Master of Science in Engineering: Computer Security
May 2018

Identifying high risk targets in a corporate multi-user network

Oskar Edbro
Annika Hansson

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Engineering: Computer Security. The thesis is equivalent to 20 weeks of full time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author(s):
Oskar Edbro
E-mail: osed13@student.bth.se

Annika Hansson
E-mail: anhm13@student.bth.se

University advisor:
Assistant Professor Martin Boldt
Department of Computer Science and Engineering

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

Context. A corporate multi-user network is often the target when it comes to cyber attacks. Today, there are different ways of working to secure these networks. One of them is vulnerability scanning which shows the vulnerabilities currently present in a network. This project investigates a new method where this information can be used in combination to data about users and their access to prioritise the risks in the network and recommend the order of where to start with remediation activities.
Objectives. The aim is to propose a new method for prioritisation of remediation actions for hosts in a corporate network. This will proactively aid the network administrators in their daily work of securing the network.
Methods. The proposed method uses information about users' access and activity as well as vulnerability information within the network, and is tested with an experiment. In the experiment the proposed method and the state-of-the-art method, is compared to a prioritisation set by domain experts. The measurements used for verification of improvement are accuracy, Root Mean Square Error (RMSE) and Cohen's kappa.
Results. The result of this study is a new automatised method for prioritisation which uses information about users in the network combined with the vulnerabilities currently present, for a certain host. The experiment shows that the prioritisation becomes more accurate, when using the proposed method compared to the current state of the art method, with an absolute Cohen's d value above 5, for all measures.
Conclusions. The improvement of the prioritisation indicates that more factors in the analysis provides for a more accurate prioritisation. If the risk assessment is accurate, then the prioritisation becomes more useful to the network administrator. The implementation of the proposed method is not faster than the state-of-the-art method, but is still significantly faster than doing this prioritisation manually.

Keywords: Risk management, proactive security, network management, decision support systems

Sammanfattning

Bakgrund. Företag med stora interna nätverk är ofta mål för dataintrång. Det finns idag många olika sätt att arbeta för att säkra upp dessa nätverk. Ett av dem är sårbarhetsskanning, som har som mål att presentera de sårbarheter som finns. Detta arbete undersöker en ny metod för att kunna prioritera vilken enhet i nätverket där åtgärder bör appliceras först. Detta genom att ta hänsyn till användare på enheterna och hur utbredd deras tillgång till andra enheter är.
Syfte. Målet med detta arbete är att ta fram en ny metod för prioritering av åtgärder som kan förbättra säkerheten i ett företags nätverk. Detta för att hjälpa nätverksadministratörer att proaktivt arbeta med säkerhet på ett effektivare sätt.
Metod. Den nya föreslagna metoden använder både information om användares access och aktivitet samt sårbarheter i nätverket, och har testats genom ett experiment. I experimentet har den resulterande prioritiseringen av den föreslagna metoden och den metod som används i branchen idag, jämförts mot en prioritering gjord av experter på området. Mätvärdena som använts är träffsäkerhet, Root Mean Square Error och Cohen's kappa.
Resultat. Detta arbete har resulterat i en ny, automatiserad metod, som använder information om användare, i kombination med sårbarheter i nätverket, för en host. Experimentet påvisar en klar förbättring av prioriteringen gjord av den föreslagna metoden men ett absolutbelopp av Cohen's d värden på över 5, för alla mätvärden.
Slutsatser. Förbättringen av prioriteringen gjord av den nya föreslagna metoden, indikerar att träffsäkerheten ökar med fler faktorer. Desto fler faktorer som används, desto träffsäkrare blir prioriteringen vilket i sin tur leder till en prioritering som är mer användbar för en nätverksadministratör. Utöver den förbättrade träffsäkerheten så är tidsbesparingen av prioritering i relation till vad som skulle behövas vid manuell prioritering avsevärd, dock inte riktigt lika snabb jäntemot den metoden som används i branchen idag.

Nyckelord: Riskanalys, förebyggande säkerhet, nätverksadmnistration, beslutsstödssystem

Acknowledgments

The authors would like to thank the supervisor at Blekinge Institute of Technology Assistant Professor Martin Boldt for your support, help and enthusiasm during the project. You have been a valuable help with all parts of the thesis and provided great encouragement and insights.

The authors would also like to thank the supervisors from Outpost24, Martin Jartelius and Nils Forsman, for your support, patience and inspiration during this thesis. It has been a privilege to work with you and we are very grateful to have been very well welcomed by everyone at your office in Karlskrona. Thank you for including us in the joyfulness at breakfast every morning and providing a good working environment. Also, a huge thank you goes out to the domain experts, who have helped with setting the ground truth which is a big part to ensure the validity of our work.

Without the support, enthusiasm and humour from all of you, this thesis would not have reached its current state.

Contents

Abstract i

Sammanfattning iii

Acknowledgments v

Abbrevations xvi

1 Introduction 1
- 1.1 Related Work . . . 3
  - 1.1.1 User account correlation . . . 3
  - 1.1.2 Log analysis . . . 3
  - 1.1.3 Intrusion Detection Systems . . . 4
  - 1.1.4 Penetration Testing . . . 4
  - 1.1.5 Vulnerability scanning . . . 4
  - 1.1.6 Research gap . . . 5
- 1.2 Research Questions . . . 5
- 1.3 Limitations . . . 5
  - 1.3.1 User account management and access . . . 6
  - 1.3.2 Experiment . . . 6
  - 1.3.3 Prioritisation . . . 6
  - 1.3.4 Data collection . . . 7
- 1.4 Contribution . . . 7
- 1.5 Outline . . . 7

2 Background 9
- 2.1 Lightweight Directory Access Protocol (LDAP) . . . 9
- 2.2 Kerberos . . . 9
- 2.3 FreeIPA . . . 9
- 2.4 System Security Service Daemon (SSSD) . . . 10
- 2.5 Bourne Again Shell (BASH) . . . 10
- 2.6 Windows Active Directory . . . 10
- 2.7 PowerShell . . . 11
- 2.8 Cached credentials . . . 11
- 2.9 Secure Shell . . . 12

3 Method **13**
3.1 Background study . . . 13
3.2 Experiment planning . . . 13
3.3 Experiment operations . . . 14
3.4 Experiment validity threats . . . 15
3.5 Dataset description . . . 16
3.6 Implementation . . . 17

4 Background study **21**
4.1 User account correlation . . . 21
4.1.1 Risk assessment and categorisation of users . . . 22
4.1.2 Risk assessment and categorisation of hosts . . . 23
4.1.3 Summary of background study . . . 23
4.2 Proposed analysis method . . . 24
4.2.1 Risk calculations for hosts, and users . . . 24

5 Results **27**
5.1 State-of-the-art method (SOTA) . . . 27
5.2 Proposed method without user data (PMWUD) . . . 27
5.3 Proposed method (PM) . . . 28
5.4 Other Observations . . . 29

6 Analysis and Discussion **31**

7 Conclusions and Future Work **35**
7.1 Future work . . . 35

References **37**

A PowerShell commands for data collection **43**

B BASH commands for data collection **45**

C Experiment Data **47**
C.1 Host Data . . . 47

List of Figures

1.1 Network description 2

3.1 Flow chart for the implemented python3 program. 18
3.2 An entity relationship diagram of the database. 19
3.3 Example of the visualisation generated by the proof of concept implementation. Boxes are hosts and eclipses are users. The boxes are colour coded from red, highest risk, to green, loswest risk. Each arrow is a user that can access the host, where the red ones are the users that contributes to the UHS. 20

List of Tables

3.1 Description of the dataset, and the source of date. 17

4.1 Categorisation for users as risks in a network. 24
4.2 Definitions of the terms used for the mathematical calculations. . . . 26

5.1 The prioritisations for the pilot experiment. The first dataset was used for this experiment. 28
5.2 The prioritisations for the main experiment. The domain experts prioritisations are presented as well in separate columns, these are what the expert opinion is based on. The second dataset was used for this experiment. 29
5.3 The measurements between expert opinion and the different test cases for the pilot experiment, when run 1000 times. 30
5.4 The measurements between the expert opinion and the different test cases for the main experiment, when run 1000 times. 30
5.5 The effect size of the main experiment when comparing state-of-the-art method with the proposed method. 30

Listings

A.1 The registy key which controls the information about cached credentials 43
A.2 A PowerShell command to list access logs 43
A.3 A PowerShell command that lists all active accounts on the host . . . 43
A.4 Lists user information from the domain. 43
A.5 Lists group information from the local machine. 43
A.6 Lists group information from the domain. 43
B.1 Lists logins and logouts during the last year according to the log files. Logs are reset upon reboot. 45
B.2 A command to list local acounts in linux 45
B.3 A small script for gathering md5 sums of all public ssh-keys in `~/.ssh` 45
B.4 A small script to show all info about rules applicable to the host . . . 45
B.5 Lists all users which can be authenticated through FreeIPA from the current machine 46

Abbreviations

AAA Authentication, Authorisation and Accounting

AD Active Directory

BASH Bourne Again Shell

CVE Common Vulnerabilities and Exposures

CVSS Common Vulnerability Scoring System

DC Domain Controller

DoS Denial of Service

GUBP General User Behaviour Profile

HIDS Host-based Intrusion Detection System

IdM Identity Management

IDS Intrusion Detection System

IPA Identity, Policy and Audit

IPS Intrusion Prevention System

LAN Local Area Network

LDAP Lightweight Directory Access Protocol

NIDS Network-based Intrusion Detection System

NSS Name Service Switch

PAM Pluggable Authentication Modules

PAUBP Personal Adaptive User Behaviour Profile

PKC Public Key Cryptography

RCR Relative Cumulative Risk

RHEL Red Hat Enterprise Linux

RMSE Root Mean Square Error

RODC Read Only Domain Controller

RWDC Read Write Domain Controller

SSH Secure Shell

SSO Single Sign-On

SSSD System Security Service Daemon

UBP User Behaviour Profile

Chapter 1

Introduction

As an administrator in a corporate network, it is important to have control over who has access to the network and what kind of access they have. If the users are unauthorised by the administrator or have more access than needed for their specific tasks [18], it is hard to ensure that the network is secure. At the same time, it can be problematic for a user to not have enough access to parts of the network which are required for the expected work [18, 37]. To provide sufficient access for a user it needs to be reviewed and updated depending on work tasks instead of just adding access when the tasks change. To achieve this it is common to use role-based access control [37].

The user with most privileges on a Linux host is called *root*. This user can alter anything on the host. Therefore, it is important to make sure that only the users who need it have root access, otherwise it can have devastating consequences for the company and its network. In the documentation for the Linux distribution Fedora and Red Hat Enterprise Linux 6 for example, user management is explained to be maintained with access control by using groups [10, 32]. This means that access should be assigned to different groups in the system and then the user account should be assigned to a group which allows the user to perform his or her job properly but nothing more.

The user with highest privileges in Windows Operating System is called *NT-Authority\system* but a more commonly used account is the administrator account. Administrators can alter the host and add, remove or edit users and user privileges as well as install or remove software. To control access for users in Windows, the administrator can use the Microsoft Identity Manager [1] [20]. This manager keeps track of users in hosts on the corporate network and can provide needed task-based privileges for a user.

Many companies today have large corporate networks with many users and a lot of sensitive information and resources that need protection. These networks are also connected to the internet for the purposes of email contact, web surfing and other necessities. In general, connections are a two-way street and if there is a way out, there is a way in, which makes it a target for attackers.

