UNIVERSITY OF AMSTERDAM

Faculty of Physics, Mathematics and Informatics
Graduate School of Informatics
System and Network Engineering MSc

# Analysing the feasibility of portable passive detection of Advanced Persistent Threats

Research Project 2

*Student:*
Guido Kroon
guido.kroon@os3.nl

*Supervisors:*
Marco Davids, SIDN
Christian Hesselman, SIDN

August 26, 2016

# Abstract

The Foundation for Internet Domain Registration Netherlands (SIDN) expressed their interest in researching the feasibility of portable passive detection of Advanced Persistent Threat (APT). An APT is a highly sophisticated threat that high-profile organisations and governments need to arm themselves against. This request of SIDN has been defined in a project proposal, which includes the following research question: *Can a portable passive Advanced Persistent Threat (APT) sensor be designed for easy deployment on the network and to detect the presence of potential APTs?* To determine the feasibility of such a sensor required a study on how APTs execute their attacks and what their characteristics are. It also required to research which detection methods are currently being used and what current research developments propose new ways of detection.

Based on researched information and analysis it can be determined that APTs are not only technically advanced, but are of an entirely different magnitude, compared with existing traditional threats. These threats operate differently and are not a mere continuation of traditional threats. It follows that current detection mechanisms are no longer sufficient, as they were designed to detect threats of a different magnitude. This created the need for new detection mechanisms that are designed to detect these threats based on their typical characteristics.

As a result, this research concluded that a portable detection device, as SIDN requested, is therefore unsuitable to respond to the typical characteristics of APTs, leading to an adverse answer to the research question. While there is as of now no proven effective solution to this problem, advances in this field of science already identify potential detection solutions. There are still opportunities for future work and SIDN is recommended to look into these opportunities.

# Acknowledgement

I would like to thank SIDN for providing this interesting project, especially Marco Davids for providing me with feedback and ideas for this project. Special thanks also go to the System and Network Engineering (SNE) staff for their support and guidance throughout the entire study, as well as to Kees Buiter, Koen Veelenturf, Stella Vouteva for proofreading my report for punctuation, and last but not least my father for his moral support.

Guido Kroon

# Contents

**1 Introduction** **3**
1.1 SIDN and research motivation 3
1.2 Research questions 3
1.3 Scope 4
1.4 Approach 4
1.5 Related work 4

**2 About Advanced Persistent Threats** **6**
2.1 Modus operandi of the APT 6
2.2 Comparison of modus operandi models 8
2.3 The Attack Pyramid and known APT examples 9
2.3.1 Operation Aurora 10
2.3.2 Stuxnet 10
2.3.3 Operation Shady RAT 11
2.3.4 Operation Socialist 12
2.4 Characteristics of the APT 13
2.5 Overview of the APT 14

**3 The current state of affairs regarding detection** **16**
3.1 Current defensive countermeasures 16
3.2 Offensive techniques and corresponding defensive countermeasures 17
3.3 Standard network detection shortcomings 17
3.3.1 Obvious suspicious activities 18
3.3.2 NIDS evasion 18
3.3.3 The reliable baseline problem 18
3.4 Performing big data analysis of security related events 19
3.4.1 Obstacles to overcome 19

**4 Discussion** **20**

**5 Conclusion** **22**

**6 Recommendations and future work** **24**

**Bibliography** **25**

**Glossary** **30**

# Chapter 1

# Introduction

For as long as there have been personal computers, people have used them for both good and bad purposes. After the first few personal computers entered the market in the 1980s, it did not take long before the first computer viruses were created [31]. Back then, computer viruses were simple and relatively harmless, but computer crime has since become more severe, large-scale, and results in a detrimental effect on affected parties [38]. Over the last decade, the attacks have gotten so sophisticated that the industry started calling these attacks Advanced Persistent Threats (APTs) [5]. These threats are often orchestrated by skillful professionals involved in organised crime [79][13], or even state actors involved in clandestine or covert operations, such as involving espionage [72] and sabotage [10]. The attacks are premeditated, aimed at specific targets, with a clear, long-term objective in mind, such as intelligence gathering, financial enrichment, or disruption of (foreign) infrastructures or critical services.

Organisations continue protecting themselves against numerous digital attacks by means of the usual firewalls, anti-virus suites, Network Intrusion Detection Systems (NIDSs), Network Intrusion Protection Systems (NIPSs) and Security Information and Event Management (SIEM) solutions. However, these security controls are by no means a guarantee to stop malicious actors that are especially determined to break into the network.

## 1.1 SIDN and research motivation

SIDN is the foundation responsible for registration of Internet domain names in the `nl` country code top-level domain (ccTLD). To date, SIDN is responsible for over 5,5 million registered domain names and maintains the availability and security of all its registrations. Due to this responsibility, SIDN needs to constantly remain vigilant against threats. For example, disruption of service could result in a global downtime of registered domain names in the `nl` ccTLD. And corruption of such registered domain names means that users trying to resolve a domain name residing in the `nl` zone, could be redirected to malicious servers that the attackers have under their control. Especially public or governmental services that have their domain name registered at SIDN are affected by such attacks. So is the Internet community as a whole that relies on the integrity of registered domain names residing inside the `nl` ccTLD.

As such, SIDN continuously seeks to improve their security to defend themselves against such threats, and expressed their interest into exploring the characteristics and *modus operandi* of APTs. They also wish to know how organisations can detect such suspicious behaviour, and finally wishes to design a portable, passive prototype to effectively detect such threats. This lead to a project proposal, containing the motivation, scope and research questions for this research, which was agreed upon by both the University of Amsterdam and SIDN.

## 1.2 Research questions

This research aims to better understand the modus operandi of APTs and, consequently, how to detect the presence of a potential APT on the network. Therefore, this paper derived the following main research question:

*Can a portable passive Advanced Persistent Threat (APT) sensor be designed for easy deployment on the network and to detect the presence of potential APTs?*

If the above research question is answered positively, then that means such a sensor can indeed be designed. To help answer this main research question, the following sub-questions have been devised:

- What is the modus operandi of an APT?
- What are the characteristics of APTs?
- What is the current state of the art regarding detection of APTs?

## 1.3 Scope

This project will focus on analysing the modus operandi and the characteristics of an APT, how such a threat can be detected and whether it is possible to design a basic portable prototype. This project will not focus on social engineering aspects, nor does this project aims to prevent or detect vulnerabilities in systems. Furthermore, as per the requirements of SIDN, the APT sensor itself should be an easy to deploy, small, portable device, such as a single board computer. The sensor should not be designed to lure intruders in, rather it should be just an inconspicuous device that generates alarms when hit. Regarding detection, the sensor will not do detection or monitoring of activity on its own platform. The sensor should only focuses on detecting suspicious behaviour on the network plane.

## 1.4 Approach

The approach to this project contains a number of steps. In order to answer the main research question, to research the feasibility of a portable passive APT sensor, first requires more research into the threat that it needs to detect. Hence is why the first two research subquestions have been devised to gain more insight into how an APT operates, and which typical characteristics they possess, which sets them apart from other threats. Several models and other literature from the industry and current research developments will be consulted and compared during the first phase of this project. This literature study shows a representation of characteristics of attacks, as well as the current state of research developments. Further comments, observations and additions will be reserved for the discussion chapter.

After knowing more about the threats that need to be detected, the second phase of this project will focus on identifying currently feasible detection methods, as well as new scientific progress regarding new detection methods and models. Armed with this new information this paper will analyse the feasibility of detection of APTs with portable passive sensors, thereby answering the main research question.

## 1.5 Related work

Previous research has already been performed regarding APTs - how they operate, supported by publicly known examples, and how to detect this new type of threat. In 2011, Colin Tankard published a basic article [64] describing this new threat, its characteristics and examples of attacks. He concludes that continuous (anomaly) detection, with controls continuously monitored for their effectiveness, and real-time alerting is necessary to stay ahead of these new threats.

In 2013, Virvilis and Gritzalis analysed [74] similar APT attacks, such as Duqu, Stuxnet, Flame and Red October. After analysis and comparison between these attacks, they concluded that these complex attacks cannot be solved using a single security appliance. They point out that a wide-range of security countermeasures and hardening procedures to provide a multi-layered and robust defense are needed instead.

Chen et al. recognised in 2014 that an objective approach to the APT issue is currently lacking and therefore present a study [9] on APTs. They described distinguishing characteristics, their attack model and analysing techniques commonly seen in APT attacks. They also conclude that these threats are so sophisticated that no single solution exists to effectively detect their presence. They then enumerate some countermeasures that can help to mitigate APTs, thereby highlighting the directions for future research. Countermeasures include more security awareness training and to keep deploying traditional detection and prevention systems to make things harder for potential intruders. But more importantly, to continue behavioural analysis on these complex malware, to focus more on anomaly detection and on, what they refer to as, more intelligence-driven detection.

Current research indeed seems to agree and is already proposing new models [1][35][49][30] in order to aggregate numerous security related events of all security systems, and performing big data analysis and anomaly detection.

# Chapter 2

# About Advanced Persistent Threats

In order to detect APT attacks, it is first required to gain insight into the modalities of such attacks. This chapter will describe the typical modus operandi, as well as the typical characteristics of the APT.

## 2.1 Modus operandi of the APT

The term Advanced Persistent Threat (APT) is used to refer to an attack, or series of attacks, performed by *skilled and well-resourced criminals who employ a wide range of sophisticated reconnaissance and information-gathering tools, as well as attack tools and methods* [64]. APTs distinguish themselves from other, classical types of threats in the sense that these are targeted and predetermined attacks. They are, in contrast to classical threats, much more skillful and ongoing for extended periods of time due to their stealthy approach, circumventing all kinds of security controls on the network. APTs deliberately target parties to launch a series of cunning attacks to circumvent security and gain control of assets on the network.

