

Marius Ødegård Lindvall  
May-Liss Amundsen  
Martin Wahl  
Steinar Vrenne

## Multipurpose platform for Security Analysts

Bachelor's project in IT-Operations and Information Security  
Supervisor: Jia-Chun Lin

May 2021



Marius Ødegård Lindvall  
May-Liss Amundsen  
Martin Wahl  
Steinar Vrenne

# **Multipurpose platform for Security Analysts**

Bachelor's project in IT-Operations and Information Security  
Supervisor: Jia-Chun Lin  
May 2021

Norwegian University of Science and Technology  
Faculty of Information Technology and Electrical Engineering





## Sammendrag av Bacheloroppgaven

Kongsberg Gruppen er et internasjonalt teknologikonsern som utvikler høyt teknologiske systemer til kunder innen olje og gass, frakt, forsvar og romfartsindustri. Kongsberg Cyber Security Center er avdelingen innen Kongsberg Gruppen som er ansvarlig for analyse, deteksjon og hendelseshåndtering for alle områder innen konsernet. Den type arbeid krever muligheten til å koble seg til forskjellige miljøer for å utføre forskjellige oppgaver, noe som i dag krever bruk av et større antall datamaskiner i analysearbeidet. Hovedmålet med denne bacheloroppgaven er å utvikle en multifunksjons analyseplattform for sikkerhetsanalytikere, som kan redusere behovet for flere datamaskiner ved at én maskin kobler seg på flere av miljøene samtidig, og som kan holde disse miljøene isolert fra hverandre. I denne oppgaven vil vi gjøre rede for teknologier som muliggjør slik type soneinndeling, og undersøke hvordan en slik teknologi kan tilpasses for å møte Kongsbergs krav til en analyseplattform. I tillegg vil det bli utformet og presentert en risikovurdering av det endelige produktet, og anbefalinger for systemkrav vil utarbeides. Resultatet vil være en rapport om hvorvidt en slik type plattform kan brukes som et reelt analyseverktøy, og dokumentasjon som Kongsberg kan bruke for videre intern utvikling av plattformen.

## **Abstract of Bachelor Assignment**

Kongsberg Group is an international technology group developing high technology systems for customers within the oil and gas, shipping, defence and aerospace industries. Kongsberg Cyber Security Center is the department within Kongsberg Group that handles analysis, detection and incident response for all divisions of the group. This kind of work necessitates connecting to several environments to perform various tasks, which presently requires the use of a large number of computers for analysis work. The main goal of this bachelor project is to develop a multipurpose platform for security analysts to reduce the need of multiple computers by several environments on a single machine, while being capable of keeping those environments isolated from each other. In the thesis, we will explain technologies that enable the abovementioned multipurpose platform, and explore how such technologies can be adapted to meet Kongsberg's requirements for a secure analysis platform. Additionally, a risk assessment of the final product, and recommendations on hardware specifications will be presented. The result will be a report indicating whether or not such a platform can be used as a useful analysis tool in the real world, as well as documentation for Kongsberg to further develop the platform.

# Preface

“Multipurpose platform for security analysts” is a Bachelor thesis written by four IT-Operations and Information Security students at Norges Teknisk-Naturvitenskapelige Universitet (NTNU) in Gjøvik.

We hope this thesis can help other students and companies decide on a similar project and what platform to base it upon.

We would like to thank the following for their contribution and help towards this thesis:

- David Lee Andersen and Erlend Hammer at Kongsberg Cyber Security Center for providing this subject and laptops to test this product on as well continually giving counsel and feedback on our thesis.
- Jia-Chun Lin at NTNU in Gjøvik for providing good counsel and continually providing feedback for helping us improve our thesis.

Gjøvik, May 2021

# Contents

<b>Preface</b> . . . . .	<b>v</b>
<b>Contents</b> . . . . .	<b>vi</b>
<b>Figures</b> . . . . .	<b>ix</b>
<b>Tables</b> . . . . .	<b>x</b>
<b>Code Listings</b> . . . . .	<b>xi</b>
<b>Acronyms</b> . . . . .	<b>xii</b>
<b>Glossary</b> . . . . .	<b>xiv</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Background . . . . .	1
1.2 Project Goals . . . . .	2
1.3 Limitations . . . . .	2
1.4 Project Group . . . . .	2
1.5 Thesis Structure . . . . .	3
<b>2 Research Questions and Work Breakdown</b> . . . . .	<b>5</b>
2.1 High-level work breakdown . . . . .	6
<b>3 Background</b> . . . . .	<b>7</b>
3.1 Malware . . . . .	7
3.2 Malware analysis . . . . .	7
3.3 Virtual machines . . . . .	8
3.4 Virtualization and hypervisors . . . . .	9
3.5 Configuration Management . . . . .	10
<b>4 Investigation of Hypervisors</b> . . . . .	<b>11</b>
4.1 Xen Hypervisor . . . . .	12
4.2 Star Lab Crucible / Star Lab Titanium Secure Hypervisor . . . . .	12
4.3 AIS SecureView . . . . .	13
4.4 HP Sure Click . . . . .	15
4.5 sVirt . . . . .	15
4.6 Hysolate . . . . .	16
4.7 PolyXene . . . . .	17
4.8 Qubes OS . . . . .	17
4.9 Cloud-based solutions . . . . .	18
4.10 Investigation summary . . . . .	18
<b>5 Technical Design</b> . . . . .	<b>19</b>
5.1 Qubes OS . . . . .	19
5.2 Virtual machines . . . . .	20
<b>6 Networking</b> . . . . .	<b>22</b>
6.1 OpenVPN . . . . .	22
6.2 PiVPN . . . . .	23
6.3 Network architecture in Qubes OS . . . . .	23
6.4 Network design of the multipurpose platform . . . . .	26



<b>7</b>	<b>Security Information and Event Management</b>	<b>29</b>
7.1	SIEM software	30
7.2	Splunk	30
7.3	Splunk integration with Qubes OS	30
7.4	Script run-down	31
7.5	Investigation summary	34
<b>8</b>	<b>Windows in Qubes OS</b>	<b>35</b>
8.1	Installing Windows in Qubes	35
8.2	Qubes Windows Tools	37
8.3	Speakers and microphone (audio)	37
8.4	Web camera	38
8.5	Screen sharing	38
8.6	Investigation summary	38
<b>9</b>	<b>Identity and Access Management</b>	<b>39</b>
9.1	Microsoft Azure Active Directory (Azure AD)	39
9.2	Other solutions	40
9.3	Implementation	41
9.4	Multifactor authentication	43
9.5	Investigation summary	43
<b>10</b>	<b>Configuration Management</b>	<b>44</b>
10.1	Tools	44
10.2	Implementation and centrally managing laptops running Qubes OS	46
10.3	Recommendations	49
<b>11</b>	<b>Qubes OS Hardware Compatibility</b>	<b>51</b>
11.1	Our experience with the provided laptops	51
11.2	Laptop vs desktop vs cloud	53
11.3	Investigation Summary	55
11.4	Hardware recommendations	55
<b>12</b>	<b>Risk Assessment of the Multipurpose Platform</b>	<b>56</b>
12.1	Main assets	57
12.2	Use cases	57
12.3	Potential attacker profiles	63
12.4	Abuse cases	65
12.5	Findings	82
<b>13</b>	<b>Evaluation</b>	<b>86</b>
13.1	Client	87
13.2	Bandwidth	87
<b>14</b>	<b>Discussion</b>	<b>89</b>
14.1	Related Work	90
<b>15</b>	<b>Development Process</b>	<b>91</b>
15.1	Development model	91
15.2	Documentation	91
15.3	Routines	92
<b>16</b>	<b>Closing Remarks</b>	<b>94</b>
16.1	Learning outcome	94
16.2	Conclusion	94
16.3	Future work	95
	<b>Bibliography</b>	<b>96</b>
<b>A</b>	<b>Code files</b>	<b>104</b>
A.1	splunkforwarder.py	104

A.2	splunkforwarder.conf.json	107
A.3	gitscript.sh	107
A.4	InitializeUsername.py	109
A.5	Install-VPN.py	110
A.6	machines.sls	110
A.7	top.sls	114
A.8	pullUpdate.sls	114
A.9	proxySoftware.sls	115
A.10	AnalysisSoftware.sls	115
<b>B</b>	<b>Emails</b>	<b>116</b>
B.1	AIS Secureview reply	116
<b>C</b>	<b>Timesheets</b>	<b>117</b>
C.1	Example of Timesheet	117
<b>D</b>	<b>Project plan</b>	<b>118</b>
D.1	Goals and limitations	118
D.2	Scope	120
D.3	Project management	121
D.4	Planning, follow-up, and reporting	122
D.5	Quality assurance	123
D.6	Implementation plan	124
<b>E</b>	<b>Group rules</b>	<b>128</b>
<b>F</b>	<b>Project agreement</b>	<b>130</b>
<b>G</b>	<b>Meeting minutes</b>	<b>135</b>
G.1	Jan 14: Supervisor meeting	135
G.2	Jan 14: Group decisions	136
G.3	Jan 20: Client meeting	136
G.4	Jan 20: Daily scrum meeting	137
G.5	Jan 20: Weekly scrum meeting	137
G.6	Jan 21: Supervisor meeting	138
G.7	Jan 21: Daily scrum meeting	139
G.8	Jan 22: Daily scrum meeting	139
G.9	Jan 27: Weekly scrum meeting	139
G.10	Jan 28: Supervisor meeting	140
G.11	Jan 28: Daily scrum meeting	142
G.12	Feb 03: Weekly scrum meeting	142
G.13	Feb 05: Daily scrum meeting	143
G.14	Feb 10: Weekly scrum meeting	143
G.15	Feb 11: Supervisor meeting	143
G.16	Feb 11: Daily scrum meeting	145
G.17	Feb 17: Weekly scrum meeting	145
G.18	Feb 18: Supervisor meeting	146
G.19	March 3: Weekly scrum meeting	147
G.20	March 10: Supervisor meeting	148
G.21	March 17: Client meeting	150
G.22	March 24: Supervisor meeting	151
G.23	March 31: Client meeting	153
G.24	April 7: Supervisor meeting	154
G.25	April 7: Weekly scrum meeting	157

# Figures

3.1	Overview of the two types of hypervisors [24]	9
4.1	SecureView desktop [37]	14
4.2	Hysolate architecture and Hysolate desktop [42, 43]	16
5.1	The different qubes in a typical Qubes OS installation [53]	20
6.1	Network architecture within Qubes OS [61]	24
6.2	The architecture of a VPN domain [61]	25
6.3	Networking connections diagram	26
6.4	Proxy diagram	28
9.1	Joining to Azure AD	41
9.2	List of licenses we were granted in the working proof of concept	42
9.3	Joined Windows 10 VM	42
10.1	Image from Qubes OS press release	45
10.2	Overview	47
10.3	Example configuration screenshot	50
11.1	Comparison between a Ryzen 7 3700 and 3700U [104]	54
12.1	Use case diagram	58
12.2	Probability level requirements used in risk assessment	66
12.3	Consequence level requirements used in risk assessment	67
12.4	Abuse case diagram	68
12.5	Risk overview before Qubes OS & countermeasures	83
12.6	Risk overview after Qubes OS & countermeasures	84
12.7	List of color-coded risks before and after countermeasures.	85
13.1	Data transfer through no VPN tunnel	87
13.2	Data transfer through VPN tunnel	88
15.1	An average work week per person.	92
2.1	AIS Secureview reply.	116
3.1	An example selection of a group members hours.	117

# Tables

- 8.1 Qubes Windows Tools compability table [68] . . . . . 37
  
- 11.1 Laptop configuration . . . . . 51
- 11.2 Comparison between laptop, desktop and cloud . . . . . 53
  
- 12.1 Main assets . . . . . 57
- 12.2 Use case 1 - Log in to the machine . . . . . 59
- 12.3 Use case 2 - Browse the Internet/non-work related tasks . . . . . 59
- 12.4 Use case 3 - Malware analysis . . . . . 59
- 12.5 Use case 4 - Office related work . . . . . 60
- 12.6 Use case 5 - Write software . . . . . 60
- 12.7 Use case 6 - Administer the platform . . . . . 61
- 12.8 Use case 7 - Initial installation on platform . . . . . 62
- 12.9 Use case 8 - Change/connect to NetVM . . . . . 62
- 12.10 Use case 9 - Connect to network . . . . . 62

# Code Listings

6.1	Installation of OpenVPN client . . . . .	22
6.2	Startup script on VPN VM . . . . .	23
6.3	Installation of PiVPN . . . . .	23
7.1	qubeLogHandler function in splunkforwarder.py . . . . .	31
7.2	getFileandCopy function in splunkforwarder.py . . . . .	32
7.3	splunkforwarder.conf.json example . . . . .	32
7.4	dom0Handler function in splunkforwarder.py . . . . .	33
8.1	Install Windows 10 . . . . .	35
10.1	Script variables . . . . .	48
10.2	Configuration downloading script . . . . .	48
10.3	Create temporary virtual machine (VM) . . . . .	49
10.4	Cloning repository . . . . .	49
10.5	Copying to dom0 . . . . .	49

# Acronyms

**AD** Microsoft Active Directory. 5, 11, 38, 40

**AI** artificial intelligence. 30

**AWS** Amazon Web Services. 18, 40

**Azure AD** Microsoft Azure Active Directory. 34, 39–41, 43

**CPU** central processing unit. 8, 9, 51–55

**GUI** graphical user interface. 38

**IaaS** infrastructure as a service. 39

**IAM** identity and access management. 3, 6, 16, 17, 39–41, 43, 57, 69–71, 73, 75, 79, 80, 86

**ICMP** Internet Control Message Protocol. 22

**KCSC** Kongsberg Cyber Security Center. 1, 2

**KDA** Kongsberg Defence & Aerospace. 1–3, 5–7, 18, 20, 39, 43, 56, 57, 64, 65, 70, 71, 73, 75, 77, 79, 80, 86, 87, 93

**KVM** kernel-based virtual machine. 11

**MFA** multi-factor authentication. 5, 39, 43

**PaaS** platform as a service. 39

**QWT** Qubes Windows Tools. x, 33, 35–38, 43

**RPC** remote procedure call. 36

**SaaS** software as a service. 39, 54

**SIEM** Security Incident and Event Manager. 3, 5, 6, 11, 12, 15–17, 29–31, 34, 40, 57, 69, 70, 72, 73, 75, 80, 82, 86

**SSO** single sign-on. 6, 39–41

**TLS** Transport Layer Security. 22

**VM** virtual machine. xi, 5–9, 11–13, 15, 17–21, 23, 26, 27, 32, 35–41, 43, 46, 48, 49, 53, 57, 61, 69–73, 75, 77, 79–81

**VoIP** Voice over Internet Protocol. 93

**VPN** virtual private network. ix, 3, 6, 20, 22, 23, 25–27, 55, 57, 60–62, 69, 70, 73, 75, 77, 80, 87

# Glossary

**admin VM** A qube that has access to start and manage other qubes. 36, 44

**cron job** A script that is run periodically at set intervals on Linux systems, such as daily or hourly, via the built-in cron tool. 31

**disposable VM** A qube that is created from a template VM upon launch, and is deleted when the running application closes. Disposable VMs have no persistent storage. 36

**dom0** The virtual machine in Qubes OS that is responsible for management of the system, such as creating and managing qubes. Dom0 runs the toolchain that manages the Xen hypervisor and is the first virtual machine to be started by the Xen hypervisor [1]. Dom0 is thus the most ultimately trusted domain in the Qubes OS security model, and has full administrative access to the entire system. 19, 20, 30–34, 36, 38–40, 43, 44, 46, 49, 50, 54, 56, 69, 81, 82

**domain** According to the Qubes OS glossary a domain is "an area or set of activities in one's digital life that has certain security requirements and therefore involves the use of certain qubes" [1]. A domain is the Qubes OS-specific terminology for a zone. ix, 20, 22, 23, 25, 30, 31, 38, 43, 57, 59, 60, 69–71, 73, 75, 77–80

**dynamic analysis** Analysis of an executable file that involves executing the file on a live system and observing its behavior. This includes e.g. executing the file and monitoring its system calls, file system interaction, and interaction with other processes, as well as more advanced techniques such as altering the flow of execution through a debugger, or reading and modifying in-memory values set and read by the program. 7, 8

**hash** A short string generated as a result of running a cryptographic one-way hash function against some data input. A one-way hash function  $h$  is designed such that it produces a value  $h(X)$  for an input  $X$  where  $X$  can be arbitrary length,  $h(X)$  has a fixed length  $n$  where  $n \geq 128$ , it is "hard" to find  $X$  when only given  $h(X)$ , and it is "hard" to find an  $X' \neq X$  such that  $h(X') = h(X)$  [2]. 36, 46, 74

**host machine** The physical machine on which a hypervisor is installed. Through the hypervisor, the host machine runs one or more virtual machines. 5, 8, 9, 11, 16, 19, 23

**hypervisor** Software that creates and runs virtual machines. The hypervisor is responsible for providing virtual hardware that an operating system can interact with. 3, 6–9, 11–13, 15–19, 80, 81, 86, 95

**impact** The negative effects of a security event. This can be expressed qualitatively or quantitatively, but is a measure of the damage that would take place if a risk occurred. 46

**incident response** Methodology and actions taken by a company to respond to and handle an attack against their systems and infrastructure. 1, 29



- iperf3** A tool used to measure maximum download speeds between two nodes on an IP network. 87
- ISO** An archive file containing a full digital copy of the contents of a single optical media, such as an operating system installation disc. 35, 36, 38, 43
- key pinning** Retrieving the cryptographic public key used by a service that utilizes Transport Layer Security and storing that key as a hard-coded value within the configuration, or code, of a script or program. All connections made by the script or program to that service in the future will refuse to connect if the service does not present the same public key. 36
- malware** Malicious software, a catch-all term typically used to refer to any software designed to cause damage to a single computer, server, or computer network, whether it's a virus, spyware, et al. [3]. 1, 7, 8, 15, 18, 20, 29, 36, 57, 60, 77, 79, 80, 82
- malware analysis** The art of dissecting malware to understand how it works, how to identify it, and how to defeat or eliminate it [4]. 1, 3, 7, 18, 57, 69–71, 73, 75, 80
- NetVM** A qube that provides network access to other qubes. x, 60, 62
- OpenVPN** An open-source VPN application that encrypts the connection using the cryptographic library OpenSSL and authenticates through either SSL/TLS and certificates, or preshared keys. 22, 23
- PCI device** A device connected to the central processing unit in a computer using Peripheral Component Interconnect (PCI) or PCI-Express (PCIe). Such devices include both separate hardware devices that are connected to the system motherboard using a PCI or PCIe connector, as well as devices integrated into the motherboard itself. Examples of PCI devices include the processor chipset, network adapters, USB controllers and graphics processing units. 37, 38, 52, 54, 55
- PiVPN** An installer for OpenVPN and Wireguard used to quickly setup and harden a VPN server on Debian, often used for installing VPN servers on Raspberry Pi devices. 23
- ProxyVM** A qube that acts as both a network provider and a network client. The ProxyVM acts as the network gateway for other qubes connected to it, and forwards it on to the qube that the ProxyVM itself uses as its network gateway (typically sys-net). It thus acts as a proxy for network traffic. ProxyVMs are used to facilitate VPN connections in Qubes OS, by performing VPN encapsulation on network traffic that passes through it before it is sent to the upstream network qube. 23, 60
- qube** The Qubes OS-specific terminology for a virtual machine; an informal term used to make it easier for users to understand the Qubes OS architecture [1]. ix, 19, 20, 23, 30–38, 40, 43, 44, 46, 49, 50, 52, 54, 56, 57, 60–62, 69, 71–73, 75, 77–80, 82
- Qubes OS** An operating system designed with privacy and security in mind. ix, 3, 6, 15, 18–20, 22–24, 26, 30, 33–41, 43–46, 49–57, 59, 62, 65, 81–84, 86, 87, 93, 95
- regular expression** A sequence of characters that specifies a search pattern in a string of text. 33
- SkyHiGH** The self-provisioning cloud on campus at NTNU Gjøvik. 22
- standalone VM** A type of VM created by cloning a TemplateVM, but does not share root filesystem with other VMs. 31, 35, 43

**static analysis** Analysis of an executable file performed without actually executing the file. This includes e.g. looking for strings of text inside the file's metadata, attempting to reverse engineer or decompile the binary code of the executable, and graphing the flow of execution of the program based on how different parts of the binary code are linked together. 7, 8

**template VM** A qube that acts as a snapshot from which other qubes can be created. Template VMs contain full installations of an operating system, and forms the base layer upon which other qubes are built. 31, 35, 36, 61

**virtualization** The process of creating virtual instances of traditionally physical computing components, such as virtual machines. 2, 7, 9, 15, 17, 20, 81

**zone** A group of tools, data and activities on a digital platform that as a whole are expected to conform to a shared set of security requirements, and that should be kept isolated from other zones. For example, a business zone would be expected to contain confidential data, and may have broader access to the network than a personal zone, which would be intended for personal use and should not have access to sensitive or confidential business documents. 1, 5, 6, 11–13, 15–17, 22, 54, 56, 86

# Chapter 1

## Introduction

This chapter includes a brief background to our bachelor project, goals and limitations. A description of the thesis and group structure is also included.

### 1.1 Background

Kongsberg Group [5] is an international technology group supplying high technology systems and solutions to customers within the oil and gas, shipping, defence and aerospace industries. As of 2019, Kongsberg Group had more than 11,000 employees [6]. Kongsberg Defence & Aerospace (KDA) [7] is a wholly owned subsidiary of Kongsberg Group, consisting of seven business areas. KDA has 3,100 employees, and is a supplier of defence products, systems for command and control, surveillance, space, tactical communications and more [8].

Kongsberg Cyber Security Center (KCSC) is a department within the Kongsberg Group that handles analysis, detection and incident response for all divisions of the group. This type of work necessitates the ability to connect to multiple environments to perform various tasks, such as evidence gathering, log collection and incident handling. KCSC's current routines involve using machines and tooling from each environment, which is cumbersome and unproductive for the analysts.

To prevent usage of a large number of different machines and inconsistent methods for secure file transfer, it is desirable to have a single laptop to perform these different tasks on. The laptop would be connected to multiple zones, where each zone can be broadly described as a group of tools, data and activities that as a whole are expected to conform to a shared set of security requirements. For example, a business zone would be expected to contain confidential data, and may have broader access to the network than a personal zone, which would be intended for personal use and should not have access to sensitive or confidential business documents.

Because malware analysis often include executing live malware, this laptop must have precautions against executing malware in enterprise zones, and at the same time, it should be able to securely transfer data (such as logs, reports and malware samples) from a secure zone to an insecure zone.

## 1.2 Project Goals

Overcoming the aforementioned challenges and implementing this kind of platform, is where our multipurpose platform for security analysts comes in. The goal of our bachelor project can be divided into three main parts.

The first goal is to come up with a design that encompasses and answers the research questions stated by KDA. This involves research into which solutions currently exist on the market, evaluating them against each other to identify pros and cons, and then selecting the solutions that best meet the requirements of KDA.

The second goal is to implement the most promising set of solutions. Specifically, this involves installing the operating system chosen in the design stage on laptops provided by KDA. After operating system installation, the laptops will then be configured, and additional software will be installed to fulfill the technical requirements specified by KDA.

The third and final goal is to evaluate how well our final implementation meets the requirements specified by KDA. This involves matching the implementation against the requirements specification, identifying potential short-comings, performing a security assessment of the implementation, and discussing possible future improvements.

## 1.3 Limitations

The final product of this project consists of an operating system installation with configuration, that is installed on KDA's provided laptops, and a detailed description of the components and setup of this configuration. The configuration is to be considered as a prototype and proof of concept, and as a basis from which the system can be further configured to meet additional future requirements by Kongsberg. The purpose of this proof of concept is to explore whether or not a multipurpose platform for security analysts is feasible, in order to meet strict security requirements while still serving as a useful tool for analysts. This evaluation is also part of the final product.

The project is limited to the specific requirements provided by KDA, and only explores existing solutions on the market. Development of new systems and tooling is out of scope. The platform is not created to meet the expectations of a production-grade system. Furthermore, the platform is only installed on the specific laptops provided by KCSC. Therefore, compatibility with other laptops, or computer hardware in general, is not tested or guaranteed.

## 1.4 Project Group

The project owners are David Lee Andersen and Erlend Hammer from KDA. The supervisor of the bachelor group is Jia-Chun Lin, assistant professor at NTNU. The project group consists of four bachelor students from the IT Operations and Information Security study program. The students have a varied skill set within information security, operating systems knowledge, and understanding of Linux and virtualization.

## 1.5 Thesis Structure

### **Chapter 1 - Introduction**

Description of the project's background, goals, limitations and thesis structure.

### **Chapter 2 - Research Questions and Work Breakdown**

Research questions given to us by KDA, and a high-level work breakdown to address these questions.

### **Chapter 3 - Background**

Background knowledge which is relevant for why the project is helpful.

### **Chapter 4 - Investigation of Hypervisors**

Comparison of various hypervisors available on the market, to determine the hypervisors that works best for our purpose.

### **Chapter 5 - Technical Design**

Technical design of the functionality and setup of the platform.

### **Chapter 6 - Networking**

Implementation of networking and virtual private networks (VPNs).

### **Chapter 7 - Security Information and Event Management**

Explanation of the purpose of a Security Incident and Event Manager (SIEM) and the integration of the Splunk SIEM in the platform.

### **Chapter 8 - Windows in Qubes OS**

Discussion on the compatibility between Windows and the Qubes OS hypervisor.

### **Chapter 9 - Identity and Access Management**

Implementation of an identity and access management (IAM) service in the platform.

### **Chapter 10 - Configuration Management**

Discussing ways of implementing configuration management in our hypervisor.

### **Chapter 11 - Qubes OS Hardware Compatibility**

Discussion on hardware configurations used for the proof-of-concept implementations of the platform, and hardware-related challenges and compatibility issues the group has faced.

### **Chapter 12 - Risk Assessment of the Multipurpose Platform**

A comprehensive risk assessment of the system as a whole.

### **Chapter 13 - Evaluation**

An evaluation of how well the resulting platform meets the requirements set by KDA, including how well suited the platform is for performing malware analysis, and a risk assessment of the platform.

### **Chapter 14 - Discussion**

A discussion of the end result and work done in the thesis.

**Chapter 15 - *Development Process***

The development process that the group has used during the development of the platform.

**Chapter 16 - *Closing Remarks***

Project conclusion, and suggestions for future changes and improvements.

## Chapter 2

# Research Questions and Work Breakdown

In this chapter, we will describe all research questions that were given by KDA. We will additionally give a high-level work breakdown on how we plan to address the research questions.

KDA has provided us with four laptops, and they would like us to explore the possibility to run a multi-purpose platform on each of them. The laptops should be configured such that different applications are launched on dedicated VMs in different security zones. At least four different security zones are expected. These are listed as below:

- A work zone in which Microsoft Teams will be used.
- A analysis zone in which malware will be run and analyzed.
- A development zone in which software can be tested.
- A surfing zone through which the user can surf the web.

These zones should be appropriately configured based on their required level of access to the Internet, as well as to Kongsberg's intranet. In addition, the multipurpose platform must be able to transfer system logs to a Security Incident and Event Manager (SIEM), which is a type of system that consolidates security logs and events from different sources in an enterprise network, and stores, normalizes and correlates them in order to identify malicious activity in real time [9]. Furthermore, we were asked to evaluate and document how the zones are isolated from each other and from the host machine.

Based on the above description, the main research question in this project is to evaluate the feasibility of realizing such a multipurpose platform on the four assigned laptops, and to provide suggestions on how this would work in practice. This evaluation should also contain a risk assessment. Furthermore, KDA hopes that we can document all non-trivial code to further support development of the project at KDA.

In addition, KDA has provided a list of optional features that would be nice to have in our proposed multipurpose platform, but which are not strictly required:

- Support for multi-factor authentication (MFA).
- Implementation of a location policy that bans travel considered to be impossible, such as moving between physical locations unreasonably quickly (implemented via geo-IP resolution).
- Integration of identities with Microsoft Active Directory (AD) [10].
- Centralized configuration and patch management via e.g. SaltStack [11], Puppet [12] or Ansible [13].
- An evaluation of the compatibility and performance of the project implementation against the laptops provided by KDA.

## 2.1 High-level work breakdown

To answer the research question mentioned in the previous section, we break down our research work as listed below:

1. Investigate current hypervisor solutions on the market and choose one that best meets the requirements of KDA. Chapter 4 details our investigation on state-of-the-art hypervisors and the hypervisor we chose for this project.
2. Implement the chosen hypervisor on each of the four laptops to create the different zones required by KDA. Chapter 5 introduces the technical design of the proposed multipurpose platform.
3. Make a working proof of concept for how to connect the VMs to different virtual private networks (VPNs). Chapter 6 describes our research into the concepts behind a VPN, and how to integrate it into the hypervisor's network architecture.
4. Connect our proposed multipurpose platform to a configuration management system, so that it can be centrally managed by IT staff at KDA. Chapter 10 introduces some configuration management systems on the market that are relevant to Qubes OS, and discusses possible designs for such a system along with some of the difficulties in doing so on our platform hypervisor.
5. Connect our proposed multipurpose platform to a Security Incident and Event Manager (SIEM). Chapter 7 details our investigation into this topic and the steps needed to successfully collect and send data to a SIEM.
6. Install the Windows operating system on the chosen hypervisor, which is primarily designed for Linux. Chapter 8 explains how this was solved in practice, discusses compatibility issues, and compares the pros and cons between different ways of installing Windows.
7. Integrate our multipurpose platform into an identity and access management (IAM) for single sign-on. Chapter 9 introduces some of the IAM solutions available on the market today, and showcases how one of them can be integrated into our platform.
8. Compare the different laptop models that were provided by KDA to identify suitable minimum system requirements for our multipurpose platform. Chapter 11 details our investigation into the system requirements of the hypervisor, as well as our own experience using the platform on the different laptops, in order to identify a minimal set of system requirements and recommendations for KDA.
9. Perform a risk assessment of the multipurpose platform. Chapter 12 identifies the various assets, use and abuse cases, and threat actors that affect the multipurpose platform, and consolidates this into a comprehensive assessment of the risks and security of the platform.



# Chapter 3

## Background

This chapter will cover the existing solutions used for malware analysis, and what components KDA currently requires as part of their analysis process. It will give an introduction to concepts often used in malware analysis, such as the definition and inherent risks of working with malware, and an introduction to commonly used components, such as virtual machines, hypervisors and virtualization.

### 3.1 Malware

Malware can broadly be described as any software designed to cause damage to a single computer, server, or computer network, such as viruses, spyware, etc. [3]. The motivation behind creating such software is usually for financial gain, for acquisition of sensitive data and trade secrets, or to intentionally cause disruption of a business for political or economical reasons [14–16].

There have been many high-profile malware attacks in recent years that have left many businesses and agencies unable to operate for extended periods of time. A good example of this is the WannaCry ransomware [17], a form of malware designed to hold a computer or computer system ransom. In a matter of days, WannaCry infected more than 230,000 computers in 150 countries, affecting large scale organizations such as the British National Health Service, impacting their ability to provide health care, among others [18].

### 3.2 Malware analysis

As cyber attacks increase in frequency and complexity, understanding how malware works is imperative to formulating a response plan, and to understand and analyze malware authors' intentions, and the extent of damage caused by malware to a system. To that end, malware analysis has become a common topic of research for security specialists. Malware analysis can be defined as the art of dissecting malware to understand how it works, how to identify it, and how to defeat or eliminate it [4], and can broadly be split into two main types of analysis: static analysis and dynamic analysis.

### 3.2.1 Static analysis

Static analysis is the process of analyzing the functionality of an executable program without actually running the program [19], thus preventing the program from causing damage to the system. An everyday analogy to this would be to look at an unknown baking recipe and using the ingredients and steps to formulate a hypothesis about what the final product would look like, without actually baking the product.

Static analysis usually involves using specialized tooling to look at the metadata of the malware sample, extracting data such as the operating system functions the sample calls, strings of text contained within the sample that may hint at functionality or servers it communicates with, and possible identifying information about the author of the malware sample [4]. As static analysis does not require executing the malware sample, the risk of the sample causing damage to the environment is generally limited to the analyst's ability to avoid executing the malware sample by accident.

### 3.2.2 Dynamic analysis

As opposed to static analysis, dynamic analysis involves actually running the suspected malicious program and analyzing its behavior [19]. A more fitting analogy for this process is following the steps of a baking recipe, making notes on how the product develops throughout the process, and then observing the end result once the baking process is completed.

Dynamic analysis involves usage of tools that can extensively collect and store system calls and changes to the file system and settings registry, and modify or halt the execution of a program while it is running [4]. As running malware on a live production system is extremely risky, analysts frequently use virtual machines (VMs) for this purpose. VMs are explained in detail in Section 3.3. In short, they allow the sample to run on an operating system installation that is completely separated from the host machine. Most VM hypervisors allow its users to create snapshots of VMs at any given time, so the analyst can roll back to essentially undo the malware sample ever being executed. This allows execution of the malware sample repeatedly from a consistent, untouched environment, allowing the analyst to observe how changes to the environment, and to the sample itself and its memory, changes the malware sample's behavior.

## 3.3 Virtual machines

A virtual machine (VM) is a software-based virtual instance of a computer system that runs on top of a real physical system [20]. Most of today's commonly used operating systems can be installed as a VM, such as Windows, Linux, BSD, Solaris and more, spanning different versions and distributions. The VM will have delegated access to the host machine's central processing unit (CPU), memory, and other resources, such as network connectivity and storage. These resources are separated from the host machine and other VMs and provisioned by a hypervisor, which will be introduced in the next section. Resources allocated to a VM can easily be reallocated to other VMs or provisioned to new VMs [21–23].

VMs are by default isolated from the host machine and each other, but the hypervisor can allow them to communicate, e.g. by sharing files or folders between each other. VMs are designed to be easily moved, removed, destroyed and rebooted and reverted to a previous state. They are therefore most commonly designed only to serve one function, usually working in tandem with other VMs to provide a more encompassing service. Separating VMs based on functionality allows for easier troubleshooting, improved security and better cost efficiency at the expense of higher overhead, i.e. more system resources such as processing power and memory capacity needed to achieve the desired functionality.