Even if the general security is there, such as firewall, Intrusion Detection System (IDS), Intrusion Prevention System (IPS) and centralised logging, the network administrator will not know of a breach until it has already occurred. To proactively work with securing company assets, penetration testing or vulnerability scanning can

[1] `https://docs.microsoft.com/en-us/microsoft-identity-manager/microsoft-identity-manager-2016`

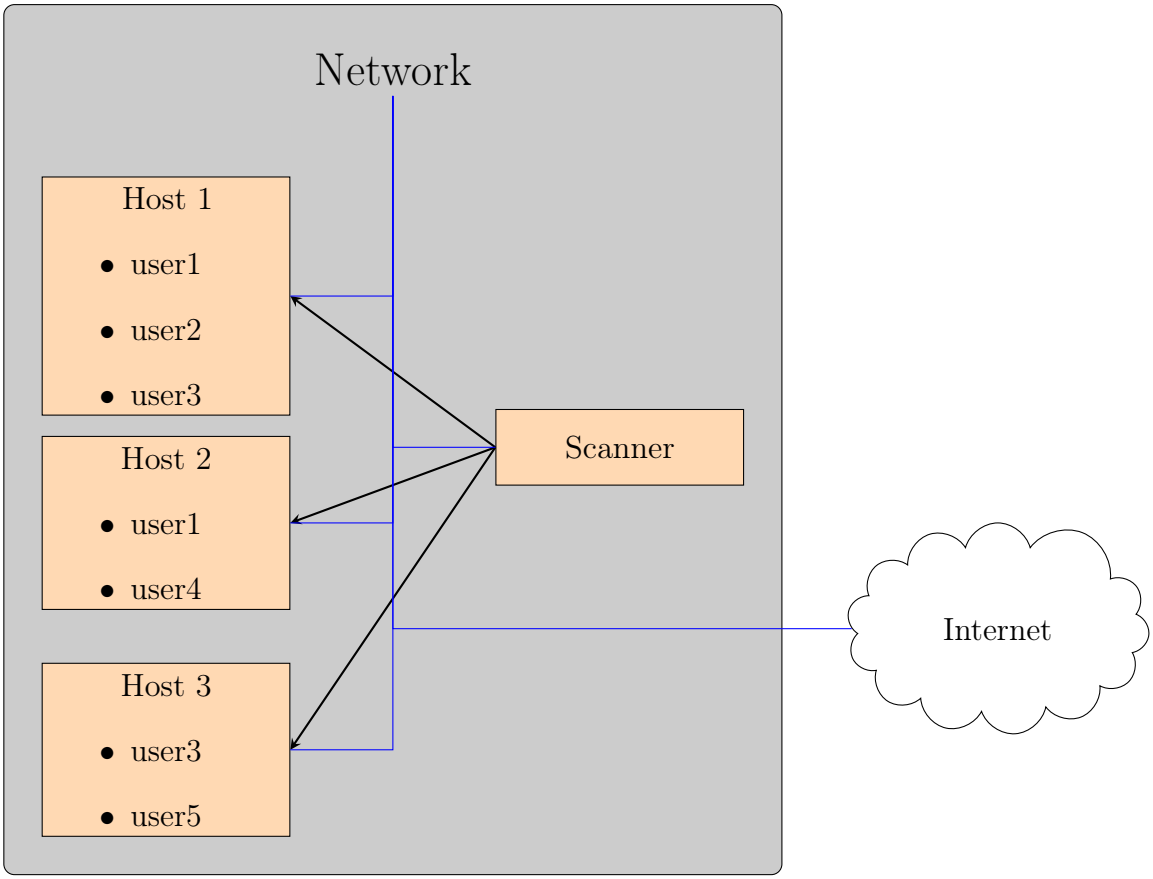

Figure 1.1: Users and hosts in a network in relation to the scanner. The blue lines are network cables that connect the hosts and the arrows represent scans run on the hosts.

be used.

A vulnerability scanner can be a part of the network it scans, and collects vulnerability data by scanning the hosts, as demonstrated in figure 1.1. These services do however, mostly focus on vulnerabilities on hosts when proposing vulnerability remediation activity recommendations. This assessment does most often not include users and their activity in the network.

Users can pose a risk to a corporate network depending on how often they access resources and what privileges they have when performing a task, or in the network in general. By looking at what access users have and how this access is actually being used, with regards to their host access within a network, could improve the overall security in the system. It is important to evaluate high risk hosts and users because if they are compromised, it could have a devastating affect for a whole network and ultimately, an entire company. This makes some parts of the network more valuable to keep secure and would benefit the most from mitigating vulnerabilities.

The risk concept described in this report, refers to the risk of spreading malicious activity inside the network. If a compromised host or user account is classified as a potential high risk, then all connected accounts and hosts could also become compromised or infected. That way, large parts of a network can get infected or compromised if those potential entry points are not discovered in time.

1.1 Related Work

The existing research within this problem domain, with focus on how to proactively prevent security flaws and how to prioritise which host to apply remediation activities to based on user activity in a multi-user network, is very limited. However, some recent research in a parallel domain has focused on evaluating different tools for log analysis in order to track activity and collect evidence from a forensic perspective [4, 16]. There are also effective methods for linking digital identities but mainly for social media accounts [11, 31]. These methods require a certain amount of data in the form of usernames, geographical tags and posts with text and pictures.

1.1.1 User account correlation

Research has shown that it is possible to correlate user accounts across multiple social media sites [11, 31] due to the amount of information people publish about themselves. This can be useful during forensic investigations, when companies merge, for targeted adds, or for an attacker when monitoring their targets.

Digital identities can also be linked together by analysing usernames and linked accounts with different usernames [31]. Even if a user has the option to change the username, it is rarely used. However, when it is used, the change is often very predictable and can therefore still be linked to the user's other digital identities.

1.1.2 Log analysis

In Windows the system event logs can be used effectively to extract information of what events have occurred [16]. The collected information can then be used for forensic purposes for example, to find an intrusion and to identify what was stolen or was changed.

There are some tools available today, which can be used when extracting and analysing information to find anomalies. One of them is The Open Source Elastic Stack [2] which includes Logstash [3], Kirbana [4] and Elasticsearch [5] Combined, these tools presents a good presentation of the activity based on the collected logs [16]. An investigation shows that the analysing of the logs is efficient for security auditing [16]. Since most operating systems or services already have some type of logging available, it is possible to collect them in one place with centralised logging, for analysing with these tools. It is also recommended to use centralised logging on a dedicated external resource to minimise the risk of compromising the information in the logs [16]. This is because in the event of a breach, the log server is isolated from the rest of the network and not be accessible for the intruders and the chances of the digital evidence to stay intact, increases.

[2] https://www.elastic.co/products
[3] https://www.elastic.co/products/logstash
[4] https://www.elastic.co/products/kibana
[5] https://www.elastic.co/products/elasticsearch

1.1.3 Intrusion Detection Systems

Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) are designed to alert if there is any abnormal activity in a network or isolate the suspicious activity to minimize the damage [25]. They provide an extra layer of security for a network because they can detect known attacks and malware. There are two different types of IDS systems, Host-based Intrusion Detection System (HIDS) and Network-based Intrusion Detection System (NIDS). The NIDS are used to detect anomalies in an entire network based on the network traffic and Host-based Intrusion Detection System (HIDS) do the same but for anomalous activity on a host [1]. An IDS does not however, prevent intrusions proactively but enables the network administrator to respond to the incident when the network is or has been under attack.

1.1.4 Penetration Testing

One way to improve security for a large network, is to do manual penetration testing. This is when a person tries to find vulnerabilities, which can be exploited to gain unauthorised access to resources in the target network [14]. These tests can be made from different perspectives or scopes, which depends on what the network owner wants to have tested. The penetration tester can also be familiar with the target network (white box) or have no previous knowledge of the network set up (black box).

The results from a penetration test should be summarised in a report [14], and can then be used as a base for recommendations on where to start patching the network. This technique relies on finding possible entry points and vulnerabilities in the network in order to gain unauthorised access to valuable resources [14], rather than study the activity of the users inside the network. Even if the penetration tester successfully performs a social engineering attack for example, to gain access to the target network there is not necessarily any knowledge of other user activity in the network.

1.1.5 Vulnerability scanning

A study of how effective automated vulnerability scanning is to find flaws in a network [13], has been conducted and compares seven different vulnerability scanners on two different platforms, Linux and Windows. The scanners where tested for both authenticated and unauthenticated scans. The results show that automated scans are useful but should be complemented with other methods as well, since only a subset of the vulnerabilities where found [13]. There were significant differences between scans for Linux and Windows machines [13], both for found vulnerabilities and false positives.

The number of attack paths in a network increases with the number of vulnerabilities for neighbouring hosts in a network. By using the Common Vulnerability Scoring System (CVSS) to classify vulnerabilities, the risk can be calculated for a single host but does not consider the vulnerabilities on the neighbouring hosts. To get a more accurate risk assessment of the combination of hosts in a network, the respectively CVSS scores can be adjusted by checking the neighbouring CVSS score as well [43]. The result is a formula to calculate the Relative Cumulative Risk (RCR)

which is based on the CVSS score but considers neighbouring risks (hosts inside the network or connections to the internet) too. This can then be used to prioritise the remediation activities for the hosts in the network.

1.1.6 Research gap

Previous research have focused on identifying abnormal activity within a network or on a host by analysing the event logs. These tools can in some cases provide sound forensic evidence for malicious activity on a host. An IDS system monitors the network activity and will react to anomalies in the network traffic. The proactive methods found for securing a network are manual penetration testing and vulnerability scanning which can be useful to detect possible attack vectors before an actual intruder.

Research for how users affect the risk value of a host in combination with vulnerabilities, were very limited. The risk methods discussed today does not include users access and activity but only vulnerabilities present in the network. Users can pose a risk to the network if a person has more access than needed in a network or frequently accesses sensitive data which was not intended. If that person's account is compromised, then significantly larger losses could be the reality for the corporation.

1.2 Research Questions

The aim of this thesis is to identify neighbouring effects from user's inter-host activities combined with existing vulnerabilities in a corporate multi-user network, which can be of interest to potential hackers and the system administrator of a target network. The users, and hosts will be classified depending on the level of risk they pose to the system by analysing the user activity. Then this classification is merged with the existing analysis of the vulnerabilities present on the different hosts in the network.

To reach the described aim, the following research questions will be answered:

RQ1: What methods are available today to correlate user accounts on several hosts in Windows and Linux corporate networks?

RQ2: How can a user be defined and classified as a risk based on its activity and access in Windows and Linux corporate networks?

RQ3: How can a host be defined and classified as a risk based on the user classification and its vulnerability score in Windows and Linux corporate networks?

RQ4: How does the proposed method, that takes into account risk classifications of hosts and users in a network, perform at prioritising the overall risks related to the hosts in a network when compared to a state-of-the-art risk prioritisation method?

1.3 Limitations

Some limitations has been set for this project to specify the network setup that is going to be tested. This delimitation is due to the impossibility of taking every available option into account.

1.3.1 User account management and access

The mapping of users in the target network, will consist of tracking their user accounts on the different hosts. The user accounts can be both local and distributed on different hosts.

When checking the type of access a user has on a host, it will be recognised as root or administrator or just regular user without any special access. This is because groups are created and managed by the network administrator and therefore can have odd names or special logic behind it. Administrator or root groups on the other hand are created by default when the operating system is installed.

If protocols like Secure Shell (SSH) are used, the assumption will be made that the generated keys are placed in the default folder. In the case of SSH this will be in the '`.ssh`' folder in a user's home directory.

1.3.2 Experiment

The data used in the experiment will be collected from one network consisting of both Windows and Linux machines. The network is a controlled environment and all tests are run with the consent of the network owner. The number of hosts will be limited to seventeen (17) and the number of users will be limited to eleven (11). By keeping the dataset small the workload for the domain experts are kept reasonable.

The hosts where chosen based on convenience sampling, in the test network of an industry-leading company in vulnerability management. To get an accurate assessment of the test network, 4 domain experts where sampled from the same company to take advantage of their knowledge and experience within the domain.

Default configurations will be assumed as much as possible, since there are many different options for the network administrator. In reality, configurations for a network can differ a lot depending on knowledge and experience of the network administrator and by making this assumption, the proposed method will be applicable for a large variety of networks.