An APT first proceeds by performing a number of steps in order to complete their objective, which Dell SecureWorks refers to as the *Kill Chain* [56]. Figure 2.1 shows the Kill Chain, outlining the process an APT typically follows.

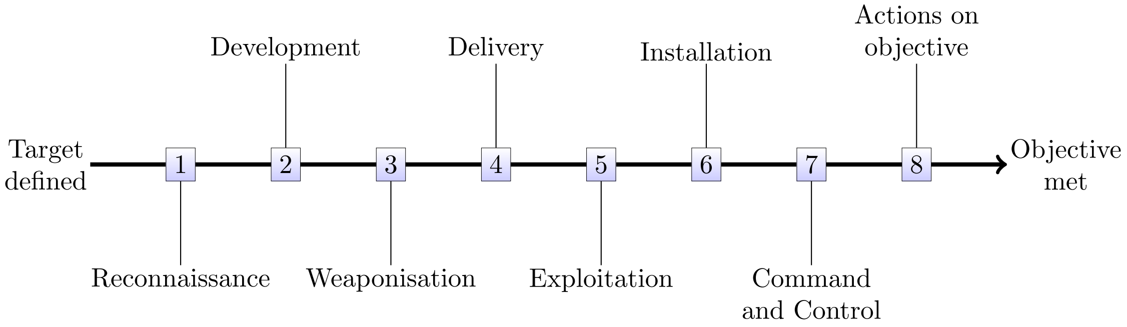

Figure 2.1: Kill Chain [56].

The Kill Chain steps in figure 2.1 are described as follows:

1. **Reconnaissance**: gathering of intelligence about the target. Usually by means of Open Source Intelligence (OSINT) or other non-technical means;

2. **Development**: gathering of technical intelligence about the target;

3. **Weaponisation**: development of a malicious payload used for attacking the target. Usually a Remote Access Toolkit (RAT) trojan horse;

4. **Delivery**: transmission of the malicious payload and tools to the target;

5. **Exploitation**: performing the attack at the target, also by means of custom zero day exploits and/or social engineering;

6. **Installation**: necessary methods and artefacts left on compromised systems in order to implant malicious code;

7. **Command and Control**: interaction with compromised resources which also serve as the point of data exfiltration;

8. **Actions on objective**: Exfiltration of classified information.

Giura et al. show a more concise model [28] that outlines these stages (see figure 2.2). They see the delivery phase as both building and delivering a malicious payload on the network, typically by making use of a *spear phishing* attack. A spear phishing attack is a specific and directed form of a phishing attack that aims to deceive a specific victim. For instance, to make them open a malicious attachment that contains an exploit, enabling an APT to break into the victim's computer.

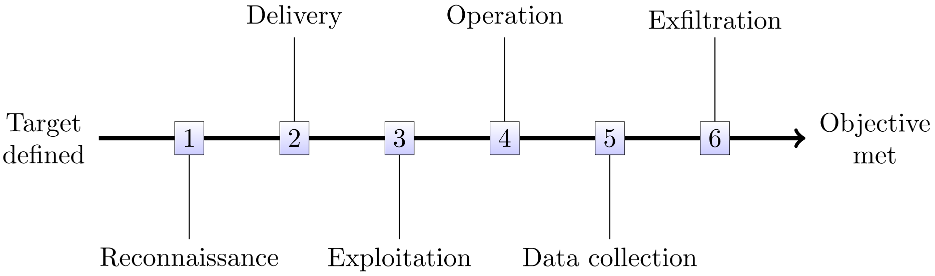

Figure 2.2: Typical stages of an APT [28].

The steps in figure 2.2 are described as follows:

1. **Reconnaissance**: gathering of information;

2. **Delivery**: delivery of malicious payloads, or performing a social engineering attack;

3. **Exploitation**: exploiting a vulnerability using the installed malicious payload;

4. **Operation**: privilege escalation and making presence persistent, such as installing a backdoor for easy future access;

5. **Data collection**: harvesting of credentials to collect valuable information

6. **Exfiltration**: packaging (and encryption) of collected information to copy over to external servers. Usually multiple servers for obfuscation (counter forensics).

When observing both models in figure 2.1 and figure 2.2, a similar modus operandi of the APT is described. More interestingly however is that there appears to be an overlap to how a typical *penetration tester* usually operates. Both aforementioned models share a similar process as the Zero Entry Hacking (ZEH) methodology [23], which a penetration tester usually follows when *pentesting* the network infrastructure of a customer. The ZEH methodology in figure 2.3 consist of four steps.

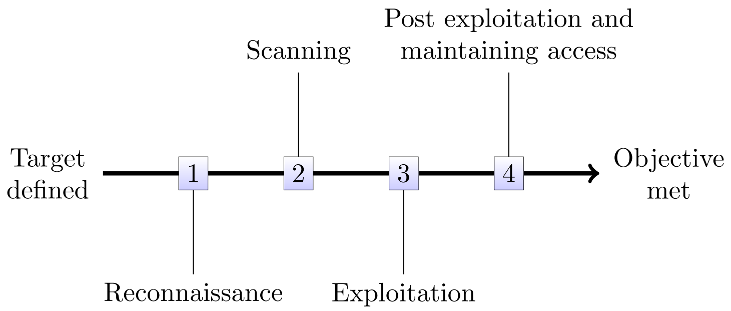

Figure 2.3: Zero Entry Hacking (ZEH) methodology [23].

The steps in figure 2.3 are described as follows:

1. **Reconnaissance**: gathering of information;
2. **Scanning**: subdivision of two distinct steps;
   (a) **Port scanning**: to find potentially exploitable services;
   (b) **Vulnerability** scanning: to find feasible exploits on newly discovered services;
3. **Exploitation**: exploiting a vulnerability using newly discovered vulnerable services;
4. **Post exploitation and maintaining access**: maintaining access by making access persistent (e.g. by installing a backdoor). From here, harvesting and exfiltration of valuable information also begins.

Here it can be observed that APTs follow roughly the same procedure as the earlier models that describe the operation of a different actor.

## 2.2 Comparison of modus operandi models

When observing all three models in figures 2.1, 2.2, and 2.3 next to each other we can observe they all share the same general steps. Some models just make a clear distinction between certain steps, whereas a different model combines them into a single step (see table 2.1).

| | **Kill Chain** [56] | **Giura et al.** [28] | **Zero Entry Hacking** [23] |
|---|---|---|---|
| **1** | Reconnaissance | Reconnaissance | Reconnaissance |
| **2** | | | Scanning |
| **3** | Development | Delivery | Exploitation |
| **4** | Weaponisation | | |
| **5** | Delivery | | |
| **6** | Exploitation | Exploitation | |
| **7** | Installation | Operation | Post exploitation and maintaining access |
| **8** | Command & Control | | |
| **9** | Actions on objective | Data collection | |
| **10** | | Exfiltration | |

Table 2.1: All three models show a similar procedure.

The following similarities and slight differences can be observed:

- The ZEH model distinguishes the first two *Reconnaissance* and *Scanning* phases, which separates gathering of information, and scanning newly found systems for exploitable vulnerabilities. The other two models just combine these steps in their *Reconnaissance* phase;

- The Kill Chain distinguishes the *Development*, *Weaponisation* and *Delivery* phases, whereas the Giura et al. model joins these phases together in its *Delivery* phase;
- The Kill Chain further distinguishes the *Installation* and the *Command & Control* phases, which the Giura et al. model merges together in its *Operation* phase;
- The Giura et al model. separates the *Data collection* and *Exfiltration* phases, whereas the Kill Chain combines these actions together in its *Actions on objective* phase;
- The ZEH model simply merges all activity after its *Exploitation* phase together in the *Post exploitation and maintaining access* phase, whereas the other two models subdivide these activities over more phases.

Despite some slight differences, there is a certain overlap between the modus operandi of an APT and a penetration tester. The first two models aim to provide insight in how APT carries out its objective, whereas the third model shows a typical procedure a penetration tester usually follows. Of course, a distinct difference between these different actors can be found in their underlying motives. A penetration tester is typically bound to responsible disclosure, whereas the goal of an APT can be to cause damage without further ramifications. One role of the penetration tester would be to report weaknesses, discovered during their attack. An APT, on the other hand, could publicly release confidential information, without informing the target.

To summarise, looking at the aforementioned models, there seems to be a consensus regarding the modus operandi of the APT, albeit with some slight divisions between steps. Firstly, the threat performs his preliminary research about the target, which most models refer to as the *Reconnaissance* phase. Within this step the attacker aims to gain more insight into the target, also to find potential weaknesses for exploitation. After this reconnaissance, the attacker proceeds with devising a plan of attack, a stratagem to infiltrate the target. Typically by means of developing a custom attack to be launched at a weakness within a target's identified weakness. Once successfully exploited this weakness, the attacker gained its first foothold into the target. From there, the attacker proceeds with further privilege escalation and installation of malware to ensure easy access when returning, as well as retain full control of the compromised assets. Upon realisation, the attacker proceeds with the final steps of executing its goal. This could range from a variety of motives, such as espionage, stealing classified information, or sabotaging a critical asset or service.

## 2.3 The Attack Pyramid and known APT examples

The *Attack Pyramid* [28] (see figure 2.4) of Giura et al. proposes a new model to depict the evolution of an APT. This model shows that the APT may span several attack planes before it achieves its final objective, which gains insight into how an APT operates.

