HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

COMPUTER SCIENCE

STUDYTHESIS

Readiness for Tailored Attacks and Lateral Movement Detection

Authors:

Claudio MATTES
claudio.mattes@hsr.ch

Lukas KELLENBERGER
lukas.kellenberger@hsr.ch

Supervisor:

Cyrill BRUNSCHWILER
Hochschule für Technik Rapperswil
cyrill.brunschwiler@hsr.ch

DEPARTEMENT COMPUTER SCIENCES
HSR UNIVERSITY OF APPLIED SCIENCES RAPPERSWIL
CH-8640 RAPPERSWIL, SWITZERLAND

November 11, 2018

Abstract

Management Summary

Initial Situation

Procedure

Results

Outlook

Task Definition

Einführung

Es werden vermehrt Cyberangriffe publik, wo Schadcode im Einsatz ist, welcher sich nicht nur auf einem infizierten System niederlässt, sondern weitere Systeme im Netz befällt. Das Ziel oder Resultat ist dabei oft die komplette Infiltrierung einer Organisation. In der Analyse solcher Fälle sind Information und Zeit ein Schlüssel zum Erfolg. Folglich ist die Bereitschaft "Readiness" für ein solches Ereignis ein entscheidender Faktor.

Aufgabe

Ziel dieser Arbeit ist es, ein Tool zu erstellen, welches die Bewertung der eigene Readiness erlaubt aber auch im Analysefall eine Unterstützung bietet. Readiness betrifft viele Aspekte und einfache Dinge wie korrekte Zeitstempel in Logs, deren Vollständigkeit oder die Bereitstellung von Backups. In der konkreten Aufgabenstellung soll die Readiness-Analyse primär für Windows-Infrastrukturen anhand von Logs und spezifischen Events erfolgen. Unter anderem soll auf den neusten Publikationen des japanischen Computer Emergency Response Teams (JPCERT/CC) und der öffentlichen Datenbank der MITRE Corporation, dem Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK^{TM}) Wissenspool, basiert werden. Das JPCERT und MITRE haben dabei die Werkzeuge und das generelle Vorgehen von Angreifern analysiert und geben Hinweise, welche Events auf eine mögliche Verseuchung hinweisen.

Abgrenzung

Es geht nicht darum neue Angriffsvektoren zu finden.

Tätigkeiten

- Projektmanagement und Dokumentation
- Einarbeitung in Incident Handling und Forensik
- Einarbeitung in Angriffstechniken und Werkzeuge
- Einarbeitung in Abwehrtechniken und Härtung von Systemen
- Studium öffentlicher Quellen und verfügbaren Tools
- Umsetzung eines Analyzers gemäss Anforderungen basierend auf etablierten Frameworks

Vorgehen

Im Rahmen der allgemeinen Richtlinien zur Durchführung von Studien- und Bachelorarbeiten gemäss eigenem Projektmanagementplan. Dieser Projektmanagementplan ist als Erstes zu erstellen und enthält insbesondere:

- Die Beschreibung des dem Projektcharakter angepassten Vorgehensmodells.
- Eine erste Aufteilung der Aufgabe in gemeinsam und einzeln zu bearbeitende Teile unter Berücksichtigung der vorgegebenen Teilaspekte. Die genaue Aufteilung muss spätestens nach der Technologiestudie (Elaboration) erfolgen.
- Den Projektplan (Zeitplan) und die Meilensteine.

Anforderungen

Es geht primär darum einen Analyzer zu erstellen um die "Readiness for Tailored Attacks and Lateral Movement Detection" beurteilen zu können. Idealerweise kann dieses Tool von einem IT Administrator ohne spezielle Kenntnisse und grossartige Installationsprozeder ausgeführt werden.

Schematisch aber nicht bindend werden folgende Schritte auszuführen sein

- Definition der Requirements für einen neuen/verbesserten Analyzer
- Design und Analyse basierend auf den Vorgaben
- Vorschläge für die Umsetzung oder Verbesserung eines
  - Readiness Analyzers
  - Readiness Optimizers
  - Compromise Analyzers
- Implementation der Funktionalität und Erstellung eines Benutzerhandbuch
- Erweiterung der Analyzer um neue Erkenntnisse, Werkzeuge und Indicators
- Dokumentation der Software und Skripte

Technologien

- Windows Workstations, Windows Server, Windows Security generell
- Windows Event Logs, Security und Audit Logs
- Windows On-Board Tools, Sysinternals Toolkit
- Active Directory Service (AD) Services
- Group Policy Objects (GPO)
- PowerShell, .NET, Python, Windows Batch

Infrastruktur

Die Arbeiten werden auf den Rechnern der Studenten durchgeführt. Zusätzlich benötigte Software oder Hardware wird bei Bedarf und nach Rücksprache mit Compass Security zur Verfügung gestellt.

Erwartete Resultate

In elektronischer Form

- lauffähiges Toolkit und kompletter Source Code
- komplette Software Dokumentation (Use Cases, Klassenmodell, Sequenzdiagramme usw. in UML)
- komplette Use Cases und Erfolgs-Szenarien resp. Musterlösungen
- alle Dokumente und Protokolle (vorzugsweise in englischer Sprache)

Auf Papier

Gemäss der Anleitung der HSR: \\hsr.ch\root\alg\skripte\Informatik\Fachbereich\Studienarbeit_Informatik Es muss aus den abgegebenen Dokumenten klar hervorgehen, wer für welchen Teil der Arbeit und der Dokumentation verantwortlich war (detaillierte Zeiterfassung).

Termine

Termine gemäss der HSR: \\hsr.ch\root\alg\skripte\Informatik\Fachbereich\Studienarbeit_Informatik\SAI\Termine

Datum	Task
17.09.2018	Beginn der Arbeit, Ausgabe der Aufgabenstellung durch den Betreuer.
18.12.2018	Erfassung des Abstracts im Online-Tool https://abstract.hsr.ch/ Die Studierenden geben den Abstract für die Diplomarbeitsbroschüre zur Kontrolle an ihren Betreuer/Examinator frei. Der Betreuer/Examinator gibt das Dokument mit dem korrekten und vollständigen Abstract zur Weiterverarbeitung an das Studiengangsekretariat frei Vorlagen sowie eine ausführliche Anleitung betreffend Dokumentation stehen auf dem Skripteserver zur Verfügung.
21.12.2018	Der Betreuer/Examinator gibt das Dokument mit dem korrekten und vollständigen Abstract der Broschüre zur Weiterverarbeitung an das Studiengangsekretariat frei. Hochladen aller verlangten Dokumente auf archiv-i.hsr.ch Abgabe des Berichts an den Betreuer bis 12.00 Uhr

Zeitplan und Meilensteine

Zeitplan und Meilensteine für das Projekt sind von den Studenten selber zu erarbeiten und zusammen mit dem Projektmanagementplan abzuliefern. Die Meilensteine sind bindend. Der erste Meilenstein ist vorgegeben. Mit den Betreuern werden regelmässige Sitzungen zur Fortschrittskontrolle durchgeführt.

Betreuung

Die Arbeiten werden durch Cyrill Brunschwiler betreut. Der Gegenleser ist noch nicht bestimmt.

Kontakt

Cyrill Brunschwiler, Managing Director, Compass Security Schweiz AG
Weststrasse 50, 8003 Zürich, Switzerland
Werkstrasse 20, 8645 Jona, Switzerland

+41 55 214 41 73
cyrill.brunschwiler@compass-security.com
cyrill.brunschwiler@hsr.ch
https://fb.compass-security.com/inbox/hUGXMr2EeZ2V7b

Unterschriften

Jona, 28. September 2018


Cyrill Brunschwiler


Claudio Mattes


Lukas Kellenberger

Contents

Part I

Technical Report

10 Introduction and Overview

As described in the introduction of the task definition, the key for a successful analysis in case of an advanced persistence threat (APT) or lateral movement in a network, it is fundamental to have solid event logging of all systems participating in the network.

11 Analysis

This chapter describes the first step of this project, the research of published technical reports and tools which are considered interesting for this project.

11.1 BloodHound / SharpHound

11.1.1 Description

BloodHound describes itself on its wiki page on GitHub as follows:

> *"BloodHound is a single page Javascript web application, built on top of Linkurious, compiled with Electron, with a Neo4j database fed by a PowerShell/C# ingestor. BloodHound uses graph theory to reveal the hidden and often unintended relationships within an Active Directory environment. Attacks can use BloodHound to easily identify highly complex attack paths that would otherwise be impossible to quickly identify. Defenders can use BloodHound to identify and eliminate those same attack paths. Both blue and red teams can use BloodHound to easily gain a deeper understanding of privilege relationships in an Active Directory environment." [1]*

11.1.2 Difficulties

BloodHound was tested in the test environment which is described later in this chapter. Both the C# and Python ingestors were successfully installed and tested. The only problem which occurred was that the Python-ingestor does not yet run on the latest Python release. One must have a Python 2.7.x version installed to run the scripts successfully.

