January 2021

The Treatment of Advanced Persistent Threats on Windows Based Systems

Peter Bentley

PhD 2021

PETER BENTLEY

A thesis submitted to The University of Gloucestershire in accordance with the requirements of the degree of Doctor of Philosophy in the School of Computing and Engineering.

January 2021

Chapters 1-10 Word Count: 83,756

Total Word Count: 102,888

AUTHOR'S DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of the University of Gloucestershire and is original except where indicated by specific reference in the text. No part of the thesis has been submitted as part of any other academic award. The thesis has not been presented to any other education institution in the United Kingdom or overseas.

Any views expressed in the thesis are those of the author and in no way represent those of the University.

Signed: Peter Bentley Date: January 2021

doi: 10.46289/CPAQ6617

ACKNOWLEDGEMENTS

I would like to thank: The University of Gloucestershire for their support; Prof Kamal Bechkoum and Paula Thomas for their supervisory support and guidance; Emeritus Professor Clare Morris for all matters statistical; the university's Post Graduate Research Office; everyone in the university's IT Support, Library and Administration; the refectory staff and university support staff.

Finally, thanks to mum and dad.

Without you all this thesis would not have come to pass.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

GLOSSARY

ADS - Alternate Data Stream

API - Application Programming Interface

APT - Advanced Persistent Threat

ARP - Address Resolution Protocol

AV - Anti-Virus

BeEF - The Browser Exploitation Framework

C&C - Command and Control

C2 - Command and Control

C3 - Command, Control and Communications

CBC – Cipher Block Chaining

CD - Compact Disc

CIFS - Common Internet File System

CKC - Cyber Kill Chain

CLSID - Class Identifier

COM - Common Object Model Objects

COM - Component Object Module

CPD – Continuing Professional Development

CPNI - UK’s Centre for the Protection of Nation Infrastructure

CRM - Customer Relationship Management

DDOS - Distributed Denial of Service q.v. DOS

DGA - Domain Generation Algorithm

DLL - Dynamic-linked library

DNS - Domain Name System

DOS - Denial of Service (DOS) q.v. DDOS

DSL - Digital Subscriber Line

DTN - Disruption Tolerant Networking

DVD - Digital Versatile Disc

EA - Extended Attributes

EACs - Estimates-At-CompletionEPO - Entry Point Obscuring

EVFS - Encrypted Virtual File System

EWS - Microsoft Exchange

GPO - Group Policy Objects

HDD - Hard Disk Drives

HKCU - HKEY COMMON USER

HKLM - HKEY LOCAL MACHINE

HMG - Her Majesty's Government

HTRAN - HUC Packet Transmit Tool

ICAs - Independent Cost assessments

ICS - Industrial Control System

IGCEs Independent Government Cost EstimatesIoC - Index of Coincidence

IOC - Indicator of Compromise

IP - Internet Protocol or Intellectual Property

IPL - Initial Program Loader

IT - Information Technology

KB - Kilobyte

KMCS - Kernel Mode Code Signing

LCCE – Life Cycle CostLCG - Linear Congruential Generator

LMKC - Lockheed Martin Cyber Kill Chain

LSA - Local System Authority

MAC – Media Access Control

Malware - Malicious Software

MB - Megabyte

GB - Gigabyte

MBR - Master Boot record

MFT - Master File Table

NASA - National Aeronautics and Space Administration

NDIS - Network Driver Interface Specification

NIST -National Institute of Standards and Technology

NVA - National Crime Agency

P3T - Personal, Policy, Physical and Technical (Security)

PC - Personal Computer

PDB - Program Database

PE - Portable Executable

PEB - Process Environment Block (PEB)

PID - Process Identification

PoS – Point of Sale

PUA - Potentially Unwanted Application (PUA)

PUP - Potentially Unwanted Program

QC - Quality Control

RAID - Redundant Array of Independent Disks

RAT - Remote Access Tool

ROM - Rough Order of Magnitude Estimates

RDP - Remote Desktop Protocol

ROP - Return-oriented programming

RTF - Rich Text Format

SCADA - Supervisory Control and Data Acquisition

SCaN -NASA’s Space Communications and Navigation program office

SMB - Server Message Block

SQL - Structured Query Language

SSL - Secure Sockets Layer

TB - Terabyte

TCP - Transmission Control Protocol

TIB - The Thread Information Block

TLS - Transport Layer Security

Tor - The Onion Router

TTP - Tactics, Techniques and Procedures

UAC - User Account Control

UDP - User Datagram Protocol

UK - United Kingdom

UN - United Nations

URL - Uniform Resource Locator

USAF - United States Air Force

USN - Update Sequence Number

VBR - Volume Boot Record

VFS – Virtual File System

VNC - Virtual Networking Computer

VPN - Virtual Private Network

VPS - Virtual Private Servers (VPS)

VSS - Volume Shadow Copy Service

WFP - Windows Filtering Platform (WFP)

WMI - Windows Management Instrumentation

WSA - Windows Sockets API

XSS - Cross-site Scripting

ABSTRACT

Advanced Persistent Threat (APT) is the name given to individuals or groups who write malicious software (malware) and who have the intent to perform actions detrimental to the victim or the victims' organisation. This thesis investigates ways in which it is possible to treat APTs before, during and after the malware has been laid down on the victim's computer. The scope of the thesis is restricted to desktop and laptop computers with hard disk drives. APTs have different motivations for their work and this thesis is agnostic towards their origin and intent.

Anti-malware companies freely present the work of APTs in many ways but summarise mainly in the form of white papers. Individually, pieces of these works give an incomplete picture of an APT but in aggregate it is possible to construct a view of APT families and pan-APT commonalities by comparing and contrasting the work of many anti-malware companies; it as if there are alot of the pieces of a jigsaw puzzle but there is no box lid available with the complete picture. In addition, academic papers provide proof of concept attacks and observations, some of which may become used by malware writers. Gaps in, and extensions to, the public knowledge may be filled through inference, implication, interpolation and extrapolation and form the basis for this thesis.

The thesis presents a view of where APTs lie on windows-based systems. It uses this view to create and build generic views of where APTs lie on Hard Disc Drives on Windows based systems using the Lockheed Martin Cyber Kill Chain. This is then used to treat APTs on Windows based IT systems using purpose-built software in such a way that the malware is negated by. The thesis does not claim to find all malware on but it demonstrates how to increase the cost of doing business for APTs, for example by overwriting unused disc space so APTs cannot place malware there.

The software developed was able to find Indicators of Compromise on all eight Hard Disc Drives provided for analysis. Separately, from a corpus of 228 files known to be associated with malware it identified approximately two thirds as Indicators of Compromise.

“The king hath note of all that they intend,

By interception which they dream not of.”

- Bedford, Henry V, Act II Scene II

(Shakespeare, 1599, p. 13)

1 INTRODUCTION

1.1 Background and Motivation

This is a Cyber Security thesis within the Computer Science Discipline. It focuses on the deployment by Advanced Persistent Threats (APT) (Jeun, Lee and Won, 2012) of malicious software (malware) on the Windows family of operating systems on PCs and laptops which use Hard Disk Drives (HDDs). The technical analysis is framed within the business context of the user and is a file-based view.

This thesis is agnostic towards the origin and intent of APTs.

APTs attempt to install malware on a victim's machine using a variety of techniques. Cyber Security companies publish white papers (Lemay *et al.*, 2018) on the work they have performed researching the various APTs, their malware and their Tactics, Techniques and Procedures (TTP).

This thesis presents a cyber defence philosophy and associated software aligned with the Lockheed Martin Kill Chain (LMKC) (Hutchins, Clopperty and M., 2011) for the defence of Windows based operating systems against malware deployed by APTs. It presents the idea that it is not necessary to eradicate malware from the operating system; the minimum that is needed is to disrupt the mechanism of action (Lynch, 1972), i.e. operation, of the malware. The thesis also presents the idea of increasing the business costs of the attackers. APTs impose negative externalities on the victim's business; this thesis seeks to impose negative externalities on APTs.

The world is increasingly interconnected, one that is supported by computers. Many previous actions that involved face-to-face contact e.g. banking, can now be performed remotely, online, by the customer and with that has followed cyber-crime exploiting the gaps in the new technology and associated business practices. This thesis will assert that, although the fundamental problem is one of poor Quality Control (QC) in the development and deployment of hardware and software by the IT industry, APTs are still able to construct and install malware onto computers remotely even when defenders use industry standard techniques such as risk registers built on Personal, Policy, Physical and Technical (P3T) measures.

Cyber Security affects the business bottom line: it is a cost but the cost of lost data or Intellectual Property could be higher. This thesis asserts that the cost of cyber security should not be greater that the value of the assets it is protecting. Additionally, one cannot prove that a given computer is malware free and no evidence of malware does not mean there is no malware. In some ways a cyber security success can be a failure in that the success is finding malware but the failure is that malware has been found on the computer. The thesis starts with the assumption that malware is on the computer, so what can be done to treat it? The approach is supported by software which may be used to help increase the future business cost of the APT beyond the point where it is cost-effective for them to continue attacking a given company or machine.

1.2 How the Thesis was Approached

Companies that provide cyber security advice and guidance openly publish on their internet web sites white papers discussing their work. The start for the thesis was to "crank start" or "boot strap" the research by using the search term "Advanced Persistent Threat" in an internet search, selecting results from these companies. The search returned links to white papers, APT names, designators, with alternatives and further useful search terms which informed the next round of searches for an iterative approach until there were perceived diminishing returns. Similarly, academic papers are published on the subject and the search was repeated for several journals and academic websites. Later research was needed to support software development for proposed solutions.

In this thesis gaps in knowledge were identified through inference, implication, interpolation and extrapolation. The thesis was based on informed assumptions, leading to hypotheses that were tested by the software developed.

1.3 Aims and Objectives

The aim of this thesis is to identify novel approaches to counter APTs, and their malware, and construct implementable methods of protection against them while increasing the business costs of the attackers. This research will produce real-time software to protect users, systems and information from APTs and other attacks by analysing the strengths and weaknesses of current APT approaches. It also identifies novel approaches to counter APTs as well to design and test a new, implementable approach which is APT independent.

This research objectives are to:

- discover where APTs lie on windows-based systems;
- create/build generic views of where APTs lie on Windows based systems using an appropriate Cyber Kill Chain (CKC);
- treat APTs on Windows based IT systems in such a way that the malware is negated;
- increase the cost of doing business for APTs, for example by overwriting unused disc space so APTs cannot place malware there.

The thesis title uses the word "Treatment" instead of the words "delete" and "disrupt". This is a deliberate choice of words - as will be shown in Chapter Four these latter two verbs are used in the LMKC Courses of Action Matrix. A Cyber Kill Chain (CKC) formalises the APT stages of attack and the LMKC was selected from 21 CKCs as the best candidate to support this thesis. It also contains the Courses of Action Matrix which describes mitigations for each stage of the CKC (full discussion later). It is not necessary to delete all the malware. It is not even necessary or possible to locate all parts of the APT attack; it is only necessary to find just enough (necessary and sufficient) malware and malware mechanisms of action for the counter-malware technique to be successful and so may leave, what becomes, impotent malware on the computer.

1.4 Scope

The scope of this thesis is for malware on the Windows family of Operating System on the platters of a Hard Disc Drive (HDD) contextualised within the Lockheed Martin Cyber Kill Chain (LMKC). Microcode on the HDD controller is out of scope.

Furthermore, the scope of this thesis does not include the LMKC stage Reconnaissance as at this point there is no APT deployed malware on the HDD, but logs from such, as “after event evidence”, may be a useful resource when cross-checking for example IP addresses and ports in conjunction with the port scanner written for this thesis.

1.5 Why Study a Windows PC based IT System?

Malware on computers costs users in lost time and money. Such malware may be deployed by what the IT industry calls Advanced Persistent Threats (APTs), There are several operating systems for computers; one of the most popular is Windows which uses the proprietary file system NTFS (New Technology File System) (Microsoft, 2019a).

Windows is a family of computer operating systems produced by the Microsoft corporation in the US with a lineage back to 1985. Microsoft started in 1975, shipping their first operating system, MS-DOS, in 1981 (Microsoft, 2015). Market analysis (Appendix A) shows that, in January 2017, the various version of Windows had, in total, approximately 90.58% of the market share for desktop systems. In September 2019 the Windows share had declined to 86.38%.

It was observed in 2018 that eight of the top ten vulnerabilities which were exploited via phishing attacks, exploit kits or RATs (Remote Access Tools) targeted Microsoft products – one more than in 2017 – and that Adobe Flash had only one vulnerability on the top ten. This is a change from 2015 and 2016 when Adobe had the majority of the top ten places (Kuczma, 2019, p. 1). Again, in 2019, eight of the top ten vulnerabilities were for Microsoft products (Kuczma and Manalo, 2020, p. 2).

Although this thesis concentrates on secondary storage (HDD), the transition from larger devices (PCs, laptops etc.) to mobile devices for work and entertainment (Lion Gu, 2014, p. 2) is acknowledged and supported elsewhere: In May 2016 the market share for different hardware was: Desktop/Laptop 70.3%; Mobile – 24.42%; Tablet, 5.26%; Other - 0.02%. By September 2019 the market share for different hardware was: Desktop PCs/Laptop 41.84%; Mobile - 53.79%; Tablet, 4.37%; Other - 0.00% and this has been steady for at least a year (Netmarketshare, 2019b), now giving Windows on PCs over a third of the market. Mobile devices are good for "on the go" communications but not for business applications, e.g. word processing, spreadsheets, simply because it is easier to manipulate such data on these desktop PCs or laptops than on a mobile phone. It is for this reason that that laptops and desktop PCs is the primary hardware for such work. It is noted that the Windows share of the Top 500 registered fastest supercomputers in the world is now zero (Top500, 2019a). Finally, this author has more experience on Windows based systems. It is therefore a good fit.

1.6 What is New About This Piece of Research

This research has been developed from ideas that the author has had as an IT Security professional, using real-life observations of the subject as well as an extension of the author's M.Sc. dissertation (Bentley, 2008) and reading for Continuing Professional Development (CPD) purposes. The Aims and Objectives, discussed earlier in this chapter, were developed from these observations.

For the first two Aims and Objectives, there seemed to be no literature, academic or otherwise, looking at the subject of APT threats holistically using a file-based approach while understanding the generic problem of mechanisms of action. Academic and white paper seemed to be concerned with particular issues of particular malware families and particular APTs. Interesting though they are, they seemed to be nothing linking them together, drawing wider conclusion and then extending that within the Windows and other operating systems. This apparent lack of linkage extended to advice provided by the IT Security industry. This is the third point of the Aims and Objectives.

This seemingly lack of holistic view extended to CKCs and how malware was aligned to the CKC concept.

Finally, for the fourth Aim and Objective, there seemed to no consideration of how APTs were organised and resourced. They are, after all, a business just like any other (either a one-person unit or a fully resourced team) and viewing them in this way sheds light on how their knavish tricks may be thwarted.

1.7 Structure

The thesis is divided into nine chapters:

Chapter 1, this chapter, provides an overview for the thesis presenting the reasons for the thesis, aims and objective, scope and structure. It reviews the aims and objectives to reflect and identify what will be new about this piece of research.

Chapter 2 is the academic literature review and academic justification for the thesis. It discusses the investigative process for the thesis, the philosophical aspects of the research and contextualises the intellectual framework common for the research aims and the choice, design and implementation of the research method. For the academic justification it includes the literature choice and boundaries. It also includes the hypotheses that are in other chapters. All of this is then linked to the objectives There is a discussion of the legal issues, including consent, and a view of the security actioned in support of the thesis.

Chapter 3 identies gaps in the knowledge. It concludes with that idea that cyber security issues are fundamentally ones of poor Quality Control (QC) (25010:2011, 2011). It is a scene setting chapter which discusses how a HDD works and the aspects of the Windows operating relevant to the research for example: the Master Boot record (MBR), Master File Table (MFT) and Registry.

Chapter 4 opens with the historical context followed by a discussion of the international (United-Nations, 2018) and UK national (CPNI, 2018) provenance of the legitimacy of cyber defence. This is followed by a view of the size of the problem and cost to users. The motivation of APTs is discussed prior to a review of

Cyber Kill Chains (CKCs) and the reasons for selecting the Lockheed Martin Kill Chain (LMKC.)

Chapter 5 reviews academic literature on the subject and blends it with some white papers published by cyber security companies, aligning the derived knowledge with the LMKC.

Chapter 6 presents the main review of white papers aligned with the LMKC. The chapter briefly discusses stage 1 of the 7-stage LMKC before focusing on stage 2-7. The stage 1 discussion is brief as no malware has yet been delivered and hence is not on the HDD. It concludes with a review of APT errors.

Chapter 7 starts with a discussion of APT business organisation, their business costs and how these may be increased by the defender. The chapter continues with a distillation of the information uncovered into a manageable form and present a structure for the software suite developed to treat malware on the machine. It re-introduces to IT Security the concept of a "Mechanism of Action" – how the malware is actioned – and uses the concepts of the Windows operating system previously described to contextualise the malware and its actions. It is the analysis based on the literature reviews and discusses the research and the implementable methods of protection against APTs and APT disruption.

Chapter 8 presents the software developed in support of the analysis. It also introduces the concept of benware – software which acts as malware but is benign.

Chapter 9 presents the results of analysis derived from the application of the software written. It presents a hitherto publicly unknown, mechanism of action in support of the Windows Registry which could be used to action malware. It then discusses another unknown attribute, this time on the length of Windows executable files. This is followed by a philosophical presentation of an approach to cyber security which frames the software developed in support of the thesis.

The chapter demonstrates that, on every one of a variety of Hard Disk Drives (HDDs) analysed an indicator of possible malware were found. It also applies the software developed to a known corpus of malware and presents the results.

Chapter 10 is the conclusion containing a critical review of the thesis and a discussion of the future including possible lines of research.

Throughout this thesis the term Hard Disk Drive will be abbreviated to HDD. Compact Disc and Digital Versatile Disc will be referred to as CD and DVD respectively. Also, although a misnomer, the term hacking (IETF, 1993) will be used for those who develop or place malware on a computer for which they are not authorised to access. The correct term should be “cracker” but common use suggests the use of the root “hack” for the noun and verb.

A Glossary is provided before Chapter 1.

2 METHODOLOGY AND APPROACH FOR THE THESIS

2.1 Chapter Overview

This chapter presents a discussion of the steps taken in the investigative process. It includes academic justification (literature choice and boundaries) and includes the hypotheses that are in other chapters. These are then linked to the objectives.

The chapter contextualises the choice, possible design and implementation of research method(s) in relation to the author's research. It draws on relevant reading and reflects critically the author's role as a researcher. It also discusses personal security for the thesis and security for the internet searches in support of the thesis. It is an chapter that lays the groundwork for the rest of the thesis.

The chapter discusses epistemologically: how we know what we know and how we know our knowledge is complete; dealing with the data; interpreting the data; modelling and presenting the Kill Chain world; legal issues including obtaining consent; the issue of hero culture within the IT industry; and the model for the thesis. This is followed by a discussion of the philosophy of the thesis, modelling, legal issues, concluding with a model for the thesis. This supports the first two Aims and Objectives of thesis

Overall the chapter contextualises the intellectual framework common for the research aims and contextualises them in relation to key paradigms and theoretical perspectives employed within the area of study as well as providing an orientation within that study. It also discusses the rationale for choosing Windows and the possibility of bias based on Anti-virus companies' research, business and cultural agendas, conscious or otherwise. There is a dearth of literature on a file-based view of an operating system. The support Aims and Objectives three and four.

2.2 Background

To help mitigate hacking there has been a growth in the IT Security as a subset of the IT industry. This subset includes IT Security or Anti-virus (AV) companies helping to counter APTs and their malware. These AV companies produce white papers, which discuss the work of APTs. It is mainly from these papers, as well as academic research, that this literature review is based.

The title of the thesis is “The Treatment of Advanced Persistent Threats on Windows Based Systems”, with the aim being to identify novel approaches to counter APTs and construct implementable methods of protection against them. It is possible that this research will help produce real-time software to protect users, systems and information from APTs and other attacks by:

- Analysing the strengths and weaknesses of current APT approaches;
- Identifying novel approaches to counter APTs;
- Designing and testing a new, implementable approach which is independent of APT;

The fundamental purpose of the thesis is as described by Prof Sir Peter Ratcliffe:

> “We make knowledge, that's what I do as a publicly funded scientist. That knowledge has only one quality that is definable really, it's good knowledge, it's true, it's correct. It is important that scientists have the courage and are allowed to derive knowledge for its own sake, independent of the perceived value at the point of creation.”

(ABC, 2019a)

2.3 Securing and Protecting the Research Environment

This author is agnostic towards the origin and motives of APTs which is a point re-enforced by Forcepoint (Forcepoint-Security_Labs, 2016, p. 8) who strongly advise against getting distracted chasing attribution and that resources are better spent on prevention and remediation.

However, APTs may not reciprocate this view and so a reasonable amount of security should be built in to the research. Guererro-Saade (Guerrero-Saade, 2016) discusses the ethics and perils of APT research from the individual researcher’s and anti-virus company viewpoint. There is no evidence in Guererro-Saade's paper to back up what can be inferred with respect to threats against researchers and a cited forthcoming paper does not appear to be available.

Guererro-Saade does make a valid point about verification of the work of IT Security researchers however. This thesis verifies some elements of that work: by reviewing a large number of white paper and academic papers it is possible to build a view of APTs, across AV companies, noting agreements and contradictions.

A research topic of this type could draw unwanted attention to the university and the student and an APT may wish to lay down malware on the university's network. In protecting my research the security environment needed to be protected. The intent was to try to draw as little attention to the University of Gloucestershire as possible. The Head of IT for the University met with the student to discuss the research proposal, and possible threats. There was agreement that for foreign language searches the university would issue a laptop to which would be used geographically distant from the university.

On the other hand, repeated searches to company websites using more advanced and focused search techniques may give the impression of a form of "Google Dork" (Bort, 2015) – using advanced search techniques to gather information. Use was made of the "site: and "filetype:" parameters would draw attention this research.

Some AV companies require registration details for some reports. In order to remain relatively anonymous, it was decided not to register for any reports. A lot of the reports that required registration were repeated on different sites.

Interestingly, more than one pdf report from more than one AV company had a modification date and time the same as the date and time, to the minute, that it was downloaded by this author. Another company had a modification date of the pdf before the creation date. When starting the literature write-up and comparing metadata, the "original" on the internet did not have the download date as the latest modification date (Wyke, 2012b).

Although the author has full confidence in the University's IT data protection procedures, backups to personal offline storage was frequently performed during thesis and software development with backups being done on two separate USB memory sticks that were only ever used on university machines. The date and time of each backup, along with which memory was being used, was recorded. Complete loss of research data would no longer support any of the Aims and Objectives.

Technical security was achieved by using firewalls, AV software, Icognito or InPrivate browsing. Many tracking websites (Al-Fannah, Li and Mitchell, 2018) were also blocked.

No security is perfect but for a small effort on this author's part potential attackers would be put to a great deal of work. i.e. increasing the cost of their business which is one of the research objectives.

Every website which potentially contained white papers was checked with using a "Whois" website for provenance. So much use was made of different websites that some websites blocked this researcher. A web search checked further details of the website. In some cases, the use of Google Maps facilitated the checking of addresses returned by "Whois" and the country of origin noted. The laptop loaned by the university facilitated checks of non-English language websites. However, in some cases the registrations of some websites were hidden behind privacy registration providers.

There were at some "interesting" companies: the first one was noted as having a link to a foreign white paper. Initial checks indicted no problems but as the "whois" address was an obvious UK home address. A further check with Google maps returned a blurred out image what was possibly the house at the correct address; the second company "whois" had a US address but four of the six phone and fax contact numbers were all 9s. Google maps had a medium sized building with a "For Lease" banner outside but with some lights on (it is noted that Google maps is not a live feed); other "companies" were shared offices, warehouses, or industrial premises. One website registrant had a mailing address in a commercial mail centre. A further search on the address found a US government department webpage containing a copy of a “cease and desist” letter telling them to stop unauthorised sale of a certain medical testing kit in the US. The registrant’s website was not accessed for this thesis.

It is clear that a number of fake registrations were uncovered as well as possible illegal activity.

A number of searches of companies’ websites were done with a laptop away from the university’s campus using Wi-Fi that was for public use. These searches and subsequent downloading of relevant information were from companies where it was judged that UK academic interest may raise warning flags.

Some mistakes were made by this researcher but it is thought that in general good IT Security practices were upheld.

2.4 The Thesis Structure

This thesis is built on the Opening Out Model selected from one of Dunleavy’s thesis models (Dunleavy, 2003, pp. 51-61). This provides a wide amount of room to discuss, develop and present original research. At thesis inception the literature review was expected to be large but this author had a good idea of the direction of travel. This has come from working in the IT industry and continuous professional development including reading academic and white papers. This is not to say that this knowledge is complete; there was no intention to put the cart before the horse and this thesis is evidence driven.

Dunleavy (Dunleavy, 2003, pp. 51-61) suggests splitting an 80,000-word thesis into 8, roughly equal, 10,000 word chapters: the first two being Lead-in materials; the next five core subject; and the last being Lead-out materials. Dunleavy also suggests having a Research Methods Appendix. Dunleavy goes to say that a thesis should convey the PhD candidate’s original work and examiners may wish to see this original work in the thesis as soon as reasonably possible. A discussion on Research Methods is a discussion on a supporting function. The examiners need to see evidence that there has been rigour to the research, and that the research has not been random (q.v. Hume Section 2.8), but they do not need to wade through 20,000 words until they come to the candidate's original contribution.

Although not a social scientific thesis it considers Harrington who concludes that:

> " … social theory is the study of ways of thinking about society scientifically. Further, we have seen that it is the discipline about how far it is *possible* for society to be studied scientifically."
> (Harrington, 2005, p. 12)

Harington inherently states that there are limits to the extent of studying society scientifically. It is possible to study people (or society) in s statistical manner but not to apply results as self-evident truths.

Hammersley & Atkinson suggest that there are at least two textual approaches: thematic and chronological (Hammersley and Atkinson, 2009, p. 194). As is demonstrated this thesis follows the LMKC which will follow both approaches. APT attacks must follow a path of cause and effect which this thesis will describe as "Mechanisms of Action". There may be external probing of a victim's computer system, followed by laying down of the malware. This must be done chronologically – one cannot deliver the malware, manually or automatically, before scoping the system on which it is to work. Even automatically delivered malware (speculative or opportunist attacks) will not run if there is not software compatibility and with that compatibility is an inherent scoping. A CKC follows a logical chronological progression. However, within the chronological deployment of malware there may be themes to explore e.g. drive-by infections, PKI certificate fraud. This may be illustrated thus:

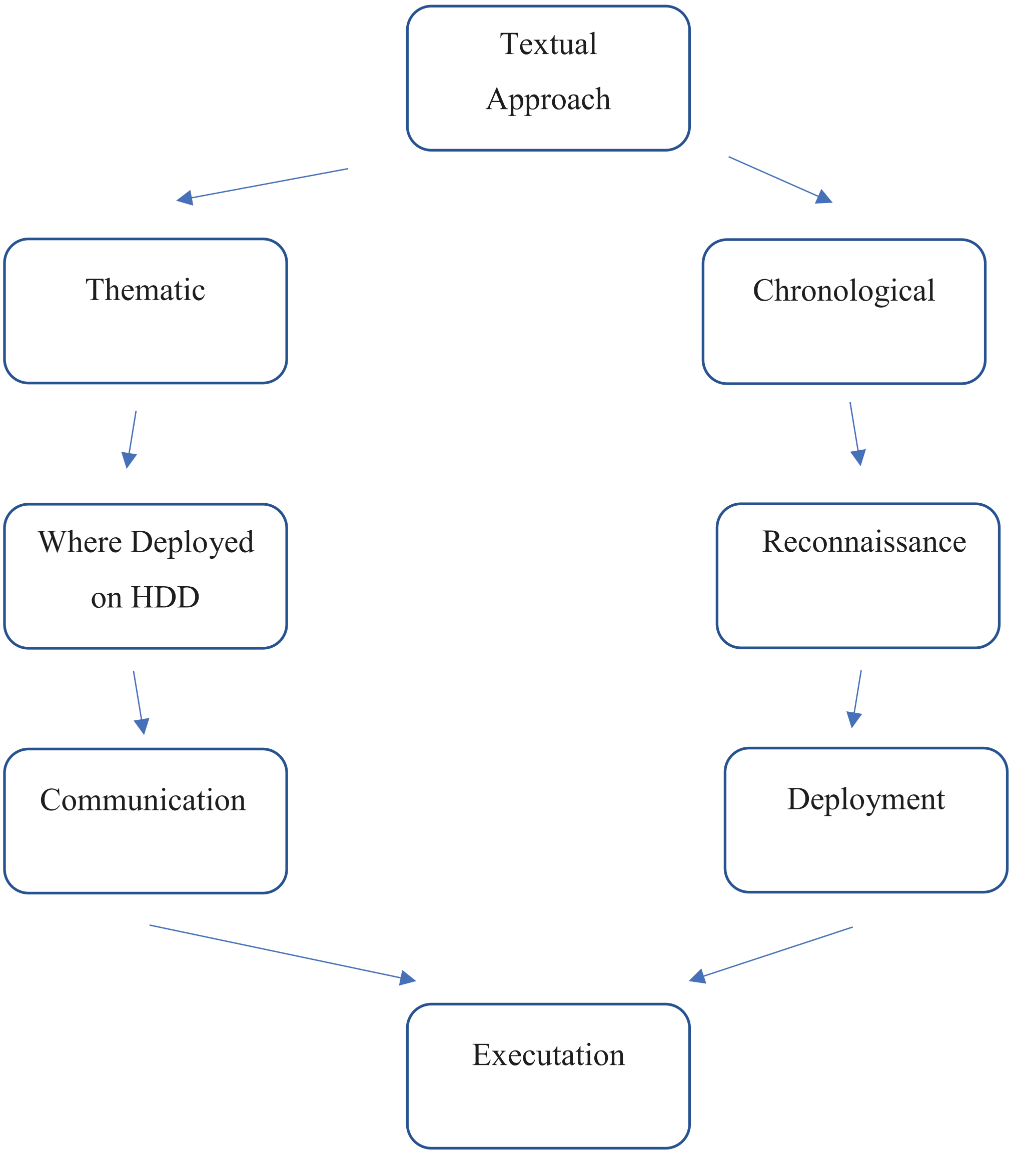


Figure 2-1: Textual Approach Representation

Layder (Layder, 1998, pp. 100-127) states that theorizing involves the ability to ask questions and come up with answers; think analytically and conceptually; and the ability to move from the concrete and particular to more general and abstract concerns and ideas. These concepts and background concepts allow this author to "crank start" the research. It is from the specifics of academic and white paper documentation that conceptual and general themes for the thesis are developed. These themes also give rise other ideas developed for this thesis which are explored e.g. Attributes by Design; Attributes by Discovery and Mechanisms of Action. This supports the first Aim and Objective of the thesis.

2.5 The Model for the Thesis

In support of the first Aim and Objective openly available literature is reviewed and the thesis is guided using the Lockheed Martin APT Kill Chain. The research uses Mixed Method Approaches of Quantitative Methods, Qualitative Methods and Discipline Driven (cyber industry) Methods. Given that academic papers and white papers are both used as an evidential base there also are Concurrent Quantitative Methods (Saunders, Lewis and Thornhill, 2016, p. 170)

It will be asserted in the next section, that malware exploits poor IT (QC) and the research will assume that malware has exploited this poor QC and is on the system. AV vendors produce freely available non-proprietary information on APTs in the form of white papers, blogs etc. From this information it is possible to construct a view of APT families and pan-APT commonalities. The ethical hacking community produces similar information. In addition, academic papers provide proof of concept attacks and observations, some of which may become used by malware writers.

Gaps in, and extensions to, the public knowledge may be filled through inference, implication, interpolation and extrapolation. This supporst the second Aim and Objective.

The author's professional model is remarkably similar to the 36-Cell Sherwood Applied Business Security Architecture (SABSA) Matrix (Sherwood, Clark and Lynas, 2005) except the author has three conceptual layers (Strategic, Tactical, Implementation as well as six practical considerations for each layer: what, who, where, when, why, and how; with personal actions in the cells rather than business functions.

As stated, the initial reading is from academic papers and AV companies' white papers and. The former is, obviously, material from academia; the latter describes what has been observed, perhaps leaving out some details for their own IP reasons. The analysis leading to the latter may even miss salient points. What the writer meant and what the reader reads may, in some cases, lead to an incorrect view as to what really happened. The problem for inference, implication, interpolation and extrapolation is that this author will have to make intelligent assumptions, even guesses, leading to hypotheses, as to what may be in the gaps and then test these hypotheses. Iser in Lodge and Wood understands this phenomenological approach (Lodge, 2000). Although written for fiction, Iser points out that the two poles (artistic and aesthetic) are linked by what was meant by the author and what was realised by the reader and that there are gaps for the reader to fill in.

2.6 A Question of Poor Quality Control?

This thesis infers that there is an acknowledgement at the international level of a lack of good software QC as evidenced by the International Standards Organization (25010:2011, 2011). QC also suggests system design and code audit. This thesis suggests that if there were good IT QC then such a standard might not be needed. With simple systems QC may not be required but the complexity and interconnectedness of computer systems has produced poor outcomes. The advent of the internet as a business tool has seen the development of a world wide web of interconnected computers supported by associated software but with this there are inherent problems, most of which relate to design and implementation of software. This thesis asserts that the technical and business design of a computer system to support a business should be that which is necessary and sufficient to perform the business function it is required to do but, in many cases is not. As shall be demonstrated in this thesis, hacking exploits this poor hardware and software architectural design and subsequent implementation, as well as manipulating business practices.

To help mitigate hacking there has been a growth in the IT or Cyber Security as a subset of the IT industry. This subset includes Anti-virus (AV) companies helping to counter APTs. These AV companies produce white papers, which discuss APTs “Tactics, Techniques and Procedures” (TTP) (J. N. Stewart, 2014). TTP is a term used in IT Security and can be used to discuss both an attacker’s and defender’s modus operandi. It is mainly from these papers, as well as academic research, that this literature review is based. White papers are explored in more detail later in Chapters Three to Six.

Such lack of QC is highlighted by Little who quotes Perrow:

"In his book, *Normal Accidents*, Charles Perrow described numerous failures of tightly coupled, complex systems.* In the search for speed, volume, efficiency, and the ability to operate in hostile environments, he maintains, ***we have neglected the kind of system designs that provide reliability and security*** (***This thesis author's emphasis***). A particularly troubling characteristic of these tightly coupled, complex systems is that they predictably fail but in unpredictable ways. Similar chains of events do not always produce the same phenomena, but system-level or "normal" accidents of major consequence continuously recur."

*These occur where the systems involved are sufficiently complex to allow unexpected interactions of failures to occur such that safety systems are defeated, and sufficiently tightly coupled to allow a cascade of increasingly serious failures ending in disaster.

(Little, 2002)

This thesis asserts that it is this neglect of reliability and security that have allowed APTs to flourish and which will not be fully resolved (which is consistent with no resolution) until Perez's (2002:a) MATURITY stage of IT technological development has been passed. (Perez, 2002, p. 74).

Perrow (Perrow, 1999, pp. 3-5) suggests that if a system is complex and tightly coupled then it does not matter how effective conventional safety devices are: there is a form of accident that is inevitable, which is not good should the system have catastrophic potential. Clearly an operating system is complex and coupled (but perhaps not tightly from a security point of view otherwise there would be no gaps for APTs to exploit). It also has catastrophic potential as is evidenced by APTs actions on objectives Q.v. LMKC.

However, these "inevitable accidents" are not welcomed by Phillip Davies who presents a criticism of the IT industry's assertion that self-driving cars will learn over time:

"This is the mentality of computer programmers who have released defective software for their entire careers: let the customer find out the problems and then charge them for the fixes. It is not a proper mentality of someone who is building machines that will kill people."

(NPR, 2018)

This also supports the QC argument and SANS (SANS, 2011), with their list of the Top 25 programming errors. For there to be a Top 25 the same errors must be being repeated in different systems by different designers and implementers programmers).

The errors are not just technical: the manipulation of business practices is not central to this thesis but, briefly, in order to avoid the manipulation of business practices there should be data checking and verification; not just for, for example, alphabetic data in an expected numeric field but also as a business sanity check e.g. why is this person 143 years old?

It is therefore inferred that the technical and business design of IT systems suffers from poor QC which includes a lack of, or sub-standard, auditing of code and aggregated systems. In many cases hacking is facilitated by exploiting poor hardware and software architectural design and implementation as well as by manipulating business practices. This is a question of Quality Control.

2.7 Ontological Discussion

Conceptually, IT Security is like the Plato's Parable of the Cave (Cornford, 1945): Prisoners in a cave see shadows on the cave wall of events in the real world and they have to make sense of them. Unlike Plato's prisoners who see time bounded shadows (there is no artefact providing proof of a previous existence), APTs may leave artefacts of attacks victims' computers. These artefacts are elements which APTs need, or needed, to progress their attack and which are the outcomes of poor QC – something from which APTs are not immune. In IT Security the Plato's shadows are manifestations or artefacts of actions of APTs, part of which are made by malware developers and deployers. Some artefacts may be pieces of software left over from a completed attack but not completely erased. Again, a QC issue.

In looking for malware on a piece of IT equipment one is searching for causality. It is easy to see patterns in data when none exists. Indeed, malware may be designed to be hidden or obscured; to blend into the operating system and associated software.

Hume argues that there are three conditions for causal inference: contiguity, succession; and necessary connexion (sic) (Hume, MDCCXXXIX). This is elaborated on by Teddlie et al as:

- Physical contiguity between the presumed cause and effect;
- Temporal precedence (the cause has to preceded the effect in time);
- Constant conjunction such that the cause has to be present when the effect if obtained;

(Teddlie and Johnson, 2009)

It may be added that the cause and effect should be necessary and sufficient conditions i.e. the cause only influences the effect and the effect is brought about by the cause. This reduction of analysis to the easiest describable event is covered by Ockham's Razor. The "Principle of Parsimony" of Ockham's Razor should be observed as described by Epstein:

> "Ockham proposed a rule of logic which has come to be called "Ockham's Razor." He stated it variously: "Plurality is not to be posited without necessity" *(Plumlitas non est ponenda sine necessitae)* Or "What can be explained by the assumption of fewer things is vainly explained by the assumption of more things" *(Frustra fit per plum quod potest fieri per pauciora)."*
>
> "Where we have no reason to do otherwise and where two theories account for the same facts, we should prefer the one which is briefer, which makes assumptions with which we can easily dispense, which refers to observables, and which has the greatest possible generality"

(Epstein, 1984)

A model may thus be constructed. A generalisation from repetitive observations may be constructed but suggesting, as Popper does (Popper, 1968, pp. 249-250), that just because something has always happened it does not mean it always will. With malware, certain actions or outcomes may be observed but there is a chance that the APT operators have chosen to not yet attack using a different combination of inputs which would give rise to other outputs. Therefore, existence of the same malware may not always mean the same outcome. Additionally, operating systems change over time and the vulnerabilities that were once present may not later be. New vulnerabilities may also come to light.

2.8 Research Orientation

The research orientation is not phenomenological. The author of this thesis has a mathematical background and considers himself a positivist. However, there is a danger of applying the science to people and Comte warns against an overly mathematical approach in Simpson:

> "The most perfect methods may, however, be rendered deceptive by misuse and this we must bear in mind."

(Simpson, 1982)

This has its extremes in Taylor-ist management (Taylor, 1919) or political thought where people may be seen as numbers. Halfpenny in McNamee asserts that:

> "First, positivism refers to a theory of historical development in which the growth of knowledge contributes to the development of progress and social stability."

(McNamee, 2005, p. 3)

Crotty is somewhat dismissive towards positivism:

> "To say that 'A' is 'A', or that 'not~A' is not 'A' is hardly an almighty contribution to human knowledge. Logical positivists would agree. Analytic propositions are either tautologies or contradictions. Nothing more, nothing less. On this accounting, logic and mathematics are merely formal in character. They are quite

empty of factual content.”

(Crotty, 1998, p. 25)

Crotty is just plain wrong: it is an almighty contribution to human knowledge full of factual content. These are fundamental statements in a branch of Mathematics called Logic. Logic is the basis for Boolean Algebra which itself is part of the subset of Number Theory called ”Groups, Rings and Fields”. It is the basis of arithmetic. Without any of this computer would not work and more importantly would not be proven to work. As computers are ubiquitous in modern society, modern society would unravel very quickly. One does equal one and one plus one does equal two. Fact.

2.9 Problem Approach

As a natural scientist, this author has an epistemological position of a logical positivist that will support ontological claims. The thesis will not be solely based on academic and white papers - it is a consilience thesis. This supports the second Aim and Objective.

This work is an extension of this author's MSc in IT Security dissertation titled “Secure Erasure” (Bentley, 2008) and considers how to identify and safely (from the viewpoint of the OS) affect the mechanism of action for malware and increase the APTs business costs. It is aligned with the Lockheed Martin Cyber Kill Chain (Hutchins, Clopperty and M., 2011) which, as is demonstrated in Chapter Four, is selected from a number of Kill Chains.

This approach aligns with the assertion of Popper who discusses the use of deductive reasoning to create hypothesis which may then be tested (Popper, 1968, pp. 27-34). This thesis will also be based on inductive and deductive reasoning to search for individual pieces of malware.

However, there may be a Problem of Demarcation as described by Popper:

“ … my main reason for rejecting inductive logic is that *it does not provide a suitable distinguishing mark* of the empirical, non-metaphysical, character of a theoretical system; or in other words that *it does not provide a suitable 'criterion of demarcation'*". (Popper, 1968, p. 34).

Popper goes on to discuss falsifiability criteria (Popper, 1968, pp. 86 - 91). A theory is falsifiable if it divides the class of basic statement into two: those statements which are potential falsifiers of the theory and those which do not contradict (or permit) the theory. Falsification of the theory occurs when basic statements that contradict the theory are accepted. This helps us scope the problem.

The Problem of Demarcation for this thesis is solved thus: The theoretical system is bounded by the physical system and the users of that system. The physical universe (Demarcation limit) of our work is platters of the HDD. Any malware must be located, and locatable, on the HDD and only on the HDD. Malware may manifest itself as such; for example, an executable file (.exe) may be "hidden" partly by renaming with another suffix (Clearsky, 2015, p. 15) or it may be hidden from the user by use of the Windows Hidden files option (Microsoft, 2017aj). Inductive logic is used as, although malware may be present, not all types and variations of malware may be present. There may even be cases where no malware is located because no malware is present. Deductive logic is used to follow chains of event and mechanisms of action.

The physical system is initially demarcated by technology. The selection of HDDs as the physical medium is deliberate. Computers may have HDDs or a solid-state disk drive (SSD) as secondary storage. As shall be shown later in the thesis, a SSD presents itself to the Windows OS as a HDD and Windows acts in the same manner for both HDDs and SSDs but SSDs automatically delete data whereas HDDs do not – when a user deletes data the area it uses is simply marked as free by the operating system. However, the market for such secondary storage devices is changing. Based on shipments data, in 2021 the number of SSDs (360 million) will exceed the number of HDDs (330 million) (statista, 2019c).This differs from 2015 where the number of SSDs shipped was less than a quarter of the number of HDDs shipped (470 million viz 105 million). This data tallies well, where it overlaps, with the HDD quarterly shipment data (statista, 2019a).

Demarcation also extends to the use of the data on a computer. A computer system and the information it contains should only be available to the legitimate users, for agreed business reasons, in a timely manner. Not all parts of the systems and contained information should be available to all users of the business. An organisation's requirements to control information should be documented by that organisation (ISO, 2013). This thesis asserts that malware writers and deployers are not legitimate users but they fall within the demarcation lines of this thesis. This supports all four Aims and Objectives.

A Cyber Security professional legitimately installing software on a system to investigate a possible intrusion is a legitimate user. Therefore, any analysis software is within the demarcation limit.

The intent of the thesis is to stop all non-legitimate users, and software from running, from using the system under investigation. This thesis does not claim 100% eradication of malware but proposes that the ideas developed provide a framework to increase the business costs of the malware writers and deployers (collectively APTs), to a point where it is prohibitively expensive for the APTs to continue. Should these costs outweigh the APTs perceived gains then the APTs may cease the attack and move to a potential victim where their gains outweigh their costs.

This thesis deliberately includes multiple references to support the evidence. Independent evidence for the same assertion provides greater support to that assertion. It some place, for example the "Run" registry sub-key, multiple references are included to demonstrate how many APTs use that particular element of the Windows operating system to support their attack.

Many subsections start with direct quotes from Microsoft documentation. This is to allow the reader, who may not have complete knowledge of that subject, to gain an understanding.

Ellerton (ABC, 2019c) suggests that critical thinking is not analytic thinking; it is not just sophisticated or difficult thinking; it more than that. It is not just being intelligent to whatever extent that means. It is about metacognitive – aware of some aspects of our own thinking and there– some intentionality behind that and can understand how one arrives at certain decision and are aware of the inferential chains or pathways that have been followed to come to the decisions. The second aspect is evaluative – to make our thinking an object of study. Was that the best way to go? Am I thinking in a way to get the best answer regularly? Nature of argumentation. What make for a good argument? The needs to apply a set of values e.g. accuracy, precision, clarity, significance, relevance, simplicity, reproducibility, coherence which overlaps with Kuhn's model of Accurate, Consistent, Broad Scope, Simple, Fruitfulness (Kuhn, 1977, pp. 321-322). The attribute of simplicity has already been considered earlier prior to thesis development in the form of Ockham's Razor. Ellerton also asserts that people think that one thing is more important than truth and that is coherence. If their world view is coherent - they have theories about how and why people act in the way they do and that is consistent with the way that they interpret the world - then that feels true. People mistake a sense of coherence for truth.

In a broadly similar way, Murdoch, who almost certainly predates Ellerton, asserts that

"Knowledge (language) is essentially related to morality by the idea of truth. Science too depends on truthfulness. ….. We, as individuals, live in different worlds, we *see* (visual metaphor) different things, not just in general but down to the last details."

(Murdoch, 1992, p. 474).

Although Murdoch is more concerned with morals, the idea of people having different views of the world, it is the link with truth that is of concern to this thesis. Looking to Plato's shadows on the cave wall one can imagine prisoners creating a view of the world which the shadows create and having a coherence even though this may not be a truth.

Derrida, as summarised by Newheiser (ABC, 2019b), says that there is an abyss of meaning; people complain that post-modernism leaves us in the abyss: uncertainly is destabilising on an emotional, effective, level and so people often want to assert a certainty that they do not possess in order to make life seem more liveable which, in Derrida's view, provides a false comfort but that is an hopeful resilience that acknowledges that the abyss is there and that people do not possess certainty but it does overlap with Ellerton and the concept of coherence and truth: coherence for truth; assertation of a certainty as a truth.

Postmodernism may offer the concept that there is no reliable knowledge but this inherently rejects the concept of scientific, and therefore provable, knowledge. This falls into the same trap as Crotty whose ideas are discussed earlier in this chapter.

It is the intent of this thesis to reconcile the above set of values with the truth and ensure that the thesis world's coherence overlays identically with the truth.

Anti-malware techniques lean on the concept of Indicators of Compromise. These are unique indicators which are a form of inculpatory evidence i.e. they point to there being malware but may not, in themselves, uniquely identify malware. They point to a truth.

This thesis is agnostic towards the origin and intent of APTs. A research topic of this nature could draw unwanted attention to the student and the university and hence an APT may wish to lay down malware on the university's network. The intent is to try to draw as little attention to the University of Gloucestershire for as long as possible. In order to mitigate any threat a meeting was held with the Head of IT for the University to discuss the research proposal, and possible threats. It was agreed that, for foreign language searches, a laptop would be issued which would be used away from the university.

This is a sensible approach as some internet users try to give the impression that they are based in another country, to either provide themselves with cachet or obfuscate their true origins. For example, the Financial Times states that a Turkish start-up based in Istanbul has offices registered in US and other offices outside of US. This is because there is a cachet by having a registered office in Silicon Valley or in this case San Francisco (Financial-Times, 2017b).

In order to support the thesis, multiple downloads from individual sites were to be needed and may attract attention. The hope was to be unobtrusive simply by being a reader of documents. Initially, there was a desire to perform non-English language searches but the scope had to be restricted (the clean university laptop was still needed). Such an action may also have drawn attention to the work and the university; and gatekeepers, IT passive or otherwise, would have to have been employed. In dealing with this the views of the university's IT Security department were sought. A lab development machine was sought, and obtained, on which downloaded software could run. In addition, a personal development machine was used. No software was downloaded nor run, on an openly used, university library based, machine

A search was performed using the keywords “Advanced Persistent Threat” on all available English language thesis repositories (University-of-Gloucestershire, 2018). Fortunately only one thesis of relevance was found and this was from the Athens University of Economics and Business, in English (Βιρβίλης-Κολλητήρης, 2015). Although it followed a similar research path to this thesis the software outcomes were completely different those of this thesis.

The hope is that, after the end of the research process and following publication of the PhD, there may be a tertiary elaboration, co-opting the general theory and that this researcher or others will take it forward either in an academic or a business context.

The research started from the zero-base described above with an internet search of “Advanced Persistent Threat”. This search yielded websites containing such information. The names of these websites were collected and then for each site searches were performed looking for .pdf files which contained the words or string “APT”, “Advanced persistent Threat” or “APT White Paper” (all white papers were thought to be .pdf). This information was stored in folders for the respective websites. These white papers yielded references to more white papers and websites which were collected and collated as well as names of APT campaigns and types of attacks which in turn were searched for. This provided an iterative method: as more websites were found, more APT campaigns names were found etc. Approximately

250 academic papers were reviewed as well as “numerous” websites and other less formal sources of information. Over 1100 white papers were downloaded from over 200 webpages visited and stored on the author’s university account in company specific folders. Using the Pareto principle to analyse the white papers from AV firms, the Concentration Ratio for the top 10 company white paper downloads account for just over 53% of the white papers while the top 20 account for just under 70% of the same.

McNamee asserts that:

“Fragmentation of the process is now essential to much modern science. … I have often met PhD. Students who have already completed one of more experiments but performed no literature review.”
(McNamee, 2005)

The need to review literature is to gain a view of what is already, or not already, known is appreciated in this thesis (and needed for this thesis) but fragmentation of the research meant that was testing of parts of an emerging hypothesis in parallel with the literature review.

It is recognised that bias and confirmation bias may be a problem: the white paper analysis was started with the company with the highest number of white paper downloads. This may anchor the analysis in a company specific view. Other approaches which could have been used include: build up a view from companies with fewer downloaded articles; review articles in a random order; start with the most recently published white paper and work chronologically in reverse; or start with oldest first. No reason could be argued for one over the other so descending white paper count by company was used.

As previously discussed, different AV companies work on the same sets of malware on different victims’ networks and machines. This may introduce conformation bias in the analysis: it reports the same attack more than once. It is unclear if different AV company white papers are reporting the same attacker and/or event or the reports are of different attackers and/or events.

Verizon's research, which they assert is consistent with other similar research is that:

"Consistent with some other recent vendor reports, we found that 70 to 90% (depending on the source and organization) of malware samples are unique to a single organization.

We use "unique" here from a signature/hash perspective;…"

(Verizon, 2015, p. 22).

However O'Connor asserts that signature development as a form of anti-malware is not sufficient (O'Connor, 2014, p. 17)

The question of bias is discussed in Yeadon who quotes Hay discussing long jump athletes:

"Hay (1987) cautions against statistical analysis in which several jumps by each of a number of athletes are treated as if they were single trials performed by different athletes, since trails by the same athlete cannot be regarded as independent trials." (Yeadon, 2005)

This is a salutary warning against reading too much into a specific AV company's set of white papers; AV companies have products to sell and the information provided in white papers, while true, may be selective information to direct the reader's thinking into buying their product. There may also be unintentional company bias promulgated by the culture, direction of research, writer's personal views or other reasons. It is therefore prudent to think about how to organise the analysis of white papers to produce evidence that is as unbiased as possible. This supports the second Aim and Objective.

For the various reasons given above the thesis may seem over referenced. Some references in support of observation noted in the thesis may refer to the same APT or even attack.

The author reviewed white papers in sets of five years, starting from the present and working backwards. This has the advantage of ensuring that the research is current and it also allows a view on which pieces of malware are being recycled as highlighted by Dela Paz (Dela Paz, 2012, p. 1) who notes that the number of targeted campaigns continue to increase the techniques some of the attacks have existed for several years..

There is industrialization of "hacking" with there being evidence of APTs adapting to environments and using change control (Fireeye, 2015a, pp. 5-7) with prioritisation (Fireeye, 2015a)

Initially there was intent to perform searches not just in English but other languages e.g. Russian, Chinese to gain a non-Anglo-Saxon view of APTs. However, with the abundance of reports to read written in English it became apparent that including non-English language papers, and the associated cost ion time of translation, would be too much. One wonders what the Lusophone view of APTs might be.

The literature search started from a zero base with an internet search of "Advanced Persistent Threat". This yielded websites containing such information. A record was kept of the names of these websites and the number of papers retrieved. As all of the whitepapers were in .pdf format, the search was modified to look for .pdf files, which contained the words or string "APT", "Advanced persistent Threat", "White Paper" or "malware" and the whitepapers downloaded and stored in the appropriate folder. These primary searches also highlighted additional websites as well as names of APT campaigns, which in turn provided starting points for more searches. This provided an iterative method: the discovery of more websites led to more APT campaign names, which led to more websites etc.

In total almost 200 AV company websites were accessed, of which over 130 had at least one whitepaper referenced in this thesis. Over 1150 whitepapers or similar were downloaded and reviewed as were approximately 300 academic papers. The number of webpages accessed was not recorded

The majority of results were analytic papers describing various APTs but some results were product or publicity brochures. The problem of "Information Tautology" had been addressed earlier in this thesis.

Another problem is that of knowing what has been done before. As has been stated, during the academic research part of the work the author came across another thesis which is similar to this one (Βιρβίλης-Κολλητήρης, 2015). The approach is at a more macro level and the outcomes (software written) are different but it demonstrates how easy it is to do overlapping research.

Once the search was complete and the information reviewed it was decided to halt any further reading. However, later in the research it was decided to review documents that had been subsequently written. This was done about three-quarters of the way through the research and revealed papers which overlapped with the reading performed (Lemay *et al.*, 2018).

A world-wide categorisation of PhD thesis would be helpful but a repository of commercial company white papers was found (CyberMonitor, 2019).

Should there be subtle disruption of an APT attack the malware operators may not work out what is really going on. As pointed out in Futility Closet (Futility-Closet, 2017) with the psychology of deception: one cannot get to someone to believe something they are not already disposed to believe but can play on their hopes and fears to persuade them to believe something they want to believe. Deception is a course of action from the Lockheed LMKC. It is not the intent of this thesis to antagonise APTs (as previously stated, this thesis is agnostic towards the origin and intent of APTs) but simply to treat their operations so that they are not successful. They, and their malware, are unwanted visitors.

The University of Gloucestershire has an ethics policy (Univeristy-of-Gloucestershire, 2019) which was followed. Grix, quoting Punch sums up the main areas in which ethical issues arise in research: Harm; Consent; Deception; Privacy; Confidentiality.(Grix, 2010).

Within this thesis, all of the research reading is from open literature which by its nature is freely available and so there is neither harm done nor consent needed. Again, as the open literature is freely available, there are no deception, privacy or confidentiality issue. Software development to implement ideas produced was on a University of Gloucestershire owned or a personally owned machine. No interviews were undertaken and the thesis comes from standalone work. There is nothing unethical about the treatment of software that has been illegally placed on a computer owned by the organisation, or person, by those legitimately performing that treatment.

Any issues that could have arisen would have been worked through with supervisors and the University of Gloucestershire’s Legal School.

2.10 Discovery – How will we know that we know?

Supporting the second Aim and Objective King and Horrocks (King and Horrocks, 2010) discuss epistemology as what we can know and what we might want to know. However, knowledge of knowledge, or lack of it, goes beyond this and into the concept of known knowns, known unknowns, unknown knowns, and unknown unknown, which is an extension of the Johari window (USC, 2003).

This Johari window view casts doubt on King and Horrocks' assertion that Ontology may be classed as Realist or Relativist. This thesis asserts that both exist in parallel and cannot be decoupled. The real world does exist (Realist) and in some cases may be measured (e.g. temperatures, pressure) but the real world is also unstructured and diverse as evidenced by people and other living organisms which populate it (Relativist). It is not an ontological binary choice. For example, the actions of any one organism may be statistically predictable but interactions within the whole make predictions impossible (forecasts less so). In a similar manner attributes of, and interactions between, software in an operating system may have unknown and undesired consequences. This theme is developed in the section which introduces “Attributes by Discovery”.

The extension of the Johari window means that the epistemological definition of what we can know and what we what to know is blunted by not knowing what we do not know. In taking into consideration the Jorari window this thesis considers the full universe of knowledge. This is later condensed into what is known to the Windows operating system and what is not and is a key part to the thesis of malware on the HDD.

There was almost no quantitative data from which to work, being mainly a review of published academic and commercial works on malware and APTs. The thesis uses these works to explore ideas (some of which have already been generated post-Master's degree) and interpolate and extrapolate to produce a view of IT Security that has not yet been explored. There were no interviews as the data will come from white papers, academic papers, other open sources, research and technical analysis. It could have been possible as the PhD evolved, and the thesis became known, that interviews may have been later performed. The research will not end with this thesis.

This thesis asserts that extensions to the public knowledge may be filled through inference, implication, interpolation and extrapolation. Inherently, interpolation and extrapolation make use of implication and inference. Logical extension allows a final point to be concluded or argued. Austin describes the leading to this point as expositional performatives (Austin, 1975, p. 85). This thesis interprets Austin's work to mean that a logical conclusion must be attained from supporting evidence. Assumption does not pass Austin's four tests (Does it make sense? Could the action be performed without uttering the normative? Could it be done deliberately? Could it be literally false?) (Austin, 1975, pp. 83-84) and so this thesis must be careful to support every assertion.

This author of this thesis has a scientific background and this is a STEM subject thesis so the stance is one of pragmatism within the 5Ps (Paradigms, Pragmatism, Praxis, Proficiency, Publishing), as described by Cameron (Cameron, 2011, pp. 96-108). Discovery is by reading published work of others who may not be familiar with the 5Ps. This thesis asserts that researchers and authors do not need to be cognisant of epistemological positions and do things "naturally". Recent moves from monolithic programmes have led IT projects to the Agile philosophy; this means that people are encouraged to fail quickly and move on. This is an extension of the idea that one cannot be a successful entrepreneur until one has had commercial failures and learned from them. Such pragmatism is applied during the research thought not always successfully!

A research agenda for what is left to be known as mentioned by Financial Times (Financial-Times, 2016) could be drawn up. In personal investment, some people are unable to imagine their “Future Self” (Frydman and Camerer, 2016, pp. 661-675). This author has no problems in imagining a Future Self with possession of a PhD, perhaps leading a start-up company which has been incubated locally or a transition into academia.

McHoul in Balnaves and Caputi discusses first and second order interpretations of data. Natural scientists deal with first order data (atoms, planets, cyclones) whereas people (objects of knowledge have already interpreted the data before the social scientist arrives (Balnaves, 2001, p. 5). This is a valid point; with interviews, people are generally offering subjective data whereas scientific measurement of natural events is generally objective. The analysis of data in this thesis will, in the first instance, be people's interpretations of APT attack outcomes but, as has been discussed earlier, may have biases and information tautologies. The data which has been analysed is largely scientific but the researchers' analysis may also include personal bias and personal interpretation.

It is sad to see this first chapter of Balnaves and Caputi apologetically defending statistical analysis, even when published statistics was mature; it is even more so now with the arrival of the fashionable name "Big Data" (HMG, 2014)**.**

This thesis deals with few, if any, quantitative data but the quantitative data that has been analysed will, as has been discussed, have already been interpreted in some manner. Research is not binary. This thesis will also be dealing with gaps in the data: some AV companies may see a part of the picture; they may also wish to withhold some information for business reasons. This thesis will interpolate and extrapolate from published information as well as try to fit together the pictures interpreted from different AV companies.

2.11 The Personal Position

Supporting the first Aim and Objective Hart discusses literature searches and personal transferable skills (Hart, 2001, pp. 2-21). This author has successfully completed two master's degrees where referencing was not an issue. The skills developed in these degrees plus a background in IT means that a transition to a PhD may not be as formidable as it might otherwise be. However, it is clear from Hart that the fountain of knowledge repositories is wide, deep and broad. The thesis will depend on a lot of research.

Once found, Hart suggests subdividing the literature into categories: Rationale; Research design; Findings; Discussion; Conclusions; Recommendations. This thesis is aligned to an Intrusion Kill Chain - the Lockheed Martin Intrusion Kill Chain – which was been selected from a candidate group of Kill Chain. The list, below, are the seven stages of the Lockheed Martin Intrusion Kill Chain; information in parenthesis are practical applications of malware:

- Reconnaissance (strategy; infrastructure);
- Weaponization (development, testing, maintenance, DNS registration);
- Delivery (delivery mechanisms);
- Exploitation (passwords, encryption, PKC certificates);
- Installation (storage on disc, in memory, graphics cards etc.);
- Command & Control (communications, system configuration, inter and intra attack communications);
- Actions on Objectives (thesis inspired software).

Within this the evidence and literature is aligned, as per Hart, to the relevant stage of the Lockheed Martin Intrusion Kill Chain.

Interpreting the Data

McKee (2004) defines textual analysis as:

> "When we perform textual analysis on a text we make an educated guess at some of the most likely interpretations that might be made of that text."

(McKee, 2003)

This supports the second Aim and Objective: this thesis is derived partly from interpolating and extrapolating from ideas presented in white papers and research papers. The authors of these papers are reporting on what they have seen but just as importantly they are not reporting what they have not seen and, for their own company's Intellectual Property and business reasons, omitting some things that have been seen. This has been seen in a discussion of one family of ransomware. Sophos mention setting file length to 0, in a technique described as "0 allocation" (Loman, 2019, p. 10), yet Panda Security discussing the same ransomware make no mention of the technique (Panda-Security, 2017). Another explanation is that this technique may not have reached Panda Security's reporting threshold.

This thesis therefore performs textual analysis using interpolation and inference on what is in the gaps. The purist researcher may argue that this is not textual analysis but this author argues that textual analysis may be performed on known unknowns.

Initially the background reading to the thesis was to draw on searches in different languages. The language would have depended on the perceived origin of the APT. This can present problem with interpretations of a cultural nature. A non-Anglo-Saxon approach to being an APT may view the vulnerabilities of IT systems and hence CKCs in different ways; although this author is hard pressed to see how a technical (i.e. scientific) gap in IT defences can be exploited culturally. Perhaps this is part of the research journey? The real problem here is one of translation: it is known that automatic translations work best on short phrases or a search in a language other than English on individual words or short phrases. Translating an online search in this manner should produce best results but the problem is then how to translate the results. The same online translating regime may provide a sense of the content but not a translation in the way a natural bi-lingual person would.

As individuals and societies form their own model or view of the world and they are all not identical, then discrepancies of view arise. In extremes these discrepancies lead to wars and possible subjugation of one culture or society by another, this thesis denotes this as “Cultural subduction”. This is end point of Cultural Rejection, Cultural Blending, Cultural Submission, Cultural Subduction. An idea to be developed.

Harford 2016 quotes Ross:

> “... describes the problem as “naive realism”. By this he means the seductive sense that we’re seeing the world as it truly is, without bias or error. This is such a powerful illusion that whenever we meet someone whose views conflict with our own, we instinctively believe we’ve met someone who is deluded, rather than realising that perhaps we’re the ones who could learn something."

(Harford, 2016)

However, time became the limiter and few searches were performed in any language other than English.

2.12 Modelling the World

The author asserts that there are at least as many models of the world as there are people (some models may survive their originators). There are many similar models that are rooted in similar philosophies and there are conflicting philosophies which result in conflicting models. The epistemological position taken is that each world model is constructed by the individual based on experience.

Schulz asserts that to be fallible is to be human and that people get stuck in an internal state of being right which partly stems from early years where to be wrong meant being mocked by peers (and sometimes by pedagogues) (Schulz, 2011). People are then conditioned to work within our comfort zone. Of course, making mistakes that lead to catastrophic outcomes is not acceptable and such outcomes are generally not the result of a single bad action but an aggregation of seemingly unrelated actions, many of which are sensible and based on the known information of the time (HMSO, 1967).

However, feeling that one is right is a state when the observation fits one's internalised model of the world; but the model may be wrong. Being wrong, and learning from being wrong, should always lead to a modification the model.

For this author being wrong is not an issue: being wrong means that one has tried something outside one's bounds of knowledge or experience or made an incorrect assumption or interpretation.

Taleb (2007) asserts that people focus on a few well-defined sources of uncertainty, at the expense of others that do not come to mind (Taleb, 2007). Taleb, an ex-financial market trader, discusses the impact and modelling of low probability, high impact events and takes the title of the book “The Black Swan” from Australian black swans. Until the discovery of Australia, all swans known to the west were white. Taleb's assertion can equally apply to research: just because all observations fit the model (and further model-fitting observations strengthen the model) without an underlying proof there should always be doubt that the model is correct.

Taleb describes a form of sampling, however there is a problem with using small samples sizes to draw wider conclusions. Picking up on the one perspective, HESA (2016) reports that

> “45% (89,225) of academic staff on 1 December 2014 were female, the same proportion as in 2013/14. "

(HESA, 2016)

to take a statistically valid sample for any poll regarding UK women academics (95% confidence level, 2% confidence interval) the author of any research paper must engage with 2,338 female academics as calculated by the Australian National Statistics Service (NSS, 2016) and Creative Research Systems (Creative-Research-Systems, 2016). Reassuringly, both references agree.

In building a hypothesis, this author hypothesised that more time was spent on the “sexy, headline grabbing” issues related to IT Security and not enough was being done on the holistic view. This view is supported by Jacoby and Jartelius:

“We both feel that this really is one of the key problems today; we spend more time on new exciting vulnerabilities and threats than actually looking into the real problems.”

(Jacoby and Jartelius, 2013, p. 11):

The analogy provided by this thesis is that there are many pieces of the jigsaw, through reports published by AV companies and freely available, but the jigsaw box lid is not available. This is suggested by the second Aim and Objective.

2.13 Presenting to the World

Hammersley and Atkinson discuss Impression Management (Hammersley and Atkinson, 2009). The author has already been doing this at university by attending supervisor meetings and modules in a suit, white shirt and no tie. This is a brand which conveys position, informality with a hint of the formal and is then backed up in conversation with expert knowledge. The absence of a tie is a recent western statement but the suit sets one apart from the informality of hands-on IT professionals. The absence of tie is a hint of rebellion yet the suit allows presentable appearance before senior managers. This is the brand that will continue into the field for research should such action be necessary. Of course, this may backfire and the author ends up looking like some bloke trying to be trendy!

2.14 Legal Issues

All analysis is performed on machines and HDDs owned by the University of Gloucestershire or the author of this thesis. No testing was done over the internet. The author’s benevolent software (benware), below, was never transmitted over the internet.

Any access to these pieces of hardware is authorised:

The UK Computer Misuse Act (1.b) states that

> “A person is guilty of an offence if … the access he intends to secure is unauthorised.”

(HMG, 1990)

Any work which had the potential to adversely affect a computer and/or operating system was done on the author's personal machines or the University's forensics machines. Substantial testing was done before any more delicate software was run.

As far as possible software was written for this project and the least number of executables were imported and run. There are two reasons for this: the first is that there is no guarantee that such software does not harvest personal information and intellectual property; the second is that such software does not come with levels of assurance that it has been written to an acceptable professional standard that has been independently verified. For example, in the testing of a pristine USB HDD assurance was needed that the new HDD would be viewed as sent from the manufacturer. Therefore, software to ensure that USB devices may be used in read only mode but it came with no assurance. The author simply used the Microsoft command in a batch to add the relevant registry key and associated value to block USB devices.

A potential problem with a thesis of the nature of this one is that it becomes a handbook and tool for the black hat ("hacker") community. In the case of a successful PhD submission, an electronic copy of the thesis will be placed with the British Library (British-Library, 2020) with open access which may inadvertently contravene export control regulations:

"You will need an export licence to export any controlled dual-use items from the UK to another country outside the EU. Most dual use items do not require a licence if they are exported to the EU or the Channel Islands."

(HMG, 2019b)

The Windows Device Driver which was produced in support of the fourth Aim and Objective writes directly to given sectors of a HDD. This software overwrites data which is no longer being used. However, this may be viewed as jamming software under export control guidance as it could be modified to overwrite areas of the HDD (and SSD in other devices) which are being used:

> "Jamming equipment specially designed or modified to intentionally and selectively interfere with, deny, inhibit, degrade or seduce mobile telecommunications services and …"

(EU, 2009, p. L 134/161)

This device driver could be modified to "deny" or "inhibit" device use. For this reason, all computer source code reproduced and techniques to run executables will not be placed within the thesis for publishing with the British Library and will be retained separately.

However, this author believes that aggregating information such as that in this thesis, and building on it, to provide mitigations does not contravene export control guidance and is, therefore, not a reason for not doing the PhD.

Use of independent software does raise the question with the author about how assurance of the software this is viewed in legal circles.

The provenance of informed consent may be traced from the Nuremberg Code (Unknown, 1947) into guidance on informed consent is provided by the UK Government's Civil Service Government Social Research Unit (Civil-Service, Undated) and also the Social Research Association:

> "4.2 Obtaining informed consent
>
> Inquiries involving human subjects should be based as far as practicable on the freely given informed consent of subjects. Even if participation is required by law, it should still be as informed as possible. In voluntary inquiries, subjects should not be under the impression that they are required to participate. They should be aware of their entitlement to refuse at any stage for whatever reason and to withdraw data just supplied. Information that would be likely to affect a subject's willingness to participate should not be deliberately withheld, since this would remove from subjects an important means of protecting their own interests.

(Social-Research-Association, 2003)

Homan discusses the ownership of data and states that:

> "The granting to respondents of rights over data means that they may alter or withdraw data up to the point of publication."

(Homan, 1991)

This may be problematic if such data, or groups of data, are key to the thesis; years of work may be lost due to withdrawal of consent. It is therefore imperative to gain consent and custody of data from acquisition to publication.

O'Reilly suggests that ethnographers may introduce themselves informally into a group (O'Reilly, 2012). There is even an unwritten, inherent, suggestion that the group need not know that they are part of a piece of research. This raises ethics questions. It also affects Intellectual Property. Should the interviewee not know that they are, in effect, being interviewed and they pass on a piece of useful, unique information and this is then used in the research there may be Intellectual Property ownership issues. It is worse (and unethical) if the interviewer does not attribute the information. Whichever path is chosen this author suggests that the interviewer should always be "up front" with the interviewee.

There should be no instance where research is being performed and the subjects are not informed. Within this researcher's thesis, most of the research is performed after researching open literature. By its nature, open literature is freely available and there is an inherent consent behind it. Literature that is not open is, by definition, literature that has been withheld and not available to the world. Miller and Bell in Mauthner et al discuss what consent means but get a bit bogged down by it (Mauthner *et al.*, 2002). There was a slim chance that this author would need to conduct interviews of people who have been affected by malware or who work in the AV industry.

This author always presents himself as a PhD student before any interview or guided conversation takes place. The interviewee or group members are always aware that they are part of a PhD process. There is interpretivism, realism and relativism in these guided conversations.

The author did not need interviewing and listening skills. In previous professional work the author was trained to interview internal and external candidates for positions, listen carefully and make judgements against a defined scoring system.

Interviewing people for the PhD did not come to pass but should it have happened a proforma would have been constructed and will be read out. This proforma would have been constructed with the help of the university's Legal.

In short: people are different; they have different experiences; they have different views; they have different interpretations. There are different forces at work. Be aware of it.

2.15 The Individual vs The Group in Computing

Computing has a heroic culture, where the "visionary leader" points the way and all follow: recall the heads of Microsoft®, Apple and Facebook. The garage California counter-culture of engineering q.v. Intel, Apple (and in the case the latter using hardware to spoof dial tones in order to get free long distance calls) which see their counter-culture progeny manifesting as Wikileaks , Anonymous and The Pirate Party.

On the other hand, legislative bodies catch up through international bodies such as ASCII, ISO. Even the counter-culture has to communicate: it is no use having different computer architectures that cannot communicate.

2.16 Linking to the Objectives

To recap, this research objectives are to:

- discover where APTs lie on windows-based systems;
- create/build generic views of where APTs lie on Windows based systems using an appropriate Cyber Kill Chain (CKC);
- treat APTs on Windows based IT systems in such a way that the malware is negated;

- increase the cost of doing business for APTs, for example by overwriting unused disc space so APTs cannot place malware there.

The scope of the thesis has been defined academically, physically with respect to the hardware and conceptually with respect to the APTs software deployment. The academic literature with respect to HDDs and malware is explored using the academic techniques discussed. The white papers produced by AV companies will form the basis of discovery. Form this analysis a view of where APTs lie on a Windows based system is developed. These ideas are used to develop software to increase the business cost of APTs

2.17 Conclusion

This chapter has presented a discussion of the thesis scope and the steps taken in the investigative process. It has included academic justification (literature choice and boundaries) and includes the hypotheses that are in other chapters. Finally, these were then linked to the objectives.

3 APTS IN INFORMATION TECHNOLOGY

3.1 Chapter Overview

This chapter sets the scene by reviewing existing literature and it supports the first Aim and Objective. It commences with a discussion about APT motivation and a review of malware surveys. It progresses through software development costs, and the difference between a virus and malware. This is followed by a description of how an HDD works, the issues relating to HDD storage and where malware may reside on an HDD. It then segues to Windows operating systems. This is followed by a short discussion of where APTs may get their information to attack victims. The chapter concludes with a discussion of the literature gaps and possible lines of research.

3.2 The Problem of Tautology

A problem identified early in the literature review is that which this thesis calls "tautological information". In some cases, APTs are analysed and the APTs actions are written up by different academics and AV vendors over a number of years. Different AV companies work on the same APTs, gathering evidence from different customers and it is difficult to know if the different AV companies are reporting the same attack originators or not. Different APTs use the same or similar attack vectors which may appear to the analyst to be the same attack or originator

There are examples of within APT information tautology e.g. (Goncharov, 2012), (Goncharov, 2014) where the former is the initial write-up and the latter a write-up of revisiting the subject. It is difficult to see in the latter if any of the information is a repeat of previous information. Another problem is the "borrowing of ideas" across different APTs. One AV firm believes that APTs are actively emulating each other through the review of publications and data sources. This helps to protect their infrastructure, plant false flags and muddies the attribution waters (Insikt-Group, 2018a, p. 16). It is not surprising that the idea for this thesis – of reviewing freely available publications – would be independently developed. A malicious extension of “borrowing ideas” is mimicking the TTP of another APT and the commandeering of their old infrastructure (Insikt-Group®, 2019a, pp. 2-3).

3.3 What is, and What Motivates, an APT?

It was not the intent of this thesis to define an APT but after careful consideration it became clear that one would be needed.

This thesis asserts that the term “Advanced Persistent Threat” came into being for a specific purpose and has since become the generic term, used interchangeably, for cyber attacker and deployment of malware. It will be demonstrated later in this thesis that not all threats are advanced or persistent – they may be trivial, transitory, issues. Following a review of definitions, a broader definition will be proposed.

The term APT may be traced back to Cloppert who first heard it used by the United States Air Force’s (USAF) 8th Air Force in a small meeting in 2006 (Cloppert, 2009 p. 14). While Holland attributes APT toretired USAF General Greg Rattray

(Holland, 2013)

This may have evolved from Rattray’s earlier work (Rattray, 2001, pp. 79-118) on “The Cyberterrorism Threat” which could have been in response to Tenet’s definition of the growing cyber threat which he coined as weapons of mass disruption.

(Tenet cited in Rattray (Rattray, 2001, p. 80)).

As the term APT became to be used more widely one definition began to stand out:

> “In 2006, the United States Air Force (USAF) analysts coined the term advanced persistent threat (APT) to facilitate discussion of intrusion activities with their uncleared civilian counterparts. Thus, the military teams could discuss the attack characteristics yet without revealing classified identities and explains the components of the terminology.

- **Advanced** means the adversary is conversant with computer intrusion tools and techniques and is capable of developing custom exploits.
- **Persistent** means the adversary intends to accomplish a mission. They receive directives and work towards specific goals.
- **Threat** means the adversary is organized, funded and motivated."

(Jeun, Lee and Won, 2012).

This APT definition is widely quoted in papers, books and on the internet e.g. (Cert-Polska, 2014, p. 17) but an extensive search has been unable to find the original 2006 reference in the citation. This search includes the internet archive (Wayback-Machine, 2017). A Google Trends search (Google-Trends, 2017) found the first five references to the term "Advanced Persistent Threat" in October 2007 which aligns with the 2006 origin, allowing for the phrase to spread organically. NIST also defines an APT

(NIST, 2011)

This thesis view that APT is used interchangeably as an attacker and malware is supported elsewhere where it is suggested that the entire APT space suffers from definitional uncertainty (Edwards, Ford and Szappanos, 2014). However, this thesis asserts that the problem is worse than this in that APT is a misnomer as used by the IT community. This thesis further asserts that a "Threat" cannot do anything other than "Threaten". This misnomer is highlighted elsewhere (IETF, 2007, pp. 304-306) where a threat is a "potential for violation of security". Indeed, a threat may only be a threat because it is perceived as such by an organisation in their Risk Register. Just because an adversary "is organized, funded and motivated" it does not mean that that particular adversary will attack any network or machine. Threats may be made in the generic sense; they may be directed at specific organisations, entities or a wider industry. Drawing on the above NIST definition the threat may be persistent in time or not. The threat may change or not. The threat may be persistent in level or not. The threat may be advanced, primitive or thuggish. However, a threat only threatens.

In the organisational risk management world a threat may highlight a Risk (in a Risk register) to an organisation and when the threat is acted upon or materialises it becomes an issue (APM, 2018). The Risk Management process may, or may not, have considered the threat. At the stage it becomes an Issue the “materialised Risk” needs mitigations. The four treatments of Risk are: “Avoid, Address, Accept, or Transfer” (Shostack, 2014). This thesis chooses “Address”. The use of the word “Treatment” in risk management is one reason for inclusion of the word in the title of this thesis; another being a collision of definitions with “Disrupt” which is in the LMKC Course of Action Matrix to be discussed later.

The recent development of Risk Management in wider UK business has its roots in the UK stock market where it is a condition of the UK stock market Listing Rules. These rules require companies to make a statement of how they have applied the Principles of the UK Corporate Governance Code. Principle C states that the board should have an effective mechanism to manage risk (FRC, 2018, pp. 2, 4, 10-12). These risks are generally aggregated in the form of a Risk Register. Not all threats may be real – they may have the perception of reality in the mind of the perceived victim (q.v. Ellerton’s assertion, discussed later, that coherence matters more than truth). There may be a genuine advanced, persistent, threat to business, or the IT industry in general, but for any one business or organisation this may not be the case.

Different types of attack may be embraced by the APT term. A DDOS attack or well-hidden Trojan may be advanced. Cross-site scripting is not. A threat, advanced or otherwise, may become an Issue which is, or is not, persistent. A DDOS attack or well-hidden Trojan may be persistent. A one-off ransomware attack is not. Neither of these hypothetical examples are threats. They are issues. This may seem like pedantic semantics to some readers, but it is important to distinguish between the descriptions. This point is also taken up by Bejtlich who states that people use “threat” to mean “malware”

(Bejtlich, 2010)

(Inherent in Beitlich is the concept of “Mechanism of Action” – “someone to control” which will be taken up later in this thesis).

There seems to be no consensus about what is an APT and there is blurring in the use of the term APT between threat, threat actor (those who carry out development and deployment of malware) and malware. Sophos assert that some common traits are: they are targeted; goal orientated; Persistent; Patient; and Call Home (Hudson, 2014, pp. 2-3). While others (Friedberg *et al.*, 2015) suggest that they are slow moving. “Slow moving” may be compatible with “Patient”. F-Secure assert that while APTs do not impact the majority of consumers, they are of particular interest to governments and major corporations (F-Secure, 2016, p. 10). It may be true that APTs do not impact the majority of consumers but a non-directed attack against consumers which looks for success by simply infecting as many devices as possible and “plays the numbers”, e.g. for banking details, is still a problem and as long as the associated malware resides on the consumers’ device it is still persistent. It may also be an impact in the cost of electricity (as in the case of cryptomining), anti-virus (AV) defences etc. and these attacks may be deployed by an advanced, persistent, team.

Blue Coat (Blue-Coat, 2011, p. 3) state that it is better to think of an APT as “Advanced Persistent Attackers” and that they are advanced because the attacks are well-planned and co-ordinated with every available tool. They are persistent because they are patient and are focused on avoiding detection. Blue Coat further state that the difference between an APT and mass-market malware is that APTs do not necessarily need unique malware or zero days. They assert that mass-market attacks use blended threat using multiple tools. Building on this, Technical Persistent Attacker (TPA) is a better description than APT. Furthermore, a TPA delivers malware. A “Threat” is unable to do this.