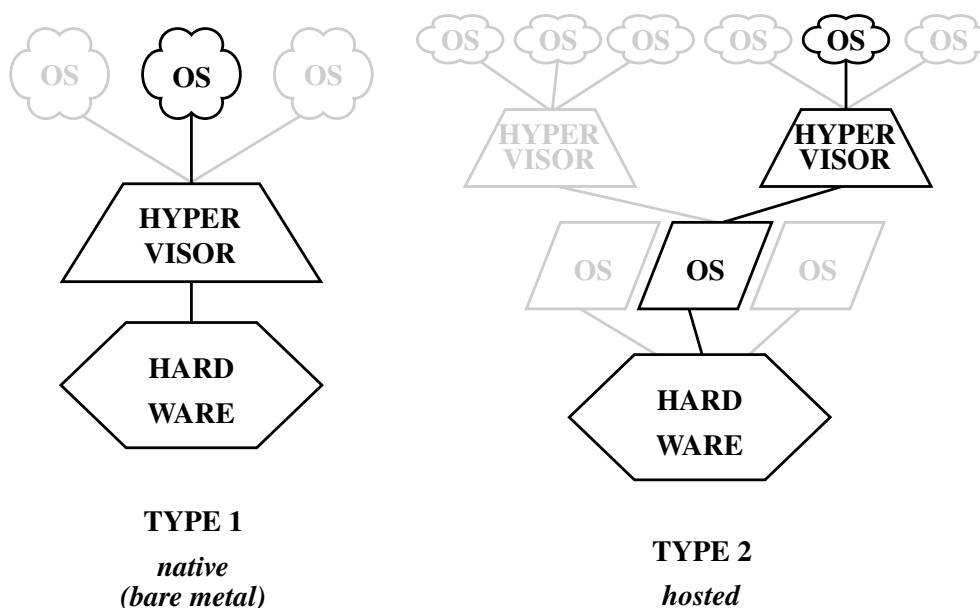
### 3.4 Virtualization and hypervisors

Virtualization is a technology which allows an environment to create multiple virtual environments (e.g. VMs, computer systems) that emulate traditionally physical hardware, such as computer components and dedicated resources (USB, microphones, speakers). Each VM runs its own operating system on top of virtual hardware, effectively allowing for multiple different operating systems running on a single physical machine [21–23].

This allows a user to make use of tools across different operating systems quickly and efficiently. Rather than booting up different physical machines with different operating systems, a user can use one machine for all tasks. All processes inside a VM takes place inside an isolated environment, meaning that they cannot communicate or affect other VMs or the host machine. This is especially useful when running untrusted software, as all potential damage is contained within a single VM and can be easily removed or reverted.

A hypervisor is the software that initiates the virtualization process and allows creating and running VMs. There are two types of hypervisors [23], as illustrated in Figure 3.1:

- Type 1 runs directly on the physical hardware of the host machine, and is commonly referred to as a bare-metal hypervisor.
- Type 2 runs as an application inside a host operating system, and it can be started and stopped like a regular program.



**Figure3.1:** Overview of the two types of hypervisors [24]

A hypervisor is responsible for resource sharing between host machine and VMs, and the VMs themselves. The hypervisor treats all host resources (CPU, memory, storage etc.) as a pool of resources which can be delegated to the VMs depending on use. The operating systems running on the VMs may require a minimum set of resources needed to operate, but beyond that, the resources they need can be customized depending on their usage.

Some hypervisors (VMware, XenServer, Xen) also allow VMs to use more resources than is available by allowing the VMs to share allocated resources between them. Dynamic allocation allows the hypervisor to set minimum and maximum values for storage and memory, the range of which will change dynamically depending on the resource requirements of the VMs [25, 26].

## 3.5 Configuration Management

Configuration management can be described as “a process for maintaining computer systems, servers, and software in a desired, consistent state. [...]. Managing IT system configurations involves defining a system’s desired state—like server configuration—then building and maintaining those systems.” [27] There are open source tools built to assist in these tasks. Some of these tools, e.g. Ansible [13], Puppet [12], and Salt [11] work by writing machine-readable code which describes the desired configuration state. This state is compared to the current system state, then files, users, etc. described in the desired state will be added if they are missing. A daemon running on the client is required by some of these products.

## Chapter 4

# Investigation of Hypervisors

There are many different hypervisors on the market, and we will discuss a selection of them below. To comply with the requirements specifications of the project, we have to set some requirements for the hypervisor we choose.

- **Type:** If the host machine of a type 2 hypervisor gets compromised, all the VMs on that machine could also be compromised. This security risk is mitigated by choosing a type 1 hypervisor. To best ensure separation between VM and host machine, and the ability to work in different zones at the same time, the underlying hypervisor should be type 1 [28].
- **Zones and previous work:** The hypervisor needs to have some notion of previous work akin to our project, creating a new operating system is not the goal of this project. This previous work must allow for the ability to create isolated zones of varying trust levels within one laptop, the ability to work in several zones simultaneously, and to securely transfer files between the zones.
- **Operating systems:** The chosen hypervisor needs to be able to host, at minimum, a recent Windows version (version 7 or 10), recent Linux distribution versions (Ubuntu 20.04, Red Hat Enterprise Linux, Arch Linux), and, if possible, FreeBSD and/or OpenBSD.
- **Logs:** Must be able to connect and ship host-logs to a Security Incident and Event Manager (SIEM).
- **Identity service:** Should be able to connect to some form of identity service such as FreeIPA or Microsoft Active Directory (AD).
- **Central management:** Be configurable by a well known declarative configuration language such as SaltStack, Puppet or Ansible.
- **Hardware support:** Hypervisor supports on-market, recent hardware and can provide a satisfying user experience.

The above requirements limit our options to hypervisors with newer releases and support, either commercial or open source.

There are several free and open source hypervisors, such as kernel-based virtual machine (KVM) and Red Hat Enterprise Virtualization (RHEV, a commercialized implementation of KVM), to name a few.

The free and open source hypervisors mentioned above, KVM and Red Hat Enterprise Virtualization would not fit our project as it has no notion of work done on top of the hypervisor. The aim for this project is not to create our own operating system.

We will look at a few of the most widely used type 1 hypervisors in the following sections. We will then investigate any previous work based on those hypervisors, that offers the ability to create the type of zones we need for our project, and consider which ones can fulfill the remaining requirements.

## 4.1 Xen Hypervisor

**Type:** Type 1

**Zones and previous work:** Bare-bones hypervisor, does not allow for zones.

**Operating systems:** Can create VMs of all required operating systems

**Logs:** Can connect to SIEM

**Identity service:** Can connect to identity service

**Central management:** Can be managed through third party software

**Hardware support:** Has support for the latest and purchaseable hardware components

Xen Hypervisor is a type 1, open source hypervisor maintained and developed by the Xen Project community under the Linux Foundation [29]. The Xen code is used as a basis for multiple projects and security solutions, several of which are relevant candidates for our project.

The Xen hypervisor has limited support for guest operating systems. While it supports most of today's commonly used operating systems, some older operating systems such as Solaris do not work.

## 4.2 Star Lab Crucible / Star Lab Titanium Secure Hypervisor

**Type:** Type 1

**Zones and previous work:** Work built on the Xen hypervisor, allows for zones.

**Operating systems:** Supports Linux, Windows and VxWorks [30].

**Logs:** Unknown

**Identity service:** Unknown

**Central management:** Unknown

**Hardware support:** Unknown

The Xen based [30] Crucible/Titanium Secure Hypervisor [31] is developed by the American company Star Lab Software, which specializes in secure combat systems [32].

The hypervisor provides security by separation and isolation. Offering isolated zones which they call execution and service domains. Malicious code in one domain will not be able to affect a different domain, but the domains are still able to communicate with each other. It is possible to enforce mandatory access control policies on this inter-domain communication [30].

To minimize the attack surface, they have removed Xen features that they consider non-essential. Their secure boot functionality ensures that sensitive application software cannot be decrypted on non-authorized, instrumented or modified hardware. The hypervisor provides fail-over function (redundancy/fault tolerance) in case of suspected compromise, and maintains trusted snapshots for quick recovery when fail-over is not possible. Major releases with new functionality are released every 6 months [30].

According to their whitepaper, the hypervisor was expected to be approved for 6 National Information Assurance Partnership (NIAP) [33] protection profiles in 2019. This approval will allow it to be used for systems that requires compliance with the Multiple Independent Levels of Security (MILS) separation

of kernel and hypervisor, and with the Commercial Solutions for Classified (CSfC) data-at-rest (file-based, and software full disk encryption for Linux) [30]. We have not found confirmation on the approval, however, according to their data sheet their solution address 100% of NIST 800-53 operational controls for federal systems [34]. With this, the solution looks like a very good candidate to fulfill several of our research questions.

Crucible/Titanium Secure Hypervisor is not an open source solution, and we have found little documentation beyond the data sheet and whitepaper from online sources. We have contacted Star Lab via email requesting more information, but at the time of writing we have not received an answer. Due to the lack of information, we cannot choose this solution for our project.

### 4.3 AIS SecureView

**Type:** Type 1

**Zones and previous work:** Work built on the Xen hypervisor, allows for zones.

**Operating systems:** Windows, Linux and Solaris [35]

**Logs:** Unknown

**Identity service:** Unknown

**Central management:** Unknown

**Hardware support:** Unknown

SecureView [36] is a Xen based solution [37] developed by the American software company Assured Information Security (AIS). It allows for the user on a single computer to simultaneously access multiple applications on various VMs belonging to different zones. The VMs can be divided into isolated zones with varying levels of classification [36, 37]. In the overview published in 2015, it is stated that “SecureView is NIST 800-53 certified as High in both Confidentiality and Integrity, and Medium in availability” [37]. As of 2019, it supports Windows, Linux and Solaris [35].

Although this looks like a very promising solution for our project, it is developed for the U.S Air Force, and is Government-off-the-Shelf software [35], and is only available to users from the U.S. Department of Defence or the U.S. Intelligence Community, and can only be exported to five eyes nations (Canada, UK, Australia and New Zealand) [B.1].

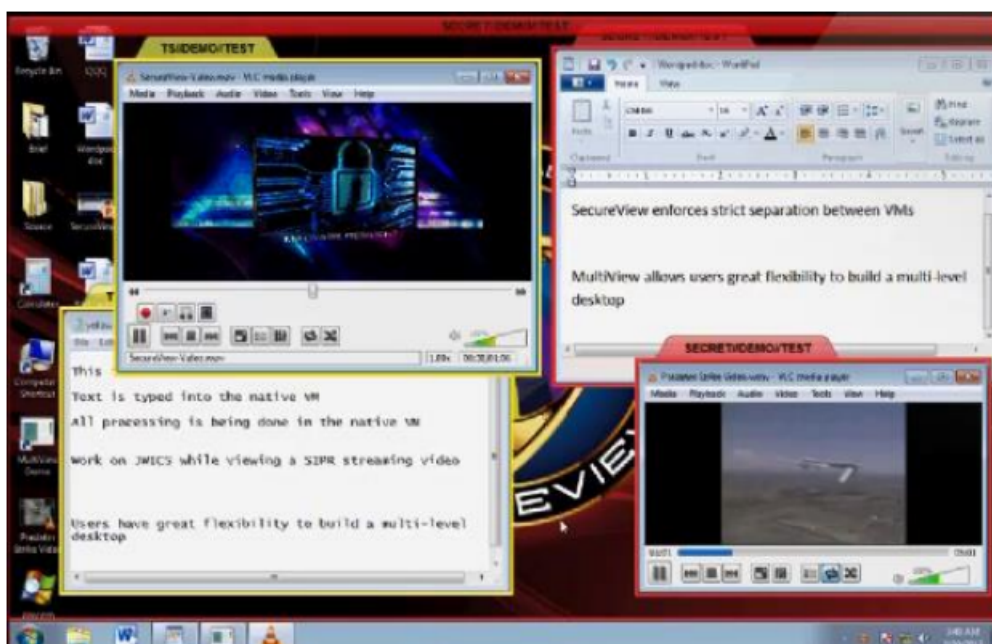


Figure4.1: SecureView desktop [37]



## 4.4 HP Sure Click

**Type:** Unknown

**Zones and previous work:** Lets the user safely open attachments and websites in micro-VMs, but does not allow for zones.

**Operating systems:** Windows

**Logs:** Unknown

**Identity service:** Unknown

**Central management:** Unknown

**Hardware support:** Unknown

Sure Click [38] is developed by the HP acquired company Bromium. Their micro-virtualization technology allows for users to securely perform tasks such as opening emails, downloading files and browsing the web, by ensuring that untrusted attachments, browser tabs, links etc. are opened in isolated micro-VMs. If a micro-VM is compromised, the threat is contained inside the VM and it can easily be disposed of by closing the attachment or window [39]. According to the whitepaper and website [38], there has not been any documented malware escapes or reported breaches.

Using Sure Click Enterprise, administrators can set access control policies for both users and user groups [40]. This solution allows for users to open files with potentially malicious content without any risk to their system, but it does not offer the option to create multiple security zones. In addition to this, HP Sure Click only supports Windows 10 [39], which means that it is not the right solution for this project.

## 4.5 sVirt

**Type:** Unknown

**Zones and previous work:** Bare-bones hypervisor, does not allow for zones

**Operating systems:** Can create VMs of Linux distributions

**Logs:** Can connect to SIEM

**Identity service:** Can connect to identity service

**Central management:** Can be managed through third party software

**Hardware support:** Unknown

sVirt [41] is an open source hypervisor implemented in Red Hat Linux. Co-developed by the American National Security Agency, it focuses on security by isolating each VM as protected processes, with rules to govern what can be accessed. It is incorporated in the SELinux (Security-Enhanced Linux) project as its hypervisor.

Kongsberg may find sVirt a better solution than Xen. However, there is little documentation on the hypervisor, and it is not open source. Compared to Qubes OS, a lot more research and work would be needed to create a working operating system, as well as the software vulnerabilities that may or may not exist in the sVirt hypervisor, as well Red Hat Linux vulnerabilities. As sVirt lacks documentation and support it is far more likely exploits and vulnerabilities will be available for attackers before sVirt will be updated and/or

patched. Partners may not even be notified, and in the worst case, patching and security will become a work task for Kongsberg themselves.

Recently, as of writing, the SELinux project has concerning security problems, for example the recently found vulnerability in SELinux Permissive mode. The exploit<sup>1</sup>, named Magica, can be installed into any arbitrary application, giving the application, thus attacker, permanent root access to the machine<sup>2</sup>. The security issues should be taken into account before choosing sVirt.

### 4.6 Hysolate

**Type:** Architecture suggests that it is type 1

**Zones and previous work:** Work based on Hyper-V, currently allows for two zones

**Operating systems:** Currently it only supports Windows 10, a windows workspace on MAC will be possible in the near future [42]

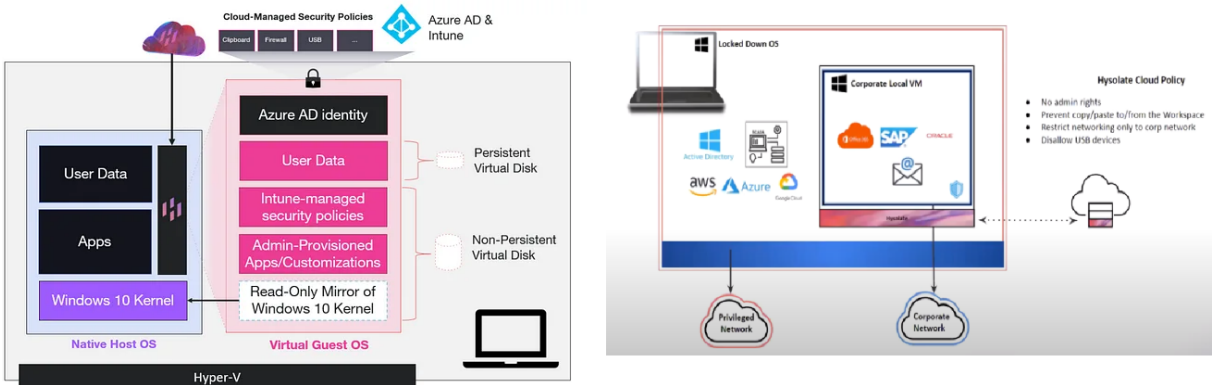
**Logs:** Can connect to SIEM [42]

**Identity service:** Can connect to identity and access management (IAM) service [43]

**Central management:** Can be managed from the cloud [44]

**Hardware support:** Unknown

Hysolate [45] is a fairly recent project based on the type 1 [46] hypervisor Hyper-V, which allows separation of environments on a Windows 10 machine. As it is designed for end-user machines, it allows for a better user experience and more security for the average user. The solution consists of a host and a workspace, and only one Windows license is needed [44]. As of January 2021, it is only possible to deploy one workspace, but several workspaces might be possible in the future. Copy and paste policies can be assigned between the host machine and the workspace, and the solution can be configured so that when the user tries to open specific applications, websites and files from within the host machine, it will be automatically be redirected and opened in the workspace [42]. The user can either switch between the host and workspace in full screen, or view both windows side by side [47]. The architecture figure and documentation suggests that this is a type 1 hypervisor, but we have not found any official documentation where they state the hypervisor type.



**Figure4.2:** Hysolate architecture and Hysolate desktop [42, 43]

<sup>1</sup> <https://github.com/vvb2060/Magica>  
<sup>2</sup> <https://twitter.com/topjohnwu/status/1359054106019565571>

Our limitations in the project leads us away from such a solution, as at the time of writing the project can only support Windows 10 virtualization, and only one VM. Our proof of concept is required to run several VMs with at least later versions of Windows and Linux.

## 4.7 PolyXene

**Type:** Type 1

**Zones and previous work:** Allows for separate applications of different security levels

**Operating systems:** Windows and Linux

**Logs:** Logging is possible

**Identity service:** Role-based access control

**Central management:** Centralized administration possible

**Hardware support:** Unknown

PolyXene [48] is a bare metal hypervisor [49], designed to separate applications of different sensitivity levels on the same work station. It includes functions such as role based access control, centralized administration and management, logging of security events, as well as automated deployment and updates. The hypervisor supports both Linux and Windows VMs [49].

According to their own website, PolyXene is developed as a collaboration between the software solutions provider Bertin IT and the French arms procurement agency DGA, and has been used by the French army headquarters since 2014 [48].

In 2009 version 1.1 of the product was granted the Common Criteria Evaluation Assurance Level 5: “semiformally designed and tested” [50] [51].

A report made by FFI in 2015 [52], suggests that PolyXene is based on the Xen hypervisor, but the sources used in the report are no longer available. We could not confirm this from Bertin IT’s current documentation.

PolyXene is not an open source solution, and we have found little documentation from online sources. We have contacted Bertin IT via email requesting more information, but at the time of writing we have not received an answer. Due to the lack of information, we cannot choose this solution for our project.

## 4.8 Qubes OS

**Type:** Type 1 hypervisor (through Xen hypervisor)

**Zones and previous work:** Allows for different zones.

**Operating systems:** Can create VMs of all required operating systems

**Logs:** Can connect to SIEM (we will discuss this in depth in Chapter 7)

**Identity service:** Can connect to IAM service (we will discuss this in depth in Chapter 9)

**Central management:** Can be administrated by SaltStack or Ansible (we will discuss this in depth in Chapter 10)

**Hardware support:** Has support for some of the latest and purchasable hardware components (we will discuss this in depth in Chapter 11)

Qubes OS is a free and open source operating system that is designed specifically with security in mind, and is based on the Xen hypervisor [53]. As Qubes OS meets all our demands, we will explain it in details in the coming chapter.

## 4.9 Cloud-based solutions

We do not consider cloud-based solutions for our project due to limitations set by KDA. We believe that in theory, cloud would be a better solution, as multiple malware analysts would be able to work on the same sample, the VMs would be easier to start, kill and revert, there is better configuration management, and it would be more cost-efficient (considering a central server for all workers instead of each employee having a rather expensive laptop). We will discuss different cloud technologies below, but they will not be relevant to the project.

It is possible to use cloud VMs in the form of Desktop as a Service, from known service providers such as Microsoft Azure Windows Virtual Desktop, Amazon WorkSpaces and IBM Cloud. Amazon Web Services (AWS) does not state malware analysis as either allowed or disallowed in their terms of service (such as running malware on Amazon servers), but the transmission of malicious files is prohibited within the AWS environment [54]. Users report they receive warnings having malicious programs run in their instances. Special forms and applications must be sent to AWS to make them aware of the nefarious traffic, which makes AWS unsuitable as a platform for malware analysis.

Azure Windows Virtual Desktop allows malware on their platform, they do however note that such a solution is not a reliable method considering the nature of the malware. The malware may disconnect Remote Desktop Connections and then kick the user out of the analysis platform with no way to get back in other than killing the platform or reverting to a previous state [55]. An on-premise solution, with for example vSphere, would allow a local connection to rescue the lost VM because of a constant console connection.

There are also several providers for cloud malware analysis platforms, such as antMan. antMan functions essentially the same as vSphere or OpenStack, but with more focus on security on the virtualized machine and custom networks [56]. It is a paid solution.

## 4.10 Investigation summary

There are many hypervisors to choose from. Commonly they all lack the work already included in Qubes OS. Similar solutions can only be found in cloud solutions at the time of writing. It would be possible to work from the hypervisors discussed and create a similar tool to Qubes OS, however, that is not the focus of this thesis.

Cloud is a good solution with already built frameworks ready to deploy in a user friendly environment, but could not be considered in this project due to limitations from KDA.

We therefore chose the Xen based hypervisor Qubes OS as the tool for our project, as it meets all of our requirements.

# Chapter 5

## Technical Design

In this chapter, we will describe the technical design of the proposed multipurpose platform. This includes an introduction to the chosen hypervisor Qubes OS, as well as the interaction between Qubes OS, the Internet, and our central management server.

### 5.1 Qubes OS

As mentioned in Section 4.8, Qubes OS is a free and open source operating system that is designed specifically with security in mind [53]. Qubes OS uses Xen, which is a Type 1 hypervisor and is able to run directly on the host machine. All operating systems installed on the machine, including Qubes OS itself, run in VMs, known in Qubes OS terminology as qubes. Qubes in Qubes OS, as explained in the Qubes OS documentation [53], have the following properties:

- Specific purposes: Each qube is designed to be used for a specific purpose, for example, personal use, work-related activity, or online shopping.
- Specific natures: The type of VM, such as a fully-fledged, or a stripped down VM.
- Specific levels of trust: As perceived by the user from a security perspective, based on how hard they would be to compromise, and their functionality. For example, the operating system management VM may be completely trusted, while the VM responsible for managing USB devices may be less trusted, due to the potential of security issues in the machine's USB controllers. The level of trust should affect how the qube has access to resources such as hardware components and networks.

Figure 5.1 shows how various qubes (or VMs) in Qubes OS interact each other, the hypervisor, and the host machine hardware. The different types of VMs are explained in more detail in Section 5.2.

Qubes OS enforces strong isolation between VMs, other VMs and the host machine:

- Every application runs in a qube, and each qube only has access to the resources that the Qubes OS user explicitly grants access to.
- The system is managed via dom0, which is the highest privileged VM in Qubes OS. Dom0 is started by the hypervisor directly on boot, and can manage Xen via hypervisor management toolchains.
- Network hardware is assigned exclusively to the `sys-net` qube, which is used as the upstream network gateway for the `sys-firewall` qube, which handles firewall rules. Other qubes must then be configured to use `sys-firewall` as its gateway to have network connectivity.
- All USB controllers are exclusively assigned to `sys-usb` by default. The user can then grant access to individual devices for a specific qube via dom0.

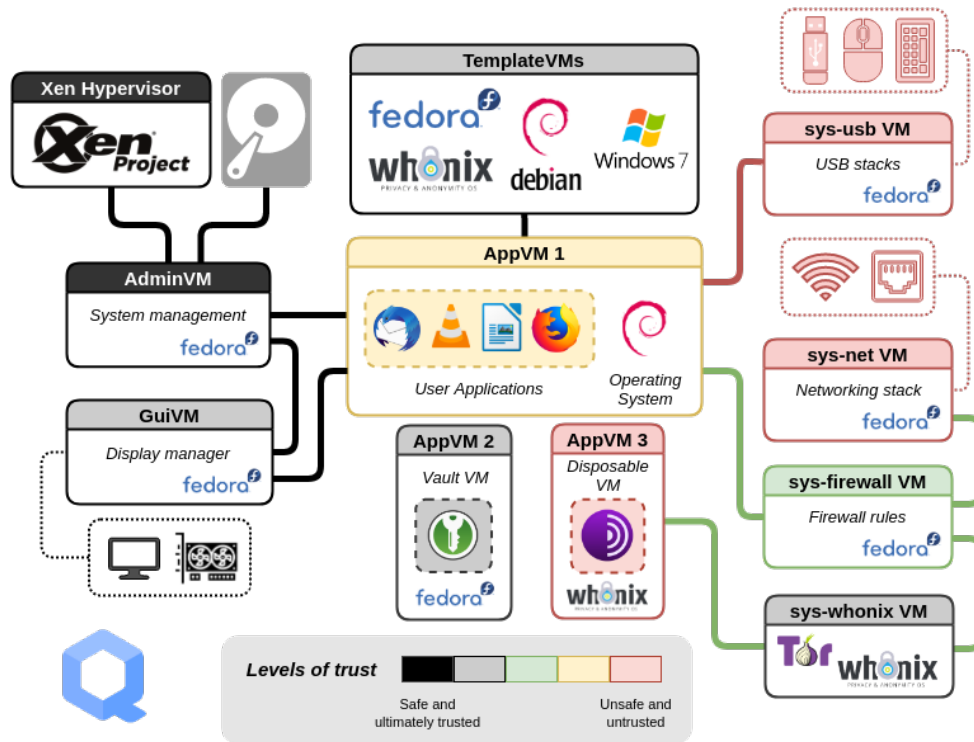


Figure 5.1: The different qubes in a typical Qubes OS installation [53]

For example, if the user wants to give one qube access to USB devices, they have to explicitly choose to connect that device from `sys-usb` to the qube in question using the Qubes OS management interface in `dom0`.

Figure 5.1 also shows usage of a `sys-whonix` VM. In this configuration, the qube “AppVM 3” is configured to use `sys-whonix` as its network gateway. `sys-whonix` is a qube that itself uses `sys-firewall` as its upstream gateway, but is configured to pass all traffic through the Tor network before passing it on to downstream qubes. “AppVM 3” thus has all of its network traffic tunneled through Tor. Such network rules are configured on a per-qube basis. This is also how networking features such as VPNs can be used in Qubes OS, which is explained more in depth in Section 6.4.1.

## 5.2 Virtual machines

The main feature of the laptop configuration is the virtualization. Qubes OS allows us to have multiple domains that are segmented from each other, but still have some interconnectedness. The types of VMs used in the project will be described here.

Each laptop has Qubes OS installed. This operating system controls VMs for the following functionality:

- A VM used for applications such as Microsoft Teams. This VM will have access to the Internet.
- A domain with multiple VMs in which malware can be run and analyzed. VMs in this domain are connected not to the Internet, but to a network on KDA premises through a VPN tunnel.
- A domain with multiple VMs in which software products can be developed and tested. VMs in this domain are connected not to the Internet, but to a network on KDA premises through a VPN tunnel.
- Two VMs to act as VPN proxies for the two domains mentioned above.
- A VM that can be used to surf the web. This VM will have access to the Internet.

- A VM to collect and send out logs to Splunk.

These VMs are based on template files.

In order to centralize configuration, a repository storing configuration management code is required.

A server running Splunk software is where our logs are sent.

## Chapter 6

# Networking

One of the research questions of this project requires figuring out whether two zones can each be isolated from the Internet, but have access to files on their own private networks. The machines in the analysis zone need to access the potentially malicious files they are meant to analyze. This is done via configuration of VPNs.

A virtual private network (VPN) can be simply explained as a “private network constructed within a public network infrastructure, such as the global Internet” [57]. When a client connects to a VPN, it establishes a connection to a server that is accessible on a broader network, such as the Internet, and tunnels traffic to that server, which in turn routes that traffic to other clients that are also connected to the same server, using standard IP routing.

We used SkyHiGh [58], the self-provisioned cloud at NTNU Gjøvik, to create two VPN servers. We then made one client configuration for each of these two servers, allowing the domains that need access to a VPN to connect to such a network, and then sending Internet Control Message Protocol (ICMP) pings to test network connectivity from each domain.

### 6.1 OpenVPN

There are multiple good VPN technologies out there. We chose OpenVPN, an open source application using the cryptographic library OpenSSL for encryption, offering authentication via Transport Layer Security (TLS) and client certificates, or through usage of usernames and passwords [59]. It is well documented, and we have previous experience with it, making it a suitable choice for testing VPN configuration in Qubes OS. The following commands install OpenVPN and setup a directory in which certificates can be placed and used.

**Listing 6.1:** Installation of OpenVPN client

```
1 sudo su
2 apt install openvpn
3 mkdir -p /etc/openvpn/client
4 chown root:root /etc/openvpn/client
5 chmod 700 /etc/openvpn/client
6 mv whatever.ovpn /etc/openvpn/client/
```

After installation, the following script was made, and will cause the system to wait for network connectivity, and connect to the VPN server once a connection is established.



**Listing 6.2:** Startup script on VPN VM

```
1 # The following code was appended to /rw/config/rc.local
2 while ! ping -c 1 -W 1 1.1.1.1; do
3     sleep 1
4 done
5 sudo openvpn /home/user/Documents/target-[no].ovpn
```

## 6.2 PiVPN

PiVPN [60] is an open source project designed to quickly setup and harden OpenVPN servers. It was intended for installation on a Raspberry Pi, but can be installed on any server running Debian. The following commands will install PiVPN and add a client certificate and configuration, which can then be transferred to the Qubes OS machine to set up the VPN client.

**Listing 6.3:** Installation of PiVPN

```
1 sudo su
2 curl -L https://install.pivpn.io | bash
3 # Go through the installation wizard
4 pivpn add # creating a new certificate
```

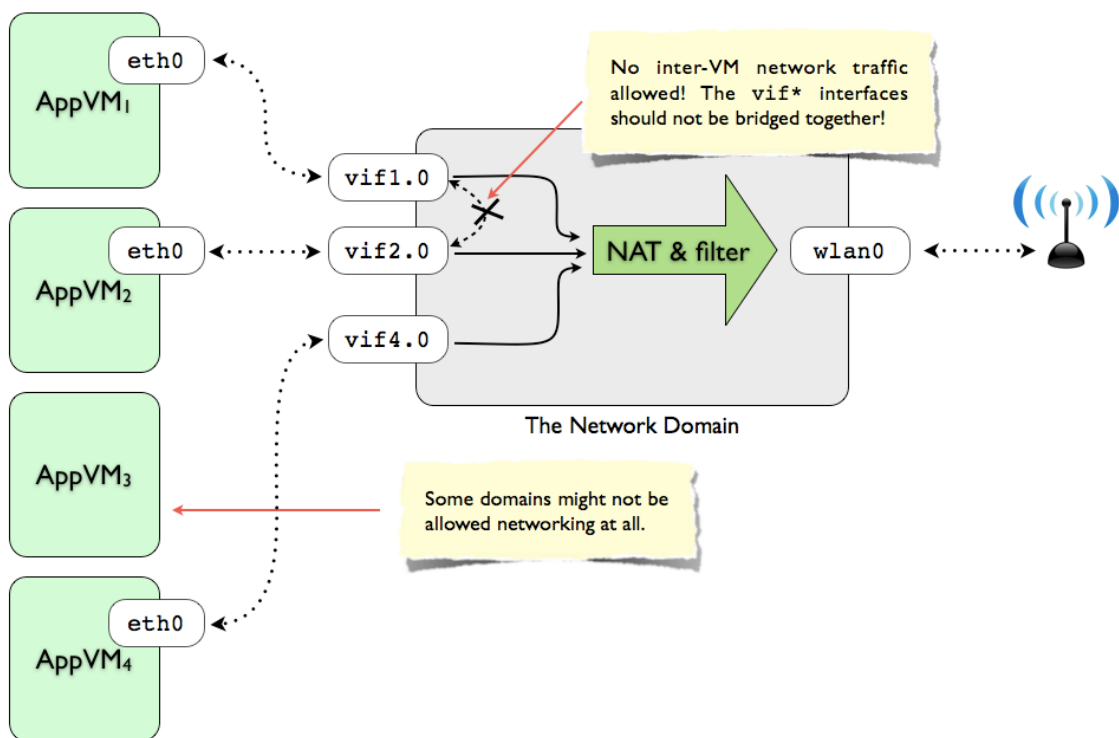
Other applications like Wireguard or Cisco AnyConnect would also be able to deliver the functionality required in this prototype, and OpenVPN/PiVPN were only used as a convenience to demonstrate VPN connectivity as a proof of concept.

## 6.3 Network architecture in Qubes OS

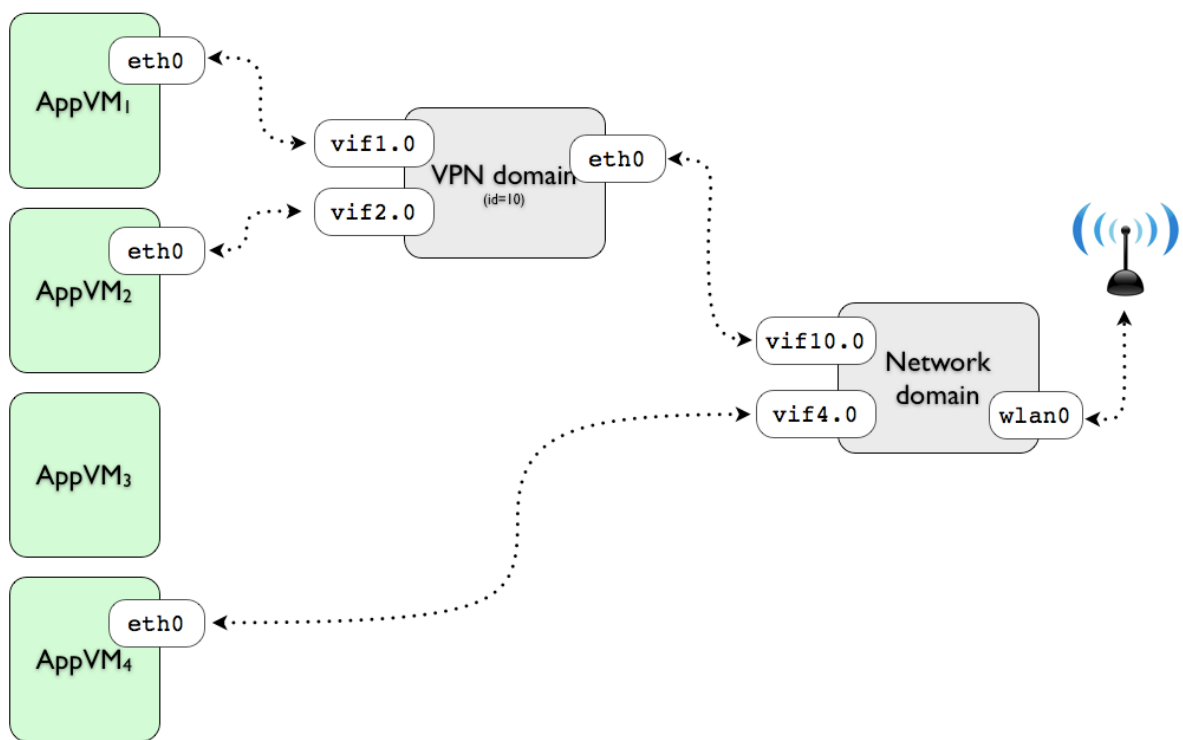
As mentioned in Chapter 5, in Qubes OS all applications run within a VM. This is also the case for both VPN client software, and the network hardware management subsystem itself. In Qubes OS, the interface between the system itself and the network hardware is facilitated through a network domain. This domain is responsible for communicating with network adapters on the host machine, and facilitates virtual networks that other qubes on the machine can utilize as their network gateway. Figure 6.1 illustrates this interaction.