11.1.3 Conclusion

The most interesting aspect of BloodHound for our project is the way it retrieves its data. Due to the decision that the application, in a first step, only reads the data of the local computer and not the whole domain, BloodHound will only be important in a later part of the project. Their so called ingestor will be used to retrieve the data of a whole network instead of only a local computer.

11.2 WEFFLES

11.2.1 Description

WEFFLES (Windows Event Logging Forensic Logging Enhancement Services) is a Threat Hunting/Incident Response Console with Windows Event Forwarding and PowerBI, coded and published by Microsoft-Security-Employee Jessica Payne. It is build to help setting up the Windows Event Forwarding, so that all the collected logs of a system are stored on one centralised server, and afterwards to analyse the collected data. Jessica Payne wrote an installation instruction on the Microsoft TechNet blog https://blogs.technet.microsoft.com/jepayne/2017/12/08/weffles/. Once the data is collected the generated weffels.csv file can simply imported into Excel and start filtering the logs to gain the needed. Jessica Payne recommends to use PowerBI, a business analytics tool designed by Microsoft. In her published blog she also gives a short introduction on what to look out for, which event ids are important and other useful tips and tricks for detecting suspicious activities in the network.

11.2.2 Conclusion

WEFFELS will not be the product on which this project is based, but could become an important point of reference. The installation guide and other WEFFELS-related documents collected by Jessica

Payne provide a lot of information for reading and understanding logs, which will be very helpful for this project. Also an interesting aspect of WEFFLES and the Jessica Payne article is how she visualised the logs, using Microsoft PowerBI.

11.3 Microsoft Security Compliance Toolkit

11.3.1 Description

The Microsoft Security Compliance Toolkit (SCT) [2] allows security administrators to analyse their configured enterprise Group Policy Objects (GPO) in comparison to the Microsoft-recommended GPO baselines. The toolkit is handed with several baseline GPO's for different versions of Microsoft Windows Client and Servers:

- Windows 10 security baselines
  - Windows 10 Version 1803 (April 2018 Update), 1709 (Fall Creators Update), 1703 (Creators Update), 1607 (Anniversary Update), 1511 (November Update), 1507
- Windows Server security baselines
  - Windows Server 2016
  - Windows Server 2012 R2
- Microsoft Office security baseline
  - Office 2016

11.3.2 Difficulties

The toolkit is very simple and could be understood and used without any difficulties. The handling is very intuitive and does not require much training. Please note, however, that the toolkit cannot be used with Windows 10 Home, since active directory support is not provided with this version.

11.3.3 Conclusion

This toolkit can be used for a very baseline GPO in enterprise environment. With the handed baselines it is easy to compare the configured GPO and to see the readiness of the enterprise GPO. The toolkit enables the comparison of different local GPO's installed on different Clients or Servers to check their consistency. In addition, the handed baselines can be used for building new GPO's. Furthermore, Microsoft delivers with the SCT a Local Group Policy Object Utility (LGPO.exe) to:

- Import and apply policy settings
- Export local policy to a GPO backup
- Parse a registry.pol file to "LGPO text" format
- Build a registry.pol file from "LGPO text"

This toolkit is very interesting, but cannot be used to build on it. The reason for this is that the source code of the complete toolkit is not available. However, it can be used as additional help for checking the readiness of an enterprise environment.

11.4 LogonTracer

11.4.1 Description

JPCERT/CCs LogonTracer is a tool built to investigate malicious logons on a system based on the research described in section 11.9 JPCERT/CC - Detecting Lateral Movement in APTs. The tool links hostnames or IP addresses with the *"[...] account name found in logon-related events and displays it as a graph"*. [3] The following event ids are checked with the tool:

- 4624:Successful logon
- 4625:Logon failure
- 4768:Kerberos Authentication(TGT Req.)
- 4769:Kerberos Service Ticket (ST Req.)
- 4776:NTLM Authentication
- 4672:Assign special privileges

The following figure depicts a sample graph of logins from different users in the test environment:

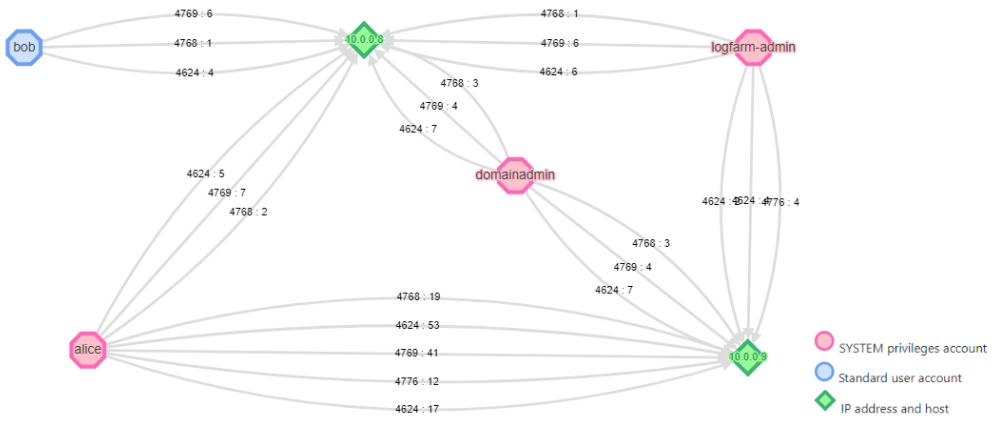

Figure 1.1: LogonTracer: Sample Graph from Test Environment

To use the LogonTracer, only a .evtx-File (Windows XML Event Log - export of Windows event logs) is necessary to be uploaded. To get the best result out of LogonTracer an export of the security event log from the domain controller should be used - to get as much information of the network as possible. With the built-in analysis of logins, by using machine learning models and statistical analysis, LogonTracer is able to provide a ranking of the most malicious users which tried to log in. [4]

In addition, LogonTracer provides a timeline for all or selected users to show when each user logged in. The timeline can also be displayed as a graph with the LogonTracer, allowing anomalies to be detected more quickly.

The test environment showed that this graph can quickly become confusing - especially in a larger corporate environment. Although only a small environment as described in the section 12 "Test environment" was used, it turned out that various users wanted to log on to the virtual machines. The reason for this is that the test environment was built in the Microsoft Azure Cloud and is accessible via public IP addresses in the cloud.

Nevertheless, with meaningful filters the search can be restricted and the graph can be used efficiently, as shown in the figure 1.1 LogonTracer: Sample Graph from Test Environment

11.4.2 Difficulties

During the test phase of LogonTracer some difficulties were faced. It is pretty easy to get the docker container, but starting LogonTracer was a bit of a challenge. JPCERT/CC gives the following instructions for starting the docker container:

Listing 1.1: LogonTracer: given docker run command

```
1 $ docker run --detach \
2 --publish=7474:7474 --publish=7687:7687 --publish=8080:8080 \
3 -e LTHOSTNAME=[IP_Address] jpcertcc/docker-logontracer
```

The problem was that the parameter `[IP_Address]` was not described well. If the command `docker ps` was executed it always showed the following `PORTS`:

Listing 1.2: LogonTracer: docker ps (PORTS)

```
1 PORTS
2 0.0.0.0:7474->7474/tcp, 0.0.0.0:7687->7687/tcp, 7473/tcp, 0.0.0.0:8080->8080/tcp
```

After some time of investigation and further tests, it turned out that under `PORTS` the ports respectively ip addresses of the container can be bound to the host. But these are not relevant for the LogonTracer, because it provides a web application under the defined parameter `[IP_Address]` and it can eventually be reached via `localhost:8080`. If this parameter was set to `127.0.0.1`, the database containing the imported .evtx file could not be accessed. Thus the graph was never displayed. The parameter `[IP_Address]` set to `localhost` solved this problem.

Listing 1.3: LogonTracer: recommended docker run command

```
1 $ docker run --detach \
2 --publish=7474:7474 --publish=7687:7687 --publish=8080:8080 \
3 -e LTHOSTNAME=localhost jpcertcc/docker-logontracer
```

11.4.3 Conclusion

The LogonTracer is unique in its form and should not be underestimated for the detection of lateral movements. This is because user access to various components available in the network, can be visualised simply and graphically, hence conclusions can be drawn about what has happened.

However, the LogonTracer is not suitable for detection readiness and cannot be used to build on it. Nonetheless, approaches for reading the event log for further work could be used. This tool is also extremely interesting and recommendable for a further detection of lateral movements.