However, not all attacks are persistent or technical (non-technical, business process manipulation, attacks have been highlighted earlier in this thesis). They may be just a “smash and grab”. They may delete the MBR or MFT and although, strictly, being persistent, the machine is rendered inoperable. Technical Cyber Attack (TCA) is a better fit. Are there any non-Technical Cyber Attacks? There will be a return to a better definition after a review of current APT definitions, a summary of which is:

- Chen et al: Specific targets; Highly organized; long-term campaign; stealthy and evasive (Chen, Desmet and Huygens, 2014);
- RSA: Highly targeted; Well-funded: Well-researched; Designed to evade detection; Multi-modal and multi-step: (Robinson and Keswani, 2016, p. 2);
- Stewart: APT as Cyber-espionage activity targeting government, industry or activists (Stewart, 2011, p. 3);
- Websense: Targeted; Evasive; Persistent; Complex (websense, 2013, p. 3);
- Alien Vault: Exploit publicly known vulnerabilities but the attackers also are highly skilled etc.; accomplish a mission that can take place over months; Dedicated organized groups are behind the attack motivated by political, economical [sic] or military reasons. (Blasco, 2010, p. 3);
- WatchGuard: An unknown, zero-day attack that has malware payloads and uses kernel rootkits and evasion-detection technologies; It doesn't stop. Sophisticated difficult-to-detect threats are launched daily at targets ranging from large corporations to small and midsize businesses. (WatchGuard® Technologies, 2016, p. 2);
- Imperva: APTs leverage malware and hacking techniques; APT hackers continue the assault until the attack is successful; APTs are growing and require organisations to adapt their security strategies and practices. (Imperva, 2011, p. 2);

- CA: The attacker has the significant technical capabilities required to take advantage of weaknesses in the target; APTs often unfold over the course of years; have the motivation, ability and resources needed to be successful. (ca-Technologies, 2014, p. 3);
- Bejtlich: adversary can operate in the full spectrum of computer intrusion; the adversary is formally tasked to accomplish a mission; the adversary is not a piece of mindless code (Bejtlich, 2010);
- Trivett: A target of value; Multiple types of stealthy attack vectors; sophisticated adversary (Trivitt, 2013, p. 3).

Clearly, there are differences of opinion. Funding, technical ability, persistence, and patience are themes. Imperva suggests that attackers continue until the attack is successful. This thesis is about increasing the costs of attackers, perhaps to the point where they give up and go elsewhere.

Working with the HMG legal definition above (HMG, 1990) this thesis only deals with illegal acts and therefore the description should be Illegal Cyber Attacks (ICA) which are perpetrated by threat actors using unauthorised techniques for gain by deploying malicious software (malware).

As previously discussed, this thesis is agnostic towards the origin and motives of APTs but this thesis will touch on motivation, first drawing on a Nietzche's definition of man's drive for power which provides a basis for motivation

. "… and use of weapons and power as an extension in support of this"

(Nietzsche, 1968, pp. 383-386)

While Maslow has a similar view of drive or needs with Man as a "perpetually wanting animal

(Maslow, 2016, p. 4)

It is clear, therefore, that at least Nietzsche and Maslow see Man as a being who has a drive or thirst for power and is prepared to weaponise that drive to bring about a desired outcome.

No literature was found on the motivation of APTs and only three items were discovered relating to the insider threat. It is from these definitions, and related work, that this thesis will develop a general definition which includes all threats.

Wall, in a wider discussion, breaks down IT breaches into two categories: internal and external (Wall, 2013). It is hard to gain a view as he provides percentages from a sample size of 3001 but a non-scientific review for this thesis looks as if the ratio is about two to one in favour of breaches from an external source. This thesis will consider the holistic view of cyber security breaches and, where the malware lies on the system, align it with the LMKC.

The insider threat has been defined as

"… an insider is a trusted entity that is given the power to violate one or more rules in a given security policy."

(Bishop, 2005).

This thesis will now modify and develop the definition to become (This thesis author's emphasis):

"An insider is a trusted entity ***who has*** the power to violate one or more rules in a given security policy."

An insider has not been given the power; it is given the opportunity to exert the power through poor QC. This power may have been taken but not given.

Elsewhere, work by Burkett comes from a "traditional espionage" background and he refers to insider motivation as: money, ideology, coercion and ego. This neatly fits in an acronym: MICE (Burkett, 2013).

Kaspersky suggests that motivation for malicious insiders is hard to predict and anticipate but suggests financial gain, disaffection, coercion and simple carelessness (Kaspersky, 2016b). However, this thesis suggests that carelessness is not an attacker motivation but a victim attribute and the other three still fit Burkett's model.

The UK's Centre for the Protection of Nation Infrastructure (CPNI) describes an Advanced Persistent Treat as an entity (an individual, collection of individuals into an entity or collection of entities) that has a quiddity to wish to have a comparative advantage over its victims or target and categorises the motivation of insider threat: Financial gain (47% of cases); Ideology (20% of cases) ; Desire for recognition (14% of cases); Loyalty to friends/family/country (14% of cases); Revenge (6% of cases) (CPNI, 2013). However, by 2015 this had changed to: Financial gain; Revenge or notoriety; Fear or coercion; Ideological. (CPNI, 2015).

"Loyalty" and "Ideology" may be one and the same so they may be combined under the latter heading which is generic (one may not have the same ideological beliefs as one's master but the ideology of loyalty leads one to follow or do the bidding). Additionally, this thesis asserts that "Notoriety" in CPNI's model is one of a number of Ego related reasons and that Revenge should be kept separate as it overlaps Ideology and Ego. Furthermore, although Revenge may be considered a form of ideology for the purposes of this thesis there is less overlap than loyalty and ideology.

For this thesis the MICE acronym for motivation can now be extended to a new acronym, **CRIME**: **C**oercion, **R**evenge, **I**deology, **M**oney, **E**go.

These findings may now be aggregated for this thesis and applied to ICAs/APTs by extending Bishop's definition to any threat but for which includes ICA:

"An Illegal Cyber Attack (ICA) is a technique used by an entity which is motivated by a combination of any of financial gain, revenge, coercion, money, ego or ideology and who believes that they have the power to illegally violate one or more rules in a given security policy and perform an action or actions to affect a computer or computer system to which that given security policy applies."

In summary: The ICA is the modus operandi, Illegal Cyber Attack Team (ICAT) is the group of one or more people performing the ICA. Malware is the malicious software laid down by the ICAT. However, for consistency with the rest of the world this thesis, as presented earlier will used the terms APT for the team of attackers and malware for the malicious software laid down on victims' machines.

3.4 Malware Surveys

Several malware surveys and categorisations have been produced. Most concentrate on the business, entry to victims' systems and high-level technical aspects of the malware (Uppal, Mehra and Verma, 2014), (Saeed, Selamat and Abuagoub, 2013), (Damshenas, Dehghantanha and Mahmoud, 2013), (Landage and Wankhade, 2013). Others are a blend of this techniques: keyword identification (e.g. SetWindowsHookEx - to which this thesis will return) and functionality of the malware to statistical attributes of the malware presented visually (Egele *et al.*, 2012). Elsewhere keywords and some statistical properties (Rudd *et al.*, 2019) are discussed, touching on encryption and commenting about the high number of false positives when searching for hooks. It also includes N-grams, instructions counting, as well as a short categorisation of obfuscation techniques:

"like dead code insertion, register reassignment, subroutine reordering, instruction substitution, code transposition, and code integration"

(Gandotra, Bansal and Sofat, 2014)

A breakdown of malware detection techniques is provided in (Gaikwad, Motwani and Shinde, 2015) their Figure 1 and in more detail elsewhere (Ponnambalam, 2015). Although Ponnambalam mentions malware encryption one has to look elsewhere for a classification of encryption methods (Singh and Supriya, 2013).

3.5 The Difference between Virus and Malware and Identification

Zamora (Zamora, 2015) states that a virus self-replicates in order to achieve its goals, while malware is an umbrella term that includes rootkit, virus, trojan, spyware, worms, adware and ransomware. This thesis asserts that this is the correct way to view the terms but, like the term APT, both virus and malware are used interchangeably.

No academic evidence could be found about how commercial anti-malware works. This could be because terms and conditions prohibit reverse engineering of products (McAfee, 2018b), (MalwareBytes, 2018b). The researcher, therefore, is left with comparing and contrasting outputs from AV companies in the form of whitepapers: AV software appears to users, in a way, as a Hidden Markov Model.

Symantec provide a illustrative view of heuristics (Schmall, 2018) which looks for previously unexamined functionality of malware using a weighting system which, when the weight passes a threshold, triggers an alarm. This is tantalising as Schmall does not appear to have written any academic papers on the subject.

A combination of any of Heuristic-based detection, behavioural-based detection, sandbox detection, data mining techniques (Eliz, 2018). Suspicious behaviours may include modification of registry entries or start-up list changes (MalwareFox, 2018). One can envisage this being work testing observation against a known sample e.g. modified operating system against a pristine version of that system. Both the scoring mechanism and observations against a known base were part of the thinking behind the development of the submission of the proposal for this thesis.

Given the titles and order, some of these definitions may be copies of other websites e.g. Comodo website (Comodo Security Solutions, 2018)).

3.6 HDD Geometry

3.6.1 How a HDD Works

This section describes how a HDD works with particular emphasis on the allocation and de-allocation of storage. This is crucial to the understanding of where malware may reside. Some of this section paraphrases and updates earlier work by the thesis author (Bentley, 2008) for a master’s degree.

The mechanism and geometry of a HDD is described by Sobey (Sobey, 2004, pp. 4-12) as well as Arpaci-Dusseau and Arpaci-Dusseau (Arpaci-Dusseau and Arpaci-Dusseau, 2018, pp. 1-17). A HDD is comprised of a number of platters. Each platter contains concentric tracks which are divided into 512 user byte sectors. Encoding and error detection needed for the user data means that the sector size increases to 600 bytes. Every HDD disc contains areas that are faulty, and therefore unusable (and inaccessible), either at the time of manufacturing or during the life of the disc. The former is known as the P-list (Primary list) and the latter is known as the G-list (Growth list). The number of areas in the P-list does not affect the performance of the disk but the number of areas in the G-list, as it grows, does. More recently with the Advanced Format supported by IDEMA, the International Disk Drive Equipment and Materials Association (IDEMA, 2019), HDDs have been able to support sector sizes larger than 512 bytes in multiples of 512-byte logical sectors of user data, notably 1KB, 2KB, 4KB (Chicoine *et al.*, 2007, p. 10). It is noted elsewhere that the sector size can be greater than 512 bytes (Microsoft, 2018j). The hardware and software in this thesis use 512-byte sectors.

Each sector of disc is delimited at the start and end by the use of delimiters. User data may look like these delimiters, so the user's data is run-length limited (RLL) encoded. RLL algorithms restrict the consecutive number of binary zeros and ones. Not having RLL encoding would mean that the hardware would not be able to correctly interpret the user's data. RLL encoding increases by 1% - 12.5% the amount of data in the sector. User data may be encoded up to five times and may also include insertion of timing information before the application of run-length limiting.

3.6.2 Different types of HDD and Storage

This thesis notes that there are different types of HDDs such as ATA and SCSI, and that HDDs may be internal to a laptop or PC as well as external USB enabled HDDs.

Information is stored on computers in many ways for example on HDDs, on-chip RAM, on flash memory and on graphics cards. This thesis is based on HDDs.

3.6.3 Master Boot Record (MBR)

"The master boot record (MBR), the most important data structure on the disk, is created when the disk is partitioned. The MBR contains a small amount of executable code called the master boot code, the disk signature, and the partition table for the disk. At the end of the MBR is a 2-byte structure called a signature word or end of sector marker, which is always set to 0x55AA. A signature word also marks the end of an extended boot record (EBR) and the boot sector."

(Microsoft, 2009a)

3.6.4 The Master File Table (MFT)

"The NTFS file system contains a file called the *master file table*, or MFT. There is at least one entry in the MFT for every file on an NTFS file system volume, including the MFT itself. All information about a file, including its size, time and date stamps, permissions, and data content, is stored either in MFT entries, or in space outside the MFT that is described by MFT entries."

(Microsoft, 2018q).

Descriptions of the MFT are openly available from Microsoft (Microsoft, 2018q) or discussed by researchers (Hsu *et al.*, 2016).

Deletion of a file by the operating system does not mean it is deleted or erased. Every file has an entry in the Master File Table (MFT). This is a file index. On deletion the file's MFT entry is changed to tell the operating system that the area of storage occupied by the file is now available for use. This concept is later addressed (LaBarge, Mazzuchi and Sarkani, 2014). Schneier states that eventually the area will be used again (and hence over-written).and also states that erasing data is much harder that one might think (Schneier, 2000, p. 253). The HDD may be tidied up by defragmentation. A defragmenter brings together data blocks that belong together logically and releases unused smaller blocks into larger contiguous sets of disc space.

Files can be read by the user using available software within the Windows operating system. In this scenario the file is available and has not been deleted. After deletion the file is moved to the Recycle bin from where it may be recovered by the user using Windows software. Once deleted from the Recycle bin a file is not recoverable using Windows software. As previously discussed, the data appears to have been deleted, but what has occurred is that the MFT link has been destroyed and the data is still available, depending on how long ago it was deleted. Commercially available software can recover the data if the operating system has not overwritten it.

3.6.5 The Windows Registry

"The *registry* is a system-defined database in which applications and system components store and retrieve configuration data.."

(Microsoft, 2017aa).

Microsoft provide an overview of the registry (Microsoft, 2012). Both Thomassen (Thomassen, 2008) and Morgan (Morgan, 2008) independently discuss registry forensics and places where data could hide.

> However, it has been demonstrated that programs which uninstall registry may inadvertently corrupt those

(Kahvedžić and Kechadi, 2009).

It has also been shown that, on uninstallation, not all registry keys for the software are deleted (Kim, Lee and Hong, 2008). Although this demonstrates how deleted registry keys and key fragments may be recovered forensically, using hidden parts of the registry to hide malware, and expect it to persist, does appear to be a risky strategy for attackers.

3.6.6 Update Sequence Number (USN) Change Journal and Cluster Usage.

"… the NTFS file system maintains an update sequence number (USN) change journal. When any change is made to a file or directory in a volume, the USN change journal for that volume is updated with a description of the change and the name of the file or directory."

(Microsoft, 2018d).

There appears to be little on forensic examination of the USN change journal. Lees provides evidence (Lees, 2013) of changes when InPrivate Browsing and CCleaner, a disc wiper, are used but nothing about where malware might be able to reside. This thesis postulates that malware may be able to manipulate the USN change journal and/or cluster usage. However it is asserted that it does not appear to be possible to directly change the USN change journal (Eterovic-Soric *et al.*, 2018) but surely software running in kernel mode could directly amend the contents on the HDD?

It is suggested (Singh and Singh, 2018) that using a correlated model for program execution artefacts, which would include the USN journal, could be designed. This would be a form of Process driven defence and this author believes that this analysis should go further and look at the USN change journal as a file, and hence Landscape Driven Defence, to determine inclusion of malware in the USN change journal and holistic issues relating to the USN change journal.

At least one piece of malware deletes the USN journal (Wüest and Anand, 2017, p. 16) which may be an indicator of the presence of malware although Microsoft recommend housekeeping by administrators by deleting the journal when the USN approaches the maximum USN (Microsoft, 2018h). However, the ability to delete the USN journal suggests a conflict with the observations of Eterovic-Soric, above.

3.7 How Data is Stored on a HDD

Data is stored on the HDD in several logical ways leaving unused space on the HDD. The existence of this unused space is a point to which this thesis will return.

Data is also stored by programs in different ways depending on the function. Wilson et al describe how memory is allocated in Ramps, Peaks, and Plateaus:

- Ramps: these programs accumulate data monotonically over time;
- Peaks: these programs use memory in burst, allocating and releasing it when needed;

- Plateaus: these programs allocate memory at the start and use it for long periods.

(Wilson *et al.*, Undated, p. 15)

Knuth describes how data in the storage device is assigned in blocks by software, (known as an allocator) using complex mathematical algorithms (Knuth, 1997, pp. 435–456).

Jones and Lins discuss the three ways that storage can be allocated by a compiler for high-level languages:

- Static Allocation is where the names in the program are bound to storage locations at compile time;
- Stack allocation is where an activation record or frame is pushed onto the system stack as each procedure is called, and popped when it returns;
- Heap allocation is where data structures may be allocated in any order.

(Jones and Lins, 1996, pp. 2-3)

Randell describes how data may be inefficiently allocated by one of two means: (1) the size of the data is greater than the block size which means that small areas of free storage space are skipped over and the data is allocated elsewhere causing external fragmentation; (2) the size of the data is less than the size of the block size which causes internal fragmentation; (Randell, 1969).

"The page size is often a parameter that can be chosen by the operating system" (Tanenbaum, 2001, p. 237) and from which it can inferred that choosing operating system page sizes as multiples of HDD sector size would be more efficient than otherwise. However, Tanenbaum (Tanenbaum, 2001, p. 237) later goes on to point out that "On average, half of the final page will be empty." And later still that external fragmentation is also called checkboarding (Tanenbaum, 2001, p. 253). Analysis of such space, also known as slack space, is known to the forensics community (Prem, Selwin and Mohan, 2017), (Srinivasan and Kolli, 2013), (Slot, 2015, p. 16). The latter asserts the difficulty of detection of rootkits in slack spaces but the MFT should have the file size and hence disc usage. Any slack space can therefore be overwritten without penalty to the machine's user.

Knuth also provides examples of different allocator algorithms but the results are the same: the storage allocated is not contiguous. Garbage collectors may be needed to collect up free space (Knuth, 1997, pp. 408–423).

Zelkowitz has produced a graphic demonstrating data hiding tactics for Windows and Unix file systems (Zelkowitz, 2008, p. 5) e.g. end of partitions.

The size of slack space can be non-trivial: A review of 18 different versions of Windows for slack space found slack (unused) space of the order of tens of megabytes (Mulazzani *et al.*, 2013).

It can therefore be seen that memory is being continuously allocated and unallocated leaving empty space on the HDD. This, along with the allocation methods described above, means that the data is not uniformly distributed on the HDD.

3.8 Persistence in the HDD Firmware

Seagate defines firmware as

> "… as a computer program that is hard-coded into the hard drive and contains its basic operational storage programming. Another way to think of it is that firmware is a software middleman that allows your hardware to talk to software (i.e., Windows operating systems, etc). For a hard drive, firmware is a program that governs the behaviour and factory settings, even the identity, of that drive. Any drive that comes out of the factory and is in use in a computer or server has firmware installed on it from the factory."

(Seagate, Unknown)

Kaspersky have observed one APT reprogram the firmware of the HDD and provide an API into a set of hidden HDD sectors. (Kaspersky, 2015c, pp. 16-18).

The plugin version 3 can reprogram six HDD "categories" (although not entirely clear from the reference it appears that these are manufacturers and models); while the later version 4 can reprogram 12 "categories".

Microsoft offer guidance on how to modify firmware (Microsoft, 2016) while others have provided proof of concept (Read *et al.*, 2013). The reason that firmware is modifiable may be that, as suggested by Gruhn, that HDD firmware is not digitally signed (Gruhn, 2017) and that the contents of given HDD sectors may be hidden by the modified firmware to: only return zeros from specific sectors being read; only return zeros for all reads; or return returns for reds and overwrite the requested data with zeros.

However McAfee lead the reader to believe that HDD firmware has signatures which can be checked (McAfee, 2016, pp. 1-2).

The move from BIOS to UEFI is noted. Each HDD partition has a Volume Boot Record (VBR) and at least one attack has been seen overwriting VBR areas (Dell-SecureWorks-Counter-Threat-Unit-Threat-Intelligence, 2013).

MBRs may be manipulated (Grill, Platzer and Eckel, 2014) and forensically analysed (Gruhn, 2017) and Symantec (Symantec-Security-Response, 2015d, pp. 5, 19) have observed one APT overwriting the MBR:

McAfee have also seen an APT with the facility to wipe a MBR (Sherstobitoff, Liba and Walter, 2013, p. 5) using a small, 24KB, executable and F-Secure have seen modification of the MBR to achieve persistence:

(F-Secure-Labs, 2014d, pp. 5, 8)

F-Secure-Labs note that this method of persistence has stability issues and crash dumps are controlled.

Mandiant observed an APT using Volume Boot record modification as their persistence mechanism

(Mandiant, 2017, pp. 14-15).

3.9 Windows is a File-Based Operating System

It is asserted that there is a lack of up-to-date anti-forensic research literature between 2010 and 2016 (Eterovic-Soric *et al.*, 2018). While this does not give this thesis carte blanche it does indicate that, supported by the academic evidence reviewed above, there is a wide scope for this thesis. Also, it appears that all the research points to artefacts and fragments of malware from APT attacks rather than viewing them holistically. It is for this reason and, as will be shown that Windows is a file-based system, that the thesis will be based on looking at malware at the file level. This will now be further explored.

In NTFS "Everything is a file"

(Carrier, 2005, p. 274), (Casey, 2007, p. 134) (Richard Russon, 2018).

> "NTFS is based round a relational database. This is the MFT or Master File Table. All "objects" stored on the volume are regarded as files, except the Partition Boot Record"

(Sammes and Jenkinson, 2007, p. 217)

Before finding these assertions, this author recognised that for the purposes of this thesis "the file is the unit of currency". Whitepapers and academic papers re-enforced this belief and finding the evidence, above, was a vindication. There are very few references to this concept and non are directly from Microsoft and it is sparsely repeated (Russon and Fledel, Undated, p. 5). It is recognised that the assertion is a small sample size but, on the one hand this sparseness demonstrates gaps in the academic knowledge of NFTS (and hence Windows) and on the other hand it provides direction and support for this thesis.

Eterovic-Soric et al also assert, and later demonstrate, that that it is possible to run malware without modifying the change journal. Should this be the case then it might be possible to cross-reference file creation, existence and deletion directly from the HDD with a missing USN change journal entry. This, together with a test of kernel level software making change to the USN change journal, would be an interesting avenue of exploration for this thesis.

One research group (Garba *et al.*, Unknown) describe the Lockheed Martin Intrusion Kill Chain and claim that the research will effectively detect APTs based on the kill chain. The work seems to only detect at the perimeter for incoming attacks, not those which have got past the perimeter. One could argue 100% detection at the perimeter means there is no resident malware on the system but it is a bold claim. In addition, they only seem to address the Detect part of the LMKC "Courses of Action Matrix".

There are a variety of papers looking at statistical technique for malware identification. One looks at Major Block Comparisons (Kang *et al.*, 2012), while another suggests a visualisation (Han, Lim and Im, 2013). However, these rely on malware being unencrypted. Something which this thesis will address.

Cyber Security presents the concept of "Data at Rest" (on a permanent storage medium), "Data in Transit" (traversing a network) and "Data in Use" (in virtual storage) (Willet, 2008, p. 159). The first two definitions are generally used (Microsoft, 2019b) as it may be argued that "Data in Use" is either in a store, albeit virtual, or is being processed i.e. is in transit. The academic literature points to researchers observing a process and then building a generic process to counter-act this. This thesis denotes this as "Process Driven Defence". This thesis holistically views the HDD and builds a HDD view which this thesis calls "Landscape Driven Defence".

3.10 File Sizes

Analysis of executable file sizes appears to be a little researched area. An early paper (Lee *et al.*, 1998) briefly discusses Windows NT executable file sizes in the context of performance of desktop applications for Windows NT. Methodologies for identifying file types (Li *et al.*, 2005), (Karresand and Shahmehri, 2006) look at content and not file size - the first paper's authors deliberately truncated files before statistical analysis. Elsewhere (Shahzad, Haider and Lavesson, 2010) there is a discussion about using binary features of files but nothing on the file size. This work is broadly similar to n-gram analysis (Kim *et al.*, 2014). Another paper discusses identification of executable signatures on the basis of statistical criteria (Krivtsova, Lebedev and Salakhutdinova, 2017) but, again, there is no analysis of file sizes. The importance of the detection of executable files may lead to discovery of malware, either individually or as part of a botnet (Satrya, Cahyani and Andreta, 2015).

Further evidence supporting the lack of study of executable files lengths it is noted that a study of data mining techniques for malware detection using file features (Siddiqui, Wang and Lee, 2008) makes no mention of file length.

A file size study (Tanenbaum, Herder and Bos, 2005) comparing a university's UNIX system 20 years apart is one of the closest to the method adopted in this research. Although it is based on UNIX systems and all file sizes are a grouped in bands of a multiple of 2, the authors state that the median file size if 2475 bytes i.e. an odd number. One other study (Downey, 2001) suggested a statistical distribution. Although not an exhaustive search it appears that little to no work has been performed on Windows file lengths.

This research takes two different lines of analysis by separately looking at the distribution of the most and least significant digits (first and last digits) of executable file sizes. It is asserted that the introduction of malware is a form of fraud and that any insertion of malware may affect either or both distributions. Within the accountancy profession Benford's Law (Nigrini, 2012, p. 5) is a well-known method for highlighting potentially fraudulent activity. This is performed by analysis of the distribution of the most significant digit of each number in a set of numbers. Given that analysis of the most significant digit is to be performed a hypothesis was constructed that there was no difference in the distribution of the least significant digit (last digit) of the same set of numbers. A search of academic literature found no such work had previously been performed. Analysis of the last digit of the same set of number – is it odd or even – might provide other insights.

Benford's Law states the distribution of leading digits (provided in Figure 2-1, below) of numbers in many naturally occurring datasets follow a negative exponential distribution. The expected frequencies for these single digits are given by:

$$\text{Prob } (D_i = d_i) = \log (1 + (1/d_i)) \quad d_i \in \{1, 2, \dots , 9)$$

Number	Probability
1	0.30103
2	0.176091
3	0.124939
4	0.09691
5	0.079181
6	0.066947
7	0.057992
8	0.051153
9	0.045757

Table 3-1: Benford Probabilities

3.11 From Where the APTs May Get Their Information

A simple web search using the keywords "Exploit Database" produces a number of websites such as that hosted by Offensive Security (Offensive-Security, 2018).

Some APTs use the same malware for their attacks and, of course, different AV companies are sometimes analysing the same APT without knowing it until publication of their results. Attempts have been made to deduplicate both instances but there may be some references which support the same observation (Kaspersky, 2014b, pp. 12-19). This overlap between AV companies' analysis and separately between APTs' malware deployment made it difficult to disaggregate some data for this thesis.

APT can also make use of well-known TTP (pwc-BAe, 2017, pp. 16, 18-19).

3.12 Where are the Gaps in the Knowledge?

This chapter has shown that one tenet of the Windows OS is that everything is a file. Malware surveys and other papers showed no evidence of making use of this. Hardly any academic papers which discuss the malware problem addressed shared attributes of files that define file types. For example, these attributes may come from aggregation of ostensibly similar files based on filename extension or on shared properties. There is little discussion in AV companies' white papers about how AV products work. The only hints are vague definitions like heuristics and keyword search. Finally, these white papers displayed no direct references to academic papers.

This apparent lack of research at the file level is the gap in the knowledge that this thesis will explore.

3.13 Conclusion

This chapter supports the first Aim and Objective, through a review of academic literature and some supporting white papers. It has laid out APT motivation, a discussion on the difference between a virus and malware. It followed with a description of how a HDD works, issues relating to HDD storage and where malware may reside on a HDD and ended with a review of where are the gaps in the knowledge. It then looked at evidence of analysis of Windows as a file-based operating system, No such evidence could be found for a file-based review centred on file attributes and unused disk space.

The thesis will now progress to discuss the wider, global, business issues and taxonomy issues of APT attacks.

4 THE TECHNOLOGICAL PERSPECTIVE

4.1 Chapter Overview

This chapter commences with a review of technological revolutions and continues with the reasons for caring about IT Security. This is followed by a review of a history of attacks on IT systems, parallels to attacks in the non-IT physical world and also popular culture, which affect the public perception of an APT and IT Security. It then follows with a discussion on the costs of malware infection to the defender and the business positions for both attackers and defenders. It also discusses how bad the problem of malware and APTs might be, followed by a definition of an APT.

The chapter highlights the governance chain through international co-operation to company governance with respect to Risk management. It continues with a review and comparison of various Cyber Kill Chains, with discussion and argument for selecting the Lockheed Martin Intrusion Kill Chain. This supports the second Aim and Objective

4.2 Technological Revolutions and The Long-term Business View

The emergence and use of computers by all and the rise and ubiquitousness of the internet have has been nothing short of a technological, business or industrial revolution and, no doubt, still has far to run. However, the existence of such a revolution is not new. Perez asserts that technological revolutions recur every half century and are based on the causal mechanism of capitalism (Perez, 2002, pp. 5, 78). She lists the five technological Revolutions as:

- The Industrial Revolution;
- Age of Steam;
- Age of Steel, Electricity and Heavy Engineering;
- Age of Oil, Automobiles and Mass Production;

- Age of Information and Telecommunications.

Perez describes the four stages of a technological revolution: IRRUPTION (Love affair of FK (Financial Capital) with revolution); FRENZY (Decoupling FK-PK (Production Capital)); SYNERGY (Recoupling FK-PK); and MATURITY (Signs of separation). (Perez, 2002, p. 74). Perez also defines FK and PK respectively as the criteria and behaviour of the agents who possess wealth and the motives and behaviour of those who generate new wealth. Okasha re-enforces this view with:

"Kuhn's characterization of the history of science as long periods of normal science punctuated by occasional scientific revolutions …"

(Okasha, 2002)

This argument of technological revolutions and technological bubbles is elaborated on by Perez:

"New actors, usually young, burst into action shaking a firmly established and complacent world. Investment in the new industries is carried out by new entrepreneurs while the young financial tycoons create a whirlpool that sucks in huge amounts of the world's wealth to reallocate it in more adventurous or reckless hands: …When the financial down comes, the party is over and the time comes for analysing what went wrong and how it can be prevented from happening again. Though the debate about the causes and the culprits can go on forever, the more practical task of setting up an adequate regulatory system and a set of effective safeguards is soon undertaken. Thanks to the crash and the recession, there is a newfound readiness to accept such rules on the part of the – until recently arrogant – financial wizards, now sobered up."

(Perez, 2002, pp. 5-6)

This author suggests that social stability does not directly follow but may do after a period of time. Disruptive technologies may destroy old industries, put people out of work and create a breakdown in social structure c.f. coal mining in the UK as a source of energy. This point is taken up by Schumpeter when he states that

"Introducing a new thing is difficult as one of the ways the environment resists is physical attack the man who tries to produce it. Following the embedding and acceptance of the new technology the social stability settles to a new norm. This norm may take many years to come about; perhaps waiting until those who resisted die of natural causes. Some historical developments linger: the “South” in the USA, colonialism in ex-colonies etc.”

(Schumpeter, 2010, p. 117)

This thesis suggests that some development of knowledge may contribute towards progress which turns out to be a historical cul-de-sac, not just for technologies (e.g. Betamax video) but also for entire countries and peoples. Kuhn suggests that when the paradigm becomes inadequate and anomalies arise then “nature has somehow violated the paradigm induced expectations that govern normal science”

and that a paradigm shift is when:

“Normal science is being turned on its head and an era of ‘extraordinary science’ is being ushered in. “ (Kuhn, 1996, pp. 52-53, 91)

This is defined by Kuhn as a scientific revolution within positivism. This thesis should play a small part in an Information Technology (IT) Security revolution.

This thesis asserts that the “tech bubbles” are the result of scientific revolutions coming to market with the market trying to make sense of, and capitalise on, the scientific revolutions that were discovered perhaps many years before (in the case of computers developed in the 1940s coming to business in the 1960s with mainframe computers and the wider population with desktop PCs hosting for example Microsoft OS in the 1980s). Furthermore, the thesis has asserted, and will further assert, and support with evidence, that this rush to market by the new IT industrial revolution has led to a lack of good QC which in turn has led to opportunities for APTs to develop and thrive.

In the case of the current “IT Revolution”, this lack of good QC has led to an opening for Advanced Persistent Threats to come into existence and thrive. However, as shall be seen later, this lack of good QC also applies to APTs and allows defenders to gain insight into the modus operandi of APTs.

The lack of recent good QC may have its roots in the “move fast and break things” IT development culture, documented as first being heard in public in 2005 (Sutton and Rao, 2014, p. 308) but not reaching the wider public until 2008 (Google-Trends, 2019). This “move fast and break things” culture was a reaction to the problem of delivery of IT projects. As computers became more mainstream it was recognised that computer projects were overrunning on budget and time as well as needing the right quality (Haughey, 2014) (Larman and Basili, 2003). However poor project management is not a recent problem (Seymour and Hussein, 2014).

The IT Security industry has grown to meet the threat and issues through specialised IT Security or Cyber Security companies. Larger companies have developed their own IT Security sub-divisions. Many anti-virus (AV) companies have grown from entrepreneurial beginnings and produce white papers and blogs discussing APTs, as evidenced by F-Secure (F-Secure, 2017), Symantec (Symantec, 2017) and Trend Micro (Trend-Micro, 2017a). There is also a world-wide academic requirement for research in this area (Marchetti *et al.*, 2016), (Friedberg *et al.*, 2015) and (Banna , Singh and Samsudin, 2015).

4.3 IT Security - A Global Concern

4.3.1 History

The problem of illegal manipulation of IT assets is not new, with a history dating back to at least 1973. Although no examples could be found earlier than 1973, an abstract of a report issued by the US Government on the guidance on controls over data processing equipment in disbursing operations indicates that thought was being given to the problem of computer security in the 1960s (US-Dept-of-Defense, 1965).

In 1973 the Chief Teller at the Park Avenue, New York, branch of the Union Dime Savings Bank (Fosburgh, 1973, p. 1) was able to embezzle $1.5 million by manipulating hundreds of individual accounts on the bank’s computer. The publicised details are vague but it seems that the teller removed money from accounts and ensured that it was redeposited when quarterly interest payments were due. In addition, the computer was manipulated so that records always contained the correct figures. One can infer that a process surrounding the quarterly interest payment was the method of governance to monitor issues at the account level and the Chief Teller was able to abuse his governance position.

In the late 1980s a true-life popular science story on hacking emerged: in earlier days of computing some organisations charged computer users for use. These charges were at the departmental or individual level and the IT department was paid out of the user's IT budget. At Lawrence Berkeley National Laboratory in the US, Clifford Stoll, a PhD student and systems administrator, discovered and tracked a hacker after he, Stoll, discovered a 75-cent financial discrepancy on a computer log (Stoll, 1988).

These two examples, 15 years apart, demonstrate the two methods of computer, or cyber, crime: the first is a manipulation of the business process, including governance systems; the second is a technical "hacking" attack which is, perhaps, the idea of cybercrime more common in the public consciousness.

APTs attacking computer systems are regularly in the news (Barrett, 2017). The idea of computer hacking, or what is now referred to in the cyber security (IT Security) profession, as Advanced Persistent Threat, may have first come to the public's wider attention as entertainment with the popular films "The Italian Job" (Collinson, 1969) and "War Games" (Badham, 1983). In the former the character played by Benny Hill, Professor Simon Peach, is able to gain physical access to the computer controlling the Turin traffic system and replace a reel-to-reel magnetic tape containing legitimate software with a magnetic tape containing illegitimate software. In the later, the character played by Matthew Broderick, David, is able to gain access, and complete control, through a software back door and password guessing, of a computer which controls the US military's nuclear arsenal.

However, philosophically, computer hacking techniques are not new and have their origins in the non-IT world. In 16th century England, Sir Francis Walsingham was Queen Elizabeth I's minister in charge of espionage and Thomas Phelippes, one of Walsingham's most confidential assistants, uncovered a conspiracy to overthrow Elizabeth I by followers of Mary Queen of Scots. All messages between Mary and her supporters, which were encrypted, were copied and decrypted by Phelippes who, unknown to Mary and her supporters, added a forged post script to one message to obtain the names of Mary's courtiers who were to assassinate Elizabeth (Kahn, 1973, pp. 86-90). This is an example of a man-in-the-middle attack, later to evolve as a man-in the-browser attack, as well as signature forging which can be done with fake public key computer certificates. All of these will be discussed later in this thesis.

In 1834 two brothers, François and Joseph Blanc, who were bankers in Bordeaux, subverted the French Government's telegraph semaphore network. News of the French stock market could take several days to reach Bordeaux. The Blanc brothers bribed an operator to insert a backspace, which deleted the previous character, into a message. The spurious character, which was not written down on arrival, indicated the previous day's movement in the French stock exchange. An accomplice, who was a former telegraph operator, viewed the spurious character outside Bordeaux through a telescope (Standage, 2017) and passed the information to the brothers Blanc. Manipulation of data streams will be discussed later in this thesis.

These two examples demonstrate that it does not take long for something new to be abused and in this regard the internet is nothing new: On June 23th 1989 3.26.53pm Australia was connected to the internet via Hawaii. Within eight months two young Australians had been arrested by the Australian Federal Police (AFP) for hacking (ABC, 2017).

It is clear, therefore, that technology and processes can be exploited for uses other than what they were intended and that some form of governance is needed to re-enforce technological security.

4.3.2 International and National Governance

It has been demonstrated earlier (Perez, 2002, pp. 5-6) that regulatory oversight comes after a technological or financial crash and this oversight comes from the United Nations (UN). Part of the preamble to The United Nations' Charter states that:

"We the peoples of the United Nations determined … to establish conditions under which justice and respect for the obligations arising from treaties and other sources of international law can be maintained …"

(United-Nations, 1945)

Also, under Article 10 of the UN Charter The General Assembly may discuss questions or matters with the scope of the Charter and may make recommendations to the Security Council or members of the UN (United-Nations, 1945).

There are five UN General Assembly resolutions related to cybersecurity (United-Nations, 2018) the most recent of which, in 2010, was to take stock of national efforts to protect Critical National Infrastructure (CNI) which the UK government officially defines as:

> "Those critical elements of infrastructure (namely assets, facilities, systems, networks or processes and the essential workers that operate and facilitate them), the loss or compromise of which could result in:
>
> a) Major detrimental impact on the availability, integrity or delivery of essential services – including those services whose integrity, if compromised, could result in significant loss of life or casualties – taking into account significant economic or social impacts; and/or
>
> b) Significant impact on national security, national defence, or the functioning of the state."

(CPNI, 2018)

The 2010 UN Resolution builds on the January 2003 UN Resolution to create a global culture of cybersecurity which itself builds on the call to Member States:

" … to promote at multilateral levels the consideration of existing and potential threats in the field of information security;"

(United-Nations, 1998)

This thesis suggests that what is good for critical infrastructure is also good for wider business. Therefore, CPNI's definition could be extended to the wider business as:

> "Those critical elements of business (namely assets, facilities, systems, networks or processes and the essential employees that operate and facilitate them), the loss or compromise of which could result in major detrimental impact on the confidentiality, availability, integrity of the business's data, information, intellectual property or delivery of the business – including those aspects whose confidentiality, availability and integrity, if compromised, could result in significant business loss, both financial and physical, including reputational and emotional damage."

4.3.3 The Size of the Computer Hacking Problem

Before discussing the size of the problem, it is necessary to define the problem. In UK law the Computer Misuse Act 1990 is defined as:

> "An Act to make provision for securing computer material against unauthorised

(HMG, 1990)

This thesis interprets hacking as "unauthorised access or modification". As telecommunication equipment is IT based this Act, therefore, applies not just the traditional idea of a desktop computer but all aspects of electronic communication, processing and storage. This thesis further suggests that computer hacking is the manifestation of a stated intent or desire in the form of an action to affect a computer or computer system.

In the US the modern definition of a Cyber Attack is:

"An attack, via cyberspace, targeting an enterprise's use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information"

(NIST, 2013, p. 58).

Having defined the problem, this thesis will now turn to the size of the problem.

The UK's National Crime Agency (NCA) states that the cost to the UK economy of cyber-crime is billions of pounds a year and growing (NCA, 2016, p. 3) and is such a well-recognised problem that the UK government has seen fit to invest £1.9bn to fight cyber attacks (Reuters, 2016). In 2016, research by Ponemon sponsored by IBM derived from of 383 companies, found that the average total cost of a data breach had increased from $3.79 million in 2015 to $4 million in 2016. There was also an increase for the average cost paid for each lost or stolen record containing sensitive and confidential information from $154 in 2015 to $158 in 2016 (Ponemon, 2016, p. 1) and the $141 and $146 in 2017 and 2018 respectively The total cost of a data breach was $3.62M in 2017 and $3.86M in 2018 (Ponemon, 2018, p. 10)

Ponemon define a compromised record as

> "… one that identifies the individual whose information has been lost or stolen in a data breach."

(Ponemon, 2016, p. 1)

Clearly this is the cost of the loss of personal records to a company, not a cost to those whose records have been lost. i.e. the cost to wider society in rectifying the problems caused by the data loss. Ponemon's definition does not explicitly consider intellectual property which is discussed by Deloitte:

"Valuation of both the impact of the stolen IP and the lost contract employs the following generally accepted principles:

- **The with-and-without method.** This approach estimates the value of an asset after an attack, compared with its value in the absence of the theft. The difference is the value of the impact attributed to the incident.

- **Present value of future benefits (and costs).** To calculate an asset's projected benefits while accounting for the time value of money, the cost is associated with the specific point in time at which the attack is discovered.

- **Industry benchmark assumptions.** Typical industry benchmarks are used to arrive at the value or financial impact associated with various assets. Examples include royalty rates for the licensing of technology or trade name.”

(Mossburg, Fancher and John, 2016, pp. 106 - 121)

This thesis asserts that it is difficult to assess the cost of lost information, opportunity cost, disciplinary action against internal culprits, rebuilding databases or files, adding new security software, and adding extra security procedures. Reputational cost is, for practical purposes, immeasurable.

Accenture group the range of costs into two sets: the first by internal cost activity centre for the attack (Discovery, Investigation, Containment and Recovery); the second by business function (Cost of information loss or theft; cost of business disruption and Cost of equipment damage) (Accenture, 2019, pp. 34-35). This thesis will later discuss Cyber Kill Chains but post-thesis development could overlay the internal cost centre on the Cyber Kill Chain to have a structured approach to the Kill Chain, including the Courses of Action matrix i.e. a “Cyber Recovery Chain”.

This view of a range of costs is also held elsewhere (Center-for-Strategic-and-International-Studies, 2013, p. 16) where it is suggested a financial range (rather than a specific monetary figure) should be used. A theme also followed by the UK Government (HMG, 2019a, pp. 51-54). However, not all such reports consider all aspects of the cost of doing business. In a work commissioned by the UK Government the authors do not include, with no reason given, reputational costs in their report (Anderson *et al.*, 2014, p. 269).

The problem is not just about monetary and reputational value - it may directly affect lives. In the US medical field, it was reported to the Information Security and Privacy Advisory Board that medical devices could become infected with malware to the point where they cannot provide the information needed to deliver care. They also stated that device vendors say they are 510 certified and so they cannot patch the devices or put firewalls on them. Furthermore, after being cleaned and re-attached to the network, devices become re-infected in about 10-12 days with some devices using older operating systems being re-infected in less than a day (Information-Security-and-Privacy-Advisory-Board, 2012, p. 8 of 18). The 510 certification relates to US Federal Food, Drug, and Cosmetic Act (US-Government, 2014) summarised by the US Food and drug Administration:

> "… or reintroduce a device that will be significantly changed or modified to the extent that its safety or effectiveness could be affected."

(FDA, 2020)

Although the statute relates to "knowingly and intentionally" … "… causing serious adverse health consequences or death …" the penalty is a maximum of not more than 20 years imprisonment or a fine of up to $1,000,000 (US-Government, 2014, p. 23). This would be sufficient deterrent even to those who, with good faith, updated the software of medical devices but who would probably know that much software is faulty.

Fireeye state that in a world-wide proof-of-value (PoV) trials made up of 1,216 organisations from October 2013 to March 2014, 97% of the organisations were breached with three quarters of all systems running active Command & Control sessions (Fireeye and Mandiant, 2014, pp. 7,9-10). The top six AV vendors, the majority, missed 62% of the malware at the time of FireEye detection. In addition a quarter of the malware was not detected by any of those vendors. (Fireeye and Mandiant, 2014, p. 11). Cisco claim that 93% of websites accessed by customers contain malware (CISCO, 2016). This figure is consistent with a trial bot infection where the attacker retained 108 out of 110 cryptocurrency miners overnight (Barysevich, Moriuchi and Hatheway, 2017, p. 9) and is comparable to research by Check Point which showed that in 75% of organisations scanned a host accessed a malicious website (Check-Point, 2013, pp. 11-12).

Villeneuve (Villeneuve, 2011, p. 13) asserts that many of the attacks are simple and have simple remediation. In many cases, the persistence mechanism consists of using OS program execution techniques e.g. the Windows Startup folder, Windows Registry Run keys or Windows Service installation. Villeneuve goes on to say that Mandiant found that 97% of the targeted malware they analysed used these mechanisms.

This re-enforces the view that the same issues come up time after time, year after year, as highlighted by SANS (SANS, 2011) and this is compounded with assertions that anti-virus techniques are still not working:

> "These anti-virus are clearly not working. …. We [Malwarebytes] started off as a remediation product. Meaning we would clean up the mess caused by the other anti-viruses. … I had an anti-virus when I first got infected. One of the big ones that you've heard of and it didn't solve the problem. It didn't prevent the problem and it couldn't solve it after"

(Financial-Times, 2018).

In nearly all of Mandiant's investigations, the user's anti-virus software failed to hinder [the APT], despite "the tool's wide reach and reputation". The APT would typically modify and recompile source code to evade detection. (Mandiant, 2015a, p. 18). The lack of efficacy is acknowledged in the business world - an organisational survey quoted by Mandiant states that approximately 88% of organisations say their threat-hunting programs need to be improved (CrowdStrike, 2016, p. 8). This improvement should not be difficult and Villeneuve describes how 97% of malware is actioned, above, and evidence from Verizon states that most of the malware may be easy to identify: "96% of attacks were not highly difficult" and that "97% of incidents were avoidable through simple intermediate controls" (Verizon, 2012, p. 3). It is reassuring that that both Verizon and Fireeye and Mandiant both use figures in the 90% range. However, O'Connor offers a view of why attacks may be successful:

> "Briefly consider the implications: this particular malware existed for several years on a public website and antivirus vendors did not write a signature for it. Understanding this, we begin to realize the lack of usefulness in most antivirus vendor products. Instead of examining the indicators of malicious activity (for example, logging keystrokes, turning on the webcam, installing a persistence

> module), most antivirus programs use a sequence of bytes or an MD5 hash to identify a malicious program.”

(O’Connor, 2014, p. 17)

A signature, as an Indicator of Compromise, may be useless when malware is “Living Off the Land” i.e. using legitimate software which is combined in such a way to produce malicious intent, as will be discussed later in the thesis. O’Connor’s view is consistent with the ideas behind this thesis which will be discussed later.

Given the lack of AV software efficacy it is perhaps not surprising that APTs can successfully mount and maintain an attack. Clearly malware can reside on machines for a long period of time. The research of Jacoby and Jartelius (Jacoby and Jartelius, 2013, p. 4), based on an undisclosed sample size, states that it takes companies about 60-70 days to fix a vulnerability and in another attack attackers were able to maintain persistent control and that the average length of compromise was 145 days, with the longest infection being 660 days (Villeneuve, 2011, p. 5). This thesis will return to the subject of length of compromise.

The problem is not just related to single machine infection and intent for direct financial gain. For just one piece of botnet malware with an estimated 1,000,000 nodes on the bot, Sophos estimated that there was 895 Terabytes of network traffic per month, before file downloads, and lost computing cycles i.e. a larger electricity bill through use of more computing cycles for running the bot and cooling the machines (Wyke, 2012b, p. 46).

This use of resources for purposes other than an attack against the host organisation may seem trivial however but some APTs use victims’ machines for their computing power for cryptomining and, as highlighted above, the cost of electricity for powering and cooling of the hardware is not trivial. An estimate of the cost of electricity of using malware to mine cryptocurrencies can be ascertained (F-Secure-Labs, 2015c), (Marosi, 2016). Although the malware may not steal information, IP personal or otherwise, cryptomining does steal electricity and the cost in commodity hardware of bitcoin mining now exceeds the value of the rewards and that in 2014 the electricity consumption for mining bitcoin is similar to that of the electricity consumption of Ireland (O'Dwyer and Malone, 2014).

Clearly this is non-trivial and it can be inferred from this that illegal bitcoin mining on a network can exceed the capital investment and take a large proportion of the running costs of that network. Comparative costs which demonstrate the most profitable cryptocurrency, from a basket of eight, are given in a mining calculator which also highlights that profitability is directly related to the time the malware is undetected on the victims' machines (Barysevich, Moriuchi and Hatheway, 2017, p. 8). Cryptocurrency mining may be embedded in a website and hence use a victim's CPU – a method which may lock the victim's browser and drain the battery of the victim's device. It is not that the mining algorithm is not as efficient as custom made ASIC mining chips (Insikt-Group, 2018b, pp. 18-20) a subject to which this thesis will return when discussing mitigations to APTs and malware.

All of the above discuss the size and cost of the problem but, this thesis suggests, the problem may get worse. Although good IT Security will close off some avenues of attack, the barriers to entry for computer crime are becoming lower. Exploit kits, first seen in 2006 and which offer "Crimeware-as-a-Service", have allowed people with less coding experience and skill levels to gain access to systems (Kuczma and Manalo, 2020, p. 10).

4.4 Space and The Internet of Things (IoT)

A set of international standards is being developed at NASA's Space Communications and Navigation (SCaN) program office. These standards are collectively known as Disruption Tolerant Networking (DTN) standards and will support internetworking in space. The standards are designed for a secure, reliable network service similar to TCP/IP, but implemented differently. The difference is that distances in space are so vast and end-to-end paths may not be available. An example of this is when a space vehicle in orbit receives data from Earth and then needs to wait before it can forward it to a lander on a planet. (NASA, 2017). One can see how there might be "two internets", one on earth and one in space, with a need for a gateway between them or the DTN might find a use on earth when data cannot be forwarded in real time. This too may have vulnerabilities.

4.5 The Cyber Security and Anti-Virus Market Size

In 2018 the UK Government estimated the 2015/16 UK total for cyber security revenue at £5.7bn (to the nearest £100 million) (RSM and CSIT, 2018, p. 22). The size of the global cyber security market is expected to grow from $137.63Bn in 2017 to $248.26Bn in 2023 (statista, 2019d) i.e. an annualised growth rate of over 10%. At £1.3 to the US dollar this translates to a UK share of the global market of over 5.3%.

The market share for the 10 leading anti-malware application vendors sums to 77.14% (statista, 2019b). Another report provides similar data from the same top ten AV companies (but in a different order and market share) with a total of 77.57% (OPSWAT, 2019). These concentration ratios indicate that almost 25% of the market is held by companies which each hold around 1% of the market. This market is expected to shrink in terms of revenue from US$ 3770 million in 2019 to US$3500 million in 2024. Clearly there is a data conflict: The evidence in the previous paragraph provides a view on the global cyber security market yet the AV market share (which surely must be a constituent part of the global security market) is $3.77Bn. Appendix A analysis is based on desktops while the other references just refer to Windows. To borrow a phrase: There is "Plenty of Room at the Bottom" (Feynman, 1959).

How might AV companies provide a service in this market? McAfee's Chief Technology Officer states (NPR, 2019b) that McAfee has a billion sensors deployed around the world which allows McAfee to see new threats and track them. MalwareBytes similarly claim (Malwarebytes, 2018a, p. 3) that they have intel and statistics from their Intelligence, Research, and Data Science teams along with telemetry from their consumer and business products. Their software is deployed on millions of machines.

We may infer from this that McAfee takes the business view that each machine which hosts McAfee AV software is not only performing AV work for the customer but is part of a wider network which will pass back to McAfee new threats. It can be further inferred that this could include date and time of possible infection allowing McAfee to see the evolution and deployment of the threat, rate of infections etc. On the one hand this helps all customers but on the other it gives access to machines by an AV company which has security consequences for the customers – possible unrestricted access to every file on the machine. There has been an example of one commercial AV company being breached (Kaspersky, 2015b) and it is known that files are uploaded to the cloud and analysed using big data. Another APT loaded information to a file hosting service (Insikt-Group and Rapid7, 2019, pp. 2,8,21-23,26).

McAfee's privacy notice states that:

"In order to provide you our Services, including to detect and evaluate malware and spam, we may scan, collect, and store data from your files, including emails, attachments, email addresses, metadata, and URLs and traffic data."

(McAfee, 2019)

This means that a customer's Intellectual Property may find its way into the repositories of this AV company. Malwarebytes states something similar (Malwarebytes, 2019). From this one could infer that other AV companies do something similar. While not suggesting that AV companies would directly profit from IP acquisition, potential AV customers should consider the origin of AV software. Cyber security professionals might consider if it is a good idea to buy AV software from a geographic area that may host business competitors.

It has been demonstrated that there are gaps in the global cyber security market as AV products, individually and in aggregate, do not fulfil their promise. There is also a large enough market for a start-up to fill these gaps in a multi-billion-dollar market. To gain such a market share, this thesis asserts that the first thing to do is review APTs and their modus operandi, Before getting into the details of malware it is necessary to review how APTs do their business and this may be brought into focus by a review of CKCs as stated in the thesis objectives.

4.6 The Cost of Software Development and IT Security

IT projects have been notoriously difficult to cost and manage with some ending as failures after much time and money has been spent (Charette, 2005). The first methodologies were based on physical production process: simply take the number of widgets needed and divide by cost for, or time to make, each widget. (Putnam, 1978). Brooks helped to expose this mythology and claims for several years to have used the rule of thumb for a software task as: "⅓ planning; ⅙ coding; ¼ component test and early system test; ¼ system test, all components in hand.".

(Brooks, 1995)

While this may be an acceptable technique for a relative view of software production there is no absoluteness to this – how does one cost or price the work? ⅓, ⅙, ¼ must be ⅓, ⅙, ¼ of something. Anyone who approaches their management board with this as a cost is surely going to be asked what is the currency value of their work? In addition, Brooks does not address documentation and nowhere in his book does he address the full life-cycle costs. It is not for want of trying that the IT industry has had problems with costs, as highlighted by one survey (Boehm, Abts and Chulani, 2000).

This is compounded by the cost of business in different areas of the world. For example, the World Bank states:

"The most popular reform is making it easier to start a business. More than a quarter of economies did just that in 2017/18. It now takes an average of 20 days and costs 23% of income per capita to start a business, compared to 47 days and 76% of income per capita in 2006.".

(World-Bank, 2018)

Where it can be seen that costs are measured in comparative terms per country rather than in absolute terms of a major currency or currency of the host's country.

4.7 Costing the Technical Business Model

It has been shown that malware writers use AV sites to test and hone their malware (Zetter, 2014). Such sites may be used to help with defence. At the front end use may be made of AV (Fisher 2019), (Raymond.cc, 2019), (Comodo Security Solutions, 2019).

The Top500 supercomputer list (Top500, 2019b) provides evidence of the growth of compute power. One may view the list in various ways but the Performance Development graph gives an idea of progression over the years in terms of Flops (floating point operations per second). For example, the number one machine in 1994 would not make the Top 500 in 2001 and this is a broad trend through subsequent years: in any given year the number one does not make the Top 500 about 7-9 years later.

One could calculate the cost (CPU time, electricity etc.) for a given attack and then make predictions about when that attack would be feasible based on the Top 500 list. It is not just about one particular machine but also a network of Top 500 machines.

Historically, calculating the number of flops was the number of floating-point operations per second multiplied by cycles per second but working out comparative speeds has become more difficult. Modern machines may have levels of parallelism, multiple silicon devices per nodes (socket) multiple cores, multiple threads per core as well as more sophisticated instructions (Dolbeau, 2018). Perhaps, for the purpose of this thesis, there should not be an attempt to calculate the maximum number of flops but the calculation should be the minimum number. This calculation should also include test data. For a 2Ghz CPU with an estimation of one Flop per cycle then there are 2GFlops. Such a machine would be number 500 in June 1995 but is now an affordable desktop or laptop. A business with 100 such machines would have the equivalent of the world’s number one supercomputer in mid-to-late 1993.

However, the best measure may just be to run programs on a sample of data and cost the whole work by multiplying up. Apart from normal operating system functions, the programs had sole use of the 1.7GHz machine on which it was running. This device has two cores each of which can run two threads (Intel, 2019a). In addition, each program in the thesis outputs the start and end time of the program. Form the individual timings a reasonable estimation could be made of longer runs. For example, the work for the search programs divided the disc size by 10 and the first decile run and costed (both machines had 460GB discs). One of the search programs ran on a faster machine (Intel i5-4690 3.5GHz). This device has four cores each of which can run four threads (Intel, 2019b) and the elapsed run time was four times faster (roughly one hour down from four hours). A doubling of each of: the processor speed; number of cores; and number of threads; would suggest an elapsed time difference of a factor of eight but the difference of a factor of four gives weight to Dolbeau's observations, above, about the difficulty of working out comparative differences.

On a much smaller scale a "supercomputer" can be constructed from a network of machines by dividing up the work into manageable packages in the manner of the seti@home (University-of-Berkeley, 2019) project so an attack could be produced by linking various universities' networks, for example. Microsoft provide guidance for a proof of concept cluster (Microsoft, 2013). Assuming that a business has hardware that satisfies the recommend specifications (2GHz or faster CPU, 2GB or more RAM and 80GB or more of disc space) then the capital costs are zero as the hardware costs are sunk costs. The costs become setup costs and running cost. Other software options are available e.g. Hadoop (Apache, 2019), Beowulf (Beowulf.org, 2019). One hardware option is a Raspberry Pi cluster (Los-Alamos-National-Laboratory, 2017).

Dividing the work into deciles gives the added benefit of a form of check pointing: should any job fail for any reason then the results up to that decile of data are preserved. Simple arithmetic demonstrates that a search run on the faster machine reduces the elapsed time from 40 hours to 10 hours. Further dividing the work space across 100 of the faster speed machines will reduce the elapsed search time to .1 hours or six minutes allowing many up to 240 such searches to be perfomed.in a calendar day. Another option for the search process is a series of FPGAs and at least one product uses USB 3.0 (Xilinx, 2019) which can transfer data at up to 625 MB per second but, again, the speed calculations are difficult (Strenski *et al.*, 2008). A 460GB disc would take a minimum of 1.3 hours when the data transfer rate is 100MB or just under 15 minutes at the theoretical maximum.

4.8 A Cyber Kill Chain Model

4.8.1 Overview

This section examines the origins of the term "Cyber Kill Chain"; it will define a Cyber Kill Chain (CKC), review a selection of CKCs and select one, the Lockheed Martin APT Kill Chain, for the thesis. A tabular comparison of some CKCs will be produced.

Many Kill Chains follow similar designs. The thesis asserts that the Lockheed Martin Kill Chain is the seminal design for cyber or IT Security and its intellectual paternity can be seen in other Kill Chains, as will be demonstrated.

A Google Trends search on "Cyber Kill Chain" (Google-Trends, 2020) from 2004 to 2020 shows some interest November 2009 and September 2010 and then almost constant, rising, interest from November 2011. However, as shall be demonstrated, the groundwork had been laid by the USAF in the late 20th century.

The term "Kill Chain" has military origins from a speech given to a USAF symposium by Gen Ronald R. Fogleman, the then Air Force Chief of Staff, in which he said:

" … support … the Joint Vision by recognizing the reality that in the first quarter of the 21st century, it will become possible to find, fix or track, and target anything that moves on the surface of the earth."

(Fogleman, 1996)

The prediction was elaborated on to include "engage" and "assess" under the acronym F2T2EA (Tirpak, 2008) and within this Tirpak states that "fix" is a part of what is sometimes called the "kill chain". The [Offensive] Kill Chain is summarised, again within the USAF as:

"Typical elements of the kill chain include target reconnaissance, detection, identification, tracking, decision, and order to attack the target, destruction of the target, assessment, and reporting.“

(Contratto, 2012, pp. 2, 26).

This thesis has earlier discussed the genesis and evolution of the term APT. This thesis notes that, again, this work was within the USAF during the same period i.e. 2006-2012, based on turn of the 21st Century ideas. This author asserts that a Cyber, or Intrusion, Kill Chain is merely a manifestation of the USAF F2T2EA Kill Chain as applied to IT Security.

A Cyber Kill Chain allows the defenders, attackers and academics to logically view the APT attack scenario and logically work through it. The main body of this thesis will be organised accordingly. This organisation of the thesis is a valid form of coding (Coffey and Atkinson, 1996).The code tags used in the thesis will be the individual pieces of malware: for example, Domain Name Service (DNS) issues, polymorphic issues. By grouping the data in this way, this thesis can segment the attacks primarily aligned with the chosen kill chain and secondarily under which individual pieces of malware are being discussed. This also provides the thesis with the opportunity to be consistent across AV companies' results, highlighting overlap and gaps in published knowledge of malware families. From here the author can ask "What if ...?" and "Why?" In some ways searching for malware is like working out a Hidden Markov Model – outcomes are observable and there are fragments (clues) about the underlying event. The minimum model, Ockham’s Razor view, to describe what has been seen may then be constructed.

This thesis will now try to construct that minimum model.

4.8.2 The Business View

Kaspersky in Smeets state: “[t]he ‘ecosystem’ of malware breaks down into known threats (70%), unknown threats (29%) and advanced threats (1%).” (Smeets, Undated). However, this is too simplistic: this thesis asserts that this description is confusing the technical abilities of attackers with their business sense. As has discussed earlier in this thesis it is sensible to deploy the minimum technical attack against a potential victim. This thesis further asserts that a true APT will deploy low level attacks to appear to be something else. Another problem with the Kaspersky view is that there are known and unknown threats i.e. it is a binary choice and so there cannot be the third outcome of advanced threats. Additionally, threats may be advanced or not.

Even APTs should apply the philosophy of Ockham's razor to an attack. A low-level attack may be devised, built and actioned by an APT but a high-level technical attack by a non-APT may not. Why would, or should, an attacker deploy their most technical attack, on which they may have spent much time and money, against an easy target?

Smeets takes up this issue of the business aspects of malware development:

> "First, weapon development requires *know-how*; the knowledge required to design and acquire the weapon. Second, there are the *material* and *economic* costs. Third, weapon development often requires an *organizational structure* as various actors have to work together to develop a certain capability."

Indeed, for well-funded and resourced APTs one can imagine it being a business with associated business controls and economies of scale and economies of scope. This would need to be well-managed, perhaps with the victims being viewed as customers with associated attributes. This is also an idea touched on, but not further developed, with respect to Customer Relationship Management (CRM) (Wrolstad and Vengerik, 2015, p. 11) who speculate that web analytics tools may be used. This thesis suggests that a good, well-organised, well-led and disciplined team of attackers should have a list (database) of every attack under development, every attack deployed, a unique serial number for each piece of malware and for every attack as well as every victim.

What has just been described is a business flow for the APT. This may be formalised into a CKC. This thesis will now discuss what a CKC should look like.

4.8.3 What Should a Kill Chain Look Like?

This thesis asserts that a Kill Chain model should cover every part of the attack lifecycle.

Velazquez asserts that

> “Detecting and preventing attacks earlier in the kill chain is critical in defending against cyber threats. By implementing defence layers to detect and block attacks earlier in the kill chain organizations decreases the amount of remediation that needs to be performed by the security team.”

(Velazquez, 2017, p. 17)

Velazquez provides no evidence for this and this thesis argues that all stages should cover detection. Remediation is only needed if an attacker is on the system. No remediation is needed is the attacker has been blocked. Velazquez provides no evidence for the assertion. Arguably, placing mitigations at the gateway stops malware but there are the associated costs of software blocking, monitoring logs, and looking for changes in probes to the defences.

Pols builds a Unified Kill Chain (UKC) from the action of one APT by overlaying the APT’s actions over the Lockheed Martin Intrusion Kill Chain. The UKC has three main stages (Initial Foothold, Network Propagation, Actions on Objectives) and subsequent sub-stages (Pols, 2017, p. 78). For completeness Pols’ work is included in the Kill Chain comparison table.

Pols (Pols, 2017, p. 7) highlights criticisms of the Lockheed Martin Intrusion Kill Chain by Engel

> “As sexy as it is, the Cyber Kill Chain model can actually be detrimental to network security because it reinforces old-school, perimeter-focused, malware-prevention thinking.”

(Engel, 2014)

Engel goes on to confuse method of delivery with malware. This is similar to the mixed use of APT as a delivery team and APT as software which has been discussed earlier in this thesis. Furthermore, Engel does not acknowledge the LMKC “Courses of Action Matrix” (Hutchins, Clopperty and Amin, 2011, p. 5) but Pols does (Pols, 2017, p. 20). This thesis has found no evidence for a scenario where the Lockheed Martin Intrusion Kill Chain does not describe the attack and asserts that the Lockheed Martin Intrusion Kill Chain is the best available model from those reviewed into which all cyber-attacks may be fitted and mitigated. Not all attacks need to be fitted into the model at once and not every cyber-attack needs to be deployed against it all mitigations in the mitigation universe. The Lockheed Martin Intrusion Kill Chain is a model into which the defender inserts selected mitigations based on the Risk Analysis that has been performed. These mitigations may be brought to light by populating Lockheed Martin's "Courses of Action Matrix".

Pols raises another criticism in that insiders are not hackers and that:

"The Intrusion Kill Chain is excellent for attacks, but doesn't exactly work for insider threats."

(Reidy, 2013)

This is simply not true. An insider will perform reconnaissance (How do I not get caught given the security I have seen in place?), Weaponization (build malware), Delivery (Insert USB or type on keyboard, run programs, get data to which I should not have access, access website with known malicious code) etc. The Insider Threat is a Threat. They are Persistent (they attend their place of work and have access to the IT either at work or remotely). They may or not be Advanced but as has been asserted and as will be demonstrated, not all APTs need to be advanced.

4.8.4 A Review of Kill Chains

There appears to be little academic literature on comparisons of CKC. The closest to such a comparison (Cho *et al.*, 2018) diagrammatically depicts five CKCs Chains and references four. The authors chose the Lockheed Martin Intrusion Kill as the most representative without reason and without pointing out its seminal place in the evolution of CKCs, as has been discussed, above. Elsewhere (Grant, Burke and van Heerden, 2012) there is a comparison of seven different attacks which produces three different CKC models. This will be discussed later in this section. It appears that this thesis is one of the first times that a comprehensive comparison has been made.

Given that there appears to be little academic literature on the subject, primary data for the author will be mainly white papers produced by AV companies.

The white papers are the results and analysis (both qualitative and quantitative) of APT attacks which leads this author to suggest that they fall into the class of Retrospective Primary data i.e. compiled by the writer after the event, as described by Brewer (Brewer, 2000, p. Figure 3.2), although this is categorised as a subset within the "Personal documents" theme of Ethnography, they may also fall under "Case studies and generalizations". However, this thesis will draw the generalisation across APT attacks through interpolative and extrapolative thoughts and actions. Arguably academic research papers do not fall into any of the Brewer categories.

The thesis will now briefly discuss TTP which has been introduced earlier in this thesis as a proto-Kill Chain. TTP provides a view of the attack across the whole operation which is different to the Cyber Kill Chain (CKC). For defenders it provides a similar way to defend. For example, a theoretical defence TTP model has been developed (J. N. Stewart, 2014) and has given rise to an example based on an attack (RSA, 2016, p. 2). However, the problem with viewing an attack with the TTP model is that one must consider all three items at once and it is not necessarily linear or temporal. Therefore, using the more simplistic TTP model will be rejected and this thesis will follow the Kill Chain route.

Kadivar suggests that a cyber-attack has at least five attributes (Actors; Assets targeted (sic); Motivation; Effect on targeted assets; Duration) and from an examination of 10 cyber attacks goes on to suggest six additional attributes (Attack vector; Vulnerability; Malicious software; Botnet reliance; Origin; Destination.) (Kadivar, 2014). However, this thesis asserts that these 11 attributes, in two different sets, are incomplete and that the two sets maybe be further refined (Origin, Assets targeted, Vulnerability, Attack vector, Malicious software, Effect on assets targeted, Destination, Duration.) omitting Botnet reliance, Actors, Motivation. A Botnet is a specific type of attack, Actors are the people doing the attack i.e. the APT and Motivation is why the APT is doing it. None of the these three add to Kadivar's CKC.

It is noted that the Lockheed Martin Intrusion Kill Chain which, this thesis will select, precedes Kadivar.

For completeness of the selection review this thesis notes that Grant et al (Grant, Burke and van Heerden, 2012) review seven different attacks and distil them into a model using SADT (Structured Analysis and Design Technique – a meaning not elaborated on in the paper but inferred for this thesis), Rational Reconstruction and a Canonical Model to produce a five-stage model with multiple sub-stages and sub-sub-stages

Fujitsu acknowledge the existence of malware on machines and inherently accept the existence of a CKC but are not explicit about it (Torri *et al.*, 2014). One could argue that they have proposed a 3-stage CKC: Spying, System Research, Breakout. It is a pity that they did not develop this model to make their counter-measure process clearer.

One CKC (Trend-Labs, 2014, p. 2) (Trend-Micro, 2013b, pp. 5-7) has six stages of a Targeted Attack but another Trend Micro CKC has eight stages (Kellermann, 2012, p. 7) while a semi-product brochure has “Compromise” as stage 2 (TrendLabs, 2012a, p. 2). Trend Micro change their model frequently (Trend-Micro, 2017b) (TrendLabs, 2015b, p. 6), (Pernet and Sela, 2015, p. 12) (TrendLabs, 2013b, p. 2) which may have evolved from an earlier model (Villeneuve, 2011, p. 7). It is possible that Trend Micro are, at least partly, fitting CKC models to observed attacks but if they are this in not clear. A consistent CKC, with change management, could be used in their analysis. Deviations from this norm could then be documented. While it is good practice to develop a specific rule from observations it is difficult to see how, if at all, Trend Micro are building a theory from the observations. It is possible that different teams within Trend Micro have their own views and hence CKC models.

Pernet and Sela seem to be trying to fit the TrendLabs model, above. to the specific APT under discussion, which some other AV companies seem to be trying to do. To be fair, all of the above is consistent with what we shall see of the philosophy of the Lockheed Martin Intrusion Kill Chain of an informed analysis of adversary campaigns and therefore an evolving model but at some point, a model should be fixed.

Symantec observed one APT with a 6-stage architecture:

- Stage 0 - Dropper. Installs onto the target computer;
- Stage 1 – Loads driver, decrypts and executes next stage;
- Stage 2 – Loads driver, decrypts and executes next stage;
- Stage 3 – Kernel Framework;
- Stage 4 - User Framework and Kernel Modules;
- Stage 5 – Payload Models.

"The initial Stage 1 driver is the only plainly visible code on the computer. All other stages are stored as encrypted data blobs, as a file, or within a non-traditional file storage area such as the registry, extended attributes, or raw sectors at the end of disk." (Symantec-Security-Response, 2015c, pp. 8-11) (n.b. This method clashes with Symantec's observation of another APT using the end of the disk to help support its attack.. Symantec go on to suggest that "The dropper is quite likely built into the infection vector exploit code and is never written to disk.", while Stage 1 may be registered as a system service, Stage 2 "is encrypted within an extended attribute or a registry key blob".

The CKC definitions, above, are now summarised and a selection of Kill Chains are cross-referenced in the following table (N.b. An asterisk (*) next to a number indicates that the description in the first column of the table is sufficiently close to that Kill Chain's description but it is not identical):

	Lockheed Martin	Trend Micro (2013b)	Trend-Micro (2017)	TrendLabs (2015b)	Pernet and Sela	MITRE ATT&CK™	Mandiant (2017)	IPA (2013)	Symantec (2011)	Blue Coat (2011)	Forcepoint (2017)	Damballa (2016)	F_Secure (2016)	Assante and Lee (2015)	McKew (2017)	CA Technologies (2014)	LogRhythm (2013)
Reconnaissance	1					1	1*/5*				1				1	1	
Intelligence Gathering		1	1	1	1												
Planning								1						1			
Research										1							
Inception													1				
Weaponization	2					2											
Lure											2						
Redirect											3						
Preparation								2						2			
Point of Entry		2	2	2	2					2*							

	Lockheed Martin	Trend Micro (2013b)	Trend-Micro (2017)	TrendLabs (2015b)	Pernet and Sela	MITRE ATT&CK™	Mandiant (2017)	IPA (2013)	Symantec (2011)	Blue Coat (2011)	Forcepoint (2017)	Damballa (2016)	F_Secure (2016)	Assante and Lee (2015)	McKew (2017)	CA Technologies (2014)	LogRhythm (2013)
Delivery	**3**					**3**											
Incursion									**1**					**3***	**2**		**1***
Initial Compromise							**2**	**3**								**2***	**2***
Droppers												**1**					
Exploitation	**4**					**4**					**4***						
Intrusion													**2**				
Establish Foothold							**3**										
Downloaders												**2**					
Crimeware Installation												**3**					

	Lockheed Martin	Trend Micro (2013b)	Trend-Micro (2017)	TrendLabs (2015b)	Pernet and Sela	MITRE ATT&CK™	Mandiant (2017)	IPA (2013)	Symantec (2011)	Blue Coat (2011)	Forcepoint (2017)	Damballa (2016)	F_Secure (2016)	Assante and Lee (2015)	McKew (2017)	CA Technologies (2014)	LogRhythm (2013)
Attacking Infrastructure Building								**4**									
Dropper File											**5**						
Lateral Movement		**4**	**4**	**4**	**4**												
Penetration/ Exploration								**5**		**3***						**4***	
Escalate Privileges							**4**									**3**	
Installation	**5**																
Infection													**3**				
Asset/Data Delivery					**5**												

	Lockheed Martin	Trend Micro (2013b)	Trend-Micro (2017)	TrendLabs (2015b)	Pernet and Sela	MITRE ATT&CK™	Mandiant (2017)	IPA (2013)	Symantec (2011)	Blue Coat (2011)	Forcepoint (2017)	Damballa (2016)	F_Secure (2016)	Assante and Lee (2015)	McKew (2017)	CA Technologies (2014)	LogRhythm (2013)
Discovery									2						3		
Command & Control	6	3	3	3	3	5					6						
Management & Enablement														4			
Call Home																	
Actions on Objectives	7																
Invasion													4				
Capture									3						4		
Data Exfiltration		6	6	6					4*						5*		3*
Data Theft											7						

	Lockheed Martin	Trend Micro (2013b)	Trend-Micro (2017)	TrendLabs (2015b)	Pernet and Sela	MITRE ATT&CK™	Mandiant (2017)	IPA (2013)	Symantec (2011)	Blue Coat (2011)	Forcepoint (2017)	Damballa (2016)	F_Secure (2016)	Assante and Lee (2015)	McKew (2017)	CA Technologies (2014)	LogRhythm (2013)
Execute						6		6 *									
Harvest										4							
Data of Interest		5															
Maintenance			5	5		7											
Sustainment, Entrenchment, Development & Execution														5			
Re-Infiltration								7									
Complete Mission							6										

Table 4-1 Kill Chain Comparison Table (Selected Comparison)

.:

From the literature and Table 4-1 it can be seen that:

- Lockheed Martin have a 7-stage model (Hutchins, Clopperty and M., 2011);
- IPA have a 7-stage attack Scenario for targeted email attacks: (IPA, 2013, pp. 11-14);
- Trend Micro/Trend Labs have a number of models (Trend-Labs, 2014, p. 2), (Trend-Micro, 2013b, pp. 5-7), (Kellermann, 2012, p. 7), (TrendLabs, 2012a, p. 2), (Trend-Micro, 2017b) (TrendLabs, 2015b, p. 6), (Pernet and Sela, 2015, p. 12), (TrendLabs, 2013b, p. 2), (Villeneuve, 2011, p. 7);
- Grant et al (Grant, Burke and van Heerden, 2012) have a 5-stage model with sub-stages and sub-substages;
- Mandiant's model (Mandiant, 2010, pp. 3, 6) has a 7-stage model. However Mandiant later changes this model (Mandiant, 2017, p. 10) (Mandiant, 2013, p. 27), (Madiant-Consulting, 2016, p. 27), (Mandiant, 2015a, p. 14);
- Symantec uses a four-stage model (Symantec, 2011, pp. 2-6);
- Blue Coat uses a four-stage model (Blue-Coat, 2011, p. 6);
- Forcepoint uses a seven-stage model (Forcepoint-Security_Labs, 2017);
- Damballa have different models for different types of malware: How the Malware Lifecycle Worked in the Past (1-stage); How the Idealized Dropper Lifecycle Works (6 stages); How the Idealized Downloader Lifecycle Works (7 stages); How the

Crimeware Installation Lifecycle Works (12 stages) (Damballa, 2016, pp. 2-7);

- F-Secure offer four stages in their user centric Chain of Compromise: (F-Secure, 2016, pp. 2, 22-25, 28-34)
- RSA suggest a 4-stage model (Maczuba, 2016, p. 2), (RSA, 2016, p. 2) which is a subset of the Lockheed Martin Intrusion Kill Chain;
- Assante and Lee have a 2-stage model based on the Lockheed Martin Intrusion Kill Chain. (Assante and Lee, 2015, pp. 4-12);
- McKew has a five phases model (NPR, 2017);
- Websense have evolved their 3-stage model (websense, 2011, pp. 3-5) through a five-stage model (websense, 2013) to 7-stage model (Clark and Robinson, 2013);
- Imperva has five stages (Imperva, 2014, p. 2);
- EY propose six stages (Miller and Zaveri, 2016, p. 3) as do M86 Security (M86-Security, 2010, p. 3);
- InfoSec Institute propose seven stages (InfoSec-Institute, 2017);
- CA Technologies have a 12-stage model (ca-Technologies, 2014, p. 4);
- LogRhythm have a 3-stage (LogRhythm, 2013, p. 3) and a later 5-stage model (LogRhythm, 2014, pp. 2-4);
- MITRE ATT&CK™ has a 7-stage model (Mitre, 2018) as well as a later 11-stage model (Mitre, 2019a);

McAfee illustrate a 5-stage “Anatomy of a Hack” (McAfee®-Foundstone®-Professional-Services and McAfee-Labs™, 2011, p. 3).And that while all CKCs broadly align, there are some that appear disjoint against the majority

4.8.1 Critical Analysis of The Presented Cyber Kill Chains

The review of CKCs highlighted that many Kill Chain models can trace their lineage directly or indirectly to the Lockheed Martin Intrusion Kill Chain. For example Chen et al (Chen, Desmet and Huygens, 2014) directly reference Hutchins (Hutchins, Clopperty and M., 2011), the Kill Chain phases of which can be seen to be very similar to the Lockheed Martin Intrusion Kill chain.

The point of a Kill Chain is surely in the title; there is a need to "Kill" the malware or at least control and minimise the damage, hence the LMKC "Courses of Action Matrix".

One taxonomy (Kiwiaa *et al.*, 2018) follows the LMKC without an explanation of why this particular Kill Chain was chosen. The taxonomy is very detailed but fails to match this against the LMKC "Courses of Action Matrix" of Detect, Deny, Disrupt, Degrade, Deceive, Destroy against the seven phases of the Kill Chain. This seems to be a common restriction. This taxonomy also claims that Trojans generally consist of two parts: one for the server; and the other for the client. This may be true but both a server and a client are computers and both parts of the Trojan will have mechanisms of persistence, as well as mechanisms of action, and will need to possess similar methods of concealment from defenders as both parts will face the same sort of searches by defenders.

Although there is no reference to the USAF Kill Chain in the Lockheed Martin APT Kill Chain (Hutchins, Clopperty and M., 2011), the lineage is obvious (Lockheed Martin Terms first; USAF terms follow after "-"):

- Reconnaissance - find, fix;
- Weaponization - fix, target;
- Delivery - target, engage;
- Exploitation - engage;
- Installation - engage;
- Command & Control - track;

- Actions on Objectives - engage, assess.

Many CKCs have the LMKC in their ancestry and some appear to have been developed from observation of a single attack without regard for the wider needs and issues. This thesis will now reduce the field of competitors to a final selection.

4.8.2 Moving towards Selection of the Lockheed Martin Kill Chain

In selecting a CKC as a model for this thesis, all models that listed "Exfiltration" as the final step were rejected. This is because data exfiltration might not be the objective. The objective may be a complete take down on the victim's machine or network; it may to use the victim's machine or network as part of a DDOS attack; it may be for ransomware etc.

Therefore, the models of Trend Micro, Mandiant, Symantec, Blue Coat, Force Point, McKew, Imperva, EY, Miller and Zaveri, CA Technologies, LogRhythm Mitre and McAfee were all rejected. In addition, Trend Micro, Damballa and M86 were rejected as they did not have one consistent model, even if they had good ideas within those inconsistencies. Assante and Lee were too heavily based on the Lockheed Martin Model as were RSA and Maczuba and were therefore rejected.

F-Secure have a very clean and memorable Kill Chain, however it concentrates on entry to wider infection without a clear description of what are the outcomes of the attack and so their model was rejected.

The last three candidates are Lockheed Martin, IPA and Grant et al. All were strong, well-documented and thoughtful but the first two concentrated on malware delivery by email (Other attacks for example, manufacturers' password use, were ignored) and the third was designed from an attacker's point of view with a final stage being called "Lessons Learned" which included Unintended Effects, Assess Damage, Evaluate operation, Disseminate LL (Lessons Learned). The Grant et al model was rejected as it was from an APT viewpoint,

Both Lockheed Martin and IPA separate the initial compromise and persistence into two steps but this author feels that any destructive intent could be achieved in one step and would prefer to see initial compromise and persistence in one step or at worst one step with two sub-divisions.