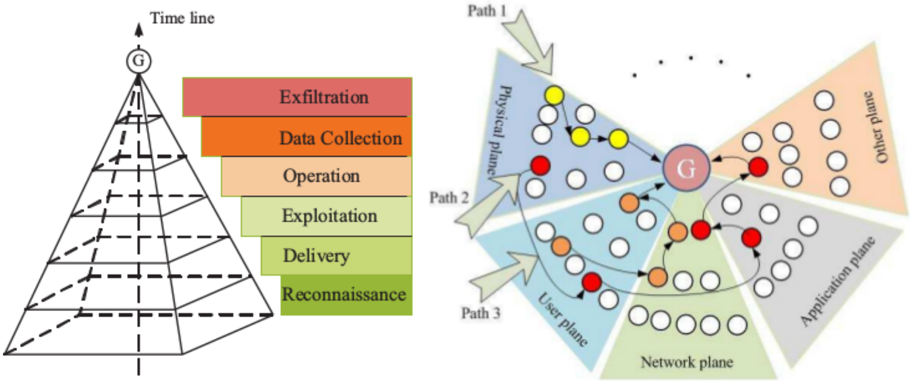

Figure 2.4: Attack Pyramid [28].

On the left of the picture we can see the earlier mentioned linear model in figure 2.2. However, it now adds another dimension, introducing several attack planes. These attack planes are shown in an unfolded pyramid to the right. When folding the pyramid back together, each side of the pyramid represents an attack plane. The threat could then be tracked during each phase of its attack, starting from the bottom at the *Reconnaissance* phase, all the up to the top to the *Exfiltration* phase. During this time, the APT could be tracked over different sides of the pyramid, meaning that the threat moved through different attack planes.

To illustrate this model, the attacker may first exploit a weakness on the *user plane* (see figure 2.4) by using a spear phishing attack. Once the victim is persuaded to open, for example, a malicious attachment, this may trigger a vulnerability in the application that opens the attachment. This creates an opportunity for the attacker to exploit this vulnerability, at which point the attack has evolved from the *user plane*, to the application plane. From there, the attacker has a way into a compromised system. From here the attack may evolve onto the network plane, in order to infiltrate a valuable target, such a server containing classified information.

The researchers discussed and elaborated their model, but have not demonstrated this model with some actual APT examples. This section will therefore analyse its applicability by using some publicly known APT examples, in order to gain a better insight into how and where such attacks evolve over time. To use this model, a couple of publicly known APT examples have been analysed in the following subsections, which are then placed inside the Attack Pyramid at section 2.3.4. The descriptions will also highlight where the attack took place inside the Attack Pyramid. For example, during Reconnaissance on the user plane.

### 2.3.1 Operation Aurora

One notable early example of a large scale APT was Operation Aurora, originating from China in 2010. It is believed the attacks were already ongoing since 2009, but no records of that clearly support it. However, the Aurora codebase does have compiler time stamps dating back to May 2006 [62]. The large operation aimed to steal source code from several high-profile companies, such as Google and Adobe, as well as from parties within the financial and defense sectors [79].

The attack started with spear phishing victims by sending a message from a trusted source (*Reconnaissance on user plane*). The spear phishing mails were sent to the victims over the network (*Delivery on network plane*). The message, typically an email or an instant message, contained a link to a website, which, when browsed to, loads a malicious JavaScript payload. The payload exploits a zero-day vulnerability in Internet Explorer 6 [15][67], enabling the attacker to execute arbitrary code [64][5] (*Exploitation on application plane*). An example of this exploit [45] uses Python to launch a web server, which serves a document with a malicious JavaScript payload. The browser loads a dummy picture, which then triggers the execution of the payload. This effectively compromised the victim's system, at which point the attacker gained a foothold into the organisation.

The attack progressed by installing custom made malware on the software configuration management (SCM) systems, which was received through the initial infection. The backdoor program is decompressed from the initial infection and an embedded library is inserted into the Windows system32 directory and loads it as a service. The library is then modified to avoid detection, and the initial malware deletes itself from the system [80]. The malware communicates over a custom made Command and Control (C2) protocol, which abuses TCP port 443, a port commonly associated with encrypted traffic, such as HTTPS [9]. The compromised system is then part of a C2 network (*Operation on network plane*). The SCM systems held proprietary source code, which was collected using the malware (*Data collection on application plane*). Using encryption over a standard port which is often used for encrypted communication anyway makes the exfiltration of data less suspicious when uploading the proprietary source code to their own servers [5] (*Exfiltration on network plane*). The attack got detected by recurring connections to suspicious domains [73].

### 2.3.2 Stuxnet

After Operation Aurora, the *Stuxnet* computer worm was discovered [24] in 2010, yet early variants of the Stuxnet code stemming from 2009 have been found [36]. Stuxnet is an advanced malware, allegedly developed by an Israeli/United States joint effort, although neither governments publicly acknowledged this [10]. It was speculated to have been used for corporate espionage, but was actually developed to attack a specific military target, namely the nuclear power programme in Iran [37].

The attack was aimed at air gapped (i.e., systems not connected to any untrusted networks) industrial Supervisory Control and Data Acquisition systems (*Reconnaissance on physical plane*). Infection was caused by physically attached media, such as a USB flash drive (*Delivery on physical plane*). From there, the Stuxnet worm exploited multiple zero-day vulnerabilities on the systems. For example, it exploited the publicly disclosed MS10-046 [68][14], by exploiting a remote code execution vulnerability inside the Windows shell through a special crafted `.lnk` or `.pif` shortcut file. Stuxnet also exploited the already known MS08-067[66][17] vulnerability, which allowed for remote code execution by exploiting the Server service, which was vulnerable to special crafted RPC requests. It also exploited MS10-061 [69][19], a new vulnerability inside the Print Spooler service, again allowing for remote code execution through special crafted print requests over RPC. It also performed other means of privilege escalation by exploiting MS10-073[70] and MS10-092[71][18] (*Exploitation on application plane*). Stuxnet also installed a rootkit that obfuscated its binaries and masqueraded the malicious PLC code as values within limits of normal operation [74]. It also performed anti-virus evasion techniques and utilised a distributed updating system (peer-to-peer) [43]. If a Stuxnet worm would get updated to a newer version, it would use the LAN to automatically update the other local Stuxnet installations as well [25] (*Operation on network plane*). The worm's objective was to modify the code running in Programmable Logic Controllers (PLCs), intercepting communication between the industrial control systems and the PLCs they are connected to [25]. Stuxnet is known to only attack specific PLCs, by installing malware onto the PLC, resulting into periodic changes in rotation speed of the centrifuges. When these centrifuges were rotating outside their programmed limit, it eventually destroyed them. The malware would also contact C2 servers (which addresses were hard-coded in the malware), only if it detected an Internet connection [36].

Stuxnet differs from Aurora in that the initial infection would take place on air gapped systems; the actual systems needed to be infected physically. Another difference is that Stuxnet was not collecting or exfiltrating any data. It's sole purpose was sabotage, which the Attack Pyramid model unfortunately does not allow for. Stuxnet also makes use of several "off-the-shelve" zero-day exploits, which Farwell & Rohozinski claim [25] that this makes Stuxnet more hacky and stitched together than sophisticated. They do, however, agree that the amalgam of components within Stuxnet contributes to its concealment, pointing out that *deliberate ambiguity is an effective shield against retribution.* Karnouskos claims [36] its modularity, features and concealment through code reusage sufficiently demonstrated its sophistication. Yet, it did make the worm quickly and effectively disarmed. It would appear that the malware was causing arbitrary computer reboots and crashes, which a VirusBlokAda anti-virus programmer named Sergey Ulasen noticed. After investigation, the team found anomalies on infected systems and soon found out about the rootkit, two stolen digital certificates and more complex code causing the crashes [11], all of which led to the discovery of the Stuxnet worm.

### 2.3.3 Operation Shady RAT

Operation Shady RAT was a 2011 large-scale attack, predominantly targeting the US and attacking for over 70 global parties within all kinds of sectors, such as IT, energy, governmental, non-profits, financial, and more. Earliest log records found date back from 2006, although McAfee speculates that the attack has likely been set in motion even before that [3]. The attackers have allegedly stolen intellectual property, bidding information, prospecting data and computerised topographical maps "worth millions of dollars" showing the locations of potential oil reserves [13]. Exfiltrated information is estimated at over a petabyte [8].

Similar to Operation Aurora, the attack relied on spear phishing attacks (*Reconnaissance on user plane*). Sources claim [29][46] that the victim needed to click on a malicious link he or she received via email, embedded in a malicious Microsoft Excel file (*Delivery on network plane*). The malicious Excel file would create and open a clean copy of the Excel file, thereby not making the victim suspicious. During this time, the remote access tool (RAT) is being installed on the system (*Exploitation on application plane*). The RAT would create a backdoor for the attackers to infiltrate the network (*Operation on network plane*), escalate user privileges, collect data (Data collection on application plane) and to begin exfiltrating data (*Exfiltration on network plane*). In March later that year, McAfee discovered [29] the logs stored on the attackers' servers, which enabled the identification of victims by name (using their IP addresses) and to track the pattern of infections in detail. It was after these findings on the C2 servers that investigation pointed out Chinese involvement of the attacks.

McAfee claims [3] the attack was very sophisticated, but this has been contested by rival companies

Symantec [46] and Kaspersky [42]. Both are claiming that the attacks used were relatively simple (for example, the Microsoft Excel attack was already known [46]) and therefore not worthy of classifying this threat as an APT. SANS Institute [42], however, notices that although the attacks were not that sophisticated, they were nonetheless a real threat and it highlights well-organised and funded attacks coming from China.