11.5 Microsoft Monitoring Active Directory for Signs of Compromise

11.5.1 Description

This article "Microsoft Monitoring Active Directory for Signs of Compromise" [5] is about configuration of an solid event log monitoring for Microsoft servers. The article gives a quite a good overview about the audit policy in Microsoft systems and what each policy stands for. The article gives information about the most important audit policies and how noisy (if a lot of data is produced by them) they are. This study does not go into the details of the audit policies in detail. Furthermore, the article describes how the policies can be read with powershell.

In this article Microsoft compiles in Appendix L [6] all important event ids which are necessary for a successful detection of APTs and lateral movements.

11.5.2 Conclusion

Due to the fact that audit policies are an important setting for solid event logging, this article and appendix L will be a central part of the toolkit to be built. As a next step and part of this study these event ids have to be correlated with the event ids found in the JPCERT/CC's study "Detecting Lateral Movement through Tracking Event Logs" [7] to make a clear statement which event ids have to be logged.

11.6 MITRE ATT&CK

11.6.1 Description

MITRE ATT&CK introduces itself on its website as follows:

> *"MITRE ATT&CK™ is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community." [8]*

The portal offers a variety of attacks and their patterns, which are currently known in different operating systems. MITRE ATT&CK describes the attack in short words and then lists possibilities for detection and mitigation. The portal also describes various attack tools, their targets and effects on the system. In addition, the corresponding attacks are always cross-referenced. This is a great advantage for a quick search, especially when time is of the essence.

11.6.2 Conclusion

Although many attacks are described and how they can be detected and fended off, MITRE ATT&CK is not quite suitable for our task. The readiness of a system to detect tailored attacks and lateral movements is only roughly described and would be associated with a time-consuming analysis in order to draw exact conclusions.

11.7 sysmon-modular

11.7.1 Description

With sysmon-modular [9] a clean configuration of the Windows system service System Monitor (Sysmon), an xml-file which is loaded by Sysmon, is provided. Noisy process creations, which are made by legitimate programs, are suppressed as far as possible by Sysmon. The tool offers the possibility and it is expressly recommended by the developer to adapt the configuration to the respective organization. Furthermore, sysmon-modular implements various attacks in MITRE ATT&CK for detection with Sysmon. It offers the possibility to detect the attacks shown in the figure 1.2 with Sysmon.

Figure 1.2: Detectable attacks with sysmon-modular

11.7.2 Conclusion

Sysmon-modular offers a very good basic configuration for Sysmon based on the platform MITRE ATT&CK which is widely used in the security scene. Unfortunately, sysmon-modular was discovered when decisions were made to develop a tool based on the study "Detecting Lateral Movement through Tracking Event Logs" by JPCERT/CC. The readiness of a system with the basis of MITRE ATT&CK patterns would probably have had an even greater impact. However, Sysmon-modular will most likely not be included in the tool during this study, unless there are still enough time reserves for such an integration.

11.8 Sysmon Tools

11.8.1 Description

Sysmon Tools [10] contains some useful functions to make better use of Sysmon. Among other things there are different views for the representation of the single entries which were recorded by Sysmon. A Process View is provided which can be used to examine a process in more detail. Related processes are taken into account and represented in a simple data-flow-like view, sorted by chronological order. With the Map View you can include geo-locate IP addresses during the import phase and Map View tries to geo-map the network destinations with ipstack [11]. The All Events View represents a full search by Sysmon and can be filtered and grouped accordingly. Furthermore, Sysmon Tools offers a Sysmon Shell, which can be used to create a customized XML configuration for Sysmon using a GUI. Templates are also provided for further building.

11.8.2 Conclusion

This tool can also be a great help for detecting attacks and, with the Sysmon Shell, a robust configuration for Sysmon can be created. However, Sysmon Tool will have no basis for this project.

11.9 JPCERT/CC - Detecting Lateral Movement in APTs

11.9.1 Description

This document [12] is from a presentation by Shingo Abe, a JPCERT/CC employee. In it he describes how to find system intruders more effectively using Windows Event Logs. The collected data is used to to detect inconsistencies more effectively, such as when an administrator logs on to another machine or when an administrator logs on suspiciously often.

11.9.2 Conclusion

This presentation contains interesting information which could be built into the project at a later point. The information this document contains is more suitable for monitoring purposes than for checking the readiness of a system.

11.10 JPCERT/CC - Detecting Lateral Movement through Tracking Event Logs

11.10.1 Description

This is a document [7] the Japan Computer Emergency Response Team Coordination Center, or short JPCERT/CC, has published in the year 2017. It describes how, in their experience, attackers proceed with lateral movement. In a very detailed 81-page report they describe the procedure step-by-step, the tools used and what is most interesting for the project, the logs generated while doing so.

11.10.2 Conclusion

This report will have the biggest impact on this project, it shows which logs have to be read in any case. In addition, JPCERT/CC describes in this report which configurations are necessary for solid logging. The appendix not only describes the individual event log IDs, but also the audit policy that can be used to achieve them. For this reason, the checklist to be used will mainly be based on this report. With the provided information we see the greatest potential to develop a suitable tool for the accomplishment of the task in the given time. The given information of the configuration settings in JPCERT/CCs study appendix must be correlated with the "Advanced security auditing FAQ" [13] in order to define the right auditing settings so that the right events are captured.

11.11 Decision for a new Tool

At the beginning it was not clear how the tool should be built exactly and what the functionality and scope should be based on. After a detailed analysis of different tools, reports and studies, it was possible to better estimate how an efficient detection of the readiness of a system can be implemented. It would have been desirable to be able to build on an existing tool, but as shown in a five-week analysis, there is no such tool. For this reason it was decided to develop a tool based on JPCERT/CCs study. The configurations in the Advanced Audit Settings of the GPOs are to be checked accordingly and in a second step the event logs are to be searched for the EventIDs.

11.12 Mandatory Event Logs

The following tables lists the event logs which are mandatory and must be logged based on the study JPCERT/CC - Detecting Lateral Movement through Tracking Event Logs:

Security	
EventID	**Description**
104[1]	The System log file was cleared
4624	An account was successfully logged on
4634	An account was logged off
4648	A logon was attempted using explicit credentials
4656	A handle to an object was requested
4658	The handle to an object was closed
4660	An object was deleted
4661	A handle to an object was requested
4663	An attempt was made to access an object
4672	Special privileges assigned to new logon
4673	A privileged service was called
4688	A new process has been created
4689	A process has exited
4690	An attempt was made to duplicate a handle to an object
4720	A user account was created
4726	A user account was deleted
4728	A member was added to a security enabled global group
4729	A member was removed from a security enabled global group
4768	A Kerberos authentication ticket (TGT) was requested
4769	A Kerberos service ticket was requested
4946	A change has been made to Windows Firewall exception list. A rule was added
5140	A network share object was accessed
5142	A network share object was added
5144	A network share object was deleted
5145	A network share object was accessed
5154	WFP has permitted an application or service to listen on a port for incoming connections
5156	WFP has allowed a connection
7036[1]	The service state has changed
7045[1]	A service was installed in the system

Table 1.1: Mandatory Security Event Logs

[1] Recorded by default Windows settings

System	
EventID	**Description**
8222[1]	TShadow copy has been created
20001[1]	Driver Management concluded the process to install driver

Table 1.2: Mandatory System Event Logs

Applications & Service > Microsoft > Windows > TaskScheduler > Operational	
EventID	**Description**
102[1]	Task completed
106[1]	A task has been registered
129[1]	A task process has been created
200[1]	The operation that has been started
201[1]	The operation has been completed

Table 1.3: Mandatory TaskScheduler Event Logs

Applications & Service > Microsoft > Windows > Windows Remote Management > Operational	
EventID	**Description**
6[1]	Creating WSMan Session
169[1]	User authentication authenticated successfully

Table 1.4: Mandatory Windows Remote Management Event Logs

Applications & Service > Microsoft > Windows > TerminalServices-LocalSessionManager > Operational	
EventID	**Description**
21[1]	Remote Desktop Services: Session logon succeeded
24[1]	Remote Desktop Services: Session has been disconnected

Table 1.5: Mandatory TerminalServices-LocalSessionManager Event Logs

Applications & Service > Microsoft > Windows > Sysmon > Operational	
EventID	**Description**
1[2]	Process Create
5[2]	Process Terminated
8[2]	CreateRemoteThread detected

Table 1.6: Mandatory Sysmon Event Logs

[1] Recorded by default Windows settings
[2] Recorded by default Sysmon settings

11.13 Correlation: Advanced Audit Policy Setting and Event Log IDs

In this section, the "Advanced Audit Policies" required to trigger the corresponding event logs are shown in tables. Based on these tables, the "Advanced Audit Policies" are checked for correctness with the tool.