IPA have a seven stage “Attack Plan”:

- Planning;
- Preparation;
- Initial Compromise;
- Attacking Infrastructure Building;
- Penetration/Exploitation;
- Mission Execution;
- Re-Infiltration.

The fourth stage, “Attacking Infrastructure Building”, is too specific. The Attack Plan is directed to towards targeted email attacks but even so, one does not need to attack a building’s infrastructure for a successful attack. However, defenders need to protect that infrastructure as this may be a way into the wider IT system.

IPA have their seventh and final stage as re-infiltration but Lockheed Martin have a very good general final stage as “Actions on Objectives”. This may map to IPA’s statement that Advanced Persistent Threats are a combination of multiple factors but IPA are not explicit. The use of “re-infiltration” by IPA may be a language or translation issue (it is a Japanese model). There is no need for re-infiltration if the malware has not been mitigated and is persistent but the IPA authors may have meant it to be part of a Command & Control stage.

Like Lockheed Martin, IPA generalise the executing phase. As discussed, other models only state exfiltration but the outcome may be to put information onto a machine or network e.g. illegal material that would get the victim removed from their position or worse.

Lockheed Martin's only weakness would be there is no continuous monitoring but one could argue this is inherent in the Command & Control Stage

The LMKC also covers a "non-standard" attack - what this thesis will call, a quasi-insider attack. Attackers were able resolve the pseudorandom number generator of slot machines to predict outcomes and bet accordingly (NPR, 2019a). It seems likely that the attackers were able to acquire a least one type of slot machine, reverse engineer it and build C&C to support an attack to make money by adjusting betting based on the slot machine outcomes by sending electronic messages to the players. The LMKC is flexible enough to support description of this attack:

- Reconnaissance – Obtain and reverse engineer a slot machine. Search for casinos which use the slot machine;
- Weaponization – reverse engineer a slot machine, build an app to aid exploitation;
- Delivery – send people to casinos with the slot machines;
- Exploitation – play the slot machines;
- Installation – continue to play the slot machines;
- Command & Control – record slot machine outcomes, deliver outcomes results in real-time, send betting directions back to player;
- Actions on Objectives – player bets according to directions received. Cash out as appropriate.

The weakest part of this description is "Installation" as Lockheed Martin assume persistence after installation. However, "Installation" is flexible enough to assume direct access of the APT team to, and not software on, the machine. The LMKC (or mechanism of action) for this attack was broken when members of the team were arrested in the US.

Lockheed Martin’s seventh and final step of Actions on Objectives is a sufficient “catch-all” but a strong piece of support is Lockheed Martin’s Table 1: the defender’s “Courses of Action Matrix” to defend against stages of the Kill Chain. This “Courses of Action Matrix”, provides mitigations for each of the Kill Chain stages. The “Destroy” column is completely empty and this is where part of this thesis fits. This thesis suggests that the course of action table could be improved by the addition of “Deter” as discussed by Smith and Rothwell (Smith and Rothwell, 2015, p. 4) bringing the full table to Deter, Detect, Deny, Disrupt, Degrade, Deceive, Destroy. This thesis combines all such elements.

It is this view that leads this thesis to discuss APT attacks through the vehicle of the Lockheed Martin Intrusion Kill Chain.

4.9 Conclusion

This chapter has provided the wider business context of APTs and malware. It supports the first Aim and Objective and has:

- discussed the reasons why IT Security is important;
- given a history of malware and some of their parallels in the non-IT, physical world;
- addressed the costs of malware infection to the defender and the business positions for both attackers and defenders;
- reviewed and compared various Kill Chains and selected the Lockheed Martin Intrusion Kill Chain for the thesis.

As stated in the Financial Times (Financial-Times, 2017a) that if one is doing something new one should have a good explanation presented in an effective way. The subject said that she was going to solve the biggest problem for people who ride bikes. It is the biggest problem for a big bunch of people. This gets attention and people ask:

“What is the biggest problem?”

This thesis will address the biggest problem in IT Security: APTs - and continue with evidence from the real world.

5 LITERATURE PRESENTATION

5.1 Chapter Overview

This chapter introduces open source white papers on APTs and malware and supports the first and second Aims and Objectives. It explains aspects of the HDD and Windows operating systems where APTs may lie. It links to the first two points of the Aims and Objectives:

- discover where APTs lie on windows-based systems;
- create/build generic views of where APTs lie on Windows based systems using an appropriate Cyber Kill Chain (CKC);

In order to understand where malware lies and what it looks like the chapter start with a review of a pristine HDD and Windows OS.

Each sub-section starts, where appropriate, with a definition and/or description of the subject under review, taken from Microsoft webpages. This is to allow the reader to gain an understanding of the subject and how the mitigations against APTs fit in.

The malware deployed by an APT may hide in plain sight (i.e. the files may be known to the operating system or may be largely invisible to the operation intercepting communications between application and the operating system) or not. APTs use a variety of techniques to make a defender's job harder. It is the intent of this chapter, building on the previous one, to highlight commonalities across APTs and malware to produce a set of observations that can be used as a basis for anti-malware computer programs and associated software.

5.2 HDD and Windows OS Analysis

5.2.1 What does Pristine and Re-formatted HDD Look Like?

Having discussed how, philosophically, malware may be analysed this thesis asserts that first one should know what a pristine system looks like. Without knowing what is legitimate and, therefore, expected, one cannot identify the unexpected or abnormal i.e. Categorisation and analysis should not be done on a HDD without first having a baseline of what a pristine HDD should look like.

The author bought a new SP USB 3.0 1TB portable HHD and analysed random sectors of the HDD by writing and using the program *sec_check*. No information on what a pristine HDD should look like could be found and this author speculated that it could be all zeros, repeating 0s and 1s or a check pattern. Given that it was 1TB it was hypothesised that not all of the HDD would be written to (such a business proposition for HDD manufacturers would just take too long for all of the product line) and that it would be all 0s. This indeed was the case as far as could be ascertained. A number of consecutive 5000 sector areas of storage were viewed. Apart from the start which contained an MFT (there were indications that it was a FATS HDD and some documentation), the observed sectors were all zeros.

Elsewhere, on one of this author's personal machines, there were many sectors with all zeros but some had repeating binary pattern) The pristine HDD both confirmed (sectors all zeros) contradicted (no repeating binary patterns) what had been seen on the author’s personal machine.

A previously used Windows HDD was analysed before and after reformatting. The program *filelist* was used to output the MBR and the full list of files. The program sec_check output five different random sectors before and after reformatting. These sectors were equal across reformatting. After reformatting all files on the disc could not be accessed by Windows 10 but they could be read by software developed for this thesis. Simply using Windows OS software to reformat the disc does not delete files or cause them to be unreadable by all software.

The before and after reformatting MBRs were different with different MFT Record Size (1024 viz a negative, presumably unallocated number), Sectors Per Cluster (8 viz 1), Clusters Per File Record Segment (246 viz 2) MFT logical cluster number start (786432 viz 6291456) and two different VSNs. Additionally, two separate files for each format containing random sectors were analysed using the MS command “comp” and found to be identical. i.e. the reformatting only affects the MBRs. It can, therefore, be stated that although the original files on the reformatted

HDD were not visible to the operating system they would be visible to forensics' software. The MBR difference table may be viewed in Appendix C.

This leads to an observation on one Windows tool: the GUI for reformatting with Disk Management in the Control Panel is not strictly true which. It states that on reformatting "All data on the partition will be lost." A simple internet search reveals tools that can recover data. One demonstration of a product has the GUI stating "Deleting this volume will erase all data on it" (Hetman-Recovery, 2019). Loss of data after reformatting is not true.

Given the amount of time it would take to overwrite the HDD one assumes that Microsoft decided to make it appear to the operating system that the files had been deleted. Similarly, this thesis infers that, a pristine HDD may not contain all zeros but any position that has not been magnetised will return zero by the HDD microcode.

Analysis of the pre and post reformatting of the HDD highlights a contradiction in the VSN fields structure description. The VSN is a function of the date and time HDD creation or copying as described by Lunt in Section 28.2 (Lunt, 2004) and is eight hex digits (The conversion is not a 1-1 mapping. Using the example given in the reference of 1995 and 21:55, the same result (VSN part), 0x1D02, can be obtained with 1996 as the year and 21:54 as the hour and minutes). Elsewhere the VSN is claimed to be 16 hex digits (Wilkinson, 2017), (LSoft-Technologies-Inc., 2020a) yet the Microsoft command "vol" only returns eight hex digits – the first four bytes of the 16 hex digits read as little Endian. The leaves four bytes, eight hex digits the middle two overlap with the first two of the VSN.

5.2.2 What Does a Windows Operating System Look Like?

Having discussed what a pristine HDD looks like attention is now turned to the OS of choice for this thesis. For a commercially available desktop or laptop a Windows OS permits 16 physical drives (0-15) and 26 logical drives (A-Z). Logical drives may reside on more than one physical disk. The program *read_mbr*, written for this thesis, lists all physical and logical drives permitting the analyst the opportunity to check the drive structure on the computer. It also list the attributes of the drives e.g. drive size, sector size etc from the MBR. At the high-level Windows is laid out one a disc thus (LSoft-Technologies-Inc., 2020b):

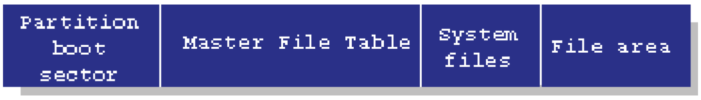


Figure 5-1: Windows Schematic on a HDD

5.2.3 Overview of Windows Components

The major components of the Windows operating system are well known (Microsoft, 2017y).

Kernel Mode vs. User Mode:

“NOTE The architecture of the x86 and x64 processor defines four privilege levels, or rings, to protect system code and data from being overwritten either inadvertently or maliciously by code of lesser privilege. Windows uses privilege level 0 (or ring 0) for kernel mode and privilege level 3 (or ring 3) for user mode. ... two levels is that some hardware architectures that were supported in the past (such as Compaq Alpha and Silicon Graphics MIPS) implemented only two privilege levels.”

(Russinovich and Solomon, 2009, p. 16)

This is graphically illustrated by Atwood (Atwood, 2008) with the x86 CPU hardware protection rings:

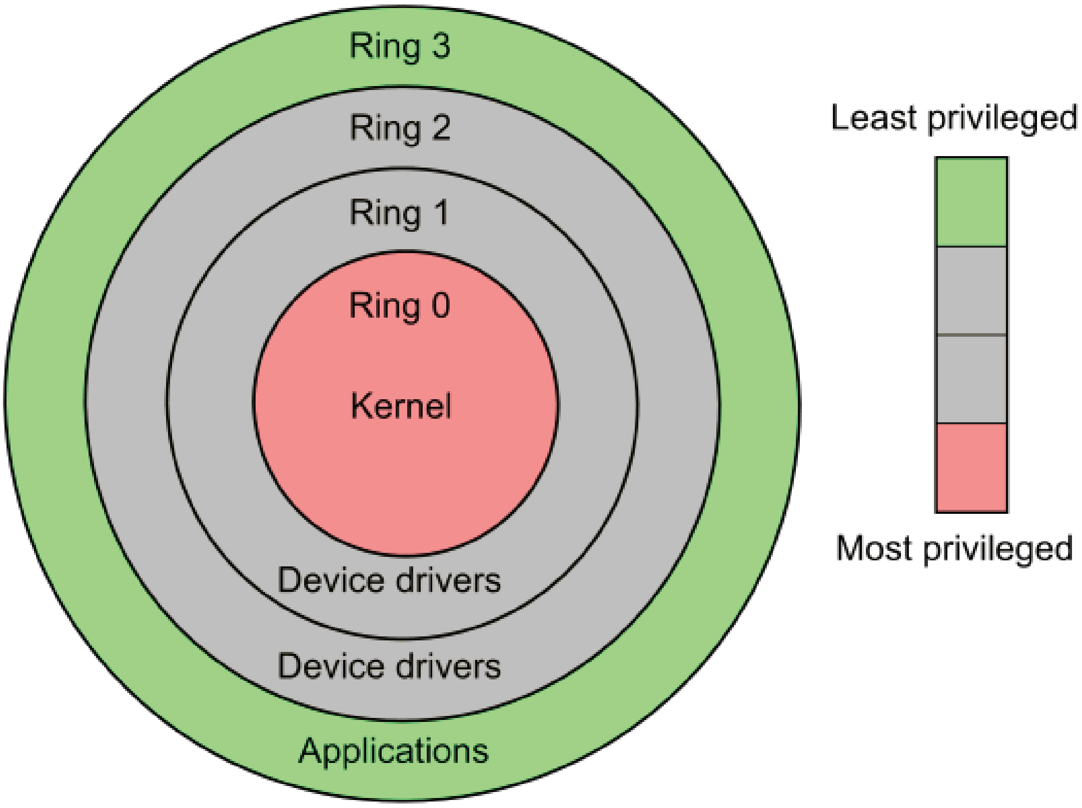


Figure 5-2: x86 CPU hardware protection rings

Ring 0, or Kernel mode, makes use of kernel specific routines which begin Zw. Outside of the kernel, machine information may be acquired by routines such as DeviceIOControl and programs may be suspended by the use of the sleep routine. It has been demonstrated earlier in this thesis that APTs build malware out of such routines and this observation will be built on.

5.2.4 What happens at Startup?

It has previously been shown that the Registry RunOnce and Run keys at HKLM and HKCU run at start up and both are separated by a call to the start-up folder. This is discussed by Skoudis and Zeltser (Skoudis and Zeltser, 2004, pp. 191-201) who quite rightly suggest that their list of registry keys that start programs on login or reboot is not exhaustive. It is noted that the registry key name has a limit of 255 characters while the value name has a limit of 16,383 characters (Microsoft, 2018u).

Many APTs (Symantec, 2016a, pp. 13-14) use what is called fileless malware to store and run their malware. Fileless is when the malware is stored not as a file listed in the MFT but as a file in a key in the Registry. File based protection (stored in, and from, the registry) is not a new concept (Symantec-Security-Response, 2012c, p. 6) and the software developed for this thesis could be modified to work in this way. However, as will be discussed the file related date information is not reliable. Software developed for this thesis addresses this form of attack.

5.3 Executables and DLLs

The Microsoft Portable Executable (PE) format is well documented (Pietrek, 1994), (Pietrek, 2002), but is provided here pictorially for convenience:

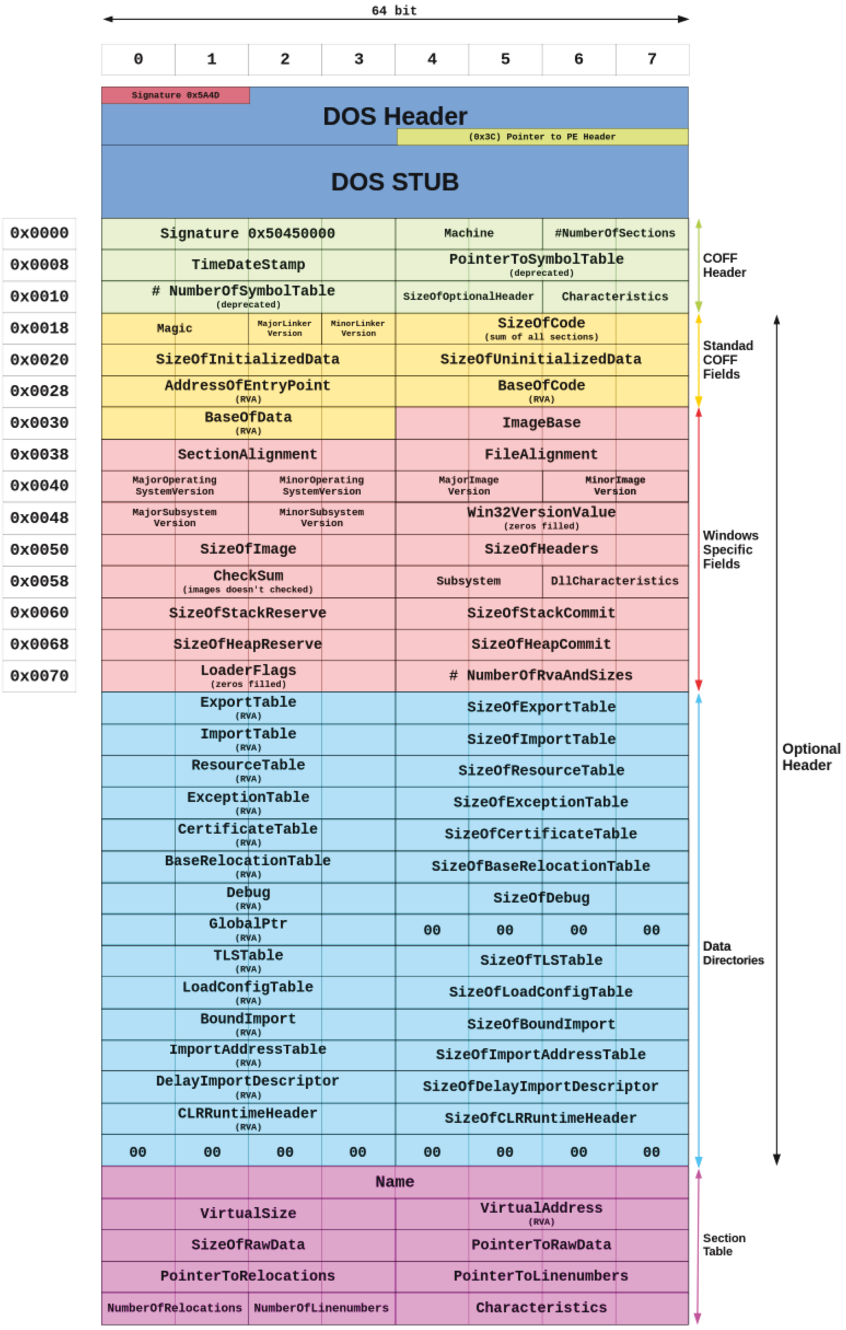


Figure 5-3: Structure of a 64-bit Portable Executable (Patel, 2016)

An executable may be considered a standalone program while a DLL is a Dynamic-linked library which may be considered subroutine. The difference between an executable and .dll is a one-bit flag (Microsoft, 2018r).

It is generally necessary to lay down malware on the victim's machine in order to commence the attack. This malware may be in the form of a set of commands, e.g. .bat files, or executables. The malware may be enciphered, or not, and may contain hardcoded information, for example URLs to connect (Yaneza, 2015a, p. 8). Executables may be identified by the filename extension of .exe and the use of a specific format, including the MZ two-byte header (Pietrek, 1994). However, one APT replaces the MZ bytes "0x4D 0x5A" with "0x9B 0x8A", possibly to confuse AV software (Fidelis-Cybersecurity, 2015a, p. 5). These bytes are later changed back to MZ when the file is needed.

Malware has also been seen as a Visual Basic Script, .VBS (O'Brien, 2016, p. 20), (Symantec-Security-Response, 2012b, p. 3), Assembly, Basic, C, Delphi and Visual Basic (Suenaga, 2012, pp. 2-3)

Symantec have observed one APT using fileless persistence in the registry. The meaning of fileless in this context is that the file, malware, does not have an entry in the OS file system. As will be later discussed in this thesis, malware creates a Run subkey entry which points to a malicious executable. This is then executed at logon. In some cases, the APT uses rundll32.exe, a legitimate Windows executable used to load DLLs, and passes several parameters. These parameters include JavaScript results in malware being loaded into memory and executed." (O'Murchu and Gutierrez, 2015, p. 18).

Symantec (Neville and Gibb, 2013, p. 26) have observed a piece of malware which places other malware onto the victim's machine (a dropper) flipping the DLL flag in the PE header to change itself from an executable to a DLL.

While Sophos have seen files dropped into folders and registered in Registry keys (Szappanos, 2014a, pp. 3, 6, etc.).

It has been noted that zero length files may be malware (Bontchev, 2006, p. 9) although no explanation of the mechanism of action is given. However, Hamre (Hamre, 2005) highlights the use of alternate data streams (ADS) to “hide” data. The use of ADS is not restricted to zero-length files; one or more ADS may be used with any length file. The use of ADS is noted elsewhere (O’Murchu and Gutierrez, 2015, p. 6) and will be addressed later in this thesis.

F-Secure have seen malware constructing concatenating names from a common list of two to four long strings .e.g. the list contains the strings “url”, “lsa”, “ras” and “dns” so, for example, the malware may be named urllsa.exe or rasdns.exe (F-secure-Labs, 2014b, p. 7).

It has earlier been referenced that AV tools easily identify the malware as the malware entry point lies outside the legitimate code section (Symantec, 2005). Such an action would be easily identifiable but will not be considered in this thesis as it has already been done. Elsewhere executable path extensions (.COM; .EXE; .BAT; .CMD; .VBS; .VBE; .JS; .JSE; .WSF; .WSH; .MSC) may be displayed by the command “echo %PATHEXT%” in a command window and found at:

Computer\HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Session Manager\Environment

and

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment

Services may be found at:

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services

and

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Services

The Services automatic update program is wuauclt.exe and Process Hollowing (removing legitimate code and inserting malware) has been observed (Malwarebytes, 2018a, p. 7).

Installed Software maybe be found in the registry keys:

https://stackoverflow.com/questions/25131413/wmi-get-the-list-of-installed-softwares

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall

HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\

5.4 Filename Extensions including Compressed Files

As previously discussed in this thesis (Villeneuve and Bennett, 2012, p. 5), a common method of entry onto a victim's system is the use of spear phishing with .doc files containing malware (Alintanahin, 2015, pp. 2-3), (O'Brien, 2016, p. 18), infected Microsoft™ Excel® (Pernet and Lu, 2015, p. 2) and .exe files being made to appear as "ordinary" documents (e.g. .doc) (Microsoft, 2017q). It is also possible to make an executable's icon look like a folder (Villeneuve, 2011, p. 10), (Suenaga, 2012, pp. 41-43) (with the latter hiding - "If the worm finds a folder, it hides the original folder, copies itself using the folder name, and gives it the file extension of ".exe".) or folders by using the associated icon and using right-to-left override (RTLO) in UNICODE (TrendLabs, 2014, p. 10), (Fireeye, 2015b, p. 19), (Kaspersky, 2015a, p. 12) or other software (Nart Villeneuve *et al.*, 2014, p. 9), (F-secure-Labs, 2014b, p. 4). One APT (Pernet and Lu, 2015, p. 8) used free online cloud storage and the malicious file had a Microsoft Power Point (.ppt) icon but whose extension was .ppt.exe i.e. an .e.g. disguised as a .ppt file. Re-labelling one type of file as another is not restricted to .exe, .dll, or .pdf etc.: one piece of malware propagates through removable drives (TrendLabs, 2012b, p. 2) or one which uses a font file as a .dll extension name (Dela Paz, 2011) or Encrypted Virtual File System

(EVFS) containers which contain for example ..evt and .imd file extensions (Symantec-Security-Response, 2015c, pp. 11, 14).

Panda Security have seen a self-extracting file disguised with a .pdf icon to extract six files into a folder which it creates (Panda-Security, 2015, p. 4). This attack makes no use of malware, just legitimate tools and scripts (i.e. Living Off the Land). Kaspersky (Kaspersky, 2015a, p. 24) have also seen an APT use attacks which demonstrate that zero-day techniques are not necessarily needed for efficient targeted attacks. Using phishing emails, social engineering and homemade tools and backdoors, the APT infected hundreds of victims' computer systems or mobile devices. A Living Off the Land "Benign Ware (Benware)" was developed in support of this thesis and will be discussed later.

Spear-phishing emails can have various file type attachments. During one example of TrendLabs organisational monitoring (TrendLabs, 2012c, p. 2) TrendLabs found that the most common and shared file types in organisations (e.g., .xls, .pdf, .doc, .docx, and .hwp) accounted for 70% of the of spear-phishing email attachments.

TrendLabs asserted (TrendLabs, 2012c, pp. 2-3) that executable files were not commonly used as spear-phishing email attachments. This is probably because emails with .exe file attachments are detected and blocked by AV software. TrendLabs later assert that this is why .exe files are usually compressed and archived before being sent as .lzh, .rar, and .zip files. This is consistent with observations about .exe files, above.

This is somewhat reflected by one APT attack as documented by Symantec (Symantec-Security-Response, 2013, pp. 15-16). Of 35 attachment files the extensions were .exe (3 times), .xls (8 times), .doc (7 times), .scr (5 times), .pdf (12 times). .scr and .7z have been seen by others (Arborsert, 2014, p. 1), .CSS files to obfuscate .7z, .RAR files as well as using .PST files (Chang *et al.*, 2015, pp. 6, 7)

Fireeye support the TrendLabs view (Fireeye, 2014g, p. 6) stating that there is a move away from .exe as a filename extension with .zip representing the vast majority at 76.91% of extension names, followed by pdf (11.70%), .exe (3.98%), .doc (2.67%) and .pif (1.09%)

Symantec also report (Candid Wüest, 2014, p. 23) their analysis where, of the attachments analysed, 38% were .exe and 12% were .src files. Only 6% used double extensions like .pdf.exe to fool the user. Also 23% were Microsoft Word documents using an exploit to execute custom code on the victim's machine.

One APT uses non-standard filename extensions e.g. .-MP, .wA-, .Bz7 (Katsuki, 2012, p. 5) and another uses .Enc (Nart Villeneuve *et al.*, 2014, p. 8).

At least one APT specialising in an given economic sector has used Control Panel Applet (.cpl) files (Kaspersky-Lab, 2015, p. 3). "Each tool in Control Panel is represented by a .cpl file in the Windows\ System folder. The .cpl files in the Windows\System folder are loaded automatically when you start Control Panel. Note that Control Panel files are sometimes loaded using entries in the [MMCPL] section of the Control.ini file. " (Microsoft, 2017g)

Forcepoint have seen the use of encrypted binary blobs (proprietary device drivers) "… typically packaged into a malicious document via an encrypted binary blob within that document. This binary blob often contains a legitimate decoy document that is shown to the user. … The encoded VBScript uses a file extension which is not associated, by default, as being a VBScript file. The extensions ***.**domx* and ***.**lgx* have been observed. The shellcode is responsible for adding a new file association for the file extension which specifies that they should be interpreted as an encoded VBScript." (Settle, Griffin and Toro, 2016, p. 17). Other filenames include .tmp (Symantec-Security-Response, 2015a, p. 10) and Cabinet Files .cab (Kaspersky, 2014f, pp. 8-11, 13).

"A cabinet is a single file, usually with a .cab extension, that stores compressed files in a file library. The cabinet format is an efficient way to package multiple files because compression is performed across file boundaries, which significantly improves the compression ratio."

(Microsoft, 2018c)

There is no single compression algorithm or methodology used in Information Technology and each has its own extension name. Christiansen provides a review of a variety of compression formats (Christiansen, 2010, p. 25).

5.5 Alternate Data Streams (ADS)

Russinovich (Russinovich, 2016) observes that NTFS allows applications to create alternate data streams. All data is stored, by default in a file's main unnamed data stream, alternates may be written to and read". A Microsoft graphic supports this definition (Microsoft, 2009b). An ADS carrying malware may also be known as "Stream Companion" virus (Symantec, 2007).

The problem with ADS is that Windows and many applications make use of it (Broomfield, 2017). For example the USN change journal uses two alternate data streams (Russon and Fledel, Undated, pp. 60-62). However, ADS cannot be sent over TCP/IP to a remote network. Attempting to do so results only the main stream being sent. ADS cannot be copied, transferred or exist in a non-NTFS environment. Any attempt results in transferring the main stream and not the ADS (Martini, Zaharis and Ilioudis, 2008).

It is claimed (Mahajan, Singh and Miglani, 2014) that ADS can be hidden behind any existing system file and that multiple files can be added behind a single file. ADS are not affected by moving or copying; and that the existing file is unaffected by the ADS but the original reference cannot be found. The same authors also state that file system backup only backup the default streams of the file and therefore backups of ADS cannot be created.

Mahajan et al further claim that Microsoft Word uses ADS attached to the document to store annotations but the reference (Berghel and Brajkovska, 2004) does not mention this. A wider list of ADS features is available (Mahajan, 2016)

5.6 Use of Hidden Files

Microsoft® files may be Hidden or not. The Windows OS Hidden attribute ensures that the files are not viewable by the user with the normal Windows operating system software (Sancho *et al.*, 2012, p. 2). They are viewable, however, using the command line “dir /H” in the cmd.exe window. The use of the Hidden attribute has also been seen in association with setting the file attribute as “System” (Chang *et al.*, 2015, p. 26). While Fireeye (Fireeye, 2014a, p. 37) note the use of a hidden file.

Rootkits can hide processes, registry keys, and other evidence of the existence of malicious software in a computer. (Goncharov, 2012, p. 15). Symantec have noted (Falliere, Murchu and Chien, 2011, p. 24). The file sizes are between 4Kb and 8Mb. They also note the uses of “Copy of Copy of Copy of Copy of Shortcut to .lnk” and “Copy of Copy of Copy of Shortcut to. .lnk” etc.

F-Secure have also seen the use of a LNK file in the start-up folder with the LNK file being a shortcut to the main dll using rundll32.exe (F-Secure-Labs, 2014a, p. 10).

Although not a hidden file in the true sense of the word at least one APT has been using software that is legitimately used to hide contents from those who would pirate those contents (Chen and Li, 2015, pp. 6-8).

Kaspersky have seen malware set its attributes to Hidden and System if they are not already set as such before connecting to the C&C server (Kaspersky, 2013d, p. 29) as well as seeing a LNK file in the Startup folder which is later removed (Kaspersky, 2013d, p. 38).

Kaspersky have seen (Kaspersky, 2013e, p. 16) a driver whose purpose is to hide malware network connections which the driver has established. For example, if the user decides to check a list of connections while the bot (malware) is communicating to its control, the driver will protect and hide the malware connections. Many Windows rootkits use this technique.

F-Secure (F-Secure-Labs, 2015c, p. 3) have seen an APT which requires the victim to click on a zipped file. This is launched and looks for a specific Dynamic Link Library (DLL) file, fbgen.dat, from a defined Dropbox file sharing link. If the file is found and successfully downloaded, the Java executable (JAR) file executes it using “regsvr32.exe”, with “/s” key for silent registration of DLL. This ensures that dialogue boxes are not displayed and so hides execution from the user.

There are few references in AV company literature to the use of Alternate Data Streams (eset, 2017, p. 14), (Wyke, 2012a, p. 17), (Giuliani, 2011, p. 2) which cover just two APTs. Either AV companies are withholding this information as intellectual property, AV companies are not looking for the use of ADS or AV companies have been looking for, and have not found, the use of ADS.

The use of a hidden console window has been seen in support of a keylogger (Kaspersky, 2014c, p. 2).

5.7 Use of the Windows Recycle Bin

Both RSA (RSA, 2014, p. 33), Sophos (Wyke, 2012b, p. 7) and Kaspersky (Kaspersky, 2013e, pp. 31, 47) have seen malware hidden in the Windows recycle bin. While Fidel Cybersecurity have seen use of %APPDATA%, %USERPROFILE%, %PROGRAMDATA%, AND %TEMP% as well as a USB drive $RECYCLE.BIN (Fidel-Cybersecurity, 2016, p. 4). Novetta have seen %TEMP% (Novetta, Unknown, p. 7), %PROGRAMFILES%, %WINDIR% (Novetta, Unknown, p. 10), %SYSDIR% (Novetta, Unknown, p. 17) and Kaspersky %SYSTEM%, %APPDATA% (Kaspersky, 2014f, p. 10)

5.8 Process Camouflage

Process Camouflage is the description given to well-named processes that do not attract attention. They may be slight variations on legitimate operating system process names whose binaries reside in non-standard locations. They may also be variations of common processes (Harbour, 2007). Fireeye (Fireeye and Mandiant, 2014, p. 14) re-enforce this by stating that malware is given a benign looking name. Alternatively the malware may inject itself into a legitimate process such as svchost.exe (Dela Paz, 2012, p. 4), taskmgr.exe (Kharouni, 2015, p. 4) or services.exe (Symantec-Security-Response, 2015c, p. 11). Symantec report that one APT injects into svchost.exe using ZwQueueApcThread (Symantec-Security-Response, 2015b, p. 13). McAfee also report the injection of svchost.exe by an unspecified mechanism with one of two DLLs depending on operating system (Sherstobitoff, Liba and Walter, 2013, p. 14) . Others may simply install itself as a Windows service as described Section 5.14.4.

Mandiant (Mandiant, 2010, p. 8) provides a diagram of "APT: Persistence Backdoors 60% of APT backdoor samples were persistent on the machine". This 60% was made up of Windows Service (76%); HKLM Run Registry Key, other (3%). The diagram is accompanied by a footnote stating "APT: Non-Persistent Backdoors 30% used process injection to avoid detection". The 3% of persistence backdoors presumably includes the Startup folder.

These statistics are somewhat confusing: presumably this means that 40% malware used Non-Persistent Backdoors and of these 30% used process injection to avoid detection. Otherwise there is 10% of malware "missing" (60% + 30%). Such injection may be done by a binder (Pernet, 2016, pp. 22-23), (Cylance, 2016, p. 33) or a joiner (Goncharov, 2012, p. 2) and although not identified as being modified by a binder Katsuki (Katsuki, 2012, pp. 6-7) noted that a legitimate application had an entry point modified to launch appended code. Katsuki goes on to note that a legitimate SSH client had been modified to become an installer.

Injection may also be by reflective injection (Fewer, 2008) which the ability of a program to modify its own code (Malenfant, Jacques and Demers, 1996)

In Section 5.8, Process Camouflage, one malware builder makes the malware self-contained within the dropper document and "generates malicious documents, such as Word and PDF documents, Excel workbooks, CHM-compiled help files, and HTML pages." The process may also set the font code to match the background colour making it look like there is no content (Gábor Szappanos, 2016, pp. 4, 15).

Ször and Ferrie discuss different types of obfuscation (e.g. encryption, oligomorphism, polymorphism and metamorphism) and state that:

"A file can be infected if it smaller than 448 KB, begins with "MZ" (Windows does not support "ZM" format applications), is not infected already (the infection marker is “Z” at offset 0x1C in the MZ header. Windows applications do not generally use this field), and is a Portable Executable file.”

(Ször and Ferrie, 2003, p. 16).

Svchost.exe has been seen in conjunction with DeviceIOControl (Florio and Kasslin, 2009, p. 5) which will be discussed later in this thesis.

5.9 Miscellaneous Windows Functionality

5.9.1 Shadow Copy

“The Volume Shadow Copy Service (VSS) is a set of COM interfaces that implements a framework to allow volume backups to be performed while applications on a system continue to write to the volumes.” (Microsoft, 2018w)

Sophos have seen ransomware deleting shadow storage using vssadmin.exe (Wyke and Ajjan, 2015, pp. 4, 14, 53), as well using other software to perform the same task (Wyke and Ajjan, 2015, p. 51). Panda Security have also seen malware delete shadow copy (Panda-Security, 2017, pp. 2, 5). While in a review of 10 ransomware families Sophos (Loman, 2019, pp. 13, 15, 17, 19, 21, 22, 25, 27-28) have seen ransomware deleting shadow storage in eight of the families.

5.9.2 Use of Directories

Following payload execution the decoy document is decrypted and then saved into the Windows temporary directory (Trend-Micro-Threat-Research-Team, 2012, p. 4), %ALLUSERPROFILE%\Application Data\default (Kruse, Hacquebord and McArdle, 2012, p. 9).

5.9.3 Use of Computer Specific Information

The Volume Serial Number (VSN) may be used in the attack (Kruse, Hacquebord and McArdle, 2012, p. 11) as a bot id. Fireeye (Fireeye, 2015a, p. 16) have seen malware with 45 encoded hard disk serial numbers hardcoded within the controller binary and this will only run if there is a match. Kaspersky have seen (Kaspersky, 2013d, p. 26) the VSN used as a basis for an encryption key: a unique system ID is constructed from a hash of the system drive VSN which comprises two hash algorithms: a custom algorithm and then MD5.

5.9.4 Bypassing Security Controls

Some APTs check the infected machine for AV protections and disable them. One such APT disables the Firewall, default Windows protection and UAC. Using WMI it checks for other security products (Yaneza and Mendoza, 2016, p. 2). Another APT (Alintanahin, 2015, pp. 4-5) performs a similar check while yet another (Falliere, Murchu and Chien, 2011, p. 14) looks for known AV products to inject malware.

Kaspersky (Kaspersky, 2017a, p. 8) have also seen UAC bypassing where a worm connects to a remote machine’s registry and disables Remote UAC changing the relevant registry key.

5.9.5 Residing in Unused HDD Space

An APT may reside (“hide”) in any unused space on the HDD (Symantec-Security-Response, 2012c, p. 3). Palumbo has seen one APT overwriting its stage #1 component on the HDD before freeing the memory space (Palumbo, 2014b, pp. 6-7). An attack which also sees the use of an APT using a enciphered VM before payload deployment of a PE32+.

5.10 Windows Management Instrumentation (WMI)

“Windows Management Instrumentation (WMI) is the infrastructure for management data and operations on Windows-based operating systems. You can write WMI scripts or applications to automate administrative tasks on remote computers but WMI also supplies management data to other parts of the operating system and products,” (Microsoft, 2017al).

WMI runs as a service and starts automatically at system start-up under the LocalHost account. If it is not running it will start automatically when the first script or management application request connection to a WMI namespace. Stopping WMI is not an option as other services including Windows firewall will also stop (Microsoft, 2018v).

Microsoft (Microsoft, 2009c) present WMIC which is a way of taking command line control of WMI, something observed by Mandiant (Mandiant, 2017, p. 14) for persistence and Mandiant express the opinion that PowerShell is the most powerful way to interact with WMI (Mandiant, 2015b)

Persistence may be established by using WMI (Trend-Micro.Forward-Looking-Threat-Research-Team, 2012, p. 3) and also being a repository from which attacks may be run (Dizon, Galang and Cruz, 2010). In addition WMI may be used to check the files opened by victims and APTs can easily determine and collate these files for exfiltration (Trend-Micro, 2013c, p. 4).

Trend Micro (TrendMicro, 2013, p. 5) observe that remote control tools allow APTs to access other desktops in the network and execute programs, schedule tasks, and manage data collections on other systems. Cylance note (Cylance, 2016, p. 39) that PsExec can be used to run commands on any other machine which accepts those domain credentials and that if this is combined with cached credential dumping, it can be used on a compromised network to jump from machine to machine.

FireEye have produced a discussion of WMI attacks and defence (Ballenthin, Graeber and Teodorescu, 2015).

Kaspersky have seen WMI used to collect information from profile directory lnk files and user information that available from remote registries (Kaspersky, 2015b, p. 20) as have Mandiant to collect run data (Mandiant, 2015a, p. 16),

5.11 Multiple APTs on One System

Kaspersky have seen an APT which stored malware at the end of the last partition on disk (Kaspersky, 2014e, p. 6), while Palumbo has seen an APT store the payload in the gap between the past partition on the disc and the end of the disk (Palumbo, 2014b). This has the potential for APT collisions on a system which may cause problems for one, both or all APTs on the system and may look like good defence by the victim.

Although having two different APTs on the same machine sounds unbelievable, Kaspersky have identified one computer which is infected by at least six different, probably unrelated, APTs (Kaspersky, 2015c, p. 31), (Kaspersky, 2014e, p. 3). As an aside this is interesting for at least two reasons: either that some of the APTs know that others are on the machine and seem to have an unspoken tacit agreement to leave each other alone; or they do not check for other APTs on the system and therefore may not get all the data they require as another APT may be restricting access to files.

However, aggressive responses are unusual but not unknown: one APT found itself the victim of a spear phishing attack and responded with malware (Emm *et al.*, 2015, p. 6). Elsewhere Kaspersky have seen false flag operations where malware used by one APT was deployed by another (Kaspersky, 2014d, pp. 71-72).

5.12 Windows Kernel, ZwXxx and NtXxx Routines and Other Interventions

"The **Zw*Xxx*** routines provide a set of system entry points that parallel some of the executive's system services. Calling a **Zw*Xxx*** routine from kernel-mode code results in a call to the corresponding system service. Calling a **Zw*Xxx*** routine from user mode is not supported; instead, native applications (applications that bypass the Microsoft Win32 subsystem) should call the **Nt*Xxx*** equivalent of the **Zw*Xxx*** routine." (Microsoft, 2018x).

Zw routines are called from kernel mode and use of these routines, while not evidence of malware, may be an Indicator of Compromise (IOC). For example, one piece of malware uses a .dll which uses at least seven Zw routines (Falliere, Murchu and Chien, 2011, pp. 13, 30) and another uses at least two (Neville and Gibb, 2013, p. 19).

Symantec has seen the use of at least eight Zw routines (Symantec-Security-Response, 2015d, pp. 12, 14) while McAfee highlight the use of ZwAllocateVirtualMemory to allocate user space from the kernel (McAfee-Labs-Threat-Advisory, 2014). F-Secure have also seen the use of Zw routines where the bootkit created a system thread via PsCreateSystemThread which loads the kernel payload by reading it from the raw disk's sector via ZwOpenFile and ZwReadFile" (F-Secure-Labs, 2014d, p. 9).

5.13 Hooks

"A hook is a point in the system message-handling mechanism where an application can install a subroutine to monitor the message traffic in the system and process certain types of messages before they reach the target window procedure."

(Microsoft, 2018m)

An APT Hooks onto the kernel filesystem device stack and obfuscates the IRP file buffer for protected files (Bingham, 2012, p. 4). File buffering allows drivers which service slow devices (i.e. those which transfer small amounts of data) to improve memory usage. It is generally used by such as mouse, keyboard and video (Microsoft, 2017ah).

A hook may be used to evade file-based Sandboxes. This is done by using the SetWindowsHookEx function (Microsoft, 2017ae) (Singh and Bu, 2014, pp. 4-5). The function monitors the system for a specific type of event e.g. a mouse click or other human actioned events. When this occurs, malware is called. Similarly a keyboard logger has been seen using GetAsyncKeyState (Fireeye, 2015b, pp. 21-22). This code may have been based on a publicly available keylogger (Fireeye, 2015b, p. 22)

File-based sandboxes can identify malware by monitoring OS processes. This is done by PsSetCreateProcessNotifyRoutine. One piece of malware uses PsSetCreateProcessNotifyRoutine to remove all registered call backs which would include those from security software (Singh and Bu, 2014, pp. 14-15).

Return-oriented programming (ROP) is a technique where an APT can bring about arbitrary behaviour in a program from control flow diverted by the APT. This is done without code injection. ROP chains are short instruction sequences in the program's address space, each of which ends in a return instruction. A ROP Chain attack has been noted by Sophos (Gabor Szappanos, 2016), (Szappanos, 2015a).

Kaspersky (Kaspersky, 2015b, p. 4) have seen one APT using a Word Document which contained an exploit for a zero-day vulnerability. This malware relied on a malicious embedded True Type Font File (TTF) which allowed the APT to jump directly into Kernel mode from the Word Document, which Kaspersky assert, is “a very powerful, extremely rare, technique.” Other computers in the domain were infected by a few different strategies and in most of the attacks, the APT prepared Microsoft Windows Installer Packages (MSI), which were then deployed remotely to other machines. These were then launched as a service using msiexec. The Task Scheduler was also used to start msiexec remotely.

5.14 Persistence through Startup Mechanisms

5.14.1 General

Villeneuve observes that:

“In many cases, the persistence mechanism will consist of simple methods such as adding the malware executable to the windows Startup folder, modifying the Run keys in the Windows Registry or installing an application as a Windows Service. The security form (sic) Mandiant found that 97% of the targeted malware they analyzed used these simple mechanisms.”

(Villeneuve, 2011, p. 13).

Mandiant list (Madiant-Consulting, 2016, pp. 28-35) a number of persistence mechanisms: Windows Services; Windows Registry; DLL search-order hijacking; Modification of Group Policy Objects (GPO); Use of Common Object Model (COM) objects; Modification of existing binaries; Windows scheduled tasks; Windows Management Instrumentation (WMI); Malicious Windows Security Packages; and MBR and VBR bootkits. Glyer (Glyer, 2010, p. 6) provides a variety of examples as well as Mandiant (Mandiant, 2017, p. 14)

Chien states (Chien, 2005, pp. 13-14) that Windows has load points at different times during start-up e.g. when Windows starts, user log in, when the shell (Explorer) starts, and when applications start and that almost all malware uses at least one load point to ensure persistence across reboots. Having more than one mechanism of action, load point or persistence mode is not unusual: one APT has the ability to gain persistence through six different persistence modes: ShellAutorun; HiddenTaskAutorun; ScreenSaverAutorun; StartupAutorun; Task SchedulerAutorun; LinkAutorun (eset, 2017, pp. 9-11).

One APT has been seen changing the path of the startup folder (eset, 2013, p. 12).

5.14.2 Registry Keys

"The *registry* is a system-defined database in which applications and system components store and retrieve configuration data. The data stored in the registry varies according to the version of Microsoft® Windows. Applications use the registry API to retrieve, modify, or delete registry data."

(Microsoft, 2017aa)

Malware persistence may be achieved at reboot by installing Registry entries (Microsoft, 2017ab) which facilitate automatic start-up of software, (Microsoft, 2017ag) as directed by userint (Microsoft, 2010). Persistence can also be achieved by adding entries to one or more of the following registry keys and the following list of malware using these keys are documented:

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run (Symantec-Security-Response, 2013, p. 14), (Symantec-Security-Response, 2014, p. 11), (Virus-Bulletin, 2012, p. 10), (F-Secure-Labs, 2015a, p. 4), (Chang *et al.*, 2015, pp. 24-25), (Anthe *et al.*, 2015, p. 11);

- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run (Trend-Micro-Cyber-Safety-Solutions-Team, 2016, p. 2), (Trend-Micro-Threat-Research-Team, 2012, p. 3), (Kruse, Hacquebord and McArdle, 2012, p. 7), (Pernet and Sela, 2015, p. 24), (Trend-Micro-Incorporated, 2015, p. 3), (Symantec-Security-Response, 2013, p. 14), (Katsuki, 2012, p. 6), (Symantec-Security-Response, 2014, p. 11), (O'Murchu and Gutierrez, 2015,

p. 8), (Fireeye, 2015b, p. 27), (Szappanos, 2014b, p. 4), (Rivera and Inocencio, 2016, pp. 73, 77-78 etc.), (Kaspersky-Lab, 2015, p. 13), (Kaspersky, 2013d, p. 27), (Kaspersky, 2013e, p. 38), (Settle, Griffin and Toro, 2016, p. 22), (Damballa, 2014b, p. 3), (Manos Antonakakis *et al.*, 2012, p. 5), (F-Secure-Labs, 2015a, p. 4), (F-Secure-Labs, 2015c, p. 3), (Gross and Cylance-Spear-Team, 2016, p. 13), (Langill, 2014, pp. 13-15), (Fidelis-Cybersecurity-Solutions, 2015, p. 12), (Fidelis-Cybersecurity, 2015a, p. 11), (eset, 2016b, p. 14), (eset, 2015, p. 17), (Check-Point, 2015, p. 33), (Arborsert, 2014, p. 3), (Arborsert, 2015, p. 6), (Guarnieri and Anderson, 2016, p. 48), (CrowdStrike, 2014b, p. 35), (Clearsky, 2015, p. 16), (Haq *et al.*, 2014, p. 10), (Panda, 2017, pp. 14, 33);

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce (Kharouni, 2015, p. 5), (Settle, Griffin and Toro, 2016, pp. 19, 42), (Check-Point, 2015, p. 33), (Fireeye, 2015a, p. 47), (Panda, 2017, p. 33);
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce (Chang *et al.*, 2015, p. 36).

There is an order for actioning registry keys and running programs in the startup folder (Microsoft, 2018p).

Modification of other registry keys to facilitate attacks is known to occur (Kruse, Hacquebord and McArdle, 2012, p. 9) (Villeneuve and Sancho, 2011, p. 7),

Fireeye have observed a common piece of malware being custom load and only extracted into memory to bypass AV controls. The custom malware uses HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run for persistence (Fireeye, 2015b, p. 25).

Kaspersky have seen (Kaspersky, 2015c, p. 10) persistence through shellcode from the registry: The malware starts Windows and launches a multi-stage (four to five stages) mechanism of decryption before execution within the Windows. The malware then runs the modules that are stored inside the Windows registry. Each stage decodes and executes the next, and the entire suite will start after successful execution of all levels. Should an error occur the entire malware suite self-destructs. The implementation may have been designed this way to make it invisible to AV products. When used with the bootkit, all the modules as well as the stolen data are stored encrypted in the registry and dynamically decrypted and executed. No executable malware modules are on the infected filesystem (Kaspersky, 2015c, p. 12).

5.14.3 Startup Folder

Another way to ensure persistence is to create a shortcut in the Startup folder (Villeneuve, 2011, p. 13) or to change the common Startup folder to a specially created one. The existence of the new folder is constantly checked and redone if it is changed back to normal (Villeneuve and Sancho, 2011, p. 7).

5.14.4 Windows Service. Kernel Service, Raised Privileges

“Microsoft® Windows services, formerly known as NT services, enable you to create long-running executable applications that run in their own Windows sessions. These services can be automatically started when the computer boots, can be paused and restarted, and do not show any user interface. These features make services ideal for use on a server or whenever you need long-running functionality that does not interfere with other users who are working on the same computer.” (Microsoft, 2017s). A database of installed services is maintained by SCM in the registry at HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services (Microsoft, 2017f).

Malware may be installed as a Windows service, (Symantec-Security-Response, 2015d, p. 5) or may be installed by looking for inactive services and, on success, killing one at random and then starting up malware with that service’s name (Trend-Micro-Threat-Research-Team, 2012, p. 5). MacAfee have also seen (McAfee®-Foundstone®-Professional-Services, 2017, pp. 11-12) persistence being achieved through the use of Windows Service or Kernel Service. Damballa have seen an APT installing a kernel driver (Damballa, 2014b, p. 6).

Fireeye have seen at least one APT use schtasks, the Microsoft tasks schedular, to a create a service (Carr, 2017). F-Secure have seen (F-secure-Labs, 2014b, p. 7) use of a Windows Service to open a handle to explorer.exe process and duplicate its process token, It then reads the path of the malware binary from a registry and starts the malware using the duplicated process token.

Mandiant have seen use of phishing for OAuth tokens (Mandiant, 2017, p. 16). OAuth is an open standard to share information without a password. With these tokens an attacker can bypass multi-factor authentication to access a victim's cloud data.

Kaspersky have also seen (Kaspersky, 2013c, pp. 26-27) the use of system services - a Win32 PE executable file compiled in Microsoft Visual C++ 6.0. It drops a .dll file on the victim's machine and registers it as a system service. The malware takes a suitable service name from one of the values in the registry.

5.15 Intra and inter-process Communication

5.15.1 Threads

Many APTs are automated and intra and inter-process communication allows APTs to check to see the malware is running on the target machine.

"A *thread* is the entity within a process that can be scheduled for execution. All threads of a process share its virtual address space and system resources. In addition, each thread maintains exception handlers, a scheduling priority, thread local storage, a unique thread identifier, and a set of structures the system will use to save the thread context until it is scheduled."

(Microsoft, 2018b)

Kaspersky have seen the use of threads for intra-process C&C (Kaspersky, 2013d, p. 29).

5.15.2 Export Tables

"A DLL file has a layout very similar to an .exe file, with one important difference — a DLL file contains an exports table. The exports table contains the name of every function that the DLL exports to other executables. These functions are the entry points into the DLL; only the functions in the exports table can be accessed by other executables."

(Microsoft, 2017m)

"The simplest way to export functions from your DLL is to export them by name. This is what happens when you use **__declspec(dllexport)**, for example. But you can instead export functions by ordinal. With this technique, you must use a .def file instead of **__declspec(dllexport)**. To specify a function's ordinal value, append its ordinal to the function name in the .def file."

(Microsoft, 2017n)

As APTs run, the malware may pass execution to other routines. For example, Kaspersky have seen (Kaspersky, 2015b, p. 10) one piece of malware pass the execution to the second exported function in DLL's export table. This ignores the export name and relies on the order of functions in the table of PE export ordinals. At time export function is called a next stage structure pointer is passed to it so that it can use some of the values set on the upper layer.

Export by Ordinal has been seen by Palumbo (Palumbo, 2014b, pp. 6, 14)

5.15.3 Server Message Block (SMB)

"The Server Message Block (SMB) Protocol is a network file sharing protocol, and as implemented in Microsoft Windows is known as Microsoft SMB Protocol. The set of message packets that defines a particular version of the protocol is called a dialect. The Common Internet File System (CIFS) Protocol is a dialect of SMB. Both SMB and CIFS are also available on VMS, several versions of Unix, and other operating systems."

(Microsoft, 2017w).

Cyrus (Cyrus, 2016, p. 4) gives examples of APTs using SMB. At least one of the references (Nart Villeneuve *et al.*, 2014) does not explicitly mention SMB but elsewhere (Cylance, 2016, pp. 10, 28, 39) it does.

5.15.4 Mutexes and Pipes

A mutex is:

"A synchronization primitive that can also be used for interprocess synchronization." (Microsoft, 2017x). APTs may use more than one mutex in their malware (Kharouni, 2015, pp. 2,4-5,8-9).

One malware tool offers creation of random mutex names (Villeneuve and Bennett, 2014).

"A *pipe* is a section of shared memory that processes use for communication. The process that creates a pipe is the *pipe server*. A process that connects to a pipe is a *pipe client*. One process writes information to the pipe, then the other process reads the information from the pipe."

(Microsoft, 2018s)

Malware has been seen communicating through named pipes (Symantec-Security-Response, 2015b, p. 14), (Symantec-Security-Response, 2015d, p. 15):

A list of mutexes and pipes recovered from the white paper reading is retained separately to this thesis.

5.15.5 Mailslot

"A mailslot is a pseudofile that resides in memory, and you use standard file functions to access it. The data in a mailslot message can be in any form, but cannot be larger than 424 bytes when sent between computers. Unlike disk files, mailslots are temporary. When all handles to a mailslot are closed, the mailslot and all the data it contains are deleted."

(Microsoft, 2018a)

As part of a C&C mechanism Kaspersky have seen (Kaspersky, 2015b, pp. 33-34) the use network pipes and mailslots as well as raw filtering of network traffic and masking C&C traffic inside image files. Kaspersky also mention that the use of mailslots has been inspired by another APT they have seen (Kaspersky, 2014e, p. 14). Interestingly, Symantec note the same use of mailslots by the other APT (Symantec-Security-Response, 2015c, p. 12). F-Secure, who have seen the same APT do not mention the use of mailslots in either of their reports 0n the other APT (Palumbo, 2014a) (Palumbo, 2014b).

5.16 Anti-Anti-Malware Techniques

In order to evade detection, some APTs use anti-anti-malware techniques which Symantec summarise (Symantec-Security-Response, 2015b, pp. 12, 20-21) as: Anti-debug; Anti-emulation; Anti-VM; Packing/compression; Obfuscation; Host-based encryption; Network-based encryption; Server-side tricks; Anti-AV Company. as well as "… password encrypting malware in a zip archive then compressing the file again hoping to bypass email gateway filters (Candid Wüest, 2014b, p. 15).

Fireeye discuss a number of ways to evade sandboxes (Singh and Bu, 2014). For example, "At least one of the … samples contained a simple anti-virtual machine heuristic. Specifically, the GetTickCount function is called and a loop is executed 999,999,990 times that simply increments a variable. After this loop completes, GetTickCount is called again and the values are compared. If they are the same, the process terminates." (Nart Villeneuve *et al.*, 2014, p. 7). FireEye also discuss the use of anti-virus and sand box detection, one of example of the latter being with WMI and PowerShell (Ballenthin, Graeber and Teodorescu, 2015, pp. 21-24)

SecureWorks reports five ways that advanced malware avoids sandboxes:

- Stalling - Malware performs useless CPU cycles disguised to look like non-malicious activity;
- Interaction - Malware determines whether it is on a real-live PC by lying dormant until predetermined human interaction is initiated;

- Environment Check - Malware checks the environment for a virtual machine or well-known registry keys/files that would signify a sandbox;
- Fingerprint - Malware computes a unique host fingerprint upon arrival in environment. When malware starts execution, a new host fingerprint is computed and compared against original to determine if in a different environment;
- Sleep - Using sleep calls, malware refrains from suspicious behavior during monitoring.

(Dell-SecureWorks, 2015)

Fireeye also highlights one APT attempting to disable an AV security product driver to prevent it scanning any process called “service.exe” (McAfee-Labs-Threat-Advisory, 2014, p. 3) while Forcepoint see “Anti Killers” against a range of AV products as well as disabling features (Settle, Griffin and Toro, 2016, pp. 39-40) for example disabling the registry (Microsoft, 2017i)

Although not-strictly an AV mechanism, McAfee have seen a piece of malware check for a debug file, exiting if found. It is believed that this is included to stop the malware infecting the authors’ machine (McAfee-Labs, 2016, p. 29).

Another IOC may be the use of DeviceIoControl function. Also the use of IRP_MJ_READ request which can be used when a kernel-mode component has called ZwReadFile (Kaspersky, 2014f, pp. 51, 54). Other IRP requests may be an IOC.

Sophos have seen one piece of malware attempting to disable some AV software using the Windows security feature “Software Restriction Policies” (Microsoft, 2017ai), (Wyke, 2014, p. 10). Sophos have also seen one piece of malware change the dropper so that checksum detection will not work (Wyke, 2011, p. 6).

Virus Bulletin report malware choking the anti-virus engine by entering a loop of 271 API calls using a combination of GetWindowThreadProcessId, GetWindowRect and GetDlgItemTextA APIs. This loop has (0x9C40) 40,000 iterations, generating a total of 10,840,000 API calls. Following this the malware sleeps for 200 milliseconds then continues with the rest of the code. (Virus-Bulletin, 2013, p. 6).

Kaspersky have seen (Kaspersky, 2014e, p. 10) an APT analysing 24 VFSes (Virtual File Systems) from victims around the world. These have generally been random names and are located in several places in the infected system. Kaspersky have also seen a VFS stored as a .exe file (Kaspersky, 2016a, p. 7). Elsewhere Kapsersky have observed the modus operandi of one APT which suggested the APT had analysed the patches and implemented a better way to patch new changes (Kaspersky, 2017b, p. 5). In the same attack the malicious payload was spilt into two pieces in two different zones of responsibility, to attempt to hide from anyone who would investigate. Kaspersky go on to state that they have seen this technique at least twice and believe that it is not a coincidence. (Kaspersky, 2017b, pp. 11-12). For this same APT Kaspersky go on to say that the APT:

> “knows the value of quality code, which is why we normally see rudimentary backdoors being pushed during the first stage of infection. Burning those doesn't cause too much impact on the group. However, if the first stage backdoor reports an interesting infection it starts deploying more advanced code, carefully protecting it from accidental detection on disk. The code is wrapped into a DLL loader or stored in an encrypted container, or maybe hidden in a binary encrypted registry value. It usually comes with an installer that only the attackers can use, because they password protect it.”

(Kaspersky, 2017b, p. 24)

F-Secure have seen (F-Secure-Labs, 2015a, p. 3) an APT which first checks to see if an anti-virus product installed on the victim’s machine. Should one be found, it will be checked against a list of product names. Should there be a match, the dropper exits. Newer versions of the dropper will perform checks see if it is in a

VM or a known malware analysis sandbox environment. Again, should either occur, the dropper will exit.

Elsewhere an APT has been observed (Fidelis-Cybersecurity-Solutions, 2015, pp. 5-7) disabling AV software by changing the values of various registry keys.

Microsoft's Windows Defender, amongst other Microsoft products, could be exploited: "Microsoft Malware Protection Engine does not properly scan a specially crafted file, leading to memory corruption.". "Microsoft is not aware of active attacks using this vulnerability" (Microsoft, 2017d, 2017e).

Placing malware in AV products would be a good way to gain persistence as AV products, by definition, need to run to look for malware.

Elsewhere (Barysevich, Moriuchi and Hatheway, 2017, p. 8) have seen persistent malware which is hidden from the Task Manager and can be restored if deleted. Also seen are counterfeit certificates registered under legitimate corporations (Barysevich, 2018, pp. 1-2)

5.17 Coding

Attacks are coded in a range of languages: NET (Yaneza, 2015a, p. 7) Python (.pyc) (Symantec-Security-Response, 2012a, p. 6). C++ and Python libraries (Fireeye, 2015b, p. 28).

5.18 Auto-Obfuscation

Symantec state that obfuscation might deceive automated analysis but not an observant security analyst. They further assert that if code [a PowerShell script] is obfuscated then it is likely to be malicious and suggest looking for a high number of quotation marks or curly brackets (Symantec, 2016a, p. 17). They also briefly discuss entropy. They assert that most PowerShell scripts use ExecutionPolicy and NoProfile parameters which are good indicators of compromise. Other keywords e.g. MSBuildShell may be invoked. PsSetCreate…, Sleep, mutex, DeviceIOcontrol, IRP_. It will also be necessary to identify IPv4 and IPv6 addresses as well as website names.

5.19 Malware File Lengths

Fireeye (Fireeye and BT, 2015, p. 13) point out that with rare exceptions, malware typically has a small size which is usually no larger than a few hundred kilobytes. For example, Fireeye (Lee, Ahl and Hanzlik, 2014, p. 12) have seen one piece of malware (a Web shell) with a size of 73 bytes, for the ASPX version, or four kilobytes on disk. This malware written in a variety of languages such as ASP, ASPX, PHP, JSP, and CFM. (Lee, Ahl and Hanzlik, 2014, pp. 7, 15), Delphi (Nart Villeneuve *et al.*, 2014, p. 7). Other Web shells seen are 619 bytes and 8,527 bytes (Lee, Ahl and Hanzlik, 2014, p. 12)

Kaspersky have seen a file of 27Kb (Kaspersky, 2014a, p. 15) and also a file of 4-5Kb (Kaspersky, 2016a, p. 7) and a file of 20KB (Kaspersky, 2013b, p. 6).

However Fireeye also observed APTs using a 10MB file (it was padded with nulls) presumably this is because most AV do not have the ability to scan larger files (Haq *et al.*, 2014, p. 8)

Loman describes how one piece of ransomware sets files for deletion and then sets their file lengths (Loman, 2019, p. 10) to zero, something which is described as “0 allocation” yet Panda Security discussing the same ransomware make no mention of this technique (Panda-Security, 2017).

Elsewhere a binary of 9511 bytes has been seen (Chohan, DeSombre and Grosfelt, 2018). Although a Unix based attack, this is an odd number of bytes. Analysis of executable file length is discussed in Section 5.19.

5.20 Prefetch

Prefetching allows Windows to fetch files at bootup time (Russinovich and Solomon, 2009, pp. 823-827). One APT has been seen deleting prefetch entries Mandiant (Mandiant, 2017, p. 14) and they can also be manipulated by changing the value of a registry key (Russinovich and Solomon, 2009, p. 824). The Prefetch folder may be viewed at C:\windows\prefetch (Rivera and Inocencio, 2016, p. 88). Kaspersky found a Windows prefetch file in this directory which was created when the malicious attachment was opened. (Kaspersky, 2013e, p. 35) and also use of Prefetch by an APT (Kaspersky, 2017b, pp. 19-20).

5.21 Suspect Domains, Names

Some work has been done looking at suspect domain and file names with "odd" internal properties (Szabo and Huq, 2013) but simply looking at URL and file names which can be changed is not enough.

It might be possible to build a language model of DSN for each language type against the country code tope level domain e.g. .uk and then score the main part of the DSN (the characters between "www" and the country against the language model. Alternatively, sites may be blocked by pattern matching as in the proof of concept Firefox browser add-on software written for this thesis in Appendix H

At the simplest level, the use of no-ip could be blocked by searching for the string "no-ip" (Kharouni, 2015, p. 4), No-ip and six other similar sites (Fireeye, 2014e, p. 13).

Domain names have been registered which appear to belong to legitimate companies. For example, the domain name may be something like legitmatecompany.com or have a small typo. Additionally, some registrations clearly contain false contact details (e.g. telephone and/fax numbers). It would not be too much effort for Domain Name System (DNS) registrars to offer a service to legitimate companies to highlight such registrations or check against online telephone books the legitimacy of the registrant.

5.22 Miscellaneous

Evidence of APTs may be found on corporate log files but (i.e. not on users' machines). The log files on C&C servers for one APT were securely deleted on a regular basis. Additionally this APT had the ability to delete itself from users' computers (Symantec-Security-Response, 2012a, pp. 2-3, 7).This may be indicative of a wider security awareness by this APT and help defenders to categorise this APT's ability.

"System Monitor (SYSMON) is the application programming interface (API) that you use to configure the Microsoft System Monitor ActiveX control. The System Monitor control lets you view real-time and previously logged performance counter data such as memory, disk, and processor counter data"

(Microsoft, 2017af).

After one APT used sysmon they forgot to remove evidence of their attack from the sysmon log files (Kaspersky, 2017b, p. 9).

Novetta have seen evidence of clearing MRU lists (Novetta, Unknown, p. 24)

Fireeye have observed the use of Skype and avatars. An APT may pose as an attractive woman and engage in a Skype conversation with the potential victim. A "personal" photograph would be sent to the victim after ascertaining what device the victim would be using. The photograph would contain malware. (Fireeye, 2015b, pp. 4, 11)

The Process Environment Block (PEB) contains information about every running process. The Thread Information Block (TIB) is a Windows data structure which holds information and every running thread and there is a TOB for every running thread. The PEB and TIB have been used by fake AV and other malware for anti-emulation/anti-debugging. Kuser Shared Data (KSD) is a shared Windows data structure area. Should the KSD not be populated with the appropriate values malware uses that to detect the presence of an emulator (Chandraiah, 2012, pp. 8-9)

5.23 Conclusion

This chapter has supported the first and second Aims and Objectives. It has demonstrated that APTs may place their malware anywhere on a computer's HDD, either within the Windows OS or in free: Malware has been seen placed on a HDD and the HDD controller. From here it can extrapolated that malware will be placed anywhere there is storage: on-chip memory, HDD, wireless card, graphics card etc. This thesis will concentrate on the HDD platters.

It has been shown that APTs can modify the system to hide the existence of files with a given name or less than a given size. One mitigation is to create an executable of length zero and then recursively list files directory list (either “dir” or “forfiles”). Should the file not be listed then there is a possibility that the command used to list files has been modified.

This chapter has demonstrated that APTs use malware that:

- Have the true file extension disguised by icon;
- Have file filename disguised (and hence use more than one full stop in the filename);
- Replaces a legitimate program with a malware that uses the same name;
- Uses Zw, Nt and other routines e.g. SetWindowsHookEx, GetAsyncKeyState;
- is able to gain persistence on machines by various mechanisms of action e.g. Windows Services; Windows Registry; DLL search-order hijacking; Modification of Group Policy Objects (GPO); Use of Common Object Model (COM) objects; Modification of existing binaries; Windows scheduled tasks; Windows Management Instrumentation (WMI); Malicious Windows Security Packages; the Startup folder; and MBR and VBR bootkits;
- uses C&C communications;
- Intra-communicates by mutexes, pipes and mailslots;
- uses encryption;
- use files of a small size;

- may not be detected by the Process Identification (PID) number sequence;

Malware may be identified by the use of little used routines and given keywords.

All of these observations link to the first two Aims and Objectives of this thesis.

6 USING THE LOCKHEED MARTIN INTRUSION KILL CHAIN TO SUPPORT THE SOLUTION

6.1 Chapter Overview

The thesis has, so far, presented the academic background to the research methodology, selection of the LMKC, description of a HDD, a discussion of the difference between a virus and malware and an introduction to APT's work. It will now review, in depth, freely available AV companies' whitepapers and other supporting sources, aligned with the LMKC, to present real-world examples of malware on machines and how the malware gets there. Sub-section 5.3 to 5.9 (inclusive) are the seven LMKC stages. At least one other recently produced taxonomy of APT malware is available (Mitre, 2019a).

The purpose of this chapter is to distil salient information from the whitepapers and use this information to build a solution to treat the malware deployed by APTs. This chapter is aligned with the thesis' first two Aims and Objectives.

6.2 Where the Thesis Solution fits into the Kill Chain

Damballa assert that "APT's are most effectively identified, contained and disrupted at the network level." (Damballa, 2014a, p. 3). This thesis will demonstrate that effective identification, containment and disruption can occur at the individual computer level.

The thrust of this thesis is to increase the business costs of APTs and this may be partially achieved by minimising APT breach longevity or dwell time. Douglas states that Cyber dwell time is measured by tracing the threat back to its origin and that it begins when an attacker enters the network and continues until they are removed or leave (Douglas, 2015, pp. 4-5). However, this thesis argues that measurement should only be to the point of entry on the network or machine, not the origin. In some cases, it may not be possible to discover the origin and, given the philosophy of this thesis – it is agnostic to the origin and intent of the APT – it is not necessary to discover the origin, the malware is on the system; just treat it. However, should there be a need, it may be possible to identify the Reconnaissance stage of the LMKC by examining logs which contain evidence of contact (e.g. use of ping from outside). National Computer Emergency Response Teams (CERTs) are the vehicle for pan-industry collaboration should it be necessary to identify the APT origin.

As has been previously discussed in the selection of the LMKC for this thesis, two of the courses of action in the LMKC are Degrade and Deceive. Therefore, part of the counter-APT strategy may be to leave the malware on the machine. This contains the malware (e.g. restricting access) to waste the attacker's time and hence increase their business costs. An example of a manifestation of this strategy is the use of Honey Pots – machines or files that look attractive to the APT but are designed to waste their time and allow victims to learn more about the APT modus operandi. A further treatment is to infect files on the Honey Pot area with a virus which the attacker may download (LMKC Destroy course of action). Leaving the malware on the machine also provides the opportunity to gather evidence.

Counter-measures should not be done in isolation but as part of a holistic defence and it is worth now taking a short historical detour. AV defences could be mapped against good practice. Schneier provides a view on the security of systems using Attack Trees (Schneier, 1999) which is claimed (Giura and Wang, 2012, p. 3) to be based on earlier Threat Tree work by Edward Amoroso (Amoroso, 1994). However, this type of modelling attacks appears to be based on, and is very similar, to Structured Design Methodology (SDM) (Jackson, 1975) which was used by some organisations in the 1970s. Although SDM is a design methodology and not a security tool, this was a departure from the then norm of using flowcharts to design programs. SDM is, arguably, the forerunner of Object Orientated (OO) programming.

Jackson's design modelled the world through the structure of the data in structure diagrams rather that the sequential operations that were to be performed on the data. Jackson's design viewed the data being made up of records and the operations on those records would be defined in terms of "…three structural forms:

concatenation (sequential flow), iteration (DO WHILE or REPEAT UNTIL) and selection (IF THEN ELSE or CASE).

The GO TO statement should be avoided completely or so far as possible." (Jackson, 1975, p. 1).

In his ACKNOWLEDGEMENTS statement (Jackson, 1975, p. ix), Jackson states that "Many of the sources of ideas for this book are already in the public domain:" and he goes on in the book to acknowledge "… a brilliant description of these ideas by Professor E. W. Dijkstra …" (For the reader who wishes to delve more into the subject, Dijkstra summarises these ideas as "concatenation", "selection" and "repetition" (Dijkstra, Hoare and Dahl, 1972, pp. 16-23).

However, the idea of categorising and compartmentalising data is common to both attack trees and data structures. This is an idea that forms the basis of the software written for this thesis.

Having discussed how this solution fits into the LMKC, the thesis will continue with the white paper review, with the next seven subheadings aligned with the seven stages of the LMKC.

6.3 Reconnaissance

6.3.1 Definition

The LMKC presents the components of this stage as crawling internet websites for information on specific topics e.g. conference proceedings, email address, social relationships as well as technologies. From the victim's viewpoint these are all passive actions in that they can be performed without touching infrastructure or websites owned by the victim. Other techniques may include searches of official and commercial databases, social media, email address harvesting. It is possible to use vulnerability scanners and proof of concept code to targeting specific vulnerabilities (Dela Vega and Ingal, 2010, p. 5) but this particular attack, when successful, also leaves logs on the victim's machine.

6.3.2 Victim Selection

Installation may be indiscriminate or specific. For example, Dela Vega and Ingal provide four different types of vicitm selection:

"

- **Geo-targeting or IP delivery:** This utilizes users' IP addresses to determine their geographic locations in order to deliver location-specific content to their systems.
- **Blog scraping:** This refers to regularly scanning blogs to search for and to copy content using an automated software.
- **Referrer page checking:** This ensures that only users arriving via search engines will be included in the infection chain and prevents security analysts or system administrators from seeing anything malicious when they directly access a doorway page.
- **User-agent filtering:** This refers to distinguishing between browsers to enable OS- specific download of payloads."

(Dela Vega and Ingal, 2010, p. 11).

Sophos have seen the use of poisoning search engine results by filling webpages with topical keywords and phrases which will be harvested by search engine crawlers. Victims arriving via search engines are processed differently than those who arrive by those just happening on the page or are search engine crawlers (Wyke, 2012a, pp. 2-4). Kaspersky have seen the opposite occurring, where users from specific countries, based on IP address, are not infected (Kaspersky, 2015c, pp. 24-25). F-Secure have seen location tracking malware which track the device's GPS location (F-Secure-Labs, 2014c).

This thesis asserts that minimising the electronic footprint is a good start to any defence – if one does not have to provide information, do not offer (advertise) it on the internet.

6.3.3 DNS Hi-jacking, ARP Cache poisoning and Spoof Websites

"DNS is the Internet protocol that resolves human-readable domain names into IP addresses that are assigned to computer servers on the web. Most Internet users automatically use their ISPs' DNS servers and are probably unaware that DNS even exists. DNS changer Trojans discreetly modify computers' settings so these will use foreign DNS servers set up by malicious third parties and translate certain domains into malicious IP addresses. As a result, victims are redirected to possibly malicious sites without their knowledge or consent."

(Trend-Micro, 2012, p. 2)

The standard for domain names is well documented (IETF, 1987b), (IETF, 1987a).

Hacquebord (Hacquebord, 2017, p. 17) documents that, following acquisition of administrator of DNS credentials, a change can be made to the domain's MX record so that it points to a proxy IP address controlled by the APT who can receive all incoming email.

"The address resolution protocol (ARP) cache is a table in computer memory that maps a limited number of IP addresses to their physical adapter addresses. A computer's ARP cache contains its own entry, entries for machines that have made ARP broadcasts to it, and entries for machines to which it has made broadcasts." (Microsoft, 2017a).

Cylance have seen ARP cache poisoning (Cylance, 2016, p. 41) while Kaspersky have seen an APT's C&C where a second-level domain was created without a DNS A-record, i.e., there was no IP address assigned to it. Where there was an A-record, the IP address assigned was typically 127.0.0.1. Some of the second-level domains created by the APTs for their C&C had similar names to the domain hosting the site of a certain real company. The APT's domain was resolved to the same IP address of the real company and the third-level domains resolved to IP addresses assigned to the attackers' actual C&C servers (Kaspersky, 2013e, p. 6).

The Windows hosts file (C:\Windows\System32\drivers\etc\hosts) allows the mapping of any website to any IP address. DiMaggio notes DNS hi-jacking performed by modification of the host file (DiMaggio, 2015, pp. 12-13).

Kaspersky have seen a URL which looks very similar to a legitimate social network URL (Kaspersky, 2015a). They also note that sometimes the malware had a local IP address (e.g. 192.168.1.136) for the C&C. They speculate that there may have been infected machine with no Internet connection, but the APTs needed control over it. The APT had deployed a dedicated local C&C with Internet connection on another compromised machine within the same local network. From this C&C the first machine could be controlled. System administrators try to isolate critical computers from the outside world to decrease the probability of random infection, but this does not always help in a targeted attack (Kaspersky, 2013e, p. 7).

ETags (or entity tags – an optional HTTP header) can be used to check if different URLs (webpages) have the same content or not (Damballa, 2015, pp. 10-12).

Domain name construction may be manipulated: Domain Fluxing, also known as Domain Generation Algorithm (DGA) has been seen generating over 1900 domains in four days (Manos Antonakakis, Roberto Perdisci, Yacin Nadji*, et al.*, 2012, p. 1) Damballa also assert that

"The purpose of a DGA is to:

- Make it impossible for static reputation systems to maintain an accurate list of *all* possible C&C domains.
- Allow the cybercriminals to evade perimeter based network filtering technologies.
- Maintain a small but agile physical C&C infrastructure that only needs to be configured and turned on for short periods of time.
- Provide "just-in-time" registration of domain names to avoid reactive counter-measures and law enforcement.

- Allow crimeware agents to propagate and establish a large infection base without exposing the C&C infrastructure."

(Damballa, 2012, p. 2)

and that some APTs that employ DGAs are able to register their C&C domain and configure DNS in less than an hour to the DGA generating the domain. This DGA domain is then closed within 24 hours (Damballa, 2012, p. 2).

DGA has also been seen by F-Secure (F-Secure-Labs, 2014d, p. 11) and Fidelis Cybersecurity (Fidelis-Cybersecurity, 2015b, pp. 4-10). While McAfee report seeing the use of the Mersenne Twister algorithm random number generator to generate domains (McAfee-Labs, 2016, p. 34). Sophos have seen IP addresses generated randomly (Marosi, 2016, p. 9) while F-Secure have seen malware filenames generated randomly (F-secure-Labs, 2014b, p. 7). However, malware-related domain names may be detected, "weeks" before they appear in public blacklists and security fora (Manos Antonakakis, Roberto Perdisci, Wenke Lee*, et al.*, 2012).

6.4 Weaponization

6.4.1 Definition

The LMKC presents weaponization as:

"Coupling a remote access trojan with an exploit into a deliverable payload, typically by means of an automated tool (weaponizer). Increasingly, client application data files such as Adobe Portable Document Format (PDF) or Microsoft Office documents serve as the weaponized deliverable."

Trojan (horse) malware is software designed to hide its true intent and takes its name from the Ancient Greek story of the Trojan Horse which, after delivery to the Trojans, led to the fall of Troy. A deliverable payload is malware that can be deployed onto the victim's machine.

6.4.2 Persistence

Imperva suggest that the initial detection rate of a new virus is less than 5% and that for some AV companies it may take up to four weeks to detect a new virus (Imperva, 2012, p. 1). This thesis has already discussed length of time of compromise and has described an assertion that attackers spend an average of more than 200 days inside a network before being discovered (InfoSec-Institute, 2017). There is no inconsistency between this statement and the four weeks example. The new virus detection time may denote the maximum time otherwise another time would have been used and some organisations may have poor, or no, anti-virus software, policies, procedures etc. so it could take, on average, 200 days for discovery. One APT managed to "stay under the radar" for over seven years (eset, 2016a, p. 28) in spite of the malware having been seen in the wild since 2008. This APT did not display technical sophistication or novel techniques but they did just what was necessary and sufficient (needed) to succeed. Malware may remain effective for a number of years with, eight years being seen (Fireeye, 2014d, p. 2). Note: effectiveness of malware is not the same as malware being on a system.

A number of APTs (Fireeye, 2016, p. 6), (Singh, Gomez and Malik, 2014), (Fireeye, 2015f, p. 5), (Fireeye, 2015e, p. 6), (Fireeye, 2017a, p. 10) use "off-the-shelf" malware i.e. they have not written the malware but acquired it and perhaps modified it. This means that the same malware may be used across APT and campaigns. Fireeye demonstrates co-ordination and similarities across attacks, producing a report on the subject (Fireeye, 2014f).

Others (Panda-Security, 2015), (Mandiant, 2013, p. 35), (Wüest, 2012, p. 22) provide evidence of attackers using little to no bespoke software; only freely available software provided by Microsoft for Windows 8.1 machines. This is an example of "Living Off the Land" with "Dual Use Tools" (Wüest and Anand, 2017, pp. 7, 15-16) and again may be used by different APTs in different campaigns. This re-use makes forensics analysis harder.

Kaspersky have observed (Kaspersky, 2017b, p. 9) Persistence implemented as Windows service dynamic-link library (DLL), registered inside the group of Network Services (netsvcs).

McAfee categorise fileless malware: Memory resident; Rootkit; and Windows registry; with the memory resident malware using a legitimate Windows file's memory space. One piece of malware loads the code into that memory space and remains until accessed or reactivated. Execution occurs within the legitimate file's memory space, and there is a dormant physical file that initiates or restarts the execution which means that the malware is not completely fileless (McAfee-Labs, 2015, pp. 9-11).

One Fileless APT does not exist as a file on disk but in the registry. It does this by hijacking a valid CSLID which is run when the machine is restarted and uses a protected registry subkey preventing it being opened. This is done by invoking NTCreateKey (O'Murchu and Gutierrez, 2015, pp. 9, 17). Malware injected into the browser ensures persistence should the registry be cleaned. It also has the ability to inject itself into other executables. (O'Murchu and Gutierrez, 2015, pp. 6-7, 11-12).