The Windows machines will be connected to a Domain Controller (DC) in the network and for the Linux machines, FreeIPA will be used as a connection between the server and clients. All clients will be assumed to run System Security Service Daemon (SSSD) for communication with the server. Both Windows Active Directory (AD) and FreeIPA uses Lightweight Directory Access Protocol (LDAP) and Kerberos as authentication protocols and can communicate with each other. The servers assumes also to be present in the target network and not placed on a remote geographical site.

Due to the fact that the hosts that are used in the experiment are vulnerable to different vulnerabilities, all data about them will be anonymised and thereby not reversible to the actual hosts. This means that there is no possibility to know where the vulnerable hosts does exist, and thereby minimise the risk of exploitation.

1.3.3 Prioritisation

The proposed method will prioritise the risks that each host poses and not the risk a vulnerability poses. This is because the resulting prioritisation should be used as

a recommendation to the network administrator of which host to start remediation activities on first.

1.3.4 Data collection

The data collection is out of scope of this project and will be provided by a state-of-the-art vulnerability scanner. However, the user data will be collected from the network manually, based on the commands presented in Appendix A for Windows 10, and B for Centos7, and by using the graphical user interface of the Operating Systems or centralised login in all other cases.

1.4 Contribution

The main contribution of this thesis is a new decision making process to proactively secure a corporate multi-user network by proposing a new method of prioritising hosts where remediation activities should be applied first. This based on the risk the host poses to the rest of the network if it were to be targeted. The risk analysis focuses on the users of hosts, their access, privileges and login frequency alongside the ordinary vulnerability information from a vulnerability scanner to propose a prioritisation.

This thesis also contributes with a definition of what a high risk user is in both Linux and Windows environments based on the information gathered in the background study. It also provides a definition of what a high risk host can be in a network, based on its users. These definitions are used in the proposed method to identify these high risk targets.

To investigate to what extent the new method compares to the state-of-the-art method, an experiment is conducted in a controlled environment. The result of both methods are then analysed in comparison to a ground-truth set by experienced domain experts.

In addition to this, the decision making process is implemented as a proof of concept, resulting in automatisation of the analysis. This saves a lot of time for the person in charge of the network security, for a large corporate network. The prioritisation enables the network administrator to start remediation activities for the hosts that poses the highest risk for the network.

1.5 Outline

The following parts of this report are chapter 2, which presents a brief background for this problem domain. Then the method is presented in chapter 3. Chapter 4 contains a background study which is used to answer RQ1 - 3, which is followed by the result of the experiment in chapter 5, that is used to answer RQ4. Discussions and analysis are presented in chapter 6 and then the very last part presents conclusions and proposals for future work in chapter 7. Last but not least, is the appendix containing additional information referenced in the report.

Chapter 2

Background

This part will give an overview of some of the key technical components that is used in this research. It will give a brief introduction, that focuses on the parts that can, and will, be used when it comes to data gathering.

2.1 Lightweight Directory Access Protocol (LDAP)

Lightweight Directory Access Protocol (LDAP) is a way to centralise management of multiple hosts. It can be used for authorisation, access control or to share a phone directory [42] with a client-server model [33].

LDAP uses a model based on entries. Each entry contains of a number of attributes which is a type and one or more values [33]. This can be used to store user information among other things. In that case, the entry is a new user which has an attribute of type name and a value which is the user's name.

2.2 Kerberos

Kerberos[1] is originally a project from MIT and was developed in the 1980s [6]. Today it has become one of the most commonly used authorisation and authentication system in modern networks [6].

Kerberos is a system for Single Sign-On (SSO) [7], which means that a user receives a token on login and then that token is used for authentication on multiple systems [5]. In reality, Kerberos is often used in combination with LDAP, where Kerberos is used for authentication and LDAP for authorisation on the local host[5].

AD [17], SSSD [29] and FreeIPA [46] are implementations that uses a combination of LDAP and Kerberos for secure access control.

2.3 FreeIPA

Identity, Policy and Audit (IPA) can be used to handle Authentication, Authorisation and Accounting (AAA) when it comes to network access for Linux and Unix. One implementation of this, is FreeIPA[2] which is supported for Fedora and Red Hat Enterprise Linux (RHEL).

[1] `http://kerberos.org/`

[2] `https://www.freeipa.org/`

FreeIPA can be managed both through the command line and through the web interface. As default the admin account is created which can be used to create new users. The admin is also placed in the access group *admins*. As default when new users are created they are added to the user group *ipausers*.

2.4 System Security Service Daemon (SSSD)

System Security Service Daemon (SSSD) [3] is a service that can be used to connect to LDAP, AD, or IPA when handling system security [51]. It is the recommended client for centralised authentication in RHEL. With the close relation between RedHat and CentOS, it is commonly used in CentOS as well.

SSSD supports online and offline support as well as caching through a common interface [29]. It also provides interfaces like Pluggable Authentication Modules (PAM)[4] which is the standard authentication mechanism in RHEL and Name Service Switch (NSS)[5] which is used to instruct the SSSD to retrieve user information [15]. It is also possible to use SSSD to establish communication between Windows AD domain and for example FreeIPA for Linux [29].

2.5 Bourne Again Shell (BASH)

The Bourne Again Shell (BASH) shell is a command interpreter and a programming language [34] which is most commonly used in the GNU operating system. The combination of programming language and interpreter allows for the commands to be placed together in a file and that way, become new commands.

BASH is a newer version of the Unix shell but is compatible with predecessor, `sh`. Features from Korn Shell (ksh) and C Shell csh can also be used from the BASH shell [34].

2.6 Windows Active Directory

Windows Active Directory (AD) is Microsoft's own network Operating System [9] which has a hierarchical structure for objects in Windows [19]. The objects contains information of the user's phone number, email address etc.

An AD has a Domain Controller (DC) which is used to authenticate users on different resources in the network as well as control and distribute the resources [9, 19]. It can also be used to change settings for the clients connected to the domain.

A host that belongs to a Windows domain are in contact with a DC [9]. The DC can be layered with, for example, a Read Write Domain Controller (RWDC) at the top, a Read Only Domain Controller (RODC), and a set of hosts as leafs. The RWDC has all the data stored, that the RODC then will ask for and cache the needed information, that is asked for by the hosts [9].

[3] `https://pagure.io/SSSD/sssd`

[4] `https://www.centos.org/docs/5/html/Deployment_Guide-en-US/ch-pam.html`

[5] `https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/deployment_guide/configuration_options-nss_configuration_options`

To control access in Windows for users in a domain, site and organisational unit a Windows AD can be used [24]. As default, it is the domain administrator, enterprise administrator, the operating system and the policy group creator owner, who can create and manage new policy groups[6]. If a regular user needs access to manage policies, then that user can be added to the policy group creator owners security group [20].

Another option for Windows machines is to interact with each other. This is to form a Local Area Network (LAN) consisting of a number of Windows host. This LAN is called a workgroup and all host connected to that workgroup are peers. Through this workgroup the hosts can share files, internet connection and access to printers and other devices.

A local account can be either a local user, or a Microsoft account that is defined on the local host [21][7]. The local user is a user account that is created on, and does not exist outside, the local host. The Microsoft account on the other hand is a online account that is accessible via the internet, but is defined to be allowed on the local host, by the local host.

2.7 PowerShell

PowerShell is a command line tool for Windows Operating System. It is an open source project [22] developed for administrator and has an integrated scripting environment attached [23][8]. PowerShell can be used to extract system and user information of current events on the current machine as well as make additional configurations [23]. If the current machine happens to be a DC then authentication logs can be extracted covering activity of all client machines in the Windows domain.

2.8 Cached credentials

In Windows, a client that is connected to a domain, does store cached credentials as default [39, 44]. This means that the client can authorise a user to the domain without being in contact with the DC. If the user wants to access another machine in the network, the user will be able to do that if the resource does not require the DC to validate the user credentials [44].

For usability, cached credentials are recommended [39] since users can still access their workstations even if there is no connection to the DC. Therefore, work hours are not lost if the network goes down or if someone is working from a remote location.

From a security aspect, it is not a very good solution, since there exist malware which uses cached credentials to gain unauthorised access [30]. If an intruder manages to get a hold of the administrators credentials from a client machine, then he or she has administrative rights in the entire network. This results in multiple possible attack paths for the attacker.

[6] https://technet.microsoft.com/en-us/library/cc978262.aspx

[7] https://msdn.microsoft.com/en-us/library/aa394507(v=vs.85).aspx

[8] https://docs.microsoft.com/en-us/powershell/scripting/powershell-scripting?view=powershell-5.1#getting-started-with-powershellgetting-startedgetting-started-with-windows-powershellmd

In Linux Environments where FreeIPA is used, the clients also store cached credentials for usability [50]. It allows local authentication operations for the local machine [50]. However, if a user wants to jump further through the network on the same credentials, he or she will be asked to provide the password or key each time.

2.9 Secure Shell

Secure Shell (SSH) is a protocol which provides a secure encrypted connection for the user, even from a remote location [3]. It was designed to replace other remote log in protocols originally, which sends unprotected data through a network like for example, Telnet [48].

The SSH key exchange protocol consists of a few stages. The first stage is to establish a connection from the client to the remote server. This is done in plain text and the client provides the server with supported algorithm [48]. The second stage is to decide on a shared key which is done with Diffie-Hellman's key exchange algorithm. When a shared key is established the server sends a certificate, a public key, a signature of the digest message transcript as well as a shared secret key to confirm its identity to the client [48].

Chapter 3

Method

In this section the methodology used to answer the posed research questions and reach the aim, will be presented.

3.1 Background study

A background study of relevant publications will be conducted to answer RQ1 - 3 posed in section 1.2. By investigating the problem domain, a background will be provided as well as a summary of the current research in the area. It will also reveal what current methods are used today to solve the posed problem.

The answers from the background study will be used to create a proposed method, and its implementation. The intention is to formulate an accurate method which works for the described network and probably other corporate networks.

3.2 Experiment planning

To compare the proposed method to the current state-of-the-art method, an experiment will be conducted. First, a case study was considered, but due to the sensitive nature of the data it was discarded in favour of an experiment in a controlled environment.

The experiment will be divided into three different cases which tries the different prioritisation methods. The aim of this experiment will be to measure the improvement of the prioritisation when using the user activity information in addition to vulnerability data. This experiment will provide an answer to RQ4.

The experiment will be tested for a dataset generated for the sole purpose of performing this experiment. The user correlation process will not be prioritised at this time, but focus on how vulnerabilities are affected by already correlated users. The user data gathering will be done in a manual fashion, using both graphical user interface and command line scripts. In the cases applicable the commands explained in Appendix A and B are used.

The independent variables selected for the experiment is the prioritisation method used, and the user data. This means that there are three cases that will be tested:

1. **The current state-of-the-art method**

   On a per vulnerability basis, it looks at how much CVSS score can be mitigated with a fix. That fix is the most important one. When it comes to the host

part, the current state-of-the-art method is based on the simple count of all vulnerabilities on the host, even informational vulnerabilities which have a CVSS of 0.

This test case will be referenced as SOTA (State-of-the-art), in this report.

2. **The proposed method without user data**

   This test case makes a prioritisation with the proposed method without user data. In this case, that is very close to using the current state-of-the art method. This test case will be referenced as PMWUD (Proposed Method Without User Data), in this report.

   The main difference between these methods, 1 and 2, is that the proposed method does not count vulnerabilities that have not been assigned a Common Vulnerabilities and Exposures (CVE), and summarises the CVSS instead of counting them.

3. **The proposed method**

   This test case uses only the proposed method which takes user activity into account. The calculations for the vulnerabilities are the same as for the previous test case.

   This test case will be referenced as PM (Proposed Method), in this report.

The dependent variables will be the Cohen's d of accuracy, Cohen's Kappa and RMSE of the prioritisation, compared to a expert opinion, when compared to the same scores for the current state of the art method. The expert opinion is set by four domain experts for the dataset, selected via convenience sampling. The variables have been chosen for two different reasons, accuracy because it is well known and easy to understand. Cohen's Kappa and RMSE was chosen because it gives good view of the correlation between the rankings. The Cohen's d is chosen to statistically prove the improvements, which will be the value that answers RQ4.

