



**HSR**  
**HOCHSCHULE FÜR TECHNIK**  
**RAPPERSWIL**

**COMPUTER SCIENCE**

# Readinizer

BACHELOR THESIS

SPRING TERM 2019

## **Authors:**

Claudio MATTES  
claudio.mattes@hsr.ch

Lukas KELLENBERGER  
lukas.kellenberger@hsr.ch

## **Supervisor:**

Cyrill BRUNSCHWILER  
University of Applied Sciences Rapperswil  
cyrill.brunschwiler@hsr.ch

## **External Co-Examiner:**

Dr. Christian FOLINI  
netnea.com  
christian.folini@time-machine.ch

## **Internal Co-Examiner:**

Prof. Dr. Olaf ZIMMERMANN  
University of Applied Sciences Rapperswil  
olaf.zimmermann@hsr.ch

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

June 12, 2019

# Abstract

## Introduction

The number of cyber-attacks where malicious code is used has massively increased recently. These attacks not only settle on the infected system, but can also infect other systems through lateral movements in the network. The outcome is often the complete infiltration of the organization due to the use of advanced persistent threats (APT). Although the configuration of these targeted networks varies depending on the organization, common patterns in the attack methods can be detected. In the analysis of such patterns and events, information and time are key factors to success. Hence, readiness and a fast access through an entire environment for such an event is a decisive factor.

## Approach

The main aspect of this project was to analyze readiness of the configured settings - through an entire Active Directory (AD) environment - and give a statement to improve those. On the other hand, but still with a significant importance, an optimization part was planned to improve the present state of the environment. In a first step, a benchmarking of the defined recommended audit settings from the previous Proof of Concept (PoC) was performed against several Computer Emergency Response Teams (CERT) all over the world. Simultaneously, architectural and design decisions for the application have been made. After further research in upcoming topics, the construction of the application “Readinizer” for the analysis part was performed. The construction phase also contained the optimization part. Last but not least, manuals for the application and the entire project have been documented.

## Result

The application “Readinizer” analyzes an entire AD forest and gathers information about all domains, sites, organizational units (OU) and member computers/servers. As soon as this information is gathered and all relationships between these objects are resolved, the “Readinizer” calls one computer/server of each OU to receive a Resultant Set of Policies (RSoP). A RSoP is a summary of the applied computer settings that were made locally or distributed via group policy objects (GPO). Since an OU has the highest precedence when applying GPOs, it is sufficient to query only one computer of each OU. Then an analysis is performed for each received RSoP, comparing the current settings in the AD forest with the recommended settings - based on the benchmark. The result of the analysis is then presented to the user in form of a percentage figure whereby a tree structure of the forest depicts the analyzed RSoPs and gives a first view of the readiness. In addition, the user has the possibility to simultaneously perform a Sysmon check. Sysmon is a tool by Mark Russinovich which logs the same events as the default event logger but where the executables are hashed. Hence, compromise of such executables can be detected. The user can then drill down the RSoPs to a detailed view over all applied / recommended settings and which GPO applied those settings. With the optimization part of the “Readinizer”, the distribution of Sysmon to an entire fleet is simplified for the user, as well as the setup of central logging by Windows Event Forwarding - with appropriate templates - is made available in the form of manuals. The “Readinizer” also includes a GPO of recommended settings which can be imported.

# Management Summary

## Initial Situation

The number of cyber-attacks where malicious code is used, which not only settles on the infected system, but also infects other systems in the network, has massively increased recently. The outcome is often the complete infiltration of the organization. In the analysis of such an event, information and time are key factors to success. Consequently, readiness for such an event is a decisive factor.

This bachelor thesis was preceded by a study thesis in which a proof of concept (PoC) was developed. The PoC checks the readiness of a system using the Windows logging settings. One goal of this Bachelor thesis was to extend the PoC and create a tool that can determine the readiness of a complete Windows network. Furthermore, guidelines for improving the readiness of a system are to be provided.

## Procedure

The project was initially limited to Windows machines running on the operating system Windows 10 Pro or Windows Server 2016. The project was handled according to common project management and software engineering principles. Unlike in the PoC, where we chose PowerShell to realize the project, we chose C# as the programming language. The reason why C# was chosen is that it is close to the Microsoft operating system and it is better suited to developing a tool than PowerShell. Due to the complexity of the domain model, we decided to use LocalDB to store the data. The decision on how to display the gathered and analyzed information was made in favour of Windows Presentation Foundation.

During the construction phase the “Readinizer” was developed, a tool that collects both forest and domain information, checks and analyzes Windows logging settings to give an impression of the readiness of a system. In addition, manuals were written on how to improve the readiness of a Windows network, for example the central collection and storage of Windows event logs.

## Results

The resulting tool can analyze a complete forest with all its domains and subdomains. The readiness of the system is then illustrated using colored graphs, it also shows where which setting is not correct and how it should be changed. The user is offered a Group Policy Object which he can embed in his domain to increase the readiness, additionally there are manuals available which pursue the same goal.

## Outlook

The “Readinizer” is a small useful tool to get an overall impression of the Readiness of a system. Nevertheless, there are still some improvements or extensions to be performed. For example, the “Readinizer” is currently still dependent on a dynamic link library of the Remote Server Administrator Tool, it would be advantageous to make the tool independent of it. Additionally, one could analyze the sysmon logs in more detail and create a corresponding sysmon-config file. It would also be useful if the user rights of the active user are checked before the analysis is performed and other credentials can be specified. On a larger scale, one can combine all three parts “Readinizer, Visualizer and Optimization” into a monitoring tool.



# Contents

<b>Abstract</b>	<b>I</b>
Introduction . . . . .	I
Approach . . . . .	I
Result . . . . .	I
 <b>Management Summary</b>	 <b>II</b>
Initial Situation . . . . .	II
Procedure . . . . .	II
Results . . . . .	II
Outlook . . . . .	III
 <b>Table of Contents</b>	 <b>IX</b>
 <b>I Technical Report</b>	 <b>X</b>
1 Introduction and Overview . . . . .	1
1.1 Purpose and Scope . . . . .	1
1.2 Audience . . . . .	1
1.3 Document Structure . . . . .	1
2 Analysis . . . . .	2
2.1 Windows Network Environment . . . . .	2
2.1.1 Active Directory Domain Services . . . . .	2
2.2 Group Policy Objects . . . . .	5
2.2.1 Filtering and Scope of GPOs . . . . .	6
2.2.2 Inheritance and Processing Rules of GPOs . . . . .	7
2.2.3 Resultant Set of Policies . . . . .	9
2.2.4 GPOs Storage Location . . . . .	9
3 Benchmark . . . . .	10
3.1 Computer Emergency Response Team - Europe . . . . .	10
3.1.1 Comparision PoC - CERT-EU . . . . .	12
3.1.2 Conclusion CERT-EU . . . . .	12
3.2 National Security Agency . . . . .	13
3.2.1 Comparision PoC - NSA . . . . .	16
3.2.2 Conclusion NSA . . . . .	19
3.3 Australian Cyber Security Center . . . . .	20
3.3.1 Comparision PoC - ACSC . . . . .	22
3.3.2 Conclusion ACSC . . . . .	24
3.4 MITRE Adversarial Tactics, Techniques and Common Knowledge . . . . .	25

	3.4.1	Comparision PoC - MITRE ATT&CK . . . . .	25
	3.4.2	Conclusion MITRE ATT&CK . . . . .	29
3.5		SysAdmin, Networking and Security Digital Forensics and Incident Response .	30
	3.5.1	Comparison PoC - SANS . . . . .	30
	3.5.2	Conclusion SANS . . . . .	32
3.6		Overall conclusion . . . . .	33
	3.6.1	GPO Settings Readinizer . . . . .	33
	3.6.2	Additional Readinizer Settings . . . . .	37
4		Test Environment . . . . .	38
	4.1	Domain User . . . . .	39
	4.2	Domain readinizer.ch . . . . .	40
	4.3	Difficulties . . . . .	41
5		Design . . . . .	42
	5.1	Domain Analysis . . . . .	42
	5.2	Graphical User Interface Design . . . . .	44
	5.2.1	Start screen . . . . .	44
	5.2.2	Result overview . . . . .	45
	5.2.3	Result per domain . . . . .	46
	5.2.4	Result per RSoP . . . . .	47
	5.2.5	Navigation bar . . . . .	47
	5.3	Data Model . . . . .	48
	5.4	Differences to the Domain Model . . . . .	49
	5.4.1	General . . . . .	49
	5.4.2	Rsop . . . . .	49
	5.4.3	RsopPot . . . . .	49
	5.4.4	GPOSetting . . . . .	49
6		System Architecture . . . . .	50
	6.1	Use Cases Readinizer (UC- <i>R<sub>n</sub></i> ) . . . . .	50
	6.1.1	UC-R1 - Discovering all organizational units and their members . . .	51
	6.1.2	UC-R2 - Collecting Resultant Set of Policies of a fleet . . . . .	52
	6.1.3	UC-R3 - Analyzing the collected data . . . . .	53
	6.1.4	UC-R4 - Visualize the analyzed data . . . . .	54
	6.2	Use Cases Optimizer (UC- <i>O<sub>n</sub></i> ) . . . . .	55
	6.2.1	UC-O1 - Provide a recommended Group Policies . . . . .	55
	6.2.2	UC-O2 - Provide manual for fleet-wide Sysmon installation . . . . .	55
	6.2.3	UC-O3 - Provide manual for fleet-wide central logging installation . .	56
	6.3	Non Functional Requirements (NFR) . . . . .	56
	6.4	Logical Architecture - Package Diagram . . . . .	57
	6.4.1	Design Decisions . . . . .	57
	6.5	System Architecture - Deployment Diagram . . . . .	59
	6.5.1	Rejected System Architecture . . . . .	59
	6.5.2	Accepted System Architecture . . . . .	59
	6.6	Technologies . . . . .	60
	6.6.1	Chosen Technologies & Frameworks . . . . .	60
	6.6.2	Rejected Technologies . . . . .	61
7		Implementation . . . . .	63
	7.1	Application Structure and Central Components . . . . .	63
	7.1.1	Dependency Injection . . . . .	63
	7.1.2	Generic Repository . . . . .	64

	7.1.3	Unit of Work . . . . .	64
7.2		Frontend Implementation (Presentaion Layer) . . . . .	65
	7.2.1	ApplicationView(Model) . . . . .	66
7.3		UC-R1: Discovering all Organizational units and their members . . . . .	68
	7.3.1	Logic flow . . . . .	68
	7.3.2	Implementation - ADDomainService . . . . .	69
	7.3.3	Implementation - SiteService . . . . .	71
	7.3.4	Implementation - OrganizationlUnitService . . . . .	72
	7.3.5	Implementation - ComputerService . . . . .	73
7.4		UC-R2: Collecting Resultant Set of Policies of a fleet . . . . .	74
	7.4.1	Logic flow . . . . .	74
	7.4.2	Implementation - PingService . . . . .	74
7.5		Implementation - RSoPService . . . . .	75
	7.5.1	Implementation - SysmonService . . . . .	77
7.6		UC-R3: Analysing the collected data . . . . .	78
	7.6.1	Logic flow . . . . .	78
	7.6.2	Implementation - AnalysisService . . . . .	78
	7.6.3	Implementation - RsopPotService . . . . .	89
7.7		UC-R4: Visualize the analyzed data . . . . .	93
	7.7.1	Logic flow . . . . .	93
	7.7.2	Implementation Forest Overview . . . . .	93
	7.7.3	Implementation Domain Overview . . . . .	94
	7.7.4	Implementation GISS Overview . . . . .	94
	7.7.5	Implementation OU Overview . . . . .	95
	7.7.6	Implementation Sysmon Overview . . . . .	95
7.8		UC-O1: Provide a recommended Group Policy Object . . . . .	96
7.9		UC-O2: Provide manual for fleet-wide Sysmon installation . . . . .	96
7.10		UC-O3: Provide manual for fleet-wide central logging installation . . . . .	96
8		Conclusion and Outlook . . . . .	97
	8.1	Conclusion Achieved Work . . . . .	97
	8.2	Conclusion Technologies and Frameworks . . . . .	98
	8.3	Outlook . . . . .	99
	8.3.1	Dependencies . . . . .	99
	8.3.2	User Permissions . . . . .	99
	8.3.3	Sysmon . . . . .	99
	8.3.4	Parallelisation . . . . .	99
	8.3.5	Log Pattern . . . . .	99
	8.3.6	Monitoring . . . . .	99
	8.3.7	Monitoring - Anomalies . . . . .	99

<b>Glossary</b>	<b>XI</b>
<b>Listings</b>	<b>XVI</b>
<b>List of Figures</b>	<b>XVIII</b>
<b>List of Tables</b>	<b>XX</b>
<b>Bibliography</b>	<b>XXVI</b>
<b>II Appendix</b>	<b>XXVII</b>
<b>Testing and Code Metrics</b>	<b>XXVIII</b>
Testing . . . . .	XXVIII
System Tests . . . . .	XXVIII
Unit Testing . . . . .	XXXIII
Code Metrics . . . . .	XXXIV
<b>Time Management</b>	<b>XXXV</b>
Time by Activity Type . . . . .	XXXV
Time by Phase . . . . .	XXXVI
Inception . . . . .	XXXVI
Elaboration . . . . .	XXXVI
Construction . . . . .	XXXVII
Transition . . . . .	XXXVII
Sprints - Estimated Time vs. Actual Time . . . . .	XXXVIII
Conclusion Time Management . . . . .	XXXVIII
<b>Task Definition</b>	<b>XXXIX</b>
Einführung . . . . .	XXXIX
Aufgabe . . . . .	XXXIX
Abgrenzung . . . . .	XXXIX
Tätigkeiten . . . . .	XXXIX
Vorgehen . . . . .	XL
Anforderungen . . . . .	XL
Technologien . . . . .	XL
Infrastruktur . . . . .	XLI
Erwartete Resultate . . . . .	XLI
In elektronischer Form . . . . .	XLI
Auf Papier . . . . .	XLI
Termine . . . . .	XLI
Zeitplan und Meilensteine . . . . .	XLII
Betreuung . . . . .	XLII

Kontakt . . . . .	XLII
Unterschriften . . . . .	XLII

<b>Sysmon Deployment Through GPO</b>	<b>I</b>
1.1 Overview . . . . .	I
1.2 Organization of the Manual . . . . .	I
1.1 What is Sysmon? . . . . .	II
1.2 Why Sysmon? . . . . .	II
1.1 Requirements . . . . .	III
1.2 Limitations . . . . .	III
1.1 Domain Folder . . . . .	IV
1.2 Sysmon Executable . . . . .	IV
1.3 Sysmon Configuration File . . . . .	IV
1.4 Batch File . . . . .	V
1.4.1 What does it do... . . . .	V
1.5 Group Policy Object . . . . .	VII

<b>Windows Event Forwarding Deployment Fleet-Wide</b>	<b>I</b>
1.1 Overview . . . . .	I
1.2 Organization of the Manual . . . . .	I
1.1 What is Windows Event Forwarding? . . . . .	II
1.2 Advantages with WEF . . . . .	II
1.1 Requirements . . . . .	III
1.2 Limitations . . . . .	III
1.3 Additional Information . . . . .	III
1.1 Windows Event Collector . . . . .	IV
1.1.1 Enable WinRM . . . . .	IV
1.1.2 Enable Event Forwarding . . . . .	V
1.1.3 Group Policy Objects for the subscribers . . . . .	V
1.1.4 WEF Subscription . . . . .	IX
2 Encryption of Event Logs . . . . .	XIII

<b>Installation &amp; User Manual</b>	<b>I</b>
1.1 Overview . . . . .	I
1.2 Organization of the Manual . . . . .	I
2.1 Operating System . . . . .	II
2.2 User Authorizations . . . . .	II
2.3 Firewall Settings . . . . .	II
2.4 Pre-Installed Software . . . . .	III
2.4.1 Remote Server Administration Tool . . . . .	III
2.4.2 SQLLocalDB . . . . .	IV
3.1 Download . . . . .	VI
3.2 Installation . . . . .	VI
3.2.1 Installer . . . . .	VI
3.2.2 Portable Application . . . . .	VI
4.1 Starting the Readinizer . . . . .	VII
4.1.1 Home Screen . . . . .	VII

---

4.1.2	Forest Result Screen . . . . .	VIII
4.1.3	Domain Result Screen . . . . .	IX
4.1.4	Group of Identical Security Settings Result Screen . . . . .	X
4.1.5	Organizational Unit Result Screen . . . . .	XI
4.1.6	Sysmon Result Screen . . . . .	XII
4.1.7	Navigationbar . . . . .	XIII

Part I

Technical Report

# 1 Introduction and Overview

## 1.1 Purpose and Scope

The key for a successful analysis in case of an advanced persistence threat (APT) and lateral movement in a network, is to have a solid event logging of all systems participating in the network. That was shown by the research that was carried out in the study thesis. The findings and recommended settings of this thesis were tested against guidelines from well-known cyber security specialist, such as the National Security Agency (NSA). The findings from this comparison have been used to improve the Readinizer.

This report also contains information about the Readinizer. What architectural decisions were made and why, what design approaches were followed, and detailed descriptions of how core elements were implemented.

## 1.2 Audience

This document is intended for software developers, security advisors and engineers who want to gain an insight into the relationship between APTs / lateral movement and event logging as well as basic information about Windows networks and Group Policy Objects (GPO). Furthermore, this document gives an insight about the Readinizer tool and the optimization manuals, which were the results of this bachelor thesis.

## 1.3 Document Structure

This technical report is structured in several sections:

- **Analysis:** Describes the research on Active Directory (AD) structure and Group Policy Objects.
- **Benchmark:** Compares the proof of concept (PoC) against other guidelines from reputable and well-known security organizations and draws conclusions for the Readinizer and its settings.
- **Design:** Describes the decisions for the tool which are derived from the analysis and addresses the problem domain.
- **Test Environment:** Describes the test environment used to test tools during the research and test the developed tool during the implementation.
- **System Architecture:** Based on the design, this section will answer the question how the problem domain will be fulfilled. Therefore, the use cases developed and the technology decisions are discussed.
- **Implementation:** Describes the structure and details about central components of the application as well as a detailed description of the frontend and backend implementation.
- **Testing and Code Metrics:** Describes how and what was tested. Gives an overview over the code metrics.
- **Conclusion and Outlook:** Is a retrospective of the thesis and makes statements about findings. In addition, an outlook on further development and expansion in this area will be drawn on the basis of this work.



## 2 Analysis

In this chapter we deal with the detailed structure of a Active Directory structure as well as how Group Policy Objects proceed and are deployed.

### 2.1 Windows Network Environment

One of the greatest challenges in this bachelor thesis is to run the application over entire fleets. For this reason, research was started on how a large Windows environment is built and will work. This section describes which components make up a typical Windows environment and how they work together.

#### 2.1.1 Active Directory Domain Services

The Active Directory Domain Services (AD DS), or short Active Directory, is responsible for structuring and storing network information and provides services to retrieve this stored information. It helps to organise network objects into a hierarchical collection of containers. The top logical container in this hierarchy is the forest, within a forest there are domain containers, and within a domain there are organization units (OU). Together these containers form the logical structure. An Active Directory provides domain, configuration, schema, and optional application information that applies to all containers.

**Organizational Units** Organizational Units are the smallest container units and also the smallest units to which group policy settings can be assigned. They are used to group objects, such as computers, resources and users, for administrative purposes. Organizational Units can not contain objects from other domains.

**Domains** Domains are created to manage the administrative requirements of the organization such as the delegation of administrative authority and operational requirements. A domain is a partition of an Active Directory; partitioning data helps to store data only where it is needed. This allows the directory to scale globally over the network without overloading the bandwidth. Every domain possesses administrators who have full control over every object in the domain, with their power being limited to their domain only.

Within a forest, a domain is a container whereby objects in this container trust each other and the security services inherently. If a new domain is created within a forest, a two-way transitive trust relationship is automatically created between the new domain and its parent. Thus, in a forest, every object of a domain trusts every other domain and its objects by a two-way transitive trust.

Each domain has at least one domain controller. It stores all of the domain's information on its domain partition; when changes occur, it stores them and reports them to all other domain controllers. Every domain controller has also two non domain partitions where they store forest-wide data, which includes the directory schema and configuration data.

**Domain Trees** A tree consists of several domains that share the same schema and configurations, forming a continuous namespace. Domains in a tree are not only bound by the same namespace but also share a trust relationship. Each tree has a root domain tree, all domains beneath it are its children and inherit its namespace. If there are domains that are superior to the tree root domain, this can only be the forest root domain. The definition of a forest root domain can be found in the chapter below.

**Forests** A forest is the top-level container and can therefore be used synonymously for Active Directory, it consists of at least one or several trees. Nevertheless, trees belonging to the same forest do not form a contiguous namespace. In other words, they make up a non-contiguous namespace which means each tree is based on a different Domain Name Systems (DNS) root domain name. All members in a forest share a common directory schema, directory configuration and global catalogue. All domains in the same forest trust each other inherently by a two-way transitive trust. As a result, every authentication request made from any domain to an other domain in the same forest will be granted. On the other hand, forests can be seen as security boundaries, non-members of the forest are not allowed to access the resources of the domains.

Like every tree has a tree root domain, the forest also has a forest root domain. The first domain created in a forest is defined as the forest root domain. There are two user groups within a forest root domain:

- Enterprise Admins
- Schema Admins

These user groups possess forest-wide administrative credentials. The forest root domain cannot be deleted but only restructured and renamed.

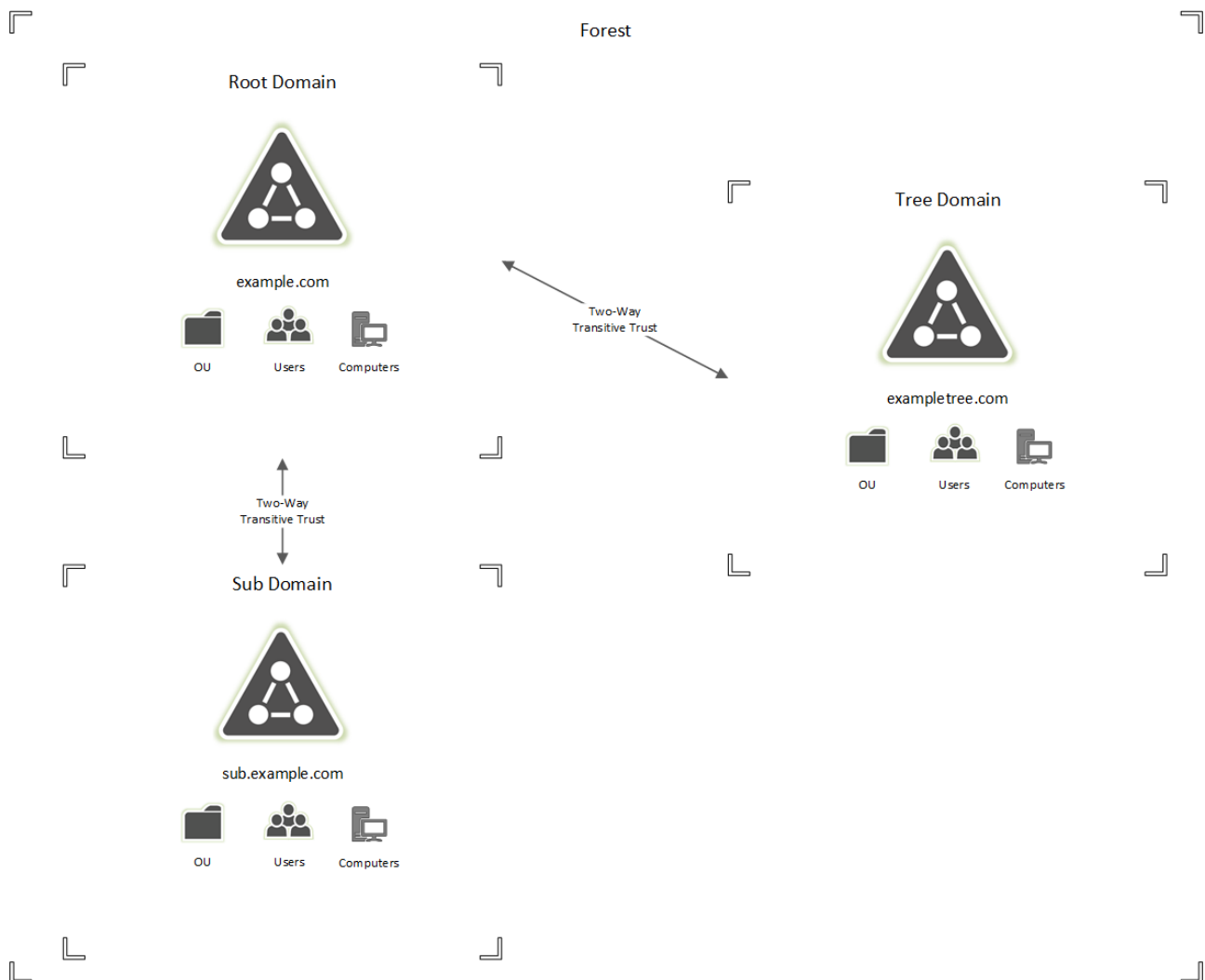


Figure 2.1: Windows Network Environment

**Site Objects** Sites are container objects with the goal to pool a physical network. This allows administrators to set up a domain controller within the site in order to redirect traffic to a physical nearby domain controller instead of burdening the network.

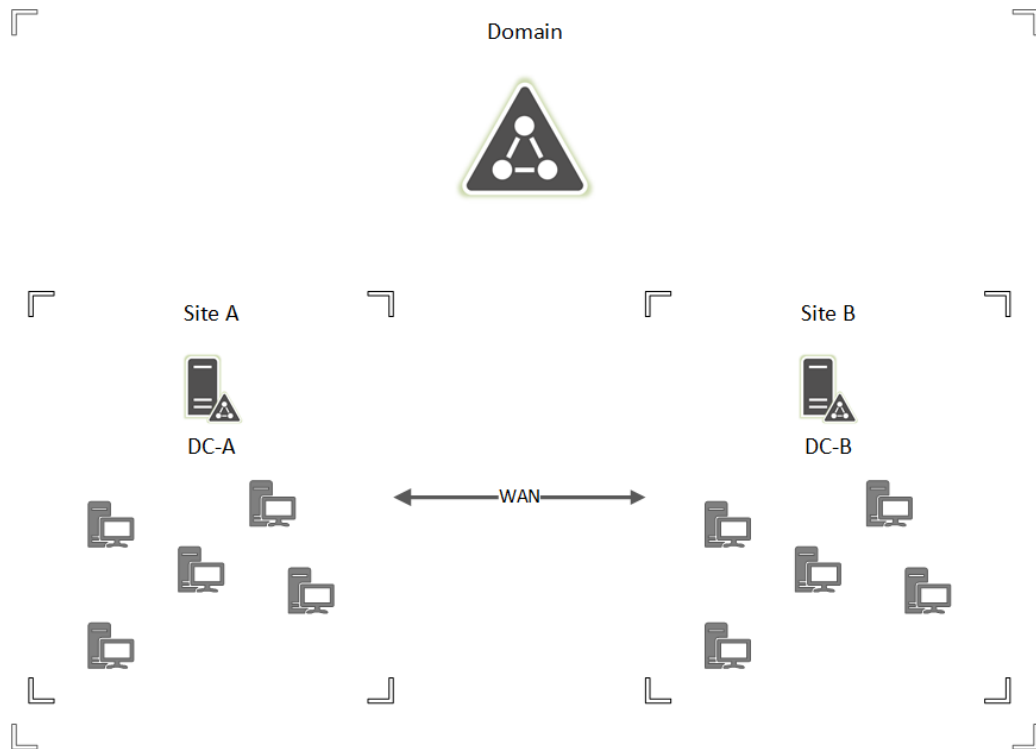


Figure 2.2: Site Objects

## 2.2 Group Policy Objects

Group Policy Objects (GPO) are used to apply one or more desired configurations or policy settings to a set of users and computers located in an Active Directory environment. Within this infrastructure, a Group Policy engine and multiple client-side extensions (CSEs) are used to write specific policy settings to each target computer. These computers will process the policies at each start of the computer and in regular refreshes during runtime.

The following figure 2.3 Group Policy Engine [1] shows the Group Policy engine and its interactions with other components. The table 2.1 Group Policy Engine [1] [2] describes the core components.

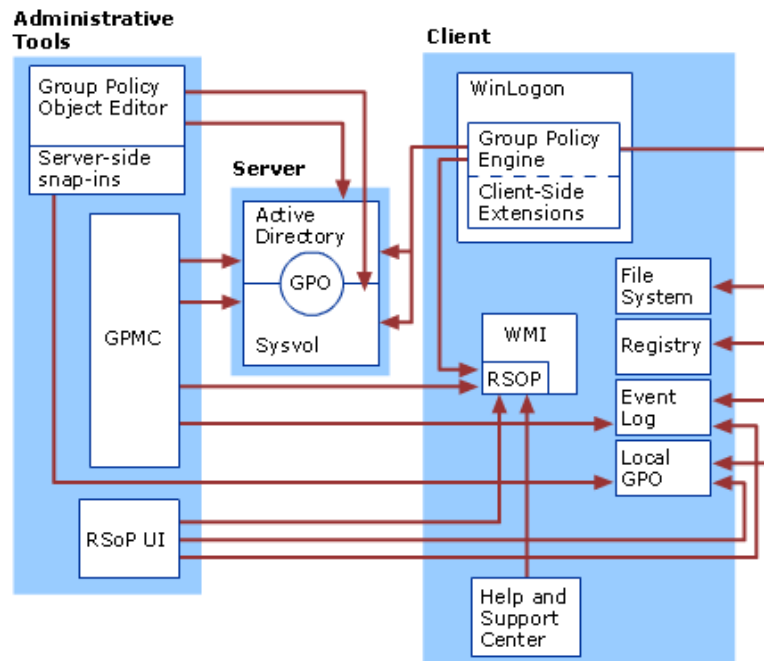


Figure 2.3: Group Policy Engine [1]

Component	Description
Group Policy Management Console (GPMC)	<i>[...] create, view, and manage GPOs [...]</i>
Group Policy Object Editor	<i>[...] set and configure the policy settings in GPOs</i>
Server (domain controller)	<i>[...] contains writable copy of Active Directory database, participates in Active Directory replication, controls access to network resources.</i>
Active Directory	Stores information about objects in a network and makes this information available for network participants.
Sysvol	<i>[...] is a set of folders containing important domain information, stored in a subfolder of the systemroot folder %\systemroot\sysvol\sysvol. The Sysvol contains the largest part of a GPO: the Group Policy template, which includes Administrative Template-based policy settings, security settings, script files and information regarding applications that are available for software installation. [...]</i>

Group Policy object (GPO)	A GPO is a set of Group Policy settings, stored at the domain level as a virtual object consisting information about the properties and its set values of a GPO.
Local Group Policy object	<i>The Local Group Policy object (Local GPO) is stored on each individual computer [...] are always processed, but are the least influential GPOs in an Active Directory environment [...]</i>
Registry	A database repository for information about a computer's configuration which can be controlled through Group Policies.
Winlogon	<i>A Component of the Windows operating system that provides interactive logon support. Is the service in which the Group Policy engine runs.</i>
Event Log	A Service that records events which occur at system runtime.
Windows Management Instrumentation (WMI)	<i>[...] supports monitoring and controlling of system resources through a common set of interfaces and provides a logically organized, consistent model of Windows operation, configuration, and status.</i>
Resultant Set of Policy (RSOP) infrastructure	<i>All Group Policy processing information collected and then determines which policy settings are effectively applied to users and computers. [...]</i>

Table 2.1: Group Policy Engine [1] [2]

In an Active Directory policy settings are stored as Group Policy objects, establishing how domain resources can be accessed, configured and used. The scope for which a policy associated with a GPO can be applied, is only within the domain and not across other domains. These GPOs, living in a domain, can be linked to Active Directory containers such as sites, domains or Organizational Units. Each GPO and its defined settings is only effective when it is linked to one or more of those containers. Group Policy settings can be designed as specialized or as general according to the respective organization's environment. [1] GPO links influence users and computers in different ways depending to which container it is linked:

**Linked to a site:**

- GPO applies to all users and computers in the site

**Linked to a domain:**

- GPO applies directly to all users and computers in the domain
- inherently to all participants of its child OUs
- (not inherited across domains)

**Linked to an OU:**

- GPO applies directly to all users and computers in the OU
- inherently to all participants of its child OUs

### 2.2.1 Filtering and Scope of GPOs

In order to use the whole benefit of GPOs in an AD DS, the GPOs shall be applied properly to AD containers. By applying GPOs to AD containers, called "scoping the GPO", the determination which users and computers will receive the setting takes places. There are three different ways to set the scope of each GPO in more and more detail. First, links to a site, domain, OU are used to define which computers and users will receive the particular GPO. Secondly, security filtering is used to further reduce and specify the participants which will receive the GPO even more. In addition, WMI filtering can be used to be even more precise. [1] [2]

### 2.2.2 Inheritance and Processing Rules of GPOs

Inheritance of GPOs is a possible way to distribute the GPO settings in an organization. GPOs linked to higher containers in an AD environment are inherited by their respective child containers and then combined. Basically, when multiple GPOs apply to a computer or user, the settings defined are aggregated if possible. It might lead to conflicts between individual GPO settings and therefore a resolution has to take place as to which setting should be applied. In such cases, the setting with the higher precedence wins. A GPO has a higher precedence if the GPO is processed later, this is referred to the “later writer wins” model, resulting in an override of the earlier applied setting. [1] [2]

GPO settings are processed in the following hierarchical order - the later the higher the precedence:

#### Local GPO:

- Exactly one GPO exists for each computer, which is stored locally

#### Site:

- The GPO with the lowest link order in sites (an attribute set in Group Policy Management Console) is processed last and thus has the highest precedence

#### Domain:

- The GPO with the lowest link order in domains (an attribute set in Group Policy Management Console) is processed last and thus has the highest precedence

#### Organizational Units:

- GPOs linked to OUs higher in an AD hierarchy are processed first
- Then GPOs linked to OUs lower in the AD hierarchy will be processed level for level
- Finally, the GPOs that are linked to the Organizational Unit which contains the user or computer are processed

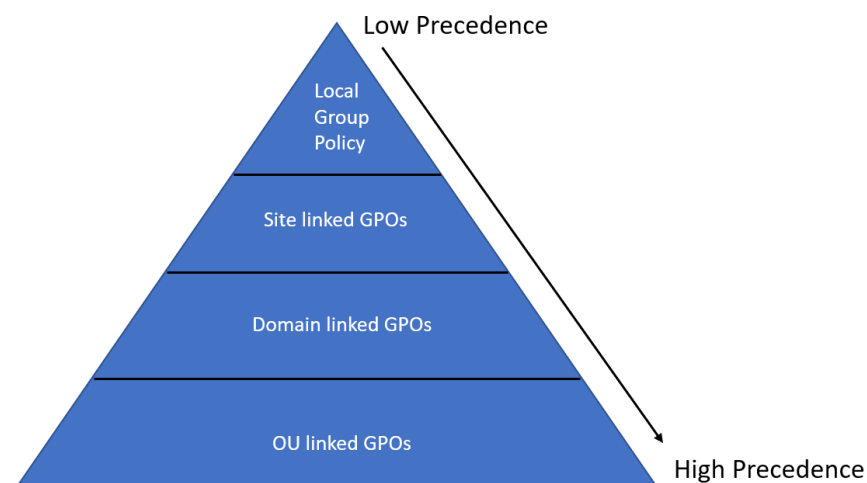


Figure 2.4: GPO Precedence

**Note:** There are exceptions which will overrule the precedence of the hierarchical order of GPOs. These exceptions must therefore also be taken into account:

**Enforce option:**

- Any domain-based GPO can be enforced by setting the corresponding option
- If this option is enabled, this GPO cannot be overwritten
- Enforced GPOs are always processed last, hence enforced GPOs overwrite all other GPO settings
- Multiple enforced GPOs will be processed by the link order precedence (see 2.4 GPO Precedence)

**Block Inheritance:**

- Any GPO inheritance, on domain or OU level, can be selectively designated as Block Inheritance
- Stops containers inheriting policies from parent containers
- Block Inheritance does not prevent enforced policies from applying

**Loopback:**

- GPOs apply based on the location of the computer rather than the user object
- Allows to apply “User Group Policy” based on computer which the user is logging onto
- Different Modes: Merge and Replace

**Example:** For a better understanding of which GPO applies, we provide the example from Microsoft’s article “What is a Core Group Policy” [1]:

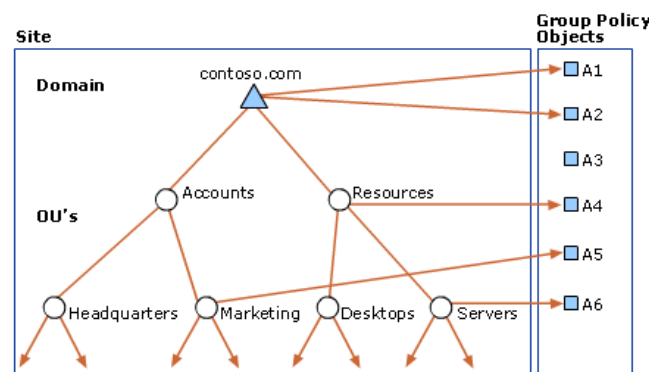


Figure 2.5: Example which GPO will apply

GPOs applied to Servers OU:

- A1, A2, A3<sup>1</sup>, A4, A6

GPOs applied to Marketing OU:

- A1, A2, A3<sup>1</sup>, A5

Loopback Merge Mode (Marketing OU user logs into Server OU computer):

- A3, A1, A2, A5, A3<sup>1</sup>, A1, A2, A4, A6

Loopback Replace Mode (Marketing OU user logs into Server OU computer):

- A3<sup>1</sup>, A1, A2, A4, A6

<sup>1</sup>Site applied GPO

### 2.2.3 Resultant Set of Policies

One of the most interesting features for administrating Group Policies is the Resultant Set of Policies (RSoP) - especially for determining applied Group Policy settings. RSoP comes in two different modes: logging mode and planning. The logging mode determines the resultant effect of Group Policy settings which have been applied to an existing user or computer based on its domain container (site, domain, OU). This mode is available on every Windows operating system controlled by Winlogon and is part of the normal GPO processing operation. The planning mode provides the possibilities to model and simulate resultant effects of Group Policy settings which will be applied to users and computers. The planning mode is only available on Windows Servers acting as domain controllers. All gathered information about Group Policy processing by computers is stored in WMI databases, also known as the CIMOM (Common Information Management Object Model) database. [2]

### 2.2.4 GPOs Storage Location

GPOs are virtual objects in the domain which stores its data in two locations: Group Policy container and Group Policy template. A Group Policy container is a Active Directory location where GPOs and their properties are stored. Properties describe computer and user Group Policy informations. The Group Policy container can be accessed through the Lightweight Directory Access Protocol (LDAP) syntax and each container lies at the path **CN=Policies,CN=System,DC=Domain\_Name** whereby **Domain\_Name** is the fully qualified domain name (FQDN). These containers can also be found in the "Active Directory Users and Computers" snap-in under **System\Policies**. Every Group Policy container has its unique name - a Globally Unique Identifier (GUID). There are two GPOs which always have the same GUID in every AD:

- **Default Domain Policy:** {31B2F340-016D-11D2-945F-00C04FB984F9}
- **Default Domain Controllers Policy:** {6AC1786C-016F-11D2-945F-00C04FB984F9}

The Group Policy container of each GPO contains attributes that are used to deploy GPOs as well as a link to the file system component of each GPO - the Group Policy template. [2]

The Group Policy templates contain the majority of the Group Policy settings and are stored in the Sysvol of the domain controller in the **\\domain\_name\Sysvol\domain\_name\Policies** folder. These templates provide the actual data for the policy extensions like Security Settings (.inf files), Administrative Templates based policy settings (.adm and .pol files) or potentially scripts. Moreover, this folder includes the relevant settings for the "Advanced Audit Policy Configuration" (can be found at **\MACHINE\Microsoft\Windows NT\Audit**). A section from the Microsoft documentation shows the following additional contents which influences the computer policies:

**Scripts/Startup** Contains the scripts that are to run when the computer starts up.

**Scripts/Shutdown** Contains the scripts that are to run when the computer shuts down.

**Applications** Contains the advertisement files (.aas files) used by the Windows installer. These are applied to computers.

**Microsoft/Windows NT/Secedit** Contains the Gpntmpl.inf file, which includes the default security configuration settings for a Windows Server 2003 domain controller.

**Adm** Contains all of the .adm files for the GPO.



### 3 Benchmark

A benchmark is carried out in this section. There are various guidelines regarding Lateral Movement Detection and Windows Event Logging from different Cyber Security Centers. These policies are compared and clarified as to how the Proof of Concept (PoC [3]) covers different policies.

There are almost as many policies and papers as there are cyber security centers. For this benchmark, guidelines from some of the largest and most important institutions have been used. The Japan Computer Emergency Response Team / Coordination Centers (JPCERT/CC) paper “Detecting Lateral Movement through Tracking Event Logs” [4] and Microsofts “Events to Monitor” [?] are deliberately not listed because the Proof of Concept is based on these guidelines.

Each section discusses the differences between the group policy setting of the individual papers and the PoC which was based on the JPCERT/CC paper “Detecting Lateral Movement through Tracking Event Logs” [4].

#### 3.1 Computer Emergency Response Team - Europe

The Computer Emergency Response Team - Europe (CERT-EU) published their paper “Detecting Lateral Movements in Windows Infrastructure”[5] in the year 2017. The paper provides guidelines to detect lateral movement in a Windows Vista/7 and Windows Server 2008 based environment. A plan to bring out guidelines for Windows 10 has unfortunately not happened yet. [5, p. 1]

The first chapter gives a background on “Lateral Movement Attacks” in general. It is described as a 2-step attack; step one is to capture credentials from a source host. Step two is to use these stolen credentials - preferably from an administrator - to access another host or resource. Attackers can use techniques known as pass-the-hash, pass-the-ticket or Kerberos tickets. Lateral movement is not only limited to access other hosts but also servers like an Exchange server etc. This picture shows a “typical” lateral movement attack [5, p. 2-3]:

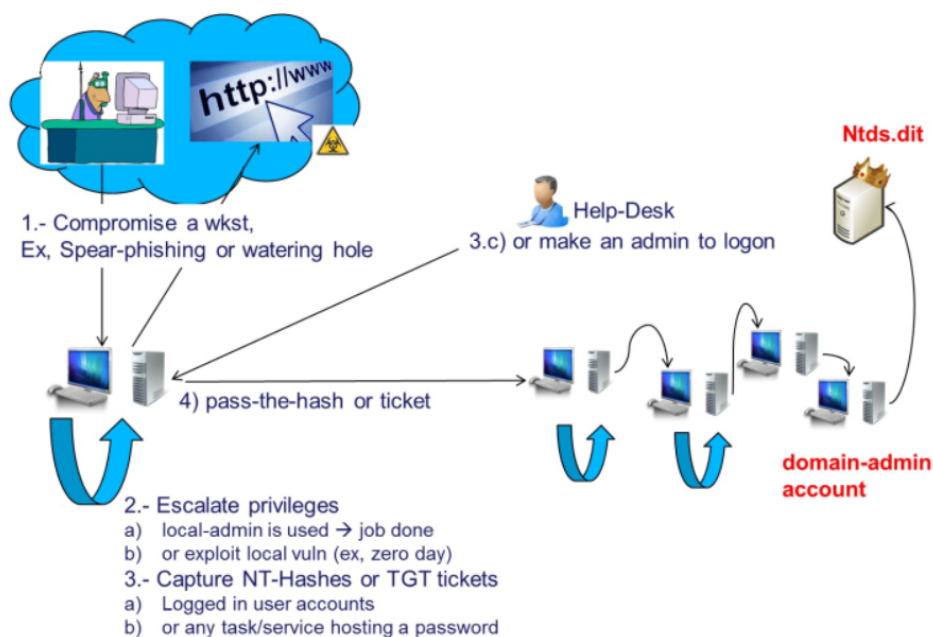


Figure 2.6: Typical lateral movement attack [5, p. 4]

The paper explains that credentials are cached in workstations where the users are connected to or run commands from, for instance when using the Remote Desktop Protocol (RDP) or “runas” commands. Attackers can use tools to capture these credentials. CERT-EU tested different logon types to see which type saved their credentials in the memory of the workstations. [5, p. 4-5] These are the results:

Logon Type	Cached NT Hash <sup>2</sup>	Cached TGT <sup>3</sup>
remote assistance	No	No
RDP	Yes	Yes
run as admin with interactive	Yes	Yes
run as admin with interactive + network	Yes	Yes
network to a remote (admin) share	No	No

Table 2.2: Caching credentials [5, p. 4-5]

Thanks to logging and monitoring Windows events is it possible to detect lateral movement. The CERT-EU guidelines are focused on how to detect “pass-the-hash” and “pass-the-ticket” attacks. They recommend to log these events on the domain controller [5, p. 3-10]:

EventID	Description
4776	The domain controller attempted to validate the credentials for an account ( <i>Key-Element to monitor</i> )
4768	A Kerberos authentication ticket (TGT) <sup>3</sup> was requested ( <i>May not occur because tickets was previously stolen</i> )
4769	A Kerberos service ticket was requested ( <i>request access to the target system or resource, Key-Element to monitor</i> )

Table 2.3: Recommended Event Loggings on the domain controller [5, p. 3-10]

<sup>2</sup>Authentication through NT LAN Manager (NTLM)

<sup>3</sup>Authentication through “Ticket Granting Ticket” (TGT) by Kerberos

And to log these events on all important\* accounts [5, p. 3-10]:

EventID	Description
4648	A logon was attempted using explicit credentials ( <i>Useful for forensics purposes</i> )
4624	An account was successfully logged on ( <i>Useful for forensics purposes</i> )
4625	An account failed to log on ( <i>Useful for forensics purposes</i> )
4634	An account was logged off ( <i>Useful for forensics purposes</i> )
4647	User initiated logoff ( <i>Useful for forensics purposes</i> )

Table 2.4: Recommended Event Loggings on Workstations [5, p. 3-10]

\* *service accounts, rarely used accounts, emergency accounts, business-critical accounts, etc.*

In the subsection “Detecting the Golden Ticket” the reference is made to their published paper about “Kerberos Golden Ticket”. The golden tickets allows the attacker to authenticate anybody in the domain. But to produce such a ticket, the attacker needs full administrator rights and control over the Domain Controller. To achieve this goal the attacker most likely needs to do lateral movement. Therefore, the same events have to be monitored. [5, p. 10]

In the “Additional Monitoring” section, they recommend the National Security Agency reference, which provides additional events that might be useful in order to monitor to detect possible attacks.

### 3.1.1 Comparision PoC - CERT-EU

The paper of the CERT-EU focuses mainly on logging the “pass-the-hash” and “pass-the-ticket” attacks. JPCERT/CC’s paper [4], on which the PoC is based, covers much more than just these two attacks and logs more events. The events proposed by CERT-EU are therefore also covered by the PoC. The PoC covers also the mentioned “Golden Ticket” and goes even further to detect so called “Silver Tickets”. [5]

The CERT-EU refers to the NSA paper [6] when it comes to logging other events, these recommendations are discussed in the next section. [5, p. 10]

### 3.1.2 Conclusion CERT-EU

“Detecting Lateral Movements in Windows Infrastructure”[5], which CERT-EU published, is a very slim paper based heavily on NSA recommendations. The PoC [3] covers all recommendations of the CERT-EU. However, the weighting of this paper is also very low.

### 3.2 National Security Agency

The National Security Agency (NSA) does not provide a specific paper about detecting lateral movement in a network. In the year 2013 “Spotting the Adversary with Windows Event Log Monitoring” [6] was published, this guide is a general recommendation on what events to monitor in order to detect attackers as well as other useful settings to secure the network.

The first part of the report describes how to set up Windows event forwarding; it is recommended to install a server only to collect and analyze event logs. To ensure the integration of event logs it is recommended to reduce the administrator groups privileges from “Full” to “Read & Execute” permissions. [6, p. 1-7] It is also recommended to disable “Windows RemoteShell” and to restrict the “Windows RemoteAccess” to certain internet protocol (IP) addresses and/or ranges.[6, p. 15-20] Suggestions are also made for hardening the event collection by changing the “Windows Remote Management” authentication method and/or by encrypting the payload. [6, p. 20-24]

The most relevant part of the paper for the Readinizer is the “Recommended Events to Collect” which is summarized in these categories by the NSA:

**Application Whitelisting** These events should be collected in order to search for applications that were blocked from execution. Blocked applications could either be malware or unapproved software, which may be investigated. [6, p. 24]

- **Consideration necessary:** No, auditing Application Whitelisting is not a part of the PoC and will also not be part of the Readinizer. But not because it is negligible, Application Whitelisting can be a very useful and powerful tool to prevent Lateral Movement and APTs. Unfortunately, Application Whitelisting is very elaborate and the PoC as well as the Readinizer will be used by small and medium-sized enterprises.

**Application Crashes** These events should be collected because they can be an indication that malware is in use. They may justify an investigation. In particular, Windows Error Reporting and Blue Screen of Death. [6, p. 24]

- **Consideration necessary:** No, application crashes occur in the working day and would mostly only be producing noise. Most of these events are logged by default anyway.

**System or Service Failures** These events should be collected because they can be interesting. Services do not normally fail or crash. They may justify an investigation by an administrator. [6, p. 24]

- **Consideration necessary:** No, these events are logged by default. No further consideration necessary.

**Windows Update Errors** These events should be collected to find out if there are machines which have failed to install an update. Machines should be kept up to date to eliminate known vulnerabilities. [6, p. 25]

- **Consideration necessary:** No, these events are neither logged in the PoC nor the Readinizer. Most enterprises use other tools to roll out software and updates.

**Windows Firewall** These events should be collected to detect changes on the built-in host-based Windows Firewall. Normal users should not make any changes to the local firewall rules. [6, p. 25]

- **Consideration necessary:** Yes, the PoC does not check these events, but they should be logged. Normal users should not change firewall settings. This may be a strong indication of malicious attacks.

**Clearing Event Logs** These events should be collected to detect when an event log was cleared. This is a strong indication that an attacker wants to cover his tracks. [6, p. 25]

- **Consideration necessary:** Yes, the PoC as well as the Readinizer do check these event logs. If one of these events occurs, that's a strong indication of malicious attacks.

**Software & Service Installation** These events should be logged because as a part of normal network operations, new software and services will be installed. Through logging these events administrators can check and verify that these are no risk to the network. [6, p. 26]

- **Consideration necessary:** Yes, some of these events can be very useful, especially from a forensic point of view. Some of them are collected in the PoC as well as in the Readinizer.

**Account Usage** These events should be collected to detect Pass the Hash, Pass the Ticket and other unauthorized account usage. Many other useful information regarding remote desktop login, users added to privileged groups or account lockouts can be tracked using these events. [6, p. 26-27]

- **Consideration necessary:** Yes, these events are collected in the PoC as well as with the Readinizer. These logs can be crucial in understanding and tracking attacks.

**Kernel Driver Signing** These events should be collected because any alert thrown may indicate malicious activity. [6, p. 27-28]

- **Consideration necessary:** No, with the kernel driver signing in the 64-bit version of Windows Vista Microsoft improved the defense against insertion of malicious drivers or activities in the kernel massively. These events are neither logged in the PoC nor the Readinizer.

**Group Policy Errors** These events should be collected because manage domain computer with group policies allows administrators to improve the security and regulation. These events may be investigated by an administrator. [6, p. 28]

- **Consideration necessary:** No, these events are neither logged in the PoC nor the Readinizer. These events occur rarely and can be neglected.

**Windows Defender Activities** These events should be logged because any notification of detecting, removing and preventing malicious programs is worthy to investigate. If the Windows Defender is not functioning properly, this issue should be corrected to prevent infection or further infection of the system. [6, p. 28-29]

- **Consideration necessary:** Yes, these events were not considered in the PoC but are in the Readinizer. But if a third-party antivirus or antispysware product is used, these logs can be neglected. [6, p. 29-30]

**Mobile Device Activities** These events should be collected because they may be critical for an enterprise to log. An infected wireless device can travel between different networks, regardless of the communication protocol. Tracking which mobile device enter and exit the network can be helpful. [6, p. 30-31]

- **Consideration necessary:** No, these events are neither logged in the PoC nor the Readinizer, they produce too much noise.

**External Media Detection** These events should be collected to detect USB devices inserted into the network. [6, p. 31]

- **Consideration necessary:** No, these events are neither logged in the PoC nor the Readinizer, they produce too much noise.

**Printing Services** These events should be collected to be able to trace who printed what, when and where. [6, p. 32]

- **Consideration necessary:** No, these events are neither logged in the PoC nor the Readinizer, they produce too much noise.

**Pass the Hash Detection** These events should be collected to be able to detect lateral movement in a system. [6, p. 32-33]

- **Consideration necessary:** Yes, these events are collected in the PoC as well as with the Readinizer. They overlap with the events from Account Usage.

**Remote Desktop Logon Detection** These events should be collected because Remote Desktop should only be used by certain administrators. [6, p. 33-34]

- **Consideration necessary:** Yes, these events are collected in the PoC as well as with the Readinizer. They overlap with the events from Account Usage.

The NSA recommends to set the size of the log file server to 1 Gigabyte (GB) and to enable the “Archive the log when full, do not overwrite events.” The restriction helps the security advisor to get a better overview of the events that have occurred. However, this makes it all the more important to check the logs regularly (once a day). [6, p. 34-35]

### 3.2.1 Comparison PoC - NSA

In this section we compare the recommendations of the NSA paper “Spotting the Adversary with Windows Event Log Monitor” [6] and the ones that were made in the PoC [3]. In 3.2 National Security Agency a list was made and decided whether the log recommendation needs further consideration. The events in which this question was answered in the affirmative are once again examined in detail here.

**Windows Firewall** Changes on the built-in host-based Windows Firewall can be an indicator for malicious attacks. Normal users should not make any changes to the local firewall rules.

Event Log	EventID	Event	NSA	PoC	Readinizer
Microsoft-Windows/ Windows Firewall with advanced Security/ Firewall	2004	Firewall Rule Add	Yes	Yes	Yes
Microsoft-Windows/ Windows Firewall with advanced Security/ Firewall	2005	Firewall Rule Changed	Yes	Yes	Yes
Microsoft-Windows/ Windows Firewall with advanced Security/ Firewall	2006/2033	Firewall Failed to load Group Policy	Yes	Yes	Yes
Microsoft-Windows/ Windows Firewall with advanced Security/ Firewall	2009	Firewall Rule Add	Yes	Yes	Yes

Table 2.5: Windows Firewall [6, p. 25] [7]

The events that show the operational status are logged by default, there is no need to change the settings. [7]

**Clearing Event Logs** This event is triggered when an event log was cleared; this is a strong indication that an attacker wants to cover his tracks. [6, p. 25]

Event Log	EventID	Event	NSA	PoC	Readinizer
System	104	Event Log was cleared	Yes	Yes	Yes
Security	1102	Audit Log was cleared	Yes	Yes	Yes

Table 2.6: Clearing Event Logs [6, p. 25] [3, p. 17] [8]

Both of these events are recorded by default, there is no need to change the settings. [8] [4, p. 63]

**Software & Service Installation** As a part of normal network operations, new software and services will be installed. Through logging these events administrators can check and verify that these are of no risk to the network. [6, p. 26]

Event Log	EventID	Event	NSA	PoC	Readinizer
System	6	New Kernel Filter Driver	Yes	No	No
System	7045	New Windows Service	Yes	Yes	Yes
Application	1022/1033	New MSI File Installed	Yes	No	No
Application-Experience/ Program-Inventory	903	New Application Installation	Yes	No	No
Application-Experience/ Program-Inventory	905	Updated Application	Yes	No	No
Application-Experience/ Program-Inventory	907	Removed Application	Yes	No	No
Application-Experience/ Program-Inventory	800	Summary of Software Activities	Yes	No	No
Setup	2	Update Packages Installed	Yes	No	No
System	19	Windows Update Installed	Yes	No	No

Table 2.7: Software & Service Installation [6, p. 26] [3, p. 17]

The event “New Kernel Filter Driver”, “New MSI File Installed”, as well as the “Application Installation/Updated/Removed” and “Summary of Software Activities”, are not logged due to the noise these events generate. Furthermore, the significance is not to be estimated so highly. The event “11707 - Installation operation completed successfully” is more important and logged by default. The “New Windows Service” event is logged by default. [9] Although it is important that the machines are updated, these events are not considered important.



**Account Usage** Collecting account usage can help when detecting “pass-the-hash” activities and can be very useful in forensic terms.

Event Log	EventID	Event	NSA	PoC	Readinizer
Security	4740	Account Lockouts	Yes	Yes	Yes
Security	4728/4732/4756	User Added to Privileged Group	Yes	Yes	Yes
Security	4735	Security-Enabled group Modification	Yes	Yes	Yes
Security	4624	Successful User Account Login	Yes	Yes	Yes
Security	4625	Failed User Account Login	Yes	Yes*	Yes
Security	4648	Account Login with Explicit Credentials	Yes	Yes	Yes

Table 2.8: Account Usage [6, p. 26-27] [3, p. 18-19]

\* The event “4625 - Failed User Account Login” is logged in the PoC, but only by the “Audit Logon - Success and Failure ” but not by “Audit Account Lockout - Failure”

All the events are logged in the PoC and will be logged in the Readinizer. [3, p. 18-19]

**Windows Defender Activities** Every notification of detecting, removing and preventing malicious programs is worthy to investigate. If the Windows Defender is not functioning properly, this issue should be corrected to prevent infection or further infection of the system. [6, p. 28-29]

Event Log	EventID	Event	NSA	PoC	Readinizer
Microsoft-Windows-Defender/Operational	1005	Scan Failed	Yes	Yes	Yes
Microsoft-Windows-Defender/Operational	1006	Detected Maleware	Yes	Yes	Yes
Microsoft-Windows-Defender/Operational	1008	Action on Maleware Failed	Yes	Yes	Yes
Microsoft-Windows-Defender/Operational	1010	Failed to remove item from quarantine	Yes	Yes	Yes

Microsoft-Windows-Defender/Operational	2001	Failed to update signature	Yes	Yes	Yes
Microsoft-Windows-Defender/Operational	2003	Failed to update engine	Yes	Yes	Yes
Microsoft-Windows-Defender/Operational	2004	Reverting to last known good set of signatures	Yes	Yes	Yes
Microsoft-Windows-Defender/Operational	3002	Real-Time Protection failed	Yes	Yes	Yes
Microsoft-Windows-Defender/Operational	5008	Unexpected Error	Yes	Yes	Yes

Table 2.9: Windows Defender Activities [6, p. 28-29] [10]

All of these events are recorded by default, there is no need to change the settings [10].

**Pass the Hash Detection** The “pass-the-hash” logs match those from “Account Usage”. The PoC logs the recommended events. [6, p. 32-33] [3, p. 18-19]

**Remote Desktop Logon Detection** When an account remotely connects to a client, a successful logon event is created. Therefore the events with the id 4624 and 4634 are logged, which matches with the events from “Account Usage”. The PoC logs the recommended events. [6, p. 33-34] [3, p. 18-19]

### 3.2.2 Conclusion NSA

The NSA is a globally recognized cyber security agency and has published “Spotting the Adversary with Windows Event Log Monitor” [6], a very comprehensive and powerful paper. Although it covers a larger area than intended for the PoC or the Readinizer, the important parts of the PoC [3] mostly correspond very well with the paper. Any discrepancies that occurred were checked and adjusted for the settings in the Readinizer.

### 3.3 Australian Cyber Security Center

The Australian Cyber Security Center (ACSC) provides a paper called “Windows Event Logging and Forwarding” [11] with detailed recommendations of logging events in Windows environments, as well as accurate guidance for group policies settings in perspective to event logging.

In one of the first sections of the paper the ACSC states the recommendation for the event log retention time to be at least 18 months, due to the fact of appropriate traceability in case of attacks. However, this time should be mentioned as a minimum time because it may differ in respect of regulations and audit requirements.[11, p. 1]

ACSC recommends group policies, defined for auditing event logs, should always be defined in Group Policy Objects apart from other GPOs and with a scope set for every system in the Active Directory. [11, p. 2]

The ACSC mentions the Windows event log sizes which should be increased from its default value. Unfortunately, the ACSC does make no suggestions of precise sizes. They state the maximum sizes which are possible and refer this decision to be made individually with regard to the environment. [11, p. 2]

The next part of the paper addresses the different event categories which should or must be logged for good forensic baseline, as well as which configurations have to be done to achieve them. These categories describe a superset of GPO settings in Microsoft environments which are mostly adaptable to the analyzed settings of the previously done Proof of Concept. [3, p. 18-19] The following table shows a brief form of the category table from the ACSC and states which additional categories should be considered (row “Consideration necessary”- Yes/No) or need additional attention (Partially). The descriptions are quotations from the table it self. [11, p. 3-5] In general, the ACSC states almost all event categories as highly recommended (row “Value”). [11, p. 3] The row “Noise” describes how many events are thrown in an average manner and thus how disk space consuming they are.

Event Category	Description	Value	Noise	Integration in the Readinizer
Sysmon	<i>Provides visibility of process creation and termination, driver and library loads, network connections, file creation, registry changes, process injection, and more.</i>	Very High	Very High	Yes but benefits already discussed in the PoC [11, p. 11] - development of a manual for the fleet-wide installation of sysmon
Account lock-out	<i>Records account lockout activity - detects password brute-forcing attempts.</i>	High	Low	Yes
Account modifications	<i>Records unauthorized creation or modification of accounts and groups with administrative privileges.</i>	High	Low	Partially (additional policy settings)
Event collection	<i>Forwards changes and errors with auditing, event collection and event forwarding. Detects attempts by an adversary to suppress logging evidence.</i>	High	Low	Yes

Account logon	<i>Records activity related to accounts logging in and out - detects unauthorized use of accounts.</i>	High	Medium	Partially (configured policy settings)
Process tracking	<i>Provides visibility of (malicious) process creation and termination, including command line arguments.</i>	High	High	Partially (additional policy settings)
Services	<i>Detects installation of services that are used for persistence or lateral movement by an adversary.</i>	Medium	Low	No - default logged by Windows [11, p. 9] [9]
Windows Error Reporting	<i>Detects exploitation attempts and unstable applications, which may indicate malicious activity.</i>	Medium	Low	No - default logged by Windows [11, p. 9] [12]
Code Integrity	<i>Records code integrity violations for drivers and protected processes. Detects malware or restricted applications that are being audited or prevented from executing by code integrity checks.</i>	Medium	Low or Medium	Yes
File shares	<i>Detects access and modification of file shares. This includes lateral movement and access to file shares used to exfiltrate data from the network.</i>	Medium	Medium	Partially (configured policy settings)
Scheduled tasks	<i>Detects scheduled tasks being added or modified. This may include tasks used for lateral movement, persistence or elevation to system privileges.</i>	Medium	Medium	No (already considered in PoC [3, p. 23])
Object access auditing	<i>Detects some forms of unauthorized changes to sensitive files and registry keys, and some forms of credential and password hash access.</i>	Low	Medium	No (already considered in PoC [3, p. 19])

Table 2.10: ACSC event categories [11, p. 3-5]

In addition to the event categories, the ACSC states Sysmon, as already recognized in the PoC [3, p. 11], as a very useful additional tool and important for a solid detection of attacks. Since the benefits have already been discussed in the PoC, ACSC's paper only provides little information on system-wide installation. [11, p. 5-6]

At last, the ACSC recommends the event logs to be forwarded to a central server, implemented with the native ability of Microsofts Windows Event Forwarding (WEF). [11, p. 13-17]

### 3.3.1 Comparison PoC - ACSC

Settings already contained in the PoC are no longer taken into account, unless the set value of the setting differs.

**Account lockout** This group policy records account lockout activities. For a good forensic base to detect brute-force attacks of account logins, this policy should be considered and activated. However, the ACSC has not correctly observed the settings and recommends “Success”, but only “Failure” events are thrown with this setting, so the setting “Failure” is implemented in the Readinizer. [13] In addition, Microsoft’s default setting is wrong, because it is also set to “Success”, whereby only “Failure” events are thrown. This has been reported to Microsoft.

Advanced Audit Policy Configuration\Logon/Logoff			
Setting	ACSC	PoC	Readinizer
Audit Account Lockout	Success	Not included	Failure

Table 2.11: ACSC vs. PoC - Account lockout

**Account modifications** This group policy records account modifications whether they are authorized or not. [14] [15] The setting “Audit User Account Management” should log “Success and Failures” to track not just successful events like “An attempt was made to change an account’s password” but also failures. [16] The other settings throw just “Success” logs and will therefore not be set to “Success and Failure”, as recommended by the ACSC.

Advanced Audit Policy Configuration\Account Management			
Setting	ACSC	PoC	Readinizer
Audit Computer Account Management	Success and Failure	Not included	Success
Audit Other Account Management Events	Success and Failure	Not included	Success
Audit User Account Management	Success and Failure	Success	Success and Failure

Table 2.12: ACSC vs. PoC - Account modifications

**Account logon** This group policy records logon and logoff events. With the setting “Audit Other Logon/Logoff Events” terminal session login/logoff events can be tracked in addition to the regular login/logoff events. [17] Furthermore, this setting allows the detection of a Kerberos replay attacks. The setting “Audit Group Membership” allows to audit the group membership information in the user’s logon token which might be useful in case of a detection. [18] The setting “Audit Special Logon” only needs the to be set to “Success” because this setting has just “Success’-events and no “Failures”. [19]

Advanced Audit Policy Configuration\Logon/Logoff			
Setting	ACSC	PoC	Readinizer
Audit Other Logon/Logoff Events	Success and Failure	Not included	Success and Failure
Audit Group Membership	Success	Not included	Success
Audit Special Logon	Success and Failure	Success	Success

Table 2.13: ACSC vs. PoC - Account logon

**Event collection** This group policy records changes within the event auditing as well as modifications on user rights and domain relationships. These events can be very helpful in order to detect what happened during an attack or a lateral movement and therefore shall be logged. [20] The recommendation "Success and Failure" from the ACSC is unnecessary, because only "Success" is logged.

Advanced Audit Policy Configuration\Policy Change			
Setting	ACSC	PoC	Readinizer
Audit Audit Policy Change	Success and Failure	Not included	Success

Table 2.14: ACSC vs. PoC - Event collection

**Process tracking** The PoC focused just on tracking process creation and termination events. But it is important to increase the value of the process creation events by including command line arguments with process creation events.[21]

System\Audit Process Creation			
Setting	ACSC	PoC	Readinizer
Include command line in process creation events	Enabled	Not included	Enabled

Table 2.15: ACSC vs. PoC - Process tracking

**Code integrity** These logs can detect failures which occur within the event logging. Moreover, it logs processes which use an invalid local procedure call (LPC) port in an attempt to impersonate a client, reply to a client address space, read to a client address space, or write from a client address space. [22] Hence, these events can evolve the use of logs during attack detection.

Advanced Audit Policy Configuration\System			
Setting	ACSC	PoC	Readinizer
Audit System Integrity	Success and Failure	Not included	Success and Failure

Table 2.16: ACSC vs. PoC - Code integrity

**File shares** The "Audit Detailed File Share" policy is no longer recommended due to high noise level. [23] Audit events related to file shares like creation, deletion, modification, and access attempts are still logged with the policy "Audit File Share". [24] In combination with the policy "Audit File System" it is possible to track what content was accessed, the source (IP address and port) of the request, and the user account that was used for the access. [25]

Advanced Audit Policy Configuration\Object Access			
Setting	ACSC	PoC	Readinizer
Audit Detailed File Share	Not Configured <sup>4</sup>	Success and Failure	Not Configured <sup>4</sup>

Table 2.17: ACSC vs. PoC - File shares

<sup>4</sup>Policy setting "Not Configured"  $\Rightarrow$  there is no default value that this policy would log

### 3.3.2 Conclusion ACSC

The paper “Windows Event Logging and Forwarding” [11] from the Australian Cyber Security Centre provides a very good insight of solid event logging and highlights which group policies must be set. Although the Proof of Concept [3] has already covered many of the settings recommended by the ACSC, some adjustments to the final group policy settings have been made. It shows that the Proof of Concept has been made in the right direction and the Readinizer can be built on top of this path. Surprisingly, even an error could be found in the default event log policy “Audit Account Lockout” setting of Microsoft (see Account lockout).

### 3.4 MITRE Adversarial Tactics, Techniques and Common Knowledge

The MITRE Adversarial Tactics, Techniques and Common Knowledge (MITRE ATT&CK) platform is a globally accessible knowledge base of adversary tactics and techniques based on real-world observations by many key players in the security scene. The platform is filled with recommendations how an effective hardening of computer systems or networks can be applied. However, it is difficult to find suggestions for prevention in the form of clean event logging because the platform focuses more on mitigating specific attacks. Although there is also a detection section for each attack, it does not always show exactly how the attack can be detected, but is more general in form. It is difficult to conclude on group policy settings which have the effect that the attack can be traced in the event log.

Since over 220 types and techniques of windows enterprise attacks, malicious code execution, exfiltration of data, etc. are described on the platform, the scope has been limited to the following subsets of attacks:

- Execution
- Privilege Escalation
- Discovery
- Persistence
- Credential Access
- Lateral Movement

Since not every description of an attack leads to a conclusion of what should be logged, only attacks are described which have an effect on auditing events.

#### 3.4.1 Comparision PoC - MITRE ATT&CK

This section examines each attack technique which has an effect on event auditing and is yet not considered within the PoC. For each attack technique a brief description quotation from MITRE ATT&CK [26] is provided as well as how this attack can be detected. Finally, a statement is made as how to it will influence the Readinizer.

**LSASS Driver (Execution, Persistence)** *The Windows security subsystem is a set of components that manage and enforce the security policy for a computer or domain. The Local Security Authority (LSA) is the main component responsible for local security policy and user authentication. [...] Adversaries may target lsass.exe drivers to obtain execution and/or persistence. [27]*

- **Detection:** *With LSA Protection enabled<sup>5</sup>, monitor the event logs (Events 3033 and 3063) for failed attempts to load LSA plug-ins and drivers. [27]*
- **Consideration:** Enable LSA Protection in Registry [29]

**PowerShell (Execution)** *PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. [...] Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. [30]*

- **Detection:** *[...] turn on PowerShell logging to gain increased fidelity in what occurs during execution. [30] [31]*
- **Consideration:** Enable “Turn on Module Logging” and “Turn on PowerShell script Block Logging” in Administrative Templates \Windows Components \Windows PowerShell [32]

<sup>5</sup>To enable LSA Protection, Secure Boot must be enabled → follow the instructions on [28] and do the opposite to enable Secure Boot



**Scheduled Task (Execution, Persistence, Privilege Escalation)** *Utilities such as `at` and `schtasks`, along with the Windows Task Scheduler, can be used to schedule programs or scripts to be executed at a date and time. [...] An adversary may use task scheduling to execute programs at system startup or on a scheduled basis for persistence, to conduct remote Execution as part of Lateral Movement, to gain SYSTEM privileges, or to run a process under the context of a specified account. [33]*

- **Detection:** *Configure event logging for scheduled task creation and changes by enabling the “Microsoft-Windows-TaskScheduler/ Operational” setting within the event logging service. [33]*
- **Consideration:** Use Windows Event Log Tools Utility (`wevutil`) [34] to enable “Microsoft-Windows-TaskScheduler/ Operational”

**Process Injection (Privilege Escalation)** *Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process’s memory, system/network resources, and possibly elevated privileges. [35]*

- **Detection:** *Monitor processes and command-line arguments. Code injection may also be performed using PowerShell so additional PowerShell monitoring might be required to gather all needed information. [35]*
- **Consideration:** Partially - Process execution (Audit Process Creation and Termination) is already considered in the PoC [3, p. 18] but not command-line arguments (Include command line in process creation events). Enable additional PowerShell monitoring - see PowerShell (Execution)

**Credential Dumping (Credential Access)** *Credential dumping is the process of obtaining account login and password information, normally in the form of a hash or a clear text password, from the operating system and software. Credentials can then be used by adversaries to perform Lateral Movement and access restricted information. [...] Mostly performed with custom tools like Mimikatz. [36]*

- **Detection:** [...] *monitor Windows Logs for LSASS.exe creation to verify that LSASS started as a protected process, therefore LSA Protection must be enabled. [...] Monitor processes and command-line arguments for program execution that may be indicative of credential dumping. [36]*
- **Consideration:** Enable LSA Protection in Registry [29] as well as “Include command line in process creation events” [21]

**Subject Interface Packages and Trust Provider Hijacking (Persistence)** *In user mode, Windows Authenticode digital signatures are used to verify a file's origin and integrity, variables that may be used to establish trust in signed code. [...] Because of the varying executable file types and corresponding signature formats, Microsoft created software components called Subject Interface Packages (SIPs) to provide a layer of abstraction between API functions and files. SIPs are responsible for enabling API functions to create, retrieve, calculate, and verify signatures. [...] Adversaries may hijack SIP and trust provider components to mislead operating system and whitelisting tools to classify malicious (or any) code [37]*

- **Detection:** *Enable CryptoAPI v2 (CAPI) event logging to monitor and analyze error events related to failed trust validation as well as any other provided information events. Code integrity event logging may also provide valuable indicators of malicious SIP or trust provider loads [37]*
- **Consideration:** CAPI2 is already considered in the PoC [3, p. 13] but not code integrity → enable “Audit System Integrity” (see 3.3.1 Code integrity)

**New Service (Persistence, Privilege Escalation)** *When operating systems boot up, they can start programs or applications called services that perform background system functions. [...] Adversaries may install a new service that can be configured to execute at startup by using utilities to interact with services or by directly modifying the Registry. [38]*

- **Detection:** *Monitor service creation through changes in the Registry and common utilities using command-line invocation. Creation of new services may generate an alterable event (ex: Event ID 4697 and/or 7045). [...] Monitor processes and command-line arguments for actions that could create services. [38]*
- **Consideration:** Enable the setting “Audit Security System Extension” in the advanced audit policy configurations under “System” [39]. Enable command-line arguments (Include command line in process creation events) [21]

**Valid Accounts (Privilege Escalation)** *Adversaries may steal valid credentials of a specific user or service account using Credential Access techniques. [40]*

**Create Account , Account Manipulation (Persistence, Privilege Escalation)** *Account manipulation may aid adversaries in maintaining access to credentials and certain permission levels within an environment. [...] [41] Adversaries with sufficient access may create a local system or domain account. [42]*

**Brute Force (Credential Access)** *Adversaries may use brute force techniques to attempt access to accounts when passwords are unknown or when password hashes are obtained. [43]*

**(External) Remote Services (Persistence, Lateral Movement)** *Adversaries may use remote services with a valid user account to access and persist within a network. [44]*

**Security Identifier (SID) History Injection (Privilege Escalation)** *An account can hold additional SIDs in the SID-History Active Directory attribute, allowing inter-operable account migration between domains [...] Adversaries may use this mechanism for privilege escalation. [45]*

**Windows Admin Shares (Lateral Movement)** *Windows systems have hidden network shares that are accessible only to administrators and provide the ability for remote file copy and other administrative functions. [...] Adversaries may use this technique in conjunction with administrator-level Valid Accounts to remotely access a networked system over server message block (SMB) to interact with systems using remote procedure calls (RPCs), transfer files, and run transferred binaries through remote Execution. [46]*

- **Detection:** Monitor authentication logs for system and application login failures as well as account management operations
- **Consideration:** These logs are already taken into account through the PoC [3, p. 18-19] within the settings “Account Management” and “Logon/Logoff”

**Pass the Hash (Lateral Movement)** *Pass the hash (PtH) is a method of authenticating as a user without having access to the user’s cleartext password. This method bypasses standard authentication steps that require a cleartext password, moving directly into the portion of the authentication that uses the password hash. [47]*

- **Detection:** Audit all logon and credential use events and review for discrepancies. [47]
- **Consideration:** These logs are already taken into account through the PoC [3, p. 18-19] within the settings “Audit Account Management” and “Audit Logon/Logoff”

**Kerberoasting and Pass the Ticket (Credential Access, Lateral Movement)** *Adversaries possessing a valid Kerberos ticket-granting ticket may request one or more Kerberos ticket-granting service (TGS) service tickets from a domain controller. [48] Pass the ticket is a method of authenticating to a system using Kerberos tickets without having access to an account’s password. Kerberos authentication can be used as the first step to lateral movement to a remote system. [49]*

- **Detection:** Enable Audit Kerberos Service Ticket Operations to log Kerberos TGS service ticket requests. [48]
- **Consideration:** These settings (“Audit Kerberos Authentication Service” and “Audit Kerberos Service Ticket Operations”) were already included in the PoC [3, p. 18]

**Remote File Copy (Lateral Movement)** *Files may be copied from one system to another to stage adversary tools or other files over the course of an operation. Files may be copied from an external adversary-controlled system through the Command and Control channel to bring tools into the victim network or through alternate protocols with another tool such as File Transfer Protocol (FTP). [50]*

- **Detection:** Monitor for file creation and files transferred within a network [...] [50]
- **Consideration:** These logs are already taken into account through the PoC [3, p. 19] within several settings of “Object access” in the advanced audit policy configuration

**Process execution** All the following attack techniques were examined, and it became clear that they could be monitored via the detailed tracking in the advanced audit policy configuration. This setting (Audit Process Creation and Termination) has already been taken into account in the PoC [3, p. 18]. There is only one additional setting which shall be included in the Readinizer: command-line arguments (Include command line in process creation events) [21]

- |                                                      |                                                           |
|------------------------------------------------------|-----------------------------------------------------------|
| • Permission Groups Discovery (Execution)            | • XSL Script Processing (Execution)                       |
| • Peripheral Device Discovery (Execution)            | • Command-Line Interfaces (Execution)                     |
| • Password Policy Discovery (Execution)              | • Windows Management Instrumentation (Execution)          |
| • Network Share Discovery (Execution)                | • User Execution (Execution)                              |
| • Network Service Scanning (Execution)               | • XSL Script Processing (Execution)                       |
| • File and Directory Discovery (Execution)           | • Windows Remote Management (Execution, Lateral Movement) |
| • Browser Bookmark Discovery (Execution)             | • Application Shimming (Persistence)                      |
| • Permission Groups Discovery (Execution)            | • Screensaver (Persistence)                               |
| • Process Discovery (Execution)                      | • Service Registry Permissions Weakness (Persistence)     |
| • Remote System Discovery (Execution)                | • Access Token Manipulation (Privilege Escalation)        |
| • Security Software Discovery (Execution)            | • Account Discovery (Discovery)                           |
| • System Information Discovery (Execution)           | • Application Window Discovery (Discovery)                |
| • System Network Configuration Discovery (Execution) | • File and Directory Discovery (Discovery)                |
| • System Network Connections Discovery (Execution)   | • Network Share Discovery (Discovery)                     |
| • System Owner/User Discovery (Execution)            | • Password Policy Discovery (Discovery)                   |
| • System Service Discovery (Execution)               | • Peripheral Device Discovery (Discovery)                 |
| • System Time Discovery (Execution)                  | • Permission Groups Discovery (Discovery)                 |
| • CMSTP (Execution) (Execution)                      | • Process Discovery (Discovery)                           |
| • Command-Line Interface (Execution)                 | • Remote System Discovery (Discovery)                     |
| • InstallUtil (Execution)                            | • Security Software Discovery (Discovery)                 |
| • Local Job Scheduling (Execution)                   | • System Information Discovery (Discovery)                |
| • Mshta (Execution)                                  | • System Network Configuration Discovery (Discovery)      |
| • Regsvcs/Regasm (Execution)                         | • System Network Connections Discovery (Discovery)        |
| • Regsvr32 (Execution)                               | • System Owner/User Discovery (Discovery)                 |
| • Rundll32 (Execution)                               | • System Service Discovery (Discovery)                    |
| • Signed Binary Proxy Execution (Execution)          | • Shared Webroot (Lateral Movement)                       |
| • Signed Script Proxy Execution (Execution)          | • Taint Shared Content (Lateral Movement)                 |
| • Trusted Developer Utilities (Execution)            | • Windows Remote Management (Lateral Movement)            |

### 3.4.2 Conclusion MITRE ATT&CK

The MITRE ATT&CK platform serves a wide spectrum of attack techniques from different platforms. Nevertheless, for each attack technique a detection is described, but they do not focus in particular on how the event log settings have to be set. Hence, it was difficult to make a clear statement about which group policy setting must be configured in order to achieve a solid event logging for a good readiness of a system. However, the platform served a lot information regarding which critical aspects have to be audited.

### 3.5 SysAdmin, Networking and Security Digital Forensics and Incident Response

The SysAdmin, Networking and Security-Institute (SANS) published a poster with information according to “Windows Forensic Analysis” [51]. This poster is not a guideline for lateral movement or advanced persistence threat detection but more of a cheat-sheet for forensic analysis. It describes its use as follows:

*“[...]Use this poster as a cheat-sheet to help you remember where you can discover key Windows artifacts for computer intrusion, intellectual property theft, and other common cyber crime investigations.” [51]*

The only part of the poster that is important to us, i.e. which contains logging recommendations, is “Account Usage”, which is divided into these subcategories:

- Last Login
- Last Password Change
- RDP Usage
- Services Events
- Logon Types
- Authentication Events
- Success/Fail Logons

These subcategories are described and compared to the PoC in the next section.

#### 3.5.1 Comparison PoC - SANS

In this section the “Account Usage” parts from the SANS poster [51] will be described and compared to the PoC.

**Last Login** *Lists the local accounts of the system and their equivalent security identifiers.[...] [51]*

- **EventIDs:** -
- **Consideration:** No, does not contain any recommended event logs but only a location where the last logon time is stored. [51]

**Last Password Change** *Lists the last time the password of a specific local user has been changed.[...] [51]*

- **EventIDs:** -
- **Consideration:** No, does not contain any recommended event logs but only a location where the last password change time is stored. [51]

**RDP Usage** [...] *Track Remote Desktop Protocol logons to target machines.* [...] [51]

- **EventIDs:**

...

- *Event ID 4778 - Session Connected/Reconnected*

- *Event ID 4779 - Session Disconnected*

...

[51]

- **Consideration:** Yes, although the PoC did not include these events, the Readinizer will by logging the “Audit Other Account Management Events”. [51]

### Services Events

- *Analyze logs for suspicious services running at boot time*
- *Review services started or stopped around the time of a suspected compromise*

[51]

- **EventIDs:** *All Event IDs reference the System Log*

- *7034 - Service crashed unexpectedly*

- *7035 - Service sent a Start/Stop control*

- *7036 - Service started or stopped*

- *7040 - Start type changed (Boot / On Request / Disabled)*

- *7045 - A service was installed on the system (Win2008R2+)*

- *4697 - A service was installed on the system (from Security log)*

[51]

- **Consideration:** All Events from the System Log are logged by default. [9] The event 4697 is not included in to PoC, but will be in the Readinizer by “Audits Security System Extension”.

**Logon Types** *Logon Events can give us very specific information regarding the nature of account authorizations on a system if we know where to look and how to decipher the data that we find. In addition to telling us the date, time, username, hostname, and success/failure status of a logon, Logon Events also enables us to determine by exactly what means a logon was attempted.* [51]

- **EventIDs:** [...] *Event ID 4624* [51]

- **Consideration:** No, these events are logged both in the PoC as well as in the Readinizer. [3, p. 17]

### Authentication Events *Authentication mechanisms [51]*

- **EventIDs:** *Event ID Codes (NTLM protocol)*
  - 4776: *Successful/Failed account authentication Event ID Codes (Kerberos protocol)*
  - 4768: *Ticket Granting Ticket was granted (successful logon)*
  - 4769: *Service Ticket requested (access to server resource)*
  - 4771: *Pre-authentication failed (failed logon)*

[51]

- **Consideration:** No, these events are logged both in the PoC as well as in the Readinizer. [3, p. 18]

**Success/Fail Logon** *Determine which accounts have been used for attempted logons. Track account usage for known compromised accounts.[51]*

- **EventIDs:**

...

- 4624 – *Successful Logon*
- 4625 – *Failed Logon*
- 4634 / 4647 – *Successful Logoff*
- 4648 – *Logon using explicit credentials (Runas)*
- 4672 – *Account logon with superuser rights (Administrator)*
- 4720 – *An account was created*

[51]

- **Consideration:** No, these events are logged both in the PoC as well as in the Readinizer. [3, p. 17-19]

#### 3.5.2 Conclusion SANS

The poster published by SANS [51] cannot be compared with the other documents in the benchmark because it has a different purpose. It serves as a tool for forensic analysis, but helpful information has been obtained anyway. Some settings that were not implemented in the PoC could be adjusted for the Readinizer. The weighting of the poster is not as high as other papers in this benchmark.

### 3.6 Overall conclusion

This chapter compares the different papers, shows their similarities and the most important differences. It also discusses their importance for our project and which settings were effectively used for the Readinizer.

All analyzed sources, despite possible differences, show the importance of solid event logging. The common denominator of all documents was the Logon/Logoff logging as well as the Kerberos authentication to detect tools such as “Mimikatz”.

The main difference between the papers was the scope and level of detail of their recommendations. While CERT-EU focuses minimalistically on the logging of pass-the-hash and pass-the-ticket attacks, the NSA paper, for example, covers a much larger scope up to the logging of wireless devices. Additionally the style of the papers was very different. While CERT-EU and NSA focused on event IDs, ASAC and MITRE focused on the audit setting. While the PoC also focused on EventIDs, this approach has now been dropped. In the Readinizer, more focus is placed on audit settings. Serious differences in content between the individual proposals could not be identified.

Neither the CERT-EU nor the SANS paper have a significant influence on the Readinizer settings. Almost all recommended logs were already covered by the PoC [3]. The MITRE ATT&CK as well as the NSA paper cover a larger scope than intended for the Readinizer. Despite this, they were able to provide an interesting point of view and some changes were made based on their statements. For example, the logs of the Windows Defender were included in the Readinizer. ACSC's paper was very useful to check it against the PoC. Although it covered many of the settings recommended by the ACSC, some adjustments to the Readinizer settings have been made.

To conclude the benchmark, it can be said that the Proof of Concept forms a good and solid basis for the Readinizer. Nevertheless, some adjustments are made based on the other papers. The precise settings for the Readinizer can be found in the next chapter.

#### 3.6.1 GPO Settings Readinizer

There are several combinations of settings which can be configured:

**Not Configured:**

Nothing selected

**0 - No Auditing:**

“Configure the following audit events:”

**1 - Success:**

“Success”

**2 - Failure:**

“Failure”

**3 - Success and Failure:**

“Success” and “Failure”

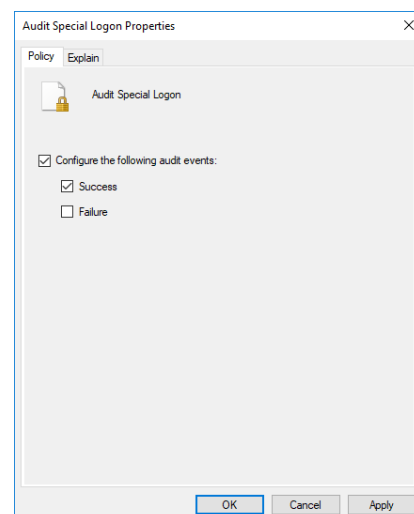


Figure 2.7: Advanced Audit Policy - Logon/Logoff - Audit Special Logon



Computer Configuration\Policies\Windows Settings\Security Settings\ Local Policies\Security Options	
GPO Setting	Configuration
Audit: Force audit policy subcategory settings (Windows Vista or later) to override audit policy category settings	Enable

Table 2.18: Force Audit Policy Subcategory Settings

Computer Configuration\Policies\Windows Settings\Security Settings\ Advanced Audit Policy Configuration\Account Logon	
GPO Setting	Configuration
Audit Kerberos Authentication Service	Success & Failure
Audit Kerberos Service Ticket Operations	Success & Failure

Table 2.19: Advanced Audit Policy Setting Account Logon

Computer Configuration\Policies\Windows Settings\Security Settings\ Advanced Audit Policy Configuration\Account Management	
GPO Setting	Configuration
Audit Computer Account Management	Success
Audit Other Account Management Events	Success
Audit Security Group Management	Success
Audit User Account Management	Success & Failure

Table 2.20: Advanced Audit Policy Setting Account Management

Computer Configuration\Policies\Windows Settings\Security Settings\ Advanced Audit Policy Configuration\Detailed Tracking	
GPO Setting	Configuration
Audit Process Creation	Success
Audit Process Termination	Success

Table 2.21: Advanced Audit Policy Setting Detailed Tracking

Computer Configuration\Policies\Windows Settings\Security Settings\ Advanced Audit Policy Configuration\DS Access	
GPO Setting	Configuration
Audit Directory Service Changes	Success

Table 2.22: Advanced Audit Policy Setting DS Access

Computer Configuration\Policies\Windows Settings\Security Settings\ Advanced Audit Policy Configuration\Logon/Logoff	
GPO Setting	Configuration
Audit Account Lockout	Failure
Audit Group Membership	Success
Audit Logoff	Success
Audit Logon	Success & Failure
Audit Other Logon/Logoff Events	Success & Failure
Audit Special Logon	Success

Table 2.23: Advanced Audit Policy Setting Logon/Logoff

Computer Configuration\Policies\Windows Settings\Security Settings\ Advanced Audit Policy Configuration\Object Access	
GPO Setting	Configuration
Audit File Share	Success & Failure
Audit File System	Success & Failure
Audit Handle Manipulation	Success
Audit Kernel Object	Success & Failure
Audit Other Object Access Events	Success & Failure
Audit Registry	Success & Failure
Audit SAM	Success & Failure

Table 2.24: Advanced Audit Policy Setting Object Access

Computer Configuration\Policies\Windows Settings\Security Settings\ Advanced Audit Policy Configuration\Policy Change	
GPO Setting	Configuration
Audit Audit Policy Change	Success
Audit MPSSVC Rule-Level Policy Change	Success

Table 2.25: Advanced Audit Policy Setting Policy Change

Computer Configuration\Policies\Windows Settings\Security Settings\ Advanced Audit Policy Configuration\Privilege Use	
GPO Setting	Configuration
Audit Non Sensitive Privilege Use	Success & Failure
Audit Sensitive Privilege Use	Success & Failure

Table 2.26: Advanced Audit Policy Setting Privilege Use

Computer Configuration\Policies\Windows Settings\Security Settings\ Advanced Audit Policy Configuration\System	
GPO Setting	Configuration
Audit Security System Extension	Success
Audit System Integrity	Success & Failure

Table 2.27: Advanced Audit Policy Setting System

Computer Configuration\Policies\Administrative Templates\ System\Audit Process Creation	
GPO Setting	Configuration
Include command line in process creation events	Enabled

Table 2.28: Administrative Template System

Computer Configuration\Policies\Administrative Templates\ Windows Components\Windows PowerShell	
GPO Setting	Configuration
Turn on Module Logging	Enabled Add wildcard value in <b>Module Names</b> : *
Turn on PowerShell script Block Logging	Enabled

Table 2.29: Administrative Template Windows Components Windows PowerShell

Computer Configuration\Policies\Administrative Templates\SCM: Pass the Hash Mitigations	
<b>Prerequisites</b>	<p>Note: <b>Must be performed on every computer system which will include this setting!</b></p> <ol style="list-style-type: none"> <li>1. To enable LSA Protection, Secure Boot must be enabled → follow the instructions on [28] and do the opposite to enable Secure Boot</li> <li>2. Download the Microsoft “Microsoft Security Compliance Toolkit 1.0”, extract it and copy “.\PtH.admx” and “en-US\PtH.adml” from “. \MSSecurityComplianceToolkit\Win81-WS2012R2-IE11-Baselines-FINAL\Win81-WS2012R2-IE11-Baselines\Administrative Template\PolicyDefinitions” to “C:\Windows\PolicyDefinitions”</li> </ol>
<b>GPO Setting</b>	<b>Configuration</b>
Lsass.exe audit mode	<p>Enabled</p> <p>Adds the following registry key: SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\LSASS.exe\ "AuditLevel"=dword:00000008</p>
LSA Protection	<p>Enabled</p> <p>Adds the following registry key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\ "RunAsPPL"=dword:00000001</p>

Table 2.30: Add Registry Key for LSA Protection [29]

### 3.6.2 Additional Readinizer Settings

#### Sysmon

Sysmon, short for System Monitor, is a Windows system service that... It is strongly recommend to have Sysmon [52] installed and running by both, the ACSC [11, p.3] as well as by the PoC [3, p.11]. The importance of Sysmon for the PoC and Readinizer was summarized in the PoC:

*“Due to the fact that Sysmon will log not only the name of an executable but also the corresponding hash value, Sysmon is an important tool to be enabled for solid detection of attacks. So Sysmon has to be detected if it is running or not to prepare an environment for a good readiness.” [3, p.11]*

## 4 Test Environment

This chapter of the report describes the setup of the testing environment in which not only the tools during the research were tested, but were also used to test the Readinizer itself.

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 three domains split up into two trees, a parent-child domain and a separate tree domain. 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:

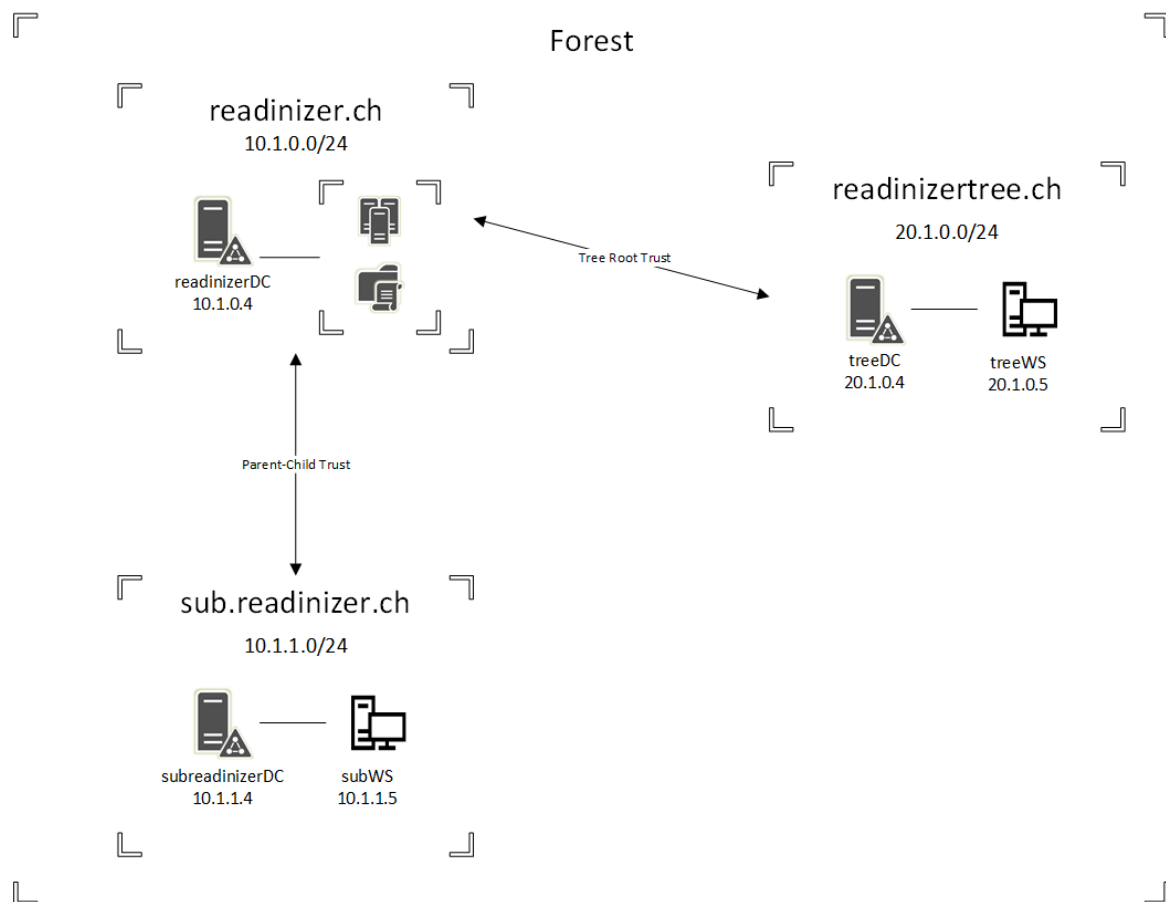


Figure 2.8: Test Environment

Each domain has its own domain controller, which is marked with a DC at the end of its name. The “readinizerDC” form the domain readinizer.ch is the forest root domain. The sub.readinizer.ch is its subdomain/childdomain. The treereadinizer.ch is located in a separate tree, because it is the only domain in the tree it is also the tree root domain. Hence, it has a two-way transitive trust with the forest root domain.

#### 4.1 Domain User

Three users were configured for the logfarm-network:

Domain	Name	Privileges
readinizer.ch	domainadmin	Enterprise Administrator
sub.readinizer.ch	domainadmin	Domain Administrator
readinizertree.ch	domainadmin	Domain Administrator
readinizertree.ch	Alice	User

Table 2.31: Test Environment User

## 4.2 Domain readinizer.ch

The readinizer.ch domain was set up to test the Readinizer first in a single domain, because the Azure Cloud Subscription limits the number of Virtual-Cores. For this purpose, the following structure of Organizational Units and Members was established:

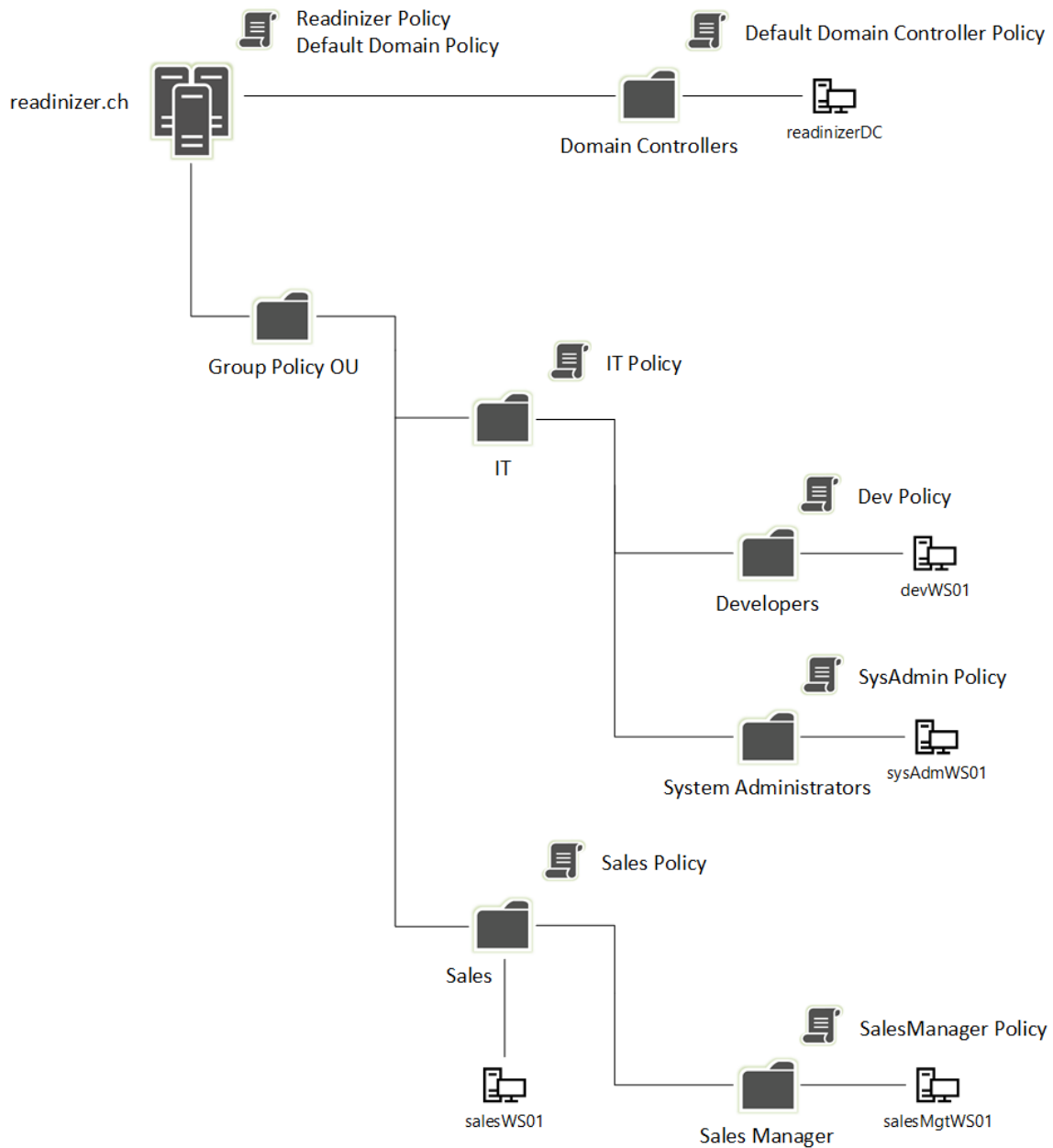


Figure 2.9: Domain readinizer.ch

### 4.3 Difficulties

Various difficulties occurred which are presented in this subsection.

#### Firewall setting for Internet Control Message Protocol (ICMP)

After setting up the “readinizer.ch” domain, its subdomain “sub.readinizer.ch” and the tree domain “readinizertree.ch”, their domain controller and DNS servers were installed. To make sure that the domains can reach each other, a virtual network peering was made. But they still could not ping each other. This issue did occur in an earlier project. After changing the in- and outbound rules in the firewall settings for the ICMP the pinging worked for the readinizer.ch and its subdomain, to make it work for the tree domain it was necessary to set up a DNS Zone.

#### DNS Zone Forwarding

To allow the tree domain to communicate with the forest root domain and its subdomains, it was necessary to install a DNS Forwarding zone on the domain controller. After the setup was done, it was possible to ping the “treeWS” but no one could reach the “treeDC” from outside the domain. After quite some research, the development-team decided to reboot the whole system. Subsequently, the pinging worked flawlessly.



## 5 Design

This section focuses on answering the following question:

*"How can the problem domain be defined and how do the individual components interact with each other?"*

Accordingly, a Unified Modeling Language (UML) class diagram and a description are used to illustrate the problem domain.

### 5.1 Domain Analysis

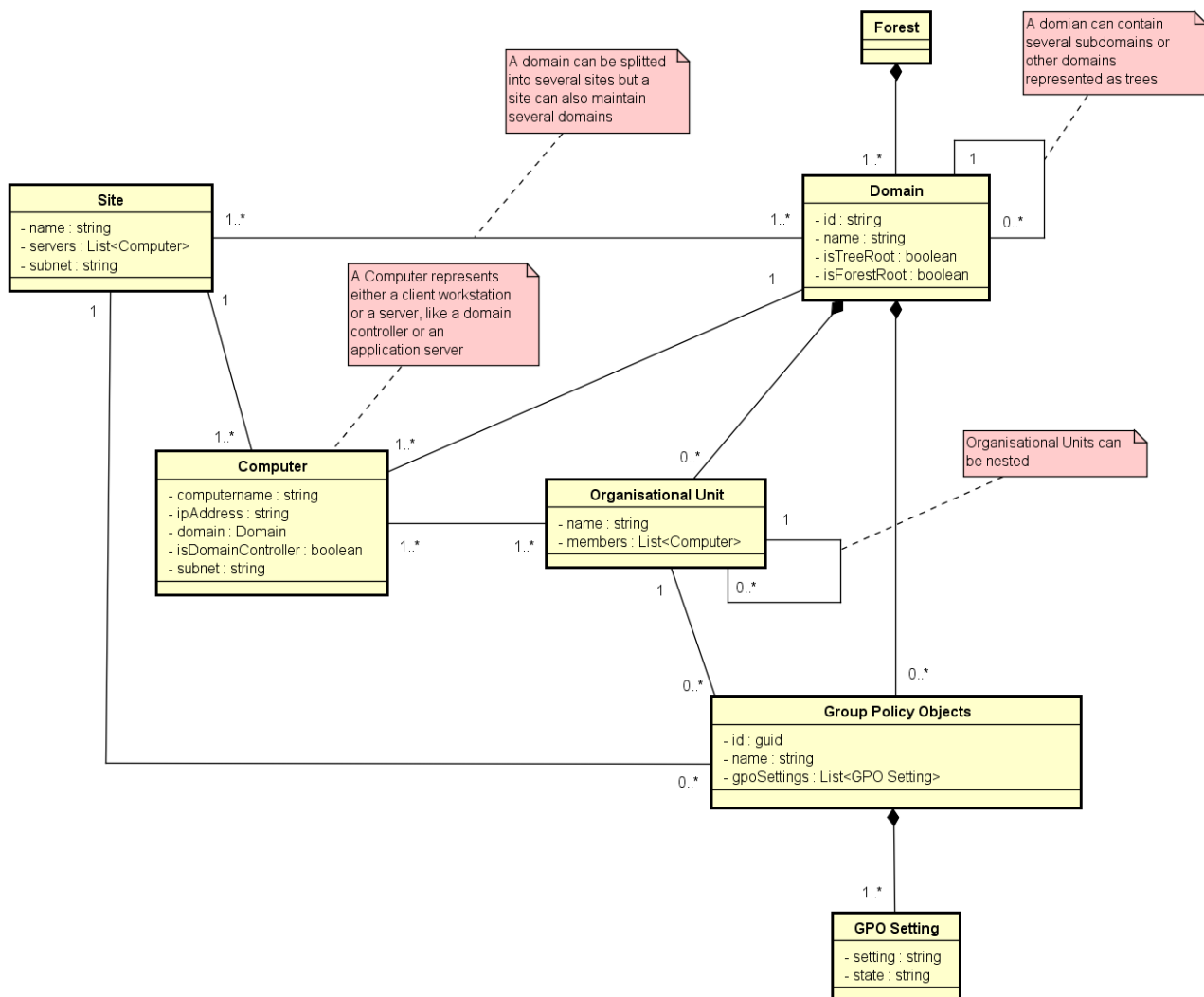


Figure 2.10: Active Directory Domain Model

#### Forest

An Active Directory consists of a forest whereby all its other objects like domains, sites, organizational units live - according to 2.1.1 Forests. Seeing as the forest is just a container for all these objects, it will not deliver important information for the Readinizer and therefore not many attributes are necessary to store, besides the name.

## Domain

A domain in an Active Directory is one form of grouping and is used to manage objects of the same namespace. Every forest contains exactly one forest root domain and hence is the root of the forest tree. It is possible to add several subdomains to this forest root domain to split the objects in different namespaces (like `sub.readinizer.ch`) and have an easier logical management. Tree domains are used to link domains with a non-contiguous namespace (i.e. `readinizertree.ch`) in respect of the forest root domain (`readinizer.ch`). These subdomains and tree domains can have subdomains as children as well (e.g. `child.sub.readinizer.ch` or `child.readinizertree.ch`), which is shown in the domain model as a self reference. For more information see 2.1.1 Domains.

## Site

Sites are used to split objects of an Active Directory in a physical manner to avoid high network traffic exchange of domain information (see 2.1.1 Site Objects). A domain has at least one site but can have several sites in addition. But it is also possible to have several domains at a site, for example a domain and its subdomain.

## Computer

An Active Directory has the possibility to store several different objects (e.g. client computer, servers, printers etc.). The class “computer” in this domain model is used to describe just two of these objects: client computers and servers. To fulfill the requirement only those two sorts of objects are necessary. Although we differentiate between client computers and servers, a server can have different functionalities in its domain. For example a server can be a usual server like an application or database server but can also act as a domain controller. Domain controllers can also be differentiated by the mode they are acting in. A domain controller can act as the forest root domain controller, a tree root domain controller or a normal domain controller. In order to visualise the whole Active Directory as a tree, it is necessary to store this information. In addition, this information is used to gather information about the next level - the organizational units.

## Organizational Unit

Organizational units are used to bring another level of logical abstraction into an Active Directory (see 2.1.1 Organizational Units). An organizational unit can only be created in a domain and has therefore a composite relation to the domain. Although organizational units are the smallest units of an Active Directory, they can be used as buckets for other organizational units and hence organizational units can be nested as required. It is necessary to gather all information about the member computers in a organizational unit to gather the RSoP in a further step for analysis.

## Group Policy Objects

Group policy objects are used to build sets of different configuration setups for computers in an Active Directory (see 2.2 Group Policy Objects). Hence, group policy objects have a set of group policy settings which define the effective settings. These group policy objects can only be created in a domain and therefore depend on the domain. Group policy objects can not only be linked to domains but also to organizational units and sites.

## GPO Setting

A group policy setting describes a single computer setting which is delegated through a group policy object to a computer.

## 5.2 Graphical User Interface Design

This section describes and shows how the graphical user interface (GUI) is planned to look like.

### 5.2.1 Start screen

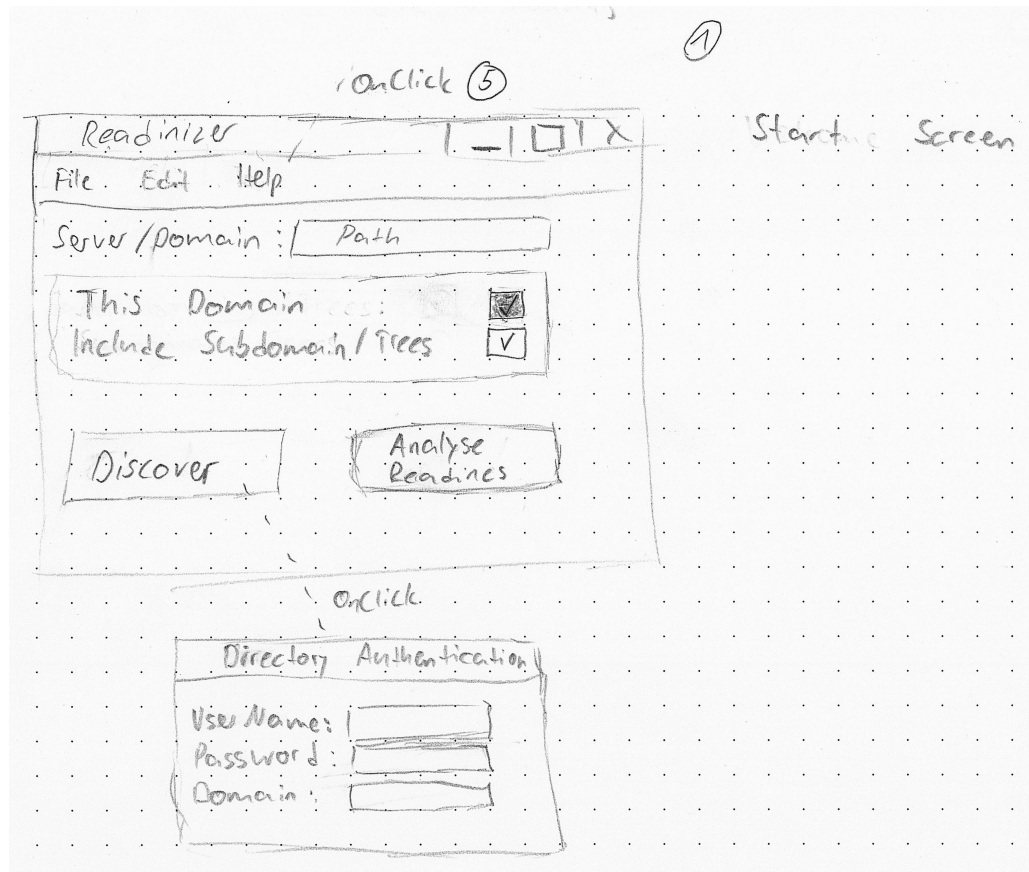


Figure 2.11: GUI Readinizer StartUp View

This sketch shows the draft of the start screen. The user specifies the domain from which he wants to check the GPO settings, as well as if he also wants to check its subdomains and treedomains. The “Discover” button checks if the specified domain is reachable and if the current user has enough rights to read the objects from the Active Directory. If this is not the case, a new window will pop up where the user can provide credentials which have these rights. The “Analyze Readiness” button is greyed out until the discovery process runs successfully. When it is executed, the application collects the domain and its subdomains, the contained organizational units and the members of the units.

### 5.2.2 Result overview

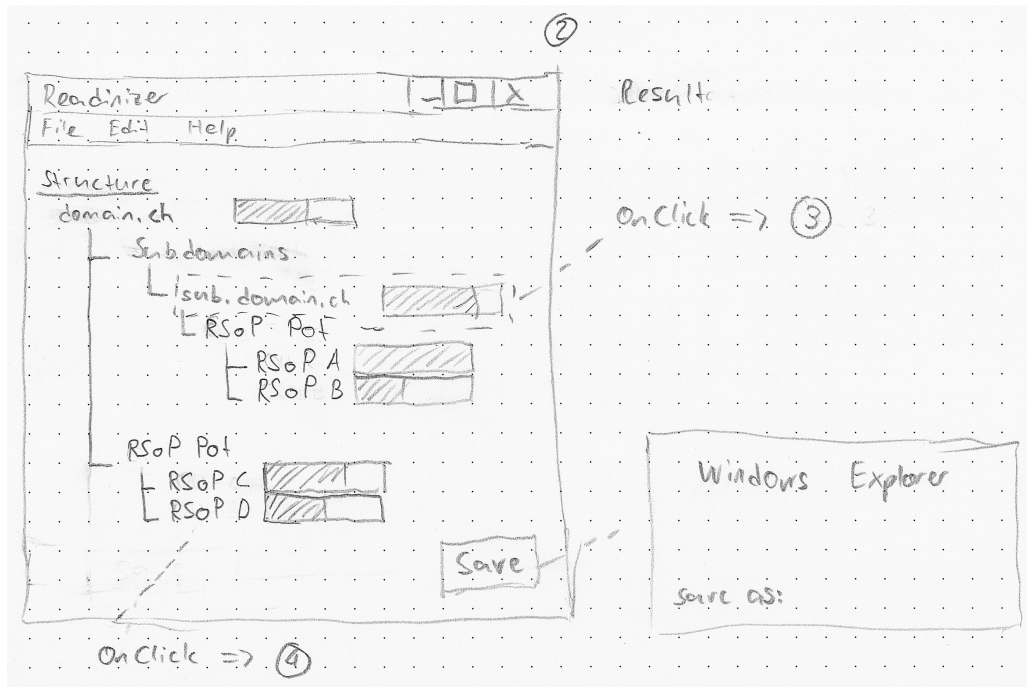


Figure 2.12: GUI Readinizer Result View

After running the analysis, an overview of the results will be shown. The overview displays the structure of the network. A bar gives a rough overview of the readiness for each domain, subdomain or RSoP. A more detailed overview can be opened by clicking on the respective name. A “Save” button will open a new window where the user can decide where he wants to save this data.

### 5.2.3 Result per domain

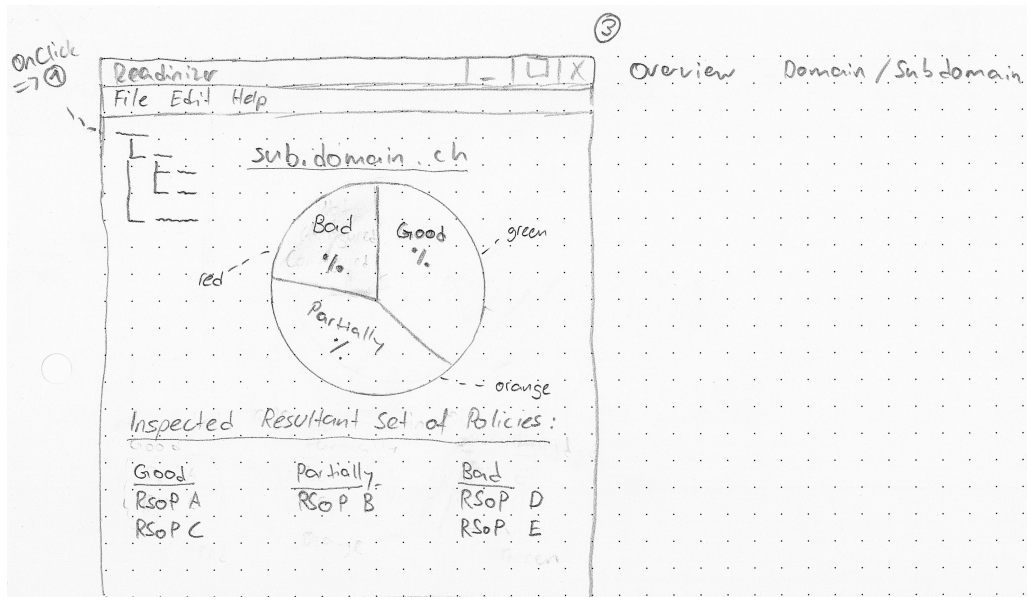


Figure 2.13: GUI Readinizer Overview Domain/Subdomain View

This view displays a pie chart which is divided into three parts. The “Good” section which is green, the “Partially” section which is orange and the “Bad” section which is red. This section will give a brief overview of the GPO settings in the domain. Underneath the chart is a more detailed listing of the different RSoPs of the domain, split into the same three categories.

### 5.2.4 Result per RSoP

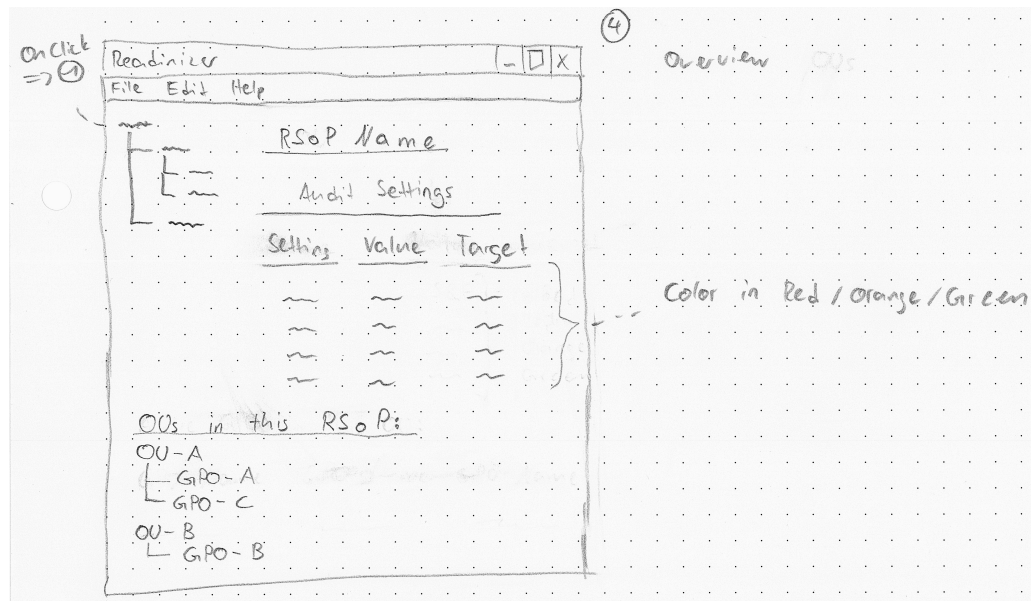


Figure 2.14: GUI Readinizer Overview OUs View

This view displays a more detailed overview of each RSoP. It will list the crucial GRO settings, the current value and the value that is recommended. Underneath that is an overview of which Group Policies this RSoP is composed of.

### 5.2.5 Navigation bar

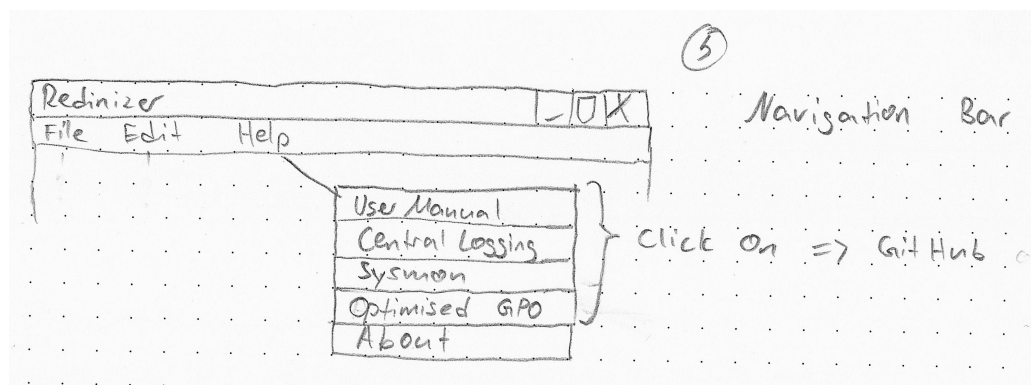


Figure 2.15: GUI Readinizer Navigation Bar

The main part of the navigation bar is the "Help" section. This is where the user manual is provided, as well as other documents to optimise the readiness of a system. The navigation bar is available in every screen.

## 5.3 Data Model

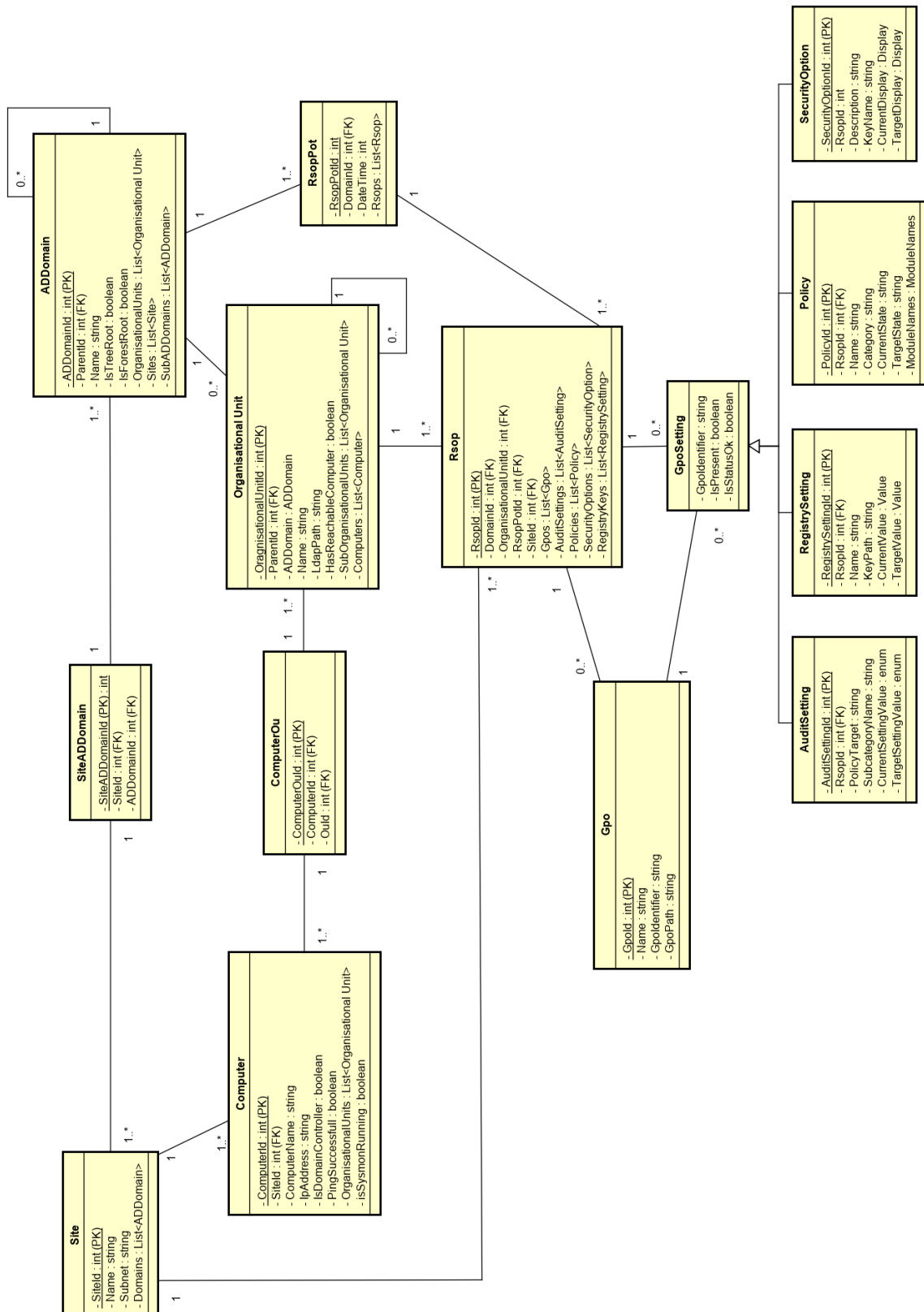


Figure 2.16: Data Model



## 5.4 Differences to the Domain Model

### 5.4.1 General

The data model differs slightly from the developed domain model. During the implementation it was noticed that not only the properties developed in the domain model are needed, but also others.

### 5.4.2 Rsop

The class Rsop describes the RSoP received from the clients and the data which are analyzed with the Readinizer. It also contains information about GPOs, site, domain and OUs. This information is then used to then make a statement about where the configured settings on the client originate from.

### 5.4.3 RsopPot

The class RsopPot reflects all RSoP with identical security settings. This class does not occur in any Microsoft AD environment. However, it was created to abstract and simplify the problem. For the user, these RsopPots are represented as “Group of identical security settings”. This grouping of RSoP ensures that there is no unnecessary configuration work within the GPO management. The grouping allows the user to identify which OUs still contain incorrect settings and allows to make changes more efficiently.

### 5.4.4 GPOSetting

The GPO settings could not be implemented quite as trivially as was intended in the domain model. A GPO setting is divided into different parts of computer settings, resulting in a corresponding RSoP as an XML file. As a result, the more specific classes **AuditSetting**, **RegistrySetting**, **Policy** and **SecurityOption** were defined (for further details see section 7.6.2 Implementation - AnalysisService).



## 6 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.

### 6.1 Use Cases Readinizer (UC-R<sub>n</sub>)

The system administrator and security advisor are grouped in the use cases as the user and is considered to be the main actor of the use cases. The following figure 2.17 Use Case Diagram shows the use cases and their relationships with the actors.

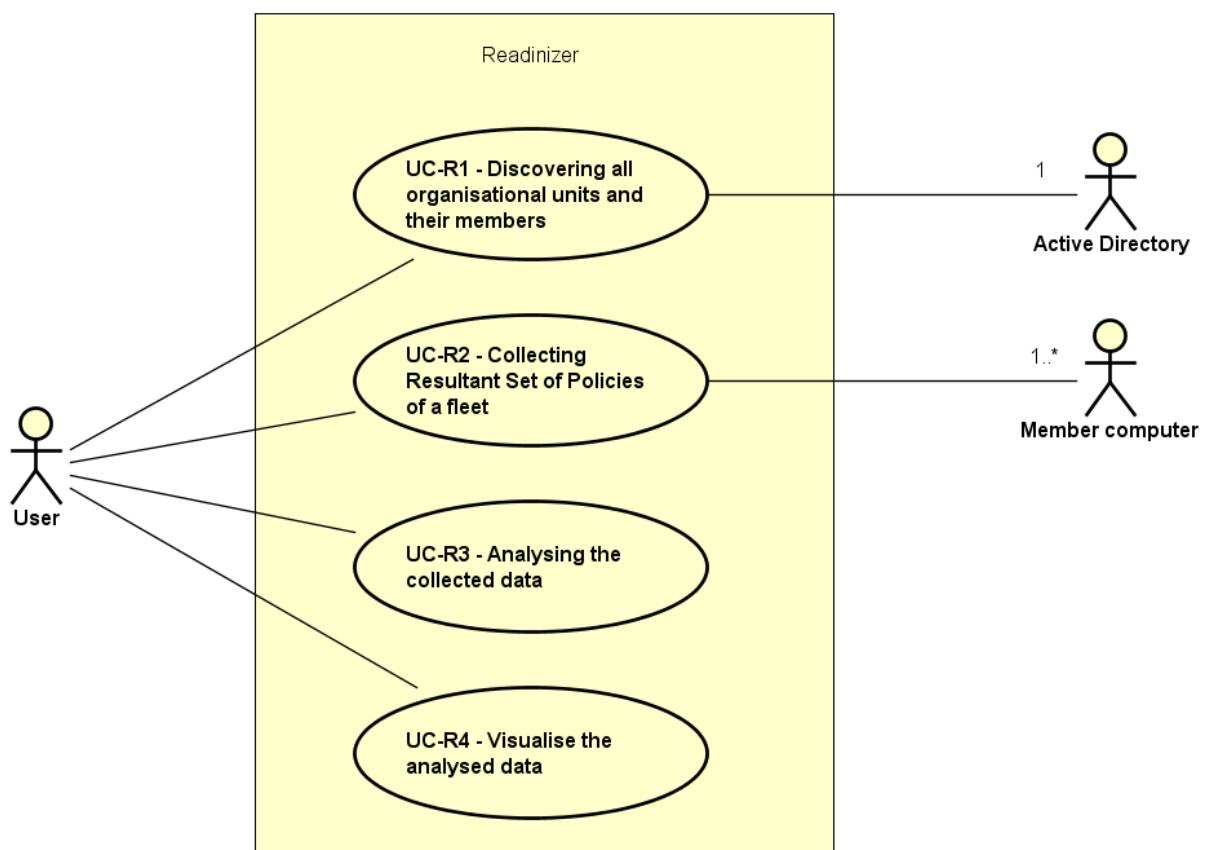


Figure 2.17: Use Case Diagram

### 6.1.1 UC-R1 - Discovering all organizational units and their members

**Description**

The application discovers all domains, sites and organizational units with their member computers in the Active Directory environment. The application organizes the data into domains, subdomains/trees and organizational units.

**Precondition**

The user has to be part of the forest which is to be analyzed. Therefore, the application has a connection to the forest and its domains which will be analyzed. Either the executing user has the necessary privileges (read access on the Active Directory) or credentials are provided to give the application these access rights.

**Main Success Scenario**

1. The user defines the Active Directory, which has to be discovered, by providing the fully qualified domainname.
2. The user who is running the application, has access read rights or is in possession of credentials with these access read rights of the respective Active Directory.
3. The application collects all organizational units and their members of the Active Directory environment
4. The application stores the collected data in a datastore for further processing.

**Postcondition**

The application is in a blocked state in order to fulfill UC02. The data is stored in a temporary datastore.

**Error-condition**

- If the application does not have sufficient access rights for the Active Directory, the application issues an error message alerting the user of this specific error.
- If the provided fully qualified domainname is incorrect, the application issues an error message alerting the user of this specific error.
- If the application does not receive an answer from the Active Directory, the application issues an error message alerting the user of this specific error.

### 6.1.2 UC-R2 - Collecting Resultant Set of Policies of a fleet

**Description**

The application collects the Resultant Set of Policies from one member computer of each organizational unit in the Active Directory environment.

**Precondition**

For each organizational unit a member computer is available and accessible. The precondition from UC01 also applies to this use case.

**Main Success Scenario**

1. The application collects for each organizational unit a RSoP from a member computer.
2. The application stores the collected data in a datastore for further processing.

**Postcondition**

The application is in a blocked state in order to fulfill UC03. The data is stored in a temporary datastore.

**Error-condition**

- If the computer member of an organizational unit cannot provide a RSoP, the next computer will be called to provide its RSoP.
- If no computer member is available from an organizational unit, no data or an appropriate value to indicate such a case is stored for this organization unit.

### 6.1.3 UC-R3 - Analyzing the collected data

**Description**

The application analyzes the collected Resultant Set of Policies according to the predefined group policy settings on the basis of the proof of concept and the findings of the benchmark. The RSoPs are grouped into collections of identical RSoP.

**Precondition**

The application has successfully fulfilled the UC02. Therefore, the data stored in UC02 is available.

**Main Success Scenario**

1. The application analyzes the collected data (RSoP)
2. The application prepares the data for export as machine-readable output as well as to be visualized

**Postcondition**

The application is in a blocked state in order to fulfill UC04. The analyzed data is stored in a temporary datastore.

**Error-condition**

If in an organizational unit no RSoP is received, this organizational unit is marked in an appropriate way to indicate this case for the visualization.

#### 6.1.4 UC-R4 - Visualize the analyzed data

**Description**

The application visualizes the analyzed data in an appealing design which represents an abstract form of the Active Directory. The representation is divided into a forest of domains, subdomains/trees and groups of identical Resultant Set of Policies which contain the respective organizational units.

**Precondition**

The application has successfully fulfilled the UC03. Therefore, the analyzed data is available to visualize.

**Main Success Scenario**

1. The application visualizes the analyzed data.
2. The user is able to navigate through the visualized data, whereby each level provides data in more detail.
3. The user is able to export the visualized data as a machine-readable or user-friendly output.

**Postcondition**

The application is ready for the user to navigate through the visualization.

**Error-condition**

If no analyzed data is provided, the application will visualize this case in an understandable way for the user.

## 6.2 Use Cases Optimizer (UC-On)

### 6.2.1 UC-O1 - Provide a recommended Group Policies

**Description**

The application serves a link to a repository whereby a recommended group policy is provided as a minimal set of configuration settings which can be imported system-wide. This recommended group policy is not a customer-specific recommendation, but a basic recommendation for solid event logging.

**Precondition**

The user has access to the internet. The repository provides a brief documentation on how to implement the group policy.

**Main Success Scenario**

1. The user downloads the recommended group policy
2. The user implements the group policy as described in the repository

**Postcondition**

The system logs all recommended events. If the Readinizer is executed again, no more deviations from the recommended settings should be detected.

**Error-condition**

If there are still deviations from the recommended settings after a new execution, the system administrator must carry out a manual investigation.

### 6.2.2 UC-O2 - Provide manual for fleet-wide Sysmon installation

**Description**

A manual developed for a system administrator is provided, which describes how to detect if Sysmon is installed as well as giving instructions how to install Sysmon on a fleet.

**Precondition**

The user has access to the internet. The repository provides a manual on how to install Sysmon fleet-wide.

**Main Success Scenario**

1. The system administrator downloads the manual
2. The system administrator checks if Sysmon is installed
3. The system administrator is able to install Sysmon fleet-wide

**Postcondition**

On every system of the fleet - which is relevant for the system administrator - Sysmon must be installed.

### 6.2.3 UC-O3 - Provide manual for fleet-wide central logging installation

#### Description

A manual developed for a system administrator is provided which describes how to install central logging for a fleet.

#### Precondition

The user has access to the internet. The repository provides a manual on how to install a simple central logging.

#### Main Success Scenario

1. The system administrator downloads the manual
2. The system administrator is able to install central logging

#### Postcondition

On every system of the fleet - which is relevant for the system administrator - must send its logs to a central logging server.

## 6.3 Non Functional Requirements (NFR)

NFR-No.	Description
NFR01	<b>Minimal target version</b>  The minimal target version of the system for the application to run must be Microsoft Windows 10 Professional v1709 and Microsoft Server 2016 Datacenter.
NFR02	<b>Performance</b>  The user shall not notice significant performance degradation of the system when using the application. While using the application, the functionality of the Active Directory shall not be altered in each way during the readiness analysis.
NFR03	<b>Network Performance</b>  The usage of the application shall not produce significant network traffic. More precisely, the application shall not exceed 5% of the regular network traffic of the organization. In addition, the network must remain stable in any case during the use of the application.
NFR04	<b>Runtime</b>  For smaller networks, up to 100 workstations, the runtime of the Readinizer Use Cases 6.1.1 UC-R1, 6.1.2 UC-R2, 6.1.3 UC-R3 and 6.1.4 UC-R4 must not exceed 15 minutes. Larger networks may have a longer runtime.

<b>NFR-No.</b>	<b>Description</b>
NFR05	<b>Usability</b> The application is delivered with as little as possible installation dependencies.
NFR06	<b>Usability</b> A system administrator is able to use the application without further instructions or manuals. An interested party of this application is able to use it after reading the provided manual.
NFR07	<b>Integrity</b> The Group Policy Objects provided to the user must not impact the function of the domain. No existing Group Policy Objects settings may be changed.
NFR08	<b>Integrity / Security</b> The event logs are to be sent protected during transmission for event forwarding. The minimal strength of the encryption must be 128-bit.
NFR09	<b>Security</b> Only certain machines are allowed to commit their event logs to the central logging station.
NFR10	<b>Security</b> Only privileged users are allowed to read the collected event logs on the central logging station.

Table 2.32: Non Functional Requirements

## 6.4 Logical Architecture - Package Diagram

### 6.4.1 Design Decisions

#### Repository Pattern

The Repository pattern is used so that the logic of loading data from the database can take place on the business layer, regardless of the technology used in the data access layer. Thus the layers are cleanly separated with an abstraction and a new database technology can be used without problems.

#### Unit of Work Pattern

The Unit of Work (UoW) pattern is used to support the Repository pattern. In fact, the UoW orchestrates the work of several repositories by creating one single database context which is shared through all the repositories. A single database context is especially important so that the n:m table relationships contained in the database can be filled and modified correctly and without great effort. Otherwise multiple context would exist in the application and would hold different knowledge about the data set. If one of these contexts receives a change on an existing dataset and a second one holds the same dataset, conflicts would occur when saving to the database. With a single context this problem is solved.

#### Services

Business tasks that belong together are offered as call methods in individual services.



### Model View ViewModel Pattern

The Model View ViewModel pattern (MVVM) is extremely common in C# WPF applications and has proven to be extremely suitable for the project, which is why this pattern is used for the presentation layer. This pattern brings the presentation layer another possibility to abstract view and viewmodel. However, the view is only responsible for displaying data and is bound to the viewmodel where it gets the data from. Nevertheless, no commands or other logical code are added to the view, the viewmodel is responsible for this - in the viewmodel the whole logic is handled for the view.

### Package Diagram

Figure 2.18 Logical Architecture - Package Diagram shows how the application is abstracted and which dependencies exist.

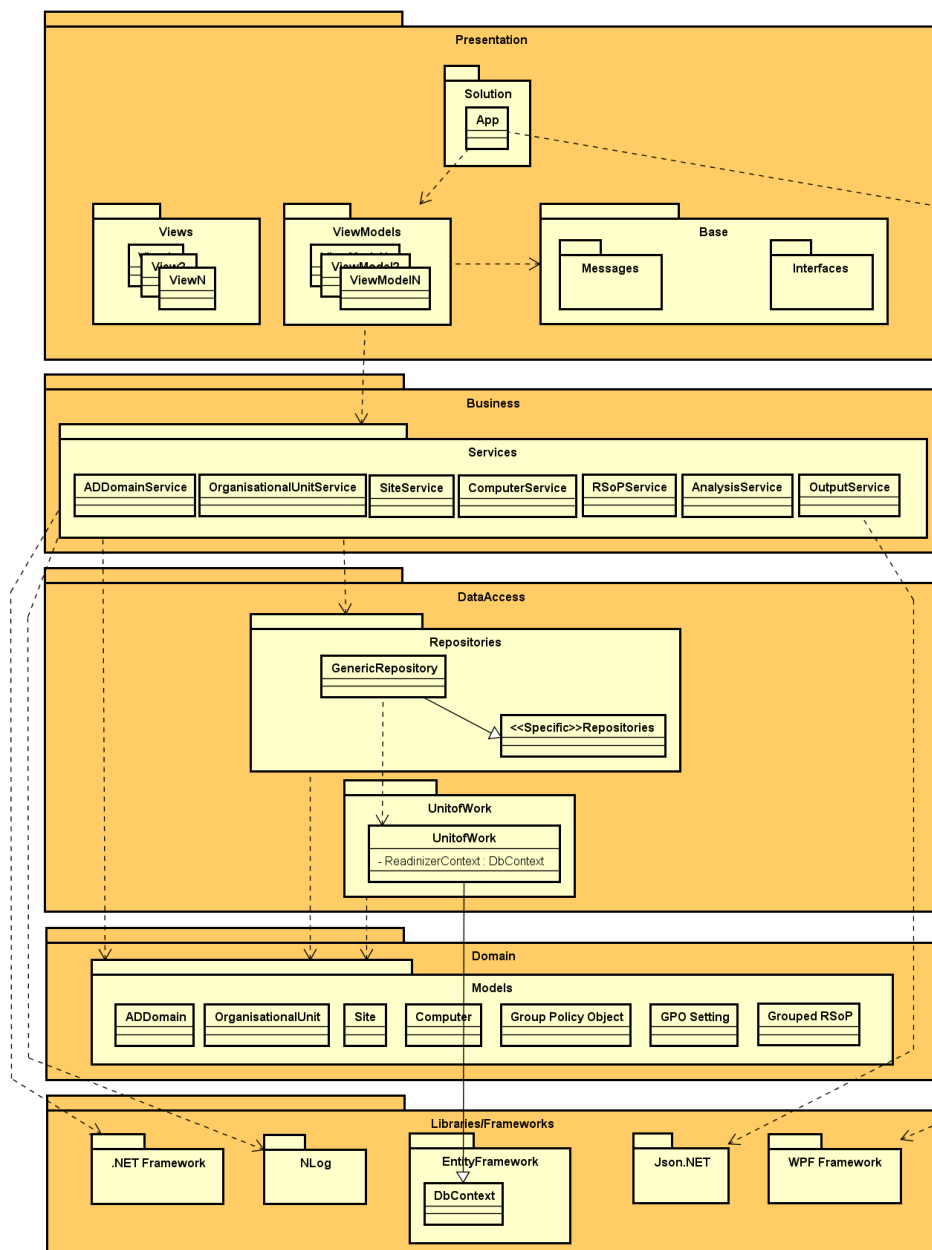


Figure 2.18: Logical Architecture - Package Diagram

## 6.5 System Architecture - Deployment Diagram

This section addresses the question *“How will the application on the system/systems deployed?”*.

### 6.5.1 Rejected System Architecture

In this system architecture, several components are used to split the functionality of the application in multiple parts. It includes a web application for the frontend, a web service to provide a RESTful HTTP backend for the frontend, a Windows service for gathering information about the Active Directory and a MSSQL database to store all of this information.

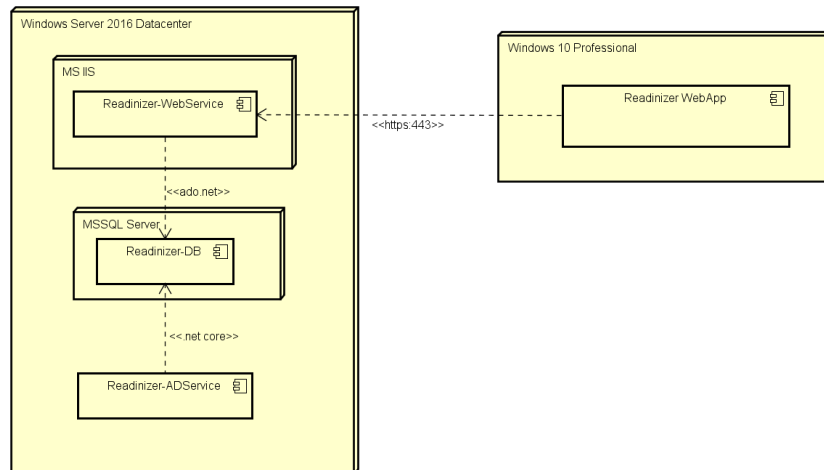


Figure 2.19: System Architecture - Deployment Diagram Rejected

This system architecture is rejected because it includes too many components and dependencies which would have to be installed before using the application. It is a non functional requirement that the application is delivered with as little as possible installation dependencies.

### 6.5.2 Accepted System Architecture

Due to the rejected system architecture, the decision was made to build a single host application with a Windows Presentation Foundation (WPF) frontend and a .NET Framework backend. After research and some testing, this decision proved to be a good choice because it is possible to gather all necessary information with the Lightweight Directory Access Protocol and Windows Management Instrumentation. This information is then stored in a LocalDB database.

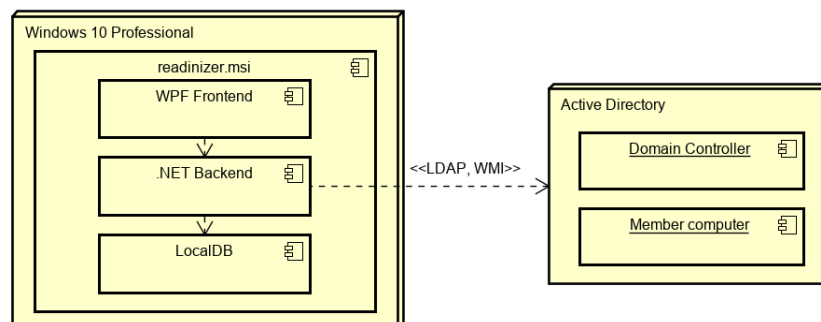


Figure 2.20: System Architecture - Deployment Diagram Accepted

## 6.6 Technologies

### 6.6.1 Chosen Technologies & Frameworks

#### C-Sharp

The application is written in the programming language C-Sharp (C#) [53] because the scope of this thesis and application is set to Windows environments. Moreover, with C# as the programming language, the services and functions of the application can be built in a very system close fashion. Another great benefit is that the application can use already existing namespaces and classes of Windows environments.

#### Windows Presentation Foundation Framework

The frontend is decided to build with the Windows Presentation Foundation (WPF) framework. [54] This decision is based on the chosen technology C# whereby WPF suits perfectly for a Windows single host application.

#### .NET Framework

It was decided to implement the backend part with the .NET framework. [55] Several namespaces and classes (e.g. `DirectoryEntry` [56] or `DirectorySearcher` [57]) which will be necessary to gather Active Directory information are included in the .NET framework.

#### LocalDB / Entity Framework

The Entity Framework (EF) [58] is used to store gathered information about the Active Directory in a LocalDB [59] database. If no database were used and the application ran in a large Active Directory environment, it is possible that there would not be enough volatile memory available to store this information. The database will prevent the system - on which the application is running - from falling into a memory overflow.

#### Newtonsoft Json.NET

Json.NET [60] is a high-performance JSON framework to serialize data into JSON-file-structures. This framework will give the ability to provide a machine-readable output which is one of the functional requirements for the Readinizer.

#### NLog

NLog [61] is used as the logging framework. NLog offers very good and simple configuration possibilities and can be integrated into the .NET framework.

#### LaTeX & Visual Studio Code

The documentation is written with LaTeX in Visual Studio Code with 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 had the experience that with LaTeX the formatting is more reliable than for example when Microsoft Word is used.

#### Microsoft Azure DevOps

Although not very bad experiences with Redmine as a project management tool were had, for this project it was decided to use Microsoft Azure DevOps [62] as a project management tool. New experiences with this platform should be gained. Azure DevOps is not only a project management tool, but also offers the possibility of continuous integration. Furthermore, Azure DevOps is designed for agile project handling which benefits us greatly.

### Azure Cloud

Microsofts Azure Cloud platform [63] is used for the test environment which is built with several different virtual machines for the domain controllers and clients in the respective domains. Two clients serve as the development clients and due to the fact that the development of the Readinizer depends on calls to the domain controllers this approach has been taken.

### GitHub

GitHub [64] is used as a version control tool for source code and documentation. GitHub has been elected due to its good reputation and the experience the developers already gained with. Moreover, the GitHub repositories can be linked to Microsoft Azure DevOps Pipelines which are used for continuous integration.

#### 6.6.2 Rejected Technologies

##### PowerShell

The decision not to use PowerShell as the programming language for this project is based on the following evidence from the proof of concept:

*The chosen technology PowerShell offered a simple implementation of the problem. However, an object-oriented approach and the realization of a classic software project is not ideal with PowerShell. It is noticeable that the language was originally a scripting language. [3, p. 55]*

For this reason, this programming language was no longer an option at all. Moreover, PowerShell can not be used to provide a user interface which is a functional requirement.

##### .NET Core Framework

Although it would be a great advantage for the further implementation of the Readinizer on other platforms (e.g. MacOS or Linux distributions), the decision not to use the .NET Core framework [65] depends on considerations of the other .NET technologies and libraries to be used. Since the Active Directory was developed on the .NET framework and still has many dependencies on it today, the new technology .NET Core may not support all abilities to gather all necessary information. There would be a risk that the requirements could not be fully met. In addition, the features of high performance, microservices or the possibility for docker containers are not required, all of which can be implemented with .NET Core. [66]

##### Prism Framework

In a first approach of the project it was intended to use the Prism Framework [67] which is described as follows:

*Prism is a framework for building loosely coupled, maintainable, and testable XAML applications in WPF, Windows 10 UWP, and Xamarin Forms. [68]*

This framework was rejected because the application is not as big as it has to be when put in such a strong guarded framework. Nevertheless, the applications architecture is set up to be loosely coupled, maintainable as well as testable. It will be built in several layers (i.e. Data Access, Business, Domain) to support loose coupling between classes, services, views and viewmodels.

**Neo4J**

Neo4J [69] is a graph database management system. With certain extensions, Neo4J makes it possible to present the collected data in a graphically appealing way. Other tools such as BloodHound [70] did use Neo4J to store and visualize their data. This technology was rejected due to the installation effort experienced by the user as well as the complexity to display the data in the desired way.

**Redmine**

Redmine [71] as a project management tool is rejected due to the decision of the usage of Microsoft Azure DevOps for project management. Redmine was used in the previously done Proof of Concept and served well, but in order to have everything (project management and continuous integration) in one tool, it was decided to set up Azure DevOps. In addition, Redmine does not provide such a suitable plug-in for agile project management, whereby this feature is already integrated in Azure DevOps.

## 7 Implementation

This sections shows the implementation of the application logic. First, a short description about the structure and details about central components of the application is given, followed by the description of the frontend implementation. Subsequently, the logical sequence is displayed according to the defined use cases. In a second step, special or important code fragments are described in detail. The last step of each implementation part deals with the problems that occurred.

### 7.1 Application Structure and Central Components

#### Structure

The application is divided into four projects within the Visual Studio solution and an additional project for tests. These four projects represent the package diagram (see 2.18 Logical Architecture - Package Diagram) which ensures a clean separation of the individual layers: Presentation (called Frontend in the Solution), Business, Data Access and Domain. This separation using layers also aims at a loose coupling of the individual components. The following figure shows the corresponding dependency diagram:

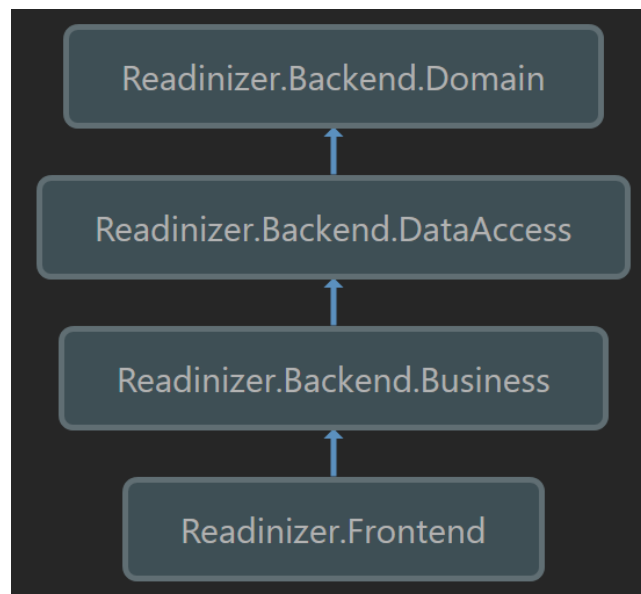


Figure 2.21: Dependency Diagram

#### 7.1.1 Dependency Injection

The Dependency Injection (Inversion of Control) approach is used for the application, i.e. the responsibility for object generation is transferred to an independent component. The code thus becomes more independent of its environment and testing by means of mocking is facilitated. The framework “Unity” was used for Dependency Injection.

The following listing 2.1 is a shortened summary of the registrations of the individual classes. An interface of the class to be injected and the class itself must be registered. It also makes sense to register certain components as singletons, such as the database context, the UnitOfWork class, or the SnackbarMessageQueue. These components should be instantiated exactly once within the application because they are reusable form everywhere within the application.

Listing 2.1: Unity Dependency Injection

```

1  protected override void OnStartup(StartupEventArgs e)
2  {
3      base.OnStartup(e);
4
5      IUnityContainer container = new UnityContainer();
6
7      container.RegisterType<IApplicationViewModel, ApplicationViewModel>();
8
9      container.RegisterType<IADDomainService, ADDomainService>();
10
11     container.RegisterSingleton<IReadinizerDbContext, ReadinizerDbContext>();
12     container.RegisterSingleton<IUnitOfWork, UnitOfWork>();
13
14     container.RegisterSingleton<ISnackbarMessageQueue, SnackbarMessageQueue>();
15
16     var applicationView = container.Resolve<ApplicationView>();
17     applicationView.Show();
18 }

```

### 7.1.2 Generic Repository

The repository pattern intends an abstraction layer between the data access layer and the business logic layer (see 6.4.1 Repository Pattern). In the beginning, we worked with individual repositories, each of which had its own implementation. However, since all repositories basically have the same tasks, it was decided to create a generic repository. This generic repository provides all methods for storing and reading the data in the database, as shown in the following listing 2.2. The implementation is based on the instructions “Implementing the Repository and Unit of Work Patterns in an ASP.NET MVC Application” from Microsoft. [72]

Listing 2.2: Generic Repository

```

1  public interface IGenericRepository<TEntity> where TEntity : class
2  {
3      Task<List<TEntity>> GetAllEntities();
4      void Add(TEntity entity);
5      void AddRange(List<TEntity> entities);
6      void Update(TEntity entityToUpdate);
7      void DeleteById(object id);
8      void Delete(TEntity entityToDelete);
9      TEntity GetByID(object id);
10 }

```

### 7.1.3 Unit of Work

As described in the architecture (see 6.4.1 Unit of Work Pattern) the Unit of Work (UoW) pattern is used so that within the application a single database context is instantiated and no problems occur with  $n:m$  table relationships. In addition, the UoW pattern complements the repository pattern and supports the abstraction of the individual layers and leads to looser coupling of the individual components. The implementation is based on the instructions “Implementing the Repository and Unit of Work Patterns in an ASP.NET MVC Application” from Microsoft. [72]

A database context is created once within the UoW class. When the repositories are called from the UoW, the individual repositories are then instantiated with the created database context as a parameter, if they have not yet been created.

Normally, the `SaveChangesAsync` and the `Dispose` methods would be implemented in the individual repositories where the database context would live. However, since the database context lives in the UoW as singleton instance, these methods are implemented in this class. This ensures that when these methods are called, all associated changes to the data are properly coordinated via the database context. If the repositories had each been implemented with their own database context, this could have caused problems when saving dependent objects.

Listing 2.3: Unit of Work

---

```

1 public class UnitOfWork : IDisposable, IUnitOfWork
2 {
3     private ReadinizerDbContext context = new ReadinizerDbContext();
4     private GenericRepository<ADDomain> adDomainRepository;
5     private GenericRepository<OrganizationalUnit> OrganizationalUnitRepository;
6     private GenericRepository<Computer> computerRepository;
7     private GenericRepository<Site> siteRepository;
8
9     public GenericRepository<ADDomain> ADDomainRepository
10    {
11        get
12        {
13            if (this.adDomainRepository == null)
14            {
15                this.adDomainRepository = new GenericRepository<ADDomain>(context);
16            }
17
18            return adDomainRepository;
19        }
20    }
21    public GenericRepository<OrganizationalUnit> OrganizationalUnitRepository { ... }
22    public GenericRepository<Computer> ComputerRepository { ... }
23    public GenericRepository<Site> SiteRepository { ... }
24    public Task SaveChangesAsync() { ... };
25    public void Dispose(bool disposing) { ... };
26    public void Dispose() { ... };
27 }

```

---

## 7.2 Frontend Implementation (Presentaion Layer)

The frontend is designed so that there is a main view (called `ApplicationView`). This `ApplicationView` is used to handle controls that are used on all views in a single view. In the following figure, the red border shows the `ApplicationView` with the menu which is always to be displayed. The `snackbar`, which is used to display errors or other current information to the user, is also integrated in this view. The green border shows the container which contains the other views with the actual functionality of the application.





Figure 2.22: ApplicationView

### 7.2.1 ApplicationView(Model)

The ApplicationViewModel is the associated class that maps the logic for this view (see 6.4.1 Model View ViewModel Pattern). This viewmodel mainly contains the logic for the handling of view changes. The approach of viewmodel first was chosen, in which the views are bound to the viewmodels and the corresponding views are created when the respective viewmodels are instantiated. Therefore, all viewmodels which are displayed in the container of the ApplicationView (green border) have to be fetched from the dependency injection in the constructor (see Listing 2.4 ApplicationViewModel - Constructor) and stored in this class.

Listing 2.4: ApplicationViewModel - Constructor

```
1 public ApplicationViewModel(StartupViewModel startUpViewModel,  
   TreeStructureResultViewModel treeStructureResultViewModel, ISnackbarMessageQueue  
   snackbarMessageQueue)  
2 {  
3     this.startUpViewModel = startUpViewModel;  
4     this.treeStructureResultViewModel = treeStructureResultViewModel;  
5  
6     this.SnackbarMessageQueue = snackbarMessageQueue;  
7  
8     ShowStartupView();  
9     Messenger.Default.Register<ChangeView>(this, ChangeView);  
10    Messenger.Default.Register<SnackbarMessage>(this, OnShowMessage);  
11 }
```

Furthermore, in this viewmodel the class **ChangeView** (see Listing 2.5 ChangeView Class) with the method **ChangeView** (see Listing 2.6 ApplicationViewModel - ChangeView Method) is called via a messenger (which is already given by the framework MVVM Light). registered. This allows you to switch from any viewmodel to a new viewmodel/view with the following command:

```
Messenger.Default.Send(new ChangeView(typeof(<<<Any ViewModel>>>)));
```

Listing 2.5: ChangeView Class

```
1 public class ChangeView
2 {
3     public Type ViewModelType { get; private set; }
4
5     public ChangeView(Type viewModelType)
6     {
7         ViewModelType = viewModelType;
8     }
9 }
```

Listing 2.6: ApplicationViewModel - ChangeView Method

```
1 private void ChangeView(ChangeView message)
2 {
3     if (message.ViewModelType == typeof(StartupViewModel))
4     {
5         ShowStartupView();
6     }
7     else if (message.ViewModelType == typeof(TreeStructureResultViewModel))
8     {
9         ShowTreeStructureResultView();
10    }
11 }
```

The following listing 2.7 View - ViewModel Binding shows how the views are bound to the viewmodels. The **DataContext** of the **UserControls** is defined as design time creatable and bound to the corresponding viewmodel.

Listing 2.7: View - ViewModel Binding

```
1 <UserControl x:Class="Readinizer.Frontend.Views.StartupView"
2     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
3     xmlns:viewModels="clr-namespace:Readinizer.Frontend.ViewModels"
4     ...
5     d:DataContext="{d:DesignInstance viewModels:StartupViewModel,
6         IsDesignTimeCreatable=True}">
```

### 7.3 UC-R1: Discovering all Organizational units and their members

The goal of this use case is to collect all necessary information of a forest and its domains, subdomains, treedomains and sites to successfully handle the subsequent use cases.

#### 7.3.1 Logic flow

##### Discover domain

- The user provides domain name from which he wants to check the readiness
  - If the user does not provide any domain name, the forest root domain will be the start domain
- The user can decide to discover all subdomains and treedomains too
- The user has the ability to check the fleet if Sysmon is installed
- The user can click the “Analyze Readiness” button

##### Collect domain data

- The application builds up connection to provided domain and its Active Directory
  - If the connection fails, the user will receive an error message
- The application recursively collects all the subdomains and treedomains and saves them into the local database
- The application collects all sites
- In each found domain, the application collects all organizational units and recursively all its sub-organizational-units and saves them into the local database
- In each found organizational unit the application collects all computer members
- For each found organizational unit the application collects one RSoP
- The collected RSoPs are analyzed against the recommendation from the benchmark (see 3.6 Overall conclusion)

### 7.3.2 Implementation - ADDomainService

The `ADDomainService` is responsible for the search of all domains, subdomains and treedomains located in the Active Directory forest. The search for these domains is solved recursively. If no domain name is provided through the user, the root domain of the forest is searched (Listing 2.8 - line 10) by calling `GetCurrentForest` [73] which brings the property `RootDomain` with it. Otherwise the `startDomain` will be the one provided by the user (Listing 2.8 - line 14). If the user also wants the sub- and treedomains to be searched, the methods `AddAllTreeDomains(startDomain, treeDomainsWithChildren, unavailableDomains)` and `AddAllChildDomains(startDomain, domains, unavailableDomains)` are called (Listing 2.8 - line 19 - 20). Both methods are called including the `startDomain`, a list for either the treedomains or subdomains and a list of unavailable domains. The list of unavailable domains will be used to display all unreachable domains.

Listing 2.8: `ADDomainService` - `SearchDomains()` Part 1

```

1  var domains = new List<AD.Domain>();
2  var treeDomainsWithChildren = new List<AD.Domain>();
3  var unavailableDomains = new List<string>();
4
5  try
6  {
7      AD.Domain startDomain;
8      if (string.IsNullOrEmpty(domainName))
9      {
10         startDomain = Forest.GetCurrentForest().RootDomain;
11     }
12     else
13     {
14         startDomain = AD.Domain.GetDomain(new
15             DirectoryContext(DirectoryContextType.Domain, domainName));
16     }
17     if (subdomainsChecked)
18     {
19         AddAllTreeDomains(startDomain, treeDomainsWithChildren, unavailableDomains);
20         AddAllChildDomains(startDomain, domains, unavailableDomains);
21     }
22     else
23     {
24         domains.Add(startDomain);
25     }
26 }
27 catch (Exceptions e)
28 {
29     ...
30 }
31 ...

```

In order to find out which domain is a treedomain, all trust relationships are additionally retrieved from the root domain. After that all these trust relationships are iterated to find the treedomains. If a treedomain is found, the corresponding domain object is fetched by `GetDomain()` [74] and added to a list of treedomains (2.9 - line 13 - 14). If the domain could not be found, it will be added to the list of unavailable domains (2.9 - line 18). After all treedomains are found, all subdomains will be searched for these treedomains (2.9 - line 23 - 26).

Listing 2.9: ADDomainService - AddAllTreeDomains

```
1 private static void AddAllTreeDomains(AD.Domain startDomain, List<AD.Domain>
   treeDomainsWithChildren, List<string> unavailableDomains)
2 {
3     var domainTrusts = startDomain.GetAllTrustRelationships();
4     List<AD.Domain> treeDomains = new List<AD.Domain>();
5
6     foreach (TrustRelationshipInformation domainTrust in domainTrusts)
7     {
8         if (domainTrust.TrustType.Equals(AD.TrustType.TreeRoot))
9         {
10             try
11             {
12                 var treeDomain = AD.Domain.GetDomain(new
13                     DirectoryContext(DirectoryContextType.Domain, domainTrust.TargetName));
14                 treeDomains.Add(treeDomain);
15             }
16             catch
17             {
18                 unavailableDomains.Add(domainTrust.TargetName);
19             }
20         }
21
22         foreach (var treeDomain in treeDomains)
23         {
24             AddAllChildDomains(treeDomain, treeDomainsWithChildren, unavailableDomains);
25         }
26     }
```

The recursive method `AddAllChildDomains()` adds the given domain to the list of all domains. Subsequently, the recursive method with the respective child is called by its children, i.e. subdomains. This is done until the array of children is empty and the for-loop is no longer processed. If the corresponding subdomain could not be called, it will be added to the list of unavailable domains.

Listing 2.10: ADDomainService - AddAllChildDomains

```

1 private static void AddAllChildDomains(AD.Domain root, List<AD.Domain> domains)
2 {
3     domains.Add(root);
4
5     for (var i = 0; i < root.Children.Count; ++i)
6     {
7         try
8         {
9             var subDomain = AD.Domain.GetDomain(new
10                 DirectoryContext(DirectoryContextType.Domain, root.Children[i].Name));
11             AddAllChildDomains(subDomain, domains, unavailableDomains);
12         }
13         catch
14         {
15             unavailableDomains.Add(root.Children[i].Name);
16         }
17     }
18 }

```

Back in the `SearchAllDomains()` method, the lists with treedomains, subdomains and unavailable domains contain now all domains which could be gathered. Since not all information is needed by the system domain objects, they are created with the method `MapToDomainModel()` into corresponding domain objects, which are customized for the Readinizer. Via the `ADDomainRepository`, the generated domain objects are then written into the database.

Listing 2.11: ADDomainService - SearchAllDomains() Part 2

```

1 ...
2 var models = MapToDomainModel(domains, treeDomainsWithChildren);
3 unitOfWork.ADDomainRepository.AddRange(models);
4
5 var modelsUnavailable = unavailableDomains.Select(x => new ADDomain { Name = x,
6     IsAvailable = false }).ToList();
7 unitOfWork.ADDomainRepository.AddRange(modelsUnavailable);
8
9 await unitOfWork.SaveChangesAsync();

```

## Occurred problems - Class Domain

### Class Domain

Due to the fact that in the .NET Framework already a class “Domain” exists, the domain model had to be slightly modified. The domain class for the Readinizer application is therefore called `ADDomain` in order to prevent naming conflicts.

### 7.3.3 Implementation - SiteService

The `SiteService` does not differ much from the `ADDomainService` and will therefore not be described in detail. In fact, the `SiteService` is much simpler because it just gets all forest sites, maps them to the domain models and finally saves them in the database.

### 7.3.4 Implementation - OrganizationlUnitService

The **OrganizationlUnitService** is looking for all Organizational Units in each domain. Because organizational units can have child organizational units, the search is set up recursively. First, all domains found are loaded from the database and stored in a list. Each domain is then examined individually for its organizational units. Before this happens, it is checked if the domain is available (Listing 2.12 - line 5). Subsequently, all objects of the category organizational units (Listing 2.12 - line 9), without their child organizational units (Listing 2.12 - line 10), are searched in the domain, specified by the Lightweight Directory Access Protocol path (Listing 2.12 - line 6), and are stored temporary in a list. Important information of each organizational unit is read out and saved (Listing 2.12 - line 16 -19). To find all child organizational units and their child organizational units the recursive method 2.13 **GetChildOUs** is called. After all child organizational units have been found, the organizational units are stored in the database.

Listing 2.12: OrganizationlUnitService - GetAllOrganizationalUnits

```

1  List<ADDomain> allDomains = await unitOfWork.ADDomainRepository.GetAllEntities();
2
3  foreach (ADDomain domain in allDomains)
4  {
5      if (domain.IsAvailable)
6      {
7          DirectoryEntry entry = new DirectoryEntry("LDAP://" + domain.Name);
8          DirectorySearcher searcher = new DirectorySearcher(entry);
9          searcher.Filter = "(objectCategory=organizationalUnit)";
10         searcher.SearchScope = SearchScope.OneLevel;
11         var foundOUs = new List<OrganizationalUnit>();
12
13         foreach (SearchResult searchResult in searcher.FindAll())
14         {
15             OrganizationalUnit foundOU = new OrganizationalUnit();
16             foundOU.Name = searchResult.Name;
17             foundOU.LdapPath = searchResult.Path;
18             foundOU.ADDomainRefId = domain.ADDomainId;
19             foundOU.SubOrganizationalUnits = GetChildOUs(foundOU.LdapPath, foundOU);
20
21             foundOUs.Add(foundOU);
22         }
23         unitOfWork.OrganizationalUnitRepository.AddRange(foundOUs);
24     }
25 }
26 await unitOfWork.SaveChangesAsync();
27 }
```

When calling the function `GetChildOUs`, the LDAP-path as well as the parent organizational unit must be passed. As when searching in the domain, all organizational units in this organizational unit are searched using the LDAP path and are temporarily stored (Listing 2.13 - line 13 - 15). The `GetChildOUs` function is then called for each of the found child organizational units (Listing 2.13 - line 16). The list of child organizational units will be returned to the parent and stored in the database (Listing 2.13 - line 20 & 23).

Listing 2.13: OrganizationalUnitService - GetChildOUs

---

```

1  public List<OrganizationalUnit> GetChildOUs(string ldapPath, OrganizationalUnit
   parentOU)
2  {
3      List<OrganizationalUnit> childOUs = new List<OrganizationalUnit>();
4
5      DirectoryEntry childEntry = new DirectoryEntry(ldapPath);
6      DirectorySearcher childSearcher = new DirectorySearcher(childEntry);
7      childSearcher.Filter = "(objectCategory=organizationalUnit)";
8      childSearcher.SearchScope = SearchScope.OneLevel;
9
10     foreach (SearchResult childResult in childSearcher.FindAll())
11     {
12         OrganizationalUnit childOU = new OrganizationalUnit();
13         childOU.Name = childResult.Name;
14         childOU.LdapPath = childResult.Path;
15         childOU.ADDomainRefId = parentOU.ADDomainRefId;
16         childOU.SubOrganizationalUnits = GetChildOUs(childOU.LdapPath, childOU);
17
18         childOUs.Add(childOU);
19
20         unitOfWork.OrganizationalUnitRepository.Add(childOU);
21     }
22
23     return childOUs;
24 }

```

---

### 7.3.5 Implementation - ComputerService

The computers are searched and stored in a very similar way to the organizational units in the domain (`GetAllOrganizationalUnits`). For this reason, it is not explained in more detail.



## 7.4 UC-R2: Collecting Resultant Set of Policies of a fleet

The goal of this use case is to collect the Resultant Set of Policies for one member of each found organizational unit.

### 7.4.1 Logic flow

#### Check reachability of computer

- All organizational units are loaded from the database
- For each organizational units the computer members are loaded from the database
- One by one, the computers are tried to be pinged, until one ping is successful
- The computer will be marked as reachable, the organizational unit is set to have a reachable member

#### Collect Resultant Set of Policy

- The RSoP for this computer is read out via remote access
- The gather data is saved as a temporary XML-file for later use

#### (Optional) Ckeck if Sysmon is running

- If this feature is activated, each available computer is checked if the Sysmon service is running

### 7.4.2 Implementation - PingService

The reachability of a computer is checked by its response on a ping request. A ping is not the perfect solution because it requires certain firewall settings which may not be set, but it is the most performant variant. A more detailed explanation can be found in the Occured Problems section.

To ping the target computer the classes, Ping Class [75] and PingReply Class [76] are used as well as the IPStatus Enum [77].

Listing 2.14: Ping target computer

```
1 using System.Net.NetworkInformation;
2
3 bool pingable = false;
4 Ping pinger = new Ping();
5
6 PingReply reply = pinger.Send(ipAddress, 200);
7 pingable = reply.Status == IPStatus.Success;
```

The `PingReply` is used to catch the status of the reply received from the ping. The `Ping.Send` method sends an ICMP echo message and receives a echo reply message from the target computer. The target can be specified with the first parameter, either by the computers IP-address or its hostname. The second parameter defines the maximum number of milliseconds to wait for the reply message. The default time-out is 4000 milliseconds (4 seconds) [78].

The ping reply status is compared against the enum `IPStatus Success`; if these match, the device can be contacted successfully. The boolean `pingable` is set to true.

## Occurred problems

### Address Resolution Protocol-Request Timeout

Ping should not be used to check the reachability of a computer, as it is recommended to block pings in the network. So we first tried without checking the reachability of a computer and immediately started requesting an RSoP. This worked, but only as long as a computer was active. If the computer was not turned on or could not be contacted, it took about 15 seconds to query a single computer. By sniffing the network, we also found out why. Before the connection is established, an Address Resolution Protocol (ARP) request is sent. Four times such a ARP-request is sent and each time about 4 seconds waited for a response before the connection was aborted. The same problem occurred when checking via Transmission Control Protocol (TCP) connection. For this reason, the ping variant was finally retained.

## 7.5 Implementation - RSoPService

To collect the Resultant Set of Policy from the computers, the GPRsop class [79] was used. To use this class the Microsoft.GroupPolicy.Management Dynamic Link Library (DLL) must be available. This library can be found when installing the “Remote Server Administration Tool” (RSAT) [80] and must then be added as a reference.

Listing 2.15: GPRsop use

```
1 using Microsoft.GroupPolicy;  
2  
3 GPRsop rsop = new GPRsop(RsopMode.Logging, "");  
4     rsop.LoggingMode = LoggingMode.Computer;  
5     rsop.LoggingComputer = "computer";  
6     rsop.LoggingUser = "user";  
7     rsop.CreateQueryResults();  
8     rsop.GenerateReportToFile(ReportType.Xml, "C:\\path\\rsop.xml");
```

The created GPRsop will be in the RsopMode Logging, this means it will gather the actual Group Policies and calculate the effective settings. The second parameter is the “WMI-Namespace” and as described in the constructor [81] this parameter can be an empty string.

The LoggingMode [82] is a enum and can be set to three different values. The value used is Computer, it generates a report based on the Group Policy settings of the specified computer, the user configuration section is empty.

The LoggingComputer [83] can be specified with this parameter, the fully qualified domain name of the computer is used. The LoginUser[84] is to be specified in domain\domain format. If these parameters are not set, the current computer respectively the current user is used.

The CreateQueryResults [85] requires the LoggingMode, LoggingComputer and LoginUser to be set. It connects to the specified computer and collects the RSoP and delivers it back to the executing computer.

The GenerateReportToFile [86] generates a report in a file from the results the CreateQueryResults delivered. The first parameter defines the type of the report, this can either be as a Extensible Markup Language file (XML) or as a Hypertext Markup Language file (HTML). The second parameter specifies the path where these files are saved.

## Occurred problems

### COM Exception: Class not found

After the RSAT was installed and the library added as a reference to the project, the application threw this COM Exception:

```
Class not found (Exception from HRESULT: 0x80040154 (REGDB_E_CLASSNOTREG))
```

After carrying out internet research, it turned out that this library is only available in 64-bit systems. After changing the CPU settings in Visual Studio the application run successful.

### GPRsop.LoggingComputer name

The description of the GPRsop.LoggingComputer [83] mentions that the LoggingComputer can be specified in three different formats:

- Computername, this will logically only work in the same domain
- Domain\Computername
- Fully qualified domain name of the computer

In a first step the format Domain\Computername was used. This worked well for computers in the same domain, but when connecting to a computer in another domain, this error occurred:

```
The RPC server is unavailable. 0x800706BA
```

Searching the internet for this error brought up various explanations and solutions such as wrong firewall settings or insufficient privileges. However, none of these solutions solved the problem.

After the communication between the client SysAdmWS01 (in the domain readinizer.ch) and the domain controller subreadinizerDC (in the domain sub.readinizer.ch) was sniffed with Wireshark, these packets were found:

DNS	89	Standard query	0xde08	A subreadinizerDC.readinizer.ch
MDNS	81	Standard query	0x0000	A subreadinizerDC.local, "QM" question
MDNS	101	Standard query	0x0000	A subreadinizerDC.local, "QM" question
DNS	162	Standard query response	0xde08	No such name A subreadinizerDC.readinizer.ch SOA readinizerdc.readinizer.ch

Figure 2.23: Sniffed communication

The computer was searched in the wrong domain. Why this is the case is not known. The case was reported to Microsoft through their feedback platform.

After changing the format to the fully qualified domain name of the computer, the error did not occur again.

### 7.5.1 Implementation - SysmonService

After it has been determined if a computer is running, it is checked if the Sysmon service is installed and has the status “Running”. For the reason that Sysmon can be “hidden”, the service then runs under a different name, the service name of Sysmon can be specified. First a connection to the target computer is established (Listing 2.16 - line 3 - 5), then all Windows services are loaded (Listing 2.16 - line 6 - 7). Each service is then checked if the service name equals the provided name (default value is “Sysmon”). If this is the case, the service stat is checked if it is set to “Running” (Listing 2.16 - line 11). If both cases apply, “True” will be returned, otherwise the method returns “False”, Sysmon is not running.

Listing 2.16: Check if Sysmon service is running

```
1  public bool isSysmonRunning(string serviceName, string user, string computerName,  
2      string domain)  
3  {  
4      ConnectionOptions op = new ConnectionOptions();  
5      ManagementScope scope = new ManagementScope(@"\\" + computerName + "." + domain +  
6          "\\root\\cimv2", op);  
7      scope.Connect();  
8      ManagementPath path = new ManagementPath("Win32_Service");  
9      ManagementClass services = new ManagementClass(scope, path, null);  
10  
11     foreach (var service in services.GetInstances())  
12     {  
13         if (service.GetPropertyValue("Name").ToString().Equals(serviceName) &&  
14             service.GetPropertyValue("State").ToString().ToLower().Equals("running"))  
15         {  
16             return true;  
17         }  
18     }  
19     return false;  
20 }
```

## 7.6 UC-R3: Analysing the collected data

### 7.6.1 Logic flow

#### Analyze all Resultant Set of Policies

- Get all XML-Files (RSoP) from the path where they are buffered
- Convert the XML-Files to JavaScript Object Notation (JSON) for easier handling and to remove the namespaces
- Get all GPOs from each RSoP
- Analyze the audit settings against the recommended ones
- Create a RSoP object and save it to the database for each RSoP

### 7.6.2 Implementation - AnalysisService

#### Preamble

In order to describe the **AnalysisService** accurately, the received RSoP XML-File must be discussed in advance because the RSoP-XML has a not such simple structure. The following listing will only deal with the relevant tags of the XML, because otherwise it would go beyond the scope and fill probably over 20 pages with XML code. Only the structure is described. Note that the real RSoP-XML gives a lot more information than described here.

At the beginning of the XML, some general information about where this RSoP comes from is stated:

Listing 2.17: RSoP-XML - General Information

```

1  <?xml version="1.0" encoding="utf-16"?>
2  <Rsop>
3    <ComputerResults>
4      <Name>Systemname</Name>
5      <Domain>Domainname</Domain>
6      <SOM>Scope Of Management</SOM>
7      <Site>Sitename</Site>
8      <SearchedSOM>
9        <Path>SOM Path</Path>
10       <Type>Type of SOM (Domain, Site, OU)</Type>
11       <Order>Order in which OUs are applied to this System (Treestructure OUs)</Order>
12       <BlocksInheritance>If GPO inheritance is blocked</BlocksInheritance>
13       <Blocked>If GPO is blocked</Blocked>
14       <Reason>Normal/Loopback GPO Processing</Reason>
15     </SearchedSOM>
16     ...
17     More SearchedSOM
18     ...

```

The next relevant part of the XML are the GPOs. Especially the part in the tag `<Link>` is not trivial to understand. However, not all information from this tag is necessary for the application “Readinizer” because all applied settings are stated at a later stage in the XML with its respective GPO identifier. To know from which GPO - throughout all GPOs - each setting came from, the respectively identifier can be resolved. Therefore, this information brings no benefits for the “Readinizer” but can bring a lot of understanding how GPOs are processed and applied.

Listing 2.18: RSoP-XML - GPO

```

1 <GPO>
2   <Name>GPO Name</Name>
3   <Path>
4     <Identifier>GPO ID(i.e. {38BD7D39-23EB-469E-9B93-1D2E5645C43E})</Identifier>
5     <Domain>Domainname</Domain>
6   </Path>
7   More GPO Information
8   ...
9   <Link>
10    <SOMPath>
11      SOM Path of GPO
12      (i.e. readinizer.ch/Group Policy OU/IT/System Administrators/Recommended GPO)
13    </SOMPath>
14    <SOMOrder>1</SOMOrder>
15    <AppliedOrder>3</AppliedOrder>
16    <LinkOrder>4</LinkOrder>
17    <Enabled>Whether GPO is enabled or not</Enabled>
18    <NoOverride>Whether GPO overrides other GPOs or not</NoOverride>
19  </Link>
20  <SecurityFilter>
21    Who has to apply this GPO on the system
22    (i.e. NT AUTHORITY\Authenticated Users)
23  </SecurityFilter>
24  <ExtensionName>
25    Which Extensions (Settings) will be applied
26    (i.e. Audit Policy Configuration)
27  </ExtensionName>
28 </GPO>
29 ExtensionData (i.e. Scheduled Tasks)
30 ...

```

For a better understanding how the GPOs are linked and the way they are processed, a detailed explanation follows:

**SOMOrder** Scope of management (SOM) order of respectively SOM containers (Site, Domain, OU)  
- in case of multiple GPOs in such a container, this number will increase so on every container-  
“layer” a hierarchical order exist

**AppliedOrder** Overall order of GPOs which effect the system

**LinkOrder** Link order in relation to all other GPOs in the hierarchy of the according system in which the GPO is applied

**Note:** The GPO with the highest precedence will have the biggest effect. The later a GPO is applied the higher is its precedence and it might override settings from a GPO with a smaller precedence.

An example of how the GPOs are linked and processed will now be discussed. SOMOrder, AppliedOrder and LinkOrder can be seen in from of a triple (**SOMOrder ; AppliedOrder ; LinkOrder**). The order how a GPO is applied seems to be inverse lexicographical ordered.

**Inverse Lexicographical Order**<sup>6</sup> of two n-tuples is recursively defined as:

$$(x_n, x_{n-1}, \dots, x_2, x_1) > (y_n, y_{n-1}, \dots, y_2, y_1)$$

Example:

$$(1;4;5) (1;3;4) (1;1;1) (1;0;3) (2;2;2) (1;5;6) (2;6;7)$$

$$i=1 (2;2;2) (2;6;7) (1;4;5) (1;3;4) (1;1;1) (1;0;3) (1;5;6)$$

$$i=2 (2;6;7) (1;5;6) (1;4;5) (1;3;4) (2;2;2) (1;1;1) (1;0;3)$$

$$i=3 (2;6;7) (1;5;6) (1;4;5) (1;3;4) (1;0;3) (2;2;2) (1;1;1)$$

Lets say we do have a computer in the OU “A.A.A” and the “Default Domain Policy” has no settings for computers and users, then the triples are as follows:

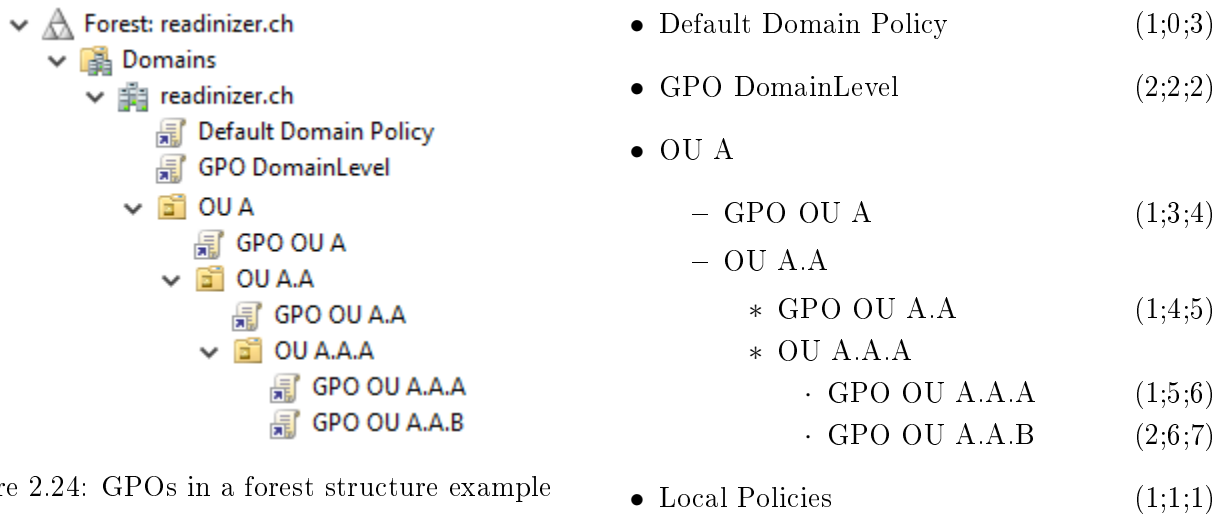


Figure 2.24: GPOs in a forest structure example

If we look at the example sequence from right to left, the correct order of the GPO precedence’s results:

### Effectively applied order:

$$(1;1;1) (2;2;2) (1;0;3) (1;3;4) (1;4;5) (1;5;6) (2;6;7)$$

Local Policies > GPO DomainLevel > Default Domain Policy > GPO OU A > GPO OU A.A > GPO OU A.A.A > GPO OU A.A.B

As you can see, the policy with the triple (2;6;7) - GPO OU A.A.B will be applied at the very end of the chain and has therefore the highest precedence. If previous GPO settings vary with the GPO at the ver end of the chain, they will be overwritten. This example does not include enforced GPOs. Settings from enforced GPOs will be applied in any case.

<sup>6</sup>Common Lexicographical Order:  $(x_1, x_2, \dots, x_{n-1}, x_n) < (y_1, y_2, \dots, y_{n-1}, y_n)$

The AuditSettings<sup>7</sup> are one of the most important part regarding to the overall conclusion of the benchmark (see section 3.6 Overall conclusion).

**XML Namespaces & Transformation** From here on namespaces are introduced in the RSoP on ExtensionData level. These namespaces start with a **q** followed by a more or less arbitrary number - i.e. **q2:**. This number is determined according to which settings occur in the RSoP. In the following case, **ScheduledTasksSettings** are applied to the system, which gives the AuditSettings the namespace **q2:**. If no **ScheduledTasksSettings** were applied, the namespace of the AuditSettings would be **q1:**. Since these namespaces are more or less arbitrary, it was decided to remove the namespaces of the RSoP-XML before the analysis and at the same time to transfer the RSoP-XML into a JSON. This has the effect that the search for the settings is simplified a lot. Another reason for removing the namespaces is discussed when addressing the SecurityOptions.

**GPO** Each AuditSetting contains the GPO identifier and domain from which the setting was applied

**PolicyTarget** Which “Advanced Audit Policy”-group is concerned

**SubcategoryName** Which “Advanced Audit Policy”-setting is concerned

**SubcategoryGuid** Which “Advanced Audit Policy”-GUID is concerned

**SettingValue** Which “Advanced Audit Policy”-setting is applied (0: Not Configured/No Auditing, 1: Success, 2: Failure, 3: Success and Failure - see 3.6.1 GPO Settings Readinizer)

Listing 2.19: RSoP-XML - AuditSettings

```
1 <ExtensionData>
2   <Extension xsi:type="q1:ScheduledTasksSettings">
3   </Extension>
4 </ExtensionData>
5 <ExtensionData>
6   <Extension xsi:type="q2:AuditSettings">
7     <q2:AuditSetting>
8       <GPO>
9         <Identifier>{38BD7D39-23EB-469E-9B93-1D2E5645C43E}</Identifier>
10        <Domain>readinizer.ch</Domain>
11      </GPO>
12      <q2:PolicyTarget>System</q2:PolicyTarget>
13      <q2:SubcategoryName>Audit Security System Extension</q2:SubcategoryName>
14      <q2:SubcategoryGuid>{0cce9211-69ae-11d9-bed3-505054503030}</q2:SubcategoryGuid>
15      <q2:SettingValue>1</q2:SettingValue>
16    </q2:AuditSetting>
17    ...
18  </Extension>
19 <Name>Advanced Audit Configuration</Name>
20 </ExtensionData>
21 ...
22 ExtensionData (i.e. ServiceSettings)
23 ...
```

<sup>7</sup>Can be found in the GPO settings under: Computer Configuration > Windows Settings > Security Settings > Advanced Audit Policy Configuration > System Audit Policies



With the SecurityOptions<sup>8</sup> the GPO setting “Force audit policy subcategory” (see table 2.18 Force Audit Policy Subcategory Settings) settings can be detected.

As you can see here, even within the same namespaces there are different settings (**Account** and **SecurityOptions**). This is another reason why the namespaces are removed.

**GPO** Each SecurityOption contains the GPO identifier and domain from which the setting was applied

**KeyName** Which registry key is affected

**SettingNumber**

**Display - Name** Text which is displayed in the GUI

**Display - DisplayBoolean** Whether the setting is enabled or not

Listing 2.20: RSoP-XML - SecurityOptions

```
1 <ExtensionData>
2   <Extension xsi:type="q4:SecuritySettings">
3     <q4:Account>
4       ...
5     </q4:Account>
6     ...
7     <q4:SecurityOptions>
8       <GPO>
9         <Identifier>{38BD7D39-23EB-469E-9B93-1D2E5645C43E}</Identifier>
10        <Domain>readinizer.ch</Domain>
11      </GPO>
12      <q4:KeyName>
13        MACHINE\System\CurrentControlSet\Control\Lsa\SCENoApplyLegacyAuditPolicy
14      </q4:KeyName>
15      <q4:Display>
16        <q4:Name>Audit: Force audit policy subcategory settings (Windows Vista or
17          later) to override audit policy category settings</q4:Name>
18        <q4:Units />
19        <q4:DisplayBoolean>true</q4:DisplayBoolean>
20      </q4:Display>
21    </q4:SecurityOptions>
22    <q4:Blocked>>false</q4:Blocked>
23  </Extension>
24  <Name>Security</Name>
25</ExtensionData>
26...
27ExtensionData (i.e. Security by other authority, Public Key, Windows Firewall)
28...
```

<sup>8</sup>Can be found in the GPO settings under: Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options

Finally there are the RegistrySettings where you can also find the Policies. Both concern the settings which can be found in the GPO settings as follows:

Computer Configuration > Policies > Administrative Templates > ...

These applies to the settings recommended in the tables 2.28 Administrative Template System, 2.29 Administrative Template Windows Components Windows PowerShell and 2.30 Add Registry Key for LSA Protection [29].

**GPO** Each Policy/RegistrySetting contains the GPO identifier and domain from which the setting was applied

**Policy - Name** Which setting is affected

**Policy - State** Whether the setting is enabled or not

**Policy - Category** Where the setting can be found in the GPO settings

**RegistrySetting - KeyPath** Which registry key is affected

**RegistrySetting - Value - Name** Which setting is affected

**RegistrySetting - Value - Number** How the setting is configured

Listing 2.21: RSoP-XML - Policy and RegistrySettings

```

1      <ExtensionData>
2          <Extension xsi:type="q8:RegistrySettings">
3              <q8:Policy>
4                  <GP0>
5                      <Identifier>{38BD7D39-23EB-469E-9B93-1D2E5645C43E}</Identifier>
6                      <Domain>readinizer.ch</Domain>
7                  </GP0>
8                  <q8:Name>Include command line in process creation events</q8:Name>
9                  <q8:State>Enabled</q8:State>
10                 <q8:Category>System/Audit Process Creation</q8:Category>
11             </q8:Policy>
12             ...
13             <q8:RegistrySetting>
14                 <GP0>
15                     <Identifier>{38BD7D39-23EB-469E-9B93-1D2E5645C43E}</Identifier>
16                     <Domain>readinizer.ch</Domain>
17                 </GP0>
18                 <q8:KeyPath>SYSTEM\CurrentControlSet\Control\Lsa</q8:KeyPath>
19                 <q8:Value>
20                     <q8:Name>RunAsPPL</q8:Name>
21                     <q8:Number>1</q8:Number>
22                 </q8:Value>
23             </q8:RegistrySetting>
24             ...
25         </Extension>
26         <Name>Registry</Name>
27     </ExtensionData>
28     ....
29 </ComputerResults>
30 </Rsop>

```

## Implementation

Now that we have a good understanding of the RSoP and its structure, we can go into the actual implementation of the AnalysisService.

The AnalysisService is needed to analyze the RSoP received from the clients during the initial readiness check and for any missing RSoPs that can be imported. In order for the workflow of the AnalysisService to be reusable for both applications, a distinction is made at the beginning by the given parameter `importPath`. If this string `IsNullOrEmpty` is an initial analysis and the RSoP are fetched from the directory where they were stored by the `RSoPService`. In the other case it is a manual import of RSoP-XMLs and therefore the user has to specify the path to them. The analysis (`AnalyzeEachXml` - Listing 2.22 - line 8) is then performed for each XML.

Listing 2.22: AnalysisService - Analyze()

```
1 public async Task<List<Rsop>> Analyze(string importPath)
2 {
3     var rsopPath = string.IsNullOrEmpty(importPath) ?
4         ConfigurationManager.AppSettings["ReceivedRSoP"] : importPath;
5     var directoryInfo = new DirectoryInfo(rsopPath);
6     var rsopXml = directoryInfo.GetFiles("*.xml");
7     var rsops = new List<Rsop>();
8
9     AnalyzeEachXml(rsopXml, rsops);
10    unitOfWork.Repository.AddRange(rsops);
11
12    await unitOfWork.SaveChangesAsync();
13    return rsops;
14 }
```

The method `AnalyzeEachXml()` loops over the array of RSoP-XMLs and performs the following steps for each:

- Load XML
- Transform XML to JSON (see XML Namespaces & Transformation)
- Analyzes all settings (`AuditSettings`, `Policies`, `SecurityOptions`, `RegistrySettings`)
  - Detailed description follows on the next page
- Creates a corresponding RSoP object which is stored in the database
  - The RSoP contains a list of each setting group `AuditSettings`, `SecurityOptions`, `Policies`, `RegistrySettings`
  - The RSoP contains additional information about which domain, OU and site is affected by getting this information from the general information of the RSoP-XML (see Listing 2.17 RSoP-XML - General Information)

Because the individual settings vary slightly, an appropriate method had to be implemented for each group of settings (**AuditSettings**, **SecurityOptions**, **Policies**, **RegistrySettings**) to extract the current settings. However, only the **AuditSettings** method is discussed here, since the methods only differ in the properties.

First of all, the recommended settings are fetched into a list of **recommendedAuditSettings** from the according JSON-File (Listing 2.23 - line 3).

Listing 2.23: AnalysisService - AnalyseAuditSettings() Part 1

```
1 private static List<AuditSetting> AnalyzeAuditSettings(JObject rsop)
2 {
3     var recommendedAuditSettings =
        GetRecommendedSettings(ConfigurationManager.AppSettings["RecommendedAuditSettings"],
        new List<AuditSetting>());
```

Afterwards the ExtensionData **AuditSetting** is searched in the transformed JSON **rsop** and all existing settings are extracted into a list **auditSettings** (Listing 2.24).

Listing 2.24: AnalysisService - AnalyseAuditSettings() Part 2

```
1 var jsonAuditSettings = rsop.SelectToken("$.AuditSetting");
2 var auditSettings = new List<AuditSettingJson>();
3 GetSettings(jsonAuditSettings, auditSettings);
```

The two lists differ in their classes. The current settings from the JSON were transferred to a list of **JSONAuditSetting** and the recommended settings to a list of **AuditSetting**. The reason for this is that the class **JSONAuditSetting** is exactly adapted to the JSON and the values to be extracted. The class **AuditSetting** is designed in such a way that the current and recommended settings can be reconstructed. It also contains additional information that will be needed later on in the application.

If the current and recommended settings are present in both lists, these are joined together (Listing 2.25). This results in a list of **AuditSetting** called **presentAuditSettings**, which contains only the actual present settings in the RSoP.

Listing 2.25: AnalysisService - AnalyseAuditSettings() Part 3

```

1  var presentAuditSettings = recommendedAuditSettings.Join(auditSettings,
2      recommendedAuditSetting => recommendedAuditSetting.SubcategoryName,
3      auditSetting => auditSetting.SubcategoryName,
4      (recommendedAuditSetting, x) => recommendedAuditSetting).ToList();

```

As mentioned above, the class **AuditSetting** contains further information, which is now finally filled into the list of **recommendedAuditSettings**. It is checked which recommended settings are really present by cross-checking the **presentAuditSettings** with the **recommendedAuditSettings** (Listing 2.26 - line 3).

The property **CurrentSettingValue** is then transferred from the **presentAuditSettings** to the **recommendedAuditSettings** (Listing 2.26 - line 4 - 5). If this setting is not available, a default value is written. (Listing 2.26 - line 6) Finally, the GPO Identifier is extracted to see later which GPO made this setting (Listing 2.26 - line 8 - 11).

Listing 2.26: AnalysisService - AnalyseAuditSettings() Part 4

```

1  return recommendedAuditSettings.Select(x =>
2  {
3      x.IsPresent = presentAuditSettings.Contains(x);
4      x.CurrentSettingValue = auditSettings.Where(y =>
5          y.SubcategoryName.Equals(x.SubcategoryName))
6          .Select(z => z.CurrentSettingValue)
7          .DefaultIfEmpty(AuditSettingValue.NoAuditing)
8          .FirstOrDefault();
9      x.GpoId = auditSettings.Where(y => y.SubcategoryName.Equals(x.SubcategoryName))
10         .Select(z => z.Gpo.GpoIdentifier.Id)
11         .DefaultIfEmpty("NoGpoId")
12         .FirstOrDefault();
13     return x;
14 }).ToList();

```

## Occurred problems

### Is the setting an array or an object

Due to the fact that the RSoP XML is generated by Microsoft and it is not completely consistent, the values in method `GetSettings()` could not simply be deserialized to a list of the corresponding settings. Because if there is only one setting within the RSoP - e.g. for the `AuditSettings` - it is not available as an array of `JToken` but as a single object. This leads to problems, because in the normal case the corresponding array (`Children`<sup>9</sup>) would be iterated, the individual `AuditSettings` would be deserialized and written to a list. However, if there is a single setting, it will only be represented as a single object in the RSoP. Thus, the `Children` are not an array of audit settings, but the `JToken` itself is the setting looked for.

For this reason, the first step is to detect whether the setting from the JSON is an array or the object itself (Listing 2.27 - line 3 & 20). If deserialization does not work, a new undefined object will be created and added to the list of settings (Listing 2.27 - line 13 - 17 & 27 - 31).

Listing 2.27: `GetSettings()`

```
1  if (!(jsonSettings is null))
2  {
3      if (jsonSettings.Type is JTokenType.Array)
4      {
5          var jsonSettingsList = jsonSettings.Children().ToList();
6          foreach (var jsonSetting in jsonSettingsList)
7          {
8              try
9              {
10                 var setting = jsonSetting.ToObject<T>();
11                 settings.Add(setting);
12             }
13             catch
14             {
15                 var setting = new T();
16                 settings.Add(setting);
17             }
18         }
19     }
20     else
21     {
22         try
23         {
24             var setting = jsonSettings.ToObject<T>();
25             settings.Add(setting);
26         }
27         catch
28         {
29             var setting = new T();
30             settings.Add(setting);
31         }
32     }
33 }
```

<sup>9</sup>Listing 2.19 - line 7 - 16 is a single object, followed by possibly further `AuditSetting` objects, which leads to an array

### SingleValueArrayConverter

The previous problem exists not only with the settings to be deserialized, but also within the object itself. There is, for example, a subobject GPO, which in turn occurs in the form of a single object or as a corresponding array. Therefore a corresponding SingleValueArrayConverter was created which overrides the original method `ReadJson` (Listing 2.28) from the Newtonsoft-library. It then is defined as annotations on the subobjects in the classes of the settings (Listing 2.29 - line 9). The SingleValueArrayConverter checks the subobject (in this case `Link`) to be deserialized and returns a corresponding array with one or more subobjects in any case.

Listing 2.28: ReadJson Override

```
1 public override object ReadJson(JsonReader reader, Type objectType, object
  existingValue, JsonSerializer serializer)
2 {
3     object returnValue = new Object();
4     if (reader.TokenType == JsonToken.StartObject)
5     {
6         T instance = (T)serializer.Deserialize(reader, typeof(T));
7         returnValue = new List<T>() { instance };
8     }
9     else if (reader.TokenType == JsonToken.StartArray)
10    {
11        returnValue = serializer.Deserialize(reader, objectType);
12    }
13    return returnValue;
14 }
```

Listing 2.29: SingleValueArrayConverter in class Gpo

```
1 public class Gpo
2 {
3     public int GpoId { get; set; }
4
5     ...
6
7     [JsonProperty("Link")]
8     [JsonConverter(typeof(SingleValueArrayConverter<Link>))]
9     public List<Link> Link { get; set; }
10
11     ...
12 }
```

### 7.6.3 Implementation - RsopPotService

#### RsopPot - Groups of identical security settings

It might happen that several organizations have the same audit/security settings, although other settings vary. Due to the fact that the Readinizer only examines a certain set of audit/security settings, it makes sense to group RSoPs. In other words, all organizational units that have the same audit/security settings are grouped and sorted into “groups of identical security settings”. Within the application these groups of identical security settings are called **RsopPots**.

For the mapping of the Rsops with the same audit/security settings into a corresponding RsopPot a separate service was implemented. The service loads all existing Rsops from the database and fills them into RsopPots.

The algorithm takes the first rsop from this list and creates the first RsopPot from it (Listing 2.30 - line 4). Then the whole list of Rsops is iterated, skipping the first one, because a RsopPot has already been created from it (Listing 2.30 - line 6 - 14). The algorithm now checks each Rsop against the Rsops in the already existing RsopPots (Listing 2.30 - line 8). If there is no match (`foundPot == null`), a new RsopPot will be created.

Listing 2.30: RsopPotService - FillRsopPotList()

---

```

1  public List<RsopPot> FillRsopPotList(List<Rsop> sortedRsopsByDomain)
2  {
3      var rsopPots = new List<RsopPot>();
4      AddRsopPot(sortedRsopsByDomain.First());
5
6      foreach (var rsop in sortedRsopsByDomain.Skip(1))
7      {
8          var foundPot = RsopPotsEqual(rsopPots, rsop);
9
10         if (foundPot == null)
11         {
12             AddRsopPot(rsop);
13         }
14     }
15
16     void AddRsopPot(Rsop rsop)
17     {
18         rsopPots.Add(RsopPotFactory(rsop));
19     }
20
21     return rsopPots;
22 }

```

---



The following method checks whether the given rsop (**currentRsop**) already occurs in one of the existing RsopPots. For this purpose, all existing RsopPots are iterated and the first Rsop is taken. The first rsop can be taken, because all rsops in a RsopPot are identical.

Afterwards the **AuditSettings**, **Policies**, **RegistrySettings** as well as the **SecurityOptions** are checked with the method **SettingsEqual** (see Listing 2.32) for equality (Listing 2.31 - line 10 & 13 & 16 & 19). If one of these checks results in **false** - so not all settings are the same - the check is aborted with this RsopPot and the **currentRsop** is compared with the Rsop of the next RsopPot (Listing 2.31 - line 11 & 14 & 17 & 20).

If all settings are the same, the domain and the organizational unit will be checked for equality. Since it is difficult to recommend creating separate GPOs for each domain, the domain is checked. If the domains are the same, the **currentRsop** is compared with the rsop from the next RsopPot (Listing 2.31 - line 22 - 23). Finally it is checked whether the organizational units are equal, if this is not the case, this rsop will be appended to the list of rsops of the corresponding RsopPots. So all rsops with the same settings will be mapped to the same RsopPot (Listing 2.31 - line 27 - 28). The loop can be aborted because the rsop could be assigned to a RsopPot.

Listing 2.31: RsopPotService - RsopPotsEqual()

```

1 public RsopPot RsopPotsEqual(List<RsopPot> rsopPots, Rsop currentRsop)
2 {
3     RsopPot foundPot = null;
4
5     foreach (var pot in rsopPots)
6     {
7         var rsop = pot.Rsops.FirstOrDefault();
8         if (rsop == null) continue;
9
10        var auditSettingsEqual = SettingsEqual(rsop.AuditSettings,
11        currentRsop.AuditSettings);
12        if (!auditSettingsEqual) continue;
13
14        var policiesEqual = SettingsEqual(rsop.Policies, currentRsop.Policies);
15        if (!policiesEqual) continue;
16
17        var registrySettingsEqual = SettingsEqual(rsop.RegistrySettings,
18        currentRsop.RegistrySettings);
19        if (!registrySettingsEqual) continue;
20
21        var securityOptionsEqual = SettingsEqual(rsop.SecurityOptions,
22        currentRsop.SecurityOptions);
23        if (!securityOptionsEqual) continue;
24
25        var domainsEqual = rsop.Domain.Equals(currentRsop.Domain);
26        if (!domainsEqual) continue;
27
28        if (RsopAndRsopPotsOuEqual(currentRsop, rsop)) continue;
29
30        pot.Rsops.Add(currentRsop);
31        foundPot = pot;
32        break;
33    }
34
35    return foundPot;
36 }

```

A generic method was created to check whether the settings are the same. For this to work, the corresponding classes (**AuditSetting**, **Policy**, **RegistrySetting**, **SecurityOption**) had to overwrite the **Equals** method. This method was overwritten so that not the whole object is compared, but only the property **CurrentSetting**.

Since the settings are stored sorted in the respective Rsops, a normal for-loop can be used to iterate over all settings and can be aborted as soon as a setting is no longer equivalent to the setting from the RsopPot to be checked (Listing 2.32 - line 13 - 19).

Listing 2.32: RsopPotService - SettingsEqual()

```
1 public bool SettingsEqual<T>(ICollection<T> currentSettings, ICollection<T>  
   otherSettings)  
2 {  
3     if (currentSettings == null || otherSettings == null)  
4     {  
5         return (currentSettings == null && otherSettings == null);  
6     }  
7  
8     if (currentSettings.Count() != otherSettings.Count())  
9     {  
10        return false;  
11    }  
12  
13    for (var i = 0; i < currentSettings.Count(); i++)  
14    {  
15        if (!currentSettings.ElementAt(i).Equals(otherSettings.ElementAt(i)))  
16        {  
17            return false;  
18        }  
19    }  
20  
21    return true;  
22 }
```

If the user later imports additional RSoPs that have not yet been analyzed, the first step is to analyze the RSoP and then check whether a RsopPot already exists that has the same settings as the imported Rsop. For this the method `UpdateRsopPots()` (Listing 2.33) is used for an import.

Therefore the list of imported RSoPs is iterated and it is checked if there is a RsopPot with the same settings (Listing 2.33 - line 5 - 20). If a RsopPot has the same settings as an imported RSoP, then this RSoP has already been assigned to the corresponding RsopPot (Listing 2.31 - line 27) and only the date of the RsopPot has to be adjusted and the RsopPot on the database has to be updated (Listing 2.33 - line 9 - 13). If no RsopPot could be found for the RSoP and it is an RSoP from a not yet captured organizational unit, a new RsopPot for the RSoP will be created (Listing 2.33 - line 14 - 19).

Listing 2.33: RsopPotService - UpdateRsopPots()

```

1 public async Task UpdateRsopPots(List<Rsop> rsops)
2 {
3     var rsopPots = await unitOfWork.RsopPotRepository.GetAllEntities();
4
5     foreach (var rsop in rsops)
6     {
7         var foundPot = RsopPotsEqual(rsopPots, rsop);
8
9         if (foundPot != null)
10        {
11            foundPot.DateTime = DateTime.Now.ToString("g", CultureInfo.InvariantCulture);
12            unitOfWork.RsopPotRepository.Update(foundPot);
13        }
14        else if (!rsopPots.Any(x => RsopAndRsopPotsOuEqual(rsop, x.Rsops.First())))
15        {
16            foundPot = RsopPotFactory(rsop);
17            rsopPots.Add(foundPot);
18            unitOfWork.RsopPotRepository.Add(foundPot);
19        }
20    }
21
22    await unitOfWork.SaveChangesAsync();
23 }

```

## Occurred problems

### Algorithm

There were basically no problems with the implementation of the service. However, it took some time until the algorithm was implemented, since checking individual RSoPs in the RsopPots didn't turn out to be easy.

Furthermore, it should be mentioned that the algorithm has a not quite perfect runtime. This is because each RSoP has to be compared with the already existing lists of RSoPs in the RsopPots. In addition, all recommended settings are checked individually. However, this should not take very much time, because the number of settings is limited to 34. Basically it was taken care that each iteration is aborted as early as possible and no unnecessary comparisons and iterations are required.

## 7.7 UC-R4: Visualize the analyzed data

### 7.7.1 Logic flow

- Load all domains and the according Group of Identical Security Settings (GISS) from the database
- Representation of the domains and GISS with visualization of the respective readiness
- Detailed representation of domain, GISS and organizational unit can be displayed

### 7.7.2 Implementation Forest Overview

The domains and GISS are displayed in a tree structure. In addition to the name, the time of the analysis is also displayed. This is to show the user that the content of the GISS can change after every use. On the right side a progressbar shows how good the readiness is. For the GISS the percentage of correctly set settings is displayed in green, for the domains the value of the worst member.

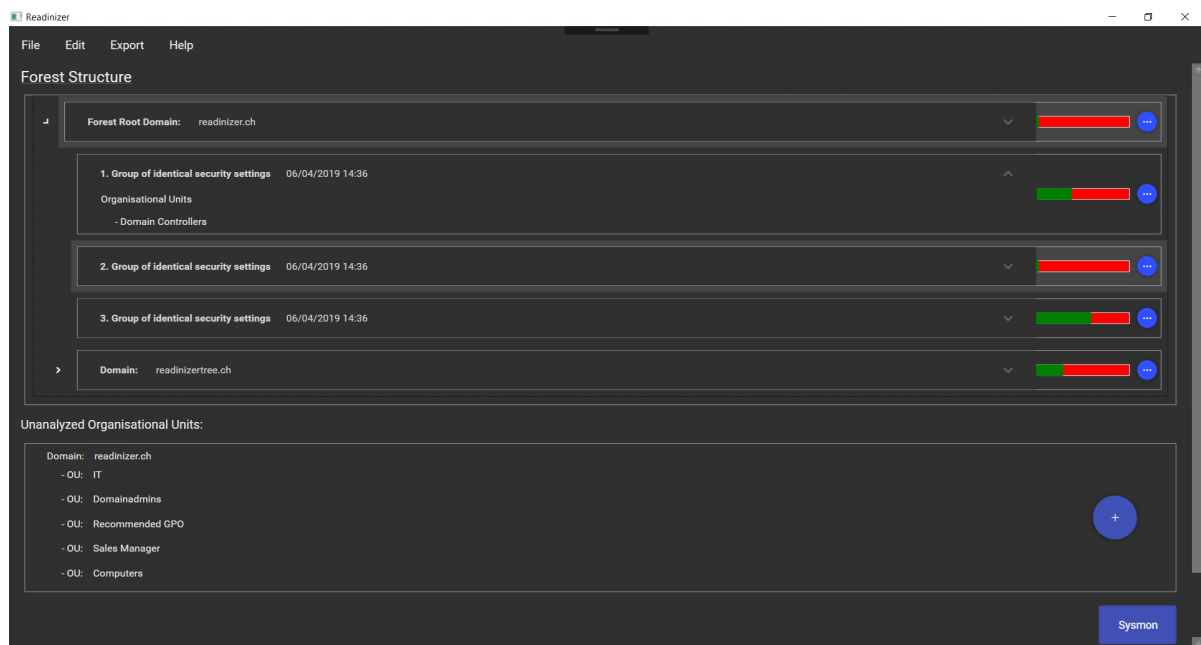


Figure 2.25: Forest Overview

### 7.7.3 Implementation Domain Overview

A pie chart is used to give an overview of the readies of the domain. Below are the good and bad GISS listed, these are clickable.

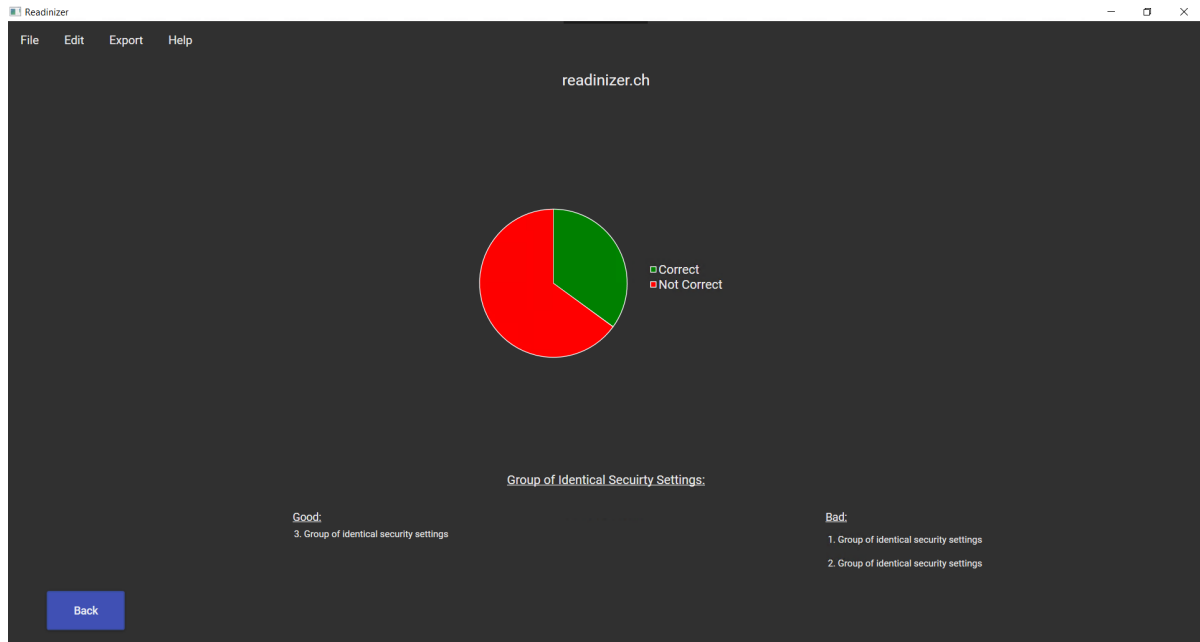


Figure 2.26: Domain Overview

### 7.7.4 Implementation GISS Overview

The GISS overview shows, based on an RSoP, the settings which are set in this group. In addition, the settings are compared with the recommended settings.

3. Group of identical security settings

Audit Settings:

Setting	Target Value	Current Value	Status
Audit Kerberos Service Ticket Operations	SuccessAndFailure	Success	✗
Audit Kernel Object	SuccessAndFailure	SuccessAndFailure	✓
Audit Logoff	Success	Success	✓
Audit Logon	SuccessAndFailure	Success	✗
Audit MPSSVC Rule-Level Policy Change	Success	NoAuditing	✗
Audit Non Sensitive Privilege Use	SuccessAndFailure	SuccessAndFailure	✓
Audit Other Account Management Events	SuccessAndFailure	SuccessAndFailure	✓

OU's in this GISS:

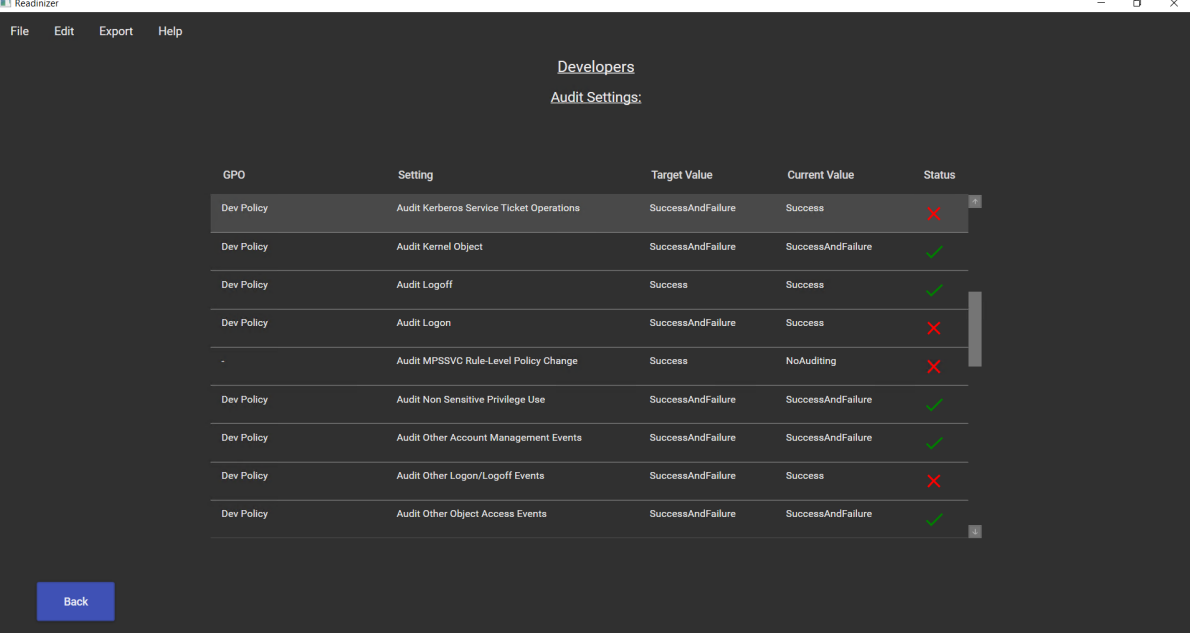
Sales  
Developers

Back

Figure 2.27: GISS Overview

### 7.7.5 Implementation OU Overview

This view is similar to the **GISS Overview**, but this time it is a specific RSoP. It also shows which Group Policy Object made this setting.



The screenshot shows the Readinizer application window with the 'Developers' tab selected. Below the tab, the 'Audit Settings' section displays a table with the following data:

GPO	Setting	Target Value	Current Value	Status
Dev Policy	Audit Kerberos Service Ticket Operations	SuccessAndFailure	Success	✗
Dev Policy	Audit Kernel Object	SuccessAndFailure	SuccessAndFailure	✓
Dev Policy	Audit Logoff	Success	Success	✓
Dev Policy	Audit Logon	SuccessAndFailure	Success	✗
-	Audit MPSSVC Rule-Level Policy Change	Success	NoAuditing	✗
Dev Policy	Audit Non Sensitive Privilege Use	SuccessAndFailure	SuccessAndFailure	✓
Dev Policy	Audit Other Account Management Events	SuccessAndFailure	SuccessAndFailure	✓
Dev Policy	Audit Other Logon/Logoff Events	SuccessAndFailure	Success	✗
Dev Policy	Audit Other Object Access Events	SuccessAndFailure	SuccessAndFailure	✓

A 'Back' button is located at the bottom left of the window.

Figure 2.28: OU Overview

### 7.7.6 Implementation Sysmon Overview

The Sysmon overview shows with a pie chart on how many devices in the network Sysmon Service is running. Beneath it, all devices are listed on which the Sysmon Service is not running.

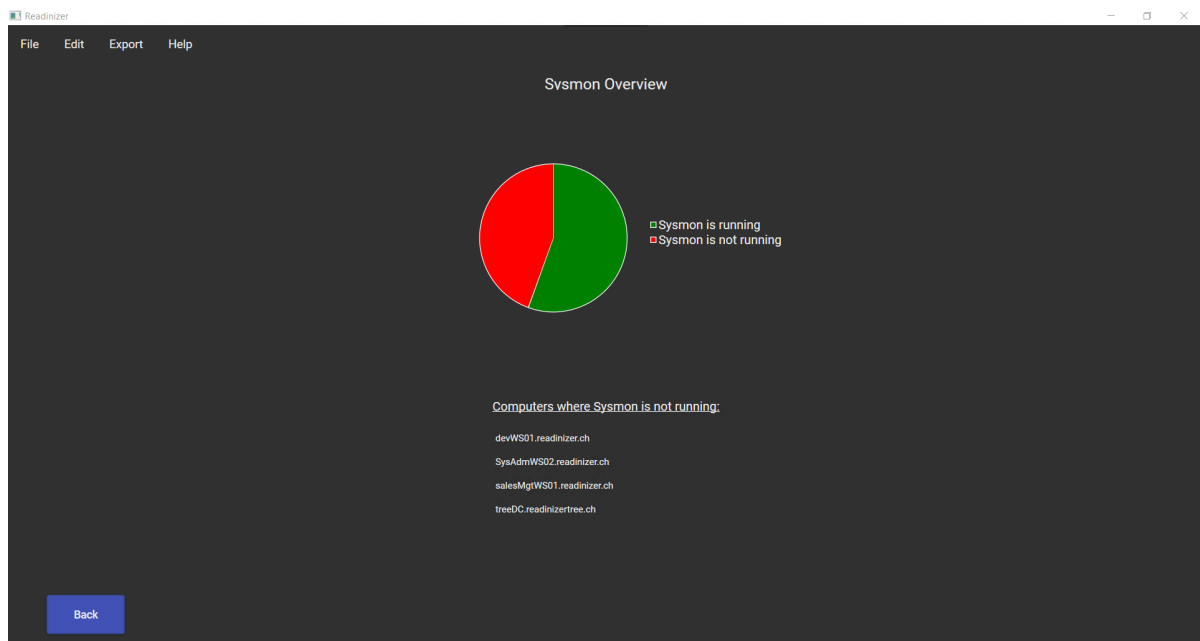


Figure 2.29: Sysmon Overview

## 7.8 UC-O1: Provide a recommended Group Policy Object

The Group Policy Object settings that are recommended in the section 3.6.1 GPO Settings Readinizer can be downloaded here:

<https://github.com/clma91/Readinizer/releases>.

This is the backup file of a recommended Group Policy Object. Download the ZIP file and unpack it. Open the Group Policy Management Console and create a Group Policy Object. Right click on the newly created GPO and select **Import Settings...**

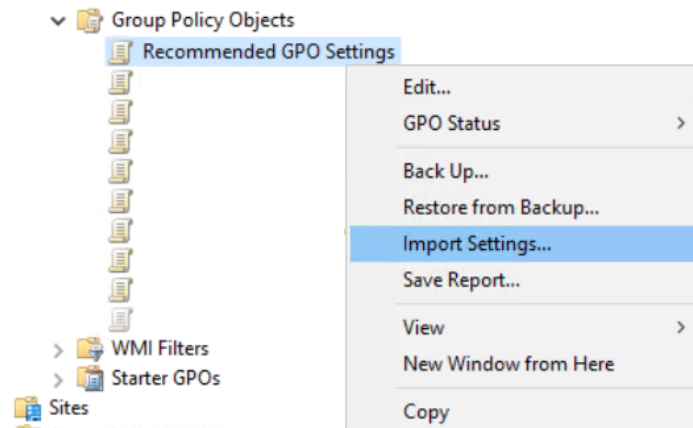


Figure 2.30: Import Settings

The “Import Settings Wizard” opens, provide the path to the downloaded backup file and import the settings. Link the new GPO to your domain.

## 7.9 UC-O2: Provide manual for fleet-wide Sysmon installation

This use case was solved with the document “Optimization: Installation of Sysmon via GPO”, this can be found in the appendix: “Sysmon Deployment Through GPO”.

## 7.10 UC-O3: Provide manual for fleet-wide central logging installation

This use case was solved with the document “Optimization: Provide manual for fleet-wide central logging installation”, this can be found in the appendix: “Windows Event Forwarding Deploying Fleet-Wide”.

## 8 Conclusion and Outlook

This part deals with the overall conclusion about the achieved work and delivered product as well as the used technologies and frameworks. Moreover, it will provide an outlook for further development and expansion in this area on the basis of this work.

### 8.1 Conclusion Achieved Work

The “Readinizer” application created during the thesis meets all the requirements set at the beginning of the project. The application “Readinizer” makes it possible to capture and analyze a complete Active Directory environment and its Resultant Set of Policies (see 6.1.1 UC-R1 - Discovering all organizational units and their members, 6.1.2 UC-R2 - Collecting Resultant Set of Policies of a fleet, 6.1.3 UC-R3 - Analyzing the collected data). Furthermore, a graphical user interface displays the result to the user so that a conclusion can be drawn about the readiness of a system and the configured audit settings (see 6.1.4 UC-R4 - Visualize the analyzed data).

The requirements for the optimization part have also been met and the user is provided with appropriate manuals (see 6.2.2 UC-O2 - Provide manual for fleet-wide Sysmon installation, 6.2.3 UC-O3 - Provide manual for fleet-wide central logging installation). With these the user can further harden his Active Directory infrastructure and prepare it for a solid analysis of lateral movements and APTs. In addition, prepared GPOs are made available to the user (see 6.2.1 UC-O1 - Provide a recommended Group Policies).

The non-functional requirements (see 6.3 Non Functional Requirements (NFR)) were met as follows:

- **NFR01** - Minimal target version
  - *Fulfilled*
  - The application meets the the minimal target version of Microsoft Windows 10 Professional v1709 and Microsoft Server 2016 Datacenter.
- **NFR02** - Performance
  - *Fulfilled*
  - On an average base the application does not exceed 6% of CPU usage (dual core Intel® Haswell 2.4 GHz E5-2673 v3 processors or better [87]) and 250MB of memory usage. The performance is additionally minimized by using the relational LocalDb in relation to a larger network.
- **NFR03** - Network Performance
  - *Partially fulfilled*
  - Unfortunately, the application was not tested on a large scale but the network traffic generated by the application is not significant because just one member of each organizational unit is requested. Hence, only a fraction of the entire fleet is affected.
- **NFR04** - Runtime
  - *Unknown*
  - Unfortunately, the application was not tested on a large scale. Therefore, it is not known if the application meets this non functional requirement. It does not only depend on the number of clients, but also how strongly structured the Active Directory network is.



- **NFR05** - Usability (Dependencies)
  - *Partially fulfilled*
  - The ultimate goal would have been to make the application available without any dependencies. However, the application depends on the RSAT library, which is required to get information from the Active Directory forest. In addition, a relational LocalDb is used to avoid memory overflow on a large networks and to keep the data complexity to a minimum.
- **NFR06** - Usability (Handling)
  - *Fulfilled*
  - The handling of the application should be self-explanatory for system administrators. Additional manuals with descriptions of the functionalities were created.
- **NFR07** - Integrity (GPO changes)
  - *Fulfilled*
  - The use of the application does not require any changes to the existing GPOs. Only additional GPOs are required.
- **NFR08** - Integrity / Security (WEF encryption)
  - *Fulfilled*
  - An encryption strength of 128 bit can be achieved by allowing AES as the encryption type.
- **NFR09** - Security (WEF authorization)
  - *Fulfilled*
  - The GPO can be used to define who should participate in central logging.
- **NFR10** - Security (WEC authorization)
  - *Fulfilled*
  - By setting up the WEC on a dedicated server, on which only certain users have access, it can be ensured that no unauthorized access will occur.

## 8.2 Conclusion Technologies and Frameworks

It turned out that the used programming language C-Sharp and the corresponding frameworks WPF, .NET Framework as well as Entity Framework were very suitable for the implementation of the task. This is due to the fact that the technologies used are products from Microsoft or are close to the operating system Windows. For this reason the AD forest queries could be implemented quite fast and easily.

With the LocalDb used, the data could easily be managed in a relational database, which greatly simplified the object-oriented approach with the relationships of the individual objects.

Furthermore, Microsoft Azure DevOps has proven to be an excellent project management platform. The main reasons for this were the easy handling (in areas like scrum and task management) as well as the already included continuous integration. Azure Cloud has proven its worth through very simple, agile and fast deployments of test machines and modification of the test environment. Github is not discussed here, because it is considered obvious to use the platform in a project of this size.

### 8.3 Outlook

Even though most of the requirements were met, there are further expansion options for the application.

#### 8.3.1 Dependencies

Make the application free of any dependencies (RSAT and LocalDb). The dependency to the library RSAT will hardly be preventable, but this library will be provided as an optional feature with newer Windows operating system versions, so this dependency could become obsolete. The dependency on LocalDb could be achieved by using a non relational database (NoSQL, documented-based store). However, we consider this a major task as the relationships between the classes have to be normalized. Neither we would not recommend an in-memory solution as this could lead to massive performance problems within larger networks.

#### 8.3.2 User Permissions

The user permissions could be checked before the analysis is started and the user could be given a chance to impersonate another user for the application.

#### 8.3.3 Sysmon

The Sysmon logs could be analyzed in detail and a config-file for the Sysmon could be created. So it could be defined exactly what should be logged over the Sysmon.

#### 8.3.4 Parallelisation

Possibly there is an approach to parallelize the queries that are triggered on the network. This, however, escapes our knowledge as we did not visit the corresponding module Parallel Programming.

#### 8.3.5 Log Pattern

It could be examined in detail which information an event must contain in order to be classified as possible lateral movement or APT.

#### 8.3.6 Monitoring

On a larger scale, one can combine all three parts “Readinizer, Visualizer and Optimization” into a monitoring tool. For example Winlogbeat [88] or HELK [89] could be used. With Winlogbeat event logs can be brought into a structured format so that they can be filtered and aggregated more easily in Elasticsearch. HELK deals with the same topic, but tailored to a Windows Event Forwarding environment and enables data science for possible machine learning.

#### 8.3.7 Monitoring - Anomalies

Through the use of central logging, certain patterns (or anomalies) could be detected across all stored logs, which would result in corresponding alarm messages.

# Glossary

<b>ACSC</b>	Australian Cyber Security Center, Australian government agency responsible for cyber security
<b>AD</b>	Active Directory, a directory service that Microsoft developed for the Windows domain networks
<b>AD DS</b>	Active Directory Domain Service, a server role in Active Directory that allows admins to manage and store information about resources from a network in a distributed database
<b>APT</b>	Advanced Persistent Threat, a stealthy computer network attack in which the attacker gains unauthorized access to a network and remains undetected for an extended period
<b>ARP</b>	Address Resolution Protocol, that determines the physical address of the network access layer for a network address of the Internet layer
<b>C#</b>	Csharp, a general-purpose programming language developed by Microsoft
<b>CAPI2</b>	CryptoAPI2, a Microsoft Windows platform specific Cryptographic Application Programming Interface from Windows Vista or newer, offers function for encrypting and decrypting data and strong authentication with digital certificates and secure generation of random numbers
<b>CERT-EU</b>	Computer Emergency Response Team for EU institutions, bodies and agencies
<b>CI</b>	Continuous Integration, continuous assembly of components into an application, mostly on a server with automatic builds and tests
<b>CSE</b>	Client-Side Extensions, an integral component of enterprise group policy administration that applies Group Policy to users or endpoint systems
<b>DLL</b>	Dynamic Link Library, is Microsoft's implementation of the shared library concept in the Microsoft Windows
<b>DNS</b>	Domain Name Systems, a decentralized naming system that breaks down IP addresses into names and vice versa
<b>EF</b>	Entity Framework, is a framework for object-relational mapping
<b>EXE</b>	Executable File for different OS
<b>FQDN</b>	Fully-Qualified Domain Name, uniquely identifies a particular computer

<b>FTP</b>	File Transfer Protocol, Network protocol for transferring files over IP networks.
<b>GB</b>	Gigabyte, is a unit of measurement for digital technology and computer science, 1 GB is 10 <sup>9</sup> Byte
<b>GISS</b>	Group of Identical Security Settings, a group where the relevant security settings in a RSoP are identical
<b>GPMC</b>	Group Policy Management Console, a tool to administrate Group Policy Objects and their deployment
<b>GPO</b>	Group Policy Objects, is a digital policy for various settings under Microsoft Windows 2000 and its successors
<b>GUI</b>	Graphic User Interface, a form of user interface of a computer, make application software operable for humans on a computer by means of graphic symbols and control elements
<b>GUID</b>	Globally Unique Identifier, to uniquely identify objects in Windows networks
<b>HTML</b>	Hypertext Markup Language, a text-based markup language for structuring electronic documents such as texts with hyperlinks, images and other content
<b>HTTP</b>	Hypertext Transport Protocol, is an application protocol for distributed, collaborative, hypermedia information systems, foundation of data communication for the World Wide Web
<b>ICMP</b>	Internet Control Message Protocol, is used in computer networks to exchange information and error messages via the IP protocol
<b>ID</b>	Identifier, is a characteristic linked to a particular identity for the unique identification of the load-bearing object
<b>IDE</b>	Integrated development environment, is a software application that provides comprehensive facilities to computer programmers for software development
<b>IP</b>	Internet-Protocol, widely used network protocol, represents the basis of the internet
<b>JPCERT/CC</b>	Japan Computer Emergency Response Team Coordination Center, a Computer Security Incident Response Team established in Japan
<b>JSON</b>	JavaScript Object Notation, compact data format in an easily readable text form for the purpose of data exchange between applications
<b>KB</b>	Kilobyte, is a unit of measurement for digital technology and computer science, 1 KB is 1000 Byte
<b>LaTeX</b>	Lamport TeX, is a software package that simplifies the use of the TeX typesetting system with the help of macros
<b>LDAP</b>	Lightweight Directory Access Protocol, Network protocol for querying and changing information in an Active Directory

<b>LGPO / Local GPO</b>	Local Group Policy Object, is a GPO which applies only to a single computer
<b>LPC</b>	Local Procedure Call, an interprocess communication facility for high-speed message passing available, only to Windows operating system components
<b>LSA</b>	Local Security Authority, a protected subsystem that authenticates and logs users onto the local system
<b>MB</b>	Megabyte, is a unit of measurement for digital technology and computer science, 1 MB is 10 <sup>6</sup> Byte
<b>MITRE ATT&amp;CK</b>	MITRE Adversarial Tactics, Techniques, and Common Knowledge, a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations by MITRE, a non-profit organization which manages federally funded research and development centers supporting several U.S. government agencies
<b>MPSSVC</b>	Is part of Windows Firewall, which protects computers by preventing unauthorized users from gaining access through the Internet or a network
<b>MSSQL</b>	Microsoft SQL Server, a relational database management system from Microsoft.
<b>MVVM</b>	Model View ViewModel, architectural pattern, the view model is responsible for converting the data objects from the model in such a way that objects are easily managed and presented by the view
<b>NFR</b>	Non Functional Requirement, a statement about a property or performance to be fulfilled by a product, system or process, usually measurable
<b>NSA</b>	National Security Agency, intelligence agency of the United States Department of Defense
<b>OU</b>	Organizational Unit, smallest grouping container in a Windows network
<b>PoC</b>	Proof of Concept, a milestone at which the basic feasibility of a project has been demonstrated
<b>RDP</b>	Remote Desktop Protocol, is a Microsoft network protocol for remote access to Windows computers
<b>REST</b>	Representational State Transfer, a software architectural style that defines a set of constraints to be used for creating Web services
<b>RPC</b>	Remote Procedure Calls, a technique for implementing interprocess communication that enables functions to be called in other address spaces.
<b>RSAT</b>	Remote Server Administration Tool, a Windows server component for remote computer management
<b>RSoP</b>	Resultant Set of Policies, is an overview of all group policy settings within the Active Directory structure

<b>SAM</b>	Security Accounts Manager, is a Microsoft Windows service that stores user information such as logon name and password as hash values in a database.
<b>SANS</b>	SysAdmin, Networking and Security, US-company that specializes in information security, cybersecurity training and selling certificates
<b>SCT</b>	Microsoft Security Compliance Toolkit, this set of tools allows enterprise security administrators analyze, test, edit and store Microsoft-recommended security configuration baselines for Windows and other Microsoft products, while comparing them against other security configurations
<b>SID</b>	Security Identifier, A unique security identifier that Microsoft Windows NT automatically assigns
<b>SIP</b>	Subject Interface Packages, allow the CryptAPI to interact with specific parts of the files in order to put, get, calculate and verify digital signatures on files
<b>SMB</b>	Server Message Block, mainly used for providing shared access to files, printers, and serial ports and miscellaneous communications between nodes on a network
<b>SOM</b>	Scope of Management, containers that can contain user and computer accounts that can be managed through Group Policy
<b>SYSVOL</b>	System Volume, a shared directory that stores the server copy of the domain's public files that must be shared for common access and replication throughout a domain
<b>TCP</b>	Transmission Control Protocol, is one of the main protocols of the Internet protocol suite, is a connection-oriented protocol
<b>TGS</b>	Ticket Granting Server?,
<b>TGT</b>	Ticket Granting Ticket, is a small file that, similar to a password, but more secure, allows access to a data exchange
<b>UC</b>	Use Case, the externally visible behavior of a system is described from the user's point of view
<b>UC-On</b>	Use Case Optimization, Use Case for the optimization part, the n stands for the ID of the Use Case
<b>UC-Rn</b>	Use Case Readinizer, Use Case for Readinizer application, the n stands for the ID of the Use Case
<b>UML</b>	Unified Modeling Language, a graphical modeling language for specification, design and documentation of software parts and other systems
<b>UoW</b>	Unit of Work, a pattern that keeps track of everything done during a business transaction that can affect the database
<b>VPN</b>	Virtual Private Network, a private network that enables users to send and receive data securely and encrypted over public or shared networks

<b>WEF</b>	Windows Event Forwarding, reads any operational or administrative event log on a device in a organization and forwards the events chosen to a Windows Event Collector server
<b>WEFFLES</b>	Windows Event Logging Forensic Logging Enhancement Services, a Threat Hunting/Incident Response Console with Windows Event Forwarding and visualized with PowerBI
<b>wevtutil</b>	Windows Event Log Tools Utility, enables to get information about event logs and publishers.
<b>WFP</b>	Windows Filtering Platform, set of API and system services that provide a platform for creating network filtering applications
<b>WMI</b>	Windows Management Instrumentation, a Interface for administration and remote maintenance of workstations and servers
<b>WPF</b>	Windows Presentation Foundation, Graphic framework and window system of the .NET Framework from Microsoft
<b>XML</b>	Extensible Markup Language, a markup language for the representation of hierarchically structured data in the format of a text file, which is readable both by humans and by machines

# Listings

2.1	Unity Dependency Injection . . . . .	64
2.2	Generic Repository . . . . .	64
2.3	Unit of Work . . . . .	65
2.4	ApplicationViewModel - Constructor . . . . .	66
2.5	ChangeView Class . . . . .	67
2.6	ApplicationViewModel - ChangeView Method . . . . .	67
2.7	View - ViewModel Binding . . . . .	67
2.8	ADDomainService - SearchDomains() Part 1 . . . . .	69
2.9	ADDomainService - AddAllTreeDomains . . . . .	70
2.10	ADDomainService - AddAllChildDomains . . . . .	71
2.11	ADDomainService - SearchAllDomains() Part 2 . . . . .	71
2.12	OrganizationalUnitService - GetAllOrganizationalUnits . . . . .	72
2.13	OrganizationalUnitService - GetChildOUs . . . . .	73
2.14	Ping target computer . . . . .	74
2.15	GPRsop use . . . . .	75
2.16	Check if Sysmon service is running . . . . .	77
2.17	RSOP-XML - General Information . . . . .	78
2.18	RSOP-XML - GPO . . . . .	79
2.19	RSOP-XML - AuditSettings . . . . .	81
2.20	RSOP-XML - SecurityOptions . . . . .	82
2.21	RSOP-XML - Policy and RegistrySettings . . . . .	83
2.22	AnalysisService - Analyze() . . . . .	84
2.23	AnalysisService - AnaylseAuditSettings() Part 1 . . . . .	85
2.24	AnalysisService - AnaylseAuditSettings() Part 2 . . . . .	85
2.25	AnalysisService - AnaylseAuditSettings() Part 3 . . . . .	86
2.26	AnalysisService - AnaylseAuditSettings() Part 4 . . . . .	86
2.27	GetSettings() . . . . .	87
2.28	ReadJson Override . . . . .	88
2.29	SingleValueArrayConverter in class Gpo . . . . .	88
2.30	RsopPotService - FillRsopPotList() . . . . .	89
2.31	RsopPotService - RsopPotsEqual() . . . . .	90
2.32	RsopPotService - SettingsEqual() . . . . .	91
2.33	RsopPotService - UpdateRsopPots() . . . . .	92
G.1	Sysmon Installation Batch File . . . . .	V
G.2	Check if service exists . . . . .	V
G.3	Check if Service exists . . . . .	VI
G.4	Installing Sysmon . . . . .	VI
K.1	Subscription XML . . . . .	XI
M.1	ReadinizerWEFRecommendation . . . . .	XVI



# List of Figures

2.1	Windows Network Environment . . . . .	3
2.2	Site Objects . . . . .	4
2.3	Group Policy Engine [1] . . . . .	5
2.4	GPO Precedence . . . . .	7
2.5	Example which GPO will apply . . . . .	8
2.6	Typical lateral movement attack [5, p. 4] . . . . .	10
2.7	Advanced Audit Policy - Logon/Logoff - Audit Special Logon . . . . .	33
2.8	Test Environment . . . . .	38
2.9	Domain readinizer.ch . . . . .	40
2.10	Active Directory Domain Model . . . . .	42
2.11	GUI Readinizer StartUp View . . . . .	44
2.12	GUI Readinizer Result View . . . . .	45
2.13	GUI Readinizer Overview Domain/Subdomain View . . . . .	46
2.14	GUI Readinizer Overview OUs View . . . . .	47
2.15	GUI Readinizer Navigation Bar . . . . .	47
2.16	Data Model . . . . .	48
2.17	Use Case Diagram . . . . .	50
2.18	Logical Architecture - Package Diagram . . . . .	58
2.19	System Architecture - Deployment Diagram Rejected . . . . .	59
2.20	System Architecture - Deployment Diagram Accepted . . . . .	59
2.21	Dependency Diagram . . . . .	63
2.22	ApplicationView . . . . .	66
2.23	Sniffed communication . . . . .	76
2.24	GPOs in a forest structure example . . . . .	80
2.25	Forest Overview . . . . .	93
2.26	Domain Overview . . . . .	94
2.27	GISS Overview . . . . .	94
2.28	OU Overview . . . . .	95
2.29	Sysmon Overview . . . . .	95
2.30	Import Settings . . . . .	96
A.1	Rest Result . . . . .	XXXIII
B.1	Time by Activity Type . . . . .	XXXV
B.2	Time in Inception . . . . .	XXXVI
B.3	Time in Elaboration . . . . .	XXXVI
B.4	Time in Construction . . . . .	XXXVII
B.5	Time in Transition . . . . .	XXXVII
B.6	Sprints - Estimated Time vs. Actual Time . . . . .	XXXVIII
G.1	SYSVOL-folder . . . . .	IV
G.2	DefinitelyNotSysmon Service . . . . .	IV

G.3 Create Scheduled Tasks . . . . .	VII
G.4 Scheduled Task - General . . . . .	VII
G.5 Scheduled Task - Trigger . . . . .	VIII
G.6 Scheduled Task - Action . . . . .	VIII
G.7 Scheduled Task - Action . . . . .	IX
K.1 Enable WinRM . . . . .	IV
K.2 Enable Subscriptions in the Event Viewer . . . . .	V
K.3 Information about the Security Event Log . . . . .	V
K.4 Configure GPO - Define WEC . . . . .	VI
K.5 Configure GPO - Log Access . . . . .	VI
K.6 Enable WinRM Service . . . . .	VII
K.7 GPO Summary . . . . .	VIII
K.8 Subscription Properties . . . . .	IX
K.9 Subscription Computer Groups . . . . .	IX
K.10 Subscription Query Filter . . . . .	X
K.11 Subscription Advanced Settings . . . . .	X
K.12 Kerberos encryption . . . . .	XIV
O.1 Domain Rights . . . . .	II
O.2 Manage optional features . . . . .	III
O.3 RSAT: Group Policy Management Tools . . . . .	III
O.4 SQL Server Express 2017 . . . . .	IV
O.5 SQL Server Express 2017 . . . . .	IV
O.6 SQL Server Express 2017 . . . . .	V
Q.1 Readinizer Home Screen . . . . .	VII
Q.2 Readinizer Forest Result Screen . . . . .	VIII
Q.3 Readinizer Domain Result Screen . . . . .	IX
Q.4 Readinizer Group of Identical Security Settings Result Screen . . . . .	X
Q.5 Readinizer Organizational Unit Result Screen . . . . .	XI
Q.6 Readinizer Sysmon Result Screen . . . . .	XII
Q.7 Navigationbar File . . . . .	XIII
Q.8 Navigationbar Export . . . . .	XIII
Q.9 Navigationbar Help . . . . .	XIII

# List of Tables

2.1	Group Policy Engine [1] [2]	6
2.2	Caching credentials [5, p. 4-5]	11
2.3	Recommended Event Loggings on the domain controller [5, p. 3-10]	11
2.4	Recommended Event Loggings on Workstations [5, p. 3-10]	12
2.5	Windows Firewall [6, p. 25] [7]	16
2.6	Clearing Event Logs [6, p. 25] [3, p. 17] [8]	17
2.7	Software & Service Installation [6, p. 26] [3, p. 17]	17
2.8	Account Usage [6, p. 26-27] [3, p. 18-19]	18
2.9	Windows Defender Activities [6, p. 28-29] [10]	19
2.10	ACSC event categories [11, p. 3-5]	21
2.11	ACSC vs. PoC - Account lockout	22
2.12	ACSC vs. PoC - Account modifications	22
2.13	ACSC vs. PoC - Account logon	22
2.14	ACSC vs. PoC - Event collection	23
2.15	ACSC vs. PoC - Process tracking	23
2.16	ACSC vs. PoC - Code integrity	23
2.17	ACSC vs. PoC - File shares	23
2.18	Force Audit Policy Subcategory Settings	34
2.19	Advanced Audit Policy Setting Account Logon	34
2.20	Advanced Audit Policy Setting Account Management	34
2.21	Advanced Audit Policy Setting Detailed Tracking	34
2.22	Advanced Audit Policy Setting DS Access	34
2.23	Advanced Audit Policy Setting Logon/Logoff	35
2.24	Advanced Audit Policy Setting Object Access	35
2.25	Advanced Audit Policy Setting Policy Change	35
2.26	Advanced Audit Policy Setting Privilege Use	35
2.27	Advanced Audit Policy Setting System	36
2.28	Administrative Template System	36
2.29	Administrative Template Windows Components Windows PowerShell	36
2.30	Add Registry Key for LSA Protection [29]	37
2.31	Test Environment User	39
2.32	Non Functional Requirements	57
A.1	ST1 - Execute Readinizer	XXVIII
A.2	ST2.0 - Analyze a specified domain	XXIX
A.3	ST2.1 - Domain Overview	XXIX
A.4	ST2.2 - GISS Overview	XXX
A.5	ST2.3 - Organizational Unit Overview	XXX
A.6	ST3 - New Analysis	XXXI
A.7	ST4 - Analyze all domains and check Sysmon service	XXXI

A.8 ST5 - Add a Organizational Unit that could not be analyzed . . . . .	XXXII
A.9 ST6 - Sysmon Overview . . . . .	XXXII
A.10 ST7 - Export to JSON . . . . .	XXXIII
A.11 Code metrics - Visual Studio . . . . .	XXXIV
A.12 Code metrics - Visual Studio . . . . .	XXXIV
K.1 Kerberos Encryption Types [90] . . . . .	XIII

# Bibliography

- [1] Microsoft. What Is Core Group Policy? [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc779077\(v%3dws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc779077(v%3dws.10)), October 2009. Accessed: 21.03.2019.
- [2] Microsoft. How Core Group Policy Works. [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc784268\(v%3dws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc784268(v%3dws.10)), November 2011. Accessed: 21.03.2019.
- [3] Lukas Kellenberger, Claudio Mattes. Readiness for Tailored Attacks and Lateral Movement Detection, Fall Term 2018.
- [4] JPCERT/CC. Detecting Lateral Movement through Tracking Event Logs. , June 2017.
- [5] CERT-EU. Detecting Lateral Movements in Windows Infrastructure. , February 2017.
- [6] NSA. Spotting the Adversary with Windows Event Log Monitoring. , December 2013.
- [7] Microsoft. Step 8: Enabling Firewall Logging. [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc754451\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc754451(v=ws.10)), December 2009. Accessed: 04.04.2019.
- [8] Microsoft. Other Events. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/other-events>, April 2017. Accessed: 06.04.2019.
- [9] Microsoft. How to: Log Information About Services. <https://docs.microsoft.com/en-us/dotnet/framework/windows-services/how-to-log-information-about-services>, March 2017. Accessed: 05.04.2019.
- [10] Microsoft. Review event logs and error codes to troubleshoot issues with Windows Defender Antivirus. <https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-antivirus/troubleshoot-windows-defender-antivirus>, September 2018. Accessed: 05.04.2019.
- [11] Australian Cyber Security Center - ACSC. Windows Event Logging and Forwarding. , January 2019.
- [12] Microsoft. Windows error reporting. <https://docs.microsoft.com/en-us/windows/deployment/upgrade/windows-error-reporting>, March 2018. Accessed: 04.04.2019.
- [13] Microsoft. Audit Account Lockout. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-account-lockout>, July 2018. Accessed: 19.03.2019.
- [14] Microsoft. Audit Computer Account Management. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-computer-account-management>, April 2017. Accessed: 19.03.2019.

- [15] Microsoft. Audit Other Account Management Events. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-other-account-management-events>, April 2017. Accessed: 19.03.2019.
- [16] Microsoft. Audit User Account Management. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-user-account-management>, April 2017. Accessed: 20.03.2019.
- [17] Microsoft. Audit Other Logon/Logoff Events. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-other-logonlogoff-events>, April 2017. Accessed: 19.03.2019.
- [18] Microsoft. Audit Group Membership. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-group-membership>, April 2017. Accessed: 19.03.2019.
- [19] Microsoft. Audit Special Logon. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-special-logon>, April 2017. Accessed: 20.03.2019.
- [20] Microsoft. Audit policy change. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/basic-audit-policy-change>, April 2017. Accessed: 19.03.2019.
- [21] Microsoft. Command line process auditing. <https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/manage/component-updates/command-line-process-auditing>, May 2017. Accessed: 19.03.2019.
- [22] Microsoft. Audit System Integrity. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-system-integrity>, April 2017. Accessed: 19.03.2019.
- [23] Microsoft. Audit Detailed File Share. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-detailed-file-share>, April 2017. Accessed: 19.03.2019.
- [24] Microsoft. Audit Detailed File Share. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-file-share>, April 2017. Accessed: 19.03.2019.
- [25] Microsoft. Audit File System. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-file-system>, April 2017. Accessed: 20.03.2019.
- [26] MITRE ATT&CK. MITRE ATT&CK Website. <https://attack.mitre.org/>, September 2018. Accessed: 07.03.2019.
- [27] MITRE ATT&CK. LSASS Driver. <https://attack.mitre.org/techniques/T1177/>, March 2019. Accessed: 28.03.2019.
- [28] Microsoft. Disabling Secure Boot. <https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/disabling-secure-boot>, May 2017. Accessed: 20.04.2019.
- [29] Microsoft. Configuring Additional LSA Protection. <https://docs.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/configuring-additional-lsa-protection>, October 2016. Accessed: 28.03.2019.
- [30] MITRE ATT&CK. PowerShell. <https://attack.mitre.org/techniques/T1086/>, March 2019. Accessed: 28.03.2019.
- [31] MITRE ATT&CK. Scripting. <https://attack.mitre.org/techniques/T1064/>, March 2019. Accessed: 28.03.2019.

- [32] Malware Archaeology. WINDOWS POWERSHELL LOGGING CHEAT SHEET - Win 7/Win 2008 or later. <https://static1.squarespace.com/static/552092d5e4b0661088167e5c/t/5760096ecf80a129e0b1763> June 2016. Accessed: 28.03.2019.
- [33] MITRE ATT&CK. Scheduled Task. <https://attack.mitre.org/techniques/T1053/>, March 2019. Accessed: 28.03.2019.
- [34] Glenn Slayden. How can I enable the Windows Server Task Scheduler History recording? <https://stackoverflow.com/questions/11013132/how-can-i-enable-the-windows-server-task-scheduler-history-recording>, June 2012. Accessed: 28.03.2019.
- [35] MITRE ATT&CK. Process Injection. <https://attack.mitre.org/techniques/T1055/>, March 2019. Accessed: 28.03.2019.
- [36] MITRE ATT&CK. Credential Dumping. <https://attack.mitre.org/techniques/T1003/>, March 2019. Accessed: 28.03.2019.
- [37] MITRE ATT&CK. SIP and Trust Provider Hijacking. <https://attack.mitre.org/techniques/T1198/>, March 2019. Accessed: 28.03.2019.
- [38] MITRE ATT&CK. New Service. <https://attack.mitre.org/techniques/T1050/>, March 2019. Accessed: 28.03.2019.
- [39] Microsoft. Audit Security System Extension. <https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-security-system-extension>, April 2017. Accessed: 28.03.2019.
- [40] MITRE ATT&CK. Valid Accounts. <https://attack.mitre.org/techniques/T1078/>, March 2019. Accessed: 28.03.2019.
- [41] MITRE ATT&CK. Account Manipulation. <https://attack.mitre.org/techniques/T1098/>, March 2019. Accessed: 28.03.2019.
- [42] MITRE ATT&CK. Create Account. <https://attack.mitre.org/techniques/T1136/>, March 2019. Accessed: 28.03.2019.
- [43] MITRE ATT&CK. Brute Force. <https://attack.mitre.org/techniques/T1110/>, March 2019. Accessed: 28.03.2019.
- [44] MITRE ATT&CK. External Remote Services. <https://attack.mitre.org/techniques/T1133/>, March 2019. Accessed: 28.03.2019.
- [45] MITRE ATT&CK. SID-History Injection. <https://attack.mitre.org/techniques/T1178/>, March 2019. Accessed: 28.03.2019.
- [46] MITRE ATT&CK. Windows Admin Shares. <https://attack.mitre.org/techniques/T1077/>, March 2019. Accessed: 28.03.2019.
- [47] MITRE ATT&CK. Pass the Hash. <https://attack.mitre.org/techniques/T1075/>, March 2019. Accessed: 28.03.2019.
- [48] MITRE ATT&CK. Kerberoasting. <https://attack.mitre.org/techniques/T1208/>, March 2019. Accessed: 28.03.2019.
- [49] MITRE ATT&CK. Pass the Ticket. <https://attack.mitre.org/techniques/T1097/>, March 2019. Accessed: 28.03.2019.

- [50] MITRE ATT&CK. Remote File Copy. <https://attack.mitre.org/techniques/T1105/>, March 2019. Accessed: 28.03.2019.
- [51] Rob Lee. SANS DFIR. 2019.
- [52] Mark Russinovich, Thomas Garnier. Sysmon v9.0. <https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon>, February 2019. Accessed: 28.02.2019.
- [53] Microsoft. C# Guide. <https://docs.microsoft.com/en-us/dotnet/csharp/index>, January 2018. Accessed: 16.04.2019.
- [54] Microsoft. Windows Presentation Foundation. <https://docs.microsoft.com/en-us/dotnet/framework/wpf/index>, January 2018. Accessed: 16.04.2019.
- [55] Microsoft. .NET Framework Guide. <https://docs.microsoft.com/en-us/dotnet/framework/>, April 2018. Accessed: 16.04.2019.
- [56] Microsoft. DirectoryInfo Class. <https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.directoryentry?view=netframework-4.7.2>, March 2019. Accessed: 06.04.2019.
- [57] Microsoft. DirectorySearcher Class. <https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.directorysearcher?view=netframework-4.7.2>, March 2019. Accessed: 08.04.2019.
- [58] Microsoft. Entity Framework Documentation. <https://docs.microsoft.com/en-us/ef/>, March 2019. Accessed: 16.04.2019.
- [59] Microsoft. LocalDB. <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/sql-server-2016-express-localdb?view=sql-server-2017>, March 2019. Accessed: 16.04.2019.
- [60] Newtonsoft. Json.NET. <https://www.newtonsoft.com/json>, March 2019. Accessed: 16.04.2019.
- [61] NLog. NLog Project. <https://nlog-project.org/>, March 2019. Accessed: 16.04.2019.
- [62] Microsoft. Azure DevOps. <https://azure.microsoft.com/en-us/services/devops/>, March 2019. Accessed: 11.03.2019.
- [63] Microsoft. Azure. <https://azure.microsoft.com>, September 2018. Accessed: 11.03.2019.
- [64] GitHub. . <https://github.com/>, September 2018. Accessed: 11.03.2019.
- [65] Microsoft. .NET Core Guide. <https://docs.microsoft.com/en-us/dotnet/core/>, August 2018. Accessed: 16.04.2019.
- [66] Microsoft. Choosing between .NET Core and .NET Framework for server apps. <https://docs.microsoft.com/en-us/dotnet/standard/choosing-core-framework-server>, June 2018. Accessed: 16.04.2019.
- [67] Prism Library. Prism Library. <https://prismlibrary.github.io/docs/>, April 2019. Accessed: 12.04.2019.
- [68] Prism Library. Introduction to Prism. <https://prismlibrary.github.io/>, April 2019. Accessed: 12.04.2019.



- [69] Neo4j. Neo4j. <https://neo4j.com/>, March 2019. Accessed: 16.04.2019.
- [70] Andrew Robbins, Rohan Vazarkar, Will Schroede. BloodHoundAD. <https://github.com/BloodHoundAD/BloodHound>, March 2019. Accessed: 12.04.2019.
- [71] Redmine. Redmine. <https://www.redmine.org/>, March 2019. Accessed: 16.04.2019.
- [72] Microsoft, Tom Dykstra. Implementing the Repository and Unit of Work Patterns in an ASP.NET MVC Application. <https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions/getting-started-with-ef-5-using-mvc-4/implementing-the-repository-and-unit-of-work-patterns-in-an-asp-net-mvc-application#implement-a-generic-repository-and-a-unit-of-work-class>, July 2013. Accessed: 26.05.2019.
- [73] Microsoft. Forest.GetCurrentForest Method. [https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.activedirectory.forest.getcurrentforest?view=netframework-4.8#System\\_DirectoryServices\\_ActiveDirectory\\_Forest\\_GetCurrentForest](https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.activedirectory.forest.getcurrentforest?view=netframework-4.8#System_DirectoryServices_ActiveDirectory_Forest_GetCurrentForest), April 2019. Accessed: 03.06.2019.
- [74] Microsoft. Domain.GetDomain(DirectoryContext) Method. <https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.activedirectory.domain.getdomain?view=netframework-4.8>, April 2019. Accessed: 03.06.2019.
- [75] Microsoft. Ping Class. <https://docs.microsoft.com/en-us/dotnet/api/system.net.networkinformation.ping?view=4.8>.
- [76] Microsoft. PingReply Class. <https://docs.microsoft.com/en-us/dotnet/api/system.net.networkinformation.pingreply?view=netframework-4.8>.
- [77] Microsoft. IPStatus Enum. <https://docs.microsoft.com/en-us/dotnet/api/system.net.networkinformation.ipstatus?view=netframework-4.8>.
- [78] Microsoft. ping. <https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/ping#syntax>, November 2018. Accessed: 26.05.2019.
- [79] Microsoft. GPRsop. [https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi\\_v2/class-library/gprsop-class-microsoft-grouppolicy](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi_v2/class-library/gprsop-class-microsoft-grouppolicy), October 2016. Accessed: 10.05.2019.
- [80] Microsoft. Remote Server Administration Tools for Windows 10. <https://www.microsoft.com/en-us/download/details.aspx?id=45520>, February 2018. Accessed: 10.05.2019.
- [81] Microsoft. GPRsop Constructor (RsopMode, String). [https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi\\_v2/class-library/gprsop-constructor-rsopmode-string-microsoft-grouppolicy](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi_v2/class-library/gprsop-constructor-rsopmode-string-microsoft-grouppolicy), October 2016. Accessed: 12.05.2019.
- [82] Microsoft. LoggingMode Enumeration. [https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi\\_v2/class-library/loggingmode-enumeration-microsoft-grouppolicy](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi_v2/class-library/loggingmode-enumeration-microsoft-grouppolicy), October 2016. Accessed: 12.05.2019.
- [83] Microsoft. GPRsop.LoggingComputer Property. [https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi\\_v2/class-library/gprsop-loggingcomputer-property-microsoft-grouppolicy](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi_v2/class-library/gprsop-loggingcomputer-property-microsoft-grouppolicy), October 2016. Accessed: 16.05.2019.
- [84] Microsoft. GPRsop.LoggingUser Property. [https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi\\_v2/class-library/gprsop-logginguser-property-microsoft-grouppolicy](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi_v2/class-library/gprsop-logginguser-property-microsoft-grouppolicy), October 2016. Accessed: 16.05.2019.

- [85] Microsoft. GPRsop.CreateQueryResults Method (). [https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi\\_v2/class-library/gprsop-createqueryresults-method-microsoft-grouppolicy](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi_v2/class-library/gprsop-createqueryresults-method-microsoft-grouppolicy), October 2016. Accessed: 16.05.2019.
- [86] Microsoft. GPRsop.GenerateReportToFile Method (ReportType, String). [https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi\\_v2/class-library/gprsop-generatereporttofile-method-reporttype-string-microsoft-grouppolicy](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/wmi_v2/class-library/gprsop-generatereporttofile-method-reporttype-string-microsoft-grouppolicy), October 2016. Accessed: 16.05.2019.
- [87] Corey Sanders. Introducing B-Series, our new burstable VM size. <https://azure.microsoft.com/de-de/blog/introducing-b-series-our-new-burstable-vm-size/>, September 2017. Accessed: 07.06.2019.
- [88] Elasticsearch. Transfer von Windows-Ereignisprotokollen leicht gemacht. <https://www.elastic.co/de/products/beats/winlogbeat>, June 2019. Accessed: 07.06.2019.
- [89] Roberto Rodriguez - Cyb3rWard0g. HELK. <https://github.com/Cyb3rWard0g/HELK>, June 2019. Accessed: 07.06.2019.
- [90] Microsoft. Network security: Configure encryption types allowed for Kerberos. <https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-security-configure-encryption-types-allowed-for-kerberos>, April 2017. Accessed: 11.06.2019.
- [91] Microsoft. Code metrics values. <https://github.com/SwiftOnSecurity/sysmon-config>, November 2018. Accessed: 03.06.2019.
- [92] SwiftOnSecurity. sysmon-config. <https://github.com/SwiftOnSecurity/sysmon-config>, May 2019. Accessed: 03.06.2019.
- [93] GovCERT.ch. APT Case RUAG (Espionage Case at RUAG). Technical report, MELANI:GovCERT, May 2016.
- [94] Jessica Payne. Monitoring what matters – Windows Event Forwarding for everyone (even if you already have a SIEM.). <https://blogs.technet.microsoft.com/jepayne/2015/11/23/monitoring-what-matters-windows-event-forwarding-for-everyone-even-if-you-already-have-a-siem/>, November 2015. Accessed: 25.05.2019.
- [95] Microsoft. Use Windows Event Forwarding to help with intrusion detection. <https://docs.microsoft.com/en-us/windows/security/threat-protection/use-windows-event-forwarding-to-assist-in-intrusion-detection>, February 2019. Accessed: 07.06.2019.
- [96] K. Jaganathan, L. Zhu, J. Brezak. The RC4-HMAC Kerberos Encryption Types Used by Microsoft Windows. <https://tools.ietf.org/html/rfc4757>, December 2006. Accessed: 11.06.2019.

## Part II

# Appendix

# Testing and Code Metrics

This section describes in which ways the Readinizer was tested. Furthermore, the most important code metrics are shown and analyzed.

## Testing

### System Tests

Due to cost reasons, the existing test environment was used as the basic scope for the system test. Within this network, a new client was set up in each of two different domains. Both clients run on the operating system Windows 10 Version 1709. Before the testing began the two needed dependencies were installed, the SQL Server 2017 LocalDB as well as the Windows Remote Server Administration Tool. On one client the Readinizer was installed via the installer, on the other client the portable application was used. Neither of the distribution types encountered any difficulties.

For the system test the following protocol was used:

<b>Name</b>	<b>ST1.0 - Execute Readinizer</b>
<b>Test objective</b>	The Readinizer application starts
<b>Preconditions</b>	<ul style="list-style-type: none"><li>• SQLLocalDB &amp; RSAT is installed</li><li>• User can execute as administrator</li></ul>
<b>Execution</b>	<ol style="list-style-type: none"><li>1. Open the Readinizer.exe</li><li>2. Allow the Readinizer to make changes</li></ol>
<b>Expectation</b>	Readinizer starts up, either the home view or the last result is displayed.
<b>Result</b>	<b>Fulfilled</b>

Table A.1: ST1 - Execute Readinizer

<b>Name</b>	<b>ST2.0 - Analyze a specified domain</b>
<b>Test objective</b>	The security settings of a specified domain in the forest are collected and analyzed
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>
<b>Execution</b>	<ol style="list-style-type: none"> <li>1. Provide a domain which is in the forest</li> <li>2. Click the “Analyze Readiness”-button</li> </ol>
<b>Expectation</b>	Readinizer runs and collects the data of the specified domain. The forest overview opens and displays the analyzed data, the “Sysmon”-button is NOT visible.
<b>Result</b>	<b>Fulfilled</b>

Table A.2: ST2.0 - Analyze a specified domain

<b>Name</b>	<b>ST2.1 - Domain Overview</b>
<b>Test objective</b>	The domain overview opens
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• ST2.0 successfully executed</li> </ul>
<b>Execution</b>	<ol style="list-style-type: none"> <li>1. Click the “...”-button next to the domain name</li> </ol>
<b>Expectation</b>	The domain overview opens and displays the readiness in a pie chart as well as lists the “good” and the “bad” groups of identical security settings.
<b>Result</b>	<b>Fulfilled</b>

Table A.3: ST2.1 - Domain Overview

<b>Name</b>	<b>ST2.2 - GISS Overview</b>
<b>Test objective</b>	The GISS overview opens
<b>Preconditions</b>	<ul style="list-style-type: none"><li>• ST2.1 successfully executed</li></ul>
<b>Execution</b>	<ol style="list-style-type: none"><li>1. Click a random GISS name of either the “good” or “bad” list</li><li>2. Check audit settings against the “predicted” audit settings</li></ol>
<b>Expectation</b>	The GISS overview opens and displays security settings, which match the predicted security settings. Furthermore, a list of organizational units in this GISS are displayed.
<b>Result</b>	<b>Fulfilled</b>

Table A.4: ST2.2 - GISS Overview

<b>Name</b>	<b>ST2.3 - Organizational Unit Overview</b>
<b>Test objective</b>	The organizational unit overview opens
<b>Preconditions</b>	<ul style="list-style-type: none"><li>• ST2.2 successfully executed</li></ul>
<b>Execution</b>	<ol style="list-style-type: none"><li>1. Click a random organizational unit name</li><li>2. Review the audit settings, and the Group Policy Object responsible for it, against the “predicted” list</li></ol>
<b>Expectation</b>	The organizational unit overview opens and displays security settings and the responsible Group Policy Object, which match the predicted list.
<b>Result</b>	<b>Fulfilled</b>

Table A.5: ST2.3 - Organizational Unit Overview

<b>Name</b>	<b>ST3.0 - New Analysis</b>
<b>Test objective</b>	Readinizer is prepared for a new analysis
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>
<b>Execution</b>	1. Click on “File” in the navigation bar, select “New Analysis”
<b>Expectation</b>	All database tables are truncated and the home view opens up.
<b>Result</b>	<b>Fulfilled</b>

Table A.6: ST3 - New Analysis

<b>Name</b>	<b>ST4.0 - Analyze all domains and check Sysmon service</b>
<b>Test objective</b>	The security settings of all domains are collected and analyzed
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• ST3 successfully executed</li> </ul>
<b>Execution</b>	<ol style="list-style-type: none"> <li>1. Leaf the domain name field empty</li> <li>2. Select the toggles “Select subdomains/treedomains” and “Check Sysmon”</li> <li>3. Provide no other Sysmon name</li> </ol>
<b>Expectation</b>	Readinizer runs and collects the data of all domains in the forest. The forest overview opens and displays the analyzed data. The Sysmon button is visible.
<b>Result</b>	<b>Fulfilled</b>

Table A.7: ST4 - Analyze all domains and check Sysmon service

At this point, repeat ST2.1 - Domain Overview, ST2.2 - GISS Overview and ST2.3 - Organizational Unit Overview.

<b>Name</b>	<b>ST5.0 - Add a Organizational Unit that could not be analyzed</b>
<b>Test objective</b>	An organizational unit that could not be contacted/analyzed is inserted manually
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• ST4 successfully executed</li> <li>• At least one organizational unit could not be analyzed</li> </ul>
<b>Execution</b>	<ol style="list-style-type: none"> <li>1. Click on the “+”-button</li> <li>2. Add a previously collected RSoP as an XML file of a missing organizational unit</li> </ol>
<b>Expectation</b>	The added organizational unit either shows up in an existing GISS or creates a new one. The added organizational unit disappears in the “Unanalyzed Organizational Units” list.
<b>Result</b>	<b>Fulfilled</b>

Table A.8: ST5 - Add a Organizational Unit that could not be analyzed

<b>Name</b>	<b>ST6.0 - Sysmon Overview</b>
<b>Test objective</b>	The Sysmon overview opens
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• ST4 successfully executed</li> </ul>
<b>Execution</b>	<ol style="list-style-type: none"> <li>1. Click on the “Sysmon”-button</li> </ol>
<b>Expectation</b>	The Sysmon overview opens and displays in a pie chart the percentage of computers on which Sysmon is running in green. Beneath two list shows the name of computer on which Sysmon is running respectively not running.
<b>Result</b>	<b>Fulfilled</b>

Table A.9: ST6 - Sysmon Overview



<b>Name</b>	<b>ST7.0 - Export to JSON</b>
<b>Test objective</b>	Export to a JSON file
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>ST4 successfully executed</li> </ul>
<b>Execution</b>	1. Click on “Export” in the navigation bar, select “Export Grouped Security Settings (RSoP per OU)”
<b>Expectation</b>	The collected and analyzed data is exported into a JSON file. This JSON file is structured hierarchically and contains information about the security settings per organizational unit as well as the Readinizer’s assessment.
<b>Result</b>	<b>Fulfilled</b>

Table A.10: ST7 - Export to JSON

This test protocol was also performed on servers that already existed in the network. The same results were achieved as in the tests with the newly added clients.

## Unit Testing

Unit tests were only written for the two services **AnalysisService** and **RsopService**. These two services are not dependent on any underlying system calls and test data can be easily mocked. The rest of the business logic depends heavily on other system calls, for example on data from the Active Directory or from Resultant Set of Policy objects. Nevertheless, tests - especially with regard to further development of the product - are of great importance and therefore edge cases for both services were tested.

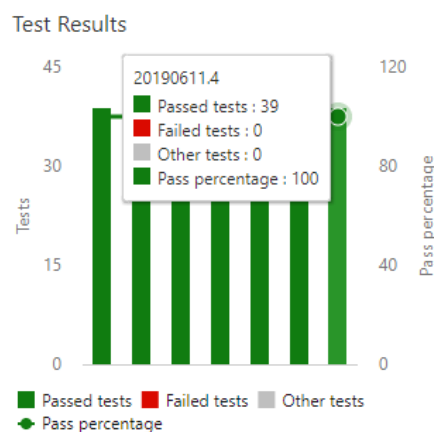


Figure A.1: Rest Result

Furthermore, no unit tests were written for the data access layer, since it makes no sense to check the already well tested and widely used Entity Framework for its functionality.

## Code Metrics

<b>Classes &amp; Interfaces</b>	83
<b>Lines of compiled Code</b>	1439
<b>Views</b>	8

Table A.11: Code metrics - Visual Studio

The following code metrics were calculated by Visual Studio:

<b>Project</b>	<b>Maintainability Index</b>	<b>Depth of Inheritance</b>	<b>Lines of Code</b>
Business	84	1	478
DataAccess	84	2	148
Domain	92	2	400
Frontend	90	9	413

Table A.12: Code metrics - Visual Studio

The **Maintainability Index** returns a value between 0 and 100 and describes the relative ease of maintaining the code. The higher the value the better.

The **Depth of Inheritance** represents the number of different classes that inherit from one another. The lower the value, the better. The high value of the frontend is due to the inheritance of the WPF View classes.

The **Lines of Code** displays the approximate number of lines in the code. This number is based on the Common Intermediate Language and does not reflect the exact lines of code in the source file [91].

# Time Management

This section describes the time management over the entire project. The project was implemented during the spring semester 2019 which lasted from 18.06.2019 to 14.06.2019. 17 semester weeks were available, whereby a time effort of about 40 hours<sup>1</sup> per week was expected. Note that the school semester lasted only 15 weeks with two additional weeks before the submission of the bachelor thesis. During these two weeks the students worked fulltime on the bachelor thesis - thus, totally 9 days à 8 hours each. This results in an estimated effort of 720 hours over the total duration of the project.

## Time by Activity Type

The project was divided into 8 different activities within the time management. The following figure shows the distribution of time per activity over the entire project.

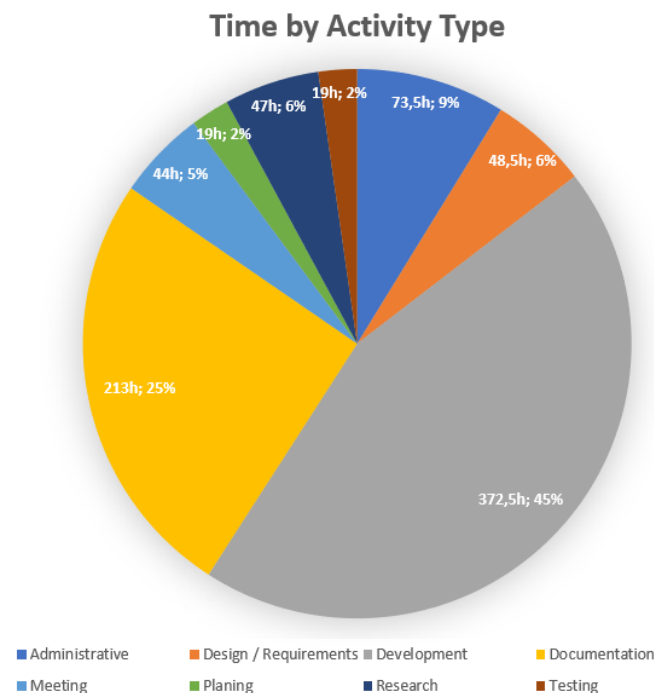


Figure B.1: Time by Activity Type

---

<sup>1</sup> 20 hours per person

# Time by Phase

## Inception

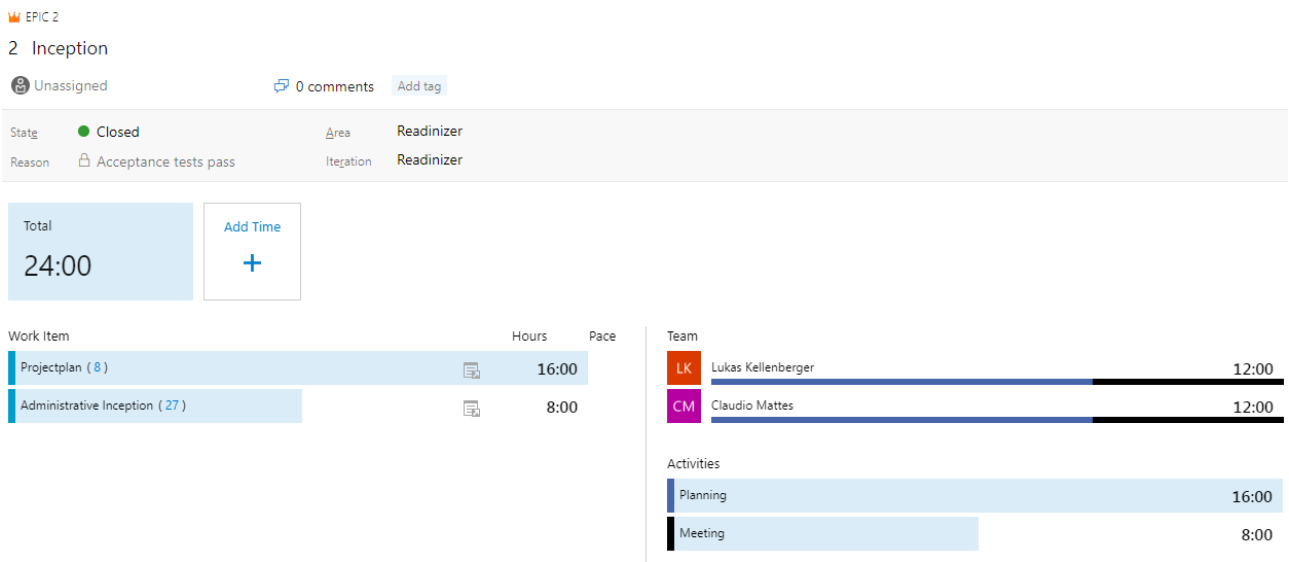


Figure B.2: Time in Inception

## Elaboration

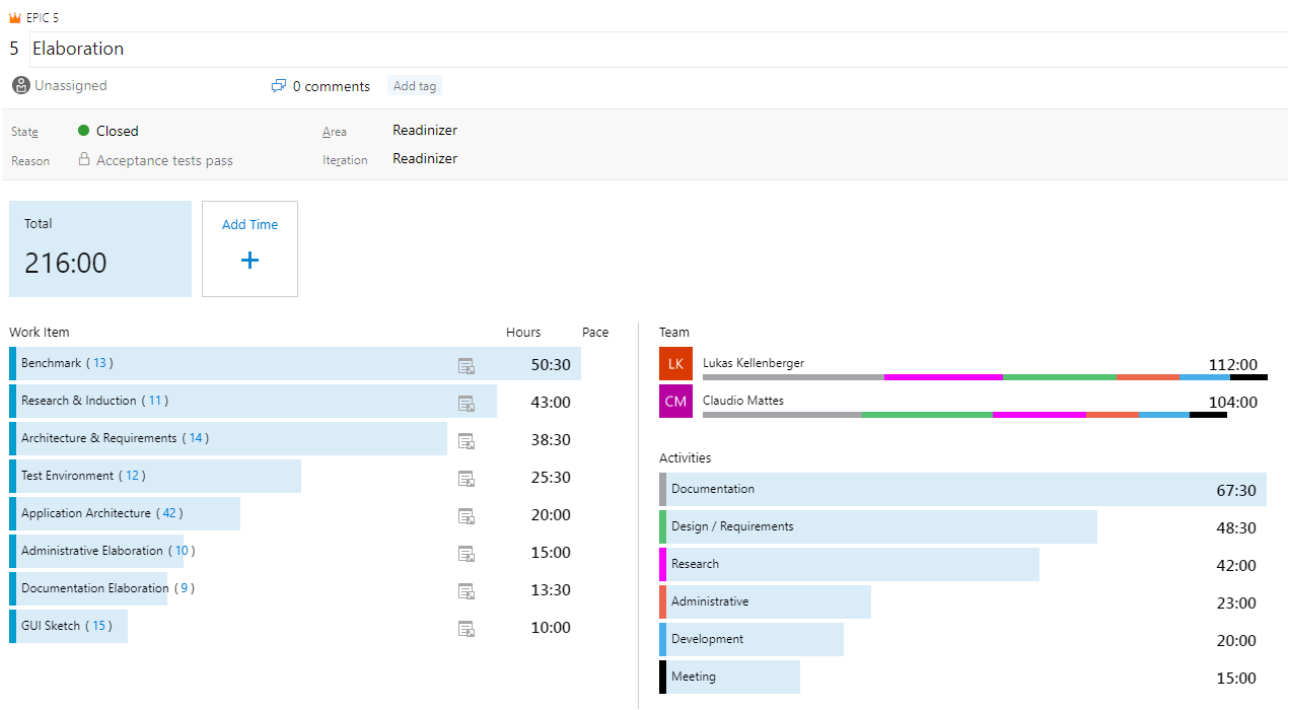


Figure B.3: Time in Elaboration

Construction

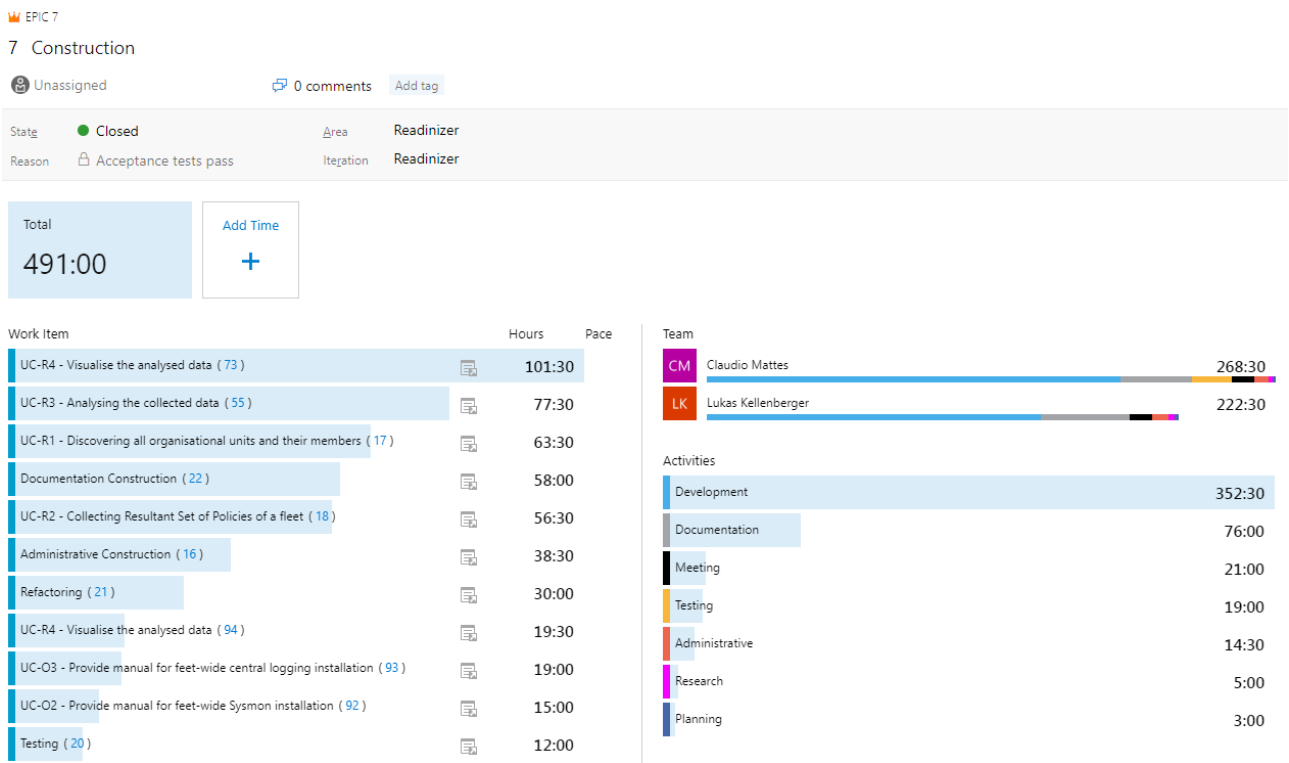


Figure B.4: Time in Construction

Transition

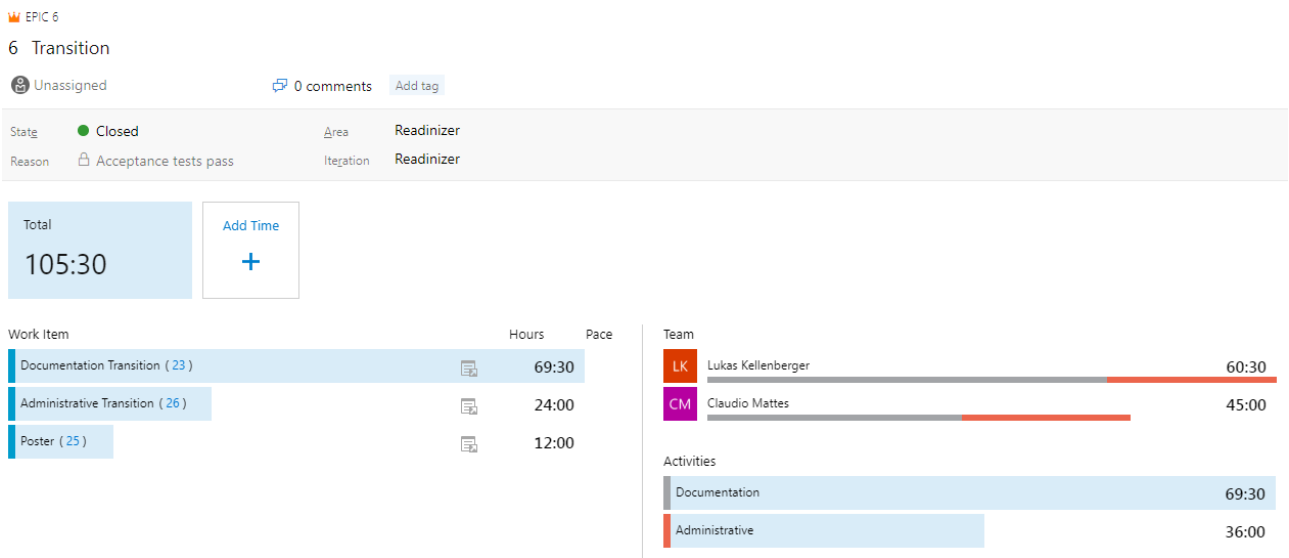


Figure B.5: Time in Transition

## Sprints - Estimated Time vs. Actual Time

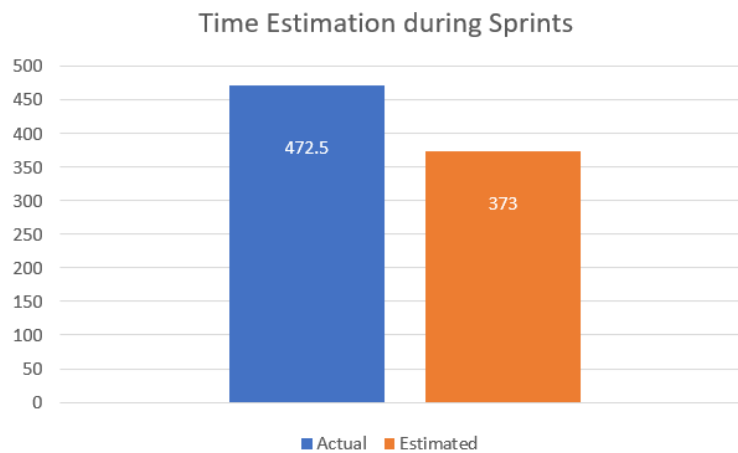


Figure B.6: Sprints - Estimated Time vs. Actual Time

## Conclusion Time Management

All in all, the estimated time was exceeded by approximately 16%. The main reason for the time overrun is that both students had relatively little experience in C-Sharp. Thus, some implementation work took longer than expected.

**Estimated Total:** 836.5 hours

**Actual Total:** 720 hours

Nevertheless, all milestones were met in the given time. Furthermore, the bachelor thesis was submitted on time. Due to the misjudgements (especially estimation and actual time) during the project, we were able to gain important experience for future projects. We are now aware of the fact that a more accurate estimate must be done which includes certain disruptions.

# 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

Im Rahmen einer Studienarbeit wurde ein Proof-of-Concept (PoC) erstellt, welcher die Readiness von Windows Computern prüfen kann. Readiness betrifft viele Aspekte und einfache Dinge wie korrekte Zeitstempel in Logs, deren Vollständigkeit oder die Bereitstellung von Backups. Der vorliegende PoC fokussiert auf Windows-Infrastrukturen und prüft dabei deren Konfiguration (Group Policy Objects, GPOs). Er berücksichtigt dabei hauptsächlich die Publikationen des japanischen Computer Emergency Response Teams (JPCERT/CC).

Das Ziel der Bachelorarbeit ist es nun, den PoC auszubauen um komplette Windows Netze (Forests, Domänen) untersuchen zu können. Zudem soll das Tool eine visuell ansprechende Auswertung (Zielpublikum: Entscheider) sowie Hilfestellung für die Verbesserung der Readiness bereitstellen

## Abgrenzung

Es geht nicht darum neue Angriffsvektoren zu finden.

## Tätigkeiten

- Projektmanagement und Dokumentation
- Einarbeitung in Incident Handling, Forensik, Angriffstechniken und Werkzeuge dafür
- Einarbeitung in Abwehrtechniken und Härtung von Systemen
- Studium öffentlicher Quellen und verfügbaren Tools
- Umsetzung eines Analyzers, Visualizer und Optimizer gemäss Anforderungen

## 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, Visualizer und Optimizer 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 Installationsprozedere ausgeführt werden.

- Definition der Requirements für den verbesserten Analyzer
  - Benchmark des PoC gegen CERT-EU, NSA, MITRE und MS Guidelines
  - Flottenfunktionalität (Forests, Domains, OE abbilden)
  - Ansprechendes User Interface
  - Ansprechender Report (XML, XSLT, PDF)
- Mögliche Themen für einen Optimizer
  - Policy Files generieren
  - Sysmon Install Package
  - Simple Central Logging (bspw. gemäss WEFFLES)
  - Performance/Logmenge Netzload
- Design, Architektur und Implementation
- Webseite mit Download und Benutzerhandbuch
- 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 (UseCases, 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: \skripte\Informatik\Fachbereich\Bachelor-Arbeit\_Informatik Es muss aus den abgegebenen Dokumenten klar hervorgehen, wer für welchen Teil der Arbeit und der Dokumentation verantwortlich war (detaillierte Zeiterfassung).

## Termine

Datum	Task
18.02.2019	Beginn der Bachelorarbeit, Ausgabe der Aufgabenstellung durch den Betreuer.
07.06.2019	Die Studierenden geben den Abstract für die Diplomarbeitsbroschüre zur Kontrolle an ihren Betreuer/Examinator frei. Die Studierenden erhalten vorgängig vom Studiengangsekretariat die Aufforderung mit den Zugangsdaten zur Online-Erfassung des Abstracts im DAB-Tool.  Die Studierenden senden per Email das A0-Poster zur Prüfung an ihren Examinator/Betreuer. Vorlagen sowie eine ausführliche Anleitung betreffend Dokumentation stehen auf dem Skripteserver zur Verfügung.
12.06.2019	Der Betreuer/Examinator gibt das Dokument mit dem korrekten und vollständigen Abstract der Broschüre zur Weiterverarbeitung an das Studiengangsekretariat frei.  Für die Ausstellung der Bachelorarbeiten das A0 Posters per Email bis 10.00 Uhr an das Studiengangsekretariat senden.
14.06.2019	Hochladen aller verlangten Dokumente auf archiv-i.hsr.ch Abgabe des Berichts an den Betreuer bis 12.00 Uhr
14.06.2019	Präsentation und Ausstellung der Bachelorarbeiten, 16 bis 20 Uhr
05.08.2019 bis 23.08.2019	Mündliche BA-Prüfung
27.09.2019	Bachelorfeier

## 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 Brunswiler betreut. Der Gegenleser ist noch nicht bestimmt.

## Kontakt

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

+41 55 214 41 73

cyrill.brunswiler@compass-security.com

cyrill.brunswiler@hsr.ch

## Unterschriften

Jona, 19. März 2019



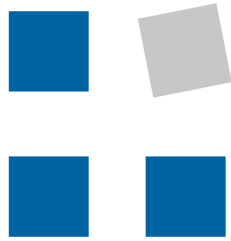
Cyrill Brunswiler



Claudio Mattes



Lukas Kellenberger



**HSR**  
**HOCHSCHULE FÜR TECHNIK**  
**RAPPERSWIL**

**COMPUTER SCIENCE**

# Optimization

SYSMON DEPLOYMENT THROUGH GPO

## **Authors:**

Claudio MATTES  
claudio.mattes@hsr.ch

Lukas KELLENBERGER  
lukas.kellenberger@hsr.ch

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

June 12, 2019

# General Information

## 1.1 Overview

This manual is a step by step guide on how to install Sysmon on a Windows domain without the use of an automatic software deployment tool. To achieve this goal, Sysmon is deployed through Group Policy Objects.

Sysmon is a monitoring service that logs events such as process creation, network connections and file access and changes. Sysmon logs events which Windows does not log and/or does this in a much more detailed way.

A network folder will be created to which each client has access. Three files are stored in this folder:

- **Sysmon Executable:**  
The regular Sysmon executable.
- **Sysmon Configuration File:**  
A XML-file which contains the configuration that will be applied to the Sysmon service.
- **Batch File:**  
The batch file will be executed remotely and check whether the Sysmon service is all ready installed and running. If this is not the case, it will install Sysmon on the computer.

A Group Policy Object will be created and applied to the domain. Within this GPO a Scheduled Task is set, which will execute the batch file in regular, defined intervals.

This manual was developed during a bachelor thesis by the two Bachelor of Science in Computer Science students, Claudio Mattes and Lukas Kellenberger.

## 1.2 Organization of the Manual

The user manual consists of four parts:

- **General Information:**  
The General Information section explains the manual and the purpose for which it is intended.
- **Sysmon:**  
The Sysmon section explains what Sysmon is and why it should be installed.
- **Requirements and Limitations:**  
The Requirements and Limitations section defines which requirements are needed to be able to deploy Sysmon and which systems this guide is limited to.
- **Implementation:**  
The Implementation section is the actual guideline, it is a step by step guide on how to install Sysmon on an entire fleet.

# Sysmon

## 1.1 What is Sysmon?

Sysmon is a monitor service developed by Mark Russinovich and Thomas Garnier. They describe Sysmon as followed:

*“System Monitor (Sysmon) is a Windows system service and device driver that, once installed on a system, remains resident across system reboots to monitor and log system activity to the Windows event log. It provides detailed information about process creations, network connections, and changes to file creation time.” [52]*

## 1.2 Why Sysmon?

“Sysmon logs several events on the system which are partly logged by default too. For example, the event “A new process has been created” with the identifier (ID) 4688 is logged by Sysmon with the ID 1 “Process Creation” . The problem is that the default logged event with the ID 4688 logs only the executable file (EXE) name as well as the including path. But attackers want to stay below the radar, so they might replace the original EXE a with malicious one and rename it like the original. Hence, there is no way to determine with the system based event log entry 4688 if the original EXE was executed. Sysmon eliminates exactly this gap by logging not only the name and path of the EXE but also the hash value of the EXE. Ergo Sysmon brings a big advantage to detect if a malicious EXE was executed or not. Therefore a reference hash value of the executed EXE is required to compare the hash values on its correctness.” [3, p. 11]

More detailed information can be found at:

<https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon>

# Requirements and Limitations

## 1.1 Requirements

Sysmon runs on clients with the operating system “Windows 7” and higher, and on servers with “Windows Server 2008 R2” and higher. Moreover, to run the Scheduled Task “Windows 7” is the minimum required version.

The user needs permission to create a network folder and edit files in it.

The user needs access to the Active Directory and must be able to create and set a Group Policy Object on a domain.

## 1.2 Limitations

This guide was only tested on “Windows 10” and “Windows Server 2016” and is therefore limited to these operating systems. Theoretically this guide should work for all operating systems from Windows 7 and higher, or Windows Server 2008 R2 and higher. This however, was not tested for this guide.

# Implementation

## 1.1 Domain Folder

Create a folder in your domain that is accessible for every client on your domain. The SYSVOL folder would be a good choice.

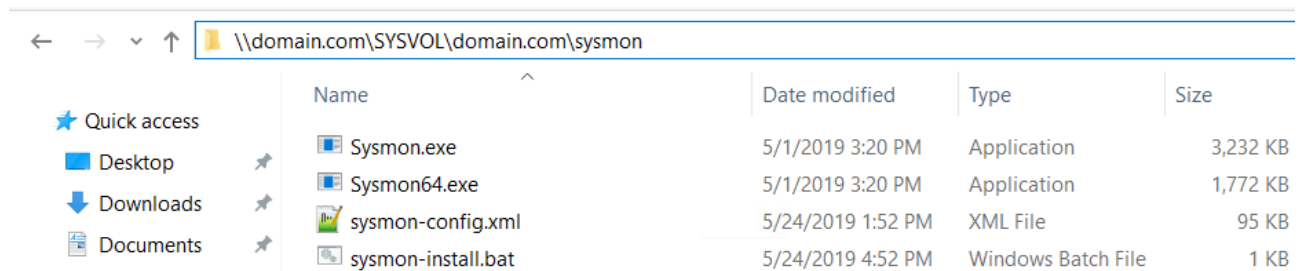


Figure G.1: SYSVOL-folder

If another folder is used, it is important to keep the access permissions restrictive. An attacker otherwise has the option of exchanging or modifying the files, which are then distributed across the entire domain.

## 1.2 Sysmon Executable

Add a Sysmon executable to the created domain folder, can be downloaded here:  
<https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon>

**(Optional) Change service name** For security reasons it is recommended to rename the Sysmon service to make it more difficult for an attacker to detect Sysmon. To do so, just change the name of the executable, for example to “DefinitelyNotSysmon.exe”. This will then be the name of the service.

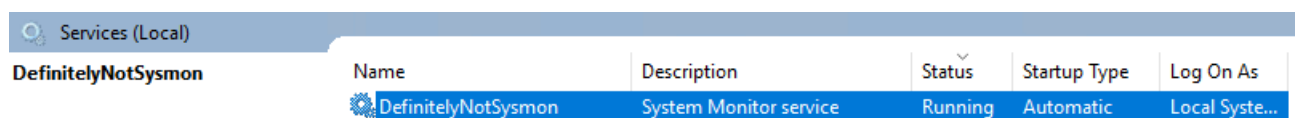


Figure G.2: DefinitelyNotSysmon Service

**Attention!** If the name of the executable is changed, the batch file has to be adjusted as well.

## 1.3 Sysmon Configuration File

The Sysmon configuration is used to make deployment easier and to filter captured events. To get started, a Sysmon sample configuration from SwiftOnSecurity [92] can be used:  
<https://github.com/SwiftOnSecurity/sysmon-config>.

## 1.4 Batch File

The batch file is the logical component of the deployment. It checks if Sysmon is already installed and if the service is running or not. This is the code contained in the batch file.

Listing G.1: Sysmon Installation Batch File

```
1  @echo off
2
3  SC QUERY Sysmon > NUL
4  IF ERRORLEVEL 1 GOTO MissingInstallSysmon
5  ECHO Exists
6
7  FOR /F "TOKENS=3 DELIMS=: " %%S in ('SC QUERY "Sysmon" ^| FINDSTR "STATE"') DO (
8      IF /I "%%S" NEQ "RUNNING" (
9          ECHO Sysmon is not running
10         net start Sysmon
11         GOTO EOF)
12
13     ECHO Sysmon is running
14
15 )
16 GOTO EOF
17
18 :MissingInstallSysmon
19 ECHO Is missing, sysmon is beeing installed
20 IF NOT EXIST "C:\sysmon" (mkdir "C:\sysmon" & copy /v
21     "\\domain.com\SYSVOL\domain.com\sysmon\config.xml" "C:\sysmon\config.xml")
22     "\\domain.com\SYSVOL\domain.com\sysmon\sysmon.exe" -accepteula -i
23     C:\sysmon\sysmon-config.xml
24 GOTO EOF
25
26 :EOF
27 END && EXIT
```

### 1.4.1 What does it do...

#### Check if this service exists

Listing G.2: Check if service exists

```
1  SC QUERY Sysmon > NUL
2  IF ERRORLEVEL 1 GOTO Missing
3  ECHO Sysmon exists
```

This piece of code checks if the service with the name Sysmon exist. If the service is not found, the errorlevel is set to 1 and the “Service is missing and needs to be installed” code is executed. If the service is found, the errorlevel is set to 0 and the message Sysmon exists will display. The code continues to run and will check as in the sector “Check the status of the service” shows the status of the service.



## Check the status of the service

Listing G.3: Check if Service exists

```
1  FOR /F "TOKENS=3 DELIMS=: " %%S IN ('SC QUERY "Sysmon" ^| FINDSTR "STATE"') DO (  
2    IF /I "%%S" NEQ "RUNNING" (  
3      ECHO Sysmon is not running  
4      GOTO StartService  
5      net start Sysmon  
6      GOTO EOF  
7    )  
8  
9    ECHO Sysmon is running  
10 )  
11 GOTO EOF
```

First the properties of the service are loaded and the property with the name “State” is searched. They are then provided with tokens and the third token, which contains the state, is compared with the string “Running”. If the service exists, but is not yet running, the service is started with the command “net start Sysmon” and then the batch file is terminated. Otherwise the service is running and the batch file is terminated directly.

## Service is missing and needs to be installed

Listing G.4: Installing Sysmon

```
1  :MissingInstallSysmon  
2  ECHO Is missing, sysmon is beeing installed  
3  IF NOT EXIST "C:\sysmon" (mkdir "C:\sysmon" & copy /v  
4    "\\domain.com\SYSVOL\domain.com\sysmon\config.xml" "C:\sysmon\config.xml")  
5    "\\domain.com\SYSVOL\domain.com\sysmon\sysmon.exe" -accepteula -i C:\sysmon\config.xml  
6  GOTO EOF
```

First it is checked whether the directory “C:\sysmon” exists. If this is not the case, this directory is created. The XML file “sysmon-configuration” is then copied into that directory. Subsequently the “Sysmon.exe” file is executed and Sysmon is installed with the defined configuration from the copied “sysmon-config” file.

## 1.5 Group Policy Object

Create a new Group Policy on your domain controller and link it to your domain. Right click on the newly created GPO and choose “Edit”. Then navigate to **Computer Configuration** -> **Preferences** -> **Control Panel Setting**. Right click on Scheduled Tasks or inside the Scheduled Tasks window, select “New” and choose “Scheduled Task (At least Windows 7)”.

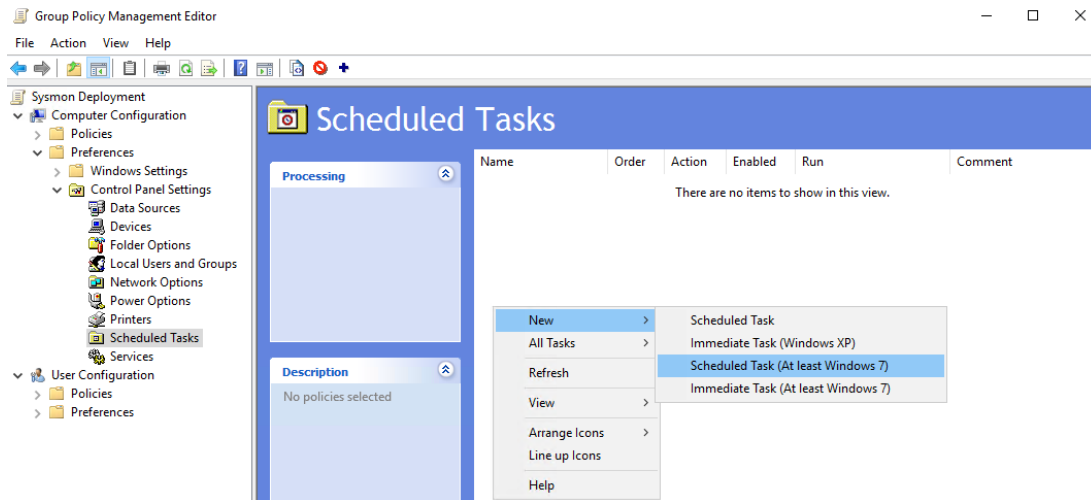


Figure G.3: Create Scheduled Tasks

Now configure the general settings, give the task a name and choose a user which will execute the task. We recommend to use the user “System”, to install or start the Sysmon service administrator rights are needed. Set “Configure for:” to “Windows 7, Windows Server 2008R2”, this setting also worked with Windows 10 and Windows Server 2016 during our tests.

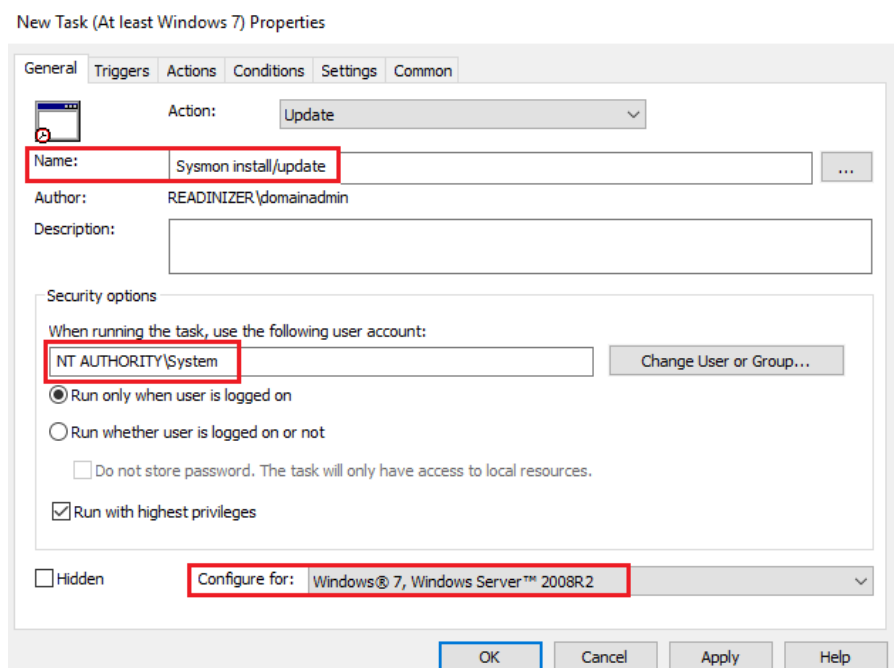


Figure G.4: Scheduled Task - General

Change to the tab “Trigger”. Click on “New” to generate a new Trigger for the task. With this settings it is checked once a day if Sysmon is installed or if there is an update.

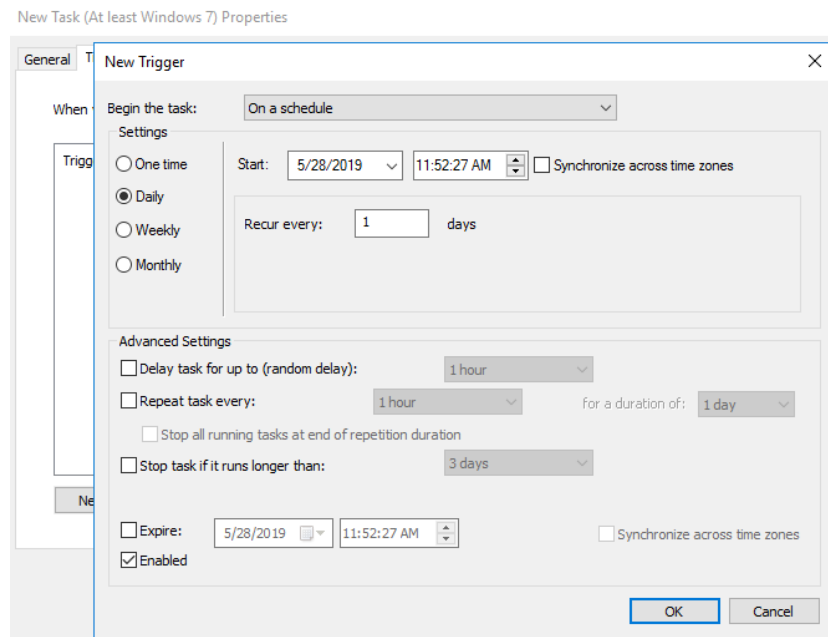


Figure G.5: Scheduled Task - Trigger

On the tab “Action” it is defines what to do. Click on “New”, set the “Action” to “Start a program” and enter the path to the sysmon-install.bat file on the domain folder.

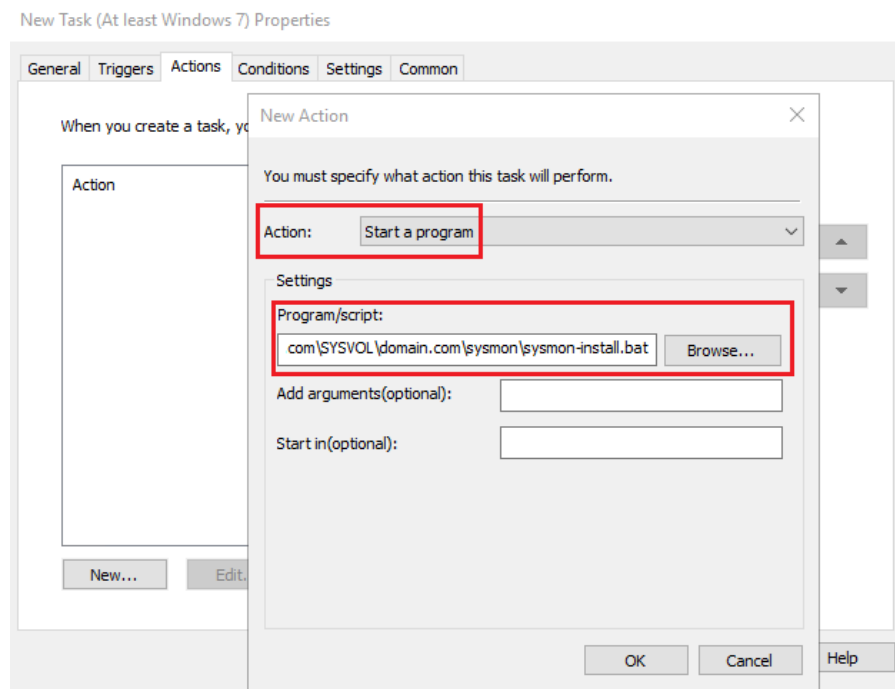


Figure G.6: Scheduled Task - Action

(Optional) Enable the “Allow task to be run on demand” to manually trigger the task on an computer. This makes testing easier.

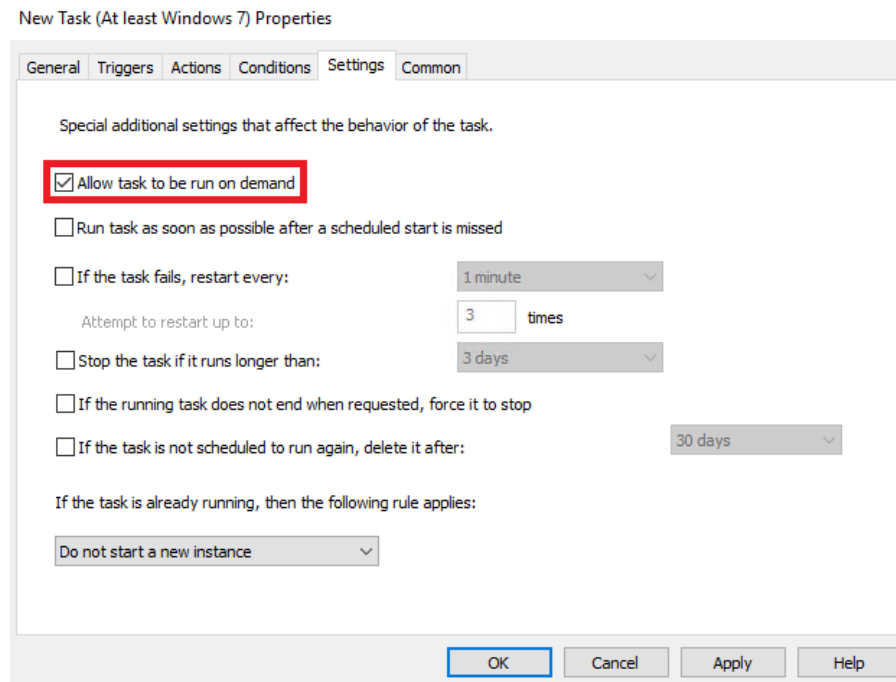
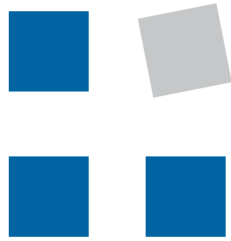


Figure G.7: Scheduled Task - Action



**HSR**  
**HOCHSCHULE FÜR TECHNIK**  
**RAPPERSWIL**

**COMPUTER SCIENCE**

# Optimization

WINDOWS EVENT FORWARDING DEPLOYING FLEET-WIDE

## **Authors:**

Claudio MATTES  
claudio.mattes@hsr.ch

Lukas KELLENBERGER  
lukas.kellenberger@hsr.ch

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

June 12, 2019

# General Information

## 1.1 Overview

This manual describes step by step how Windows Event Forwarding (WEF) - also known as central logging - can be integrated over an entire Windows domain.

A Windows Event Collector (WEC) is installed, which is responsible for the collection of all logs of the clients. A GPO is defined which allows the selected clients to send their logs to the WEC. The subscription is described, which defines what to log.

## 1.2 Organization of the Manual

The user manual consists of five parts:

- **General Information:**  
The General Information section explains the tool and the purpose for which it is intended.
- **Windows Event Forwarding (WEF):**  
This section describes what WEF actually is and the advantages of using it.
- **Requirements and Limitations:**  
The Requirements and Limitations section defines which requirements are needed to be able to deploy WEF and what limitations apply.
- **Deployment:**  
The Deployment section describes in detailed steps how a central logging is deployed over a entire fleet.
- **Appendix:**  
Additional information about which event log ids will be logged with the recommended subscription as well as the recommended subscription itself.

# Windows Event Forwarding

## 1.1 What is Windows Event Forwarding?

Windows Event Forwarding (WEF) allows system administrators that logs are no longer stored on individual clients and servers (further referred as only clients) within the organisation, but centrally on a server. A Windows Event Collector (WEC) server is defined as the central instance responsible for collecting the client logs. The event logs are written on the individual clients and then forwarded to the WEC.

On the WEC, subscriptions can be created for the clients, which define which event logs the clients should forward to the WEC. WEF subscription can be set up as push or pull procedure. In principle, however, the pull procedure should not be used, as the WEC queries all clients for their event logs that have not yet been sent. This means that at certain times the network is stressed by many clients. In contrast, the push procedure does not stress the network as much as the clients themselves decide when to send the event logs to the WEC.

## 1.2 Advantages with WEF

WEF is a passive system with regard to event logging, which ensures the completeness and a longer lifetime of the event logs. Even with WEF, events are still logged on clients and servers, but forwarded to the central instance. This in turn allows a much faster forensic analysis in case of advanced persistent threat (APT) or lateral movement - conventional event logging (like specific application logs) can also be stored centrally. With the extended lifetime of event logs, APTs can be better tracked and analyzed. From the technical report on the “RUAG cyber espionage case”<sup>1</sup> it is clear that a long lifetime of log files can improve a complete forensic analysis:

*Unfortunately, log files at RUAG only go back until September 2014, where we still see C&C activity. Additionally, many suspicious devices have been reinstalled in the meantime; Hence we cannot determine the initial attack vector. [93, p. 5]*

This manual is mostly based on Jessica Paynes article “Monitoring what matters – Windows Event Forwarding for everyone (even if you already have a SIEM.)” [94]

---

<sup>1</sup>Further information about the espionage case available in the technical report: [https://www.melani.admin.ch/melani/en/home/dokumentation/reports/technical-reports/technical-report\\_apr\\_case\\_ruag.html](https://www.melani.admin.ch/melani/en/home/dokumentation/reports/technical-reports/technical-report_apr_case_ruag.html)

# Requirements and Limitations

## 1.1 Requirements

A WEC environment can be deployed on any “Windows 10” or “Windows Server 2012R2” system and above. However, it is highly recommended to use a “Windows Server” with enough disk space. Furthermore, it is recommended to use disks which have a “high speed” write capability to increase the number of events per second that a one WEC can handle.

## 1.2 Limitations

The following limitations have been derived from microsoft <sup>1</sup>

- There are no recommendations in this manual for disk sizes, as this can vary greatly depending on the number of clients within the network.
- A WEC can only handle a limited number of clients due its limitation of available TCP ports. Therefore, the number of clients which subscribe to a single WEC must be considered.
- The registry size of the WEC can increase to an unmanageable size over time. Because for every client - which connects to a WEF subscription - a registry key is created in order to store bookmark and source heartbeat information. Unfortunately, inactive or no longer existing clients are not removed. A quote from Microsoft in this regard:
  - *When a subscription has >1000 WEF sources connect to it [...] Event Viewer can become unresponsive for a few minutes when selecting the Subscriptions node in the left-navigation, but will function normally afterwards.*
  - *At >50,000 lifetime WEF sources, Event Viewer is no longer an option and wecutil.exe (included with Windows) must be used to configure and manage subscriptions.*
  - *At >100,000 lifetime WEF sources, the registry will not be readable and the WEC server will likely have to be rebuilt. [95]*

## 1.3 Additional Information

- WEF can handle VPN, RAS and DirectAccess connected clients
- The clients local event log acts as a buffer in case of connection loss
- Supports IPv4 and IPv6
- In a Active Directory environment there is no need for additional settings to encrypt the events which will be sent to the WEC. By default the events are encrypted using Kerberos (with NTLM as a fallback option). More information see 2 Encryption of Event Logs

---

<sup>1</sup>Further information about WEF deployment: <https://docs.microsoft.com/en-us/windows/security/threat-protection/use-windows-event-forwarding-to-assist-in-intrusion-detection>



# Deployment

## 1.1 Windows Event Collector

Choose a system (preferably a “Windows Server 2012R2” or above) for the WEC.

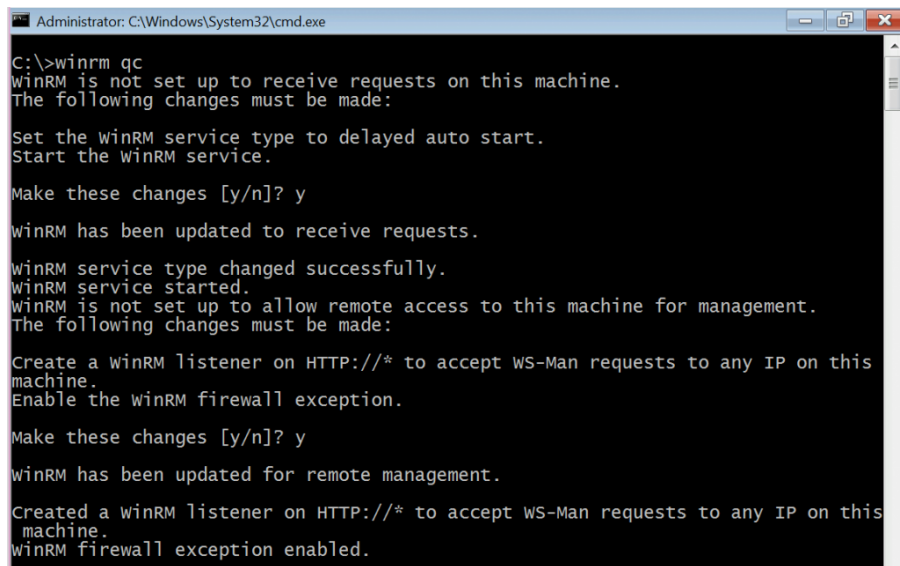
**NOTE:** Keep in mind that the logs should not be made accessible to all users! In the best case, a dedicated server to which only certain users have access is used as the WEC.

### 1.1.1 Enable WinRM

To be able to receive events the Windows Remote Management Service (WinRM) must be enabled.

Run an administrative command prompt and execute the following command: `winrm qc`

Answer the followed two questions `Make these changes[y/n]` with yes.



```
Administrator: C:\Windows\System32\cmd.exe
C:\>winrm qc
WinRM is not set up to receive requests on this machine.
The following changes must be made:

Set the winRM service type to delayed auto start.
Start the winRM service.

Make these changes [y/n]? y
WinRM has been updated to receive requests.

winRM service type changed successfully.
winRM service started.
WinRM is not set up to allow remote access to this machine for management.
The following changes must be made:

Create a winRM listener on HTTP://* to accept WS-Man requests to any IP on this
machine.
Enable the winRM firewall exception.

Make these changes [y/n]? y
WinRM has been updated for remote management.

Created a winRM listener on HTTP://* to accept WS-Man requests to any IP on this
machine.
WinRM firewall exception enabled.
```

Figure K.1: Enable WinRM

Do not close the administrative command prompt yet because it is needed in a further step.

### 1.1.2 Enable Event Forwarding

The next step is to enable “Event Forwarding”. Open the “Event Viewer” either by enter **WIN + x** or by open “Run” (**WIN + r**) and enter **eventvwr**.

In the “Event Viewer” click on “Subscriptions” and confirm with yes.

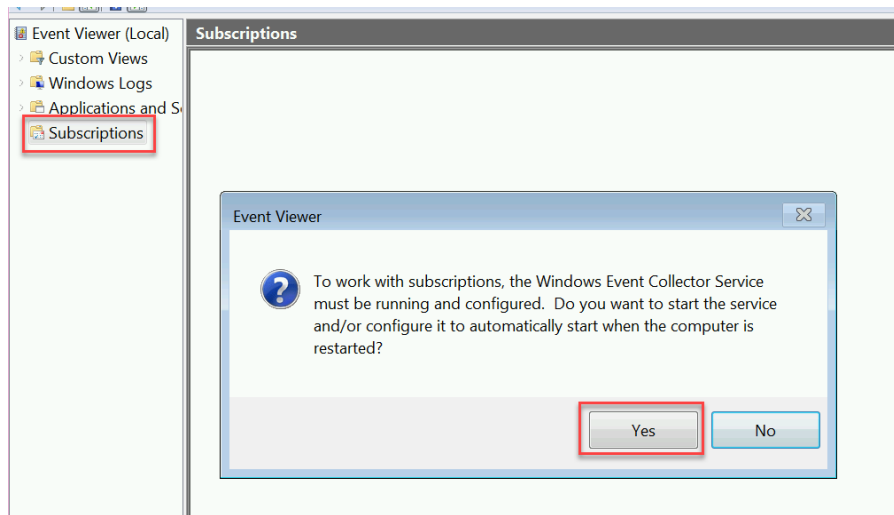


Figure K.2: Enable Subscriptions in the Event Viewer

### 1.1.3 Group Policy Objects for the subscribers

In order for the clients to also send their events to the configured WEC, a corresponding Group Policy Object (GPO) must be defined. For the creation of this GPO, however, information is required in advance.

Use the administrative command prompt and execute the following command: **wevtutil gl security**

This command will give us the information about the “Security Event Log” where the permissions on the log are stored. Copy out **channelAccess** from **0:BAG:SYD...** through the last parenthesis and put it into a Notepad:

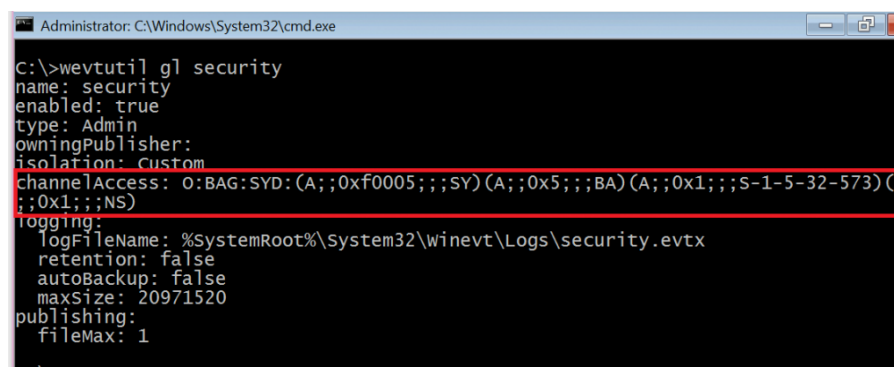


Figure K.3: Information about the Security Event Log

If your output does not include **(A;;0x1;;;NS)** at the end - like depicted - append it to your temporary stored one in Notepad.

1. Register your created WEC for the clients:

- Edit the GPO and go to: Computer Configuration > Policies > Admin Templates>Windows Components > Event Forwarding > Configure target subscription manager
- Enter: Server=http://<FQDN-WEC>:5985/wsman/SubscriptionManager/WEC,Refresh=n
- Modify <FQDN-WEC> with the FQDN of your created WEC
- Modify the refresh interval Refresh=n where n is in seconds (e.g. Refresh=1800 means that every 30minutes the clients will check for new subscriptions)



- Edit the GPO and go to: **Computer Configuration > Policies > Admin Templates > Windows Components > Event Log Service>Security > Configure log access**
- Enter the temporary stored access string (0:BAG:SYD...)



3. Configure the “WinRM” Service to be started automatically, so the clients are able to send their events to the WEC.

- Edit the GPO and go to: **Computer Configuration > Preferences > Control Panel Settings > Services**
- Right click on **Services** and create a new Service (**New > Service**)
- Set the “Startup” to “Automatic”
- Set the “Service action” to “Start service”

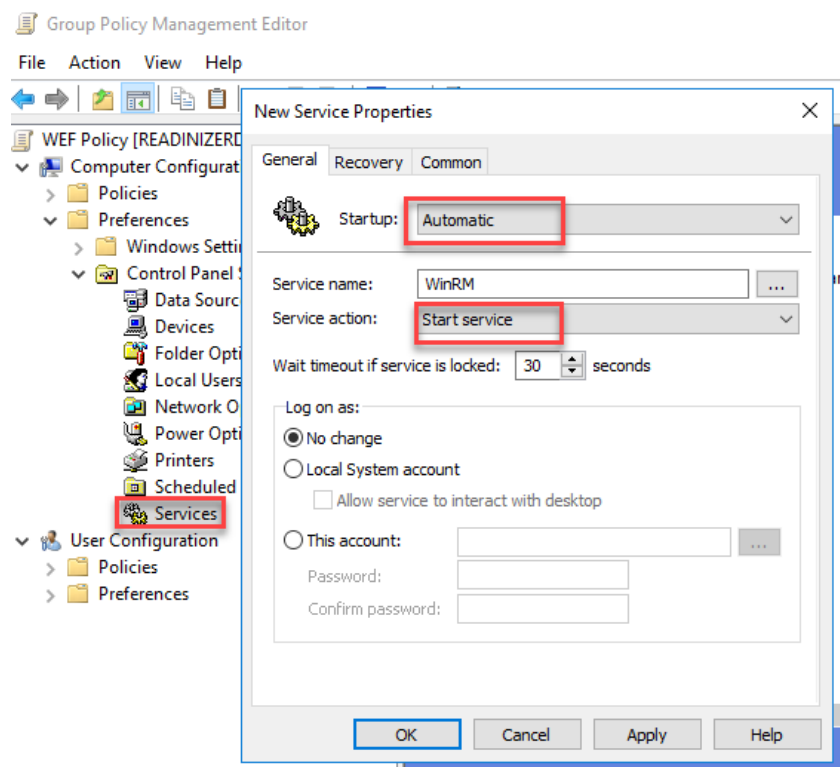


Figure K.6: Enable WinRM Service

When everything is set, your GPO should look like this summary:

ScopeDetailsSettingsDelegation

Policies

Windows Settings

Security Settings

Administrative Templates

Policy definitions (ADMX files) retrieved from the local computer.

Windows Components/Event Forwarding

Policy	Setting	Comment
Configure target Subscription Manager	Enabled	
SubscriptionManagers		
Server=http://readinizer:5985/wsman/SubscriptionManager/WEC.Refresh=60		

Windows Components/Event Log Service/Security

Policy	Setting	Comment
Configure log access	Enabled	
Log Access		
O:BAG-SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-573)(A;;0x1;;;NS)		

Preferences

Control Panel Settings

Services

Service (Name: WinRM)

WinRM (Order: 1)

General

Service name	WinRM
Action	Start service
Startup type:	Automatic
Wait timeout if service is locked:	30 seconds
Service Account	
Log on service as:	No change
Recovery	

Figure K.7: GPO Summary

Apply this GPO to all your computers which you want to forward their events to the WEC.

#### 1.1.4 WEF Subscription

The last step is to create a proper subscription that defines which events the clients will forward to the WEC. Therefore, a template was defined according to the bachelor thesis. The included event log IDs can be shown in the Appendix A - Event Log IDs.

There are two ways for defining a subscription - either you configure your subscription through the GUI or you import the subscription as a XML-file via the command line. This document describes both ways to give a good insight and to ensure maintainability when adjustments are needed.

##### 1. Define a subscription through the GUI

- Open the “Event Viewer” either by enter **WIN + x** or by open “Run” (**WIN + r**) and enter **eventvwr**.
- Click “Create Subscription” within the tab “Actions” on the left hand side of the Event Viewer.
- Enter a proper name for your new subscription and choose if the events get pulled (“Collector initiated”) or pushed (“Source computer initiated” - recommended).

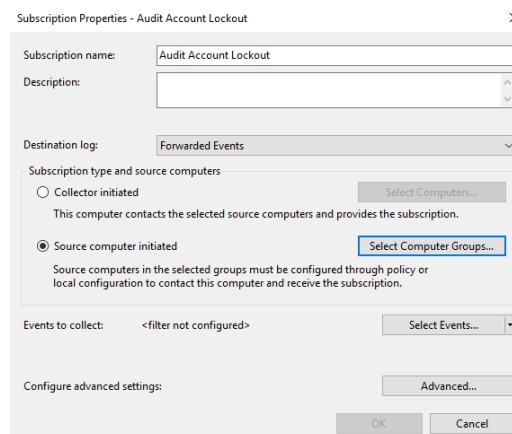


Figure K.8: Subscription Properties

- Click on “Select Computer Groups...” and choose which computers should follow their events to the WEC. In our template the groups “Domain Controllers” and “Domain Computers” are included, which should cover a wide part of the Active Directory.

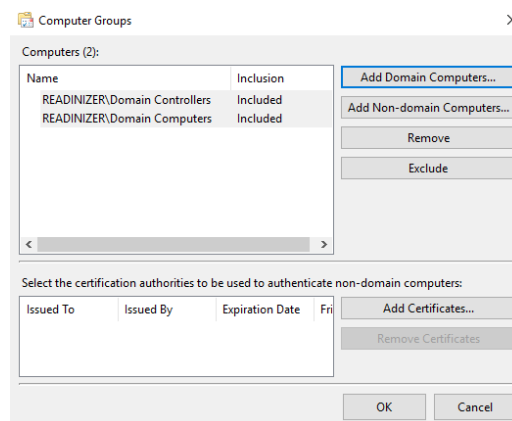


Figure K.9: Subscription Computer Groups

- Within the next step we will configure the query filter which defines the events that get forwarded by the subscribed clients, therefore open “Select Events..”. Select the setting “By log” and then “Security” or whatever you want to log. Afterwards select the Event ID which you want to track. Click “OK” when finished.

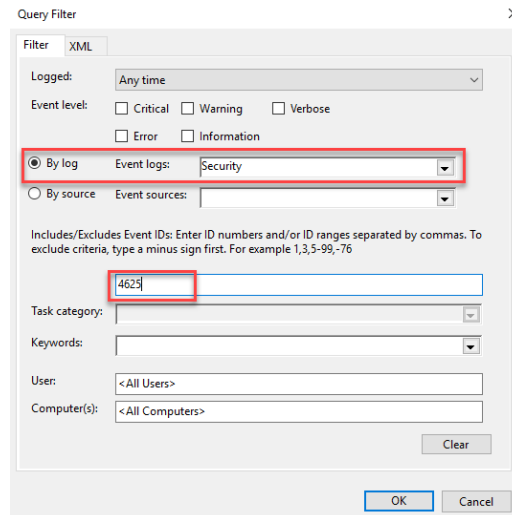


Figure K.10: Subscription Query Filter

- The last step is to define the delivery mode. You can select between the following options:
  - Normal: Ensures reliability and does not conserve bandwidth (batch timeout 15 minutes with 5 items)
  - Minimize bandwidth: Ensures bandwidth (batch timeout 6 hours)
  - Minimize latency: Ensures minimal delay of event delivering

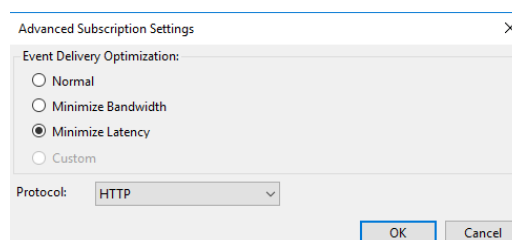


Figure K.11: Subscription Advanced Settings

This settings can now be exported with the command (no administrative rights needed):

```
wecutil gs "Subscription name" /f:xml > filename.xml
```

This will create a XML-file with the following content:

Listing K.1: Subscription XML

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
3   <SubscriptionId>Audit Account Lockout</SubscriptionId>
4   <SubscriptionType>SourceInitiated</SubscriptionType>
5   <Description></Description>
6   <Enabled>true</Enabled>
7   <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
8   <ConfigurationMode>MinLatency</ConfigurationMode>
9   <Delivery Mode="Push">
10     <Batching>
11       <MaxLatencyTime>30000</MaxLatencyTime>
12     </Batching>
13     <PushSettings>
14       <Heartbeat Interval="3600000"/>
15     </PushSettings>
16   </Delivery>
17   <Query>
18     <![CDATA[
19       <QueryList>
20         <Query Id="0">
21           <Select Path="Security">*[System[(EventID=4625)]]</Select>
22         </Query>
23       </QueryList>
24     ]>
25   </Query>
26   <ReadExistingEvents>true</ReadExistingEvents>
27   <TransportName>HTTP</TransportName>
28   <ContentFormat>RenderedText</ContentFormat>
29   <Locale Language="en-US"/>
30   <LogFile>ForwardedEvents</LogFile>
31   <PublisherName>Microsoft-Windows-EventCollector</PublisherName>
32   <AllowedSourceNonDomainComputers>
33     <AllowedIssuerCAList></AllowedIssuerCAList>
34   </AllowedSourceNonDomainComputers>
35   <AllowedSourceDomainComputers>
36     0:NSG:BAD:P(A;;GA;;;DC)(A;;GA;;;DD)S:
37   </AllowedSourceDomainComputers>
38 </Subscription>
```

**SubscriptionId:** Subscription name

**SubscriptionType:** Clients which listens on subscriptions of the WEC (defined in the GPO 1.1.3 Group Policy Objects for the subscribers) will pull the subscription (does not mean pull procedure regarding event forwarding)

**ConfigurationMode:** Defines the delivery mode options (normal, minimize bandwidth, minimize latency)

**Delivery Mode="<Mode>":** Defines the delivery mode (pull/push)

**Query:** Defines the query which EventIDs are forwarded

**ReadExistingEvents:** Defines if existing events should be forwarded (**true**) or only new ones (**false**)

**LogFile:** Defines where the events - which the clients forwarded to the WEC - should be saved (you can define your own .evtx-File for each subscription)



**AllowedSourceDomainComputers:** Defines which computers should listen to this subscription. If you have multiple domains, you have to get the identifiers of every domain connecting to your Windows Event Collector (WEC). The most simple way to do this is to make a new subscription from the GUI and export -> then copy the identifiers and import the new file. With the identifier of the above XML "Domain Users" and "Domain Computers" will listen to this subscription.

## 2. Import the subscription with a XML-file

- Run an administrative command prompt and execute the following command to import a single subscription: `wecutil cs C:\path\to\filename.xml`
- To import multiple subscriptions defined as an XML-file run, an administrative command prompt and execute the following command: `for %f in (C:\path\to\your\subscriptions\*.xml) do wecutil cs "%f"`
- The subscription "ReadinizerWEFRecommendation.xml" (see 2 Appendix B - ReadinizerWEFRecommendation.xml which is based on 2 Appendix A - Event Log IDs) defines event logs which should be audited with the Windows Event Collector for a better lateral movement analysis.

## 2 Encryption of Event Logs

As already in the section 1.2 Requirements and Limitations explained, the event logs in a Windows environment are encrypted by default using Kerberos. This section briefly explains which encryption standard is used and which strength it provides.

The following list shows all encryption types and their key strength supported for Kerberos:

Encryption Type	Description	Key Strength
DES_CBC_CRC	Data Encryption Standard with Cipher Block Chaining using the Cyclic Redundancy Check function	56 BIT
DES_CBC_MD5	Data Encryption Standard with Cipher Block Chaining using the Message-Digest algorithm 5 checksum function	56 BIT
RC4_HMAC_MD5	Rivest Cipher 4 with Hashed Message Authentication Code using the Message-Digest algorithm 5 checksum function	56 - 128 BIT
AES128_HMAC_SHA1	Advanced Encryption Standard in 128 bit cipher block with Hashed Message Authentication Code using the Secure Hash Algorithm (1)	128 BIT
AES256_HMAC_SHA1	Advanced Encryption Standard in 256 bit cipher block with Hashed Message Authentication Code using the Secure Hash Algorithm (1)	256 BIT
Future encryption types	Reserved for upcoming additional encryption types	UNKNOWN

Table K.1: Kerberos Encryption Types [90]

Since Windows 7 and Windows Server 2008, Microsoft has disabled the weak encryption types DES\_CBC\_CRC and DES\_CBC\_MD5 by default. These encryption types are since those versions deprecated but can still be activated manually for legacy support. Although, this is definitely not recommended!

The encryption type RC4\_HMAC\_MD5 can reach a strength of 128 bit, but both sides (client / server) must support the full-strength encryption. Otherwise the weak encryption type is used as described in RFC4757:

*A Kerberos client and server can negotiate over key length if they are using mutual authentication. If the client is unable to perform full-strength encryption, it may propose a key in the "subkey" field of the authenticator, using a weaker encryption type. [...] [96, Section 6]*

Thus the encryption type RC4\_HMAC\_MD5 does not guarantee sufficiently strong encryption. Only

## 2. Encryption of Event Logs

the two encryption types AES128\_HMAC\_SHA1 and AES256\_HMAC\_SHA1 use a minimum key length of 128 bit. In principle, however, the strongest encryption is always automatically negotiated between both parties. Nevertheless it is recommended to only allow the two encryption types AES128\_HMAC\_SHA1 and AES256\_HMAC\_SHA1.

This can be achieved with the following GPO setting (Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > Security Options > Network security: Configure encryption types allowed for Kerberos):

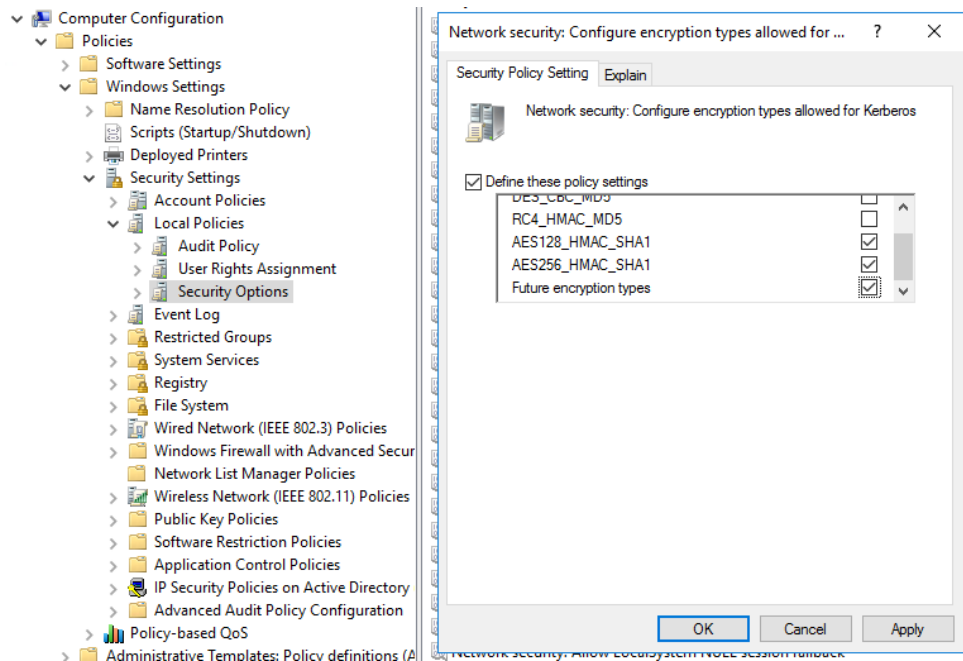


Figure K.12: Kerberos encryption

# Appendix A - Event Log IDs

- Account Logon
  - Audit Kerberos Authentication Service  
4768, 4772
  - Audit Kerberos Service Ticket Operations  
4769, 4770
- Account Managemenet
  - Audit Computer Account Management  
4741, 4742, 4743
  - Audit Other Account Management Events  
4782, 4793
  - Audit Security Group Management  
4731, 4732, 4733, 4734, 4735, 4764, 4799
  - Audit User Account Management  
4720, 4722, 4723, 4724, 4725, 4726, 4738, 4740,  
4765, 4766, 4767, 4780, 4781, 4794, 4798, 5376,  
5377
- Detailed Tracking
  - Audit Process Creation  
4688, 4696
  - Audit Process Termination  
4689
- Logon/Logoff
  - Audit Account Lockout  
4625
  - Audit Group Membership  
4627
  - Audit Logoff  
4634, 4647
  - Audit Logon  
4624, 4625, 4648, 4675
  - Audit Other Logon/Logoff Events  
4649, 4778, 4779, 4800, 4801, 4802, 4803, 5378,  
5632, 5633
  - Audit Special Logon  
4964, 4672
- Object Access
  - Audit File Share  
5140, 5142, 5143, 5144, 5168
  - Audit File System  
4656, 4658, 4660, 4663, 4664, 4985, 5051, 4670
  - Audit Handle Manipulation  
4658, 4690
  - Audit Kernel Object  
4656, 4658, 4660, 4663
  - Audit Other Object Access Events  
4671, 4691, 5148, 5149, 4698, 4699, 4700, 4701,  
4702, 5888, 5889, 5890
- Audit Registry  
4663, 4656, 4658, 4660, 4657, 5039, 4670
- Audit SAM  
4661
- Policy Change
  - Audit Audit Policy Change  
4715, 4719, 4817, 4902, 4906, 4907, 4908, 4912,  
4904, 4905
  - Audit MPSSVC Rule-Level Policy Change  
4944, 4945, 4946, 4947, 4948, 4949, 4950, 4951,  
4952, 4953, 4954, 4956, 4957, 4958
- Privilege Use
  - Audit Non Sensitive Privilege Use  
4673, 4674, 4985
  - Audit Sensitive Privilege Use  
4673, 4674, 4985
- System
  - Audit Security System Extension  
4610, 4611, 4614, 4622, 4697
  - Audit System Integrity  
4612, 4615, 4618, 4816, 5038, 5056, 5062, 5057,  
5060, 5051, 6281, 6410
- Clear Application Event Log
  - 104
- Clear Security Event Log
  - 1102
- Task Scheduler
  - 102, 106, 129, 200, 201
- Windows Remote Management
  - 6, 169
- System Events
  - 8222, 20001
- Terminal Services - Local Session Manager
  - 21, 24
- Sysmon Event Logs
  - \*

# Appendix B - ReadinizerWEFRecommendation.xml

Listing M.1: ReadinizerWEFRecommendation

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
3   <SubscriptionId>Readinizer Recommendation</SubscriptionId>
4   <SubscriptionType>SourceInitiated</SubscriptionType>
5   <Description>Event logs which should be audited with the Windows Event Collector for
      a better lateral movement analysis. These log IDs are based on the bachelor
      thesis 'Readinizer' (spring term 2019 - C.M. and L.K.).</Description>
6   <Enabled>true</Enabled>
7   <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
8   <ConfigurationMode>MinLatency</ConfigurationMode>
9   <Delivery Mode="Push">
10     <Batching>
11       <MaxLatencyTime>30000</MaxLatencyTime>
12     </Batching>
13     <PushSettings>
14       <Heartbeat Interval="3600000"/>
15     </PushSettings>
16   </Delivery>
17   <Query>
18     <![CDATA[
19       <QueryList>
20         <!-- Audit Kerberos Authentication Service -->
21         <Query Id="0"><Select Path="Security">*[System[(EventID=4768 or
            EventID=4772)]]</Select></Query>
22
23         <!-- Audit Kerberos Service Ticket Operations -->
24         <Query Id="1"><Select Path="Security">*[System[(EventID=4769 or
            EventID=4770)]]</Select></Query>
25
26         <!-- Audit Computer Account Management -->
27         <Query Id="2"><Select Path="Security">*[System[(EventID=4741 or EventID=4742
            or EventID=4743)]]</Select></Query>
28
29         <!-- Audit Other Account Management Events -->
30         <Query Id="3"><Select Path="Security">*[System[(EventID=4782 or
            EventID=4793)]]</Select></Query>
31
32         <!-- Audit Security Group Management -->
33         <Query Id="4"><Select Path="Security">*[System[(EventID=4731 or EventID=4732
            or EventID=4733 or EventID=4734 or EventID=4735 or EventID=4764 or
            EventID=4799)]]</Select></Query>
34
35         <!-- Audit User Account Management -->
36         <Query Id="5"><Select Path="Security">*[System[(EventID=4720 or EventID=4722
```

```

    or EventID=4723 or EventID=4724 or EventID=4725 or EventID=4726 or
    EventID=4738 or EventID=4740 or EventID=4765 or EventID=4766 or
    EventID=4767 or EventID=4780 or EventID=4781 or EventID=4794 or
    EventID=4798 or EventID=5376 or EventID=5377))]]</Select></Query>

```

```

37
38 <!-- Audit Process Creation -->
39 <Query Id="6"><Select Path="Security">*[System[(EventID=4688 or
    EventID=4696)]]</Select></Query>
40
41 <!-- Audit Process Termination -->
42 <Query Id="7"><Select
    Path="Security">*[System[(EventID=4689)]]</Select></Query>
43
44 <!-- Audit Account Lockout -->
45 <Query Id="8"><Select
    Path="Security">*[System[(EventID=4625)]]</Select></Query>
46
47 <!-- Audit Group Membership -->
48 <Query Id="9"><Select
    Path="Security">*[System[(EventID=4627)]]</Select></Query>
49
50 <!-- Audit Logoff -->
51 <Query Id="10"><Select Path="Security">*[System[(EventID=4634 or
    EventID=4647)]]</Select></Query>
52
53 <!-- Audit Logon -->
54 <Query Id="11"><Select Path="Security">*[System[(EventID=4624 or
    EventID=4648 or EventID=4675)]]</Select></Query>
55
56 <!-- Audit Other Logon/Logoff Events -->
57 <Query Id="12"><Select Path="Security">*[System[(EventID=4649 or
    EventID=4778 or EventID=4779 or EventID=4800 or EventID=4801 or
    EventID=4802 or EventID=4803 or EventID=5378 or EventID=5632 or
    EventID=5633)]]</Select></Query>
58
59 <!-- Audit Special Logon -->
60 <Query Id="13"><Select Path="Security">*[System[(EventID=4964 or
    EventID=4672)]]</Select></Query>
61
62 <!-- Audit File Share -->
63 <Query Id="14"><Select Path="Security">*[System[(EventID=5140 or
    EventID=5142 or EventID=5143 or EventID=5144 or
    EventID=5168)]]</Select></Query>
64
65 <!-- Audit File System and Audit Kernel Object -->
66 <Query Id="15"><Select Path="Security">*[System[(EventID=4656 or
    EventID=4658 or EventID=4660 or EventID=4663 or EventID=4664 or
    EventID=4985 or EventID=5051 or EventID=4670)]]</Select></Query>
67
68 <!-- Audit Handle Manipulation -->
69 <Query Id="17"><Select
    Path="Security">*[System[(EventID=4690)]]</Select></Query>
70
71 <!-- Audit Other Object Access Events -->
72 <Query Id="18"><Select Path="Security">*[System[(EventID=4671 or
    EventID=4691 or EventID=5148 or EventID=5149 or EventID=4698 or
    EventID=4699 or EventID=4700 or EventID=4701 or EventID=4702 or
    EventID=5888 or EventID=5889 or EventID=5890)]]</Select></Query>
73
74 <!-- Audit Registry -->
75 <Query Id="19"><Select Path="Security">*[System[(EventID=4657 or

```

```

      EventID=5039)]]</Select></Query>
76
77 <!-- Audit SAM -->
78 <Query Id="20"><Select
      Path="Security">*[System[(EventID=4661)]]</Select></Query>
79
80 <!-- Audit Audit Policy Change -->
81 <Query Id="21"><Select Path="Security">*[System[(EventID=4715 or
      EventID=4719 or EventID=4817 or EventID=4902 or EventID=4906 or
      EventID=4907 or EventID=4908 or EventID=4912 or EventID=4904 or
      EventID=4905)]]</Select></Query>
82
83 <!-- Audit MPSSVC Rule-Level Policy Change -->
84 <Query Id="22"><Select Path="Security">*[System[(EventID=4944 or
      EventID=4945 or EventID=4946 or EventID=4947 or EventID=4948 or
      EventID=4949 or EventID=4950 or EventID=4951 or EventID=4952 or
      EventID=4953 or EventID=4954 or EventID=4956 or EventID=4957 or
      EventID=4958)]]</Select></Query>
85
86 <!-- Audit Non Sensitive Privilege Use and Audit Sensitive Privilege Use -->
87 <Query Id="23"><Select Path="Security">*[System[(EventID=4673 or
      EventID=4674)]]</Select></Query>
88
89 <!-- Audit Security System Extension -->
90 <Query Id="24"><Select Path="Security">*[System[(EventID=4610 or
      EventID=4611 or EventID=4614 or EventID=4622 or
      EventID=4697)]]</Select></Query>
91
92 <!-- Audit System Integrity -->
93 <Query Id="25"><Select Path="Security">*[System[(EventID=4612 or
      EventID=4615 or EventID=4618 or EventID=4816 or EventID=5038 or
      EventID=5056 or EventID=5062 or EventID=5057 or EventID=5060 or
      EventID=6281 or EventID=6410)]]</Select></Query>
94
95 <!-- Clear Application Event Log -->
96 <Query Id="26"><Select
      Path="System">*[System[Provider[@Name='Microsoft-Windows-Eventlog'] and
      Level=4 and EventID=104]]</Select></Query>
97
98 <!-- Clear Security Event Log -->
99 <Query Id="27"><Select
      Path="Security">*[System[Provider[@Name='Microsoft-Windows-Eventlog']
      and Level=4 and EventID=1102]]</Select></Query>
100
101 <!-- Task Scheduler -->
102 <Query Id="28"><Select Path="Application">*[System[(EventID=102 or
      EventID=106 or EventID=129 or EventID=200 or
      EventID=201)]]</Select></Query>
103
104 <!-- Windows Remote Management -->
105 <Query Id="29"><Select Path="Application">*[System[(EventID=6 or
      EventID=169)]]</Select></Query>
106
107 <!-- System Events -->
108 <Query Id="30"><Select Path="Application">*[System[(EventID=8222 or
      EventID=20001)]]</Select></Query>
109
110 <!-- Terminal Services - Local Session Manager -->
111 <Query Id="31"><Select Path="Application">*[System[(EventID=21 or
      EventID=24)]]</Select></Query>
112

```

```

113         <!-- Sysmon Event Logs -->
114         <Query Id="32" Path="Microsoft-Windows-Sysmon/Operational"><Select
            Path="Microsoft-Windows-Sysmon/Operational">*</Select></Query>
115     </QueryList>
116     <]]>
117 </Query>
118 <ReadExistingEvents>false</ReadExistingEvents>
119 <TransportName>HTTP</TransportName>
120 <ContentFormat>RenderedText</ContentFormat>
121 <Locale Language="en-US"/>
122 <LogFile>ForwardedEvents</LogFile>
123 <PublisherName>Microsoft-Windows-EventCollector</PublisherName>
124 <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
125 <!-- If you have multiple domains, you have to get the identifiers of every domain
        connecting to your Windows Event Collector (WEC) and put them below. -->
126 <!-- The most simple way to do this is to make a new subscription from the GUI and
        export -> then copy the identifiers and import the new file. -->
127 <!-- Export command: wecutil gs "subscriptionname" /f:xml > filename.xml -->
128 <!-- Import command: wecutil cs filename.xml | Import command multiple subscriptions:
        for %f in (C:\path\to\your\subscriptions\*.xml) do wecutil cs "%f" -->
129 <!-- The SDDL below is just the well known identifiers for "Domain Users" and "Domain
        Computers" -->
130 <AllowedSourceDomainComputers>
131     O:NSG:BAD:P(A;;GA;;;DC)(A;;GA;;;DD)S:
132 </AllowedSourceDomainComputers>
133 </Subscription>

```





**HSR**  
**HOCHSCHULE FÜR TECHNIK**  
**RAPPERSWIL**

**COMPUTER SCIENCE**

# Installation & User Manual

READINIZER

## **Authors:**

Claudio MATTES  
claudio.mattes@hsr.ch

Lukas KELLENBERGER  
lukas.kellenberger@hsr.ch

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

June 12, 2019

# General Information

## 1.1 Overview

The Readinizer is an application which helps to check the readiness of an entire Windows Network. Therefore it gathers information from an Active Directory about the domains, sites, organizational units and their member computers/servers. The Readinizer then collects a “Resultant Set of Policy”, the active security settings that are set on the computer, for one computer of each organizational unit. The found settings are then compared against the recommended settings. The result of the analysis is then presented to the user in form of a percentage figure whereby a tree structure of the forest depicts the analyzed RSoPs and gives a first view of the readiness. In addition, the user has the possibility to simultaneously perform a Sysmon check. Sysmon is a tool by Mark Russinovich which logs the same as default event logger but where the executables are hashed, hence compromise of such executables can be detected. The user can then drill down the RSoPs to a detailed view over all applied / recommended settings and which GPO applied those settings

The Readinizer was developed during a bachelor thesis by the two Bachelor of Science in Computer Science students, Claudio Mattes and Lukas Kellenberger.

## 1.2 Organization of the Manual

The user manual consists of four parts:

- **General Information:**  
The General Information section explains the application and the purpose for which it is intended.
- **System Requirements:**  
The System Requirements section provides a general overview of the system requirements. Which operating systems are supported, what software must be pre-installed, and what authorizations the user must have.
- **Getting Started:**  
The Getting Started section explains how to obtain and install the Readinizer on your device.
- **Using the Readinizer:**  
The Using the Readinizer section provides a detailed description of the system functions.

# System Requirements

## 2.1 Operating System

The Readinizer runs on all Windows 10 Professional Version 1709 operated systems as well as on all servers with the operating system Windows Server 2016.

## 2.2 User Authorizations

To run the Readinizer successfully, the user needs administrator rights on the executing machine. Additionally, he needs Local Administrator and Remote Desktop User rights in every domain that is going to be analyzed. It is recommended to create a custom user/user group. In this case a user group called “Local Admins” has been created:

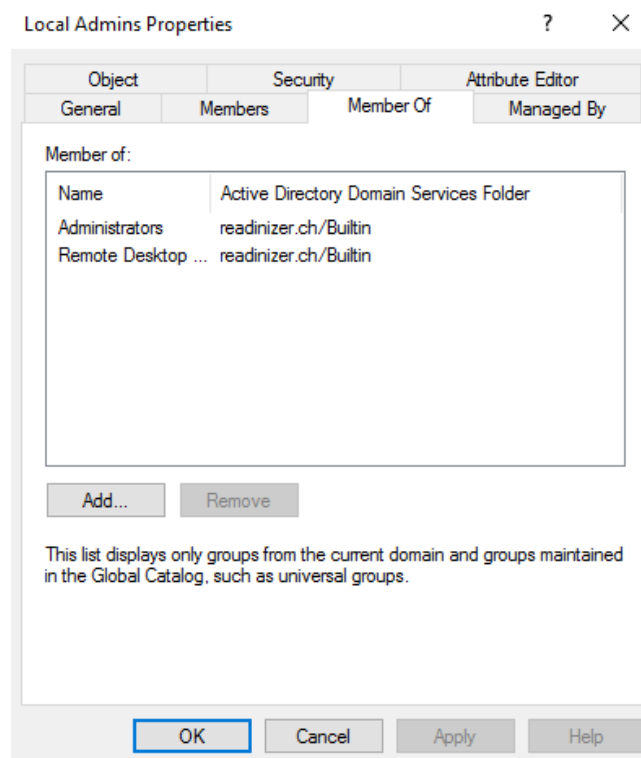


Figure O.1: Domain Rights

## 2.3 Firewall Settings

The firewalls of the target computers must allow WMI connections and Ping (ICMP echo requests) for the Readinizer to run successfully.

## 2.4 Pre-Installed Software

### 2.4.1 Remote Server Administration Tool

To enable the Readinizer to read the Resultant Set of Policies, the Remote Server Administration Tools (RSAT) must be installed/activated on the executing device.

#### Version 1803 and older

For computers with Windows 10 Version 1803 and older the RSAT can be downloaded here:  
<https://www.microsoft.com/en-us/download/details.aspx?id=45520>

The installation is simple and self-explanatory.

#### Version 1809 and newer

Since the October 2018 update, the RSAT is pre-installed on Windows Professional machines. However, it still has to be activated. To do so open **Settings** → **App**. Then click on **Manage optional features**.

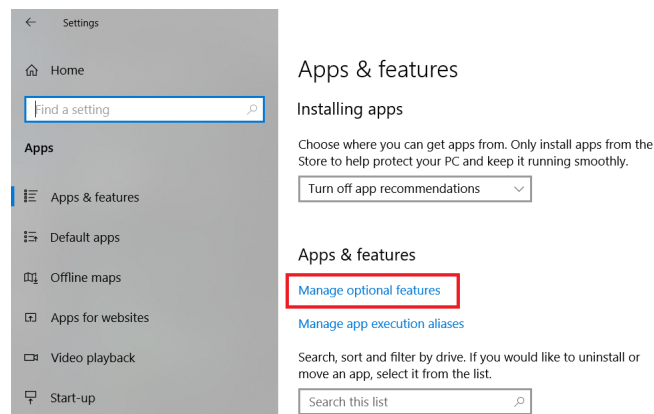


Figure O.2: Manage optional features

Then click the **Add a feature** button.

Scroll down until you see the **RSAT: Group Policy Management Tools** and install this feature.

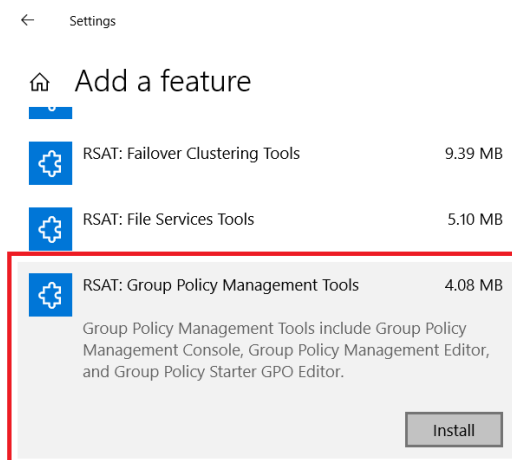


Figure O.3: RSAT: Group Policy Management Tools

### 2.4.2 SQLLocalDB

To display the complexity of an Active Directory, the Readinizer needs a database. For this a lightweight database is used, a SQLLocalDB. To install the LocalDB download the SQL Server Express installer. It can be downloaded here:

<https://www.microsoft.com/en-us/sql-server/sql-server-editions-express>

After executing the downloaded installer, a installation type has to be selected. Choose **Download Media**.

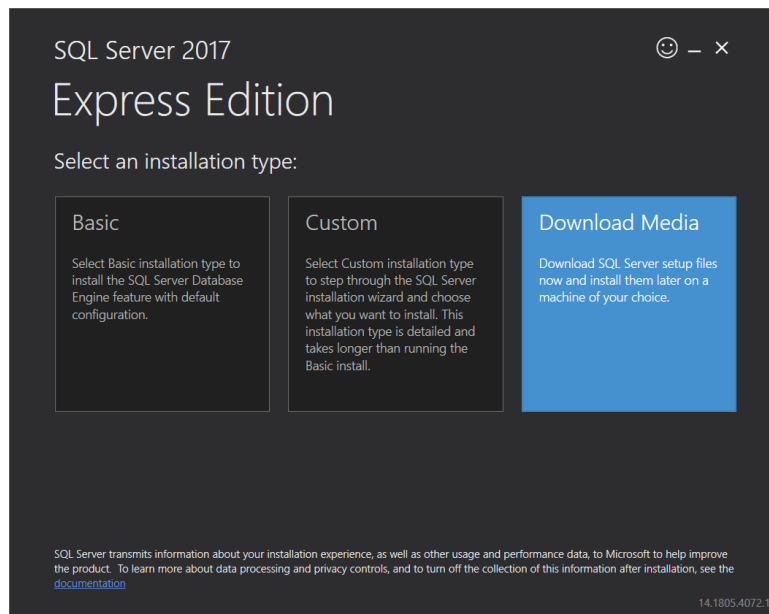


Figure O.4: SQL Server Express 2017

Activate the **LocalDB**-toggle and select where the LocalDB-installer should be saved.

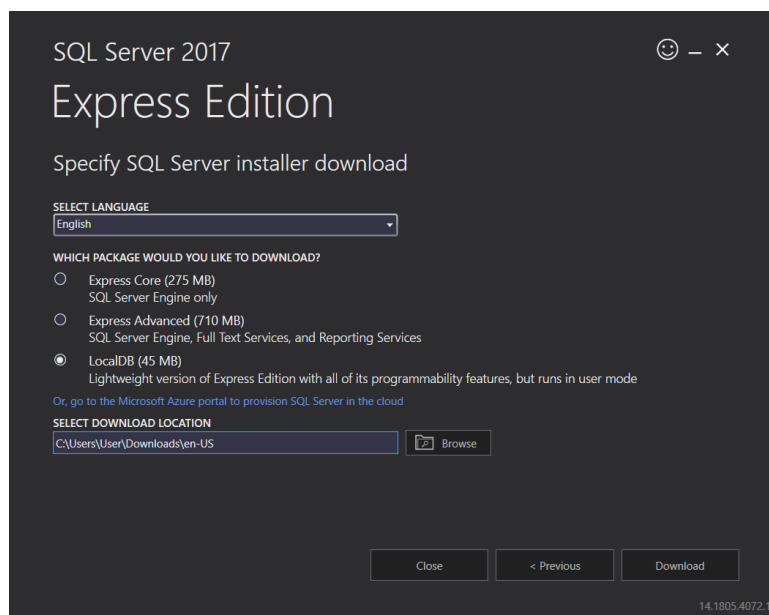


Figure O.5: SQL Server Express 2017

After this installer is downloaded, open it at the provided path. Install SQLLocalDB by using the installation wizard.



Figure O.6: SQL Server Express 2017

# Getting Started

## 3.1 Download

You can find the installer, the portable application or the plain code on this GitHub repository:

<https://github.com/clma91/Readinizer/releases/>

## 3.2 Installation

### 3.2.1 Installer

Execute the installer and click “Next”. Select the folder where the Readinizer should be installed. By default, a folder is created in Program Files. Confirm the installation by clicking on “Next”. Allow to make changes to your device. After the installation is complete, close the installer.

### 3.2.2 Portable Application

Unpack the ZIP folder. The portable application does not need any further installation. Just execute the Readinizer.exe as an administrator.

# Using the Readinizer

## 4.1 Starting the Readinizer

Execute the Readinizer.exe as an administrator. After a few seconds the Readinizer home screen opens.

### 4.1.1 Home Screen



Figure Q.1: Readinizer Home Screen

1. Provide the name of the domain that should be analyzed. If this field is left blank, the forest root domain is selected by default.
2. If this toggle is not activated, only the provided domain or the default domain is analyzed. If the toggle is activated all subdomains and treedomains of the provided domain are analyzed as well.



3. If this toggle is activated all reachable computers in the network are checked if the Sysmon service is running. Sysmon is a monitoring tool by Mark Russinovich. For more information check: <https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon>
4. For security reasons it is recommended that the Sysmon service is run under a different name. This can prevent a potential attacker from detecting that Sysmon is logging events. To check how many machines Sysmon is installed on, you can enter the changed service name here. Default value is Sysmon.
5. By clicking on this button the analysis will be performed with the above settings. First, information about domains, sites, organizations units and computers is loaded from the Active Directory. Then an attempt is made to contact a computer from each organization unit and read out a RSoP. A RSoP contains the active computer logging settings. These settings are then compared with the recommended settings and the result is displayed.

#### 4.1.2 Forest Result Screen

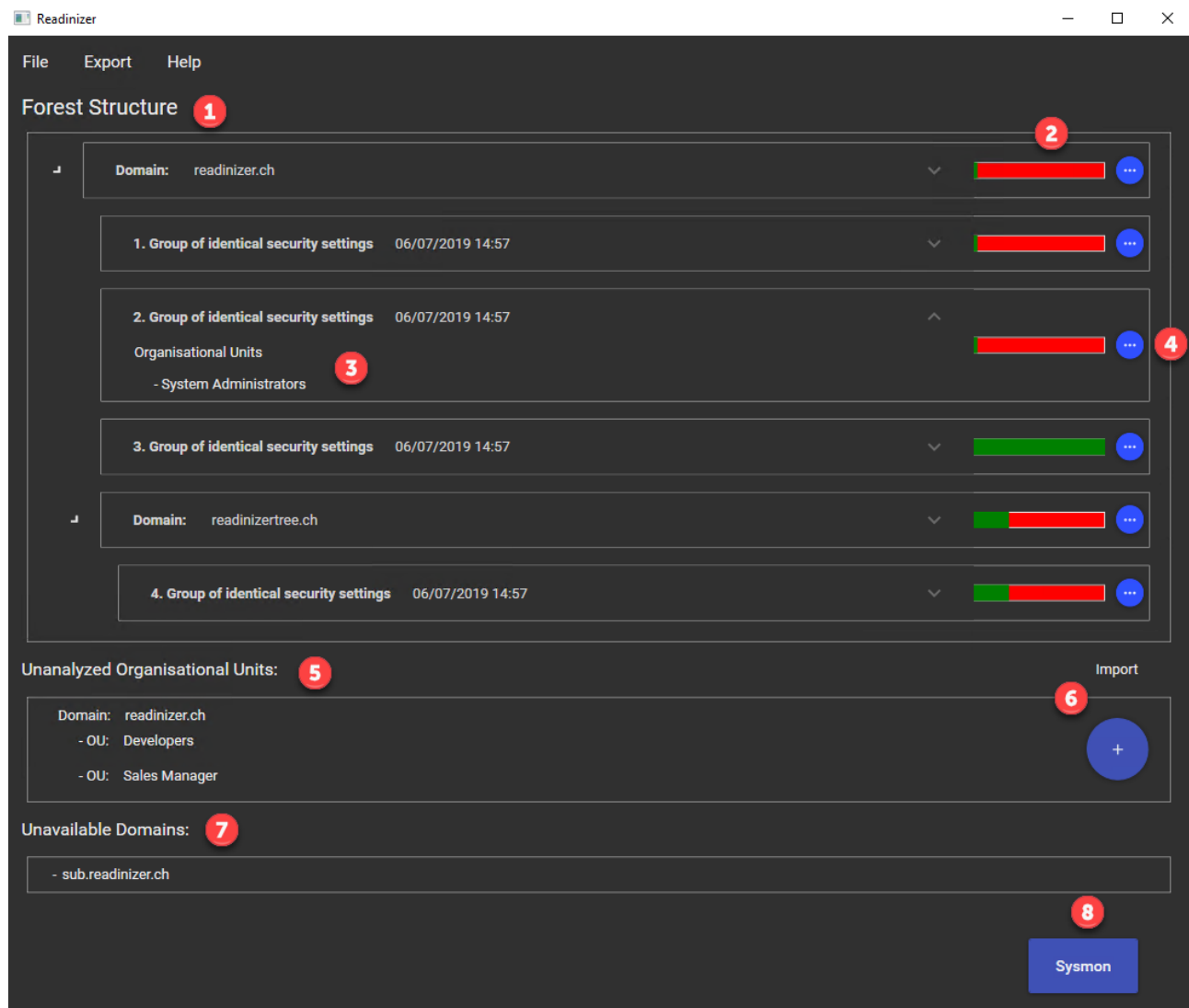


Figure Q.2: Readinizer Forest Result Screen

1. The result reflects the structure of a forest. At the top is the forest root domain or the parent domain, beneath are the subdomains and tree domains.
2. The progressbar reflects the readiness of a domain or a group of identical security settings. The progressbar of the groups of identical security settings is calculated by matching settings. For the domain, the lowest value of its children is used.
3. The group of identical security settings can be expanded to show the organizational units that are members of that group.
4. This button opens a more detailed view for domains and groups of identical security settings.
5. This section lists all organizational units where no computer could be reached and therefore no RSoP was extracted.
6. With this button RSoPs can be inserted as XML files for the unanalyzed organizational units.
7. This section lists all domains that are members of the forest but could not be contacted.
8. By clicking this button you get an overview of the Sysmon installation rate in the network. This button is only showed if the corresponding toggle on the home screen was selected.

#### 4.1.3 Domain Result Screen

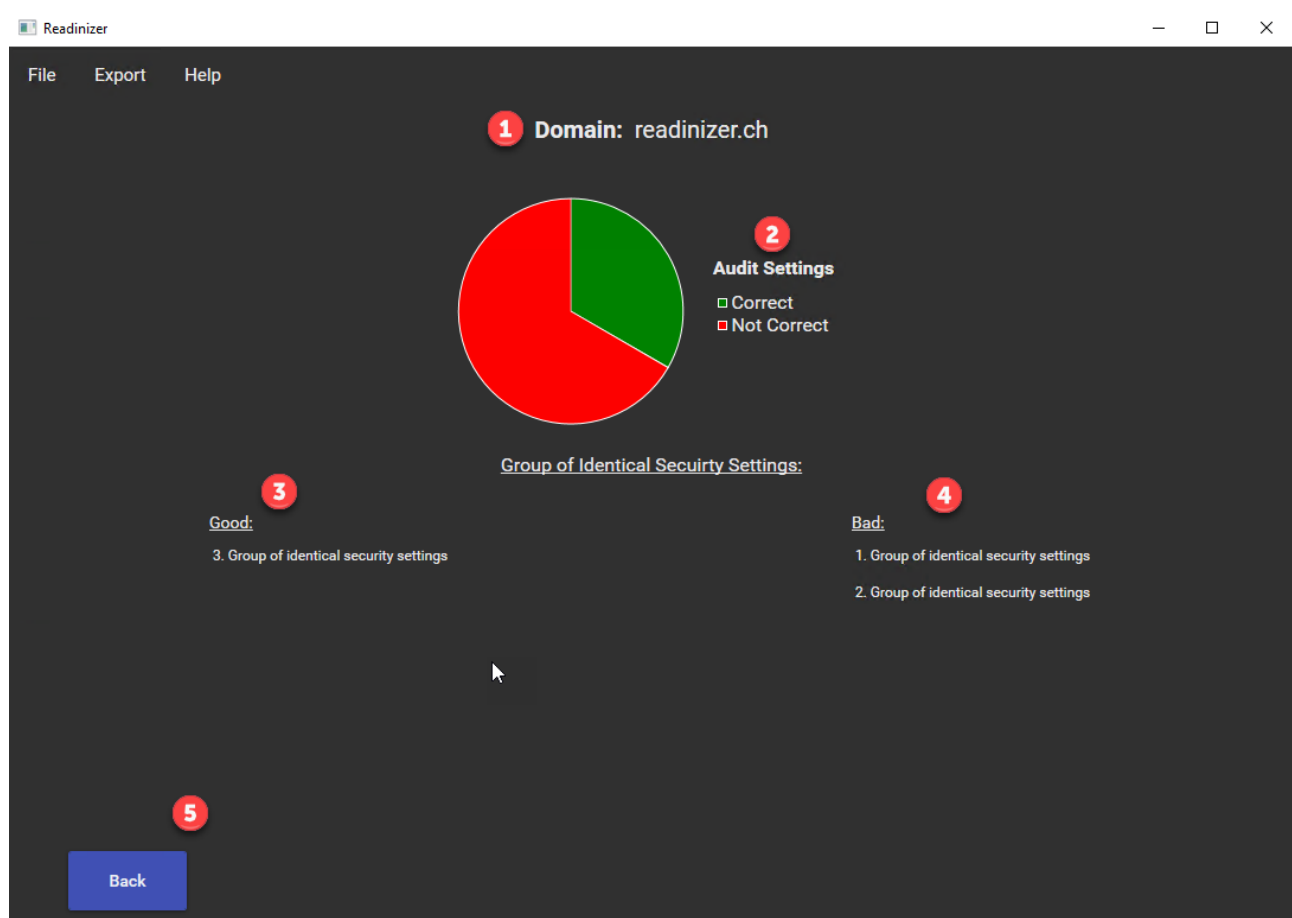


Figure Q.3: Readinizer Domain Result Screen

1. The name of the domain that is displayed.
2. The number of correct groups of identical security settings shown in a pie chart.
3. The groups of identical security settings that match the recommended settings. A click on the name opens a more detailed overview of the group of identical security settings.
4. The groups of identical security settings that do not match the recommended settings. A click on the name opens a more detailed overview of the group of identical security settings.
5. This button brings you back to the forest result overview.

#### 4.1.4 Group of Identical Security Settings Result Screen

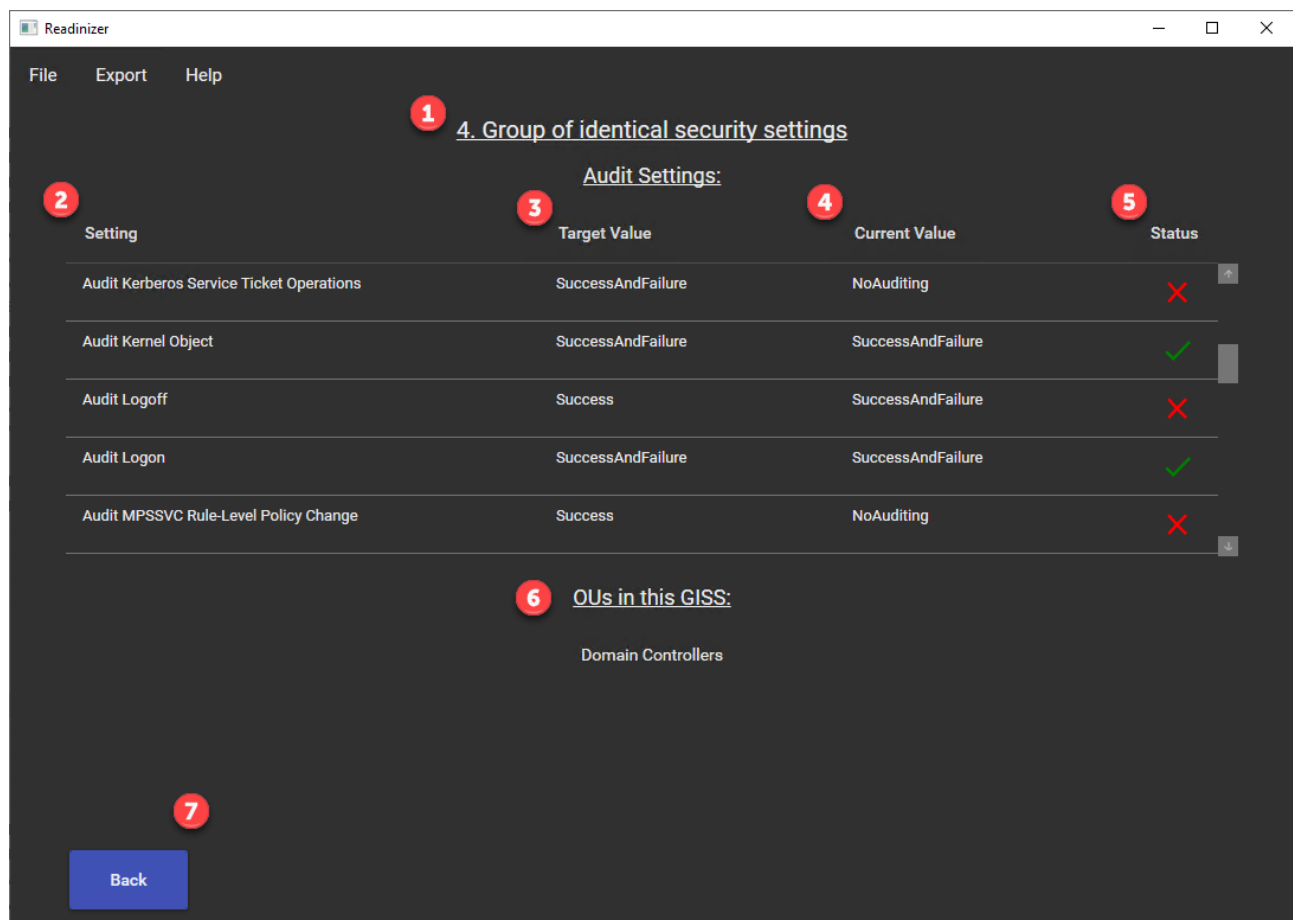


Figure Q.4: Readinizer Group of Identical Security Settings Result Screen

1. The name/number of the group of identical security settings that is displayed.
2. The name of the setting.
3. The target value of the setting.
4. The current value of the setting.
5. The status of the setting, a green checkmark if matching, a red cross if there is no match, a orange exclamation mark if undefined.

6. A list of organization units that are member in this group of identical security settings. A click on the name opens a more detailed overview of the organization unit.
7. This button brings you back to the domain result overview.

#### 4.1.5 Organizational Unit Result Screen

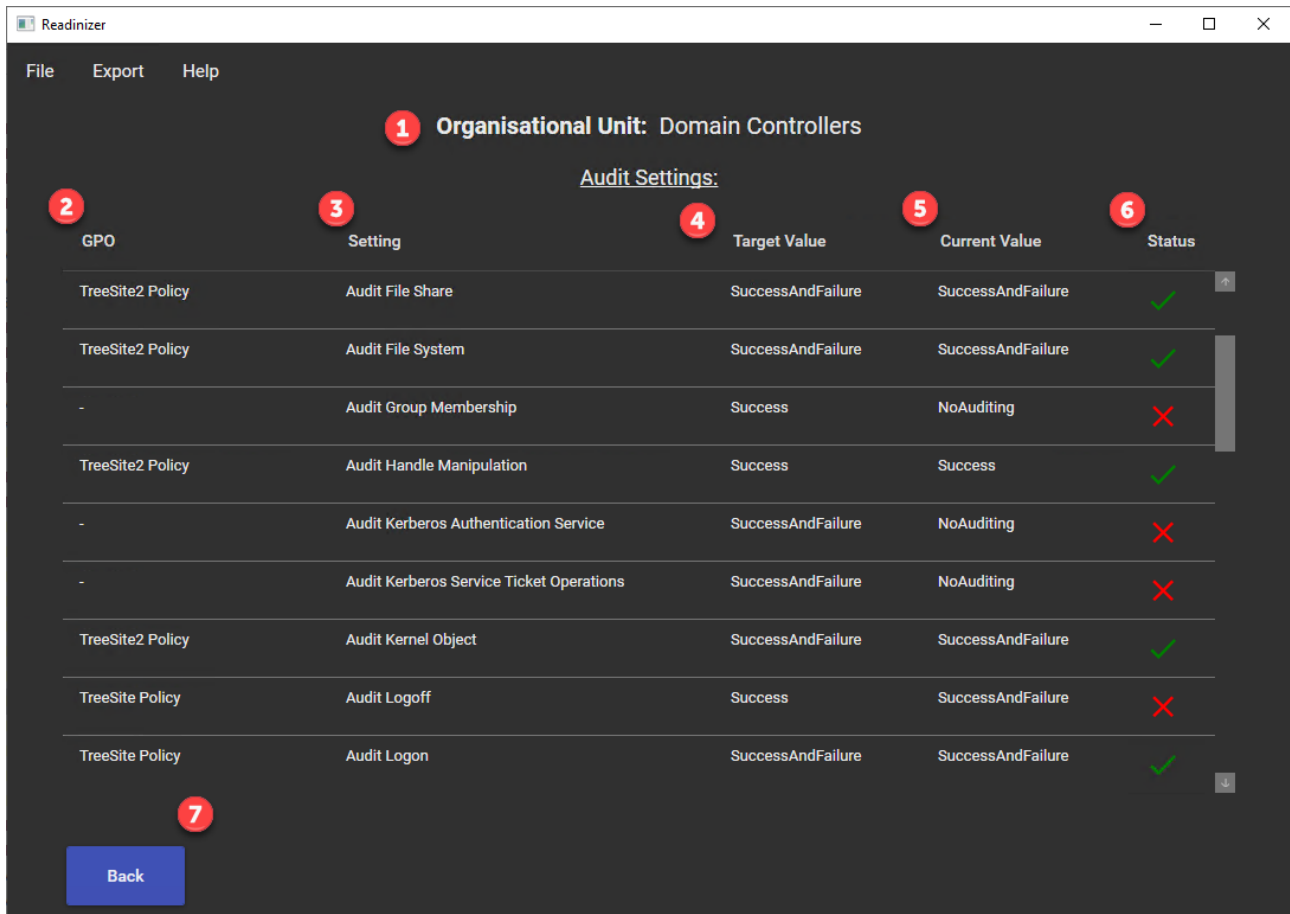


Figure Q.5: Readinizer Organizational Unit Result Screen

1. The name of the Organizational Unit that is displayed.
2. The name of the Group Policy Object which set this setting.
3. The name of the setting.
4. The target value of the setting.
5. The current value of the setting.
6. The status of the setting, a green checkmark if matching, a red cross if there is no match, a orange exclamation mark if undefined.
7. This button brings you back to the group of identical security settings result overview.

#### 4.1.6 Sysmon Result Screen

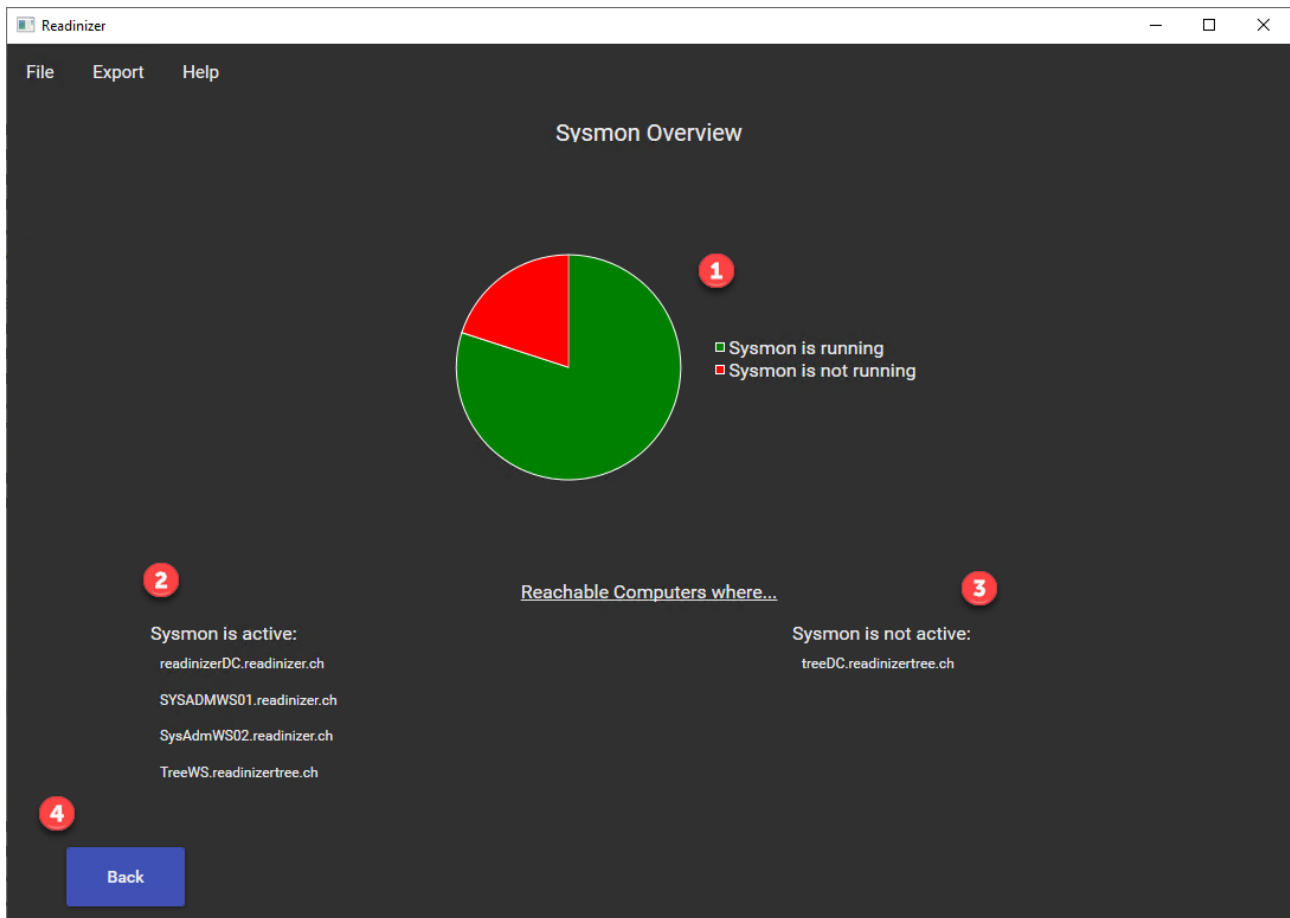


Figure Q.6: Readinizer Sysmon Result Screen

1. The percentage of computers on the network on which Sysmon is installed is displayed in a pie chart.
2. List of computer on which sysmon is installed.
3. List of computer on which sysmon is not installed.
4. This button brings you back to the forest result overview.

#### 4.1.7 Navigationbar

##### File

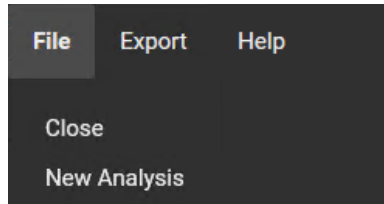


Figure Q.7: Navigationbar File

1. Close: Terminates the application
2. New Analysis: Truncates the database, prepares the Readinizer for a new analysis

##### Export

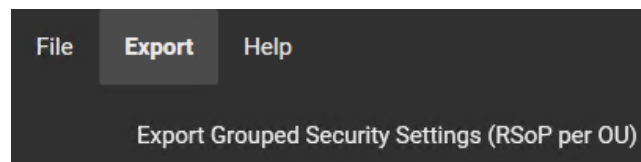


Figure Q.8: Navigationbar Export

1. Export Grouped Security Settings: Exports the collected and analyzed data to a JSON file to a specified path

##### Help

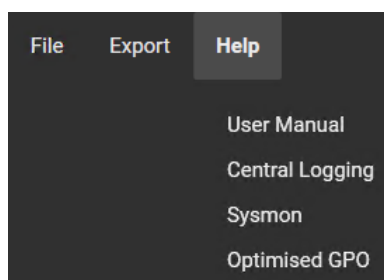


Figure Q.9: Navigationbar Help

1. User Manual: Contains a link to this document
2. Central Logging: Contains a link to a guide on how to implement central logging
3. Sysmon: Contains a link to a guide on how to install Sysmon over an entire fleet through Group Policy Objects
4. Optimized GPO: Contains a link to the recommend Group Policy File