In Qubes OS, the network domain is the domain that provides network access to other qubes, which in turn act as network clients. However, Qubes OS allows a qube to act as both a network gateway to other qubes, while at the same time being a client of another network domain. This is referred to as a ProxyVM [62]. A ProxyVM allows VPN software to be installed, and can be used to encapsulate network traffic before it is sent to the upstream network gateway. ProxyVMs appear to the network qube as any other qube, while appearing to its dependent qubes as if it was the standard network gateway VM. This is illustrated in Figure 6.2.

Due to the network architecture being structured in such a way in Qubes OS, VPN domains are protocol-agnostic. As long as a Linux client exists for the VPN protocol that is desired, and this client establishes network routes using the standard Linux networking stack, it can be used as a VPN client and network provider in Qubes OS. Linux clients exist for most major VPN protocols, such as OpenVPN, Wireguard, Cisco AnyConnect, etc.



**Figure6.1:** Network architecture within Qubes OS [61]



**Figure6.2:** The architecture of a VPN domain [61]

### 6.4 Network design of the multipurpose platform

Figure 6.3 illustrates the network connections within Qubes OS as well as to and from the external network. As seen in the figure, four VMs accessed by the user will have a connection to a network. Two of these will have access to the Internet with certain restrictions, while the other two will only have access to a VPN connection to servers on company premises holding files each type of VM will need. Figure 6.3 and Figure 6.4 each include a red and orange box filled with servers. These represent the servers on the company premises holding files relevant to these domains.

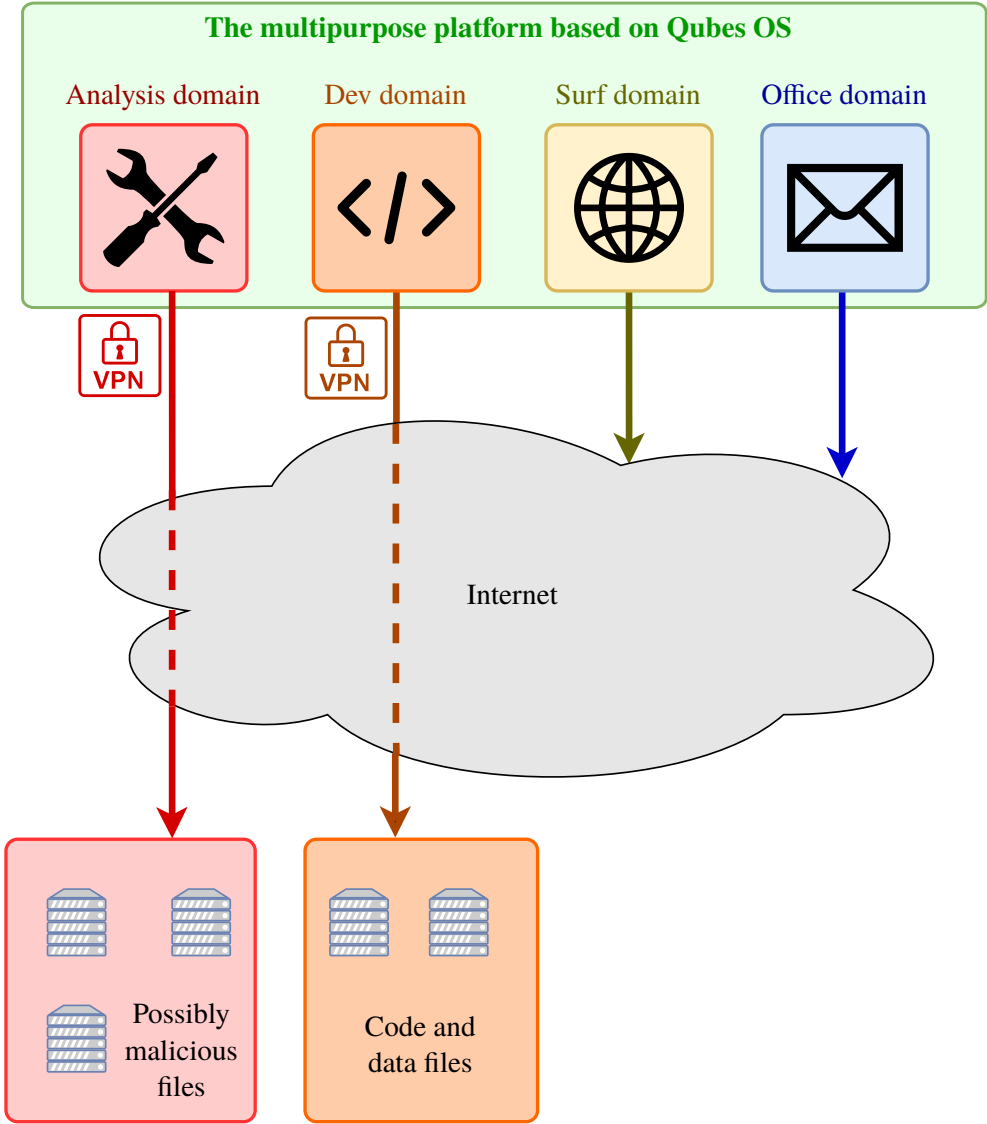
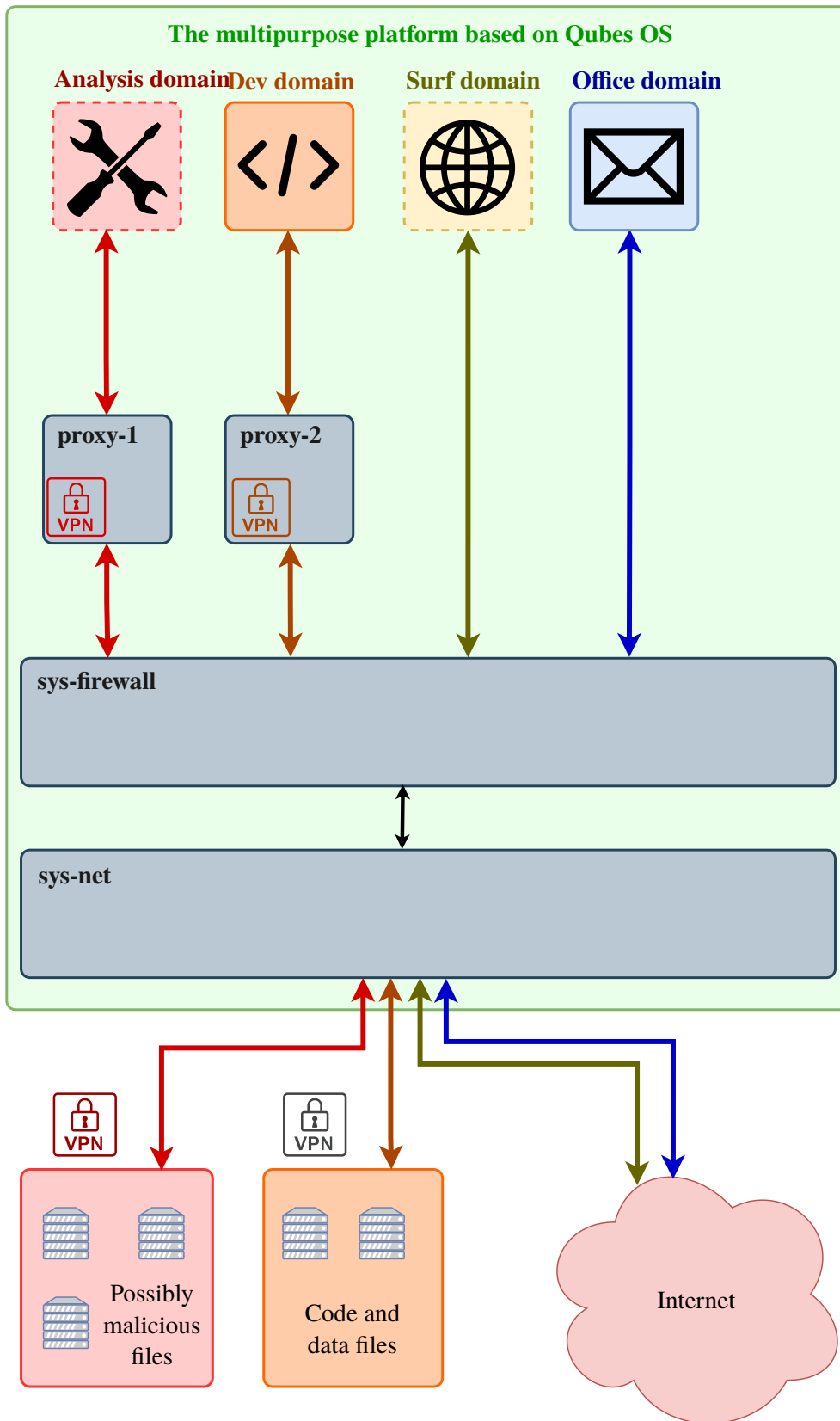


Figure6.3: Networking connections diagram

### 6.4.1 Virtual private networks

Figure 6.4 shows how the network is routed through the laptop. A VPN client is located within its own VM. This software acts as a gateway through which network activity is encrypted and rerouted to a specified location. By doing it this way, the credentials used in the VPN tunnel are separated from both the malware VMs, the testing VMs, and the network VM. Additionally, none of the VMs connected to the VPN need to have their own VPN client software installed or configured, as the VMs are presented with a standard virtual network interface, whose traffic is routed to and processed by the VPN VM.



**Figure6.4:** Proxy diagram

## Chapter 7

# Security Information and Event Management

A Security Incident and Event Manager (SIEM) is a type of system that consolidates security logs and events from different sources in an enterprise network, and stores, normalizes and correlates them in order to identify malicious activity in real time [9]. All logs and other customized piece of information is automatically synchronized to the local configured SIEM, which will hold and analyze the data. Different SIEM software provides different analysis on the data, and in some cases the administrators can even configure their own analytics. This technology and software allows IT administrators to easily get an insight into the network infrastructure and health, as well as providing easy threat modelling of the infrastructure, and incident response and event correlation when trying to figure out an incident.

The most important job for a SIEM is to be able to extract the data that is needed by the enterprise. The second job, is to be able to support data analysis (e.g. searches and machine learning). The SIEM collects data from infrastructure devices, such as laptops, workstations (desktops), router, firewalls, applications, etc. The data collected is customizable, from default files to folder paths. Collected logs are then categorized based on analytical value, and then processed by category. This is especially useful for security, as the software can report to the administrators of indicators of compromise, such as failed login attempts, impossible travel or signatures of malware or malware activity. Custom alerts can also be set up to alert the administrators of changes or events against the configured ruleset.

SIEMs are costly and may not always produce valuable data. For smaller enterprises, the entry cost can be big enough to discourage companies from investing into such a system, and for larger enterprises the price will scale with the infrastructure. Sometimes the companies must hire additional workforce to be able to read and maintain the SIEM. Splunk, which we will talk about later, also exists as a free version (Splunk Free) with limits, e.g. a limit to data collected each day and missing access to most Splunk Enterprise edition features. Commonly in SIEMs, they are very resource heavy. Minimum specifications vary, but usually require a large amount of storage, memory and processing power. In larger enterprises, a or multiple servers is usually dedicated to a SIEM. This implies a heavy up-front cost to set up, as well as a need for scalability when the enterprises infrastructure also scales up.

## 7.1 SIEM software

There are a number of SIEM software and tools on the market today. Most are closed-source and require either a monthly subscription or a big upfront cost, or maybe both. There also exists open source free alternatives, such as OSSIM and OSSEC. Most, if not all, SIEM software integrates with Linux, however, none of them integrates with Qubes OS. There is no support from the Qubes OS team nor has there been any work on supporting Qubes OS in a SIEM. The reason for this is most likely that dom0 does not have internet access, and an intrusion into dom0 from a SIEM could compromise the entire machine and is against the primary goal of Qubes OS: dom0 is the most trusted domain. We will discuss our solution to this later.

Since the nature of our task is to upload relevant logs to the SIEM, and all competitive SIEMs can support this capability on Linux and Windows, we have chosen Splunk in our proof of concept. We have been endorsed with a licence to Splunk, and it is also a requirement to test our proof of concept with the Splunk SIEM.

Other competitive SIEMs that can also support this capability are, in our research, IBM Qradar, Alient-Vault OSSIM, OSSEC and SolarWinds. However, these have not been tested due to lack of time and funding. These are not the only ones that can support this capability, but are a list of the most current dominant ones on the market with which can most likely support Qubes OS as it supports both Windows and Linux.

As our objective was to resolve integrating the laptop to a SIEM, our main goal was to upload the logs to another SIEM or storage from which the logs can be processed by other software or forwarded. This means for our task, we could chose a simple database solution which can hold our log files. The same principle for extracting data would apply to all SIEMs or log storing locations, due to dom0 not having internet access in the Qubes OS architecture.

## 7.2 Splunk

Splunk [63] is a SIEM software, capable of ingesting all forms of data in text form, from any source. All files can be integrated to Splunk with no matter to structure or need to organize it first. Splunk keeps a reference for all integrated files so users can watch logs in real-time (stream them) or go back in history to review older logs in case of incidents. Administrators can use Splunk to gain insight into their network by doing searches or leveraging machine learning and artificial intelligence (AI) to automate their tasks.

Splunk is controlled by a dashboard hosted on the Splunk instance. The dashboard is accessible through a web browser, e.g. Google Chrome. This is the main interface where the administrators can do searches, check infrastructure health and get an overview of the incoming data.

## 7.3 Splunk integration with Qubes OS

Because of the nature in Qubes OS secure architecture, dom0 does not have internet connection. To be able to upload logs and data to Splunk, the architecture requires a new domain dedicated to synchronizing with Splunk. This domain, which we will call henceforth splunkQube, will need a connection to the Splunk server and all the other logs of choice on the laptop that should be synchronized with Splunk. A connection to the Splunk server necessitates a connection to either the default `sys-firewall` qube or a customized network qube which can forward the connection between splunkQube and the Splunk server. The downsides of creating a new network qube for splunkQube is added resources usage, however, it limits



the damage potential for attacking splunkQube and attacks originating from splunkQube. On splunkQube, Splunk Forwarder must be installed, which is responsible for negotiating a connection and collecting the data on the host and sending it to either a forwarder or the Splunk server itself. The Splunk forwarder can then be configured to watch directories, within which contents will be forwarded (to either another forwarder or Splunk). Splunk must install the “App” for “Splunk Add-on for Unix and Linux” to be able to be capable to process Linux and Unix data.

It should be noted that splunkQube must either be a template VM or made a standalone VM, or that the installed software is in persistent storage on the qube, such as the `/home/user` or the `/rw` directories.

The remaining part is now to make all logs and data of choice be available to Splunk through the splunkQube, which is no easy task on its own. As splunkQube can only access the logs on its own domain, splunkQube relies on a script in dom0 to send all logs to splunkQube, as well as dom0 logs. This script should be run as a cron job or as a service at a custom schedule. The script’s purpose is to copy all relevant logs from the domains to the splunkQube which can communicate with the Splunk SIEM, and thus synchronize all log files and relevant data from the laptop to Splunk. Splunk Forwarder will be responsible for handing off the data to Splunk.

The script, named `splunkforwarder.py`, will reside anywhere on the dom0. It has a helper file, a configuration file, `splunkforwarder.conf.json`, in which user can change what location logs should upload, on any qube. Qubes can also be blacklisted if they should be ignored by the script. Lastly, the location on splunkQube the logs should be dumped. This folder should correlate with what folders the forwarder on the laptop forwards to Splunk. Inside this folder on the splunkQube, all logs for the included qubes will be dropped, with pathing to qube name and path to log files location to easily trace what qube the piece of data came from.

## 7.4 Script run-down

The full proof of concept code can be found in Appendix A.1. The script prerequisite is that a qube called `splunk` is created (this can be changed) and the helper file is also in the same folder, and they both need to be on dom0.

First all the active qubes are queried from dom0, then the blacklisted qubes from the configuration file are removed. Then each qube is queried for the files and paths in the log paths found in the configuration file, as we can see in Listing 7.1 on line 5. As this is run on dom0, we must use the sub-command `qvm-run` with the `--pass-io` flag to run the command on the qube. The output is automatically output to the terminal, and the answer is put into an array, as seen in line 8. The result is then sent to `getFileandCopy` which is responsible for creating the folders on the splunkQube in the indicated locations in the configuration file.

**Listing 7.1:** `qubeLogHandler` function in `splunkforwarder.py`

```
1 def qubeLogHandler(qube):
2     qubeList = []
3     # Get all files that need to be sent to the getFileandCopy function
4     for confPath in data["log_locations"]:
5         command = subprocess.run('/usr/bin/qvm-run -u root -p '+ qube +' "sudo_
↳find '+ confPath +' -type f"', shell=True, universal_newlines=True,
↳stdout=subprocess.PIPE)
6         files_with_path = command.stdout.splitlines()
7
```

(continues on next page)

(continued from previous page)

```
8     qubeObject = {'files': files_with_path, 'path': confPath}
9     qubeList.append(qubeObject)
10    # Recursively push all files to their respective folders and files
11    for qubeObject in qubeList:
12        for file in qubeObject['files']:
13            getFileandCopy(qubeObject['path'], file, qube)
```

Listing 7.2 shows the `getFileandCopy` function. As it is also run from `dom0`, all processes on the other qubes must run through the `qvm-run` command. First we create the folders that the files should be copied into, as seen on lines 4 and 5. Lines 8 and 9 are responsible for running the `cat` commands on the other VMs and pipe the output to the `splunkQube`. As seen on line 9, the path to the ported file resides on the `splunkQube` under the path specified by configuration file and under each qube name.

**Listing 7.2:** `getFileandCopy` function in `splunkforwarder.py`

```
1 def getFileandCopy(path, filename, vmname):
2     # Create the folder for the current file
3     pathWithoutFile = filename.rsplit("/", 1)[0]
4     subprocess.run('qvm-run -p splunk \
5         "mkdir -p '+ data["splunk_qube_log_location"] + '/imported_logs/'+
6         ↪vmname + '/' + pathWithoutFile + '"', shell=True)
7
8     # Cat and push all files to the splunk vm
9     subprocess.run('qvm-run -u root -p '+ vmname + ' "sudo cat '+ filename + '
10    ↪" \
11    | qvm-run -p splunk "cat > '+ data["splunk_qube_log_location"] + '/imported_
12    ↪logs/' + vmname + filename + '"', shell=True)
```

Listing 7.3 shows what the configuration file can look like. The keys are a must, but the values of `qubes` and `blacklisted_qubes` can be empty. Log locations should also include at least one location, or else there is no use for this script. `splunk_qube_log_location` is the location on `splunkQube` in which the user wants to store the logs from all the laptops qubes. This is the folder that must be set to forward to the Splunk server by the Splunk Forwarder.

**Listing 7.3:** `splunkforwarder.conf.json` example

```
{
  "qubes": [
    "dom0",
    "splunk",
    "sys-firewall",
    "sys-net",
    "sys-usb",
    "sys-whonix"
  ],
  "log_locations": [
    "/var/log",
    "/etc/cron.d"
  ],
}
```

(continues on next page)

```

"blacklisted_qubes": [
    "vault",
    "windows-mgmt"
],
"splunk_qube_log_location": "/home/user"
}

```

As we can see, all commands are run from the dom0 qube, therefore we need a special case for handling the dom0 qube. The premise is the same as handling all the other qubes, however, we can not run the `qvm-run` command on the dom0 qube. So excluding the `qvm-run` and running the commands as first hand instead, like we do in Listing 7.4 on line 12, we also include dom0 in our script. It should be noted that a limited amount of logs from the other qubes are stored in `/var/log/qubes` on dom0. However, in our testing, the logs in this location were lacking. Many files were empty or not up to date with recent information, or just not present at all. Especially the Windows qubes did not store any logs in this location, most likely due to lack of support for Windows in Qubes OSs and out of date Qubes Windows Tools (QWT).

**Listing 7.4:** dom0Handler function in `splunkforwarder.py`

```

1 def dom0Handler(qube):
2     def qubeListHandler(listItem):
3         for file in listItem["files"]:
4             # Create folder and then copy the files
5             pathWithoutFile = file.rsplit("/", 1)[0]
6             subprocess.run('qvm-run -p splunk \
7                 "mkdir -p '+ data["splunk_qube_log_location"] + '/imported_logs/
↳'+ qube +'/' + pathWithoutFile + "'", shell=True)
8             subprocess.run('sudo cat '+ file +' | qvm-run -u root -p splunk
↳"cat > '+ data["splunk_qube_log_location"] + '/imported_logs/' + qube +' '+
↳file + "'", shell=True)
9
10    qubeList = []
11    for confPath in data["log_locations"]:
12        command = subprocess.run("sudo find "+ confPath +" -type f",
↳shell=True, universal_newlines=True,
13            stdout=subprocess.PIPE)
14        files_with_path = command.stdout.splitlines()
15        qubeObject = {'files': files_with_path, 'path': confPath}
16        qubeList.append(qubeObject)
17    for listItem in qubeList:
18        qubeListHandler(listItem)

```

As Splunk is capable of collecting all data, all piece of data in the indicated directories are sent to Splunk. A blacklist for certain files or file extensions could be added, or regular expressions to filter out certain pieces of data. This blacklist could be crosschecked with the filename before each extraction to the `splunkQube`, and skipped if it is blacklisted.

Only qubes that are up and running will be included in this proof of concept. It is possible to add the functionality for starting and stopping qubes when they are done, but because of the limited resources the laptops have, this was not included in our proof of concept. Such an alteration would risk qubes not starting due to maximum resource usage by the user, as every qube takes their share of the shared memory pool.

Therefore, the log collection would risk not collecting all logs in a day and synchronize inconsequential data to the SIEM, making it harder to see the chain of events or obfuscating alerts that would have been triggered if the correct logs were collected and thus hiding a threat. However, only collecting logs from running qubes also run the risk of hiding potential triggers in shut down qubes as well. A better solution might be to collect logs when starting a qube and when shutting it down, at the risk of increased shutdown and startup time.

It should be noted that dom0 also stores some logs of the other qubes, however, this is a limited selection and most log files are empty. We therefore opted to make the script to be able to customize the log directories on all qubes.

The proof of concept script currently only handles Linux and Unix qubes. Due to the nature of the Qubes OS architecture, copying to and from dom0 requires usage of the Linux `cat` command, as seen in code snippet below. Windows does not support this command, and a solution is to use the command `type` instead. An example of how such a command might be like this, from dom0: `qvm-run -p <windows qube name> 'type <path\to\file.txt' | awk -F ''exit' '{print $NF}' | awk 'NR > 4' | cat > <local file name>'. Following the same pattern as the script, piping the file contents to the terminal is one of the few ways to copy files to and from other qubes from dom0. Alternatively the command can pipe the output to the qvm-run command and pipe it to another qube again, e.g. the splunk qube. We will also discuss using Microsoft Azure Active Directory (Azure AD) to send logs to Splunk by joining the Windows qube to the domain in Chapter 9.`

## 7.5 Investigation summary

A SIEM is a piece of software that can give IT administrators easy insight into the security and health of their network. In our proof of concept, we chosen to work with Splunk. Most SIEMs would be able to do the same.

A connection from any of the laptops to the SIEM is doable, however, due to the nature of the Qubes OS architecture, the Splunk Forwarder can only forward the data in the qube it is installed on. It cannot be installed on dom0, because dom0 has no network interfaces. To be able to forward all logs from a laptop, a script that resides in dom0 is necessary. This script can copy all logs from the other qubes to the qube which connects to Splunk. We have created a working proof of concept for how such a system might work.

## Chapter 8

# Windows in Qubes OS

This chapter will focus on how to install Windows in Qubes OS. We will introduce different ways to install Windows, and the difference between them. Lastly we will cover some of the problems we have found a common user may encounter while using Windows on Qubes OS.

Windows can be installed on Qubes OS like any other operating system. However, due to Windows not being a Linux or Unix-based operating system, much of the core functionality in Qubes OS is not supported out of the box on Windows, and even less so on recent versions of Windows. For example, the Qubes OS inter-qube clipboard will not work out of the box with the Windows qube. For greater integration with Qubes OS, we need Qubes Windows Tools (QWT). Without Qubes Windows Tools (QWT), the qube will act like a normal VM with a full desktop experience.

### 8.1 Installing Windows in Qubes

There are several ways to install Windows in Qubes OS. We are going to focus on the two methods we had the least difficulty with. The first one is using a guide by the official Qubes OS team, while the other is using third party tools and scripts to easily install Windows along with Qubes OS-specific tooling.

#### 8.1.1 Manual installation guide

Installing Windows manually is in itself not a complicated process. It involves creating a standalone VM or template VM, then booting it with a current Windows ISO [64]. The ISO file can be downloaded on an existing qube, or it can be stored on a USB drive. The installer will guide you through the rest of the setup. These are the commands we used to install Windows 10.

**Listing 8.1:** Install Windows 10

```
qvm-create --class StandaloneVM --label red --property virt_mode=hvm new-vm
qvm-prefs new-vm memory 4096
qvm-prefs new-vm maxmem 4096
qvm-volume extend new-vm:root 25g
qvm-start new-vm --cdrom=someVM:/home/user/isofile.iso
```

## 8.1.2 Elliot Killick's `qvm-create-windows-qube`

`qvm-create-windows-qube` [65] is a tool created by Elliot Killick and Brendan Hoar for conveniently creating Windows qubes in Qubes OS. The tool will also install QWT and configure the Windows qube to improve its compatibility with Qubes OS. Installation with this tool can be done in about five minutes. The tool is open source and available on GitHub [65]. It will allow the user to easily use a script on the `dom0` qube to create Windows qubes with different configurations, such as template VMs and disposable VMs, using different Windows versions.

It works by creating a qube, `windows-mgmt`, from which it downloads and stores Windows ISO files and helper files. All qubes are created from the script residing in `dom0`, which uses the downloaded ISOs and configuration from the `windows-mgmt` qube as input values for qube creation. The script needs to reside on `dom0`, as `dom0` and `admin` VMs are the only qubes with the access and tools needed to create other qubes. It is possible to give VMs the ability to create additional VMs. This is done through a remote procedure call (RPC), where a task is started in a VM by the command of another. The policy which dictates what RPCs are allowed in Qubes OS must be altered if this is to be achieved. This is demonstrated in Chapter 10.

The tool improves privacy for the user by disabling certain Windows functions. The `post/spyless.bat` script [66] is run automatically when first installing the Windows qube. It will make changes to the Windows system registry to reduce the amount of telemetry Windows tries to send back to Microsoft. When using this tool to create a multipurpose platform for security analysts, this function may be undesirable, as it might interfere with how the malware functions. If `qvm-create-windows-qube` is to be used, we would recommend turning this functionality off.

`qvm-create-windows-qube` has a script for downloading Windows ISO files off the Internet [67]. However, this script appears to be outdated, and no longer functions correctly. First of all, certain ISOs cannot be downloaded, due to Microsoft having disallowed downloads for those images outside of using the Windows Media Creation Tool. Only some Windows versions may still be downloaded from the official Microsoft website, such as Windows 10 Enterprise evaluation edition. Secondly, the script checks downloaded files against a hard coded hash. However, due to the script not being actively developed, some of these hashes are outdated due to newer versions of Windows ISOs having been released since the script was written, replacing the original ISOs against which the hashes were calculated. The script will give the user a warning, but allow the user to download anyway.

Lastly, the script uses key pinning to check whether the website being accessed is the legitimate site the script expects to access, so the script is not vulnerable to Man in the Middle attacks. However, due to the script not being kept up to date, the hard-coded public keys in the script no longer match the certificates presented by Microsoft's servers, preventing the user from downloading the suggested ISOs from the provided links. There is no way to turn off key pinning beyond modifying the script to disable that check.

We would recommend either changing the script and removing deprecated security checks for the script to function, or updating the values to be consistent with the current values provided by Microsoft. Since the script can no longer support download for certain ISOs from Microsoft, we would recommend creating a new tool for downloading the ISO images from a secure local network source and using up to date security checks to make sure the ISO was not compromised when downloading, e.g. via verifying file hashes. No matter the solution to this, the images should be stored in the same folder as `download-windows.sh`, so the rest of the tool can find it.

## 8.2 Qubes Windows Tools

Qubes Windows Tools (QWT) [68] is a set of software tools that can be installed on a Windows instance for greater integration with Qubes OS. With QWT, the Windows qube will support Qubes OS secure inter-qube transfer, for example secure clipboard and file exchange, direct application access from the Qubes OS application menu, and network support. QWT only supports Windows 7 and 10, but the features supported in each Windows version varies. For example, seamless mode (explained below) is not available on Windows 10, but it is possible to enable this in Windows 7. See Table 8.1 listing QWT feature availability for Windows 10 and Windows 7.

**Table8.1:** Qubes Windows Tools compability table [68]

Feature	Windows 7	Windows 10
Seamless mode	Yes	No
Qubes network setup	Yes	Yes
Private volume setup (profiles)	Yes	Yes
File send/receive (Qubes OS secure file exchange)	Yes	Yes
Clipboard copy paste (Qubes OS secure clipboard)	Yes	Yes
Application shortcuts	Yes	Yes
Copy/edit in disposable VM	Yes	Yes
Block (storage) devices	Yes	Yes
USB devices	Yes	Yes
Audio	No	No

As seen in Table 8.1, there are minor variations in feature availability for different Windows versions. The greatest difference is the lack of seamless mode in Windows 10. Seamless mode is the Qubes OS definition for rendering only a single program in the Qubes OS workspace, as opposed to the entire desktop area of the VM, in such a way that the user can easily use programs from different operating systems side by side.

QWT is, as of writing, unmaintained, and 32-bit Windows is not supported at all.

## 8.3 Speakers and microphone (audio)

Due to lack of audio support in QWT, there is no audio passthrough between Qubes OS and the Windows qubes. Qubes OS is not able to register audio or input audio to the Windows qube through the speakers or microphone. However, PCI devices can be assigned to individual qubes. The audio hardware in the laptops are presented as its own PCI device to the system, and by manually assigning this PCI device to a Windows qube, we can allow that qube access to both the speakers and microphone. As PCI devices can only be assigned to a single VM at a time, when this is done, no other qubes can use the speakers at the same time. The Windows qube is thus the only qube that has audio access in such a configuration.

Normally, the audio PCI device is not assigned to a specific qube, and in such a configuration, Qubes OS takes responsibility for the audio device, and shares access to it by providing virtual audio devices to the qubes that need audio capability.

On some machines, the audio PCI device may be presented as part of the system chipset itself, and cannot be individually assigned to a specific qube without assigning the entire chipset PCI device to the qube. Assigning the chipset PCI device to a qube will render the system non-functional, and as such, those machines will not be able to provide audio inputs and outputs to the Windows qubes at all. Allocating or

deallocating a PCI device to a qube requires restarting that qube, as such changes can only be done when the VM is powered off.

## 8.4 Web camera

QWT does not integrate well with the Qubes OS device manager, leading to problems when trying to give access to the laptop's web camera to the Windows qube. This stems from the same root issue as the audio problems discussed above. The web camera is internally connected to the system via a USB controller, which is in turn part of the system's chipset. Giving the Windows qube access to the PCI device holding the web camera's USB controller results in the qube being unable to start.

## 8.5 Screen sharing

Screen sharing is a commonly used feature in online team meetings, wherein a person shares their screen to the other participants in the meeting, and optionally allows them to take control of their machine to assist in doing a specific task. Due to all applications running inside their own VMs in Qubes OS, sharing the screen with an application running inside a qube will only show the contents of the desktop of that particular qube. The main graphical user interface (GUI) of Qubes OS is managed by dom0, and giving access to that domain from other qubes for the purpose of screen sharing is deemed by the Qubes OS developers as “too risky to even consider” [69].

Screen sharing is still possible by using an external capture card [70]. A capture card can see the same input as the screen, and can pipe the screen via USB to another computer or the Qubes OS computer itself. Using this “bypass” screen sharing is still possible in Qubes OS, but require proprietary hardware, such as a capture card.

## 8.6 Investigation summary

Using the manual method from the Qubes OS team takes longer than using Killick and Hoar's automated tool, but this method enables better customization for each laptop. However, we do not see the benefit of this as the final product will be installed on many instances of the same laptop.

An automated tool is the easiest approach for a project like this, as well as for limiting the amount of work the administrator will need to perform on each laptop before giving it to a user. However, in terms of security, an automated solution such as `qvm-create-windows-qube` should go through code review and tested before being put to use. The tool should also be altered to not rely on downloading files from the Internet, such as the ISO files and the repository on GitHub. These files should be hosted on the network the administrator is using for first time set up with the laptop.

We also experienced some issues between the different installation methods. AD would not work for Windows qubes installed by Elliot Killick's script, for instance. We will detail this in Chapter 9, but this should be considered before using either method.



## Chapter 9

# Identity and Access Management

IAM keeps track of the digital identities of employees or customers of a business. It is a way to ensure that the users have correct permissions and can be used as a single sign-on (SSO) to the organization's assets. Administrators can manage the access rules and policies of the users, for example multi-factor authentication (MFA) for login. The system can be on-premises, cloud-based or both [71].

One of the research questions specifies that the platform should be able to connect to an IAM. KDA already uses Microsoft Azure Active Directory (Azure AD) in their organization thus it was a natural choice to for us use this in our proof of concept as well. As we will further discuss in the following subsection, Azure AD is a popular solution among some of the biggest organizations in the world, and it is easily integrable with other solutions that an organization might already use. We think that using Azure AD in our proof of concept will reach a wide userbase.

Many IAMs services, provides the ability to join a laptop device to the organization's domain. This means that they can log in to the laptop with their work account credentials and directly access the organization's resources [72]. At this point, Qubes OS does not have any good way of joining the whole computer to be managed by the organization, without giving Internet access to dom0. This would defeat one of the core security principles in the Qubes OS architecture. Although we cannot join the whole computer to an IAM service in the current Qubes OS version, it is possible to join one or more VMs to access company applications and use SSO.

In the following section, we briefly describe Azure AD, the open source IAM FreeIPA [73], as well as a few popular IAM systems, and their compatibility with our platform.

### 9.1 Microsoft Azure Active Directory (Azure AD)

Microsoft Azure Active Directory (Azure AD) [74] is a cloud based IAM from Microsoft with pricing ranging from free to around 80 NOK per user per month [75]. It lets employees log in with MFA and use SSO to access the organization's own infrastructure as a service (IaaS), platform as a service (PaaS) including their cloud applications, as well as their software as a service (SaaS) such as for example Microsoft 365. Users can also gain remote access to the organization's on-premises applications using Azure AD Application Proxy [76]. There are also ways for the organization to integrate already existing applications or solutions with Azure AD [77].

The organization can manage and govern each user identity and their permissions throughout the employee's different roles in the organization (identity- and access lifecycle) [78], for example changing the

employees access rights when they are assigned to a new project, making it easier to follow a principle of least privilege.

