UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Anomaly-based Detection of Lateral Movement in a Microsoft Windows Environment

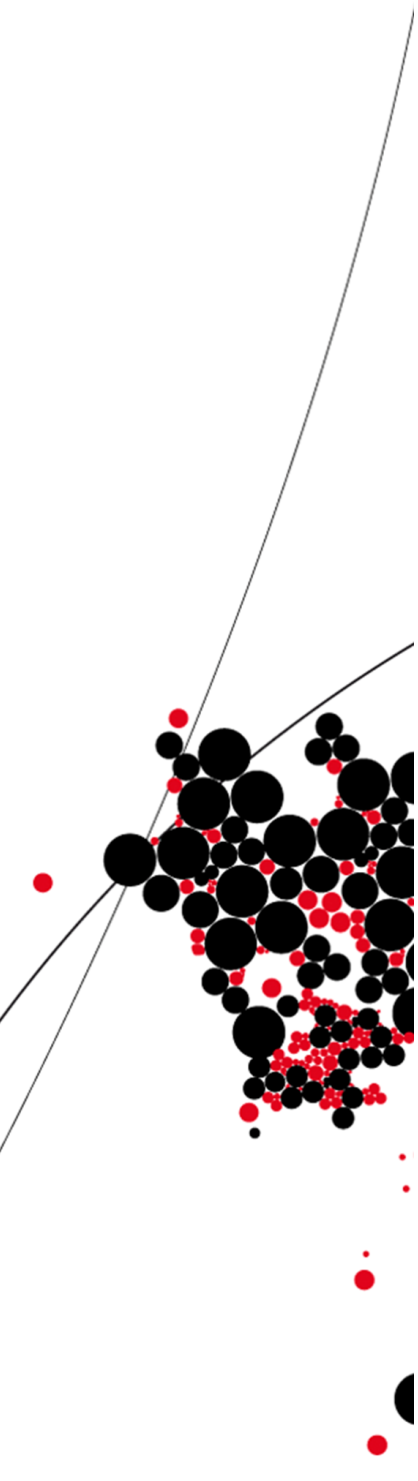
Using the Windows security event log for detecting lateral movement techniques executed by a professional red team.

by

Mart Meijerink

Thesis to obtain the degree of
Master of Science in Computer Science
with a specialisation in Cyber Security

January 2019



Supervisors:
Dr. José Jair C. de Santanna
Dr. Anna Sperotto
Angelo Perniola (KPMG)

Faculty of Electrical Engineering
Mathematics & Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

Cyber security is a very important topic for organisations and yearly many thousands of incidents occur. When adversaries intruded the information technology (IT) systems of a targeted organisation in 2017, they managed to avoid detection for a median time of 101 days. This shows that organisations are insufficiently able to detect intruders in their systems.

These long-term intrusions are often executed by sophisticated attackers, typically called advanced persistent threats (APTs). APTs are capable of using a variety of tactics and techniques, one of which is lateral movement. This tactic is the act of adversaries moving from system to system to increase their influence and reach their objectives by penetrating further into their target's IT environment. By failing to detect lateral movement, organisations are exposed to leakage of data and attackers increase their chance of staying persistent, even after detection.

Therefore, the detection techniques concerning lateral movement have been researched, based on which 2 host-based anomaly detection approaches have been implemented for the detection of anomalous logon patterns. As Microsoft Windows is adopted most by enterprise-size companies, detection efforts were focused towards a Microsoft Windows environment. The Windows security event log is chosen to supply the data used for the detection approaches, because related work has shown its potential. Despite that, little research focused on this natively available logging, although, its widespread availability to security monitoring and incident response teams.

To evaluate the implemented anomaly detection approaches, operational data has been gathered from an enterprise-size company. Additionally, a professional red team has been tasked to execute lateral movement in a typical enterprise IT environment. This has been combined into a dataset featuring 58 event logs supplemented with malicious logon records from the attack environment, resulting in a realistic dataset. The anomaly detection approaches have been implemented based on clustering, using HDBSCAN, and a statistical technique, using principal component based classification (PCC).

The results indicate that both approaches are able to identify deviating logons based on the Windows security event log. Clustering achieved a true positive rate (TPR) of 85.63% with an 8.29% false positive rate (FPR). PCC was able to detect less malicious logons with a TPR of 59.81%, however, better performance with respect to the FPR, 4.70%, was achieved as well.

This thesis shows that anomaly detection based on the Windows security event log of an individual system is an effective method for the detection of lateral movement. From the perspective of a security monitoring architecture, the main contribution of this approach is the conclusion that it is possible to distribute part of the detection efforts of a centralised monitoring solution, such as a security information and event management (SIEM) solution, toward the individual workstations in an organisation's environment.

Contents

Chapter 1

Introduction

Most companies in society are aware cyber attacks pose a risk to their organisation and cyber security, therefore, is a very important topic [23]. However, every year numerous headlines appear of businesses, hospitals, and even cities and goverments being the victim of cyber criminals. In March 2018, Wired reported on the SamSam ransomware attack on the city of Atlanta [47]. Unlike other ransomware attacks, the attackers behind SamSam infiltrated the network of their victims first, to obtain information about the network and the target organisation [47, 55]. The information gathered via reconnaissance was then capitalised on by the attackers by encrypting important systems and demanding a ransom "at price points that are both potentially manageable for victim organisations and worthwhile for attackers" [47]. Another example of adversaries infiltrating a company and maintaining persistence in its information technology (IT) systems, is the data breach of hotel chain Marriott [31]. In November 2018, Marriott disclosed adversaries intruded the network and maintained persistence over a 4 year period. The Marriott data breach affected 500 million guests who's data was stolen, including cases where sensitive personal information was leaked, such as passport numbers or financial information [31].

By November 2018, the SamSam group received an estimated $6 million in ransom payments, but had caused over $30 million in damages and losses to their victims [55]. Upon Marriott's disclosure of the data breach, their stock price plummeted 5.6 percent as analysts assessed that Marriott could be liable for fines and settlements up to $200 million [22]. This shows that the impact of intrusions is highly relevant, as well as from a financial as a business continuity perspective.

These examples are not isolated incidents, as reflected by the Online Trust Alliance review on cyber incidents [48], which mentioned 159,700 cyber incidents during the course of 2017. However, 44% of the companies do not have a security operations centre (SOC) to monitor their IT environment [23]. Related to this, FireEye, a security provider specialised in monitoring and defence of IT environments, reported the median time adversaries managed to stay undetected in the IT environments of their victims to be 101 days [37]. Of these intrusions, 72% stayed undetected for at least 1 month and over 28% even more than 1 year. These headlines and surveys regarding cyber security show organisations are insufficiently able to detect intrusions of adversaries in their network.

These long-term intrusions often involve sophisticated attackers, which are capable of advanced tactics and techniques. These adversaries execute structured attacks, during which typically low-privileged systems, such as employees' workstations, are compromised first [52], because detection efforts and preventive control measures are aimed at high-value systems. The prioritisation of detection and prevention efforts is not only due to the relevance of data stored on high-value systems, but also related to the financial and technical impact of collecting event logs from a large base of

distributed endpoints into a centralised security information and event management (SIEM) solution. Although, adversaries are able to compromise the network boundaries in this way, no access to sensitive information is gained as of yet. Therefore, attackers employ lateral movement, which is the act of adversaries moving from system to system to increase their influence and reach their objectives by penetrating further into their target's IT environment. Due to defenders' focus of monitoring efforts and as evidenced by the long periods adversaries are able to evade discovery, it is clear that detection of lateral movement is insufficient and this thesis addresses that fact.

In order to investigate the current efforts of intrusion detection research aimed at the detection of lateral movement, the following research question (RQ) has been devised:

> **RQ1:** *What are the intrusion detection techniques used for detecting lateral movement?*

Previous research focused mainly on host-based data, because attackers often execute lateral movement attacks by leveraging legitimate system features and using valid credentials [13]. Therefore, host data gives defenders a better overall picture of any anomalous actions, as opposed to network traffic. Host-based data consists of a myriad of different log types, however, authentication attempts were a commonly inspected source by related work [3, 24, 28, 34, 51, 52]. These methods mostly collected host-based data for the application of a centralised detection approach in SIEM tools. This introduces a huge burden, because of the vast amount of data involved when incorporating data of low-privilege systems [19]. Based on the findings of this question, it was therefore decided to research 2 anomaly-based detection approaches, which use host-based data for the detection of lateral movement with a special focus towards logon events. Although, this thesis introduces a new technique to apply the implementation of the 2 researched detection methods, the main contribution of this thesis lies not in the specific detection techniques. More important is the application of these techniques based on the data supplied by individual, low-privilege systems, such as workstations, which shows that part of the detection efforts can be distributed.

The investigation of a host-based anomaly detection is focused on Microsoft Windows 10 workstations in enterprise organisations. Microsoft Windows is chosen for its large adoption by businesses world-wide. For example, the Microsoft Windows 10 operating system is adopted by 200 million enterprise users [30]. Additionally, Gartner stated 85% of enterprises started deployment of Windows 10 by the end of 2017 [25]. Therefore, a major impact can be made with improved detection on this platform. Given the chosen anomaly detection approaches, the next research question is aimed at finding the host data supplied by the Microsoft Windows operating system, which could aid detection:

> **RQ2:** *What logging provided by Microsoft Windows systems can be used for detecting advanced lateral movement techniques?*

This research question is aimed to focus specifically on advanced lateral movement techniques that use legitimate, built-in operating system features, instead of exploits to work around installed security barriers. These techniques are usually harder to detect, among others because they follow the normal, intended path of execution, usually also including valid credentials. Research shows that the Windows security event log provides the logging based on which an anomaly detection approach can deviate between benign and malicious behaviour [3, 17, 28, 52]. Especially, due to the availability of logon events. Related work, providing the answers to these research questions, is discussed and explained in detail in chapter 2.

Afterwards, a proof of concept has been build to evaluate the performance of the chosen anomaly detection methods. To investigate whether host-based anomaly detection on individual machines is capable of detecting lateral movement, the last research question is:

RQ3: *Which anomaly detection method better detects a deviation in the Windows security event log?*

In chapter 3 the process is described to build a proof of concept, evaluate it, and supply it with the data necessary for evaluation. For this effort the Windows security event logs from workstations in an enterprise-size company have been gathered. Combined with the data from attacks executed by a professional red team, a realistic dataset has been constructed. Based on an evaluation of the proof of concept, the third research question is answered.

The results of this evaluation show that anomaly detection based on the Windows security event log of individual workstations in an enterprise-size company is able to detect the execution of lateral movement techniques. In chapter 4 these results are presented and discussed. Finally, chapter 5 summarises and concludes on the findings of this thesis.

Chapter 2

Background

As mentioned in chapter 1, adversaries manage to intrude organisations' IT environments and stay undetected for long periods of time. Lateral movement is a tactic commonly employed in such cyber intrusions and this chapter gives an overview of lateral movement in section 2.1. Afterwards, existing research into intrusion detection, specifically in the area of lateral movement, is discussed in section 2.2. Lastly, the Windows domain and Active Directory (AD) are introduced in section 2.3 to describe previous work covering the Windows security event log investigated in this research. This results in the concluding remarks of section 2.4, which answers the first two research questions (**RQ1:** *What are the intrusion detection techniques used for detecting lateral movement?* **RQ2:** *What logging provided by Microsoft Windows systems can be used for detecting advanced lateral movement techniques?*).

2.1 Lateral Movement

This section gives an overview of the adversaries able to execute attacks involving lateral movement in subsection 2.1.1. Followed by a description of the life cycle of a cyber attack in subsection 2.1.2. Lastly, subsection 2.1.3 covers what lateral movement comprises in general and specific techniques to execute lateral movement aimed at avoiding detection.

2.1.1 Advanced Persistent Threats

The threat actors involved in lateral movement typically display high resourcefulness, clear motivation and goals, and strong technical skills. As stated in chapter 1, FireEye reported a global median dwell time of 101 days after the initial compromise before organisations were able to discover a breach of their network [37]. Groups with the skill set to stay undetected and persistent for such a long time are often called advanced persistent threats (APTs). Chen, Desmet, and Huygens [21] introduced four distinguishing characteristics APTs display in their approach:

1. specific targets and clear objectives;
2. highly organised and well-resourced attackers;
3. a long-term campaign with repeated attempts;
4. stealthy and evasive attack techniques.

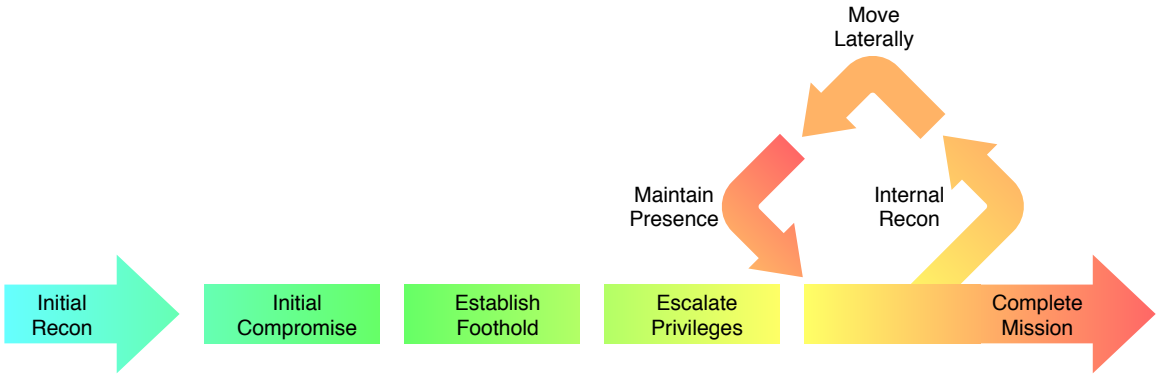


Figure 2.1: The attack life cycle, as defined by Mandiant [36].

These characteristics differentiate intrusion attempts by APTs from ordinary intrusion attempts. Besides that, they also give insight as to why and how these adversaries are able to intrude their target's network and stay undetected. An attack of an APT group is typically aimed against a specific organisation or industry, incorporating extensive reconnaissance on the target. Combined with an organised and well-resourced approach, this explains their persistence displayed by repeated attempts and the use of stealthy and evasive attack techniques to complete the mission. APTs are thus fierce opponents, which are difficult to counter. However, these threat actors do not constantly reinvent themselves and their intrusion attempts can therefore be modelled. The next section describes the steps that an intrusion consists of.

2.1.2 Attack Life Cycle

Cyber security is a cat-and-mouse game between attackers and defenders, in which the defenders have long thought to be at a continues disadvantage. Until Hutchins, Cloppert, and Amin [27] defined the notion of the cyber kill chain model, stating that attackers have "no inherent advantage over defenders". This led to the development of attack models to aid and structure detection efforts, by modelling the structured approach of APTs.

As described in chapter 1, lateral movement is executed inside the IT environment of the target and the cyber kill chain model [27] addresses the fact that defenders are able to detect traces of adversaries when being attacked. The Mandiant attack life cycle [36] views the different stages of a cyber attack in a slightly different way. Unlike the chained phases of the cyber kill chain, Mandiant defined cyber attacks as a life cycle, including a circular component, as visualised in Figure 2.1. Adversaries enter the IT environment of their target during the initial compromise stage and establish a foothold. From this point forward, attackers evade the control measures protecting an organisation's network boundaries. However, they can still be detected by the monitoring efforts of an organisation. Adversaries employ tactics, such as privilege escalation [11] and discovery [4], to expand their sphere of influence over the compromised machines. The increased privileges and obtained knowledge about the intruded environment enable attackers to spread through the network by executing lateral movement [5], searching for the high-value systems holding the information of their mission's objectives.

So far, a description of APTs, the threat actors typically executing lateral movement, has been given and attack models, such as the Mandiant attack life cycle, have been introduced. Tactics have been mentioned, which generically describe the capabilities of adversaries to intrude a target's IT

environment. Multiple efforts exist to document the tactics, techniques, and procedures (TTPs) of APTs in a generic fashion. Adversarial Tactics, Techniques & Common Knowledge (ATT&CK) [57] is an example of such a knowledge base. ATT&CK is built on information derived from open source reporting that documents the approaches of specific threat groups and red teaming efforts mimicking those techniques to provide detection and mitigation recommendations [53]. Therefore, ATT&CK is a valuable resource considering tactics, such as lateral movement, and the techniques supporting those tactics. The next section covers lateral movement and techniques to execute lateral movement.