To test the data provided for the domain experts, one of them will be randomly chosen to do a pilot experiment. By doing so the intention is to ensure that all the necessary data, as described in section 3.5, is attached. The experiment environment will be the same, but the dataset might be updated between the pilot and the main experiments, based on input from the first expert. The choice of who will be the domain expert for the pilot experiment may effect the end result, due to differences in work experience.

3.3 Experiment operations

First a dataset will be collected from a network. The dataset will then be used by the different prioritisation methods. The network that data is gathered from will be a test network with multiple hosts were the users and logs are created for the sole purpose of being used in this experiment. The current method, SOTA, for prioritisation is already implemented in the commercial product, but the proposed method will be implemented as a proof of concept.

The collected dataset will be sent to one of the domain experts, for the pilot experiment. Then the proof of concept implementation will be tested against the expert opinion, which will be the prioritisation made by the domain expert. For the main experiment, the data will be collected again, with any additions requested by the first expert, from the same network and sent to two other domain experts. Then the proof of concept implementation will be tested against the expert opinion based on a mean of those domain experts prioritisations.

The prioritisation supplied by the experts will be used as a expert opinion. The accuracy, Cohen's kappa and RMSE of the generated prioritisation will then be calculated in relation to the expert opinion.

When it comes to the main experiment, the expert opinion will be generated by summarising the hosts prioritisation positions, provided by the two domain experts and then ordered there after, to gain a new prioritisation. In the case that multiple hosts gain the same sum, they are positioned randomly. The choice to randomise the hosts that are equal, is because there are no other way to determine which has a higher risk, and this gives all of them an equal chance of being the highest and lowest risk. The expert opinion is only generated once for the pilot experiment and once for the main experiment, which is presented in chapter 5. This is done to give a clear view of the expert opinion prioritisation.

To be able to measure the improvement of the prioritisation between the proposed method and the SOTA, Cohen's d (3.1), was used. The M values are the two means that are compared and the σ is the corresponding standard deviation. The Cohen's d score is linked to the percentage of non overlap between too sets [2]. With an absolute score over 0.8, the improvement is Large.

$$\frac{M_1 - M_2}{\sqrt{\frac{(\sigma_1^2 + \sigma_2^2)}{2}}} \tag{3.1}$$

3.4 Experiment validity threats

There are four types of validity threats that are applicable to this experiment, construct, conclusion, internal and external. The construct validity is threatened by the risk that there is a strong expectancy by the domain experts, that looking at more parameters will improve the accuracy of the prioritisation. To minimise the risk of the getting biased prioritisations from the domain experts, they are not allowed to see the prioritisation made by the proposed method before making their own. The domain experts are also instructed not to communicate with each other during this phase and to send their prioritisation directly to the authors.

The threats to the validity of the conclusions, is that there is only a single dataset used for this experiment. This gives the results a low statistical power. The main threat to the internal validity is the fact that the domain experts are chosen from the company that the research is done in cooperation with. This is hard to mitigate since the dataset contains sensitive information which will not be trusted to third party domain experts. Therefore inhouse domain experts are the only possible option for this task.

The threats to external validity, and dominantly the risk that even though a new set of data with different focus instead of users, is added to the prioritisation algorithm, there are other variables that can further impact the resulting prioritisation. Another threat is that the user data is generated, and might not give a truthful representation of the users in a corporate network.

The validity of the data is ensured by asking four domain experts to generate the expert opinion prioritisation. This will allow for a comparison between the truths and allow for a use of a median truth to be used as the expert opinion. To ensure truthful statistical values in the comparison the prioritisation will be run 1000 times, after which the mean of the dependant variables, and the standard deviation, will be presented.

3.5 Dataset description

The dataset used in this experiment is collected by an external source. The contents of the dataset is described in table 3.1 and it contains the IP addresses for identification of the hosts present, in the network. As can be seen in the appendix, the hosts have been given names, *HostA - Q*, which corresponds to unique IP addresses in the network, to make it more readable. This is to disguise the data and the location of the vulnerable hosts, and therefore minimise the risk of exploitation. The hosts were chosen based on convenience sampling, in the test network of a company developing an state-of-the-art vulnerability scanner, where user data was generated. Some of this data was gathered from a centralised environment, and some directly from the hosts. The host information can be seen in Appendix C

The vulnerabilities for each host are collected with their corresponding CVE numbers, CVSS scores and CVSS vectors. What type of operating system the hosts have, are also collected and is a mix of Linux and Windows operating systems in this dataset.

To identify users, their usernames are used and information of which hosts they have accounts on. Other information collected for users are the date when they last logged in, the frequency of their logins per host during the past year, if they have root or administrator privileges on a certain host. In the case of Windows, there is also information about if cached credentials are used and what the limit is for that host.

The host information are gathered by the vulnerability scanner and the user information are manually fetched. From this data, RCR and vulnerability count per host are calculated and saved as well to increase the performance.

The domain experts are given the dataset described in section 3.5 and will then make their own prioritisations independently. Their individual prioritisations are then collected in excel sheets, same for everyone. Together the domain experts have many years of experience in this field and have been in contact with many environments like this. The expert opinion is a mean of the prioritisation made by the different domain experts.

This dataset could be representative for a small company with some techsavy personnel that keep their hosts up to date, and some that lack the know how and understanding of the importance of updates. This could result in a network that has a lot of different operating systems depending on the need of the company and an

Type	Data	Source
TEXT	IP Address	External Source
TEXT	Operating System	External Source
TEXT	Hostname	External Source
TEXT	Last Login date	Manually Fetched
INTERGER	User Login frequency	Manually Fetched
BOOL	Admin	Manually Fetched
BOOL	Cached Credentials	Manually Fetched
TEXT	Username	Manually Fetched
TEXT	CVE	External Source
TEXT	CVSS	External Source
INTERGER	RCR	Calculated
INTERGER	Vulnerability Count Per Host	Calculated

Table 3.1: Description of the dataset, and the source of date.

immense difference between how well the security of different parts of the network is handled. This does not mean that the techsavy personnel do not have access to the non updated hosts, but it might not be their task to update them.

3.6 Implementation

The proposed method is implemented as a Python3[1] program, with an execution flow as presented in figure 3.1. The required data is collected from the network by the external source and then provided as input to the python3 program. The data input will filter out any informational findings, without a CVE. Then an analysis is run on the data.

To filter the collected data and be able to make an analysis, the data is sorted and stored in a database with the tables shown in figure 3.2. This structure was created with the implementation of the prioritisation in mind, as described in section 4.2. It is implemented with SQLite[2].

The result of the analysis is a prioritisation of which host to start remediation activities on first. It is visualised as a graph, see example in figure 3.3, which shows the highest risk user for the different hosts, the risk score based on the CVSS scores of the present vulnerabilities and the connections the hosts have because of the users. If no user data is provided, then the analysis is based solely on the CVSS scores, the user score is set to one and therefore the RCR becomes the CVSS. The prioritisation will vary between runs if there are multiple hosts with the same RCR score, since their position will be randomised among each other. This means that the prioritisation presented in chapter 5 is one of the possible prioritisations.

The visualisation graph of users and hosts are colour coded in order to make it easier for the viewer to get an overview of the current state of the risks in the network. The red arrows from the users are indicators of on which hosts that particular user is

[1] `https://docs.python.org/3/whatsnew/3.6.html`
[2] `https://www.sqlite.org/index.html`

a high risk. The hosts are also colour coded where each colour represents a category. Dark green is low risk, light green is medium risk, yellow is high risk and red is critical. This visualisation is made according to the prioritisation made with the proposed method.

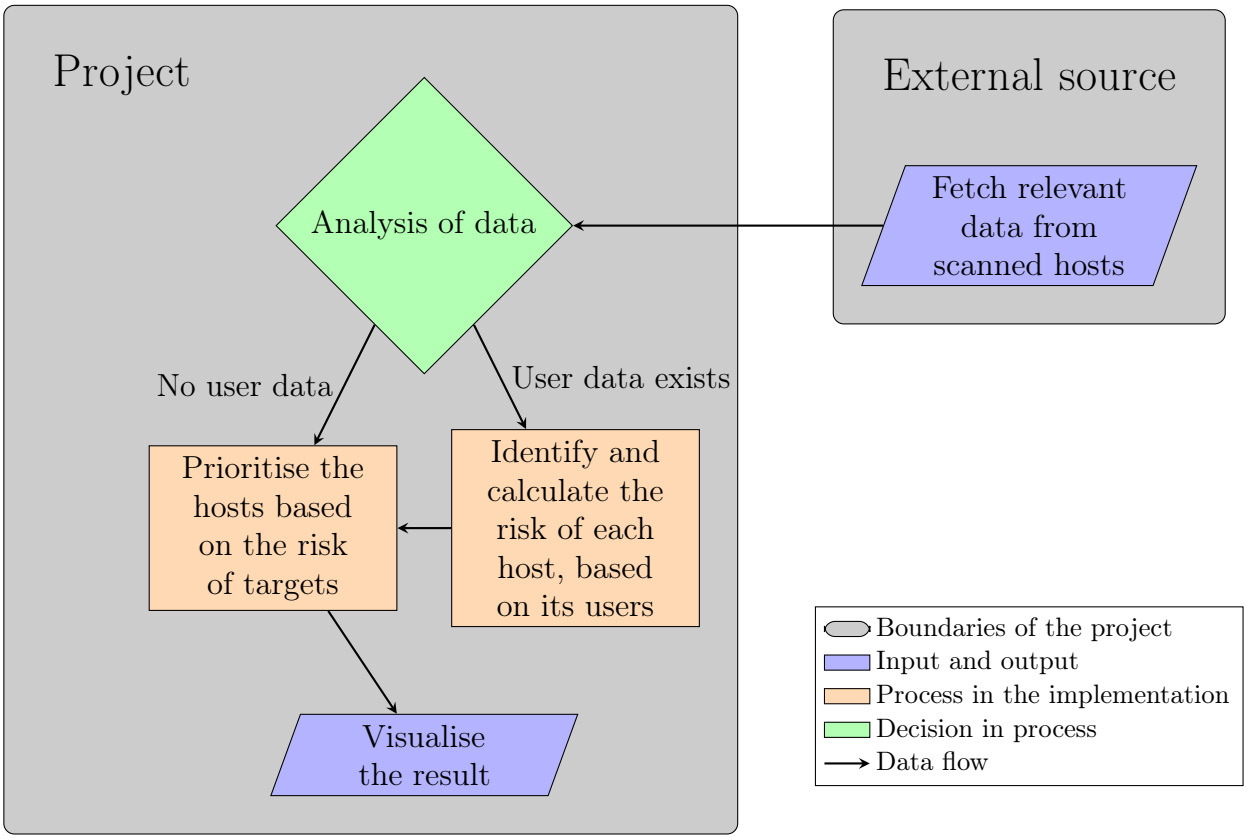

Figure 3.1: Flow chart for the implemented python3 program.

To calculate the statistics of the measuring points, they are built in to the RCR implementation. To implement the statistical functions, the library `sklearn.metrics` is used.