6.4.3 Zero Days

Zero Days (0-days) are attacks which are not publicly known but which are known to malware writers (Hacquebord, 2017, p. 30). The term zero day refers to the number of days that the attack has been in the public domain. They have been found by analysis and may be sold in the hacking community. As they are not in the public domain, they are unknown to the wider IT community and hence may not be mitigated, although the philosophy behind this thesis allows some zero days to be treated.

Kaspersky (Jacoby and Jartelius, 2013) do not put a figure on the number of zero days used. From analysis of some of the previous breaches, zero-days have been the APT's entry point - the number of attacks where zero-days have been used are still quite low but rising.

6.4.4 Bundlers, Packers and Self-Extracting Files

One method of weaponization (and hence deployment) is to combine at least the initial attack malware into one package. The software that performs the combining ma, generically, be called bundler, packer or self-extracting file; examples of which are a zip file or iExpress.

The use of this type of weaponization is varied. Panda Security have seen a disguised self-extracting file which gives the appearance of being a .pdf file, using the .pdf icon. When clicked on, it extracts six files into a folder which it creates (Panda-Security, 2015, p. 4). Similarly, other bundlers have been seen in attacks, such as ZIP and RAR files (Villeneuve, 2011, p. 10) and a self-extracting RAR file with a .pdf extension (Fireeye, 2015b, p. 11). Kaspersky have also seen this same technique (Kaspersky, 2015a, p. 11). Sophos have also seen malware contained in a zip file as well as using regsvr32.exe (Wyke, 2014, p. 3). regsvr32.exe is a Microsoft utility to register and unregister controls such as DLLs in the registry (Microsoft, 2017p). Zip files may contain self-extractor software (Wyke, 2012b, p. 6). A zip file with a self-extracting RAR file has also been seen by F-Secure, although the ZIP file was hosted on a website (F-Secure-Labs, 2015a, pp. 2-3). Elsewhere three different Industrial Control System (ICS) equipment providers had malware inserted into the software bundles which had been made available for download from their websites (Symantec-Security-Response, 2014, p. 7). Doherty et al have seen malware bundled with legitimate software (Doherty *et al.*, 2013, p. 21).

A packer is a file which contains the code for decompression as well as the data (Fireeye and Mandiant, 2014, p. 15) and packers may be found using the free tool PEiD (Lee, Ahl and Hanzlik, 2014, pp. 4-5). Yan et al reference a now broken reference which claims that in a sample of 735 malwares from in March 2006, more than 92% were packed by packers and crypters from 30 different families (Yan, Zhang and Ansari, 2008). Pan observes that two APTs use the same custom packer (Virus-Bulletin, 2012, p. 25) which, presumably helps link APT groups.

6.4.5 Digital Certificates

The legitimacy of software, if signed, may be verified by digital certificates. The fraudulent certificate attack is specifically highlighted by (Sancho, Hacquebord and Link, 2014, p. 3) where a new root SSL certificate is installed which allows the attackers to display secure content from phishing sites without producing a warning from the browser. Fraudulent digital certificates may be used to make malicious Web sites and malware look legitimate. In one attack, valid certificates were obtained for some of high-value domains, including Yahoo, Mozilla, and Google. Google discovered the use of the certificates in a large-scale, Man-In-The-Middle (MITM) attack for eavesdropping on over 300,000 Gmail users. This attack had been in use "for weeks" before it was detected (Trend-Micro, 2013b, p. 4). This is a modern IT attack similar to the historical example previously discussed - Walsingham's success against the plot to overthrow Elizabeth I, discussed earlier in this thesis which discusses fraudulent signing of messages. Symantec report that another APT gained access to a company's digital code signing certificates and signed a number of Trojans and malicious scripts (Doherty *et al.*, 2013, p. 9).

For many years McAfee Labs has followed the growth of digitally signed malware and they believe that the threat is rapidly expanding while becoming more complex. During this 2103Q4 McAfee discovered more than 2.3 million new and unique malicious signed binaries - a 52% increase over the previous quarter – For 2013 as a while the number was almost 5.7 million, more than triple the 2012 number.

The set of fraudulent certificates came from stolen, purchased, or abused certificates, the vast majority of growth is due to dubious content distribution networks (CDNs). CDNs are websites and companies which allow developers to upload programs, or URLs which link to an external application, and wraps it in a signed installer. This provides developers with a distribution channel and "a cloak of legitimacy" (McAfee-Labs, 2013, pp. 9-10). This technique is not restricted to programs as a signed Java applet has been seen (Katsuki, 2012, p. 3).

Symantec have noted one APT modifying certificate registry values to disable the certificate revocation check and warnings about invalid site certificates as well other values (O'Murchu and Gutierrez, 2015, pp. 12-14).

Weak 512-bit keys have been noted (Kaspersky, 2014a, pp. 9-11) while Kaspersky have also seen an APT's software signed with two fake digital certificates which were supposed to belong to Microsoft Corporation and Broadcom Corporation. In the infection phase, the APT injected a trusted Certificate Authority

(CA) in the certificates chain (Kaspersky, 2014e, p. 6). CAs instruct the system to trust their signatures. More worryingly, Kaspersky have seen a CA compromised which would allow the generation of digital certificates (Kaspersky, 2015b, p. 43). Kaspersky have also seen one APT extract digital certificates and signing malware. The certificates were then used by other ”malware groups” (Kaspersky, 2013e, pp. 2-4).

F-Secure have also seen the use of a false root path and certificates to give the impression of legitimacy (Palumbo, 2014b, pp. 4-5). F-Secure have seen two APTs with the ability to harvest PKI certificates and associated private keys by using PFXExportCertStoreEx (F-secure-Labs, 2014b, pp. 2-3, 8). However, Mandiant have seen an APT using unsigned malware in C:\Windows (Mandiant, 2017, p. 25).

Clearly using fake digital certificates is a problem, the use of which allows attacks to deceive victims and the software on their machines into believing that the malware is legitimate.

6.5 Delivery

6.5.1 Definition

The LMKC presents delivery as “Transmission of the weapon to the targeted environment” with the most prevalent delivery methods for 2004-2010 being email attachments, websites and USB removable media. This section will discuss and review a range of delivery methods. It will demonstrate that a range of attacks and ingress points may be used – anywhere where the system or machine interacts with the rest of the world.

One definition states that delivery may be achieved by any one of: Phishing; Unpatched vulnerabilities; DDos; SQL injection; Cross site scripting; Malicious Websites; Malicious Browser extensions (Organization-of-Amercian-States, 2015, pp. 27, 37). However, this thesis asserts that neither Unpatched vulnerability nor Malicious Browser extensions are delivery methods; the former is a weakness and the latter is deployed malware. The list is incomplete as there are more delivery methods (Phishing and social engineering (Oxford-Dictionaries, 2017) attacks, Vulnerability exploitation, Watering hole attacks, System misconfiguration exploitation, Drive-by-download attacks, Malvertising, 3rd party vendors, Man-in-the-Middle (MitM), Infected equipment, Insider job) as discussed by Huq (Huq, 2016, pp. 30-32). Again, two items are not delivery methods: Vulnerability exploitation, and System misconfiguration exploitation, are weaknesses not methods of delivery.

Even these combined lists may not be complete as there is a least one APT where the method of delivery is unknown (Palumbo, 2014a, p. 3). Browsers may also be injected with malware in what is also known as man-in-the-browser attack (RSA, 2011). Q.v. Walsingham's success against the plot to overthrow Elizabeth I mentioned earlier.

One attack had three different attack methods: In addition to infected websites (an infected medical website), a real-time file-sharing service and a direct transfer via a virtual networking computer (VNC). (Trend-Micro-Cyber-Safety-Solutions-Team, 2016, p. 1). It is not clear why this was the case. It is possible that the APT wanted to ensure that they were successful. APTs will use social engineering to get victims to click on items which contain malware in emails that contain information in which the targets would be interested, (Sancho and Hacquebord, 2016, p. 3). Such emails may contain Word documents with macros enabled so that when the document is opened the malware is delivered. Another common, and old, technique is to use macro-based malware (Yaneza, 2015a, p. 3). Macros were more highly favoured because of their ability to bypass traditional antimalware solutions (Trend-Micro, 2015, p. 17). This view is shared elsewhere where it is asserted that one of the few significant tradecraft differences is that weaponised Microsoft Word macros are more likely to be dropped onto the victim's system that the once-dominant exploit kits (DeSombre and Byrnes, 2018, p. 16). However, Microsoft later changed the default Microsoft Office configuration to prevent macro execution which protected most users. However many large organisations use macros leaving this method of entry open and malware authors have taken advantage of this with simple social engineering tricks, leading to the return of macro malware (McAfee-Labs, 2015, p. 34). McAfee have stated that it is unclear as to whether the malware can change the macro run default setting (McAfee-Labs, 2015, p. 41). This Microsoft Office macro attack view is re-enforced by Szappanos who has also seen VBA code run through the vehicle of "simple social engineering tricks." (Szappanos, 2014b, p. 1). Elsewhere, Microsoft Word documents containing obfuscated code have been seen (Insikt-Group, 2019, p. 11). The subject of obfuscated code will be taken up later in this thesis.

Another type of social engineering is in the form of fake Google Drive™, Gmail™ account or using stolen documents suggesting a legitimate cause and sender. Social media accounts may be used (Pernet and Sela, 2015, p. 11).

A Denial of Service (DOS) or Distributed Denial of Service (DDOS) attack (sometimes known as Email Based Bounce Attack) may be used to hide the real attack (and therefore software is installed on the user's system) e.g. stopping banking customers accessing their accounts so being unable to see that monies are missing. A DOS attack may also be used to swamp the security logs making it more difficult to find the attackers entries (Candid Wüest, 2014a, p. 16). Wüest goes on to speculate that there might be more DDOS attacks from mobile devices or the Internet of Things (Candid Wüest, 2014a, p. 25). However, Toro et al (Toro, Griffin and Settle, 2016, pp. 12, 17-18) point out a DDOS malware (which has a backdoor that allows the malware writer to infect users' machines), where for every ten minutes attacking websites, users receive a point and these points can be traded in for rewards. The malware waits 10 minutes to download the backdoor. On failure with further attempts are made every 1 hour and 10 minutes.

Although not Windows, the Apple App store may have contained from 30 - 300 apps infected by software (one of which showed no evidence of data theft or harm) which had been inserted when app developers used one of a number of compromised app development tool. (F-Secure, 2016, pp. 18-19). One cannot rule out the possibility of infected Windows apps especially given the malware, previously highlighted in this thesis, which has no known method of entry onto the victim's machine.

6.5.2 Pre-installation

One form of malware has been “preinstalled” in some laptop models (Trend-Micro, 2015, pp. 7-8). Pre-installed malware may be more difficult to find as malware has been deployed well before any external connection by the victim. It is inferred that this is a supply chain phishing attack or a well-directed specific attack. Modification of disc head controller micro code to support infection has been seen and this may be a supply chain attack.

6.5.3 Watering Hole and Infected Browser

Potentials victims may be encouraged to visit legitimate websites which have been infected (Hacquebord, 2017, pp. 29-30). Compromised sites have been injected with malware scripts, the Browser Exploitation Framework (BeEF) (BeEF, 2020) redirects to the APT’s private exploit kit: BeEF is useful to an APT when the potential victim leaves browser inactive tabs open. When a user opens a browser tab and visits a compromised website there is a link to a BeEF exploit URL, The APT then has time to perform reconnaissance and try different attacks until the browser tab is closed. Attacks seen are social engineering attacks, password grabbing, and vulnerability exploit (Hacquebord, 2017, pp. 29-30). Browser infection is supported elsewhere (Molinyawe, Hariri and Spelman, 2016). Fireeye note that APTs are using well-known websites which they do not need to compromise to host C&C IP addresses. The APTs use the website for legitimate purposes, such as creating profile pages or posting forum threads (Fireeye, 2015d, p. 3).

Legitimate website may also be compromised using iframes in the HTML code which point to malware on other servers (Kharouni *et al.*, 2014, p. 11) (Goncharov, 2012, p. 23), (Goncharov, 2014, p. 7), (O’Gorman and McDonald, 2012b, p. 3), (O’Gorman and McDonald, 2012a, pp. 3, 6). Once the non-trivial action of compromising a legitimate website has been done there are no visible signs of malicious activity to the victim and such a website may contain links to sites controlled by the attackers. (CrowdStrike, 2014a, pp. 7, 19)

The iframe may have a small area, for example 1x1 with a visibility of “hidden” (Symantec-Security-Response, 2014, p. 10), (Huss, 2016, p. 6) or 2x2, also with a visibility of “hidden” (Symantec-Security-Response, 2016, p. 15). The iframe may be in an infected banner or popup ad (MalwareBytes, 2016). They also may be used to drop a Java applet, possibly in conjunction with social engineering techniques (Katsuki, 2012, p. 2). One way of directing users to malicious websites is the use of homophones in website names and misspellings (Nikiforakis *et al.*, 2014).

While O'Leary et al note the use of recruitment themed emails with links to malicious HTML (.hta) application files. The ,hta files contained links to legitimate job postings on employment websites (O'Leary *et al.*, 2017).

Use of SMB (Server Message Block) protocol in a watering hole attack has been seen (Insikt-Group, 2018a, pp. 4-8,10,13,17-18). SMB is a network file sharing protocol and in Windows is known as Microsoft SMB Protocol. The set of message packets that defines a particular version of the protocol is called a dialect and the Common Internet File System (CIFS) Protocol is a dialect of SMB (Microsoft, 2017w).

6.5.4 Checking and Exploiting the Victim's System Information

Some APTs check to see if the system on which malware is to installed is exploitable or in the right geographic regions. In order to do this some malware fingerprints the victim's system. One such piece, written in JavaScript, identifies installed browser plugins. The victim is then redirected to a URL which, based on the information collected, determines the best exploit to use. (Symantec-Security-Response, 2014, pp. 7, 10). Such redirection needs a compromised website and may place a supercookie on the victims' machines (Wrolstad and Vengerik, 2015, pp. 4-7).

Fireeye have observed an APT checking system information using GetSystemInfo. Should there only be one CPU core then the attack terminates. Fireeye suggest this is to evade sandboxes and other virtualised environments as well as other analysis environments used in reverse engineering (Haq *et al.*, 2014, pp. 5-6). F-Secure have also seen malware not running after checking for a sandbox (F-Secure-Labs, 2014d, pp. 9-10).

Kaspersky have seen (Kaspersky, 2014c, pp. 3-4) collection of systems information - hardware, OS information, computer name user name, group disk space usage, user accounts, TCP and UDP information, processes, directories.

WatchGuard assert that “Most malware runs in user mode (either as a regular user or administrator) (WatchGuard® Technologies, 2016, p. 7) and that virtualised sandboxes look at Windows Application Programming Interface (API) and system calls from the user mode programs. System calls or function calls capture all interactions between a program and its environment (e.g., file reading, writing to registry keys, and the production of network traffic). However, the sandbox is unable to see everything that happens between these calls. It is this blind spot that is targeted by malware authors.

6.5.5 Default Passwords and NetBIOS Credentials

Sophos have observed one popular exploit delivered using an encrypted Excel spreadsheet. Excel will automatically decrypt a spreadsheet with a certain password. The exploit uses more than this but it highlights the use of default passwords (Chantry, 2016, p. 6).

NetBIOS credentials have been obtained (Mandiant, 2010, p. 5) and used to perform NETBIOS log-ons. Mandiant has seen attackers using compromised credentials on as few as 10 compromised systems, as many as over 150 with an average of 40 systems.

6.5.6 Phishing

NIST (NIST, 2013, p. 142) provides three definitions of phishing, one of which is:

“A digital form of social engineering that uses authentic-looking—but bogus—emails to request information from users or direct them to a fake Web site that requests information.”

(NIST, 2013, p. 142)

Forcepoint state that 81% of email scanned by Forcepoint was unwanted (Forcepoint-Security_Labs, 2016, p. 18) while IASCA observed that Spear Phishing (direct targeting of an individual) became a very common method used by those launching malware as an entry point to an enterprise. All it takes is a single user to click a link and open an attachment for the malware to begin to the first phase of an attack (ISACA, 2013, p. 7). Phishing attacks against home emails get around lack of home peripheral protection (Pernet and Sela, 2015, p. 13). This allows them to take advantage of any blurring of business and private lives. Trend Micro observe that email is used as the initial infection vector in 74% of targeted attacks (Hipolito, 2016, p. 4). Fireeye have spear phishing that directs the victim to fake Outlook Web Access (OWA) login pages (Vengerik *et al.*, 2014, pp. 5,7,11,13,14), (Nart Villeneuve *et al.*, 2014, p. 5).

Although spear-phishing is considered a successful form of attack, and in at least one 500 respondent survey respondents cited it as the single biggest attack method against which they had to defend, Unpatched vendor software vulnerabilities was a distant second (Organization-of-Amercian-States, 2015, pp. 6, 46). No figures were supplied for the success rate but a collection of phishing attempts is collated in the following table:

Year	Phishing Rate 1 in	Email Virus Rate 1 in	Reference
2010	442	282	(Symantec, 2013, p. 11)
2011	299	232	(Symantec, 2013, p. 11)
2012	414	291	(Symantec, 2013, p. 11)
2013	392	196	(Symantec, 2016b, p. 8)
2014	965	244	(Symantec, 2016b, p. 8)

2015	1,846	220	(Symantec, 2016b, p. 8)

Table 6-1: Phishing and Email Viral rates

Many phishing emails may contain attachments that, when opened, install malware on the host. The attachments have names that look like valid file names (e.g. MS Word) but on closer inspection are files that are not. This theme is taken up later with the Benign Ware developed in support of this thesis.

CISCO state that only 3% of spam has an attachment, against 25% of legitimate email. In the cases when a spam message has an attachment, it is an average of 18% larger than a typical attachment in a valid email. Therefore these malicious attachments tend to stand out (CISCO, 2013, p. 65).

O'Brien noted an APT taking care to make their spam more authentic with 74% of the APT's spam email using real company names matching the region of origin e.g. "co.uk" for UK companies (O'Brien, 2016, pp. 12-13).

However, some spear-phishing emails may not include attachments; they include a link to a website redirect as described by Kapsersky (and discussed earlier in this thesis). The APT campaign discovered relied on spear-phishing e-mails with links to a malicious website. This website contains several exploits designed to infect the victim's machine. Following a successful infection, the website redirects the user to a benign website. The malicious websites do not automatically infect victims; instead, the malware is hosted in specific folders on the website, which are not directly referenced anywhere, except in spear phishing e-mails (Kaspersky, 2014f, p. 5).

6.5.7 SQL Injection and Cross-site Scripting (XSS)

This attack manipulates the input of a website to gain access to data or the system (Kaspersky, 2016b, p. 7) by making entries to webpage forms to send unexpected commands to database using SQL (Akamai, 2020). Cross-site scripting is when malicious scripts are injected into otherwise benign websites and has been seen in legitimate processes e.g. via comment boxes in forums and discussion boards (TrendLabs, 2015a, p. 24). It is claimed (Goncharov, 2012, p. 17) that XSS is not easy to perform.

6.5.8 "Legitimate Software"

Remote Desktop Connection (Microsoft, 2017c) is a Microsoft® service that allows connection between one machine over a network (possibly the internet) to another using Remote Desktop Protocol (RDP). In order to work certain permissions are need but it has been used by attackers (Lion Gu, 2014, p. 9). Weak passwords and a default port make this easier to use but also easy to identify as does port binding (Chiu, Weng and Chiu, 2014, p. 3) which requires the absence of a firewall. It has also been seen being used with tools to patch RDP servers which support multiple user logins, as well as other hacking tools e.g. Proxy installers and sysinfo collectors (Kaspersky, 2016c, pp. 2, 13). Weak passwords were the cause of another successful attack which brute-forced RDP (Insikt-Group®, 2019b, pp. 10,12,16)

Mandiant (Mandiant, 2010, p. 5) have seen attackers users legitimate software that are seen on other systems. Mandiant conclude that the attackers have installed this software using valid credentials. While Symantec (Symantec-Security-Response, 2012a, p. 18) have seen unique certificate weaknesses used to hijack the Windows Update feature to spread across networks.

As previously discussed, three different ICS equipment providers were compromised and malware was inserted into the software bundles they had made available for download from their websites (Symantec-Security-Response, 2014, p. 7). This observation was confirmed by Langill who states that three different ICS suppliers had their support websites compromised. The APT replaced legitimate installation software on the sites with software that added malware (Langill, 2014, p. 10).

O'Brien notes the use of VNC for command and control of the victim's machine (O'Brien, 2016, p. 24) while others in Symantec note something similar (Symantec-Security-Response, 2015b, p. 15), (Symantec-Security-Response, 2015d,

p. 5). It is not clear from the papers if VNC is legitimate VNC software, modified legitimate VNC software or purpose built.

Even supply chains have been compromised (also known as “interdiction”): in one such attack the update was compromised by a Trojan malware (Candid Wüest, 2014b, p. 15); in another conference proceedings were mailed to conference participants on a CD after the conference and the CD used autorun.inf to execute an installer and attempt privilege escalation (Kaspersky, 2015c, p. 15).

Most PE file infectors change the entry point of the PE file to point to the virus body. This is easy to spot as the malware entry point lies outside the legitimate code section (Symantec, 2005). Entry Point Obscuring (EPO) replaces certain call instructions, with a jump/call as Symantec go on to say, or replace ExitProcess() API calls to point to the beginning of the malware code (Perriot, Ször and Ferrie, 2003, p. 4). Trampoline code, where execution traverses different pieces of code, is discussed later in this thesis in Extended Attributes under the Exploitation part of the LMKC.

Svchost.exe is a process which is a shell for loading services from .dll file and uses parameters at HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost\netsvcs in Windows Services

(Microsoft, 2017b)

McAfee have seen a unmodified copy of the Windows command line executable cmd.exe copied, using a remote command line shell, to the compromised system, renamed svchost.exe, and used. (McAfee®-Foundstone®-Professional-Services and McAfee-Labs™, 2011, p. 17). The use of svchost.exe as a program name to add a Run registry key has been seen elsewhere (Chang *et al.*, 2015, p. 24)

One APT has been seen using a legitimate package, TeamViewer, which is used for remote administration and online meetings with the APT writing custom components for further exploitation (Kaspersky, 2013d, pp. 1-2).

6.5.9 USB devices

APTs can use USBs to infect machines (Symantec-Security-Response, 2015d, pp. 4-5) and may also use rate limiting code which is where the malware will delete itself after a number (three) of infections (Falliere, Murchu and Chien, 2011, pp. 7, 10).

One study found 29 different USB-based attacks and constructed a taxonomy to classify them into four major categories. (Nissim, Yahalom and Elovici, 2017). These categories are: Programmable Microcontrollers; USB Peripherals (sub-divided into malicious reprogrammed and not reprogrammed) ; and Electrical (an example of a modified USB stick capable of destroying sensitive components through a power surge attack). The paper also presented a description of the USB protocol. Not only may the USB device deliver the attack, but it may also be used for data exfiltration.

One APT used a unique USB-based C&C mechanism: basic information is stored on a hidden area of the USB stick. When the USB stick is later connected to an internet connected machine the data is extracted from the hidden area and sent to the C&C infrastructure. When the APT wishes to run commands on the air-gapped network commands are saved in the USB stick's hidden area which are run when the USB stick is later plugged into that network (Kaspersky, 2015c, pp. 13-14).

Another APT has a toolkit, to be used on removable USB devices, which has a module that moves data from air gapped networks to Internet-connected systems. Once networked systems are compromised, the APTs wait for a USB drive to be connect to the infected machine. The USBs are formatted to reduce the USB partition size and reserve several hundred megabytes of hidden data at the end of the disk for malicious purposes. This reserved space is used for a new custom-encrypted partition that will not be recognized by a common OS, such as Windows. The partition has its own semi-filesystem (or virtual file system, VFS) with two directories: 'In' and 'Out'. This attack also bypasses many DLP products, as software that disables the plugging of unknown USB devices based on DeviceID would not prevent an attack or data leakage because a genuine recognized USB drive was used (Kaspersky, 2016a, p. 8).

6.5.10 Firewalls

It is to be expected that firewalls are deployed and employed however with most ICS/ SCADA (Supervisory control and data acquisition) systems firewalls are a rarity (Wilhoit, 2013, p. 2)..

Symantec have observed an APT attempting to lower the computer's security by modifying registry entries' subkeys relating to firewalls (Symantec-Security-Response, 2015d, p. 12). While others have seen have seen modifications to firewall rules (Kaspersky, 2013d, p. 28), (Manos Antonakakis *et al.*, 2012, p. 5).

6.5.11 Java

It is claimed (Jain, Gomez and Singh, 2014) that Java is widely used by developers to the point where many websites and applications will not run without it being installed on a user's system. Such ubiquity lends Java as a malware vehicle, with three common vulnerabilities making up 93% of the observed total.

6.5.12 JavaScript

Hacquebord (Hacquebord, 2017, p. 16) has seen JavaScript, which was obfuscated, hosted on an APT. It may also be embedded in a pdf document to make an HTTP request to download malware (Singh and Bu, 2014, p. 16). McAfee Labs Threat Advisory have seen obfuscated JavaScript delivered in a .zip file as an email attachment (McAfee, 2018a, pp. 1-2).

6.5.13 Attachments, Office Documents, RTF and Flash

"With Automation, you can do programmatically almost anything that the user can do manually in Microsoft Office Word. However, if you have lots of text that you want to enter and to format, it might require lots of code. If you can represent the data as a Rich Text Format (RTF) string, you can frequently reduce the Automation code. You can create an RTF string, copy the RTF string to the clipboard, and then paste the RTF string into the document."

(Microsoft, 2018n)

TrendLabs (TrendLabs, 2014, pp. 5,7) note that a variety of attachment types may be used for infection and Hacquebord (Hacquebord, 2017, p. 28) notes an APT that started to use RTF and other Microsoft® Office® documents embedded with a Flash file. The Flash file uploads information on the victims' system to a remote server and the remote server may respond with a chain of exploits, zero-days and privilege escalation that infects the victim's computer. While Sophos have also seen the use of RTF (Chantry, 2016, pp. 6-11). Although the users' system may present a warning about the file (Sancho, Hacquebord and Link, 2014, pp. 2-3)

6.5.14 Potentially Unwanted Program (PUP), Potentially Unwanted Application (PUA) and Malvertising

Malware may be delivered through infected, or malicious, advertisements (malvertising) (Huq, 2016, p. 31), (Trend-Micro, 2015, p. 4). This method allows APTs to promote malware laden advertising on legitimate websites on which victims then click. Trend Micro (Trend-Micro, 2015, p. 10) asserts that for normal users malvertisements are one of the worst threats as they can hurt users even when the right things are being done. Malvertisements can affect users who do not click links, have updated security and only go to trusted sites. Trend Micro assert that the only defence is luck. This thesis shall demonstrate otherwise.

This thesis asserts that malvertisements are particularly bad as they make use of the advertising companies' data to target users. i.e. the advertising companies are doing the reconnaissance for the APTs by highlighting users that have the attributes that the APTs are looking for.

In 2014 Panda Security (Panda-Labs, 2014, p. 4) reported a rise in PUPs through the use of software bundlers which install PUPs along with software that the user wishes to install. This rise continued into 2015 (Panda-Labs, 2016, p. 8)

At least one piece of malware generates fraudulent click to generate revenue (Wyke, 2012b, p. 22), (Damballa, 2014b). Although not necessarily an issue for the users of the host computer it could be for the entity which wishes to advertise their products or service.

Sophos have seen fake-AV promulgated by malvertising (Chandraiah, 2012, p. 5) while Kaspersky have seen malvertising redirecting to malicious websites (Kaspersky, 2013d, p. 6). In another click-fraud attack botnet (O'Murchu and

Gutierrez, 2015, p. 3) the malware silently uses a hidden browser window to visit web pages and display advertisements in that window. The APT gets paid for every advertisement shown and although the amount per advertisement is small, the compromised computers are able to show thousands of advertisements each day. A complication for victims is that the advertisements can contain malware which means that a compromised computer will often contain other threats, including ransomware.

6.5.15 MBR

Mandiant (Madiant-Consulting, 2016, pp. 30-31) have identified at least one MBR bootkit which iterates over all NTFS formatted logical drives and attempts to store malware, in two places – one as a disk file and another in unallocated sectors near the end of the file system. The latter is a backup in the event the former is removed. The installer then overwrites sections of the malicious MBR over the legitimate MBR, preserving the original partition table and error messages. The malware ensures that the MBR is only modified on the physical drive that contains the file system where %WinDir% (i.e. where the Windows operating system is installed) is located and that the MBR has not been previously modified.

6.5.16 Insiders

Last, but not least, of the specific delivery mechanism is the insider. A human who places malware onto a machine and who is either willing (Kaspersky, 2016b, pp. 10-11), or not (Falliere, Murchu and Chien, 2011, p. 7).

6.5.17 Miscellaneous

Trend Micro (Pernet and Sela, 2015, p. 7) have seen a malicious version of a legitimate penetration testing tool which has been infected using a malicious Microsoft® Office® macro and in another similar case (Chiu, 2015, pp. 3, 4, 9) " … the ring 0 port re-use techniques to hide the backdoor's communication.". The latter also is able to check the system tray (systray) to see if there is any anti-malware protection and clean event logs to make incident response more difficult. Ring 0 (kernel) use was previously highlighted - "Backdoors with this capability use the Network Driver Interface Specification (NDIS) or Windows Filtering Platform

(WFP) to listen to already-open ports on a target system. " (Chiu, Weng and Chiu, 2014, p. 14). Those backdoors with port re-use in Ring 3 can be used in user mode (Chiu, Weng and Chiu, 2014, p. 15). Other malware intercepts WinAPIs in UserMode, Ring 3 (Goncharov, 2012, p. 11). This means that the malware (bot) does not need drivers or calls in Ring 0.

A Ring 0 (kernel) rootkit subverts the Windows kernel and can "… hide files, folders, registry keys, ports and processes" and has to "… operate as a system driver … The major drawback of this implementation is that the rootkit always comes with two different binaries (one SYS driver and one EXE that installs the driver) … Moreover, the installation process requires interaction with the Windows Service Control Manager (SCM), or alternatively uses the undocumented API ZwSetSystemInformation. Both methods can create some evidence of the threat's presence or can be blocked during the installation phase." (Florio, 2005, p. 4)

Luo and Yan provide an overview of Fake Apps (Luo and Yan, 2014) and at least one piece of malware originated from Yahoo! Instant Messenger (Lee, Ahl and Hanzlik, 2014, p. 17). While Rodionov et al provide an overview of bootkits (Rodionov, Matrosov and Harley, 2016). A Volume Boot Record (VBR) is the boot record for a partition as opposed to a MBR which is the boot record for the device and contains the partition table. The Initial Program Loader (IPL) is the first 15 sectors after the VBR. One attack modifies the 'Hidden Sectors' field of the VBR, while all the other data and code of the VBR and IPL remain intact (Rodionov, Matrosov and Harley, 2014, p. 322).

The use of decoy documents has been seen by F-Secure (F-secure-Labs, 2014b, p. 5). In this case an order receipt is used as the decoy.

Finally, Microsoft promulgate end of life dates for their software (Microsoft, 2017ak). However at least one previous, out of support Microsoft operating system (Windows XP), was still causing problems accounting for around 18% of infections (Kaspersky, 2014d, p. 51)

Kaspersky have seen the creation of directories to support the attack (Kaspersky, 2013d, p. 39).

One APT uses multiple delivery methods e.g. torrent file-sharing sites for indiscriminate distribution, social engineering, lures and websites, and a video downloaded (FireEye, 2020, p. 7)

6.6 Exploitation

6.6.1 Definition

Lockheed Martin define exploitation as the triggering of the intruders' code. This may be to exploit the users or leverage the OS feature which auto-executes the code (Hutchins, Clopperty and M., 2011).

6.6.2 Victim Selection

Not all malware that arrives on a system needs to run – the APT needs to check if the potential victim is one that can, or should, be exploited.

One piece of malware displays unique info to product ID and CD-Key of certain software if it is installed (Organization-of-Amercian-States, 2015, p. 34). Although not elaborated on, this thesis infers that identification of such products aids attack selection.

Google's public DNS server at 8.8.8.8 (Google, 2019a) allows network administrators to change DNS operator from the ISP to Google Public DNS and has been seen as a method of geolocation (Wyke, 2012b, p. 19). A legitimate global website uses geolocation to cater for its customers. Malware on the victim's infected machine abuses this server to obtain the address of the legitimate website and then sends a request to that website. The website responds with geolocation information. With another attack it is not clear to what use 8.8.8.8 is being put (Kaspersky, 2015b, p. 35).

6.6.3 Booting

APTs may delete system files in such a way that means that Windows fails to boot up (Madiant-Consulting, 2016, p. 11).

There are two safeboot registry keys which allow the minimum set of device drivers and network capabilities respectively.

“HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Minimal”
“HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Network”

One APT deletes entries in both of these keys (Virus-Bulletin, 2012, pp. 22-23)

> “*Microsoft* claimed that the release of the Secure Boot technology heralded the end of the bootkit era. In practice, Secure Boot just switched the focus of the attackers towards a change in infection strategy. There are still many active machines in the world with old operating systems where Secure Boot is not supported. For non-targeted attacks, just intended to build botnets, cybercriminals will continue to use old bootkits and bootkit techniques for MBR/VBR infection until a critical mass of users have switched to modern hardware and operating systems.”

(Rodionov, Matrosov and Harley, 2014, p. 325)

6.6.4 Privilege Escalation

Privilege Escalation may be divided into Horizontal Privilege Escalation and Vertical Privilege Escalation (ICANN, 2020). The former is access to accounts with the similar privileges; the latter is elevating the single user privileges possessed.

F_Secure (F-Secure-Labs, 2014a, p. 6) have seen an APT using social engineering to install their malware. Originally the APT preferred to use the name of the Windows registry editor (regedt32.exe). It is presumed that this is because the editor needs administrator rights and would try to request for the highest available rights. This process produces a notification message to the user who, it is asserted, is more likely to grant permission if it appears. However, experienced users may be less likely to be taken in, hence decreasing the likelihood of success. This same APT also tries to bypass UAC default settings by getting a SHIM database to instruct SndVol.exe (volume control, a “Windows executables that will be automatically elevated upon execution because it is thought to be safe.”) to execute cmd.exe instead, which can then be used to install the malware while in an elevated state (F-Secure-Labs, 2014a, p. 8), (Fireeye, 2017b).

Kaspersky have seen an APT check the EnableLUA value of the "HKLM\Software\Microsoft\Windows\Current Version\Policies\System". Should it be enabled then the installation defaults to user installation to evade any user notification (Kaspersky, 2014f, p. 11).

F-Secure have a observed an attack related to TESTSIGNING option (F-Secure-Labs, 2014a, p. 8). On 64-bit Windows, Microsoft enforces a security policy requiring signed drivers. Signing identifies drivers to its author, reducing the number of malware developers willing to take the risk. To allow developers to test drivers during development, Microsoft provides a TESTSIGNING boot configuration option. In this, a watermark is displayed on the screen to making it obvious to users and to prevent malware from exploiting the option. One APT uses the TESTSIGNING option to load malicious driver components and to hide this change from the user, the malware deletes the watermark by removing the relevant strings in the user32.dll.mui of the system. In Windows 8 and up, the strings are no longer stored in user32.dll.mui, so this attack will not work

In another attack F-Secure (F-Secure-Labs, 2014a, p. 9) have observed an APT looking for an existing inactive driver service that is disabled or set to start on demand. The APT drops the driver component using the corresponding path of the service, overwriting the existing driver if necessary. This service is then set to start automatically enabling it to persist after a reboot. Using a legitimate service, the APT hopes that malicious driver will be overlooked by administrators or investigators. The driver component of this APT uses IOCTL (I/O Control) buffer command codes (Code 6 loads a driver into memory) with the 32-bit version containing additional, incomplete routines for hiding processes via DKOM Direct Kernel Object Manipulation

Another APT seen by F-Secure embeds objects in .docx email attachments. This requires the recipient to action a prompt to execute the embedded executable malware which, for the first stage, gathers basic system information and screen shots (F-Secure-Labs-Malware-Analysis, 2017, p. 4). This technique is not uncommon (Sardiwal *et al.*, 2017)

One way to bypass Windows User Account Control (UAC) for 64-bit Windows is for the malware to use the Windows Update Standalone Installer (WUSA) to copy its DLL into a protected folder (*C:\Windows\System32\oobe)* as *wdscore.dll***.** It will then execute *oobe.exe* to side-load the malicious *wdscore.dll* instead of the legitimate one. For 32-bit Windows this is a mixture of IFileOperation code and *CompMgmtLauncher.exe* (Settle, Griffin and Toro, 2016, p. 31). See Section 6.6.6 for a definition of side-loading.

Proof of concept UAC elevation by code injection is available (Davidson, 2009), if a little old and Sophos have seen one dropper inject dll malware into all running processes for which it has permission (Wyke, 2014, p. 9) while Kaspersky have seen (Kaspersky, 2013d, p. 48) one piece of malware replicate itself with several names, to several locations, the final location residing in the All Users\Application Data directory and a run key for itself added.

Use may also be made by malware of GetCurrentProcess (Microsoft, 2018t) to which "the system returns a pseudohandle with the maximum access that the DACL allows to the caller" A DACL is a Discretionary Access Control List (Microsoft, 2018i).

6.6.5 Directory Traversal

Symantec report (Symantec-Security-Response, 2015d, pp. 21-22) one APT recursively traverses the directory searching for files to infect. While Symantec observe (Symantec-Security-Response, 2012c, p. 3) an APT re-routing OS APIs by changing the address of these APIs to point to their own code. McAfee have seen (McAfee®-Foundstone®-Professional-Services, 2017, pp. 1-2) malware with the capability to spread via Admin$, C$ and D$ shares as well as trying to spread to other machines in the network by trying to connect to the hidden shares ADMIN$, C$WINDOWS and D$\WINDOWS

6.6.6 DLL Side-loading etc.

It should be noted that DLL side-loading, DLL Search-Order Hijacking, DLL-Hijacking and DLL pre-loading, are related.

The nature of windows is such that the OS will attempt to find the required DLL by searching, in order, a well-defined set of directories (Microsoft, 2017k). Microsoft further state that:

"A system can contain multiple versions of the same DLL. Applications can control the location from which a DLL is loaded by specifying a full path or using another mechanism such as a manifest. If these methods are not used, the system searches for the DLL at load time …"

(Microsoft, 2017j)

Microsoft go on to present the factors which affect searches at load time. The registry key **"**HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\KnownDLLs" plays a part.

Stewart (A. Stewart, 2014, p. 3) states that "DLL side-loading takes advantage of Windows' side-by-side (SxS or WinSxS) assembly feature, which helps manage conflicting and duplicate DLL versions by loading them on demand from a common directory." And later notes that the malware under analysis placed a spoofed malicious DLL file in a Windows' WinSxS directory so that the operating system loaded it instead of the legitimate file. The SxS listing is in the registry keys "%TEMP%\RarSFX%\%ALLUSERS PROFILE%\SXS\" or "%TEMP%\RarSFX%\%ALLUSERS PROFILE%\WinSxS\"

Mitre comment (Mitre, 2015-2019) that programs may specify the DLLs to be loaded at runtime and that any programs that improperly or vaguely specifies a DLL may be open to a vulnerability where an unintended DLL is loaded. Side-loading occurs when Windows Side-by-Side (WinSxS) manifests are not explicit enough about characteristics of the DLL to be loaded and APTs may take advantage of legitimate programs that are vulnerable to side-loading to load a malicious DLL. APTs are thought to this technique to mask actions they perform under a legitimate, trusted software process or system.

Harbour asserts (Harbour, 2010) that on a 64-bit laptop Windows 7 there are no less than 1032 path and DLL name combinations where a DLL could be placed such that it would automatically load at some point during normal boot-up. Harbour further asserts, without evidence, that with 64-bit malware DLL the number would be higher as there are more 64-bit processes running at boot time.

Side-loading has been seen as method of placing code into signed Java executable (Settle, Griffin and Toro, 2016, p. 22) and also seen by Mandiant (Glyer, 2010, pp. 3-4), pwc (pwc-BAe, 2017, p. 18) and F-Secure (F-Secure-Labs, 2014d, p. 5).

Kaspersky (Kaspersky, 2013e, pp. 10-11) have observed that for one APT the mechanism of action for the malware is that if a benign application depends on Windows winmm.dll (located in %WINDIR%\System32\winmm.dll) and the malware with the same name (winmm.dll) is in the folder of benign application, the malicious library will be loaded instead of the system one. For example, the APT place a malicious library in the %WINDIR% folder. This folder also contains explorer.exe. Kaspersky go to note that this enables the APTs to ensure that the malicious DLL is loaded at system start-up: explorer.exe loads the malicious winmm.dll from the %WINDIR% folder when it launches during system start-up. The APT used a tool which had been developed by security researchers to analyse malware: The program facilitates the analysis of malicious libraries. Input is a DLL and it produces C code which hooks the functions included in the library. This C code is then compiled back into a DLL, which can then be used as a proxy and provide a way to analyse behaviour of malicious files.

6.6.7 Stub Malware

Stub malware leaves a minimal forensic footprint on a machine and can allow an APT to upload C&C software dynamically. This software can be uploaded as and when needed and, when no longer needed would only leave a footprint in virtual memory (Mandiant, 2010, p. 20).

Stub malware may also decrypt other malware which is used for the main part of the attack (Goncharov, 2012, p. 1), (Goncharov, 2014, p. 8).

6.6.8 Browser related Malware

Malware may insert itself into browsers (Kruse, Hacquebord and McArdle, 2012, p. 10), (Fireeye, 2015b, p. 26) or flaws in browsers (TrendLabs, pp. 7, 8), (Chen and Li, 2015, p. 5), (O'Brien, 2016, p. 20). Kaspersky have seen one APT inject malware into EXPLORER.EXE (Kaspersky, 2014b, p. 12) as does another (eset, 2015, p. 16).This is called a Man in the Browser (MITB) attack. However, as Chen and Li point out Internet Explorer patches had, at that time, driven APTs to Adobe Flash Player. This thesis notes that Internet Explorer is not the only browser available.

Symantec reports (Symantec-Security-Response, 2015b, pp. 8-9) that one APT has at least two MITB methods. The first checks every webpage visited by the user and when there is a match the user is redirected to a malicious server which sends the user a fake web page and harvests credentials before being redirected again to the genuine webpage. With the second attack the APT alters the legitimate webpage in the fly with malicious code. Again, the APT harvests credentials.

In 2014 Kaspersky saw browsers accounting for 42% of vulnerable applications (Kaspersky, 2014d, pp. 24-25) while Damballa have seen malware running under explorer.exe child and also execute its own child explorer.exe (Damballa, 2014b, p. 7).

Sophos have observed one piece of malware which determines which browser and plugins are installed on the victim's machine. It then selects from a variety of exploits the malware that will be effective against the victim's machine (Wang, 2013, p. 4).

The orchestrator (part of the malware suite co-ordination software) for one APT is injected into explorer.exe (eset, 2017, pp. 7,16); a technique is known to academia (Nissim, Yahalom and Elovici, 2017).

While F-Secure have seen persistence through browser related registry keys HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run and HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run they have also seen modification of Internet Explorer registry keys during installation to prevent any prompt related to running Insafe ActiveX (F-Secure-Labs, 2016, p. 11).

Many sites, including those of financial institutions, fingerprint machines as an anti-fraud technique. However software is available to customise fingerprints to get around these anti-fraud techniques (Insikt-Group®, 2020b).

Attacks against Address Space Layout Randomization (ASLR) and Data Execution Prevention (DEP), which include leveraging Use After Free (UAF) (Fireeye, 2013, p. 19) are noted.

Malware may turn off browser warnings (Kruse, Hacquebord and McArdle, 2012, p. 7) and may use the "res://" protocol in older versions of Internet Explorer (Villeneuve, 2011, p. 10).

6.6.9 Extended Attributes

"Kernel Extended Attributes (Kernel EA's) are a feature added to NTFS in Windows 8 as a way to boost the performance of image file signature validation. It is an expensive operation to verify an images signature. Therefore, storing information about whether a binary, which has previously been validated, has been changed or not would reduce the number of instances where an image would have to undergo a full signature check."

(Microsoft, 2017t).

For Extended Attributes (Ciubotariu, 2012) and Alternate Data Streams (Palumbo, 2014a, p. 4), the former example uses ZwSetEaFile to write the malware into the Extended Attributes (EA data) of the %System%services.exe file and ZwQueryEaFile to retrieve and execute it. While the latter also notes the use of Trampoline code which will traverse different pieces of code before executing the required external subroutine before returning to the payload (Palumbo, 2014a, pp. 6,8-9).

Kaspersky have seen (Kaspersky, 2014e, p. 5). an APT use the EA to hide stages. Originally, EA were implemented in Windows NT for compatibility with OS/2 applications but they are also in later versions of Windows: 2000, XP and

Vista. The malware hides its modules in NTFS EAs and splits large files into several limited size blocks. These are then dynamically joined, decrypted and executed in memory.

6.6.10 Buffer Overflow

Buffer overflow is when data being written to storage overflows the allocated area into adjacent storage areas (Arpaci-Dusseau and Arpaci-Dusseau, March 2015, p. 5). Malware may also take advantage of buffer overflow to help with the attack (Werthmann, 2006), (Fireeye and Mandiant, 2014).

C buffer overflows may be mitigated. For example two program transformation fixed buffer overflows which originated in unsafe library functions and operations involving bad pointers (Shaw, 2014). The software automatically fixed all buffer overflows on over 4,500 programs involving over 2.3 million line of code in a NIST reference dataset.

6.6.11 Heap Spray

The heap is used for dynamically allocated memory. Such space, in C, is requested and freed by the routines malloc and free, respectively (Arpaci-Dusseau and Arpaci-Dusseau, March 2015, p. 5). The heap is vulnerable to a heap spray (Fireeye and Mandiant, 2014, p. 14) where the APT places code in the heap to allocate storage which may then be filled with the required data.

6.7 Command and Control (C&C)

6.7.1 Definition

Lockheed Martin assert that malware must beacon outbound to a control server to establish a C2 channel (Hutchins, Clopperty and M., 2011). Not all malware will need a C2 channel as the action on objective may be denial of service through MBR overwrite, for example.

6.7.2 Topology

APTs need a command and control infrastructure. This may mean registering websites and a DNS. NO-IP (no-ip, 2020) is a dynamic and managed DNS service that has been used by at least one APT (Yaneza and Mendoza, 2015, pp. 4, 20, 30).

In another attack a piece of malware is configured so that the victim's machine is the server and the client is the C3 controller (Villeneuve and Bennett, 2014).

Use of IP addresses linked to Digital Subscriber Line (DSL) - http://computer.yourdictionary.com/cable-dsl-gateway) networks (Sancho *et al.*, 2012, p. 1) can make it harder to identify attackers. For the purposes of this thesis identification is not necessary but commonality or linkage between APTs may aid analysis of attack.

Unique identifiers are used by APT campaigns (Villeneuve and Sancho, 2011, p. 8) (Hacquebord, 2017, p. 28), (Trend-Micro-Incorporated, 2015, p. 3). These may include the registration of computer name, IP address and service pack (Dela Paz, 2012, p. 6).This is presumably an APT business governance and project control technique.

C&C topology ranged from the very simple, as in the no-ip example above, to complex and numerous as described by Sancho et al (Sancho *et al.*, 2012, p. 1): in this case the APT aggregated at least 60 C&C servers. This allowed the APT to cover their tracks, as having the C&C server in the victims' corporate networks means very little C&C traffic leaves them. The APT's use of compromised machines and dynamic DNS services allowed them to hide their presence by confusing their activities with data from legitimate individuals. C&C servers may include dedicated, registered, DNS servers (Villeneuve and Sancho, 2011, p. 5) to compromised servers. (Sancho *et al.*, 2012, p. 1). However geographic location of server is not necessarily a reflection of user location. An IP address in one country was used by a Virtual Private Network (VPN) provider in another (Villeneuve and Sancho, 2011; p. 6). The number of instances of separation of user from server geographic can only increase with the take up of cloud storage.

Other C&C servers change over time (Ciancaglini *et al.*, 2015, p. 16) but bots can use the same pattern strings over time (Ciancaglini *et al.*, 2015, p. 19). One piece of malware uses the C&C server to change the URL of the doorway for redirection every 10 minutes (Dela Vega and Ingal, 2010, p. 9) while another uses a site redirection service (Dela Paz, 2012, pp. 8, 10).

Fireeye report (Nart Villeneuve *et al.*, 2014, pp. 13-14) seeing C&C infrastructure which relies primarily on domains obtained from dynamic DNS providers. The APT frequently changes IP addresses and often point their C&C domains to legitimate IP addresses when they are not in use.

Some botnets make use of The Onion Router (TOR) which contains a network of more than 3,000 volunteer nodes to allow for anonymous communications by public key cryptography (PKC) enabling concealment from network surveillance tools (Ciancaglini *et al.*, 2013, pp. 5, 21).

Analysis of one C&C server (Symantec-Security-Response, 2015a, pp. 10, 13) found that the APT used encrypted virtual machines and multi-staged C&C servers to make it difficult to investigate their activities.. Virtual Box (virtual machine software) and Truecrypt (which can encrypt a file system or single file) were installed on the server. Analysis of the attack is slim but it is likely that the 400GB ".rar" file was an encrypted Truecrypt file which contained a Virtual Box virtual machine. The APT decrypted and ran the virtual machine, redirecting SSH traffic from the physical hosting server to the virtual machine. This meant that the APT could control compromised systems from within the virtual machine. This type of attack hinders analysis without a live memory image of the C&C server.

While F-Secure (F-Secure-Labs, 2014d, p. 2) found that in one attack the payload is a kernel-mode driver protected and obfuscated by VM code, which cannot be executed natively by Windows. The VM code is a series of byte code that needs an interpreter to translate it to native Windows machine code. The byte code cannot be disassembled by a common disassembler tool. This is effective at protecting the malware, as it prevents researchers from understanding the program's functionality, or at least increases the analysis time needed on the problem.

Mandiant observed (Mandiant, 2010, p. 16) one APT continually moving laterally through the corporate network starting and stopping backdoors. This reduced the likelihood of a compromised host being identified, as the systems communicating with the APT's attack infrastructure continually changed. Multiple backdoors allowed the APT to maintain a presence on the network.

Finally, registry keys may be used to store C&C information (TrendLabs, 2013c, p. 12), or APT configuration data (Fireeye, 2014a, p. 35) and F-Secure have seen the use of the cloud to host malware (F-Secure-Labs, 2015b, p. 7).

Kaspersky noted (Kaspersky, 2017b, p. 7) one APT's attack where most of the hosts analysed were not directly controlled via a C2 server - they connected to another internal host that relayed TCP connection to the C2 using mechanism Kaspersky called the "TCP Tunnel Tool". This tool allowed the APT to chain the victim's internal hosts and relay communications to the real C2 server. This technique made it harder for administrators to identify compromised hosts as local connections usually seem less suspicious.

Fireeye report an APT receiving tasking via POP and exfiltrating data by email attachments using SMTP (Fireeye, 2014a, p. 23).

6.7.3 Anonymous FTP Server

An anonymous FTP server allows attackers to connect remotely victim's compromised machine and browse the file system, upload, download, or delete files and execute commands. (Symantec-Security-Response, 2015d, p. 6). This is analogous to TOR which has been used as a data exfiltration vehicle (Villeneuve, 2011, p. 14).

Sophos have seen cryptomining software try to spread in the manner of a worm to other machines using FTP (Marosi, 2016, p. 9), while Kaspersky have also seen cryptomining as an aside to other malware (Kaspersky, 2016c, p. 14)

Fireeye have seen the use of HUC Packet Transmit Tool (HTRAN) which proxies connection through intermediate hops to help disguise true geographic location (Haq *et al.*, 2014, pp. 17-20). Mandiant have also seen HTRAN (Mandiant, 2013, pp. 41-42).

6.8 Installation (and Uninstallation)

6.8.1 Definition

Lockheed Martin define installation as the part of the Kill Chain which allows APTs to maintain persistence (Hutchins, Clopperty and M., 2011).

6.8.2 Persistence

Installation malware may delete itself without leaving a trace (Sancho, Hacquebord and Link, 2014; Trend-Micro, 2012, p. 4) making it more difficult to detect as the result looks like a system configuration change. However some must leave a trace otherwise researchers would not be able to state that an APT existed on the machine (Chang *et al.*, 2015, p. 26). Kaspersky have seen an APT using an integrated timer which, it is presumed, is designed to self-destruct if commands are not received from the C&C after a period of several months (Kaspersky, 2015c, p. 9). Kaspersky have also seen a multi-stage malware with the ability to delete itself from the system but it does leave artefacts – the encrypted Virtual file system (VFS) from which the malware is loaded - (Kaspersky, 2014e, pp. 8, 22) as well as a self-delete after execution of a main procedure (Kaspersky, 2013d, p. 46).

Kaspersky have seen an APT gaining persistence on domain controllers using a Windows LSA (Local System Authority) (Microsoft, 2017v) password filter. This runs every time any domain, local user, or administrator logs in or changes a password and enables the APT to harvest the password in plaintext (Kaspersky, 2016a, p. 6). |At least one VM has been seen to be infected (Katsuki, 2012, pp. 9-11) which indicates that not all malware runs on the host system.

However not all APTs have the ability to remove all of their malware (Kaspersky, 2017b, p. 5) which may indicate an absence of, or incomplete, asset register of their malware.

6.8.3 Ports

Unless the APT is destructive there is a need for C&C and to exfiltrate the required data. However, for APTs who need C&C, and whose Action on Objectives is data exfiltration, ports are will be needed unless it is an insider attack using storage devices e.g. USB memory stick. This methodology may be constant over time and therefore easier to detect. Exfiltration may done using standard ports (e.g. 443) or non-standard ports (Villeneuve and Bennett, 2012, p. 4). Port numbers may be checked with IANA (IANA, 2020) for the reasons for their use.

Mandiant state that 83% of malware used TCP port 80 or 443 with 17% using another port (Mandiant, 2010, p. 8). For example, Fireeye have observed the use of port 55555 (Fireeye, 2015b, p. 28). However, in 2013 Fireeye (Fireeye, 2014c, pp. 5-6) saw a victims machines sending C&C to 33,697 different port numbers, which is almost 50% of the available ports assigned by Internet Engineering Task Force (IETF) process for standards-track protocols. The most frequently used ports were under 5000, with a large concentration of use of ports in the 15,000 to 16,000 range. This suggested to Fireeye that many corporate and government networks were infected by the malware known as ZeroAccess which uses these port numbers for peer-to-peer communications. In the 1 – 1,000 range, Fireeye found very limited usage in the 400s, which is often used for Secure Sockets Layer (SSL). There is no inconsistency between the Mandiant and Fireeye observations; Mandiant present density of the concentration ratio of port numbers while Fireeye presents spread.

Firewall protection may be by-passed by using ports that are being legitimately used, i.e. not blocked, by the system (Chiu, Weng and Chiu, 2014, p. 4).or FTP and SOCKS Bot (Goncharov, 2012, p. 4)

High level language access to ports is done using the winsock.h (Microsoft, 2019h) and winsock2.h headers (Microsoft, 2019g) which contain the sockets and binding WSA (Windows Sockets API) functions (Neville and Gibb, 2013, p. 36). This will be returned to later in the thesis.

6.8.4 Timing

Some attacks may be timed (Villeneuve and Bennett, 2012, p. 11), (Villeneuve and Sancho, 2011, p. 8) and some remain quiet for a time after infection (Sancho and Hacquebord, 2016, p. 12), (TrendLabs, 2014, p. 12). One, after "checking in" had a backdoor to take advantage of OS features such as sleep timer. It could accept a sleep command allowing it to be dormant for varying periods of time before contacting C&C servers (TrendLabs, 2014, p. 4). Mandiant identified (Mandiant, 2010, p. 15) malware configured to sleep for anywhere from a few weeks to a few months, with one piece of malware able to sleep for over a year. Symantec have observed the use of "Sleep" in VB (Suenaga, 2012, pp. 20, 27-28, 43-46) and elsewhere (Symantec-Security-Response, 2015d, p. 20) where it would sleep for 0x7530 milliseconds (30 seconds) at a time. While Kaspersky (Kaspersky, 2013a, p. 59) have seen malware trying to connect to the C&C infrastructure every 150 seconds.

One APT uses KillTimer to destroy the specified timer (Microsoft, 2017u) and then SetTimer to create a timer with a specified time-out value (Microsoft, 2017ad).

Symantec (Symantec-Security-Response, 2015a, p. 19) discovered that one APT had two parametric delay methods: one was a delay loop which could be execute a number of times; the "sleeptime" which for a number of seconds which was at different points in the code.

Fireeye report the use of Sleep to evade file-based sandboxes as well as using it to invoke malware (Singh and Bu, 2014, pp. 12-13), and C&C communications failures (Nart Villeneuve *et al.*, 2014, p. 11). Fireeye also note the capture of user data every 500 milliseconds (Fireeye, 2014a, p. 37) as well as sleep for a random amount of time using rand() (Fireeye, 2015b, p. 22). Kaspersky have seen a sequence of useless Sleep API calls, probably to break detection of some signature-based AV engines (Kaspersky, 2013d).

One APT's malware checks for the existence of a special file. The malware will not start C&C server call back until the file is 180 days old - this ensures that a critical malicious component was removed during this period. The module backs up and restores system access within six months (Kaspersky, 2014a, p. 16). In another Kaspersky observation, after installation the main modules start working as "sleeper cells". They display no activity of their own but wait for 'wakeup' commands (Kaspersky, 2016a).

One APT has been seen using "hit and run" C&C tactics: The APT sets up a C&C, create a malware sample to use it, attacks and infects the victim, communicates with the victim machine and moves on. Shared hosting would expire in a month or two and the C&C disappears (Kaspersky, 2013a, p. 24). Kaspersky go on to say that the attacks were focused and that the APT knew what they were looking for; this author wonders if the reconnaissance had previously been performed? Kaspersky later note (Kaspersky, 2013a, p. 49) that the "hit and run" nature of this operation makes it unusual. In other cases, victims remain infected for months or years, and data is continuously exfiltrated. The APT appears to know what they need and after the information is obtained, the victim is abandoned.

Although this is a Windows-based thesis it is noted that one Unix based system opens the back door for three minutes every hour and then uses a unique combination of TCP headers on the correct port to enter (Chohan, DeSombre and Grosfelt, 2018, p. 5).

6.8.5 Use of Commercial Products

Fireeye report (Fireeye, 2015c, pp. 7-11. 13) an APT using Twitter to facilitate C&C: The APT generates a Twitter handle for a specific day ahead of that day. The malware on the victim's machine generates the same Twitter handle and that tells the malware to visit a specific Twitter handle a specific day. The APT posts obfuscated instructions before malware attempts to access it. Should contact not be made, the malware waits until the next day to try to again communicate. Communications attempts may be configured to occur on weekdays or after a specified date, allowing the malware to blend into "normal" network traffic. The tweet contains a URL, which has been observed as a GitHub or a compromised website URL, and a hashtag. The URL contains a varying size image or images. The hashtag contains decryption details for the instruction which can be found in the image(s).

Virus Bulletin report hijacking of internet satellite links for C2 by sniffing and spoofing (Virus-Bulletin, 2015, p. 7).

Mandiant (Mandiant, 2010, pp. 7, 8) observed that the malware for one attack did not listen for inbound connections (it only initiated outbound connections) so unless defenders were monitoring outbound traffic for beaconing it would not be discovered.

6.8.6 Protocols

C&C may be detected by: Consistent URL paths; Detectable packet headers; Identifiable network communications; Unusual ports and protocols. Secure sockets layer (SSL) certificates (TrendLabs, 2013b, pp. 3-4). Unusual port use has been seen elsewhere (Kharouni, 2015, p. 4) and commented on in Section 6.8.3.

A Component Object Module (COM) is a system for creating binary software components that can interact (Microsoft, 2018g) and which provides an interface to allow developers to control and manipulate objects of other applications (Rascagneres, 2014). A Class Identifier (CLSID) is a globally unique identifier to identify a COM class object (Microsoft, 2018f). APTs have been seen hijack or create CLSIDs (O'Brien, 2016, pp. 22, 25), (Mcafee-Labs, 2012, p. 2) as well as hijacking COM entries (Neville and Gibb, 2013, p. 18), (Wyke, 2012b, p. 8) and (Rascagneres, 2014).

More straightforwardly, Symantec report HTTPS for secure communications to the C&C server (Symantec-Security-Response, 2015b, p. 16). This APT even encrypts the IP address of the C&C server (Symantec-Security-Response, 2015b, p. 20) using a simple XOR with the byte 0x14 for all characters of the IP address. Sophos have seen an IP address XORed with 0x4E8F9AF4 and the port is XOR‘‘ed with 0xF5AD (Wyke, 2012b, p. 28).

McAfee report the use of an IRC channel to receive real-time commands (Sherstobitoff, Liba and Walter, 2013, p. 10).

6.8.7 Encryption

“With terms such as ‘packer’, ‘obfuscator’ and ‘crypter’, the line between these popular code protection mechanisms is becoming blurred, …” (Pontiroli and Martinez, 2015, p. 16).

Encryption and obfuscation can be rudimentary: Some APTs encipher their malware and/or retrieved data stored on disc. Different attacks from the same APT can use different encryption keys (Kaspersky, 2013b, p. 6). The table, below, summarises some straightforward encryption schemes:

Encryption Algorithm	**Key**	**White Paper Reference**
XOR Repeating	0x66	(Trend-Micro, 2013a, p. 13)
XOR Repeating	0x02	(Dela Paz, 2012, p. 6)
XOR Repeating	0x90 (with a 16-byte key also being used)	(Alintanahin, 2015, p. 3)
XOR Repeating	0x95	(Gross and Cylance-Spear-Team, 2016, p. 4)
XOR Repeating	unreported byte value	(Villeneuve and Sancho, 2011, p. 7)
XOR Repeating	“1/2”	(Pernet and Sela, 2015, p. 16)
XOR Repeating	0x3E	(Kaspersky, 2013c, p. 26)
Multiplication	One-byte key	(Alintanahin, 2015, p. 7)
Unknown	Machine specific variables e.g. MAC address	(Villeneuve and Bennett, 2012, p. 5).
Unknown Repeating	16-byte key	(CrowdStrike, 2014b, p. 35)

Double encryption Repeating	0x2C and 0x7B 0x70 and 0x79	(Fidelis-Cybersecurity, 2015a, pp. 5, 7).
XOR Repeating	4-byte key	(Wyke, 2011, p. 8) as have RSA (RSA, 2014, pp. 13, 25-27, 29)
XOR Repeating after bitwise NOT of the input stream.	4-long key’\x30\x30\x34\x31’ (said to be random)	(Haq *et al.*, 2014, p. 13).
XOR Repeating	32-byte key (the first 32 bytes) from one file for two other files.	(Symantec-Security-Response, 2015b, p. 13).

Table 6-2: Selected List of APT Developed Encryption Techniques

It can be seem that many APTs use very simple encryption with some being a simple substation. This knowledge for the defender allows the defender to counter simple encryption scheme using categorisation techniques. It will be shown later in this thesis that use of the Index of Coincidence and Vigenère Square analysis techniques are necessary and sufficient to identify such encryption. The deification and publication of these techniques should force APTs to change their encryption scheme and hence increase their business costs.

The thesis will now summarise less straightforward encryption schemes.

Although strictly encryption, using a Caesar (simple) substitution (single character encryption) data may be obfuscated using different Base64 alphabets (Sancho *et al.*, 2012, p. 3), (Mandiant, 2010, pp. 12, 14-16), (Fireeye, 2014a, pp. 31, 38), (Lee, Ahl and Hanzlik, 2014, p. 19), (Clearsky, 2016, pp. 5, 10, 16), (McAfee-Labs, 2015, p. 13) with part of the last example decoding to hex characters e.g. “0x55,0x8B,0xEC,0x81,” etc..

For one APT encryption scheme each byte is XOR-ed by a letter in the string, YHCRA, and rotated three bits to the right after every XOR operation (Villeneuve and dela Torre, 2013, p. 5). This encryption scheme is commented on in detail in Section 6.8.7.

Multiple key lengths (10, 6, and unknown – the latter is given as “keylen” in the paper) have been seen in one suite of attack software (Benchea *et al.*, 2015, pp. 20-21). None of the algorithms is a straightforward XOR of key and text. The pseudo-code for the 10-long key is incomplete so it is hard to see what might be the intent.

Encryption may be used with filename extension obfuscation For example saving scrapped data as a .dll made to look part of a wider .dll family (Yaneza, 2015b, p. 4). The encryption algorithm is not given. Sophos see filename extension renaming in a ransomware attack. This then stops other ransomware enciphering the same files (Loman, 2019, pp. 8-9).

GET and POST requests may contain obfuscated data, for example an obfuscated MAC address (Trend-Micro-Threat-Research-Team, 2012, p. 3) later in the attack the obfuscated MAC address is used as an RC4 encryption key to exchange date between the victim’s machine and the C&C server (Trend-Micro-Threat-Research-Team, 2012, p. 5).

Some APTs encrypt their malware, only decrypting at run time and being held in memory producing what is called fileless infection (described in Section 5.8). Others may decrypt to disk (Chen and Li, 2015, pp. 6,7) (O’Murchu and Gutierrez, 2015, p. 18). This thesis suggests that the true meaning of the word “fileless” in the APT context is that which has no image on disc but the definition has been expanded to include registry and reflective code injection. Symantec define fileless as:

“**Memory only threats**, such as SQL Slammer

- **Fileless persistence**, such as VBS in the registry
- **Dual-use tools**, such as psExec.exe, which are used by the attacker

- **Non-PE file attacks**, such as Office documents with macros or scripts”

(Wüest and Anand, 2017, p. 10)

Wüest and Anand go on to show a JScript inside malicious SCT file that contains a large amount of hex code (Wüest and Anand, 2017, p. 12). This will be discussed later in the thesis.

Trend Micro observe “http :// microsoft.com” being used as an innocuous as decryption key for RC4 (TrendLabs, 2013c, p. 13).

Another APT uses RC4 for encryption of elements of its attack. At least one element is stored in the registry as a wide character string and is converted to a multibyte character string before the key use. The conversion varies depending on the region of the user’s system. It also checks the system. O’Brien (O’Brien, 2016, p. 22) notes the use of RC4 encryption for communications with the C&C server. RC5 has also been seen (Symantec-Security-Response, 2015c, p. 14) as well as RC5 and RC6 (Kaspersky, 2015c, pp. 27-30) and (CrowdStrike, 2014b, p. 35) also use a relatively simple XOR based method of encryption with a 16 byte long key.

AES128 has also been seen (Katsuki, 2012, p. 9) as has RSA and functions imported from the Microsoft Cryptography API library (Sherstobitoff, Liba and Walter, 2013, pp. 11, 16). Sophos have also seen the use of RSA (Wyke, 2012b, p. 14).

Symantec report seeing the use of a Linear Congruential Generator (LCG) for encryption (Symantec-Security-Response, 2016, p. 10) and 128-bit CAST in CBC mode (Symantec-Security-Response, 2016, p. 13). While Sophos have seen malware encryption based on Microsoft’s Visual C++ LCG (Wyke, 2014, p. 4). An encryption scheme that may be a poorly implemented (deliberately or otherwise) LCG is discussed later in the thesis.

Kaspersky point out that “Changing the code breaks Yara recognition and other signature-based detections.” (Kaspersky, 2017b, p. 6).

F-Secure (F-Secure-Labs, 2014d, p. 5) have seen a decryption scheme decoding encoded strings based on a shuffled jump table. Tantalisingly no further information is provided. F-Secure further noted (F-Secure-Labs, 2014d, p. 10) nidsSendNetBufferLists as part of encrypted TCP data.

Privacy is becoming more important and users are moving to secure communication partly through legal pressures (Blue-Coat, 2013) but partly through personal choice. Encrypted traffic allows malware to bypass perimeter checks. In 2014 Blue Coat (Blue-Coat, 2014, p. 2) observe that SSL-encrypted traffic, which makes up 30%-40% of all Internet traffic, is used by APTs and cannot be deciphered by IPS and web gateway defences.

6.8.8 Steganography, Obfuscation and Polymorphism

Steganography, hiding messages in text or pictures, can play a part (TrendLabs, 2015a, pp. 20, 23) and also set up a second C&C channel to counter incident response. Fireeye also report the use of steganography (Fireeye, 2015c, p. 11), (Fireeye, 2014c, p. 7) as does (Mosuela, 2016), however an installer file is needed.

Obfuscation may be layered and not just once instance. Unique counterfeit certificates may be registered under legitimate corporations, without the knowledge of those corporations. The uniqueness may be independently tested (Barysevich, 2018, pp. 1-2).

Obfuscation has been seen during the reconnaissance and delivery part of the Kill Chain by such as Virtual Private Networks (VPN), Virtual Private Servers (VPS), Transport Layer Security (TLS), and The Onion Router (Tor) (Moriuchi, 2018, p. 2).

Symantec have observed APTs obfuscating a domain name e.g. "*gstr_domain4 = "ns" & "1" & ".p" & "lay" & "e" & "r13" & "52.org""* (Suenaga, 2012, pp. 28, 46) with further examples at (Symantec, 2016a, pp. 12, 15-18) and JavaScript (Selvaraj and Gutierrez, 2010, pp. 11-13). While McAfee have seen code setting conditions which can never be met e.g. "if [number] = [number] +

1 Then End” (McAfee-Labs, 2015, p. 40). As previously discussed, McAfee have seen obfuscated JavaScript code delivered by email in a .zip file (McAfee, 2018a).

Jain et al assert that obfuscation may be spotted by using four metrics:”

- N-gram checks for the probability of occurrence of certain sequence based upon the good and the bad sample set;
- Entropy checks for the distribution of the used bytes codes;
- Word Size checks if very long strings are used;
- Size checks for the decompiled class file.”

(Jain, Gomez and Singh, 2014, p. 30)

This thesis notes the existence of The International Obfuscated C Code Contest exists. There are five goals of the contest, the first of which is to write Obscure/Obfuscated C code (Broukhis, Cooper and Noll, 2019).

Goncharov describes polymorphism as:

“Polymorphic crypters are considered more advanced. They use state-of-the-art algorithms that utilize random variables, data, keys, decoders, and so on. As such, one input source file never produces an output file that is identical to the output of another source file. This can be achieved by using several algorithms, including:

- Shuffling blocks of code while preserving a malicious file’s ability to run: Blocks of code are encrypted using a specific technique. Several decoders are then created for the malware body, which is randomly decoded. This applies also to variables and other data.
- Creating macros: A macro is created during preprocessing. When invoked, it repeatedly performs an instruction.
- Inserting garbage code: Blocks are split into sections, in-between which garbage instructions are inserted. These instructions do not

affect the code but force an emulator to "sweat." Not only are garbage instructions used in code blocks, these are also used to execute helpful actions that complicate the work of an anti-malware analyzer in every possible way.

- Combining all of the above-mentioned methods: All of the aforementioned methods, along with dynamically generating algorithms after encrypting a specific block of code based on random conditions, may also be used."

(Goncharov, 2012, pp. 1-2)

and the data is encrypted and decrypted offline by the attackers using keys unique to each client. (Symantec-Security-Response, 2012a, p. 3).

Sophos state that:

"In a polymorphic attack, code is typically encrypted to appear meaningless and paired with a decryptor that translates it back into a form that can be executed. Each time it's decrypted, a mutation engine changes its syntax, semantics, or both." They go on to state that "Traditional polymorphic viruses are self-contained and must contain the mutation engine in order to replicate."

(Sophos, 2013, p. 31)

6.9 Actions on Objectives

6.9.1 Definition

Lockheed Martin state that the objective is typically data exfiltration involving collecting, encrypting and extracting information from the victim's machines. They also state that potential objectives are violations of data integrity or availability. The APT may only access the victim's machine to use as a hop point to compromise other systems and move laterally inside the network.

6.9.2 Types of Communications

Actions on Objectives are as varied as motivations and are the logical extension on motives. For example one APT disabled direct database manipulations (Kaspersky, 2017b, p. 5).

Fireeye report (Fireeye, 2015c, p. 12) the use of cloud storage to hold exfiltrated data prior to a final downloaded at the APT's convenience. Fireeye also noted (Carr, 2017) an APT tracking their deployments with the use of cloud-based email analytics which had been designed for sales organisations on legitimate cloud storage services as well as the use of native webpage functionality linked to images hosted on infrastructure monitored by the APT.

Exfiltration of data is by many means: from the simple, unencrypted method (Trend-Micro, 2013a, p. 9), (Trend-Micro-Cyber-Safety-Solutions-Team, 2016, pp. 4-5) to use of TCP, UDP, ICMP, HTTP etc (Symantec-Security-Response, 2015c, pp. 11, 15) (Neville and Gibb, 2013, p. 8), (F-Secure-Labs, 2014a, p. 12).

The original intent of DNS tunnelling was to allow machines to resolve a DNS to an IP address. As it is critical to a network's operations ports, typically port 53, are left open. This allows APTs to use a, generally, unscrutinised port (Insikt-Group®, 2020a, p. 19).

6.9.3 PowerShell

PowerShell allows "… commands that enabled it to download files and bypass execution policies to execute the files." (TrendLabs, 2014, p. 4) with a "typical command to download" demonstrated by Symantec (Symantec, 2016a, p. 9). Symantec go on to list the 10 reasons why attackers use PowerShell:

- It is installed by default on all new Windows computers;
- It can execute payloads directly from memory, making it stealthy;
- It generates few traces by default, making it difficult to find under forensic analysis;
- It has remote access capabilities by default with encrypted traffic;

- As a script, it is easy to obfuscate and difficult to detect with traditional security tools;
- Defenders often overlook it when hardening their systems;
- It can bypass application-whitelisting tools depending on the configuration;
- Many gateway sandboxes do not handle script-based malware well;
- It has a growing community with readily available scripts;
- Many system administrators use and trust the framework, allowing PowerShell malware to blend in with regular administration work.

Of 49,127 PowerShell scripts submitted for malware analysis, Symantec found that 95.4% were malicious (Symantec, 2016a, p. 8) . 55% of scripts were started through cmd.exe on the command line (the next highest was msiexec.exe at 8%) with 95% of scripts executed through cmd.exe (the next highest being wmiprvse.exe with 9%) (Symantec, 2016a, p. 12).

Symantec have observed one APT downloading updates and tools, including powershell, "The files are download and installed with the "/quiet" and "/norestart" flags to keep them hidden from the user." (O'Murchu and Gutierrez, 2015, p. 7).

To restrict powershell use the registry key HKEY_LOCAL_ MACHINE\SOFTWARE\Microsoft\ PowerShell\ should be set to "Restricted" but this may be bypassed by use of "Bypass" (Kazanciyan and Hastings, 2014). This will ensure that "nothing is blocked and there are no warnings or prompts" (Microsoft, 2020). Fireeye have observed an APT using powershell "powershell -ExecutionPolicy bypass -WindowStyle hidden – encoded Command..." (Fireeye, 2015c, p. 12) which allows an APT to bypass without changing the registry key.

Sophos have observed one piece of malware using PowerShell in conjunction with a Registry Run key. The malware first checks to see if PowerShell is running; if not, it downloads and install it and then creates a blank or NULL Autostart entry using the ZWSetValueKey (Santos, 2014). Sophos subsequently saw more use of the registry for storing payloads but with registration in one registry key and the payload (dropped) into another (Szappanos, 2015b, pp. 15-16, 19-20).

Although Powershell offers the ability to be fileless , it does need Registry Keys for persistence (one of which is HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\(null). It could be argued that this is not truly fileless as the "file" is stored in the registry key (Rivera and Inocencio, 2016, pp. 71-74), (Rivera and Inocencio, 2015).

"After luring unsuspecting users into opening what seemed to be an innocent *Excel* spreadsheet, a password-protected macro would then decode and execute the final PowerShell payload, effectively bypassing the execution policy in place." (Pontiroli and Martinez, 2015, p. 13).

"Once again, we are shown how easy is to bypass a PowerShell execution policy. In this case, merely invoking the script with the parameter 'bypass' will do the trick." (Pontiroli and Martinez, 2015, p. 15).

It is suggested (Pontiroli and Martinez, 2015) that it is possible to aggregate Powershell into a framework. One such framework, PowersSploit, can integrate a set of PowerShell scripts and modules to be used in the attack post exploitation phases. PowerSploit executes scripts to perform administrative and low-level tasks without the need to implant malicious executables with the aim to evade timely AV detection. This is another example of Living Off the Land". Panda Security observes (Panda-Security, 2015, p. 6) the same phenomenon without PowerShell as the malware's peculiarities indicate that the proactive protection layers included in most AV solutions would not be able to detect its apparently harmless behaviour.

Mandiant have also see PowerShell mailbox harvest using EWS (Mandiant, 2017, p. 19).

The .NET Framework is a development platform for app building for various platforms. It has a common language runtime (CLR) and .NET Framework class library with broad functionality and industry standards support. It provides services which include memory management, type and memory safety, security, networking, and application deployment. It also has data structures and APIs that abstract the lower-level Windows operating system. It can be used with different programming languages (Microsoft, 2019e) and may be used maliciously with PowerShell (Pontiroli and Martinez, 2015).

6.9.4 CPU Memory (RAM Scrapper)

Trend Micro have seen credit card details scrapped from Point of Sale (PoS) RAM (Trend-Micro-Cyber-Safety-Solutions-Team, 2016, pp. 3-5). The malware looks for international credit cards that are unrestricted for authorization not requiring a pin. In a report summarising such attacks Trend Micro assert that weak PoS security allows APTs to breach networks but not necessarily for targeted attacks. The malware does provide instant, huge, profits (Trend-Micro, 2015, pp. 27-29).

6.9.5 DDOS

Distributed Denial of Service (DDOS) attacks use multiple machines to attack one victim. One DDOS attack has been seen using connectionless LDAP (CLDAP). CLDAP is typically used on networks for accessing active directory services such as usernames. This DDOS attack replaces the attackers IP address with their victim's IP address ad hence send the response to the victim (Insikt-Group®, 2020a, pp. 20-21).

6.9.6 RansomWare

Some APTs hold victims' computers to ransom by encrypting files. For example one APT encrypts random position of files, deletes Volume Shadow Copies, disables recovery reboot and then reboots the machine after 300 seconds (Solad *et al.*, 2018, pp. 9, 13).

6.10 Malware Languages

Web Shells have been seen written in JSP, CFM, ASP, ASPX, or PHP (RSA, 2014, p. 9). The advantage of Web shells is that they have: low detection rates; the inability to block or monitor an IP and no beaconing.

A means of detecting file type is needed.

6.11 APT Errors

6.11.1 Introduction

APTs are made up of computer architects and software engineers who too make mistakes – they are not super-human. It is observed that it may be easy to decompile their executables or enter their C&C infrastructure as they too make similar mistakes to their targets (Sancho and Hacquebord, 2016, pp. 4-5), (Villeneuve, 2011, p. 5). Villeneuve later goes on to point out:

> "The threat actors behind targeted malware attacks do not always use zero day vulnerabilities—exploits for vulnerabilities for which there is no patch available. While some might believe that the threat actors behind targeted malware attacks have mythical capabilities, both in terms of their operational security and the exploits and malware tools used, they, in fact, often use older exploits and simple malware. The objective of these attacks is to obtain sensitive data; the malware used in the attacks is just an instrument. They will use whatever is required to gain entry based on reconnaissance. In addition, they will adjust their tactics in reaction to the defenses of the victim."

(Villeneuve, 2011, p. 8).

It is known that some malware is used across campaigns (Trend-Micro.Forward-Looking-Threat-Research-Team, 2012, p. 2). This allows defenders to spot similarities and if necessary, disrupt more than one infection should that have occurred. However one should be cogniscent of false flags (Bartholomew and Guerrero-Saade, 2016), (Guerrero-Saade, Moriuchi and Lesnewich, 2018, p. 3) which may fool a defender and take up a lot of time and effort.

Berghel et al suggest that “Typically, forensic analysis of systems reveals that bad actors do not often take any extraordinary means to hide data beyond, perhaps, either wiping the media or encrypting the data.” (Berghel, Hoelzer and Sthultz, 2006). This has been explored earlier in the section on encryption where cycling single or multiple bytes are used as the encryption or, at the other end of the scale, for example, AES. Software to flag possibly encrypted files has been developed for, and will be discussed later in, this thesis.

6.11.2 Domain Registration and Email Addresses

Fireeye have observed the use of the same email address to register domain names used in attacks (Nart Villeneuve *et al.*, 2014, pp. 13-14, 17). While elsewhere two different emails addresses have been used to separately register a C2 domain and a another “malicious domain”, another to register 125 IP addresses (Guerrero-Saade and Chohan, 2018, pp. 3,11, 23, 25-26).

It is possible to check historical ownership of internet domains as well as for example registrant names (Kruse, Hacquebord and McArdle, 2012, pp. 15-16) however fake or comprised email addresses may be used as well as non-standard telephone and fax numbers (ICANN, 2017).

Agari note the use of linked Gmail accounts (Agari, 2019, p. 14). Any Gmail address with the same name but a dot in a different place directs to the same Gmail inbox. For example, my.name@gmail.com , m.yname@gmail.com and myn.ame@gmail.com direct to the same inbox. Agari go on to note that the vast majority of the internet treats these three addresses as three separate accounts. This makes the life of APTs easier in that they can control many victims from the same inbox - it also makes a defender’s life easier as linked Gmail addresses are easier to identify.