2.1.3 Tactic & Techniques

A tactic represents the reason why an attacker performs an action along the course of an intrusion and is strictly linked to the objective that the attacker wants to achieve. Lateral movement is one of the generic tactics [5] that are commonly employed by APTs during cyber intrusions. Lateral movement provides threat actors with deeper access into a targeted network, granting logical access and control over systems that are considered instrumental for the completion of the mission. In most network intrusions, initially a low-privileged system or user account is compromised via, for example, spear-phishing or a web drive-by [8, 52]. Although attackers now have a foothold in the network, they may still be far from their mission objective. At this point lateral movement enables them to either compromise new systems, containing sensitive data they are interested in, or pivot from previously compromised systems toward new target systems within the organisation's network.

In Figure 2.2 a simplified network setup is shown with the connection path from a) an adversary to b) a compromised workstation. Suppose c) the file share contains confidential files which are an APT's objective. The confidential files are accessible by d) administrator users, but not by the user of the compromised workstation. To gain access to those confidential files, the adversary could e) execute lateral movement as shown in Figure 2.2 to complete its objective. Lateral movement, in this case, consists of obtaining access to the admin workstation, after which the adversary is able to abuse those privileges to access the confidential files at the file share server.

Stealth and evasion are a clear indicator of the skill level which enables advanced adversaries to remain undetected in their targets' environments and this also depends on the techniques used. A generic description is given with the introduction of the tactic lateral movement, which enables adversaries to expand their presence in a network environment to fulfil their objectives. Techniques represents the way an attacker achieves a tactical objective by performing a predetermined action. In the context of lateral movement, a technique describes a specific implementation to actually execute lateral movement within a victim's network.

Lateral movement techniques applicable to Windows, as covered by ATT&CK [5], are listed in Table 2.1. For each of the 15 techniques listed, the type of logging is indicated which is able to assist in detection of that technique. As can be deduced from Table 2.1, lateral movement can be executed in many ways. However, not every technique is equally sophisticated. For example, the remote desktop protocol (RDP) is a well-known service, which attackers can misuse to connect to other systems [12], as evidenced by, among others, the SamSam attackers [47]. Legitimate users could become aware of these connection attempts when logged in, because Windows typically notifies a user when another connection is attempted. Harder to the detect, on the other hand, are techniques which misuse administrative features, such as Windows Management Instrumentation (WMI) [14], as these are easily mistaken for maintenance tasks. Many threat actors have therefore been found to misuse WMI. Another example of a popular adversarial technique is pass-the-hash [10], which "bypasses standard authentication steps (...) moving directly into the portion of the authentication that uses the password hash" [10], making it hard to detect. Focus is placed on the ability to detect these advanced lateral movement techniques, upon the assumption that less sophisticated

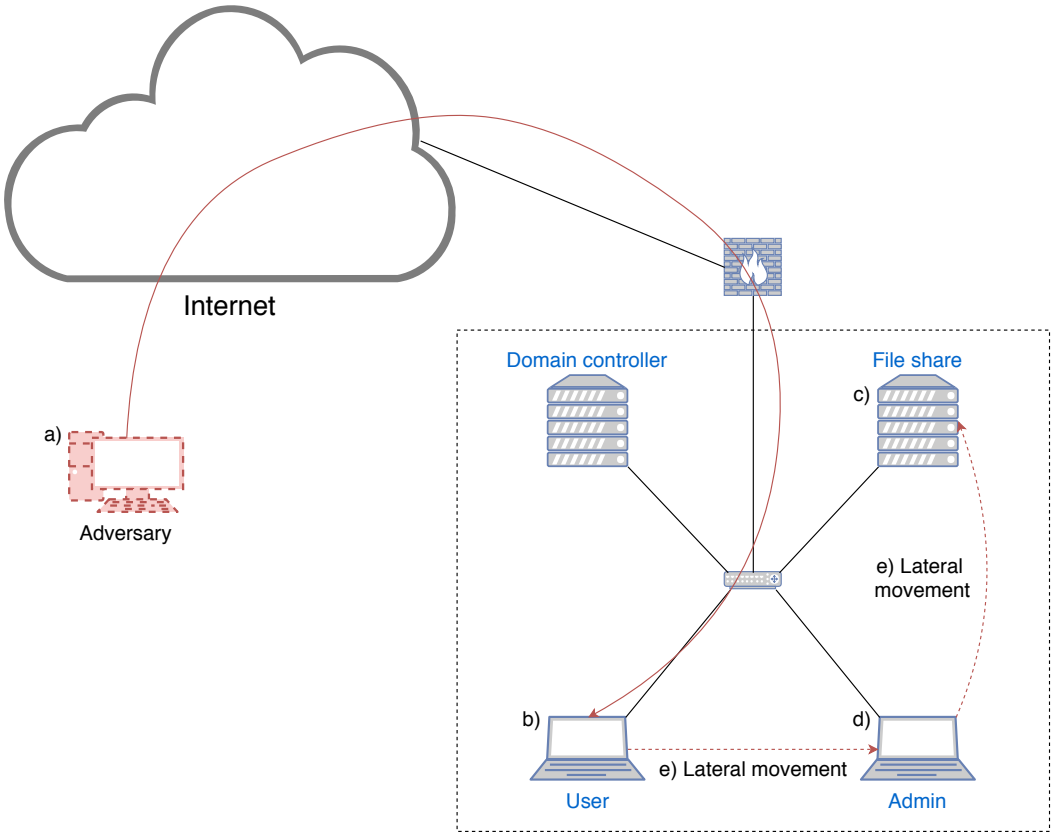


Figure 2.2: Initial compromise of network and lateral movement towards file share.

techniques will be detected by the same solution. In addition, well-known hacking tools, such as Cobalt Strike [6] and Mimikatz [9], have the built-in capability to execute these techniques. The next sections give a detailed overview of these mentioned advanced techniques.

Pass the Hash

The authentication process in Microsoft Windows does not always need direct input from a user. User credentials can also be supplied programmatically by applications [40]. The lateral movement technique pass-the-hash [10] exploits this feature to directly inject a hash of the password in the applicable authentication process. This removes the need for adversaries to obtain access to a user's cleartext password. Thereby, credentials are cached in the Security Accounts Manager (SAM) database [40], so users do not need to constantly supply their logon information. However, credential dumping tools such as Mimikatz [9] are able to retrieve these stored credentials [7], which results in adversaries obtaining password hashes to valid accounts [13].

After capturing a user's password hash and successfully executing pass-the-hash, an adversary is authenticated as that user on the attacked system. At this point the adversary has access to the system as defined by the user's privileges. Detection of pass-the-hash executions is difficult,

because the programmatic process call is not distinctive from a legitimate call to these authentication process [10, 40].

Windows Management Instrumentation

Windows Management Instrumentation (WMI) [46] is an administrative feature for local, as well as remote access. WMI allows administrators of enterprise environments to manage remote computers through automating administrative tasks, but can also distribute management data to other parts of the operating system. All Microsoft Windows desktop and server platforms have WMI installed by default. WMI can be used through programs supplying graphical interfaces, but also called directly via scripts to automate tasks.

Adversaries have been known to create PowerShell scripts, which download and execute remote access tools (RATs) [2]. This technique uses PowerShell, which is a natively available tool in Windows, and runs from memory on the target system leaving no artefacts behind for forensic analysis. By using the features offered by WMI and the deployment of PowerShell scripts remotely, attackers can execute lateral movement in the IT environment of their victims [2, 14].

To create an interactive shell at the target computer, the `Invoke-WmiMethod`-command can be used from a PowerShell at the compromised system. The command, as shown in Listing 2.1, creates a new Windows process with the supplied credentials at the system specified by the `ComputerName`-option. The credentials can be supplied either via a `PSCredential`-object or a plain text username. When a username is supplied, a password prompt will show up. Therefore, the use of a `PSCredential`-object is to be expected for automated scripts. When no credentials are supplied, the WMI method is run as the current user.

Listing 2.1: Creation of an interactive PowerShell using WMI

```
Invoke-WmiMethod -Path Win32_Process -Name create
    -ComputerName <ComputerName>
    -Credential <PSCredential | Username>
    -ArgumentList PowerShell
```

After introducing 1) APTs, the threat actors able to execute lateral movement undetected, 2) the attack life cycle, which models cyber attacks executed by APTs, and 3) the tactic and techniques, to execute lateral movement undetected, the next section covers intrusion detection. Research concerning the detection of lateral movement and other opportunities using anomaly detection are discussed.

Technique / Type of logging	API monitoring	Authentication logs	Binary file metadata	Data loss prevention	DLL monitoring	File monitoring	NetFlow	Network protocol analysis	Packet capture	Process command-line parameters	Process monitoring	Process use of network	Third-party application logs	Windows event logs	Windows error reporting	Windows registry
Application Deployment Software						✓				✓	✓					
Distributed Component Object Model	✓	✓			✓			✓		✓			✓			✓
Exploitation of Remote Services						✓				✓				✓		
Logon Scripts						✓				✓						
Pass the Hash		✓														
Pass the Ticket		✓														
Remote Desktop Protocol		✓					✓			✓						
Remote File Copy						✓	✓	✓	✓	✓	✓					
Remote Services		✓														
Replication Through Removable Media				✓		✓										
Shared Webroot						✓				✓						
Taint Shared Content						✓				✓						
Third-party Software			✓			✓				✓	✓	✓				✓
Windows Admin Shares		✓							✓	✓	✓					
Windows Remote Management		✓				✓	✓		✓	✓						
Total of techniques captured	1	7	1	1	1	9	3	1	2	2	11	4	1	1	1	2

Table 2.1: Lateral movement techniques logging sources

2.2 Intrusion Detection

Intrusion detection systems (IDSs) are either using network or host-based data [15]. Network IDSs gather log information from network traffic. Whereas a host-based IDS collects audit data from endpoints in the IT environment. The detection strategies of IDSs can be divided in two major approaches [16]: signature-based detection or anomaly detection. Signature-based systems are generally strong in detecting known attacks. Artefacts specific to certain techniques, which show up in log sources, are identified and are described in a signature to detect executions of those techniques. As an attack first needs to be described in a signature, however, these systems are unable to detect new, deviating attack techniques.

Anomaly detection IDSs, on the other hand, employ a different strategy. They learn a profile of the normal behaviour under the assumption that no intrusions are present during training. Afterwards, anomaly detection declares unusual, unseen behaviour as malicious when it does not fit the learned profile. Therefore, anomaly detection systems do hold the promise to detect unknown attacks as well and they could work in a more automated fashion. However, this strategy does also have its drawbacks as anomaly detection systems suffer from false positives, because anomalies in the data could also have been caused by new legitimate behaviour or noise [1]. Other drawbacks include the initial learning period before deployment and the tendency of anomaly detectors to be more resource intensive [16].

The remainder of this section discusses research into anomaly detection in subsection 2.2.1, describing the challenges to take into account and different anomaly detection methods. Intrusion detection research aimed at lateral movement detection is covered in subsection 2.2.2. The limitations of these approaches and log sources are described and the topics covered in this section, answers the first research question.

2.2.1 Anomaly Detection

Despite the challenges of anomaly detection mentioned, anomaly-based detection is chosen because of its possibility to detect new techniques, which adversaries rapidly keep developing. This section, therefore, first discusses the challenges related to anomaly detection. Afterwards clustering and statistical techniques, used to implement anomaly detection, are described.

Among the challenges of anomaly detection, as identified by Ahmed, Mahmood, and Hu [1], are the lack of universally applicable techniques, noisy data, a lack of publicly available datasets, and drift of the legitimate behaviour of users which has to be accounted for. Besides the challenges to implement an anomaly detection system, also the evaluation of anomaly detection research has been criticised. A comprehensive study by Tavallee, Stakhanova, and Ghorbani [56] surveyed research on anomaly-based intrusion detection, assessing the validity and reliability of the experiments. Three dimensions, relating to 1) the employed data, 2) the evaluation of the method's performance, and 3) the performed experiments, have been identified to review the research in scope. A detailed description of these reviewed dimensions is given:

1) Employed data Concerning the data used, problems in reporting existed among others due to failing to give a definition of what constituted as an anomaly [49] or a proper description of normalisation of the data or the employed features.

2) Evaluation of Method With respect to the performance evaluation it was mentioned that no universal metrics were used, however, the receiver operator curve (ROC), true positive rate (TPR), and false positive rate (FPR) are most commonly seen. Another important point

concerned that different types of attack should be evaluated and reported on separately. This thesis adheres to this practice as only lateral movement is evaluated.

3) Performed Experiments Reproducibility of an experiment is an important factor for experimental science [49], however, Tavallee, Stakhanova, and Ghorbani [56] found that the documentation of experimental process often lacked in the reviewed papers.

As mentioned, anomaly detection holds the promise of automatically detecting new types of intrusions, but also suffers from drawbacks which pose challenges to the development of such a system. In addition, shortcomings with regards to the evaluation of anomaly detection research have been identified, which impacts the applicability of that research.

Li and Oprea [34] described the design of an analytics framework based on operational security logs. One of the main insights given, is that "hosts in an enterprise network are constrained by company policies and employee job functions, and exhibit more homogeneity than those on the open internet" [34]. Dedicated hosts, utilised by a single user, are monitored by a system that identifies hosts displaying anomalous activity which does not fit the expected behaviour previously learned. This homogeneity does allow to better define the scope as well and develop a more robust detection system. Next, techniques to implement an anomaly detection system are described.

Ahmed, Mahmood, and Hu [1] covered network anomaly detection, analysing among others classification, clustering, and statistical techniques. Similarly, Buczak and Guven [18] surveyed methods for cyber intrusion detection. Anomalies with respect to lateral movement attacks can be well detected by clustering and statistical techniques [1, 18]. These attacks are targeted at hosts inside an IT environment and do not generate huge amounts of traffic, because one of the objectives is to stay undetected. Therefore, these intrusions are rare in relation to the total amount of traffic. However, traffic patterns of intrusions deviate statistically from benign traffic patterns and clustering or statistical techniques are able to detect these deviations.

The main advantage for clustering methods is that they can work unsupervised without explicit descriptions of what constitutes as an anomaly [18]. According to Ahmed, Mahmood, and Hu [1], clustering techniques work with three key assumptions which every research states about anomalies:

1. Data which does not fit clusters constructed from benign data is considered anomalous, in case of density-based clustering algorithms this assumption translates into considering noise as anomalies.

2. In clusters containing both benign and anomalous data points, anomalous points lie further from the nearest cluster centroid, therefore, a distance-based threshold can separate benign from anomalous data.

3. When data is clustered in multiple clusters, anomalous points tend to belong to smaller and sparser clusters. Separation of benign and anomalous clusters should therefore happen based on size, density, or a combination of size and density.

Statistical techniques, on the other hand, determine the expected value for a new data point based on training data. When the observed value deviates significantly from the expected value, that data point is considered an anomaly. A significant deviation typically follows the 3 sigma rule of thumb [1], which describes that a deviation of more than 3 standard deviations from the mean is typically an outlier.

A statistical anomaly detection technique can be implemented by employing a principal component analysis (PCA) [1]. The principal components express the variance in the data in a certain direction. The components explaining the most variance can be used to model the data information [50]. At the same time, this reduced the dimensionality of the data without losing any

important, because the components describing the least amount of variation are ignored. This data model, which is also free from any assumption about the statistical distribution of the data, is found to closely model specifically the expected value of normal data instances. Anomalies, however, generally show a large deviation from the expected value, because most of the variance concerning anomalous data points is captured by the dropped principal components, which explain the least variation over the total amount data [32]. This method works under the assumption that the amount of anomalous instances in the training data is negligible [50]. The next section describes intrusion detection techniques aimed at the detection of lateral movement.

2.2.2 Lateral Movement Detection

As described in section 2.1, ATT&CK [57] lists many lateral movement techniques and describes detection and mitigation strategies. The detection sources, as mentioned by ATT&CK, of known lateral movement techniques can be found in Table 2.1. The main difference regarding lateral movement detection in comparison with other intrusion detection efforts, lies in the fact that lateral movement follows an initial compromise of the network. Due to this, lateral movement is typically executed behind the security control measures installed to protect the network boundaries of an organisation. Research into anomaly detection, as described in subsection 2.2.1, mainly works with network data and shows a focus on high privilege systems. This section gives an overview of intrusion detection aimed at the detection of lateral movement.

A lateral movement detection architecture is proposed by Fawaz et al. [24] to propagate host-level monitoring to a global, network-wide view. Detection is based on monitoring inter-process communication at host-level of which the results are collected at clusters to build a host communication graph. The host communication graphs are evaluated globally to identify inter-cluster connections. By monitoring the system calls of processes, Fawaz et al. [24] argue that they are able to better detect lateral movement than by "using timing information or port numbers". While this approach shows monitoring of the network usage of processes is able to indicate lateral movement, no qualification whether or not the movement is malicious can be given [24] and, as indicated by column 'process use of network' in Table 2.1, ATT&CK states inter-process communication is only able to detect 4 out of the 15 listed techniques. In contrast though, Fawaz et al. [24] present their approach as a "first step towards discovering evidence of malicious lateral movement" for a wider scope than indicated by ATT&CK.

According to Table 2.1, lateral movement techniques using the Windows admin shares or WMI could be detected based on inspection of command line parameters and the detection approach of Hendler, Kels, and Rubin [26] looked exactly into this type of data. As discussed by Hendler, Kels, and Rubin [26] and supported by ATT&CK [14] and FireEye [37], PowerShell is increasingly used by adversaries to evade detection, while available on most Windows systems by default and rarely restricted due to its benign use cases. As adversaries are able to use PowerShell to download remote content and execute it from memory, which leaves no artefacts on disk behind, incident response efforts have great difficulty unravelling attackers actions. On top of that, scripts can be obfuscated to further cripple detection efforts. An unsupervised machine learning approach for the detection of malicious commands was employed by Hendler, Kels, and Rubin [26] and an ensemble detector of natural language processing combined with convolutional neural networks, a deep learning approach, was found to produce the best results. Evaluation took place on a labelled dataset, containing about 60,000 clean commands and 6,300 malicious commands. Most malicious commands were obtained via execution of known malicious programs in a sandbox environment, these were used for training the designed detection models. The 471 other malicious commands were contributed by Microsoft security exports and used for evaluation of the approach. According to Hendler, Kels, and Rubin [26],

this results in a realistic scenario with a detector trained by researched malicious behaviour. The detector was able to correctly detect 92% of the malicious commands at a false positive rate of 1%, while for a 0.1% false positive rate the true positive rate stayed high with 89%.

In their focus on lateral movement techniques using valid credentials in enterprise networks, Siadati and Memon [51] did observe, similar to one of the conclusions of Li and Oprea [34], that the logons of users within an organisation are somewhat structured and mostly predictable. Credential-based lateral movement, on the other hand, is very unlikely to adhere to standard logon patterns. Using pattern mining, Siadati and Memon [51] created patterns of logons to determine the typical structure. Based on the patterns found, a classifier was employed to determine whether or not a new logon adheres to the learned patterns based on the user accounts and computers involved. Promising results have been reported of an 82% detection rate with 0.3% false positives on a dataset gathered at a global financial company containing 5 months of data with millions of logons and synthetic attack traces generated based on penetration testing campaigns. The main limitation addressed by the researchers was that the dataset was limited to one company.

Research has been discussed which covers multiple of the logging sources mentioned in Table 2.1, as listed by techniques described in ATT&CK [57]. Process and file monitoring clearly have the ability to cover the detection of most lateral movement techniques. However, many intrusion detection systems focus on network traffic or log sources of critical systems and network devices. As attackers initially compromise non-critical systems and lateral movement is executed over peer-to-peer connections, that scope has its limitations regarding the detection of lateral movement. Research into the detection of lateral movement therefore shows more interest in host-based log sources. As such, Buyukkayhan et al. [19], Fawaz et al. [24], and Wijnands [58] do focus on endpoint monitoring of workstations, looking at process monitoring on a fine-grained level to develop process graphs. These approaches are prone to the specific behaviour of users and the installation image of the workstation under inspection. Confirmed by the conclusion of Wijnands [58]: specific processes running on a system hugely impact this type of monitoring, resulting in false positives. The method of Hendler, Kels, and Rubin [26], focusing on the logging of PowerShell scripts, holds promise, especially for the detection of advanced adversaries who aim to utilise techniques using administrative features such as Windows admin shares and WMI. In addition, Siadati and Memon [51] their approach is interesting as it focuses on logon patterns which used data on logons in a Windows environment. The data collected can be found in the Windows security event log. The administrative lateral movement techniques also create logon patterns and although attackers might try to follow established patterns to evade detection, this will not always be possible for them, especially in the beginning of an intrusion when the necessary credentials are not yet available [51]. The next section introduces the Windows security event log as a source for intrusion detection techniques.

2.3 Windows Security Event Log

This section focuses on the security event log Windows offers and indicates strategies for the detection of anomalies. To better understand the context, first the Active Directory Domain Services are described, which offer the services for administrators to store and manage users and resources in a Windows domain.

Computers in an organisation are subscribed to the network and user accounts are given to employees to log on to those computers. In case of Windows, that network is called the Windows domain. Active Directory (AD) is the directory service offered by Microsoft, which provides the methods to store and disseminate all data related to managing a domain [43]. Role-based access control is implemented via groups, which define the privileges given to the accounts part of that

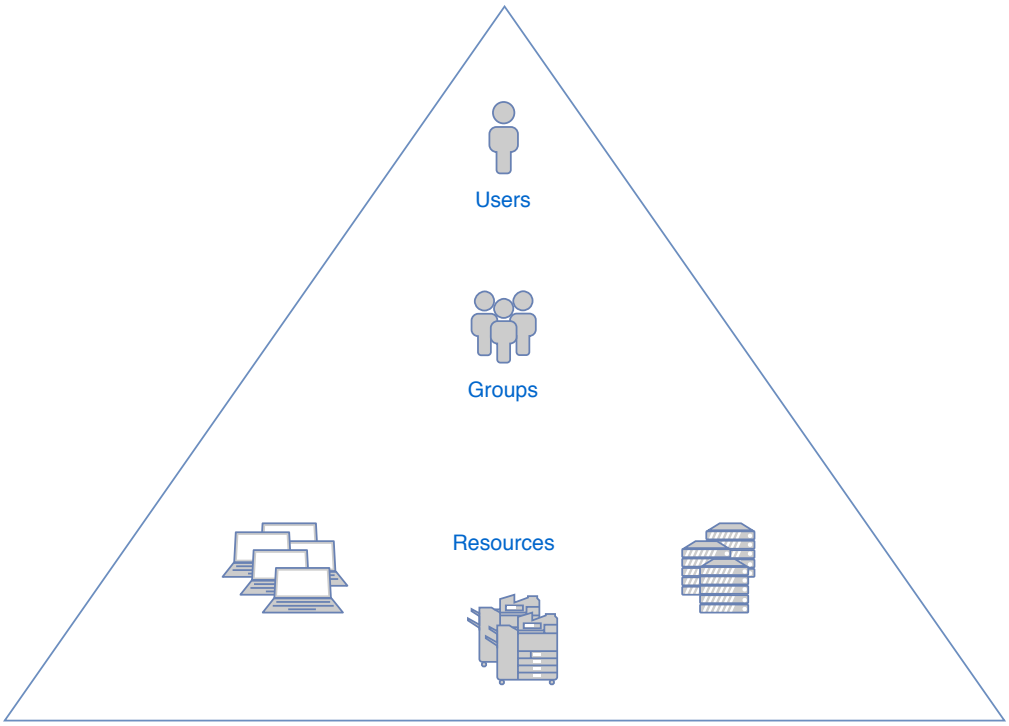


Figure 2.3: Active Directory manages resources and users via role-based access control defined in groups.

group. In AD the information about accounts and groups is stored, as visualised in Figure 2.3. Two main types of accounts exist in a Windows domain. User accounts to allow people to log on to the domain and computer accounts to manage resources. User accounts and computer accounts are administered in the same way and are part of groups to manage their privileges. The domain controller as shown in Figure 2.2 is where Active Directory resides [39]. The domain controllers are the servers responsible for providing directory information throughout the domain and therefore also validate credentials.

In order to keep track of privilege usage and actions executed by users, the Windows operating system offers event logging which features three standard types of event logs: being 1) the application log, 2) the system log, and 3) the security log. Application event records contain diagnostic logging information about installed applications and the logging message and structure depend on the source application. The system log also registers diagnostic events, however, those event records are related to the machine. Events regarding networking and other communication protocols are to be expected, as well as logging concerning machine policies. Both application and system events contain mostly a textual description of the diagnostic event record.

The Windows security event log [44], however, gathers security related events regarding logons of accounts, creation and privilege usage of processes, registration of security related processes, and also system restarts. The events logged by Windows are defined in an audit policy, which can be enforced globally inside a Windows domain network by defining the audit policy for different groups of machines. These events have predictable attributes based on the event type and depending on

the active audit policy [44]. Moreover, most attributes are categorical labels, for example defining the specific type of a logon or the privilege used by a process. The remainder of this section covers methods using the Windows security events for threat detection purposes.

Public institutions such as the Japanese Computer Emergency Response Team Coordination Centre (JPCERT/CC) and its European counterpart, CERT-EU, have researched the impact of cyber attacks in Microsoft Windows environments. JPCERT investigated almost 50 typical tools attackers have been known to use in support of various tactics, among others for lateral movement [28]. Tools able to execute pass-the-hash [10] attacks or misuse WMI [14] are both covered. The research focused on traces execution of these tools left behind as evidence in event log records. Logs have been scrutinised before and after execution of the tools to infer the changes introduced into the event logs, but also registry entries have been inspected. An overview per tool has been compiled, detailing which evidence is introduced in a specific event log [29]. The analysis results are a valuable resource in incident detection and investigations. As many tools do not show evidence of execution with the default Windows logging settings, JPCERT concluded on the importance of developing an elaborate enough audit policy. CERT-EU [52] similarly inspected the influence on log records in the case of lateral movement techniques, such as pass-the-hash and pass-the-ticket. Collecting the event logs from the domain controllers is deemed most important, while the collection of events from workstations of administrators and other high privilege accounts is to be considered. CERT-EU advises to monitor important account groups, with the 'Domain Administrators' being the most important. Other accounts mentioned are service accounts, emergency accounts, and business critical accounts. Event types of interest include the event registered when a domain controller validates credentials, event ID 4776, and logon events, event ID 4624. Monitoring could be set up rule-based and generic detection rules are offered. Focus should be on the source of an account logon, this could be the workstation or network address logged in the event record. Taking these fields into account, a logon of an admin account from a workstation or network address other than its regular, registered workstation or IP address should be triggered upon.

A manager in cyber incident response at KPMG explained the use of Windows event logs to detect lateral movement in forensic investigations. The Windows event logs of the systems in the forensic scope are gathered and the logon events are inspected. Logons from one machine to the other are chained and any unexpected events, especially connections returning to a previously found host in the chain, are flagged. The connections between different machines which create a loop back to an earlier seen host, might indicate actions from an adversary creating persistence in the network after moving laterally or the extraction of valuable information, as visualised in Figure 2.4. Anomalies do not only show up when a logon is initiated from an unexpected location, but also when an account logon occurs on an unexpected system. Most accounts in an enterprise environment are expected to be used onto a pre-defined set of systems. Any deviations from this set possibly indicate attacks and these deviations are reflected in the Windows event log.

So far, industry interests with a focus on incident response have been discussed. As shown in subsection 2.2.1, academic research into anomaly detection suffers from the availability of publicly open datasets and tends to have a focus towards network-based detection. As such, research specifically investigating the Windows security event log is limited, but efforts exist which focus on clustering security event records to deduce patterns. Basagoiti et al. [17] concluded, based on the Windows security event logs of four different domain controllers, that different servers show different patterns based on clustering series of frequent events. Records have been analysed by grouping them in series based on the event identifier. Interruptions in theses series by other type of events have been counted and found to correctly identify 15 out of the 16 servers' series after using k-means clustering. The events under consideration were network logons and logoffs, special privilege assignment, and privileged object operations. The research only considered domain controllers which gathered

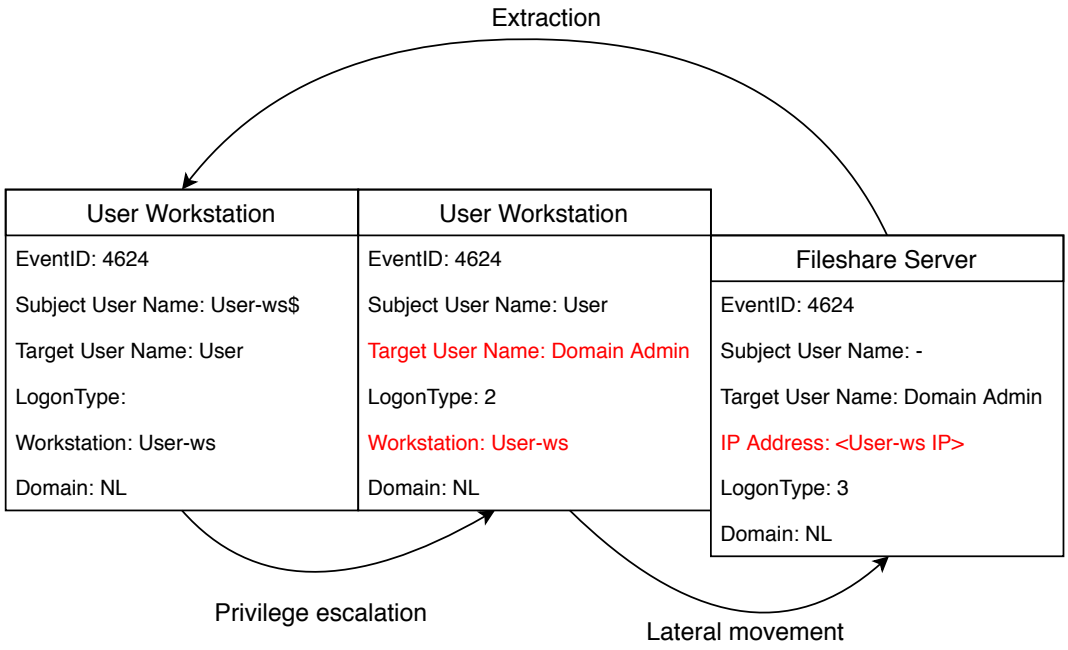


Figure 2.4: Deducing lateral movement through correlation of Windows security logon event records.

events related to the domain and its Active Directory (AD). Besides using the event identifier of an event and number of distinct users, only features regarding the series of events were captured. Selected features defined among others the total number of security event records, the total number of different event types, and other statistics regarding the frequency and percentages of the events. As no further attributes apart from the event identifier and user were inspected, this approach differs hugely as described by incident responders.

Focusing on the Windows logon events as well, Asanger and Hutchison [3] did inspect the attributes of records. Similar to Basagoiti et al. [17], the collected events mainly stem from domain controllers and they used unsupervised anomaly detection based on a global k-Nearest Neighbour approach for which the importance of normalisation and proper pre-processing is stressed [3]. The lack of workstation logs and member servers is considered to be a limitation of the research, which is also supported given the scope of the machines advised to be considered by CERT-EU [52] and KPMG's incident response manager. One of the consequences of this limitation is that account logon event records are mainly noise, caused by group policy updates. Interactive logons at a domain controller are very rare. Logon records are only meaningful on the machine at which the account actually logs on, the actual workstations. The collected event records have been aggregated into multiple data views which consider, for example, the records on a per user basis, records of different workstations aggregated per user, or the users per IP address. These views give insight into different perspectives. Multiple workstations used in attempts to authenticate the same user are suspicious in an enterprise network when every employee has his own computer. The same goes for authentication of different user accounts from the same IP address, which could indicate a password guessing attack. As anomaly detection was unsupervised, outliers have been presented to security operators of the managed services environment to request feedback. It was confirmed that the SIEM solution in place

did not detect the anomalies and continuous feedback helped to improve the process significantly. False positives were reduced by achieving better separation of accounts or computers based on their privileges, respectively, the type of computer.

This review of the research conducted into Windows security event logging indicates that events can be used to identify cyber attacks, such as lateral movement. Clustering techniques have been used to deduce the source machine based on event patterns, but also anomalies regarding authentication could be detected. Both Basagoiti et al. [17] and Asanger and Hutchison [3] mention event logs of individual workstations and servers could be of interest to extend the scope of their research. Combining the findings on intrusion detection and the Windows security event logs, the next section answers the first two research questions.

2.4 Concluding Remarks

This chapter introduced lateral movement and described intrusion detection with a special interest towards anomaly detection. Specific methods targeting the detection of lateral movement attacks have been covered and a shift from network-based towards host-based detection can be seen. Which is logical as lateral movement happens after a system inside the IT environment has been compromised. Anomaly detection is able to discern deviating behaviour related to lateral movement in logon patterns.

The detection of lateral movement attacks, therefore, focuses on identifying these patterns or patterns in the network connections between computers. As stated, these connections have a certain degree of predictability in an enterprise network. Deviations from the expected logon patterns need to be detected, as those could indicate lateral movement.

Different methods have been proposed, but clustering and statistical methods have shown promising results. Clustering currently has a focus towards density-based methods, which perform stronger than classical clustering algorithms based on distance measures due to their ability to cope with regions with differing density. Statistical methods calculate the difference between the expected and actual value and declare anomalies when the deviation exceeds a certain threshold. Typically the deviation threshold lies around 3 standard deviations, following the 3 sigma rule of thumb [1].

An audit policy can be activated for the Windows security event log to log the necessary events and attributes to detect lateral movement. Forensic investigations look into the originating location of an account logon to determine any abnormalities and research showed that clustering is able to deviate between the originating servers and detect suspicious authentication attempts and account deviations based on the Windows security event log. Monitoring is able to aid detection of lateral movement based on anomalous logons [10] or WMI commands [14]. The Windows security log is also able to provide this information.

As indicated by the literature [3, 28, 52], to obtain a complete picture of the IT environment, event logs of workstations should be used as well. Although, lateral movement detection is possible with the event logs of the domain controllers, as those together have a fairly complete picture regarding network logons, it does not show everything.

Anomaly detection techniques for the detection of lateral movement could thus be implemented by defining what constitutes as a deviation from the expected logon patterns. These deviations can then be detected using clustering or statistical methods to model the logon patterns. The data in a Windows environment to build these logon patterns could be gathered from the Windows security event log. Logons to a computer are registered in this log and attributes defining the originating location in combination with the account used are able to indicate deviating actions. Therefore, this thesis looks into host-based anomaly detection, using the Windows security event log.

Chapter 3

Methodology

As described in chapter 2, numerous methods have been developed to detect lateral movement in network-based as well as in host-based data. However, adversaries are still able to remain undetected in their targets' networks for many days. In order to counter this trend and be able to better detect attackers, the need for more fine-grained logging is stressed by Buyukkayhan et al. [19] and Lee and Lee [33]. As concluded in section 2.4, host-based intrusion detection holds promise to improve detection techniques, especially in the case of intrusions starting at low privilege workstations. This thesis therefore investigates whether the Windows security event log could be used for detection of lateral movement techniques. In particular, focus lies with those techniques were attackers leverage already available operating system features and administrative tools. Because the huge amounts of log data [19] available at endpoints, anomaly detection is chosen to process the data using machine learning techniques. Following the recommendations of Tavallaee, Stakhanova, and Ghorbani [56] to explicitly state what constitutes as an anomaly, anomalies to be detected by the proposed detection methods of this thesis are defined as:

> *Windows security event log records that deviate from benign records with respect to the credentials, originating location, or privileges involved, because of adversaries executing lateral movement techniques.*

Following the dimensions identified by Tavallaee, Stakhanova, and Ghorbani [56] for the evaluation of anomaly detection research, first the dataset gathered to research lateral movement techniques is described in section 3.1 covering the Windows security events and domain setup used. Next, section 3.2 covers the method employed to answer the research question. The experiment setup to research the effectiveness of host-based anomaly detection is described and the assumptions made and limitations involved are detailed. Lastly, section 3.3 explains the anomaly detection steps employed to handle the security event logs and apply the detection algorithms on the dataset.

3.1 Data

In order to develop a dataset which could be used to answer the research question, Windows security event logs have been gathered. This section describes the Windows security events which can be found in the dataset in subsection 3.1.1. Next, in subsection 3.1.2 the machines from which the logs have been gathered are described. Finally, the implications with respect to the limitations of the developed dataset are discussed in subsection 3.1.3.

Event ID	Description
4610	An authentication package has been loaded by the Local Security Authority
4611	A trusted logon process has been registered with the Local Security Authority
4614	A notification package has been loaded by the Security Account Manager
4616	The system time was changed
4622	A security package has been loaded by the Local Security Authority
4624	An account was successfully logged on
4625	An account failed to log on
4634	An account was logged off
4648	A logon was attempted using explicit credentials
4662	An operation was performed on an object
4670	Permissions on an object were changed
4673	A privileged service was called
4674	An operation was attempted on a privileged object
4688	A new process has been created
4697	A service was installed in the system
4797	An attempt was made to query the existence of a blank password for an account
4798	A user's local group membership was enumerated
4799	A security-enabled local group membership was enumerated
4907	Auditing settings on object were changed
4985	The state of a transaction has changed

Table 3.1: Top 15 and logon related event types present in the dataset.

3.1.1 Windows Security Events

As stated in section 2.3, the Microsoft Windows operating system logs many security related events in the event log [44]. Registered event records are, among others, different types of logons, new processes being created, privileges being requested by processes, but also security related system settings being adjusted. This section describes the type of Windows events logged and the attributes found in a security event record.

Administrators can design an audit policy to define the event types logged in the security log. Not only is it possible to define which types of events to register, but also which attributes of an event are collected in the event data. One specific example is the process creation event. Besides the standard attributes collected, such as the process name and the parent process of the process being created, it is also possible to log the command line options used to create the process.

In general, the security event log contains system data and event data. System data logs the information about the system creating the log record and contains attributes such as the event identifier, the system time, and the computer name. The event data contains the attributes which identify the specific event that occurred. The event data generally contains the user account initiating the logged action and the process executing the action. Usually, also an account is involved on which the action is executed. Examples of event data attributes which are typically specific to an event, are the parent process in case of a process creation event, the type and origin of a logon in case of a logon event, or the type of privilege requested or used in case of an event regarding the use or request of certain privileges.

In total, the dataset features 52 different event types. However, with the 15 least frequent types occurring less than 10 times each in the total dataset, it is easily concluded that not all events are

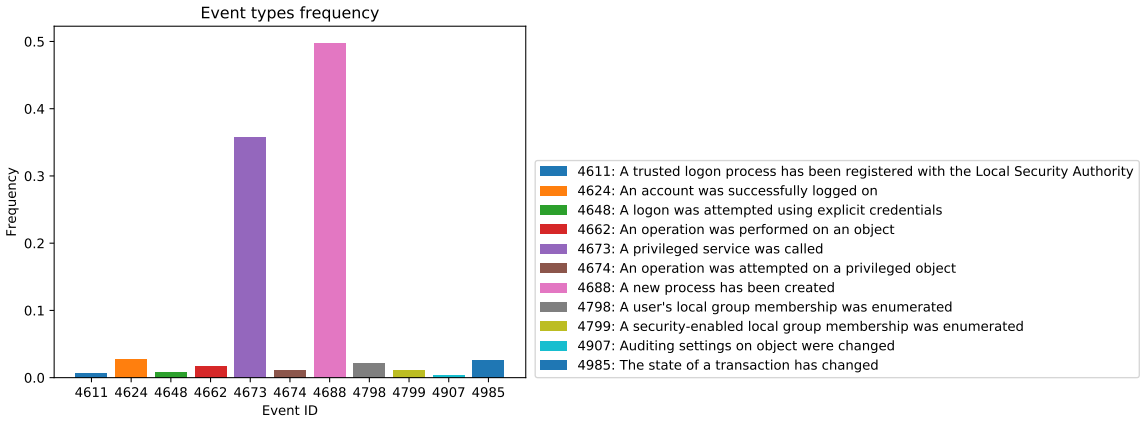


Figure 3.1: Top 10 most frequently occurring benign events.

available at every workstation. These events contain too little information for anomaly detection as no comparison to similar event records can be made. The 15 most occurring Windows security events included in the dataset are shown in Table 3.1, where each event identifier and the corresponding description of the event are listed. Besides the top 15, also less frequent events related to the logon process have been added to Table 3.1, because of the indication given by related work that logon events should be monitored to detect lateral movement, as described in section 2.3. As can be seen in Table 3.1, a variety of events is covered by the audit policy. Quite some events relate to the authentication process. Security events related to process creations and their privileges are well represented. Other events with respect to the adjustment of security related settings or startup and shutdown were less common. This is logical as settings are not adjusted daily, but logons, which also include unlocking a computer, and the usage of processes are typical actions. See Figure 3.1 for an overview of the distribution of the top 10 most frequent events in the dataset. The following section describes a few specific examples of events which could be encountered in the dataset.

Examples

This section gives two concrete examples of logged events. Detailed are a record of a process creation event, one of the most common events in the dataset, and a logon event, of interest as indicated by related work [3, 28, 52].

In Listing 3.1 an example is shown of the event attributes logged in a logon record [41]. The event data contains attributes such as *Logon Type*, *Logon Process*, and *Logon Guid* which are obvious examples of typical attributes specific to a logon event. Most events contain at least information about the user account which initiated the event, therefore, the attributes such as *SubjectUserName* and *TargetUserName* are found in the event data of most events. In the case of the example in Listing 3.1, the domain computer account was logged on locally. The type of logon, 3, specifies that "a user or computer logged on to this computer from the network" [41], which originated from localhost as signified by the *Source Network Address* `::1`. The authentication involved a `Kerberos` authentication as declared by the detailed authentication attributes. This particular example is seen at all workstations in a Windows domain environment and functions as a check-in to keep up to date with the active directory and its policies.

Listing 3.2 shows an example of the event data of a process creation record [42]. The system

Listing 3.1: Event 4624 - Logon

```
An account was successfully logged on.

Subject:
        Security ID:                S-1-0-0
        Account Name:               -
        Account Domain:             -
        Logon ID:                   0x0

Logon Information:
        Logon Type:                 3
        Restricted Admin Mode:      -
        Virtual Account:            No
        Elevated Token:             Yes

Impersonation Level:                Impersonation

New Logon:
        Security ID:                S-1-5-18
        Account Name:               <ComputerName>$
        Account Domain:             <DomainName>
        Logon ID:                   0xB287ED54
        Linked Logon ID:            0x0
        Network Account Name:       -
        Network Account Domain:     -
        Logon GUID:                 {dc7a3e49-e4c9-b4d0-fadf-7283a138c548}

Process Information:
        Process ID:                 0x0
        Process Name:               -

Network Information:
        Workstation Name:           -
        Source Network Address:     ::1
        Source Port:                0

Detailed Authentication Information:
        Logon Process:              Kerberos
        Authentication Package:     Kerberos
        Transited Services:         -
        Package Name (NTLM only):   -
        Key Length:                 0
```

Listing 3.2: Event 4688 - Process Creation

```
A new process has been created.

Creator Subject:
        Security ID:            S-1-5-18
        Account Name:           <ComputerName>$
        Account Domain:         <DomainName>
        Logon ID:               0x3E7

Target Subject:
        Security ID:            S-1-0-0
        Account Name:           -
        Account Domain:         -
        Logon ID:               0x0

Process Information:
        New Process ID:         0x8cc
        New Process Name:       C:\Windows\System32\svchost.exe
        Token Elevation Type:   %%1936
        Mandatory Label:        Mandatory Label\System Mandatory Level
        Creator Process ID:     0x318
        Creator Process Name:   C:\Windows\System32\services.exe
        Process Command Line:
```

attributes are independent of the type of event as opposed to the event data. As explained, the event data, however, contains attributes specific to an event type. Process creation events contain among others the *Creator Process Name* and *Token Elevation Type*, which indicates the privileges with which the process is created. As seen in the example of Listing 3.2, the computer account has tasked the `services` process to start a new instance of `svchost`.

Given the specification of the events in the dataset and the attributes available, the next section describes the origin and amount of security event logs gathered.

3.1.2 Log Sources

At this point the Windows security events in the dataset have been described and concrete examples have been shown. As one event log corresponds to one specific machine which gathered the events, an overview of the different sources follows. Two main sources can be distinguished in the dataset: operational logs, which originated from personal workstations in an enterprise company, and an attack environment, set up to gather lateral movement traces executed by a professional red team. In the next sections the different sources are described which have been summarised in Table 3.2.

Operational Logs

The dataset consists of 36 distinct workstation logs, collected from 1 department of an enterprise organisation. All workstations are connected to the Windows domain of the organisation and the users of the workstations had different roles in the department, ranging from more executing, techni-

Source	Machines	Logs	Records/Log	Total
Operational security logs				
All events	36	60	∼30,900	1,851,818
Logon events (4624)	34	58	∼880	51,030
Attack environment				
All events	1	1	34,881	34,881
Logon events (4624)	1	1	651	651
Total	37	61	-	1,886,699

Table 3.2: Amount of records gathered from the different type of machines

cal roles to managerial roles. Each log contained on average 31,000 events, which roughly translates to a period of 1 to 3 weeks depending on how intensively a particular workstation had been used. From the same set of workstations, another 24 security event logs were available to recollect at least 3 weeks later, to obtain a new sample from the same machines. Making the total dataset contain 60 logs, counting over 1.85 million records. As described, process creations and privilege use were the most frequent events, however, special attention has been payed to the logon events given related works [3, 28, 52], as indicated earlier.

Attack Environment Logs

An environment has been developed containing a Windows domain setup. The Windows domain consisted of four machines. Two servers, one domain controller managing the domain and one server offering file sharing services, were installed and supplemented with two workstations, one intended for a non-privileged user and one intended for an administrator user. Two domain accounts have been created reflecting the described situation. Besides the non-privileged user and the admin user account, a domain administrator account was prepared with full privileges in the domain. The non-privileged user could access the *Business files* network folder, shown in Figure 3.2, at the file share server. The administrator had unrestricted access to the file share server and was able to log on remotely. All administrator accounts in the attack environment had access to the *Administrator files* folder at the file share. An audit policy responsible for logging security related events, incorporating at least the same type of events as the collected workstation logs, was configured.

After setting up the Windows domain environment, red teaming professionals of KPMG have been asked to attack the environment. The machines set up in the attack environment are displayed in Figure 3.2 indicating the starting point for the red teamers in red: the *User* workstation. The starting point represents a system in the network where attackers have established a foothold after the initial compromise with the intent to move laterally within the target environment. The following scenario has been used to define the goal for the red team:

1. The goal of the attack was to reach a folder at the file share which was only accessible to admin users, visualised in Figure 3.2 by the *Administrator files* network folder.

2. The non-privileged user account was considered compromised, credentials to the user account and network access to the user's workstation were provided, as shown in Figure 3.2 by the red workstation *User*.

3. Tools in the support of lateral movement should be administrative Windows tools.

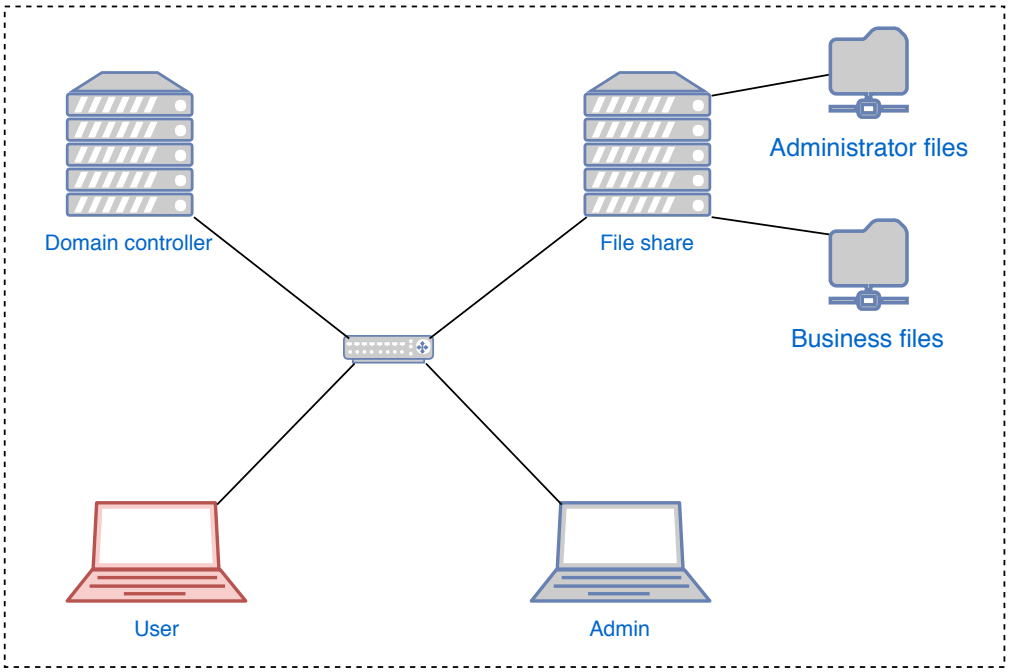


Figure 3.2: Attack environment setup.

Within these boundaries the red teamers were free to try and choose any technique which they deemed effective based on their expertise. As multiple attackers have contributed in this experiment, multiple techniques regarding discovery [4], the actual lateral movement [5], and possibly privilege escalation [11] have been executed, following different attack paths. No focus on specific threat actors or adversarial techniques was added. This ensures any positive results will be able to more generically detect intrusions in a Windows domain environment. As these red teamers use methodologies such as discussed in Mitre's ATT&CK framework [57], which have been described based on research on APTs, the attack techniques display sophistication which is roughly comparable to APTs. From the attack environment only the event log of the user workstation has been collected and the event records have been extracted during the time frames the red teamers have been active.

Limitations related to the collection and compilation of the dataset as described, are addressed in the next section.

3.1.3 Assumptions and Limitations

A limitation this dataset suffers from is that the audit policy of the operational logs could not be changed due to the corporate environment setup. However, the need to collect a sufficiently large dataset for validation was deemed more important than the influence of a more granular audit policy at this stage. Additionally, users were at the time not aware their security event logs would be requested afterwards, therefore, no explicit effort has been done to perfectly follow any guidelines or policy, because of the knowledge that the event log would be eventually collected. Based on a successful approach, future work could always research the influence of a different audit policy in more detail. This research therefore focuses on a more elemental step: whether or not detection

of lateral movement is possible in the first place by using anomaly detection on security event log records of individual machines. The audit policy of the attack environment was more elaborate, such that initial suggestions might be identified of specific events and attributes which might hold promise for future work.

Another point of attention is the different approach different attackers could take. This concern has been mainly addressed by asking multiple red teaming professionals to execute lateral movement in the environment. Within the given scenario multiple approaches have been executed by these professionals to gather a variety of attack techniques, as described. These techniques have been based on known APT behaviour, such as described in ATT&CK [57]. Although new, unknown techniques will always arise, research on zero-days indicates an attack path seldom consists of only zero-days [54]. This finding also indicates that attackers do not constantly reinvent themselves, as also indicated by attack models such as the Mandiant attack life cycle [36]. Combined with the application of an anomaly detection strategy, it is assumed that the proposed solution is able to discover new lateral movement attack techniques as long as execution of such a new technique is still recorded in the Windows security logon event.

Lastly, the event logs only stem from one company and different companies could feature a different network topology and differences in usage of the Windows domain. It is assumed the anomaly detection approach is able to learn the patterns given a certain topology, however, this can only be confirmed when data of multiple companies is available. Next, the method to research and evaluate lateral movement detection on this dataset are described.

3.2 Method

Given the dataset as described in the previous section, this section covers the method followed to answer the main research question. As stated in section 2.3, related work on the Windows security event log inspected the use of anomaly detection using the event logs of the domain controllers [3, 17], but ignored workstation logs which offer a huge potential for lateral movement detection. The method followed by this thesis to investigate host-based anomaly detection is divided in three steps:

1. A sample is drawn from the malicious records and adapted to remove biases from the workstation log under inspection.

2. The thresholds are inspected on an event log basis, to determine the optimal threshold for the dataset.

3. The anomaly detection approach is run over the complete dataset with a random amount of malicious records to inspect the overall performance.

Detection of anomalies is inspired by the work of Asanger and Hutchison [3], but the validation differs as no security operators are involved. The remainder of this section describes the process to evaluate the performance and covers its limitations. However, first the evaluation measures to express the performance are described.

3.2.1 Evaluation Measures

The feasibility of the approach is evaluated based on the true positive rate (TPR) and false positive rate (FPR). These measures express the ability to correctly detect malicious records and the rate of falsely indicated suspicious records. On a record basis the classification of a record can respectively be a true positive (TP), true negative (TN), false positive (FP), or a false negative (FN). A true

positive is a malicious record correctly classified as such. Opposite, a false positive is a benign record which has been incorrectly classified as malicious. Similarly, negative classifications are a prediction of benign actions. The TPR, as expressed in Equation 3.1, is the fraction of correctly classified malicious records with respect to all the malicious records in the dataset. The true positive rate (TPR) therefore indicates the detection capability of the solution.

$$TPR = \frac{tp}{tp + fn} \tag{3.1}$$

The false positive rate (FPR) is the amount of benign records incorrectly classified as malicious in relation to the total population of benign records. In Equation 3.2 the FPR is formulated.

$$FPR = \frac{fp}{fp + tn} \tag{3.2}$$

According to Axelsson [16], the false positive rate is the most important measure for the effectiveness of an intrusion detection method. While a high detection rate is favourable as an organisation wants to be able to detect as many anomalies as possible, this comes at a cost. Because an anomaly detector's performance operates with a trade-off between the TPR and FPR; an increased detection rate brings along a higher false positive rate (FPR) as well. As argued by Axelsson [16], this will eventually result in analysts ignoring the alarms of the system. Therefore, it is important to keep the FPR low enough, such that the impact of false positive alarms in terms of the required effort of security analysts is justified in relation to the amount of correctly detected malicious actions.

A good way to visualise this trade-off is by means of a receiver operator curve (ROC). The ROC plots the TPR against the FPR to indicate the ability to detect attacks given the rate of false alarms to be expected for a system. The area under the ROC is a performance measure of a machine learning algorithm independent of the chosen threshold. The bigger the area under the curve (AUC), the higher the TPRs achieved at lower FPRs. The ROC is therefore a helpful tool in finding well performing thresholds, based on the requirements of users.

To ensure results can be generalised, within the discussed limitations, the TPR and FPR are reported after cross-validation over the different logs with a different amount of malicious records, as is explained in the next section. The next section also covers the establishment of optimal thresholds based on the inspection of the area under the curve of the ROCs.

3.2.2 Evaluation Process

The security event log of every workstation in the dataset is evaluated using a host-centric approach. The evaluation of a machine's event log is then executed on a record basis, grouped per event type. The three steps that constitute the development and evaluation process are highlighted in this section. First, the adaptation of malicious records to the evaluated event log is discussed. Next, the inspection of the ROCs to find optimal thresholds is detailed. Finally, the cross-validation over the different available logs is described.

In order to investigate how to apply anomaly detection using the security events logged by Windows, event records of the attack environment's user workstation have been sampled and mixed with the records of the operational logs of individual workstations. The attack records have been adapted to mimic the destination event log as to remove any biases. This allowed for labelling which records in the dataset were benign or malicious. The adaptation of malicious records taken from the attack environment covers the substitution of attributes such as the domain name, username, and security identifier with the values of the user account and computer from which the workstation event log is collected. Two type of accounts are involved: the computer account and the domain

user account of the employee's workstation. Besides the accounts, also the workstation name has to be substituted. After this adaptation step, the malicious records no longer displayed any biases due to being collected from a different environment. The adaptation was possible because malicious records were identical to the benign records taken from the security log, apart from the attributes, as identified in section 2.3, which might indicate malicious actions.

After adapting the records, the test set of the inspected event log is sampled with malicious event records, similar to Hendler, Kels, and Rubin [26]. The training set is populated only with assumed benign records from the operational logs, following a typical anomaly detection setup. The test set features 50% benign events and 50% malicious events when being evaluated to build the ROCs. The ROCs are constructed by taking 5 draws of the malicious records to cross-validate with 5 different test sets for each event log. This results in an average ROC to limit the influence of outlying curves. The mean ROC is calculated per event log before an average ROC per anomaly detection method is constructed. Based on the individual mean ROCs, the optimal threshold is selected based on the maximum difference between the TPR and FPR.

With the selected threshold, the performance of each method is evaluated by inspecting the average TPR and FPR based on a test set containing a randomly sized sample of malicious actions. Influences due to a fixed percentage of malicious records are therefore countered by this approach. The next section covers limitations of this method and makes assumptions taken explicit.

Assumptions and Limitations

As mentioned, the operational logs are assumed to only contain benign records. The systems are part of a managed network that includes host and network based security controls. As no security incident has been detected during the period when logs have been collected, an assumption has been made to consider no incident has effectively occurred. It has to be noted, however, that this assumption is no guarantee that anomalies are not present in the benign samples.

Most of the limitations are mentioned in subsection 3.1.3, as they are specific to the dataset. Concerning the method, it is assumed that its applicability and performance are limited when an organisation does not supply dedicated workstations to its personnel. In that case, employees could switch workstations between days which would probably make it harder to establish a standard pattern from which lateral movement will be a deviation. Although a proper separation between departments could prove to be enough, this cannot be tested as the dataset is comprised of workstation logs assigned to a specific employee. The described method is not able to overcome this limitation with the current dataset.

Another limitation of the described method is that all events are grouped by event type before evaluation. Unlike Basagoiti et al. [17], any patterns between records of different events can therefore not be discerned and anomalies in the context of the relations between different event types cannot be detected. However, related to Asanger and Hutchison [3], the event attributes of each record are inspected more closely and records are evaluated per event type to detect any deviations between the activities represented by each record. This is the aim of the tested anomaly detection methods.

Now that the process to test an anomaly detection solution is explained, using the dataset as described in section 3.1, the actual application of machine learning used to develop an anomaly detector is introduced in the next section.

3.3 Anomaly Detection Approach

Given the dataset and the method to research the data, this section describes the approach to actually apply the anomaly detection on the data. Pre-processing, feature selection, and the specific

	Subject			Target				
Type	*Domain*	*User*	*User Sid*	*Domain*	*User*	*User Sid*	*Workstation*	*Ip*
0	-	-	S-1-0-0	NT AUTHORITY	SYSTEM	S-1-5-18	-	-
2	DOMAIN	<ComputerName>$	S-1-5-18	Window Manager	DWM-1	S-1-5-90-0-1	-	-
2	DOMAIN	<ComputerName>$	S-1-5-18	Font Driver Host	UMFD-3	S-1-5-96-0-3	-	-
3	-	-	S-1-0-0	DOMAIN.LOCAL	<ComputerName>$	S-1-5-18	-	::1
5	DOMAIN	<ComputerName>$	S-1-5-18	NT AUTHORITY	SYSTEM	S-1-5-18	-	-
5	DOMAIN	<ComputerName>$	S-1-5-18	NT AUTHORITY	LOCAL SERVICE	S-1-5-18	-	-

Table 3.3: Examples of logon events which were filtered in pre-processing.

machine learning algorithms are explained which are applied for each machine individually on a per event type basis.

3.3.1 Pre-processing

Multiple steps have been taken to sanitise the raw input of the Windows security event logs and be able to easily handle the data. The first steps were related to parsing the raw security logs, handling empty fields, filtering noise, encoding the categorical data, and normalising the data.

The raw security logs, as described in section 3.1, have been parsed with the open source tool *plaso* [35] to direct the output to a storage file for each security log extracted from the machines in the dataset. This storage file has in turn been parsed into a `JSON`-formatted file, using the *psort* module of plaso, containing the individual event records sorted by their timestamp. The attributes of each record have been extracted and labelled to the workstation which logged the events to keep track of the origins of the event log.

Independent of the actions executed on a Windows machine, certain accounts are always active. These could be attributed to, for example, system accounts such as the Desktop Window Manager [45], as these accounts also generate logons or create processes. System accounts are local to a computer and run as a service. Records of these events are unrelated to lateral movement as these accounts have specific tasks, as long as these accounts are not allowed to connect to other computers. The Desktop Window Manager, for example, is responsible for desktop composition. Event records related to these accounts can be found in an event log no matter whether benign or malicious behaviour is happening. Therefore, these event records are considered noise and have been filtered out accordingly. Besides noise, however, also event records which could be either benign or malicious have been filtered, as they were indistinguishable. Examples of such records include the logon of the local system user, which can be seen when a user switches to administrator privileges. Whether an adversary or the legitimate user executes such an action cannot be deduced as a log record only shows the local computer account logging on to the local system account. In Table 3.3 examples of records which have been filtered are shown and from the security identifiers involved, it can be deduced all events are local to the computer.

As most attributes in the Windows security event log are categorical and thus contain text, labels, or other types of data, encoding of those attributes was necessary before the machine learning algorithms could handle those attributes. This also enabled the last pre-processing step, normalising the values so the range and scale are comparable. The next step in the process is the selection of features, explained in the following section.

Feature	Description
AuthenticationPackageName	Indicates the authentication type. This could, for example, be Kerberos or NTLM.
ElevatedToken	Whether or not the logon has elevated administrator privileges.
ImpersonationLevel	The level of impersonation the logon is allowed to. Most common is to have impersonation privileges allowing to impersonate the client on the local system, but not remote systems.
IpAddress	The source IP address where the logon originated from.
IpPort	The source IP port of the remote system.
KeyLength	The key length in bits for an NTLM authenticated logon.
LmPackageName	The package name of an NTLM authenticated logon.
LogonProcessName	The trusted logon process registered with the local security authority handling the logon.
LogonType	The type of logon performed.
ProcessName	The the executable of the process which initiated the logon.
RestrictedAdminMode	Whether or not the logon was running in restricted admin mode.
SubjectDomainName	The domain of the account initiating the logon.
SubjectUserName	The username of the account initiating the logon.
SubjectUserSid	The security identifier of the account initiating the logon.
TargetDomainName	The domain the account being logged onto resides in.
TargetOutboundDomainName	The domain for the user account being used for remote connections, in case of logon type 9.
TargetOutboundUserName	The username of the account being used for remote connections.
TargetUserName	The username of the account being logged onto.
TargetUserSid	The security identifier of the account being logged onto.
WorkstationName	The computer name of the workstation from where the logon originated.

Table 3.4: Selected features for logon events (4624).

3.3.2 Feature Selection

Feature selection is an important step to ensure only meaningful attributes are taken into account. Based on the literature as described in section 2.3, it can be deduced that especially the source location in case of an account logon is important to detect anomalies. Therefore, fields such as *IpAddress* and *Workstation* in combination with the subject and target account should be important to deduce differences between benign and malicious records.

The following initial feature selection approach has been devised. Attributes with a constant value have been removed as no additional information about a record is contained in these attributes. Examples of constant attributes are the name of event log and event channel, because only the security event log is used. Identifying attributes, such as the *EventRecordID* which is an identifier given when the record is saved, have also been removed to counter overfitting. The selected attributes for logon events are listed in Table 3.4.

The next section covers the actual application of the machine learning algorithms given the pre-processing and feature selection described before.

3.3.3 Machine Learning Algorithms

This section describes the machine learning algorithms used to classify the event records in the dataset. Two different methods are described: a clustering method, HDBSCAN, and a statistical method based on PCA.

Clustering Method

Hierarchical Density-Based Spatial Clustering for Applications with Noise (HDBSCAN) [20] is a popular clustering method for anomaly detection [19]. A cluster is a non-empty set composed of all points which are *density-connected*. The concept density-connected applies to the neighbourhood of a point, which has a radius of ϵ. The ϵ-neighbourhood needs to contain at least m points for a point to be considered a *core object*. When a point is not a core object, it is called noise. All core objects which are in each others neighbourhood or transitively connected via another core object's neighbourhood are density-connected and thus form a cluster. The hierarchical component of HDBSCAN refers to the selection of clusters based on their persistence which is selected based on the value of ϵ.

The HDBSCAN Python implementation of McInnes and Healy [38] takes the minimum cluster size m, which corresponds to the m points which are minimally needed before a cluster is formed, and the minimum sample size s. The minimum sample size influences how conservatively the clustering is and corresponds to the number of points that are considered in a point's neighbourhood. In other words, when a point's ϵ-neighbourhood contains at least the minimum number of samples s and the resulting cluster contains at least m points in total, a cluster is formed.

As described in subsection 2.2.1, Ahmed, Mahmood, and Hu [1] identified three common assumptions made with regards to clustering methods for anomaly detection to mark samples as malicious, being 1) samples classified as noise, 2) the sample distance from the centroid, and 3) clusters containing a small amount of samples. This thesis focused on the second assumption by using the outlier score of the HDBSCAN implementation as developed by McInnes and Healy [38]. The outlier score is a measure of the probability that a sample is classified to another cluster when newly evaluated samples could alter the balance of clusters. The higher the outlier score, the more a sample differs from the rest of the cluster.

Principal Component based Classification

A well-known dimensionality reduction approach is principal component analysis (PCA) and first used as an anomaly detection approach for intrusion detection by Shyu et al. [50] in 2003. This is a statistical method for anomaly detection where classification is based on the principal components. PCA transforms high-dimensional data onto a new set of uncorrelated axes. In case of zero-mean, scaled data, these axes express the principal components, which have the property that they explain "the direction of maximum variance remaining in the data, given the variance already accounted for in the preceding components" [32]. Usually, most variance is explained by the first few principal components.

This fact is used by Lakhina, Crovella, and Diot [32]. A classifier based on the principal components employs PCA over the training data, retaining the principal components explaining most of the variance in the data. Based on this PCA-model, the data is fitted and outliers are removed. The remaining samples are fitted again. The current model forms the basis for the classification prediction model. Using this PCA-model, a prediction is made for data points in the test set. This is done by transforming the test data to the dimensions of the PCA-model and retrieving the expected value, which is calculated by reversing the calculation. The outcome is a value based on

the trained PCA-model, called the inverse score. The difference between the actual test data point and the inverse score is called the residual value. The residual value is the amount of unexplained information about the data point and this remaining information is explained in the principal components containing the lowest variance about the data. The residual value is introduced as these principal components were not available in the trained model. As explained by Lakhina, Crovella, and Diot [32], this method separates the data into a *modelled* and *residual* part. Generally, an anomaly in the data will be reflected by a large change in the small principal components and will thus have a high residual value.

Typically a difference of 3 times the standard deviation is an indicator for outliers, which is used by Lakhina, Crovella, and Diot [32] to mark samples as malicious. To find the best threshold, the differences between the TPR and the FPR have been calculated per event log for the different thresholds used in constructing the ROCs. By inspecting the mean of the maximum differences between all event logs and in relation to the standard deviation, the threshold for principal component based classification (PCC) has been selected.

3.4 Concluding Remarks

A dataset has been collected featuring Windows security event logs of workstations in an enterprise environment. To research whether host-based anomaly detection is able to distinguish between benign and malicious logon records in this dataset, a method and measures have been presented to report on the performance under the assumptions and limitations which have been scoped. Secondly, the implementation steps have been described stressing the importance of pre-processing and feature selection. Finally, two different methods, clustering and principal component based classification (PCC), to distinguish between different event records have been introduced.

Based on the described dataset, the proposed method, and the detailed steps taken to actually develop the two anomaly detection methods to detect deviations in the logon records of the Windows security event log, the results of this effort are presented next.

Chapter 4

Results

This chapter covers the findings regarding the implemented anomaly detection approaches, as described in chapter 3. First, the observations concerning the overall performance and individual wrongly-classified records are described in section 4.1. Second, in section 4.2 the described observations are interpreted and analysed with respect to the implemented approaches. The discussion of the results is wrapped up in section 4.3.

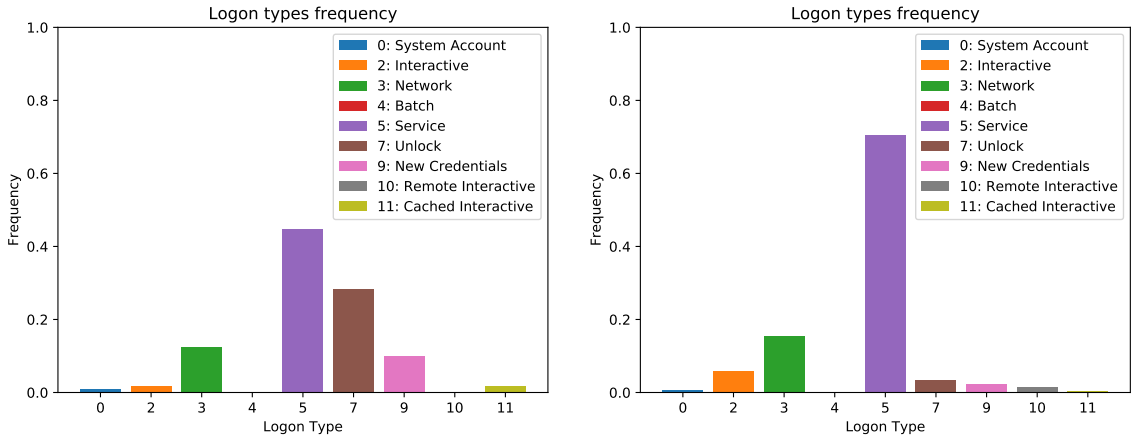
4.1 Observations

First, the logon records are inspected in subsection 4.1.1. Next, the observations with respect to the results of the different methods have been summarised in subsection 4.1.2. Afterwards, the wrongly-classified records are analysed in subsection 4.1.3 and subsection 4.1.4.

4.1.1 Exploration

The total dataset, contained 51,423 logon event records. However, the event logs of 2 users were notable as outliers, because both logs had less than 300 logon records due to the low number of working days in which the users were using their workstations. These logs have not been taken into account in the final evaluation, but will be reported on separately for the insights this delivers. In Table 3.2 the logon events under inspection have already been enumerated and as shown, with the removal of the outlying logs, 51,030 logon records of 58 event logs have been used to validate the performance of the 2 detection methods. Each workstation featured on average 880 records of which roughly 57% was filtered during pre-processing. The actual detection algorithms, therefore, had on average 373 relevant records to work with, ranging from at least 167 to at most 646 benign records. On average, the randomly added sample of malicious records constituted 10% of the total amount of benign and malicious records combined.

In Figure 4.1 the frequencies of the different logon types have been plotted. As can be deduced from the comparison between the benign records, Figure 4.1a, and the malicious records, Figure 4.1b, service logons are always the majority and account for most of the filtered event records. As described in subsection 3.3.1, the service logons are filtered, because these are non-distinctive. The biggest difference is in the amount of logons to unlock a computer. This could be explained as in a benign scenario where users lock their computer while doing other work, thus leaving the computer inactive when unattended. In a malicious scenario, however, the time frame extracted contains data of an active attack which, in most cases, set up a connection logging on using `NTLM` authentication. This



(a) Benign logon records.

(b) Malicious logon records.

Figure 4.1: Frequencies per logon type of benign and malicious logon records.

scenario does not involve an unlock. Logon types 10, remote interactive, and 11, cached interactive, are logged when connections are set up via RDP. Most important are logon types 3 and 9, because these signify network logons respectively logons using new credentials. Therefore, both logon types 3 and 9 involve domain accounts, which could be used for lateral movement, and identify the originating location of a logon by registering an IP address or workstation name of the initiator. These logon types show identifiable anomalies to the benign use cases based on their attributes. Network logons, type 3, could stand out based on an unexpected value in the *WorkstationName*-attribute or a different *IpAddress*. Logon type 9, which uses different credentials for remote connections, features the attributes *TargetOutboundDomainName* and *TargetOutboundUserName*, which are the credentials for those outgoing connections. Anomalies show in these attributes when an account is used from an unexpected origin. These logon types have therefore been at the centre of the detection efforts. Next, the methods and averaged results are discussed.

4.1.2 Summarised Results

This section discusses the choices regarding classifying records either benign or malicious and the averaged results per method, which have been summarised in Table 4.1.

Clustering has been executed over the dataset as a whole, no distinction can be made between a train and test set. This lies in the nature of the HDBSCAN clustering algorithm [38], which evaluates all records in relation to each other. Adding new points could shift the clustering and would therefore trigger a new run of the clustering algorithm. It has empirically been deduced that it works best to classify all records being given an outlier score as malicious, when defining a small minimum cluster of about 5% of the available logon records. In Figure 4.2 it can be seen that the average area under the curve of the ROC for the clustering method is 0.880 and a 0.034 standard deviation. Any records with an outlier score above 0, signifying small differences from the other clustered records, have been marked anomalous.

With clustering a mean true positive rate of 85.63% has been achieved while 8.29% of the records have been wrongly classified as positives. The standard deviation for the true and false positive rates lay respectively at 6.99% and 4.65%. The maximum true positive rate was 97.14% and the minimum

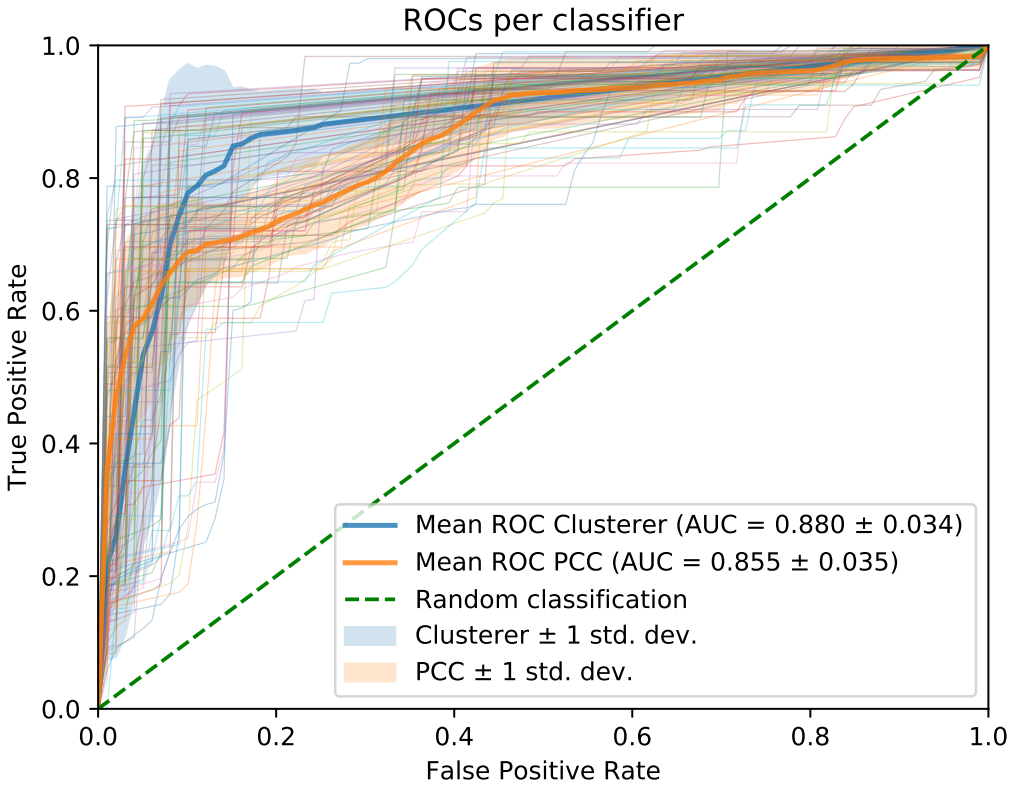


Figure 4.2: The mean ROC per method with the standard deviation marked. The mean cross-validated ROC of each individual event log has been drawn as well.

67.5%; the false positive rates ranged from 0.31% to 24.77%. There were 5 event logs for which the FPR lay above 15%.

Based on the mean ROC of the PCC method, which is displayed in Figure 4.2 and features an AUC of 0.855, it could be deduced that around a TPR of 60-65% the false positive rate would increase significantly. Therefore, a conservative threshold of 3.70 times the standard deviation is chosen to keep the false positive rate manageable. The ideal threshold between different logs differed quite significantly and therefore the highest thresholds have been inspected to select this conservative threshold.

The PCC method achieved a mean true positive rate of 59.81% with a standard deviation of 7.54%, the mean false positive rate was 4.70% with a standard deviation of 3.31%. Respectively 43.14% and 72.06% were the minimum and maximum recorded true positive rates. For all but 3 logs the FPR lay below 10% due to the conservative threshold, the minimum achieved false positive rate was 0.00% and the highest encountered FPR 13.95%.

Both Figure 4.2 and Table 4.1 show the PCC method achieves a lower TPRs than clustering. However, until a TPR of 60% the FPR is also lower. At higher true positive rates (TPRs) PCC is not able to properly distinguish between benign and malicious records and as such the FPR increases more rapidly than with clustering. The PCC method therefore lacks in detection capability, but performs initially stronger with respect to the false positive rate (FPR).

Measure	Mean	Std	Min	Max
Records	879.828	218.780	439	1411
Clustering				
AUC ROC	0.882	0.029	0.820	0.938
TPR	0.856	0.070	0.675	0.971
FPR	0.083	0.033	0.003	0.248
Principal Component based Classification				
AUC ROC	0.857	0.033	0.729	0.924
TPR	0.598	0.075	0.431	0.721
FPR	0.047	0.033	0.000	0.130

Table 4.1: Summary per measure per method.

	Subject		Target				
Type	*Domain*	*User*	*Domain*	*User*	*Workstation*	*Ip*	*Logon Process*
3	-	-	<DomainName>	<DomainAdmin>	-	Internal, different subnet	Kerberos
3	<DomainName>	<ComputerName>$	<DomainName>	<UserName>	<WorkstationName>	-	winlogon
2	<DomainName>	<ComputerName>$	<DomainName>	<UserName>	<WorkstationName>	127.0.0.1	User32
11	<DomainName>	<ComputerName>$	<DomainName>	<UserName>	<WorkstationName>	127.0.0.1	User32
4	<DomainName>	<ComputerName>$	<DomainName>	<UserName>	<WorkstationName>	-	.UBPM

Table 4.2: Examples of false positives.

As concluded, clustering based on HDBSCAN is able to achieve higher TPRs, but this inherently comes at the cost of more false positive classifications. However, with an FPR of less than 15% for the majority of the inspected event logs and a mean FPR of 8.29%, clustering has a strong performance nonetheless. Figure 4.3 shows the results of both methods for each security log individually. It clearly shows clustering achieves a better performance with fairly comparable false positive rates.

Next, the discussion explains and compares the results of the detection methods, also covering the outlying event logs.

4.1.3 False Positives

False positive classifications have been encountered and the results show that on this front improvements have to be made before the implemented methods can be part of any anomaly-based detection system. As described by Axelsson [16], the FPR is an important measure of the usefulness of a system and 8.3%, respectively, 4.7% is too high for practical use, given the total amount of records that are daily handled in an enterprise environment [3, 19]. Examples of generalised false positives have been given in Table 4.2.

Inspection of the logs showed that false positives were mainly related to two reasons. A domain admin account regularly logging on for maintenance purposes and rare logon types, particularly type 2, interactive logons, and cached interactive logons, type 11. Concerning the privileged domain administrator account, logon records of this domain administrator were, for most logs, sparsely available. As reported by Asanger and Hutchison [3], "differentiation between various account (...) types is crucial" to counter false positives. Proper naming conventions should be employed for the classification of privileged accounts [3], however, this was not the case for this domain administrator account. A re-run, filtering this specific account during pre-processing, confirmed this: the false positive rate was reduced by ∼1.4 percentage points, respectively, ∼1.1 for PCC with respect to the

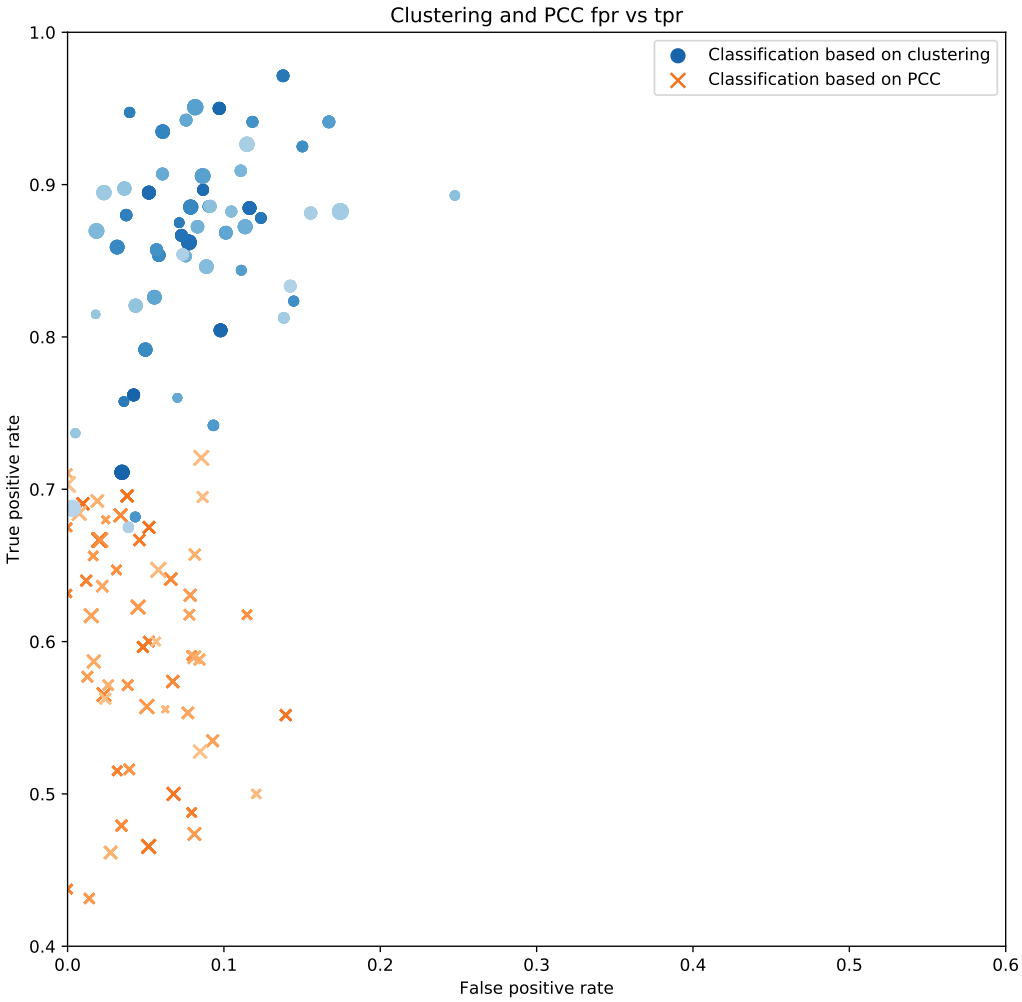


Figure 4.3: The performance for clustering and PCC per individual workstation log.

results as described in subsection 4.1.1. However, bluntly filtering is not the solution to reduce these false positives as it concerns a domain administrator account, which could be misused by adversaries. A solution should either be able to correctly handle these type of records or filter based on additional context information.

Another source of false positive records is related to rare logon types in the dataset. As mentioned and shown in Figure 4.1 as well, relatively few logons such as interactive, or cached interactive logons are available. Therefore, these type of logons stand out as anomalous as too few records are available to form a cluster, even if, their use cases are legitimate. As a domain user account is involved, these logons have neither been filtered.

Analysis of the false positives, therefore, led to the conclusion that false positive classifications are mostly the result of rare logon records. This falls in line with the challenges and expectations concerning anomaly detection systems and conclusions of related work, such as among others Ahmed, Mahmood, and Hu [1], and Buczak and Guven [18].

Type	*SubjectUserName*	*TargetUserName*	*Workstation*	*Ip*	*Logon Process*	*AuthenticationPackageName*
2	<ComputerName>$	**Other Account**	<WorkstationName>	127.0.0.1	seclogo	Negotiate
3	-	<UserName>	**Unknown workstation**	**External**	**NtLmSsp**	**NTLM**
7	<ComputerName>$	<UserName>	<WorkstationName>	**External**	User32	Negotiate
7	<ComputerName>$	<UserName>	<WorkstationName>	-	Negotiat	Negotiate
11	<ComputerName>$	<UserName>	<WorkstationName>	127.0.0.1	User32	Negotiate

Table 4.3: Examples of false negatives. The top entries display anomalous values, however, the bottom entries are equal to benign records.

4.1.4 False Negatives

Similar to the false positives, false negative classifications have been inspected as to why these records have not been detected. In Table 4.3, a distinction between the false negatives could be made based on whether or not anomalous values were present. False negatives, which did show anomalous values, concern interactive logons, logon type 2, from another domain account, as evidenced by the *TargetUserName* or unlocks, type 7, of the computer from an unknown workstation with an external IP address.

Another example are type 3 domain logons, using `NTLM`-authentication from an external IP address. No satisfying theory could be drafted as to why these records were missed by the PCC method. The records differ in the following features: *AuthenticationPackageName*, *IpAddress*, *LmPackageName*, *LogonProcessName*, and *WorkstationName*. Besides the indicated features of interest, *IpAddress* and *WorkstationName*, also the authentication process deviated because `NTLM` authentication was selected explicitly. However, despite the deviation found during manual inspection, the calculated deviation stayed under the selected threshold for multiple workstation logs. Clustering did not seem to suffer from these false negatives and correctly detected the pass-the-hash attacks, which caused these type 3 domain logon records.

As stated, false negatives that were indistinguishable from benign records have been encountered as well. The unlock, logon type 7, and cached interactive logon, type 11, in the bottom of Table 4.3 are indistinguishable from benign logon records of these types. In case the benign records have been clustered properly, thus no false positives have been classified concerning the rare logon types as described in subsection 4.1.3, it is a logical consequence that these malicious data points are missed by the detection algorithm. However, as evidenced by the logon type, because a workstation was unlocked, or credentials had been cached, these logon records were part of a set of records related to the same session. Therefore, malicious logon records are already available which do display anomalous values. As such, the related intrusions are detectable, despite these indistinguishable records.

Given the described false negatives, the malicious records containing anomalous values are of more concern. Further research into the reason why these records are being missed is needed. The next section describes the impact and decisions concerning filtering and features.

4.2 Anomaly Detection Approach Analysis

This section discusses the impact of the applied filtering and selected features related to the description of the results in the previous section.

4.2.1 Filtering

The most important step in pre-processing is filtering of logon records. Based on a study into the Windows logon event and its attributes, the semantics of the record values have been deduced. This knowledge and a comparison between benign and malicious records resulted in the filtering approach as described in subsection 3.3.1. Only records which were local to the workstation have been filtered, because these were typical to the inner workings of the Windows operating system and did not involve the Windows domain network. Besides, these records were usually either identical or benign and malicious records were indistinguishable from each other, therefore, filtered records were almost impossible to classify correctly with the applied methods. The applied filtering approach greatly improved detection results by increasing the TPR and reducing the FPR.

As discussed in subsection 4.1.3, based on the review of false positive records, more aggressive filtering was applied to asses its impact on the detection rate and FPR. However, given that a) domain user accounts were involved, b) benign and malicious records are distinctive from each other, or c) records form an important part of the baseline, as is the case with logons unlocking a computer, filtering is not the solution to improve TPRs and FPRs.

In general, filtering is applied to remove the uninteresting records, feature selection is applied to improve anomaly detection with the records that are left after filtering. Therefore, the next section covers the impact of selected features.

4.2.2 Features

In general, the selected features show comparable influence on the results for clustering, as well as, PCC. The automatic feature selection approach resulting in the feature set of Table 3.4, as described in subsection 3.3.2, has been taken as a basis for further experiments. The results of excluding specific features are shown in Table 4.4. *RestrictedAdminMode* has been excluded in all experiments, because it is only available for remote interactive logons, logon type 10. As the literature described the originating location to be most important, features regarding the authentication used, privileges assigned, and logon process handling the logon have been left out to assess their impact. The more constant features, such as the *AuthenticationPackageName*, the *ImpersonationLevel*, or *LmPackageName* for example, did show more impact deviations for malicious records than initially expected based on related work.

For example, techniques such as pass-the-hash [52] were also identifiable because of the value in *AuthenticationPackageName*, besides the *IpAddress* being logged. Normally, the package name was either `Kerberos` or `Negotiate`, meaning the connecting systems would use `Kerberos` when available or fall back to `NTLM`-authentication. However, upon execution of pass-the-hash, an `NTLM`-hash is being passed directly and no negotiation process takes place to decide the authentication method being used. Therefore, this deviates from the standard logon records and gives an extra indication of lateral movement. The same goes for the *LmPackageName*, which is usually empty as only an authentication package is transmitted after `NTLM`-authentication has been negotiated. Except, when `NTLM` is selected immediately and no negotiation takes place, as is the case with pass-the-hash attacks, the *LmPackageName* is known upon registration of the logon in the event log.

It was found that *ElevatedToken* did improve the detection capabilities, especially for clustering. Without this feature, which logs the privileges of a logon, the TPR was roughly 9% lower. However, this feature also had a big impact on the FPR, which increased with roughly 4%. The PCC method showed another picture. While *ElevatedToken* does seem to increase the detection capability of the anomaly detector as well, unlike with clustering, the FPR decreased when the *ElevatedToken* was included. In Figure 4.4 the related ROC is shown, the difference with Figure 4.2 is the exclusion of

Excluded Features	AUC ROC	TPR	FPR
Clustering			
None excluded	0.882	0.856	0.083
AuthenticationPackageName	0.874	0.850	0.081
AuthenticationPackageName & *LmPackageName*	0.876	0.849	0.083
ElevatedToken	0.857	0.743	0.046
ElevatedToken & *KeyLength*	0.856	0.763	0.042
ImpersonationLevel	0.881	0.859	0.075
ImpersonationLevel & *KeyLength*	0.882	0.848	0.078
ImpersonationLevel & *ElevatedToken* & *KeyLength*	0.855	0.760	0.039
KeyLength	0.881	0.856	0.083
Principal component based classification			
None excluded	0.857	0.598	0.047
AuthenticationPackageName	0.836	0.627	0.058
AuthenticationPackageName & *LmPackageName*	0.822	0.589	0.050
ElevatedToken	0.774	0.473	0.040
ElevatedToken & *KeyLength*	0.867	0.637	0.048
ImpersonationLevel	0.859	0.566	0.024
ImpersonationLevel & *KeyLength*	0.895	0.674	0.030
ImpersonationLevel & *ElevatedToken* & *KeyLength*	0.879	0.646	0.030
KeyLength	0.878	0.662	0.063

Table 4.4: Impact of exclusion features w.r.t. the selected features, as mentioned in Table 3.4.

ImpersonationLevel, *KeyLength*, and *RestrictedAdminMode*. As mentioned, *ElevatedToken* especially aided PCC and this is reflected by the ROC. To be able to better compare the impact of the *ElevatedToken*-attribute, the same threshold of 3.70 times the standard deviation has been employed as in subsection 4.1.2. However, the ROC suggests a better optimal threshold exists.

The *ImpersonationLevel* and *KeyLength* had a small effect on the clustering method, when included the FPR increased ∼0.8 percentage points, respectively, no effect was shown. However, regarding PCC, the *ImpersonationLevel* showed a clear impact. Generally adding about 2 percentage points to the false positive rate, while no or negative impact was shown concerning the detection rate. The detrimental effect of the combination of the *ImpersonationLevel* and *KeyLength* is especially seen with regards to the improved ROC for the PCC method in Figure 4.4, when these features are excluded. The *KeyLength*-feature showed a similar, but opposite effect as the *ImpersonationLevel* reducing the TPR. As seen in Table 4.4, the *ImpersonationLevel* and *KeyLength* form a peculiar pair of features, which are better excluded when using PCC.

From the selected features, one often used feature in intrusion detection is missing: the *SystemTime*. Based on several reasons the time of the event log has been left out. First and most importantly, time as registered in the Windows security event records is very precise and would have acted similarly as the *EventRecordID*, as if it was an almost unique identifier. Without any additional manipulations, overfitting might have occurred. Second, this would also have influenced the evaluation process, as described in subsection 3.2.2, because the time of malicious records would have to be adapted as well to the workstation log under inspection. For this task, assumptions would have to be made which could not be supported and would border to random guesses. Especially the reproducability would have been impacted, resulting in a subjective evaluation. Furthermore,

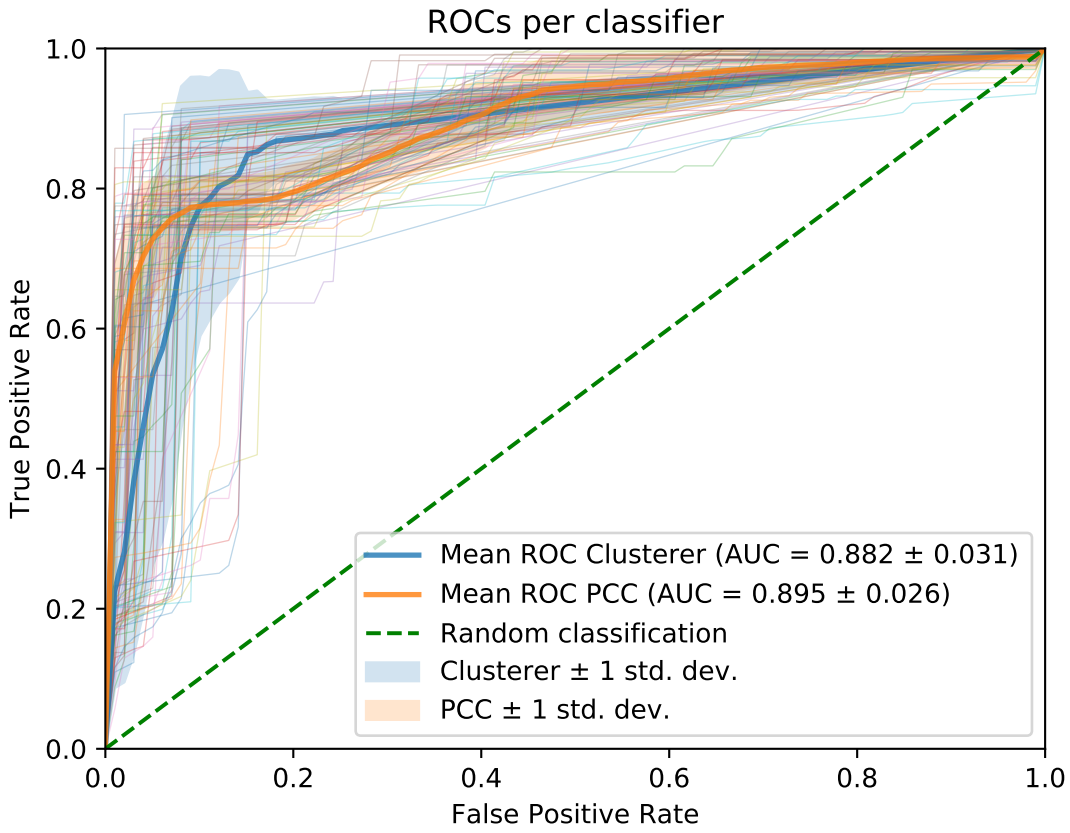


Figure 4.4: The mean ROC without *ImpersonationLevel*, *KeyLength*, and *RestrictedAdminMode*.

research investigating security event-based lateral movement detection did rarely mention time as an interesting feature [28, 52].

With the approach and its impact on the results discussed, the next section wraps up this chapter, also covering topics that have been left unanswered.

4.3 Concluding Remarks

This chapter described the results of the implemented anomaly detection approaches. The performance of clustering, using HDBSCAN, was evaluated against the statistical PCC method. It was found that, although the FPR was fairly high with 8.29%, clustering provided a better detection rate of 85.63%. Displaying an FPR of 4.70%, PCC performed thus stronger. However, with 59.81% the TPR lacked behind.

An inspection of the false positives and negatives, revealed that pass-the-hash [52] attacks were the main reason for the difference in the detection rate between clustering and PCC. The explicit declaration of `NTLM` authentication was seen as a deviation by the clustering method while the PCC method regularly miss-classified these malicious sampels. False positives mainly related to a privileged domain administrator account executing maintenance tasks and rare logon records.

A review of the implemented approaches showed the impact of feature selection on the TPR and FPR. Especially the *ElevatedToken* has a considerable impact, increasing both the TPR and FPR for both methods. The *ImpersonationLevel* and *KeyLength* turned out to be better excluded when applying PCC, as exclusion of these features resulted in both an improved TPR as well as FPR.

Lastly, limitations, mainly regarding filtering, and the outlying logs are covered in the next section.

4.3.1 Limitations

This section covers limitations of this research regarding topics that have been left unanswered.

Multiple logs showed false positives related to rare logon types, such as interactive logons of type 2 and 11 and less commonly batch logons, type 4. This thesis is not able to answer whether or not these logon types might be filtered out completely during pre-processing, because the impact of filtering these logons is unknown. Unlike the currently applied filtering, completely filtering these logon types would also filter logon records involving domain user accounts. As this remains an open question, these logon records have not been filtered based on the assumption that classification for these records should recognise them as benign and because malicious records have been encountered which do show anomalous values. Filtering of these records might have resulted in a bias towards the dataset used and artificially inflate the results with a, possible, lower false positive rate. Additionally, inspection of the selected features showed the impact of the *ElevatedToken* and *ImpersonationLevel* attributes towards the false positive rate (FPR). The most commonly encountered differences between wrongly classified benign records and correctly classified ones, resided in these attributes.

The 2 outlying event logs which contained 111 and 282 logon event records showed a TPR and FPR of 0%. This means all different records were clustered perfectly, as the differences between records was too big to fall in the same cluster. Therefore, the assumption that anomalies are signified by an HDBSCAN outlier score did not hold in these cases. Based on the other assumption, as described in subsection 3.3.3, that malicious records might form smaller clusters, it might be possible to correctly discern between the benign and malicious records of these logs.

The next chapter concludes on the findings of this thesis and offers recommendations for future work, also related to these open topics.

Chapter 5

Conclusion

This thesis covered lateral movement techniques with a focus on advanced techniques, which make detection more difficult. In order to counter these techniques, a shift is needed to distribute detection efforts towards the endpoints and this thesis shows that anomaly detection can be based on the Windows security event logs of individual workstations.

Lateral movement is executed by attackers inside their target's environment. It is an adversarial tactic to move, after the initial compromise, from system to system within a target environment in order to reach the true objectives of an intrusion. In this context, the first layer of defence, employing control measures aimed at intrusion prevention, have already been by-passed by the attackers. The next layer of an defence-in-depth strategy is aimed to detect these intrusions that successfully by-passed network boundary security controls. Detection of intruders proves a hard task, as evidenced by the median time of 101 days adversaries persisted in the information technology (IT) environments of their victims [37]. This thesis is an effort to close this gap by providing increased detection capabilities designed for the machines which are often compromised first during an intrusion, namely, employee workstations. To research detection strategies regarding lateral movement, the following question was posed:

RQ1: *What are the intrusion detection techniques used for detecting lateral movement?*

Detection techniques regarding lateral movement looked into the connection patterns between machines and specifically logons. These techniques aimed to discern deviating behaviour, based on the finding that "hosts in an enterprise network are constrained by company policies and employee job functions and exhibit more homogeneity than those on the open internet" [34]. Anomaly detection was commonly applied to find these patterns and detect deviations from them. Given that lateral movement is executed on compromised machines inside a target's IT environment, the discussed methods used host data for anomaly detection based on clustering or statistical methods. Therefore, a host-based anomaly detection approach was chosen for the detection of advanced lateral movement techniques, researching the performance of both a clustering method as well as a statistical method.