Account Logon	
Subcategory	**EventIDs**
Audit Kerberos Authentication Service	4768
Audit Kerberos Service Ticket Operations	4769

Table 1.7: Advanced Audit Policy Setting Account Logon

Account Management	
Subcategory	**EventIDs**
Audit User Account Management	4720, 4726
Audit Security Group Management	4728, 4729

Table 1.8: Advanced Audit Policy Setting Account Management

Logon/Logoff	
Subcategory	**EventIDs**
Audit Logon	4624, 4648
Audit Logoff	4634
Audit Special Logon	4672

Table 1.9: Advanced Audit Policy Setting Logon/Logoff

Object Access	
Subcategory	**EventIDs**
Audit File System	4656, 4658, 4660, 4663, 4670
Audit Kernel Object	4656, 4658, 4660, 4663
Audit Registry	4656, 4658, 4660 , 4663
Audit Handle Manipulation	4658, 4690
Audit SAM	4661
Audit File Share	5140, 5142, 5144
Audit Detailed File Share	5145
Audit Filtering Platform Connection	5154, 5156

Table 1.10: Advanced Audit Policy Setting Object Access

Policy Change	
Subcategory	**EventIDs**
Audit MPSSVC Rule-Level Policy Change	4946

Table 1.11: Advanced Audit Policy Setting Policy Change

Privilege Use	
Subcategory	**EventIDs**
Audit Non Sensitive Privilege Use	4673
Audit Sensitive Privilege Use	4673

Table 1.12: Advanced Audit Policy Setting Privilege Use

11.14 Domain Analysis

The following section describes the problem domain which is faced during this project. Despite the decision to not programm an object orientated solution, there are several things to be aware of and to think through carefully. For this reason building a domain model is a simple and suitable suitable technique to use for. The following figure 1.3 shows the domain model and will be explained in some details afterwards.

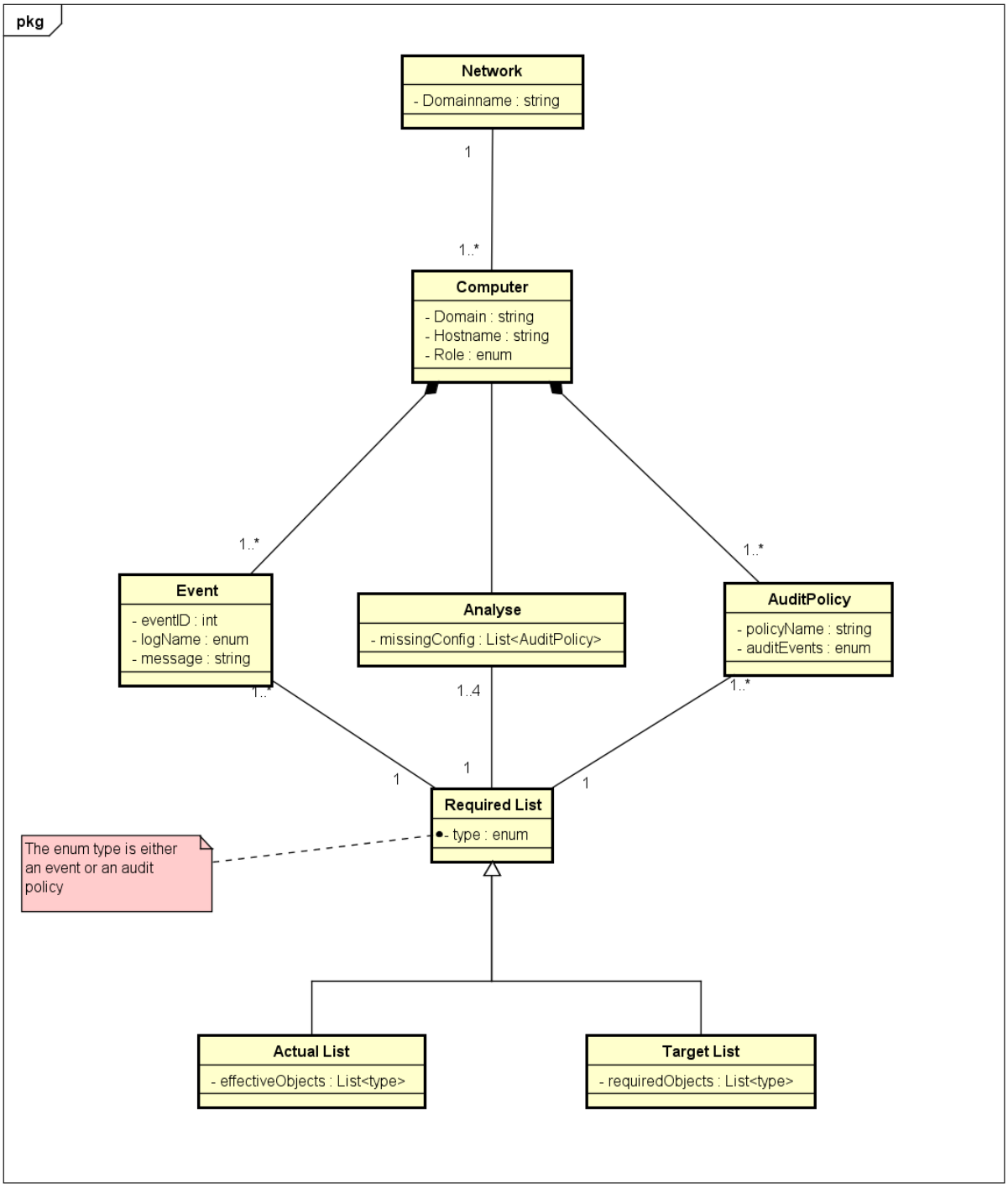

Figure 1.3: Domain Model

11.14.1 Network

The class network depicts the organizations wide network which is used to connect all clients and servers together. In this project the main goal is to locally detect the readiness of the system and not to extend the detection for a system-wide infrastructure. For further development on this project and a system-wide extension, the network is already considered in this domain model.

11.14.2 Computer

A computer illustrates either a client like a Windows 10 machine or a server in particular a domain controller running on a Windows Server 2016. In principle, however, every Windows computer is represented. A computer is a core component in our project, because the detection is done on a single client or server.

11.14.3 Event

An event represents a single event log entry in simplified form.

11.14.4 AuditPolicy

AuditPolicy displays the individual settings of the audit policies of the group policy, which can be found via `gpedit.msc` under "Computer Configuration > Windows Settings". However, only the settings under "Security Settings > Advanced Audit Policy Configuration" are considered and not the settings under "Security Settings > Local Policies > Audit Policy". The reason for this is that Microsoft recommends that only one of the two policies is used:

> *[...] do not use both the basic audit policy settings under Local Policies\Audit Policy and the advanced settings under Security Settings\Advanced Audit Policy Configuration. Using both basic and advanced audit policy settings can cause unexpected results in audit reporting.* [13]

A single audit policy setting represents one or more event IDs logged by this configuration.

11.14.5 Required List

Actual List The actual list represents the current state of the system. It reflects the event log IDs that have occurred and the audit policies that have been set.

Target List The target list represents either the list of event logs or configured audit policies which must be present for a solid detection of attacks.

11.14.6 Analysis

Based on the required lists and the current state of the computer, the analysis shows which settings are missing in the audit policies.