All Azure AD licences provide the option to log user activity, however the more expensive licenses provide more features. The logs can be viewed in the form of comprehensive security and activity reports. Security reports provide a list of possible compromised users, and possibly illegitimate sign-ins. The premium Azure AD licenses also provide different levels of detail about the underlying risks, and how to respond to them. Activity reports provide information about what the users do, for example which users are granted different rights, what applications they are using or when they last reset their passwords [79]. The premium licenses also provide information about the user sign-ins. The activity logs can be routed to various endpoints, for example to a SIEM such as Splunk [80].

Azure AD is a popular solution for many organizations that use Microsoft 365 or Microsoft's cloud platform Azure because these services already require the users to sign in with Azure AD [81]. According to Microsoft's own websites, 95% of the current fortune 500 companies use Microsoft Azure [82], and over 1.2 billion identities are managed by Azure AD [74].

One issue we ran into with Splunk, our SIEM, is that the Windows 10 qube does not support the proper command for copying log files to dom0. Although there are workarounds as described in Chapter 8, this issue can also be solved by joining the Windows 10 qubes to Azure AD and route the logs to Splunk. It should be noted that by using this method, only the Windows 10 qubes that are joined to Azure AD will be logged. This could be a solution, but we recommend using the Splunk workarounds to secure that all the appropriate logs are acquired.

Azure AD supports anomaly detection of impossible travel [83]. This means that if the user is detected to change location impossibly fast, this is an indication of malicious activity. It is possible to set various actions that will be done when this anomaly is detected. In our proof of concept, this will only be possible to apply to the VMs that are joined to Azure AD, since it is not possible to join the whole computer.

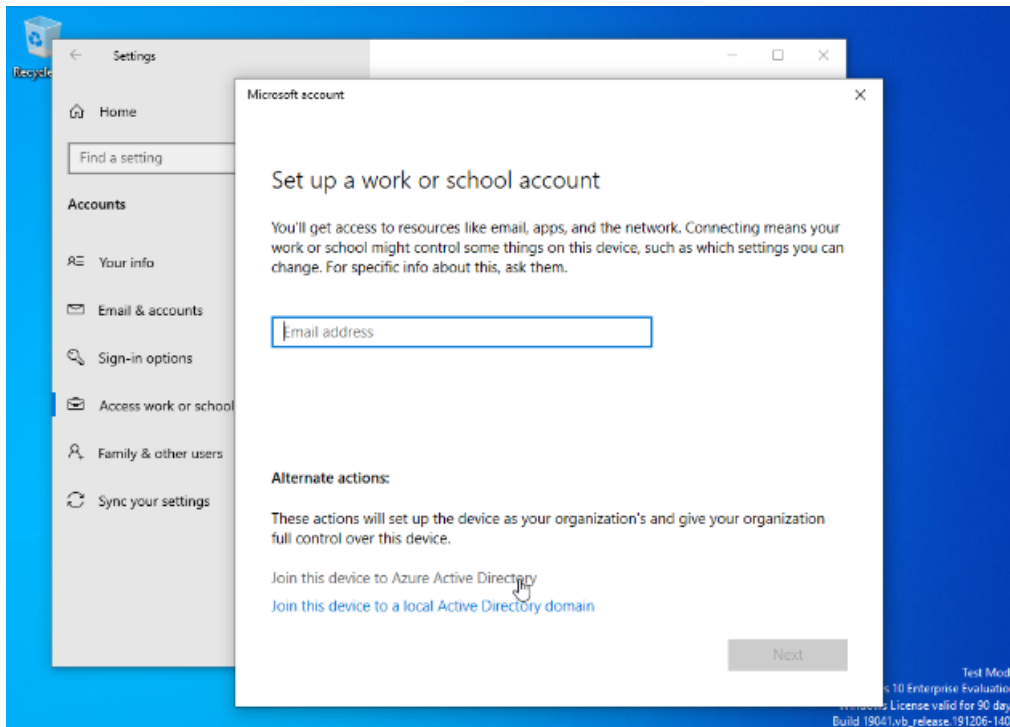
## 9.2 Other solutions

Other popular options are Google Cloud IAM and International Business Machines (IBM) IAM solutions closely resemble Azure AD and AWS Identity is Amazon's cloud based IAM. With Google and IBM, the organization can manage user devices by employees enrolling their devices in the solution and logging in using their work account. The user can use Windows, iOS, macOS and Android to join device to IBM, [84]. Google lets users join with Android, iOS, Chrome OS, Mac, Windows and Linux [85]. AWS Identity lets the employees use SSO to gain access to their Amazon cloud, but not to join their laptop, although it can be integrated with an organizations on-premises AD [86], with Azure AD, or with another identity provider if the organization wish to combine several solutions [87].

Another solution to consider is FreeIPA [73], an open source IAMs based on open source technologies such as 389 Directory Server, MIT Kerberos, NTP, DNS, Dogtag (Certificate System), combining them in a into a userfriendly solution. It enables the user to join Linux machines to the organization's domain, and offers SSO and is compatible with AD, allowing for SSO between Linux and Windows machines [88]. In the context of Qubes OS, this project is interesting. As mentioned the only current way to use Qubes OS with an IAM is to join one VM. Using an open source IAM could open possibilities for the development of a solution with a better integration with the Qubes OS system as a whole, possibly by allowing for join with seamless application VMs.

## 9.3 Implementation

In the following section, we will provide a proof of concept of joining a Windows VM to Azure AD. Because there is no good way to join the whole Qubes OS machine to an IAM at this moment, the best option for the solution is to join the VMs individually. The VM needs to have Windows 10 installed (excluding Windows 10 Home) to be able to join Azure AD [72]. We installed the Windows 10 Enterprise Evaluation edition from Microsoft [89] following the commands in the Qubes OS documentation [64], the way described in Chapter 8.

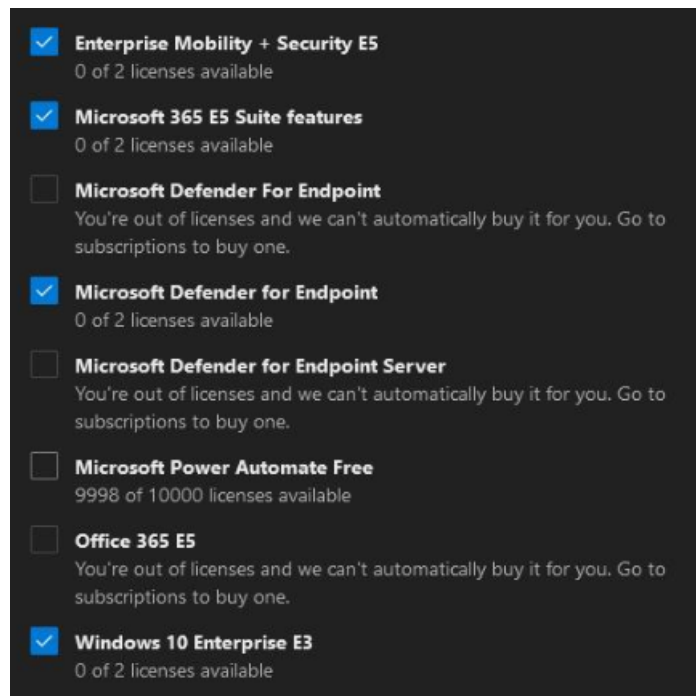


**Figure 9.1:** Joining to Azure AD

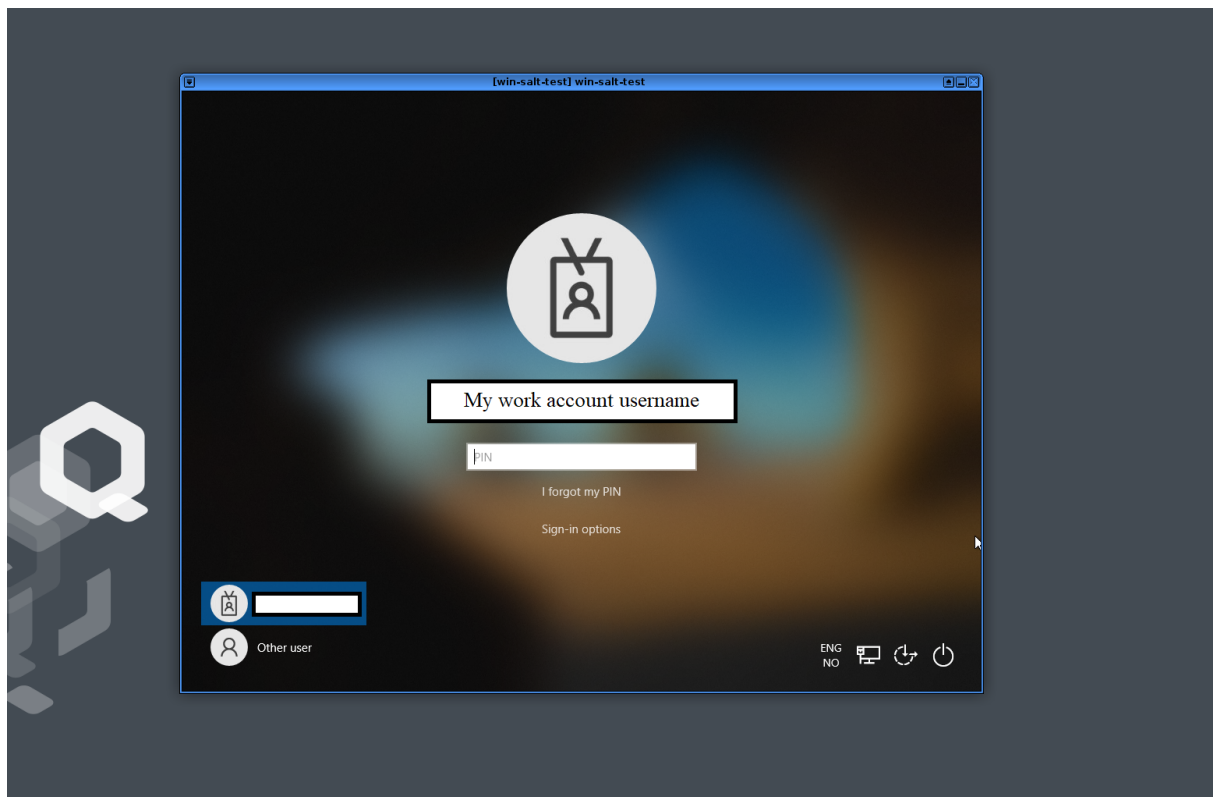
Provided that the appropriate access are granted to the user, they can now log in to Azure AD with the Windows 10 VM by logging in to the VM and then joining via “Windows Settings > Accounts > Access work or school > Connect > Join this device to Azure Active Directory”, then enter their work email address and password. We joined the client’s Azure AD with the access shown in Figure 9.2.

After joining the device the user need to log off and log back into their work account using their work credentials again. The user will now be able to use this VM using SSO to authenticate to their organization’s applications [90].

This VM is configured using SaltStack which we will discuss in chapter Chapter 10.



**Figure9.2:** List of licenses we were granted in the working proof of concept



**Figure9.3:** Joined Windows 10 VM

### 9.3.1 Problems

The first problem we encountered was that we got an error message when trying to connect to our Azure AD account in the “access work or school” settings. After KDA changed some settings with some trial and error, following the Microsoft’s troubleshooting guide [91], we managed to log in using the settings in image Figure 9.2

The next problem we encountered was when switching users in Windows and trying to log in with the Azure AD account. This only resulted in an endless spinning wheel after the sign-in process. There is a known problem [92] that gives a three hour startup delay if the workgroup name is the same as the on-premises domain NetBIOS name, but since we tried both switching the workgroup name and waiting well over three hours, we can rule out that this is our problem.

The Windows 10 VM can be automatically installed using the Elliot Killick’s `qvm-create-windows-qube`, as described in Section 8.1.2. After successfully joining a manually installed Windows 10 VM to Azure AD, we concluded that the problem is caused by using the script installation. We tried installations using the script without any of the optional alternatives ‘optimize’, ‘spyles’, ‘whonix’ and ‘packages’, using the command `./qvm-create-windows-qube.sh -n sys-firewall -i win10x64-ltsc-eval.iso -a win10x64-ltsc-eval.xml myVM`. We have also tried using the same ISO file that we used in the manual installation, and removing the QWT installation from the script, but neither had any effect on the problem with the script.

## 9.4 Multifactor authentication

In order to improve the security of the machine, it is desirable to implement some form of multi-factor authentication (MFA). Multiple MFA solutions exist, but due to dom0 being locked down from accessing the Internet, not all solutions are feasible. Qubes OS has built-in support for YubiKey [93], so we decided to try to implement YubiKey as second factor for sign-in to the machine. The group did not have a YubiKey device on hand, but we instead chose to attempt to use an OnlyKey [94], which is similar to a YubiKey.

While we were able to set up a challenge-response token on the OnlyKey device, and properly configure Qubes OS following the official documentation [93], no challenges were ever sent to the OnlyKey device. The sign-in screen on Qubes OS also did not indicate that a device was even connected, indicating that an OnlyKey does not work to provide MFA in Qubes OS. It is possible that a YubiKey would work instead, but seeing as the group did not have access to a YubiKey, we chose not to pursue that path any further.

## 9.5 Investigation summary

As long as the Windows 10 VM is installed manually, or with our central management Saltstack solution discussed in Chapter 8 the login process to Azure AD is the same as on a regular Windows 10 machine. In our proof of concept we use Azure AD with a Windows 10 standalone VM, because this would be a relevant use case for many organizations, including KDA. Linux seem to be the OS prioritized by the Qubes OS developers, and Windows 10 currently has some performance issues in Qubes OS Chapter 8 considering web camera, speakers and microphone. In terms of compatibility with Qubes OS, and considering the user experience, we would recommend using Linux instead of Windows 10 at the time of writing this. Joining a Linux VM to a IAM that supports Linux would be possible the same way we have joined a Windows 10 VM to Azure AD in our proof of concept by using a persistent standalone VM. FreeIPA being open source also opens up the possibility of further work developing an integration with Qubes OS, possibly allowing for a more seamless login shared between several qubes in a domain.

## Chapter 10

# Configuration Management

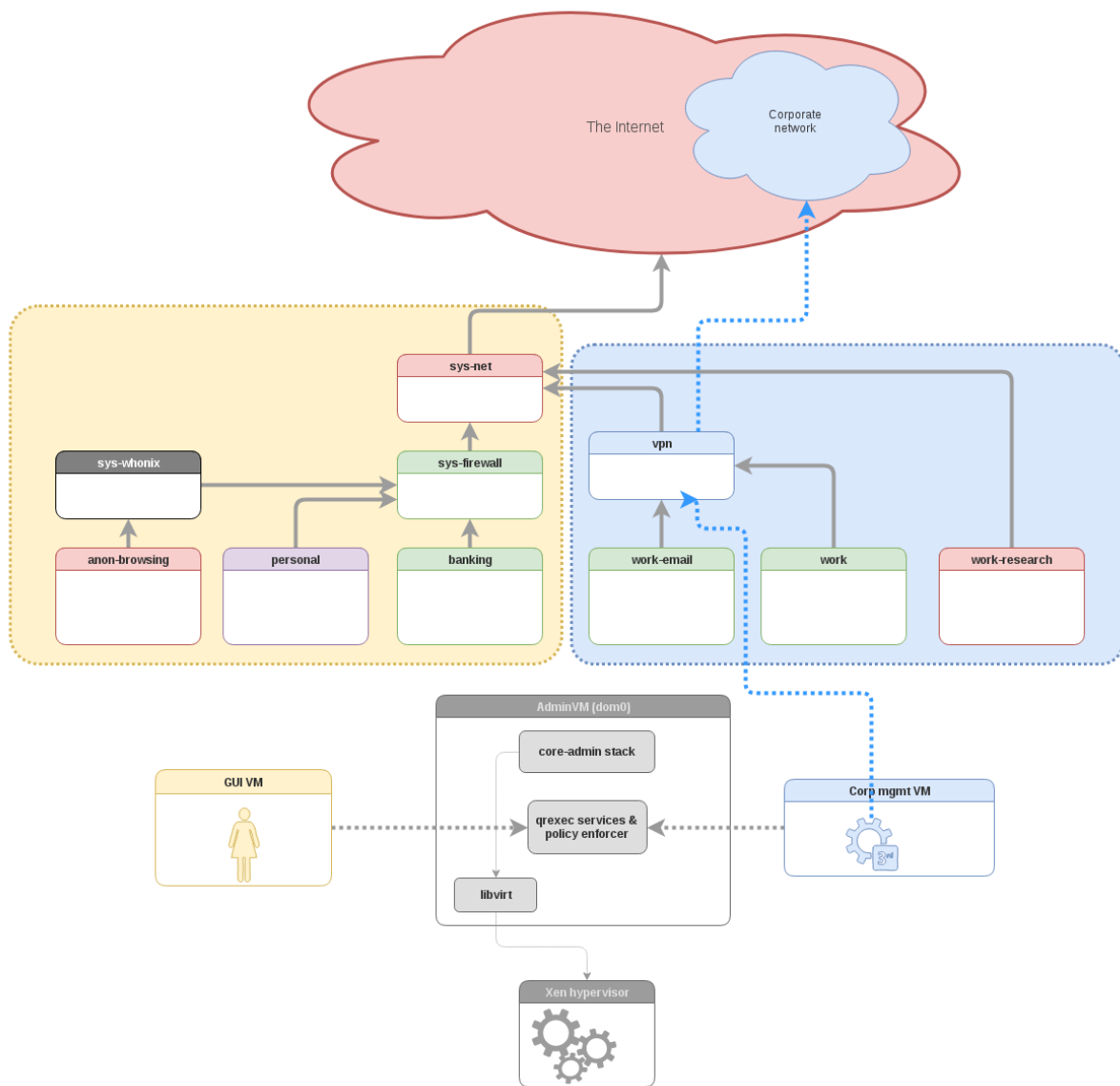
In this chapter we go into details of how the laptops we have been provided can have their configuration centrally managed. By this we mean to allow IT staff to write code that can be stored in a server on their internal network and from whence the laptops can download and apply their configuration. We will also discuss some implementation alternatives and some of the difficulties one might face when attempting to deploy such a setup for a large number of employees.

In 2017 Qubes OS introduced the “Admin API” along with future plans and a tutorial on how to create an admin VM. These are qubes that can be used to create and configure other qubes without having to use dom0. This is part of the goal of sealing off dom0 [69]. Admin VMs can for the most part replace later mentions of dom0 as configuring the laptop is a reason for admin VMs to exist. Figure 10.1 illustrates how they planned for Qubes OS to evolve.

### 10.1 Tools

There are many tools used to perform configuration management [95]. One of these, Salt [11] has been integrated with Qubes OS [96]. This integration is created to allow dom0 to manage the other qubes on the same physical machine. Writing Salt code for Qubes OS is similar to writing it as one otherwise would, but because of the integration there are additional functionality regarding the creation and configuration specific to Qubes OS. The fact that the developers have focused on Salt being a local configuration tool has been a hinderence, but a possible solution is detailed in the following section.

Another tool, Ansible [13], is meant to be a more lightweight alternative to other configuration management tools, and there is one project on Github that is of interest regarding Ansible. The project “ansible-qubes” [97] uses a custom program `bombshell-client` in order to facilitate Ansible commands. The project even has a tutorial on how to enable other laptops to remotely manage qubes on the platform. Because Salt is integrated with Qubes OS, and because we do not have the capacity nor the knowledge to proofread the entire project in time, we decided to describe a solution using Salt. In the case where Ansible is preferred over Salt, or if the program is of interest then forking the repository might be a good start for an internal project.



**Figure10.1:** Image from Qubes OS press release

## 10.2 Implementation and centrally managing laptops running Qubes OS

Figure 10.2 illustrates how the configuration files are accessed by the platform. The straight filled lines represent connectivity and ability to transfer in the direction of the arrow. The dotted curved lines represent the propagation of new configuration from dom0. Regular updates to the platform configuration will until further updates to Qubes OS be easiest to implement as a pull system. This could be done in regular time intervals so as not to catch employees by surprise. It would be bad if an update came as a surprise to employees who might have their self made configurations necessary for some task removed.

We have created a script to download configurations from a central repository and transfer them to dom0. Tools such as Anacron [98] can be used to execute commands or start these scripts in dom0 and will start them some time while the laptop is on. It is pre-installed on dom0. The script must be should be run regularly to keep the platform updated with new configurations. Make sure that the laptop has an Internet connection, and that the script is run as administrator. To protect against tampering, a hash value could be generated when downloading the repo and compared to a pre-generated one to ensure the integrity of the files.

To reduce the impact of information leakage we recommend not storing sensitive information in the repositories storing code that the laptops will pull from. As all contents will be downloaded on employee laptops, only material they are eligible to see should be put in that repo. Users should authenticate themselves to the server which provides the Saltstack code. We recommend the use of password authentication as this can be set up quickly as described in demonstration scripts below.

### 10.2.1 Clone with HTTPS or SSH

Normally when attempting to use HTTPS based `git clone` from the command line, the user will be prompted to enter their password. Since our script is run from dom0, using the command `qvm-run --pass-io` to pass commands to the Git VM, the password prompt will appear inside the Git VM. This prompt will thereby not be passed back to dom0 and the user is never prompted for their password. One solution to this problem is to include the password in the URL when cloning the repository (`git clone https://user:password@bitbucket.org/repo`).

### 10.2.2 Script

In short, the script creates a new qube that downloads the files from the Git repository, copies the files to dom0 and then deletes the newly created qube. The variables in the script need to be set and customized before it is run. The `DIR_TARGET` variables specify where the files are located in Git, and the script will currently download “other files” that are located directly in the Git repo, and SaltStack files that are located in the folder called `salt`. These locations should be customized to fit the structure of the repository being used. The `DIR_PREFIX` variables specify where the files are saved in dom0. The SSH file, or alternatively username and password must also be specified, depending on the chosen authentication method. We used code made by the github user SkypLabs as a basis for our script [99].

In the following pages we will explain how our script works. The whole script can be found in Appendix A.3



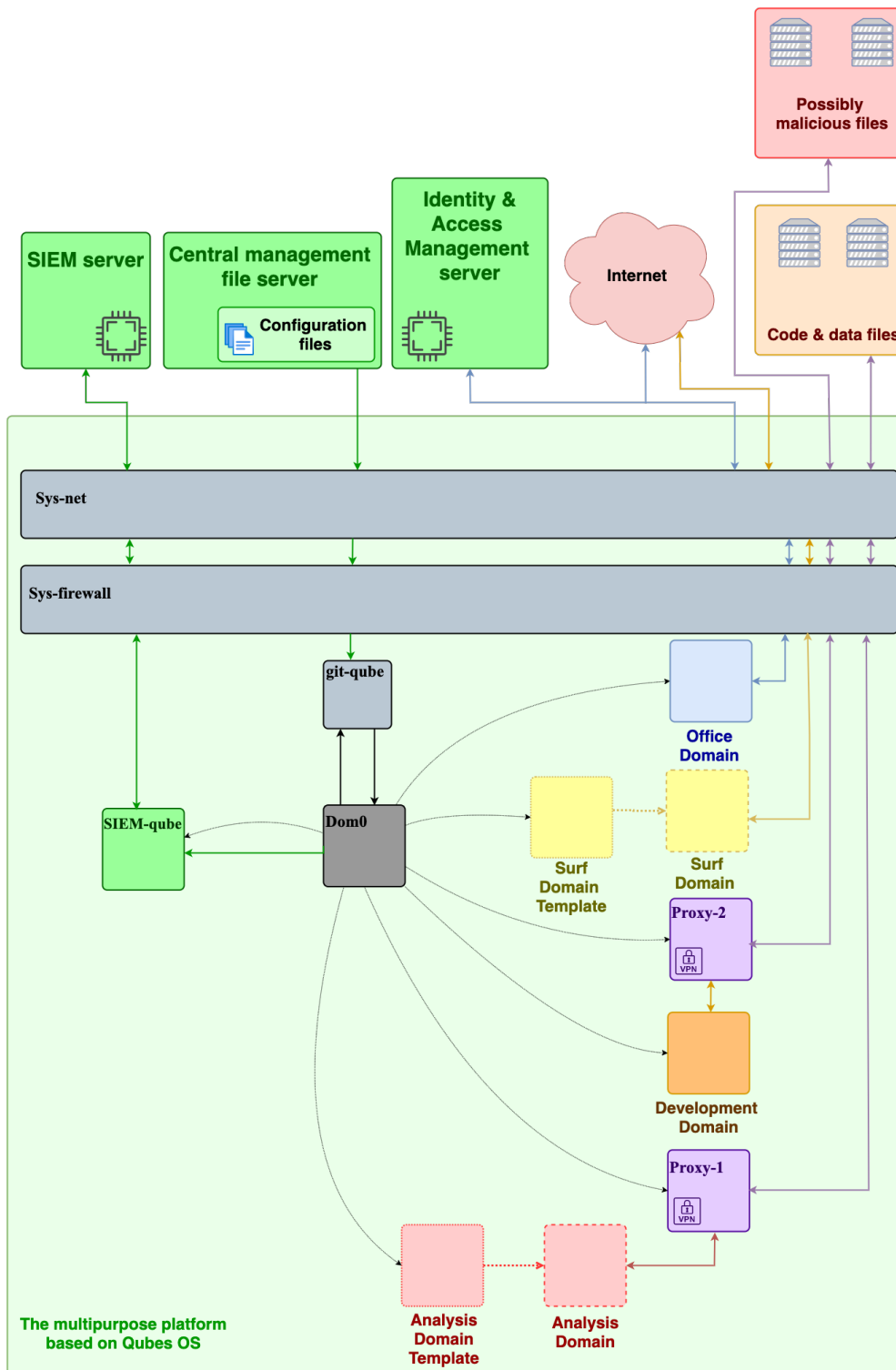


Figure10.2: Overview

**Listing 10.1:** Script variables

```
#!/bin/bash

# Variables
# -----#
VM_NAME=git-qube
GIT_PASSWORD=MyPass
GIT_USER=MyUser
GIT_REPO=MyRepo/saltstack_repo
GIT_HOST=bitbucket.org/
SALT_DIR_PREFIX=/srv/salt
SALT_DIR_TARGET=saltstack_repo/salt
OTHER_DIR_PREFIX=/home/user/git
OTHER_DIR_TARGET=saltstack_repo
```

First, the script will check if a VM with the given VM\_NAME is present on the platform checking the list of VMs with the command `qvm-ls --fields=NAME`. If it is in the list, it will be deleted without any further warning. This is because there cannot be two VMs with the same name. In order to prevent data loss from a possible deletion, the user should make sure not to name any VM the same as VM\_NAME. The removal of the VM is done by the function `remove()`. The VM is first shut down, then the command `qvm-ls --fields=NAME --halted` is used to make sure that the VM is halted before the remove command is run.

**Listing 10.2:** Configuration downloading script

```
# Functions
# -----#

#Function for removing VM:
remove() {

    #Halt vm:
    qvm-shutdown '${VM_NAME}'

    #Check the list of halted VMs until the git-vm appear
    vm=""
    until [ "$vm" == '${VM_NAME}' ]; do
        for name in $(qvm-ls --fields=NAME --halted); do
            if [ "$name" == '${VM_NAME}' ]; then
                vm="$name"
            fi
        done
        sleep 1
    done

    yes | qvm-remove '${VM_NAME}'
}
```

Then the VM is created. Here it is based on the `fedora-30` template. The template it is based on is

optional, but it must be one that already exists on the platform.

**Listing 10.3:** Create temporary VM

```
#Creating the VM that will download the git-repo
qvm-create --template=fedora-30 --label=red '${VM_NAME}'
```

After confirming that the VM is created, again using `qvm-ls --fields=NAME`, the Git repo is cloned to the VM using the regular Git clone command, and the files are then copied to dom0.

**Listing 10.4:** Cloning repository

```
#Cloning the repo
qvm-run --pass-io '${VM_NAME}' "git clone https://'${GIT_USER}':'${GIT_
↳PASSWORD}'@'${GIT_HOST}':'${GIT_REPO}'.git"
```

When we want to copy files to dom0 in Qubes OS, the files are not copied directly, but only the contents of the file are copied. This is done to protect dom0 from malicious files. The script retrieves the file names from the Git repo and lists them in the variables `otherfiles` and `saltfiles`. Then the content in all those files is retrieved using `cat` and redirected to new files in dom0 with the same names.

**Listing 10.5:** Copying to dom0

```
#Get the file names
otherfiles=$(qvm-run --pass-io '${VM_NAME}' "ls '${OTHER_DIR_TARGET}'/")
saltfiles=$(qvm-run --pass-io '${VM_NAME}' "ls '${SALT_DIR_TARGET}'/")

#Copy the contents from the files
for file in $otherfiles; do
    qvm-run --pass-io '${VM_NAME}' "cat '${OTHER_DIR_TARGET}'/'$file'" > '$
↳{OTHER_DIR_PREFIX}'/'$file'
done

for file in $saltfiles; do
    qvm-run --pass-io '${VM_NAME}' "cat '${SALT_DIR_TARGET}'/'$file'" > '${SALT_
↳DIR_PREFIX}'/'$file'
done
```

The VM is then deleted using the function `remove()`.

## 10.3 Recommendations

If some sort of central management is attempted with a large number of employees using Qubes OS, then we highly recommend the creation of a custom image to be installed with the necessary pre-requisite scripts. This will decrease time spent on each laptop setup, but will add another item that needs to be documented and kept up to date.

When discussing how to best implement configuration management one should decide whether or not the employees will be expected or allowed to create additional qubes. This is an important decision as creating and customizing different qubes is the main focus of the operating system. It would at the same time give a much greater degree of control to the IT staff, as well as a burden as they would need to at least



# Chapter 11

## Qubes OS Hardware Compatibility

We were given four laptops from Kongsberg which we could use for testing Qubes OS. The laptops have different hardware configurations. Components such as CPU, motherboard and memory installed are different from each other, as well as the amount of storage. Motherboard and memory is not important when running Qubes OS, as common use motherboards allow the user to install different operating systems and memory will always accept instructions so long as it is compatible with the motherboard. However, Qubes OS is not compatible with all CPUs [100]. Careful consideration should take place before deciding what CPU should be incorporated in the chosen laptop or desktop. Moreover, the chosen workstation which will run the platform should also host sufficient storage as storing the operating system takes much space. We will explain this in depth in this chapter. We experienced some problems with our laptops, as well as different experience when using the laptop for its intended use in our testing.

Resources in this section is defined as the total amount of processing or storage power installed in the machine, determined by the components that matter the most for this project, CPU, RAM and storage.

The CPU is the one factor in the selection that must be compatible with Qubes OS to be able to install and run Qubes OS. Due to several different kernel versions, different virtualization techniques and technology, not every CPU can support Qubes OS. The Qubes team must therefore test and rely on user reports to identify incompatible hardware. This must be taken into consideration when choosing a solution for this platform, as well at least the minium requirements, 4GB RAM and 32GB of storage [101]. However, you need to allocate more storage space and most likely more RAM or else the platform will not be able to run Windows qubes.

### 11.1 Our experience with the provided laptops

We were given four different laptops, an Elitebook 745 G6, Elitebook 820 G2, Elitebook 840 G3 and a Zbook 15, all from Hewlett-Packard (HP). All laptops are unique in terms of installed hardware components so we could differentiate our experience on the laptops and give feedback on the most important factors a solution like this would need in a professional environment. All laptops are full HD screen, 1920 by 1080 pixels. Below is a table comparing the laptops to factors that impact our experience.

**Table11.1:** Laptop configuration

Name	CPU (clockspeed, turbo)	Memory	Storage	Screen size and weight
Elitebook 745 G6	AMD Ryzen 7 PRO 3700U (2.3 GHz, 4.0 GHz)	16 GB RAM	512 GB SSD NVMe	14 inches, 1.5 kg
Elitebook 820 G2	Intel Core i7-5600U (2.6 GHz, 3.2 GHz)	12 GB RAM	512 GB SSD	12.5 inches, 1.36 kg
Elitebook 840 G3	Intel Core i7-6500U (2.5 GHz, 3.1 GHz)	8 GB RAM	256 GB SSD	14 inches, 1.48 kg
Zbook 15	Intel Core i7-4800MQ (2.7 GHz, 3.7 GHz)	16 GB RAM	512 GB SSD	15.6 inches, 2.93 kg

Upon first receiving the laptops, pre-installed with Qubes OS 4.0, we already experienced that the laptops would tackle this project differently. The AMD Ryzen based Elitebook 745 G6 had problems booting up and had issues with its screen resolution. After updating the kernel, the screen would not turn on when booting up. We therefore deem the laptop Elitebook 745 G6 unusable for this project. Neither the desktop version of the AMD Ryzen 7 3700U, the AMD Ryzen 3700, are not listed as compatible in the official Qubes OS Hardware Compatibility List [100].

The other received laptops were able to boot Qubes OS 4.0 without issue. Between them, using them for Linux based operating system, and Qubes OS itself, we experienced a usable and enjoyable experience for our tasks, testing and proof of concepts. It should be noted the user will experience slow downs compared to using a singular operating system. The user will have to wait for the qubes to start when starting applications across qubes. This is usually a “do and forget” operation when starting up the machine for the first time.

For Windows qubes, however, we experienced significant slow-downs compared to Linux based operating systems. At times, the interface was delayed or un-responsive until it could catch up. We also experienced Windows qubes to be resource hungry compared to Linux based qubes, e.g. RAM, CPU and storage. At times, other qubes must be shut down so the resources can be redirected to the Windows qube instead because there is not enough available RAM. This is of course a general problem for all qubes, even Linux based qubes if there are enough, but even more so with Windows due to high minimum requirements.

As we discussed in Chapter 8, we experienced different problems on the machines as the PCI device architecture were different between the machines. At the end of the project period, we ran into another problem after testing Qubes OS 4.1 on one of the machine, the Elitebook 840 G3. The taskbar removed itself, most likely due to a bug. We would like to mention this issue, as problems like these are common in Qubes OS. Luckily the laptop is still usable by finding alternate ways outside of using the taskbar. While tracking down this issue, we found a related issue on the Qubes OS GitHub page, open since 2016 with no remedy for our problem [102].

We also noticed some inconsistencies between the different Qubes OS installations. Even though all the laptops were installed with the same version, the Qubes OS inter-qube copy paste system would not work on the Zbook 15. This functionality did work on the other machines, so it may attributed to the fact that Qubes OS is not compatible to the fullest with all machines.

## 11.2 Laptop vs desktop vs cloud

There are three ways this platform can be delivered to the users, on a laptop, on a desktop or function as a service in the cloud. A desktop would have been the common old solution, but a more modern approach would perhaps be a laptop or cloud solution (a server accessible with Remote Desktop Protocols or other means). They all excel at different areas important to the business and this project, such as experienced performance of the platform, how flexible is the solution to the individual user, and lastly cost efficiency.

Laptops are more prone to thermal throttling, however this mostly applies to graphics cards. It should be noted that most laptops sport a laptop configured Qubes OS, which often run at lower clock speeds as their desktop alternatives (to preserve battery).

Below is a table comparing the different posts with a solution for shipping the platform.

**Table11.2:** Comparison between laptop, desktop and cloud

Solution	Laptop	Desktop	Cloud
Flexibility	Easy to carry and transport	Hard to transport	Easy to connect from all over the world with Internet access, with penalty to experienced quality
Performance	Lower performance on laptop parts compared to desktop alternatives, higher temperatures with reduced airflow in a small chassis reduce performance	Best available performance with dedicated resources per workstation	Great performance with shared server resources
Cost efficiency	Low cost efficiency, low capability for adding resources	Great cost efficiency, easy to add or swap resources according to demand	Great cost efficiency. Cost efficiency is split on how many users; more users would provide greater cost efficiency as they share server resources, easy to add resources

Laptops are common in the industry, following the bring your own device IT policy [103]. Each worker is allowed to bring their own device, or commonly given a work related device following policies set by the local IT department with limitations related to security, e.g. storing of sensitive files cannot be done on the device. Their function is the same as the old desktop workstation, but allowing a more flexible solution for the worker. The user can now easily bring the laptop to meetings or home for a flexible working environment. Laptops are an easy and flexible solution, but has some negative concerns for a project like this. Virtualizing other VMs, especially several at a time, is resource intensive on the system. To accommodate for more resources, results in either more expensive components or sacrificing the flexibility of the laptop by increasing the weight.

Below is comparison between the desktop hardware compared to the same component designed for laptop use, Ryzen 7 PRO 3700 versus 3700U found in the Elitebook 745 G6.

As we can see, the desktop variant for this processor offers almost three times better performance for the laptop equivalent. While this processor is the strongest in the laptops we tried, we were not able to test Qubes OS on it. It should also be mentioned that the Ryzen 3700U only has 4 CPU cores and 10 GPU

CPU Name	CPU Mark (higher is better)	Rank (lower is better)	CPU Value (higher is better)	Price (USD)
AMD Ryzen 7 PRO 3700	22,164	127	67.17	\$329.99*
AMD Ryzen 7 PRO 3700U	7,446	628	NA	NA

**Figure11.1:** Comparison between a Ryzen 7 3700 and 3700U [104]

cores. However, GPU passthrough (giving virtual machines, in this case a qube, direct access to the GPU to perform a job) is not a function in Qubes OS yet, only dom0 is able to use the GPU functionality, yet the only responsibility of dom0 is to function as an admin for the other qubes. This means that a lot of the power Ryzen 3700U can output is wasted when running on Qubes OS, and therefore delivers poor cost efficiency.

Desktops are less mobile, but provide greater availability to resources at lower cost and greater expandability options, such as installing RAM, harddrives and changing CPU. They also have larger upfront cost, as they also need monitors, mouse and keyboard and other peripherals to offer accessibility to the user. Desktops also allow installation of additional PCI devices, such as USB controllers, that can be dedicated to a Windows qube.

Desktop CPU solutions can also run at higher frequency longer due to better thermal management as there is more space for the air in the chassis to circulate, compared to size-limited laptops designed to be mobile. Especially under heavy workload, laptops will reduce CPU frequency to lower the temperature due to low air circulation, resulting in worse performance than a desktop equivalent.

More and more solutions migrate to the cloud for both cost efficiency and flexibility on the users. Moving the solution to the cloud is allocating a server, or several, to host the solution for users, and allowing them to use the solution remotely.

Currently Qubes OS only supports traditional laptop or desktop configurations, but plans for a SaaS (or as the Qubes team likes to call it, Qubes-as-a-Service) platform is in progress, called Qubes Air [105]. Qubes Air will function the same as a standard Qubes OS install, with different zones and maybe even include “air-gapped” devices, allowing qubes to run on separate devices. The project is aimed at allowing for better security through compartmentalization applicable for more scenarios for the modern company. Development was first announced in January 2018, but no progress has been documented to the public, other than remarks that the project is still in development [106].

A cloud solution (Qubes-as-a-service) would get rid of many of the inhibitors the user will experience and find ways to work around in Qubes OS environment on their laptop, such as screensharing of the Qubes OS desktop, or remote controlling the platform. However, this must be done at the cost of trusting the user more to keep access keys safe, as well as keeping important or dangerous files limited to the cloud solution.

This must be done at the risk the user losing the added security that Qubes OS provide to a laptop. The laptop must then be treated as a normal laptop, with access to malware analysis platform in the cloud.

Going for a cloud solution also removes the deployment cost in terms of finding a compatible laptop or desktop with Qubes OS. The Qubes Air project will most likely also not rely on the Xen hypervisor as a single point of failure for security issues [107], but at this point nothing has been released regarding this other than a wish from the Qubes team.



## 11.3 Investigation Summary

Running several operating systems on a single system demands many resources in form of CPU, memory and storage. In terms of cost efficiency, laptops will always be more expensive per unit of resource (CPU, RAM and storage) due to space limitations as the laptop must be light and easy to move. Desktops will provide more power for this resource hungry platform, but the user will not be able to easily move the platform, e.g. to work from home or move the platform to attend a meeting. A cloud solution might be the best of both worlds, as it can be accessed from desktop and laptop. E.g. in a bring-your-own-device company, the user can then access the cloud platform on their laptop from company network and at home (most likely via VPN or other secure solutions). It should be noted that a cloud solution would also rely on laptops to function, however, as the files are not stored on the device (only the access configuration), a stolen laptop would not infringe on the security of the company.

No matter the chosen solution, the PCI device architecture of the workstation must be discussed and customized to fit your project. It is unsure how this will work in Qubes Air as it has not been released as of writing, and no documentation has been released regarding this on the project.

The limiting factor for compatibility with Qubes OS is the CPU compatibility [100]. As laptops use their own mobile version of a commercialized processor, we would recommend using desktop instead. Qubes OS compatibility list, at the time of writing, is a bit outdated on the newest laptops with better processors and testing would need to be done at own expense. A desktop solution would have the positives of having better air circulation due to a bigger chassis and more fans, granting more consistent and greater clock speed for the CPU.

We would recommend looking out for the Qubes Air project and what may come of it in the future. By removing all of the limiting factors a user might experience on a laptop or desktop, the user can instead use a operating system they are comfortable with and use the secure Qubes OS in the cloud.

## 11.4 Hardware recommendations

For desktop and both laptop solutions, we would recommend the following hardware for the platform. As we mentioned above, the minimum requirements for Qubes OS is 32GB of storage and 4GB of RAM. We would recommend increasing the storage to 256 GB or more, or 128GB at minimum. The platform requires Windows machines which takes up a lot of space compared to other operating systems. We recommend at least 8GB of memory, preferably 16GB of RAM, again, because of the amount of virtual machines running at the same time. At last, due to Qubes OS being mostly CPU bound it is hard to recommend a single CPU. We would therefore recommend a later generation Intel processor, with preferably 4 or more cores.

## Chapter 12

# Risk Assessment of the Multipurpose Platform

This chapter present a risk assessment for our proposed multipurpose platform. We will follow the relevant advice and recommendations in the documents “Veileder i sikkerhetsstyring” [108] (related to ISO/IEC 27001 [109]) and “Grunnprinsipper for IKT-sikkerhet” [110] (related to ISO/IEC 27002 [111]) from The Norwegian National Security Authority [112].

The platform could possibly handle data assets of various protection degrees. Asset valuation of the client’s data assets is out of scope for this project, and for that reason the data assets mentioned in this chapter are generalized into what zone they belong to. We advice that the client builds on this risk assessment and further evaluate the platform according to the requirements in Virksomhetsikkerhetsforskriften §12 “Vurdering av risiko” [108], taking into consideration their own asset evaluation of the data that will be stored in and processed by the platform. In relation to this we also advice that the client revise our recommended countermeasures so that the assets recieve the appororiate protection degree.

We will discuss how the laptop will be used in the use-case diagram, and ways of causing harm onto or with the laptop in the abuse-case diagram. This assessment will not cover all possible scenarios, but is made to highlight how using Qubes OS on corporate laptops alter and retains certain risks concerning KDA.

The assessment will give an overview of what risks the platform is facing, what risks Qubes OS can mitigate, and what extra countermeasures we recommend. The countermeasures that we recommend in this chapter are based on Virksomhetsikkerhetsforskriften §15 “Prinsipper ved valg og utforming av sikkerhetstiltak” [108], and on the relevant recommended measures presented in “Grunnprinsipper for IKT-sikkerhet”. These measures include multifactor authentication, disk encryption, protection from malware in email and web browsers, restoring configuration, as well as logging and monitoring relevant security data.

Even though the risks below can make the platform seem inherently dangerous, it is important to note the security benefits that Qubes OS provide versus a normal company issued laptop. A company issued laptop is commonly a single operating system laptop, usually Windows or Linux depending on the user. This is also a single point of failure for the device, if malware compromises the machine the whole platform is compromised. In a Qubes OS based platform, a compromised qube is realistically not a threat, as the qube can be deleted whilst all other data on the platform is out of reach from the attacker. Qubes OS also offers protection by providing networking qubes and USB qubes, and excluding dom0 (a potential single point of failure for the platform) from the rest of the machine.

## 12.1 Main assets

Assets are any hardware, software or information that supports the activities of a client [113]. Assets should be protected against illicit disclosure, theft, destruction and/or alteration from malicious actors to protect Kongsberg Group from potential damage. Table 12.1 lists the assets we have identified for this platform:

**Table 12.1:** Main assets

Asset name	Description
Virtual machines (qubes)	Every VM on the machine has information KDA does not want disclosed, e.g. configuration files and sensitive data
Laptop	The multipurpose analysis platform
Logs (event logs, metadata, browser logs, qube logs etc..)	Computer generated logs with timestamps about past activity on the platform
Passwords	Passwords or encryption keys to the disk or to Qubes OS itself
Authentication data (to SIEM, IAM or VPN)	Signatures and login data used by the computer to connect to company services, such as SIEM, IAM or VPN
Data on internal KDA network	Data on the local network, hosted on servers that the platform can connect to
Surf domain data	Data found in the surf domain, commonly browser data, cookies and other recreational data
Office domain data	Company confidential data the user may possess
Testing/development data	Company software and testing suites
Malware analysis data	Malware analysis reports and live malware
Hardware authentication device	USB device used for multi-factor authentication when logging into the laptop
Configuration management code	SaltStack code and scripts used to configure and update the laptops

## 12.2 Use cases

We will go into detail about how the system can be misused, but first we will introduce how the system can be used by a normal user. This will be explained through the use-cases and the use-case diagram below.

Inside the domain highlighted in Figure 12.1 are VMs (a qube), and can be any operating system of choice (Windows, Linux, Unix, FreeBSD, etc..). The use case paints a picture how this platform can be used, and is needed to explain how KDA or the user can be exploited later in this chapter.

Below is an explanation per point in the use cases. They refer to Use Case 1 through 9.

**Actor:** Person doing the action described in the use case

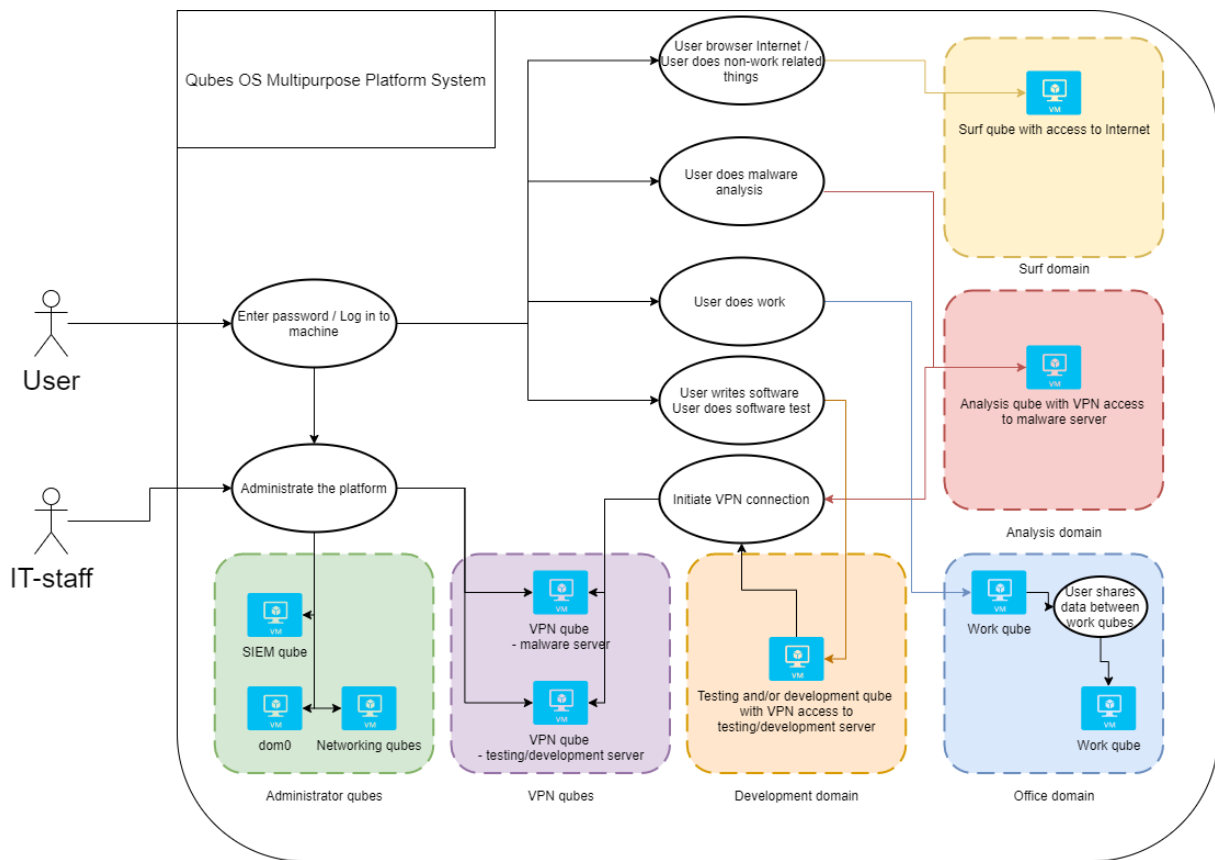
**Prerequisite:** Conditions that must be met before the use case can begin

**Outcome:** The reason why the actor(s) want to do the use case.

**Main flow:** The main event flow the actor will follow to receive outcome

**Side flow:** Optional event flow the actor might follow based on specific condition or desired outcome.

**Deviations:** Event flow that might deviate to the planned behavior due to errors or unplanned circumstances.



**Figure12.1:** Use case diagram

**Table12.2:** Use case 1 - Log in to the machine

<b>Use case 1</b>	<b>Description</b>
Actor	User
Prerequisites	The user has password to the laptop (the password is set during installation of Qubes OS, which can then be changed by the user after first login)
Outcome	The user is logged in
Main flow	<ol style="list-style-type: none"><li>1. The user enters the disk password.</li><li>2. The user enters the main account.</li></ol>
Side flow	
Deviations	<ol style="list-style-type: none"><li>1.1 The user enters the wrong disk password.<ul style="list-style-type: none"><li>• The user is asked again two times.</li><li>• The laptop must be restarted to retry.</li></ul></li><li>2.1 The user enters the wrong account password.<ul style="list-style-type: none"><li>• The user is asked to enter the password again.</li></ul></li></ol>

**Table12.3:** Use case 2 - Browse the Internet/non-work related tasks

<b>Use case 2</b>	<b>Description</b>
Actor	User
Prerequisites	The user is logged in to the laptop
Outcome	The user is browsing the Internet inside the surfing domain
Main flow	<ol style="list-style-type: none"><li>1. The user clicks the application button.</li><li>2. The user selects the surfing domain.</li><li>3. The user selects the desired application (for example Firefox).</li><li>4. The user closes application when finished.</li></ol>
Side flow	
Deviations	

**Table12.4:** Use case 3 - Malware analysis

<b>Use case 3</b>	<b>Description</b>
Actor	User
Prerequisites	The user is logged in to the laptop
Outcome	The user is ready to analyse malware inside the malware domain
Main flow	<ol style="list-style-type: none"> <li>1. The user clicks on the application button.</li> <li>2. The user selects the analysis domain and machine.</li> <li>3. The user selects the desired application.</li> <li>4. When the qube starts, the assigned NetVM also starts. In this case the assigned NetVM is a ProxyVM using the malware VPN connection.</li> </ol>
Side flow	
Deviations	<p>4.1 The ProxyVM does not function properly.</p> <ul style="list-style-type: none"> <li>• The user must restart the ProxyVM.</li> </ul>

**Table12.5:** Use case 4 - Office related work

<b>Use case 4</b>	<b>Description</b>
Actor	User
Prerequisites	The user is logged in to the laptop
Outcome	The user is ready to do business related work in the business domain
Main flow	<ol style="list-style-type: none"> <li>1. The user clicks the application button.</li> <li>2. The user selects the analysis domain and machine.</li> <li>3. The user selects the desired application.</li> <li>4. When the qube starts, the assigned NetVM also starts.</li> </ol>
Side flow	
Deviations	

**Table12.6:** Use case 5 - Write software

<b>Use case 5</b>	<b>Description</b>
Actor	User
Prerequisites	The user is logged in to the laptop
Outcome	The user is ready to write software in the testing domain
Main flow	<ol style="list-style-type: none"> <li>1. The user clicks the application button.</li> <li>2. The user selects the testing domain and machine.</li> <li>3. The user selects the desired application.</li> <li>4. When the qube starts, the assigned NetVM also starts. In this case the assigned NetVM is a ProxyVM using the development VPN connection.</li> </ol>
Side flow	
Deviations	<p>4.1 The ProxyVM does not function properly.</p> <ul style="list-style-type: none"> <li>• The user must restart the ProxyVM.</li> </ul>

**Table12.7:** Use case 6 - Administer the platform

Use case 6	Description
Actor	IT staff and user
Prerequisites	Is logged in to the laptop
Outcome	The platform is updated to the desired state
Main flow	
Side flow	<ol style="list-style-type: none"> <li>1. Change user password               <ol style="list-style-type: none"> <li>1. Change from command line</li> </ol> </li> <li>2. Download SaltStack config               <ol style="list-style-type: none"> <li>1. Download config manually by running the script</li> </ol> </li> <li>3. Change VPN password               <ol style="list-style-type: none"> <li>1. User runs a script which updates password and VPN authentication.</li> </ol> </li> <li>4. Change qube resource usage               <ol style="list-style-type: none"> <li>1. Open Qubes Manager</li> <li>2. Right click specific qube and open qube settings</li> <li>3. Change allocated resources</li> </ol> </li> <li>5. Delete qube               <ol style="list-style-type: none"> <li>1. Open Qubes Manager</li> <li>2. Right click specific qube and delete qube</li> <li>3. Type name of qube when prompted</li> </ol> </li> <li>6. Create new qube based on template               <ol style="list-style-type: none"> <li>1. Open Qubes Manager</li> <li>2. Write name and pick color of qube</li> <li>3. Select template VM.</li> <li>4. Click "Create VM"</li> </ol> </li> </ol>
Deviations	2.1 The laptop has no Internet connection so no download can occur. 4.3 The increase exceeds available resources <ul style="list-style-type: none"> <li>• Qubes OS gives user error and denies the change</li> </ul>

**Table12.8:** Use case 7 - Initial installation on platform

<b>Use case 7</b>	<b>Description</b>
Actor	IT staff and user
Prerequisites	
Outcome	Qubes OS is installed on a new laptop
Main flow	<ol style="list-style-type: none"> <li>1. IT staff installs custom Qubes OS image.</li> <li>2. IT staff runs a script to fetch the latest SaltStack code.</li> <li>3. User runs a script which updates password and VPN authentication.</li> </ol>
Side flow	
Deviations	2.1 The laptop has no Internet connection so no download can occur.

**Table12.9:** Use case 8 - Change/connect to NetVM

<b>Use case 8</b>	<b>Description</b>
Actor	User
Prerequisites	The user is logged in to the laptop.
Outcome	The qube is connected to the desired network qube.
Main flow	<ol style="list-style-type: none"> <li>1. Open Qubes Manager.</li> <li>2. Right click on the qube and select qube settings.</li> <li>3. Under networking, choose the desired NetVM.</li> </ol>
Side flow	
Deviations	

**Table12.10:** Use case 9 - Connect to network

<b>Use case 9</b>	<b>Description</b>
Actor	User
Prerequisites	The user is logged in to the laptop.
Outcome	The platform is connected to a network either through a cable or wirelessly
Main flow	<ol style="list-style-type: none"> <li>1. The user navigates the network menu in the top right to connect to public or corporate network.</li> </ol>
Side flow	<ol style="list-style-type: none"> <li>1. Connect Ethernet cable to the laptop.</li> </ol>
Deviations	



## 12.3 Potential attacker profiles

Today's digital threat landscape consist of many different actors with a wide variety of motivation and capabilities. The Norwegian National Security Authority's report "Helhetlig digitalt risikobilde 2020" highlights state actors and criminal organizations as the most prominent. These actors often wish to obtain trade secrets and advanced technology from for example the defence and maritime sector. They are known to possess extensive resources and perform sophisticated and complex long-term operations, and their attacks often consist of information retrieval and mapping of infrastructure and vulnerabilities that can be used as future sabotage opportunities. The Norwegian Intelligence Service also highlights this method for state actors, and that they often trying to gain access where they have the opportunity to gain accesses or information they might find useful in the future. Growing trends that are emphasized in the NSM report are ransomware attacks, often carried out in combination with phishing campaigns. This type of attack has shown to be directed toward targets with a high ability to pay, such as large companies, and are often carried out by highly competent attackers. In this section, we present all the potential threats we consider for this platform.

Below is a description for all points for how we describe the threats against the platform.

- **Description:** A brief description of the threat, which is an actor (person or organization) with a motivation to do intention on system via attack vector.
- **Motivation:** What makes the threat want to do action against this platform, e.g. what do they get in return for their investment (action).
- **Intention:** What is the threat's plan with the action against the platform.
- **Cyber Attack vector:** "The method or way by an adversary can breach or infiltrate an entire network/system. Attack vectors enable hackers to exploit system vulnerabilities, including the human element" [114]. Can lead to malware installation, which results in further attack vectors from the malware as point of origin.

### 12.3.1 State sponsored actor

**Description**

State sponsored person or person acting on behalf of a state

**Motivation**

Military, monetary and political gains

**Intention**

Information gathering, command and control, blackmail, sabotage

**Common attack vectors**

Social engineering (Phishing, Spear phishing), supply chain attack, theft, vulnerabilities in software & hardware, evil-maid, drive-by-USB, network interfaces (Wi-Fi and Ethernet)

### 12.3.2 Criminal organization

**Description**

Organized criminals attempting to misuse KDA assets

**Motivation**

Monetary gains

**Intention**

Ransom

**Common attack vectors**

Social engineering (phishing, spear phishing), vulnerabilities in software

### 12.3.3 User

**Description**

The user (or IT staff) using the platform. May unintentionally cause harm against KDA assets

**Motivation**

Accidents caused by time constraints, fatigue, insufficient training and information, etc.

**Intention**

Unapproved laptop alterations, cut corners

**Common attack vectors**

Accidental actions, such as deletion, wrong alterations disrupting assets, intentional modification of the platform

### 12.3.4 Insider

**Description**

A person hired by KDA, corrupted with financial gains or spite to disrupt or leak KDA assets

**Motivation**

Monetary gains, revenge

**Intention**

Data leak, sabotage, ransom

**Common attack vectors**

Theft, evil-maid, drive-by-USB, data-leak, install malicious software

### 12.3.5 Corporate spy

**Description**

Other corporations looking to steal and spy on ideas or patents from KDA to better their own competing products

**Motivation**

Competitive advantage

**Intention**

Information gathering

**Common attack vectors**

Social engineering (phishing, spear phishing), theft

### 12.3.6 External opportunist

**Description**

External opportunist, commonly a hacker, sees a vulnerability they can exploit

**Motivation**

Monetary gains, political activism

**Intention**

Data leak, sabotage, ransom

**Common attack vectors**

Theft, vulnerabilities in software, social engineering (phishing, spear phishing)

## 12.4 Abuse cases

In the following tables, we describe abuse cases that influence the KDA assets previously identified in Table 12.1. Below is an explanation to each recurring point in the abuse cases. They apply to Abuse Case 1 through 10.

- **Actor:** Person or organization doing an action against the system. The actors are our previously identified threats.
- **Prerequisite:** Conditions that must be met before the abuse case can begin
- **Purpose:** The reason why the actor(s) want to do the action. The purpose correlates to their intention in the threat table.
- **Assets influenced:** Assets that may be affected by the abuse case.
- **Description:** General information about the abuse case.
- **Risk before Qubes OS controls and countermeasures:** Describing and giving a score to the probability & consequence of the given risk before using Qubes OS or applying any countermeasures.
- **Controls in Qubes OS:** The design or implementation details of Qubes OS that alter the risk scores.
- **Countermeasures:** Actions, policies or rules that can be done before or during the abuse case timeframe.
- **Qubes OS controls and countermeasure's effects on risks:** Effects that the countermeasure may have on the assets after they are implemented.

Figure 12.2 is how we calculate the different scoring of abuse cases. The assessment is a qualitative one, but the diagrams should help indicate our thoughts on these risks.

	Probability - the average of the 3			
	Complexity (one of)		Action frequency	Qualitative assessment of Attackers' motivation
Description	Attacker	User		
Level 1	Requires custom tools and high specific knowledge within the niche area	Must be done with bad intentions	Less than every other year	Victim is not interesting for the attacker
Level 2	Requires custom tools	Does not follow decided procedure	Once every other year	Victim is useful but one of many for the attacker
Level 3	Can be done using existing scripts and tools	Can be done with carelessness	One to twelve times a year	Victim is useful and rare for the attacker
Level 4	Can be done without tools	Can be done in an attempt to perform work tasks (misunderstanding)	More than once a month	Victim is uniquely useful for the attacker

**Figure12.2:** Probability level requirements used in risk assessment

The abuse case diagram below paints a picture of where and how the system is vulnerable to attacks.

<b>Consequence - (highest of the two)</b>		
<b>Description</b>	<b>Working hours lost</b>	<b>Violation of CIA</b>
Level 1	Total <= 1 hour	Surf and similar
Level 2	Total <= 1 day's work	Data on analysis
Level 3	Total <= 3 day's work	Data on development - Other software written by KCSC - Log data
Level 4	Total > 3 day's work	Data on office

**Figure12.3:** Consequence level requirements used in risk assessment

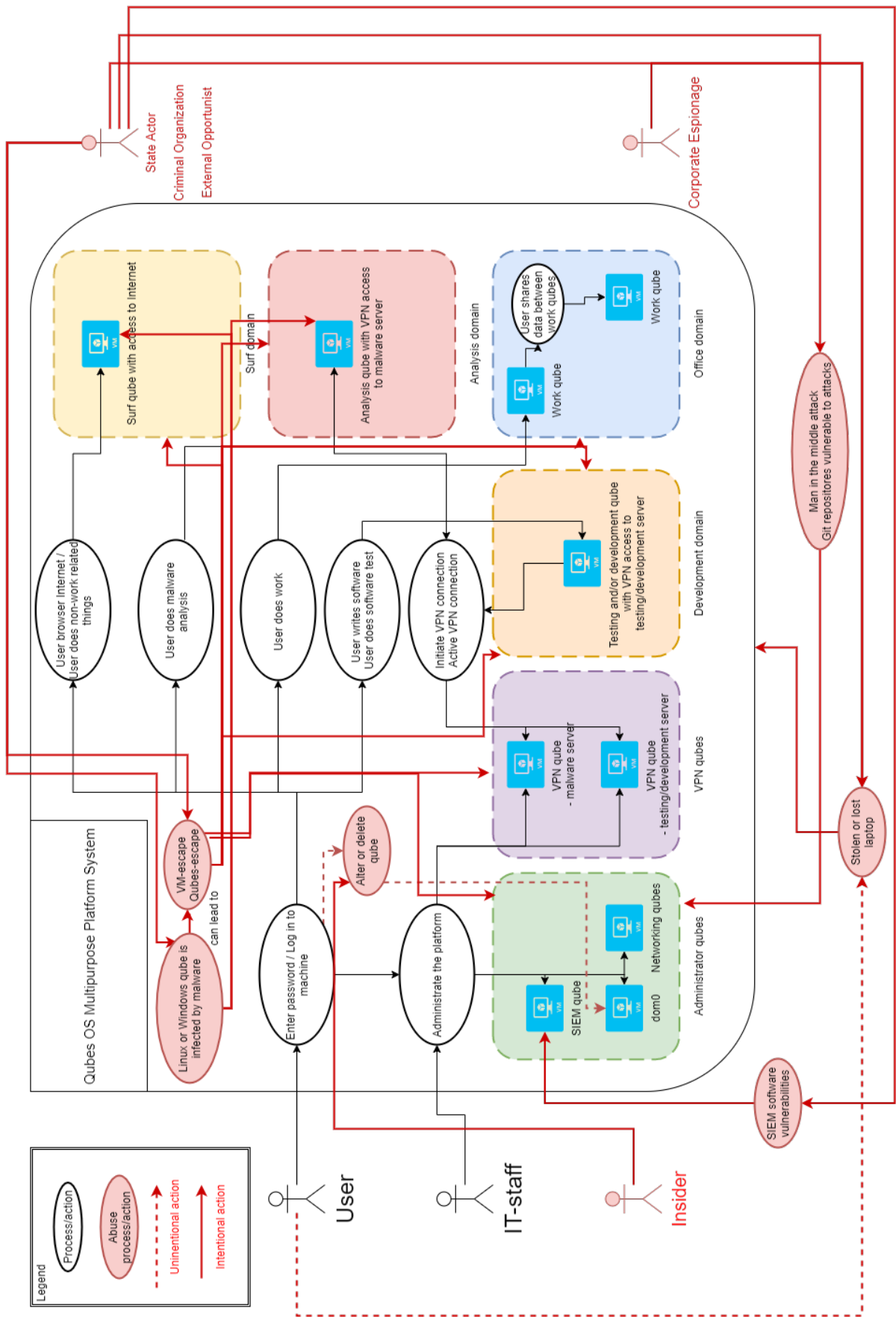


Figure12.4: Abuse case diagram

## 12.4.1 Abuse case 1 - Accidental VM deletion

**Actor** User, Insider

**Prerequisites** User is logged in, a user has the ability to delete or alter qubes

**Purpose** User: Accident; Insider: Sabotage

**Assets influenced** VMs, Authentication data (to SIEM, IAM or VPN), Surf domain data, Office domain data, Testing/development data, Malware analysis data

### Description

This abuse case is only relevant for laptops used in malware analysis. On such laptops type 2 hypervisors are often used as laboratories to test out the malware. Setting these up can take a lot of time.

Example scenario:

Actor alters or deletes VM by accident or on purpose, which results in loss of data.

### Risk before Qubes OS controls and countermeasures

Probability: The user can do this by a mistake.

Consequence: The consequence of deleting a VM depends on what VM is deleted. If a VM is deleted, all data on the VM is lost, and the user may have to make a new VM, which can be a lengthy process. Depending on what data is lost, this might set the user back a considerable amount of time (more than one day).

**Risk (PxC): 3 x 3**

### Controls in Qubes OS

The name of the qube that will be deleted must be written by the user either when using the GUI or dom0. This does not apply to disposable VMs.

[If the last application running in a qube is closed, the qube will shut down. When using a disposable VM, if the user closes an application and wants to use another instead then the qube will be deleted]

### Countermeasures

1. Inform the user of actions that can cause this behavior.
2. Instruct the users not to use the dom0 terminal (reduces the risk of accidentally deleting a qube, but also drastically reduces the users ability to fix issues on the platform).
3. Save documents on network drives

## Qubes controls and countermeasures' effects on risks

The countermeasures cannot ensure that the assets are wholly protected, but will increase the threshold and awareness around accidental data deletion. If a qube is deleted in Qubes OS, and the documents and data is saved on network drives, recreating the qube from the template takes a short amount of time, less than one hour.

**Risk (PxC): 2 x 1**

### 12.4.2 Abuse case 2 - Lost or Stolen laptop

**Actor** Insider, corporate spy, external opportunist, state actor

**Prerequisites** Laptop is in a vulnerable and unsupervised position.

**Purpose** Gain reconnaissance or steal assets and intellectual property from KDA

**Assets influenced** Laptop, Disk, VMs, Authentication data (to SIEM, IAM or VPN), Surf domain data, Office domain data, Testing/development data, Malware analysis data

#### Description

Actor finds the platform (laptop) in an unsupervised vulnerable position and/or state (such as unlocked). With easy access to the platform and the owner not watching, the actor can steal the laptop and use it for their own purposes.

Actors can also steal the laptop after terminated work relations with KDA. Insider knows the current default password on the platforms. Insider uses this information to access a user's laptop without authorization.

#### Risk before Qubes OS controls and countermeasures

**Probability:** Human negligence works together with a threat's attempt to steal laptops. If a laptop is lost, it should be treated as if it was stolen to not be surprised by the consequences. A stolen or lost laptop requires an actor to use some tools or a user to be careless, and it can happen several times pr. year for big companies.

**Consequence:** Laptops which are lost can be acquired by hostile actors and will in such cases possibly reveal confidential information. Both the confidentiality, integrity and availability of the influenced assets will be affected. Everything that is only stored locally on the laptop will be lost. With no countermeasures, a stolen laptop could potentially give the threat actor access to the whole laptop content, including access to any VPN or other service that has credentials saved on the laptop.

**Risk (PxC): 3 x 4**



## Controls in Qubes OS

Not applicable.

## Countermeasures

1. Multi-factor authentication
2. Finders reward and contact details on the laptop.
3. Have procedures in place for changing credentials when the user loses track of the laptop, such as removing the laptop from IAM and revoking active licenses.
4. Disk encryption.
5. do not only store files locally, but on an internal network as well.
6. Have IT staff present when the user is supposed to change passwords for the first time in order to avoid weak passwords.
7. Have a script in the image that runs the first time the machine is logged into that prompts the user to change their default password.

## Qubes controls and countermeasures' effects on risks

The countermeasures mitigate some of the loss of availability, as the files are not only stored locally. The confidentiality and integrity of the assets will be ensured by multifactor authentication and removing access from the laptop to internal services.

**Risk (PxC): 3 x 1**

### 12.4.3 Abuse case 3 - IT staff pushes faulty configuration code which removes functionality

**Actor** IT staff

**Prerequisites** Laptops must be set up to periodically pull configuration.

**Purpose** IT staff could in an attempt to cut corners cause damage.

**Assets influenced** VMs (qubes), Logs (event logs, metadata, browser logs, qube logs etc.) Data on internal KDA network, Surf domain data, Office domain data, Testing/development data, Malware analysis data, Configuration management code

#### Description

IT staff has a lot of immediate power when using configuration management tools. Cutting corners in such cases could result in pushing misconfigurations, which in itself could cause even further problems.

Example scenario:

A user requests a few small alterations in the configuration files. At the beginning of the week the IT staff decides to quickly implement and push out the change. After pushing out the new version it is discovered that it causes downtime in some domain. Now all users have to either manually update after another fix or they need to wait until the next automatic update.

## Risk before Qubes OS controls and countermeasures

Probability: With frequent updates comes frequent possibility of failure.

Consequence: The consequences may vary, but even temporary loss of availability for a chunk of the employees can be very disruptive.

**Risk (PxC) : 4 x 2**

## Controls in Qubes OS

Not applicable.

## Countermeasures

Use Continuous Integration / Continuous Delivery [115] in order to reduce the amount of errors & reduce the time spent to recover from them. Automated tests never cut corners.

## Qubes controls and countermeasures' effects on risks

**Risk (PxC): 2 x 1**

### 12.4.4 Abuse case 4 - SIEM software vulnerability is abused by threat

**Actor** State actor, criminal organization, external opportunist

**Prerequisites** The platform has installed proprietary SIEM software. In case of Splunk, this would be a Splunk Universal Forwarder. The installed SIEM software has security vulnerabilities.

**Purpose** State actor: Information gathering, use the Splunk qube as a further attack vector; Criminal organization, External opportunist: Ransom, use the Splunk qube as a further attack vector

**Assets influenced** VMs (the SIEM qube), Logs (event logs, metadata, browser logs, qube logs etc.) Passwords, Authentication data (to SIEM),

## Description

Depending on the vulnerability that can be exploited, the attacker can exploit the SIEM software installed on the SIEM qube (in our case the Splunk qube).

Actor may try to gather information about the system, impact the system's availability by denying access to the system or use the vulnerability to further attack the system, depending on the severity of the vulnerability.

In the event of an attack, the attacker may want to tamper with the log files (in a Man-in-the-Middle attack) to prevent SIEM from alerting IT staff to take action.

There have been instances of security vulnerabilities in Splunk [116].

## Risk before Qubes OS controls and countermeasures

**Probability:** This could be done by existing tools, and is seen to happen in Splunk less than once a year [117]. There is a possibility that KDA is an interesting and unique target for the actor.

**Consequence:** If the logs were to be altered by an attacker, it would affect the confidentiality, integrity and availability of the logs. An attacker could alter the logs to hide another attack from IT-staff monitoring, or use the logs to gather intelligence about the system.

**Risk (PxC): 3 x 3**

## Controls in Qubes OS

Not applicable.

## Countermeasures

1. Keep SIEM software up to date
2. Keep IT staff up to date regarding SIEM vulnerabilities and attacks (in the news or social media)
3. Add authentication and end-to-end encryption to the log transport if missing

## Qubes controls and countermeasures' effects on risks

1. Decreases the likelihood that a vulnerability will be exploited, this will require the threat actor to make new tools.
2. Stops third-parties from looking at and altering traffic.

**Risk (PxC): 2 x 3**

## 12.4.5 Abuse case 5 - Fault in scripts opens a security vulnerability

**Actor** State actor, Criminal organization, Insider, Corporate spy, External opportunist

**Prerequisites** Scripts authored or approved by KDA present on the platform has a security vulnerability

**Purpose** Use the vulnerability as a further attack vector

**Assets influenced** VMs (qubes), Logs (event logs, metadata, browser logs, qube logs etc.) Passwords, Authentication data (to SIEM, IAM or VPN), Data on internal KDA network, Surf domain data, Office domain data, Testing/development data, Malware analysis data, Configuration management code

## Description

Actor can exploit vulnerabilities found in customized scripts for the platform. The accessibility of the exploit is dependent on the need for the actor to be on the same network or over the Internet. The severity of the exploit depends on the vulnerability present and the capability of the actor.

A cause for this can be, e.g., lack of authentication, code-injection vulnerabilities, storing important data in clear text or reversible hashes, and any other software vulnerabilities that can be found in scripts.

Example scenario:

Script used to update the configuration management code could be written to connect via http instead of https and expose traffic while still maintaining functionality. Since the functionality is not lost, the fault is hard to detect and the code is not reviewed. Exposed traffic can reveal the configuration files and user credentials.

## Risk before Qubes OS controls and countermeasures

Probability:

The probability that the scripts has a fault: Can be done by a mistake, but the developer will try to not make the vulnerability. The probability of this happening is level 2.

Probability of attack:

The attack could be done by existing tools, There is a possibility that KDA is an interesting and unique target for the actor. The probability is at level 3

Combined probability:

$2/4 * 3/4 = 2/4$  The probability is at level 2.

Consequence:

Fault in scripts that communicate over the Internet could give away confidential information or credentials. Credentials or attacks such as code injection can be used by the actor to gain further access to the system, and could affect the confidentiality, integrity and availability of potentially any data.

**Risk (PxC): 2 x 4**

## Controls in Qubes OS

Qubes OS isolates domains and VMs, which means that an attack only would affect the qubes that the script concerns. The qubes that are not in contact with the script will not be affected, unless it is infected with malware capable of VM-escape or qubes-escape.

## Countermeasures

1. Refactor code
2. Proof read and test code before deploying
3. Penetration test the platform after big changes or with set intervals

## Qubes controls and countermeasures' effects on risks

The controls do not mitigate the risk of the particular domain affected by the script, but significantly reduces the risk to the other domains on the laptop. The countermeasures cannot ensure that the scripts are definitively secure, but will decrease the probability of failures.

**Risk (PxC): 1 x 4**

### 12.4.6 Abuse case 6 - Actor uses git repository to push malicious code

**Actor** State sponsored actor, Criminal organization, User, Insider, Corporate spy, External opportunist

**Prerequisites** Access to the git repository

**Purpose** User: Accident; State (sponsored) actor, Corporate spy: Reconnaissance; Criminal organization: Ransom; Insider, External opportunist: Sabotage

**Assets influenced** VMs (qubes), Logs (event logs, metadata, browser logs, qube logs etc.) Passwords, Authentication data (to SIEM, IAM or VPN), Data on internal KDA network, Surf domain data, Office domain data, Testing/development data, Malware analysis data, Configuration management code

## Description

An actor pushes malicious code to the git repository that will infect the users machines when they clone the SaltStack code. The malicious code can be used to spy on users, to spread ransomware, or to sabotage users platforms.

1. An attacker with access to view the git repository files can easily see that SaltStack code is used.
2. The attacker then engineers SaltStack code to download malware.
3. Plants malware in the git repo that will be downloaded.
4. Engineers SaltStack code that executes the malware.
5. This way the malware can infect the whole platform.

## Risk before Qubes OS controls and countermeasures

**Probability:** If no countermeasures are taken, it could be very easy for an attacker to gain access to the repository. No tools are needed to push the code, but the attacker needs to write the code and use new or existing malware. There is a possibility that KDA is an interesting and unique target for the actor.

**Consequence:** All data could be at risk, and the lost work could be over 3 days. An example of how much a ransomware attack can cost is the Hydro attack in 2019, estimated to 550-650 MNOK [118].

**Risk (PxC): 3 x 4**

## Controls in Qubes OS

Not applicable.

## Countermeasures

1. Keep the Git repository on trusted servers on a local trusted network.
2. Only authorizing a few trusted employee positions with access to push to the repository.
3. Authenticate code through hash sums compared to ones on the company website.

## Qubes controls and countermeasures' effects on risks

The countermeasures does not mitigate the consequences of malicious code in the repository, but it can authenticate the code, unless the website is compromised. It will also decrease the probability of compromise of the repository, because the actor has to get into the local network and gain the rights to push to the repository. This combination will make the probability small.

**Risk (PxC): 1 x 4**

### 12.4.7 Abuse case 7 - Actor uses git repository to pull and analyze information

**Actor** State (sponsored) actor, Corporate spy, External opportunist

**Prerequisites** Actor has access to the repository.

**Purpose** State (sponsored) actor, Corporate spy: Reconnaissance; External opportunist: Reconnaissance, data leak

**Assets influenced** Configuration management code

## Description

An actor pulls files from the git repository to analyze and gain knowledge of the platform structure, or obtain sensitive files possibly located in the repository. If the file server is accessible over the Internet it might be vulnerable to a man in the middle attack.

## Risk before Qubes OS controls and countermeasures

**Probability:** Probability: If no countermeasures are taken, it could be very easy for an attacker to gain access to the repository. No tools are needed to see the code.

**Consequence:** The actor can view the code which gives the actor information that can be used in future attacks. This affects the confidentiality of the software data making this level 3.

**Risk (PxC): 4 x 3**

## Controls in Qubes OS

Not applicable.

## Countermeasures

1. Keep the Git repository on trusted servers on a local trusted network.
2. Only allow authenticated users to pull files.
3. Route traffic travelling over the Internet through an encrypted VPN.
4. SSH.

## Qubes controls and countermeasures' effects on risks

The countermeasures cannot ensure that the code is definitively secure, because an attacker can potentially obtain a user account with access to the git repository, but will decrease the probability of compromise of the repository, because the actor has to get into the local network and obtain a user account. This combination will make the probability small.

**Risk (PxC): 1 x 3**

### 12.4.8 Abuse case 8 - User runs malware in an unsafe environment

**Actor** User

**Prerequisites** User is logged in, malware must be downloaded

**Purpose** Accident

**Assets influenced** VMs (qubes), Logs (event logs, metadata, browser logs, qube logs etc.) Data on internal KDA network, Surf domain data, Office domain data, Testing/development data,

## Description

A user mistakenly runs malware in an unsafe environment. If a user acquires malware for analysis purposes and does not follow proper procedure they might accidentally run it outside the virtual environment and compromise the analysis laptop. If that machine is connected to an internal network then it could propagate even further.

## Risk before Qubes OS controls and countermeasures

Risk of malware affecting the laptop 8.1:

Probability: A user could do this by accident.

Consequence: Depending on the malware, the confidentiality, integrity and availability of the data on the analysis laptop can be affected. All data could be at risk, and the lost work could be over 3 days.

**Risk (PxC): 3 x 2**

Risk of malware affecting the network 8.2 :

Probability: A user have to both make the mistake of running the malware in the wrong environment, be connected to the network, and the malware have to be capable of spreading through the network.

We take into consideration the probability of the user making the mistake, and the probability of the malware being capable of network propagation:  $3/4 * 2/4 = 2/4$  The calculation gives a probability level of 2.

Consequence: Depending on the malware, the confidentiality, integrity and availability of the data on several of the laptops on the network might be affected. All data could be at risk, and the lost work could be over 3 days. An example of how much a ransomware attack can cost is the Hydro attack in 2019, estimated to 550-650 MNOK [118].

**Risk (PxC): 2 x 4**

## Controls in Qubes OS

### Colour Labels

If different domains are associated with different colors then one will be less likely to perform actions in the wrong domain.

However, using color labels introduces a new issue:

The correct colour and rules must be applied to the qube when it is created. Unknowing users might think picking a colour alters the configuration which is not the case and can cause trouble.

Example case of colour confusion: A user creates their own qube, and labels it with the colour of the analysis domain without applying the rules of the analysis domain. The user can also change the netVM of an analysis qube manually, without the qube changing colour. This can lead to confusion, and result in the user running malware in a qube that has the right colour, but not the right configurations.

The consequences of this can be that malware is run in a qube that has access to the wrong VPN connection, or to the Internet, thereby spreading the malware.

## Countermeasures

1. Inform the users of this risk to increase awareness.
2. Clear rules and policies for color coding qubes.
3. Users can be instructed not to change any configurations in the analysis domain.
4. Users can be instructed not to change any configurations at all, and use the SaltStack configurations only.

## Qubes controls and countermeasures' effects on risks

The countermeasures cannot ensure that the malware is never run in an unsafe environment, but will decrease the likelihood of this scenario. It will also reduce the possibility that the analysis is done while connected to a network.

**Risk (PxC) 9.1: 2 x 2**

**Risk (PxC) 9.2: 1 x 4**



## 12.4.9 Abuse case 9 - Linux or Windows VM is infected with malware through social engineering

**Actor** Insider, state actor, criminal organization, corporate spy, external opportunist

**Prerequisites** Adversary can replicate a trusted source to the user User clicks malicious link

**Purpose** Criminal organization & state actor: information gathering, C2, blackmail, sabotage external opportunist: Sabotage Corporate spy: information gathering

**Assets influenced** Which assets that are influenced depend on which domain is infected with malware.

All domains except analysis domain:

- VMs (qubes) - The qube infected with malware.
- Logs (event logs, metadata, browser logs, qube logs etc.) - Logs of the qube infected with the malware

Domain specific:

- Office: Authentication data (IAM), Data on internal KDA network, Office domain data (of the qube infected)
- Development: Testing/development domain data (of the qube infected)
- Surf: Surf domain data (of the qube infected)

### Description

User clicks an untrusted link from a phishing email or website, downloads malicious files and gives the files run permissions. If the local anti-virus or network anti-virus does not stop the malware, the qube that downloaded the file will be infected by the malware. Depending on the malware, the infected qube can now encrypt data and assets. The decryption can be done in exchange for money (ransomware). Or the malware can act as a reconnaissance gatherer, data and assets exfiltrator, etc.

Example scenario:

User receives an email they believe is from a trusted source. The email includes a link to malware the user downloads. The qube the user ran the malware in is now infected.

### Risk before Qubes OS controls and countermeasures

**Probability:** Can be done with existing tools, phishing containing malware could be expected to happen 1-12 times a year, and the attacker is likely to target the business.

**Consequence:** All data could be at risk, and the lost work could be over 3 days. An example of how much a ransomware attack can cost is the Hydro attack in 2019, estimated to 550-650 MNOK [118].

**Risk (PxC): 3 x 4**

## Controls in Qubes OS

Phishing attacks is most likely to happen when the user is opening email attachments or Surfing. Surfing qube is isolated from the rest of the machine. All qubes are isolated from each other. Analysis and development machine has no internet connection. Links can be open in disposable qubes that disappears when they are closed. If the qube has not been given permission by dom0 to communicate with other qubes, then the malware will have no effect on the platform other than the infected qube (given that the malware is not capable of VM-escape).

## Countermeasures

1. Open suspicious links in a disposable qube. The malware will be deleted together with the disposable qube when it is closed.
2. Disallow untrusted files from executing on laptop, either by rules (only whitelist trusted sources) or user policies (inform the user of trusted sources)
3. Require antivirus on qubes requiring Internet connection or mail.
4. Scan files travelling on the network for malicious intent or signatures (presumes user is on the local (company) network)
5. Active SIEM monitoring and analysis to flag malicious behaviour

## Qubes controls and countermeasures' effects on risks

The countermeasures cannot ensure that the assets are wholly protected, but will instead decrease the asset's exposure to malware. Opening suspicious links in disposable qubes considerably mitigates the risk of the malware spreading on the laptop, the only exception being malware capable of VM-escape.

**Risk (PxC): 3 x 1**

### 12.4.10 Abuse case 10 - Laptop is infected with malware capable of VM escape

**Actor** State actor, criminal organization, external opportunist

**Prerequisites** User or insider has downloaded malware capable of VM escape on the Xen hypervisor

**Purpose** State actor: Sabotage, ransom, information gathering; Criminal organization: Sabotage, ransom; External opportunist: Sabotage, ransom, data leak

**Assets influenced** VMs, Disk, Logs (event logs, metadata, browser logs, qube logs etc.) Authentication data (to SIEM, IAM or VPN), Data on internal KDA network, Surf domain data, Office domain data, Testing/development data, Malware analysis data

## Description

Actors can traverse the virtual machines found on the platform by exploiting hypervisor vulnerabilities [107]. These are rare, but incredibly problematic when performing analysis as they threaten to infect the analysis platform and connected network.

### Risk before Qubes OS controls and countermeasures

Risk of malware capable of VM-escape affecting the laptop 10.1:

Probability: These attacks are rare and could aim at a wide audience.

Consequence: Depending on the malware, the confidentiality, integrity and availability of the data on the analysis laptop can be affected.

#### Risk (PxC): 2 x 2

Risk of malware capable of VM-escape affecting the network 10.2 :

Probability: Such an attack is more difficult, but is likely to be used by dedicated adversaries, who are willing to spend more resources.

Consequence: Depending on the malware, the confidentiality, integrity and availability of the data on several of the laptops on the network might be affected. All data could be at risk, and the lost work could be over 3 days. An example of how much a ransomware attack can cost is the Hydro attack in 2019, estimated to 550-650M NOK [118].

#### Risk (PxC): 2 x 4

### Controls in Qubes OS

Not all Xen vulnerabilities are applicable to Qubes OS, and this abuse case requires more dedication and resources from adversaries to be able to gain access to the entire platform (via dom0) [107].

According to testing done by the Qubes OS security team, the Xen hypervisor was susceptible to VM escape. The team has found some of these and introduced counter measures in recent versions [119]. This only applies to the paravirtualized VMs which uses software based virtualization. Windows requires hardware VMs which uses hardware-assisted virtualization. The virtualization technique can be chosen for each VM, and therefore earlier versions of Qubes OS may be utilized should compatibility problems arise.

It should be noted that Qubes OS has removed paravirtualized VM in the Qubes OS 4.0 update. “A more radical reader might be of the opinion that we should completely replace Xen with some other hypervisor. Such an opinion is surely not unfounded, as we have previously expressed our disappointment in the Xen security process [5]. Sadly, not much has improved over the past several months. Moreover, even though Qubes is now based on a hypervisor-abstracting architecture (“Odyssey”), which should make switching to a different VMM a relatively easy task, the primary problem that remains is the lack of a good alternative hypervisor to which we could move [6].” [120]

## Countermeasures

1. Disallow untrusted files from executing on laptop, either by rules (only whitelist trusted sources) or user policies (inform the user of trusted sources)
2. Require antivirus on all qubes
3. Active SIEM monitoring and analysis to flag malicious behaviour (specialized malware can turn off shipping logs to SIEM, negating this countermeasure if the malware got dom0 access)
4. Update Qubes OS whenever a new update is released
5. Store files on the internal network as well as locally
6. Create offline backups of important files.

## Qubes controls and countermeasures' effects on risks

The countermeasures cannot ensure that the assets are wholly protected, but monitoring can alert IT-staff of a spreading malware and stop the attack. An up to date Qubes OS can include the security patches to the malware and will prevent the malware from spreading. Antivirus and up-to-date Qubes OS can give protection against known attacks, but will most likely not protect against zero day attacks, depending on vulnerability.

According to the Qubes OS security team, not all vulnerabilities in Xen that lead to a VM-escape exist in Qubes OS. This means it is still possible, but is considerably more rare such an attack will succeed [107].

If the malware is capable of qubes-escape, the probability of the malware infecting other domains or Dom0 is lowered by the countermeasures but it is still possible, placing the consequence at level 4.

The consequence of VM escape that affects the laptop only (12.1) is higher after switching to QubesOS, because it could compromise all the zones, including the Office zone.

**Risk 12.1 (PxC): 1 x 4**

**Risk 12.2 (PxC): 1 x 4**

## 12.5 Findings

In Figure 12.5 we show the IDs of all the risks in the assessment in a four by four table. The figures show the risk IDs traveling from a higher risk score area to a lower risk score area. Figure 12.7 shows all the risk scores before and after countermeasures, with color coded scores. Abuse case 9 is particularly interesting as it shows some of the benefits of using Qubes OS when dealing with malware infections. Abuse case 10 shows instead some possible new challenges with Qubes OS as all machines are physically connected.

Before Qubes OS controls & countermeasures					
<b>Consequence</b>	4		5, 8.2, 10.2	2, 6, 9	
	3			1, 4	7
	2		10.1	8.1	3
	1				
		1	2	3	4
		<b>Probability</b>			

Figure12.5: Risk overview before Qubes OS & countermeasures

After Qubes OS controls & countermeasures					
<b>Consequence</b>	4	5, 6, 8.2, 10.1, 10.2			
	3	7	4		
	2		8.1		
	1		1, 3	2, 9	
		1	2	3	4
		<b>Probability</b>			

Figure12.6: Risk overview after Qubes OS & countermeasures

Risk ID	Risk Description	Probability before	Consequence before	Score	Probability after	Consequence after	Score
1	Accidental VM deletion	3	3	9	2	1	2
2	Lost or Stolen laptop	3	4	12	3	1	3
3	IT-staff pushes faulty configuration code which removes functionality	4	2	8	2	1	2
4	SIEM software vulnerability is abused by threat	3	3	9	2	3	6
5	Fault in scripts opens a security vulnerability	2	4	8	1	4	4
6	Actor uses git repository to push malicious code	3	4	12	1	4	4
7	Actor uses git repository to pull and analyze information	4	3	12	1	3	3
8.1	User runs malware in an unsafe environment	3	2	6	2	2	4
8.2	User runs malware in an unsafe environment	2	4	8	1	4	4
9	Linux or Windows VM is infected with malware through social engineering	3	4	12	3	1	3
10.1	Laptop is infected with malware capable of VM escape	2	2	4	1	4	4
10.2	Laptop is infected with malware capable of VM escape	2	4	8	1	4	4

**Figure12.7:** List of color-coded risks before and after countermeasures.

## Chapter 13

# Evaluation

The purpose of this chapter is to evaluate our project based on the research questions set by KDA and described in Chapter 2. As most of our task consisted of figuring out these questions, an evaluation of the project will be focused on those answers.

The first research question is concerned with finding a hypervisor that can be setup in accordance with the zones of the second question. This thesis describes eight hypervisors and rules some of these out for unalterable reasons. Even though we decided to go further using Qubes OS it is made clear that some of these other alternatives could be of interest. Even though these alternatives have been found it must be stated that we did not attempt to configure or investigate the other alternatives.

The research question regarding collecting and sending logs to SIEM software has been answered with a script performing such a job across all the qubes on the platform.

The risk assessment is limited to a few scenarios, but has taken into consideration how using Qubes OS will affect the different risks. The accompanying illustrations should help the reader see the overall effects of Qubes OS.

We have given an evaluation of the compatibility for Qubes OS and our received laptop. Based upon what we experienced and what is documented online, we have given a recommendation for newer purchases. We could

We have successfully managed to join the Windows qubes to Kongsbergs IAM: service, Microsoft Azure Active Directory. We also created a proof of concept for networking and VPN tunnels.

We did not include banning impossible travel in our thesis. We find this option to be server side (which we do not have access to) on the IAM:, because the IAM: is logging login locations. We stopped developing this goal, as this can be solved by a policy change server-side.

The multifactor authentication was unsuccessful. Our multifactor device did not work and we were unsuccessful in proving multifactor authentication on our proof of concept platform.

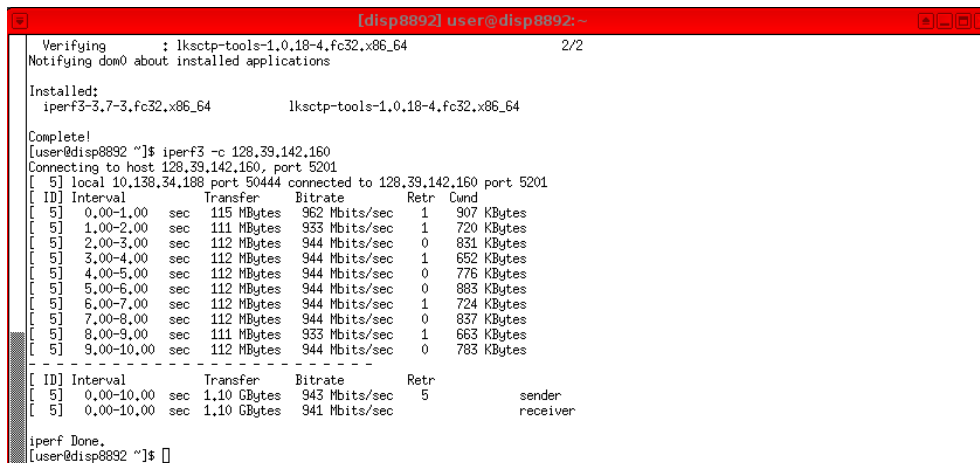


## 13.1 Client

With bi-weekly meetings and updates with KDA, we received continuous feedback on our progress and report. We have had two demonstrations for KDA, both of which were early in the project period. The first regarded using Qubes OS for teleconferencing in which we showcased that so far there is no easy way to share screens across qubes as was mentioned in Section 8. The other showcase the VPN regarding the VPNs. We have received positive response from KDA in our demonstrations and meetings.

## 13.2 Bandwidth

One of the ways we can evaluate the laptop is measuring bandwidth with iperf3. Measuring download speeds has it's limitations, as we are bottlenecked by the equipment at our disposal. Even still our figure is indicative of what our laptops are at least able to do. The images below show a connection to an iperf3 host first without, then through a VPN proxy. There is a significant decrease in bandwidth when sending through the proxies. Retransmission rates also increased with roughly 1600%.



```
[disp8892] user@disp8892: ~
Verifying          : lkstcp-tools-1.0.18-4.fc32.x86_64          2/2
Notifying dnf0 about installed applications

Installed:
iperf3-3.7-3.fc32.x86_64          lkstcp-tools-1.0.18-4.fc32.x86_64

Complete!
[user@disp8892 ~]$ iperf3 -c 128.39.142.160
Connecting to host 128.39.142.160, port 5201
[ 5] local 10.138.34.188 port 50444 connected to 128.39.142.160 port 5201
[ ID] Interval      Transfer      Bitrate      Retr  Cwnd
[ 5] 0.00-1.00    sec   115 MBytes  962 Mbits/sec  1    907 KBytes
[ 5] 1.00-2.00    sec   111 MBytes  933 Mbits/sec  1    720 KBytes
[ 5] 2.00-3.00    sec   112 MBytes  944 Mbits/sec  0    831 KBytes
[ 5] 3.00-4.00    sec   112 MBytes  944 Mbits/sec  1    652 KBytes
[ 5] 4.00-5.00    sec   112 MBytes  944 Mbits/sec  0    776 KBytes
[ 5] 5.00-6.00    sec   112 MBytes  944 Mbits/sec  0    883 KBytes
[ 5] 6.00-7.00    sec   112 MBytes  944 Mbits/sec  1    724 KBytes
[ 5] 7.00-8.00    sec   112 MBytes  944 Mbits/sec  0    837 KBytes
[ 5] 8.00-9.00    sec   111 MBytes  933 Mbits/sec  1    663 KBytes
[ 5] 9.00-10.00   sec   112 MBytes  944 Mbits/sec  0    783 KBytes
-----
[ ID] Interval      Transfer      Bitrate      Retr
[ 5] 0.00-10.00   sec   1.10 GBytes  943 Mbits/sec  5
[ 5] 0.00-10.00   sec   1.10 GBytes  941 Mbits/sec
iperf Done.
[user@disp8892 ~]$
```

Figure13.1: Data transfer through no VPN tunnel

```
[Analysis_deb] user@localhost: ~
File Edit View Search Terminal Help
user@localhost:~$ iperf3 -c 10.8.1.1
Connecting to host 10.8.1.1, port 5201
[ 5] local 10.137.0.23 port 53324 connected to 10.8.1.1 port 5201
[ ID] Interval          Transfer          Bitrate          Retr  Cwnd
[ 5]  0.00-1.00      sec  15.2 MBytes      127 Mbits/sec    5   99.5 KBytes
[ 5]  1.00-2.00      sec  16.5 MBytes      138 Mbits/sec   10   125 KBytes
[ 5]  2.00-3.00      sec  15.8 MBytes      132 Mbits/sec   13   102 KBytes
[ 5]  3.00-4.00      sec  15.8 MBytes      132 Mbits/sec    9   88.9 KBytes
[ 5]  4.00-5.00      sec  16.2 MBytes      136 Mbits/sec    7   119 KBytes
[ 5]  5.00-6.00      sec  15.7 MBytes      132 Mbits/sec   14   102 KBytes
[ 5]  6.00-7.00      sec  15.7 MBytes      131 Mbits/sec   11   87.5 KBytes
[ 5]  7.00-8.00      sec  15.9 MBytes      133 Mbits/sec    7   115 KBytes
[ 5]  8.00-9.00      sec  15.7 MBytes      132 Mbits/sec    5   101 KBytes
[ 5]  9.00-10.00     sec  15.9 MBytes      133 Mbits/sec    5   123 KBytes
-----
[ ID] Interval          Transfer          Bitrate          Retr
[ 5]  0.00-10.00     sec  158 MBytes      133 Mbits/sec    86
[ 5]  0.00-10.00     sec  158 MBytes      132 Mbits/sec

iperf Done.
user@localhost:~$
```

**Figure13.2:** Data transfer through VPN tunnel

# Chapter 14

## Discussion

In the pre-project phase, we underestimated the amount of work we would do on the report while doing research. On the GANTT the amount of time used on the main parts (research and testing) could be extended with the knowledge of also writing the thesis at the same time. We began work on the thesis much sooner than expected, e.g. setting up repositories and necessary tools, whilst we worked on the research tasks much longer than anticipated. We should have found a balance between milestones and more agile development, and we experienced our GANTT-diagram in the later stages of the project to be too stiff for the actual work that needs to be done.

We can not guarantee all our sources are correct, considering we are using much Qubes OS online documentation and a Qubes OS product. Testing every of their claim is a hard and tedious problem, and we would mostly be incapable of doing so due to lack of knowledge. However, we have been critically to our sources and rather used official documentation from the Qubes OS team rather than using reader submissions, e.g. from StackOverflow, Reddit or Medium articles.

We had little luck with e-mails, and instead of waiting for responses that never came, we realise in hindsight we should rather have queried the community. While also having a better chance of giving us a response, the answers may also represent less bias towards a solution or product. Some of the communities we could have contacted could be the Qubes subreddit, QubesOS twitter or StackOverflow.

To keep this thesis not classified, we opted, with consent from KDA, to not ask for KDA assets or internal risk assessments. Therefore, all assets, threats and abuse cases are made after our own best expertise. Some of them may not be relevant to KDA, but rather be more suited for general companies. The risk assessment should be taken with a grain of salt, and evaluated before being used.

We experienced several times leads that lead to dead ends. Much of the documentation and relevant research paper is old, and contains unrelated or outdated work or not the necessary information for our needs. Often times we realized this too late, and had wasted a large amount of time on unnecessary research. To try to prevent this, we started becoming more aware of the version we used and the version described in the research.

Using LaTeX [121], we met some easy problems early in the thesis stage. The problems were easy to fix with Internet resources. However, as the project became bigger and more chapters, texts and images were added, we required LaTeX to do more when compiling. Later in the project we experienced more and more problems compiling, only to be solved by compiling again. At the end, the re-compile count was up to 6 times each time we generated the PDF. At points we had to use more time figuring out how to generate the thesis rather than actually writing it. During the project period we tried to find other, better and more efficient document compilers, but we found nothing that would yield better result than we already had at the time. Next time we would like to research better writing tools before we start writing thesis.

## 14.1 Related Work

While there has been previous work on Qubes OS, we found no attempts to use Qubes OS as a platform for security analysts. We did find work for using Qubes OS as a secure platform for more sensitive work, such as SecureDrop Workstation [122]. SecureDrop Workstation is a project aimed at journalists to more easily upload important and vulnerable documents to a SecureDrop server. SecureDrop Workstation is an add-on to Qubes OS which creates several qubes in different domains to best secure the transaction of the journalists work between SecureDrop Workstation and the SecureDrop server.

An existing approach to what we are trying to achieve could be done in certain type 2 hypervisors, such as VMWare. VMWare allows the user to enter “unity mode” between the host and the virtual machine [123]. This allows the applications in the virtual machine to appear on the hosts desktop, much alike the seamless mode in Qubes OS. Contrary to Qubes OS, this will only work with Windows XP and later, including Windows 10, but not Linux.

# Chapter 15

## Development Process

This chapters covers the development process used for the platform, explaining the development model used, and the choices made during design and development of the platform.

### 15.1 Development model

The project group used a hybrid Scrum and Kanban approach [124] to structure and delegate work between group members. Daily and weekly scrum meeting were held intermittently to organize the work that needed to be done in a given day or week, and to check up on the status of the work assigned thus far. As the project moved on, the group members largely felt they had good overview of the tasks assigned and their responsibilities within the project, and as such, daily and weekly meetings were not always necessary.

As the project largely consists of research and implementation of existing products, rather than development of new tools and programs, no specific development plan was made for scripts and programs required for the project. Programs needed to fulfill specific requirements or purposes within a scope of work already delegated to a specific person, such as Appendix A.1, were developed by the group member in question on an ad-hoc basis.

### 15.2 Documentation

To organize our work, guidelines have been defined to organize our report and documentation.

#### 15.2.1 Report writing

The report was structured using Sphinx, which is a tool designed to make it easy to write documentation, originally created for documenting Python [125]. We used the markup language reStructuredText, which is the default for Sphinx [125], designed for extensibility and made to be easy to read [126]. Custom extensions to Sphinx were written to facilitate some features required for our thesis, such as the ability to embed a glossary, vector graphics, and more. The thesis was then compiled to PDF via Sphinx using L<sup>A</sup>T<sub>E</sub>X as an intermediary compiler. While the group recognizes T<sub>E</sub>X as one of the most widely used markup formats in academia [121], reStructuredText was chosen instead due to it being easier to understand and work with.

The report itself was hosted on Bitbucket, and managed via Git, which is a version control system used to track changes to source code, often used in software development [127]. As the report is written in reStructuredText, a plain-text format, Git is a suitable tool to manage changes and updates to the thesis. All source code for scripts and programs developed as part of the project are integrated into the Sphinx directory structure in Git, making it easier to integrate the code into the report as needed.

## 15.2.2 Meeting minutes

During every meeting, notes were taken to summarize what had been discussed. This included meetings internally in the project group, as well as meetings with our supervisor and the client. The meeting minutes have been included in Appendix G.

## 15.2.3 Time management

To facilitate time management, the group documented the time spent performing different tasks. The time and hours logged were primarily used to personally check one's own performance against the expectations set by the group, to see if the workload assigned and performed is comparable with, and in line with those of the other group members.

The figure below is a typical work week. The star represents days where we were not working together, but were rather encouraged to do individual work. This is due to other obligations each team member had, such as part-time jobs.

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
*	*	5-8 h	5-8 h	5-8 h		

**Figure15.1:** An average work week per person.

We averaged around 30 hours per week. Appendix C.1 is an example of individual hours in a selection of days.

## 15.3 Routines

The group established various routines to coordinate group work. These routines were primarily intended for use internally within the group for work on the thesis.

### 15.3.1 Work policy

As part of the pre-project plan, the group developed a policy governing the work and attendance requirements for each group member. This policy is included in the pre-project plan.

## COVID-19

While working on this thesis, the world is still under a pandemic and the group had to take certain measures to prevent spread of the COVID-19 virus. The group prioritized working physically together, as we found it gained us more quality work and discussions. At times, due to the pandemic, the NTNU Gjøvik campus was closed and the group was forced to work from home, for example during a period of high infection rates in Norway after Easter.

All communications to actors outside of the group, to KDA or our guidance councillor, had to be completed online. This limited our ability to present our project and receive feedback, e.g. Qubes OS does not have an easy solution to share its desktop screen over the Internet due to security concerns.

### Failed Communication Attempts with the Hypervisor and Qubes OS community

During the project period we have sent mails to Qubes OS projects and potential projects like Qubes OS. We sent out four mails, three to potential Qubes OS replacements and one mail to “2020 Google Summer of Code” to query about audio support for Windows qubes. The three potential Qubes OS like projects and hypervisors we queried for more information are PolyXene, Titanium Secure and AIS SecureView. We only received answer to one mail, AIS SecureView. When we tried to further enquire about their product our mail got rejected from them, and further communication was not possible.

Due to the lack of response, we had no other solution than choosing Qubes OS. As touched we touched on in Chapter 4, a lot of the similar projects to Qubes OS were lacking online documentation and information, and some of them were strictly for only US governmental organs (such as AIS SecureView). If we had received documentation or useful information that may have swayed our recommendation, we might have chosen a different hypervisor and the final platform could be based on another product.

### 15.3.2 Tools

Various tools were agreed upon to be used by the project group. These are outlined below.

#### Communication

Internal group communication was facilitated using text and voice chats on the free Voice over Internet Protocol (VoIP) application Discord [128]. All communications with the project supervisor, as well as the client, were facilitated with Microsoft Teams [129]. The group has prioritized attending group meetings on campus for internal meetings where possible, following all relevant guidelines regarding the covid-19 pandemic.

#### Report

The report was developed using the text editor of choice for each group member, primarily Atom [130] or Visual Studio Code [131]. Collaboration was facilitated via usage of Git [127], and rendering and report compilation facilitated via Sphinx [125]. The report and all of its contents were hosted on Bitbucket [132].

# Chapter 16

## Closing Remarks

This chapter includes our learning outcomes, a conclusion to our work and what we think can be worked on in the future regarding the multipurpose platform.

### 16.1 Learning outcome

During this project, we have used many of the things we have learnt during our course, IT-operations and Cybersecurity, and many new things. We have learnt much more about operating systems and how they function, hypervisors, SaltStack, SIEM and IAM and much more highlighted through our thesis.

Besides what we have talked about in this thesis, we have learnt much how to complete a project from start to finish. First, we all have learnt how to best work in a group and what roles we take naturally. This includes what role some of us had to slip into to better the group. Secondly, we have learnt much from communicating with guidance councillor and clients, as well as the challenge COVID-19 provides by limiting our interactions to online. As we have touched on earlier, our meetings could only be facilitated over the Internet, which proved it hard to demo our platform. We needed to be creative to demonstrate the platform to the client.

### 16.2 Conclusion

After half a year of intense work, we are proud to present our final product, Multipurpose platform for Security Analysts. Our work presents the possibility of using Qubes OS as a stable platform in a professional environment. We hope our work will be used to further this product, and even be used by KDA to better Kongsberg Gruppens defensive capabilities.

Throughout this project we have discussed and reflected on the research questions given to us by KDA. It has been a good learning experience and fun to use our knowledge in new and creative ways to produce a final product.



## 16.3 Future work

We hope this thesis and our work can be used to create a multipurpose platform for security analysts and be used for real life scenarios. We hope the platform can help reduce security concerns against their respective companies, and help analysts analyze malware to help common users, workers and companies against attack.

If we had more time and resources, we would have liked to create this platform and tested in a real life scenario. After judging its effectiveness, we would have liked to scale up the test, by introducing more platforms and users. At each step we would have liked to survey the users about their experiences and used the result to improve the platform from a user perspective as well. Only then could we definitively recommend or not recommend this platform to KDA.

Our scripts are not perfect, and they could have much more potential. As this is a proof of concept, we focused more on showing that it can be done and how. If we had more time and resources, we could expand the scripts with more functionality. The scripts could potentially work together in sequence to require the least amount of human input possible.

Not every use/abuse case, asset or threat we have brought forth is relevant in every scenario. If someone were to adopt this project, a new risk assessment would have to be done adapted to their own scenario.

Since Qubes OS is open source, we would like to functionality we found lacking or missing. For example, we would like to add better integration with Qubes OS and interfaces to third party services, like a SIEM, IAM or configuration manager.

Continuing the fact that Qubes OS is still being updated, it should be taken into consideration development for Qubes OS can stop at any minute. It is open source and Invisible Things Lab has no commitment for updating other than initiative. Any other open-source project or hypervisor similar to Qubes OS can also present itself in the future. If either of these problems were to appear, we would recommend evaluating moving the platform to the better solution.

Furthermore, we have mentioned the work on Qubes Air in Chapter 11. We would recommend looking out for this project in the future, and evaluate the platform and move to the better solution.

There were many avenues we never took in this project. One of the biggest is the fact that we never tested any other hypervisor than Qubes OS and doing so might reveal that some of those are possible other candidates. Configuration management is another area were someone could try something different, possibly with the

# Bibliography

- [1] Qubes OS, *Glossary of qubes terminology*, [Online; accessed 8-April-2021]. [Online]. Available: <https://www.qubes-os.org/doc/glossary/>.
- [2] B. Preneel, “Cryptographic hash functions,” *European Transactions on Telecommunications*, vol. 5, no. 4, pp. 431–448, 1994. doi: 10.1002/ett.4460050406.
- [3] R. Moir, *Defining malware: FAQ*, Apr. 2009. [Online]. Available: [https://docs.microsoft.com/en-us/previous-versions/tn-archive/dd632948\(v=technet.10\)](https://docs.microsoft.com/en-us/previous-versions/tn-archive/dd632948(v=technet.10)).
- [4] M. Sikorski and A. Honig, *Practical malware analysis the hands-on guide to dissecting malicious software*. No Starch Press, 2012.
- [5] Kongsberg Group, *Kongsberg*, [Online; accessed 08-April-2021], Mar. 2021. [Online]. Available: <https://www.kongsberg.com/>.
- [6] Wikipedia contributors, *Kongsberg gruppen — Wikipedia, the free encyclopedia*, [Online; accessed 19-February-2021], 2021. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Kongsberg\\_Gruppen&oldid=1007511827](https://en.wikipedia.org/w/index.php?title=Kongsberg_Gruppen&oldid=1007511827).
- [7] Kongsberg Group, *Kongsberg defence and aerospace*, [Online; accessed 08-April-2021], Mar. 2021. [Online]. Available: <https://www.kongsberg.com/kda/>.
- [8] Kongsberg Group, *About us - kda*, [Online; accessed 19-February-2021]. [Online]. Available: <https://www.kongsberg.com/kda/about-us/>.
- [9] S. Bhatt, P. K. Manadhata, and L. Zomlot, “The operational role of security information and event management systems,” *IEEE security & Privacy*, vol. 12, no. 5, pp. 35–41, 2014. doi: 10.1109/MSP.2014.103. [Online]. Available: <https://doi.org/10.1109/MSP.2014.103>.
- [10] S. Kouti and M. Seitsonen, “Active directory: The big picture,” in *Inside Active Directory: A System Administrator’s Guide*, 2nd ed., ser. Microsoft Windows Server System Series. Addison-Wesley, 2005, pp. 4–7.
- [11] *Salt project*, [Online; accessed 13-May-2021]. [Online]. Available: <https://saltproject.io/>.
- [12] *Make infrastructure actionable, scalable and intelligent*. [Online; accessed 13-May-2021]. [Online]. Available: <https://puppet.com/>.
- [13] *Ansible is simple it automation*, [Online; accessed 13-May-2021]. [Online]. Available: <https://www.ansible.com/>.
- [14] M. J. Van Eeten and J. M. Bauer, “Economics of malware: Security decisions, incentives and externalities,” *OECD Science, Technology and Industry Working Papers*, 2008. doi: 10.1787/18151965. [Online]. Available: <https://doi.org/10.1787/18151965>.
- [15] E. Cole, “The changing threat,” in *Advanced Persistent Threat*, Elsevier, 2013, pp. 3–26. doi: 10.1016/b978-1-59-749949-1.00001-2. [Online]. Available: <https://doi.org/10.1016/b978-1-59-749949-1.00001-2>.

- [16] F. Li, A. Lai, and D. Ddl, "Evidence of advanced persistent threat: A case study of malware for political espionage," in *2011 6th International Conference on Malicious and Unwanted Software*, IEEE, Oct. 2011. doi: 10.1109/malware.2011.6112333. [Online]. Available: <https://doi.org/10.1109/malware.2011.6112333>.
- [17] Wikipedia contributors, *Wannacry ransomware attack — Wikipedia, the free encyclopedia*, [Online; accessed 8-April-2021], 2021. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=WannaCry\\_ransomware\\_attack&oldid=1016481123](https://en.wikipedia.org/w/index.php?title=WannaCry_ransomware_attack&oldid=1016481123).
- [18] J. M. Ehrenfeld, "WannaCry, cybersecurity and health information technology: A time to act," *Journal of Medical Systems*, vol. 41, no. 7, May 2017. doi: 10.1007/s10916-017-0752-1. [Online]. Available: <https://doi.org/10.1007/s10916-017-0752-1>.
- [19] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," *ACM Computing Surveys*, vol. 44, no. 2, pp. 1–42, Feb. 2012. doi: 10.1145/2089125.2089126. [Online]. Available: <https://doi.org/10.1145/2089125.2089126>.
- [20] J. E. Smith and R. Nair, "The architecture of virtual machines," *Computer*, vol. 38, no. 5, pp. 32–38, 2005.
- [21] Red Hat, *Understanding virtualization*, [Online; accessed 18-February-2021]. [Online]. Available: <https://www.redhat.com/en/topics/virtualization>.
- [22] Open Source @ Red Hat, *What is virtualization?* [Online; accessed 18-February-2021]. [Online]. Available: <https://opensource.com/resources/virtualization>.
- [23] Red Hat, *What is a virtual machine (vm)?* [Online; accessed 19-February-2021]. [Online]. Available: <https://www.redhat.com/en/topics/virtualization/what-is-a-virtual-machine>.
- [24] W. Commons, *File:hyperviseur.svg — wikimedia commons, the free media repository*, [Online; accessed 20-May-2021], 2021. [Online]. Available: <https://commons.wikimedia.org/w/index.php?title=File:Hyperviseur.svg&oldid=556375952>.
- [25] Xen Project, *Archive/xcp faq dynamic memory control*, [Online; accessed 19-February-2021], Nov. 2011. [Online]. Available: [https://wiki.xenproject.org/index.php?title=Archive%2FXCP\\_FAQ\\_Dynamic\\_Memory\\_Control&oldid=12076](https://wiki.xenproject.org/index.php?title=Archive%2FXCP_FAQ_Dynamic_Memory_Control&oldid=12076).
- [26] VMWare, *Dynamic storage provisioning*, Nov. 2009. [Online]. Available: <https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/VMware-DynamicStorageProv-WP-EN.pdf>.
- [27] *What is configuration management?* [Online; accessed 13-May-2021]. [Online]. Available: <https://www.redhat.com/en/topics/automation/what-is-configuration-management>.
- [28] B. I. C. Education, *Hypervisors*, [Online; accessed 11-February-2021]. [Online]. Available: <https://www.ibm.com/cloud/learn/hypervisors>.
- [29] Xen Project, *About us*, [Online; accessed 11-February-2021], Jun. 2019. [Online]. Available: <https://xenproject.org/about-us/>.
- [30] Star Lab Software, *Whitepaper*, [Online; accessed 12-February-2021]. [Online]. Available: <https://static1.squarespace.com/static/5e1f51eb1bb1681137ea90b8/t/5e8360669d266d6c8c086f47/1585668199694/Star+Lab+Whitepaper+-+Tactical+Virtualization.pdf>.
- [31] Star Lab Software, *Star lab website*, [Online; accessed 12-February-2021]. [Online]. Available: <https://www.starlab.io/titanium-secure-hypervisor>.
- [32] Star Lab Software, *About starlab*, [Online; accessed 12-February-2021]. [Online]. Available: <https://www.starlab.io/about>.
- [33] The National Information Assurance Partnership, *Niap*, [Online; accessed 12-February-2021]. [Online]. Available: <https://www.niap-ccevs.org/>.

- [34] Wind River, *Data sheet*, [Online; accessed 12-February-2021]. [Online]. Available: <https://static1.squarespace.com/static/5e1f51eb1bb1681137ea90b8/t/5fa0811b24daf529fb6e2b46/1604354332829/Star+Lab+Data+Sheet+-+Titanium+Secure+Hypervisor.pdf>.
- [35] AIS, *Secureview infosheet*, [Online; accessed 18-February-2021]. [Online]. Available: [https://n5vm24dpyjj1jvzaq12us84e-wpengine.netdna-ssl.com/wp-content/uploads/2020/01/AIS-SECUREVIEW\\_infosheet.pdf](https://n5vm24dpyjj1jvzaq12us84e-wpengine.netdna-ssl.com/wp-content/uploads/2020/01/AIS-SECUREVIEW_infosheet.pdf).
- [36] AIS, *Ais secure view*, [Online; accessed 18-February-2021]. [Online]. Available: <https://www.ainfosec.com/technologies/secureview/>.
- [37] AIS, *Secureview overview*, [Online; accessed 18-February-2021]. [Online]. Available: <https://www.ainfosec.com/wp-content/uploads/2015/09/AIS-SecureView-Overview.pdf>.
- [38] HP, *Sure click enterprise website*, [Online; accessed 12-February-2021]. [Online]. Available: <https://www8.hp.com/us/en/solutions/sure-click-enterprise.html>.
- [39] HP/Bromium, *Sure click whitepaper*, [Online; accessed 12-February-2021]. [Online]. Available: <https://h20195.www2.hp.com/v2/getpdf.aspx/4AA7-7517ENW>.
- [40] HP/Bromium, *Sure click solution brief*, [Online; accessed 12-February-2021]. [Online]. Available: <https://h20195.www2.hp.com/v2/getpdf.aspx/4AA7-7470ENUS.pdf>.
- [41] Red Hat, *Secure virtualization with svirt*, [Online; accessed 03-February-2021], 2016. [Online]. Available: <https://www.redhat.com/en/resources/secure-virtualization-with-svirt>.
- [42] *Isolate Your Workspace - Session 2*. YouTube, Jan. 2021, [Online; accessed 04-February-2021]. [Online]. Available: <https://www.youtube.com/watch?v=TKq2XsvNIIA>.
- [43] *Making virtual desktops work with azure ad and intune*, [Online; accessed 12-May-2021], May 2021. [Online]. Available: <https://www.hysolate.com/blog/making-virtual-desktops-work-with-azure-ad-and-intune/>.
- [44] *Frequently asked questions*, [Online; accessed 12-May-2021], May 2021. [Online]. Available: <https://www.hysolate.com/faq/>.
- [45] *Isolate risky and sensitive activities with an isolated workspace*, [Online; accessed 12-May-2021], May 2021. [Online]. Available: <https://www.hysolate.com/>.
- [46] *Virtualization for windows 10: A practical guide*, [Online; accessed 12-May-2021], May 2021. [Online]. Available: <https://www.hysolate.com/learn/vdi-windows/virtualization-for-windows-10-a-practical-guide/>.
- [47] *Isolate Your Workspace - Session 1*. YouTube, Jan. 2021, [Online; accessed 04-February-2021]. [Online]. Available: <https://www.youtube.com/watch?v=mY64b3kQN0A>.
- [48] Bertin IT, *Polyxene®: High security software platform for critical infrastructure & sensitive information system*, [Online; accessed 05-February-2021], Mar. 2017. [Online]. Available: <https://www.bertin-it.com/en/cybersecurity/polyxene-high-security-software-platform-for-sensitive-information-systems-critical-infrastructure/>.
- [49] Bertin IT, *The defense in-depth of information systems & critical infrastructures*, May 2016. [Online]. Available: <https://www.bertin-it.com/brochure/PolyXene-high-security-hypervisor.pdf> (visited on 02/05/2021).
- [50] Common Criteria for Information Technology Security Evaluation, *Common criteria*, Apr. 2017. [Online]. Available: <https://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf> (visited on 02/05/2021).
- [51] Secrétariat général de la défense nationale, *Certificat dcssi-2009/18*, Jul. 2009. [Online]. Available: [https://www.ssi.gouv.fr/uploads/IMG/certificat/dcssi\\_2009-18fr.pdf](https://www.ssi.gouv.fr/uploads/IMG/certificat/dcssi_2009-18fr.pdf) (visited on 02/05/2021).

- [52] N. A. Nordbotten, F. Mancini, B. H. Farsund, R. Haakseth, A. M. Hegland, and F. Lillevold, *Information sharing across security domains*, Aug. 2015. [Online]. Available: <http://rapporter.ffi.no/rapporter/2015/00456.pdf> (visited on 02/05/2021).
- [53] Qubes OS, *Introduction*, [Online; accessed 22-April-2021]. [Online]. Available: <https://www.qubes-os.org/intro/>.
- [54] *Community information file: Service terms*, [Online; accessed 11-May-2021], May 2021. [Online]. Available: <https://aws.amazon.com/service-terms/>.
- [55] M. McKittrick, *Hosting malware on azure virtual machine*, [Online; accessed 10-February-2021], Apr. 2019. [Online]. Available: <https://social.msdn.microsoft.com/Forums/en-US/c3c360ca-4d28-4cac-83ad-17d2397bb04d/hosting-malware-on-azure-virtual-machine?forum=WAVirtualMachinesforWindows>.
- [56] Antsle, *Your private cyberlab*, [Online; accessed 10-February-2021], Jul. 2020. [Online]. Available: <https://antsle.com/solutions/cybersecurity/>.
- [57] P. Ferguson and G. Huston, *What is a vpn?* 1998.
- [58] E. Obrestad, *Openstack at ntnu*, [Online; accessed 15-May-2021], Dec. 2020. [Online]. Available: <https://www.ntnu.no/wiki/display/skyhigh/>.
- [59] J. Cepek, *Authentication*, [Online; accessed 15-May-2021], Jan. 2014. [Online]. Available: <https://community.openvpn.net/openvpn/wiki/Concepts-Authentication>.
- [60] *Pivpn*, [Online; accessed 15-May-2021]. [Online]. Available: <https://pivpn.io/>.
- [61] J. Rutkowska and R. Wojtczuk, *Qubes os architecture*, Jan. 2010. [Online]. Available: <https://www.qubes-os.org/attachment/wiki/QubesArchitecture/arch-spec-0.3.pdf>.
- [62] A. D. Wong, *How to make a vpn gateway in qubes*, [Online; accessed 29-April-2021], Dec. 2020. [Online]. Available: <https://github.com/Qubes-Community/Contents/blob/2230dc0aada25da2df102b3759732feadb1b8a7f/docs/configuration/vpn.md>.
- [63] *The data-to-everything™ platform, powering security, it and devops*, [Online; accessed 15-May-2021]. [Online]. Available: <https://www.splunk.com/>.
- [64] A. D. Wong, *Installing a windows vm*, [Online; accessed 24-March-2021], Dec. 2020. [Online]. Available: <https://github.com/Qubes-Community/Contents/blob/fc841d00c50f6751ea746522201f49da34086520/docs/os/windows/windows-vm.md>.
- [65] E. Killick and B. Hoar, *Installing a windows vm*, [Online; accessed 24-March-2021]. [Online]. Available: <https://github.com/elliottkillick/qvm-create-windows-qube>.
- [66] E. Killick, *Spyless.bat*, [Online; accessed 23-April-2021], Apr. 2021. [Online]. Available: <https://github.com/elliottkillick/qvm-create-windows-qube/blob/d05103494ad8b13ffbcceb0e261e6276f0ff3a73/post/spyless.bat>.
- [67] E. Killick, *Download-windows.sh*, [Online; accessed 23-April-2021], Apr. 2021. [Online]. Available: <https://github.com/elliottkillick/qvm-create-windows-qube/blob/d05103494ad8b13ffbcceb0e261e6276f0ff3a73/windows-media/isos/download-windows.sh>.
- [68] A. D. Wong and G. Weck, *Qubes windows tools*, [Online; accessed 18-March-2021], Feb. 2021. [Online]. Available: <https://github.com/Qubes-Community/Contents/blob/2230dc0aada25da2df102b3759732feadb1b8a7f/docs/os/windows/windows-tools.md>.
- [69] J. Rutkowska, *Introducing the qubes admin api*, [Online; accessed 28-April-2021], Jun. 2017. [Online]. Available: <https://www.qubes-os.org/news/2017/06/27/qubes-admin-api/>.
- [70] C. Heckmann and C. Heckmann, *Capture cards explained - types, systems and set-ups*, [Online; accessed 6-May-2021], Mar. 2021. [Online]. Available: <https://www.studiobinder.com/blog/what-is-a-capture-card-for-streaming/>.

- [71] D. Strom, *What is iam? identity and access management explained*, [Online; accessed 12-May-2021], Apr. 2021. [Online]. Available: <https://www.csoonline.com/article/2120384/what-is-iam-identity-and-access-management-explained.html>.
- [72] J. Flores, J. Martinez, M. Turscak, and T. Pratt, *Azure ad joined devices*, [Online; accessed 12-May-2021], Jul. 2020. [Online]. Available: <https://docs.microsoft.com/en-us/azure/active-directory/devices/concept-azure-ad-join>.
- [73] *Freeipa home*, [Online; accessed 26-Mar-2021]. [Online]. Available: [https://www.freeipa.org/page/Main\\_Page](https://www.freeipa.org/page/Main_Page).
- [74] *Azure active directory*, [Online; accessed 12-May-2021]. [Online]. Available: <https://azure.microsoft.com/en-us/services/active-directory/>.
- [75] *Azure active directory pricing*, [Online; accessed 12-May-2021]. [Online]. Available: <https://azure.microsoft.com/en-us/pricing/details/active-directory/>.
- [76] K. Withee, *Remote access to on-premises applications through azure ad application proxy*, [Online; accessed 12-May-2021], Apr. 2021. [Online]. Available: <https://docs.microsoft.com/en-us/azure/active-directory/app-proxy/application-proxy>.
- [77] I. Githinji, K. Withee, D. Coulter, R. Lyon, M. Martin, M. Browne, J. Flores, C. de Guzman, K. Sharkey, D. Bahall, B. Kess, and E. Ross, *Get started integrating azure active directory with apps*, [Online; accessed 12-May-2021], May 2021. [Online]. Available: <https://docs.microsoft.com/en-us/azure/active-directory/manage-apps/plan-an-application-integration>.
- [78] A. Burnley, M. Wahl, C. Love, D. Coulter, B. Neira, M. Martin, R. Lyon, J. Flores, and K. Sharkey, [Online; accessed 12-May-2021], Jun. 2020. [Online]. Available: <https://docs.microsoft.com/en-us/azure/active-directory/governance/identity-governance-overview>.
- [79] M. Vilcinskis, D. Coulter, S. Shailaj, J. Flores, C. Adams, K. Sharkey, D. Bahall, and P. Mohanram, *What are azure active directory reports?* [Online; accessed 12-May-2021], Sep. 2020. [Online]. Available: <https://docs.microsoft.com/en-us/azure/active-directory/reports-monitoring/overview-reports>.
- [80] M. Vilcinskis, B. Wren, B. W., D. Coulter, M. Turscak, C. Adams, M. Sebolt, K. Sharkey, D. Bahall, and P. Mohanram, *What is azure active directory monitoring?* [Online; accessed 12-May-2021], Apr. 2019. [Online]. Available: <https://docs.microsoft.com/en-us/azure/active-directory/reports-monitoring/overview-monitoring>.
- [81] A. Burnley, A. Buck, R. Lyon, T. Myers, K. Withee, J. Townsend, C. Love, J. Martinez, D. Coulter, D. Arora, K. Sharkey, D. Murray, J. Flores, M. Turscak, dmc, V. Suravarapu, K. Bullen, E. Ross, T. Sanders, D. Bahall, C. de Guzman, M. Garg, B. Mathers, and B. Kess, *What is azure active directory?* [Online; accessed 12-May-2021], Jun. 2020. [Online]. Available: <https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-whatis>.
- [82] *What is azure?* [Online; accessed 12-May-2021]. [Online]. Available: <https://azure.microsoft.com/en-us/overview/what-is-azure/>.
- [83] *Anomaly detection policy*, [Online; accessed 14-Mar-2021]. [Online]. Available: <https://docs.microsoft.com/en-us/cloud-app-security/anomaly-detection-policy>.
- [84] *Ibm device management*, [Online; accessed 14-May-2021]. [Online]. Available: <https://www.ibm.com/docs/en/maas360>.
- [85] *Google device management*, [Online; accessed 14-May-2021]. [Online]. Available: [https://support.google.com/cloudidentity/answer/7582673?hl=en&ref\\_topic=7558359](https://support.google.com/cloudidentity/answer/7582673?hl=en&ref_topic=7558359).
- [86] *Aws directory service*, [Online; accessed 14-May-2021]. [Online]. Available: <https://aws.amazon.com/directoryservice/>.

- [87] *Aws single sign on*, [Online; accessed 14-May-2021]. [Online]. Available: <https://aws.amazon.com/single-sign-on/>.
- [88] *Freeipa about*, [Online; accessed 26-Mar-2021]. [Online]. Available: <https://www.freeipa.org/page/About>.
- [89] *Try windows 10 enterprise on microsoft evaluation center*, [Online; accessed 12-May-2021]. [Online]. Available: <https://www.microsoft.com/en-us/evalcenter/evaluate-windows-10-enterprise>.
- [90] C. Love, J. Martinez, D. Coulter, D. Bahall, E. Ross, N. Schonning, and K. Sharkey, *Join your work device to your organization's network*, [Online; accessed 12-May-2021], Aug. 2018. [Online]. Available: <https://docs.microsoft.com/en-us/azure/active-directory/user-help/user-help-join-device-on-network>.
- [91] L. Chen and L. Zou, *Troubleshoot windows device enrollment problems in microsoft intune*, [Online; accessed 12-May-2021], Dec. 2020. [Online]. Available: <https://docs.microsoft.com/en-us/troubleshoot/mem/intune/troubleshoot-windows-enrollment-errors>.
- [92] T. Lin and Z. Tu, *Azure active directory joined computers experience a three hours delay during boot*, [Online; accessed 12-May-2021], Jul. 2020. [Online]. Available: <https://docs.microsoft.com/en-us/troubleshoot/azure/active-directory/computers-3-hour-delay-boot>.
- [93] *Using yubikey to qubes authentication*, [Online; accessed 19-May-2021]. [Online]. Available: <https://www.qubes-os.org/doc/yubi-key/>.
- [94] *Onlykey hardware password manager*, [Online; accessed 19-May-2021]. [Online]. Available: <https://onlykey.io/>.
- [95] Wikipedia contributors, *Comparison of open-source configuration management software — Wikipedia, the free encyclopedia*, [Online; accessed 20-May-2021], 2021. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Comparison\\_of\\_open-source\\_configuration\\_management\\_software&oldid=1021999953](https://en.wikipedia.org/w/index.php?title=Comparison_of_open-source_configuration_management_software&oldid=1021999953).
- [96] *Management infrastructure*, [Online; accessed 19-April-2021]. [Online]. Available: <https://www.qubes-os.org/doc/salt/>.
- [97] Rudd-O, *Qubes os devops automation toolkit*, [Online; accessed 20-May-2021]. [Online]. Available: <https://github.com/Rudd-O/ansible-qubes>.
- [98] *Anacron(8) linux user's manual*.
- [99] *Skyplabs script*, [Online; accessed 13-Apr-2021]. [Online]. Available: <https://github.com/SkypLabs/my-qubes-os-formula/blob/master/copy-from-vm-to-dom0.sh>.
- [100] *Hardware compatibility list (hcl)*, [Online; accessed 07-April-2021]. [Online]. Available: <https://www.qubes-os.org/hcl/>.
- [101] *System requirements*, [Online; accessed 16-April-2021]. [Online]. Available: <https://www.qubes-os.org/doc/system-requirements/>.
- [102] A. D. Wong, *System tray icons disappearing*, [Online; accessed 14-April-2021]. [Online]. Available: <https://github.com/QubesOS/qubes-issues/issues/2242>.
- [103] *What is bring your own device (byod)?* [Online; accessed 22-April-2021]. [Online]. Available: <https://www.ibm.com/services/digital-workplace/byod>.
- [104] *Cpu benchmarks*, [Online; accessed 14-April-2021]. [Online]. Available: [https://www.cpubenchmark.net/cpu\\_list.php](https://www.cpubenchmark.net/cpu_list.php).
- [105] J. Rutkowska, *Qubes air: Generalizing the qubes architecture*, [Online; accessed 15-April-2021], Jan. 2018. [Online]. Available: <https://www.qubes-os.org/news/2018/01/22/qubes-air/>.

- [106] M. Marczykowski-Górecki and M. Marczykowska-Górecka, *Qubes architecture next steps: The gui domain*, [Online; accessed 15-April-2021], Mar. 2020. [Online]. Available: <https://www.qubes-os.org/news/2020/03/18/gui-domain/>.
- [107] *Xen security advisory (xsa) tracker*, [Online; accessed 5-May-2021]. [Online]. Available: <https://www.qubes-os.org/security/xsa/>.
- [108] *Nsm veiledere i sikkerhetsstyring*, [Online; accessed 05-May-2021]. [Online]. Available: <https://nsm.no/getfile.php/132933-1591350417/Demo/Dokumenter/Veiledere/veiledere-i-sikkerhetsstyring.pdf>.
- [109] *Iso/iec 27001*, [Online; accessed 13-Apr-2021]. [Online]. Available: <https://www.iso.org/isoiec-27001-information-security.html>.
- [110] *Nsm grunprinsipper for ikt-sikkerhet*, [Online; accessed 05-May-2021]. [Online]. Available: <https://nsm.no/getfile.php/133735-1592917067/Demo/Dokumenter/Veiledere/nsm-grunprinsipper-for-ikt-sikkerhet-v2.0.pdf>.
- [111] *Iso/iec 27002*, [Online; accessed 13-Apr-2021]. [Online]. Available: <https://www.iso.org/standard/54533.html>.
- [112] *Nsm website*, [Online; accessed 05-May-2021]. [Online]. Available: <https://nsm.no/om-oss/dette-er-nsm/>.
- [113] *Glossary*, Aug. 2009. [Online]. Available: <https://web.archive.org/web/20120229151151/http://www.enisa.europa.eu/act/rm/cr/risk-management-inventory/glossary/#G3>.
- [114] *8 common cyber attack vectors and how to avoid them*, [Online; accessed 6-May-2021], Feb. 2021. [Online]. Available: <https://www.balbix.com/insights/attack-vectors-and-breach-methods/>.
- [115] *What is ci/cd? continuous integration and continuous delivery explained*, [Online; accessed 20-May-2021]. [Online]. Available: <https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html>.
- [116] KOF2002, *Splunk < 7.0.1 - information disclosure*, [Online; accessed 5-May-2021], Jun. 2018. [Online]. Available: <https://www.exploit-db.com/exploits/44865>.
- [117] *Splunk : Security vulnerabilities*, [Online; accessed 20-May-2021]. [Online]. Available: [https://www.cvedetails.com/vulnerability-list/vendor\\_id-10963/Splunk.html](https://www.cvedetails.com/vulnerability-list/vendor_id-10963/Splunk.html).
- [118] *Cyberangrep på hydro*, [Online; accessed 17-May-2021]. [Online]. Available: <https://www.hydro.com/no-NO/media/pa-dagsorden/cyberangrep-pa-hydro/>.
- [119] L. Constantin, *Xen hypervisor faces third highly critical vm escape bug in 10 months*, [Online; accessed 03-February-2021], May 2017. [Online]. Available: <https://www.infoworld.com/article/3194007/xen-hypervisor-faces-third-highly-critical-vm-escape-bug-in-10-months.html>.
- [120] Qubes OS, *Qubesos/qubes-secpack*, [Online; accessed 04-February-2021], Jul. 2016. [Online]. Available: <https://github.com/QubesOS/qubes-secpack/blob/master/QSBs/qsb-024-2016.txt>.
- [121] *What are T<sub>E</sub>X and its friends?* [Online; accessed 6-May-2021]. [Online]. Available: <https://www.ctan.org/tex>.
- [122] Freedomofpress, *Freedomofpress/securedrop-workstation*, [Online; accessed 19-May-2021]. [Online]. Available: <https://github.com/freedomofpress/securedrop-workstation>.
- [123] Djohn, *Use unity mode*, [Online; accessed 19-May-2021]. [Online]. Available: <https://docs.vmware.com/en/VMware-Workstation-Pro/16.0/com.vmware.ws.using.doc/GUID-8C477788-7700-4030-8C4A-039C02AABB74.html>.
- [124] H. Kniberg and M. Skarin, *Kanban and Scrum-making the most of both*. Lulu.com, 2010.
- [125] The Sphinx team, *Overview*, [Online; accessed 6-May-2021]. [Online]. Available: <https://www.sphinx-doc.org/en/master/>.



- [126] D. Goodger, *An introduction to restructuredtext*, [Online; accessed 6-May-2021], Jan. 2012. [Online]. Available: <https://docutils.sourceforge.io/docs/ref/rst/introduction.html>.
- [127] Wikipedia contributors, *Git — Wikipedia, the free encyclopedia*, [Online; accessed 6-May-2021], 2021. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Git&oldid=1021092067>.
- [128] *Your place to talk and hang out*, [Online; accessed 6-May-2021]. [Online]. Available: <https://discord.com/>.
- [129] *Microsoft teams*, [Online; accessed 6-May-2021]. [Online]. Available: <https://www.microsoft.com/en-us/microsoft-teams/group-chat-software>.
- [130] *A hackable text editor for the 21st century*, [Online; accessed 6-May-2021]. [Online]. Available: <https://atom.io/>.
- [131] *Code editing. redefined.* [Online; accessed 6-May-2021]. [Online]. Available: <https://code.visualstudio.com/>.
- [132] *The git solution for professional teams*, [Online; accessed 6-May-2021]. [Online]. Available: <https://bitbucket.org/product/>.