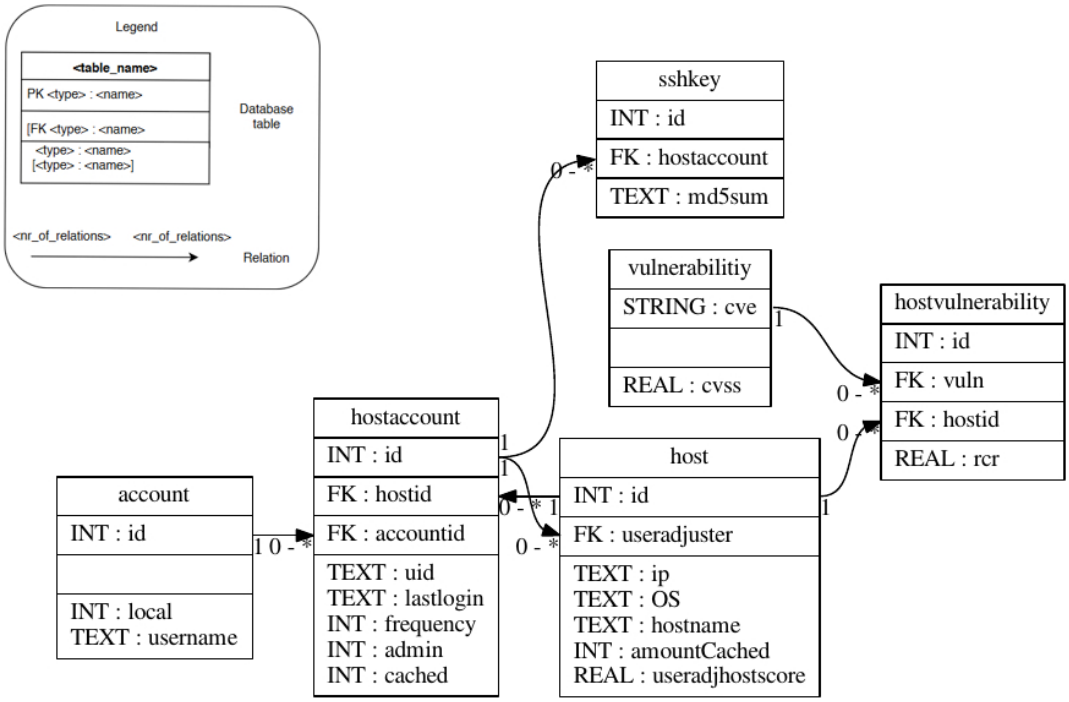

Figure 3.2: An entity relationship diagram of the database.

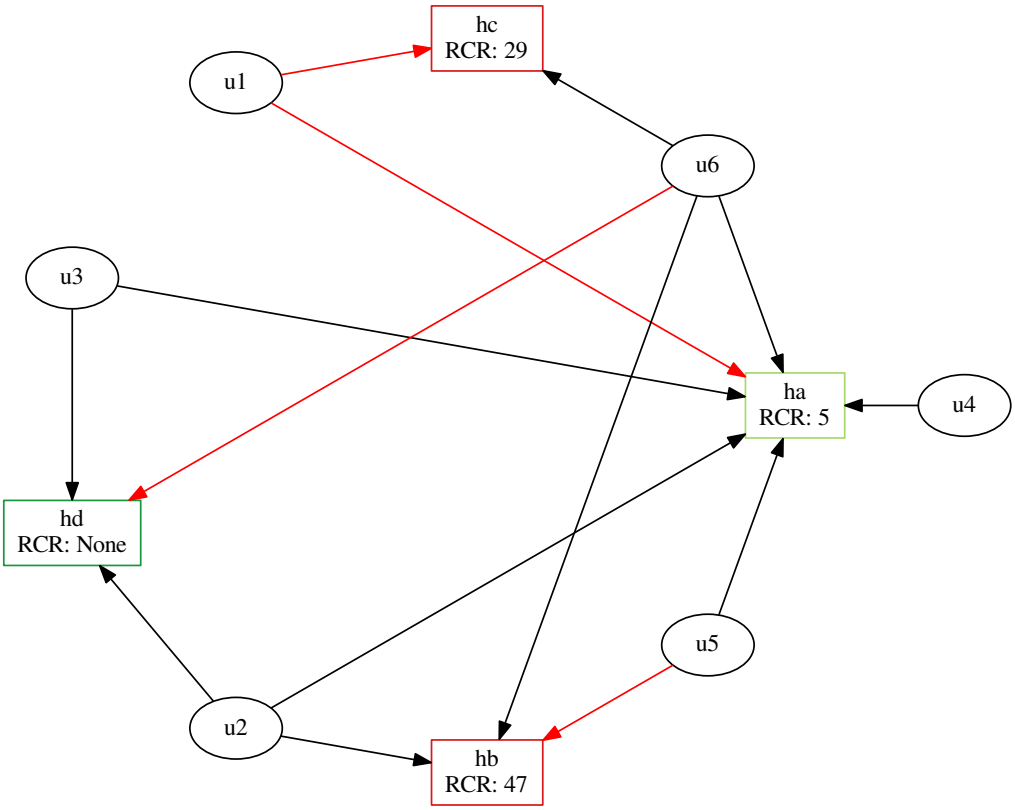

Figure 3.3: Example of the visualisation generated by the proof of concept implementation. Boxes are hosts and eclipses are users. The boxes are colour coded from red, highest risk, to green, loswest risk. Each arrow is a user that can access the host, where the red ones are the users that contributes to the UHS.

Chapter 4

Background study

This background study is based on the existing research within the problem domain and will provide information which is used to answer the research questions 1-3, posed in section 1.2. Lastly, a method for prioritising where in a network to apply remediation activities first is proposed, based on the background study.

4.1 User account correlation

In Windows, logon and logoff events are recorded to great extent, both for local machines and domain clients [12, 45]. The logs contain information about the type of logon event (remote, local etc), the success status and the authentication method (cached credentials or contact with the DC) [12]. Cached user credentials are used when a computer that belongs to a domain, does not have a connection to the DC. This information can be extracted from the security logs [12, 45], and used to track a user in a forensic investigation by identifying suspicious activity.

To track a user in Windows, whom has a domain account is a simple task if you have access to the DC. The DC and the local workstation or server with `account logon` and `logon/logoff` policies enabled [12], logs the logon events. By looking at these events one can track the users' activity in the network and on that machine.

For local accounts, the logon activity can be found on the local workstation [12, 45]. The system events generated by that user is also logged by the same workstation [12], and can therefore provide a good picture of what that user has done during the session. All events in windows has a specific event ID which can be used to track a specific user with a local account. By looking at outgoing connections to other workstations or machines, it can be possible assume that the tracked user has access to another local account in the network.

When it comes to correlating user accounts there is a lack of studies done for user accounts on hosts. The focus is rather on how accounts can be correlated between different web pages. This can be done with great success by only looking at user chosen usernames [31]. The study states that humans have a tendency to use identical, or similar, usernames on different platforms. This allows for accurate user correlation with minimum effort put in. This is not applicable to a corporate environment where the usernames are not chosen by the user, but by the administrator, but the conclusions about how humans chose usernames are still relevant.

Another way to find out where users have access in a network is to look for SSH keys. In order to authenticate a user remotely SSH keys can be used as an alternative for passwords. Since SSH is a Public Key Cryptography (PKC) [26], a key pair is

generated and then placed on the hosts that the user wants to authenticate against. This means that a public key can be found on multiple servers in the network to which a specific user has access to, except if the user account belongs to an Identity Management (IdM).

4.1.1 Risk assessment and categorisation of users

Studies have shown that a lot of threats to a corporate network consists of internal threats [8, 38, 40], so called *insiders*. Mostly, these are employees that are unhappy [49] with the company in some way or are about to be let go.

A user can pose as a security threat to a corporate multi-user network, by both mistake and if there is malicious intent but this is none the less based on end user behaviour [8, 40, 49]. It is one thing to intentionally change the configuration of network equipment as a Denial of Service (DoS) attack, and another to mistakenly change it so that the up-time is affected negatively. Even if both mistakes and malicious intent are risks to the corporate network, one is more serious than the other.

By analysing a user's habits and its intentions [38], the user can be categorised as different levels of security threats to a network or organisation [40]. In a theoretical study [40], a categorisation has been created were users are divided based on levels of intent. These categories are malicious, neutral and beneficial, and the level of technical expertise is high or low. The reasoning is that a user with high expertise and malicious intent is a bigger threat to the corporate network than a user with neutral intent and low technical expertise.

There exists also a proposed theoretical framework [38], for detecting insider attacks before they happen by analysing user activity. This framework is a combination of HIDS and NIDS in order to track a user in the network. The framework considers different aspects of a user's behaviour before determining if there is a possible insider or not. For example, it looks at how access is used in the system by the user, or if the user is trying to extract as much data as possible from a vulnerable host in the network as preparations as well as social aspects like personal traits and hostility in language. In order to trigger an alert that a specific user can be an insider, there must exist more than one sign in the activity.

Another recent study have shown that it is possible to profile users by looking at their behaviour by making a so called User Behaviour Profile (UBP) [28]. To do this, a baseline is first established [49] with what is considered *normal* activity for a specific user or role. However, this baseline can be altered by a malicious insider by slowly changing his or her activity pattern over time [28, 49]. Profiling of a specific user is called Personal Adaptive User Behaviour Profile (PAUBP) and profiling for different roles, contain patterns for multiple users is called General User Behaviour Profile (GUBP). Then comparisons can be made to the current activity in order to find anomalous behaviour. Experiments shows that PAUBP more effectively can detect abnormal user activity than GUBP [28].

4.1.2 Risk assessment and categorisation of hosts

A target system can be classified on accessibility [27] based on an analysis of the requests sent to and from the target system. The analysis will help determine if it is a normal request or an abnormal request based on who the user is, the user's role, the type of request and if present, the Personal Adaptive User Behaviour Profile (PAUBP) for that specific user. The calculations of the classification does take into account mistakes made by a user and therefore calculates a threshold for the amount of abnormal requests [27]. This threshold is dynamically updated based on the requests, and eliminating manual updating of a fixed value for the threshold over time.

In an another recent study there is proposed risk assessment of hosts, based on network flow and inter-dependency between hosts [35]. This method takes into account not only the host itself but the neighbouring hosts (based on network separation), the source and destination of the data flow in the network [35]. The method was tested for two public datasets, Honeynet and MIT Lincon, to measure effectiveness to the existing approach explained in [47]. Both methods found the same risk elements in the network but the new proposed method [35] turned out to be more efficient. There are mathematical formulas for calculating the risk scored for both hosts and the network flow published in later research by the same authors [36].

These methods have a deep dependency on the dataflow in the network, to define a high risk host. To gather this data in a efficient way is quite hard, and will require extensive data about what communications are transmitted. This data requires extra logging capabilities, and cannot be extracted from a ordinary host with a default configuration.

Vulnerability scanning allows to find many but not all vulnerabilities on hosts in a network [13] and is effective for multiple platforms. This information can then be used to calculate a vulnerability score for a certain host, based on the CVSS scores of the found vulnerabilities. To get a more accurate scoring for a certain host, the neighbouring vulnerability scores should have some influence [43]. This means that if neighbouring hosts have high vulnerability scores, this will have an impact on its neighbour hosts as well.

4.1.3 Summary of background study

In a modern enterprise environment an IdM system has a core function for managing users [50]. This could be one reason that there are a lack of research in the area of user correlation, because if all users are managed by the IdM system, there is no need to correlate them. The proposed method, will therefore, only be based on the trivial same username correlation, in combination with correlation of public keys in an SSH based environment.

The background study on risk assessment and categorisation of hosts has shown that there are a lot of aspects to take into account when categorising a high risk user. Many of these aspects are of a non technical nature, such as intent or knowledge. The proposed method for assessing the risk of a user will take into account technical aspects, such as to which hosts the user has access, what kind of access the user has (privileged or regular), with what frequency the user access resources and if the host has cached credentials in Windows.

The vulnerabilities and overall activity that connects the hosts, both user and network activity, contributes to the risk a host poses to the network. To identify a host as a risk one needs to consider as many of these factors as possible.

4.2 Proposed analysis method

Based on the results of the background study a method is proposed for prioritising what vulnerabilities should be patched first. The method is heavily inspired by the method proposed by C. Suh-Lee and J. Jo in Quantifying security risk by measuring network risk conditions [43], but it is modified so that the definition of what a neighbouring host depends on user access instead of network hops. The result of this proposed method will still be a RCR for a vulnerability on a host. The RCR, AVS, and $NAVS$ value during the calculations, are not confined, and can therefore only be compared within an unchanged network. Since this method is aimed to be automated, it will stick to analysing the technical information only. This allows utilisation without manual monitoring of human behaviour.