12 Test environment

A virtual network was set up on the Microsoft Azure Cloud as a test environment. The test network was set up in the cloud so that the development team can access the network regardless of its location. The test network consists of a Windows server and two Windows clients. Active Directory service was configured on the server to manage the client computer and to have the possibilities to create group policies. Group policies are used in almost every corporate environment to build rule sets for configurations. These configurations are a core element to check the readiness of a system. The following operating systems were installed in this test network:

Server:

- Windows Server 2016

Clients:

- Windows 10 Pro, Version 1709

The network is structured as followed:

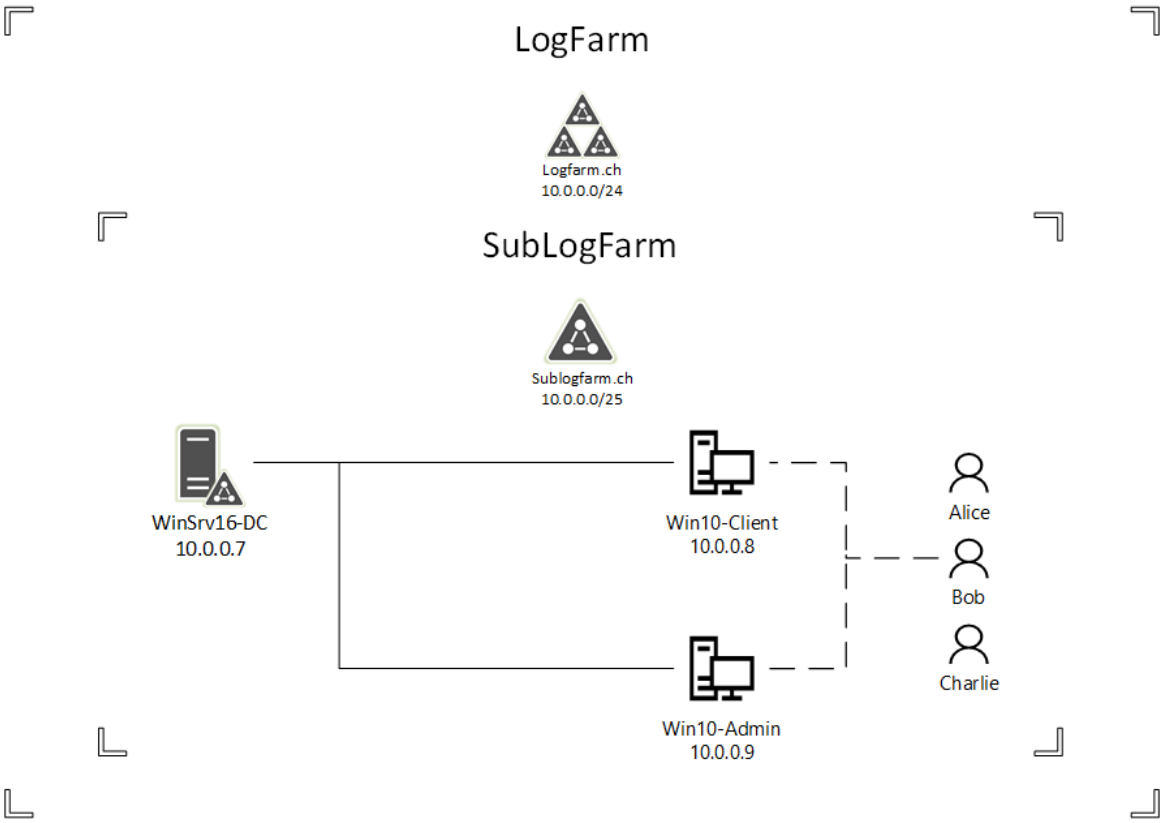

Figure 1.4: Test Environment

12.1 User

Three users were configured for the logfarm-network:

Name	Privileges
alice	Domain administrator
bob	User
charlie	User

Table 1.13: Test Environment User

12.2 Difficulties

Various difficulties occurred which are presented in this subsection.

Connect to the virtuel machines via RDP
After setting up the virtual machines on Azure, the developers tried to connect to the devices via the Remote Desktop Protocol but failed. First, the developers suspected the issue was the incoming port rules, so the machines were reinstalled. However, this did not fix the issue. It became apparent that the problem were not the virtual machines, but with the network used to connect to the Microsoft Azure Cloud. Some firewall rules blocked the RDP-connection. In order to avoid this, the developers used a VPN-Connection in which these rules did not apply.

Firewall setting for ICMP
After the virtual network had been set up, the developers tested the connections in the virtual network. The configured DNS ran without any problem and could translate all hostnames. Testing the network using Pings showed that almost all clients were receiving pings, but the ping-requests by one client remained unanswered. It transpired that, for some inexplicable reason, the incoming ICMP-firewall-settings were different on this client. After adjusting the setting, the ping-requests were answered positively.

RDP connection for Bob and Charlie
Due to the fact that the user `alice` owns domain administrator privileges, this user was able to connect over RDP without an error. Bob and Charlie on the other hand did not have this permissions. The developers had to create a group for them, the RDP-Group. This group was then allowed to login over RDP on the clients Win10-Client and Win10-Admin.

13 System Architecture

In this section the following main question is answered:

"What would a system architecture look like to fulfill the described problem domain?"

This includes the coverage of use cases, non-functional requirements, technologies used and how the tool will be designed.

13.1 Use Cases

A visual representation of the use cases with a use case diagram was deliberately omitted, because there is only one actor involved - the security advisor. The actor is not specifically mentioned in the use cases every time, because it is always the same.

13.1.1 UC01 - Read Resultant Set of Policies

Description
The specified audit policies are read and saved in a temporary file.

Precondition
The system is running and the tool must possess administrator permissions.

Main Success Scenario

1. Read the specified audit policies from the system
2. Save the needed information from the audit policies in a temporary file for analysis purposes.

13.1.2 UC02 - Analyse Audit Policies

Description
The list which was created in UC01 is compared to a "perfect settings"-list. Missing or wrong settings are going to be exported into a separate file.

Precondition
UC01 is fulfilled: the temporary file is available.

Main Success Scenario

1. The temporary files can be read
2. Creates a list of incorrect settings

13.1.3 UC03 - Find Event Logs

Description
Event logs are search by ID and marked in an external file as found or missing.

Precondition
The system is running and must have valid event logs. The tool must possess administrator permissions.

Main Success Scenario

1. Search for the specified event logs from the local system
2. Save the result from the search in a temporary file for analysis purposes.

13.1.4 UC04 - Analyse Found Event Logs

Description
The implemented logic analyses, by defined event ids, which events occurred or are missing and creates a list of events that did not occurred or are not logged yet.

Precondition
UC03 is fulfilled: the temporary file is available.

Main Success Scenario

1. The temporary file can be read
2. The list with the defined event ids is available
3. Create a list of events which occurred and which are missing

13.1.5 UC05 - Display missing or wrong system configuration

Description
Based on the list created in UC02 and UC04 the user gets an overview of missing configurations (the result) which would improve the readiness of the system for a good attack detection.

Precondition
The lists from UC02 and UC04 are available.

Main Success Scenario

1. Displays a visual output of missing or wrong system configurations

13.1.6 UC06 - Save Result to specific path

Description
The actor has the possibility to save the overview from UC05 to a file in a specific path defined by the actor himself. This file contains the result from UC05 in a descriptive way.

Precondition
UC05 is fulfilled: the result, respectively the overview is available

Main Success Scenario

1. A file is saved to a specific path with the result from UC05
2. The path can be defined by the actor

13.2 Non Functional Requirements

NFR-No.	Description
NRF01	After using the Toolkit the system must remain in the status quo. More specifically the system shall not deliberately alter any existing entry in the event logs and registry. However, the tool may produce new event logs.
NFR02	The user shall not notice significant performance degradation from the system when using the Toolkit.
NFR03	The Toolkit must be portable with no installation procedure before use.
NFR04	The minimal target version of the system for the Toolkit to run must be Microsoft Windows 10 Professional or Microsoft Server 2016.
NFR05	The Toolkit runs in one go, but can also be executed in single steps with the possibility to skip single steps (pause/abort in case of performance problems)

Table 1.14: Non Functional Requirements

13.3 Technologies

13.3.1 Chosen Technologies

PowerShell & Visual Studio Code
The decision as to which technology to use, was made in favour of PowerShell. The reason why PowerShell was used, was that it is close to the Microsoft Operating System and that it has a large and detailed documentation at its disposal.

The scripts are written in Visual Studio Code with the extension packet "PowerShell". Visual Studio code is preferred to PowerShell ISE because it only requires working in one IDE for implementation and documentation.

LaTeX & Visual Studio Code
The documentation is written with LaTeX in Visual Studio Code wit the LaTeX Workshop extension. The main reason for LaTex was that the developers are already familiar with it. Furthermore, LaTeX offers a very simple way for referencing sources. On the other hand, we made the experience that with LaTeX the formatting is more reliable than for example when Microsoft Word is used.

Azure Cloud
The test environment is set up, as described in section 12 "Test environment", in the azure cloud. One server and two clients form a virtual network, this enables developers to access it from anywhere to any given time. A disadvantage is the changing public IP-addresses to access the VMs. In the end, the advantages outweigh the disadvantages.

GitHub
GitHub is used as a version control tool for source code and documentation. GitHub has been elected because of its good reputation and the experience the developers already gained with.

Continuous Integration
Continuous Integration (CI) for Powershell is unfortunately not very widespread as has been shown after some time of research. Fortunately the article "Converting a PowerShell Project to use Azure DevOps Pipelines" [14] by Daniel Scott-Raynsford was found, which describes in detail how a CI environment can be set up in Microsoft Azure DevOps. Due to the fact that Azure DevOps offers a very simple and clear handling, as well as supports all common operating systems (Linux, Windows and MacOS), it was decided to set up the CI environment in Azure DevOps. The structure and the important findings are described in the Continuous Integration manual.

13.3.2 Rejected Technologies

Python
The decision to use PowerShell and maybe C# for a GUI instead of Python was made because the developers do not have much experience with Python. Also PowerShell is closer to the Microsoft-OS. With Python there is no guarantee that the libraries which would be used are as powerful to solve the requirements.

13.4 Tool Design

This section describes the process of the toolkit and explain the individual steps in detail. As mentioned in the Use Cases, the actor of this toolkit will be a security advisor, who will execute the toolkit.

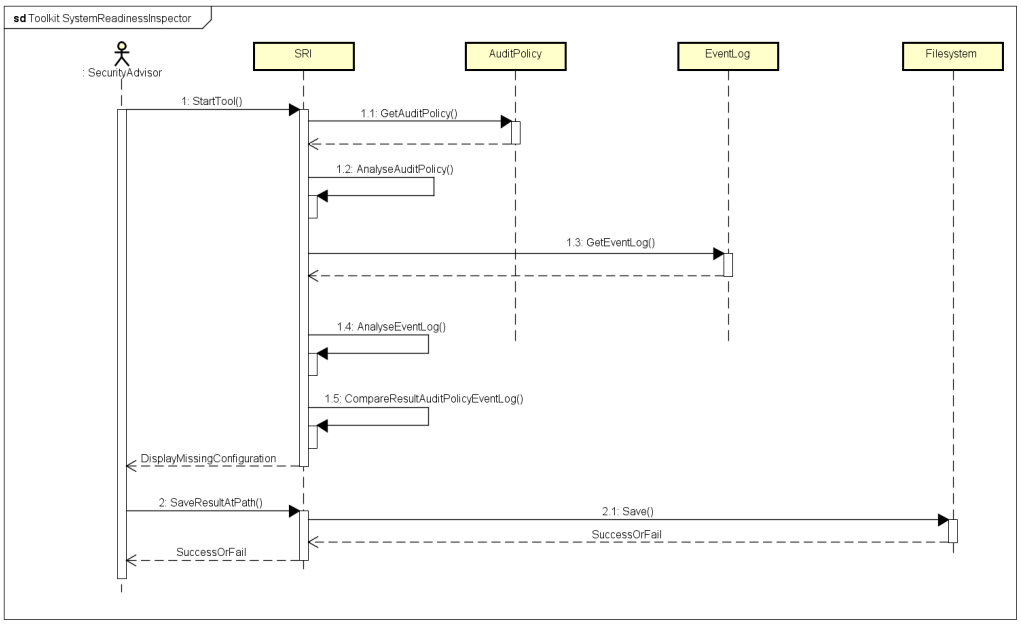

Figure 1.5: Sequence Diagram SystemReadinessInspector - SRI

13.4.1 GetAuditPolicy()

This task is responsible to get all Audit Policies, which are relevant for logging the right events according to JPCERT/CCs study. To gather all information about the Audit Policies and the current state of its configuration the "Resultant Set of Policies" (RSoP) must be read. [15] RSoP is a Microsoft snap-in to create a detailed report about the applied policy settings.

13.4.2 AnalyseAditPolicy()

In this task the gathered information from the task GetAuditPolicy(), which is represented as a XML-File, is going to be analysed against the recommendation from JPCERT/CCs study (see 11.10 JPCERT/CC - Detecting Lateral Movement through Tracking Event Logs). the result of this analysis will be stored in a XML-based format in a temporary file.

13.4.3 GetEventLog()

This task is responsible for getting the event logs from the system. Therefor the command `Get-EventLogs` [16] retrieves all logs from 'System', 'Application' and 'Security'. This logs are, to be analysed later, saved as a 'CSV' file to the current path were the PowerShell is running.

13.4.4 AnalyseEvent()

In this task the created 'CSV' file from GetEventLog() is used to analyse the collected logs. They are compared to a list provided by JPCERT to find out how ready the system would be if an enemy attack had taken place. The result of this comparison will be stored as a 'XML' file in order to visualise it.

14 Implementation

14.1 Script: GetAndAnalyseAuditPolicies

The basic idea was to implement the use case "UC01 - Read Resultant Set of Policies" separately from the use case "UC02 - Analyse Audit Policies". However, during the implementation it quickly became clear that these two use cases could be merged and did not have to be implemented separately. Therefore, both use cases were integrated into one script. The following describes how the two use cases were implemented.

14.1.1 Result

Both Use Cases were implemented together in one script. The script follows the following schedule:

- Reading and caching of the RSoP
- Verification that all defined audit policies are in place
- Verification that all defined audit policies are correctly configured
- Check if "Audit: Force audit policy subcategory settings (Windows Vista or later) to override audit policy category settings" is enabled in registry to prevent conflicts between similar settings
- Check if Sysmon is installed and running as a service
- Check whether CAPI2 is enabled

Each result of the individual steps is then collected in a common XML file. Finally, the environment and files that are no longer needed are deleted, so that only the result XML is available for further processing. A result could possibly look like the following listing:

Listing 1.4: Example Result Audit Policy Analysis

```
<?xml version="1.0"?>
<AuditPolicies>
   <AuditRegistry>NotConfigured</AuditRegistry>
   <AuditProcessTermination>NotConfigured</AuditProcessTermination>
   <AuditProcessCreation>NotConfigured</AuditProcessCreation>
   <AuditFilteringPlatformConnection>NotConfigured</AuditFilteringPlatformConnection>
   <AuditKernelObject>NotConfigured</AuditKernelObject>
   <AuditNonSensitivePrivilegeUse>NotConfigured</AuditNonSensitivePrivilegeUse>
   <AuditHandleManipulation>NotConfigured</AuditHandleManipulation>
   <AuditDetailedFileShare>NotConfigured</AuditDetailedFileShare>
   <AuditFileSystem>Success</AuditFileSystem>
   <AuditKerberosAuthenticationService>Failure</AuditKerberosAuthenticationService>
   <AuditKerberosServiceTicketOperations>Success</AuditKerberosServiceTicketOperations>
   <AuditLogoff>SuccessAndFailure</AuditLogoff>
   <AuditFileShare>NoAuditing</AuditFileShare>
   <AuditSAM>SuccessAndFailure</AuditSAM>
   <AuditSensitivePrivilegeUse>SuccessAndFailure</AuditSensitivePrivilegeUse>
   <AuditUserAccountManagement>SuccessAndFailure</AuditUserAccountManagement>
   <AuditSecurityGroupManagement>SuccessAndFailure</AuditSecurityGroupManagement>
   <AuditSpecialLogon>SuccessAndFailure</AuditSpecialLogon>
   <AuditLogon>SuccessAndFailure</AuditLogon>
   <AuditMPSSVCRule-LevelPolicyChange>Failure</AuditMPSSVCRule-LevelPolicyChange>
   <ForceAuditPolicySubcategory>Enabled</ForceAuditPolicySubcategory>
   <Sysmon>Installed</Sysmon>
   <CAPI2>EnabledGoodLogSize</CAPI2>
</AuditPolicies>
```

14.1.2 Approach

Read Resultant Set of Policies

Research was done to read the corresponding audit policy configurations from the system. At the beginning, the approach was to read the required configurations using the command `auditpol`. [17] This command can be used to read out and manipulate the currently valid information on the audit policies. However, the manipulation of the audit policies is not necessary within the tool and can be ignored. The command provides exactly the information needed to fulfill this use case:

Listing 1.5: auditpol

```
PS C:\Windows\system32> auditpol /get /category:Logon/Logoff
System audit policy
Category/Subcategory                    Setting
Logon/Logoff
   Logon                                Success and Failure
   Logoff                               Success and Failure
   Account Lockout                      No Auditing
   IPsec Main Mode                      No Auditing
   IPsec Quick Mode                     No Auditing
   IPsec Extended Mode                  No Auditing
   Special Logon                        Success and Failure
   Other Logon/Logoff Events            No Auditing
   Network Policy Server                No Auditing
   User / Device Claims                 No Auditing
   Group Membership                     No Auditing
```

Unfortunately, this output is not very ideal for a suitable further processing and analysis of the current configuration. The return value of the command is an ordinary array filled with corresponding strings and therefore the complete array should have been checked for correct content by string comparisons. Furthermore, the command `auditpol` does not offer the possibility of remote configuration with regard to an extension of the tool to a whole fleet of computers. For this reason, the idea of building the tool on the basis of this command was rejected.

Further research has shown that Microsoft provides a Resultant Set of Policies (RSoP) [15] for reading audit policies. This can also be accessed via a PowerShell command. Microsoft offers the command `Get-GPResultantSetOfPolicy` [18] for this purpose. This command can be used to generate an XML-based report of the currently valid GPOs. Since traversing an XML-based file via PowerShell proves to be very simple, this variant is preferable to the `auditpol` command. After a short test, it quickly became clear that the generated XML provides all necessary information for the further analysis.

Analyse Audit Policies

The current configuration of the system's audit policies is then to be evaluated from the temporarily cached file. The basis for this provides the section "11.13 Correlation: Advanced Audit Policy Setting and Event Log IDs" based on "11.10 JPCERT/CC - Detecting Lateral Movement through Tracking Event Logs".

14.1.3 Implementation

This section describes the implementation of `GetAndCheckAuditPolicies` in detail. For this purpose, the following is referred to in the section 14.1.1 Result described schedule.

The first step is to read the RSoP from the local system with the command `Get-GPResultantSetOfPolicy`. The XML that is retrieved is then temporarily cached in the execution path of the script and read in again for further processing.

Listing 1.6: Get-GPResultantSetOfPolicy

```
$currentPath = (Resolve-Path .\).Path
$pathRSOPXML = $currentPath + "\LocalUserAndComputerReport.xml"
Get-GPResultantSetOfPolicy -ReportType Xml -Path $pathRSOPXML | Out-Null;
[xml]$rsopResult = Get-Content $pathRSOPXML;
```

The generated XML is an extraction of the GPOs and contains only the configurations set from them. Afterwards the analisis begins and the entries are searched in the XML file, in which the required configurations for the "Advanced Audit Policies" are stored (see figure 1.6).

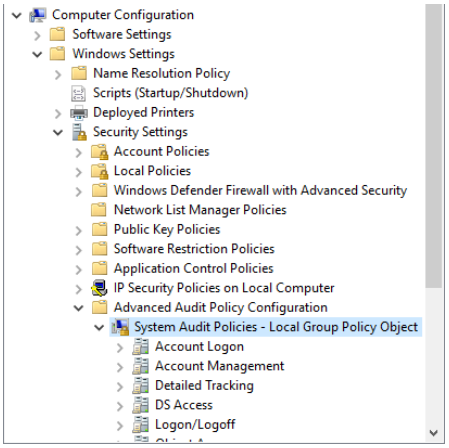

Figure 1.6: GPO - Advanced Audit Policies

The first step of the analysis is to search for missing configurations. The system iterates over the queried `AuditSettings` and searches for missing configurations.

Listing 1.7: GetAndCheckAuditPolicies: Search missing configurations

```
$auditSettings = $rsopResult.Rsop.ComputerResults.ExtensionData.Extension.AuditSetting

foreach($auditSettingSubcategoryName in $auditSettingSubcategoryNames) {
   if($auditSettings.SubcategoryName -notcontains $auditSettingSubcategoryName){
      # Write to XML
   }
}
```

After checking for missing configurations the set settings are checked for correctness according to JPCERT/CC - Detecting Lateral Movement through Tracking Event Logs.

```
foreach($auditSetting in $auditSettings) {
   # check if $auditSetting is not Null
   if($auditSetting) {
      # try to get audit setting value, if not No Auditing is configured system will
          throw exception -> set value to 0 for NoAuditing
      try {
         $auditSettingValue = $auditSetting.SettingValue
      }
      catch {
         $auditSettingValue = 0
      }
      $auditSubcategoryName = $auditSetting.SubcategoryName
      switch ($auditSettingValue) {
         NoAuditing {
            # Write SubcategoryName as tag with value 'NoAuditing' to XML
            continue
         }
         Success {
            # Write SubcategoryName as tag with value 'Success' to XML
            continue
         }
         Failure {
            # Write SubcategoryName as tag with value 'Failure' to XML
            continue
         }
         SuccessAndFailure {
            # Write SubcategoryName as tag with value 'SuccessAndFailure' to XML
            continue
         }
         Default { continue }
      }
   }
}
```

After checking the audit policies which can be configured through "Advanced Audit Policies" the next step is to verify if the setting "Audit: Force audit policy subcategory settings (Windows Vista or later) to override audit policy category settings" is enabled as considered in section "11.14.4 AuditPolicy". This had to be solved via the registry, because this information is not available in the RSoP.

Listing 1.8: Function IsForceAuditPoliySubcategoryEnabeled

```
Function IsForceAuditPoliySubcategoryEnabeled {
   $path = "HKLM:\System\CurrentControlSet\Control\Lsa"
   $name = "SCENoApplyLegacyAuditPolicy"
   try {
      $auditPoliySubcategoryKey = Get-ItemProperty -Path $path -Name $name -ErrorAction
          Stop
      if ($auditPoliySubcategoryKey.SCENoApplyLegacyAuditPolicy -eq 1) {
         return "Enabled"
      } else {
         return "Disabled"
      }
   }
   catch {
      return "NotDefined"
   }
}
```

The next step is to check if Sysmon as a service (also not contained in the RSoP) is installed and if so is it running or not.

Listing 1.9: Function IsSysmonInstalled

```
Function IsSysmonInstalled {
   $service = $null

   try {
      $service = Get-Service -Name Sysmon*
   } catch {
      return "NotInstalled"
   }

   if ($service.Status -ne "Running") {
      return "InstalledNotRunning"
   } else {
      return "Installed"
   }
}
```

As a last step the script is checking whether CAPI2 is enabled and has the right minimum log size of 4MB. Unfortunately, this information is also not available via the RSoP. Therefore the command `wevtutil` is used to query CAPI2 in the event log. The reason for this is that CAPI2 can only be enabled via the Event Viewer. [19]

Listing 1.10: Function IsCAPI2Enabled

```
Function IsCAPI2Enabled {
   [xml]$capi2 = wevtutil gl Microsoft-Windows-CAPI2/Operational /f:xml
   $capi2Enabled = $capi2.channel.enabled
   $capi2LogSize = $capi2.channel.logging.maxsize -as [int]
   if ($capi2Enabled -eq "true" -and $capi2LogSize -ge 4194304) {
      return "EnabledGoodLogSize"
   } elseif ($capi2Enabled -eq "true" -and $capi2LogSize -lt 4194304) {
      return "EnabledBadLogSize"
   } else {
      return "Disabled"
   }
}
```

At the end of the script all temporary files are removed.

14.2 Script: GetAndCompareLogs

This section describes the implementation of the "UC03 - Find Event Logs" as well as "UC04 - Analyse Found Event Logs". Both use cases were fulfilled in the PowerShell script "GetAndCompareLogs". Here is a description how the use cases were implemented.

14.2.1 Result

The script "GetAndCompareLogs", where both use cases were implement, runs as follows:

- Reading and caching the Event Logs "System" & "Security"
- Filter cached Logs by EventID, group EventIDs that occur more than once. Found EventIDs are exported as "CSV"
- Checking and caching whether a list of EventIDs from "Application and Service" Logs can be read out
- Export result set of found EventIDs as "CSV"
- Import list of found Event Logs and compare it with the predefined checklist
- Result of the comparison is written into an "XML" file
- Import and compare found Application and Service Logs with predefined checklist
- Result of the comparison is written into the same "XML" as before

The now no longer needed CSV files are deleted. The XML with the result set is now available for any further processing. A result could possibly look like the following listing:

Listing 1.11: Example Result Audit Policy Analysis

```
<?xml version="1.0"?>
<Logs>
   <EventLogsID>
      <6>present</6>
      <21>missing</21>
      <24>missing</24>
      <102>missing</102>
      <104>missing</104>
      <106>missing</106>
      <201>missing</201>
      <4624>present</4624>
      <4634>present</4634>
      <4648>present</4648>
      <4656>present</4656>
      ...
   </EventLogsID>
   <AppAndServID>
      <106>present</106>
      <200>present</200>
      <129>present</129>
      <201>present</201>
      <102>present</102>
      <6>missing</6>
      <169>missing</169>
      <21>present</21>
      <24>present</24>
   </AppAndServID>
</Logs>
```

14.2.2 Approach

Get Event Logs
After research was done on how to read out the Event Logs "System" and "Security" the desicon was made to use to PowerShell command`Get-EventLog` [16]. This command allows to read out the whole EventLog by the LogName or also to search after a specific EventID. The first approach was to search for each EventID individually. The EventIDs to search for were taken from the JPCERT Appendix B in the "Detecting Lateral Movement through Tracking Event Logs" report. [7]. The script run successfully, but the runtime was not practicable. It took over 5 minutes to search for all EventIDs in an Event Log of the size of about 37 000 Logs, or in other words 300 Kb. The developers than started to calculate the worst case scenario, in this case the worst case scenario is that none of the searched EventIDs is found in the EventLog. There are n EventIDs in the checklist and m entries in the EventLogs, if no EventID is found, every entry is called m times. That results in O(n*m). The developers decided to cache the Event Logs, reducing the runtime to O(m). The cached Logs are then grouped into EventIDs and export into a "CSV" file.

To read out the "Application and Service" Logs we can not use Get-EventLog. The first approach used the `Get-WinEvent` [20] command. The logic stayed the same, read out all events, group and export them into a 'CSV' file. Unfortunately the Get-WinEvent is very slow, it took over 10 minutes to read out just under 6000 logs. The developers found an other, much quicker command called `wevtutil` [21]. Unfortunately it is not quite simple to read out all Logs, for that reason each EventID will be searched if it appeared. Unlike Get-EventLog, this is not a problem because the command is faster, the EventIDs are more likely to occur and the amount of Logs is smaller. On the testing environment with a machine with 4 GB Ram and an Intel Xeon E5 with 2 cores it took about 10 seconds to check for 9 EventIDs in 15 000 Log entries. If and EventID was found it was added to an ArrayList, after all IDs are checked the file is exported as a 'CSV'.

Analyse Found Event Logs
To analyse the occurred EventIDs the two generated "CSV" files are imported into the PowerShell script. The respective checklists, which are based on the JPCERT/CC - Detecting Lateral Movement through Tracking Event Logs, are embedded in the script. Each id from the checklist is checked if it is present in the respective CSV file. Is this the case, the id is added to the XML-file and marked as present. Did the id no occur in the it will be added and marked as missing. The file looks like the example in "Result" shown.

14.2.3 Implementation

This section describes the implementation of `GetAndCompareLogs` in detail. For this purpose, the following is referred to in the section "14.2.1 Result" described schedule.

The first step is to read out the "System" and "Security" Logs. To achieve this goal the command `Get-EventLog` is used in the first part of the function `GetEventLogsAndExport`.

Note The code has been adapted for better readability and easier understanding

Listing 1.12: Function GetEventLogsAndExport Part 1

```
1  $logNames = @("System", "Security")
2  $eventLogs = New-Object System.Collections.ArrayList
3
4  Function GetEventLogsAndExport{
5     foreach($log in $logNames){
6     $eventLogs += Get-EventLog -LogName $log
7     }
8  ...
```

The second part of the function filters the EventIDs from the chaced logs. Subsequently, multiple EventIDs are grouped together.

Listing 1.13: Function GetEventLogsAndExport Part 2

```
1  $currentFolder = (Resolve-Path .\).Path
2  $exportEventLogsIntoCSV=$currentFolder + "\myeventlogs.csv"
3
4  $eventLogs| Select EventID -Unique |Export-CSV $exportEventLogsIntoCSV -NoTypeInfo
       -Encoding UTF8
5  }
```

After the export the function `GetApplicationAndServiceLogs` is called. As before, we divide this function into two parts, first how to get the data. The same procedure is used three times, for the "TaskScheduler", "WindowsRemoteManagement" and "LocalSessionManager". Due to the fact that the code is very similar it is only shown once. To search for the EventIDs `wevtutil` is used.

Listing 1.14: Function GetApplicationAndServiceLogs Part 1

```
1  $appAndServLogs = New-Object System.Collections.ArrayList
2  $idsForTaskScheduler = ("106", "200", "129", "201", ...)
3
4  $appAndServLogs += "EventID"
5
6  Function GetApplicationAndServiceLogs{
7
8  foreach($id in $idsForTaskScheduler){
9  if(wevtutil qe Microsoft-Windows-TaskScheduler/Operational
       /q:"*[System[(EventID="$id" )]]" /uni:false /f:text){
10    $appAndServLogs += $id
11     }
12 }
13 ...
```

After all three Logs were checked and all found EventIDs were added, the information is exported into a "CSV"-file.

Listing 1.15: Function GetApplicationAndServiceLogs Part 2

```
$exportApplicationAndServiceLogsIntoCSV = $currentFolder +
    "\myapplicationandsiencelogs.csv"

$appAndServLogs | Out-File -FilePath $exportApplicationAndServiceLogsIntoCSV
}
```

The next point on the list is importing the found "EventLogs" and "Service And Application" Logs. Due to the similarity we only show one code.

Listing 1.16: Function ImportCompareExport

```

$eventLogIdsToCheck = (6, 21, 24, 102, 104, 106, 129, ...

$resultXML = "resultOfEventLogs.xml"
$xmlWriter = New-Object System.XMl.XmlTextWriter($resultXML,$Null)
$xmlWriter.Formatting = "Indented" #$xmlWriter
$xmlWriter.Indentation = 1
$xmlWriter.IndentChar = "‘t"
$xmlWriter.WriteStartDocument()
$xmlWriter.WriteStartElement("Logs")
$xmlWriter.WriteStartElement("EventLogsID")

$importEventLogs = $exportEventLogsIntoCSV
$myEventLogs = Import-Csv $importEventLogs -Encoding UTF8

Function ImportCompareExport{
   foreach($id in $eventLogIdsToCheck){
     if( $myEventLogs | where {$_.EventID -eq $id}){
      $xmlWriter.WriteStartElement("EventID" +$id)
      $xmlWriter.WriteValue("present")
      $xmlWriter.WriteEndElement()
         }
         else{
            $xmlWriter.WriteStartElement("EventID" +$id)
            $xmlWriter.WriteValue("missing")
            $xmlWriter.WriteEndElement()
         }
   }
   $xmlWriter.WriteEndElement()
...
}

$xmlWriter.WriteEndElement()
$xmlWriter.WriteEndDocument()
$xmlWriter.Flush()
$xmlWriter.Close()
```

The same happens with the "App and Service" Logs in the `GetApplicationAndServiceLogs` function. At the end all temporary files are deleted.

15 Results

16 Conclusion

Glossary

List of Figures

List of Tables

Bibliography

[1] harmj0y Andrew Robbins, Rohan Vazarkar. BloodHound - Wiki. https://github.com/BloodHoundAD/BloodHound/wiki, September 2018.

[2] Microsoft. Microsoft Security Complience Toolkit. https://docs.microsoft.com/en-us/windows/security/threat-protection/security-compliance-toolkit-10, June 2018.

[3] JPCERT/CC. LogonTracer. https://github.com/JPCERTCC/LogonTracer, September 2018.

[4] Shusei Tomonaga. Visualise Event Logs to Identify Compromised Accounts - LogonTracer -. https://blog.jpcert.or.jp/2017/11/visualise-event-logs-to-identify-compromised-accounts—logontracer-.html , November 2017.

[5] Microsoft. Monitoring Active Directory for Signs of Compromise | Microsoft Docs. https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/monitoring-active-directory-for-signs-of-compromise, May 2017.

[6] Microsoft. Appendix L: Events. https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/appendix-l–events-to-monitor, July 2018.

[7] JPCERT/CC. Detecting Lateral Movement through Tracking Event Logs. , June 2017.

[8] MITRE ATT&CK. MITRE ATT&CK Website. https://attack.mitre.org/, September 2018.

[9] Olaf Hartong. sysmon-modular. https://attack.mitre.org/, October 2018.

[10] Nader Shalabi. Sysmon Tools. https://github.com/olafhartong/sysmon-modular, October 2018.

[11] ipstack. Locate and identify website visitors by IP address. https://ipstack.com/, October 2018.

[12] Abe, Shingo. Detecting Lateral Movement in APTs - Analysis Approach on Windows Event Logs Introduction to JPCERT / CC. June 2016.

[13] Microsoft. Advanced security auditing FAQ. https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/advanced-security-auditing-faq, April 2017.

[14] Daniel Scott-Raynsford. Converting a PowerShell Project to use Azure DevOps Pipelines. https://www.powershellmagazine.com/2018/09/20/converting-a-powershell-project-to-use-azure-devops-pipelines/, September 2018.

[15] Microsoft. RSoP - Resultant Set of Policy. https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc772175(v=ws.11), February 2017.

[16] Microsoft. Get-EventLog. https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-eventlog?view=powershell-5.1, October 2018.

[17] Microsoft. auditpol. https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/auditpol, September 2017.

[18] Microsoft. Get-GPResultantSetOfPolicy. https://docs.microsoft.com/en-us/powershell/module/grouppolicy/get-gpresultantsetofpolicy?view=win10-ps, October 2018.

[19] Microsoft. Troubleshooting PKI Problems on Windows Vista - CAPI2 Diagnostics in Windows Vista. https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-vista/cc749296(v=ws.10)#capi2-diagnostics-in-windows-vista, July 2008.

[20] Microsoft. Get-WinEvent. https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.diagnostics/get-winevent?view=powershell-6, October 2018.

[21] Microsoft. wevtutil. https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/wevtutil, October 2017.