### 2.3.4 Operation Socialist

Governments are also known for espionage, which has evolved into highly advanced digital attacks over the years. In 2013, it was revealed that the British intelligence agency, the Government Communications Headquarters (GCHQ), spied on the Belgacom International Carrier Services (BICS), a subsidiary of the main Belgian telecommunications company, Belgacom [72]. According to leaked documents, this attack was already ongoing since 2011 [34]. Belgacom claims that the GCHQ breached hundreds of their internal systems, yet played down on the extent of the compromise by insisting that no customer data was compromised during the attack. However, classified GCHQ documents show that the GCHQ was also able to intercept encrypted and unencrypted streams of private communications handled by Belgacom [27].

The attack started with the agency secretly mapping out the Belgacom network and identifying key employees related in maintenance and security. Leaked documents reveal that the agency developed a special tool, named NOCTURNAL SURGE, for looking up these key employees [34] (*Reconnaissance on the network plane*). After identifying several employees, the agency proceeded by implanting malware onto network, a so-called *Quantum Insert attack* [34], a type of watering hole attack [32] in which the potential targets' browsing habits are observed for recurring visited domains. The attack caused an instant redirect to a fake LinkedIn page once these specific employees browsed to this domain, thinking they connected to the official LinkedIn page [33] (*Delivery on the network plane*). The malicious LinkedIn page further infected the employees' computers, allowing the Government Communications Headquarters (GCHQ) to infiltrate their systems [41]. Regin, as the malware is named, is known to infect a system through several ways, such as through web browsers or exploiting applications (*Exploitation on application plane*). It features a modular design, similar but not related to Stuxnet, giving it flexibility to equip the malware with several payloads [6]. Usually the payload features a RAT (*Operation on network plane*), which can perform actions such as capturing screenshots, gaining control over the mouse, harvesting passwords, monitoring network traffic and recovering deleted files (*Data collection on application and network plane*). Other payload modules were also discovered, such as a Microsoft IIS web server traffic monitor, a traffic sniffer of the administration of mobile telephone base station controllers [51]. Some variants also contained the Qwerty keylogger, logging all input it receives from the keyboard. Qwerty uses two libraries and a malicious driver for registering the plugin, and special routines to perform the actual keylogging [6] (Operation on app).

The attack first got detected by Belgacom after a faulty email server. Their internal security team suspected some form of malware when they could not figure out why the email server stopped receiving emails. They hired the Dutch computer security firm Fox-IT to investigate the incident, which soon noticed suspicious files disguised as legitimate Microsoft software. Fox-IT then worked together with Belgian's federal computer crime unit and military intelligence to further investigate the incident, which finally lead to the discovery of the malware [34].

### Displaying the APT examples inside the Attack Pyramid model

Now that several APT examples have been technically described, they can be placed inside the Attack Pyramid (see figure 2.4). For simplification, instead of a pyramid layout, a table has been used to map the APTs in (see table 2.2). Each letter corresponds to a certain APT example, of which the key is described in the caption of the same table. For example, each *A* occurrence in the table stands for Operation Aurora.

| | Physical plane | User plane | Network plane | Application plane |
|---|---|---|---|---|
| **Reconnaissance** | S | A, R | O | |
| **Delivery** | S | | A, R, O | |
| **Exploitation** | | | | A, S, R, O |
| **Operation** | | | A, S, R, O | |
| **Data collection** | | | O | A, R, O |
| **Exfiltration** | | | A, R, O | |

Table 2.2: Mapping the publicly known APT examples in the Attack Pyramid; Operation Aurora (A), Stuxnet (S), Operation Shady RAT (R), Operation Socialist (O).

As this report strictly focuses on the network plane, this mostly narrows it down to the *Delivery*, *Operation* and *Exfiltration* phases. Note however, in the case of Stuxnet, it is only during the Operation phase that the threat may be detected (such as while it is communicating to C2 servers). Remember that Stuxnet targeted air gapped systems, meaning that physical access to the machines was necessary for initial infection.

Also note that some APTs aim to sabotage (such as Stuxnet) and do not aim to exfiltrate data. Unfortunately, the Attack Pyramid, nor any of the models previously described in section 2.1 allow for representing sabotage. Hence is why Stuxnet is not placed in any of the attack planes in the *Data collection* and *Exfiltration* phases.

## 2.4 Characteristics of the APT

Now that it is known how these advanced threats operate, as described in section 2.3, this allows for describing the typical characteristics of an APT. A typical APT has the following (non-exhaustive) characteristics, also supported by numerous other sources [16][56][64][77][26]:

- **Inquisitive**: a strong desire to know as much as possible about potential weaknesses of the target, such as by means of OSINT or social engineering. This is an important part, prior to their infiltration. An APT differs in this from other unorganised groups or script kiddies. Doing one's research is necessary for an attack in which an APT needs to get away with as much as valuable information as possible. This is something the low hanging fruit of the threats typically do not bother with in the first place, because these threats rarely aim to infiltrate a target as long as an APT would;

- **Stealthy approach**: circumventing all kinds of security controls to avoid detection. An APT needs to do so to prolong its existence on the network, which it acquired during their reconnaissance. This also means that an APT will spend a significant amount of time cleaning up after committing the act. This involves, for example, removing traces, wiping logs and removing malware that is no longer needed. For example, Operation Socialist masqueraded their malware as legitimate software, demonstration the attacker made an effort to make the malware seem endogenous;

- **Preparation**: premeditated plan of execution by using newly acquired information. This is similar to other types of threats, but an APT really goes the extra mile in comparison to less sophisticated threats. An APT will spend a lot of time fingerprinting as much resources on the network, finding out what services are running and then devising a plan of attack using this newly acquired information;

- **Infiltration**: exploiting a vulnerable asset to gain a foothold into the target. These type of initial attacks may not exclusively involve exploiting a digital weakness at first. Oftentimes an APT may also use spear phishing attacks to gain a target's trust and persuade them to do something. Typically, to have them run an infected email attachment, triggering a vulnerability in the software, enabling an APT to compromise the system;

- **Resourceful**: an APT is known for its sophisticated and custom designed attacks. Some attacks demand custom built malware to be used during attack, something lesser threats would most likely not bother with. For example, Operation Aurora used custom malware to steal intellectual property and Operation Socialist used custom malware to collect bulk information on a large telecommunications company;

- **Patient:** an APT is very patient by nature, which also separates them from lesser threats, which will simply move on to newer targets if their attack fails. As the publicly known APT examples in section 2.3 also point out, patience is very characteristic of these threats, infiltrating its target for months, or even years to carry out its objectives. For example, Operation Aurora was ongoing for over at least 4 years and Operation Shady RAT for at least 5 years. Because of their lengthy approach to reach their goal, these threats are also expected to remain inactive for extended periods of time in between.

An APT takes a considerable amount of time to carry out its objectives. This is also observed from the publicly known APT examples in section 2.3. Infiltration may remain unnoticed for years. The good news is that because of this typical characteristic, in theory, there is plenty of time to detect such long-term attacks. It can also be observed from these characteristics that, because they are so patient and stealthy, their network traffic is typically kept to a bare minimum. These new threats no longer strike fast and leave the next day with or without success, generating a lot of obvious malicious network traffic in their wake. Rather, they spend a significant amount of time planning and infiltrating to carry out their objectives as quietly as possible. Chapter 3 will discuss the effectiveness of current detection methods, as well as new research developments.

## 2.5 Overview of the APT

To gain a full understanding of APTs, it is necessary to gather more background information about what the attackers are usually affiliated with, what their likely targets are, what attacks they typically execute and what they are eventually after. The Joint Universities Computer Centre Information Security Task Force (JUCC ISTF) published [26] an article that researched this information, which is depicted in table 2.3.

| | **Organised crime** | **State-affiliated** | **Activists** |
|---|---|---|---|
| **Victim industry** | Finance<br>Retail<br>Food | Manufacturing<br>Professional<br>Transportation | Information<br>Public<br>Other services |
| **Region of operation** | Eastern Europe<br>North America | East Asia (China) | Western Europe<br>North America |
| **Common actions** | Tampering (physical)<br>Brute force (hacking)<br>Spyware (malware)<br>Captured stored data (malware)<br>Adminware (malware)<br>RAM scraper (malware) | Backdoor (malware)<br>Phishing (social)<br>Command/control (C2) (malware)<br>Export data (malware)<br>Password dumper (malware)<br>Downloader (malware)<br>Stolen credentials (hacking) | SQLi (hacking)<br>Stolen credentials (hacking)<br>Brute force (hacking)<br>RFI (hacking)<br>Backdoor (malware) |
| **Target assets** | ATM<br>POS controller<br>POS terminal<br>Database<br>Desktop | Laptop/desktop<br>File server<br>Mail server<br>Directory server | Web application<br>Database<br>Mail server |
| **Desired data** | Payment cards<br>Credentials<br>Back account info | Credentials<br>Internal organisation data<br>Trade secrets<br>System info | Personal info<br>Credentials<br>Internal organisation data |

Table 2.3: Taxonomy of the APT [26].

Notice that the JUCC ISTF distinguishes three main groups of attackers, namely those involved in organised crime, state-affiliated, or involved in activism. However, some of these attackers may not be after classified information at all and are instead determined to sabotage their targets, such as what Stuxnet was designed for. Such attacks seem to be missing from this overview.

They further assert that the following parties are more at risk for being infiltrated by APTs:

- Pharmaceutics;
- Research institutions;
- Financial institutions;
- Government entities.