4.2.1 Risk calculations for hosts, and users

Each user will first be categorised depending on the calculated risk they pose to the system *low*, *medium*, *high* or *critical*. These are the four possible categories for a user and each of them gives the user a certain risk value according to table 4.1. The reason these values were chosen was because the lowest value could not be 0, since division by 0 is undefined and the higher risks should be represented with increasingly higher values, representing higher risks. The binary system is logical and provides the sought increase in value for each category representing a higher risk. The scale of the user categorisation is from 1 to 8.

The users access in the network will be seen as a subset of hosts in the network and be called *Host User Set* (HUS_u).

Risk Value (R_v)	**User Type**	**Risk Level**
2^0	Normal user	Low
2^1	Normal user and cached credentials exist	Medium
2^2	If root/admin	High
2^3	If root/admin and cached credentials exist	Critical

Table 4.1: Categorisation for users as risks in a network.

For each user (U) in the network a *User Score* (US) is calculated for a certain host (H), that relates to the frequency (f) of that users logon activity for that host, the number of users that has access to that host (H_u) and how many hosts the user has access to (U_h). The US is calculated with the following equation

$$US(f, H_u, U_h, R_v) = \frac{1}{f \cdot H_u \cdot U_h \cdot R_v}. \tag{4.1}$$

The interval for the US is $0 < x <= 1$, where close to 0 is the highest risk and 1 is the lowest risk.

To find the user that poses the highest risk to a host (4.1) is used when calculating an *User Adjusted Host Score (UHS)* with

$$UHS(H) = min(US_i) \tag{4.2}$$

which is finding the user with the highest risk for that host based on the *US* from equation 4.1.

Each host also has a number of vulnerabilities that poses a risk to the network. In order to combine the vulnerabilities with the risk of the users an *Adjusted Vulnerability Score* (*AVS*) is calculated based on the CVSS scores of each vulnerability (V) on the host (H),

$$AVS_{vuln}(H, V) = \frac{1}{UHS(H)} \cdot CVSS(V). \tag{4.3}$$

If the *UHS* for a vulnerability is 1, then equation 4.3 is 1 multiplied by the CVSS for the vulnerability. This means that the interval for equation 4.3 is larger than or equal to the provided CVSS score.

The vulnerability score can then be calculated for the entire host by summing up the Adjusted Vulnerability Score (4.3) for each vulnerability on the host H,

$$AVS_{host}(H) = \sum_{i=1}^{n} AVS_{vuln}(H, V_i). \tag{4.4}$$

If the host has no vulnerabilities, the sum for equation 4.4 will be 0. The upper limit for the sum is infinity.

To calculate the vulnerability score for an entire subset of hosts to which a user has access in the network, the Adjusted Vulnerability Score (4.4) is summed up for each host in the set,

$$AVS_{subset}(HUS_u) = \sum_{i=1}^{n} AVS_{host}(H_i). \tag{4.5}$$

The same interval as used for equation 4.4, is used for equation 4.5, if the hosts have no vulnerabilities the sum will be 0 and the upper limit is the infinity.

Then a *Neighbour Adjusted Vulnerability Score* (*NAVS*) can be calculated for a certain vulnerability on a host, in order to know how much the neighbours effect the current host. To only get the influence score from the neighbours, the vulnerability score for the current host is removed (4.3), from the subset (4.5) to which the user has access. This means that each vulnerability on a host where a user is shared with the current host affects the score,

$$NAVS(HUS_u, H, V) = \sum_{i=1}^{n} \left\{ AVS_{subset}(HUS_i) - AVS_{vuln}(H, V) \right\}. \tag{4.6}$$

Even other vulnerabilities on the current host affects the *NAVS*.

The interval for *NAVS* is decided by the vulnerabilities of the neighbouring hosts. If there is no vulnerable hosts in the subset that a user has access to then the sum will be 0. Here too, the upper limit is the infinity.

Then a Relative Cumulative Risk (RCR) value can be calculated for the vulnerabilities on a certain host by multiplying equation 4.6 and equation 4.3,

$$RCR(H,V) = AVS_{vuln}(H,V) \cdot NAVS(H,V). \tag{4.7}$$

The interval for RCR is 0 to infinity. This is because if there is no vulnerable hosts in the subset that a user has access to, then the RCR will be low and if there are a lot of vulnerabilities in the subset, the RCR gets higher.

To identify one host as the highest risk in the network, the RCR for the host can be calculated with,

$$RCR_{host}(H) = \sum_{i=1}^{n} RCR(H,V_i) \tag{4.8}$$

The RCR for a host is dependent on the RCR scores for the vulnerabilities. This interval is also from 0 to infinity. The lower the value is, the lower risk the host poses to the rest of the network.

All the new definitions presented for the mathematical equations in this section are listed in table 4.2.

Term	**Definitions**
R_v	Risk value for a user
H_u	Number of users who has access to a certain host
U_h	Number of hosts the user has access to in the network
V	A vulnerability on a host
H	A host
US	User Score
UHS	User adjusted Host Score
HUS	Host User Subset
AVS	Adjusted Vulnerability Score
$NAVS$	Neighbour Adjusted Vulnerability Score

Table 4.2: Definitions of the terms used for the mathematical calculations.

Chapter 5

Results

This chapter presents the results from both the pilot experiment and the main experiment, as well as how the prioritisation of the different methods, compares to the expert opinion, and their improvement (Cohen's d). Lastly a small section about the performance of the proposed method compared to prioritising by hand is presented.

The expert opinion set by the domain experts, can be seen in table 5.1 (pilot experiment) and 5.2 (main experiment). In the same tables the prioritisation used as expert opinion is shown, in the last column. The expert opinion is a mean of the prioritisation made by the different domain experts.

During the experiment, some of the experts chose to leave the study. This means that out of the four voluntary experts only three, including one pilot, fulfilled the task. The expert opinion was created based on the results from the experts that chose to finish the experiment. The expert opinion was then tested against the three test cases below.

5.1 State-of-the-art method (SOTA)

The resulting prioritisation and how it compares to the expert opinion for the pilot experiment is presented in the SOTA column of table 5.1 and table 5.3. The prioritisation and comparison for the main experiment in presented in table 5.2 respectively table 5.4.

The standard deviation for this method is 0.0 which means that the runs are exactly the same each time. When it come accuracy, this method has a low score compared to the other method. It is also unchanged between the pilot experiment and the main experiment which means that it has the same accuracy undependently of the network. The Cohen's kappa for the main experiment is approximately 0.5, which means that the prioritisation is weak in comparison to the expert opinion. The RMSE value is relatively high compared to the third test case, which means that the predictions for this test case have more errors than the proposed method compared to the expert opinion.

5.2 Proposed method without user data (PMWUD)

How this test case measures to the expert opinion is presented in the PMWUD column of table 5.3 for the pilot experiment and table 5.4 for the main experiment.

The standard deviation for the proposed method without user data, is low except for the RMSE. This means that the results of the different runs are not very far from each other, but that the errors from the expert opinion are relatively many. Despite this, the accuracy of this method is still low. The Cohen's kappa are marginally higher than for the SOTAs prioritisation in the main experiment, which means that it is a weak comparison [41] to the expert opinion. The RMSE for this method is almost the same as for the SOTA, which means that the prediction contains just as many errors compared to the expert opinion.

5.3 Proposed method (PM)

The prioritisation from this test case is presented in the PM column of table 5.1 for the pilot experiment and table 5.2 for the main experiment.

How this method measures toward the expert opinion is presented in table 5.3 for the pilot experiment and table 5.4 for the main experiment.

The standard deviation for all but the RMSE is very low for this method compared to the other test cases. This indicates that the results from the different runs does not differ a lot. As shown in table 5.4 for the main experiment, this method has a significantly higher accuracy than the other two methods. The Cohen's kappa value is approximately 0.79, which translates to moderate [41], almost strong, compared to the expert opinion. The RMSE value is lower than for the other methods which indicates that the errors are fewer compared to the expert opinion, than for the other methods.

Priority	SOTA	PMWUD	PM	Expert Opinion
1	Host M	Host M	Host M	Host M
2	Host I	Host L	Host N	Host L
3	Host L	Host I	Host H	Host N
4	Host H	Host H	Host P	Host P
5	Host P	Host P	Host L	Host Q
6	Host N	Host N	Host F	Host A
7	Host J	Host J	Host Q	Host H
8	Host O	Host O	Host I	Host I
9	Host A	Host A	Host A	Host J
10	Host Q	Host C	Host J	Host O
11	Host C	Host Q	Host O	Host B
12	Host F	Host F	Host B	Host C
13	Host B	Host B	Host K	Host K
14	Host K	Host K	Host C	Host G
15	Host G	Host G	Host G	Host F
16	Host D	Host D	Host D	Host E
17	Host E	Host E	Host E	Host D

Table 5.1: The prioritisations for the pilot experiment. The first dataset was used for this experiment.

Priority	SOTA	PMWUD	PM	Expert 1	Expert 2	Expert Opinion
1	Host M	Host M	Host M	Host M	Host Q	Host P
2	Host L	Host L	Host H	Host P	Host P	Host N
3	Host I	Host I	Host P	Host L	Host N	Host Q
4	Host H	Host H	Host L	Host J	Host H	Host L
5	Host P	Host P	Host F	Host A	Host O	Host H
6	Host J	Host J	Host A	Host H	Host K	Host M
7	Host A	Host A	Host Q	Host I	Host I	Host O
8	Host Q	Host C	Host J	Host Q	Host L	Host I
9	Host C	Host Q	Host I	Host G	Host J	Host J
10	Host K	Host F	Host B	Host K	Host M	Host A
11	Host F	Host B	Host K	Host C	Host F	Host K
12	Host O	Host K	Host C	Host B	Host A	Host C
13	Host B	Host O	Host G	Host F	Host C	Host B
14	Host D	Host G	Host O	Host O	Host B	Host F
15	Host E	Host D	Host D	Host N	Host E	Host G
16	Host G	Host E	Host E	Host E	Host G	Host E
17	Host N	Host N	Host N	Host D	Host D	Host D

Table 5.2: The prioritisations for the main experiment. The domain experts prioritisations are presented as well in separate columns, these are what the expert opinion is based on. The second dataset was used for this experiment.

The improvement of all the different scores is large (see table 5.5), with an absolute Cohen's d over 5 for all values in the main experiment.

5.4 Other Observations

These results were run with the proof of concept tool built for this project in order to automatise the prioritisation process. The analysis of the dataset and statistical calculations made 1000 times, for the main experiment took 173.7 seconds and the manual labour of the same analysis was at least 5 hours.

Measurement	SOTA	PMWUD	PM
Accuracy:			
Mean	0.05882	0.17553	0.29494
Standard deviation	0.0	0.05885	0.05885
Cohen's kappa:			
Mean	0.85784	0.85660	0.84193
Standard deviation	0.0	0.00123	0.00123
RMSE:			
Mean	28.353	50.413	27.352
Standard deviation	0.0	0.05885	0.05885

Table 5.3: The measurements between expert opinion and the different test cases for the pilot experiment, when run 1000 times.

Measurement	SOTA	PMWUD	PM
Accuracy:			
Mean	0.058824	0.038235	0.21629
Standard deviation	0.0	0.043943	0.043275
Cohen's kappa:			
Mean	0.51716	0.51209	0.79244
Standard deviation	0.0	0.028956	0.028608
RMSE:			
Mean	50.353	51.110	22.889
Standard deviation	0.0	1.3951	1.3811

Table 5.4: The measurements between the expert opinion and the different test cases for the main experiment, when run 1000 times.

Measurement	Cohen's d
Accuracy	5.1459
Cohen's kappa	13.608
RMSE	-28.122

Table 5.5: The effect size of the main experiment when comparing state-of-the-art method with the proposed method.