6.11.3 Typographical Errors (Typos)

It is asserted (Pernet and Sela, 2015, p. 13) that APTs do not seem to put much effort into content QA - emails and phishing pages formulation – hence making a lot of typos and grammatical errors. This is supported (Clearsky, 2015) by the observation of another APT, not new to hacking, displaying grammatical errors, exposure of attack infrastructure, anti-analysis techniques that were easy to bypass, no code obfuscation “and more”.

6.11.4 Compiler Traces and Date Related Issues

“A program database (PDB) file holds debugging and project state information that allows incremental linking of a Debug configuration of your program. A PDB file is created when you build with /ZI or /Zi (for C/C++).” (Microsoft, 2017z)

Some APTs leave traces of their development work within their code. For example leaving the PDB path in the malware (Villeneuve and Sancho, 2011, p. 10), (TrendLabs, 2014, p. 13), (TrendLabs, 2013c, p. 14), (Fireeye, 2014b, p. 5), (Nart Villeneuve *et al.*, 2014, p. 10), (F-Secure-Labs, 2015a, p. 7), (Cylance, 2016, pp. 8, 26, 28-29) or debug strings (Pernet and Sela, 2015, pp. 15, 16), (Pernet and Lu, 2015, p. 10) both of which use the same malware.

Timestamps of when code was compiled may be analysed (Pernet and Lu, 2015, p. 4) as there are very few different samples in the wild and close timestamps may indicate small changes in APT code (Florio and Kasslin, 2009, pp. 1-2). Compile times may also be used to as supporting evidence towards identifying the origin of APTs (Fireeye, 2014a, p. 27), (Fireeye, 2015c, p. 5) although it would not be difficult to spoof this by changing the date and time of the clock where the code was compiled. The power of timestamp analysis is highlighted by Kaspersky (Kaspersky, 2014b, pp. 35, 37) who were able to analyse the compile times of 154 binaries from one APT. Elsewhere Trend Micro noted (Trend-Micro-Incorporated, 2015, p. 14) that two related APTs had similar behaviours which may or may have been a coincidence, as binaries significantly differed.

Analysis of compilation time of binaries is taken up by Fireeye (Fireeye, 2014f, p. 13) who point out that although the compilation time of binaries can be easily forged, analysis of them is still useful. The timestamp may not reveal when a binary was compiled, but it can be used to cluster samples by compile times.

APTs have been seen to leave the name or nickname of developers in compiled code (Symantec-Security-Response, 2012a, pp. 8-9), (Katsuki, 2012, p. 12).

Fireeye have observed one APT changing the modification date of files. This date is earlier than the creation date (Lee, Ahl and Hanzlik, 2014, p. 9). The unreliability of timestamps on the Windows OS is taken up in Section 6.11.4.

Kaspersky (Kaspersky, 2017a, p. 15) have seen one piece of malware with a fake PE file timestamp; however, the authors forgot to change a timestamp inside the debug directory which pointed to: 2016.11.14 21:16:45. A fake timestamp has also been seen by F-Secure (F-Secure-Labs, 2014d, p. 9).

Elsewhere metadata leaks from ”Creation Date”, “Modification Date”, “Author and Last Modified By” have been seen (Pernet and Lu, 2015, p. 9).

6.11.5 Natural Language Remnants

Fireeye have seen natural language as a passphrase and salt (Nart Villeneuve *et al.*, 2014, p. 8) as well as RC4 with a salt (Fireeye, 2014a, p. 35). Elsewhere cultural references have been seen (Kaspersky, 2014e, pp. 13, 16), (Symantec-Security-Response, 2015c, p. 13).

6.11.6 Use of Malware Scanners

Online malware scanners may be used to check to see if AV software will identify APT malware as malware (Pernet and Lu, 2015, p. 4) (Goncharov, 2012, p. 13), (Goncharov, 2014, p. 14), (Zetter, 2014). Zetter states that one can follow the development of malware but Goncharov in the later reference states that some attackers use web sites not connected with AV companies. Clearly, AV companies can build an advantage because as soon as, or even before, malware is deployed their products will have been updated. One may consider it imprudent to publicise these activities in this thesis but, once again, the idea is to increase the cost of business of the attackers. Even if attackers were to stop using these online malware testers, they would have to build their own, hence increasing the cost of their business.

6.11.7 Encryption

"If you really want something secure, this isn't the way. You want to lock something down, you do it with multiple keys, layers of security, not this peekaboo. This is … it's what you call lingerie encryption. Looks sexy, doesn't conceal anything at all." (Harkaway, 2017).

Shannon asserts that:

- The amount of secrecy should be proportionate to the effort put in to securing the message;
- The key size should be as small as possible;
- Complexity of enciphering and deciphering should minimised;
- Error propagation should be minimised;
- The encrypted message should be no longer than the message.

(Shannon, 1949)

Using simple examples where the key is a single byte: 0x66 in binary is 01100110, 0x02 is 00000010, and 0x09 is 00001001. Observe that the first encryption key will change half of the number of bits in the byte whereas the second will encrypt one bit and the third two-bits. The second and third encryption keys means that the first nibble of every byte is unencrypted.

Although one APT uses 0x2C and 0x7B as a double encryption, what they do not seem to have realised is this is equivalent a single encryption with 0x57 (0x2C XOR 0x7B = 01111011). Elsewhere in the attack they do the same with 0x70 and 0x79 which is equivalent 0x09 (0x70 XOR 0x79 = 00001001).

As has been demonstrated many APTs a single repeating byte to encrypt malware. Such encryption is easily defeated: for example, a number of fields in a PE or .dll are reserved and always set to zero. When zero, in any position in a file, is enciphered with any key character, binary mod 2, the result is always the key character. One can just look at all of these positions and check to see that they are all the same (and not zero) to recover the key character. This key can also be checked against encryption in the position in the file where “MZ” is expected (Pietrek, 2002); if , after decryption, the characters “MZ” are seen then the key is valid and the whole file may be decrypted. In addition, any bit that is enciphered by a zero will result in the original bit being the cipher. Interestingly:

> “the final payload … was delivered encoded with a single-byte XOR against the byte 0x95, skipping both the key itself and zero in an attempt to avoid exposing the key. This method of obfuscation at the time would have ensured delivery of the payload past most IDS/IPS systems. ”
>
> (Gross and Cylance-Spear-Team, 2016, p. 4)

Using a repeating key is for encipherment is, in effect, the same as using a Vigenère cipher (Gaines, 1956) and this too can be solved. However, this author asserts that APTs do not necessarily use encryption to secure their code but to obfuscate it. Minimizing CPU time means using a simple form of encryption which means a quick method of decryption.

One APT has been seen using an encryption scheme where each byte is XOR-ed by every letter in the string, YHCRA, and rotated three bits to the right after every XOR operation. However, this is equivalent to encrypting using the single byte 11110110 as shown in the table below:

Initial Text	(Null) 00000000	P (01010000)
XOR Y (01011001)	01011001	00001001
Circular Right Shift 3	00101011	00100001
XOR H (01001000)	01100011	01101001
Circular Right Shift 3	01101100	00101101
XOR C (01000011)	00101111	01101110
Circular Right Shift 3	11100101	11001101

XOR R (01010010)	10110111	10011111
Circular Right Shift 3	11110110	11110011
XOR A (01000001)	10110111	01000001
Circular Right Shift 3	11110110	01010110

Table 6-3: Analysis of "YHCRA" Encryption Scheme, Method 1

It can be seen that 11110110 XOR P (01010000) equals 01010110 which is the final value of the rightmost column.

Another way to view this is as set of streams of bits (circular left shift five equals circular right shift three) and to XOR (count the parity mod 2) of the last eight columns of each row to give the key:

Y	010110010101100101011001010110010
H	0100000101001000010010000100
C	01000011010000110100001
R	010100100101001001
A	0100000101000
Key	11110110

Table 6-4: Analysis of "YHCRA" Encryption Scheme, Method 2

The point here is to demonstrate that one must be careful with encryption as "more complicated" or "more complex" does not necessarily mean better, i.e. stronger, encryption. This example re-enforces Shannon's assertions above about minimising complexity: the designer of the above scheme has picked a bit shift of three which is co-prime to eight but has five times the number of XORs needed plus five circular shifts for each iteration i.e. a workload of at least five times for each encipherment when all that is needed is to store 11110110 as the single byte key. Depending on the hardware and software implementation the cost could be as much as 10 times (10 clock cycles) as much as using one byte.

Jscript inside a malicious SCT file has been discussed (Wüest and Anand, 2017, p. 12). This is, in effect, a simple form of obfuscation, it is also encoding but it does not meet one of Shannon's Principles in that the encoded stream should not be longer than the original q.v. the code "\x52\x32\x56\x30" etc. It is noted that the characters "\x" repeat every one characters. A theme which will be developed. Malwarebytes (Malwarebytes, 2018a, p. 8) also observe script obfuscated using hex code, for example "var _0x8aa6=["\x75\x73\x65 …."

One APT has a 3-stage encryption: a rolling XOR encryption, followed by RC4 encryption of this stream, followed by Salsa20 encryption of the intermediate stream (Insikt-Group and Rapid7, 2019, pp. 16-18). Stage 1 (the rolling XOR encryption) is a Linear Congruential Generator (LCG) with seeds of four and eight and a divisor of 255. Analysis for this thesis shows that this stream repeats after 360 iterations (Appendix G). It is possible that the developers are aware of the short comings of RC4 and thought that three different encryption stages are better than one. However, the use of indicates, or a desire to indicate, lack of cryptographic skill and knowledge.

6.11.8 Certificate Re-Use

Certificates may also be used to link attacks: Fireeye have observed six different certificates which linked a number of APTs (Fireeye, 2014f, p. 6), while Recorded Future mention the work of a rival AV company who have seen an SSL certificates shared across command and control (C2) domains (Gundert, Chohan and Lesnewich, 2018, p. 8) with later research elsewhere connecting the certificates to three other domains. WHOIS information then connected this information to a specific organisation.

6.11.9 Miscellaneous

One keylogger was so badly developed that it leaked the attackers FTP credentials (Pernet and Sela, 2015, p. 15).

Fireeye assert (Fireeye, 2014b, pp. 3-11) that, although not fool proof, a number of things can be used to ascertain the origin of APTs: Keyboard Layout; Malware Metadata; Embedded Fonts; DNS Registration; Language; Remote Administration Tool Configuration; Behaviour. Fireeye have also observed one APT using English language Word documents being used with a foreign character set and the same DNS resolution (Fireeye, 2014d, pp. 18, 20).

Sophos (Wyke, 2014, p. 9) have seen one piece of malware with debugging left on. During normal execution victims will not see the debugging but if a particular registry key is created then a message box is displayed.

One APT relied on a second Lua-script (Pontifícia-Universidade-Católica-do-Rio-de-Janeiro, 2017) for automatic wiping of stolen documents but this was not always done leaving the files in the APTs cache forever (Kaspersky, 2016a, p. 15).

Finally in an attack using decoy droppers of order and receipt (F-secure-Labs, 2014b, p. 5) an image of a receipt taken with a mobile phone was sent and this image included the metadata including the date the photograph was taken and model of the phone.

6.12 Conclusion

This chapter is aligned with the thesis' first two Aims and Objectives and has reviewed AV companies' whitepapers and other supporting sources to present a view of how real-world APT examples get malware onto victims' machines. These sources were aligned with the LMKC in sub-section 5.3 to 5.9 (inclusive). The chapter has also highlighted some APT errors.

This thesis concludes that malware may:

- enter the system from malicious websites, USB devices, business documents or importing legitimate software;

- enter the system using default passwords on legitimate software;
- hide in plain sight (i.e. the files may be known to the operating system or they be largely invisible to the operation intercepting communications between application and the operating system);
- be standalone files or infect another file;
- need a method of C&C and hence website with which to connect and ports numbers to use;
- have the same name as a legitimate file;
- be encrypted or otherwise "hidden";
- be signed by legitimate software;
- be of small size.

It is suggested that:

- a means of analysing file length is needed;
- a means of detecting file type is needed;
- date analysis of the Windows files is needed;
- certificate may be used to link attacks;
- malware may be insidious and seductive e.g. malvertising.

7 THE FOUNDATION FOR THE TREATMENT OF APTS

7.1 Chapter Overview

This chapter presents the foundation for the solution – how can data be categorised for analysis? It supports the third Aim and Objective. The thesis has discussed APTs and where the malware they deploy lie on the HDD. This chapter will discuss how an APT team may be organised for their business and discuss increasing their business costs. It will distil the information into a manageable form and present a structure for the software suite developed to treat malware on the HDD. As previously stated, this thesis is agnostic to the origin and intent of the APTs and malware. In some cases, it is not necessary to look for malware to treat it; in other cases, putative malware should first be identified. The chapter will also contain discussion of what pristine elements of the Windows OS should look like.

It has previously been stated the Windows Operating System is a file-based system. The analysis and subsequent programming will make use of this design philosophy. It will also make use of the Mechanisms of Action for malware, Attributes by Design and Attributes by Discovery.

7.2 Setting the Scene

Auty claims that the game is a little bit stacked in the attacker's favour for a few reasons:

- The attackers have unlimited time.
- The attackers have unlimited resources.
- There is little recourse that can be taken across multiple international borders.
- An organisation needs to focus on executing its business strategy, not solely pouring resources into defensive capability.

(Auty, 2015)

For the first two points this thesis asserts that no-one has unlimited time or resources. With the third point this thesis has asserted that potential victims could insert malware into their files to infect the APTs' machine. The fourth point is valid and has been discussed earlier; if the cost of security is greater than the cost of the data being protected then the defensive capability is unbalanced. Furthermore, the last two points, above, are valid insofar as it has been seen that APTs take advantage of weaknesses in business process (and technical inconsistencies) to exploit IT security. This thesis further asserts that part of this problem is that manual business processes that are internal to the organisation contain checks and balances which have been transferred to IT rather than thinking about how the IT can facilitate the business and build new, relevant checks and balances into the new technologically based business processes. In other words, users have a legal and moral duty to protect information on their machines.

There is an opportunity cost associated with the disruption caused by APTs so defenders should wish to minimise the amount of work, business disruption and overhead, to be done. Ideally, at the elemental level, making the disruption of an APT look like a Single Event Upset (Normand, 1996). However too many such events begin to look suspicious. Villeneuve points out that malware may exploit specific software on the victims' machine and can be modified so that it is not detected by the victims' security. One can turn Villeneuve's suggestion (Villeneuve, 2011, p. 14) on its head so that the victim becomes the attacker and the APT becomes the victim - one can ensure that the malware they send to the APT exploits software on the APT's computer and they can modify the machine so that it is not detected by the security solutions. This is consistent with Detect, Deny, Disrupt, Degrade, Deceive, Destroy from the LMKC Courses of Action Matrix. Such a view makes the general-purpose solution all the more valid.

Finally, this thesis will assert that AV solutions are too obvious and the defenders should take a leaf out of the attacker's book and camouflage or obfuscate the AV software. All too often AV software may be found in a directory named after the AV company with associated registry keys e.g. "C:\AV Company Name\Directory 1\AV.exe". The AV software should be in random directories using random registry keys, innocuously (and named differently) Windows Services with fake, plausible descriptions and, where necessary, enciphered. Ideally, as far as possible, the defender needs to “Live off the Land”.

7.3 Business Software Investment

Business software investment may be either purchasing new, off-the-shelf software, or development by the organisation (IBISWorld, 2021). There is no mention of freeware which some APTs make use of. IBISWorld state that compound growth rate of business software investment for the UK for 2016-2021 was 2.4% and that for 2021-2026 it is estimated to be 2.82%. They acknowledge that this is an estimate as many services and software are priced in the local currency and are subject to currency fluctuations.

However these percentage figures are little more than the Bank of England’s inflation target rate of 2% (Bank-of-England, 2021, p. 13). Therefore APTs’ business costs have to be increased by more than this or the inflation rate for the jurisdiction in which they reside.

Although this is for the UK it provides a benchmark for the world. Any attempt to increase the APT business costs must be at least this amount and preferably much more.

Pressman (Pressman, 2010, pp. 67-68) asserts that software changes are easier to make when gathering requirements and the costs escalate quickly much later in the project. By extension and APT which has deployed malware and then has it treated by the software in this thesis will have their business costs increased.

7.4 The APT Administrative Business Model

A view of the malware business model can now be built and this will be used to demonstrate how the business costs of APTs may be increased (IBISWorld, 2021)

At one end of the scale is the “lone wolf” model: someone working alone, perhaps building malware from ideas on the internet. Another is the well-financed and organised group. With knowledge inferred from the gaps it is suggested that an organisational model would need reconnaissance of victims with any attack being based on the lowest perceived technical level needed for a successful attack. Business structure may be Reconnaissance, Plan, Development, Build, Test and deploy teams with tasks being built around division of effort. There seems to be gaps in QA of software and attack management. Management of attacks perhaps consists of each attack being given a different number for each part of the attack and unique victim identifiers. Acquisition of hardware and software including compilers. Support Facilities (HR: Recruitment, maintenance, reward, exit); Financial Facilities. Financial Accounting, Management Accounting. There would also need to be a Research Facility. A possible APT organisational chart is given the next page.

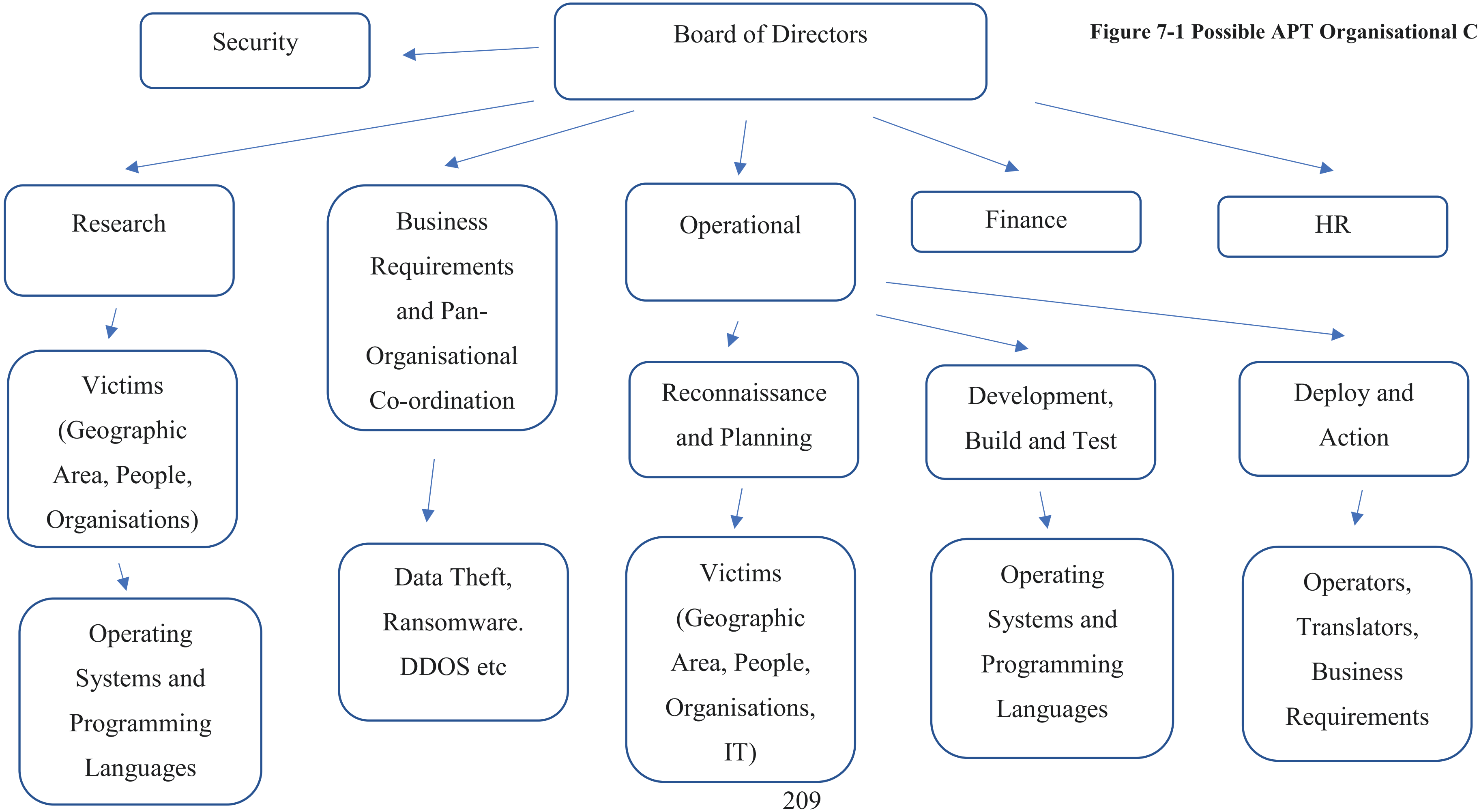

Security
Board of Directors
Research
Business Requirements and Pan-Organisational Co-ordination
Operational
Finance
HR
Victims (Geographic Area, People, Organisations)
Reconnaissance and Planning
Development, Build and Test
Deploy and Action
Operating Systems and Programming Languages
Data Theft, Ransomware. DDOS etc
Victims (Geographic Area, People, Organisations, IT)
Operating Systems and Programming Languages
Operators, Translators, Business Requirements

Figure 7-1 Possible APT Organisational Chart

7.5 Increasing the APT Costs

To re-iterate: this thesis does not claim that the work will lead to 100% identification and eradication of malware from the system. The thesis will claim to help:

- preserve the O/S legitimacy, integrity and consistency;
- increase the full life-cycle business costs (time, money, electricity, hardware and software) of malware writers and users.

This thesis uses the Lockheed Martin Kill Chain (LMKC) to treat malware (we deliberately do not use the word disrupt as this has a specific meaning in the LMKC academic paper).

Earlier the LMKC Courses of Action Matrix was been extended from six to seven courses: Deter, Detect, Deny, Disrupt, Degrade, Deceive, Destroy. Overall, the idea is to deter attackers but in order to do this it may be needed to employ a combination of the other six Courses of Action.

Simpler countermeasures which are used to increase the costs of the attacker should price out of the market the attacker with fewer resources i.e. the single person attacker who only has a PC and their brain. In reducing the number of potential attackers, the field can be reduced to more well-resourced attackers who may be more of a long-term threat to business and the intellectual property that needs protecting. These are the real Advanced, Persistent, Threats.

However, a defender does not want to increase the cost of defence to a point where these costs are disproportionate to the business. For example, if the cost of defence is greater than the cost of the Intellectual Property and its end use then the defender’s commercial value is destroyed. This thesis suggests that there are two measures of gaining an advantage: Absolute Cost and Relative Cost. For purpose of this thesis it is assumed that both defenders’ and attackers’ costs are full lifecycle costs.

Absolute cost is the absolute difference between the attacker's costs (£A) and the defender's costs (£D). For example, if the costs are respectively £A and £D then the difference is £(A-D). Relative cost is £(A/D).

Two thresholds (£T_1, £$T_{2)}$ are chosen which must be passed for a defender's work to be considered a success. i.e. £(A-D) > £T_1 and £(A/D) > £T_2. Additionally. the defender's business may only allocate £C to the defence of their assets so the defender must also satisfy £D < £C. Both of these values (£D, £C) may be related to the value the defender places on the data or information they wish to protect. (£V). This thesis defines this data as the crown jewels, loss of which may put the defender's business out of business. This might, therefore, be something like:

- £(A-D) > £1,000,000;
- £(A/D) > £10V;
- £D < £C.

For the attacker, measures will include return rates of a particular type of attack, number of LMKC Reconnaissance-stage probes, cost of maintaining and servicing a presence on the defender's machines. This thesis asserts that relatively inexpensive mitigations can substantially increase the attacker's costs.

This thesis is concerned with malware that has been placed on the system. This means that the attacker has achieved the first five of the seven stages (Reconnaissance, Weaponization, Delivery, Exploitation and Installation) of the LMKC. During these stages traces of the attack may have been left on logs or the wider system and other methods of analysis (e.g. log file analysis) may help find these traces.

We are, therefore, left with the final two stages: C2C and Actions on Objectives. This can be distilled down to looking for external communications with the attacker and looking for persistence.

The LMKC stage C2C, by definition, needs to communicate with the malware administrators. Unless the C2C part needs to jump airgaps and it is not an insider attack using, for example, open USB ports or CD/DVD drives, then the attacker needs to communicate over the internet. This means connecting to other IP addresses.

All of the software was developed on the author's development machine, first using Visual Studio 2013, and then using Visual Studio 2019, except for the device driver which was completed on a University of Gloucestershire machine.

The goal of the work contributing to this thesis is never to have claimed that it will eradicate 100% of all malware. The goal has been to increase the cost of business to APTs. i.e. cost in people, time, hardware, new software, electricity.

To what level is it desirable to increase the APT's cost of business? It has been demonstrated early that it is difficult to cost software projects.

The increase in APT's cost of business may be defined as any costs that they incur over and above those Anything the defender can do to increase the costs of their business model through the non-standard operation of their operations management, development and data analysis. Ideally the costs should be high enough for them not to deploy, and where deployed, to terminate, their operation. This is a theme supported by O'Connor (O'Connor, 2014, p. 25) and (Molinyawe, Hariri and Spelman, 2016, p. 5). Although a defender's IT security costs are sunk costs, any increase in cost to the APT which is greater that the defender's cost may be considered a success. A complete success would be a non-monetary cost of withdrawal from the defender's system.

This author asserts that increasing the cost of business varies with the needs of the attacker. This thesis has previously discussed the cost of personal records but it is further asserted that the value of intellectual property can be much higher, with its loss potentially putting a company out of business. This thesis has also highlighted the cost of the loss of personal records to a company and not to those whose records have been lost.

Increasing the business costs associated with an attacker who wishes to sell personal financial information i.e. identity theft and credit card numbers may simply be a case of how much an attacker may receive for such information multiplied by the number of records sold. However, increasing the business cost of an attacker who wishes to acquire intellectual property may be increasing the cost to the cost of research plus the cost of sales and licensing of successfully marketing that intellectual property which may run into the millions of pounds.

O'Connor asserts that:

"For their own survivability in the face of ongoing or future attacks, it is essential for an attacker to always clear the bar with the *least effective technology that will accomplish the mission."*

(O'Connor, 2014, p. 3).

O'Connor goes on to discuss the discipline of one APT with regard to military conservation of force (O'Connor, 2014, p. 12) and provides costs of exploits (O'Connor, 2014, pp. 3,4) as well as taking advantage of attackers in a Red Team test (O'Connor, 2014, pp. 19-21) and the attackers' coherence of the world which was not a truth, previously discussed by Ellerton (ABC, 2019c). This deceit aligns with the LMKC "Courses of Action Matrix".

Some of these models are complicated and complex. Elsewhere (Cohen, 1994, pp. 143 - 146) provides a cost analysis of viruses and select defences. It is suggested that an easy, non-mathematical, way of costs is to simply add the salaries of IT security staff, the cost of their interaction with business staff, cost of defence software (e.g. capital costs, licences), and a proportional cost of hardware and electricity (although one may consider the coat of hardware as sunk cost). In addition, post-incident clean-up costs and business opportunity cost may be added.

One of the LMKC mitigations is deceit (Hutchins, Clopperty and Amin, 2011). This thesis does not have to publish all of the research findings.

Many AV products have known locations on the machine and names (Falliere, Murchu and Chien, 2011) p 14. Any such product should make use of APT techniques and hide anywhere within the system. Such a technique would increase the cost of business for the APT as they would not know where to look for AV products in order to circumvent them.

Further work on this thread may be along the lines of the Nash Equilibrium etc. as highlighted by Fang at el (Fang *et al.*, 2014).

This first is the disruption of the Mechanism of Action of the malware

7.6 A Model of Malware Existence and Execution – The Mechanism of Action

The thesis will now turn to development of the concept of the Mechanism of Action with the practical benefits being described in the next chapter.

Malware also goes by the term “virus” and so this thesis again takes the liberty of borrowing from the medical world. In medicine “Mechanism of Action” is defined as

“The means by which a drug exerts its effects on cells or tissues.”

(Farlex, 2018)

Malware’s mechanism of action is performed through “time” or a call by other software which may be a consequence of the boot process or a user interaction with the machine e.g. program execution (e.g. browser), button push, mouse click.

It should be noted that “Mechanism of Action” may not be a new concept in IT as further research highlights that it has been used in the IT industry as early as 1972 (Lynch, 1972); although the term could only be found once and there is no mention of medical link.

It is desirable to explore biologically link a little further but not to go too far down this road. It is noted that a virus and a bacterium are not the same. What is meant by a computer virus? One biological definition is:

"Living things are made up of cells, maintain homeostasis, move, feed, grow, reproduce, respire, respond to environmental stimuli and excrete waste, while nonliving things do not have all of these characteristics."

(Reference, 2018)

although a virus may not be a living organism.

Antibacterial action inhibits or relegates some enzymes and also disrupts (and interferes with) the membrane structure (Merck, 2018). Inhibitors may inhibit cells wall synthesis, cell membrane function, protein and nuclei acid synthesis; and other metabolic processes. (Unknown, 2018). "In general, bioactivity profiling methods are based on the principle that compounds with the same mechanism of action will have similar behaviour across different biological assays." (Schenone *et al.*, 2013).

Elsewhere (Racchi *et al.*, 2016) provide an example of the same effect (i.e. the therapy to the disease) but the mechanism of action of two products is different. In a similar manner two different APT's may have the same effect on a victim's machine but their mechanisms of action (i.e. malware selected and used, delivery, exploitation etc.in the LMKC) may be different. This medical analogy will be further explored.

A computer virus has a cell wall/covering (structure) and internal processes may, therefore, move (or be moved on the HDD), feed (take input) reproduce (propagate within a HDD or across computers), excrete (output).

Furthermore, in biology, there is a gain of function which is

"A change in DNA that results in the synthesis of a protein with a new or different function. Gain-of-function mutations are typically dominant."

(Farlex, 2020)

Arguably malware which modifies the mechanism of action of software has a gain of function.

In the IT world this may be likened to interfering with the structure of the executable or dll; the input; the internal processes; and the output. The victim does not have to destroy (overwrite) the malware: they merely have to disrupt its mechanism action; modify its input or output; or affect its structure or internal processes.

In Section 2.6 of this thesis the concept of “Tactics, Techniques and Procedures (TTP) has been discussed: different malware acts in different ways and the modus operandi of different APTs may be observed. There is an analogy in the sports world.

The Australian netball team produced an emphatic victory over their rivals New Zealand partly through the work of Mooney et al (Werner and Webb, 2017). Mooney, inspired by the work of Ackoff (Ackoff, 1993), discovered that that each national team had its own quantifiable patterns of play and that these patterns were observable all the way down to junior level. These quantifiable patterns are a function of each national system or philosophy of netball.

Ackoff defines a system as:

“A system is a whole that consists of parts each of which can affect its behaviour or its properties.” and that ” Each part of the system, when it affects the system, is dependant for its effect on some other part. In other words, the parts are interdependent. No part of a system, or collection of parts of a system, has an independent effect on it”. “When a system is taken apart it loses its essential properties.”. “The performance of a system depends on how the parts fit, not how they act taken separately.” “Until managers take into account the systemic nature of their organisations most of their efforts to improve their performance are doomed to failure”

(Ackoff, 2010)

Systems are not a sum of their parts but products of their interactions. A car carries at least one person from one place to another but not one part of a car can do that. In this way a Windows machine may be viewed as a product of its interactions and perhaps APTs have their own quantifiable patterns beyond the basic IOCs that

AV companies observe. Developing this further, it may be possible to observe mechanisms of action that are quantifiable.

Working with the University of Sydney Mooney discovered individual fish in a shoal of fish a few basic rules for keeping at the right distance from each other and that this had been simulated by computer scientists.

Extending this work (biological analogy, systems and parts), this thesis now asserts there is necessary and sufficient grounding to treat malware. This aligns with the concept of the "Principle of Parsimony" of Ockham's Razor, previously discussed. This also neatly fits with the extended Lockheed Martin model discussed earlier for the Courses of Action Matrix: Deter, Detect, Deny, Disrupt, Degrade, Deceive, Destroy; specifically fitting in Detect, Deny, Disrupt, Degrade, Destroy. By highlighting the attack with the Courses of Action matrix it can be seen that the actions in the cells are necessary and sufficient to increase the APTs' business costs:

	Mechanism of Action	Input	Output	Structure	Internal processes
Deter	No	No	No	No	No
Detect	Yes	Yes	Yes	Yes	Yes
Deny	Modify, delete or cut	Modify or delete	Modify or delete	Modify or delete	Modify or delete
Disrupt	Modify, delete or cut	Modify or delete	Modify or delete	Modify or delete	Modify or delete
Degrade	Modify, delete or cut	Delete	Inhibit	Modify or delete	Modify or delete
Deceive	Modify	Modify	Modify	Modify	Modify
Destroy	Delete	Delete	Delete	Delete	Delete

Table 7-1: Courses of Action

It therefore follows that breaking the sequences of the mechanism of action will break the attack. This thesis asserts that attackers, as a business, will do the least they can for the maximum action on objections (as described by the LMKC) :

It has been demonstrated in this thesis that malware persists or not, how enters the computer and the mechanisms of action. These may be summarised:

Persistence -> Malware runs by:	Persistent	Non-Persistent
Time	WMI (Review); Windows Service (Review); Sleep (Review)	Switch off
Action	Registry Run Keys (Hash and Clear); AutoRun (Hash and Clear) Executable Injections: Valid Windows Software/directories (Hash/Sign and regularly check. Statistically check); Browser (Hash/Sign and regularly check. Statistically check)	Credentials (Frequently change all at the same time)

Table 7-2: Known to Operating System

This may now be reflected back on the other CKC models which were not selected for this thesis. It is not the intent to go over all 20 rejected models but just the IPA model.

IPA's stage 4 lists: attack infrastructure by establishing a backdoor, obtaining device and configuration info. Stage 5 lists Penetration/Exploitation by invading other devices and obtaining admin info. Stage 6 is Mission Execution by stealing info and destroying systems. Each of these threads of attack may by treated by the model in the table above. For example, all of the attacks are software that runs on the victim’s machine and must, therefore have a mechanism of action.

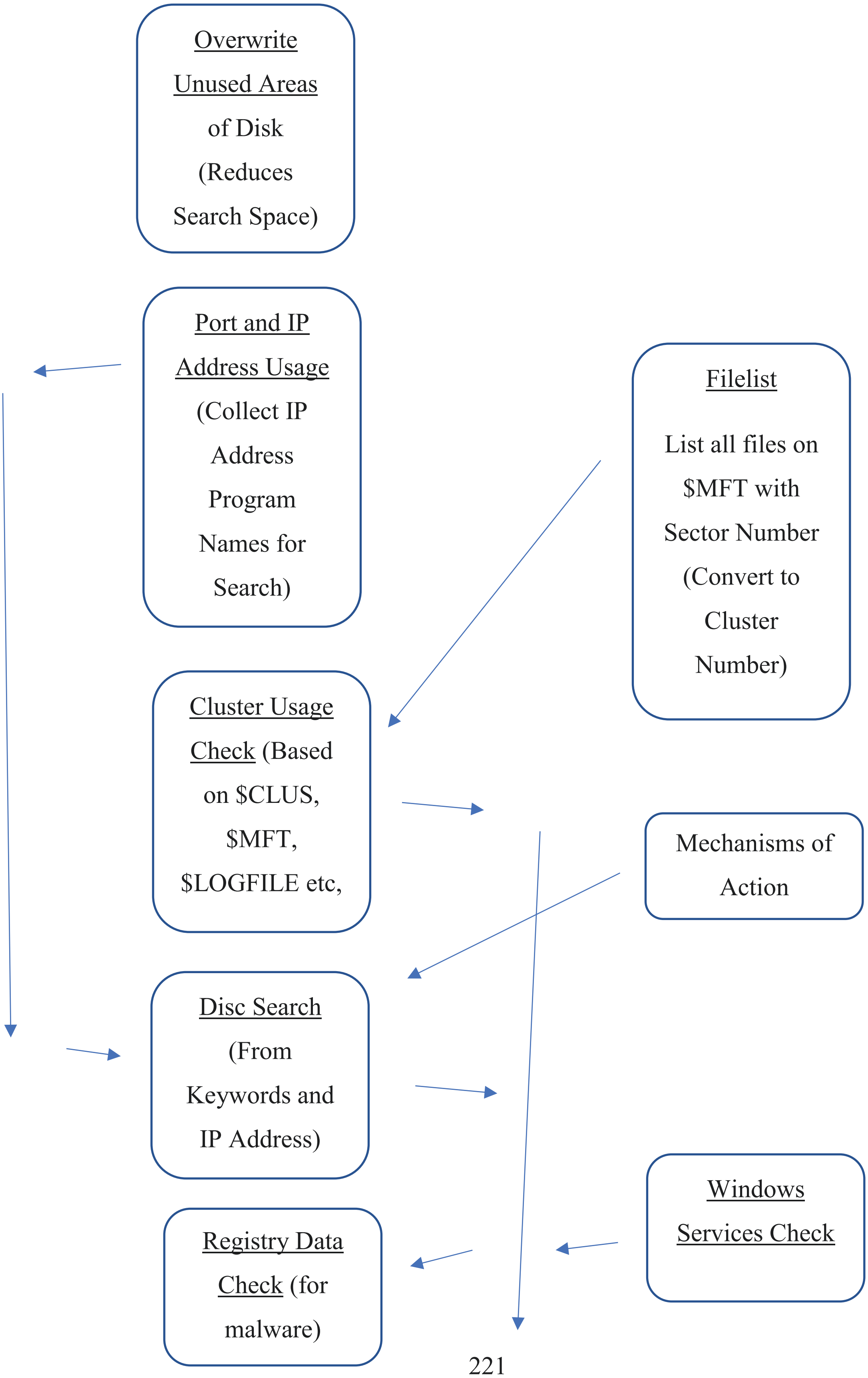
Identification of simple encryption techniques such a Caesar cipher or repeating key as in the Vigenère Square should force APTs to change their encryption scheme and hence increase their business costs.

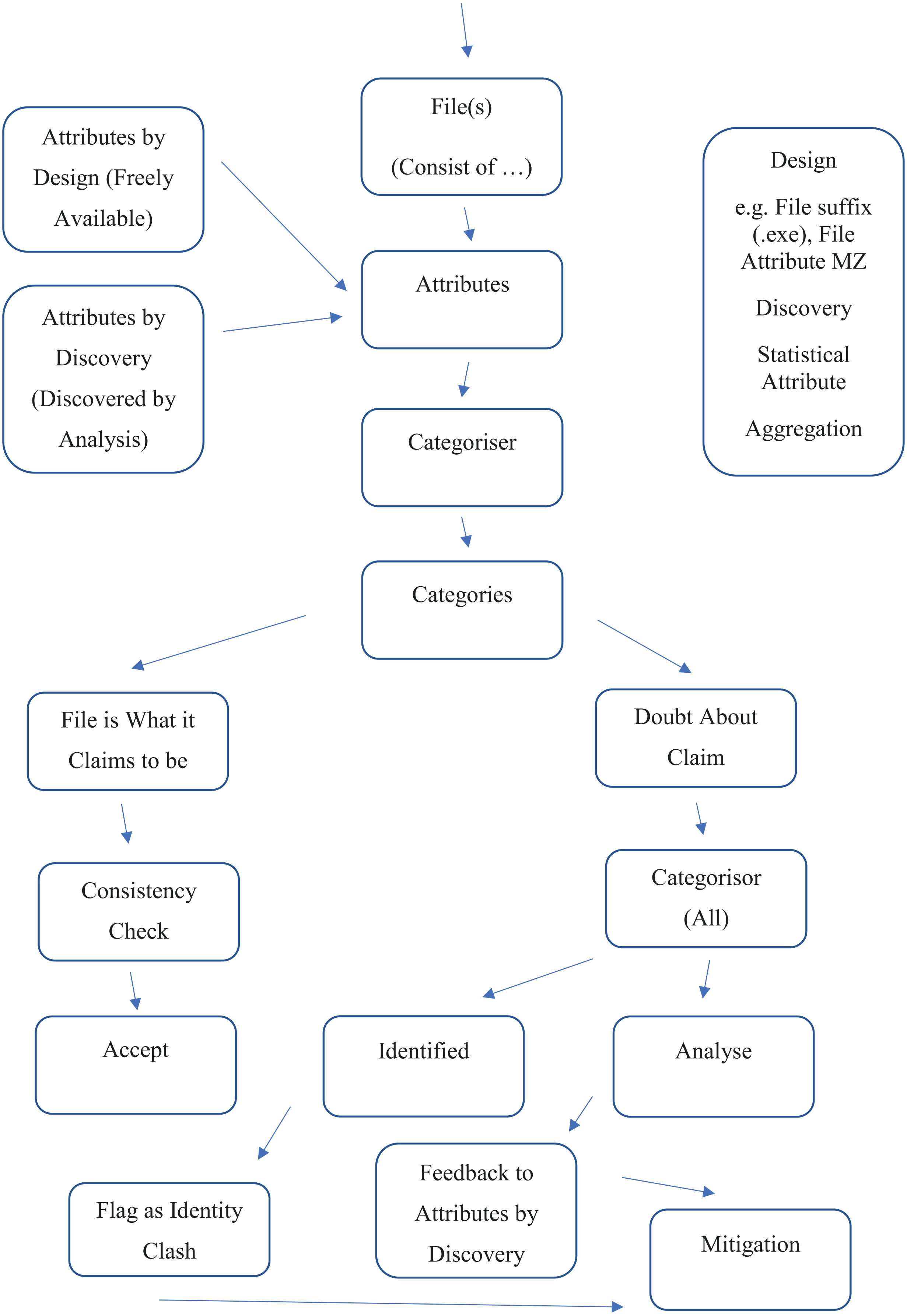
7.7 Categorising the Data

This thesis has demonstrated that Windows system programs may be replaced by malicious programs. Although it should be possible to check the integrity of these programs through hashes and signing this thesis assumes that any software not written specifically for this thesis may be suspect. Therefore, analysis programs were written for all of the analysis supporting the thesis. Ideally these programs should be compiled using different compilers and their functionality compared but this thesis uses Microsoft Visual Studio 2019. The programs are written in C++ but not using Object Orientated code: the code is, in effect, C code as this author prefers C and uses Visual Studio for its functionality and ease of use. Routines used were as close to the OS and ANSI C as possible. For example, using fread and fwrite rather than the Microsoft routines ReadFile and WriteFile. One of the underlying philosophies of this thesis is to trust no-one and trust nothing.

It has been shown earlier in this thesis that compartmentalising data is common to both attack trees and data structures and so it is now possible to categorise (and hence analyse) the data. A schematic for this process is given on the next two pages.

Figure 7-2 Anti-APT Structure and Data Flow


Overwrite Unused Areas of Disk (Reduces Search Space)
Port and IP Address Usage (Collect IP Address Program Names for Search)
Filelist
List all files on $MFT with Sector Number (Convert to Cluster Number)
Cluster Usage Check (Based on $CLUS, $MFT, $LOGFILE etc,
Mechanisms of Action
Disc Search (From Keywords and IP Address)
Windows Services Check
Registry Data Check (for malware)


File(s)
(Consist of …)
Attributes by Design (Freely Available)
Attributes by Discovery (Discovered by Analysis)
Attributes
Design
e.g. File suffix (.exe), File Attribute MZ
Discovery
Statistical Attribute
Aggregation
Categoriser
Categories
File is What it Claims to be
Doubt About Claim
Consistency Check
Categorisor (All)
Accept
Identified
Analyse
Flag as Identity Clash
Feedback to Attributes by Discovery
Mitigation

Without a structure and means of process the operating system would not function. Each part of the operating system has attributes as it has been designed. However, this is not the universe of attributes. There are those attributes which have not been designed or are at least the design is unknown to the designers. This thesis notes the format and attributes of files, specifically PEs and DLLs, and has denoted these as "Attributes by Design" i.e. any attribute which is deliberate and is therefore known to the designer. Attributes which are unknown to the designer and have been determined by analysis this thesis denotes as "Attributes by Discovery".

As discussed, this thesis is concerned with malware on a HDD. It is restricted to the platter with microcode on the HDD controller out of scope.. It has been demonstrated earlier in this thesis that the Windows OS family is file-based and so, at any one time, there will, conceptually, be N files known to the OS i.e. N active files listed in the MFT. Let the start of a computer's life be time zero and at this time there will be N_0 files. Our count may be the number of files at login or at every second or minute. At time one there will be N_1 files, at time two N_2 files and at time t there will be N_t files. At each time the rest of the HDD is unused space for further use and consists of files marked for deletion and other free space. Philosophically this thesis does not consider the latter two sets to be to be a set of files and free space but one big file that by inference is known to the OS i.e. it is not known to the MFT directly but may be inferred simply by not being listed in the MFT. The Windows OS groups sectors into clusters and marks clusters, in the cluster table, as being used or unused with the latter being available for use. Windows considers the cluster table as a file and it is listed in the MFT. One can consider, therefore, the space available for use as a file. This extra file has no structure and no attributes (mathematically the null attribute), which itself is an attribute.

Therefore, at time t, conceptually, there are N_t+1 files on the HDD.

The Microsoft Windows system is file based, therefore, each file type (e.g. PE) has been designed and an analysis of a homogeneous group of such aggregated files might reveal Attributes by Discovery. Therefore, a way of producing such sets of homogeneous files is needed i.e. there needs to be a way of extracting for analysis (for example) all PE (.exe) files or all batch (.bat) files. After analysis and the revelation of any Attributes by Discovery, these recently discovered attributes may be used for further categorisation. The program to categorise the data is called the categoriser.

Once the homogenous set of data (based on selection criteria) is collected there is a need for analysis. This program is called the analyser. Such an analyser will contain tests built from the white paper and academic paper review, attributes by design and attributes by discovery.

Symantec discuss the need for clean data and the benefits of profiling (Uscilowsk *et al.*, 2008) which include Quality Assurance (QA) and detailed information for engineers and customers. It is expected that all data will, in this sense, be clean, as the integrity of data on the HDD is assured by the Windows OS.

Indicators of Compromise (IOC) are ways to identify APTs on a machine (Sancho *et al.*, 2012, pp. 2-3, 7) with examples such as: Registry changes, file changes (Windows and other). Event log analysis, Service changes, Mutexes and network communications. IOC have been gathered for this thesis. Some mitigations will be explored later in this thesis.

7.8 Categorising the Malware

Historically Fireeye categorised malware by primary purpose and go on to note that advanced malware may have a combination of features. This is congruent with the mechanism discussion in Section 9.9. FireEye further note that the top 50 malware families generated 80% of successful infections (FireEye, 2011, pp. 3-6).

It is suggested that there should be an avoidance of guesswork and that samples should be split into malicious and non-malicious groups.(Suenaga, 2009). This may be done by categorising based on attributes by design and attributes by discovery. It may also be done by extrapolating observations and making educated guesses based on parameters and strings. This is a theme to which this thesis will return.

This concept may be extended to the legitimate software on the machine – it is known and therefore may be categorised in some manner. From this one can identify potential malware as it is not, in effect, in the machine's legitimate software asset register. This narrows our malware search considerably i.e. It can now be suggested that one approach to the categorisation of malware is not to look for malware but to look for "goodware" and the rest is therefore suspect.

7.9 Modelling exe and .dll (What does an executable look like?)

It has been previously shown that attackers sometime encipher their data using a single XOR key or short multiplies of the same key. Schneier (Schneier, 1996, pp. 14-15) highlights one way to categorise using the Index Of Coincidence (Friedman, 1935). This statistical technique can reveal the use of repeated key however testing for very long key lengths on every file can be computationally expensive. The structure of a PE can be used in our favour.

Although it is possible to identify short stretches of language (and by inference, malware) (Lui and Baldwin, 2014) the method selected for this thesis is that of n-grams. i.e. known strings

A simple method to identify encryption and obfuscation is needed. For base64 encode a simple count of the number of unique characters in a file is sufficient. A lot of encryption is by XOR-ing repeated hex keys.

The first program developed assumed that the PE has been XOR enciphered by a repeating key. XOR is chosen as it takes less CPU time than other methods of encryption such as mod 26 or mod 256. This repeating key may be of any length but it is reasonable to suppose that is one repeated byte, a multiple of two bytes or a divisor of 64 (2, 4, 8, 16, 32, 64). Given that the longest XOR encryption key recorded in the white papers is 32 bytes (Symantec-Security-Response, 2015b), as highlighted previously, this is a reasonable test. In addition the 4-bytes XOR was not on files but IP address (Wyke, 2012b, p. 28). Whatever the key length it repeats at the 65th character. Note that the key lengths 2, 4, 8, 16, 40 and 80 repeat at the 80th position. It is possible, therefore, devise a cheap test for repeated XOR decipherment.

A simple program to print out the first 100 characters of a number of PEs show that the first two characters of a PE are always MZ and that 0x0E, 0x1F are at positions 64 and 65. In addition the phrase “This program cannot be run in DOS mode” starts at column 79.

We now have a computationally inexpensive method for testing for encipherment with a repeated key starting K_1, K_2, K_3, K_4 etc. It is inexpensive as all that has to be done is to test the recovered key against its offset i.e. with a key length that is a divisor of 64 (C_1, C_2, C_3, C_4 etc. are the resulting cipher characters):

“M” XOR $K_1 = C_1$

“Z” XOR $K_2 = C_2$

The key repeats after 64 characters, therefore:

0x0E XOR $K_1 = C_3$

0x1F XOR $K_2 = C_4$

XOR the cipher that is provided by the same key:

(“M” XOR K_1) XOR (0x0E XOR K_1) = C_1 XOR C_3

the K_1s cancel which reduces to

“M” XOR 0x0E = C_1 XOR C_3

Similarly

“Z” XOR 0x1F = C_2 XOR C_4

Therefore, the test is that these two questions are true for encipherment of repeated key with a key length that divides 64. However, such a method of encipherment used has only been seen to key length 4. Similar tests may be devised for other key lengths using the structure of a portable executable.

However, some files were found not to conform to this format and so another test was needed. The Index of Coincidence model, above, is ideally placed to find data encrypted with a repeated key and also data obfuscation by “spelling out” hex characters e.g. “0x55,0x8B,0xEC,0x81,”.

Clearly sliding such a file any multiple of five characters with align the “,0x” characters and hence highlight such obfuscation. Likewise, the string “\x52\x32\x56\x30”, discussed earlier, when slid against itself would highlight multiples of four characters as the “\x” would align.

The approach considers the Vigenère Square properties of repeating keys, referenced earlier in this thesis counting the number of repeated characters at given offsets.

7.10 Conclusion

The chapter distils the AV companies literature and aligned it with the LMKC for clarity. It supports the third Aim and Objective.

This chapter has discussed:

- Dates contained within files may be analysed and grouped to highlight possible malware. Some sort of categorising software is needed;
- Files enciphered by a repeating key may be identified by “sliding” the file against itself at various offsets, counting the number of hits and performing statistical analysis. Significant regulates occur at multiples of the repeating key length. This method will also highlight files with obfuscated code such as “\x52\x32\x56\x30”;
- Base64 encoded may be identified by counting the number of unique characters in a file;
- A keyword identifier is needed;
- Mechanism of action and the concept of breaking this chain.

The chapter concluded by developing an APT business model and discuss how the APT costs may be increased.

The thesis will now be used the information gathered to develop the AV software.

8 SOFTWARE DEVELOPED IN SUPPORT OF THE THESIS

8.1 Chapter Overview

This Chapter starts with a recap of the philosophy of the thesis and from this proposes a defender/attacker cost trade-off. It then goes on to discuss the analysis performed based on the software developed. The chapter follows with some simple mitigations that could be implemented as a result of the findings and how these mitigations could be used to increase the cost of business for APTs. The chapter concludes with some technical mitigations.

The philosophy of the thesis has been previously discussed and this is translated into a business process in this chapter. Some malicious software may replicate other available legitimate software i.e. as has been shown not all software developed elsewhere cannot be trusted. The chapter supports the third and fourth Aims and Objectives.

8.2 A Short Comment on YARA

"YARA is a tool aimed at (but not limited to) helping malware researchers to identify and classify malware samples. With YARA you can create descriptions of malware families (or whatever you want to describe) based on textual or binary patterns. Each description, a.k.a. rule, consists of a set of strings and a Boolean expression which determine its logic."

(Unknown, 2019).

This thesis asserts that YARA does not fulfill of the requirements of a to identify malware. Unfortunately, there is little explanation of the underlying mathematics of the "Math Module". For example, "mean(string)" returns the mean of the string but is this with respect to an ASCII 256-bit alphabet or a base64 alphabet which has been randomly chosen by the attackers? This thesis seeks to adopt a scientific documented approach to malware categorisation.

8.3 Software Development Enabling Work

Over 50 programs were developed in support of this thesis. Many were enabling programs either to test ideas or create subroutines for the final programs.

Given that this is a thesis based on Microsoft Windows systems it was decided to write all programs using Microsoft routines in C. Visual Studio C++ was used but no use was made of the C++ facilities. Visual Studio was simply an enabler. Furthermore, it was decided that, as far as possible, software would be written using the lowest common denominator with respect to software development and that software would be written in ANSI C or something similar. Therefore _open, _read, _seeki64 etc. would be used instead of the Microsoft equivalents: OpenFile, CloseFile, SetFilePointer etc.

A number of enabling programs were written:

- ***MBRprint*** prints the Volume Boot Record (VBR) and compare the output against Microsoft routines such as GetDiscFreeSpace; sec_check. Printing the VBR also allows the user to view the VBR by eye and look for an VBR infected with malware (Mandiant, 2017, pp. 14-15);
- ***MBR_check*** outputs the Master Boot record as a hex dump. Later incorporated into other programs;
- ***drive_check*** lists all of the logical drives (A-Z) and if a drive exists then it outputs the raw MBR as a hex print with and printable characters as well as interpreted contents. It also lists the existence of the physical drove (0-15). This logical drive information is obtained using GetDiskFreeSpace, GetDiskFreeSpaceEx, DeviceIOControl using IOCTL_DISK_GET_DRIVE_GEOMETRY and FSCTL_GET_NTFS_VOLUME_DATA, as well reading and interpreting directly from the MBR. The latter allows the defender to check the information against the same obtained from

Microsoft routines. This is because there is evidence (previously discussed in this thesis) of APTs modifying Microsoft software;

- ***read_mbr*** looks for all logical and physical drives by looping through A to Z and 0 to 15 respectively, Superseded by ***drive_check***, above;
- ***sec_check*** outputs the given sectors as a hex dump. It is written using Microsoft routines, and is a program to print given sectors, in hex and ascii, 16 characters to a line;
- a program to list all ascii character with binary, decimal and hex values.
- ***hexprint*** lists any file as a hex dump. Each consecutive set of 32 hex characters is preceded by a character count and followed by the 16 characters they represent in ASCII. ***hexprint*** was also coped into various other programs to view data;
- ***sec_write*** was the first attempt to write directly to a sector It is written using ANSI C i.e. no Microsoft routines.
- ***afcount*** walks the file directory and outputs file type and length. This formed the basis of later programs which needed to access all files but not by using the $MFT;
- ***aprogram*** is an experimental program to get system information, change the computer name and modify a registry key;
- ***arcount*** is a program adapted from two others (Microsoft, 2017l), (Various, 2012) to walk the registry tree and output key types and length;
- ***timestamp*** converts date/time stamp from the format in the MFT to human readable form. It is incorporated into the program ***filelist***;
- ***test*** demonstrates that the CRC routines for ***sub_count*** work;
- ***ambrread***. Reads the master boot record of a given drive and prints its contents. Calling sequence is *ambrread* [drive letter] e.g. "ambrread C";

- ***asectorsprint***. Reads and prints sectors in hex and ascii. Calling sequence is ***asectorsprint*** [drive letter] [sector length] [start sector] [end sector] e.g. "***asectorsprint*** c 512 1 50". The sector length is obtained from the program ***ambrread***;

Microsoft routines, whilst functional, can be hard to use. This is borne out by internet comments. Early in the thesis development, therefore, it was decided to use a minimalistic programming with routines as close to the kernel as possible. C routines such as open, read and seek are much easier to work with than Openfile, Readfile, SetFilePointer etc. _int64 is a more descriptive definition than its Microsoft C code equivalent. Furthermore, any code needed to be portable should it be decided to test it elsewhere. An APT would, no doubt, think the same way, as should any developer.

A further problem with the Microsoft routines is that, when using them to read a HDD as a file, there are issues with the end of file. The C routines are able to handle this event gracefully.

8.4 The Programs Written – A Summary

A variety of programs were written. They are summarised here with detailed descriptions of provided later in the chapter:

- ***filelist*** lists all files and directories from the $MFT whether active or deleted;
- ***ports_struct*** uses Microsoft routines to look at which IP addresses are being used with associated open IP4 and IP6, TCP and UDP ports;
- ***searchall*** looks for a given string in a range of sectors;
- ***searchip*** looks for IP addresses in the form of IPv4 in a range of sectors;
- ***searchemail*** looks for email addresses in a range of sectors;
- ***clusterusg*** analyses HDD used and unused clusters;

- ***categoriser*** categorises the data according to parameters provided for the Analyser (below);
- ***Analyser*** Analysis file according to parameters provided
- ***sub_count*** builds a corpus of subroutines from a pristine Windows OS with associated probabilities.
- ***driver*** overwrites a given sector of a HDD with the phrase "hello world".

8.5 The File List Program

filelist directly lists from $MFT all files and directories whether active or deleted (Sammes and Jenkinson, 2007, pp. 215-275, 389-410), (Russon and Fledel, Undated), (Richard Russon, 2018), (Wilkinson, 2017). It also lists the start sector of the file's initial position on the HDD and has the option to list all dates associated with the existence of the file. *filelist* is inspired by the program *ntfsdump* (kusano, 2015). The program structure of f*ilelist* is different to that of *ntfsdump* and the output is slightly different. *filelist* outputs readable, relevant, MBR information as well the whole MBR. True and relative data run lengths for the MFT are output. The existence of each file is output in in a form similar to *ntfsdump* but also includes the HDD sector number on which the information resides.

It has been demonstrated that APTs will modify the software to help hide their malware. For example, the DOS dir command and Task Manager.

There are three ways which files may be listed:

- Using Microsoft Windows OS commands e.g. dir, forfiles;
- In a program using a directory walk;
- From the MFT.

filelist is a demonstration of the latter.

There was a problem with the development of *filelist:* It is asserted (Sammes and Jenkinson, 2007) that in an Attribute Header bytes four to seven should be the attribute length. During analysis this author found that this did not function, as byte six was non-zero which made the program crash. This could have been a mistake on the author's machine but why would an attribute have a maximum length of 0xffffffff, i.e. 4GB? Is there really a scenario where an attribute can be this long? Clearly one byte is not enough as that would make the longest attribute 256 bytes. An attribute length using all of two bytes would give the maximum attribute length of 65535 bytes, far beyond the two-sector length of most machines. This author reduced the field length to two bytes and the program worked.

8.6 The Sleep Program

In support of this thesis a program was written and run from the Registry RunOnce and Run keys. The program was only made to sleep. The observation is that the Registry Key values may have run in a non-sequential order. It appeared that System32 and other folder calls ran first, before the sleep program. However, the RunOnce and then Run Registry keys order was demonstrated to be true through analysis of the order of the Process Ids. After signing on another program was started under a regular user. The Process Id of this program was "out of sequence". The hypothesis is that PIDs that have completed and hence released by the O/S and reused within session. PID number is, therefore, not a good indicator of malware identification.

For administrator access, the sleep program call from HKLM RunOnce and made the start-up process hang. Only on termination of the process would the start-up process complete. It would not be too difficult to make this a recurring denial of service attack.

8.7 The Port Scanner

ports_struct uses Microsoft routines to look at which IP addresses are being used with associated open IP4 and IP6, TCP and UDP ports. Running continuously, it is similar to running "*netstat -ano*" except that it checks at random intervals which

IP addresses and associated ports have changed and outputs this information. Should information be unchanged then nothing is noted. Using ***ports_struct*** allows the defender to list gather information on IP addresses accessed with the associated ports. It may be used in conjunction with other logs.

To identify C2C it is necessary to analyse Port Usage and Associated IP Address. A program was written to check the opening and closing of ports in real time and list associated remote IPv4 and IPv6 addresses, process ID and program name. From this data may be collated that may be used in the disc search program. Use will also be made of the IETF Service Name and Transport Protocol Port Number Registry (IANA, 2020). Should there be a connection, or attempt at connection, to an IP address that the defender does not recognize, or allow, then this would be a possible Indicator of Compromise. This Registry also allows the defender to verify that the correct protocol is being used on any given port. Additionally, should there be a suspect IP address that cannot be found on the HDD then this may be an indicator that a program is using encryption or obfuscation to deny the defender viewing access.

Although some malware uses ports that are used for legitimate services, some malware uses specific ports. For example, many cryptocurrency mining uses ports 3333 and 7777; the first for low-end machines, the latter for high-end, higher-capacity machines (Insikt-Group®, 2020a).

Consideration should also be given as to how often this program needs to check for anomalous behaviour and what LMKC Course, or Courses, of Action need to be performed should potentially malicious behaviour be detected. The frequency of this checking should be a function of the size of the crown jewels divided by the internet connect rate. For example, if our crown jewels have size of 50GB and the internet connect rate is 250MB per second, then fastest the data can be extracted from the system is 50GB/250MB per second or 200 seconds. The defender, therefore, needs the ports program to perform checks at least every 200 seconds; any less frequent and the exfiltration may be missed. The defender also needs to consider how much data they are prepared to lose. Of course, the data should be encrypted but would the attacker be able to reconstruct the data from a proportion of the crown jewels and if so, what proportion? The defender also needs to build in decision making time.

50GB of data is a reasonable figure for this example: one attacker is thought to have exfiltrated 50GB from one organisation from a total of "terabytes of data" (Chang *et al.*, 2015, pp. 5, 51)

We now use the fictitious figures to demonstrate the strategy: assume that that the full set of data may be reconstructed with only 10% the known data and that it is assessed that no more than 6% of our crown jewels can be lost. Dividing the data into segments, each segment being less than half of 6% (exfiltration of one segment of 3% may not be enough to come to valid decision). Choose 2% of the data. This gives 50 segments for the full set of data. It may be calculated that the full set of data can be exfiltrated in a minimum time of 200 seconds so it is concluded that the program needs to run every four seconds (200 seconds divided by 50 segments) with the outcome of exfiltration detection being internet shutdown or machine power down. This strategy allows for two adjacent loops of the program to agree that there is problem and another four seconds to perform our exfiltration detection action. The desired outcome has, therefore, been done in eight seconds. It is acknowledged that attackers may exfiltrate data in parts to different IP addresses but increasing their costs here means that they would need to know the defender's strategy. A lot of different small data exfiltration to a number of (unknown to the defender) IP addresses is as big a "red flag" as one large data exfiltration attempt to one IP address.

The part-acquisition of data by an attacker may not be as fanciful as it at first appears. Data may be stored in RAID (Redundant Array of Independent Disks) disk arrays (Patterson, Gibson and Katz, 1988) (SNIA, 2009), (Sammes and Jenkinson, 2007, pp. 207-209) with data and parity stored on different disks. Acquisition of a subset of the data may allow reconstruction of the whole set of data simply by using the defender's legitimate error correction and business continuity techniques. This is a piece of work for the well-resourced attacker which could use the properties of CRCs e.g. n-bit error detection, (n-1) bit error correction etc.

The Port Scanner satisfies three parts of the LMKC Courses of Action Matrix: Detect, Deny, Disrupt.

The IP addresses and program names can now be used an input to the Search program.

8.8 The Search Programs

The search problem may be segmented into two different search types: non-IP address search; IP address searches. The former is split into searches for string provided by the user and searches for email addresses. The three programs are: *searchall*, *searchemail*, and *searchip*.

The various types of string search are tabulated and then discusses:

	Non-Distributed	Distributed
Software	Linear Ordered Table	Division of Work Ordered Table
Hardware	FPGA Linear	FPGA Linear

Table 8-1: String Search Comparison

There are two ways to perform a search: Should it be needed to search the data just once then a sequential search is best (Software Non-Distributed model); however if there are multiple searches then an ordered table is better (Knuth, 1973, p. 406) (Software Non-Distributed model). Ordering a multi-gigabyte or terabyte may be prohibitively expensive. Each sector or cluster could be ordered but for non-standard searches such as looking for the existence of an IP address such an order could be achieved by only sorting together numeric data.

Knuth et al (Knuth, Morris and Pratt, 1977) later develop a fast pattern matching algorithm for strings but this appears to be better for small alphabets. In addition, the expectation for matching in the problem at hand is low and so a sequential search for specific strings is low.

Fast pattern matching is possible using hardware such as FPGAs or ASICs (Woods, Teubner and Alonso, 2011) but this only possible is for wire-speed processing with predictable performance. This thesis is concerned with data resident on the HDD (Hardware Non-Distributed model).

The elapsed search time may be reduced by dividing the search area into n equal pieces and then using n networked computers in a distributed fashion as a "supercomputer". The most famous of this sort of use of idle time on computers is SETI@home (University-of-Berkeley, 2019) (Software Distributed model)..

Consider searching for a four-digit PIN. Instead of a brute-force attack of 0000 to 9999 on one machine, each test taking a unit of time, for a total of 10,000 units of time, 10 computers could be used with the first testing all PINs starting with zero, the second with all PINs starting 1 etc. Each computer performs 1,000 tests for a total of 1,000 units of time. Similarly, using the Seti example each computer could be connected to the HDD and each computer access a number of sectors: the first sectors 0 to n/10; the second (n/10)+1 to n/20 etc.

The search programs are programs to identify strings within HDD sectors. Two programs were written: the first is used to identify specific strings within sectors. For example, the registry "/Run keys" mechanisms of action; the second look for IPv4 IP addresses. The key for both programs is to write tight code. For example, in looking for IP addresses it may be decided to choose to look for specific IP addresses e.g. 127.0.0.1 but there is a need to search for all IP addresses which may range from 0.0.0.0 to 999.999.999.999 (in theory, in practice the upper bound is 255.255.255.255) so the search only needs to look in any particular string for a ratio of full stops to numbers which is between 50% and 25%. Progressing the search and moving along, drop one character off the start, add another to the end and change the counts accordingly. Other searches include, but are not restricted to, software that malware writers use e.g. ZW routines, sleep, XOR etc.

Specific searches will be guided by academic literature and white paper highlights as well as what is uncovered by the Categoriser (below).

An IP address consists of four numbers in the range [0,255], no leading zeros, separated by full stops/dots ("."). Valid IP addresses, therefore, range from 0.0.0.0 to 255.255.255.255. It can immediately be seen that the structure is four dots and four numbers up to four dot and 12 numbers. The search, therefore, is for the minimum and to look for consecutive characters in sectors that have four dots and four numbers and print them out. For 1111111 512-bytes sectors this search took five elapsed minutes. For 1.7GHz Windows 8.1 machine with 450GB this would take about 65 hours.

A new algorithm was suggested (Wood, 2019) to look for three dots, ensure that the difference the distance between them is one, two or three, that the characters in between and one after the first dot and one before the last dot are digits. For the same input test data, the program from above with this algorithm ran in three minutes i.e. 60% of the time. This would reduce the total time on the same machine to just under 40 hours. Segmenting the date equally over 10 similar machines would mean four hours i.e. half a working day. A later review of string matching algorithms did not produce a better algorithm (Cormen *et al.*, 2009, pp. 985-1013).

The output of the more efficient program was just over 30,000 hits and this was imported into a spreadsheet, sorted, obvious false positives removed and the remaining data of just over 20,000 hits was put into a pivot table. There were just over 3,000 unique hits, the vast majority of which were false positives. It was more difficult to interpret some of the data: while 4.5.6.7.8.9 may be contain a false positive how does one interpret 54.186.106.100? Is that a valid IP address or should it be 4.186.106.100 or 54.186.106.10 or 4.186.106.1? All legitimate IP addresses, as categorised by a review, were compared against ICANN (ICANN, 2019) registrations (ICANN, 2017).

It is been demonstrated that APTs use repeated code, Mutexes, Zw routines. A search for any of these (which may take some time) would be an IOC.

The string search program looks for strings as any specified combination of upper and lower cases characters. For example, the ASCII characters “A” is 01000001and “a” is 01100001. One can see that the difference is in third from left byte. This is true for All ASCII alphabetic characters. The search program performs simple Boolean operation with all input characters to a common comparison. For a five-long string this means that 32 different combination of upper and lower case can be reduced to one string for comparison.

A search was done on a personal Windows 8 machine. It discovered the traces of up to three possible pieces of malware.

The search started by looking for the a given term used by many attackers. All of a certain family of commands contain the term. It has been shown earlier on tis thesis that malware writers use these terms to identify systems which have already been infected by their malware.

The results of the search identified a number of associated interesting phrases. These were then included as search terms for further runs of the program and also kept for future reference to be used against other HDDs.

An iterative search such as this may be a candidate for an Artificial Intelligence solution.

8.9 The Cluster Analysis

clusterusg analyses HDD used and unused clusters and is controlled by parameter variables. This program accesses the $CLUS file in the $MFT and compares and contrast cluster usage within and across logon sessions. By storing a copy of the cluster table from login and comparing this against the cluster table at logout it is possible to highlight clusters that were not used and later are, or were used, and are now marked as free. Of course, this does not highlight clusters used, freed and the re-used but the program could be run at irregular intervals during a session. It is anticipated that this program and the disc overwrite program will be used in tandem. ***clusterusg*** will also analyse run lengths of consecutive clusters used and unused.

Clusterusg may also be run in real time looking for clusters that have moved state from becoming free to being used and vice versa.

8.10 The Categoriser and Analyser

categoriser categorises the data according to parameters for the Analyser. It is a collection of statistical analysis techniques. These criteria for the parameters are based on Attribute by Design (e.g. Portable Executable (PE) format) and Attributes by Discovery (q.v. PE file lengths discussion later in this thesis). The Categoriser and Analyser share a common parameter file design and some common tests. This thesis is concerned with identifying malware in any format. This program be can be enhanced iteratively by feeding back Attributes by Design and Attributes by Discovery to develop new categorisation subroutines. These routines may be, for example, statistical scoring mechanism. The statistics may include but are not limited to:

- Index of Coincidence – IoC (to help group data);
- Chi-squared (to help isolate statistically flat data);
- Encryption identifier;
- Analysis of n-grams;
- Benford's Law;
- Other statistical tests as needed once the analysis is underway.

Symantec refer to the need for "Clean Data" (Uscilowsk *et al.*, 2008) however there is not necessarily a need for clean data. With statistical analysis a model of the machine under investigation may be built using just, and only, the data on the machines. For example: in a group of 10 PE files counts may be made of all 10 files. Each file is then, in turn, analysed against the rest of the group creating two sets of data: one of the just one file and the other of all files minus the counts of the file under investigation. This way a sample of collected malware is not needed and any tests are not constrained by comparing against a known, biased set of data. This idea was developed for the PE subroutine analysis program, discussed below.

For the Registry, this program could extract keys where programs and/or batches and scripts are listed. Any Registry key that contains a (executable path extension) will need further analysis and attract closer inspected by the Analyser, below. Particular attention will also be given to Windows Service programs.

This program will also look at Registry key lengths and other Attributes by Discovery (yet to come to light).

Sharing the same parameter file format, these two programs were developed from the idea of a single program to perform analysis files which shared the same attributes. It is possible to categorise data by design or discovery and the module nature of both programs makes it easy to add subroutines for either type of attribute.

The original program became unwieldy and the decision was made to split the concept into two programs: one which would produce a list of files with the same required attributes (***categorisor***); the other would them perform the analysis (***analyser***).

The parameter file uses a pseudo-html format with every parameter being of the form <Keyword [option parameters]>. Comment lines are of the form <#### comment>. For example, a parameter file which allows extraction of all files from a certain sub-directory whose file length is in a given range [20, 3768] would be

```
<#### Directory or sub-directory on which to work>

<Diry C:/Users/user1/Documents/>

<#### File length range>

<MinL 20>

<MaxL 3768>
```

With the categoriser it is possible to:

- Work on a sub-directory or registry;
- Look for encrypted files which start MZ;
- Performa count of file which start MZ, or not and match against a .exe file type;
- Look for Base64 encoded files;
- Highlight files in a given file length range;
- List file names with more than one full stop (".") in the file name;
- List files with their Index of Coincidence;
- Calculate frequency counts for files with a given attribute for the analyser;
- Perform subroutine counts on a set of files for the analyser;
- List files with given strings in given fields.

analyser analyses data categorised by the Categoriser (above). For example, ***categoriser*** may have produced a full path name list of all PE files (all file with the .exe suffix). The Analyser then is then able to perform various tests as defined in the parameter file, highlighting any files. With the analyser it is possible to:

- Test files for a given frequency count using the chi-squared distribution;
- Look files that are encoded with a repeated key;
- Print selected columns from files.

8.11 Subroutine Count

sub_count. It has previously been demonstrated that APTs use subroutines that developers might not normally use e.g. Zw routines, Hook routines, PsSetCreate … routines. This program builds a corpus of subroutines from a pristine Windows

OS with associated probabilities. The idea is that for each executable or perceived executable subroutines may be observed and an overall probability of seeing that set of subroutines may be calculated. A plethora of low probability subroutines, as calculated from the pristine Windows OS corpus will give a low probability of total. As any probability is small the score is calculated by summing the natural log of the probability.

This program analyses the frequency of subroutines used in a corpus of executables that have been selected by the categoriser. These counts are them compared against a putative set of malware to see if they are dissimilar form the corpus.

8.12 The Overwrite Unused Areas of HDD Program

As previously discussed, the idea for this thesis came from the Master's degree. This idea was that all unused space should be overwritten by the operating system.

Two independent studies, (Bentley, 2008), (Wright, Kleiman and Sundhar, 2008), indicate that for all practical purposes, one overwrite should be sufficient. Bentley suggests that reading data by Magnetic Force Microscopy may be achievable but in reality, a lot of time, effort and money would have to be spent on the work; and the outcome would have to be worthwhile. A practical demonstration supporting these assertions was later demonstrated:

> " … a system to recover data from a hard disk drive through magnetic force microscopy was developed to do so from an un-degaussed hard disk under certain experimental conditions. … The performance has been poor even under these idealized conditions, …"
>
> (Kanekal, 2013).

This was without any overwriting of data.

It is suggested (Berghel and Hoelzer, 2006) that "It is clear that most disk wipers leave behind a lot of tell-tale information that may have proprietary or security implications.”, however this is not something with which this thesis needs to be concerned: the intent is to treatment of malware deployed by APTs and so all that has to be done is to overwrite the least amount of code to break the mechanism of action. It is noted that as well as dealing with MFT entries in *$MFT* there will be a need to duplicate any work for the MFT mirror in *$MFT_MIRROR*.

It is possible to send ATA security group commands to perform secure erase (Microsoft, 2017ac) or the Microsoft utility cipher.exe (Microsoft, 2018e), (Microsoft, 2018o), . Disc encryptors may be used to overwrite but recovery of the encryption key enables recovery of the “deleted” i.e. encrypted data. It is not clear if these encryptors encrypt the data directly on the HDD or write it to another file. leaving the original on the HDD. Other dedicated delete utilities are available (Russinovich, 2018) but seem to be recent i.e. later than the initial idea for this thesis.

This author tried to write such a utility but was unable to gain access to the kernel (Ring 0) to write directly to HDD sectors. It was possible to directly read disk sectors. However, given that there is commercially available software, this inability to gain access to the kernel is not considered a failure. The point of this thesis is to present a solution; any valid solution is acceptable.

HDD overwrite programs overwrite unused areas of disc. Microsoft now have two such programs but these appear to have been written after the author had the idea. The software overwrites free sectors, clusters and subsectors. There was a problem developing the overwriting software: initially it was thought that a device driver would be needed but internet research suggested that it could be done from a regular user, using admin privileges. Two programs were written: one using ANSI C and when that did not work another using Microsoft routines. Neither was able to overwrite given sectors. Success was finally achieved using a Kernel mode driver on a virtual machine.

The kernel mode driver writes directly to a specific sector on a HDD. This driver uses some Zw routines: ZwCreateFile, ZwReadFile, ZwWriteFile and ZwClose. This family of kernel routines have been used by APTs, as described earlier in this thesis. The program writes to a sector, reads back the sector and hex prints the sector with associated ASCII text to verify the write. It is a proof of concept to demonstrate this author's ability to write a HDD free space overwrite program and writes "hello world" to a given HDD sector. Fully developed, it can potentially overwrite any area of HDD but in reality, would write to unused areas of HDD A guide to help reproduce this work can be found in Appendix B. It is built on a host Windows machine and runs in a Microsoft VMware Windows virtual machine. This is to ensure that, during development, the any bugs in the software only affect the virtual machine which can easily be rebuilt;

The overwrite program would overwrite any area of unused HDD, including hidden sectors and break the mechanism of action of any malware using clusters not being used by the operating system (Kaspersky, 2015c, pp. 16-18). It could also contain basic attribute by design and discovery as earlier described to highlight possible malware before overwriting.

Should an attacker Live off the Land it is not clear if the action of the program sdelete shows up in program filelist output. It is possible to amend the author's filelist program to only output files with certain date/time stamps.

8.13 Benware – Benign Software as an Educational Tool

In order to try to enter the mindset of an APT and malware developers it was considered that first some pseudo-malware had to be written. This thesis offers a new portmanteau called "benware" i.e. "Benign Software". The portmanteau follows the lineage of "malware" – "malicious software". Benware is presented to the victim and acts like malware but is benign. This benware borrows from real world examples (Panda-Security, 2015), (Mandiant, 2013, p. 35), (Wüest, 2012, p. 22) and uses no bespoke software, only freely available software provided by Microsoft for Windows 8.1 machines. This is an example of "Living Off the Land" with "Dual Use Tools" (Wüest and Anand, 2017, pp. 7, 15-16). The benware collects operating system and user data and is only run as a demonstration on the standalone Windows 8.1 machine. It is neither sent nor run over the internet. Malware which lives off the land may be easily changed to present different digital signature every time it is deployed.

A new comment may be inserted into each piece of the benware which contains a one-up serial number. As well as defeating digital signature this serial number would allow an APT to build a CRM System to track deployed malware.

The benware is produced by gathering a collection of Microsoft routines into the Microsoft Bundler, iexpress (Microsoft, 2017r) and is used to demonstrate to students, in lectures, proof of concept malware. The file is presented as a .pdf file with RTLO naming i.e. “Dummy.pdf   .exe”. The “.exe” is easily hidden in an Internet Explorer Windows directory list. A victim would click to open what is thought to be a .pdf file but is the mechanism of action for deploying the benware. On clicking to open the bundler copies all of the files (Dummy.pdf; extract.bat; run.bat; and vbscript.vbs) it contains into a temporary directory (“%temp%\IXP000.TMP”) and runs a .bat file which creates a permanent directory into which all of the files in the temporary directory are copied. Adobe pdf file is opened with Adobe Reader and a Registry key value (“C:\Permanent Directory\vbscipt.vbs”) is added to the Registry Key HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run. This .vbs script is used for the extraction as .bat file would produce a Windows cmd window which would draw the user’s attention to the extraction. A .vbs script gives no indication that anything is happening other than opening a .pdf. At the next login, due to the Registry Key mechanism of action, the .vbs script calls a .bat file which runs and deletes the Registry Key (the RunOnce key could have been used as it automatically self-deletes but the chosen method was used for educational purposes) followed by collection of system information into a single file using the Microsoft commands date, time, “netstat –ano”, systeminfo, tasklist, ipconfig, route, net (various times with different parameters), ping and tracecert. This provides date, time, open ports, list of users, Route Tables, IP and system information including patching updates. Again a .vbs script is used to as it does not give any sign to the user that it is running. Again, a batch file called from a .vbs script does not produce a command window.

8.14 Some Suggested Mitigations and Gaps

There are many mitigation suggestions e.g.(TrendLabs, 2012a, p. 3), (Huq, 2016, p. 45). It could be argued that “Integrity Monitoring” covers the work of this thesis but this thesis suggests more that: it suggests actively managing secondary storage (HDD, SSD) to provide assurance that what present is what is expected to be present. What exists is necessary and sufficient to do what the business, IT administrators and users require of the machine.

A high level view of deleting data is available which helpfully mentions data in the cloud (TrendLabs, 2013a) while a de-obfuscation tool base on a number of obfuscation techniques has been suggested (Suenaga, 2009, p. 19).

Kelly suggests amongst other things a comparison of Autorun dataset per and post infection, (Kelly, 2017, p. 13)

Sans produce a 12-step guide to finding unknown malware of which it could be argued this thesis has three: Indictors of Compromise Search, Evidence of Persistence and MFT Anomalies. However “Indicators of Compromise” is a wide title and could be applied to all of the other eleven steps (SANS-DFIR, 2013). While Mitre provide a list of APTs and their persistence techniques (Mitre, 2019b).

8.15 Mitigations

It has been demonstrated that there is heavy use of the /Run ad /RunOnce Registry Keys. The mitigation developed is, at logoff, to take a copy of these keys for further forensics analysis in case they have been changed, delete them and install the desirable keys and values. This would ensure that future sessions would be free of an attack from this source. Similarly, with start-up folders.

Alternate Data Streams may be found using a one-line PowerShell script (Arntz, 2015) or programmatically though MFT analysis (Sammes and Jenkinson, 2007, pp. 415-424) and a modification to the filelist program.

8.16 “Common Sense” Mitigations

Some “common sense” mitigations are to:

- Keep patches up to date;
- Ensure permissions are setup correctly;
- Block unused ports and check regularly for open ports;
- Disable Microsoft RDP and any other remote access tool
- Remove unused and unwanted services.

8.17 Sinkholes for Advertising and Malvertising

It has been previously stated in this thesis that tracking websites were blocked (Al-Fannah, Li and Mitchell, 2018) . The European Interactive Digital Advertising Alliance (EDAA, 2019) lists websites that can be online behavioural advertising preferences can be controlled (EDAA, 2019) and at least one of these websites is in the Al-Fannah et al list and is in the EDAA list. However further analysis demonstrated that simply blocking websites was not enough. Three websites, including one from the Al-Fannah et al list were analysed. Each website had a unique IP address but this was one of a larger block of 128, 256 or 768 of consecutive IP addresses. Only three websites were selected as this author’s academic work was blocked by website and IP address identification sites because of too many requests.

Bruneau (Bruneau, 2010, p. 5) discusses the uses of sinkholes to block traffic attempting to reach the internet. A simple sinkhole may be produced by resolving DNS web address names to the hosts file using, for example, 127.0.01 as the redirect. For further forensic analysis other IP addresses may be used. More complex ones involve other servers with the added complication of DGAs or malware which connects to IP addresses and not DNS web address (Link and Sancho, 2011).

Websites may be blocked by adding “127.0.0.1 www.websitename.xx” to the C:\Windows\System32\drivers\etc\hosts file with a loopback IP Addresses for each website to be blocked e.g. 127.0.0.1 blockthiswebsite.com. Appendix E provides an indicative list of such sites built from the Al-Fannah et al list.

Additionally, the use of a mask e.g. *.*.xx or *.xx to block all websites ending xx would be sensible. A proof of concept Firefox browser add-on written for this thesis may be found in Appendix H. In this way websites from whole regions of the world or whole type of businesses could be excluded by system administrators. This is a known solution (The-Ohio-State-University, 2019), (Hofstetter, 2019). However, work supporting this thesis found that of the FANG companies (Facebook, Amazon, Microsoft and Google) Facebook.com and Microsoft.com could not be blocked by this method. It is obvious why Microsoft should not be blocked (operating system updates) but not Facebook and this, perhaps, throws some light on the business relationship between the four companies. As an aside, developing this piece of analysis found that the hosts file would be overwritten by a default hosts file, presumably by Windows Defender as this program produced pop up indicating threats.

Another way to stop malvertising would be to block all advertisements at the HTML request level. This would mean writing an internet facing parser to edit data returned from the internet. At the higher, browser, level, browsers could perform basic checks on web page source code: for example, it should not be too difficult to identify iframes that would be below the threshold for the human eye and produce a warning.

Google tag Manager (Google, 2019b) is for measurement and marketing tags codes and links to Google Analytics and Google Ads. Blocking this may help reduce the level of malvertising.

It has been shown earlier, that false information is used for registering some websites. DNS registrars could perform basic integrity checks on the information provided for registration, perhaps borrowing counter-fraud techniques from the financial industry.

It may be possible to harvest more websites from data broker links. In at one US state it is possible to request personal details that data brokers hold on the applicant (vermont.gov, 2020). California may soon be the second state (Becerra, 2020).

8.18 More Technical Mitigations

Some more technical mitigations are:

- check the Executable path extensions (.COM; .EXE; .BAT; .CMD; .VBS; .VBE; .JS; .JSE; .WSF; .WSH; .MSC) list;
- check the DLL loading order list (Microsoft, 2017j);
- run a free sector eraser program;
- use a tool to list possible 32-bit DLL locations from a clean system (Harbour, 2010) based on the criteria discussed;
- “As best as possible, figure out all running code” (Macaulay, 2014). It is possible for malware to intercept commands in the kernel and not return information about that malware;
- consider the concept of automatic overwrite is not new (LaBarge, Mazzuchi and Sarkani, 2014). This idea has been explored (Gutmann, 1996), (Bentley, 2008) with the latter being more directed;
- use FireEye suggestions for a number of defences based on WMI (Ballenthin, Graeber and Teodorescu, 2015, pp. 30-32);
- monitor the registry (Microsoft, 2019f), (Microsoft, 2019c) especially the /RUN keys. Use a logoff script to install only required /RUN keys and Services;
- Analyse cluster usage.

8.19 Development Costs

No record was kept of the costs of software development. However, an attempt will be made, using industry standards.

The US Government Accountability Office (Persons, 2020, pp. 17-25)suggests a Life Cycle Cost Estimate (LCCE) for projects. This represent all costs from inception to disposal. Other models are: Independent Cost assessments (ICAs); Budget Estimates; Rough Order of Magnitude (ROM) Estimates; Estimates-At-Completion (EACs), and Independent Government Cost Estimates (IGCEs).

This set of models is similar to those contained within an independent review (Shekhar and Kumar, 2016). Elsewhere Lee (Lee, 2013, pp. 63-64) suggest that decomposition is needed in cost estimation for example hardware, software, salaries, rents and utilities.

8.20 Conclusion

The chapter supports the third and fourth Aims and Objectives. And has presented the software developed in support of the gaps identified in previous chapters the concept of APT and defender's cost and provided a model for a comparison between them and the amount a defender should spend.

It is relatively easy to write malware by Living Off the land. Combine this with an email with a link to a malicious website that contains a malicious script and malware can be laid down on the victim's machine.

9 ANALYSIS OF HDD

9.1 Chapter Overview

This chapter discusses the analysis performed, the outcomes of that analysis and the ideas subsequently developed. It fulfils the criteria of the third Aim and Objective: treat APTs on Windows based IT systems in such a way that the malware is negated. It supports the third and fourth Aims and Objectives.

This thesis has presented the concepts of "Process Driven Defence" and "Landscape Driven Defence". It has selected and, by argument, extended the LMKC Courses of Action Matrix from six to seven courses of action by the addition of "Deter" to "Detect, Deny, Disrupt, Degrade, Deceive, Destroy". The thesis has re-introduced the concept of "Mechanism of Action" to the IT industry from its hardware use and introduced it to software (and by analogy with the medical term with respect to identifying how malware executes on a victim's machine). It has also introduced "Attributes by Design" and "Attributes by Discovery". This chapter will bring together all of these concepts and use the software devised, designed and developed to help increase the costs of APTs.

The chapter does not provide a full step-by-step analysis of any one HDD but provides a flavour of the nature of the software and analysis presented at the business level.

This thesis has presented a number of observations which now need to be brought together into a coherent whole. Fundamentally, this is a Systems Engineering problem.

9.2 Cluster usage check

Although Windows is a file-base OS, a view of how files are grouped by the OS was considered necessary. Appendix F tabulates three tables of counts of the consecutive unused and used clusters. The second table is a full count, while the third table is the same with the extreme counts removed. All counts over 100 are grouped in "101" tally.

Increased counts may be seen around powers of two for used cluster lengths. No ideas could be developed at this stage with the thesis hypothesis. Use of a single cluster, 128 clusters from the end of the logical drive, was noted on at least three HDDs. It has been highlighted earlier in this thesis that at least two APTs place malware at the end of the HDD. Cluster run usage may a suitable topic for an academic paper.

9.3 Port Scanning

The port scanner was run on a personal Windows 10 machine and found at least three connections that were unknown to the author. Two connections were to two different data analytic companies and the third to business ISP - not this author's. Analysis showed that one of the data analytic companies had already been blocked using the Windows hosts file explicitly using the web address not IP address. This connection was by IP address and seemed to be from a block of IP addresses but not one directly linked to the website address. The same was true of the business ISP. A search of the C drive was run using most significant digits of an IP address as a search term on the business ISP. For example, if the block of IP addresses were in the range 127.0.1.0 to 127.0.1.256 then the search was for 127.1.1. This search yielded no results. It is possible that the program is installed somewhere other than the C drive or the IP address is encrypted or obfuscated. It has been shown that APTs perform such an action but not a business IP addresses. The use of blocks of IP addresses was true for the other two connections.

This demonstrates that simply blocking a web address is not enough to provide unwanted internet connectivity. None of the companies which perform these actions were, to the best of this author's knowledge and memory, given permission to place software on the author's machine.

9.4 Filename Extensions

It has previously been demonstrated that some malware makes use of the Windows OS LTRO filename extension parsing and uses deceptive extensions with more than two dots on them. A review of all filename extensions (316308) on a

Windows 10 HDD gave the following counts of dots in the filename extension from 0 to 12: 22607, 129903, 48607, 10585, 66617, 8512, 19787, 7161, 1333, 922, 241, 29, 4.

A review of one HDD of 191127 files gave 16746 different filename extensions made up of 12716 unique extensions and 4030 non-unique extensions. The filename extension list was reviewed by eye and several files would be worth further analysis either for their extension name or a combination of the extension name and number of occurrences.

Windows Files may be considered as one of four types: starting with "MZ" or not; suffix of .exe or not. Clearly, there are other ways to subdivide the file space.

Analysis of 442665 files on a Windows 8.1 machine gave:

	File does not begin "MZ"	**File begins "MZ"**
File Extension is not .exe	357657	77218
File Extension is .exe	898	6753

Table 9-1: File Extension Analysis

The range of file lengths was arbitrarily set to between 0 and 999,999,999. There were 139 files outside this range which accounts for the difference of 139 in the total for the counts, above, and the total number of files on the system.

An analysis of 896 .exe files on the Windows 8.1 OS which did not start "MZ" revealed eight files which started 0x00, 0x00, 0x01, 0x00; 34 files which started 0x01, 0x00, 0x00, 0x00; 451 files which started 0x44, 0x43, 0x44, 0x01 ("DCD"); 137 files which started 0x44, 0x43, 0x48, 0x01 ("DCH"); 266 files which started 0x44, 0x43, 0x4e, ox01 ("DCN). This analysis is on the same machine as the total count above, but at a slightly different time.

9.5 What do Different Filetypes Look Like Statistically?

The Index of Coincidence parameter from the categoriser highlighted ranges of values for different file types on a sample taken from all files from the program directories. These files are Visual Studio 2019 files and examples are:

File Extension	Minimum Index of Coincidence Observed	Maximum Index of Coincidence Observed
.bat	6.5	40
.cpp	7	10
.exe	19.5	150.6 (but there was a gap from 66 to 104.3)
.log	8	20.2
.sln	6.2	6.7
.tlog	65.6	69
.user (all 168 bytes long)	7.902	7.902

Table 9-2: Index of Coincidence of Selected Filetypes

Although there is overlap for the Index of Coincidence of the different file extensions it can be deduced that any file encrypted with a single byte will give the same value and hence a range of file extensions can be inferred. This, or a more powerful single value statistic, could be used to identify file types regardless of the extension provided. Should all .user files be the same then they can be uniquely identified.

9.6 Registry Data Extraction

It has previously been demonstrated (O'Murchu and Gutierrez, 2015, pp. 9, 17) that programs may reside (hide) in the registry and this is given the misnomer of fileless malware. A program was written to extract data from the Registry and put it into a “normal” file. This can then be analysed by the Categoriser.

On a Windows 8.1 development machine, a search for the 11 executable path extension plus .PS produced:

Registry Hive	Number of Registry Values	Number of Registry Values which could not be read	Number of Registry Values Extracted
HKEY_CLASSES_ROOT	195184	2	3894
HKEY_CURRENT_USER	93520	21	2137
HKEY_LOCAL_MACHINE	(1071431) 1071426	82	36926
HKEY_USERS	(423113) 423149	83	8332
HKEY_CURRENT_CONFIG	2	0	0

Table 9-3: Registry Analysis

The program was run twice (less than 24 hours apart): the first time did not count the number of registry values extracted; the second did after program modification. Figures for both runs matched except in the two parenthesised cells where the figures in parenthesis are for the first run. Not all data extracted is valid for further analysis. For example, “.COM” extracts “.Computer”, any website which has a domain name ending “.com”.

Although any registry value which contains an executable path extension (.COM; .EXE; .BAT; .CMD; .VBS; .VBE; .JS; .JSE; .WSF; .WSH; .MSC; which may be displayed by the command “echo %PATHEXT% as described earlier in this thesis) is of interest for further analysis, also of analysis is this registry value which could not be read as it is known that APTs are able to limit access.

9.7 Looking for the existence of Malware

Approximately 12 HDDs were analysed and every HDD contained at least one Indicator Of Compromise, some in the form of malware. Possible malware discovered is listed in Appendix D.

Some HDDs had sectors which could not be read (a possible indicator of malware hiding) while at least one Windows Registry file had keys which could not be read. Again, a possible indicator of malware existence.

Looking for key phrases and IP addresses respectively yielded results. There was evidence of malicious mutexes, keyloggers, banking harvesting details, email addresses and cryptocurrency mining. A review of IPv4 addresses on HDDs produced many false positives e.g. what appeared to be software version numbers but this set of putative data was reduced by a good old fashion human review followed by use of online tools to identify the registrants. This revealed IP addresses associated with possible malicious actors. Even the key phrase results were human reviewed.

Base64 encoding highlighted files but on further review they turned out to be .bat or PowerShell files which only used 64 characters. A search for repeated key encryption highlighted unencoded files with repeated characters.

Reviews of executable file length odd/even parity highlighted a small number of files, one in $RECYCLER, one claiming to be an AMD file – the only odd length AMD file on that HDD.

.exe files were also analysed using the Chi-Squared test for statistical roughness and by extension non-statistical roughness. This is to test for a more powerful encryption scheme than XOR with repeated key. The hypothesis here is that should an APT used an encryption scheme which produces output with no statistical features then that is worthy of further analysis. Alternatively, a file which has statistical roughness then that is also worthy of further analysis.

The use of subroutines in executables was analysed. The hypothesis is that APTs will use subroutines that will have a low probability of normally being observed e.g. Zw routines, DeviceIOControl and so an executable using a number of these low probability subroutines may be malware. A probability table of observed subroutines from executables in the Windows OS was built. This was then used as the base distribution to analyse subroutines use in other executables. For ease of programming all subroutines names were hashed using a 32-bit CRC (Kientzle, 1995, pp. 276-277). As storage is a premium the least significant 16 bits were used as the unique hash. Unfortunately, this led to clashes and so the least significant 20 bits were used. Also extracting the names or subroutines was problematic as many different formats were used. However, notwithstanding the outliers from format issues t can be seen that this line of research of worthy of further work.

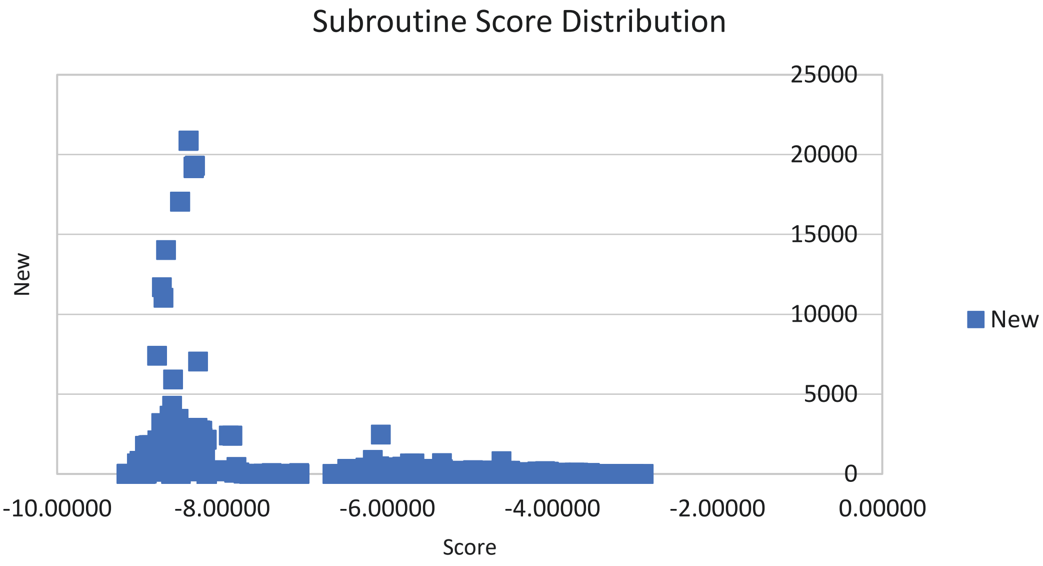


Figure 9-1: Subroutine Score Distribution

The use of subroutines in executables was analysed. The hypothesis is that subroutines which are used in malware are rarely used subroutines. This work proved to most intricate than first imagined. There are subroutines dependencies - there is not an issue with routines calling routines for the data for which is being searched (the .exe is split into sections and there may be calling of routines across sections but the search should be looking at within section). However, some routines are related; For example, for every fopen there should be an fclose and also an fread or fwrite. On the data analysed there is some mismatch of the malloc/free pair. This may be the program not working properly and hence not picking it up correctly. It may also be an indication of poor programming by the developer(s).

The choice of logs is based on the reasoning that if there are, say, a corpus of 100 subroutines with the frequencies: sub(a) 50, sub(b) 10, sub(c) 10, subs(all the rest - 30) 1, Total 100, then seeing 10 out of the rest may be more significant than just seeing sub(b). Adding the frequencies both give 10 and a reasoned choice cannot be made between them.

The null hypothesis corpus comes from a Window system that has not been used. Any hard drives presented (i.e. those which may be infected with malware) may be tested against this corpus.

The outcome, looking at the results by eye, both the "pristine data" and the "possibly infected sample" looked similar. What was highlighted, as mentioned above, was mismatched subroutine calls (malloc/free) and routines than in the "infected" sample not being in the "pristine" data. I have more work to do on these results, as my analysis has not been scientific.

This approach was not complete. Analysis by eye of PE files lead to a program to extract the dlls and create the counts for the above test. Every PE file had the same format i.e. used the dll GetCurrentProcess, and it was just a case of finding this dll and extracting the dlls close to it for the counts. When the counts software was developed it was clear that the counts were wrong. Further analysis of more PE files showed there that there was another format. The counts software was amended but still the problem persisted. Further analysis of PE files using the source code of a PE viewer (LaserMedia-AS, 2007) showed that there were many ways to parse the subroutine list (LaserMedia-AS, 2020). More ways that could be accommodated in the time left available for the development of this thesis. This idea could also be modified to build a set of counts of dlls used in malware samples (GitHub, 2020) and use those counts to statistically identify malware with the analyser program. This is a subject to which the author will return in the future.

9.8 Does the Microsoft Supply Chain Use a Different Compiler to the Rest of the World?

9.8.1 Background

Any file on a computer has a number of attributes, one of which is the file length. For Portable Executables (PEs) the file length is determined by the compiler. The outcome of the research described here, supported by statistical techniques described later in this chapter, finds that file lengths in Microsoft operating systems are overwhelmingly of even length. An odd length PE, therefore, is an indicator of not being written in the Microsoft supply chain and hence may be an IOC for malware.

It is possible that at least one compiler which Microsoft uses in its Original Equipment Manufacturer (OEM) supply chain for the compilation of constituent parts of the Windows operating system is different to other compilers which are available to the IT profession at large. This difference is that the Microsoft OEM will produce executables of even length whereas other executables may be odd or even length. It is noted that Microsoft reserves at least one field in the PE format for Borland (Microsoft, 2018r). The analysis was carried out using freely available Microsoft and OpenOffice software.

The format and attributes of PEs are noted and this thesis denotes them as "Attributes by Design" i.e. any attribute which is known to the designer. An attribute which is unknown to the designer and has been determined by analysis this thesis denotes as "Attribute by Discovery". For this particular piece of analysis, an Attribute by Discovery is the parity of the file length – is the file length odd or even – and questions what this means in aggregate across certain file types.

This thesis proposes a technique for analysing the distribution of file sizes on Windows based machines. It postulates that Microsoft, and its OEM supply chain, use at least one family of compilers for the compilation of constituent parts of the Windows operating system. It also postulates that this family of compilers is different to other compilers which are available to the IT profession at large. This difference is that the Microsoft OEM will produce executables of even length whereas other executables may be odd or even length. The analysis was carried out using non-bespoke software i.e. using available Microsoft software.

9.8.2 Most and Least Significant Digit Analysis

Most significant digit analysis consists of analysing the first digit of each number in a set of numbers. For example, if the number is 12345 then the digit “1” is selected for analysis. Least significant digit analysis consists of analysing the last digit of each number in a set of numbers. For example, if the number is 12345 then the digit “5” is selected for analysis. This would, in effect, look at the parity of the file size - is it odd or is it even? Clearly the expected result of a random set of digits is that there should be an equal number of odd and even numbers.

9.8.3 Data Collection and Analysis

Two ways of collecting the data were used: the first used the Microsoft command “forfiles” on Windows 7, 8 and 10 operating systems; the second used the “dir” commend on Windows XP. This is because “forfiles” was not available on the XP machine and it was decided not to download it. Also, the different ways of data collection provide some independent assurance that the results are valid.

I. File Size Last Digit Analysis (to test odd/even parity of the file size)

	Exe C:\Windows	Exe All Other	Dll C:\ Windows	Dll All Other
XP	135 of 2690	10 of 7945	148 of 5760	343 of 5445
7	1 of 4134	19 of 1327	14 of 36416	348 of 14614
8	377 of 3422	10 of 9033	2966 of 26053	10 of 12852
10	0 of 2138	0 of 39	6 of 14123	0 of 235

Table 9-4: Count of Odd length files

A one-line batch was written which contained the command “forfiles”. This command “Selects and executes a command on a file or set of files. This command is useful for batch processing.” (Microsoft, 2018l). This batch recursively outputs, one item per line, a list of all .exe files with associated size and full path name. The batch was modified for .dll in place of .exe and was run on Windows 7, 8 and 10. The command used within the batch is:

```
forfiles /p c:\ /s /m *.exe /c "cmd /c echo @fsize @path"
```

and, of course, “*.exe” was changed to “*.dll” for dll files. The output was written to respective text files and then imported into separate spreadsheets.

Another one-line batch was written which contained the line:

```
Dir C:\ /S
```

(Microsoft, 2017h)

Again, the output was written to text files and then imported into a spreadsheet. However, in this case a lot of hand editing and sorting was needed to produce data in the required format. The filetype extension was extracted using the formula:

```
=MID(cell containing file name],LEN(cell containing file names])-2,3)
```

The full path name was not retained. As well as extracting only .exe or .dll files the data was split between C:\Windows and all other subdirectories by hand for separate analysis. Within the various worksheets the first digit of each file size was extracted using the formula:

```
=INT([cell containing file size]/POWER(10,INT(LOG([cell containing file size],10))))
```

and the last digit of each file size was extracted using the formula:

```
=MOD[cell containing file size],10)
```

Each set of data was collated using the Pivot Table function and statistically tested using the relevant Chi-Squared test for the spreadsheet type. Spreadsheets used were Microsoft Excel and the OpenOffice equivalent. This author had admin or user rights for each machine. This meant that some directories were not available for analysis.

9.8.4 Most Significant Digit Analysis (Benford's Law)

Benford's Law has previously been discussed in this thesis.

	Exe C:\Windows	Exe All Other	Dll C:\ Windows	Dll All Other
XP	4.25E-05	0,00E-00	0.00	0.00
7	3.3E-24	3.1E-09	9.5E-306	2.1E-265
8	1.3E-06	0.00E-00	2.36E-089	4.18E-253
10	7.4E-07	Insufficient Data	3.7E-69	0.6

Table 9-5: First Digit Statistics

II. File Size Last Digit Analysis (to test odd/even parity of the file size)

	Exe C:\Windows	Exe All Other	Dll C:\ Windows	Dll All Other
XP	135 of 2690	10 of 7945	148 of 5760	343 of 5445
7	1 of 4134	19 of 1327	14 of 36416	348 of 14614
8	377 of 3422	10 of 9033	2966 of 26053	10 of 12852
10	0 of 2138	0 of 39	6 of 14123	0 of 235

Table 9-6: Count of Odd length files

It can be seen that there is a dearth of odd length files.

During this analysis this author purchased a new Windows 10 machine: Literally "out of the box" before doing anything else (including connecting to the internet) the methodology was performed. The .exe counts were:

Digit	0	1	2	3	4	5	6	7	8	9
Count	609	5	660	11	651	8	659	9	654	16

Table 9-7: Least Significant Digit Count

(New Windows 10 Machine - .exe.Files.
Count of Odd length files)

Which re-enforces the previous observations.

9.8.5 Discussion

The lack of adherence to Benford's Law is unexplained however it is aligned to the work of Evans and Kuenning who " … found that popular distributions fit poorly in terms of statistical reliability, and that anomalous spikes in distributions are the norm." (Downey, 2001). It is clear from the odd/even analysis of data that there is a significant split between the counts of odd and even file sizes across Windows operating systems.

Microsoft describe the format of PE files as follows. The Sizeof image field " ... must be a multiple of SectionAlignment " which in turn "… must be greater than or equal to FileAlignment. The default is the page size for the architecture." (Pietrek, 1994). It appears, therefore, that file sizes are based on multiples of the field FileAlignment which "… should be a power of two between 512 and 64 K, inclusive. The default is 512." One can immediately see that this will produce an even length file size. The key word here may be "should be a power" but when SectionAlignment is greater than FileAlignment, and an odd number, the file size could end up as an odd number.

Information from Microsoft is mixed: although Pietrek does not shed light on a PE file size, Pietrek (Pietrek, 2002) alludes to platform differences “You should only need to use the 32 or 64-bit specific versions of the structures if you're working with a PE file with size characteristics that are different from those of the platform you're compiling for.”. Pietrek (Pietrek, 2002) states that “The distinction between *exe* and *dll* files is entirely one of semantics. They both use the exact same PE format. The only difference is a single bit that indicates if the file should be treated as an *exe* or as a *dll*.”.

Further analysis on the machine from which the data was harvested reveals that many of the odd length files have names that contain x86 (e.g. C:\Program Files (x86)), AMD or “uninstaller”. Some are from the SysWOW64 Windows subdirectory. Why would this be the case? One inference which can be drawn from this that they could have be written by another set of developers using a different compiler.

The Windows XP and Windows 8 machines were this author’s personal machines, they had been in use for a long time and had become “tainted” by more recent files. They also contained duplicates of files and files of similar name and identical length. However, if there were a non-biased odd/even split of file lengths this should be reflected in the data. There is still a distinct odd/even split in file lengths. This observation is consistent with the work of Evans and Kuenning:

“… file-size distributions are “polluted” by large collections of similarly sized files, such as icons or configuration files associated with a particular application.”

(Evans and Kuenning, 2002)

However, this contradicts Downey’s work (Downey, 2001).

It appears that the compiler Microsoft is using in its OEM supply chain compiles code to multiples of two for the file size (i.e. even length executables) but at least one other compiler, Visual Studio 2013, compiles code to both odd and even length.

No previous analysis of *exe* and *dll* files could be found. Assurance for the work in this thesis is gained by the methodology. Using two different ways of collecting the data (*dir* and *forfiles*) means that it would be difficult to make the same mistake across all operating systems. Similarly, using two different spreadsheets (Microsoft Excel and OpenOffice) across operating systems provides independence: The Chi-Squared function calls are different in each. Digit extraction from the file length field is easily checked across the data sets.

It is concluded that using two different sets of analysis software (Microsoft Excel and OpenOffice) provides an independent check within the analysis of the methodology. The only commonality across both sets of analysis is the Microsoft program *forfiles*.

There is strong evidence that the compiled length of files in Microsoft Windows operating systems is restricted to an even length by at least one complier used by Microsoft in their supply chain when creating *.exe* and *.dll* files. It is possible that all compilers used by Microsoft have this feature. This feature may, or may not, be present in different compilers used by the wider IT profession but it has been shown that at least one these compilers will compile *.exe* and *.dll* files to an odd length as demonstrated above.

This analysis leads this author to assert that although an even length file size does not rule out infection by malware (the software could have been written and compiled by Microsoft OEM or a non-Microsoft OEM developer), an odd length file size may indicate malware i.e. The PE has been produced by an entity other than Microsoft and members of their OEM supply chain. This feature may be used as part of a wider indicator of compromise for APTs.

9.9 Mechanisms of Action

9.9.1 Getting Malware to run on victims' machines

Malware is placed on machines and persists in various ways. To recap from earlier in the thesis:

Most organisations are breached, mainly by spear-phishing (directly through email attachments or linking to a website with malware. Most malware is, at least, added by start-up folders or registry keys. Simply checking the latter two at logoff or having a “deep freeze” system which cannot be modified will reduce the probability of success of attack.

A program could be developed to hash every file in a pristine Windows system. Alternatively the hashes of software are publicly available (NIST, 2019), (GetData-Forensics, 2020).

9.9.2 Mechanism of Action: An Examination of Registry Keys

The use of the \Run and \RunOnce Registry Keys is public knowledge. However, there may be other Registry keys which act in a similar manner but which are not publicly known. It was hypothesized that there must be a list of all such Registry keys somewhere on the system.

A suite of programs was written to add a new subkey to every Registry Key. This suite, although tested on a sub-set of Registry Keys, when run on all Registry Keys crashed the machine which then ha to be completely re-built. Fortunately backups had been made as discussed earlier.

Program search was then written to search for a given string within a set of HDD sectors. The maximum number of sectors was taken from the last sector used in the MFT as provided by the program filelist. search was run twice: once looking for "\Run"; and again, looking for "\Run" with each character separated by once character. This was to look for Unicode use of the string. This gave over 150,000 lines and 500,000 lines of output respectively.

A program to identify how programs are called, and from where they are called e.g. Registry /RUN keys, Windows Services, calls from other programs.

9.9.3 Background

Malware, or a computer virus, does not run of its own accord; there has to be some a process to trigger, or initiate, it.

In IT, the term “Virus” is taken from the medical world. This thesis builds on this parallel. In a similar way that malware needs to be able to initiated, run and have an effect on a computer, in the medical world there is a definition for drugs which affect cells: This is called a “Mechanism of Action”:

> “The means by which a drug exerts its effects on cells or tissues”

(Farlex, 2018)

There is a Mechanism of Action which exerts its effect on malware so that this malware may, by extension, exert it its influence on the victim’s computer. This mechanism of action is not always well-known.

Although not well known within the IT world, “Mechanism of Action” it is a term used before by Lynch to describe Hard Disk Drive (HDD) arm movements (Lynch, 1972).

As there is a Mechanism of Action there must, by analogy, be one of inaction. This thesis calls this an Inhibitor of Action.

The work described in this thesis was performed on the HKEY_CURRENT_USER (HKCU) and HKEY_LOCAL_MACHINE (HKLM) registry hives on a Windows 8.1 machine.

9.9.4 Where are the Mechanisms of Action?

Nothing happens without power and activating the On switch starts the boot process. From this all other actions flow. The On switch may be labelled the primary mechanism of action and all others, secondary, tertiary etc. but for the purposes of this thesis all mechanisms of action are equally labelled. The chain of events from switching on the computer through to initial use this thesis calls “The “Mechanism of Action Cascade” and note that an attacker could place malware anywhere within this cascade.

At the end of various branches of this cascade are a number of malware mechanisms of action, which are the illegitimate use of legitimate features of the operation system.

Which way to follow the data? One could start with the On switch and follow the boot process or one could start where one believes the last mechanism of action to be and work back. This thesis has already seen that these mechanisms of actions may include the Windows Service, start-up folder and Registry Keys. In many cases these Mechanisms of Action are persistent i.e. they persist across logon and boot sessions. These shall be explored the latter in this thesis.

A Mechanism of Action is not restricted to making use of the actions of the operating system. This thesis has previously shown (Villeneuve, 2011, p. 13) that in many cases, the persistence mechanism will consist of methods such as adding the malware executable to the windows “start-up” folder, modifying the Run keys in the Windows Registry or installing an application as a Windows Service etc.

Yet Kaspersky have seen browsers accounting for 42% of vulnerable applications (Kaspersky, 2014d, pp. 24-25). These two figures (97% and 42% provided by Mandiant and Kaspersky respectively) may be consistent: they are measures taken at different times and different anti-malware companies may have different talents, and core skills.

Each of these forms of persistence has to start somehow. To include start-up information in the Windows Registry is fine but there must be a mechanism of action to run the contents of these keys. It is noted that there are other mechanisms of action but for the purposes of this thesis there is focus on registry keys.

9.9.5 Looking for the Mechanism of Action in the Windows Registry

This research took two routes: the first was to add values to every subkey in the registry to emulate the \Run and \RunOnce Registry Keys which are public knowledge:

“Run and RunOnce registry keys cause programs to run each time that a user logs on.” (Microsoft, 2017ab)

There are at least four registry keys that are mechanisms of action for software and it is hypothesised here, and others (Skoudis and Zeltser, 2004), that there may be more Registry keys which act in a similar manner but which are not publicly known. It was hypothesised that there must be a list of all such Registry keys somewhere on the system but where might this list be stored? The list could be integral to the Registry or in a file external to the Registry. It is important to understand which registry keys can be used as they can also be used as mechanisms of action for malware (Symantec-Security-Response, 2013, p. 14), (Gross and Cylance-Spear-Team, 2016, p. 13). This list could also contain lesser known Registry keys performing the same action and it might be possible to add to this list.

The hypothesis is that this Mechanism of Action is in the machine boot or Windows start-up procedures. The order of loading software and actioning registry keys is important.

"Because the HKEY_LOCAL_MACHINE\...\RunOnce key is loaded synchronously, all of its entries must finish loading before the HKEY_LOCAL_MACHINE\...\Run, HKEY_CURRENT_USER\...\Run, HKEY_CURRENT_USER\...\RunOnce, and Startup Folder entries can be loaded." (Microsoft, 2018p).

One might infer that the list of registry keys to be actioned at start-up or logon would be listed in in the hive of HKLM\SYSTEM\CurrentControlSet\Control Registry Key (Microsoft, 2017o) but a review of this sub-tree of Registry using regedit (Microsoft, 2012) suggests this does not appear to be the case.

The second route was to look for a list of keys on the HDD which contained the \Run and \RunOnce Registry Keys. This list should also contain other registry keys that perform a similar function.

9.9.6 The Software Written – Approach 1

The first approach was to test if live Registry keys were used in the manner of the \Run and \RunOnce Registry Keys. It is acknowledged that other registry keys not currently used and, which did not exist on the machine, could be available but there are infinite number of these and so this line of research was not pursued.

A program was written to add a new subkey to every Registry Key on a Window 8.1 system. This subkey contained a value that would action a program at login if that key were an unknown registry mechanism of action. Although tested on a sub-set of HKCU Registry Keys, when it was run on all HKCU Registry Keys it crashed the machine. The machine then had to be completely re-built.

Rethinking the implementation produced a reduced set of software. Two programs were written the result of which, at login, if a registry key were to be used in a similar manner to the /Run keys then this would be logged. A third program reversed the action of the first two and deleted all of the values that were produced.

The successful process was applied to the HKLM. This again crashed the machine but this time recovery was from a checkpoint.

9.9.7 What was Found - 1

When implemented for HKCU, the only registry keys which were actioned at login were the known \Run and \RunOnce Registry Keys.

It was not possible to successfully to do this HKLM due to the system crash.

9.9.8 The Software Written – Approach 2

The second approach was to try to find an area on the HDD that held a list of registry keys to be actioned at start-up or login. It was hypothesised this list of registry keys would contain the \Run and \RunOnce registry keys and hence other registry keys which would perform the same action at start-up.

A program (*filelist*), which had been previously written by this author for other research was used. This program lists, from information in the Master File Table ($MFT), all sectors where files are held. This program has was inspired by Kusano's (kusano, 2015) but *filelist* includes the sector number where the data resides as well as filenames and status as known to the operating system. Although the programs are functionally similar, the code for *filelist* is completely different having been researched from various sources (Sammes and Jenkinson, 2007, pp. 215-275, 389-410), (Richard Russon, 2018), (Russon and Fledel, Undated), (Wilkinson, 2017). However, the output of both programs was assured against each other.

A Program (*search*) was written to search for a given string within a range of HDD sectors and output a hex print of the containing sector. This set of sectors was taken from the program *filelist* and, in effect, listed all active and deleted files. Large files were only viewed to, at most, the first 1024 characters (two 512-byte sectors). It was reasoned that any file containing the desired list of registry keys would be relatively short. *search* was run looking for "\Run". This gave over 150,000 lines of output which were scanned by eye.

9.9.9 What was Found - 2

A number of registry keys with “\Run” in the key name were found and reviewed.

Recall that the hypothesis was that the “\Run” registry keys need a mechanism of action. It was further hypothesised that the “\Run” keys would be grouped together and that malware writers could insert additional registry keys into this file. These extra registry keys could contain links to malware. In one instance the “\Run” registry keys were found in a group in one file but there were no additional registry keys. This is not surprising, as for there to be other registry keys the machine would have to be infected with malware.

A powershell script file was discovered containing a clear indication of the Registry keys to be run. However, these registry keys were not in a specific list as hypothesised but hardwired into conditional statements. The registry keys were “\SOFTWARE\Microsoft\Windows\CurrentVersion\Run” and “\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run" both for HKCU and HKLM. The omission of \RunOnce is not surprising. As has already seen the order of loading matters (Microsoft, 2018p).

Other registry keys and fragments of keys were found. A batch was written to add registry key values for all 14 new keys that were found. As some registry keys were incomplete, there was extrapolation to the full key name. Of these registry key values, only the HKLM Wow6432Node key ran at start-up. Microsoft state that

"You can safely ignore the registry value" (Microsoft, 2011). However, the wider forensics community is aware of the use of this key value as a mechanism of action (Karp, 2010).

The file containing the group of four registry keys was analysed, as were the surrounding sectors on the HDD. This file was part of a set of similar files stored within just less than 500 512-bytes sectors. Each file started on a cluster boundary. No file contained obvious, readable, identifying data such as filename, checksum, size etc. At the end of each file the sector was padded and the rest of the cluster padded with apparent random data, which was not analysed. This padding allowed the next file to start on a cluster boundary. If a file contained enough data to extend into the 8th sector of a cluster then this sector was padded and a full cluster of apparent random data was inserted before the next file.

As all files were over 1024 bytes in length it would appear that they are $MFT non-resident files (Sammes and Jenkinson, 2007, pp. 215-275, 389-410), (Richard Russon, 2018), (Russon and Fledel, Undated), (Wilkinson, 2017).

9.9.10 An Unexpected Discovery

An unexpected outcome of inserting HKCU registry values was that after running, at login, the first set of software (the registry key value insertion program), Visual Studio 2013 did not run. Once the inserted registry values were deleted it was possible to run Visual Studio 2013 again. The same effect was seen for HKLM registry key values.

It is inferred that Visual Studio 2013 has a built-in integrity check based on the Windows Registry to inhibit tampering. i.e. Visual Studio runs if and only if a given set of registry keys and/or values exist. It is further inferred that other software may have this capability. This thesis calls this an "Inhibitor of Action" and it is something that malware authors could use to disrupt a victim's machine as a denial of service attack

In a similar way to the effect the changes had on Visual Studio 2013 it is further inferred that modifications to HKLM hive registry keys and/or values act as an inhibitor of action and that at least one version of Windows will not run if there has been at least a certain, but unknown to this author, HKLM modification.

9.9.11 Discussion and Way Forward

Given the different outcomes and of registry changes to HKLM and HKLU hives this thesis speculates that they are different teams within Microsoft and that one has better software quality control. After all, the operating system is just another program and all programs should terminate gracefully. There may also be an issue of inhibitors of action on programs. Surely a program should only action legitimate parameters and report anything unexpected?

The next stage should be to identify the name of the file containing the registry keys, the names of all files within the associated 500-sector area on HDD and to identify the mechanism of action for all of them, all the way up the branch of this part of the mechanism of action cascade. At each node in the cascade there can be a check for other mechanisms of action and progress down that part of the cascade, repeating for all nodes until all Windows mechanisms of action are found. This is a similar to walk a directory tree.

It is acknowledged that a scan of data in active clusters could have been performed but, again, this would probably produce a lot of output. This could have been done programmatically using DeviceIoControl (Microsoft, 2018k) with the FSCTL_GET_VOLUME_BITMAP control code (Microsoft, Unknown) or directly through the MFT .

It is thought that the reason that the first approach to registry key modification failed was that the registry key name being stored as a parameter value was the full registry key name and that, in some cases, this was too long for a registry key value. The shorter node registry name worked so the hypothesis was not followed up as it did not add value to the thesis.

This work has re-introduced the term “Mechanism of Action” to the IT industry and developed this into the “Mechanism of Action Cascade”. The work has also developed the concept of “Inhibitor of Action”. It has been highlighted that an attacker could place malware anywhere within the Mechanism of Action Cascade and this would be something against which it would be more difficult to defend.

It has been demonstrated that there is mechanism of action for registry keys and that there is at least one lesser-publicised Microsoft registry key (but known to the wider forensic community), Wow6432Node, which acts as such at login. Arguably, the forensics community highlight the use by attackers of HKLM and HKCU \Run registry keys but there is less publicity surround the Wow6432Node \Run key. System Administrators should be aware of this registry key.

It has also demonstrated that at least one version of Windows and one piece of software, Visual Studio 2013, has a registry-based inhibitor of action. This inhibitor of action consists of simply adding a new registry value, which the software does not recognise. It would be possible to develop this inhibitor of action as a denial of service attack.

9.10 Analysis of a Malware Corpus

The software package was brought to bear on a corpus of 228 files known to be associated with malware:

Test	Number of Files Identified
Zero Length	16
Existence of GetCurrentProcess	134
Base64	1
Odd Length	46
Length [0, 4096] i.e. short files	3
Possible Encryption	22

Table 9-8: Count of Malware Identified from Software Developed

and a score associated with each test.

Score	Number	Percentage
No Score	79	34.65%
Low Score	87	38.16%
Good Score	62	27.19%

Table 9-9: Summary of Malware Identified from Software Developed

i.e. almost two thirds of the files had some score attached to them designating them as worthy of further analysis with just over a quarter being of higher interest. At this stage one has to consider that these files are known to be associated with malware and there may be unconscious fitting of the score to the data by the author. However, it remains that the existence of possible malware of has identified using software developed from reading and hypothesis development. At the very least this may be considered a success.

9.11 Differences in Compilers

The simple "hello world" program (Kernighan and Ritchie, 1988, p. 6) was compiled using three different compilers: Tiny (Bellard, 2018), Microsoft Visual Studio 2019 (Microsoft, 2019d) and Pelles C (smorgasbort, 2003-2020). The latter two in Debug and Release modes for x86 and x64. This gave seven different executables for analysis.

Executable Origin	Size in Bytes
Tiny Compiler (hellowt.exe)	2,048
Microsoft Visual Studio Debug x64 (hellowmdx64.exe)	59,904
Microsoft Visual Studio Debug x86	38,912

(hellowmdx86.exe)	
Microsoft Visual Studio Release x64 (hellowmrx64.exe)	10,752
Microsoft Visual Studio Release x86 (hellowmrx86.exe)	9,216
Pelles C Debug (hellowpd.exe)	53,248
Pelles C Release (hellowpr.exe)	39,424

Table 9-10: Summary of Executable Sizes with Source Compiled with Different Compilers

The size of the Tiny compiler PE adds weight to the argument that APTs are able to produce malware of a small size.

Additionally, the program was compiled on different computers with Visual Studio 2019, at least twice on each. The Visual Studio PEs were then analyzed using the hexprint program a PE viewer (LaserMedia-AS, 2007). The PEs appear to contain machine specific, or Visual Studio 2019 specific, information in bytes 128 to approximately 239 and 4608 to 4879.

9.12 Putting it all Together

It is now possible to put this all together into software that will increase the business costs of APTs. This is over and above hashing techniques to validate files which claim to be legitimate Windows files. IT is also in addition to that which an organisation or individual would implement as part of their intelligence, threat driven mitigations. The Anti-APT Structure Data Flow is provided in Figure 6-1.

For analysis, any program should be written as close to the hardware and operating system as possible. This means minimising Windows specific routines and using, for example ANSI C. POSIX C is also acceptable.

An overwrite program to overwrite unused areas of disc based on cluster usage can now be written as well as a file-based analysis program. The use of both programs on the computer would be obfuscated using APT obfuscation techniques.

The overwrite program would be in two parts: analysis of unused clusters; analysis of used clusters. For unused cluster analysis, using the cluster table ($CLUS) obtained from the MFT, any unused cluster can be overwritten with binary zeros. However before overwriting the data can first be analysed for IOC of malware. These IOCs may be use mutexes, IPv4 and IPv6 address and keywords. The statistical properties of the cluster may be analysed at the binary and character level. Data which is too flat or rough can be highlighted. Any results may be sent to a central corporate server. Similarly, any clusters which are partially used may have the unused part (i.e. after a xFFFFFFFF terminator) overwritten, after analysis.

The analysis program would run perform statistical analysis on files based on the filetype. It would include, but not be restricted to, base64, Index of Coincidence, file length, encryption by repeated key, chi-squared (is it too rough or too smooth?), again sending results to a central corporate server.

Any compiler used to compiler these programs should not draw attention to itself and the paradox is that these programs should not highlight themselves on the systems. Should they be able to do that then APTs may copy the technique. However, if they can highlight themselves then an APT may be able to become aware of their existence as an AV feature and react accordingly.

9.13 Conclusion

This chapter has presented the analysis of HDDs on a variety of machine sand standalone HDDs. It supports the fourth Aim and Objective. The existence of malware has been identified using software developed for this thesis. Dubious IP addresses and port usage can be identified as have the existence of mutexes, key loggers crypto-currency mining. A corpus of files known to be associated with malware has identified at least a quarter of the files as malware related and as much as two thirds worthy of further analysis.

This chapter has demonstrated that:

- The software developed from analysis of academic papers and White papers is able to identify potential malware on HDDs;
- Microsoft and its legitimate supply chain may use a different compiler to that available to developers and the parity of .exe and .dll files can be used a malware IOC;
- By using the mechanism of action flow, it is possible to hide calls to malware earlier on in the boot process in the OS;
- It is possible to apply a denial of service attack by manipulating keys in the registry
- treat APTs on Windows based IT systems in such a way that the malware is negated;

This chapter has fulfilled the criteria of the third Aim and Objective: treat APTs on Windows based IT systems in such a way that the malware is negated;

10 CONCLUSION

10.1 Chapter Overview

This concluding chapter starts with a gentle critique of the thesis. It then follows with a broad discussion on the future including of the Internet of Things (IOT.

10.2 Fulfilling the Research Objectives

Recall that the research objectives are to:

- discover where APTs lie on windows-based systems;
- create/build generic views of where APTs lie on Windows based systems using an appropriate Cyber Kill Chain (CKC);
- treat APTs on Windows based IT systems.in such a way that the malware is negated;
- increase the cost of doing business for APTs, for example by overwriting unused disc space so APTs cannot place malware there.

10.2.1 Fulfilling the First Research Objective

This thesis has been produced following reviews of a range of academic papers and white papers produced by AV companies. It has also reviewed a large number of other sources including webpages and podcasts. From these sources has been collated a time bounded view (mainly the 2010s) on the work of APTs and malware placed on Windows based operating systems. This has been limited to Windows 7 and above.

10.2.2 Fulfilling the Second Research Objective

The thesis has been agnostic towards the origin and intent of APTs. To distil the information into a manageable form, the Lockheed Martin Cyber Kill Chain was selected from a number of candidate Cyber Kill Chains. This allowed the author to have a view of malware on Windows systems, however this view was sometimes blurred by the used of the same malware by different APTs and analysis by different AV companies of the same APTs. It is possible that apparently independent evidence from ostensibly different attacks is the same hence giving a false sense of evidence re-enforcement.

10.2.3 Fulfilling the Third Research Objective

The research used the file-based concept promulgated by Microsoft and developed the concept of Attributes by Design and Attributes by Discovery. This philosophical view helped to guide the theory and software development. The concept of Mechanisms of Action was borrowed from the pharmaceutical industry. This is a non-random connection as it builds on the concept of a computer virus. It also allows us to introduce the idea that malware does not have to be found, it merely has to have its mechanism of action severed. This is one of the inexpensive ways for a defender to increase the business costs of an attacker.

The research introduced the idea of benware (Benign Software) and developed a proof of concept attack based on the attacks of two APTs. This attack used the concept of Living Off the Land. Although not a new concept it demonstrated the ease with which malware may be written using only legitimate Microsoft software and how this may be modified for each attack to defeat AV software. It also demonstrates the idea that an attacker could, and should, use the least expensive and easiest attack to gain access and persistence on a victim's system.

10.2.4 Fulfilling the Fourth Research Objective

The research has produced a business view of the organisation of an APT with associated costs. It demonstrated the difficulty of providing a single monetary cost but did offer a high-level view for consideration. These costs were then used to justify the business techniques and software produced to mitigate the observed attacks.

10.3 A Critique of this Work and Future Research Direction

The scope of the thesis is an HDD. To provide produce proof of concept software the HDD was viewed as a single partition and file system on a single HDD. It is acknowledged that there may be multiple partitions on one HDD and that file systems may span more than one HDD q.v. RAID arrays. To keep the proof of concept software simple, further assumptions included the use of 8-bit ASCII (i.e. not UNICODE), 512 bytes to a sector and eight sectors to a cluster. It was fortunate that the MFTs of all HDDs analysed were formatted as such, although there was evidence of UNICODE. The HDD free space overwriting program did not explore all free space (e.g. within sector and cluster segments following the end of file marker but the way forward was pointed to. It is noted that such an overwriting program will destroy any evidence which may be a problem for computer forensics and law enforcement. Any development of this avenue should consider evidence preservation e.g. keep evidence of IP addresses accessed, mutexes used etc. Such an approach would need legal advice and guidance.

All programs are research, proof of concept programs and not all have been fully optimised. For example, the search programs read one sector at a time. This is less efficient than reading clusters or multiple clusters that make optimal use of the available hardware configuration. Directory walk software at the C level was used simply for programming ease. These should be changed to directly access the MFT. The driver to write directly to free areas of HDD would need to be called by a program passing to it the number of the sector to be overwritten and how (either the whole sector or after the end of file marker). This pair of programs would need to work with the Windows operating system to ensure that there are no sector access collisions.

Any implementation of the ideas provided in this thesis may need recoding from scratch.

No previous analysis of executable and .dll files could be found and this, together with the results raise the question: Why has this area of IT Security not been addressed?

The scope of this thesis is that of malware on the HDD platters and excludes HDD firmware. Should an attack modify firmware in the way described earlier in this thesis, then this thesis is nugatory, even though malware is on the platter. This is an avenue of future research.

Further research should be an analysis of the executable of a single program compiled on different machine with different compilers. It is possible that such executable may have unique compilers and machine "fingerprints", the existence of which may be incorporated into the categorisor and analyser program developed for this thesis.

Finally the subroutine program should be modified and completed to be able to identify little used Microsoft subroutines which may be Indicators of Compromise.

A further avenue of research is the Internet of Things (IoT). A world where all electronic devices are interconnected may provide security challenges. Such network-attached storage devices may include smart TVs, cars, refrigerators It is possible that not all devices will include automated update checks requiring consumers to download and install new firmware may be one task too many

Botnets may also make use of the IoT and with every device having an IP address this may get worse in the near-term.

This thesis has demonstrated that malware, or at least Indicators of Compromise, may be found by analysing the Windows OS at the file level. On eight HDDs provided by an organisation, all were found to have Indicators of Compromise. On a malware corpus of 228 files know to be associated with malware two thirds were identified. This is comparable with commercial AV companies. One does not have to find all malware just what is necessary and sufficient to meet the victim's risk management threshold. Even finding 1 Indicator of Compromise may be enough to satisfy the machine's owner.

An overwrite of the whole HDD and reload of the OS is possibly enough to satisfy most victims. A pristine HDD with OS reload should satisfy most of the rest.

HDDs are relatively inexpensive now that this may the most cost-effective solution. Any HDDs may be destroyed or analysed for the greater good.

Finally, the philosophical view, concepts and ideas in this thesis are fully transferrable to other operating systems.

10.4 Concluding Remarks

This thesis has produced a comprehensive white paper and academic discussion of where APTs lie on windows-based systems and used this create and build generic views of where APTs lie on Windows based systems using the Lockheed Martin Cyber Kill Chain. It has satisfied the thesis Aims and Objectives. It has produced a suite of software and suggested lines of further research to treat APTs on Windows based IT systems in such a way that the malware deployed is negated. This has been done using a file-based analysis including Attributes by Design and Attributes by Discovery. Finally, it has increased the cost of doing business for APTs, for example by overwriting unused disc space so APTs cannot place malware there and by severing the malware mechanisms of action.

REFERENCES

25010:2011, I. I. (2011) *Systems and Software Engineering -- Systems and Software Quality Requirements and Evaluation (Square) -- System and Software Quality Mo.* ISO.

ABC. (2017) *3.0 Phoenix + Electron*. Available at: http://www.abc.net.au/radionational/programs/sum-of-all-parts/3.0-phoenix-+-electron/9018766 (Accessed: 10th June 2018).

ABC. (2019a) *Nobel Prizes 2019*. Available at: https://www.abc.net.au/radionational/programs/scienceshow/nobel-prizes-2019/11594218 (Accessed: 17th October 2019).

ABC. (2019b) *The Philosopher's Zone*. Available at: https://www.abc.net.au/radionational/programs/philosopherszone/politics-and-the-sacred/11714022 (Accessed: 27th January 2020).

ABC. (2019c) *The Philosopher's Zone*. Available at: https://www.abc.net.au/radionational/programs/philosopherszone/nutting-it-out/10988840 (Accessed: 16th April 2019).

Accenture (2019) *The Cost of Cybercrime.* Traverse City, Michigan 49629 USA Available at: https://www.accenture.com/_acnmedia/pdf-96/accenture-2019-cost-of-cybercrime-study-final.pdf (Accessed: 14th March 2020).

Ackoff, R. (1993) 'From Mechanistic to Social Thinking', *Systems Thinking in Action Conference*. Unknown Unknown. pp. 1-22. Available at: http://acasa.upenn.edu/socsysthnkg.pdf (Accessed: 21st May 2018).

If Russ Ackoff Had Given a Ted Talk... (2010) Available at: https://www.youtube.com/watch?v=OqEeIG8aPPk (Accessed: Unknown).

Agari (2019) *Scarlet Widow Bec Bitcoin Laundry: Scam, Rinse, Repeat.* Foster City, CA 94404 Available at: https://www.agari.com/cyber-intelligence-research/whitepapers/scarlet-widow-bec-scams.pdf (Accessed: 11th August 2020).

Akamai. (2020) *Sql Injection Tutorial*. Available at: https://www.akamai.com/uk/en/resources/sql-injection-tutorial.jsp (Accessed: 21st January 2020).

Al-Fannah, N. M., Li, W. and Mitchell, C. J. (2018) *Information Security. ISC 2018. Lecture Notes in Computer Science*. University of Surrey, Guildford, Surrey, UK 2018. Springer, Cham.

Alintanahin, K. (2015) *Operation Tropic Trooper: Relying on Tried-and-Tested Flaws to Infiltrate Secret Keepers.* Trend Micro Available at: http://apac.trendmicro.com/cloud-content/apac/pdfs/security-intelligence/white-papers/wp-operation-tropic-trooper.pdf (Accessed: 31st January 2017).

Amoroso, E. (1994) *Fundamentals of Computer Security Technology.* Englewood Cliffs. N.J.: Prentice-Hall.

Anderson, R., Barton, C., Böhme, R., Clayton, R., Eeten, M. J. G. v., Michael, L., Moore, T. and Savage, S. (2014) 'Measuring the Cost of Cybercrime', in Böhme, R. (ed.) *The Economics of Information Security and Privacy.* Heidelberg New York Dordrecht London: Springer, Anthe, C., Chrzan, P., Florio, E., Chad, F., Henry, P., Jones, J., Ng, N., O'Sullivan, N., Pecelj, D., Penta, A., Ragragio, I., Rains, T. and Rebriy, P. (2015) *Featured Intelligence.* Available at: http://download.microsoft.com/download/4/4/C/44CDEF0E-7924-4787-A56A-16261691ACE3/Microsoft_Security_Intelligence_Report_Volume_19_A_Profile_Of_A_Persistent_Adversary_English.pdf (Accessed: 6th October 2019).

Antonakakis, M., Demar, J., Christopher Elisan, C. and Jerrim, J. (2012) *Dgas and Cyber-Criminals: A Case Study.* Damballa Available at: https://web.archive.org/web/20130518123544/https://www.damballa.com/downloads/r_pubs/RN_DGAs-and-Cyber-Criminals-A-Case-Study.pdf (Accessed: 5th February 2017).

Antonakakis, M., Perdisci, R., Lee, W., Vasiloglou II, N. and Dagon, D. (2012) *Detecting Malware Domains at the Upper Dns Hierarchy.* USENIX: USENIX. Available at: https://www.usenix.org/legacy/event/sec11/tech/full_papers/Antonakakis.pdf (Accessed: 26th Janaury 2020).

Antonakakis, M., Perdisci, R., Nadji, Y., Nikolaos, V., Abu-Nimeh, S., Lee, W. and Dagon, D. (2012) *From Throw-Away Traffic to Bots: Detecting the Rise of Dga-Based Malware.* (Accessed: 5th February 2017).

Apache. (2019) *Apache Hadoop*. Available at: http://hadoop.apache.org/ (Accessed: 29th June 2019).

APM. (2018) *Glossary*. Available at: https://www.apm.org.uk/body-of-knowledge/glossary/ (Accessed: 27th October 2018).

Arborsert (2014) *Asert Threat Intelligence Brief 2014-07 Illuminating the Etumbot Apt Backdoor.* Available at: https://www.arbornetworks.com/blog/asert/wp-content/uploads/2014/06/ASERT-Threat-Intelligence-Brief-2014-07-Illuminating-Etumbot-APT.pdf (Accessed: 9th February 2017).

Arborsert (2015) *Asert Threat Intelligence Report 2015-08 Uncovering the Seven Pointed Dagger.* Available at: https://www.arbornetworks.com/blog/asert/wp-content/uploads/2016/01/ASERT-Threat-Intelligence-Brief-2015-08-Uncovering-the-Seven-Pointed-Dagger.pdf (Accessed: 9th February 2017).

Arntz, P. (2015) *Introduction to Alternate Data Streams* Available at: https://blog.malwarebytes.com/101/2015/07/introduction-to-alternate-data-streams/ (Accessed: 14th December 2019).

Arpaci-Dusseau, R. H. and Arpaci-Dusseau, A. C. (2018) *Operating Systems: Three Easy Pieces*. Available at: http://pages.cs.wisc.edu/~remzi/OSTEP/file-disks.pdf (Accessed: 3rd March 2018).

Arpaci-Dusseau, R. H. and Arpaci-Dusseau, A. C. (March 2015) *Operating Systems: Three Easy Pieces*. Available at: http://pages.cs.wisc.edu/~remzi/OSTEP/vm-smalltables.pdf

Assante, M. J. and Lee, R. M. (2015) *The Industrial Control System Cyber Kill Chain.* SANS Available at: https://www.sans.org/reading-room/whitepapers/ICS/industrial-control-system-cyber-kill-chain-36297 (Accessed: 4th November 2017).

Atwood, J. (2008) *Understanding User and Kernel Mode*. Available at: https://blog.codinghorror.com/understanding-user-and-kernel-mode/ (Accessed: 15th February 2020).

Austin, J. L. (1975) *How to Do Things with Words.* London (etc.): Oxford University Press.

Auty, M. (2015) 'Anatomy of an Advanced Persistent Threat', *Network Security,* 2015(4), pp. 13-16.

War Games (1983) Directed by Badham, J.

Ballenthin, W., Graeber, M. and Teodorescu, C. (2015) *Windows Management Instrumentation (Wmi) Offense, Defense and Forensics.* Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/wp-windows-management-instrumentation.pdf (Accessed: 29th Janaury 2017).

Balnaves, M. a. C., P. . (2001) *Introduction to Quantitative Research Methods: An Investigative Approach.* London: Sage.

Bank-of-England (2021) *Monetary Policy Report.* Available at: https://www.bankofengland.co.uk/-/media/boe/files/monetary-policy-report/2021/february/monetary-policy-report-february-2021.pdf (Accessed: 16th May 2021).

Banna , L. L., Singh, M. M. and Samsudin, A. (2015) *The Third Information Systems International Conference Procedia Computer Science*.

Barrett, D. (2017) *Talktalk Failed to Encrypt Data before Third Cyber Attack This Year*. Available at: http://www.telegraph.co.uk/news/uknews/law-and-order/11949468/TalkTalk-phone-network-hit-by-significant-cyber-attack.html (Accessed: 4th March 2017).

Bartholomew, B. and Guerrero-Saade, J. A. (2016) *Wave Your False Flags! Deception Tactics Muddying Atribituin in Targeted Attacks.* Available at: https://securelist.com/files/2016/10/Bartholomew-GuerreroSaade-VB2016.pdf (Accessed: 18th July 2017).

Barysevich, A. (2018) *The Use of Counterfeit Code Signing Certificates Is on the Rise.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2018-0222.pdf (Accessed: 20th February 2020).

Barysevich, A., Moriuchi, P. and Hatheway, D. (2017) *Proliferation of Mining Malware Signals a Shift in Cybercriminal Operations.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2017-1011.pdf (Accessed: 11th February 2020).

Becerra, X. (2020) *Data Brokers*. Available at: https://www.oag.ca.gov/data-brokers (Accessed: 30th Janaury 2020).

BeEF. (2020) *Beef the Browser Exploitation Framework Project*. Available at: https://beefproject.com/ (Accessed: 10th January 2020).

Bejtlich, R. (2010) *What Is Apt and What Does It Want?* . Available at: https://taosecurity.blogspot.co.uk/2010/01/what-is-apt-and-what-does-it-want.html (Accessed: 5th Decemebr 2017).

Bellard, F. (2018) *Tiny C Compiler*. Available at: https://bellard.org/tcc/ (Accessed: 15th November 2019).

Benchea, R., Maximciuc, A., Vatamanu, C. and Luncasu, V. (2015) *Apt28 under the Scope a Journey into Exfiltrating Intelligence and Government Information.* Bitdefender Available at:

https://download.bitdefender.com/resources/media/materials/white-papers/en/Bitdefender_In-depth_analysis_of_APT28%E2%80%93The_Political_Cyber-Espionage.pdf (Accessed: 5th October 2019).

Bentley. (2008) *Secure Erasure.* University of Westminter, UK.

Beowulf.org. (2019) *Beowulf.Org*. Available at: https://www.beowulf.org/ (Accessed: 29th June 2019).

Berghel, H. and Brajkovska, N. (2004) 'Wading into Alternate Data Streams', *Communications of the ACM - Human-computer etiquette* 47 (4), pp. 21-27.

Berghel, H. and Hoelzer, D. (2006) 'Disk Wiping by Any Other Name', *Communications of the ACM* 49(8), pp. 17-21.

Berghel, H., Hoelzer, D. and Sthultz, M. (2006) *Data Hiding Tactics for Windows and Unix File Systems* Available at: http://www.berghel.net/publications/data_hiding/data_hiding.php (Accessed: 2nd February 2018).

Bingham, J. (2012) *Backdoor.Proxybox Russian Hackers, Proxy Resellers and Rootkits.* Virus Bulletin Conference on behlaf of Symantec Available at: https://www.virusbulletin.com/uploads/pdf/conference_slides/2012/Bingham-VB2012.pdf (Accessed: 19th March 2017).

Bishop, M. (2005) *NSPW '05 Proceedings of the 2005 workshop on New security paradigms*. Lake Arrowhead, California September 20 - 23, 2005. ACM: ACM.

Blasco, J. (2010) 'Fighting Advanced Persistent Threats (Apt) with Open Source Tools', *Rooted CON'2010 Congreso de Seguridad*: Alien Vault. Available at: https://www.alienvault.com/blog-content/2011/09/Fighting-Advanced-Persistent-Threats-APT-with-Open-Source-Tools.pdf (Accessed: 18th July 2017).

Blue-Coat (2011) *Blue Coat Labs Report: Advanced Persistent Threats.* Blue Coat Available at: https://www.bluecoat.com/sites/default/files/documents/files/Advanced_Persistent_Threats.0.pdf (Accessed: 16th March 2017).

Blue-Coat (2013) *Open, Manage and Accelerate Ssl Encrypted Applications.* Available at: http://www.ipexpoeurope.com/content/download/5746/72152/file/bcs_wp_Accelerate_SSL_Encrypted_Applications_EN_5b.pdf%20(1).pdf (Accessed: 30th January 2017).

Blue-Coat (2014) *Revolutionizing Advanced Threat Protection.* (Accessed: 28th January 2017).

Boehm, B., Abts, C. and Chulani, S. (2000) 'Software Development Cost Estimation Approaches — a Survey', *IEEE Annals of Software Engineering,* 10(1-4), pp. 177–205.

Bontchev, V. (2006) 'Symbos Malware Classification Problems', *Virus Bulletin*. Montreal, Canada. Available at: https://www.virusbulletin.com/uploads/pdf/conference_slides/2006/VesselinBontchevVB2006.pdf (Accessed: 13th December 2019).

Bort, J. (2015) *Term of the Day: 'Google Dorking'*. Available at: http://www.businessinsider.com/term-of-the-day-google-dorking-2014-8?IR=T (Accessed: 27th May 2017).

Brewer, J. D. (2000) *Ethnography.* Buckingham: Open University Press.

British-Library. (2020) *Ethos E-Thesis Online Service*. Available at: https://ethos.bl.uk/Home.do (Accessed: 3rd January 2020).

Brooks, F. P., Jr. (1995) *The Mythical Man-Month: Essays on Software Engineering Anniversry Edition.* MA: Addison-Wesley.

Broomfield, M. (2017) 'Ntfs Alternate Data Streams: Focused Hacking', *Network Security,* 2006(Issue 8, August 2006), pp. 7-9.

Broukhis, L., Cooper, S. and Noll, L. C. (2019) *The International Obfuscated C Code Contest*. Available at: http://www.ioccc.org/ (Accessed: 4th February 2020).

Bruneau, G. (2010) *Dns Sinkhole.* SANS Available at: https://www.sans.org/reading-room/whitepapers/dns/dns-sinkhole-33523 (Accessed: 16th February 2020).

Burkett, R. (2013) 'An Alternative Framework for Agent Recruitment: From Mice to Rascls', *Studies in Intelligence V,* 57(ol. 57, No. 1 (Extracts, March 2013)),

ca-Technologies (2014) *Defending against Advanced Persistent Threats: Strategies for a New Era of Attacks.* ca Technologies Available at: http://www3.ca.com/ar/~/media/Files/eBooks/defending-against-advanced-persistent-threats.pdf (Accessed: 5th February 2017).

Cameron, R. (2011) 'Mixed Methods Research: The Five Ps Framework', *Electronic Journal of Business Research Methods,* 9(2), pp. 96-108.

Carr, N. (2017) *Cyber Espionage Is Alive and Well: Apt32 and the Threat to Global Corporations*. Available at: https://www.fireeye.com/blog/threat-research/2017/05/cyber-espionage-apt32.html (Accessed: 1st January 2020).

Carrier, B. (2005) *File System Forensic Analysis.* Boston, MA, USA: Addison Wesley.

Casey, E. (2007) *Handbook of Computer Crime Investigation. Forensic Tools and Technology.* London, UK etc.: Academic Press. An imprint of Elsevier Science.

Center-for-Strategic-and-International-Studies (2013) *The Economic Impact of Cybercrime & Cyber Espionage.* McAfee Available at: https://www.mcafee.com/us/resources/reports/rp-economic-impact-cybercrime.pdf (Accessed: 16th February 2017).

Cert-Polska (2014) *Cert-Polska Report 2014.* Available at: https://www.cert.pl/wp-content/uploads/2015/11/Report_CP_2014.pdf (Accessed: 11th February 2017).

Chandraiah, J. (2012) *Fake Antivirus: Journey from Trojan to a Persistent Threat.* Sophos Available at: https://www.google.co.uk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwisqprI3vnVAhWsLcAKHUnMCh4QFghAMAA&url=https%3A%2F%2Fwww.sophos.com%2Fen-us%2Fmedialibrary%2FPDFs%2Ftechnical-papers%2Fsophosfakeantivirusjourneyfromtrojantpna.pdf%3Fdl%3Dtrue&usg=AFQjCNHNguvryNv0VZuh2_lRmMPBzUMGjA (Accessed: 28th August 2017).

Chang, Z., Lu, K., Luo, A., Pernet, C. and Yaneza, J. (2015) *Operation Iron Tiger: Exploring Chinese Cyber-Espionage Attacks on United States Defense Contractors.* Trend Micro: Micro, T. Available at:

https://www.erai.com/CustomUploads/ca/wp/2015_12_wp_operation_iron_tiger.pdf (Accessed: 5th October 2019).

Chantry, G. (2016) *Cve-2012-0158: Anatomy of a Prolific Exploit.* Sophos Available at: https://www.sophos.com/en-us/medialibrary/PDFs/technical%20papers/CVE-2012-0158-An-Anatomy-of-a-Prolific-Exploit.PDF (Accessed: 28th January 2017).

Charette, R. N. (2005) 'Why Software Fails ', *IEEE Spectrum,* 42(9), pp. 42 - 49.

Check-Point (2013) *Check Point 2013 Security Report.* Check Point Available at: https://sc1.checkpoint.com/documents/security-report/files/assets/common/downloads/publication.pdf (Accessed: 31st January 2017).

Check-Point (2015) *Rocket Kitten: A Campaign with 9 Lives.* Check Point Available at: https://blog.checkpoint.com/wp-content/uploads/2015/11/rocket-kitten-report.pdf (Accessed: 12th February 2017).

Chen, J. C. and Li, B. (2015) *Evolution of Exploit Kits Exploring Past Trends and Current Improvements.* Trend Micro Available at: https://documents.trendmicro.com/assets/wp/wp-evolution-of-exploit-kits.pdf (Accessed: 9th February 2017).

Chen, P., Desmet, L. and Huygens, C. (2014) *15th IFIP International Conference on Communications and Multimedia Security (CMS), Sep 2014*. Aveiro, Portugal: CMS 2014. Lecture Notes in Computer Science.

Chicoine, P., Hassner, M., Grochowski, E., Jenness, S., Noblitt, M., Silvus, G., Stevens, C. and Weber, B. (2007) *Hard Disk Drive Long Data Sector White Paper*. Available at: http://www.idema.org/wp-content/plugins/download-monitor/download.php?id=1222 (Accessed: 18th May 2019).

Chien, E. (2005) *Techniques of Adware and Spyware.* Symantec Available at: https://www.symantec.com/avcenter/reference/techniques.of.adware.and.spyware.pdf (Accessed: 29th January 2017).

Chiu, D. (2015) *Shadow Force a Technical Brief.* Trend Micro Available at: http://documents.trendmicro.com/assets/pdf/shadow-force-technical-brief.pdf (Accessed: 16th February 2017).

Chiu, D., Weng, S.-H. and Chiu, J. (2014) *Backdoor Use in Targeted Attacks.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-backdoor-use-in-targeted-attacks.pdf (Accessed: 16th April 2017).

Cho, S., Han, I., Jeong, H., Kim, J., Koo, S., Oh, H. and Park, M. (2018) 'Cyber Kill Chain Based Threat Taxonomy and Its Application on Cyber Common Operational Picture ', *2018 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)* Scotland, United Kingdom, United Kingdom 11-12 June 2018. pp. 1 - 8 (Accessed: 5th December 2018).

Chohan, S., DeSombre, W. and Grosfelt, J. (2018) *Chinese Cyberespionage Originating from Tsinghua University Infrastructure.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2018-0626.pdf (Accessed: 20th February 2020).

Christiansen, M. (2010) *Bypassing Malware Defenses.* Available at: https://www.sans.org/reading-room/whitepapers/testing/bypassing-malware-defenses-33378 (Accessed: 18th August 2020).

Ciancaglini, D. V., Balduzzi, D. M., McArdle, R. and Rösler, M. (2015) *Below the Surface: Exploring the Deep Web.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp_below_the_surface.pdf (Accessed: 31st January 2017).

Ciancaglini, V., Balduzzi, M., Goncharov, M. and McArdle, R. (2013) *Deepweb and Cybercrime It's Not All About Tor.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-deepweb-and-cybercrime.pdf (Accessed: 15th May 2017).

CISCO (2013) *2013 Cisco Annual Security Eport.* CISCO (Accessed: 4th February 2017).

CISCO (2016) *Mitigating Web Threats with Comprehensive, Cloud-Delivered Web Security.* CISCO

Ciubotariu, M. (2012) *Trojan.Zeroaccess.C Hidden in Ntfs Ea*. Available at: https://www.symantec.com/connect/blogs/trojanzeroaccessc-hidden-ntfs-ea (Accessed: 24th January 2020).

Civil-Service (Undated) *Ethical Assurance for Social Research in Government.* Government Social Research Unit Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/515296/ethics_guidance_tcm6-5782.pdf (Accessed: 3rd May 2018).

Clark, J. and Robinson, J. (2013) *7 Stages of Advanced Threats.* Princeton, New Jersey Available at: https://www.fujitsu.com/uk/Images/whitepaper-seven-stages-of-advanced-threats.pdf (Accessed: 31st January 2017).

Clearsky (2015) *Thamar Reservoir an Iranian Cyber-Attack Campaign against Targets in the Middle East.* Clearsky Available at: https://www.clearskysec.com/wp-content/uploads/2015/06/Thamar-Reservoir-public1.pdf (Accessed: 6th October 2019).

Clearsky (2016) *Operation Dustysky.* Available at: https://www.clearskysec.com/wp-content/uploads/2016/01/Operation%20DustySky_TLP_WHITE.pdf (Accessed: 6th October 2019).

Cloppert, M. (2009) *Security Intelligence: Introduction (Pt 1)*. Available at: https://digital-forensics.sans.org/blog/2009/07/22/security-intelligence-introduction-pt-1 (Accessed: 18th December 2017).

Coffey, A. and Atkinson, P. (1996) *Making Sense of Qualitative Data.* London: Sage.

Cohen, F. B. D. (1994) *A Short Course on Computer Viruses.* New York, Chichester, Brisbane, Toronto. Singapore: John Wiley & Sons Inc.

The Italian Job (1969) Directed by Collinson, P.

Comodo Security Solutions, I. (2018) *How Antivirus Works?* Available at: https://antivirus.comodo.com/how-antivirus-software-works.php (Accessed: 23rd June 2018).

Comodo Security Solutions, I. (2019) *Antivirus : Malware/False-Positive*. Available at: https://www.comodo.com/home/internet-security/submit.php (Accessed: 30th June 2019).

Contratto, M. R. (2012) *The Decline of the Military Ethos and Profession of Arms an Argument against Autonomous Lethal Engagements.* Maxwell Air Force Base, Alabama: College, A. U. A. W. Available at:

https://media.defense.gov/2017/Dec/04/2001852014/-1/-1/0/MP_0062_CONTRATTO_MILITARY_ETHOS.PDF (Accessed: 30th December 2019).

Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C. (2009) *Introduction to Algorithms.* 3rd Edition edn. Cambrdige, Massachusetts: The MIT Press.

Cornford, F. M. (1945) *The Republic of Plato Translated with Introduction and Notes by Francis Macdonald Cornford.* New York and London: Oxford University Press.

CPNI (2013) *Cpni Insider Data Collection Study. Report of Main Findings.* CPNI: CPNI. Available at: https://www.cpni.gov.uk/system/files/documents/63/29/insider-data-collection-study-report-of-main-findings.pdf (Accessed: 19th July 2018).

CPNI. (2015) *Personnel Security: An Ongoing Responsibility Understanding Insider Threats – and Minimising the Risk.* CPNI`: CPNI.

CPNI. (2018) *Critical National Infrastructure*. Available at: Critical Infrastructure (Accessed: 5th December 2018).

Creative-Research-Systems. (2016) *Sample Size Calculator*. Available at: http://www.surveysystem.com/sscalc.htm (Accessed: 12th December 2016).

Crotty, M. (1998) *The Foundations of Social Research.* London: Sage.

CrowdStrike (2014a) *Crowdstrike Global Threat Report 2013 Year in Review.* CrowdStrike (Accessed: 3rd March 2017).

CrowdStrike (2014b) *Putter Panda.* CrowdStrike: CrowdStrike. Available at: https://cdn0.vox-cdn.com/assets/4589853/crowdstrike-intelligence-report-putter-panda.original.pdf (Accessed: 6th October 2019).

CrowdStrike (2016) *Proactive Hunting the Last Line of Defense against the "Mega Breach".* Available at: https://www.crowdstrike.com/wp-content/brochures/overwatch-datasheet/WP_Overwatch.pdf (Accessed: 6th February 2017).

CyberMonitor. (2019) *Apt & Cybercriminal Campaign Collection.* Github: Github.

Cylance (2016) *Operation Cleaver.* Cylance Available at: https://www.cylance.com/content/dam/cylance/pages/operation-cleaver/Cylance_Operation_Cleaver_Report.pdf (Accessed: 24th November 2017).

Cyrus, R. (2016) *Detecting Malicious Smb Activity Using Bro.* SANS Institute InfoSec Reading Room (Accessed: 16th March 2017).

Damballa (2012) *Dgas in the Hands of Cyber-Criminals Examining the State of the Art in Malware Evasion Techniques.* Damballa Available at: https://web.archive.org/web/20130518101218/https://www.damballa.com/downloads/r_pubs/WP_DGAs-in-the-Hands-of-Cyber-Criminals.pdf (Accessed: 14th October 2017).

Damballa (2014a) *Advanced Persistent Threats (Apts).* Damballa (Accessed: 5th February 2017).

Damballa (2014b) *Behind a Malware Lifecycle and Infection Chain Linking Asprox, Zemot, Rovix and Rerdom Malware Families.* Damballa (Accessed: 5th February 2017).

Damballa (2015) *Ponyup: Tracing Pony's Threat Cycle and Multi-Stage Infection Chain.* Damballa (Accessed: 5th February 2017).

Damballa (2016) *Behind Today's Crimeware Installation Lifecycle: How Advanced Malware Morphs to Remain Stealthy and Persistent.* Damballa (Accessed: 5th February 2017).

Damshenas, M., Dehghantanha, A. and Mahmoud, R. (2013) 'A Survey on Malware Propagation, Analysis, and Detection', *International Journal of Cyber-Security and Digital Forensics,* 2(4),

Davidson, L. (2009) *Windows 7 Uac Whitelist: Code-Injection Issue (and More)*. Available at: https://www.pretentiousname.com/misc/win7_uac_whitelist2.html (Accessed: 28th Janaury 2020).

Dela Paz, R. (2011) *Worm Poses as a Font File, Uses Lnk Vulnerability to Propagate*. Available at: http://blog.trendmicro.com/trendlabs-security-intelligence/worm-posing-as-a-font-file-trigger-exploits/ (Accessed: 19th May 2017).

Dela Paz, R. (2012) *The Heartbeat Apt Campaign.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp_the-heartbeat-apt-campaign.pdf (Accessed: 31st January 2017).

Dela Vega, M. and Ingal, N. (2010) *The Dark Side of Trusting Web Searches from Blackhat Seo to System Infection.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp__trusting-web-search-blackhat-seo.pdf (Accessed: 9th February 2017).

Dell-SecureWorks-Counter-Threat-Unit-Threat-Intelligence. (2013) *Wiper Malware Threat Analysis*. Available at: https://www.secureworks.com/research/wiper-malware-analysis-attacking-korean-financial-sector (Accessed: 19th November 2017).

Dell-SecureWorks. (2015) *5 Ways Advanced Malware Evades the Sandbox*. Available at: https://www.secureworks.com/blog/bg-5-ways-malware-evades-the-sandbox (Accessed: 19th November 2017).

DeSombre, W. and Byrnes, D. (2018) *Thieves and Geeks: Russian and Chinese Hacking Communities.* Recorded Future Available at: https://go.recordedfuture.com/hubfs/reports/cta-2018-1010.pdf (Accessed: 11th February 2020).

Dijkstra, E. W., Hoare, C. A. R. and Dahl, O.-J. (1972) '1. Notes on Structured Programming', in Hoare, C. A. R. (ed.) *Structured Programming.* London, UK, UK Academic Press Ltd.,

DiMaggio, J. (2015) *The Black Vine Cyberespionage Group.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-black-vine-cyberespionage-group.pdf (Accessed: 29th January 2017).

Dizon, J., Galang, L. and Cruz, M. (2010) *Understanding Wmi Malware.* Trend Micro Available at: http://la.trendmicro.com/media/misc/understanding-wmi-malware-research-paper-en.pdf (Accessed: 20th May 2017).

Doherty, S., Gegeny, J., Spasojevic, B. and Baltazar, J. (2013) *Hidden Lynx – Professional Hackers for Hire.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/hidden_lynx.pdf (Accessed: 29th January 2017).