They remark that especially research institutions, such as universities, are at risk as these institutions are easier to infiltrate than corporate environments. Research institutions are predominantly used as a stepping stone to gain more information on their eventual targets as a means to obfuscate the attack's origin [26].

# Chapter 3

# The current state of affairs regarding detection

Now that it is known how an APT operates, what its characteristics are and in which context it operates, this chapter will describe the current state of knowledge regarding detection methods. As already mentioned in section 1.3, this research mainly focuses on detecting an APT on the network itself (i.e. the network plane). This research also focuses on which shortcomings current detection mechanisms have (see section 3.3) and in which directions current research aims to formulate an adequate response to APTs (see section 3.4).

## 3.1 Current defensive countermeasures

Numerous countermeasures exist to help organisations detect suspicious activity on the network. These are usually performed by an Intrusion Detection System (IDS). An IDS can be split up into mainly two flavours:

- **Network Intrusion Detection System (NIDS)**: these IDSs carefully monitor inbound and outbound traffic on the network segment(s) it listens on. It does so by listening on its Network Interface Cards (NICs), typically between network segments where the network flows through the NIDS. There are common detection signatures a NIDS can use to scan for, but it may also apply behavioural analysis of network traffic. Examples of NIDSs include Snort [60], Suricata [63] and Bro [7];
- **Host Intrusion Detection System (HIDS)**: these IDSs, in contrast to NIDSs, carefully monitor system files, event logs, and running processes, specifically on the host which runs the HIDS. When suspicious activity is detected, the detected events are logged and the administrator is notified. Examples of HIDSs include OSSEC [65], AIDE [76] and Samhain [78];

There are mainly two types of NIDSs:

- **Signature Based IDS (SBS)**: relies on specified patterns in network packets it should detect. These NIDSs have a database of numerous detection signatures it could generate alarms for. Examples include Snort [60] and Suricata [63];
- **Anomaly Based IDS (ABS)**: relies on a predetermined base line of how the network operates normally and generates alarms when the network behaviour significantly (depending on configuration) deviates from the established base line. Examples include Bro [7] and Snort (with a third party module[52]).

One advantage of ABSs is that they can detect new types of attacks, as long as the anomaly deviates from the threshold of the established base line. This is particularly useful in the case of detecting APTs. However, ABS NIDSs do need to be extensively trained to be efficient in their detection. Training the ABS NIDS helps eliminating false positives and improving detection accuracy [21].

This paper mainly focuses on network detection, hence is why this chapter will mainly focus on NIDSs, rather than HIDSs. These IDSs can give much insight into network activity, as well as activity on a device itself. These alerts, together with other alerts from other security systems, such as firewalls, anti-virus suites, system logs etc., can further be aggregated and correlated in a central place. This is where a Security Information and Event Management (SIEM) contributes to security:

- **Security Information and Event Management (SIEM)**: a SIEM can perform analysis on these many received security events and is therefore a more advanced approach to threat detection. Examples include OSSIM [2] and LOGalyze[1] [39].

A SIEM adds more detection intelligence by analysing aggregated security telemetry in real-time. There are also several honeypot systems that aim to perform a form of intrusion detection. Honeypots mostly contain data that appears to be legitimate, but is closely monitored for activity. This seemingly valuable data is set up as bait for potential intruders and the honeypot then logs all of the activity performed by the intruder when it takes the bait.

## 3.2 Offensive techniques and corresponding defensive countermeasures

This section shows how each offensive technique performed by an adversary can be combated with corresponding defensive countermeasures. Table 3.1 is based on a similar table that Chen et al. proposed [9], yet it is now modified to fit the Giura et al. model (see figure 2.2). This table shows where each attack takes place during each phase of the Giura et al. model, which this paper has also previously used in conjunction with their Attack Pyramid in section 2.3.4 to map APT examples inside this model.

| **Phase** | **Offensive techniques** | **Defensive countermeasures** |
|---|---|---|
| Reconnaissance | OSINT, social engineering, preparing malware | Security awareness training, patch management, firewalls |
| Delivery | Spear phishing, watering hole attack | Content filtering software, NIDS, Anti-virus software |
| Exploitation | Zero-day exploits, remote code execution | Patch management, HIDS, advanced malware detection |
| Operation | Exploiting legitimate services, privilege escalation, RAT, encryption | NIDS, SIEM, event anomaly detection |
| Data collection | Collecting data, encryption of data at rest | Access control, HIDS, NIDS, event anomaly detection, encryption of data at rest and in transit |
| Exfiltration | Compression, encryption of data in transit, counter forensics | Data loss prevention |

Table 3.1: Offensive techniques and corresponding defensive countermeasures per phase of attack.

Note that it should not be inferred from this table that deploying these countermeasures is sufficient enough to detect and thwart APTs. For example, security awareness training is only as effective as its weakest link as it takes only a single employee to be lured into a false sense of security, creating a window of opportunity for an attacker [54]. These countermeasures are extra layers of defense, but the next sections will discuss some of these detection system shortcomings, and current research developments regarding detection.

## 3.3 Standard network detection shortcomings

This section first describes some attack techniques on the network plane which classical threats often employ. However, an APT characterises itself as using new vulnerabilities, which are difficult to scan for

[1] LOGalyze has not been updated since June, 2013.

because these vulnerabilities are typically not known yet, demonstrating a difference in sophistication of these different threats.

### 3.3.1 Obvious suspicious activities

Although actively scanning the network is easiest, because one instantly knows all the hosts on a specific network segment at a certain point in time (unless devices blocked specific network traffic, which makes it a little harder). However, an actual APT would most likely not resort to classic active network scanning, as this is (far) too noisy, thereby risking obvious detection. The same applies to port scanning a target to find running services, which could prove to be exploitable.

Research suggests [53][48] that it is also possible to detect a device with a promiscuously listening NIC. Tools to detect a NIC set in promiscuous mode are readily available [20][40][22]. However, with the coming of network switches, it is unlikely an APT would consider this a feasible attack as there no longer is a single collision domain to be sniffed. Detecting a promiscuous NIC also involves active detection tools and is therefore not a feasible solution for a strictly passive APT sensor as per SIDN's request. Secondly, setting a NIC into promiscuous mode means that received network traffic is no longer filtered by the NIC itself, meaning that it is now left to the kernel whether to react on detection methods [53]. Looking at the publicly known APT examples and their level of sophisticated attacks, it is not a far-fetched idea that an attacker may change a system's kernel behaviour to avoid responding to such detection methods at all, especially when the system is under its full control.

For detection of these obvious suspicious network activities, a standard NIDS would suffice. However, it should be noted that deploying a NIDS is by no means sufficient enough as it is known that these systems can be circumvented.

### 3.3.2 NIDS evasion

One such evasion technique [58] is to alter the time to live (TTL) value of the packets. Bogus packets are sent in between the actual malicious packets, but, when configured with a smaller TTL value, these bogus packets will pass through the NIDS, but time out just before they would arrive at the target. Another technique [61] caused a denial of service attack by exploiting the signature matching algorithm of a NIDS. The researchers found out that by sending carefully crafted packets, with a lot of string repetition, would force the matching engine to repeatedly backtrack during inspection. The latter example has since been fixed [57], but NIDS evasion techniques are known to happen and should be looked out for.

Furthermore, the SBS NIDSs are mainly designed to scan for static network signatures. However, this is counterproductive for detecting APT, which are known for exploiting new vulnerabilities which have not been converted to detection signatures such a system should scan for. It would not be a far-fetched idea that an APT would still do some form of network scanning, using techniques which these network scanners simply do not (yet) scan for, because they lack the proper signatures.

### 3.3.3 The reliable baseline problem

Another problem arises when implementing ABS NIDSs, which requires a reliable baseline to detection deviations. Due to the persistent nature of an APT, it is hard to first establish such a trustworthy baseline when it can not even be ruled out yet that the network has not been compromised by an APT yet. More modern SIEM solutions further expand on this by aggregating all this information of various generated security events onto a central platform for real-time analysis and alerting. However, these solutions do not yet fully perform large-scale analysis of verbose security telemetry, which are to be thoroughly analysed, scanned and correlated for suspicious behaviour and anomalies.

Due to the advanced nature of an APT, their attacks have likely never been seen before, which makes it difficult to scan for threats. It is because of this observation that new research is aimed at finding new ways to detect these new threats [75]. Big data analysis of security telemetry may provide a missing piece of the puzzle to improve detection effectiveness and will be discussed in the next section.

## 3.4 Performing big data analysis of security related events

Numerous research proposes new big data models, which combines all sorts of security related event data (security telemetry). They assert that determining whether an event is to be classified as a threat does not have to happen in real-time [1], nor can they even be guaranteed or expected to happen in real-time [35][30]. Rather, it makes more sense to store these events and perform thorough and accurate analysis of security telemetry. This telemetry is gathered from various standard detection sources, such as network captures, (web application) firewalls, NIDSs, HIDSs, and stored in a big data appliance for data mining and further analysis [1]. The effectiveness lies in that this approach combines many detection sources, aims to reduce the amount of false positives and scans for suspicious behaviour by looking for anomalies that deviate from normal network behaviour.

### 3.4.1 Obstacles to overcome

However, s few obstacles need to be overcome first. Big data analysis takes a lot of computational power and storage to process all this bulk information [30]. There is also a need for better, or even new algorithms to process, correlate and analyse these vast amounts of data in a timely fashion. Also to visually represent them in a meaningful way and to ensure the security of sensitive indicators of compromise (IOCs) [75]. It should also be noted that advances in big data analysis has given us powerful tools to extract and correlate vast amounts of data, making it more prone to privacy violations. Therefore, research must continue developing big data applications with an understanding of privacy principles and recommendations [49].