Chapter 6

Analysis and Discussion

The first part of the background study provides an answer to RQ1, which was how user accounts can be correlated between hosts. The most common way to correlate user accounts across a large network is to use some sort of centralised authentication. This allows the system administrator to manage user accounts and set permissions for different hosts depending on what tasks they should be allowed to perform in the network. Since this is the case a trivial method is used for correlating local accounts, videlicet to see if the username match for different user accounts.

Due to this increased use of centralised authentication, the interest for correlating local user in an automated fashion seems to have decreased. With the lack of automated tools, the job falls back to a manual process where someone has to sit down and manually correlate the user accounts. This means that someone needs to check if the accounts correlate, with the help of the available data, and if needed, ask the users.

To answer RQ2, how a high risk user can be defined, the previous answer has to be taken into account of how to correlate user accounts on different hosts. Since it is possible to track a user's activity in a corporate network, it is also possible to detect diverting activity which can be malicious. This type of activity can be trying to access more resources than allowed, trying to run unauthorised commands or verbal indicators in written or spoken form. The risk can then be calculated in order to give the user a risk score and the behaviour can be categorised depending on the user's behaviour.

It is common that a user who is unhappy with his or her employeer or employment in some way, pose a higher risk to the corporate network altogether than a person who likes their job. This is because disgruntled employees often do not care as much about protecting the company's resources, since they have no attachment to the company. If the dissatisfied employee has malicious intent, then the damage could be even worse from the company's point of view. Sensitive information can be sold or outsiders can be let in to the corporate network, disguised as legitimate users.

The last part of the background study answers RQ3, how to define a high risk host. The risk a host poses in a network depends on some different factors. It depends on the network flow, vulnerabilities and how the host is accessed and by which users, not just for the current host but also for neighbouring hosts has to be taken into account.

In order to calculate a risk score for a host, one can use a combination of the CVSS score for the current host, the CVSS score for the neighbouring hosts and the calculated risk for the users accessing that host. This is the proposed method which was then compared against the state-of-the-art method.

The results from the main experiment, which is also the answer to RQ4 on how the proposed method compares to the SOTA method, shows an enhancement of all measurements as can be seen in table 5.4.If the analysis is made without any user data, the proposed method performs about as well as the state-of-the-art method, which shows the stability of the proposed method. It will not fail to deliver a usable prioritisation if no user data was gathered.

The results from the pilot experiment does not show the same degree of certainty for the proposed method, but the tendencies are there. For example, the RMSE and Cohen's kappa for the proposed method and the SOTA are almost the same but the accuracy for the proposed method is still significantly higher than for he SOTA, as shown in table 5.3. This might be an affect of the change in how the data was presented to the domain experts since the domain expert for the pilot has some additional data requests for the dataset. It can also have been affected by a slight change that occurred between the experiments in the form of patched hosts for example.

In both the experiments, as seen in table 5.3 and table 5.4, the standard deviation is the same for the combinations of accuracy, Cohen's kappa, PMWUD and PM. This is most likely due to the dataset, that contains a couple of hosts that are prioritised with the same importance with both methods.

The improvements of the main experiment are overall significant, as shown by the Cohens's d in table 5.5. This means that the results has a statistical certainty. This does not however mean that there are not room for further improvement.

Another factor to note is that the standard deviation of the RMSE in the main experiment is high in relation to the other measurements. This tells us that there are a noticeable variation between multiple runs of the same data. Considering the size of the RMSE the standard deviation value is not high enough to be concerned about, but if the method will be improved to take more factors into account in the future it is something to monitor.

The proposed method adds to the risk value for a host if it has a lot of neighbouring connections in the form of users whereas the state-of-the-art method will still only count the number of vulnerabilities. This means that another host with more high risk users could be a higher prioritisation than the host with the most vulnerabilities, that next to no one is using, with the proposed method. The experiment conducted shows that this is indeed an enhancement over the current state-of-the-art method, see tables 5.4 and 5.5.

Network administrators who are getting a prioritisation created by the proposed method instead of the current state-of-the-art method, will have an easier time improving the overall security of their network. The resulting prioritisation recommends which host to start remediation activities to and the visualisation shows high risk users for the hosts in the network. This can be used to improve user access as well in the corporate network.

For the network administrators who were not satisfied with the reports created based on the SOTA method, and remade the prioritisation, will probably not gain as much in prioritisation improvement from using the proposed method. They will however, save time in making the new prioritisation by using the implementation of the proposed method. Instead of spending hours on prioritising a small network, it will take minutes with the proposed method and still get an accurate prioritisation.

The SOTA takes a few minutes but provides a less accurate prioritisation as the experiment shows in table 5.4.

Chapter 7

Conclusions and Future Work

If a user account with high privileges throughout the network is compromised, then an attacker has the same level of access and makes that user account a higher risk to the network then a user account which has access to one machine and no administration or super user rights. By evaluating the risks that each user conveys to a network, vulnerable hosts can be identified as well and should have impact on the final risk classification and prioritisation recommendations for vulnerability remediation activities. A new method has been proposed, that is a large improvement over the current-state-of-the-art method. The Cohen's d are 5.15 for accuracy, 13.6 for Cohen's Kappa and -28.1 for Root Mean Square Error (RMSE), which indicates large improvement.

This thesis shows that there are possibilities to improve the recommendations for administrators from a vulnerability scanner on the area of prioritising hosts to apply remedies to. It proposes a method, that uses user data, to improve the recommended prioritisation for remediation activities. When tested against the expert opinion based on the prioritisations supplied by domain experts, it shows that the proposed method performs significantly better and gives a more accurate result.

The reason that the proposed method preforms better is that more factors are taken into account during the analysis. The state-of the-art-method is very limited and does only take vulnerability data into account. To improve the recommendations of remediation application further, research is needed in the area of what does affect the risk of a host, and how this can be used in the recommendations.

The background study shows some other possibilities for analysing users, hosts, and risks. These where rejected during this study, but they might be meaningful to investigate more in depth when making further improvements to the proposed method.

7.1 Future work

As a continuation of this project there are several directions that a future study can take. The first and most obvious one is to investigate if the risk values used in the proposed method can be changed to further improve the prioritisation, Another direction would be to investigate if the proposed method can be combined with other key points to further improve the RCR and therefore also the prioritisation. Some key points to investigate could be the network flow, as described in the original article "Quantifying security risk by measuring network risk conditions" by Suh-Lee and Jo, or to look at the availability of exploits for the vulnerabilities. This might be what is

needed to gain the impact of the proposed method to gain a statistical certainty to recommend the proposed method.

On the other hand, future research might focus on the data gathering and user correlation parts. More concretely there is a gap in the research about how one can correlate user accounts between different users in the network. This could be either between different hosts with the same operating system, between different operating systems, or even between services on the hosts. Ways to do this could be by mining data for similar use, as proposed by "Clustering by compression" by Cilibrasi and Vitányi or by looking at similar usernames as proposed by Perito, Castelluccia, Kaafar and Manils in "How Unique and Traceable Are Usernames?". The question is what can be gained without compromising to musch in compute time, and how can the problem with shared user accounts be solved?

Another suggestion is to investigate the impact one vulnerability has for an entire network. If you have to patch one vulnerability in the entire network, which should you choose to reduce the risk of getting hacked?

As a smaller project it might be interesting to investigate the properties of RCR, such as how it handles when comparing two scans on the same network. What would happen if a host was added? Are the scores comparable? This would help to broadened the understanding and might make it possible to modify the RCR so that it becomes confined and thereby be used to compare hosts on different networks.

References

[1] Asmaa Shaker Ashoor and Sharad Gore. Intrusion detection system (ids): case study. In *Proceedings of 2011 International Conference on Advanced Materials Engineering (ICAME 2011)*, 2011.

[2] Dr. Lee A. Becker. Efect size calculators, 2000. Accessed: 2018-05-15. URL: `https://www.uccs.edu/lbecker/effect-size`.

[3] Florian Bergsma, Benjamin Dowling, Florian Kohlar, Jörg Schwenk, and Douglas Stebila. Multi-ciphersuite security of the secure shell (ssh) protocol. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 369–381, New York, NY, USA, 2014. ACM. URL: `http://doi.acm.org.miman.bib.bth.se/10.1145/2660267.2660286`, `doi:10.1145/2660267.2660286`.

[4] Boulat Chainourov. *Log analysis using Splunk Hadoop Connect*. PhD thesis, Naval Postgradeuate School, 2017. URL: `https://calhoun.nps.edu/handle/10945/55581`.

[5] Kerberos Consortium. The role of kerberos in modern information systems, 2008. Accessed: 2018-02-12. URL: `http://kerberos.org/software/rolekerberos.pdf`.

[6] Kerberos Consortium. Documentation, 2013. Accessed: 2018-02-16. URL: `http://kerberos.org/docs/index.html`.

[7] Kerberos Consortium. Mit kerberos documentation, 2018. Accessed: 2018-02-16. URL: `http://web.mit.edu/kerberos/krb5-current/doc/`.

[8] John D'Arcy and Anat Hovav. Deterring internal information systems misuse. *Commun. ACM*, 50(10):113–117, October 2007. URL: `http://doi.acm.org.miman.bib.bth.se/10.1145/1290958.1290971`, `doi:10.1145/1290958.1290971`.

[9] Brian Desmond, Joe Richards, Robbie Allen, and Alistair G. Lowe-Norris. *Active Directory: Designing, Deploying, and Running Active Directory*. O'Reilly, Sebastopol, 5th;5; edition, 2013.

[10] Marie Doleželová, Mirek Jahoda, Maxim Svistunov, Stephen Wadeley, Tomáš Čapek, Robert Krátký, Jana Heves, Jaromír Hradílek, Douglas Silas, Barbora Ančincová, Petr Kovář, Jiří Herrmann, Peter Ondrejka, Petr Bokoč,

Martin Prpič, and Eva Majoršinová. Managing users and groups, 2018. Accessed: 2018-02-22. URL: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/deployment_guide/ch-managing_users_and_groups.

[11] Oana Goga, Howard Lei, Sree Hari Krishnan Parthasarathi, Gerald Friedland, Robin Sommer, and Renata Teixeira. Exploiting innocuous activity for correlating users across sites. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 447–458, New York, NY, USA, 2013. ACM. URL: http://doi.acm.org.miman.bib.bth.se/10.1145/2488388.2488428, doi:10.1145/2488388.2488428.

[12] Sunil Gupta. *Windows Logon Forensics*. SANS Institute InfoSec Reading Room, 2013. Accessed: 2018-03-05. URL: https://www.sans.org/reading-room/whitepapers/forensics/windows-logon-forensics-34132.

[13] Hannes Holm, Teodor Sommestad, Jonas Almroth, Mats Persson, Industriella informations-och styrsystem, Skolan för elektro-och systemteknik (EES), and KTH. A quantitative evaluation of vulnerability scanning. *Information Management & Computer Security*, 19(4):231–247, 2011.

[14] Muhammad Z. Hussain, Muhammad Z. Hasan, and Muhammad Taimoor Aamer Chughtai. Penetration testing in system administration. *International Journal of Scientific & Technology Research*, 6(6):275–278, 2017.

[15] Red Hat Inc. Red hat enterprise linux 5 deployment guide, 2013. Accessed: 2018-02-16. URL: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/pdf/deployment_guide/Red_Hat_Enterprise_Linux-5-Deployment_Guide-en-US.pdf.

[16] Collins Chandi Kimathi. *A Platform for monitoring of security and audit events: a test case with windows systems*. PhD thesis, Strathmore University, 2017. Accessed: 2018-02-20. URL: http://su-plus.strathmore.edu/handle/11071/5615.

[17] Brian Lich and Justin Hall. Threat protection, 2018. Accessed: 2018-02-16. URL: https://docs.microsoft.com/en-us/windows/security/threat-protection/.

[18] B. Malek and A. Miri. Combining attribute-based and access systems. In *2009 International Conference on Computational Science and Engineering*, volume 3, pages 305–312, Aug 2009.

[19] Bill Mathers, Sudeep Kumar, and Corey Plett. Active directory domain services overview, 2017. Accessed: 2018-03-19. URL: https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview.