Dolbeau, R. (2018) 'Theoretical Peak Flops Per Instruction Set: A Tutorial', *The Journal of Supercomputing,* 74(3), pp. 1341–1377

Douglas, J. C. (2015) *Cyber Dwell Time and Lateral Movement the New Cybersecurity Blueprint.* Raytheon Wensense Available at: https://www.raytheon.com/capabilities/rtnwcm/groups/cyber/documents/content/rtn_269210.pdf (Accessed: 7th October 2017).

Downey, A. B. (2001) 'The Structural Cause of File Size Distributions', *Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems* Cambridge, Massachusetts, USA: ACM New York, NY, USA. pp. 328-329. Available at: https://dl.acm.org/citation.cfm?id=378824 (Accessed: 8th March 2018).

Dunleavy, P. (2003) *Authoring a Phd.* Basingstoke: Palgrave.

EDAA. (2019) *European Interactive Digital Advertising Alliance*. Available at: https://www.edaa.eu/ (Accessed: 1st August 2019).

Edwards, S. P. G., Ford, R. and Szappanos, G. (2014) *Effectively Testing Apt Defences.* Sophos Available at: https://www.sophos.com/en-us/medialibrary/PDFs/technical%20papers/sophos-szappanos-effectively-testing-APT-defenses-VB2015.pdf?la=en (Accessed: 9th February 2017).

Egele, M., Scholte, T., Kirda, E. and Kruegel , C. (2012) 'A Survey on Automated Dynamic Malware-Analysis Techniques and Tools', *ACM Computing Surveys (CSUR),* 44(2),

Eliz, C. (2018) *How Exactly Does an Anti-Virus Software Work?* Available at: https://www.quora.com/How-exactly-does-an-Anti-Virus-software-work (Accessed: 23rd June 2018).

Emm, D., Garnaeva, M., Ivanov, A., Makrushin, D. and Unuchek, R. (2015) *It Threat Evolution in Q2 2015.* Available at: https://cdn.securelist.com/files/2015/08/IT_threat_evolution_Q2_2015_ENG.pdf (Accessed: 18th July 2017).

Engel, G. (2014) *Deconstructing the Cyber Kill Chain*. Available at: http://www.darkreading.com/attacks-breaches/deconstructing-the-cyber-kill-chain/a/d-id/1317542 (Accessed: 5th October 2018).

Epstein, R. (1984) 'The Principle of Parsimony and Some Applications in Psychol', *Journal Of Mind And Behavior,* 5(2 Spring 1984), pp. 119-130.

eset (2013) *Did You Say Advanced Persistent Threats?* eset Available at: https://www.welivesecurity.com/wp-content/uploads/2013/12/Advanced-Persistent-Threats.pdf (Accessed: 1st December 2017).

eset (2015) *Operation Potao Express Analysis of a Cyber-Espionage Toolkit.* eset Available at: https://www.welivesecurity.com/wp-content/uploads/2015/07/Operation-Potao-Express_final_v2.pdf (Accessed: 18th July 2017).

eset (2016a) *En Route with Sednit Part 1: Approaching the Target.* eset Available at: https://www.welivesecurity.com/wp-content/uploads/2016/10/eset-sednit-part1.pdf (Accessed: 18th July 2017).

eset (2016b) *Operation Groundbait: Analysis of a Surveillance Toolkit.* eset Available at: https://www.welivesecurity.com/wp-content/uploads/2016/05/Operation-Groundbait.pdf (Accessed: 18th July 2017).

eset (2017) *Gazing at Gazer Turla's New Second Stage Backdoor.* Available at: https://www.welivesecurity.com/wp-content/uploads/2017/08/eset-gazer.pdf (Accessed: 24th April 2018).

Eterovic-Soric, B., Choo, K.-K. R., Mubarak, S. and Ashman, H. (2018) 'Windows 7 Antiforensics: A Review and a Novel Approach', *Journal of Forensic Sciences,* 62(4), pp. 1054-1070.

Council Regulation (Ec) No 428/2009 of 5 May 2009 Setting up a Community Regime for the Control of Exports, Transfer, Brokering and Transit of Dual-Use Items, 428/2009 C.F.R. (2009).

Evans, K. M. and Kuenning, G. H. (2002) *Proceedings of the International Symposium on Performance Evaluation of Computer and TelecommunicationSystems (SPECTS)*. San Diego, CA.

F-Secure-Labs-Malware-Analysis (2017) *Callisto Group.* F-Secure Available at: https://www.f-secure.com/documents/996508/1030745/callisto-group (Accessed: 13th April 2017).

F-Secure-Labs (2014a) *Blackenergy & Quedagh the Convergence of Crimeware and Apt Attacks.* Available at: https://www.f-secure.com/documents/996508/1030745/blackenergy_whitepaper.pdf (Accessed: 31st January 2017).

F-secure-Labs (2014b) *Cosmicduke: Cosmu with a Twist of Miniduke.* F-Secure (Accessed: 13th April 2017).

F-Secure-Labs (2014c) *Mobile Threat Report Q1 2014.* (Accessed: 31st January 2017).

F-Secure-Labs (2014d) *Pitou the "Silent" Resurrection of the Notorious Srizbi Kernel Spambot.* Available at: https://www.f-secure.com/documents/996508/1030745/pitou_whitepaper.pdf (Accessed: 31st January 2017).

F-Secure-Labs (2015a) *Cozyduke: Malware Analysis.* F-Secure Available at: https://www.f-secure.com/documents/996508/1030745/CozyDuke (Accessed: 31st January 2017).

F-Secure-Labs (2015b) *The Dukes 7 Years of Russian Cyberespionage.* F-Secure Available at: https://www.f-secure.com/documents/996508/1030745/dukes_whitepaper.pdf (Accessed: 29th January 2017).

F-Secure-Labs (2015c) *Lecpetex Virtual Currency Mining Gets Social.* Available at: https://www.f-secure.com/documents/996508/1030745/lecpetex_whitepaper.pdf (Accessed: 4th March 2017).

F-Secure-Labs (2016) *Nanhaishu Rating the South China Sea.* Available at: https://www.f-secure.com/documents/996508/1030745/nanhaishu_whitepaper.pdf (Accessed: 31st Janaury 2017).

F-Secure (2016) *Threat Report 2015.* F-Secure Available at: https://www.f-secure.com/documents/996508/1030743/Threat_Report_2015.pdf (Accessed: 31st January 2017).

F-Secure. (2017) *Whitepapers the Latest Research from Labs on Threats and Technology*. Available at: https://www.f-secure.com/en/web/labs_global/whitepapers (Accessed: 4th March 2017).

F-Secure. (2020) *Backdoor:W32/Hupigon*. Available at: https://www.f-secure.com/v-descs/backdoor_w32_hupigon.shtml (Accessed: 24th October 2020).

Falliere, N., Murchu, L. O. and Chien, E. (2011) *W32.Stuxnet Dossier.* Symantec Available at: https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf (Accessed: 31st January 2017).

Fang, X., Zhai, L., Jia, Z. and Bai, W. (2014) *2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing*. Dalian, China IEEE.

Farlex. (2018) 'Mechanism of Action', *The Free Dcitionary by Farlex*. Farlex.

Farlex. (2020) 'Gain of Function Mutation', *The Free Dcitionary by Farlex*. Farlex, Inc

FDA. (2020) *510(K) Clearances*. Available at: https://www.fda.gov/medical-devices/device-approvals-denials-and-clearances/510k-clearances (Accessed: 15th March 2020).

Fewer, S. (2008) *Reflective Dll Injection V1.0.* Available at: https://pdfs.semanticscholar.org/d30c/d54b10948fef0046f61fd05cd18eed4cc561.pdf (Accessed: 12th October 2019).

Feynman, R. P. (1959) 'Plenty of Room at the Bottom', *Unknown*. Pasadena. Available at: https://web.pa.msu.edu/people/yang/RFeynman_plentySpace.pdf (Accessed: 19th May 2019).

Fidel-Cybersecurity (2016) *Down the H-W0rm Hole with Houdini's Rat.* Fidel Cybersecurity Available at: https://www.fidelissecurity.com/sites/default/files/TA_Fidelis_H-W0rm_1611.pdf (Accessed: 1st April 2017).

Fidelis-Cybersecurity-Solutions (2015) *Ratting on Alienspy.* General Dynamics Available at: https://www.fidelissecurity.com/sites/default/files/FTA_1015_Alienspy_FINAL.pdf (Accessed: 1st April 2017).

Fidelis-Cybersecurity (2015a) *Dissecting the Malware Involved in the Inocnation Campaign.* Fidelis Cybersecurity Available at: https://www.fidelissecurity.com/sites/default/files/FTA_1020_Fidelis_Inocnation_FINAL_0.pdf (Accessed: 1st April 2017).

Fidelis-Cybersecurity (2015b) *Pushdo It to Me One More Time.* Fidelis Cybersecurity Available at: https://www.fidelissecurity.com/sites/default/files/FTA_1016_Pushdo.pdf (Accessed: 1st April 2015).

Financial-Times. (2016) *The Psychology of What We Do with Our Money.* Financial Times

Financial-Times. (2017a) *Blazing a Trail for Cycling Safety* Financial-Times

Financial-Times. (2017b) *A Turkish Operation Is in the Forefront of Killing Off the Dreaded Password.* Financial Times

Financial-Times. (2018) *How Fighting a Computer Bug Turned into a Vocation* Financial Times

FireEye (2011) *Fireeye Advanced Threat Report – 2h 2011.* CA 95035 Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-firefye-advanced-threat-report-2H2011.pdf (Accessed: 15th February 2020).

Fireeye (2013) *Fireeye Advanced Threat Report: 2013.* Fireeye Available at: https://www2.fireeye.com/rs/fireye/images/fireeye-advanced-threat-report-2013.pdf (Accessed: 9th April 2017).

Fireeye (2014a) *Apt28: A Windows into Russia's Cyber Espionage Operations?* Fireeye Available at: https://www2.fireeye.com/rs/fireye/images/rpt-apt28.pdf (Accessed: 16th March 2017).

Fireeye (2014b) *Digital Bread Crumbs: Seven Clues to Identifying Who's Behind Advanced Cyber Attacks.* Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-digital-bread-crumbs.pdf (Accessed: 29th January 2017).

Fireeye (2014c) *Leviathan: Command and Control Communications on Planet Earth.* Fireeye Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-leviathan.pdf (Accessed: 29th January 2017).

Fireeye (2014d) *Poision Ivy: Assessing Damage and Extracting Intelligence.* Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-poison-ivy.pdf (Accessed: 29th January 2017).

Fireeye (2014e) *Regional Advanced Threat Report: Europe, Middle East and Africa 1h2014.* Fireeye Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/fireeye-emea-advanced-threat-report-1h2014.pdf (Accessed: 25th February 2017).

Fireeye (2014f) *Supply Chain Analysis: From Quartermaster to Sunshop.* Fireeye Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-malware-supply-chain.pdf (Accessed: 29th January 2017).

Fireeye (2014g) *Top Words Used in Spear Phishing Attacks: Successfully Compromise Enterprise Networks and Steal Data.* Fireeye Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-top-spear-phishing-words.pdf (Accessed: 29th January 2017).

Fireeye (2015a) *Apt30 and the Mechanics of a Long-Running Cyber Espionage Operation.* Available at: https://www2.fireeye.com/rs/fireye/images/rpt-apt30.pdf (Accessed: 17th February 2018).

Fireeye (2015b) *Behind the Syrian Conflict's Digital Front Lines.* Fireeye Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-behind-the-syria-conflict.pdf (Accessed: 5th February 2017).

Fireeye (2015c) *Hammertoss: Stealthy Tactics Define a Russian Cyber Threat Group.* Fireeye Available at: https://www2.fireeye.com/rs/848-DID-242/images/rpt-apt29-hammertoss.pdf (Accessed: 29th Janaury 2017).

Fireeye (2015d) *Hiding in Plain Sight: Fireeye and Microsoft Expose Obfuscation Tactic.* Available at: https://www2.fireeye.com/rs/fireye/images/APT17_Report.pdf (Accessed: 29th January 2017).

Fireeye (2015e) *Regional Advanced Threat Report: Asia Pacific 1h 2015.* Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-regional-atr-apac.pdf (Accessed: 5th February 2017).

Fireeye (2015f) *Southeast Asia: An Evolving Cyber Threat Landscape.* Available at: https://www.fireeye.com/content/dam/fireeye-www/current-threats/pdfs/rpt-southeast-asia-threat-landscape.pdf (Accessed: 5th February 2017).

Fireeye (2016) *Follow the Money: Dissecting the Operations of the Cyber Crime Grouo Fin6.* Available at: https://www2.fireeye.com/rs/848-DID-242/images/rpt-fin6.pdf (Accessed: 6th February 2017).

Fireeye (2017a) *Apt28: At the Center of the Storm Russia Strategically Evolves Its Cyber Operations.* Fireeye iSightFireeye Available at: https://www2.fireeye.com/rs/848-DID-242/images/APT28-Center-of-Storm-2017.pdf (Accessed: 5th February 2017).

Fireeye. (2017b) *To Sdb, or Not to Sdb: Fin7 Leveraging Shim Databases for Persistence*. Available at: https://www.fireeye.com/blog/threat-research/2017/05/fin7-shim-databases-persistence.html (Accessed: 7th April 2021).

FireEye (2020) *Apt37 (Reaper) the Overlooked North Korean Actor.* CA 95035, USA Available at: https://www2.fireeye.com/rs/848-DID-242/images/rpt_APT37.pdf (Accessed: 3rd October 2020).

Fireeye and BT (2015) *Regional Advanced Threat Report: Europe, Middle East and Africa 1h2015.* Fireeye Available at: https://www.fireeye.com/content/dam/fireeye-www/partners/pdfs/rpt-regional-atr-emea-web-bt.pdf (Accessed: 6th April 2017).

Fireeye and Mandiant (2014) *Cybersecurity's Maginot Line: A Real-World Assessment of the Defense-in-Depth Model.* Fireeye Available at: http://files.shareholder.com/downloads/AMDA-254Q5F/4850316747x0x762699/9915B2A4-FF47-4F45-BA98-F30EA38E73BE/fireeye-real-world-assessment.pdf (Accessed: 9th April 2017).

Fisher , S. (2019) *6 Free Online Virus Scanners*. Available at: https://www.lifewire.com/free-online-virus-scanners-1356651 (Accessed: 30th June 2019).

Florio, E. (2005) *When Malware Meets Rootkits.* Available at: https://www.symantec.com/avcenter/reference/when.malware.meets.rootkits.pdf (Accessed: 14th July 2017).

Florio, E. and Kasslin, K. (2009) *Your Computer Is Now Stoned (...Again!) the Rise of Mbr Rootkits.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/your_computer_is_now_stoned.pdf (Accessed: 14th July 2017).

Fogleman, R. R. (1996) *US Air Foirce Symposium*. Chicago Pro Rhetoric, LLC.

Forcepoint-Security_Labs (2016) *2015 Threat Report.* Available at: https://www.forcepoint.com/sites/default/files/resources/files/whitepaper-2015-threat-report-enus-08jan16-lr_0.pdf (Accessed: 9th February 2017).

Forcepoint-Security_Labs. (2017) *Knowledge Is Power: Identify the 7 Stages of Advanced Attacks*. Available at: https://www.forcepoint.com/seven-stages (Accessed: 10th October 2017).

Fosburgh, L. (1973) 'Chief Teller Is Accused of Theft of $1.5-Million at a Bank Here', *The New York Times* March 23, 1973

The Uk Corporate Governance Code, (2018).

Friedberg, I., Skopik, F., Settanni, G. and Fiedler, R. (2015) 'Combating Advanced Persistent Threats: From Network Event Correlation to Incident Detection', *Computers and Security* 48(February 2015), pp. 35-57.

Friedman, W. F. (1935) *The Index of Coincidence and Its Application in Cryptanalysis.* Washington: United States Government Printing Office.

Frydman, C. and Camerer, C. F. (2016) 'The Psychology and Neuroscience of Financial Decision Making', *Trends in Cognitive Sciences,* 20(9),

Futility-Closet. (2017) *Podcast Episode 84: The Man Who Never Was*

Gaikwad, P., Motwani, P. D. and Shinde, P. V. (2015) 'International Journal of Modern Trends in Engineering and Research', *Scientific Journal Impact Factor (SJIF),* 2(1), pp. 493-497.

Gaines, H. F. (1956) *Cryptanalysis: A Study of Ciphers and Their Solution* United States: Dover Publciations, Inc.

Gandotra, E., Bansal, D. and Sofat, S. (2014) 'Malware Analysis and Classification: A Survey', *Journal of Information Security,* 5, pp. 56-64.

Garba, F. A., Junaidu, S. B., Ahmad, B. I. and Tekanyi, A. M. S. (Unknown) 'Proposed Framework for Effective Detection and Prediction of Advanced Persistent Threats Based on the Cyber Kill Chain'. pp. 1-10. (Accessed: 5th December 2018).

GetData-Forensics. (2020) *Hash Sets*. Available at: http://www.forensicexplorer.com/hashsets.php (Accessed: 24th January 2020).

GitHub. (2020) *Malware-Samples*. Available at: https://github.com/topics/malware-samples (Accessed: 8th September 2020).

Giuliani, M. (2011) *Zeroaccess – an Advanced Kernel Mode Rootkit (Rev. 1.2).* Prevx Available at: https://www.marcogiuliani.eu/articles/prevx/zeroaccess_analysis.pdf (Accessed: 24th April 2018).

Giura, P. and Wang, W. (2012) *Using Large Scale Distributed Computing to Unveil Advanced Persistent Threats.* Available at: http://web2.research.att.com/export/sites/att_labs/techdocs/TD_101075.pdf (Accessed: 4th March 2018).

Glyer, C. (2010) *Examples of Recent Apt Persistence Mechanisms.* Available at: https://digital-forensics.sans.org/summit-archives/2010/35-glyer-apt-persistence-mechanisms.pdf (Accessed: 13th March 2017).

Goncharov, M. (2012) *Russian Underground 101.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-russian-underground-101.pdf (Accessed: 9th February 2017).

Goncharov, M. (2014) *Russian Underground Revisited.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-russian-underground-revisited.pdf (Accessed: 9th February 2017).

Google-Trends. (2017) *Google Trends*. Available at: https://trends.google.com/trends/

Google-Trends. (2019) *Move Fast Break Things*. Available at: https://trends.google.com/trends/explore?date=all&q=move%20fast%20break%20things (Accessed: 29th October 2019).

Google-Trends. (2020) *Google Trends - Cyber Kill Chain*. Available at: https://trends.google.com/trends/explore?date=all&q=Cyber%20Kill%20Chain (Accessed: 11th August 2020).

Google. (2019a) *Get Started Configure Your Network Settings to Use Google Public Dns*. Available at: https://developers.google.com/speed/public-dns/docs/using (Accessed: 25th January 2020).

Google. (2019b) *Tag Manager Overview*. Available at: https://support.google.com/tagmanager/answer/6102821?hl=en (Accessed: 11th September 2019).

Grant, T., Burke, I. and van Heerden, R. (2012) *ICIW2012-Proceedings of the 7th International Conference on Information Warfare and Security: ICIW2012*. Univeristy of Washington, Seattle, USA 22-23 March 2012. Academic Publishing International, Reading, UK.

Grill, B., Platzer, C. and Eckel, J. (2014) *EuroSec '14 Proceedings of the Seventh European Workshop on System Security*. Amsterdam, The Netherlands April 13 - 13, 2014 New York, NY, USA: ACM.

Grix, J. (2010) *The Foundations of Research. 2nd Edition.* Basingstoke, Hampshire, United Kingdom: Palgrave Macmillan.

Gross, J. and Cylance-Spear-Team (2016) *Operation Dust Storm.* Cylance Available at: https://www.cylance.com/content/dam/cylance/pdfs/reports/Op_Dust_Storm_Report.pdf (Accessed: 24th November 2017).

Gruhn, M. (2017) 'Forensic Limbo: Towards Subverting Hard Disk Firmware Bootkits', *Digital Investigation,* 23, pp. 138-150.

Gu, L. (2014) *The Chinese Underground in 2013.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-the-chinese-underground-in-2013.pdf (Accessed: 15th May 2017).

Gu, L. (2014) *The Mobile Cybercriminal Underground Market in China.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-the-mobile-cybercriminal-underground-market-in-china.pdf (Accessed: 9th February 2017).

Guarnieri, C. and Anderson, C. (2016) 'Iran and the Soft War for Internet Dominance', *Black Hat USA*: Black Hat. Available at: https://www.blackhat.com/docs/us-16/materials/us-16-Guarnieri-Iran-And-The-Soft-War-For-Internet-Dominance-wp.pdf (Accessed: 12th February 2017).

Guerrero-Saade, J. A. (2016) 'The Ethics and Perils of Apt Research: An Unexpected Transition into Inteligence Brokerage', *Virus Bulletin Conference* September 2015. Virus Bulletin. Available at: https://www.virusbulletin.com/uploads/pdf/magazine/2016/vb201601-ethics-and-perils.pdf

Guerrero-Saade, J. A. and Chohan, S. (2018) *Redalpha: New Campaigns Discovered Targeting the Tibetan Community.* Recorded Future Available at: https://go.recordedfuture.com/hubfs/reports/cta-2018-0626.pdf (Accessed: 20th February 2020).

Guerrero-Saade, J. A., Moriuchi, P. and Lesnewich, G. (2018) *Targeting of Olympic Games It Infrastructure Remains Unattributed.* Available at:

https://go.recordedfuture.com/hubfs/reports/fr-2018-0214.pdf (Accessed: 20th February 2020).

Gundert, L., Chohan, S. and Lesnewich, G. (2018) *Iran's Hacker Hierarchy Exposed.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2018-0509.pdf (Accessed: 11th February 2020).

Gutmann, P. (1996) 'Secure Deletion of Data from Magnetic and Solid-State Memory', *Proceedings of Sixth USENIX Security Symposium* pp. 77-90.

Hacquebord, F. (2017) *Two Years of Pawn Storm Examining an Increasingly Relevant Threat.* Trend Micro Available at: https://documents.trendmicro.com/assets/wp/wp-two-years-of-pawn-storm.pdf (Accessed: 28th April 2017).

Hammersley, M. and Atkinson, P. (2009) *Ethnography: Principles in Practice.* London: Tavistock.

Hamre, D. (2005) *How Many Bytes in a Zero-Length File?* Available at: http://drewhamre.com/documents/LenZero.pdf (Accessed: 13th December 2019).

Han, K., Lim, J. H. and Im, E. G. (2013) *RACS '13 Proceedings of the 2013 Research in Adaptive and Convergent Systems*. Montreal, Quebec, Canada October 01 - 04, 2013. New York, NY, USA: ACM.

Haq, T., Moran, N., Vashisht, S. and Scott, M. (2014) *Operation Quantum Entanglement.* Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/wp-operation-quantum-entanglement.pdf (Accessed: 5th February 2017).

Stealth Secrets of the Malware Ninjas (2007) Available at: https://www.blackhat.com/presentations/bh-usa-07/Harbour/Presentation/bh-usa-07-harbour.pdf (Accessed: 7th February 2020).

Harbour, N. (2010) *Malware Persistence without the Windows Registry*. Available at: https://www.fireeye.com/blog/threat-research/2010/07/malware-persistence-windows-registry.html (Accessed: 1st Janaury 2020).

Harford, T. (2016) *Delusions of Objectivity*. Available at: http://timharford.com/2016/04/delusions-of-objectivity/ (Accessed: 12th December 2016).

Harkaway, N. (2017) *Gnomon.* Kindle edn. London: William Heinemann.

Harrington, A. (2005) *Modern Social Theory.* Oxford: Oxford University Press.

Hart, C. (2001) *Doing a Literature Search.* London:: Sage.

Haughey, D. (2014) *A Brief History of Project Management*. Available at: https://www.projectsmart.co.uk/brief-history-of-project-management.php (Accessed: 9th November 2019).

HESA. (2016) *Staff at Higher Education Providers in the United Kingdom 2014/15*. Available at: https://www.hesa.ac.uk/news/21-01-2016/sfr225-staff (Accessed: 12th December 2016).

How to Recover Data after Formatting, Deleting or Creating Partitions in 2019 (2019) Hetman Recovery Available at: https://www.youtube.com/watch?v=WhyMcH4iWpc (Accessed: 18th August 2019).

Hipolito, J. (2016) *Enterprise Protection against Cyberattacks Primer: The Ukrainian Power Facility Attack.* Trend Micro Available at:

http://documents.trendmicro.com/assets/primers/enterprise-network-protection-blackenergy.pdf (Accessed: 28th May 2017).

HMG. (1990) *Computer Misuse Act, 1990*. Available at: http://www.legislation.gov.uk/ukpga/1990/18/introduction (Accessed: 12th December 2016).

HMG. (2014) *Reserach and Technologies: Big Data*. Available at: https://www.gov.uk/government/publications/eight-great-technologies-big-data (Accessed: 12th December 2016).

HMG (2019a) *Cyber Security Breaches Survey 2019.* London Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/813599/Cyber_Security_Breaches_Survey_2019_-_Main_Report.pdf (Accessed: 14th March 2020).

HMG. (2019b) *Export Controls: Dual-Use Items, Software and Technology, Goods for Torture and Radioactive Sources*. HMG. Available at: https://www.gov.uk/guidance/export-controls-dual-use-items-software-and-technology-goods-for-torture-and-radioactive-sources#dual-use-items-software-and-technology (Accessed: 3rd January 2020).

HMSO (1967) *Report of the Tribunal Appointed to Inquire into the Disaster at Aberfan on October 21st, 1966.* London

Hofstetter, C. (2019) */Etc/Hosts to Block Shock Sites Etc.* . Available at: https://gist.github.com/consti/8022703 (Accessed: 10th November 2019).

Holland, R. (2013) *Introducing Forrester's Cyber Threat Intelligence Research*. Available at: https://go.forrester.com/blogs/introducing-forresters-cyber-threat-intelligence-research/ (Accessed: 18th December 2017).

Homan, R. (1991) *The Ethics of Social Research.* London: Longman.

Hsu, F.-H., Wu, M.-H., Ou, S.-C. and Wang, S.-J. (2016) 'Data Concealments with High Privacy in New Technology File System', *The Journal of Supercomputing,* 72(1), pp. 120–140

Hudson, B. (2014) *Advanced Persistent Threats: Detection,Protection and Prevention.* Sophos Available at: http://resources.idgenterprise.com/original/AST-0112935_sophos-advanced-persistent-threats-detection-protection-prevention.pdf (Accessed: 3rd March 2017).

Hume, D. (MDCCXXXIX) *A Treatise of Human Nature : Being an Attempt to Introduce the Experimental Method of Reasoning into Moral Subjects. Vol. I. Of the Understanding.* LONDON: Printed for John Noon, at the White-Hart, near Mercer's-Chapel, in Cheapside.

Huq, N. (2016) *Cyber Threats to the Mining Industry.* Trend Mico Available at: http://documents.trendmicro.com/assets/wp/wp-cyber-threats-to-the-mining-industry.pdf (Accessed: 31st January 2017).

Huss, D. (2016) *Https://Www.Proofpoint.Com/Sites/Default/Files/Proofpoint-Operation-Transparent-Tribe-Threat-Insight-En.Pdf.* Proofpoint: Proofpoint. Available at: https://www.proofpoint.com/sites/default/files/proofpoint-operation-transparent-tribe-threat-insight-en.pdf (Accessed: 6th October 2019).

Hutchins, E. M., Clopperty, M. J. and Amin, R. M. (2011) *6th International Conference on Information Warfare and Security*. The George Washington

University, Washington, DC, USA 17-18 March 2011. Academic Conferences and Publishing International Limited.

Hutchins, E. M., Clopperty, M. J. and M., A. R. (2011) *6th International Conference on Information Warfare and Security*. The George Washington University, Washington, DC, USA 17-18 March 2011. Academic Conferences and Publishing International Limited.

Hybrid-Analysis. (2016) *Joint_Black.Exe*. Available at: https://www.hybrid-analysis.com/sample/b56684485c881ff01091a77fb1622522f3931bc7b85177e6e7e0b0baabbdef58?environmentId=1

Hybrid-Analysis. (2018a) *A59d3f6cc83d0a7247d7a080e6a7a38edc4a220b299a1650c37673f05c775381.Exe* Available at: https://www.hybrid-analysis.com/sample/a59d3f6cc83d0a7247d7a080e6a7a38edc4a220b299a1650c37673f05c775381?environmentId=100 (Accessed: 24th October 2018).

Hybrid-Analysis. (2018b) *F404c8fc0ffb3e5cb4fd1437c9ae1b22*. Available at: https://www.hybrid-analysis.com/sample/56877d13102bcc0a06bf9fc9fcf1a9209e80549a9fa8d680011395b50d783e6a?environmentId=110 (Accessed: 25th October 2020).

Hybrid-Analysis. (2018) *Adlmint.Dll.Save*. Available at: https://www.hybrid-analysis.com/sample/de8d005494736056e05583708787d8589a3f9f5fcfe575c0baf397bf88c5b94d?environmentId=120 (Accessed: 24th October 2020).

Hybrid-Analysis. (2019) *917949-F2022816.Dll*. Available at: https://hybrid-analysis.com/sample/9cf514cce4c590a5d0d81c39712e0718c5d78fd19824272b0e02523b7f0455f3?environmentId=120

Hybrid-Analysis. (2019) *Hkcmd.Exe*. Available at: https://www.hybrid-analysis.com/sample/feae8a92384d76f884d2f56cf5fb10c6a3221c39d71e0f1a181ca2199125663a?environmentId=100 (Accessed: 24th October 2020).

Hybrid-Analysis. (2020) *Cpuz_X32.Exe*. Available at: https://www.hybrid-analysis.com/sample/71e0ed0864b2e5b257bf121426efe7e51f86ab66d993fa87a349364186280ab5/5f668a738fc1bf289a2a62bb (Accessed: 24th October 2020).

IANA. (2020) *Service Name and Transport Protocol Port Number Registry*. Available at: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml (Accessed: 2nd February 2020).

IBISWorld (2021) *Uk Business Environment Profiles Report D2233* IBISWorld Available at: https://my-ibisworld-com.glos.idm.oclc.org/uk/en/business-environment-profiles/d2233/business-environment-profile (Accessed: 14th May 2021).

ICANN. (2017) *Icann Whois*. Available at: https://whois.icann.org/en/lookup?name=senseient.com (Accessed: 18th May 2017).

ICANN. (2019) *Icann*. Available at: https://www.icann.org/ (Accessed: 15th April 2019).

ICANN. (2020) *What Is Privilege Escalation?* Available at: https://www.icann.org/news/blog/what-is-privilege-escalation (Accessed: 28th Janaury 2020).

IDEMA. (2019) *Advanced Format (Af) Technology*. Available at: http://idema.org/?page_id=98 (Accessed: 18th May 2019).

IETF. (1987a) *Domain Names - Implementation and Specification.* IETF: IETF.

IETF. (1987b) *Domains Names - Concepts and Facilities.* IETF: IETF.

IETF. (1993) *Internet Users' Glossary.* IETF: IETF.

IETF. (2007) *Internet Security Glossary, Version 2.* IETF: IETF.

Imperva (2011) *Advanced Persistent Threat: Are You the Next Target?* (Accessed: 18th July 2017).

Imperva (2012) *Assessing the Effectiveness of Antivirus Solutions.* Imperva Available at: https://www.imperva.com/docs/HII_Assessing_the_Effectiveness_of_Antivirus_Solutions.pdf (Accessed: 18th July 2017).

Imperva (2014) *The Non-Advanced Persistent Threat.* Imperva Available at: https://www.imperva.com/docs/HII_The_Non-Advanced_Persistent_Threat.pdf (Accessed: 18th July 2017).

Information-Security-and-Privacy-Advisory-Board. (2012) *Minutes of Meeting October 10, 11, and 12, 2012*. Washington, D.C.: NIST. Available at: http://csrc.nist.gov/groups/SMA/ispab/documents/minutes/2012-10/ispab_meeting-minutes_october-2012.pdf (Accessed: 27th August 2017).

InfoSec-Institute. (2017) *The Seven Steps of a Successful Cyber Attack*. Available at: http://resources.infosecinstitute.com/the-seven-steps-of-a-successful-cyber-attack/ (Accessed: 3rd December 2017).

Insikt-Group (2018a) *Chinese Threat Actor Temp.Periscope Targets Uk-Based Engineering Company Using Russian Apt Techniques.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2018-1113.pdf (Accessed: 20th February 2020).

Insikt-Group (2018b) *Underlying Dimensions of Yemen's Civil War: Control of the Internet.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2018-1128.pdf (Accessed: 20th February 2020).

Insikt-Group (2019) *Talking to Rats: Assessing Corporate Risk by Analyzing Remote Access Trojan Infections.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2019-0314.pdf (Accessed: 20th February 2020).

Insikt-Group and Rapid7 (2019) *Apt10 Targeted Norwegian Msp and Us Companies in Sustained Campaign.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2019-0206.pdf (Accessed: 20th February 2020).

Insikt-Group® (2019a) *Operation Gamework: Infrastructure Overlaps Found between Bluealpha and Iranian Apts.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2019-1212.pdf (Accessed: 20th February 2020).

Insikt-Group® (2019b) *Your Organization's Network Access Is King: Here's What to Do About It.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2019-1030.pdf (Accessed: 20th February 2020).

Insikt-Group® (2020a) *How North Korea Revolutionized the Internet as a Tool for Rogue Regimes.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2020-0209.pdf (Accessed: 11th February 2020).

Insikt-Group® (2020b) *Profiling the Linken Sphere Anti-Detection Browser.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2020-0107.pdf (Accessed: 20th February 2020).

Intel. (2019a) *Intel® Celeron® Processor B820 2m Cache, 1.70 Ghz*. Available at: https://ark.intel.com/content/www/us/en/ark/products/67193/intel-celeron-processor-b820-2m-cache-1-70-ghz.html (Accessed: 29th June 2019).

Intel. (2019b) *Intel® Core™ I5-4690 Processor 6m Cache, up to 3.90 Ghz*. Available at: https://ark.intel.com/content/www/us/en/ark/products/80810/intel-core-i5-4690-processor-6m-cache-up-to-3-90-ghz.html (Accessed: 28th June 2019).

IPA (2013) *System Design Guide for Thwarting Targeted Email Attacks.* IPA IT SECURITY CENTER (ISEC) INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN Available at: https://www.ipa.go.jp/files/000035723.pdf (Accessed: 1st December 2017).

ISACA (2013) *Advanced Persistent Threat Awareness Study Results.* Trend Micro Available at: http://apac.trendmicro.com/cloud-content/apac/pdfs/business/datasheets/wp_apt-survey-report.pdf (Accessed: 9th February 2017).

ISO. (2013) *Information Technology — Security Techniques — Code of Practice for Information Security Controls (Second Edition).* ISO: ISO.

Jackson, M. A. (1975) *Principles of Program Design.* London New York San Francisco: ACADEMIC PRESS INC. (LONDON) LTD.

Jacoby, D. and Jartelius, M. (2013) *Exposing the Security Weakness We Tend to Overlook.* Available at: http://kasperskycontenthub.com/wp-content/uploads/sites/43/vlpdfs/exposing_the_security_weaknesses_we_tend_to_overlook.pdf (Accessed: 18th July 2017).

Jain, V., Gomez, J. and Singh, A. (2014) *A Daily Grind: Filtering Java Vulnerabilities.* Fireeye Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/wp-a-daily-grind-filtering-java-vulnerabilities.pdf (Accessed: 29th Janaury 2017).

Jeun, I., Lee, Y. and Won, D. (2012) 'A Practical Study on Advanced Persistent Threats', *Computer Applications for Security, Control and System Engineering,* 339, pp. 144-152

Jones, R. and Lins, R. (1996) *Garbage Collection. Algorithms for Automatic Dynamic Memory Management..* Chichester: John Wiley & Sons Ltd.

Kadivar, M. (2014) 'Cyber-Attack Attributes', *Technology Innovation Management Review,* 4(11), pp. 22-27.

Kahn, D. (1973) *The Codebreakers.* New York, New York, 10019: The New American Library, Inc.

Kahvedžić, D. and Kechadi, T. (2009) *SAC '09 Proceedings of the 2009 ACM symposium on Applied Computing*. Honolulu, Hawaii New York, NY, USA: ACM.

Kanekal, V. (2013) *Data Reconstruction from a Hard Disk Drive Using Magnetic Force Microscopy.* University of Califormia, San Diego.

Kang, B., Kim, T., Heejun, K., Choi, Y. and Im, E. G. (2012) *RACS '12 Proceedings of the 2012 ACM Research in Applied Computation Symposium* San Antonio, Texas October 23 - 26, 2012. ACM New York, NY, USA: ACM.

Karp, D. A. (2010) *Windows 7 Annoyances.* Sebastopol CA 95472: O'Reilly Media Inc.

Karresand, M. and Shahmehri, N. (2006) *Information Assurance Workshop*. West Point, NY, USA: IEEE.

Kaspersky-Lab (2015) *Carbanak Apt the Great Bank Robbery.* Available at: https://securelist.com/files/2015/02/Carbanak_APT_eng.pdf (Accessed: 18th July 2017).

Kaspersky (2013a) *The 'Icefog' Apt: A Tale of Cloak and Daggers.* Available at: http://kasperskycontenthub.com/wp-content/uploads/sites/43/vlpdfs/icefog.pdf (Accessed: 18th July 2017).

Kaspersky (2013b) *The Miniduke Mystery: Pdf 0-Day Government Spy Assembler 0x29a Micro Backdoor (or 'How Many Cool Words Can You Fit into One Title').* Available at: http://kasperskycontenthub.com/wp-content/uploads/sites/43/vlpdfs/themysteryofthepdf0-dayassemblermicrobackdoor.pdf (Accessed: 18th July 2017).

Kaspersky (2013c) *The Nettraveler (Aka 'Travnet').* Available at: http://kasperskycontenthub.com/wp-content/uploads/sites/43/vlpdfs/kaspersky-the-net-traveler-part1-final.pdf (Accessed: 18th July 2017).

Kaspersky (2013d) *The 'Teamspy' Story - Abusing Teamviewer in Cyberespionage Campaigns.* Available at: https://kasperskycontenthub.com/wp-content/uploads/sites/43/vlpdfs/theteamspystory_final_t2.pdf (Accessed: 18th July 2017).

Kaspersky (2013e) *"Winnti" More Than Just a Game.* Available at: http://kasperskycontenthub.com/wp-content/uploads/sites/43/vlpdfs/winnti-more-than-just-a-game-130410.pdf (Accessed: 18th July 2017).

Kaspersky (2014a) *The Darkhotel Apt a Story of Unusual Activity.* Available at: https://securelist.com/files/2014/11/darkhotel_kl_07.11.pdf (Accessed: 18th July 201).

Kaspersky (2014b) *Energetic Bear — Crouching Yeti.* Available at: https://securelist.com/files/2014/07/EB-YetiJuly2014-Public.pdf (Accessed: 18th July 2017).

Kaspersky (2014c) *The Epic Turla Operation: Solving Some of the Mysteries of Snake/Uroboros.* Available at: https://cdn.securelist.com/files/2014/08/KL_Epic_Turla_Technical_Appendix_20140806.pdf (Accessed: 18th July 2017).

Kaspersky (2014d) *Kaspersky Security Bulletin 2014.* Available at: https://www.securelist.com/files/2014/12/Kaspersky-Security-Bulletin-2014-EN.pdf (Accessed: 18th July 2017).

Kaspersky (2014e) *The Regin Platform Nation-State Ownage of Gsm Networks.* Available at: https://securelist.com/files/2014/11/Kaspersky_Lab_whitepaper_Regin_platform_eng.pdf (Accessed: 18th July 2017).

Kaspersky (2014f) *Unveiling "Careto" - the Masked Apt.* Available at: http://kasperskycontenthub.com/wp-content/uploads/sites/43/vlpdfs/unveilingthemask_v1.0.pdf (Accessed: 18th July 2017).

Kaspersky (2015a) *The Desert Falcons Targeted Attacks.* Available at: https://securelist.com/files/2015/02/The-Desert-Falcons-targeted-attacks.pdf (Accessed: 18th July 2017).

Kaspersky (2015b) *The Duqu 2.0 Technical Details Version: 2.1 (11 June 2015).* Available at: https://securelist.com/files/2015/06/The_Mystery_of_Duqu_2_0_a_sophisticated_cyberespionage_actor_returns.pdf (Accessed: 18th July 2017).

Kaspersky (2015c) *Equation Group: Questions and Answers.* Available at: https://securelist.com/files/2015/02/Equation_group_questions_and_answers.pdf (Accessed: 18th July 2017).

Kaspersky (2016a) *The Project Sauron Apt.* Available at: https://cdn.securelist.com/files/2016/07/The-ProjectSauron-APT_research_KL.pdf (Accessed: 18th July 2017).

Kaspersky (2016b) *Threat Intelligence Report for the Telecommunications Industry.* Available at: https://securelist.com/files/2016/08/Kaspersky_Telecom_Threats_2016.pdf (Accessed: 18th July 2017).

Kaspersky (2016c) *The Xdedic Marketplace Version: 1.0 (15 June 2016).* Available at: https://securelist.com/files/2016/06/xDedic_marketplace_ENG.pdf (Accessed: 18th July 2017).

Kaspersky (2017a) *From Shamoon to Stonedrill Wipers Attacking Saudi Organizations and Beyondversion 1.05 2017-03-07.* Available at: https://securelist.com/files/2017/03/Report_Shamoon_StoneDrill_final.pdf (Accessed: 18th July 2017).

Kaspersky (2017b) *Lazarus under the Hood.* Available at: https://securelist.com/files/2017/04/Lazarus_Under_The_Hood_PDF_final.pdf (Accessed: 18th July 2017).

Katsuki, T. (2012) *Crisis: The Advanced Malware.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/crisis_the_advanced_malware.pdf (Accessed: 29th January 2017).

Kazanciyan, R. and Hastings, M. (2014) *Investigating Powershell Attacks: Black Hat USA 2014.* Fireeye Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/solutions/pdfs/wp-lazanciyan-investigating-powershell-attacks.pdf (Accessed: 29th January 2017).

Kellermann, T. (2012) *Peter the Great Versus Sun Tzu.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/spotlight-articles/op_kellermann_peter-the-great-vs-sun-tzu.pdf (Accessed: 31st January 2017).

Kelly, K. (2017) *Indicators of Compromise Teslacrypt Malware.* SANS Available at: https://www.sans.org/reading-room/whitepapers/awareness/indicators-compromise-teslacrypt-malware-37622 (Accessed: 19th February 2017).

Kernighan, B. W. and Ritchie, D. M. (1988) *The C Programming Language.* Fourty-ninth printing edn. Upper Saddle River, NJ 07458: Prentice Hall.

Kharouni, L. (2015) *A Profile of Irs Scammers Behind Tax Fraud.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-a-profile-of-irs-scammers.pdf (Accessed: 9th February 2017).

Kharouni, L., Hacquebord, F., Huq, N., Gogolinski, J., Mercês, F., Alfred, R. and Otis, D. (2014) *Operation Pawn Storm: Using Decoys to Evade Detection.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-operation-pawn-storm.pdf (Accessed: 31st January 2017).

Kientzle, T. (1995) *The Working Programmer's Guide to Serial Protocols.* Scottsdale, Arizona, 85260: Coriolis Group Books.

Kim, Y., Lee, S. and Hong, D. (2008) *e-Forensics '08 Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop* Adelaide, Australia Adelaide, Australia: ICST.

Kim, Y., Moon, J., Cho, S., Park, M. and Han, S. (2014) *International Conference on Availability, Reliability, and Security*. Krakow, Poland: IEEE.

King, N. and Horrocks, C. (2010) *Interviews in Qualitative Research.* London: Sage. P.

Kiwiaa, D., Dehghantanhaa, A., Choo, K.-K. R. and Jim, S. (2018) 'A Cyber Kill Chain Based Taxonomy of Banking Trojans for Evolutionarycomputational Intelligence', *Journal of Computational Science,* 27(July 2018), pp. 394–409.

Knuth, D. E. (1973) *The Art of Computer Programming Volume 3 Sorting and Searching.* Vol. 3. Philippines: Addison-Wesley Publishing Company, Inc.

Knuth, D. E. (1997) *The Art of Computer Programming Volume 1 Fundamental Algorithms.* Vol. 1. United States of America: Addison Wesley Longman.

Knuth, D. E., Morris, J., James H. and Pratt, V. R. (1977) 'Fast Pattern Matching in Strings', *SIAM Journal on Computing,* 6(2), pp. 323-350.

Krivtsova, I. E., Lebedev, I. S. and Salakhutdinova, K. I. (2017) 'Identification of Executable Files on the Basis of Statistical Criteria', *2017 20th Conference of Open Innovations Association (FRUCT)*. St. Petersburg, Russia IEEE: IEEE. pp. 202 - 208. Available at: https://ieeexplore.ieee.org/abstract/document/8071312 (Accessed: 20th December 2019).

Kruse, P., Hacquebord, F. and McArdle, R. (2012) *Threat Report: W32.Tinba (Tinybanker) the Turkish Incident.* Trend Micro Available at: http://apac.trendmicro.com/cloud-content/apac/pdfs/security-intelligence/white-papers/wp_w32-tinba-tinybanker.pdf (Accessed: 9th February 2017).

Kuczma, K. (2019) *Microsoft Targeted by 8 of 10 Top Vulnerabilities in 2018.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2019-0319.pdf (Accessed: 11th February 2020).

Kuczma, K. and Manalo, B. (2020) *Criminal Underground Contiues to Target Microsoft Products in Top 2019 Exploit Vulnerabilities List.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2020-0204.pdf (Accessed: 20th February 2020).

Kuhn, T. S. (1977) *The Essential Tension. Selected Studies in Scientific Tradition and Change.* Chicago: The University of Chicago, Chicago.

Kuhn, T. S. (1996) *The Structure of Scientific Revolutions.* Chicago, IL: University of Chicago Press.

kusano (2015) *Ntfsdump* GitHub Available at: https://github.com/kusano/ntfsdump (Accessed: 24th May 2018).

LaBarge, R., Mazzuchi, T. A. and Sarkani, S. (2014) 'An Automated System for Rapid and Secure Device Sanitization', *Computers & Security,* 42, pp. 77-91.

Labs, F. (2020) *Virus W32/Gpigeon.Du!Tr.Bdr*. Available at: https://fortiguard.com/encyclopedia/virus/434903 (Accessed: 25th October 2020).

Landage, J. and Wankhade, P. M. P. (2013) 'Malware and Malware Detection Techniques: A Survey', *International Journal of Engineering Research & Technology (IJERT),* 2(12), pp. 61-68.

Langill, J. T. (2014) *Defending against the Dragonfly Cyber Security Attacks.* Belden (Accessed: 4th February 2017).

Larman, C. and Basili, V. R. (2003) 'Iterative and Incremental Developments. A Brief History', *IEEE Computer,* 36(June 2003),

LaserMedia-AS (2007) *Advance Pe Viewer - Nikpeviewer 0.4v (Beta)* CodeDebug CodeDebug Available at: http://www.codedebug.com/php/Products/Products_NikPEViewer_12v.php (Accessed: 19th November 2019).

LaserMedia-AS (2020) *Nikpeviewer*: GitHub, Inc. Available at: https://github.com/YajS/NikPEViewer (Accessed: 7th September 2020).

Layder, D. (1998) *Sociological Practice.* London: Sage.

Lee, D. C., Crowley, P. J., Baer, J.-L., Anderson, T. E. and Bershad, B. N. (1998) *25th International Symposium on Computer Architecture*. Barcelona, Spain: IEEE.

Lee, R. Y. (2013) *Software Engineering: A Hands-on Approach.* Mount Pleasant MI, USA: Atlantis Press.

Lee, T., Ahl, I. and Hanzlik, D. (2014) *The Little Malware That Could: Detecting and Defeating the China Chopper Web Shell.* Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-china-chopper.pdf (Accessed: 29th January 2017).

Lees, C. (2013) 'Determining Removal of Forensic Artefacts Using the Usn Change Journal', *Digital Investigation,* 10(4), pp. 300-310.

Lemay, A., Calvet, J., Menet, F. and Fernandez, J. M. (2018) 'Survey of Publicly Available Reports on Advanced Persistent Threat Actors', *Computers & Security,* 72(January 2018), pp. 26–59.

Li, W.-J., Ke, W., Stolfo, S. J. and Herzog, B. (2005) *Proceedings from the Sixth Annual IEEE*. West Point, NY, USA, USA: IEEE.

Link, R. and Sancho, D. (2011) 'Lessons Learned While Sinkholing Botnets - Not as Easy as It Looks!', *Virus Bulletin Conference October 2011*: Virus Bulletin. pp. 106-110.

Little, R. G. (2002) 'Controlling Cascading Failure: Understanding the Vulnerabilities of Interconnected Infrastructures', *Journal of Urban Tech,* 9(1), pp. 109-123.

Lodge, D. w. W., Nigel, Edited by. (2000) *Modern Criticism and Theory.* Harlow: Pearson Education Ltd,.

LogRhythm. (2013) *Advanced Persistent Threats Blueprint for Detection & Response.* LogRhythm.

LogRhythm (2014) *The Apt Lifecycle and Its Log Trail.* LogRhythm Available at: http://www.softbox.co.uk/pub/logrhythm_the_apt_lifecycle_white_paper.pdf (Accessed: 9th February 2017).

Loman, M. (2019) *How Ransomware Attacks.* Sophos Available at: https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophoslabs-ransomware-behavior-report.pdf (Accessed: 8th February 2020).

Los-Alamos-National-Laboratory. (2017) *Scalable Clusters Make Hpc R&D Easy as Raspberry Pi* Available at: https://www.lanl.gov/discover/news-release-archive/2017/November/1113-raspberry-pi.php (Accessed: 29th June 2019).

LSoft-Technologies-Inc. (2020a) *Ntfs Partition Boot Sector*. Available at: https://www.ntfs.com/ntfs-partition-boot-sector.htm (Accessed: 23rd April 2020).

LSoft-Technologies-Inc. (2020b) *Ntfs Partition Boot Sector*. Available at: https://www.ntfs.com/ntfs_basics.htm (Accessed: 3rd October 2020).

Lui, M. and Baldwin, T. (2014) 'Accurate Language Identification of Twitter Messages', *14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics. pp. 17–25. Available at: http://www.aclweb.org/anthology/W14-1303 (Accessed: 9th February 2017).

Lunt, B. (2004) *X86 Assembly Language Faq - General Part Iii*. Available at: http://www.faqs.org/faqs/assembly-language/x86/general/part3/ (Accessed: 23rd April 2020).

Luo, S. and Yan, P. (2014) *Fake Apps: Feigning Legitimacy.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-fake-apps.pdf (Accessed: 9th February 2017).

Lynch, W. C. (1972) 'Do Disk Arms Move? ', *ACM SIGMETRICS Performance Evaluation Review,* 1(3), pp. 3 - 16.

M86-Security (2010) *Advanced Persistent Threats.* M85 Security Available at: http://resources.idgenterprise.com/original/AST-0023216_advanced_persistent_threats_report.pdf (Accessed: 3rd March 2017).

Macaulay, S. (2014) *Killing the Rootkit - Perfect Physical Memory Process Detection.* Virus Bulletin Available at: https://www.virusbulletin.com/uploads/pdf/conference_slides/2014/Macaulay-VB2014.pdf (Accessed: 17th March 2017).

Maczuba, J. (2016) *See Everything, Fear Nothing Threat Solution Series: Spear Phishing.* RSA Available at: https://www.rsa.com/content/dam/rsa/static/usecase/collateral/so-ASOC-use-case-spearphishing.pdf (Accessed: 3rd November 2017).

Madiant-Consulting (2016) *M-Trends 2016.* Available at: https://marketingcentral.fireeye.com/ResourceFiles/9c255c9d-3a6a-4086-a000-bf614d811fee.pdf (Accessed: 4th June 2016).

Mahajan, R. (2016) *IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2016),*. Jaipur, India December 23-25, 2016.

Mahajan, R., Singh, D. M. and Miglani, S. (2014) 'Ads:Protecting Ntfs from Hacking', *International Conference on Recent Advances and Innovations in*

Engineering (ICRAIE-2014). Jaipur, India May 09-11, 2014. IEEE. Available at: https://ieeexplore.ieee.org/document/6909325/ (Accessed: 24th April 2018).

Malenfant, J., Jacques, M. and Demers, F.-N. (1996) *A Tutorial on Behavioral Reflection and Its Implementation.* Departement d'informatique et recherche operationnelle,

Universite de Montreal Available at: http://scholar.google.co.uk/scholar_url?url=http%3A%2F%2Fwww.academia.edu%2Fdownload%2F39203706%2F0deec53a9b5e7898fd000000.pdf&hl=en&sa=T&oi=ggp&ct=res&cd=0&d=10103526075438097956&ei=R6A-Xp6YKNKbmAH4gr-gDw&scisig=AAGBfm3jBA79cLb7o2IqvbuuyEK4sOmcFA&nossl=1&ws=1920x865&at=A%20tutorial%20on%20behavioral%20reflection%20and%20its%20implementation (Accessed: 8th February 2020).

MalwareBytes (2016) *What Is Malvertising?* Available at: https://www.malwarebytes.com/pdf/infographics/MalvertisingInfo.pdf (Accessed: 6th February 2017).

Malwarebytes (2018a) *Cybercrime Tactics and Techniques: Q2 2018.* Available at: https://resources.malwarebytes.com/files/2018/07/Malwarebytes_Cybercrime-Tactics-and-Techniques-Q2-2018.pdf (Accessed: 11th August 2020).

MalwareBytes. (2018b) *Malwarebytes Software License Agreement*. Available at: https://www.malwarebytes.com/eula/ (Accessed: 23rd June 2018).

Malwarebytes. (2019) *Malwarebytes Privacy Policy*. Available at: https://www.malwarebytes.com/privacy/ (Accessed: 4th Janaury 2020).

MalwareFox. (2018) *How Antivirus Works?* Available at: https://www.malwarefox.com/how-antivirus-works/ (Accessed: 23rd June 2018).

Mandiant (2010) *M Trends [the Advanced Persistent Threat]*. Mandiant Available at: https://www.slideshare.net/FireEyeInc/mtrends-2010-the-advanced-persistent-threat (Accessed: 5th June 2017).

Mandiant (2013) *Apt1: Exposing One of China's Cyber Espionage Units.* Mandiant Available at: https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf (Accessed: 20th March 2017).

Mandiant (2015a) *M-Trends® 2015: A View from the Front Lines.* (Accessed: 4th June 2017).

Mandiant (2015b) *There's Somthing About Wmi.* Available at: https://files.sans.org/summit/dfir-prague-summit-2015/PDFs/Theres-Something-about-WMI-Christopher-Glyer-and-Devon-Kerr.pdf (Accessed: 27th February 2017).

Mandiant (2017) *M-Trends® a View from the Front Lines 2017.* Milpitas, CA 95035 Available at: http://files.shareholder.com/downloads/AMDA-254Q5F/5625959308x0x938351/665BA6A3-9573-486C-B96F-80FA35759E8C/FEYE_rpt-mtrends-2017_FINAL2.pdf (Accessed: 4th June 2017).

Marchetti, M., Pierazzi, F., Colajanni, M. and Guido, A. (2016) 'Analysis of High Volumes of Network Traffic for Advanced Persistent Threat Detection', *Computer Networks,* 109, Part 2, pp. 127-141. DOI: https://doi.org/10.1016/j.comnet.2016.05.018.

Marosi, A. (2016) *Cryptomining Malware on Nas Servers.* Sophos Available at: https://www.sophos.com/en-us/medialibrary/PDFs/technical%20papers/Cryptomining-malware-on-NAS-servers.pdf (Accessed: 9th February 2017).

Martini, A. I., Zaharis, A. and Ilioudis, C. (2008) *Third International Annual Workshop on Digital Forensics and Incident Analysis, 2008. WDFIA '08. .* Malaga, Spain.

Maslow, A. H. (2016) *A Theory of Human Motivation.* Mid West Journal Press.

Mauthner, M. L., Birch, M., Jessop, J. and Tina, M. (2002) *Ethics in Qualitative Research.* London: Sage.

McAfee-Labs-Threat-Advisory (2014) *Careto Attack – the Mask.* McAfee Available at: https://kc.mcafee.com/resources/sites/MCAFEE/content/live/PRODUCT_DOCUMENTATION/25000/PD25037/en_US/McAfee_Labs_Threat_Advisory_Careto_Attack_The%20Mask_3.pdf (Accessed: 14th February 2017).

Mcafee-Labs (2012) *Pws-Gauss.* Mcafee-Labs Available at: Unknown (Accessed: 3rd March 2017).

McAfee-Labs (2013) *Mcafee Labs Threats Report: Fourth Quarter 2013.* McAfee Available at: https://www.mcafee.com/au/resources/reports/rp-quarterly-threat-q4-2013.pdf (Accessed: 29th January 2017).

McAfee-Labs (2015) *Mcafee Labs Threats Report: November 2015.* McAfee Available at: https://www.mcafee.com/us/resources/reports/rp-quarterly-threats-nov-2015.pdf (Accessed: 29th January 2017).

McAfee-Labs (2016) *Mcafee Labs Threats Report: June 2016.* McAfee Available at: https://www.mcafee.com/us/resources/reports/rp-quarterly-threats-may-2016.pdf (Accessed: 29th January 2017).

McAfee (2016) *Protecting against Firmware and Bios Manipulation.* McAfee Available at: Unknown (Accessed: 19th February 2017).

McAfee (2018a) *Js/Nemucod.* McAfee: McAfee. Available at: https://kc.mcafee.com/resources/sites/MCAFEE/content/live/CORP_KNOWLEDGEBASE/91000/KB91905/en_US/McAfee_Labs_Threat_Advisory_JS-Nemucod.pdf (Accessed: 24th Jnauary 2020).

McAfee. (2018b) *Terms of Service Agreement*. Available at: https://home.mcafee.com/supportpages/termsandconditions.aspx (Accessed: 23rd June 2018).

McAfee. (2019) *Mcafee Privacy Notice Effective Date: August 7, 2018*. Available at: https://www.mcafee.com/enterprise/en-gb/about/legal/privacy.html#kinds_information (Accessed: 19th May 2019).

McAfee®-Foundstone®-Professional-Services (2017) *Mcafee Labs Threat Advisory W32/Disttrack.* McAfee Available at: https://kc.mcafee.com/resources/sites/MCAFEE/content/live/PRODUCT_DOCUMENTATION/23000/PD23936/en_US/McAfee_Labs_Threat_Advisory-W32-DistTrack.pdf (Accessed: 18th February 2017).

McAfee®-Foundstone®-Professional-Services and McAfee-Labs™ (2011) *Global Energy Cyberattacks: "Night Dragon".* McAfee Available at: https://www.mcafee.com/us/resources/white-papers/wp-global-energy-cyberattacks-night-dragon.pdf (Accessed: 29th January 2017).

McKee. (2003) *Textual Anlysis: A Beginners Guide.* London: Sage.

McNamee, M. (2005) *Positivism, Popper and Paradigms: An Introductory Essay in the Philosophy of Science', in Mcnamee. M. (Ed) Philosophy and the Sciences of Exercise, Health and Sport.* London: Routledge.

Merck. (2018) *Mechanism of Action*. Available at: https://www.sigmaaldrich.com/life-science/biochemicals/biochemical-products.html?TablePage=14837959 (Accessed: 30th March 2018).

Microsoft. (2009a) *How Basic Disks and Volumes Work*. Available at: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc739412(v=ws.10) (Accessed: 14th December 2019).

Microsoft. (2009b) *How Ntfs Works*. Available at: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc781134(v=ws.10)#multiple-data-streams (Accessed: 12th May 2018).

Microsoft. (2009c) *Wmic - Take Command-Line Control over Wmi*. Available at: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/bb742610(v=technet.10) (Accessed: 9th February 2020).

Microsoft. (2010) *Userinit*. Available at: https://technet.microsoft.com/en-gb/library/cc939862.aspx (Accessed: 10th February 2020).

Microsoft. (2011) *Registry Key "Wow6432node" May Be Listed in System Registry on 32bit (X86) Version of Windows 7*. Available at: https://support.microsoft.com/en-us/help/2582176/registry-key-wow6432node-may-be-listed-in-system-registry-on-32bit-x86 (Accessed: 30th November 2018).

Microsoft. (2012) *Windows Registry Information for Advanced Users*. Available at: https://support.microsoft.com/en-us/help/256986/windows-registry-information-for-advanced-use (Accessed: 25th November 2018).

Microsoft. (2013) *Diy Supercomputing: How to Build a Small Windows Hpc Cluster*. Available at: https://social.technet.microsoft.com/wiki/contents/articles/2539.diy-supercomputing-how-to-build-a-small-windows-hpc-cluster.aspx (Accessed: 28th June 2019).

Microsoft. (2015) *A History of Windows*. Available at: https://web.archive.org/web/20160611182917/http://windows.microsoft.com/en-in/windows/history#T1=era0 (Accessed: 15th October 2019).

Microsoft. (2016) *Updating Drive Firmware*. Available at: https://docs.microsoft.com/en-us/windows-server/storage/update-firmware (Accessed: 23rd January 2020).

Microsoft. (2017a) *Address Resolution Protocol (Arp) Cache Functions*. Available at: https://support.microsoft.com/en-us/help/99150/address-resolution-protocol-arp-cache-functions (Accessed: 24th November 2017).

Microsoft. (2017b) *Changes to Service Host Grouping in Windows 10*. Available at: https://docs.microsoft.com/en-us/windows/application-management/svchost-service-refactoring (Accessed: 8th February 2020).

Microsoft. (2017c) *Connect to Another Computer Using Remote Desktop Connection*. Available at: https://support.microsoft.com/en-us/help/17463/windows-7-connect-to-another-computer-remote-desktop-connection (Accessed: 19th May 2017).

Microsoft. (2017d) *Cve-2017-11937 | Microsoft Malware Protection Engine Remote Code Execution Vulnerability*. Available at: https://portal.msrc.microsoft.com/en-us/security-guidance/advisory/CVE-2017-11937 (Accessed: 6th January 2017).

Microsoft. (2017e) *Cve-2017-11940 | Microsoft Malware Protection Engine Remote Code Execution Vulnerability*. Available at: https://portal.msrc.microsoft.com/en-us/security-guidance/advisory/CVE-2017-11940 (Accessed: 6th January 2017).

Microsoft. (2017f) *Database of Installed Services*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/ms682544(v=vs.85).aspx (Accessed: 3rd December 2017).

Microsoft. (2017g) *Description of Control Panel (.Cpl) Files*. Available at: https://support.microsoft.com/en-us/help/149648/description-of-control-panel--cpl-files (Accessed: 22nd September 2017).

Microsoft. (2017h) *Dir*. Available at: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/dir (Accessed: 16th May 2017).

Microsoft. (2017i) *Disable Registry Editing Tools*. Available at: https://docs.microsoft.com/en-us/previous-versions/ms811961(v=msdn.10) (Accessed: 13th February 2020).

Microsoft. (2017j) *Dynamic-Link Library Search Order*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/ms682586(v=vs.85).aspx (Accessed: 1st August 2017).

Microsoft. (2017k) *Dynamic-Link Library Security*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/ff919712(v=vs.85).aspx (Accessed: 1st August 2017).

Microsoft. (2017l) *Enumerating Registry Subkeys*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/ms724256(v=vs.85).aspx (Accessed: Unknown).

Microsoft. (2017m) *Exporting from a Dll*. Available at: https://msdn.microsoft.com/en-us/library/z4zxe9k8.aspx (Accessed: 29th October 2017).

Microsoft. (2017n) *Exporting Functions from a Dll by Ordinal Rather Than by Name*. Available at: https://msdn.microsoft.com/en-us/library/e7tsx612.aspx (Accessed: 29th October 2017).

Microsoft. (2017o) *Hklm\System\Currentcontrolset\Control Registry Tree*. Available at: https://docs.microsoft.com/en-us/windows-hardware/drivers/install/hklm-system-currentcontrolset-control-registry-tree (Accessed: 25th November 2018).

Microsoft. (2017p) *How to Use the Regsvr32 Tool and Troubleshoot Regsvr32 Error Messages*. Available at: https://support.microsoft.com/en-us/help/249873/how-to-use-the-regsvr32-tool-and-troubleshoot-regsvr32-error-messages (Accessed: 26th October 2017).

Microsoft. (2017q) *Icons*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/ms646973(v=vs.85).aspx (Accessed: 2nd October 2017).

Microsoft. (2017r) *Iexpress Wizard Command-Line Options*. Available at: https://docs.microsoft.com/en-us/internet-explorer/ie11-ieak/iexpress-command-line-options (Accessed: 23rd June 2019).

Microsoft. (2017s) *Introduction to Windows Service Applications*. Available at: https://msdn.microsoft.com/en-us/library/d56de412(v=vs.110).aspx (Accessed: 24th May 2017).

Microsoft. (2017t) *Kernel Extended Attributes*. Available at: https://docs.microsoft.com/en-us/windows-hardware/drivers/ifs/kernel-extended-attributes (Accessed: 29th October 2017).

Microsoft. (2017u) *Killtimer Function*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/ms644903(v=vs.85).aspx (Accessed: 16th June 2017).

Microsoft. (2017v) *Lsa Authentication Model*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/aa378327(v=vs.85).aspx (Accessed: 1st October 2017).

Microsoft. (2017w) *Microsoft Smb Protocol and Cifs Protocol Overview*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/aa365233(v=vs.85).aspx (Accessed: 5th November 2017).

Microsoft. (2017x) *Mutex Class*. Available at: https://msdn.microsoft.com/en-us/library/system.threading.mutex(v=vs.110).aspx (Accessed: 1st April 2017).

Microsoft. (2017y) *Overview of Windows Components*. Available at: https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/overview-of-windows-components (Accessed: 21st August 2018).

Microsoft. (2017z) *Pdb Files (C++)*. Available at: https://msdn.microsoft.com/en-us/library/yd4f8bd1(v=vs.90).aspx (Accessed: 25th October 2017).

Microsoft. (2017aa) *Registry*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/ms724871(v=vs.85).aspx (Accessed: 13th May 2017).

Microsoft. (2017ab) *Run and Runonce Registry Keys*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/aa376977(v=vs.85).aspx (Accessed: 13th May 2017).

Microsoft. (2017ac) *Security Group Commands*. Available at: https://docs.microsoft.com/en-us/windows-hardware/drivers/storage/security-group-commands (Accessed: 23rd November 2018).

Microsoft. (2017ad) *Settimer Function*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/ms644906(v=vs.85).aspx (Accessed: 16th June 2017).

Microsoft. (2017ae) *Setwindowshookex Function*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/ms644990(v=vs.85).aspx

Microsoft. (2017af) *Sysmon V6.10*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/dd408124(v=vs.85).aspx (Accessed: 29th September 2017).

Microsoft. (2017ag) *Tip: Understand and Control Startup Apps with the System Configuration Utility*. Available at: https://technet.microsoft.com/en-us/library/ee851671.aspx (Accessed: 17th April 2017).

Microsoft. (2017ah) *Using Buffered I/O*. Available at: https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/using-buffered-i-o (Accessed: 10th February 2020).

Microsoft. (2017ai) *Using Software Restriction Policies to Protect against Unauthorized Software*. Available at: https://technet.microsoft.com/en-gb/library/bb457006.aspx (Accessed: 28th August 2017).

Microsoft. (2017aj) *View Hidden Files and Folders in Windows 10* Available at: https://support.microsoft.com/en-us/help/4028316/windows-view-hidden-files-and-folders-in-windows-10 (Accessed: 29th October 2019).

Microsoft. (2017ak) *Windows Lifecycle Fact Sheet*. Available at: https://support.microsoft.com/en-us/help/13853/windows-lifecycle-fact-sheet (Accessed: 27th September 2017).

Microsoft. (2017al) *Windows Management Instrumentation*. Available at: https://msdn.microsoft.com/en-us/library/aa394582(v=vs.85).aspx (Accessed: 19th April 2017).

Microsoft. (2018a) *About Mailslots*. Available at: https://docs.microsoft.com/en-us/windows/win32/ipc/about-mailslots (Accessed: 11th August 2020).

Microsoft. (2018b) *About Processes and Threads*. Available at: https://docs.microsoft.com/en-us/windows/win32/procthread/about-processes-and-threads (Accessed: 11th August 2020).

Microsoft. (2018c) *Cabinet Files*. Available at: https://docs.microsoft.com/en-us/windows/win32/msi/cabinet-files (Accessed: 18th August 2020).

Microsoft. (2018d) *Change Journals*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/aa363798(v=vs.85).aspx (Accessed: 25th April 2018).

Microsoft. (2018e) *Cipher.Exe Security Tool for the Encrypting File System*. Available at: https://support.microsoft.com/en-us/help/298009/cipher-exe-security-tool-for-the-encrypting-file-system (Accessed: 16th December 2018).

Microsoft. (2018f) *Clsid Key*. Available at: https://docs.microsoft.com/en-us/windows/win32/com/clsid-key-hklm (Accessed: 12th February 2020).

Microsoft. (2018g) *Component Object Model (Com)*. Available at: https://docs.microsoft.com/en-us/windows/win32/com/component-object-model--com--portal (Accessed: 12th February 2020).

Microsoft. (2018h) *Creating, Modifying, and Deleting a Change Journal*. Available at: https://docs.microsoft.com/en-us/windows/win32/fileio/creating-modifying-and-deleting-a-change-journal (Accessed: 15th October 2019).

Microsoft. (2018i) *Dacls and Aces*. Available at: https://docs.microsoft.com/en-us/windows/win32/secauthz/dacls-and-aces (Accessed: 25th November 2020).

Microsoft. (2018j) *Default Cluster Size for Ntfs, Fat, and Exfat* Available at: https://support.microsoft.com/en-us/help/140365/default-cluster-size-for-ntfs-fat-and-exfat (Accessed: 14th December 2018).

Microsoft. (2018k) *Deviceiocontrol Function*. Available at: https://docs.microsoft.com/en-gb/windows/desktop/api/ioapiset/nf-ioapiset-deviceiocontrol (Accessed: Accessed 25th November 2018).

Microsoft. (2018l) *Forfiles*. Available at: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/forfiles (Accessed: 16th May 2017)).

Microsoft. (2018m) *Hooks*. Available at: https://docs.microsoft.com/en-gb/windows/win32/winmsg/hooks?redirectedfrom=MSDN (Accessed: 13th November 2019).

Microsoft. (2018n) *How to Paste a Rich Text Format String into Word with Visual Basic Automation* Available at: https://support.microsoft.com/en-us/help/258513/how-to-paste-a-rich-text-format-string-into-word-with-visual-basic-aut (Accessed: 24th January 2020).

Microsoft. (2018o) *How to Use Cipher.Exe to Overwrite Deleted Data in Windows Server 2003* Available at: https://support.microsoft.com/en-gb/help/814599/how-to-use-cipher-exe-to-overwrite-deleted-data-in-windows-server-2003 (Accessed: 19th February 2020).

Microsoft. (2018p) *Info: Run, Runonce, Runservices, Runservicesonce and Startup*. Available at: https://support.microsoft.com/en-us/help/179365/info-run-runonce-runservices-runservicesonce-and-startup (Accessed: 2nd November 2018).

Microsoft. (2018q) *Master File Table*. Available at: https://msdn.microsoft.com/en-us/library/bb470206(v=vs.85).aspx (Accessed: 27th April 2018).

Microsoft. (2018r) *Pe Format*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/ms680547(v=vs.85).aspx (Accessed: 31st March 2018).

Microsoft. (2018s) *Pipes*. Available at: https://docs.microsoft.com/en-us/windows/win32/ipc/pipes (Accessed: 11th August 2020).

Microsoft. (2018t) *Process Security and Access Rights*. Available at: https://docs.microsoft.com/en-us/windows/win32/procthread/process-security-and-access-rights (Accessed: 25th November 2020).

Microsoft. (2018u) *Registry Element Size Limits*. Available at: http://msdn.microsoft.com/en-us/library/ms724872(VS.85).aspx (Accessed: 17th February 2018).

Microsoft. (2018v) *Starting and Stopping the Wmi Service*. Available at: https://msdn.microsoft.com/en-us/library/aa826517(v=vs.85).aspx (Accessed: 4th April 2018).

Microsoft. (2018w) *Volume Shadow Copy Service*. Available at: https://msdn.microsoft.com/en-us/library/windows/desktop/bb968832(v=vs.85).aspx (Accessed: 27th April 2018).

Microsoft. (2018x) *Zwxxx / Ntxxx Routines*. Available at: https://msdn.microsoft.com/en-us/library/windows/hardware/ff567122(v=vs.85).aspx (Accessed: 27th April 2019).

Microsoft. (2019a) *1.1 Glossary*. Available at: https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-efsr/230807ac-20be-494f-86e3-4c8ac23ea584? (Accessed: 18th Auguest 2020).

Microsoft. (2019b) *Azure Data Security and Encryption Best Practices*. Available at: https://docs.microsoft.com/en-us/azure/security/fundamentals/data-encryption-best-practices (Accessed: 8th March 2020).

Microsoft. (2019c) *Description of the Windows Registry Checker Tool (Scanreg.Exe)*. Available at: https://support.microsoft.com/en-us/help/183887/description-of-the-windows-registry-checker-tool-scanreg-exe (Accessed: 16th february 2020).

Microsoft. (2019d) *Downloads*. Available at: https://visualstudio.microsoft.com/downloads/ (Accessed: 15th November 2019).

Microsoft. (2019e).*Net Framework Guide*. Available at: https://docs.microsoft.com/en-us/dotnet/framework/ (Accessed: 15th February 2020).

Microsoft. (2019f) *Process Monitor V3.53*. Available at: https://docs.microsoft.com/en-gb/sysinternals/downloads/procmon (Accessed: 16th February 2020).

Microsoft. (2019g) *Winsock2.H Header*. Available at: https://docs.microsoft.com/en-us/windows/win32/api/winsock2/ (Accessed: 2nd February 2020).

Microsoft. (2019h) *Winsock.H Header*. Available at: https://docs.microsoft.com/en-us/windows/win32/api/winsock/ (Accessed: 2nd Febuary 2020).

Microsoft. (2020) *Set-ExecutionPolicy*. Available at: https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.security/set-executionpolicy?view=powershell-7 (Accessed: 6th Februariy 2020).

Microsoft. (Unknown) *Fsctl_Get_Volume_Bitmap Control Code*. Available at: https://docs.microsoft.com/en-gb/windows/desktop/api/ioapiset/nf-ioapiset-deviceiocontrol (Accessed: Accessed 25th November 2018).

Miller, M. and Zaveri, K. (2016) *Large Scale Incident - Leveraging Encase for Advanced Data Recovery*. Ernst & Young (Accessed: 6th February 2017).

Mitre. (2015-2019) *Dll Side-Loading* Available at: https://attack.mitre.org/techniques/T1073/ (Accessed: 28th January 2020).

Mitre. (2018) *Introduction and Overview*. Available at: https://attack.mitre.org/wiki/Introduction_and_Overview (Accessed: 7th October 2018).

Mitre. (2019a) *Mitre Att&Ck Mapping.* Recorded Future.

Mitre. (2019b) *Registry Run Keys / Startup Folder* Available at: https://attack.mitre.org/wiki/Technique/T1060 (Accessed: 16th February 2020).

Molinyawe, M., Hariri, A.-A. and Spelman, J. (2016) *$Hell on Earth: From Browser to System Compromise.* Trend Micro Available at: http://documents.trendmicro.com/assets/pdf/shell-on-earth.pdf (Accessed: 16th February 2017).

Morgan, T. D. (2008) 'Recovering Deleted Data from the Windows Registry', *DFRWS DIGITAL FORENSIC RESEARCH CONFERENCE*. Baltimore, MD. United States: DFRWS. Available at: http://www.dfrws.org/sites/default/files/session-files/paper-recovering_deleted_data_from_the_windows_registry.pdf (Accessed: 17th February 2008).

Moriuchi, P. (2018) *North Korea's Ruling Elite Adapt Internet Behavior to Foreign Scrutiny.* Recorded Future Available at: https://go.recordedfuture.com/hubfs/reports/cta-2018-0425.pdf (Accessed: 20th February 2020).

Mossburg, E., Fancher, J. D. and John, G. (2016) 'The Hidden Costs of an Ip Breach', *Deloitte Review*(19),

Mosuela, L. (2016) *How It Works: Steganography Hides Malware in Image Files.* Available at: https://www.virusbulletin.com/uploads/pdf/magazine/2016/vb201604-Stegoloader.pdf (Accessed: 19th February 2017).

Mulazzani, M., Neuner, S., Kieseberg, P., Huber, M., Schrittwieser, S. and Weippl, E. (2013) *9th IFIP WG 11.9 International Conference onDigital Forensics*. Orlado, FL, USA January 28-30, 2013. IFIP International Federation for Information Processing 2013.

Murdoch, I. (1992) *Metaphysics as a Guide to Morals.* London: Chatto & Windus Ltd.

NASA. (2017) *Disruption Tolerant Networking*. Available at: https://www.nasa.gov/directorates/heo/scan/engineering/technology/txt_dtn.html (Accessed: 14th April 2020).

NCA (2016) *Cyber Crime Assessment 2016 Need for a Stronger Law Enforcement and Business Partnership to Fight Cyber Crime Version 1.2 7 July 2016.* Available at: http://www.nationalcrimeagency.gov.uk/publications/709-cyber-crime-assessment-2016 (Accessed: 12th December 2016).

Netmarketshare. (2017) *Desktop/Laptop Operating System Market Share*. Available at: https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0 (Accessed: 14th February 2017).

Netmarketshare. (2019a) *Desktp/Laptop Operating System Market Share*. Available at: https://netmarketshare.com/operating-system-market-share.aspx?options=%7B%22filter%22%3A%7B%22%24and%22%3A%5B%7B%22deviceType%22%3A%7B%22%24in%22%3A%5B%22Desktop%2Flaptop%22%5D%7D%7D%5D%7D%2C%22dateLabel%22%3A%22Custom%22%2C%22attributes%22%3A%22share%22%2C%22group%22%3A%22platform%22%2C%22sort%22%3A%7B%22share%22%3A-1%7D%2C%22id%22%3A%22platformsDesktop%22%2C%22dateInterval%22%3A%22Monthly%22%2C%22dateStart%22%3A%222019-09%22%2C%22dateEnd%22%3A%222019-09%22%2C%22segments%22%3A%22-1000%22%2C%22pageLength%22%3A1000%2C%22plotKeys%22%3A%5B%7B%22platform%22%3A%22Windows%22%7D%2C%7B%22platform%22%3A%22Mac%20OS%22%7D%2C%7B%22platform%22%3A%22Linux%22%7D%2C%7B%22platform%22%3A%22Chrome%20OS%22%7D%2C%7B%22platform%22%3A%22Unknown%22%7D%5D%7D (Accessed: 29th October 2019).

Netmarketshare. (2019b) *Device Types*. Available at: https://www.netmarketshare.com/report.aspx?options=%7B%22dateLabel%22%3A%22Trend%22%2C%22attributes%22%3A%22share%22%2C%22gro up%22%3A%22deviceType%22%2C%22sort%22%3A%7B%22share%22%3A-1%7D%2C%22id%22%3A%22deviceTypes%22%2C%22dateInterval%22%3A%22Monthly%22%2C%22filter%22%3A%7B%7D%2C%22dateStart%22%3A%222018-10%22%2C%22dateEnd%22%3A%222019-09%22%2C%22segments%22%3A%22-1000%22%7D (Accessed: 29th October 2019).

Neville, A. and Gibb, R. (2013) *Zeroaccess Indepth.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/zeroaccess_indepth.pdf (Accessed: 29th January 2017).

Nietzsche, F. (1968) *The Will to Power.* New York Random House, Inc.

Nigrini, M. (2012) *Benford's Law: Applications for Forensic Accounting, Auditing, and Fraud Detection.* Hoboken, New Jersey: John Wiley and Sons Inc.

Nikiforakis, N., Balduzzi, M., Desmet, L., Piessens, F. and Joosen, W. (2014) *Soundsquatting Uncovering the Use of Homophones in Domain Squatting.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-soundsquatting.pdf (Accessed: 9th February 2017).

Nissim, N., Yahalom, R. and Elovici, Y. (2017) 'Usb-Based Attacks', *Computers & Security,* 70(September 2017), pp. 675–688.

NIST. (2011) *Managing Information Security Risk: Organization, Mission, and Information System View.* NIST.

NIST. (2013) 'Glossary of Key Information Security Terms ', *Glossary of Key Information Security Terms* Vols. NISTIR 7298 Revision 2. Gaithersburg, MD, USA: NIST.

NIST. (2019) *Current Rds Hash Sets Rds Version 2.67 - December 2019 Iso 9660 Images of Rds Cds*. Available at: https://www.nist.gov/itl/ssd/software-quality-group/national-software-reference-library-nsrl/nsrl-download/current-rds (Accessed: 24th January 2020).

no-ip. (2020) *Dynamic Ip Address Got You Down?* Available at: https://www.noip.com/ (Accessed: 18th February 2020).

Normand, E. (1996) 'Single Event Upset at Ground', *IEEE TRANSACTIONS ON NUCLEAR SCIENCE,* 43(6), pp. 2742 - 2750.

Novetta (Unknown) *Operation Blockbuster: Destructive Malware Report.* Novetta Available at: https://www.operationblockbuster.com/wp-content/uploads/2016/02/Operation-Blockbuster-Destructive-Malware-Report.pdf (Accessed: 16th March 2017).