Research claims [1] that thus far no such off-the-shelf solutions exist. However, there has since been one open source solution developed by Cisco, named OpenSOC [47], which provides a security analytics framework for big data analysis of aggregated security telemetry. OpenSOC combines several existing technologies, such as Apache Hadoop for processing of big data sets, and Elasticsearch for indexing of aggregated telemetry. Their framework allows for monitoring various telemetry sources, anomaly detection and telemetry correlation [12]. While OpenSOC seems to be an interesting approach, which also seems in line with current research development concepts, it is still a young project. Further research of such new security big data analysis systems should first be subjected to sufficient research to measure its detection effectiveness.

Apart from OpenSOC, there are also quite some SIEM solutions that perform analysis on security related events [50]. However, such SIEM solutions perform real-time analysis on security related events detected by standard network detection appliances and many of these system's concepts are to be further developed to perform big data analysis. A typical SIEM solution does not perform big data analysis, nor were they designed to that [49]. It should therefore be noted that, although SIEMs provides a more intelligent means of detection, they still do not provide adequate detection against APTs.

# Chapter 4

# Discussion

## Much ado

There has been much ado around APTs in the news as of these last few years. Many commercial IT security companies publish articles that get much attention from the media, generating a lot of buzz as a result [59]. Some regard the term 'APT' as the latest buzz word in IT security [44] that only spreads fear, uncertainty and doubt (FUD) [4]. Some acknowledge this buzzword, but also acknowledge its usefulness to distinguish these threats from conventional threats [55]. APTs in the media may generate a fair amount of buzz, but it is not to say that APTs do not pose a threat as a result.

## Perspective

Thus far only a few cases of high-profile targets have been under attack by these threats, which implies that Small and Medium-sized Enterprises (SMEs) are typically not interesting enough for APTs, as section 2.5 points out. These attackers choose their targets specifically for financial enrichment, espionage, or even sabotage of (foreign) critical infrastructures. Although the majority of organisations are not at risk, they and society in general, suffer nonetheless damage caused by successful APT attacks.

However, the notion of an *Advanced* persistent threat implies that these threats are built on earlier types of threats. While it is indeed true that these new threats engage in highly skilled attacks, it should not be inferred from the name that these new threats are a mere continuation of an earlier, less sophisticated type of threat. An APT is of an entirely different magnitude and the notion of these attacks being *advanced* could set people on a wrong track.

## Evolutionary leap

The evolution of digital threats is currently causing a paradigm shift in defense strategies. To illustrate this paradigm shift in detection, it is quite similar to military arms races where evolutionary leaps can also be observed. Up to and including the First World War, trench warfare was an effective defense strategy as infantry was well protected against enemy small arms fire. However, with developments in military aircraft during the Second World War, trenches were no longer effective. Aircraft could now simply fly over battlefields, dropping bombs, or dropping enemy troops behind trench lines. Such advancements in military strategics demanded fundamentally new defense measures, such as armour piercing ballistics, anti-aircraft artillery, fighter aircraft, and missile defense systems.

A similar evolutionary leap is happening on the digital landscape. Due to the distinctive stealthy and patient nature of these new digital threats, current network detection measures are becoming less effective. Merely focusing on real-time analysis of known, static signature matching, which some large organisations still heavily rely on, is not effective anymore in the long run. New technological advancements of these threats have already demonstrated their detection evasiveness. Moreover, the threats no longer use well known or predictive attacks which can be picked up by traditional IDSs. Rather, as can be observed from known operations in section 2.3, and their analysed characteristic in section 2.4, the attackers masquerade their attacks so well that current detection measures are unable to pick up these new attacks.

Current detection methods are aimed at real-time analysis, yet due to the patient and stealthy nature of these new threats, the attackers infiltrate an organisation for a significant amount of time. These threats can be expected to move slow, or even lie dormant for extended periods of time as well, all in an effort to avoid arousing any suspicion. As such, because of these typical characteristics, the requirement for real-time analysis no longer applies in the long run. Instead, current research is turning to more thorough analysis to detect threats. The attacks are so advanced and new that it is counterproductive to keep detecting for static, known, and predictive malicious network attacks. It is because of these characteristically stealthy attacks that current defense strategies should not focus on improving currently existing detection methods. Instead, organisations should shift their attention to new ways regarding detection that are fundamentally different from how it is done currently. This is also why the initial research approach, with its defined research questions in section 1.2 were, in retrospect, overly focused on a portable, plug-and-play solution. Threat detection semantics are no longer to be exclusively placed at the sensors. Instead, the sensors should rather collect an overabundance of information, and sending it to a central point that can perform more in-depth big data analysis of all aggregated telemetry.

## Current detection methods

Currently, there seems to be a mismatch between detection solutions the industry currently has at its disposal, and new detection methods and models that current research developments propose. The industry uses detection systems which aim to detect classical threats based on a priori knowledge, namely signature matching. This contests these system's detection effectiveness as APTs are notoriously known for exploiting zero-day vulnerabilities, which by definition are previously unknown. Also, after having interviewed both a large Dutch telecommunications provider and a large Dutch financial institution, it appears that anomaly detection is hardly ever deployed. When dealing with dynamic and therefore unpredictable network traffic, this creates a problem when trying to setup a reliable baseline. Another problem that arises is that it cannot be ruled out that an APT is not already present, creating a catch-22, an inescapable paradoxical situation due to contradictory conditions. Because to detect the potential compromise of the network, the aforementioned baseline is needed in the first place. As section 3.4 shows, the basic consensus within current research developments is that there is an urgent need for performing more intelligent, big data analysis of, and reliable anomaly detection on all security telemetry.

## Current threat models

Last but not least, three observations can be made regarding the models that have been described and used throughout this report. First, the Attack Pyramid in section 2.3.4, previously proposed by Giura et al., has now been used for the first time to depict the evolution of several publicly known APTs. However, secondly, it can be observed that not all APTs fit within this model. Some threats "merely" aim to sabotage high-profile targets, which this model does not allow for. After the Operation phase, the Attack Pyramid follows a predetermined path of *Data collection* and *Exfiltration*, something not all APTs do. This was found during section 2.3.4 when trying to display Stuxnet in the Attack Pyramid, which, unfortunately, this model does not allow for due to the model's inherent design. Note this is not so much a flaw in the design of the Attack Pyramid itself. Rather, and this brings us to the third observation, it seems that the root cause of the problem lies within current research, which this model is build on. Current research seems very focused at a specific type of APT that steals information. The models such as the Kill Chain and the Giura et al. model in section 2.1, as well as the Attack Pyramid in 2.3 and the taxonomy matrix in section 2.4, further support this observation. These models appear to be overly focused at threats that aim to exfiltrate information, leaving no room for threats that aim to sabotage. As a result, these models are not universally applicable. However, this also present itself as an opportunity for future work, to perform more studies on different types of APTs, which should not be overlooked.

# Chapter 5

# Conclusion

## Modus operandi

This research gives insight into the modus operandi by analysing current models that outline the procedure of an APT. This shows that current research reached a certain consensus regarding the modus operandi of APTs. Subsequently, using the Giura et al. Attack Pyramid in section 2.3, several publicly known APT examples have been analysed. Analysis confirms such similarities in their method of operation, albeit that these models tend to be very focused on a specific type of APT that aim to exfiltrate information, leaving no room for APTs that instead aim to sabotage.

## Characteristics

Analysis of the modus operandi enables to further analyse the typical characteristics, which distinguishes an APT from other, classical threats. The characteristics, described in section 2.4, mostly consist of an APT being inquisitive, stealthy, well prepared, infiltrative, resourceful and patient. These characteristics, in conjunction with the modus operandi, concludes that an APT is of an entirely different magnitude than the usual, classical threats. This is characterised by their behavioural properties, such as that the attacks are stealthy to such an extent that their actions even appear to be endogenous, normal behaviour. These threats are also very persistent, which is demonstrated by the fact that some of these threats infiltrate their targets for years to carry out their objectives in secrecy. This leads to the conclusion that these threats are no mere continuation of classical threats. Rather, an evolutionary leap can be observed on the digital landscape, changing the rules of the game and demanding new ways of detection to keep up with this arms race.

## The current state of art regarding detection

To gain insight into new detection developments, it is necessary to first analyse the detection methods the industry currently has at its disposal, such as NIDSs, HIDSs and SIEMs. This leads to the conclusion that these current network detection systems, although very effective to detect the threats they are designed for, are currently not sufficiently attuned to the characteristics of APTs. Therefore, the current detection systems are not deemed effective to reliably detect an APT. This creates a serious need for new and better ways of detection, which in turn needs to be of a different magnitude as well, in order to be better attuned to the characteristics of these new threats. Science will have to make an evolutionary leap in response and is diligently searching for new means of detection. A common thread within current research developments is that they shift their focus towards big data analysis of mass aggregated security telemetry. The current state of art shows that such systems have yet to be designed and properly field tested for their effectiveness in terms of detecting these new threats. However, research already indicates that such systems need to store and analyse an extreme amount of data, which in turn takes a considerable amount of computational power and storage capacity. Furthermore, there is also a lack of optimal detection algorithms to accurately process these vast amounts of data in a timely fashion.

However, the industry has already made some progress with the coming of SIEM appliances. Although these systems are not yet developed enough to perform big data analysis, these systems are at least a step forward in the right direction.