The anomaly detection approach is aimed at detecting lateral movement on Microsoft Windows workstations, because Microsoft Windows is the most widely adopted operating system by enterprises [25]. The second research question was aimed to investigate the type of logging provided by machines running Windows, which could aid in this effort:

RQ2: *What logging provided by Microsoft Windows systems can be used for detecting advanced lateral movement techniques?*

The Windows operating system offers the security event log, which registers numerous types of events, including logon events. As related work indicated specifically the logon event to be of interest for lateral movement detection [3, 28, 52], anomaly detection was aimed at the logon event records from the security event logs. Combined with the findings from the first research question and a study of the attributes of logon events, an initial feature set of 20 features has been devised, which provides the information necessary for anomaly detection to discern a logon event as deviating. Within this feature set the features registering the origin of a logon in combination with the actual user account have found to be most important. The origin of a logon is registered by the *IpAddress* and *WorkstationName* features. The *UserName* and *UserSid*, both available for a subject as well as a target user, declare the user accounts involved. Based on these features anomaly detection can be applied to discern logon patterns and deviations from the predominant patterns. Other features in the initial feature set have been inspected and it was found that, especially, the *ElevatedToken* had impact, which registers the account's activated privileges. Unfortunately, not only the detection rate increased, but also the false positive rate (FPR) did.

To research and compare the 2 chosen anomaly detection methods in a host-based implementation using the logon event of the Windows security event log, the last research question was devised:

> **RQ3:** *Which anomaly detection method better detects a deviation in the Windows security event log?*

In order to evaluate the detection methods, a dataset consisting of 58 event logs has been collected from workstations in an enterprise-size company. The workstation logs have been supplemented with malicious logon records taken from a workstation in an attack environment, containing a typical enterprise setup of a Windows domain. A professional red team has executed attacks involving lateral movement, creating realistic records of malicious actions. In total, 51,030 benign logon records have been gathered and supplemented with 651 malicious logons.

The first method employed clustering, using an implementation [38] of HDBSCAN [20], which classified Windows security event log records based on the outlier score. This score expresses how well a record fits in the cluster it has been assigned to. The second method employed principal component analysis (PCA) to detect statistical deviations between records [32, 50]. Dubbed principal component based classification (PCC), PCC models a training set based on the largest principal components. Anomalous records show more deviation in the smaller principal components, therefore, fitting the modelled data less. A conservative threshold has been used to classify records as malicious when the residual value exceeded 3.70 times the standard deviation.

Both methods, clustering and PCC, proved to be able to identify deviating logons. The results show that, although, clustering is able to achieve higher detection rates with an 85.63% true positive rate (TPR), PCC showed not only a lower TPR, 59.81%, but also a lower FPR with 4.70%. Clustering, however, achieved an 8.29% FPR, which is too high given the amount of users logging on daily to their computer in an enterprise-size company. Even though, PCC performs better with respect to the false positive rate, which is deemed most important [16], the TPR also stays behind and especially false negative classification of logon records related to pass-the-hash [10] attacks are of concern. Therefore, clustering is deemed better and improvements should focus on lowering the FPR, which is discussed in the next section.

5.1 Future Work

The scope of this thesis focused on whether or not a host-based detection mechanism based on the Windows security event log is able to indicate the presence of malicious activity in the form of

Event ID	Description	Source
4610	An authentication package has been loaded by the Local Security Authority	
4611	A trusted logon process has been registered with the Local Security Authority	
4625	An account failed to log on	[3, 52]
4648	A logon was attempted using explicit credentials	[3, 17]
4672	Special privileges assigned to new logon	[3, 29]
4688	A new process has been created	[29, 26]
4798	A user's local group membership was enumerated	
4799	A security-enabled local group membership was enumerated	

Table 5.1: Possible interesting events for host-based future work, as indicated by the literature.

lateral movement. Based on previous work choices have been made regarding the use of an anomaly detection approach, also, only the logon event has been used for detection. This section offers pointers for future work that has been identified.

Based on the applied approach concerning feature selection and the focus on only logon events, the effects of pattern mining techniques have not been researched. However, research, such as Basagoiti et al. [17] executed, suggests an anomaly detector on the Windows security event log should incorporate pattern mining. Combined with the approach in this thesis, improvement of the presented anomaly detector might be possible with the addition of pattern mining techniques. Future work in this area could look into the third assumption concerning clustering approaches that anomalous data points tend to belong to smaller and sparser clusters [1].

As stated by Tavallaee, Stakhanova, and Ghorbani [56], the definition of what constitutes as an anomaly should be determined and because of this thesis' interest in lateral movement, filtering was based on the conclusions about logon records showing deviations concerning the involved user accounts or originating location. However, when looking to privilege escalation [11] techniques, for example, filtering will have to be adjusted. Because in the current approach, records of a local system logon have been filtered as no differences are present between a benign logon or a malicious logon of the administrator user, as shown in Table 3.3. This is explained, because malicious use cases of these type of logons relate to privilege escalation in preparation of, among others, lateral movement. While no distinction can be made based on a comparison of the attributes, the application of pattern mining could be researched to investigate whether a deviation can be detected from the normal pattern of these type of logons.

Besides focusing on only logon events, also other events could be inspected. Again, proper care should be taken on defining the scope of the detection capabilities and during pre-processing filtering needs to be adjusted accordingly. When looking to other events, other event types concerning logons might be of interest in the case of lateral movement detection. A few suggestions based on the literature and this research are indicated in Table 5.1. Pattern mining might detect anomalies in the registration of logon processes, events 4610 and 4611, prior to the actual logon attempts. Events 4625, 4648, and 4672 are different type of logons and although less information is contained in these events as compared to logon event 4624 as used in this thesis, an extended picture of the security situation might be supplied by these events. Another example of interesting events relate to the enumeration of group membership as these event have been seen in the dataset prior to lateral movement related logon records. These events might indicate the execution of discovery

techniques [4].

Furthermore, the limitations of this thesis could also be investigated, to research improvements on anomaly-based lateral movement detection. In particular, the application of the described approach in real-time would be a huge improvement towards an actual detection solution. The Windows security event log seems a valuable source for host-based detection mechanisms, which warrants more research.

References

[1] M. Ahmed, A. N. Mahmood, and J. Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, January 2016.

[2] D. Alperovitch. Deep in Thought: Chinese Targeting of National Security Think Tanks. `https://www.crowdstrike.com/blog/deep-thought-chinese-targeting-national-security-think-tanks/`, July 2014. Accessed: 2018-09-16.

[3] S. Asanger and A. Hutchison. Experiences and Challenges in Enhancing Security Information and Event Management Capability Using Unsupervised Anomaly Detection. In *2013 International Conference on Availability, Reliability and Security*, pages 654–661, September 2013.

[4] ATT&CK. Discovery. `https://attack.mitre.org/wiki/Discovery`, August 2017. Accessed: 2018-09-08.

[5] ATT&CK. Lateral Movement. `https://attack.mitre.org/wiki/Lateral_Movement`, August 2017. Accessed: 2018-09-08.

[6] ATT&CK. Cobalt Strike. `https://attack.mitre.org/software/S0154/`, October 2018. Accessed: 2018-12-06.

[7] ATT&CK. Credential Dumping. `https://attack.mitre.org/techniques/T1003/`, October 2018. Accessed: 2018-12-05.

[8] ATT&CK. Initial Access. `https://attack.mitre.org/wiki/Initial_Access`, March 2018. Accessed: 2018-09-12.

[9] ATT&CK. Mimikatz. `https://attack.mitre.org/software/S0002/`, October 2018. Accessed: 2018-12-06.

[10] ATT&CK. Pass the Hash. `https://attack.mitre.org/techniques/T1075/`, October 2018. Accessed: 2018-12-05.

[11] ATT&CK. Privilege Escalation. `https://attack.mitre.org/wiki/Privilege_Escalation`, March 2018. Accessed: 2018-09-08.

[12] ATT&CK. Remote Desktop Protocol. `https://attack.mitre.org/wiki/Technique/T1076`, October 2018. Accessed: 2018-12-05.

[13] ATT&CK. Valid Accounts. `https://attack.mitre.org/wiki/Technique/T1078`, April 2018. Accessed: 2018-09-27.

[14] ATT&CK. Windows Management Instrumentation. `https://attack.mitre.org/wiki/Technique/T1047`, January 2018. Accessed: 2018-09-19.

[15] S. Axelsson. Intrusion Detection Systems: A Survey and Taxonomy. Technical report, Chalmers University of Technology, March 2000.

[16] S. Axelsson. The Base-rate Fallacy and the Difficulty of Intrusion Detection. *ACM Transactions on Information and System Security (TISSEC)*, 3(3):186–205, August 2000.

[17] R. Basagoiti, U. Zurutuza, A. Aztiria, G. Santafé, and M. Reyes. Clustering of Windows Security Events by Means of Frequent Pattern Mining. In Á. Herrero, P. Gastaldo, R. Zunino, and E. Corchado, editors, *Computational Intelligence in Security for Information Systems*, pages 19–27, Berlin, Heidelberg, July 2009. Springer Berlin Heidelberg.

[18] A. L. Buczak and E. Guven. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys Tutorials*, 18(2):1153–1176, October 2015.

[19] A. S. Buyukkayhan, A. Oprea, Z. Li, and W. Robertson. Lens on the Endpoint: Hunting for Malicious Software Through Endpoint Data Analysis. In M. Dacier, M. Bailey, M. Polychronakis, and M. Antonakakis, editors, *Research in Attacks, Intrusions, and Defenses*, pages 73–97. Springer International Publishing, October 2017.

[20] R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-Based Clustering Based on Hierarchical Density Estimates. In *Advances in Knowledge Discovery and Data Mining*, pages 160–172. Springer Berlin Heidelberg, April 2013.

[21] P. Chen, L. Desmet, and C. Huygens. A Study on Advanced Persistent Threats. In *IFIP International Conference on Communications and Multimedia Security*, pages 63–72. Springer, 2014.

[22] P. Clark. Marriott's $13.6 Billion Starwood Deal Bought Security Risk. *Bloomberg*, November 2018. Accessed: 2018-12-04.

[23] Ernst & Young. Path to cyber resilience: Sense, resist, react. *EY's 19th Global Information Security Survey 2016-17*, 2016.

[24] A. Fawaz, A. Bohara, C. Cheh, and W. H. Sanders. Lateral Movement Detection Using Distributed Data Fusion. In *2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS)*, pages 21–30, September 2016.

[25] A. A. Forni and R. van der Meulen. Gartner Survey Finds Government CIOs Spend 21 Percent of Their IT Budget on Digital Initiatives. `https://www.gartner.com/en/newsroom/press-releases/2017-04-25-gartner-survey-shows-85-percent-of-enterprises-will-have-started-windows-10-deployments-by-end-of-2017`, April 2017. Accessed: 2018-12-05.

[26] D. Hendler, S. Kels, and A. Rubin. Detecting Malicious PowerShell Commands Using Deep Neural Networks. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ASIACCS '18, pages 187–197, New York, NY, USA, June 2018. ACM.

[27] E. M. Hutchins, M. J. Cloppert, and R. M. Amin. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. *Leading Issues in Information Warfare & Security Research*, 1(1):80, 2011.

[28] JPCERT-CC. Detecting Lateral Movement through Tracking Event Logs. Technical Report Version 2, Japan Computer Emergency Response Team Coordination Center, December 2017.

[29] JPCERT-CC. Tool Analysis Result Sheet. https://jpcertcc.github.io/ToolAnalysisResultSheet/, December 2017. Accessed: 2018-09-19.

[30] G. Keizer. Microsoft: 200M now use Windows 10 in the enterprise. *Computerworld*, May 2018.

[31] B. Krebs. Marriott: Data on 500 Million Guests Stolen in 4-Year Breach. https://krebsonsecurity.com/2018/11/marriott-data-on-500-million-guests-stolen-in-4-year-breach/, November 2018. Accessed: 2018-12-03.

[32] A. Lakhina, M. Crovella, and C. Diot. Diagnosing Network-wide Traffic Anomalies. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, pages 219–230. ACM, September 2004.

[33] R. M. Lee and R. Lee. The Who, What, Where, When, Why and How of Effective Threat Hunting. *SANS Institute*, February 2016.

[34] Z. Li and A. Oprea. Operational Security Log Analytics for Enterprise Breach Detection. In *2016 IEEE Cybersecurity Development (SecDev)*, pages 15–22, November 2016.

[35] Log2Timeline. Plaso. https://github.com/log2timeline/plaso, January 2018. Accessed: 2018-09-06.

[36] Mandiant. APT1 Exposing One of China's Cyber Espionage Units, February 2013.

[37] Mandiant. M-Trends 2018. Special report, FireEye, 2018.

[38] L. McInnes and J. Healy. Accelerated Hierarchical Density Based Clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42. IEEE, 2017.

[39] Microsoft. Directory data store. https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc736627(v=ws.10), June 2011. Accessed: 2018-11-15.

[40] Microsoft. Credentials Processes in Windows Authentication. https://docs.microsoft.com/en-us/windows-server/security/windows-authentication/credentials-processes-in-windows-authentication, October 2016. Accessed: 2018-12-06.

[41] Microsoft. 4624(S): An account was successfully logged on. https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4624, April 2017. Accessed: 2018-09-21.

[42] Microsoft. 4688(S): A new process has been created. https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4688, April 2017. Accessed: 2018-09-22.

[43] Microsoft. Active Directory Domain Services Overview. https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview, May 2017. Accessed: 2018-11-15.

[44] Microsoft. Basic security audit policies. `https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/basic-security-audit-policies`, April 2017. Accessed: 2018-09-18.

[45] Microsoft. Desktop Window Manager. https://docs.microsoft.com/en-us/windows/desktop/dwm/dwm-overview, May 2018. Accessed: 2018-11-19.

[46] Microsoft. *Windows Management Instrumentation*. Microsoft, May 2018. `https://docs.microsoft.com/en-us/windows/desktop/wmisdk/wmi-start-page`, accessed 2018-09-16.

[47] L. H. Newman. The Ransomware That Hobbled Atlanta Will Strike Again. *Wired*, March 2018. Accessed: 2018-12-03.

[48] Online Trust Alliance. Cyber Incident & Breach Trends Report. *The Internet Society*, January 2018.

[49] H. Ringberg, M. Roughan, and J. Rexford. The Need for Simulation in Evaluating Anomaly Detectors. *SIGCOMM Comput. Commun. Rev.*, 38(1):55–59, January 2008.

[50] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. A Novel Anomaly Detection Scheme Based on Principal Component Classifier. Technical report, Miami University Coral Gables, FL. Deptartment of Electrical and Computer Engineering, January 2003.

[51] H. Siadati and N. Memon. Detecting Structurally Anomalous Logins Within Enterprise Networks. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 1273–1284, New York, NY, USA, November 2017. ACM.

[52] M. Soria-Machado, D. Abolins, C. Boldea, and K. Socha. Detecting Lateral Movements in Windows Infrastructure. Technical report, Computer Emergency Response Team - European Union (CERT-EU), February 2017.

[53] B. Strom, A. Applebaum, D. Miller, K. Nickels, A. Pennington, and C. Thomas. MITRE ATT&CK: Design and Philosophy. Technical report, The MITRE Corporation, July 2018.

[54] X. Sun, J. Dai, P. Liu, A. Singhal, and J. Yen. Towards Probabilistic Identification of Zero-day Attack Paths. In *2016 IEEE Conference on Communications and Network Security (CNS)*, pages 64–72. IEEE, October 2016.

[55] Symantec. SamSam: Targeted Ransomware Attacks Continue. `https://www.symantec.com/blogs/threat-intelligence/samsam-targeted-ransomware-attacks`, October 2018. Accessed: 2018-12-02.

[56] 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)*, 40(5):516–524, September 2010.

[57] The MITRE Corporation. Adversarial Tactics, Techniques & Common Knowledge. `https://attack.mitre.org/wiki/Main_Page`, January 2018. Accessed: 2018-09-14.

[58] K. J. Wijnands. Using endpoints process information for malicious behavior detection. Master's thesis, Delft University of Technology, September 2015.