[20] Microsoft. Technet library, 2017. Accessed: 2017-12-05. URL: https://technet.microsoft.com/en-us/library.

[21] Microsoft. Microsoft api and reference catalog, 2018. Accessed: 2018-02-12. URL: https://msdn.microsoft.com/en-us/library.

[22] Microsoft. Powershell, 2018. Accessed: 2018-02-05. URL: https://docs.microsoft.com/en-us/powershell/scripting/powershell-scripting?view=powershell-5.1#getting-started-with-powershellgetting-startedgetting-started-with-windows-powershellmd.

[23] Microsoft. Powershell, 2018. Accessed: 2018-02-12. URL: https://docs.microsoft.com/en-us/powershell/.

[24] Microsoft. Technet library, 2018. Accessed: 2018-02-12. URL: https://technet.microsoft.com/en-us/library.

[25] Daesung Moon, Sung Bum Pan, and Ikkyun Kim. Host-based intrusion detection system for secure human-centric computing. *The Journal of Supercomputing*, 72(7):2520–2536, Jul 2016. URL: https://doi.org/10.1007/s11227-015-1506-9, doi:10.1007/s11227-015-1506-9.

[26] Shireen Nisha and Mohammed Farik. Rsa public key cryptography algorithm a review. *International Journal of Scientific & Technology Research*, 6(7):187–191, 2017.

[27] P. A. Osipov, L. Ya. Aleksejeva, A. N. Borisov, Yu. A. Chizhov, T. P. Zmanovska, and V. M. Zabiniako. Implementation and operation aspects of a system for detecting abnormally level of user activity. *Automatic Control and Computer Sciences*, 51(6):417–425, Nov 2017. URL: https://doi.org/10.3103/S0146411617060050, doi:10.3103/S0146411617060050.

[28] P. A. Osipov, A. E. Mrochko, and A. N. Borisov. Identification of differences of user behavior profiles and user class templates. *Automatic Control and Computer Sciences*, 48(2):65–79, Mar 2014. URL: https://doi.org/10.3103/S0146411614020072, doi:10.3103/S0146411614020072.

[29] Fedora Pagure. Sssd - security system service daemon, 2017. Accessed: 2018-02-16. URL: https://pagure.io/docs/SSSD/sssd/.

[30] Matthew Pemble. Evolutionary trends in bank customer-targeted malware. *Network Security*, 2005(10):4 – 7, 2005. URL: http://www.sciencedirect.com/science/article/pii/S1353485805702889, doi:https://doi.org/10.1016/S1353-4858(05)70288-9.

[31] Daniele Perito, Claude Castelluccia, Mohamed Ali Kaafar, and Pere Manils. How unique and traceable are usernames? In Simone Fischer-Hübner and Nicholas Hopper, editors, *Privacy Enhancing Technologies*, pages 1–17, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[32] Fedora project. Fedora documentation. Accessed: 2017-12-05. URL: https://docs.fedoraproject.org/.

[33] The OpenLDAP Project. Openldap software 2.4 administrator's guide, 2011. Accessed: 2018-02-16. URL: https://www.openldap.org/doc/admin24/.

[34] Chet Ramey and Brian Fox. Bash reference manual, 2016. Accessed: 2018-03-19. URL: https://www.gnu.org/software/bash/manual/bash.pdf.

[35] M. Rezvani, A. Ignjatovic, E. Bertino, and S. Jha. Provenance-aware security risk analysis for hosts and network flows. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–8, May 2014. doi:10.1109/NOMS.2014.6838250.

[36] M. Rezvani, V. Sekulic, A. Ignjatovic, E. Bertino, and S. Jha. Interdependent security risk analysis of hosts and flows. *IEEE Transactions on Information Forensics and Security*, 10(11):2325–2339, Nov 2015. doi:10.1109/TIFS.2015.2455414.

[37] Arindam Roy, Shamik Sural, Arun Kumar Majumdar, Jaideep Vaidya, and Vijayalakshmi Atluri. On optimal employee assignment in constrained role-based access control systems. *ACM Trans. Manage. Inf. Syst.*, 7(4):10:1–10:24, December 2016. URL: http://doi.acm.org.miman.bib.bth.se/10.1145/2996470, doi:10.1145/2996470.

[38] E.Eugene Schultz. A framework for understanding and predicting insider attacks. *Computers & Security*, 21(6):526 – 531, 2002. URL: http://www.sciencedirect.com/science/article/pii/S016740480201009X, doi:https://doi.org/10.1016/S0167-4048(02)01009-X.

[39] Randy Franklin Smith. Get answers to your windows security questions. *Windows IT Security*, 5(5):1 – 16, 2005. URL: http://miman.bib.bth.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=17349763&site=ehost-live.

[40] Jeffrey M. Stanton, Kathryn R. Stam, Paul Mastrangelo, and Jeffrey Jolton. Analysis of end user security behaviors. *Computers & Security*, 24(2):124 – 133, 2005. URL: http://www.sciencedirect.com/science/article/pii/S0167404804001841, doi:https://doi.org/10.1016/j.cose.2004.07.001.

[41] Bömches B Greis C Solleder P. Steinijans VW, Diletti E. Interrater reliability: the kappa statistic, 2012. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3900052/.

[42] Ellen J. Stokes, Russel F. Weiser, Ryan D. Moats, and Richard V. Huber. Lightweight directory access protocol (version 3) replication requirements. RFC 3384, RFC Editor, October 2002.

[43] C. Suh-Lee and J. Jo. Quantifying security risk by measuring network risk conditions. In *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*, pages 9–14, June 2015. doi:10.1109/ICIS.2015.7166562.

[44] Windows Support. Cached domain logon information, 2017. Accessed: 2018-03-09. URL: https://support.microsoft.com/en-us/help/172931/cached-domain-logon-information.

[45] Javad Talebi, Ali Dehghantanha, and Ramlan Mahmoud. Introducing and analysis of the windows 8 event log for forensic purposes. In Utpal Garain and Faisal Shafait, editors, *Computational Forensics*, pages 145–162, Cham, 2015. Springer International Publishing.

[46] FreeIPA Development Team. Directory server. Accessed: 2018-05-08. URL: https://www.freeipa.org/page/Directory_Server.

[47] S. Wang, R. State, M. Ourdane, and T. Engel. Riskrank: Security risk ranking for ip flow records. In *2010 International Conference on Network and Service Management*, pages 56–63, Oct 2010. doi:10.1109/CNSM.2010.5691334.

[48] Stephen C. Williams. *Analysis of the SSH Key Exchange Protocol*, pages 356–374. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. URL: https://doi.org/10.1007/978-3-642-25516-8_22, doi:10.1007/978-3-642-25516-8_22.

[49] W. T. Young, A. Memory, H. G. Goldberg, and T. E. Senator. Detecting unknown insider threat scenarios. In *2014 IEEE Security and Privacy Workshops*, pages 277–288, May 2014. doi:10.1109/SPW.2014.42.

[50] Aneta Šteflová Petrová, Filip Hanzelka, Lucie Maňásková, Marc Muehlfeld, Tomáš Čapek, and Ella Deon Ballard. *Linux Domain Identity, Authentication, and Policy Guide*. Redhat Foundation, 2018. Accessed: 2018-03-12. URL: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/pdf/linux_domain_identity_authentication_and_policy_guide/Red_Hat_Enterprise_Linux-7-Linux_Domain_Identity_Authentication_and_Policy_Guide-en-US.pdf.

[51] Aneta Šteflová Petrová, Filip Hanzelka, Lucie Maňásková, Marc Muehlfeld, Tomáš Čapek, and Ella Deon Ballard. *Red Hat Enterprise Linux 7, System-Level Authentication Guide*. Redhat Foundation, 2018. Accessed: 2018-02-09. URL: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/pdf/system-level_authentication_guide/Red_Hat_Enterprise_Linux-7-System-Level_Authentication_Guide-en-US.pdf.

Appendix A

PowerShell commands for data collection

All the following commands can be used in Windows PowerShell to gather user information from the local system and from an AD.

Listing A.1: The registy key which controls the information about cached credentials

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\
```

Listing A.2: A PowerShell command to list access logs [1] [2]

```
(Get-WinEvent -FilterHashtable @{logname='security'; id=4624} | Where {$_.properties[8].value -ne 5}).toXml()
```

Listing A.3: A PowerShell command that lists all active accounts on the host[3]

```
Get-LocalUser | Where Enabled -eq True | Select *
```

Listing A.4: Lists user information from the domain.

```
Get-AdUser -Properties *
```

Listing A.5: Lists group information from the local machine.

```
Get-LocalGroup | ForEach-Object {\$group = \$_; \$users=@();
Get-LocalGroupMember \$group.name | ForEach-Object {\
$users += \$_.ObjectClass + ':' + \$_.Name}; \$group |
Add-Member users \$users -PassThru }
```

Listing A.6: Lists group information from the domain.

```
Get-ADGroup | ForEach-Object {$group = $_; $users=@();
Get-ADGroupMember $group.name | ForEach-Object {$users
+= $_.ObjectClass + ':' + $_.Name}; $group | Add-Member
 users $users -force -PassThru }
```

Appendix B

BASH commands for data collection

All the following commands can be used in a bourne again shell on CentOS to gather user information from the local system and from an IPA domain.

Listing B.1: Lists logins and logouts during the last year according to the log files. Logs are reset upon reboot.

```
journalctl _TRANSPORT=syslog SYSLOG_FACILITY=10
    ↪ SYSLOG_FACILITY=4 --since "1 year ago" | egrep "session
    ↪ [(opened)(closed)]" | cut -d ' ' -f 1,2,3,8,11 | sed "
    ↪ s/opened/login/"| sed "s/closed/logout/"
```

Listing B.2: A command to list local acounts in linux

```
lslogins -o USER,UID,HOMEDIR,GROUP,GID,SUPP-GROUPS,SUPP-GIDS
```

Listing B.3: A small script for gathering md5 sums of all public ssh-keys in `~/.ssh`

```
for keys in $(grep --exclude=known_hosts -lr ssh- /home/*/.
    ↪ ssh/); do while read key; do  echo $(echo $keys | cut -
    ↪ d / -f3 ) $(echo $key | cut -d ' ' -f2 | md5sum) ; done
    ↪  < $keys; done;
```

Listing B.4: A small script to show all info about rules applicable to the host

```
IFS=$',';

rules=$(ipa host-show $(hostname) | grep -i rule | sed -E "s
    ↪ /^.*\sof\s//" | cut -d' ' -f1,3-)
sudorules=("$(echo "$rules" | grep -i sudo | cut -d' ' -f2- |
    ↪ sed -E -e ':a' -e 'N' -e '$!ba' -e 's/\n/\,/')")
hbacrules=("$(echo "$rules" | grep -i hbac | cut -d' ' -f2- |
    ↪ sed -E -e ':a' -e 'N' -e '$!ba' -e 's/\n/\,/')")
hbacrules=($hbacrules)
sudorules=($sudorules)

for i in "${sudorules[@]}";
do
    ipa sudorule-show  "$i";
    echo -e "\n"
```

```
13 done
14
15 echo "^ Sudorules / HBACrules V";
16
17 for i in "${hbacrules[@]}";
18 do
19     ipa hbacrule-show "$i";
20     echo -e "\n"
21 done
22 unset IFS
```

Listing B.5: Lists all users which can be authenticated through FreeIPA from the current machine

```
1 ipa user-find --all
```

Appendix C

Experiment Data

C.1 Host Data

Host	OS	Vulnerability count
hostA	centos7	78
hostB	centos7	11
hostC	centos7	38
hostD	debian10	0
hostE	debian10	0
hostF	debian9	19
hostG	debian9	1
hostH	rhel72	565
hostI	rhel7	568
hostJ	win10pro	219
hostK	win10pro	3
hostL	win10pro-vr-test	625
hostM	win10pro	1553
hostN	win2012r2	0
hostO	win2016std-nano	2
hostP	win2k12r2	505
hostQ	win2k8r2entfull	29

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden