Delft University of Technology

MSc Thesis

Towards a roadmap for development of intelligent data analysis based cyber attack detection systems

Date	July 5, 2012
Author	J.A. de Vries
Student number	1116967
E-mail	J.A.deVries@student.tudelft.nl
Faculty	Technology Policy & Management
Section	Information and Communication Technology
Program	System engineering, Policy Analysis and Management (SEPAM)
Graduation company	Fox-IT
Graduation supervisors:	Prof. dr. Y.-H. Tan (TU Delft, TBM) Dr. ir. J. van den Berg (TU Delft, TBM) Dr. M.E. Warnier (TU Delft, TBM) Ing. H. Hoogstraaten (Fox-IT)

Keywords: Cyber espionage, intrusion detection, machine learning, intelligent data analysis

Abstract

Cyber-attacks against companies and governments are seeing an increase in complexity and persistence. These more complex attacks are aimed at penetrating corporate and government networks to obtain classified information. The difference with cyber attacks from a couple of years ago is that attackers take more time and effort to remain undetected. Common intrusion detection methods lack in their ability to detect such complex attacks. They do not correlate individual suspicious events to detect these advanced attacks. A new approach to detection is therefore needed which takes the multistep characteristic of these advanced persistent threats into account. A framework is proposed to relate attack characteristics to analysis methods and business criteria. Traditional signature based detection algorithms still remain useful but the advanced approach of the attackers requires the use of intelligent data analysis. The framework is used as a roadmap to design a system capable of detection advanced persistent threats. A test case is used to evaluate the resulting system design. The test case shows that not all activities of attackers will be detected and that human analysis of warnings produced by the system remains necessary. The application of the framework does provide a good approach for attack analysis and detection design aspects. Intelligent data analysis methods are shown to improve detection but remain too inaccurate to replace signature detection.

Contents

Appendixes

1. Introduction

1.1. General background on cyber attacks

In our society computer networks are used to store proprietary information and to provide services for organizations and society. The security of information and services are important issues in which the confidentiality, the integrity and the availability of information are core principles. These principles are often abriviated to the CIA triangle. Laws have been created to enforce these principles. An example of such a law is the Dutch “Wet Bescherming Persoonsgegevens” which sets rules for the protection of personal information in information systems. A breach of one of these principles can be considered to be a security incident. An intentional breach of these security principals by an outsider is often considered a criminal offence by law. For example; in the Netherlands this law is called “Wet Computercriminaliteit”. Intentional breaches of security by electronic means are often called cybercrimes.

Cybercrime, in the broad sense of the word, is around since the advent of computer systems. Cybercrime entails more than viruses and malware. Espionage, disruption of services, banking fraud and other activities through electronic means are also considered cybercrimes. These crimes have clear economic consequences when services are disrupted or when bank accounts are emptied. At the present-day the economic impact of cybercrime is large but hard to quantify according to research by Van Eeten et al. [1]. Their research showed that numbers are often of a high level of aggregation but even so they show that the costs of individual cases can run into millions of euros. Others estimate estimates that the costs run from 100 billion to 1 trillion US dollars annually [2]. The financial consequences are often high when a cyber-attack is targeted specifically to disrupt services or to steal proprietary information. The consequences can go as far as bankruptcy of the target.

Research shows that cybercrimes are lucrative for criminals since they can operate internationally and not all countries have laws against cybercrimes [2]. Cyber criminals can also easily move their operations to other countries to avoid capture and prosecution. The result is that even when they are identified by law enforcement they still run a low risk of being convicted. Working in the cybercrime industry is even more attractive in countries with high unemployment rates, like eastern European countries. In Russia for example are criminals paying IT graduates ten times more than normal jobs would [2]. Economic motives are not the only motivators for cybercrimes. Fun and political motives are other drivers for hackers to attack companies and governments [2]. Examples of political motivated hacks are for example information breaches providing information to Wikileaks and activities of the hackers collectives who are targeting governments and companies whom are acting against what they belief to be right.

The growth of the internet and the increased use of computers in society are followed by an increase in the number of attacks. As a result of the rising number of attacks security was tightened. The result is a rat race between cyber criminals and security expert leading to increasingly complex attacks and defenses. The most complex and advanced attacks are targeted attacks which are specifically aimed at companies or governments to reach a predetermined goal. The security company Symantec identifies targeted attacks as one of five recurring themes of cyber security threats in 2010. The other four are often part of a targeted attack [3]. Targeted attacks that received a lot of media attention where the Aurora Trojan, the Stuxnet worm in 2010 and the Diginotar hack in 2011. The Aurora trojan was aimed at obtaining intellectual property of large corporations [3]. The Stuxnet worm was aimed at disruption of industrial systems in Iran [4]. The Diginotar hack was aimed at generating rogue signed certificates with the goal of spying on traffic to websites. The best known example of such a generated rogue certificate was for *.google.com [5].

Targeted attacks with political motives are often easy to detect since they aim to disrupt services to inflict visible damage which gives them publicity. This is often achieved by simple Denial of Service attacks. Targeted attacks with an economic purpose or espionage, like stealing trade or state secrets are executed with more stealth and advanced means. Such attacks are also called Advanced Persistent Threats (APT) in literature and by companies [6]. These APTs differ from other attacks in the sense that the attackers take their time and will

go to great lengths to gain access to their targets. The attackers themselves can be individuals or groups but also governments and companies. The term cyber warfare is often used if one government is targeting another. Stuxnet is considered to be an example of cyber warfare. Defending against targeted attacks is harder and requires a new approach to detection to counter the stealthy nature of targeted attacks like APTs. This has motivated Fox-IT to start development in the Cyber Attack Detector (CAD), which is aimed at detection of such targeted attacks [7].

A crucial element in detection of intrusions, malware, viruses and other cyber threats is intelligent data analysis (IDA). There are a lot of different IDA methods proposed for intrusion detection in academic studies in intrusion detection. Tavallaee et al. identify three main categories: Classification based methods, clustering based methods and statistics based methods [8]. Within these categories several different analysis methods like decision trees, neural networks, support vector machines, Bayesian networks and nearest neighbor algorithms are used. Another aspect related to data analysis is the choice of input data. The choice of input data and preprocessing of data for analysis have a large influence on the success of IDA methods [9].
The application of IDA methods for intrusion detection is hampered by a high number of errors in analysis, in other words the rate of false positives and false negatives [10] [11]. The rat race between attackers and security experts which results in continues changes in attack methods make it even harder to implement IDA methods.

1.2. Goal and research questions

1.2.1. Goal of the research

The goal of the CAD sensor is to detect APTs as well as the more common attacks. An analysis of the general structure of APTs in light of detection is therefore necessary while keeping the continues changes of attacks in mind. The CAD is developed as a commercial product and is therefore also influenced by business aspects. Examples of such business aspects are scalability, effectiveness and costs. The choice of IDA methods for attack detection depends on attack characteristics, system design choices and business aspects.

Existing classifications or taxonomies of intrusion detection are mostly focused on the design of intrusion detection systems or approaches against specific attack types. Tucker et al. are categorizing systems on data source and actions of IDS's [12]. Axelsson has created a taxonomy of detection systems based on detection principles [13]. Singh & Silakari have made a classification of detection systems based on purpose [14]. Research into business aspects is mostly focused on the impact of attacks and the costs of detection and not on the design of detection [15].
Research into attacks is focused on attack structure and goals. Igure & Williams have done an extensive literature research into taxonomies of attacks [16]. They show different purposes of such taxonomies including some which are aimed at providing support for signature based intrusion detection.
However none of these researches link attacks and their features to intrusion detection system design and data analysis methods with the purpose of supporting the choice of analysis methods. In this thesis an approach to IDS design and analysis method choice is proposed based on attack characteristics and business aspects.

1.2.2. Research/design questions

The goal and research gap defined in the previous paragraph raises a number of questions which need to be answered for the goal to be fulfilled. The overall research goal is:

An approach to an analysis based roadmap for detection of advanced persistent threads with intelligent data analysis.

This question cannot be answered without addressing underlying questions first. One should first know what the nature is of APTs before a design can be made to detect them. An overview of methods currently applied in research is required to find appropriate methods for detection of APTs. The requirements and limitations on an APT detection system given by business aspects is also required before a design using IDA methods can

be made. Knowing the possibilities and requirements makes it possible to make a design approach. This combines the previous elements and gives insight into the applicability of IDA methods in APT detection.

These steps can be summarized into the following sub research questions:

0) Main research question. (See above)
1) What is the structure of Advanced Persistent Threats?
2) Which intelligent data analysis methods can be used for detection of Advanced Persistent Threats?
3) Which business related aspects are influencing the design of the CAD sensor?
4) Which design choices lead to a detection system which can detect APTs?
5) To what extend can the system design detect APTs?

1.2.3. Research methods used

The goal of this research is a design which utilizes intelligent data analysis to detect advanced persistent threats. The first step, identifying the structure of APTs is necessary to identify key points and elements for detection. A short literature review is used to identify the main structure and the most important elements of APTs. An overview of the current use of intelligent data analysis methods for intrusion detection is created by analyzing systematic reviews of the research field of intrusion detection. These reviews are used to create an overview of the popularity of methods and the structure of intrusion detection systems. The relationship between business goals of APT detection (development as well as deployment) come from literature and a discussion with an expert. These steps form the input for a rough conceptual design which is used to show where intelligent data analysis methods can and should be applied.

1.3. Thesis outline

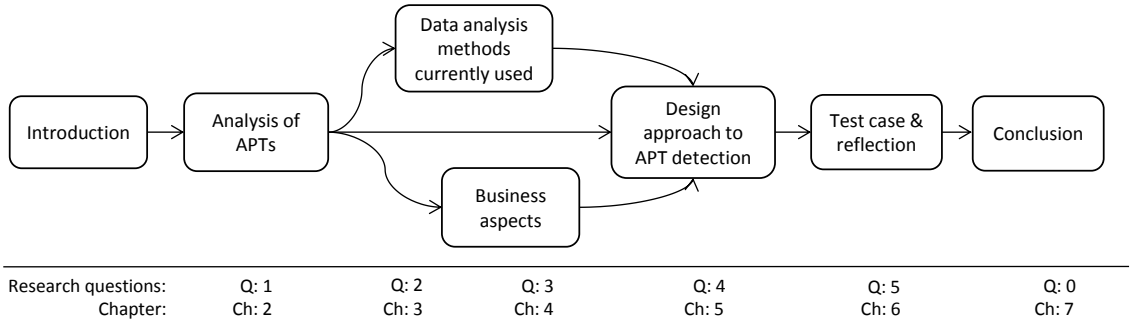

Figure 1; Visual representation of the thesis outline

Figure 1 gives an overview of this thesis report following the structure of the research questions related to the chapters. In chapter two an analysis of APTs is performed using a new framework. The purpose of this framework is to show the interrelatedness of all elements involved in the design of an APT detection system. In the third chapter intelligent data analysis methods currently in use for intrusion detection are discussed. Chapter four is dedicated to the business aspects related to an APT detection system. The second, third and fourth chapters form the input for a conceptual design of an APT detection system utilizing intelligent data analysis in chapter five. The sixth chapter presents a test case of the system and reflection on the results. The final chapter will answer the main research question as a conclusion and give future research topics.

2. Introduction to Advanced Persistent Threats

Attackers are constantly changing their attack approaches, the general structure of such attacks however is important to understand when designing a detection system. Knowledge of attacks aids the choice of attack features for detection. General attack methods also remain in use by attackers over a longer period of time even if their workings change. This chapter gives a concise introduction into advanced persistent threats and proposes a framework which is used to analyze attacks and support the design of intrusion detection systems. In addition, an example attack scenario is presented.

2.1. Advanced persistent threats

The term “Advanced Persistent Threat” is loosely used for a wide variety of cyber threats. In essence it always implies a threat in which the attacker is determined and has a specific goal [17]. This section introduces and describes APTs with the goal to give insight into the nature of APTs.

2.1.1. Introduction

The term cyber attack encompasses a lot of different attack types. The question is what the difference is between APTs and older attack approaches. A meta-literature study on taxonomies by Igure & Williams shows that there are a number of attack taxonomies in literature [16]. They show that these taxonomies differ in their goal. Some try to give a general overview of attacks categories and methods while others are aimed at aiding the development of intrusion detection systems. The latter classify attacks by attack signatures which can be used in intrusion detection. The attack signatures are not limited to attacks which use only one type of action to achieve their goal. Attacks existing out of a series of single action attack events over time are also included in these taxonomies. The single action attacks can be considered to be low level attacks when they are viewed from a series perspective. These series of events are high level attack events containing lower level events. Especially such high level event sequences give useful information for analysis by security analysts and detection systems [16].

Traditional defenses are aimed at the low-level attacks and less at the higher level event sequences. For example: A virus scanner only looks for viruses. Identifying high level attacks is a research topic with the goal to identify high level attacks by correlating low-level attacks. Correlating low level attacks to high level attacks would reduce the number of warnings for security analysts [18]. This idea is extended by Cheung et al. to events gathered from multiple sensors and other security applications [19]. The aim of their research is to create a means to describe and identify scenarios of multistep cyber-attacks. These steps each use a different attack method to achieve a sub goal of the multistep attack. Cheung et al. show that their approach of attack analysis works for denial of service attacks as well as for attacks aimed at gaining access to a network to extract information. Yang et al. uses data fusion on alerts from intrusion detection systems to identify multistep attacks. They present an attack guidance template with seven attack stages. The first stage contains recon attacks from an external network, the final stage is reaching an attack goal on the internal network [20].

The targeted attacks called APTs introduced in chapter one are a form of such multistep attacks. APTs also contain different steps using different attack methods. The difference is that APTs have a predetermined goal, use more advanced low level attack approaches and they try harder to remain undetected. Tankard [6] identifies several generic steps in ATPs. The first step is identification of the target and reconnaissance after which social engineering and malware are used to gain access. After this the attacker starts to explore the target network and will extend its access until the attacker has reached his goal. The former Dutch GOVernment Cyber Emergency Response Team (GOVCERT.NL), now the Dutch national cyber security centre, does not explicitly define individual steps. They do recognize a sequence of events with different attack methods and a specific order of execution from first contact to extraction of data [21]. The Dutch research institute TNO identified eight steps within a cyber-attack; reconnaissance, gaining access, internal reconnaissance, gathering information, information extraction, control of malware and erasing tracks [22]. These steps each have their specific goal within the attack and can be identified by the methods used within a step.

2.1.2. Attack steps in APTs

APT can be seen as a form of multistep attacks because they contain different steps each with a specific goal. Identification of the different steps based on their goal is helpful in identifying corresponding low level attacks. These low level events can be used for detection, while the structure of APTs helps with correlating low level events. The advanced nature of APTs and the attempts to obscure APTs results in constant changes to the specific low level methods used. The goals of the general steps on the other hand do not constantly change. The research by Tankard [6] and GOVCert show a similar order of events. The eight steps defined by TNO are more detailed and are more in line with attack related literature. These eight steps will therefore be used as general steps of APTs.

The eight steps defined by TNO each have a different purpose in an APT. Some of the steps are overlapping and steps can be very similar to earlier steps having only a distinctly different location. A general description of activities in the eight steps is given below. In this description some of the steps are combined because they often occur simultaneously. They do however have different purposes and are therefore still separate steps.

Step 1; reconnaissance**:** Reconnaissance of the target gives the attacker information about the technical environment and about the people working at the target. The attackers are looking for possible means of entry like services accessible from the internet, but they are also looking for information about employees to use in social engineering attack methods [21].

Step 2; gaining access**:** The information used in step one is used to obtain a foothold in the ICT infrastructure of the target. One method is to use weaknesses in software to install malware which is used in the following steps of the attack [6].

Step 3&4; internal reconnaissance and expanding access**:** When a foothold is obtained the attack is continued by looking for information about the internal infrastructure. Server addresses, accounts with higher privileges and software used on workstations and servers. This information is used to expand access in the network. Internal reconnaissance and expansion of access are therefore steps with distinct activities, which are overlapping in time.

Step 5; gathering information**:** The fifth step starts once the attacker has an overview of the network structure and locations of the wanted proprietary information. In this step the attacker starts gathering proprietary information to a single location where it is stored until extraction. Information can for example be found on file servers, databases or other information sources like enterprise resource systems planning or customer relation management systems.

Step 6; information extraction: The information gathered in the previous step can be extracted from the network to locations on the internet. These locations might be, legally or illegally, under control of the attackers, but public services, like file storage sites, are also used to offload information.

Step 7&8; control and erasing tracks: Botnet clients outside the target network are sometimes used to obscure the attacks by using them as proxies. Dedicated proxy servers are also used to obscure the attack source. Malware might also be controlled through a botnet infrastructure from which they are part. APTs are however not always using botnets, more often they are performed using dedicated malware. Removal of the malware and other stored information from a network are standard ways of erasing traces of the attack.

It is important for the attacker to limit the possibility of detection and to stay in the target network as long as needed for the purpose of the attack. Using clients in a botnet to distribute attacks is one way of obscuring elements of an attack. Altering logs or file dates are other means inside a network to hide activity on servers and clients. Erasing tracks to hide the attack is an activity during all stages of the attack and is performed often through the botnet control.

2.2. Example attack scenario

A fictitious example based on the technical descriptions of ATPs by Tankard [6] and Symantec [3] and the social aspects described by Kshetri [2] is given in this paragraph as illustration.

A company is interested in a product of a competitor. They would like to know detailed technical and production information as well as financial information about the production costs of the product. They pay some unemployed IT graduates to gain access to the competitor's network in order to obtain the wanted information. The competitor must remain unaware of the network breach to avoid an investigation and possible lawsuits or closure of foreign markets to the company.

Step 1; reconnaissance: The first step of the attackers is reconnaissance of the target company. They start by browsing corporate websites for names and mail addresses, check DNS registrations to find public accessible services and check the open web for social media profiles of people claiming to work at the target company. The main goal is to find handles for social engineering approaches and to find version information on servers and website content management systems, to find exploitable vulnerabilities.

Step 2; gaining access: After the first step the attackers proceed to use the profile information of employees to construct phishing emails which look legitimate. These emails contain a link to an infected website which uses a zero day exploit to install a malware component on the victims computer.

Step 3&4; internal reconnaissance and expanding access: Once the attackers have gained a foothold in the network through the malware they will try to expand their access to other parts of the network. The malware starts to monitor connections to servers in the network, gather information about installed programs and network users to identify server addresses, network structure and possibilities for expanding access. Un-patched programs or operating systems create possibilities for further expansion of the attackers access to network clients and servers. The attackers also perform active reconnaissance on the network themselves by connections through the malware clients.

Step 5&6; gathering and extracting information: After a while the attackers are successful and have found the wanted technical documentation and have access to the financial systems of the target. They slowly gather all the information on one of the clients they control and prepare the information for extraction. Finally they extract the information to a legitimate file storage application on the internet to make the extraction look as normal as possible. They also continue snooping around for other data they can sell and extract this as well.

Step 7&8; control and erasing tracks: The attackers have continuously monitored progress through direct access through a backdoor created by the malware and by updates from the malware to servers on the internet. After extraction of the last of the wanted information the attackers start to hide their tracks by uninstalling the malware. Botnet clients are used as proxies to hide the origins of traffic.

2.2.1. Possibilities for detection of APTs

Detection of APTs is harder because of the effort of the attacker to remain undetected but not impossible because there is traffic generated and malware is active on workstations and or servers. The use of unknown exploits makes it harder for common detection methods, like virus scanners, to detect an APT. On the other hand are APTs still another form of attack scenario and they consist out of several different lower level attack elements. The result is that it is still possible to find traces of attacks which can be put together to see if there is an ongoing APT present. Common firewalls, HIDS and NIDS systems have a hard time finding an APT because they look mostly to known attack methods and do not take the structure of APTs in account. They do not connect different low level events to each other to form an attack scenario. An approach that does link low-level attack elements can possibly detect such attacks [18] [19].

Network traffic can be used to detect the different steps of APTs. The eight steps each have a different traffic pattern in a network. An example of these patterns is given in an example network in Figure 2. This example network is a simplified network containing only three network segments. The first segment is the public network or the Internet. The second segment contains public services like web servers. These can be reached

from the public network and internal network, but do not have access to other internal network segments. This segment is often called a DMZ. The workstation segment represents segments in networks containing workstations and other network clients that do not provide services. In practice there are often multiple segments of this type. The third segment houses servers providing services to the internal network. In practice networks are more complex than this example. They contain many physical and logical segments. But the example holds in practice because attackers will not have immediate access to internal services or workstations from the public network or from servers in the public services segment.

Each attack step goes deeper into the network. The streams in Figure 2 show these advances through the network. These traffic streams are mixed with the normal traffic streams making them harder to detect. The attack related changes in traffic content or the volume and destinations of traffic however can be detected and used for detection. The traffic streams associated with normal network use are shown in Figure 12 in Appendix 1. These normal traffic streams are also subjected to change. Looking for changes in behavior to detect attacks is made more complex because of these normal changes. Traffic analysis on the other hand does have the advantage of being unobtrusive in the network.

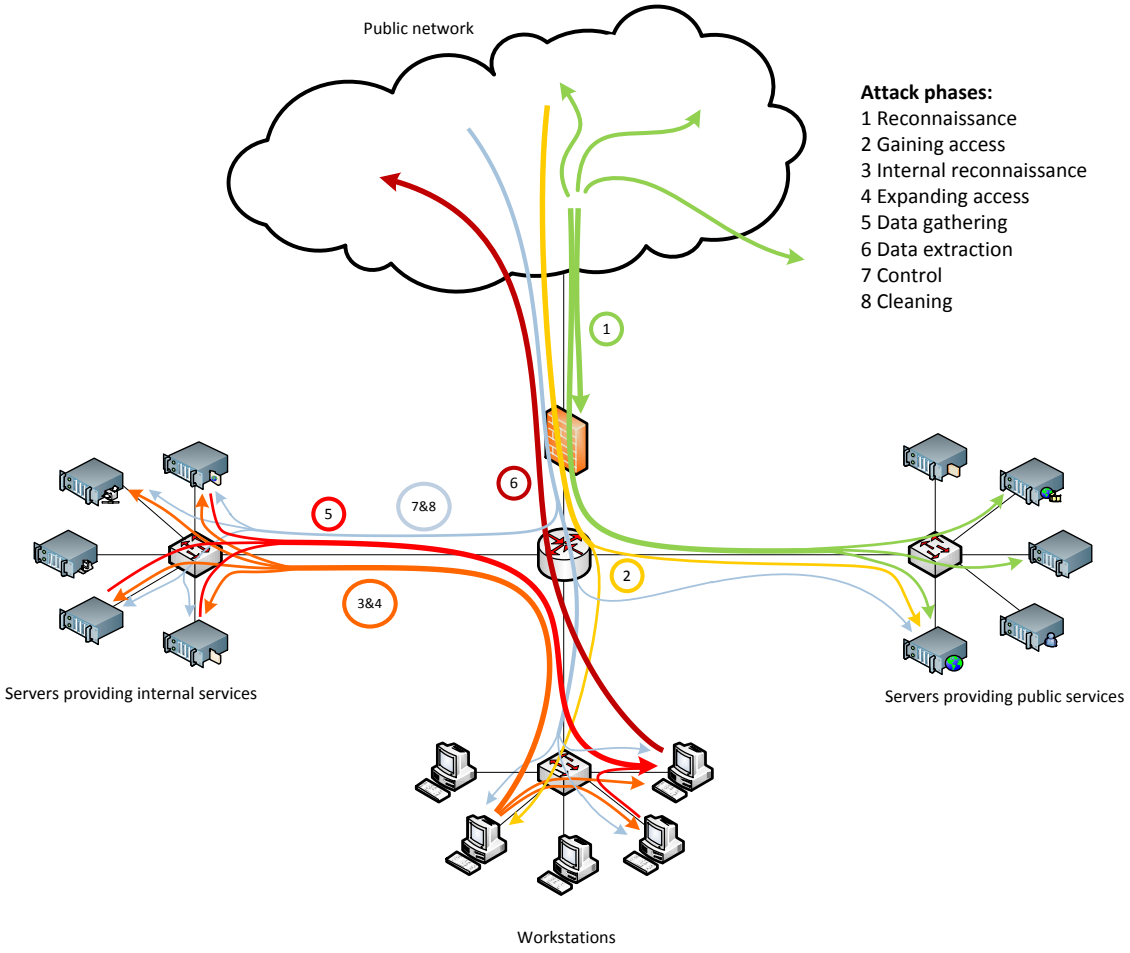

Figure 2; Network with general traffic streams per step for an example attack

Knowledge of the structure of high level attack sequences and low level attack elements is crucial for detection. This knowledge is necessary for the selection of attack features which can be detected, for example in network traffic. The choice of attack features has consequences for the detection system design and choices of analysis methods and their success in detecting attacks [9]. Approaching the choice of analysis methods from an attack perspective utilizes this knowledge to improve detection.

2.3. A framework for attack steps and aspects of APTs

In the previous paragraphs of this chapter APTs where introduced and an example of an APT was given. APTs are attacks with multiple steps which have a different purpose in an attack and contain lower level attack methods. These lower level attack methods can provide aspects which can be detected. Knowledge of these aspects can help in choosing where and how to detect them. Knowledge of attack features also helps in choosing analysis methods for attack detection.

A framework is proposed to link the high level attack structure of multistep attacks and low level attack methods to the design of detection systems. The proposed framework is constructed as a matrix in which the rows are used to represent the different attack steps and the columns represent the different aspects related to attacks and detection. The proposed framework is shown in Figure 3. The rows in the framework correspond to the attack steps. The first three columns describe the goal, method and detectable features of attacks. The information in these columns is similar to the first dimension in the attack taxonomy proposed by Hansman & Hunt [23]. The difference is that Hansman & Hunt do not consider multistep attacks. The first dimension of Hansman & Hunt could therefore be applied to the second and third column of each row with only the relevant attacks. The detection locations, detection methods and analysis are derived from ids related attack taxonomies which link attacks to intrusion detection systems [16]. The business aspects are derived from the relationships between intrusion detection systems and investments and the relation between the impact of events and their financial consequences. Individual research on these elements exists but does not link all these aspects in one framework to multistep attacks. The first three columns describe what must be detected. The fourth column describes where attack features can be detected. This could, for example be in network traffic through a firewall or in log-files on servers. The fifth and sixth columns describe how the attacks can be detected. And finally the seventh row gives the impact motivated reasons for detection choices. The result is a complete overview of all APT detection related information.

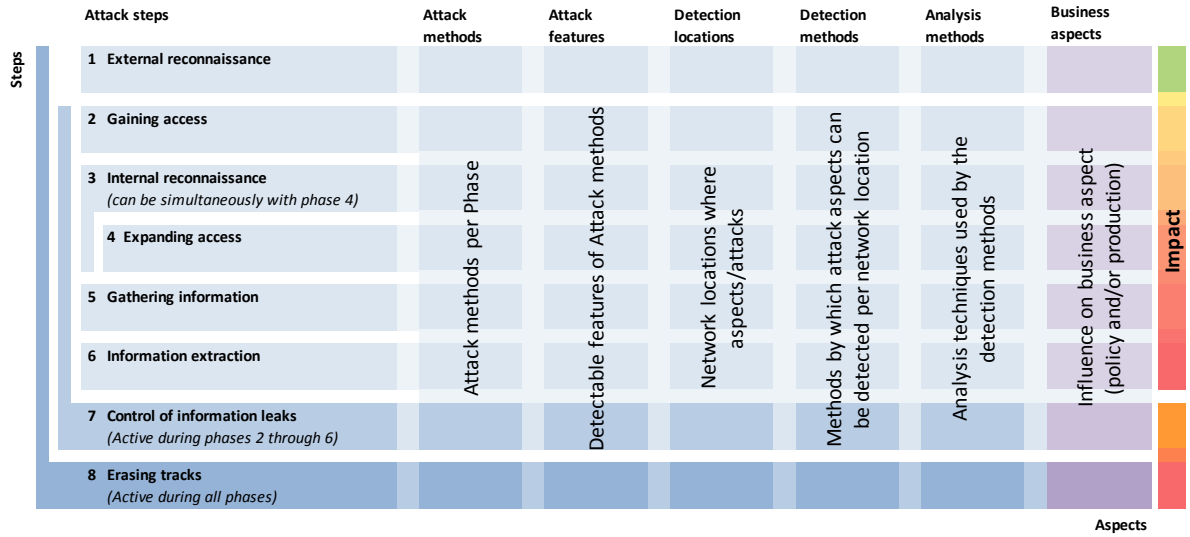

Figure 3; Overview of the framework

The first column contains the different identifiable steps of an APT which can be identified, it also shows possible overlap amongst the steps containing activities which are distinct in nature but are executed at the same time. An example of this are steps three and four. The first column contains the distinguishable steps of an APT. The presented framework has eight steps, but this might differ per specific attack or due to future changes in attacks. The second column lists the attack methods which are known to be used within the steps. The third column goes into more detail, needed for analysis techniques, on the attack methods from the first column. The locations where these attacks and aspects can be detected are in the fourth column. The next column links the previous columns to common detection methods. The sixth column should contain descriptions of analysis methods used in the detection methods in the previous columns. These methods should be capable of analyzing the aspects mentioned in the second column. The last two columns contain aspects which are not part of an attack but do influence choices on defenses against them. The first of these

two are the business aspects like the willingness to invest in defenses and the incentive to create new means of detection. The business aspects are influenced by the possible impact of an APT or the occurrence of one of its steps.

2.3.1. Use of the framework

The framework can be used in different ways. One way is to use the framework for analysis of attack scenarios ranging from detailed to general analysis. Figure 4 shows 2 steps of the example attack in the model. This is still at a high level. Attack method could be made more specific by malware type and detectable features could go into more details about which mail contents are important identifiers. In appendix 2 the complete framework of the example attack from which Figure 4 is a fragment is given. This appendix also contains a framework with a general description of the contents of the different blocks in the framework. Using the framework this way gives the possibility to describe APTs in a more general way.

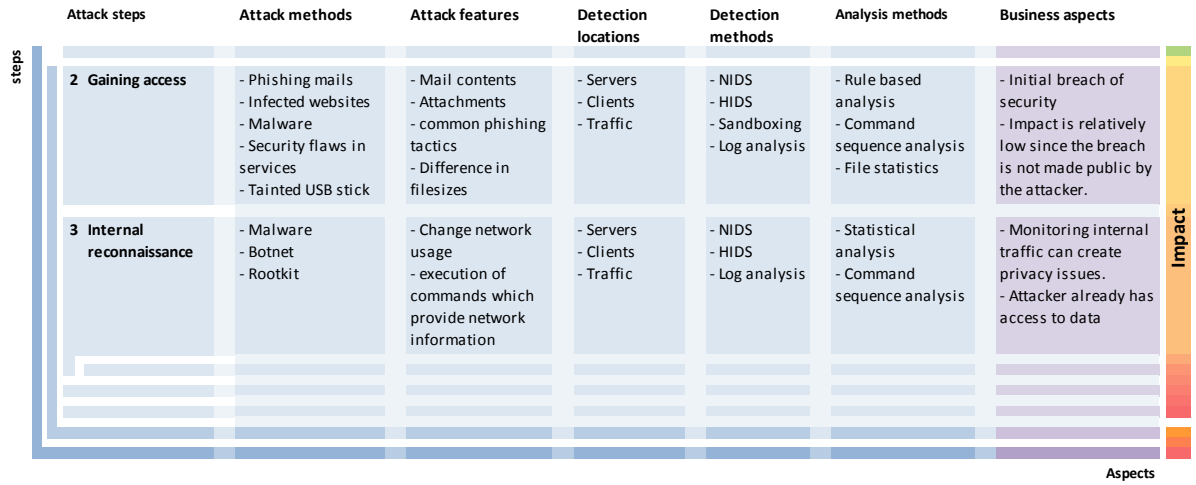

Figure 4; Steps two and three of the example attack in the framework

The second way of using the framework is to use it as a roadmap for development of a detection system. The attack analysis is the input for design choices of the system and ultimately the choice of analysis methods. The first column gives the attack steps and with that the structure of the APT. The second column with attack methods indicates real methods like denial of service attacks or malware related to botnets. Aspects like specific types of attacked ports or control messages are general aspects of these attacks that can be used for detection. These are the features which can be used for detection. The location column forms a link between the attack aspects and the detection methods. The choice of location limits the available attack features for detection. The detection method column contains detection technology like virus-scanners, firewalls, honeypots and traffic capturing equipment, currently in use for detection. The analysis techniques are used in the detection methods to find the attack features. The business aspects are different from the other aspects since they are not determined by technical relations between the columns. Chapter 4 discusses the relationship of the business aspects with the other aspects in detail, but the idea of the business column is that these aspects do control the choices made for deployment or development of detection methods. Chapter 5 shows the use of the framework as a roadmap for an attack detection system design.

2.4. Conclusion

APTs are a form of complex attack scenarios in which different low-level attack methods are used in a multistep approach to achieve a predetermined goal. They are executed with more stealth than normal attacks. This is often achieved by specifically targeting employees or by using zero day exploits.

This stealthy approach makes it harder to detect APTs with standard defenses like firewalls and virus scanners. An approach for detection of APTs should be aimed at detecting different elements of attack scenarios. The framework presented in this chapter can help analyze APTs with the purpose of creating a means of detection of APTs.

3. Intelligent data analysis for intrusion detection

The challenge in detecting attacks lies in choosing the attack right attack data features. Based on this choice follow choices for locations in networks for data gathering, (pre)processing and intelligent analysis of the data to identify attacks. For APTs this is even more difficult since attacks are stealthier and the different steps of an APT need to be correlated to identify an attack. The framework presented in the previous chapter shows the elements of attacks and their relations between the different aspects and steps. The detection methods and data analysis aspects are found in the fifth and sixth column of the framework. This chapter introduces intelligent data analysis methods used in intrusion detection and the structure of intrusion detection systems found in literature. The chapter is concluded with a discussion on the challenges of applying intelligent data analysis in APT detection systems.

3.1. Feature selection and data preprocessing

Feature selection and data preprocessing need to be done before data can be used for analysis. Davis & Clark name three reasons for preprocessing relevant for intrusion detection: Training dataset creation, feature selection and data reduction. Data preprocessing is estimated to take up to 50% of the time used in data analysis [9]. Feature selection is the first and most important step because the selection of features ultimate determines if an attack can be detected. The choice of analysis methods determines the input data type and therefore the necessary preprocessing. Analysis methods like neural networks require numeric input data to calculate the appropriate class [24]. For example; Preprocessing is required if text based features need to be analyzed by a neural network.

The selection of feature types begins with the choice of input data for analysis algorithms. For network traffic this can be streaming data, also called flow data, or packet data. Common feature types for analysis in intrusion detection are packet headers, protocol features or stream contents [9]. Another possibility is the construction of new data features based on the previous mentioned features. New features can be useful in analysis and they can also reduce the amount of features needed for analysis.

The first choice when capturing network traffic is a choice between individual IP-protocol packets or data on traffic streams. The individual packets are the actual traffic between computers and servers. They contain information on the source, destination, protocol and other transmission related information. But they also contain the actual data which was transmitted. Data on streams contains information on the connection like source, destination, protocol, etc. but usually not the transmitted data. Streams are a sequence of IP-protocol packets belonging to one connection. The advantage of using streams for analysis is that it reduces the amount of data that needs to be stored. They are also easier to use for behavior analysis.

Useful data features for anomaly detection are according to Dua & Du; “A feature extractor derives basic features that are useful in event analysis engines, including a sequence of system calls, start time, duration of the stream, source and destination address, port numbers, protocol, number of bytes and number of packets.” [24](p.4). This selection might change over time so flexibility is needed in the system design.

Creating new data features like statistics on sources, destinations and protocols can help in detection of attacks. Creating such statistics allows for easy interpretation and application of algorithms because they are, per definition, always numeric. Such statistics can be created live by using counters or later by using stored data. Expert knowledge of attack approaches is required to make the right choices for counters and/or ratio’s to detect attacks. This approach works only for attacks which result in anomalies in traffic patterns. It should therefore be combined with one or more methods that look at packet content instead of derived meta-data. Timestamps are not included in the internet protocol suite and are therefore not part of the packet data. Packets should therefore be time stamped when they are captured. This is necessary for the creation of frequency based data in the preprocessing step.

3.2. Introduction into intelligent data analysis in intrusion detection

There are, in general terms, three different approaches to intrusion detection [24]. The first, and most common one, is detection through known signatures. This approach is also known as misuse or rule based detection. In this approach data analysis techniques compare data against a set of signatures of attack features. The signatures used by such methods are human made. The advantage of this approach is that it is relatively simple to implement. Intrusions can be detected once attack features of an attack method for example a virus or malware are known. The problem however is that attackers know this and continuously change their attacks so that signatures continuously need updating.

The second approach is anomaly detection. This approach compares behavior or features and looks for deviations from what is considered to be normal. The advantage of these methods lies in the fact that they do not require constant updating of manmade signatures. This makes them also better at detecting novel attack methods. These methods are often called machine learning algorithms, since they require the computer to be trained to know what is normal. Training can be done supervised and un-supervised. Supervised training is done with datasets in which data is labeled as normal or anomalous. The disadvantage of this approach is that it is time consuming as it requires manual labeling of existing data. Unsupervised learning does not require a labeled training dataset. It operates from the assumption that normal behavior makes up the majority of the data and that anomalies differ from normal behavior. The disadvantage from unsupervised learning is that novel or rarely occurring normal traffic is also considered anomalous and therefore bad [24]. Another disadvantage of unsupervised training methods is that they generate a large amount of false positives and false negatives. These detection errors can happen through changes in protocols, but also due to normal behavior which was not present during the training period. Human interaction is often required to evaluate these possible threats to determine if detected patterns are intrusions or normal.

The third approach combines the first two methods because they complement each other [24]. The signature detection approach is used because it has a low false positive rate. Anomaly detection is added to be able to detect unknown attacks. The high false positive rate of some anomaly detection methods can be reduced by creating signatures of deviations which are considered to be normal. This combined approach does however increase the complexity of the analysis because it needs some form of reasoning to combine the results from both methods. The approach used to combine signature and anomaly detection should balance low false positive rate with a high detection rate [24]. Some false positives must be accepted to prevent false negatives which are more harmful. But the false positives should not overwhelm system operators.

3.3. Single methods for anomaly detection

Single method systems use only one algorithm for analysis and are often used for specific detection research questions or detection goals. These methods are often aimed at groups of similar low-level attack methods. Systems utilizing multiple algorithms are discussed in the next section. These systems often use multiple single method approaches to detect different attack methods.

3.3.1. Usage in literature

A meta study was performed on a book by Dua and Du [24] and on three papers by respectively; Tavallaee [25], Lappas & Pelechrinis [8] and Tsai et al. [26]. The book and papers give an overview and discussion of single algorithm detection methods in scientific literature. Analyzing these meta studies gives a percentage wise distribution of the analysis methods used in research over categories of analysis methods. An overview with the main disadvantages and advantages and a description of the basic concepts of these methods is given in Table 1. More elaborate explanations of analysis methods are given in appendix 3. The resulting distribution over categories is presented in Figure 5. Figure 6 gives a distribution of the most popular classification methods.

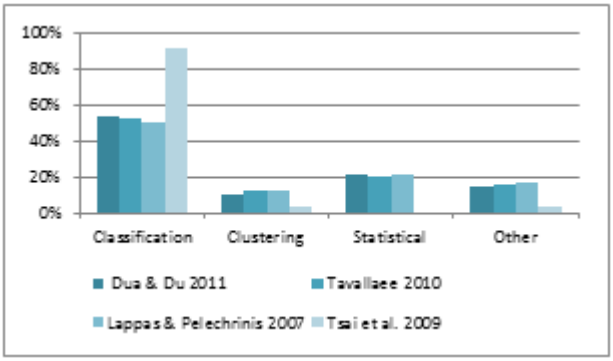

Figure 6; distribution of papers over the categories

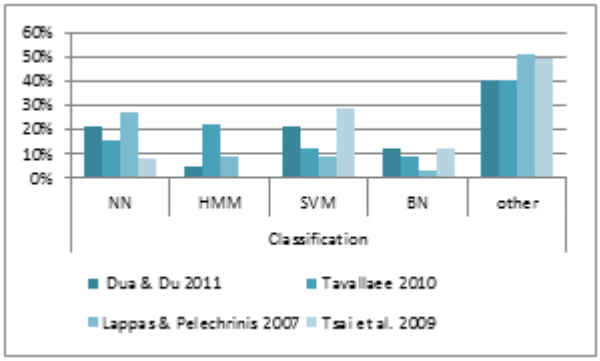

Figure 5; distribution of papers on classification methods

The amount of papers published for each method and category can be used as indicators of their popularity. Popular methods are those which give good results which are worth publishing. An overview of the frequency of methods is biased since very common methods, like simple rules, fall outside the scope of most research papers. Comparing the distribution of papers over the categories is further complicated by different interpretations of categorization of certain methods in literature. Lappas et al. for instance, categorize Bayesian networks and support vector machines as statistical measures. The paper by Tsai et al. does not mention Hidden Markov Models (HMM), but has an overly large count of Support Vector Machine (SVM) papers. They also have a low number of statistics based papers compared to the other sources. An explanation might be that they consider HMM being a purely statistical method and that they do not consider statistical methods in their paper which is supported by the fact that they only introduce pattern classification as goal for machine learning. They do use this boundary lightly since they also count self-organizing maps and Fuzzy logic as classification methods, where the first is more often considered a clustering algorithm and the latter is a method to handle linguistic uncertainty in other methods. For this comparison the categorization used by Tavallaee et al. was used since they performed the most extensive literature study to describe the research in the entire field of anomaly-based intrusion detection. These differences in interpretation have no large influence on the general distribution pattern visible in Figure 5 and Figure 6.

Figure 6 shows that classification methods are by far the most popular methods in scientific literature for intrusion detection by intelligent data analysis. Classification categories are generally Supervised learning methods which have a lower false positive rate. This could explain their popularity in research. The overview of the distribution of methods in the classification category as presented in Figure 6 shows that the four most popular methods cover 50-60% percent of all papers. The most thorough literature study by Tavallaee only uses these categories, so a more detailed overview is hard to give. The other papers however indicate that k-Nearest Neighbor (k-NN) and Decision trees (DT) are the most popular in the “other” category in Figure 6.

3.3.2. Accuracy of single methods in literature

Comparison studies of single method approaches show that these methods can reach detection accuracies close to 100% for probe and denial of service attacks. Results for privilege escalation attacks are below 20% for most methods [27] [28]. The accuracy figures of most of the studies are based on a dataset created by DARPA in 1999. The attacks in this dataset are more and more considered to be outdated. The problem is that the DARPA dataset is one of very few public datasets which can be used for performance evaluation [29].

Table 1; Summary of intelligent data analysis method families discussed in appendix 3.

Category	Method family	Disadvantage	Advantages	Basic concept
Classification methods	Simple rules	Complex situation are hard to define in simple rules	Simple to use	If…then… like rules
	Neural Networks	Black box	Can handle complex relations	Mathematical model calculating output based on inputs
	Support Vector Machines	Slow on large sample sizes	Small chance at over fitting and possible to use dynamically	Classes are separated by a hyperplane by calculating support vectors to the closest points from each class
	Bayesian Networks	Cannot handle missing data well	Efficient with causal relations	Probability of events
	Decision Trees	Very hard to find optimal solution	Can handle numeric and text data types	Classification by If..then.. like tree structure
	k-Nearest Neighbor	Storage intensive and susceptible to noise	Simple to implement	Distance/density calculation between case and classes
	Hidden Markov Models	Events must be independent. (The events may not provide a probability of a following event.)	Can analyze sequences of events in which the events are not independent	The probability of a sequence of observed events is used to calculate the probability of a sequence of non-visible events.
	Kalman Filter	Accuracy declines when noise and initial states are not normal distributed	Can analyze sequences of events in which events have consequences for future events	The observed system state and previous states are used to estimate the next state
Clustering methods	Shared Nearest Neighbor	Initial choice of parameter values	Insensitive to noise and cluster shape	Clustering based on equality
	k-Means	Hard to find optimal solution and sensitive to cluster shape.	Pre-classification not necessary	Clusters data into a given number of k clusters by minimizing the mean distance to a cluster center
	Self-organizing maps	Resulting model is a black box and creating a model is computational intensive	Good reduction of data feature dimensionality while maintaining relationships between the features	Neural network where output neurons are pixels of a density map and similar cases are mapped close to each other
Statistics	Statistical Methods	Needs more preprocessing and is susceptible to training by attackers	Statistical methods can help to reduce the feature space for supervised learning methods	Frequency analysis, correlation, regression calculations
Supporting methods	Boosting	Requires more calculating time and configuration	Increases accuracy	Strengthening weak classifiers
	Fuzzy logic	Increases complexity	Gives a possibility to indicate certainty of classification.	Handling of linguistic uncertainty problems by overlapping classes

3.4. Hybrid detection systems

Hybrid systems use multiple algorithms for detection. Each algorithm has a specific purpose and the combination of algorithms ensures that as many attack methods as possible are detected.

3.4.1. Systems designs

More complex attacks like the multistep attacks discussed in chapter two require correlation of different low-level attack methods for detection [18]. This means that multiple single method detection algorithms need to be combined for successful detection of APTs to detect the different lower level attacks. Since the late 80's a long list of research systems has been created aimed to identify intrusions or attacks in computer networks. The idea behind most of these systems is that accurate and successful detection of intrusions can only be achieved when different methods are combined. For most systems this means combining signature detection methods with anomaly detection or multiple detection algorithms and correlation. Early research systems like NIDES, EMERALD, MIDAS and Haystack use statistics to find behavior anomalies. Later research systems like MINDS, PAYL, Bro and Poseidon use machine learning based anomaly detection algorithms like neural networks, fuzzy logic and Markov models [13].

The same goes for commercial products, although they do not state which specific methods are used. They often use statements like signatures, log analysis, traffic analysis and heuristics. Especially for large scale products this does imply the use of some sort of intelligent algorithm. In some cases it is publicly stated which algorithm is used, like the application of neural networks in the detection of boot sector viruses in Norton Antivirus. In other systems like Trend Micro's Deep Discovery [30] or HBGarys' Digital DNA [31] the use of intelligent analysis methods are implied but not made explicit.

The above mentioned systems combine different methods. The way these methods are combined differs. Dua and Du identify three main types: Anomaly detection followed by signature detection or vise-versa, anomaly detection parallel to signature detection followed by a correlation algorithm and a complex mixture of methods [24].

The series design utilizes an anomaly or signature detection algorithm as preprocessing method for another signature or anomaly detection algorithm. Such an approach limits the amount of data for the latter algorithm to analyze, but has an impact on the choice of data analysis algorithm. Especially classification algorithms that define anomalies as outliers will not work with this approach if the first step is a reduction of data by eliminating the normal traffic [32].

In parallel designs different algorithms are analyzing the input data and report possible attacks. A reasoning or correlation system uses the output as to decide whether an attack is occurring. The advantage of such a system is that it does not require all systems to do real-time analysis. The different algorithms used to analyze the input data could even use separate hardware. This design is similar to the distributed system design when the input data is not the same for each algorithm. A parallel design offers possibilities for detection of targeted attacks as each algorithm could focus on attacks from different steps and the final decision system could then decide if an attack is occurring. This could be considered the simplest design capable of detecting targeted attacks.

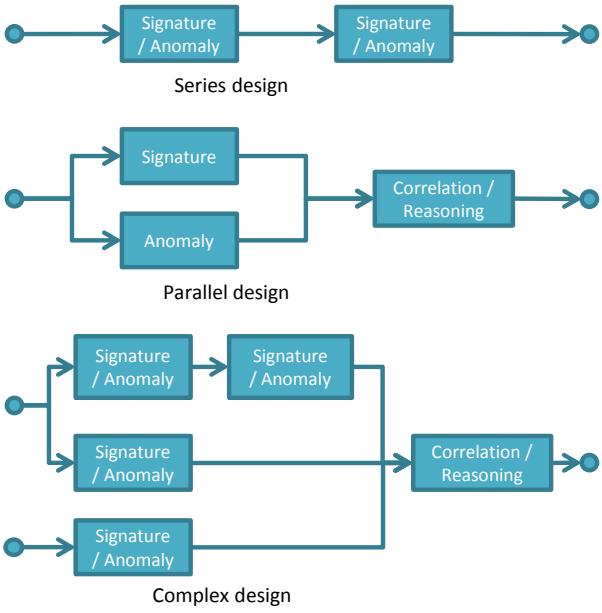

Figure 7; conceptual system designs

The third option, a complex design, is a combination of the series and parallel designs. The different series or parallel elements of such a design might be distributed over different systems to distributed the analysis and accompanying costs in analysis time, over multiple hardware systems. Detection of APT requires such a complexity to detect the subtleties of targeted attacks. The fact that not each steps has to raise an alarm already implies a more complex system since it creates a dependency on relations betweens events at different places and times in stead of the maliciousness of a single event. The success of such a system depends on the correlation or reasoning on the results of the separate parts of the system.

All three designs combine analysis methods and are therefore sensitive to propagation of detection errors [33].

3.4.2. Data acquisition

The design of a system is dependent on the location of the system. This location determines the input data type. The simplest solution is to use data from only one source. This scheme is used, for example, by Snort [34], using capture traffic from one location or by virus scanners analyzing behavior and files on a single computer or server. These systems focus only on single attack methods and not on attack scenarios. They might not even detect all lower level attacks in an APT because of the use of attack methods unknown to the detection application. This reduces the chance of detecting an APT.

A second option is to create distributed or collaborative systems which are spread across a network and collaborate to detect attacks. Such spreading of systems is be able to cover all the traffic streams associated with the different attack steps as shown in Figure 2. The distribution of data acquisition over the network does increase the chance of detecting attacks.

Distributed data acquisition introduces propagation of errors which increases the amount of false classifications. This is even more the case when a complex design is used for the analysis system design.

3.5. Challenges of intelligent data analysis for APT detection

The last decade has seen a lot of research on intrusion detection with intelligent data analysis and even implementations in commercial products. But there are still some challenges remaining when applied to intrusion detection.

First of all anomaly detection methods still suffers from large numbers of false positives which makes them less dependable and harder to use in production environments [24] [29]. Experts need to be available to analyze all alerts to identify the true positives. Another aspect is the high computational need, especially in networks with large amounts of traffic. Another challenge faced by network intrusion detection systems is the increasing use of encryption of data. The only usable features of encrypted traffic are some packet header elements or statistical information on traffic. A challenge for research is the lack of good datasets to use for testing. Most Academic Researchers are still using the datasets created by MIT in the late 90's, while attackers have been increasing the complexity of their attacks over the last years up to the point of the targeted attacks described in chapter two and they will keep changing continuously [32] [29].

A system which uses different algorithms and combines the results of these for further analysis runs the risk to accumulate errors resulting in an even higher false classification rate [33]. A design for detection of APTs therefore needs to handle the uncertainty on classifications to prevent an overkill of false positives. Confidence levels for classifications and alerts next to the common severity level reports could be used to mediate a possible increase. Another approach to reduce false positives is to use data labeled by experts to classify attacks [10].

3.6. Conclusion

This chapter started with introducing three main approaches to intrusion detection. The first was detection based on signatures, which are known detectable features of attacks. These systems check for these features and give a warning when a match is found. The second approach is anomaly detection. This approach does not use signatures but is trained to detect deviations from normal behavior. The first approach lacks the

ability to detect unknown attack methods, but has a high accuracy. The second approach can detect unknown attack methods, but has a lower accuracy rate. The third approach in intrusion detection combines these two approaches. The aim of this is to reduce the lower accuracy of anomaly detection while keeping the ability to detect unknown attack methods.

A meta study of literature on anomaly detection showed that Neural Networks, Hidden Markov Models, Support Vector Machines and Bayesian Networks are the most common intelligent data analysis methods used for classification. This implies that these methods are most successful in research in anomaly detection. The success rate of these methods is dependent on good data feature selection and data preprocessing. Anomaly detection has the potential to make APT detection more successful because APTs often use unknown attack methods. Accuracy of detection is also important. The third approach to intrusion detection is therefore a logical choice for an approach to intrusion detection. Combining signature and anomaly detection into one system can be done in several ways. Three general approaches can be identified. From these three a complex design offers the most flexibility for combining signature detection with anomaly detection methods. This also creates the possibility to create a distributed system with multiple data source in the network and on computer and servers.

Such a system to detect APTs does have some challenges. Anomaly detection suffers from false positives and combining these methods introduces the risk of error accumulation reducing the accuracy of the total system. Introducing confidence levels and using only supervised training methods can help in improving the accuracy.

4. Business aspects of APT detection

Successful cyber attacks can have a large impact on corporations. One should not only consider the financial costs of fixing the damage but also damage to the corporate image and possible secondary damages for customers. Detection of intrusions is therefore important especially when the risks increase. The first paragraph discusses the relationships between the different steps of a targeted attack and the business perspective. This is followed by an analysis of the consequences of this relationship for the other aspects in the framework from the second chapter.

4.1. Relations of the business aspects within the framework

The framework shows the different steps of an attack and accompanying aspects related to detection. The attack related aspects influence the system design and the choice of analysis methods. The business aspects are also influenced by the attack steps through the possible impact when an attack occurs. The research by Rakes et al. states that investments in security measures are influenced by the expected financial impact of an attack [15]. The impact of attacks increases as an attack has more access to a network [35]. The return on investment in an intrusion detection system depends on the ability to reduce the impact of an attack. The effectiveness of an intrusion detection system is influenced by the system design and the choice of algorithms [36]. The possible impact of attacks therefore also determines the system design and the choice of analysis methods.

4.2. Business aspects of APT detection

Business aspects are influencing the design of a detection system as requirements and limitations. The previous paragraph showed that performance of a detection system is the most important requirement from a business perspective. In literature different approaches to measuring the performance of a detection system are used. Lee et al. state that there are four performance objectives for intrusion detection systems; good detection, economic resource usage, resilience to stress and secure against attacks [37]. Lee et al. propose a cost based approach to determine tradeoffs. Iheagwara et al. state that an IDS, from an investment perspective, needs to show that it reduces the losses incurred through cyber-attacks [36]. The business consultancy firm PWC mentions accuracy, performance, security, scalability and cost as important aspects of IDS usage [38]. Dua and Du name privacy issues as an important issue in data mining for intrusion detection [24]. Performance, cost and privacy aspects are discussed below.

Effectiveness, accuracy and scalability are aspects related to the technical design of an IDS. These aspects are influenced by the nature of APTs as discussed in chapter two and the choices in system design and algorithms discussed in chapter three. An important measurement of the correctness and accuracy of detection are the number of false positives and false negatives. Many false alarms require a lot of manual analysis to determine if an attack is really occurring. A system is useless if it is not capable of detecting attacks or when it considers attacks to be normal behaviour. A system which uses data from multiple locations in a network shows a higher accuracy than systems with single detection systems [36]. A survey of intrusion detection systems by Zhou et al. shows that a distributed systems offers better scalability and that successful systems use multiple analysis methods for detection [33].

According to Igheagwara et al. should the cost of a system not exceed the possible incurred losses by cyber-attacks [36]. Lee et al. calculate the total cost of a system by subtracting the cost incurred by an attack from the implementation cost of an intrusion detection system [37]. They use this principle to show that a better economic performance is reached when multiple algorithms are used for a higher accuracy. The impact of an APT can be considered high if proprietary information is stolen and according to Rakes et al. this approves the investment in more complex security measures [15]. The consequence for the design approach is that a balance between the cost and the accuracy of the design is important. A system with multiple sensors and data analysis algorithms is warranted as long as this improves the accuracy and therefore reduces losses due to an APT.

Capturing, storing and analyzing network traffic can be considered to be invasive to privacy [24]. Creating detailed profiles of network use by individual network clients could be considered as invasive. Deep packet inspection techniques are also considered harmful to the privacy of network users. On the other hand is detailed information important for successful detection of APTs. Countermeasures like obfuscation or privacy preserving data mining can help to ensure privacy of network users. The un-obfuscated data on the other hand has to remain accessible for detailed analysis in the case of a real attack [24]. Technical measures do still remain necessary to ensure that the data cannot be used in a privacy breaching manner. These privacy issues might not directly apply at the data analysis level, but they might become an issue when traffic needs to be analyzed.

4.3. Conclusion

Business aspects in the framework are influenced by design choices and the impact of an APT. The cost of an APT occurring can be considered to be an estimate for investment in an APT detection system. The consequence is that the accuracy of a detection system is the most important aspect from a business perspective. A system with too many false warnings has higher operation costs but a system that misses attacks does not prevent losses. Cost analysis of systems in scientific literature shows that systems use multiple methods are considered better from a cost perspective.

Finally privacy should be considered when designing a detection system. Data can be obfuscated before analysis and reporting, but analysis of warning might require the original data. Privacy issues might not be a big influence on the choice of analysis algorithms but they should be considered for the front end part of a system design.

5. Road map towards APT detection system design

The framework presented in chapter two provides an overview of attack steps and aspects related to attacks. The attack related aspects are discussed in chapter two, the analysis and detection method related aspects in chapter three and the business aspects in chapter four. These aspects together determine the goals, requirements and design of an intrusion detection system against APTs. This chapter presents an approach to a design starting with defining the design space as it is delimited by the previous chapters. Following the design space a conceptual design considering the different options on detection location, data and analysis methods within the design space is discussed. The conceptual design will help to determine the answer to the main research question in the next chapter.

5.1. The framework as roadmap

The goal of this thesis is to design a system to detect APTs. The framework presented in chapter two defined seven aspect columns of attacks related to detection of APTs. The influence on the design of a detection system of the aspects is shown in Figure 8. The attack related aspects describe the attacks and therefore what should be detected. The business aspects are driven by the impact of the attacks and they give therefore the reason for detection. Detection locations are the places where attacks can be detected. A choice of detection location determines the input data for analysis. Detection and analysis methods determine how attacks are detected. The inward arrows in Figure 8 indicate the individual influence of the aspects in the framework on the design. This direct influence is not the only influence direction. The attack aspects also influence the business aspects and these in turn influence choice on detection aspects. Attack aspects also influence the network locations and these in turn influence the detection aspects. These direct and indirect relations determine the design space.

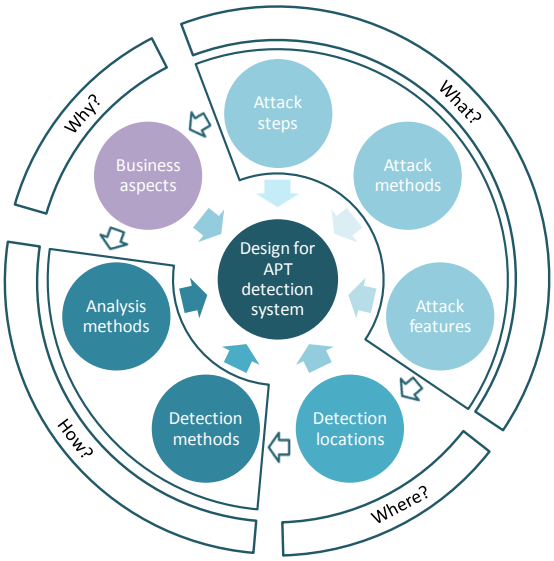

Figure 8; Elements of the framework grouped by type

The analysis of APTs in the second chapter of this thesis showed that APTs are harder to detect than traditional multi-step attacks because of the efforts of attackers to avoid detection. Traditional intrusion detection systems are therefore limited in their capabilities of detecting APTs. Multistep attacks can be identified by correlating events. This requires that individual low-level attacks are treated as possible elements of multistep attacks. The stealthy nature of APTs requires that events which are possibly part of an attack but not necessarily harmful are also recorded for analysis. In chapter four Iheagwara et al. are cited stating that detection on one single location in the network is also not enough to effectively detect attacks. Multiple instances of traffic analysis sensors where shown to have a higher efficiency rate [36]. Capturing not only traffic but also events on clients and servers can further improve detection. This results in the need for a central location where events from the individual systems is combined and analyzed. Combining information

from different sources requires a careful handling of uncertainty about detection accuracy to prevent the propagation of detection errors, especially in the case of anomaly detection algorithms [9].

The more technical design oriented aspects above are not the only design factors. Business aspects also influence the design. Companies will balance the impact of an attack with the costs of a detection system as mentioned in chapter four. This requires a balance between effectiveness and the costs of a system.

5.1.1. What?

The first columns in the framework are attack related aspects. The attack steps provide insight into the high level structure of attacks. The attack methods column and the detectable attack features column describe the low level attacks which can be used to achieve the goals of each attack step. Linking individual attack methods to attack steps through the framework makes it possible to link low level attacks to high level attack steps. Finding a sequence of attacks each corresponding to another step is an indication of a high level attack.

The attack features column in the framework provides a non-exhaustive list of features which are input for the selection of data features for the analysis methods. These features can be input for the creation of signatures for signature detection. They can also indicate how the behavior of a successfully attacked client can change due to a specific attack method.

The system must not only look for specific features in, for example, network packet contents, but also at changes in behavior and sequences of detected events.

5.1.2. Where?

Detection of the attack steps and attack features is limited by the location where data is gathered. Data can be gathered from log-files, by looking at activity in computer memory or by capturing network traffic. A choice for one type of detection location limits the features which can be detected. This in turn limits the amount of low level attacks which can be detected. The result might be that not all attack steps of a high level attack are detected. In a worst case scenario does this result in a false negative: An attack has occurred but is not reported.

Research into the cost effectiveness of intrusion detection showed that multiple sensors gave the most accurate results [36]. This idea is supported by research into IDS system designs by Zhou et al. They found that IDSs aimed at complex attack scenarios used multiple analysis methods and correlation of the results of these methods for detection in distributed systems [33]. In chapter two it was shown that most steps of APTs generate network traffic. It is therefore a logical choice to start with analysis of network traffic to detect APTs. Multiple probes which capture network traffic should therefore be deployed on the network. Minimizing the number of probes would keep the costs down but this also decreases the accurateness of detection. The physical segmentation of large networks creates possibilities to strategically locate probes. The traffic captured in these probes should be sent to a system that analysis the traffic captured by one or more probes. These systems in turn should send events and statistics to a central system which can do analysis on a higher level. Such a layered approach reduces the load on the individual analysis systems while maintaining scalability.

Analyzing traffic alone is not enough. Activities in attack step four, expanding access, are partly executed on clients or servers without generating traffic. Client applications monitoring activities on the workstations and servers can fill this information gap. The development and maintenance of clients for workstations and servers is costly and IT departments might object to installing such software clients as it might conflict with application used. A solution might be to use reports from other programs for analysis. But using of the shelf software is difficult since there are no common communication standards available. An attempt to create such a protocol, called IDMEF, has never gone further than an experimental state [39].

5.1.3. Why?

The impact of a cyber attack is the main driver to invest in defenses. The risk of a high impact event can be reduced by reducing the chances at a successful attack. From a business perspective this means that a

detection system should be effective, accurate and secure against attacks itself. For the system design this requires the system to be able to detect attacks without to many false warnings and to keep operating if the system itself is under attack.

Investments in these aspects are limited by their costs. The maximum accepted cost of a system can be determined by cost/benefit analysis. The costs of the system should not be higher than the expected losses through successful attacks. A system with a lower accuracy produces more false classifications and requires more attention of system operators. This makes the system more expensive to operate. Reducing the number of false warnings, by using only more accurate signature detection, results in a less effective system which does not reduce the risk of an attack. A balance between accuracy and effectiveness is therefore necessary to achieve a good cost/benefit ratio.

5.1.4. How?

The previous paragraphs determined what needs to be detected, where that can be detected and why it should be detected. All these aspects influence possibilities and choices on how it should be detected. Detectable features of known attack methods can be found in network traffic, workstations and servers. Creating signatures and performing signature detection ensures accurate detection but the chances at detecting unknown attack methods are slim. Anomaly detection provides a possibility of detecting unknown attack methods by looking at changes in behavior. A system for detection of APTs therefore needs both signature and anomaly detection.

Anomaly detection

When intelligent data analysis is applied to anomaly detection a choice must be made for a learning approach. The first approach is supervised learning which uses a labeled dataset to create a classification model. The second approach is unsupervised learning which classifies on the assumption that anomalies are differ from a normal situation [24]. The challenge with the first method is that creating a labeled dataset requires a lot of time. This time can also be used to create signatures for the signature detection blocks. An advantage of unsupervised learning methods is that they adapt to changes in what is to be considered normal. These changes are one of the causes for the false positive problems experienced by anomaly detection methods. On the other hand does this also introduce a weakness: an attacker can train unsupervised methods to accept attack related behavior by slowly starting the attack [24].

Anomaly detection on captured data

Capturing data at multiple locations in a network is needed for effective detection of APTs. Comparing clients by means of clustering algorithms can identify clients which behave differently. This approach can however create false classifications if the analyzed physical network segments cross organizational boundaries. For example: A client which behaves differently might belong to a different department instead of being taken over by an attacker. Knowledge of the network and careful choice in placements of probes can prevent such problems.

The type of data captured determines the choices for algorithms. Self-Organizing Maps are effective for analysis of data features. SOM is especially effective when the feature space is large like data from network packets. Behavior analysis requires preprocessing to create data for analysis. A list of connections from a client needs to be changed to connections per time interval to be useful for behavior analysis. The choices in preprocessing determine if anomaly detection at a client level can effectively and accurately classify attacks as anomalies. Clustering algorithms are the most effective unsupervised learning algorithms for anomaly detection. Clustering algorithms that have shown good results are k-means clustering and self-organizing maps [24]. To prevent false classifications semi-supervised methods can be used. In such cases are a limited number of normal events labeled, these labeled events create a start for clustering algorithms [24].

Anomaly detection on reported events

Anomaly detection in the central analysis element is more difficult because of the large number of possible sequences of low-level attack methods in multistep attacks like APTs [33]. The consequence of the large number of sequences is that it is harder to define normal behavior. Unsupervised learning by clustering

algorithms can still be used to identify sequences of anomalous behavior but they will generate a high number of false classifications. The false classification rate can be improved by combining the results from different clustering algorithms like shared nearest neighbor and k-means. Event sequences classified as anomalous by both algorithms have a higher chance of being a true positive than those which are only classified as anomalous by one. Such an approach is called a boosting configuration.

Supervised learning algorithms can be used but face the challenge of the creation of labeled datasets for learning. Single methods capable of analyzing event sequences, like Hidden Markov Models and Kalman Filters, have successfully been applied to system call sequences [24], but not to alert sequences related to multistep attacks. More complex approaches like the ones proposed by Ning et al. [18] and Yang et al. [20] use knowledge about lower level attacks to correlate events to create attack scenarios. Ning et al. use this knowledge to create hyper alerts which describe prerequisites and consequences of general attack types. Yang et al. tries to match alert sequences to known attack sequences and tries to match the results to information exposure sequences. The information exposure sequences are seven stages ranging from external reconnaissance to achieving an internal network goal. These stages are very similar to the eight steps from the framework defined in chapter two. Yang et al. state that alert correlation methods are still in the infancy state and that a lot of research is still required.
The approach of using knowledge about the structure of APTs by labeling the events from the local analysis according to the steps they belong to can help to create better event sequences for anomaly detection.

Other applications of intelligent data analysis

Another useful implementation of intelligent algorithms is to add more intelligence to signature detection. Examples are the creation of decision trees for rule application to reduce the analysis time when there are a lot of rules in the system [40]. Another option is to implement rule learning approaches. An example is fuzzy rule-based anomaly detection [24]. This approach uses labeled datasets which can be derived from the clustered data from the anomaly detection block described above. The accuracy of this labeled dataset can be increased by using the decisions of system operators on reported alerts as labels.

5.2. System Design

The result of following the framework as a roadmap is a distributed system which uses both signature and anomaly detection to look for low level and high level attacks. A possible system design with four main elements is proposed in this paragraph. The next chapter will test this design and gives a reflection on the design choices and ultimately the effectiveness of using the framework as a design roadmap.

The four elements of the proposed system design are: A probing element, a local analysis element, a central analysis element and a reporting element. The probes gather data. This might be network traffic data in IP-packet format or data from software clients on computers and servers. Multiple probes are deployed in a network. The probes pass the data to local analysis elements. These elements perform analysis to detect low level attack methods. They report an event to the central analysis element when they find a possible low level attack. The central analysis element combines all attack events and tries to correlate low level attack events to APT attack scenarios. The central analysis element finally passes possible attacks to a reporting element. The general structure of this system is shown in Figure 9. The analysis methods in the elements are discussed in the next paragraph.

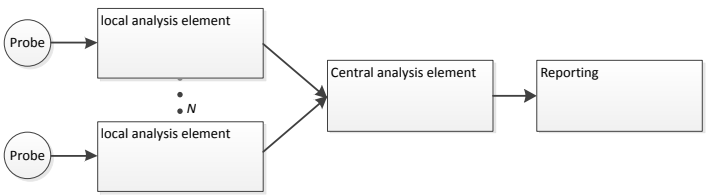

Figure 9; General system design elements

5.2.1. Analysis elements

The two different analysis elements from the system design each have a different internal structure. This paragraph describes the structure of both elements. Both elements follow the third approach for intrusion detection presented in paragraph 3.1. The signature detection elements use manually constructed signatures like those used in SNORT. Paragraph 5.2.3 will discuss the intelligent data analysis algorithms used in the anomaly detection elements. Each analysis element does not necessarily imply a piece of hardware. The analysis elements can be combined on a single server if system requirements allow for this.

Local analysis element

The internal structure of the local analysis element is shown in Figure 10. The local analysis element of the system receives packet data from a traffic probe. The data from the probe is input for the preprocessing block which provides input for a signature and an anomaly detection block. The preprocessing block acts as a filter for the signature detection block by removing all packets related to protocols which are not analyzed. This selection might differ per location. The preprocessing block creates frequency data per time interval on network client level for anomaly analysis. Creating frequency data on network client level allows for combining events from both the signature and the anomaly detection blocks in cases where both blocks generate an event on for the same attack. It also creates the possibility for the central analysis element to correlate events reported by different local analysis elements. The resolver block combines events reported by the signature and the anomaly blocks in cases where they both detect the same attack event. These matches are made on a client and timestamp basis. Finally the resolver sends event reports to the central analysis element.

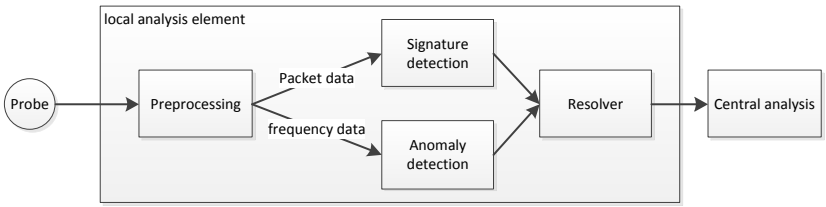

Figure 10; internal structure of local analysis element

A traffic based intrusion detection system can only analyze network traffic and cannot analyze activity on computers and servers. Adding software clients allows gathering information on these systems. The use of such clients helps to detect more low-level attack methods and it also reduces the dependency on traffic analysis. The structure of such clients can be the same the local analysis elements described earlier. For software clients a choice can be made to put the probe and the analysis elements both on a client or to separate them and have the probe send its data to a central system. The first choice might use more CPU time on a workstation which can be prevented by the second choice. The second choice has the advantage that it can compare workstations. The disadvantage is that it needs to perform the analysis for all clients on a single location.

Central analysis element

The central analysis element receives event reports from all local analysis elements. The internal structure of the central analysis element is shown in Figure 11. The first block in the internal structure of this element is the input resolver. This resolver receives the individual events from the local analysis elements and tries to combine events by comparing time and client information in the events. The input resolver has two outputs; one output delivers the combined events are reported to the output resolver as low level events and to sequence preprocessor. The sequence preprocessor maintains overviews of events ordered by timestamps and clients. These overviews are input for the detection blocks. The signature detection block uses known multistep attack scenarios to identify attacks. The anomaly detection block compares sequences on client level to identify anomalous sequences. Both detection blocks report to the output resolver. This resolver checks for duplicate events. Duplicate events are sequences reported by both the signature and the anomaly detection blocks. Duplicate events are also low level events reported by the input resolver which are part of a detected event sequence. The events remaining after elimination of duplicates are passed to a reporting element.

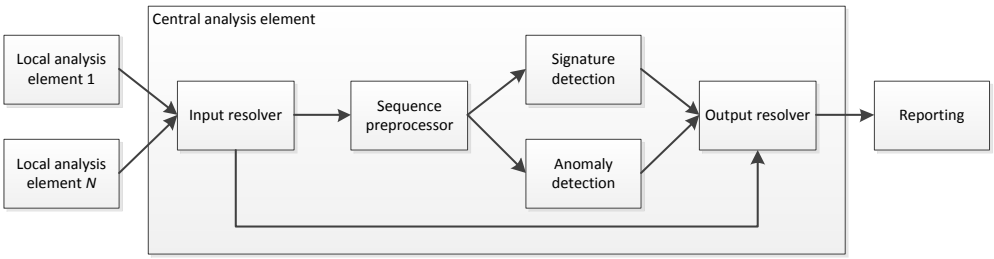

Figure 11; internal structure of central analysis element

Reporting element

The reporting element is the systems interface for analysts. Events are ordered here by severity. This severity is based on impact. The impact of an event is higher when it comes closer to access to proprietary information. The reporting element receives both detected attack scenarios and events that could not be related to an attack scenario. A single event can still have a high impact for an organization and should therefore not be lost.

5.3. Conclusion

The framework presented in chapter 2 can be used as a roadmap for the design of a detection system. The information in the framework provides choices for the system design by analyzing attacks and possible detection approaches. The business aspects in the framework act as limits to the design choices provided by the first six columns of the framework. The content in the framework can be summarized as answers to four questions: What must be detected? Where can it be detected? Why must it be detected? How can it be detected? These summaries provide design choices for a system design.

To detect APTs the system must be able to detect known attacks and unknown attacks. Known attacks can be detected through signature detection methods. Unknown attacks can be detected by looking at behavior changes or anomalies in behavior. Not all attacks can be observed in network traffic. Data from workstations and servers must therefore also be analyzed. Distributed system designs also improve the effectiveness and accuracy of a system. The design of the system is influenced by a balance between effectiveness and accuracy on one side and the cost of the system on the other. Investments in defenses against attacks are determined by the expected financial impact of attacks. This relationship puts a limit on the costs of the system. The choice of analysis methods ultimately determines how effective and accurate a system design will be in detecting attacks. Signature detection should be the basic technology and should be supported by anomaly detection which should be focused on changes in behavior. Unsupervised learning methods are inaccurate in identifying attacks, their added value however is important enough to use them. Their in-accuracy can be improved by using input from experts to shift to semi-supervised learning, or to use multiple clustering algorithms in a boosting configuration.

A system design consisting out of four elements is proposed which corresponds to the answers to the above mentioned questions. Different probes and local analysis elements are used to analyze captured data. A central analysis element uses the warnings of these systems to look for high level attacks. The low level and the high level elements are presented in a reporting system to experts which can decide on appropriate actions.

6. Test case and reflection

In this chapter the system design is applied to the example attack presented in chapter two. The result is used to reflect on the system design and the design approach with the framework as roadmap.

6.1. Application of the design approach to the example attack

This paragraph describes for the target attack, per described step, where and how it would be detected in the design approach and how it would report. The italic text is the example from chapter two. The design approach is considered to have the necessary number of probes and the probes contain a signature detection element and a statistical profiling element which is used to classify events. The central processing system uses signature and sequence analysis as well as comparisons and classification of traffic profiles.

Step 1; reconnaissance: The first step of the attackers is reconnaissance of the target company. They start by browsing corporate websites for names and mail addresses, check DNS registrations to find public accessible services and check the open web for social media profiles of people claiming to work at the target company. The main goal is to find handles for social engineering approaches and to find version information on servers and website content management systems, to find exploitable vulnerabilities.

Activities in this step are performed in open source on the internet and are therefore outside the monitored network of the target company. Browsing of the corporate website can be logged, but this does not give any information on attacks since this is normal behavior. Providing information about the company is what the corporate website is for. This first step is therefore not detected.

Step 2; gaining access: After the first step the attackers proceed to use the profile information of employees to construct phishing emails which look legitimate. These emails contain a link to an infected website which uses a zero day exploit to install a malware component on the victim's workstation.

This step is the first to generate events. The signature detection element in the first probe detects the presence of the hyperlink in the email. The mail is therefore flagged and relevant information, like the link address, addressee and source is recorded. The system starts looking for traffic to the address in the link. This is also recorded with relevant information like the address of the internal workstation from which a connection to the linked website is initiated and a timestamp. The malware installed from the website is not detected by the signature detection block in the local traffic analysis element because of the lack of a signature. The recorded information from both the mail and the website visit is reported to the central analysis system as separate events. This system links them as a possible attack, but does not report it since the probability of this series of events being normal is too high.

Step 3&4; internal reconnaissance and expanding access: Once the attackers have gained a foothold in the network through the malware they will try to expand their access to other parts of the network. The malware starts to monitor connections to servers in the network, gather information about installed programs and network users to identify server addresses, network structure and possibilities for expanding access. Un-patched programs or operating systems create possibilities for further expansion of the attackers' access to network clients and servers. The attackers also perform active reconnaissance on the network themselves by connections through the malware clients.

These steps generate traffic over the internal network like connections to internal services and traffic to the internet like command and control traffic to the malware. The signature element in the probes can detect port scans and other known reconnaissance techniques. Such behavior might be found if the attacker has access to a workstation of an employee which does not use the financial applications and the attacker is looking for a means to connect to these applications.
The anomaly detection block using traffic data frequencies identifies the unusual traffic between clients and servers raising another possible suspicious event. Individually this event might not be suspicious. It could be legitimate traffic which just doesn't occur often. But reported to the central analysis element it is linked to the

previous events resulting in a sequence of events which could be an attack. A low level warning is raised indicating that an attack might be ongoing and manual analysis is desired.
Expansion of user privileges on an infected workstation or infections of clients in the same segment as the infected machine goes unnoticed by the probe since the first does not create suspicious traffic (assuming that the command and control traffic to and from the internet is obfuscated). The latter activity generates traffic which does not pass the probe and can therefore not be detected.

Step 5&6; gathering and extracting information: After a while the attackers are successful and have found the wanted technical documentation and have access to the financial systems of the target. They slowly gather all the information on one of the clients they control and prepare the information for extraction. Finally they extract the information to a legitimate file storage application on the internet to make the extraction look as normal as possible. They also continue snooping around for other data they can sell and extract this as well.

The anomaly detection element will see a change in traffic volume in step 5. This is classified as suspicious and reported. In the central analysis system this increases the warning level. This does require that the increase of volume is on the system which was initially flagged by the email. If this is not the case then the system will still report the anomaly but it won't be linked to the already present warning in the central system.

A change in traffic volume to the internet from a workstation is classified as an attack and is given a high warning level in the central system. This might be legitimate traffic, but in such a case it is preferable to have a false positive then a false negative. The chance at a false positive is lower when there is a direct link to other steps. Weaker links, like individual steps are reported with different workstations involved but in the same network segment also reduce the chance at a false positive. Such relations can be identified by the signature element in the central analysis system. If the uploading to the internet is done from internal services it can be immediately flagged as suspicious by both signature and anomaly detection since this can be considered to be uncommon behavior for almost all internal services.

Step 7&8; control and erasing tracks: The attackers have continuously monitored progress through direct access through a backdoor created by the malware and by updates from the malware to servers on the internet. After extraction of the last of the wanted information the attackers hide their tracks by uninstalling the malware.

The removal of the malware will be unnoticed by the network probes since it occurs on the workstations. The command and control traffic of the malware clients and the traffic generated by the activities of the attacker through the malware could be detected by the network probes. Signature detection will only work if such traffic contains known suspicious elements. Statistical profiles have a better chance at detecting such traffic as unusual. It is probably hard to distinguish such a change and this would therefore only justify a warning when at least steps two or three are detected.

6.2. Evaluation of the system design

In the example the system raises some warnings but finds no complete attack sequence. Successful detection is therefore still dependent on the analysis done by experts who monitor the system. The system also does not actively block an attack which means that constant monitoring is required to stop attacks.

The example also shows critical weak points. Attacks that trigger only one event are not reported with a higher level warning except for step six. This situation should not be considered uncommon since it also includes attacks where different steps are executed on different workstations or even network segments since these are also difficult to link to each other. The example shows a dependency of the system on these links to increase the warning level on events. This brings us to another possible weakness. A strong dependency on the event sequences will miss events that do not follow the pattern of eight steps. The eight steps are not always present, as was mentioned in chapter two, and attacks are almost always completely different. The idea of the eight steps was that they are not absolute but that elements do return in other forms of targeted attacks. This makes it useful to detect as many steps as possible, but one should be careful to draw strict conclusions from the fact that steps are missing. The reported single events can reduce the

consequences of depending on sequence analysis, but if the number of alerts in the reporting element is high they might go unnoticed. A reasoning engine can help in reducing the amount of warnings which reach experts. The rules in such a system should be constructed carefully because filtering warnings might result in missing an attack.

The propagation of detection errors through the system is not tested in this test case. The system reports both low level and high level events, but this might result in experts only looking at high level attack warnings. Especially these warnings suffer from detection errors because they are based on the results from other methods with errors. The system design tries to reduce the chances of missing an attack by giving individual events with a high impact a higher warning level. But this does not address the issues of error propagation through the system.

6.3. Evaluation of the framework as roadmap

Using the framework as a roadmap to design a system ensures a focus on all elements involved in the design process beforehand. It also resulted in a system design which is capable of detecting attacks. It provides the necessary information for design decisions but the relations between the different aspects in the framework prove to be less strict than implied by the framework. Relations exist between the different aspect groups in the framework which are not shown in Figure 8. For example: The answer to the how-question is not only influenced by the detection locations and the business aspects, but also directly by the attack features.

The test case shows that detection is not assured, even when all attack methods are known. The framework does not provide a means of decision support to determine how much may remain undetected. This is even more difficult because of the difficulties in determining economic losses due to cyber attacks.

6.4. System improvements from the test case

The test case shows that manual analysis of events remains necessary to make a decision to act on a warning. This requires that data as available for analysis. All traffic related to events should therefore be stored and accessible for analysis. The problem is that the starting point is hard to detect. Directing people to a website which contains malware can also be done by phone. If an event belonging to step three is detected it might be interesting to see what happened before. This can give an insight into how long an attack is going on but can also help to create signatures for the signature detection elements in the design.

The importance of the manual analysis also means that special attention should be paid to the interface of system. This raises design questions about which information should be given and how information should be presented. For example: Should statistics be accessible for users and what about classifications. What types of graphical representations should be available? A general answer would be that all information in the system should be accessible to analysts. This might however be limited by development costs or hardware specifications on storage or processing.

6.5. Applicability to enterprise networks

Enterprise networks differ from the simple network example presented in chapter two. They have many more segments, physically as well as virtually. A larger network also means more traffic, more probes and more possible attack elements and the result is that the computational requirements of central processing increase. The proposed system can be scaled to larger networks by deploying multiple probes and local analysis elements. Problems might surface in the central analysis element when the number of alerts increases due to the network size. On the other hand do existing intrusion detection systems show that it is possible to capture and maintain information about large networks and to use this information for analysis. This goes for research as well as commercial systems [29].

6.6. Comments of experts

Conversations with experts during the thesis project confirmed that the chosen design approach is a possible way to go. Probing at strategic locations in networks, to capture traffic between different segments, is

considered a common approach needed to get a good overview. The number of probes in a network is in practice not only determined by technical means but also by customer wishes.

The application of IDA methods for detection of attacks is a point to which the experts remain skeptical. Application of such methods in the central analysis system is on the other hand considered to be somewhat more useful. The overall consensus is that IDA methods might give added value, especially in statistical profiling, but that the common signature approach remains to be the core technology in detection.
This opinion of experts might be biased because they have learned to work with and depend on signature based systems for which security intelligence experts continuously develop new signatures against new attacks.

6.7. Conclusion

A test case with the example attack shows theoretically how the system should work and react. The result is that the design can find activities from different steps. It also shows that its focus on the different steps might hinder detection of attacks which differ too much from the step attack. Attacks different form the example attack can however be detected if the influence of the sequence based analysis is limited in reporting. Reducing the weight of sequence analysis in the reporting element also reduces the risks of error propagation through the system. The test case shows that manual analysis is an important aspect of the system. A logical consequence is that this element should receive the appropriate attention. Analysis of events also requires traffic to be stored. The dilemma is to decide how long to keep captured traffic.

The test case and system design show that the framework can be useful when designing a system for APTs. But the evaluation shows that the reality is not as strictly separated as the framework pictures it.

7. Conclusions

This chapter tries to answer the main research question from the answers to the sub-questions. The remaining uncertainty is input for proposals for future research.

7.1. Conclusions

Detection of the new form of multistep attacks called APTs is difficult. The persistent approach of the attacker in the sense of patience and the advanced approach in the sense that complex and novel methods and exploits are used complicate the detection. APTs are a new form of existing multi-step attacks. Related to these steps are attack methods and detectable features. The result of the stealthy approach is that only looking for known attack methods is no longer enough.

A framework is therefore proposed to provide a means for analysis of multistep attacks. This framework links the attack related aspects, which show what needs to be detected, to detection locations and detection methods. By doing so the framework links attack aspects to detection related question like where to look for attacks and how to find ongoing attacks.

Intrusion detection is mainly done with signature detection methods which offer high accuracy in detection. Intelligent data analysis methods are being used, but their use is limited. The continuous changes in attack methods make it hard to keep signature detection up to date. Anomaly detection can be used to find anomalies in behavior which can indicate that an attacker is active. Research since the eighties has shown that methods using statistical profiles can have some success in anomaly detection. Methods to replace signature detection or which compare against known attack patterns have limited success. Anomaly detection systems are not yet capable of replacing signature detection methods which use manually created signatures. A combination of signature detection and multiple anomaly detection algorithms can provide better results than using only signature detection. Such a system which combines methods must able to handle the propagation of classification errors to prevent a low accuracy in detection.

Using the framework as a roadmap for system design ensures that attack related aspects are linked to detection methods. It also ensures that a system design keeps the business aspects of intrusion detection in mind. Following the framework results in a system design which uses both signature and anomaly detection distributed through a computer network. Data for analysis must be obtained both from the network and from network clients to obtain a high effectiveness of the system. Anomaly detection methods should be self-learning clustering algorithms. Their inaccuracy can be improved by using multiple algorithms from whom the results are combined for final classification.

Evaluation of a system design based on the roadmap provided by the framework shows that elements of attacks can be detected. But human evaluation of warnings remains important. Intelligent data analysis only adds warnings about attacks which currently remain undetected. It is therefore useful but cannot replace traditional signature detection. Detailed analysis and knowledge of attacks is important for feature selection. Feature selection is the most important factor to determine the success of intelligent data analysis. The proposed framework can help in analyzing attacks and linking attack features to data analysis methods. Using the framework to do so creates a roadmap towards a detection system which uses intelligent data analysis to detect advanced persistent threats.

7.2. Future research

Cyber attacks are constantly changing requiring defenses to adapt. Currently this means that human intelligence is used to find unique aspects of new attacks which are then used in rules which form the core of signature detection systems. Content based anomaly detection is already subject of research and will be so for the coming years before it can be implemented as a replacement for signature detection methods.

The low success rate of unsupervised learning anomaly detection algorithms in the detection of multistep attack scenarios creates a dependency on human analysis. This means that the user interface is an important

element of the system. Existing research into user interfaces should be applied to the specific case of intrusion detection systems to make this part of the systems more efficient.

The nature of constant change also makes it hard to test new algorithms and systems. Good training and testing datasets are not available and hard to create. For APTs this is especially the case since they are relatively rare compared to more standard malware attacks. The costs of creating training and testing data sets are high but perhaps these can be derived using data from network monitoring systems manned by experts. This is however complicated by privacy issues on such data from production environments. Research should be done to find a better way to create relevant training and testing data sets.

References

[1] M. Van Eeten, J. Bauer and S. Tabatabaie, "Damages from internet security incidents: A framework and toolkot for assessing the economic costs of security breaches.," Delft university of Technology, Delft, 2009.

[2] N. Kshetri, The global cybercrime industry: economic, institutional and strategic perspectives, Springer, 2010.

[3] Symantec, "Symantec Internet Security Threat Report," Symantec, 2011.

[4] GOVCERT.NL, "Stuxnet - een geavanceerde en gerichte aanval," GOVCERT.NL, The Hague, 2011.

[5] Fox-IT, "Interim Report, DigiNotar Certificate Authority breach," Fox-IT Business Unit Cybercrime, Delft, 2011.

[6] C. Tankard, "Persistent threats and how to monitor and deter them," *Network security,* vol. 2011, no. 8, pp. 16-19, 2011.

[7] Fox-IT, "www.Fox-IT.com," 01 11 2011. [Online]. Available: https://www.fox-it.com/nl/nieuws-en-events/nieuws/laatste-nieuws/nieuwsartikel/fox-it-en-tno-werken-aan-systeem-voor-het-detecteren-van-digitale-spionage/192. [Accessed 06 12 2011].

[8] M. Tavallaee, N. Stakhanova and A. A. Ghorbani, "Toward credible evaluation of anomaly-based intrusion-detection methods," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews,* vol. 40, pp. 516-524, 2010.

[9] J. Davis and A. Clarck, "Data preprocessing for anomaly based network intrusion detection: A review," *Computers & Security,* vol. 30, pp. 353-375, 2011.

[10] T. Pietraszek and A. Tanner, "Data mining and machine learning - Towards reducing false positives in intrusion detection.," *Information Security Technical Report,* vol. 10, pp. 169-183, 2005.

[11] H. Kai, Y. Liu and L. Zhou, "Reducing false negatives in intelligent intrusion detection decision response system," Sanya, 2012.

[12] C. Tucker, S. Furnell, B. Ghita and P. Brooke, "A new taxonomy for comparing intrusion detection systems," *Internet Research,* vol. 17, pp. 88-98, 2007.

[13] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Department of Computer Engineering, Chalmers University of Technology, Goteborg, 2000.

[14] S. Singh and S. Silakari, "A survey of cyber attack detection systems," *International Journal of Computer Science and Network Security,* vol. 9, pp. 1-10, 2009.

[15] T. Rakes, J. Deane and L. Rees, "IT security planning under uncertainty for high-impact events," *Omega,* vol. 40, pp. 79-88, 2012.

[16] V. Igure and R. Williams, "Taxonomies of Attacks and Vulnerabilities in Computer Systems," *IEEE Communications Surveys & Tutorials,* vol. 10, no. 1, pp. 6-19, 2008.

[17] GOVCERT.NL, "Cybersecuritybeeld Nederland," GOVCERT.nl, Den Haag, 2012.

[18] P. Ning, Y. Cui and D. Reeves, "Constructing attack scenarios through correlation of intrusion alerts," in *Proceedings of the 9th ACM conference on Computer and communications security*, New York, 2002.

[19] S. Cheung, U. Lindqvist and M. Fong, "Modeling Multistep Cyber Attacks for Scenario Recognition," in *Proceedings of the DARPA Information Survivability Conference and Exposition*, Washington, 2003.

[20] S. Yang, A. Stotz, J. Holsopple, M. Sudit and M. Kuhl, "High level information fusion for tracking and projection of multistage cyber attacks," *Information Fusion,* vol. 10, pp. 107-121, 2009.

[21] GOVCERT.NL, "Nationaal Trendrapport Cybercrime en Digitale Veiligheid 2010," GOVCERT.NL, The Hague, 2011.

[22] TNO, "Dreigingen: aanvalsstappen," TNO, Delft, 2011 [internal document].

[23] S. Hansman and R. Hunt, "A taxonomy of network and computer attacks," *Computers & Security,* vol. 24, no. 1, pp. 31-43, 2005.

[24] S. Dua and X. Du, Data mining and machine learning in cybersecurity, Taylor & Francis Group, 2011.

[25] T. Lappas and K. Pelechrinis, "Data mining techniques for (network) intrusion detection systems," *Riverside, CA: University of California,* 2007.

[26] C. Tsai, Y. Hsu and W. Lin, "Intrusion detection by machine learning: A review," *Expert Systems with Applications,* vol. 36, pp. 11994-12000, 2009.

[27] S. Mukkamala and A. Sung., "A Comparative Study of Techniques for Intrusion Detection," in *15th IEEE International Conference on Tools with Artificial Intelligence*, Sacramento, 2003.

[28] S. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Applied Soft Computing,* vol. 10, pp. 1-35, 2010.

[29] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez and E. Vazquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security,* vol. 28, pp. 18-28, 2009.

[30] Trend Micro, "Deep Discovery 3.0," Trend Micro Incorporated, [Online]. Available: http://www.trendmicro.com/us/enterprise/security-risk-management/deep-discovery/index.html. [Accessed 2 3 2012].

[31] HBGary, "Solutions :: Digital DNA," HBGary, [Online]. Available: http://www.hbgary.com/core-technology. [Accessed 8 3 2012].

[32] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *2010 IEEE Symposium on Security and Privacy*, Oakland, 2010.

[33] C. Zhou, C. Leckie and S. Karunasekera, "A survey of coordinated attacks an collaborative intrusion detection," *Computers & Security,* vol. 29, pp. 124-140, 2010.

[34] Sourcefire, "Snort," Sourcefire, [Online]. Available: http://www.snort.org/. [Accessed 19 3 2012].

[35] C. Iheagwara, "The effect of intrusion detection management methods on the return on investment," *Computers & Security,* vol. 23, pp. 213-228, 2004.

[36] C. Iheagwara, A. Blyth, T. Kevin and D. Kinn, "Cost effective management frameworks: the impact of IDS deployment technique on threat mitigation," *Information and Software Technology,* vol. 46, pp. 651-664, 2004.

[37] W. Lee, W. Fan, S. Stolfo and M. Miller, "Cost-Sensitive Modeling for Intrusion Detection," in *Machine Learning and Data Mining for Computer Security*, London, Springer, 2006, pp. 125-136.

[38] PricewaterhouseCoopers, Information Security, a strategic guide for Business, San Jose: PricewaterhouseCoopers Global Technology Centre, 2003.

[39] H. Debar, F. Telecom, D. Curry, Guardian, B. Feinstein and I. SecureWorks, "The Intrusion Detection Message Exchange Format (IDMEF)," The IETF Trust, Network Working Group, 2007.

[40] C. Kruegel and T. Toth, "Using Decision Trees to Improve Signature-Based Intrusion Detection," in *RAID 2003*, Pittsburgh, 2003.

[41] A. Webb, Statistical Pattern Recognition, Wiley, 2002.

[42] Symantec, "Symantec Adds IBM Neural Network Boot Detection Technology to Norton Antivirus," 4 8 1999. [Online]. Available: http://www.symantec.com/about/news/release/article.jsp?prid=19990804_02. [Accessed 19 3 2012].

[43] Y. Tang, Y.-Q. Zhang, N. Chawla and S. Krasser, "SVMs Modeling for Highly Imbalanced Classification," *Journal of Latex class files,* vol. 1, no. 11, pp. 1-9, 2002.

[44] W. Tylman, "Anomaly-Based Intrusion Detection Using Bayesian Networks," in *Third International Conference on Dependability of Computer Systems*, 2008.

[45] Cisco, "Securing Your Network with the Cisco Centri Firewall," Cisco Systems, Inc., [Online]. Available: http://www.cisco.com/en/US/products/sw/secursw/ps743/products_user_guide_chapter09186a008007f2ef.html#xtocid14. [Accessed 19 3 2012].

[46] M. Berthold and D. J. Hand, Intelligent Data Analysis, an introduction, Springer, 1999.

[47] V. Kumar, Y. Kim, J. Srivastava, Z.-L. Zhang, M. Shaneck, V. Chandola, H. Liu, C. Chango, S. Gyorgy, E. Eilertson and P. Desikan, "Situational awareness analysis tools for aiding discovery of security events and patterns," Air Force Research Laboratoy, Rome, New York, 2005.

Appendix 1

Reference network with normal traffic streams

The green stream is traffic from the public network to the public services and vice versa. This traffic is initiated by clients in the public network. The purple streams are normal browsing or work related traffic. The orange stream depicts information streams between the clients and internal services like mail, files, applications etc.. Traffic like updates and remote installations is also contained in this stream. The red stream might be traffic like mail coming from a spam-filter/mail-proxy or form a VPN-(file)server. This traffic is of a single type and always initiated between specific host/clients. The blue line is traffic going to the internet from the internal services. This might for example be e-mails. This traffic is mostly predictable as it comes from specific, known sources.

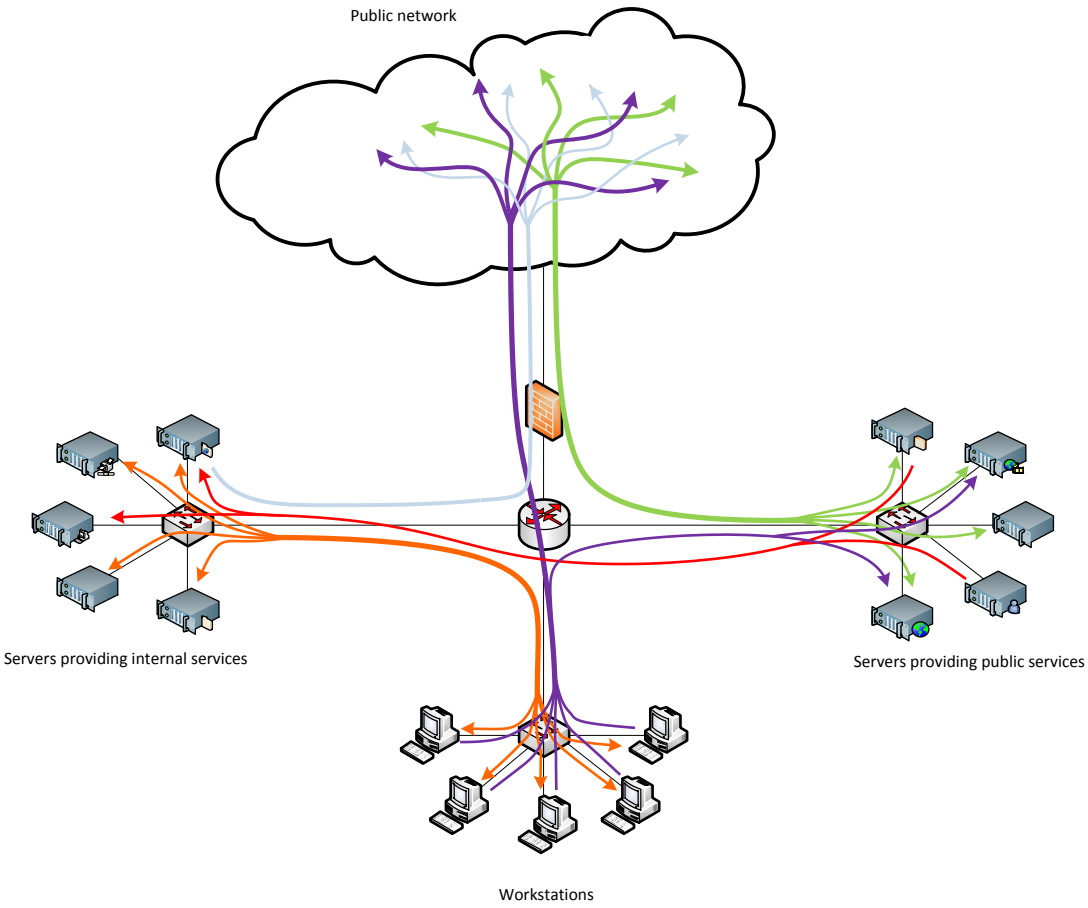

Figure 12; Example network with normal traffic streams.

Appendix 2

Figure 13 describes, on a general level, the kind of information to be found in the framework. Figure 14 shows a high level use example in which the attack example from chapter two is placed in the framework with the relevant locations, methods and techniques.

Framework with general descriptions

Figure 13; general description of block content in the framework

Framework with the attack example description

steps

Attack steps	Attack methods	Attack features	Detection locations	Detection methods	Analysis methods	Business aspects
1 External reconnaissance	- Check domain names - Browse corporate website - Searc h social media	- Uncharacteristic site usage	- Internet - Servers	- Server Log parsing - NIDS	- Use sequences - Time based Traffic statistics	- Security awareness of employees
2 Gaining access	- Phishing mails - Infected websites - Malware - Security flaws in services - Tainted USB sticks	- Attachments - common traits of phishing mails - Difference in filesizes	- Servers - Clients - Traffic	- NIDS - HIDS - Sandboxing	- Rule based analysis - Command sequence analysis - File statistics	- Anomaly detection needed - Wide coverage of measures wished
3 Internal reconnaissance *(can be simultaneously with phase 4)*	- Malware - Rootkit - Botnet	- Change network usage - execution of commands which provide network information	- Servers - Clients - Traffic	- NIDS - HIDS - Log analysis	- Statistical analysis - Command sequence analysis	- Privacy - HIDS clients might conlict - HIDS costly to develop
4 Expanding access	- Malware - Exploits - Rootkit	- Different traffic patterns - Traffic content	- Servers - Clients - Traffic	- NIDS	- Statistical analysis - Time based analysis - Content rules	- Privacy issues - Choice of locations
5 Gathering information	- Malware - Botnet	- File usage pattern changes - Services use changes - quantity of files accessed increases	- Servers - Clients - Traffic	- HIDS - NIDS - Log analysis	- Rule based analysis - Use Statistics	- Privacy - HIDS client issues
6 Information extraction	- Malware - Botnet	- Volume of traffic	- Traffic	- NIDS	- Statistical analysis	- Worst case scenario
7 Control of information leaks *(Active during phases 2 through 6)*	- Botnet	- Internet usage (segments, sites) - Traffic contents	- Traffic	- NIDS	- Statistical analysis - Rule based analysis	- Privacy
8 Erasing tracks *(Active during all phases)*	- Botnet	- Changes to system - Changes in file sizes	- Servers - Clients	- HIDS	- Rule based analysis - File statistics	

Impact

Aspects

Figure 14; Example attack aspects in the framework

Appendix 3

Description of data mining methods used for intrusion detection

Many different algorithms for data analysis exist. Researchers have studied many single intelligent data analysis methods to detect anomalies. Some of these are tailored for specific problems and others are meant as overall improvements. All these different methods use standard machine learning algorithms which can be categorized in four main categories; classification, clustering, statistical and other methods [8] [25]. These general categories are discussed below with attention to applications in research.

Classification methods

Classification methods classifying events by a model which is constructed from labeled dataset. This is called supervised learning. After the training period data that does not comply with what is considered normal can be classified as anomalous.

A common input data type for classification methods is network traffic (TCP/IP) data in packet or flow form. Frequency of system calls is another type of input data often used with classification algorithms.

Neural Networks (NN)

Neural Network models are based upon the working of biological brains. The interconnected network of neurons of a brain is simulated by computational units which have inputs, outputs and activation functions. The activation function strikes when the inputs achieve a pre-set level causing the neuron to send a signal over its output to another neuron's input. Training of such a network changes weights on the inputs in the activation functions. Inputs to the network are linked to the appropriate output states through hidden layers. During training labeled data is used to determine the right weights for connections in the neural network such that the network is capable of predicting with a high certainty level [41].

Disadvantages of neural networks are the complexity of the models with a larger number of neurons. The result is that neural networks are harder to understand and very hard to visualize. This characteristic does enable neural networks to modeling complex relations and to detect and incorporate latent variables into the model. Larger neural networks are also very fault tolerant and resistant to noisy input data. Errors in the input data have therefore less impact on the classification accuracy. On the other hand are the computational costs increasingly higher for larger neural networks. And to get good results large datasets are needed for training.

Neural networks can be considered a family of methods which are similar in structure but differ in implementation. For example, the choice of activation function for a neuron determines strongly how the network functions and has a large impact on its quality for a specific task. The consequence is that neural networks are very versatile and flexible but they are complex and their structure is often hard to understand.

When implementing neural networks an appropriate training method has to be chosen with a sufficient amount of training data. This data should be of ample size and diversity, otherwise one might run the risk of over fitting the model on the training data. Often used training methods are back propagation, radial base functions or self organizing maps.

In intrusion detection literature neural networks are used for traffic analysis based on signatures to determine the anomalous state of traffic [24]. Known commercial applications are for example the use by IBM to detect boot sector viruses this technology was later acquired and used by Symantec in Norton AntiVirus [42].

Support Vector Machines (SVM)

Support Vector Machines divide known data into two classes by calculating a hyperplane that has the largest distance to both classes and classify new cases by checking the position relative to the hyperplane. A hyperplane is n-1 dimensional "plane" which separates two classes. Multiple classes are not directly

supported since SVM's are binary models. One way to identify multiple classes is to group classes and make subdivisions by a newly calculated hyperplane in such groups. Dimensions in SVM's are dataset features like source address, destination address and destination port in TCP/IP data.

Support vector machines are fast and do not require large datasets for training. They only require the features used in the model for analysis resulting in a smaller amount of data to process since unused data can be discarded. It is possible to calculate the hyperplanes dynamically by creating a training model that adapts to environmental changes [24]. The chances of overfitting the training data are small with SVM. A disadvantage of SVM is that they slow down dramatically when they are used for large sample sizes. In general are SVMs are considered to be more accurate than NN but when the distance to the hyperplane gets smaller the accuracy of classification also decreases [24].

Support Vector Machines are used in signature detection on traffic data and command sequences [27]. For anomaly detection SVM's are used supervised as well as unsupervised to detect anomalies in traffic data and system calls. Commercially SVM's are used by McAfee Labs for their TrustedSource platform for classification and behavior analysis [43].

Bayesian Networks (BN)

A Bayesian network is an inference based network of probability functions which give the probability of events. These events are not considered separately but the interdependency of events upon each other is important. This interdependency is implied by the network structure. The underlying statistics are based upon Bayes' theorem which in this case is used to state the probability of an event to be occurring given the current observation. The Bayesian network uses observed events to calculate the probability of an unobserved event. The unobserved event can be an attack or an expected normal event.

The causal relations implied in a BN are easy to understand. The method however is dependent on the existence of the dependency amongst the events in the BN. The method is also sensitive to missing values requiring the missing data to be estimated.

Bayesian networks have been used to do feature selection for signature detection thus enhancing rule based detection [24]. BN also have been used for anomaly detection [44].

Decision Trees (DT)

Decision trees are tree like structures in which branches are the different outcomes of IF-THEN statement like comparisons. The leaves of a decision tree represent the different classes of the model. The tree is constructed starting at the root by identifying the feature with the highest information gain, or in other words; the feature which most strongly distinguishes different classes.

Decision trees create a very understandable model of classification and are easy to use. Trees can include probabilities which results in a classification with a probability of this classification given the input. Decision trees are simple to interpret and use. It is however difficult to find the most optimal tree from a large dataset. An optimal tree can therefore not be guaranteed, this can result in computing overhead [24].

Decision trees are often used to enhance signature detection models. This is done either by providing a decision tree for selection of rules to check the input against or by using DT to classify input data. Combinations of decision trees called random forests are used in anomaly detection. By creating trees based on bootstrap data and a number of randomly selected features many different trees are calculated. A new event is input for all the trees and then the summation of the output of all trees forms a weighted decision on the classification of the input event. Outliers, not strongly classified events, are calculated and these form possible anomalies. The construction of random trees results in a high variance and therefore a lower accuracy.

Decision trees have been used to improve the performance of SNORT by creating a decision tree to reduce the number of rules that need to be checked for any given input [40]. The Cisco Centri firewall uses a similar approach to analyze traffic [45].

k-Nearest Neighbor (KNN)

The idea behind the k-Nearest Neighbor algorithm is that events within a class have similar characteristics which separate them from other classes. The algorithm classifies new cases by finding a preset number (k) of neighbors in a pre-labeled data set and then determining the most common class among those neighbors.

The main advantage of KNN is its relatively simple implementation and the easy interpretation of the model. The basic model also does not require any parameter training, only a labeled dataset. However, noisy data reduces the accuracy of the method and creation of a labeled dataset is time consuming and creates a static classification. The method also requires more storage space since it keeps requiring the labeled dataset unlike models that only need a trained dataset for learning. Even though KNN does not require parameter training, it does require a choice of the number of neighbors. An odd number of neighbors is advised to prevent a draw [41].

Detecting anomalies is more difficult with KNN since the basic method implies that each case is a member of a class. Using distance thresholds makes it possible to identify anomalies. Another method to determine outliers is to calculate the density of a class and use this metric to determine if a case is an outlier and thus a anomaly. Finding the right calculation method is an important success factor for anomaly detection by KNN.

k-nearest neighbor is used for signature as well as anomaly detection. For anomaly detection a threshold needs to be defined based on distance or density metrics. It has been used to create a selection method for other methods and also to analyze system call data [24]. In the last case it proved to be performing better with short fragments than other methods.

Hidden Markov Models (HMM)

Hidden Markov Models are based upon the Markov property which states that the only the current state determines the future. But opposite from Markov models are the states not necessarily observed directly. The probability of an event happening is then calculated using the probability of the observations.

If the steps in sequences are truly independent of each other than HMM models are accurate, if the events are not independent of each other than the accuracy drops. Large training sets are also required to train HMM with a sufficient accuracy.

The sequential nature of HMM makes it very useful for time series data like network flows [24]. In such cases observed packet flow data is used to indicate an attack or some other sort of network activity. Identification of attacks is performed by calculating the chances for known signatures. An advantage of HMM is that not the entire sequences is necessary. HMM can also be used for anomaly detection if a threshold is implemented and normal traffic is identified by signatures.

Kalman filter (KF)

The Kalman filter is the opposite of HMM in the sense that it assumes that future states are dependent on past states. KF estimates the future state and uses the prediction error to improve the estimation for the next future state. An advantage of this approach is that it does not need a database of previous state data. This approach also makes that the model can be deployed in a online state. The basic KF is a linear model, the Extended Kalman Filter and the Unscented Kalman Filter are non-linear versions of KF to handle non-linear models.

The KF can provide accurate estimates of future states, even with noisy data. The only requirement is that the noise has a normal distribution. Noise filtering should be applied if this is not the case, or when this cannot be guaranteed that the noise has a normal distribution.

Kalman Filters are primarily used to predict the normal state, anomalies are deviations of the expected normal state. The method is mostly used to filter “normal” traffic so that only the anomalous data remains which can then be used as input for other analysis methods [24].

Clustering methods

Clustering methods are based on the idea that clusters share similar identifiers and that data in a dataset can be divided into clusters. These clusters are based on distance or density metrics. The result is that detected behavior which falls in a cluster with identified malicious behavior is probably also malicious in nature. Clustering differs from classification in that classification is aimed at putting a sample into a class (or cluster) whereas clustering tries to identify clustering among samples, or in other words, tries to identify classes.

Shared Nearest Neighbor (SNN)

The Shared Nearest Neighbor method is similar to the classification method KNN except that it creates classes instead of classifying a single event. Its basic premise is still that each event shares characteristics with its neighbors. SNN however calculates the neighbors of each event based on the distance to other events or the density of events. A cluster is a group of events which share neighbors according to the distance or density metric. An advantage of the SNN method is that it is insensitive to noisy data or to the shape of clusters.

k-Means

The k-means method creates a pre-defined number (k) clusters. The standard algorithm tries to minimize the mean distance to a center point for a cluster. To find the optimal solution with this method is considered NP-hard and may take a long time. There are however algorithms that are fast in finding a good solution, even though it might not be the optimal one. Important for success is the right choice of k at the start since this cannot be changed during the algorithm.

The k-means clustering method is sensitive to shape. This is the result of minimizing the mean distance by relocating the center position with each optimization round. The shape of points cloud has a strong influence in the final center position since it has influence on the distance between points in the cloud. Choosing the right distance calculation method can help in reducing the influence of this shape sensitivity.

In literature k-means clustering is mostly used to profile traffic [24]. By clustering normal types of traffic the outliers can be defined as anomalies.

Subspace

The subspace algorithm is an algorithm that takes only a subset of the dimensions of the input data to create clusters. The hypothesis is that not all dimensions are relevant and that those dimensions which are irrelevant only act as noise in methods like k-means that do not use a subset of dimensions [24].

Self-Organizing Maps (SOM)

Self-organizing maps is a special form of neural network. It is a visualization technique in which, simply put, the output neurons of the neural network represent the pixels in a grid. Neutrons that active on similar characteristics, since their input data share characteristics, are spaced close to each other in this grid showing the clustering in the input data. The self-organizing nature of SOM is due to the learning process in which the neurons are learned to activate on similar input as their neighbors. The result is that relationships in the data are preserved which is not the case in NN. This behavior is useful in intrusion detection when intrusions have distinguishable differences from normal behavior [46].

General use of clustering methods

In practice clustering is used to obtain profiles of the input data [47] [25]. In all cases the main challenges are to find the right amount of clusters and to obtain a satisfying goodness of fit.

Statistical methods

Another category of commonly used methods for intrusion detection are statistical methods. Statistical methods also form the basis for most of the methods in the other categories. Methods in this category are general methods like regression analysis methods, multivariate data analysis or principal component analysis [29].

The methods in this category are numerous and an in depth discussion falls outside the scope of this thesis. The most common methods are part of methods discussed in the other categories.

Supporting methods

Boosting is often used to improve the results gained by individual methods. One example of boosting which is often used is taking bootstraps from the training data to create multiple models which are then used in combination and the result is obtained by a majority vote from the models [24].

Fuzzy logic is often applied when there is uncertainty about the linguistic interpretation of attribute classification. With fuzzy logic boundaries between groups or classifiers are not binary but gradual. Fuzzy logic is for example applied to association rules to allow for variances in normal behavior. Fuzzy logic can also be applied within the above mentioned classification and clustering methods creating methods like fuzzy neural networks or fuzzy decision trees [46]. Fuzzy logic differs from interval in the sense that intervals still use binary boundaries whereas fuzzy logic allows for overlap between states. Fuzzy logic also allows for better text to numerical comparison [46].