NPR. (2017) *Spy Terms of the Internyet*. Available at: https://www.npr.org/podcasts/452538677/note-to-self

NPR. (2018) *Why Does Amazon Think It Can Solve Health Care?* Available at: https://www.npr.org/podcasts/381443930/future-tense

NPR. (2019a) *Episode 773: Slot Flaw Scofflaws*. Available at: https://www.npr.org/2019/12/04/784799724/episode-773-slot-flaw-scofflaws (Accessed: 773).

NPR. (2019b) *Stopping Key Tech Exports to China Could Backfire, Researchers and Firms Say*. Available at: https://www.npr.org/2019/05/14/722933448/stopping-key-tech-exports-to-china-could-backfire-researchers-and-firms-say

NSS. (2016) *Sample Size Calculator*. Available at: http://www.nss.gov.au/nss/home.nsf/pages/Sample+size+calculator (Accessed: 12th December 2016).

O'Dwyer, K. J. and Malone, D. (2014) *25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014)*. Limerick, Ireland 26-27 June 2014.

O'Brien, D. (2016) *Dridex: Tidal Waves of Spam Pushing Dangerous Financial Trojan.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/dridex-financial-trojan.pdf (Accessed: 29th January 2017).

O'Connor, T. J. (2014) *The Hacker Always Gets Through.* SANS Available at: https://www.sans.org/reading-room/whitepapers/hackers/paper/34550 (Accessed: 18th February 2017).

O'Gorman, G. and McDonald, G. (2012a) *The Elderwood Project.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-elderwood-project.pdf (Accessed: 18th February 2017).

O'Gorman, G. and McDonald, G. (2012b) *Ransomware: A Growing Menace.* Symantec Available at: https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/ransomware-growing-menace-12-en.pdf (Accessed: 29th January 2017).

O'Leary, J., Kimble, J., Vanderlee, K. and Fraser, N. (2017) *Insights into Iranian Cyber Espionage: Apt33 Targets Aerospace and Energy Sectors and Has Ties to Destructive Malware*. Available at: https://www.fireeye.com/blog/threat-research/2017/09/apt33-insights-into-iranian-cyber-espionage.html (Accessed: 1st January 2020).

O'Murchu, L. and Gutierrez, F. P. (2015) *The Evolution of the Fileless Click-Fraud Malware Poweliks.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/evolution-of-poweliks.pdf (Accessed: 7th July 2017).

O'Reilly, K. (2012) *Ethnographic Methods.* London: Routledge.

Offensive-Security. (2018) *Offensive Security's Exploit Database Archive*. Available at: https://www.exploit-db.com/ (Accessed: 13th May 2018).

Okasha, S. (2002) *Philosophy of Science: A Very Short Introduction.* Oxford: Oxford University Press.

OPSWAT. (2019) *Windows Anti-Malware Market Share Report*. Available at: https://metadefender.opswat.com/reports/anti-malware-market-share#!/?date=2019-04-29 (Accessed: 18th May 2019).

Organization-of-Amercian-States (2015) *Report on Cybersecurity and Critical Infrastructure in the Americas.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/reports/critical-infrastructures-west-hemisphere.pdf (Accessed: 9th February 2017).

Oxford-Dictionaries. (2017) *Social Engineering*. Available at: https://en.oxforddictionaries.com/definition/social_engineering (Accessed: 17th October 2017).

Palumbo, P. (2014a) *W32/Regin, Stage #1.* F-Secure Available at: https://www.f-secure.com/documents/996508/1030745/w32_regin_stage_1.pdf (Accessed: 31st January 2017).

Palumbo, P. (2014b) *W64/Regin, Stage #1.* F-Secure Available at: https://www.f-secure.com/documents/996508/1030745/w64_regin_stage_1.pdf (Accessed: 31st January 2017).

Panda-Labs (2014) *Quarterly Report Q2 2014.* Available at: http://www.pandasecurity.com/mediacenter/src/uploads/2014/07/Informe-Trimestral-Q2-2014-EN.pdf (Accessed: 7th February 2017).

Panda-Labs (2016) *Pandalabs' Annual Report 2015.* Available at: http://www.pandasecurity.com/mediacenter/src/uploads/2014/07/Pandalabs-2015-anual-EN.pdf (Accessed: 7th February 2017).

Panda-Security (2015) *Operation "Oil Tanker" the Phantom Menace.* Panda Security Available at: http://www.pandasecurity.com/mediacenter/src/uploads/2015/05/oil-tanker-en.pdf (Accessed: 7th February 2017).

Panda-Security (2017) *Ransomware from the Crysis/Dharma Family Report.* Spain Available at: https://www.pandasecurity.com/mediacenter/src/uploads/2017/11/Ransomware_Crysis-Dharma-en.pdf (Accessed: 8th February 2020).

Panda (2017) *#Wannacry Report.* Available at: https://www.pandasecurity.com/en/mediacenter/src/uploads/2017/05/1705-Informe_WannaCry-v160-en.pdf (Accessed: 25th November 2020).

Patterson, D. A., Gibson, G. and Katz, R. H. (1988) 'A Case for Redundant Arrays of Inexpensive Disks (Raid)', *SIGMOD '88: Proceedings of the 1988 ACM SIGMOD international conference on Management of data*. ChicagoIllinoisUSA June 1988. Association for Computing Machinery, New York, NY, United States. pp. 109–116. Available at: https://dl.acm.org/doi/abs/10.1145/50202.50214

Perez, C. (2002) *Technological Revolutions and Financial Capital. The Dynamics of Bubbles and Golden Ages.* Cheltenham, UK: Edward Elgar Press.

Pernet, C. (2016) *The French Underground under a Shroud of Extreme Caution.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-the-french-underground.pdf (Accessed: 9th February 2017).

Pernet, C. and Lu, K. (2015) *Operation Wollen-Goldfish When Kittens Go Phishing.* Trend Micro Available at: http://www.trendmicro.co.uk/media/wp/operation-woolen-goldfish-whitepaper-en.pdf (Accessed: 9th February 2017).

Pernet, C. and Sela, E. (2015) *The Spy Kittens Are Back: Rocket Kitten 2.* Trend Micro Available at: http://documents.trendmicro.com/assets/wp/wp-the-spy-kittens-are-back.pdf (Accessed: 31st January 2017).

Perriot, F., Ször, P. and Ferrie, P. (2003) *Striking Similarites: Win32/Simile and Metamorphic Virus Code.* Symantec Available at: https://www.symantec.com/avcenter/reference/striking.similarities.pdf (Accessed: 29th January 2017).

Perrow, C. (1999) *Normal Failures* Princeton, New Jersey 08540: Princeton University Press.

Persons, T. M. (2020) *Cost Estimating and Assessment Guide - Gao-20-195g.* Available at: https://www.gao.gov/assets/gao-20-195g.pdf (Accessed: 12th May 2021).

Pietrek, M. (1994) *Peering inside the Pe: A Tour of the Win32 Portable Executable File Format*. Available at: https://msdn.microsoft.com/en-us/library/ms809762.aspx (Accessed: 12th July 2017).

Pietrek, M. (2002) 'An in-Depth Look into the Win32 Portable Executable File Format', *MSDN Magazine*(February 2002),

Pols, P. (2017) *The Unified Kill Chain. Designing a Unified Kill Chain for Analyzing, Comparing and Defending against Cyber Attacks.* Unpublished Master. Cyber Security Academy (CSA).

Ponemon (2016) *2016 Cost of Data Breach Study: Global Analysis.* Available at: https://public.dhe.ibm.com/common/ssi/ecm/se/en/sel03094wwen/SEL03094WWEN.PDF (Accessed: 3rd May 2018).

Ponemon (2018) *2018 Cost of Data Breach Study: Impact of Business Continuity Management.* Traverse City, Michigan 49686 USA Available at: https://www.ibm.com/downloads/cas/AEJYBPWA (Accessed: 11th August 2020).

Ponnambalam, P. (2015) *Measuring Malware Evolution.* San Jose State University.

Pontifícia-Universidade-Católica-do-Rio-de-Janeiro. (2017) *The Programming Language Lua*. Available at: https://www.lua.org/ (Accessed: 1st October 2017).

Pontiroli, S. M. and Martinez, F. R. (2015) *The Tao Of .Net and Powershell Malware Analysis.* Available at: https://cdn.securelist.com/files/2015/10/Pontiroli_Martinez-VB2015-2.pdf (Accessed: 18th July 2017).

Popper, K. R. (1968) *The Logic of Scientific Discovery.* London: Hutchinson & Co Ltd.

Prem, T., Selwin, V. P. and Mohan, A. K. (2017) 'Disk Memory Forensics Analysis of Memory Forensics Frameworks Flow', *International Conference on Innovations in Power and Advanced Computing Technologies [i-PACT2017]*. Vellore, India 21-22 April. pp. 1 - 7. Available at: https://ieeexplore.ieee.org/document/8244977/ (Accessed: 24th April 2018).

Pressman, R. S. (2010) *Software Engineering a Practioner's Approach.* Boston etc. USA: McGraw Hill.

Putnam, L. H. (1978) 'A General Empirical Solution to the Macro Software Sizing and Estimating Problem', *IEEE Transactions on Software Engineering,* 4(4), pp. 345 - 336.

pwc-BAe (2017) *Operation Cloud Hopper.* pwc-BAe Available at: https://www.pwc.co.uk/cyber-security/pdf/cloud-hopper-report-final-v4.pdf (Accessed: 1st December 2017).

Racchi, M., Govoni, S., Lucchelli, A., Capone, L. and Giovagnoni, E. (2016) 'Insights into the Definition of Terms in European Medical Device Regulation', *Expert Review of Medical Devices,* 13(10), pp. 907-917.

Randell, B. (1969) ' A Note on Storage Fragmentation and Program Segmentation', *Communications of the ACM,* 12(7), pp. 365-372.

Rascagneres, P. (2014) *Com Object Hijacking: The Discreet Way of Persistence.* Available at: https://www.gdatasoftware.com/blog/2014/10/23941-com-object-hijacking-the-discreet-way-of-persistence (Accessed: 25th October 2017).

Rattray, G. J. (2001) *The Terrorism Threat and U.S. Government Response: Operational and Organizational Factors.* Vol. March 2001. US Air Force Academy, Colorado: USAF Institute for National Security Studies.

Raymond.cc. (2019) *5 Online Url Scanners to Check If Website Is Malicious* Available at: https://www.raymond.cc/blog/urlvoid-scans-websites-for-viruses-with-multiple-scanning-engines/ (Accessed: 30th June 2019).

Read, H., Xynos, K., Sutherland, I., Davies, G., Houiellebecq, T., Roarson, F. and Blyth, A. (2013) 'Manipulation of Hard Drive Firmware to Conceal Entire Partitions', *Digital Investigation,* 10(4), pp. 281-286.

Reference. (2018) 'What Is the Difference between Living Things and Nonliving Things?', *Reference*. Ask Media Group, LLC.

Combating the Insider Threat at the Fbi: Real World Lessons Learned (2013) Available at: https://media.blackhat.com/us-13/US-13-Reidy-Combating-the-Insider-Threat-At-The-FBI-Slides.pdf (Accessed: 6th October 2018).

Reuters. (2016) *Britain to Spend 1.9 Billion Pounds on Boosting Cyber Defenses*. Available at: http://www.reuters.com/article/us-britain-cyber-idUSKBN12W39K?il=0 (Accessed: 12th December 2016).

Rivera, B. S. and Inocencio, R. U. (2015) *Doing More with Less: A Study of Fileless Infection Attacks.* Virus Bulletin, Trend Micro Available at: https://www.virusbulletin.com/uploads/pdf/conference_slides/2015/RiveraInocencio-VB2015.pdf (Accessed: 22nd September 2017).

Rivera, B. S. and Inocencio, R. U. (2016) 'Doing More with Less: A Study of Fileless Infection Attacks': Trend Micro. Available at: https://www.virusbulletin.com/uploads/pdf/conference/vb2015/RiveraInnocencio-VB2015.pdf (Accessed: 17th March 2017).

Robinson, L. and Keswani, N. (2016) *Security for Business Innovation Council Report #8 Synopsis.* RSA Available at: https://www.rsa.com/content/dam/rsa/PDF/2016/03/h9005-sbic8-syn.pdf (Accessed: 7th February 2017).

Rodionov, E., Matrosov, A. and Harley, D. (2014) 'Bootkits: Past, Present & Future'. Virus Bulletin. Available at: https://www.virusbulletin.com/uploads/pdf/conference/vb2014/VB2014-RodionovMatrosov.pdf (Accessed: 22 September 2017).

Rodionov, E., Matrosov, A. and Harley, D. (2016) *Bootkits: Past, Present & Future.* Virus Bulletin Available at: https://www.slideshare.net/matrosov/vb2014-slides (Accessed: 19th March 2017).

RSA (2011) *Making Sense of Man-in-the-Browser Attacks.* Available at: https://www.rsa.com/content/dam/rsa/PDF/Making_Sense_of_Man_in_the_browser_attacks.pdf (Accessed: 7th February 2017).

RSA (2014) *Shell_Crew.* Available at: https://www.rsa.com/content/dam/en/white-paper/rsa-incident-response-emerging-threat-profile-shell-crew.pdf (Accessed: 9th October 2019).

RSA (2016) *Malicious Protocols: Gh0st Rat.* Available at: https://www.rsa.com/content/dam/rsa/PDF/2016/09/gh0st-rate-use-case.pdf (Accessed: 7th February 2017).

RSM and CSIT (2018) *Uk Cyber Security Sectoral Analysis and Deep-Dive Review.* UK Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/751406/UK_Cyber_Sector_Report_-__June_2018.pdf (Accessed: 30th October 2019).

Rudd, E. M., Rozsa, A., Günther, M. and Boult, T. E. (2019) 'A Survey of Stealth Malware Attacks, Mitigation Measures, and Steps toward Autonomous Open World Solutions', *IEEE Communications Surveys & Tutorials,* 19(2),

Russinovich, M. (2016) *Streams V1.6*. Available at: https://docs.microsoft.com/en-us/sysinternals/downloads/streams (Accessed: 25th April 2018).

Russinovich, M. (2018) *Sdelete V2.02*. Available at: https://docs.microsoft.com/en-us/sysinternals/downloads/sdelete (Accessed: 16th December 2018).

Russinovich, M. E. and Solomon, D. A. (2009) *Windows Internals Fifth Edition.* Redmond, Washington 98052-6399: Microsoft Press.

Russon, R. (2018) *Ntfs*. Available at: https://flatcap.org/linux-ntfs/ntfs/index.html (Accessed: 14th June 2018).

Russon, R. (2018) *Ntfs - Home*. Available at: https://flatcap.org/linux-ntfs/ntfs/files/index.html (Accessed: 3rd October 2020).

Russon, R. and Fledel, Y. (Undated) *Ntfs Documentation*. Unknown.

Saeed, I. A., Selamat, A. and Abuagoub, A. M. A. (2013) 'A Survey on Malware and Malware Detection Systems', *International Journal of Computer Applications,* 67(16),

Sammes, T. and Jenkinson, B. (2007) *Forensic Computing a Practioner's Guide.* London 2010: Springer-Verlag.

Sancho, D., dela Torre, J., Bakuei, M., Villeneuve, N. and McArdle, R. (2012) *Ixeshe: An Apt Campaign.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp_ixeshe.pdf (Accessed: 31st January 2017).

Sancho, D. and Hacquebord, F. (2016) *Operation C-Major: Information Theft Campaign Targets Military Personnel in India.* Available at: http://documents.trendmicro.com/assets/pdf/Indian-military-personnel-targeted-by-information-theft-campaign-cmajor.pdf Micro, T. (Accessed: 16th February 2017).

Sancho, D., Hacquebord, F. and Link, R. (2014) *Finding Holes: Operation Emmental.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-finding-holes-operation-emmental.pdf (Accessed: 1st June 2017).

SANS-DFIR. (2013) *Finding Unknown Malware – Step-by-Step.* SANS.

SANS. (2011) *Cwe/Sans Top 25 Most Dangerous Software Errors*. Available at: https://www.sans.org/top25-software-errors/ (Accessed: 12th December 2016).

Santos, R. (2014) *Poweliks: Malware Hides in Windows Registry*. Available at: http://blog.trendmicro.com/trendlabs-security-intelligence/poweliks-malware-hides-in-windows-registry/ (Accessed: 26th August 2017).

Sardiwal, M., Cannon, V., Fraser, N., Londhe, Y., Richard, N. and O'Leary, J. (2017) *New Targeted Attack in the Middle East by Apt34, a Suspected Iranian Threat Group, Using Cve-2017-11882 Exploit*. Available at:

https://www.fireeye.com/blog/threat-research/2017/12/targeted-attack-in-middle-east-by-apt34.html (Accessed: 1st January 2020).

Satrya, G. B., Cahyani, N. D. W. and Andreta, R. F. (2015) *ICEC '15 Proceedings of the 17th International Conference on Electronic Commerce 2015*. Seoul, Republic of Korea: ACM.

Saunders, M., Lewis, P. and Thornhill, A. (2016) *Research Methods for Business Students. Seventh Edition.* Harlow, England: Pearson Education Limited.

Schenone, M., Dančík, V., Wagner, B. K. and Clemons, P. A. (2013) 'Target Identification and Mechanism of Action in Chemical Biology and Drug Discovery', *Nature Chemical Biology* 9(30th March 2018), pp. 232–240

Schmall, M. (2018) *Heuristic Techniques in Av Solutions: An Overview*. Available at: https://www.symantec.com/connect/articles/heuristic-techniques-av-solutions-overview (Accessed: 23rd June 2018).

Schneier, B. (1996) *Applied Cryptography Protocols, Algorithms, and Source Code in C.* USA: John Wiley & Sons, Inc.

Schneier, B. (1999) *Attack Trees*. Available at: https://www.schneier.com/academic/archives/1999/12/attack_trees.html (Accessed: 4th March 2018).

Schneier, B. (2000) *Secrets and Lies. Digital Security in a Networked World.* Indianapolis, Indiana, USA: Wiley Computer Publishing.

Schulz, K. (2011) *On Being Wrong* (Accessed: 12th December 2016).

Schumpeter, J. A. (2010) *Capitalism, Socialism and Democracy.* London: Routledge.

Seagate. (Unknown) *Does My Drive Need a Firmware Update?* Available at: https://www.seagate.com/gb/en/support/kb/does-my-drive-need-a-firmware-update-206091en/ (Accessed: 23rd January 2020).

Selvaraj, K. and Gutierrez, N. F. (2010) *The Rise of Pdf Malware.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the_rise_of_pdf_malware.pdf (Accessed: 29th January 2017).

Settle, A., Griffin, N. and Toro, A. (2016) *Monsonn - Analysis of an Apt Campaign Espionage and Data Loss under the Cover of Current Affairs.* Available at: https://www.forcepoint.com/sites/default/files/resources/files/forcepoint-security-labs-monsoon-analysis-report.pdf (Accessed: 9th March 2017).

Seymour, T. and Hussein, S. (2014) 'The History of Project Management', *International Journal of Management & Information Systems,* 18(Third Fourth 2014),

Shahzad, R. K., Haider, S. I. and Lavesson, N. (2010) *International Conference on Availability, Reliability, and Security*: IEEE.

Shakespeare, W. (1599) *Henry V.*

Shannon, C. E. (1949) 'Communication Theory of Secrecy Systems ', *The Bell System Technical Journal* 28(4), pp. 656 - 715.

Shaw, A. (2014) *ICSE Companion 2014: Companion Proceedings of the 36th International Conference on Software Engineering*. Hyderabad, India ACM.

Shekhar, S. and Kumar, U. (2016) 'Review of Various Software Cost Estimation Techniques', *International Journal of Computer Applications (0975 – 8887),* 141 – No.11, Sherstobitoff, R., Liba, I. and Walter, J. (2013) *Dissecting Operation Troy: Cyberespionage in South Korea.* McAfee Available at: https://www.mcafee.com/us/resources/white-papers/wp-dissecting-operation-troy.pdf (Accessed: 29th January 2017).

Sherwood, J., Clark, A. and Lynas, D. (2005) *Enterprise Security Architecture: A Business-Driven Approach.* Boca Raton, FL: CRC Press.

Shostack, A. (2014) *Threat Modeling: Designing for Security.* Indianapolis, IN 46256: John Wiley & Sons, Inc.

Siddiqui, M., Wang, M. C. and Lee, J. (2008) *Proceedings of the 46th Annual Southeast Regional Conference on XX*. Auburn, Alabama: ACM.

Simpson, G. i. C. (1982) *"Auguste Comte 1798-1857." Origins and Growth of Sociological Theory. Chicago, Nelson Hall (1982) P80 in Crotty, M. (1998).* London: Sage.

Singh, A. and Bu, Z. (2014) *Hot Knives through Butter: Evading File-Based Sandboxes.* Fireeye Available at: https://www.fireeye.com/content/dam/fireeye-www/current-threats/pdfs/pf/file/fireeye-hot-knives-through-butter.pdf (Accessed: 29th January 2017).

Singh, A., Gomez, J. and Malik, A. (2014) *Brewing up Trouble: Analyzing Four Widely Exploited Java Vulnerabilities.* Fireeye Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-java-vulnerabilities.pdf (Accessed: 29th Janaury 2017).

Singh, B. and Singh, U. (2018) 'Program Execution Analysis in Windows: A Study of Data Sources, Their Format and Comparison of Forensic Capability', *Computers & Security,* 74, pp. 94–114.

Singh, G. and Supriya. (2013) 'A Study of Encryption Algorithms (Rsa, Des, 3des and Aes) for Information Security', *International Journal of Computer Applications,* 6(19), pp. 33-38.

Skoudis, E. and Zeltser, L. (2004) *Malware Fighting Malicious Code.* 1st edn. Upper Saddle River, New Jersey 07458: Prentice Hall Education Inc.

Slot, T. (2015) *Detection of Apt Malware through External and Internal Network Traffic Correlation.* University of Twente.

Smeets, M. (Undated) *What It Takes to Develop a Cyber Weapon.* Columbia Available at: https://sipa.columbia.edu/sites/default/files/WorkingPaperSeries_1.pdf (Accessed: 3rd March 2017).

Smith, J. A. and Rothwell, W. (2015) *Mitigating Cyber Threat from Malicious Insiders.* Available at: https://www.royalholloway.ac.uk/isg/documents/pdf/research/jason-anthony-smith-computer-weekly.pdf (Accessed: 18th March 2018).

smorgasbort. (2003-2020) *Pelles C*. Available at: http://www.smorgasbordet.com/pellesc/ (Accessed: 15th November 2019).

SNIA. (2009) *Common Raid Disk Data Format Specification Version 2.0 Revision 19*. Available at: https://www.snia.org/sites/default/files/SNIA_DDF_Technical_Position_v2.0.pdf (Accessed: 12th August 2019).

Sobey, C. (2004) *Recovering Unrecoverable Data.* ChannelScience Available at: https://www.researchgate.net/publication/237443436_Recovering_Unrecoverable_Data (Accessed: 3rd March 2018).

Social-Research-Association (2003) *Ethical Guidelines (2003).* Available at: http://the-sra.org.uk/wp-content/uploads/ethics03.pdf (Accessed: 30th November 2016).

Solad, A., Hatheway, D., López, M. R. and Fokker, J. (2018) *Kraken Cryptor Ransomware Gains Popularity among Cybercriminals.* Available at: https://go.recordedfuture.com/hubfs/reports/cta-2018-1030.pdf (Accessed: 20th February 2020).

Sophos (2013) *Security Threat Report 2013.* Sophos Available at: https://www.sophos.com/en-us/medialibrary/PDFs/other/sophossecuritythreatreport2013.pdf (Accessed: 9th February 2017).

Sophos. (2019) *Emotet 101, Stage 3: The Emotet Executable*. Available at: https://news.sophos.com/en-us/2019/03/05/emotet-101-stage-3-the-emotet-executable/ (Accessed: 25th October 2020).

Srinivasan, A. and Kolli, S. (2013) 'Steganographic Information Hiding That Exploits a Novel File System Vulnerability', *International Journal of Security and Networks,* 8(2), pp. 82-93.

Standage, T. (2017) *The Crooked Timber of Humanity* Available at: https://www.1843magazine.com/technology/rewind/the-crooked-timber-of-humanity (Accessed: 12th October 2018).

statista. (2019a) *Global Hard Disk Drive (Hdd) Shipments 2010-2019, by Quarter*. Available at: https://www.statista.com/statistics/275336/global-shipment-figures-for-hard-disk-drives-from-4th-quarter-2010/ (Accessed: 30th Octpber 2019).

statista. (2019b) *Market Share Held by the Leading Windows Anti-Malware Application Vendors Worldwide, as of January 2019*. Available at: https://www.statista.com/statistics/271048/market-share-held-by-antivirus-vendors-for-windows-systems/ (Accessed: 18th May 2019).

statista. (2019c) *Shipments of Hard and Solid State Disk (Hdd/Ssd) Drives Worldwide from 2015 to 2021* Available at: https://www.statista.com/statistics/285474/hdds-and-ssds-in-pcs-global-shipments-2012-2017/ (Accessed: 30th October 2019).

statista. (2019d) *Size of the Cybersecurity Market Worldwide, from 2017 to 2023 (in Billion U.S. Dollars)*. Available at: https://www.statista.com/statistics/595182/worldwide-security-as-a-service-market-size/ (Accessed: 30th October 2019).

Stewart, A. (2014) *Dll Side-Loading: A Thorn in the Side of the Anti-Virus Industry.* Fireeye Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-dll-sideloading.pdf (Accessed: 29th Janaury 2017).

Stewart, J. (2011) *Htran and the Advanced Persistent Threat.* Dell SecureWorks Available at: https://www.ncsc.nl/binaries/content/documents/ncsc-nl/conference/conference-2011/speakers/joe-stewart/1/Presentation%2BJoe%2BStewart.pdf (Accessed: 19th November 2017).

Stewart, J. N. (2014) 'Advanced Technologies/Tactics Techniques, Procedures: Closing the Attack Window, and Thresholds for Reporting and Containment', in Hathaway, M. E. (ed.) *Best Practices in Computer Network Defense: Incident Detection and Response.* Amsterdam Berlin Tokyp Washington DC: IOS Press, pp. 30 - 42.

Stoll, C. (1988) *The Cuckoo's Egg: Tracking a Spy through the Maze of Computer Espionage.* USA: Simon & Schuster

Strenski, D., Simkins, J., Walke, R. and Wittig, R. (2008) *Second International Workshop on High-Performance Reconfigurable Computing Technology and Applications (HPRCTA'08)*. Austin, Texas: IEEE.

Suenaga, M. (2009) *A Museum of Api Obfuscation on Win32.* Symantec Available at: https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/a_museum_of_api_obfuscation_on_win32.pdf (Accessed: 29th January 2017).

Suenaga, M. (2012) *W32.Changeup: How the Worm Was Created.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_changeup_how_the_worm_was_created.pdf (Accessed: 29th January 2007).

Sutton, R. I. and Rao, H. (2014) *Scaling up Excellence: Getting to More without Settling for Less.* United States: Crown Business.

Symantec-Security-Response (2012a) *Have I Got Newsforyou: Analysis of Flamer C&C Server.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_flamer_newsforyou.pdf (Accessed: 29th January 2017).

Symantec-Security-Response (2012b) *The Luckycat Hackers.* Symantec Available at: https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/luckycat-hackers-12-en.pdf (Accessed: 29th January 2017).

Symantec-Security-Response (2012c) *Rootkits.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/rootkits.pdf (Accessed: 29th January 2017).

Symantec-Security-Response (2013) *Comment Crew: Indicators of Compromise.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/comment_crew_indicators_of_compromise.pdf (Accessed: 20th March 2017).

Symantec-Security-Response (2014) *Dragonfly: Cyberespionage Attacks against Energy Suppliers.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/Dragonfly_Threat_Against_Western_Energy_Suppliers.pdf (Accessed: 31st January 2017).

Symantec-Security-Response (2015a) *Butterfly: Corporate Spies out for Financial Gain.* Symantec Available at: https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/butterfly-corporate-spies-out-for-financial-gain.pdf (Accessed: 11th February 2017).

Symantec-Security-Response (2015b) *Dyre: Emerging Threat on Financial Fraud Landscape.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/dyre-emerging-threat.pdf (Accessed: 27th February 2017).

Symantec-Security-Response (2015c) *Regin: Top-Tier Espionage Tool Enables Stealthy Surveillance.* Semantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/regin-analysis.pdf (Accessed: 4th February 2017).

Symantec-Security-Response (2015d) *W32.Ramnit Analysis.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32-ramnit-analysis.pdf (Accessed: 29th January 2017).

Symantec-Security-Response (2016) *The Waterbug Attack Group.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/waterbug-attack-group.pdf (Accessed: 4th February 2017).

Symantec. (2005) *Fighting Epo Viruses*. Available at: https://www.symantec.com/connect/articles/fighting-epo-viruses (Accessed: 23rd January 2020).

Symantec. (2007) *W2k.Stream*. Available at: https://www.symantec.com/security-center/writeup/2000-121416-2928-99 (Accessed: 25th April 2018).

Symantec (2011) *Advanced Persistent Threats: A Symantec Perspective.* Symantec Available at: https://www.symantec.com/content/en/us/enterprise/white_papers/b-advanced_persistent_threats_WP_21215957.en-us.pdf (Accessed: 29th January 2017).

Symantec (2013) *Istr Internet Security Threat Report 2013.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_v18_2012_21291018.en-us.pdf (Accessed: 9th July 2017).

Symantec (2016a) *The Increased Use of Powershell in Attacks.* Symantec Available at: https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/increased-use-of-powershell-in-attacks-16-en.pdf (Accessed: 29th January 2017).

Symantec (2016b) *Istr Internet Security Report Volume 21, April 2016.* Symantec Available at: https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf (Accessed: 1st April 2017).

Symantec. (2017) *Security Center White Papers*. Available at: https://www.symantec.com/security-center/white-papers (Accessed: 4th March 2017).

Szabo, P. and Huq, N. (2013) 'Trapping Unknown Malware in a Context Web', *Virsu Bulletin*. Available at: https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/HuqSzabo-VB2013.pdf (Accessed: 28th January 2017).

Szappanos, G. (2014a) *The Rotten Tomato Campaign.* Sophos Available at: https://www.sophos.com/en-us/medialibrary/PDFs/technical%20papers/sophos-rotten-tomato-campaign.pdf (Accessed: 28th January 2017).

Szappanos, G. (2014b) *Vba Is Not Dead.* Available at: https://www.virusbulletin.com/uploads/pdf/magazine/2014/vb201407-VBA.pdf (Accessed: 17th March 2017).

Szappanos, G. (2015a) *Exploit This: Evaluating the Exploit Skills of Malware Groups.* Sophos Available at: https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophos-exploit-this-evaluating-exploit-skills-of-malware-groups.pdf?la=en (Accessed: 28th January 2017).

Szappanos, G. (2015b) *Plugx Goes to the Registry (and India).* Sophos Available at: https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/plugx-goes-to-the-registry-and-india.pdf (Accessed: 28th January 2017).

Szappanos, G. (2016) *Ancalog – the Vintage Exploit Builder.* Sophos Available at: https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/Ancalog-the-vintage-exploit-builder.pdf?la=en (Accessed: 28th January 2017).

Szappanos, G. (2016) *Is It Time for Cve-2012-0158 to Retire?* Sophos Available at: https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/Is-it-time-for-CVE-2012-0158-to-retire.PDF?la=en (Accessed: 9th February 2017).

Ször, P. and Ferrie, P. (2003) *Hunting for Metamorphic.* Symantec Available at: http://www.symantec.com/avcenter/reference/hunting.for.metamorphic.pdf (Accessed: 29th January 2017).

Taleb, N. N. (2007) *The Black Swan.* United States of America: Random House.

Tanenbaum, A. S. (2001) *Modern Operating Systems Second Edition.* Upper Saddke River, New Jersey 07458: Prentice-Hall Inc.

Tanenbaum, A. S., Herder, J. N. and Bos, H. (2005) 'File Size Distribution on Unix Systems - Then and Now', *ACM SIGOPS Operating Systems Review,* 40(1),

Taylor, F. W. (1919) *The Principles of Scientific Management.* New York: W.W.Norton.

Teddlie, C. and Johnson, R. B. (2009) *Methodological Thought before the 20th Century in Teddlie, C. & Tashakkori, A. (2009) Foundations of Mixed Methods Research.* London: Sage.

The-Ohio-State-University. (2019) *Building / Testing Via the Hosts File*. Available at: https://web.osu.edu/technical-support/tips-tricks/hosts-file/ (Accessed: 10th November 2019).

Thomassen, J. (2008) *Forensic Analysis of Unallocated Space in Windows Registery Hives Files.* The University of Liverpool.

Tirpak, J. A. (2008) 'Find, Fix, Track, Target, Engage, Assess', *Air Force Magazine*,

Top500. (2019a) *Operating System Family / Windows* Available at: https://www.top500.org/statistics/details/osfam/2 (Accessed: 30th October 2019).

Top500. (2019b) *Performance Development*. Available at: https://www.top500.org/statistics/perfdevel/ (Accessed: 27th June 2019).

Toro, A., Griffin, N. and Settle, A. (2016) *Sledgehammer Gamification of Ddos Attacks (for Ideology Profit & Mischief).* Available at: https://www.forcepoint.com/sites/default/files/resources/files/datasheet_sledgehammer_the_gamification_of_ddos_attacks_en.pdf (Accessed: 9th February 2016).

Torri, S., Morinaga, M., Yoshioka, T. and Terada, T. (2014) 'Multi-Layered Defense against Advanced Persistent Threats (Apt)', *FUJITSU Sci. Tech. J.,* Vol. 50, No. 1(January 2014), pp. 52-58.

Totalhash. (2020) *Malware Analysis Database*. Available at: https://totalhash.cymru.com/analysis/?631d571646e0dadb47f6b06094e7525b8fd29987 (Accessed: 25th October 2020).

Trend-Labs (2014) *The Enterprise Fights Back (Part Iv): Building Threat Intelligence,.* Trend Micro Available at: http://about-threats.trendmicro.com/cloud-content/us/ent-primers/pdf/Building/Threat/Intelligence_final.pdf (Accessed: 9th February 2017).

Trend-Micro-Cyber-Safety-Solutions-Team (2016) *Fastpos: Quick and Easy Credit Card Theft.* Trend Micro Available at: http://documents.trendmicro.com/assets/fastPOS-quick-and-easy-credit-card-theft.pdf (Accessed: 16th February 2017).

Trend-Micro-Incorporated (2015) *Operation Arid Viper Byassing the Iron Dome.* Trend Micro Available at: http://www.trendmicro.co.uk/media/wp/operation-arid-viper-whitepaper-en.pdf (Accessed: 26th June 2017).

Trend-Micro-Threat-Research-Team (2012) *The Taidoor Campaign an in-Depth Analysis.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp_the_taidoor_campaign.pdf (Accessed: 15th April 2017).

Trend-Micro (2012) *Operation Ghost Click: The Rove Digital Takedown.* Trend Micro Available at: http://www.trendmicro.co.uk/media/misc/rove-digital-takedown-research-paper-en.pdf (Accessed: 27th June 2017).

Trend-Micro (2013a) *2q Report on Targeted Attack Campaigns.* Trend Micro Available at: http://about-threats.trendmicro.com/cloud-content/us/ent-primers/pdf/2q-report-on-targeted-attack-campaigns.pdf (Accessed: 15th April 2017).

Trend-Micro (2013b) *Countering the Advanced Persistent Threat Challenge with Deep Discovery.* Trend Micro Available at: http://apac.trendmicro.com/cloud-content/apac/pdfs/products/enterprise/wp01_deepdiscovery_130219us.pdf (Accessed: 9th February 2017).

Trend-Micro (2013c) *Data Exfiltration How Do Threats Actors Steal Your Data?* Trend Micro Available at: http://about-threats.trendmicro.com/cloud-content/us/ent-primers/pdf/how_do_threat_actors_steal_your_data.pdf (Accessed: 15th April 2017).

Trend-Micro (2015) *Bad Ads and Zero Days: Reemerging Threats Challenge Trust in Supply Chains and Best Practices.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/rpt-trendlabs-2015-1q-security-roundup-bad-ads-and-zero-days-reemerging-threats-challenge-tr.pdf (Accessed: 6th February 2017).

Trend-Micro. (2017a) *Research and Analysis*. Available at: https://www.trendmicro.com/vinfo/us/security/research-and-analysis. (Accessed: 4th March 2017).

Trend-Micro. (2017b) *Understanding Targeted Attacks: Six Components of Targeted Attacks*. Available at:

https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/targeted-attacks-six-components (Accessed: 16th April 2017).

Trend-Micro.Forward-Looking-Threat-Research-Team (2012) *Lucky Cat Redux inside an Apt Campaign with Multiple Targets in India and Japan.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp_luckycat_redux.pdf (Accessed: 31st January 2017).

TrendLabs *Setting the Stage: Landscape Shifts Dictate Future Threat Response Strategies.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/reports/rpt-setting-the-stage.pdf (Accessed: 14th February 2017).

TrendLabs (2012a) *Detecting the Enemy Insider the Network How Tough Is It to Deal with Apts?* Trend Micro Available at: http://apac.trendmicro.com/cloud-content/apac/pdfs/business/white-papers/wp_apt-primer.pdf (Accessed: 31st January 2017).

TrendLabs (2012b) *Maintaining Vulnerable Servers What's You Window of Exposure?* Available at: http://about-threats.trendmicro.com/cloud-content/us/ent-primers/pdf/wp_vulnerability-shielding-primer.pdf (Accessed: 17th May 2017).

TrendLabs (2012c) *Spear-Phishing Email: Most Favored Apt Attack Bait.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-spear-phishing-email-most-favored-apt-attack-bait.pdf (Accessed: 31st January 2017).

TrendLabs (2013a) *How to Erase Data Securely.* Trend Micro Available at: https://documents.trendmicro.com/images/tex/guides/how-to-erase-data-securely.pdf (Accessed: 16th Fenruary 2017).

TrendLabs (2013b) *Malicious Network Communications What Are You Overlooking?* Trend Micro Available at: http://about-threats.trendmicro.com/cloud-content/us/ent-primers/pdf/wp_tl_malicious_network_communications.pdf (Accessed: 15th April 2017).

TrendLabs (2013c) *Targeted Attack Trends 2h 2013 Report.* Trend Micro Available at: https://documents.trendmicro.com/threat-intelligence/targeted-attack-trends/rpt-targeted-attack-trends-2h-2013.pdf (Accessed: 15th April 2017).

TrendLabs (2014) *Targeted Attack Trends in Asia-Pacific.* Trend Micro Available at: http://apac.trendmicro.com/cloud-content/apac/pdfs/security-intelligence/reports/rpt-1h-2014-targeted-attack-trends-in-asia-pacific.pdf (Accessed: 16th February 2017).

TrendLabs (2015a) *A Rising Tide: New Hacks Threaten Public Technologies.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/reports/rpt_a_rising_tide.pdf (Accessed: 9th February 2017).

TrendLabs (2015b) *Understanding Targeted Attacks: The Six Components.* Trend Micro Available at: https://documents.trendmicro.com/assets/primers/understanding-targeted-attacks.pdf (Accessed: 16th February 2017).

TrendMicro. (2002) *Worm_Frethem.A*. Available at: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/archive/malware/worm_frethem.a (Accessed: 24th October 2020).

TrendMicro (2013) *Lateral Movement: How Do Threat Actors Move Deeper into Your Network?* Trend Micro Available at: http://about-threats.trendmicro.com/cloud-content/us/ent-primers/pdf/tlp_lateral_movement.pdf (Accessed: 15th April 2017).

TrendMicros. (2009) *Pe_Sality.Az-2*. Available at: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/archive/malware/pe_sality.az-2 (Accessed: 24th October 2020).

Trivitt, B. (2013) *The Evolution of Apts (Advanced Persistent Threats).* Kern/ISSA Available at: http://kern.issa.org/wp-content/uploads/2013/08/The-Evolution-of-APTsv1_2.pdf (Accessed: 18th February 2017).

Un Charter 1945 Chapter c. Available at: http://www.un.org/en/sections/un-charter/un-charter-full-text/index.html (Accessed: 27th October 2018).

53/70. Developments in the Field of Information and Telecommunications in the Context of International Security, (1998).

United-Nations. (2018) *Un Resolutions Related to Cybersecurity*. Available at: https://www.itu.int/en/action/cybersecurity/Pages/un-resolutions.aspx (Accessed: 27th October 2018).

Univeristy-of-Gloucestershire. (2019) *Research Ethics*. Available at: https://www.glos.ac.uk/research/Pages/research-ethics.aspx (Accessed: 23rd December 2019).

University-of-Berkeley. (2019) *Seti@Home*. Available at: https://setiathome.berkeley.edu/ (Accessed: 12th April 2019).

University-of-Gloucestershire. (2018) *Library and Information Services* Available at: https://infonet.glos.ac.uk/departments/lis/Pages/Research.aspx# (Accessed: 24th Marahc 2018).

Unknown. (1947) *Nuremberg Code Para 1*. Available at: https://history.nih.gov/research/downloads/nuremberg.pdf (Accessed: 30th November 2016).

Unknown. (2018) *Antimicrobial Resistence Learning Site for Veterinarians*. Available at: https://amrls.cvm.msu.edu/pharmacology/antimicrobials/mode-of-action (Accessed: 30th March 2018).

Unknown. (2019) *Welcome to Yara's Documentation!* Available at: https://yara.readthedocs.io/en/v3.10.0/ (Accessed: 4th July 2019).

Uppal, D., Mehra, V. and Verma, V. (2014) 'Basic Survey on Malware Analysis, Tools and Techniques', *International Journal on Computational Sciences & Applications,* 4(1),

US-Dept-of-Defense. (1965) *Controls over Data Processing Equipment Utilized in Disbursing Operations (from the Accounting and Finance Manual, Section 10214, January 1965)*. Available at: https://www.ncjrs.gov/App/abstractdb/AbstractDBDetails.aspx?id=65114 (Accessed: 28th July 2019).

Federal Food, Drug, and Cosmetic Act 2014 Chapter c. Available at: https://legcounsel.house.gov/Comps/FDA_CMD.pdf (Accessed: 15th March 2020).

USC (2003) *Johari Window* Available at: https://www.usc.edu/hsc/ebnet/Cc/awareness/Johari windowexplain.pdf (Accessed: 12th December 2016).

Uscilowsk, B., Weber, J., Robinson, C. and Parsons, T. (2008) *Clean Data Profiling.* Symantec Available at: https://www.virusbulletin.com/uploads/pdf/conference_slides/2008/Uscilowski-Weber-VB2008.pdf (Accessed: 12th October 2019).

Various. (2012) *Enumerating Registry Subkeys*. Available at: https://stackoverflow.com/questions/8132786/enumerating-registry-subkeys (Accessed: Unknown).

Velazquez, C. (2017) *Detecting and Preventing Attacks Earlier in the Kill Chain.* SANS (Accessed: 29th January 2017).

Vengerik, B., Dennesen, K., Berry, J. and Wrolstad, J. (2014) *Hacking the Street? Fin4 Likely Playng the Market.* Fireeye Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-fin4.pdf (Accessed: 9th February 2017).

Verizon (2012) *2012 Data Breach Investigations Report.* Available at: http://www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2012_en_xg.pdf (Accessed: 7th June 2017).

Verizon (2015) *Data Breach Investigations Report 2015.* Available at: http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigation-report_2015_en_xg.pdf (Accessed: 31st October 2017).

vermont.gov. (2020) *Data Brokers*. Available at: https://www.sec.state.vt.us/corporationsbusiness-services/data-brokers.aspx (Accessed: 30th Janaury 2020).

Villeneuve, Nart and Sancho, D. (2011) *The "Lurid" Downloader.* Available at: http://la.trendmicro.com/media/misc/lurid-downloader-enfal-report-en.pdf (Accessed: 31st January 2017).

Villeneuve, N. (2011) *Trends in Targeted Attacks.* Trend Micro Available at: https://www.trendmicro.es/media/wp/trends-in-targeted-attacks-whitepaper-en.pdf (Accessed: 27th September 2020).

Villeneuve, N. and Bennett, J. (2012) *Detecting Apt Activity with Network Traffic Analysis.* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-detecting-apt-activity-with-network-traffic-analysis.pdf (Accessed: 27th February 2017).

Villeneuve, N. and Bennett, J. T. (2014) *Xtremerat: Nuisance or Threat?* Available at: https://www.fireeye.com/blog/threat-research/2014/02/xtremerat-nuisance-or-threat.html (Accessed: 23rd February 2020).

Villeneuve, N., Bennett, J. T., Moran, N., Haq, T., Scott, M. and Geers, K. (2014) *Operation "Ke3chang": Targeted Attacks against Ministries of Foreign Affairs.* Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/wp-operation-ke3chang.pdf (Accessed: 29th January 2013).

Villeneuve, N. and dela Torre, J. (2013) *Fakem Rat Malware Disguised as Windows® Messenger and Yahoo!® Messenger.* Trend Micro Available at: http://apac.trendmicro.com/cloud-content/apac/pdfs/security-intelligence/white-papers/wp-fakem-rat.pdf (Accessed: 31st January 2017).

Villeneuve, N., Moran, N., Haq, T. and Scott, M. (2014) *Operation Saffron Rose.* Fireeye Available at: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-operation-saffron-rose.pdf (Accessed: 29th Janaury 2017).

Virus-Bulletin (2012) *Virus Bulletin Fighting Malware and Spam November 2012.* Virus Bulletin Available at: https://www.virusbulletin.com/uploads/pdf/magazine/2012/201211.pdf (Accessed: 19th February 2017).

Virus-Bulletin (2013) *Virus Bulletin Covering the Global Threat Landscape December 2013.* Virus Bulletin Available at: https://www.virusbulletin.com/uploads/pdf/magazine/2013/201312.pdf (Accessed: 19th February 2017).

Virus-Bulletin. (2015) *Turlasat: The Fault in Our Stars Turla's Exquisite Satlink Appropriation.* Virus Bulletin

Wall, D. S. (2013) 'Enemies Within: Redefining the Insider Threat in Organizational Security Policy', *Security Journal,* 28(2), pp. 107–124

Wang, R. (2013) *Malware B-Z: Inside the Threat from Blackhole to Zeroaccess.* Sophos Available at: https://www.sophos.com/en-us/medialibrary/Gated%20Assets/white%20papers/sophos_from_blackhole_to_zeroaccess_wpna.pdf (Accessed: 27th February 2017).

WatchGuard® Technologies, I. (2016) *Advanced Zero Day Protection with Apt Blocker.* WatchGuard® Technologies, Inc. Available at: http://www.watchguard.com/docs/whitepaper/wg_advanced-zero-day-protection_wp.pdf (Accessed: 19th March 2017).

Wayback-Machine. (2017) *Internet Archive Wayback Machine*. Available at: http://archive.org/web/

websense (2011) *Advanced Persistent Threats and Other Advanced Attacks: The Threat Analysis and Defense Strategies for Smb, Mid-Size, and Enterprise Organizations.* Available at: http://www.websense.com/assets/white-papers/whitepaper-websense-advanced-persistent-threats-and-other-advanced-attacks-en.pdf (Accessed: 3rd March 2018).

websense (2013) *Five Essentials for Protecting against Advanced Persistent Threats (Apts).* websense Available at: https://www.fujitsu.com/uk/Images/whitepaper-5-ways-to-protect-against-apt-en.pdf (Accessed: 5th February 2017).

Werner, J. and Webb, J. (2017) *Shoal Attack: How a School of Fish Helped Aussie Netballers Win Gold*. Available at: http://www.abc.net.au/news/2017-10-05/australian-netballs-secret-weapon-is-a-school-of-fish/9014326 (Accessed: 20th May 2018).

Werthmann, T. (2006) *Survey on Buffer Overflow Attacks and Countermeasures.* Available at: http://www.nds.rub.de/media/nds/attachments/files/2010/11/Survey.on.Buffer.Overflow.Attacks.and.Countermeasures.pdf (Accessed: 20th March 2017).

Wilhoit, K. (2013) *Who's Really Attacking Your Ics Equipment?* Trend Micro Available at: https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-whos-really-attacking-your-ics-equipment.pdf (Accessed: 9th February 2017).

Wilkinson, M. (2017) *Ntfs Reference Sheet*. Available at: http://www.writeblocked.org/resources/ntfs_cheat_sheets.pdf (Accessed: Unknown).

Willet, K. D. (2008) *Information Assurance Architecture.* Boca Raton, FL 33487-2742: Taylor & Francis group, LLC.

Wilson, P. R., Johnstone, M. S., Neely, M. and Boles, D. (Undated) *Dynamic Storage Allocation a Survey and Critical Review.* University of Texas Available at: ftp://ftp.cs.utexas.edu/pub/garbage/allocsrv.ps (Accessed: 3rd March 2018).

Wood, J. (2019) *Ip Address Search Algorithm.*

Woods, L., Teubner, J. and Alonso, G. (2011) *2011 IEEE 27th International Conference on Data Engineering* Hannover, Germany: IEEE.

World-Bank. (2018) *Doing Business 2019: A Year of Record Reforms, Rising Influence*. Available at: https://www.worldbank.org/en/news/immersive-story/2018/10/31/doing-business-2019-a-year-of-record-reforms-rising-influence (Accessed: 26th July 2019).

Wright, C., Kleiman, D. and Sundhar, S. (2008) *Information Systems Security 4th International Conference, ICISS 2008*. Hyderabad, India: ICISS: International Conference on Information Systems Security.

Wrolstad, J. and Vengerik, B. (2015) *Pinpointing Targets: Exploiting Web Analytics to Ensnare Victims.* Available at: https://www2.fireeye.com/rs/848-DID-242/images/rpt-witchcoven.pdf (Accessed: 6th October 2019).

Wüest, C. (2012) *RSA Conference Europe 2012*. Europe: Synmantec.

Wüest, C. (2014a) *The Continued Rise of Ddos Attacks.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-continued-rise-of-ddos-attacks.pdf (Accessed: 29th January 2017).

Wüest, C. (2014b) *The State of Financial Trojans 2014.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-state-of-financial-trojans-2014.pdf (Accessed: 29th January 2017).

Wüest, C. (2014) *Targeted Attacks against the Energy Sector.* Symantec Available at: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/targeted_attacks_against_the_energy_sector.pdf (Accessed: 29th January 2017).

Wüest, C. and Anand, H. (2017) *Living Off the Land and Fileless Attack Techniques.* Symantec Available at: https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-living-off-the-land-and-fileless-attack-techniques-en.pdf (Accessed: 12th October 2019).

Wyke, J. (2011) *What Is Zeus?* Sophos Available at: https://www.sophos.com/medialibrary/PDFs/technical%20papers/Sophos%20what%20is%20zeus%20tp.pdf (Accessed: 28th January 2017).

Wyke, J. (2012a) 'Zeroaccess',

Wyke, J. (2012b) *The Zeroaccess Botnet – Mining and Fraud for Massive Financial Gain.* Sophos Available at: https://www.sophos.com/en-us/medialibrary/PDFs/technical%20papers/Sophos_ZeroAccess_Botnet.pdf (Accessed: 28th January 2017).

Wyke, J. (2014) *Vawtrak – International Crimeware-as-a-Service.* Sophos Available at: https://www.sophos.com/en-us/medialibrary/PDFs/technical%20papers/sophos-vawtrak-international-crimeware-as-a-service-tpna.pdf (Accessed: 28th January 2017).

Wyke, J. and Ajjan, A. (2015) *The Current State of Ransomware.* Sophos Available at: https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophos-current-state-of-ransomware.pdf (Accessed: 19th February 2017).

Xilinx. (2019) *Xilinx*. Available at: https://www.xilinx.com/products/intellectual-property/1-8dyf-1597.html (Accessed: 29th June 2019).

Yan, W., Zhang, Z. and Ansari, N. (2008) 'Revealing Packed Malware', *IEEE Security & Privacy,* 6(5), pp. 65 - 69

Yaneza, J. (2015a) *Gamapos the Andromeda Botnet Connection.* Trend Micro Available at: http://documents.trendmicro.com/assets/GamaPOS-TechnicalBrief.pdf (Accessed: 16th February 2017).

Yaneza, J. (2015b) *Malumpos Technical Brief.* Trend Micro Available at: http://documents.trendmicro.com/images/tex/pdf/MalumPOS/Technical/Brief.pdf (Accessed: 16th February 2017).

Yaneza, J. and Mendoza, E. (2015) *The Cuckoo Miner Campaign.* Trend Micro Available at: http://documents.trendmicro.com/assets/Cuckoo-Miner-Technical-Brief.pdf (Accessed: 16th February 2017).

Yaneza, J. and Mendoza, E. (2016) *Fighterpos Malware Gets Worm Routine.* Trend Micro Available at: http://documents.trendmicro.com/assets/threat-reports/fighterpos-malware-gets-worm-routine_ver2.pdf (Accessed: 16th February 2016).

Yeadon, M. R. (2005) *What Are the Limitations of Experimental and Theoretical Approaches in Sports Biomechanics?' in Mcnamee. M. (Ed) Philosophy and the Sciences of Exercise, Health and Sport.* London: Routledge.

Zamora, W. (2015) *What's the Difference between Antivirus and Anti-Malware?* Available at: https://blog.malwarebytes.com/101/2015/09/whats-the-difference-between-antivirus-and-anti-malware/ (Accessed: 4th Jnauary 2020).

Zelkowitz, M. (2008) *Advances in Computers: Software Development.* Vol. 74. London: Elsevier Inc.

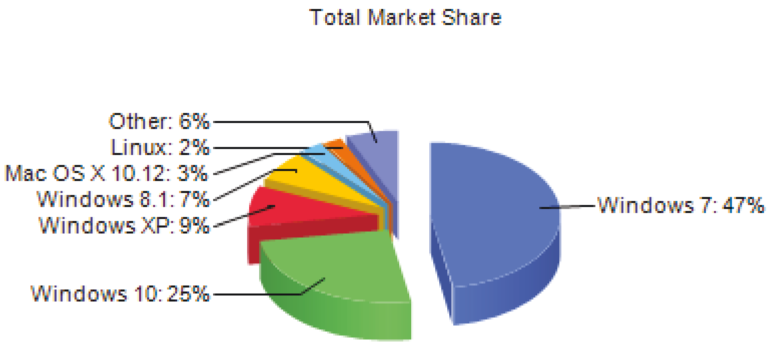
Zetter, K. (2014) *A Google Site Meant to Protect You Is Helping Hackers Attack You*. Available at: https://www.wired.com/2014/09/how-hackers-use-virustotal/ (Accessed: 23rd March 2018).

Βιρβίλης-Κολλητήρης, N. K. (2015) *Fighting an Unfair Battle: Unconventional Defenses against Advanced Persistent Threats.* Athens University of Economics and Busines.

APPENDICES

A. Desktop Operating System Market Share

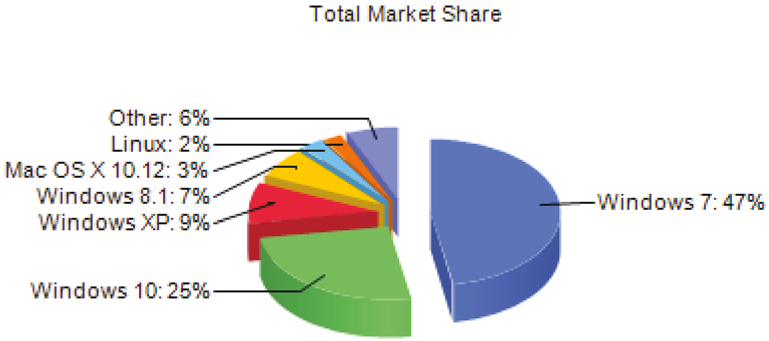
Desktop Operating System Market Share January, 2017



Operating System	Total Market Share
Windows 7	47.20%
Windows 10	25.30%
Windows XP	9.17%
Windows 8.1	6.90%
Mac OS X 10.12	2.75%
Linux	2.27%
Mac OS X 10.11	1.73%
Windows 8	1.62%
Mac OS X 10.10	1.07%
Windows Vista	0.84%
Mac OS X 10.9	0.39%
Windows NT	0.36%
Mac OS X 10.6	0.13%
Mac OS X 10.8	0.12%
Mac OS X 10.7	0.11%
Mac OS X 10.5	0.02%
Windows 2000	0.02%
Windows 98	0.01%
FreeBSD	0.00%
Mac OS X 10.4	0.00%

Source: Desktop Operating System Market (Netmarketshare, 2017)

Desktop Operating System Market Share September 2019



Operating System	Total Market Share
Windows	86.38%
Mac OS	11.16%
Linux	1.80%
Chrome OS	0.40%
Unknown	0.26%
BSD	0.00%

Source: (Netmarketshare, 2019a)

B. Setting up a machine for Windows Device Driver Testing

This section describes the facilitating steps necessary for testing device driver software in the Windows kernel. Depending on how much of the software is already on the developer's machine and internet link speed this process could take up to 3 hours plus fault resolution time. A knowledge of Operating systems and the Windows kernel is desirable. The developer will need administrator access on a host machine to install on a virtual machine (VM) and will need to ensure that the hardware of the machine on which the software is to be run is sufficient.

Install the latest version of Visual Studio with Windows Driver Kit (WDK and SDK) and qspectre libraries. The latter may be done by the Virtual Studio Installation software.

Obtain a Windows .iso file with associated product key;

Obtain Windows VMware VM software. Create a virtual machine using the Windows .iso file. Defaults (e.g. 60GB disc) are acceptable. Ensure bi-directional copy is enabled.

Obtain a driver loader. A commercial one is available or one can be written in four short lines (see below). Should the commercial loader be used ensure that the correct hardware is selected (for example, for Windows 7 x64 and above use: myfolder -> kit -> WLH -> AMD64 – FRE). Drag and drop the loader into the virtual machine.

Download to Virtualkd windbg for debugging. Click the download button and follow the download instructions at the bottom of the page. Drag and drop virtualkd into the VM.

Restart the VM. Run VMinstall from the target folder. Follow the instructions. The VM will re-boot. Select the VM with signature disabled. From the VM monitor select the debugger path (there should be a windbg.exe default). Select F8 from the VM load and select "Disable Driver Signature Enforcement". A debugger windows should popup. Select Debug -> Go and the VM should resume booting;

Configure the VM. In VM Machine Settings add a serial port. Use a named pipe e.g. \\.\pipe\MyPipe. (\\.\pipe\ is compulsory. Add your own pipe name after this). Select “This end is the server”. “Other end is an application” machine”. On the VM search for the program “msconfig”. Select “System Configuration”, Boot -> Advanced Options, Select, Debug, COM2, Baud Rate 11520. On the VM search for the program “Computer Management”, Device Manager -> Ports (COM & LPT). COM2 should exist and under “General” should be working properly. Exit and Restart;

In the host machine debugger File -> Kernel Debug – COM. Select Baud Rate as 11520. Check Pipe box and change Port to the pipe name in the previous step (in this case \\.\pipe\MyPipe.

Write batch with the two BCD edit commands (Examples here and here) and include as the this command line:

REG ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Debug Print Filter " /v DEFAULT /t REG_DWORD /d 0xf

Write and compile the driver the host machine. File -> New Project -> Installed -> Visual C++ -> Legacy -> Empty WDM Driver. Create a project name. When project has v=been created right click in Solution Explorer project name -> add -> new item. Select C++ and change file extension from .cpp to .c. Write code. In Solution Explorer project name -> property -> Properties – Configuration Manager. Choose the platform for your host machine. Configuration Properties -> Warnings Level 1. Linker -> Warnings to “No”. Driver Settings -> OS to VM OS. Select Target Platform. Inf2Cat -> Use Local time. Now compile (Build) the solution.

Drag and drop into the VM and use the Loader (commercial or otherwise) to run the driver. Debug information should appear in the debugger window.

Troubleshooting

The debugger may hang adding symbols may help:

“SRV*c:\MyServerSymbols*https://msdl.microsoft.com/download/symbols”

Ensure compatibility between COM ports

Three YouTube videos

Windows Kernel Programming Tutorial 1 - Setting up Environment - Part 1

Windows Kernel Programming Tutorial 2 - Setting up Environment - Part 2

Windows Kernel Programming Tutorial 3 - Writing a simple driver (Good for Visual Studio setup)

Kernel Debugging with windbg - How to start Windows Kernel live debugging

Two YouTube background videos

Windows Driver Development Tutorial 1 – Introduction

Windows Driver Development Tutorial 2 - How Our Driver Works

BCD batch

```
bcdedit /debug on

bcdedit /dbgsettings serial debugport:2 baudrate:115200

REG ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Debug Print Filter " /v DEFAULT /t REG_DWORD /d 0xf
```

Bespoke driver loader batch

```
sc create MyService binPath= "C:\Users\s0109817\Desktop\hellodriver.sys" type= kernel start= demand

sc start MyService

sc stop MyService

sc delete MyService
```

C. Pre and Post HDD MBR Difference Table

Row	Value															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	0	6C	0	0	0	0	0	12	0	0	0	0	0	0	0
64	F4	0	0	0	9	0	0	0	47	E8	3	3A	21	0	3A	76
80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
96	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
112	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
128	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
144	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
176	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
192	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
208	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
224	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
240	0	0	0	0	0	22	1F	0	0	0	0	0	0	0	0	0
256	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
272	22	3	0	2E	3	0	0	0	0	A1	0	0	0	0	0	0
288	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
304	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
336	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
352	0	0	0	0	0	0	0	0	0	0	1	E	0	0	0	0
368	1	1	0	0	0	0	0	0	0	3F	F1	27	CC	AC	48	9
384	C0	7	F	9	BB	CA	10	26	E2	28	FF	C9	4C	2A	25	49
400	17	2	53	19	45	13	1	41	1	52	17	1D	0	4F	1D	43
416	C	16	11	7	17	16	65	69	A	4F	45	D	1B	2	13	1F
432	67	3B	53	49	10	4F	0	19	1	16	1A	1D	2	64	D	7
448	48	1F	3D	31	3E	34	72	63	1D	1	4C	48	2E	1	4	59
464	21	16	1F	45	10	6F	2D	78	35	1	11	12	1	54	4E	7E
480	72	6C	2B	41	6C	74	2B	44	65	6C	20	74	6F	20	72	65
496	73	74	61	72	74	D	80	1	2B	A8	1	D7	0	0	0	0

D. Putative Malware Identified on HDDs (Identified Putative Malware Referenced)

	DISK 1 – 80GB	DISK 2 – 250GB	DISK 3 – 160GB	DISK 4 – 160GB	DISK 5 – 500GB	DISK 6 – 160GB	DISK 7 - 40GB	DISK 8 – 160GB
FORMSREGISTRY (TrendMicros, 2009)	X	X		X				
Hacker.com.cn_MUTEX (F-Secure, 2020) (Totalhash, 2020)		X		X		X +X(almost)		X
HOMEKEYLOGGER_MUTEX				X				
CHECKFORMUTEXES				X				
PORT_INSTANCE_MUTEX				X				
.COM.CN_MUTEXION (F-Secure, 2020)		X		X				X

	DISK 1 – 80GB	DISK 2 – 250GB	DISK 3 – 160GB	DISK 4 – 160GB	DISK 5 – 500GB	DISK 6 – 160GB	DISK 7 - 40GB	DISK 8 – 160GB
Hacker.com.cn_MUTEX (F-Secure, 2020)		X		X		X+X(almost)		X
STATREPORTERMUTEX (Hybrid-Analysis, 2018a)		X		X		X		X
REDIST_MUTEX_CHEC				X				
SACCUPDATER_MUTEX				X				X
IEXPLORE_MUTEX _AABBCCDDEEFF (TrendMicro, 2002)	X	X		X				X
HACK83.COMMUTEXTVERSION/RUN				X		X(almost)		X
CKR_MUTEX_BAD				X				
BER_OF_RAKPEER_MUTEXES		X						

	DISK 1 – 80GB	DISK 2 – 250GB	DISK 3 – 160GB	DISK 4 – 160GB	DISK 5 – 500GB	DISK 6 – 160GB	DISK 7 - 40GB	DISK 8 – 160GB
USEMUTEX KEYLOGGERUSEPASS		X						
LOGIN		X						
SEVENMUTEXN\RUNÿ Ä-ÿ————————————		X				X(almost)		
MIR3MUTEXLEGEND OF MIR 3		X				X		X(almost)
GKEMAM@?5?5?5?5MUTEX?5mutex?5?$DN?5?$CBv		X						
PITFILEWRITEMUTEX (Hybrid-Analysis, 2018)		X						
ERSACCUPDATER_MUTEXÚ&		X						
MICK_DOWNLOAD_MUTEXÿ2Ä0ÿ ————————————————		X				X		

	DISK 1 – 80GB	DISK 2 – 250GB	DISK 3 – 160GB	DISK 4 – 160GB	DISK 5 – 500GB	DISK 6 – 160GB	DISK 7 - 40GB	DISK 8 – 160GB
votnews.co (Hybrid-Analysis, 2019)		X				X		X
DBMIRROR_DBM_MUTEX		X						
FT_METADATA_MUTEX		X						
DTS_I_GOTMUTEXFROMWAIT		X						
TASK_MUTEX_12		X						
NACL_DESC_MUTEX		X						
LIB_LOG_MUTEX								
HIMLIB_L OG_MUTEX	X							
EX_CONVE RT_MUTEX	X							
TEM_EXIT_OWNED_MUTEX	X							

	DISK 1 – 80GB	DISK 2 – 250GB	DISK 3 – 160GB	DISK 4 – 160GB	DISK 5 – 500GB	DISK 6 – 160GB	DISK 7 - 40GB	DISK 8 – 160GB
MICROSOFT_WMDM_MUTEX	X							
EDC_DB_MUTEX	X							
MAPI_UIDGEN_MUTEX	X							
igfxpers.exe				X				
ASP_PERFMON_MUTEX (Hybrid-Analysis, 2016)			X	X	X			
IGFXHKMUTEXT (Hybrid-Analysis, 2019) (Totalhash, 2020)					X		X(almost)	
MSJVM-JPM-MUTEX					X			
NO_UCB OCI_NO_MUTEX					X			

	DISK 1 – 80GB	DISK 2 – 250GB	DISK 3 – 160GB	DISK 4 – 160GB	DISK 5 – 500GB	DISK 6 – 160GB	DISK 7 - 40GB	DISK 8 – 160GB
STPRXEIDGENMUTEX					X			
=HOMEKEYLOGGER_MUTEX					X			
KEYLOGGER_MUTEXtÿµ ______________________________						X		X
SU_MUTEX					X			
AC2005_MUTEX (Hybrid-Analysis, 2020)					X			
DTS_E_CANTGETMUTEX							X	
IMLIB_LO€G_MUTEX							X	
IGFX€HKMUTEX! (Hybrid-Analysis, 2018b)							X	
SL_MODEM_Setup_MUTEX							X	

	DISK 1 – 80GB	DISK 2 – 250GB	DISK 3 – 160GB	DISK 4 – 160GB	DISK 5 – 500GB	DISK 6 – 160GB	DISK 7 - 40GB	DISK 8 – 160GB
PDATER_SERVICE_MUTEX							X	
UIDGEN@_MUTEX (Sophos, 2019)	X(almost)						X	
RE.EXEHacker_MUTEX						X		
DarkCoderScMUTEX						X		X
MUTEX_sg3012_hot						X		
MS_ZJM_000001_MUTEX						X		
ION_EDONKEY_MUTEX						X		
NBA_MUTEX HookEnter (Hybrid-Analysis, 2019)						X		X

	DISK 1 – 80GB	DISK 2 – 250GB	DISK 3 – 160GB	DISK 4 – 160GB	DISK 5 – 500GB	DISK 6 – 160GB	DISK 7 - 40GB	DISK 8 – 160GB
NBA_DOWNLOAD_MUTEX (Hybrid-Analysis, 2019)						X(almost)		X
DTS_E_CANTGETMUTEX							X	
IMLIB_LO€G_MUTEX							X	
VCHOST.exeFH_MUTEX								X
EGHOSTIPARMOR								X
MUTEX1212 (Totalhash, 2020)								X
MS_ZJM_000001_MUTEX								X
POEVTINFMUTEX								X
GPigeon (Labs, 2020)								X

	DISK 1 – 80GB	DISK 2 – 250GB	DISK 3 – 160GB	DISK 4 – 160GB	DISK 5 – 500GB	DISK 6 – 160GB	DISK 7 - 40GB	DISK 8 – 160GB
REMOTE_THREAD_MUTEXURLDownloadToFile								X

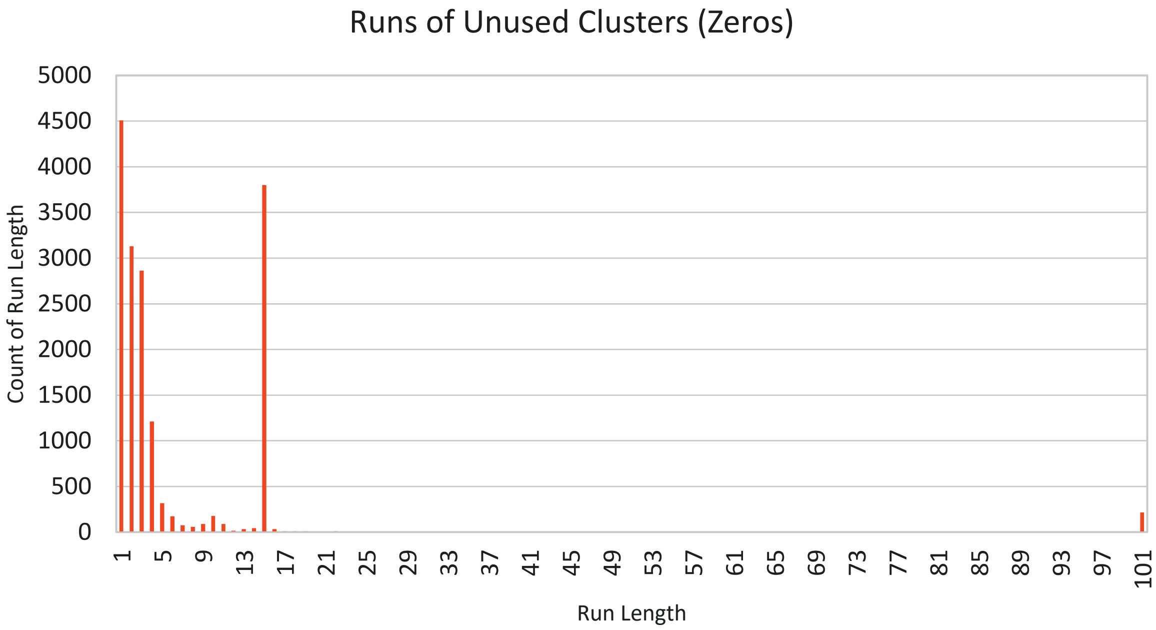
E. Modified Hosts File to Block Advertising and Malvertising

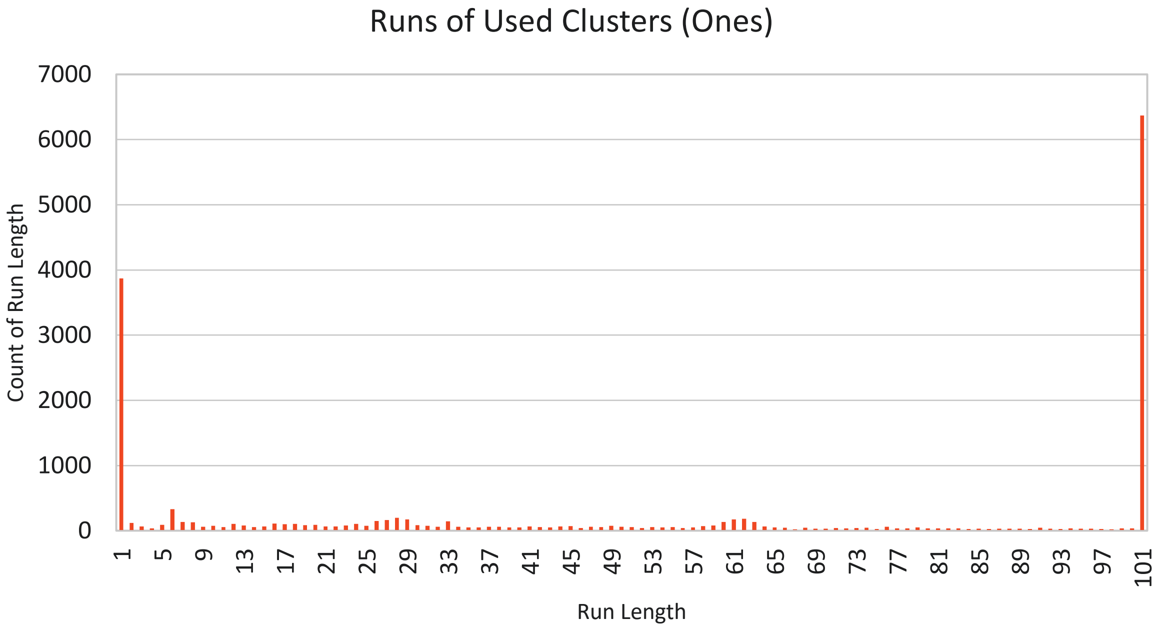
```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#          127.0.0.1       localhost
#          ::1             localhost

127.0.0.1       www.facebook.co.uk
127.0.0.1       www.facebook.com
127.0.0.1       www.youtube.com
127.0.0.1       prod.reuters.tv
127.0.0.1       queso-cdn.prod.reuters.tv
127.0.0.1       player.brightcove.net
```

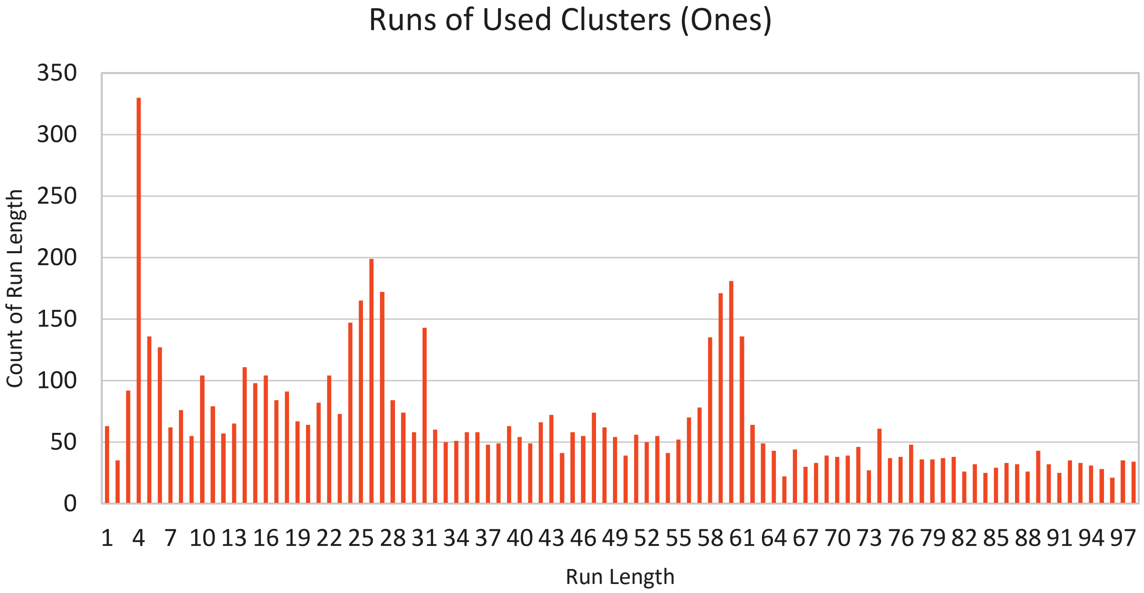
```
127.0.0.1       sadmin.brightcove.net
127.0.0.1       static.independent.co.uk
127.0.0.1       sb.scorecard.research.com
127.0.0.1       googletagmanager.com
127.0.0.1       static.chartbeat.com
127.0.0.1       ads.rubiconproject.com
127.0.0.1       ads.yahoo.com
127.0.0.1       c.amazon-adsystem.com
#
127.0.0.1       www.yandex.ru
127.0.0.1       www.optimizely.com
127.0.0.1       www.caslemedia.com
127.0.0.1       www.doubleverify.com
127.0.0.1       www.adnxs.com
127.0.0.1       www.skimresources.com
127.0.0.1       www.lkqd.net
127.0.0.1       www.tealiumiq.com
127.0.0.1       www.pubwise.io
127.0.0.1       www.google-analytics.com
127.0.0.1       www.doubleclick.net
127.0.0.1       www.quantserve.com
127.0.0.1       www.rubiconproject.com
127.0.0.1       www.omtrdc.net
127.0.0.1       www.openx.net
127.0.0.1       www.googlesyndication.com
127.0.0.1       www.bing.com
127.0.0.1       www.moatads.com
127.0.0.1       www.adsafeprotected.com
127.0.0.1       www.baidu.com
```

```
127.0.0.1       www.parsely.com
127.0.0.1       www.bluekai.com
127.0.0.1       www.pubmatic.com
127.0.0.1       www.2o7.net
127.0.0.1       www.criteo.com
127.0.0.1       www.cloudfront.net
127.0.0.1       www.googleadservices.com
127.0.0.1       www.cnzz.com
127.0.0.1       www.krxd.net
127.0.0.1       www.pinterest.com
#127.0.0.1       www.heap.io
#127.0.0.1       www.quantcast.com
#127.0.0.1       www.spotex.com
#
#127.0.0.1       www.virginmediabusiness.co.uk
#127.0.0.1       www.amazontechnologies.com
```

F. Cluster Length Analysis






Runs of Used Clusters (Ones)
Count of Run Length
Run Length
0
50
100
150
200
250
300
350
1
4
7
10
13
16
19
22
25
28
31
34
37
40
43
46
49
52
55
58
61
64
67
70
73
76
79
82
85
88
91
94
97

G. Analysis of Repeated Encryption Scheme (Insikt-Group and Rapid7, 2019)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	8	12	20	32	52	84	136	220	101	66	167	233	145	123	13
1	136	149	30	179	209	133	87	220	52	17	69	86	155	241	141	127
2	13	140	153	38	191	229	165	139	49	188	237	170	152	67	219	31
3	250	26	21	47	68	115	183	43	226	14	240	254	239	238	222	205
4	172	122	39	161	200	106	51	157	208	110	63	173	236	154	135	34
5	169	203	117	65	182	247	174	166	85	251	81	77	158	235	138	118
6	1	119	120	239	104	88	192	25	217	242	204	191	140	76	216	37
7	253	35	33	68	101	169	15	184	199	128	72	200	17	217	234	196
8	175	116	36	152	188	85	18	103	121	224	90	59	149	208	102	55
9	157	212	114	71	185	1	186	187	118	50	168	218	131	94	225	64
10	34	98	132	230	107	82	189	16	205	221	171	137	53	190	243	178
11	166	89	0	89	89	178	12	190	202	137	84	221	50	16	66	82
12	148	230	123	98	221	64	30	94	124	218	87	50	137	187	69	1
13	70	71	141	212	98	55	153	208	106	59	165	224	134	103	237	85
14	67	152	219	116	80	196	21	217	238	200	183	128	56	184	240	169
15	154	68	222	35	2	37	39	76	115	191	51	242	38	25	63	88
16	151	239	135	119	254	118	117	235	97	77	174	251	170	166	81	247
17	73	65	138	203	86	34	120	154	19	173	192	110	47	157	204	106
18	55	161	216	122	83	205	33	238	16	254	15	14	29	43	72	115
19	187	47	234	26	5	31	36	67	103	170	18	188	206	139	90	229
20	64	38	102	140	242	127	114	241	100	86	186	17	203	220	168	133
21	46	179	225	149	119	13	132	145	22	167	189	101	35	136	171	52
22	223	20	243	8	251	4	0	4	4	8	12	20	32	52	84	136

H. Firefox website Blocker

manifest.json

```
{
  "name": "Website Blocker",
  "description": "Block Websites based on Filter",
  "version": "1.0",
  "manifest_version": 2,
  "author": "Me",
  "permissions": [
    "webRequestBlocking",
    "webRequest",
    "storage",
    "activeTab",
    "tabs",
    "http://*/*",
    "https://*/*"
  ],

  "web_accessible_resources": ["/siteblkr.html"],

  "background" : {
    "scripts":  ["siteblkr.js"]
  }

}
```

siteblkr.js

```
// Website Blocker

// Block Websites based on Filter in the variable "urllist"


var urllist = ['*://*.co.uk/*', '*://*.com/*'];


function blocksite(details) {

    return {redirectUrl: browser.runtime.getURL('/siteblkr.html')};

//   return {cancel: true}


;

}


browser.webRequest.onBeforeRequest.addListener(blocksite, {urls: urllist},
['blocking']);
```

siteblkr.html

```
<!DOCTYPE html>
<html>

<head>
  <title>Blocked Website</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>

<body>
  <div class="spacer"></div>
  <div class="content">
    <span>This website has been blocked<br /></span>
    <span>Please contact SysAdmin<br /></span>
  </div>
  <div class="spacer"></div>
</body>

</html>
```