## Designing a portable solution

After answering all three research sub questions, it allows for answering the main research question, which leads to the conclusion that designing an effective portable passive APT sensor is not feasible. As per the requirements of SIDN, the sensor needs to be easily deployable, small and portable, such as a single board computer. However, due to the persistent nature of the APT, it would not seem logical to design a portable solution for a persistent problem. Furthermore, research shows that detection mechanisms need much more computational power to cope with analysis of continuous mass aggregation of all security telemetry. This hints at dedicated, powerful systems and large storage clusters - certainly not something trivial to be deployed in a portable and plug-and-play manner. Such a portable solution, which lacks computational power and storage, could only be equipped with standard, low-end sensor detection software, which sophisticated threats would most likely slip through. To conclude, due to the fact that the devised main research question has an adverse outcome, this report leads to more insight that a new era has dawned, one that requires new defense strategies against these new looming threats.

# Chapter 6

# Recommendations and future work

As the industry currently lacks sufficient, proven tools to effectively detect APTs, this research recommends SIDN to implement a SIEM solution to further improve their security. This is a second best scenario, yet it is the only feasible approach while no proven tools exist to perform much needed big data analysis of security telemetry. When deploying a SIEM solution, a separate management network is highly recommended to isolate all security telemetry flows to the SIEM from the actual LAN. Standard network detection systems can still be used, however, their alerts are now to be aggregated at the SIEM. It is also recommended to perform more anomaly detection on network segments. However, reliable baseline establishment may prove to be a challenge, due to reasons earlier described in the discussion. This is suboptimal, but it further supports the need for better detection methods which the industry does not have at its disposal yet.

## Opportunities for future work

This research also leaves some opportunities for future work, which SIDN is recommended to look into:

- The first opportunity is to frequently redo this research, to keep up with latest developments regarding the modus operandi of APTs, as well as their characteristics and detection methods and models.

- The second opportunity is to research the feasibility and effectiveness to implement such a big data security analysis system. Solutions such as Cisco's OpenSOC or other solutions that claim to perform this big data analysis are currently new developments that require more research before they are to be implemented for what they claim to detect or prevent. This may pose some intricacies.

  For example, in order to detect threats that infiltrate their targets for years, analysing such a system's detection effectiveness could therefore also take years. Moreover, if such a system during such time fails do detect APTs, it cannot simply be concluded that therefore there are no APTs. In other words, providing negative proof of evidence is inherently harder than providing positive proof of evidence. Of course, this may be different with APTs that aim to sabotage a target, as disruption of service is something the target organisation could simply find out due to its devastating results.

- The third opportunity is to perform more research on different types of APTs, which thus far seems very focused at a specific type of APT that aims to exfiltrate data. A (behavioural) study on threats that sabotage high-profile targets could point out that these threats may behave differently. They may therefore require slightly different detection methods than threats that are focused at stealing classified information.

# Bibliography

[1] Sung-Hwan Ahn, Nam-Uk Kim, and Tai-Myoung Chung. Big data analysis system concept for detecting unknown attacks. In *16th International Conference on Advanced Communication Technology*, pages 269–272. IEEE, 2014.

[2] AlienVault. OSSIM: The Open Source SIEM — AlienVault. `https://www.alienvault.com/products/ossim`, 2016. [Online; accessed 25-July-2016].

[3] Dmitri Alperovitch et al. *Revealed: operation shady RAT*, volume 3. McAfee, 2011.

[4] Frank Andrus. Is APT the new FUD? `http://www.scmagazine.com/is-apt-the-new-fud/article/205457/`, 2011. [Online; accessed 20-August-2016].

[5] Beth Binde, Russ McRee, and Terrence J O'Connor. Assessing outbound traffic to uncover advanced persistent threat. *SANS Institute. Whitepaper*, 2011.

[6] Guillaume Bonfante, Jean-Yves Marion, and Fabrice Sabatier. Gorille sniffs code similarities, the case study of qwerty versus regin. In *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)*, pages 82–89. IEEE, 2015.

[7] Bro. The Bro Network Security Monitor. `https://www.bro.org/`, 2016. [Online; accessed 25-July-2016].

[8] Michael A Champion, Prashanth Rajivan, Nancy J Cooke, and Shree Jariwala. Team-based cyber defense analysis. In *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, pages 218–221. IEEE, 2012.

[9] Ping Chen, Lieven Desmet, and Christophe Huygens. A study on advanced persistent threats. In *IFIP International Conference on Communications and Multimedia Security*, pages 63–72. Springer, 2014.

[10] Thomas M Chen. Stuxnet, the real start of cyber warfare?[editor's note]. *IEEE Network*, 24(6):2–3, 2010.

[11] Mark A Cobos. Nodes and codes: The reality of cyber warfare. Technical report, DTIC Document, 2012.

[12] I Collier and R Wartel. Maintaining traceability in an evolving distributed computing environment. In *Journal of Physics: Conference Series*, volume 664, page 022011. IOP Publishing, 2015.

[13] Jorge L Contreras, Laura DeNardis, and Melanie Teplinsky. Mapping today's cybersecurity landscape. *Am. UL Rev.*, 62:1113, 2012.

[14] MITRE Corporation. CVE-2010-2568. `https://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2568`, 2010. [Online; accessed 20-August-2016].

[15] MITRE Corporation. CVE - CVE-2010-0249. `https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0249`, 2016. [Online; accessed 12-August-2016].

[16] Terry Cutler. The Anatomy of an Advanced Persistent Threat. `http://www.securityweek.com/anatomy-advanced-persistent-threat`, 2010. [Online; accessed 5-July-2016].

[17] cve search. CVE-2008-4250. http://cve.circl.lu/cve/CVE-2008-4250, 2008. [Online; accessed 20-August-2016].

[18] cve search. CVE-2010-2549. http://cve.circl.lu/cve/CVE-2010-2549, 2010. [Online; accessed 20-August-2016].

[19] cve search. CVE-2010-2729. http://cve.circl.lu/cve/CVE-2010-2729, 2010. [Online; accessed 20-August-2016].

[20] Ademar de Souza Reis Jr. and Milton Soares Filho. sniffdet - Remote Sniffer Detection Tool/Library. http://sniffdet.sourceforge.net/, 2016. [Online; accessed 04-August-2016].

[21] Roberto Di Pietro and Luigi V Mancini. *Intrusion detection systems*, volume 38. Springer Science & Business Media, 2008.

[22] Embyte and Snifth. nast - Network Analyzer Sniffer Tool. https://manpages.debian.org/cgi-bin/man.cgi?query=nast&apropos=0&sektion=0&manpath=Debian+8+jessie&format=html&locale=en, 2016. [Online; accessed 04-August-2016].

[23] Patrick Engebretson. *The basics of hacking and penetration testing: ethical hacking and penetration testing made easy*, chapter 1, pages 14–18. Elsevier, 2013.

[24] Nicolas Falliere, Liam O Murchu, and Eric Chien. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 5:6, 2011.

[25] James P Farwell and Rafal Rohozinski. Stuxnet and the future of cyber war. *Survival*, 53(1):23–40, 2011.

[26] Joint Universities Computer Centre Information Security Task Force. Advanced Persistent Threat (APT). wikisites.cityu.edu.hk/sites/netcomp/articles/Pages/AdvancedPersistentThreat.aspx, 2010. [Online; accessed 20-August-2016].

[27] Ryan Gallagher. The Inside Story of How British Spies Hacked Belgium's Largest Telco. https://theintercept.com/2014/12/13/belgacom-hack-gchq-inside-story/, 2016. [Online; accessed 01-August-2016].

[28] Paul Giura and Wei Wang. A context-based detection framework for advanced persistent threats. In *Cyber Security (CyberSecurity), 2012 International Conference on*, pages 69–74. IEEE, 2012.

[29] Michael Joseph Gross. Exclusive: Operation shady rat—unprecedented cyber-espionage campaign and intellectual-property bonanza. *Vanity Fair*, 2, 2011.

[30] Ki-Hyoung Han, Hyung-Jong Jeong, Doog-Sik Lee, Myung-Hui Chae, Cheol-Hee Yoon, and Kyoo-Sung Noh. A study on implementation model for security log analysis system using big data platform. *Journal of Digital Convergence*, 12(8):351–359, 2014.

[31] Harold Joseph Highland. The brain virus: fact and fantasy. *Computers & Security*, 7(4):367–370, 1988.

[32] Ehinome J Ikhalia. A new social media security model (smsm). *International Journal of Emerging Technology and Advanced Engineering Website: www. ijetae. com (ISSN 2250-2459, ISO 9001: 2008 Certified Journal, Volume 3, Issue 7*, 2013.

[33] The Intercept. Quantum Insert Diagrams. https://theintercept.com/document/2014/03/12/quantum-insert-diagrams/, 2014. [Online; accessed 14-August-2016].

[34] The Intercept. The Inside Story of How British Spies Hacked Belgium's Largest Telco. https://theintercept.com/2014/12/13/belgacom-hack-gchq-inside-story/, 2014. [Online; accessed 14-August-2016].

[35] Kyung-Sik Jeon, Se-Jeong Park, Sam-Hyun Chun, and Jong-Bae Kim. A study on the big data log analysis for security. *International Journal of Security and Its Applications*, 10(1):13–20, 2016.

[36] Stamatis Karnouskos. Stuxnet worm impact on industrial cyber-physical system security. In *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society*, pages 4490–4494. IEEE, 2011.

[37] Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011.

[38] MinJae Lee and JinKyu Lee. The impact of information security failure on customer behaviors: A study on a large-scale hacking incident on the internet. *Information Systems Frontiers*, 14(2):375–393, 2012.

[39] LOGalyze. LOGalyze - Open Source Log Management Tool, SIEM, Log Analyzer. `http://logalyze.com/`, 2016. [Online; accessed 17-August-2016].

[40] Marek Majkowski. sniffer-detect NSE Script. `http://nmap.org/nsedoc/scripts/sniffer-detect.html`, 2016. [Online; accessed 04-August-2016].

[41] Morgan Marquis-Boire, Claudio Guarnieri, and Ryan Gallagher. Secret Malware in European Union Attack Linked to U.S. and British Intelligence. `https://theintercept.com/2014/11/24/secret-regin-malware-belgacom-nsa-gchq/`, 2014. [Online; accessed 14-August-2016].

[42] Angela Moscaritolo. McAfee fires back at Shady RAT criticism. `http://www.scmagazine.com/mcafee-fires-back-at-shady-rat-criticism/article/210116/`, 2011. [Online; accessed 13-August-2016].

[43] NCSC. Factsheet stuxnet - an advanced targeted attack. *Survival*, 2011.

[44] Infosec Nirvana. APT – What you need to know? `http://infosecnirvana.com/apt-what-you-need-to-know/`, 2012. [Online; accessed 20-August-2016].

[45] Ahmed Obied. Microsoft Internet Explorer 6 - Aurora Exploit. `https://www.exploit-db.com/exploits/11167/`, 2016. [Online; accessed 13-August-2016].

[46] Brian Prince. Digging Deeper into Operation Shady RAT. `http://www.securityweek.com/digging-deeper-operation-shady-rat`, 2011. [Online; accessed 13-August-2016].

[47] The OpenSOC Project. Open Security Operations Center. `https://opensoc.github.io/`, 2016. [Online; accessed 17-August-2016].

[48] Mohammed Abdul Qadeer, Arshad Iqbal, Mohammad Zahid, and Misbahur Rahman Siddiqui. Network traffic analysis and intrusion detection using packet sniffer. In *Communication Software and Networks, 2010. ICCSN'10. Second International Conference on*, pages 313–317. IEEE, 2010.

[49] M Chithik Raja and Munir Ahmed Rabbani. Big data analytics security issues in data driven information system, 2014.

[50] Mosaic Security Research. Log Management & Security Information and Event Management (SIEM) Software Guide. `http://mosaicsecurity.com/categories/85-log-management-security-information-and-event-management`, 2016. [Online; accessed 17-August-2016].

[51] Symantec Security Response. Regin: Top-tier espionage tool enables stealthy surveillance. `http://www.symantec.com/connect/blogs/regin-top-tier-espionage-tool-enables-stealthy-surveillance`, 2012. [Online; accessed 14-August-2016].

[52] Łukasz Saganowski, Marcin Goncerzewicz, and Tomasz Andrysiak. Anomaly detection preprocessor for snort ids system. In *Image Processing and Communications Challenges 4*, pages 225–232. Springer, 2013.

[53] Daiji Sanai. Detection of Promiscuous Nodes Using ARP Packets. `http://www.securityweek.com/anatomy-advanced-persistent-threat`, 2001. [Online; accessed 7-July-2016].

[54] M Angela Sasse and Ivan Flechais. Usable security: Why do we need it? how do we get it? 2005.

[55] Bruce Schneier. Advanced Persistent Threat (APT). https://www.schneier.com/blog/archives/2011/11/advanced_persis.html, 2011. [Online; accessed 20-August-2016].

[56] Dell SecureWorks. Advanced Threat Protection with Dell SecureWorks Security Services. http://www.secureworks.com/assets/pdf-store/white-papers/wp-advanced-threat-protection.pdf, 2016. [Online; accessed 27-June-2016].

[57] Debian Security. CVE-2006-6931. https://security-tracker.debian.org/tracker/CVE-2006-6931, 2006. [Online; accessed 16-August-2016].

[58] Umesh Shankar and Vern Paxson. Active mapping: Resisting nids evasion without altering traffic. In *Security and Privacy, 2003. Proceedings. 2003 Symposium on*, pages 44–61. IEEE, 2003.

[59] Surendra Singh. Advanced Persistent Threats: From "FUD" To Facts. http://www.cxotoday.com/story/advanced-persistent-threats-from-fud-to-facts/, 2014. [Online; accessed 20-August-2016].

[60] Snort. Snort - Network Intrusion Detection and Prevention System. https://www.snort.org/, 2016. [Online; accessed 25-July-2016].

[61] RandySmith CristianEstan SomeshJha. Backtracking algorithmic complexity attacks against a nids. 2006.

[62] Joe Stewart. Operation aurora: Clues in the code. *Secureworks-The information security experts. Retrieved*, pages 01–23, 2010.

[63] Suricata. Suricata — Open Source IDS / IPS / NSM engine. https://suricata-ids.org/, 2016. [Online; accessed 25-July-2016].

[64] Colin Tankard. Advanced persistent threats and how to monitor and deter them. *Network security*, 2011(8):16–19, 2011.

[65] OSSEC Project Team. Open Source HIDS SECurity. https://ossec.github.io/, 2016. [Online; accessed 25-July-2016].

[66] Microsoft Technet. Microsoft Security Bulletin MS08-067 - Critical. https://technet.microsoft.com/en-us/library/security/ms08-067.aspx, 2008. [Online; accessed 20-August-2016].

[67] Microsoft Technet. Microsoft Security Bulletin ms10-002 - Critical. https://technet.microsoft.com/en-us/library/security/ms10-002.aspx, 2010. [Online; accessed 20-August-2016].

[68] Microsoft Technet. Microsoft Security Bulletin MS10-046 - Critical. https://technet.microsoft.com/en-us/library/security/ms10-046.aspx, 2010. [Online; accessed 20-August-2016].

[69] Microsoft Technet. Microsoft Security Bulletin MS10-061 - Critical. https://technet.microsoft.com/en-us/library/security/ms10-061.aspx, 2010. [Online; accessed 20-August-2016].

[70] Microsoft Technet. Microsoft Security Bulletin MS10-073 - Important. https://technet.microsoft.com/en-us/library/security/ms10-073.aspx, 2010. [Online; accessed 20-August-2016].

[71] Microsoft Technet. Microsoft Security Bulletin MS10-092 - Important. https://technet.microsoft.com/en-us/library/security/ms10-092.aspx, 2010. [Online; accessed 20-August-2016].

[72] Ian Traynor. Gchq: Eu surveillance hearing is told of huge cyber-attack on belgian firm. *The Guardian, October*, 2013.

[73] Rohit Varma. McAfee Labs: Combating Aurora. https://kc.mcafee.com/resources/sites/MCAFEE/content/live/CORP_KNOWLEDGEBASE/67000/KB67957/en_US/Combating%20Threats%20-%20Operation%20Aurora.pdf, 2010. [Online; accessed 20-August-2016].

[74] Nikos Virvilis and Dimitris Gritzalis. The big four-what we did wrong in advanced persistent threat detection? In *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*, pages 248–254. IEEE, 2013.

[75] Nikos Virvilis, Oscar Serrano, and Luc Dandurand. Big Data Analytics for Sophisticated Attack Detection. http://www.isaca.org/Journal/archives/2014/Volume-3/Pages/Big-Data-Analytics-for-Sophisticated-Attack-Detection.aspx, 2010. [Online; accessed 20-August-2016].

[76] Hannes von Haugwitz. Advanced Intrusion Detection Environment. http://aide.sourceforge.net/, 2016. [Online; accessed 25-July-2016].

[77] J Vukalović and D Delija. Advanced persistent threats-detection and defense. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on*, pages 1324–1330. IEEE, 2015.

[78] Rainer Wichmann. The SAMHAIN file integrity / host-based intrusion detection system. http://www.la-samhna.de/samhain/, 2016. [Online; accessed 25-July-2016].

[79] Kim Zetter. Google hack attack was ultra sophisticated, new details show. *Wired Magazine*, 14, 2010.

[80] Zeljka Zorz. CVE-2010-2549. https://www.helpnetsecurity.com/2010/02/10/operation-aurora-malware-investigated/, 2010. [Online; accessed 20-August-2016].

# Glossary

**ABS** Anomaly Based IDS. 16, 18

**air gap** A network security measure to ensure that systems are isolated from the (untrusted) network, such as the Internet.. 10, 11, 13

**APT** Advanced Persistent Threat. 1, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24

**BICS** Belgacom International Carrier Services. 12

**GCHQ** Government Communications Headquarters. 12

**HIDS** Host Intrusion Detection System. 16, 17, 18, 22

**IDS** Intrusion Detection System. 16, 20

**modus operandi** A method of operating or functioning; A subject's manner of working. 3, 4, 7, 9, 22, 24

**NIDS** Network Intrusion Detection System. 3, 16, 17, 18, 22

**NIPS** Network Intrusion Protection System. 3

**OSINT** Open Source Intelligence. 6, 13, 17

**phishing** An attempt to acquire personal information, such as credentials or financial details, by masquerading as a trustworthy entity in an electronic communication. 7

**RAT** Remote Access Toolkit. 6, 17

**SBS** Signature Based IDS. 16, 18

**script kiddie** A term to refer to unskilled individuals that uses scripts or applications developed by others to attack computer systems. 13

**SIDN** Foundation for Internet Domain Registration Netherlands. 1, 3, 4, 18, 23, 24

**SIEM** Security Information and Event Management. 3, 16, 17, 19, 22, 24

**spear phishing** Phishing attempts directed at specific individuals or companies have been termed spear phishing. 7, 10, 11, 13, 17

**Supervisory Control and Data Acquisition** system for remote monitoring and control that operates with coded signals over communication channels. 10

**watering hole attack** A technique to study a target's specific browsing habits, which the attacker then injects with malware. 12, 17

**ZEH** Zero Entry Hacking. 7