# Appendix A

## Code files

### A.1 splunkforwarder.py

```
#!/usr/bin/env python3

import json
import subprocess

# Import the configuration json file.
# Contains:
# All log locations that should be imported
# All qubes that should export to Splunk - DO NOT EDIT, edit blacklisted qubes,
↳instead
# A list of blacklisted qubes which should not be exported to Splunk
# A log location on the splunk qube for where the log locations are stored,
↳and where
# the splunkforwarder should monitor.
# The script assumes the splunk qube is called splunk. Rename or change in,
↳this script if otherwise

with open("splunkforwarder.conf.json") as json_data_file:
    data = json.load(json_data_file)

# Get a list of running qubes which we can import data to
qubesList = subprocess.run("/usr/bin/qvm-ls --raw-list --running", shell=True,
↳universal_newlines=True,
    stdout=subprocess.PIPE)

qubesList = qubesList.stdout.splitlines()

data["qubes"] = qubesList

print("Found all qubes")
```

(continues on next page)

```

#Remove all the qubes that are blacklisted. Name must match exactly with the
↳config file equivalent

for blacklistedQube in data["blacklisted_qubes"]:
    for qube in qubesList:
        if qube == blacklistedQube:
            qubesList.pop(qubesList.index(qube))

print("Removed blacklisted qubes")

# dom0 needs special handling to copy from:

# copy all files from the config file to the splunk qube

# error: switch case dom0 or Windows

import glob

def qubeLogHandler(qube):
    qubeList = []
    # Get all files that need to be sent to the getFileandCopy function
    for confPath in data["log_locations"]:

        command = subprocess.run('/usr/bin/qvm-run -u root -p '+ qube +' "sudo_
↳find '+ confPath +' -type f"', shell=True, universal_newlines=True,
            stdout=subprocess.PIPE)
        files_with_path = command.stdout.splitlines()

        qubeObject = {'files': files_with_path, 'path': confPath}
        qubeList.append(qubeObject)
    # Recursively push all files to their respective folders and files
    for qubeObject in qubeList:
        for file in qubeObject['files']:
            getFileandCopy(qubeObject['path'], file, qube)

def getFileandCopy(path, filename, vmname):
    # Create the folder for the current file
    pathWithoutFile = filename.rsplit("/", 1)[0]
    subprocess.run('qvm-run -p splunk \
        "mkdir -p '+ data["splunk_qube_log_location"] +'/imported_logs/'+ vmname +
↳/'+ pathWithoutFile +'", shell=True)

    # Cat and push all files to the splunk vm
    subprocess.run('qvm-run -u root -p '+ vmname +' "sudo cat '+ filename +'
↳" \
        | qvm-run -p splunk "cat > '+ data["splunk_qube_log_location"] +'/imported_
↳logs/' + vmname + filename +'", shell=True)

```

```

# We need to handle dom0 as a special qube
def dom0Handler(qube):
    def qubeListHandler(listItem):
        for file in listItem["files"]:
            # Create folder and then copy the files
            pathWithoutFile = file.rsplit("/", 1)[0]
            subprocess.run('qvm-run -p splunk \
                "mkdir -p '+ data["splunk_qube_log_location"] + '/imported_
↳logs/'+ qube + '/' + pathWithoutFile + '", shell=True)
            subprocess.run('sudo cat '+ file + ' | qvm-run -u root -p splunk
↳"cat > '+ data["splunk_qube_log_location"] + '/imported_logs/'+ qube + "' +
↳file + '", shell=True)

        qubeList = []
        for confPath in data["log_locations"]:
            command = subprocess.run("sudo find "+ confPath + " -type f",
↳shell=True, universal_newlines=True,
                stdout=subprocess.PIPE)
            files_with_path = command.stdout.splitlines()
            qubeObject = {'files': files_with_path, 'path': confPath}
            qubeList.append(qubeObject)
        #print(qubeList)
        for listItem in qubeList:
            qubeListHandler(listItem)

for qube in qubesList:
    if qube == "dom0":
        print(qube + " will be handled next")
        dom0Handler(qube)
    else:
        print(qube, " will be executed next")
        qubeLogHandler(qube)

print("Ported logs from folders and qubes designated in config file to the
↳splunk qube")

with open("splunkforwarder.conf.json", "w") as outfile:
    json.dump(data, outfile, indent=4)

```

## A.2 splunkforwarder.conf.json

```
{
  "qubes": [
    "dom0",
    "splunk",
    "sys-firewall",
    "sys-net",
    "sys-usb",
    "sys-whonix"
  ],
  "log_locations": [
    "/var/log",
    "/etc/cron.d"
  ],
  "blacklisted_qubes": [
    "vault",
    "windows-mgmt"
  ],
  "splunk_qube_log_location": "/home/user"
}
```

## A.3 gitscript.sh

```
#!/bin/bash

# Variables
# -----#

VM_NAME=git-qube
GIT_PASSWORD=MyPass
GIT_USER=MyUser
GIT_REPO=MyRepo/saltstack_repo
GIT_HOST=bitbucket.org/
SALT_DIR_PREFIX=/srv/salt
SALT_DIR_TARGET=saltstack_repo/salt
OTHER_DIR_PREFIX=/home/user/git
OTHER_DIR_TARGET=saltstack_repo

# Functions
# -----#

#Function for removing VM:
remove () {
```

(continues on next page)

```

#Halt vm:
qvm-shutdown '${VM_NAME}'

#Check the list of halted VMs until the git-vm appear
vm=""
until [ "$vm" == '${VM_NAME}' ]; do
    for name in $(qvm-ls --fields=NAME --halted); do
        if [ "$name" == '${VM_NAME}' ]; then
            vm="$name"
        fi
    done
    sleep 1
done

yes | qvm-remove '${VM_NAME}'
}

# Script
# -----#

#If there is a VM with the same name, delete it first

for name in $(qvm-ls --fields=NAME); do
    if [ "$name" == '${VM_NAME}' ]; then
        remove
    fi
done

#Creating the VM that will download the git-repo
qvm-create --template=fedora-30 --label=red '${VM_NAME}'

#Check the list of VMs until the git-vm appear
vm=""
until [ "$vm" == '${VM_NAME}' ]; do
    for name in $(qvm-ls --fields=NAME); do
        if [ "$name" == '${VM_NAME}' ]; then
            vm="$name"
        fi
    done
    sleep 1
done

```

```

#Cloning the repo
qvm-run --pass-io '${VM_NAME}' "git clone https://'${GIT_USER}':'${GIT_
↳PASSWORD}'@'${GIT_HOST}':'${GIT_REPO}'.git"

#Copying to dom0:

#Creates directories if they do not already exist
mkdir -p '${OTHER_DIR_PREFIX}'
mkdir -p '${SALT_DIR_PREFIX}'

#Get the file names
otherfiles=$(qvm-run --pass-io '${VM_NAME}' "ls '${OTHER_DIR_TARGET}'/")
saltfiles=$(qvm-run --pass-io '${VM_NAME}' "ls '${SALT_DIR_TARGET}'/")

#Copy the contents from the files
for file in $otherfiles; do
    qvm-run --pass-io '${VM_NAME}' "cat '${OTHER_DIR_TARGET}'/'$file'" > ${OTHER_
↳DIR_PREFIX}/${file}
done

for file in $saltfiles; do
    qvm-run --pass-io '${VM_NAME}' "cat '${SALT_DIR_TARGET}'/'$file'" > '${SALT_
↳DIR_PREFIX}'/'$file'
done

#Remove VM (using remove function):

remove

echo "Done"

```

## A.4 InitializeUsername.py

```

import os

# Add username as default user on windows machine
username = input("What is your username?")

os.system(f"qvm-prefs Work default_user '{username}'")

```

## A.5 Install-VPN.py

```
import os

# activate use of password file on proxy machines | NOT NECESSARY AS QUBES
↳CREATE THIS ON II'S OWN
#os.system("salt '*[p,P]roxy*' file.line /etc/openvpn/openvpn.conf mode=replace
↳match=auth-user-pass content=auth-user-pass auth.txt")

# Add username and passwords to password file.
username = input("What is your username?")
password = input("What is your password?")

os.system(f"qubesctl '*[p,P]roxy*' file.write /rw/config/NM-system-connections/
↳secrets/passwd-file.txt '{username}\n{password}'")
```

## A.6 machines.sls

```
##### Create qubes #####

### Analysis qube ###

CreateAnalysisQubes:
  qvm.present:
    - name: Analysis_deb
    - template: debian-10
    - label: red
    - mem: 2000
    - vcpus: 2

  qvm.present:
    - name: Analysis_fed
    - template: fedora-32
    - label: red
    - mem: 2000
    - vcpus: 2

  qvm.present:
    - name: Analysis_win7
    - template: win7-temp
    - label: red
    - mem: 2048
    - vcpus: 2

  qvm.present:
    - name: Analysis_win10
```

(continues on next page)



- **template:** win10-temp
- **label:** red
- **mem:** 2000
- **vcpus:** 4

**SetupAnalysisQubes:**

**qvm.prefs:**

- **name:** Analysis\_deb
- **maxmem:** 4000
- **include-in-backups:** True
- **netvm:** Analysis\_Proxy
- **virt-mode:** hvm
- **qrexec-timeout:** 120

**qvm.prefs:**

- **name:** Analysis\_fed
- **maxmem:** 4000
- **include-in-backups:** True
- **netvm:** Analysis\_Proxy
- **virt-mode:** hvm
- **qrexec-timeout:** 120

**qvm.prefs:**

- **name:** Analysis\_win7
- **maxmem:** 4000
- **include-in-backups:** True
- **netvm:** Analysis\_Proxy
- **virt-mode:** hvm
- **qrexec-timeout:** 120

**qvm.prefs:**

- **name:** Analysis\_win10
- **maxmem:** 4000
- **include-in-backups:** True
- **netvm:** Analysis\_Proxy
- **virt-mode:** hvm
- **qrexec-timeout:** 120

*### Dev qube ###*

**CreateTestQubes:**

**qvm.present:**

- **name:** Dev\_fed
- **template:** fedora-32
- **label:** orange
- **mem:** 2000
- **vcpus:** 2

```
qvm.present:
- name: Dev_deb
- template: debian-10
- label: orange
- mem: 2000
- vcpus: 2

qvm.present:
- name: Dev_win10
- template: win10-temp
- label: orange
- mem: 2000
- vcpus: 2

qvm.prefs:
- name: Dev_win10
- maxmem: 4000
- include-in-backups: True
- netvm: Dev_Proxy
- virt-mode: hvm
- qrexec-timeout: 120
```

*### Proxy qube ###*

#### CreateProxyQubes:

```
qvm.present:
- name: Analysis_Proxy
- template: debian-10
- label: purple
- mem: 2000
- vcpus: 4
- flags:
- proxy
```

```
qvm.present:
- name: Dev_Proxy
- template: debian-10
- label: purple
- mem: 2000
- vcpus: 4
- flags:
- proxy
```

*### Business qube ###*

#### CreateWorkQube:

```
qvm.present:
- name: Work
```

- **template:** win10-temp
- **label:** blue
- **mem:** 2000
- **vcpus:** 2

**qvm.prefs:**

- **name:** Work
- **maxmem:** 4000
- **include-in-backups:** True
- **netvm:** sys-firewall
- **virt-mode:** hvm
- **qrexec-timeout:** 120

*### Surfing qube ###*

**CreateSurfQube:**

**qvm.present:**

- **name:** Surf
- **template:** fedora-32
- **label:** yellow
- **mem:** 2000
- **vcpus:** 2

**qvm.prefs:**

- **name:** Surf
- **include-in-backups:** True
- **netvm:** sys-firewall

*### SIEM qube*

**CreateSiemQube:**

**qvm.present:**

- **name:** splunkQube
- **template:** fedora-32
- **label:** green
- **mem:** 2000
- **vcpus:** 2

**qvm.prefs:**

- **name:** splunkQube
- **include-in-backups:** True
- **netvm:** sys-firewall

## A.7 top.sls

```
base:
  dom0:
    #Creating the machines
    - machines
    # adding anacron job that git pulls every 7 days
    - pullUpdate

    # matches all analysis machines except for Proxy
    ^Analysis_(?![Pp]roxy).*$:
      - AnalysisSoftware
    # matches all machines with proxy in name
    ^.*(?:[Pp]roxy).*$:
      - proxySoftware
    'Surf':
      # Matches all development machines except for Proxy
      ^Dev_(?![Pp]roxy).*$:
```

## A.8 pullUpdate.sls

```
# Identifier
Configuration:
# path to file
/etc/anacrontab:
# module used
file:
  - managed:
  - contents: |

  SHELL=/bin/sh
  PATH=/sbin:/bin:/usr/bin
  MAILTO=root
  # the macimal random delay added to the base delay of the jobs
  RANDOM_DELAY=45
  # the jobs will be started during the following hours only
  START_HOUR_RANGE=5-21

#period in days | delay in minutes | job-identifier | command
7                3                salt-update      /home/user/git/gitscript.sh
```

## A.9 proxySoftware.sls

```
#identifier
InstallProxySoftware:
  pkg.installed:
    - pkgs:
      - openvpn
      - pivpn

#identifier
Configuration:
  # this directory is not deleted each powecycle and retains changes
  /rw/config/rc.local:
    file:
      - managed:
      - contents: |
        #!/bin/bash

        while ! ping -c 1 -W 1 1.1.1.1; do
          sleep 1
        done
        sudo openvpn /home/user/Documents/target.ovpn

  /rw/config/NM-system-connections/secrets/passwd-file.txt:
    file:
      - managed:
      - contents: "vpn.secrets.password:bachelor2021"
```

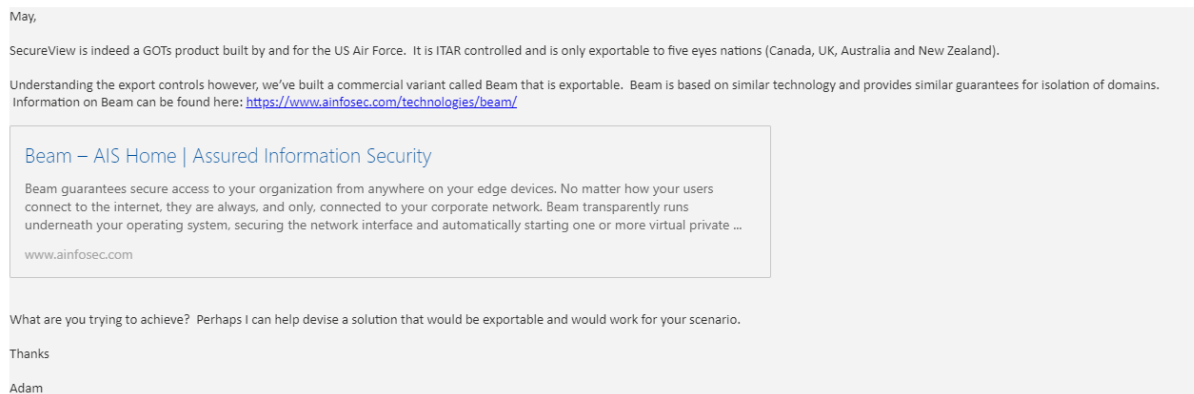
## A.10 AnalysisSoftware.sls

```
AnalysisSoftware:
  pkg.installed:
    {% if (grains['os.family'] == 'Ubuntu') or
      (grains['os.family'] == 'Debian') or
      (grains['os.family'] == 'Fedora') or
      (grains['os.family'] == 'Ubuntu') %}
      - pkgs:
        -
    {% endif %}
```

# Appendix B

## Emails

### B.1 AIS Secureview reply



**Figure2.1:** AIS Secureview reply.

# Appendix C

## Timesheets

### C.1 Example of Timesheet

3.02	9:30	12:30	Meeting with Kongsberg, investigating different hypervisors		3:00
3.02	13:30	17:45	Investigating different hypervisors		4:15
4.02	10:00	13:00	Investigating different hypervisors		3:00
4.02	14:00	17:30	Investigating different hypervisors		3:30
5.02	10:00	13:00	Investigating different hypervisors, trying to install Windows in Qubes OS		3:00
5.02	13:30	17:30	Investigating different hypervisors		4:00
9.02	14:00	15:00	Investigating different hypervisors		1:00
10.02	10:00	12:45	Investigating different hypervisors		2:45
10.02	13:30	15:00	Investigating different hypervisors		1:30
10.02	15:00	16:00	Meeting with Kelly		1:00
10.02	17:30	18:30	Investigating different hypervisors		1:00
11.02	10:00	16:00	Investigating different hypervisors		6:00
12.02	10:00	17:00	Configuring QubesOS		7:00

**Figure3.1:** An example selection of a group members hours.

# Appendix D

## Project plan

### D.1 Goals and limitations

#### D.1.1 Background

The Kongsberg Cyber Security Center (KCSC) is a department within the Kongsberg Group that handles analysis, detection and incident response for all divisions of the group. This type of work necessitates the ability to connect to multiple environments to perform various tasks, such as evidence gathering, log collection and incident handling. KCSC's current routines involve using machines and tooling from each environment, which is cumbersome and unproductive for the analysts.

Analysis of potentially malicious and sensitive content should, for security reasons, be performed in zones separated from the rest of the enterprise.

To prevent usage of a large number of different machines and inconsistent methods of secure file transfer, it is desirable to have a single laptop to perform this work on. Because malware analysis often include executing the malware, this laptop must be unable to run malware in enterprise zones, and at the same time be able to transfer data (such as logs, reports and malware samples) from a secure zone to an insecure zone. This is where the multipurpose platform for security analysts come in. The platform aims to allow proper separation between zones, and still allow results from an eventual analysis to be sent between zones.

#### D.1.2 Project goals

##### Effect goals

It is expected that the finished multipurpose platform will improve the work flow for security analysts by:

- Reducing the number of machines needed to perform analysis on malicious files and incident response work.
- Establishing a defined method by which files, such as malware samples, images, and text, can be securely transferred to and from machines during this work.
- Reducing time required to perform analysis work.



## Result goals

The following baseline requirements have been set forth by KCSC:

- A proof of concept of the multipurpose platform will be realized on the machines provided by KCSC.  
The machines will be configured such that different applications are launched on dedicated virtual machines in various zones. These zones will be appropriately configured based on their need to access the Internet and Kongsberg's intranet.  
The platform must be able to transfer system logs to Splunk, which is a Security Information and Event Manager (SIEM).  
The proof of concept will be installed on several different types of laptops to test compatibility of the platform with different hardware components, as provided by KCSC. Each such machine constitutes its own standalone platform.
- The final report will contain an evaluation of the feasibility of using the multipurpose platform, and suggestions on how this would work in practice. This evaluation will contain a risk assessment, as well as an investigation into how data in virtual machines and on the host can be securely separated from each other.
- Non-trivial code must be documented in such a way that it supports further development of the project by KCSC.

In addition to these goals, KCSC has provided a list of optional features that, while not strictly required, would be nice to have in the project implementation:

- Support for multi-factor authentication.
- Implementation of a location policy that bans travel considered to be impossible, such as moving between physical locations unreasonably quickly (implemented via geo-IP resolution).
- Integration of identities with Microsoft Active Directory.
- Centralized configuration and patch management via e.g. SaltStack, Puppet or Ansible.
- An evaluation of the compatibility of the project implementation against the hardware provided by KCSC.

Result goals that are not requested by KCSC:

- Develop a program which takes input from the user and creates a configuration with a declarative programming language. This way other malware analysts can get a configuration similar to the proof of concept.

### D.1.3 Constraints

#### Time frame

The time frame for the project is January 11, 2021 through May 20, 2021. The estimated workload will be 30 hours per week per student.

## Legal

The student group will comply by regulations set forth by the Norwegian University of Science and Technology on bachelor theses (<https://innsida.ntnu.no/bacheloroppgave>).

In external libraries or source code is used, we will comply with the given licenses and append it to the Bachelor thesis.

## D.2 Scope

### D.2.1 Task description

Working safely with malware can be a challenging and delicate process, with many hurdles that impact the efficiency of analysis work. Our goal is to create a safe, proof of concept multi-use platform for security analysts, integrated with event logging and centralized management for easy deployment and maintenance.

The proof of concept will be a multi-zone virtualization system on one laptop. As various applications are optimized for different operating systems, the host laptop will allow applications from multiple operating systems to run alongside each other on the same laptop. Software running on the host laptop will be divided into zones depending on their security level - for example, a web surfing zone will allow the user access to the Internet, while a malware zone will have strict security limitations to secure the platform. Each zone may include several virtual machines, each running a different operating system, allowing great flexibility in terms of tools and usability. The platform should be integrated with a local SIEM and a configuration management system for easy, centralized monitoring and management.

Several laptops provided by Kongsberg will be used for realizing the proof of concept, and for evaluating compatibility of the platform with different underlying hardware. The group will make recommendations on the best supported hardware to Kongsberg based on the results of this evaluation.

A risk analysis will be performed of the proof of concept platform according to baseline security requirements provided by Kongsberg. Such requirements may include e.g. evaluating the risk of an attacker acquiring Kongsberg's intellectual property in case a machine is stolen, and how such a risk can be properly mitigated. The platform must also be evaluated against NIST-800 and ISO 27000-series security requirements in terms of how data is stored and shared between virtual machines and the host system. Kongsberg further requested looking into whether data can be stored encrypted while virtual machines are offline, but the host is running.

### D.2.2 Field of study

The Bachelor thesis will cover various topics that relate to the group members' field of study. Support materials and insight obtained from previous courses will form the background knowledge required to complete the thesis. These courses include:

**Infrastructure as Code** Configuration, rules and policies sent to the platform from a central manager, to allow for easy configuration of the multiple laptops running the platform.

**Network Security** Configuring the platform's firewall and VPN access.

**Computer Networks** Use basic networking knowledge to configure the laptops networking requirements for Internet connectivity.

**ITSM, Security and Risk Management** Performing risk assessments in an information security context to give a good overview of the risks Kongsberg is facing.

**Operating Systems** Making use of knowledge of virtualization technologies, and understanding how the operating system interacts with hardware.

### D.2.3 Limitations

Due to operating system-specific hardware requirements, not all hardware is capable of running the proof of concept platform. Identifying compatibility issues with the platform, and making a guideline for Kongsberg to use for sourcing hardware for the platform, is within scope of the project.

## D.3 Project management

### D.3.1 Roles and responsibilities

**Team leader** Martin Wahl

**Communications** Steinar Vrenne

**Secretary** May-Liss Rosendahl Amundsen

**Supervisor / Guidance Counselor** Jia-Chun Lin, NTNU

**Contracting Authority** David Lee Andersen, Kongsberg Defence and Aerospace

### D.3.2 Group rules and guidelines

#### Group rules

- All members will attend group meetings every Wednesday, Thursday and Friday, from 10:00 till 16:00, with certain exceptions (e.g. doctor visits, sickness). These meetings will primarily require in-person attendance. If guidelines or regulations concerning covid-19 from the university or local and national government makes this infeasible, or if group member(s) are in quarantine, meetings will be held digitally.
- All working hours shall be logged.
- If a member can not attend a meeting or other designated working hours, they shall prior notice to the other group members. Failure to provide prior notice of absence may result in sanctions being enacted on the group member in question as outlined below.

## Conflict resolution and decisions

In case of disagreements within the group, the group members shall attempt to reach a compromise that all group members in question can agree on. Should this not be possible, a vote shall be cast to determine the outcome. If such a decision cannot be made because there are an equal number of votes for either option, a single-elimination rock-paper-scissors tournament shall be held where the winner is given the deciding vote.

## Sanctions

Depending on the severity of rule or guideline violations, or interpersonal disputes, the following sanctions may be used:

- Conversations between group members concerning the problem.
- Formal write-up by the group leader where the violation is detailed, specific actions required from the offender in question to “make up for” the incident, as well as information on further consequences for repeated and/or continued violation of the guidelines/rules.
- Conversation between group and/or group leader, the offender and guidance counselor.
- Formal decision on exclusion from the group. All parties involved shall be notified. Exclusion can not take place in the last 20 days leading up to the final deadline.

## D.4 Planning, follow-up, and reporting

### D.4.1 Development and writing process

Scrum-ban will be used as the group’s development process, combining the flexibility of daily and weekly scrums with the overview gained from using a Kanban board.

Usage of the waterfall model would be too restrictive, given that we cannot perfectly assess the time required to finalize the various stages of the project, or predict whether or not our system designs will work. Scrum allows the group to plan out the work in a more flexible manner, and allows large work units to be split into smaller, more manageable units if needed.

Other agile development processes such as eXtreme programming and iterative development were also explored, but they were found to include too many aspects that are irrelevant to our task, such as pair programming, due to our project not involving a significant amount of programming.

The group has designated weekly group sessions for Wednesdays through Fridays, each of which will begin with a brief discussion of each group member’s current progress, as well as establishing goals to be achieved that day. Every Wednesday, a weekly progress meeting will be held to synergize group efforts and to ensure everyone is aligned with the current status. Despite planning on working rather dynamically, the group has created a Gantt-diagram to visualize and make predictions on when various stages are expected to be worked on (see preproject-impl-plan).

## D.4.2 Meetings

- Weekly meetings will be held with our advisor.
- Biweekly meetings will be held with Kongsberg representatives.
- Internal scrum meetings are held daily at 10 AM on Wednesdays, Thursdays and Fridays.
- At the end of every Friday, the group will briefly discuss its progress that week.

## D.5 Quality assurance

### D.5.1 Documentation, standards and source code

Project documentation will be written in reStructuredText format and compiled into LaTeX. This will be appended to the final report along with all code written for the project.

Kongsberg needs the platform to comply with their security policies - this implies evaluating it against requirements set forth in NIST-800 and ISO 27000 series.

To reduce the chance of data loss, and to improve cooperation in production, we have decided to use Bitbucket as our version control system, and its pipeline functionality in order to catch mistakes using automated tests.

Kongsberg is lending us testing equipment to allow realistic benchmarking and testing.

### D.5.2 Configuration management

To offer configuration for the platform, we will use known programming languages, for example SaltStack, Ansible, Puppet or other depending on compatibility with the platform. We will use BitBucket as our repository for source code.

### D.5.3 Risk management

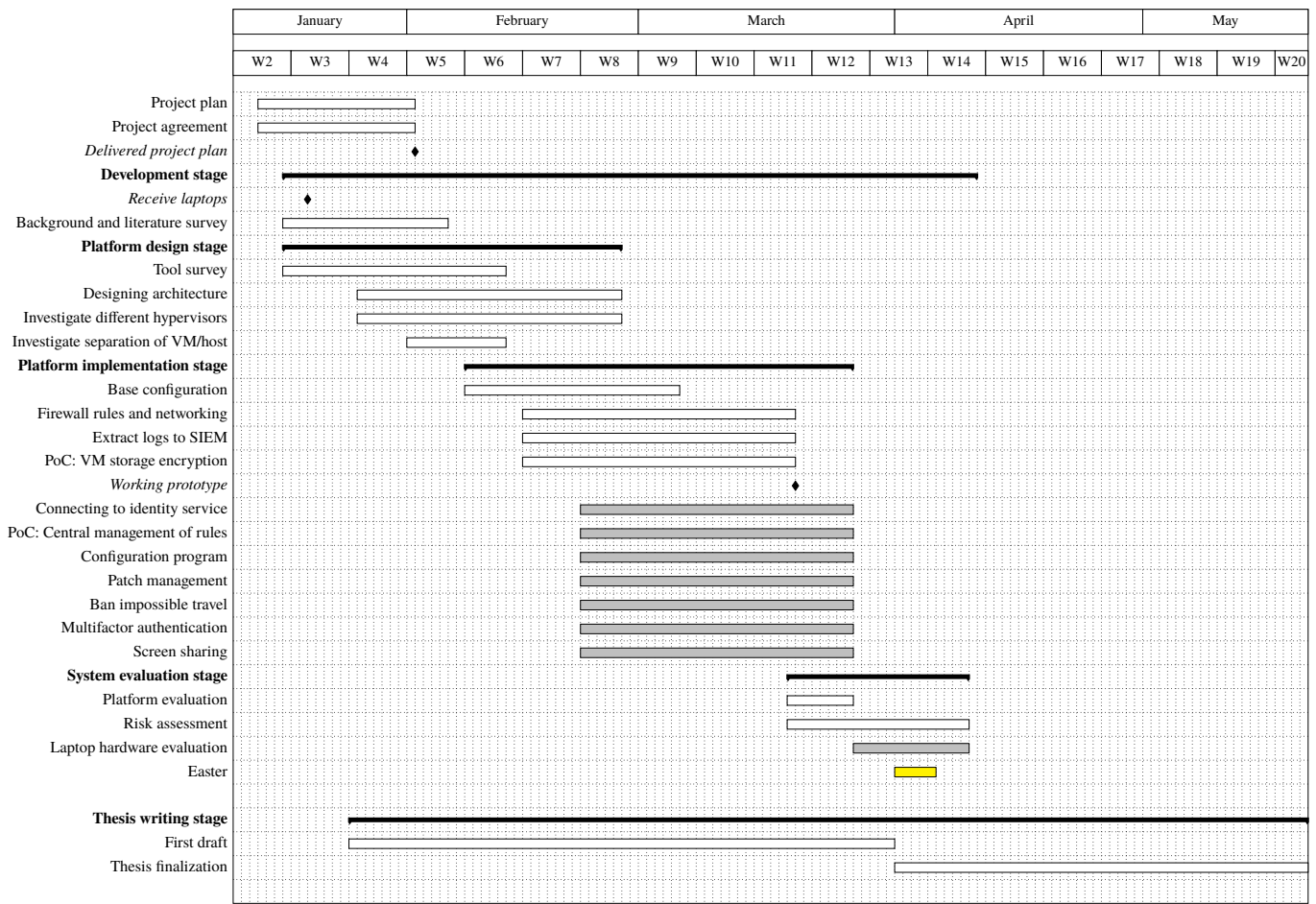
Below is a table listing project related risks and measures for the most severe risks.

Risk	Probability	Consequence	Measures
Report not delivered on time	Low	High	Our projection of work includes “nice to have” features which can be cut if we have mismanaged our time.
Miscommunication with client or advisor	Low	Medium	Bi-weekly meetings with client, and weekly meetings with advisor. Single point of contact for e-mail.
We use unacceptable assumptions to complete our task	Medium	High	Bi-weekly meetings with client, and weekly meetings with advisor. Single point of contact for e-mail.
Project related data loss	Low	High	Source code and thesis are both written with version control systems (Bitbucket). Weekly backups of info stored in cloud services.
Disruptive disagreements regarding project details	Medium	Low	We have set up procedures to ensure disagreements are dealt with quickly.
Quarantining and loss of productivity due to sickness	Medium	Medium	All group members perform social distancing and follow local and national requirements & guidelines.
Equipment provided by client is lost or fails	Low	Medium	Equipment stored at designated locations and not left unattended in open spaces.

## D.6 Implementation plan

### D.6.1 Project stages

The project is divided into different stages shown as black bars in the Gantt chart. The development stage concerns the entire development of the system, and includes the design stage, the implementation stage and the evaluation stage. The required tasks that need to be performed in each of the stages are shown as white bars in the chart. Tasks representing optional features are shown as gray bars. The thesis will be worked on and written continuously throughout the project as the development progresses.



**Investigate different hypervisors:** Investigate different hypervisors needed for the project. Give recommendations for the best hypervisor for this project, based on incompatibilities, security concerns and previous work with the hypervisor.

**Investigate separation of VM/host:** To protect data against VM escapes, separation between the VMs and the host needs to be established. Read research papers and documentation to investigate how VMs perform this separation and how VM escape can be avoided.

**Base configuration:** Investigate the appropriate configuration methods for the chosen hypervisor and configure the zones and associated VMs with regards to the current best practices (e.g. ISO-2700, NIST-800 and NSM's guidelines).

**Firewall rules and networking:** Only certain zones should have access to the open Internet. In the remaining zones, only VPN connection to approved IP addresses should be available. Malware must not be able to enter zones where it does not belong. Traffic to RFC1918 (private) IP addresses from untrusted VMs (e.g. malware lab) must be blocked. Based on the approved architecture drawings, configure firewall rules.

**Extract logs to SIEM:** Evaluate what value the logs will give in regards to the security of the platform, and what type of logs are of interest. Synchronize all possible logs with a SIEM, so that it is possible to investigate and uncover events and their causes without the need to send the laptop itself to Kongsberg for repair.

**PoC: VM storage encryption:** Investigate and make a proof of concept of how data stored in a VM can be encrypted separately from the machine host.

**Connecting to identity service:** Investigate and find a way for the platform to integrate with Kongsberg's

identity service and incorporate itself into their domain. Kongsberg's preferred identity service is Microsoft Active Directory.

**PoC: Central management of rules and Patch management:** Use declarative programming to program the platform to receive rules, configurations (e.g. firewall rules, software updates) and policies for all laptops from a central configuration management system.

**Configuration program:** Investigate what type of configuration and functionalities a malware analyst might need or want, and define parameters for the different configurations. Create a program which can automatically create a secure environment based on the parameter input from the user. Create a manager that will be able to push this configuration out to all the relevant laptops or workstations.

**Ban impossible travel:** If a user tries to log in to the platform from a physical location that is “impossibly” far away from where they last logged in, this might indicate that the user has been compromised. If a user tries to log in to the platform from a physical location that is not recognized or approved by Kongsberg, this might indicate that the laptop is stolen. Introduce security measures based on geolocation: Assess the possibility to incorporate geolocation policies to lock the laptop if it detects impossibly quick travel or login attempts from unapproved geolocations. This may be incorporated with Security as a Service from a cloud provider.

**Multifactor authentication:** Investigate if it is possible to require the use of multifactor authentication from the user before performing certain tasks. This could be done either via a mobile app or USB key.

**Screen sharing:** Screen sharing is a convenient feature for sharing the project progress with the client. It is also a convenient feature for the client to be able to screen share in their daily use of the platform. Investigate the possibility to share or capture the screen in the platform, possibly with software or a physical capture card.

**Platform evaluation:** Test the platform with feedback from KCSC. This will involve user testing and assessment of usability.

**Laptop hardware evaluation:** Test the proof of concept on different laptops with varying hardware configuration, and report any compatibility issues.

## Milestone 1: Receive laptops

The first milestone of the project is when the project group receives the laptops from the client. Reaching this milestone no later than the beginning of the design stage is an advantage because it allows for hardware knowledge and experience to be included early in the design decisions. This milestone needs to be reached before the implementation stage can begin.

## Milestone 2: Delivered project plan

This milestone marks the deadline for the completed project plan. For this milestone to be reached, both the project plan and the project agreement must be delivered. The background and literature survey, as well as the design stage, will start and progress parallel to project planning.



### **Milestone 3: Working prototype**

At this point the design stage needs to be completed, and the development stage needs to be in the final stages. All functionality necessary for the platform to work correctly must be implemented and behave as expected. Optional features in the platform implementation stage will have low priority until this milestone is reached. After the milestone is reached, the remaining part of the implementation stage should be focused on implementing optional features. This is also when the platform evaluation stage begins.

#### **D.6.2 Resources**

We have requested and been approved for 32 virtual CPUs, 50 GB RAM, and 500 GB of block storage in SkyHiGh, NTNU Gjøvik's local datacenter.

We have received laptops from Kongsberg to be used for testing the platform.

We have sourced a capture card to record the desktop for demonstration purposes.

## **Appendix E**

### **Group rules**

## Group rules

- 1) All members will attend group meetings every Wednesday, Thursday and Friday, from 10:00 till 16:00, with certain exceptions (e.g. doctor visits, sickness). These meetings will primarily require in-person attendance. If guidelines or regulations concerning covid-19 from the university or local and national government makes this infeasible, or if group member(s) are in quarantine, meetings will be held digitally.
- 2) All working hours shall be logged.
- 3) If a member can not attend a meeting or other designated working hours, they shall provide prior notice to the other group members. Failure to provide prior notice of absence may result in sanctions being enacted on the group member in question as outlined below.

## Conflict resolution and decisions


In case of disagreements within the group, the group members shall attempt to reach a compromise that all group members in question can agree on. Should this not be possible, a vote shall be cast to determine the outcome. If such a decision cannot be made because there are an equal number of votes for either option, a single-elimination rock-paper-scissors tournament shall be held where the winner is given the deciding vote.

## Sanctions

Depending on the severity of rule or guideline violations, or interpersonal disputes, the following sanctions may be used:

- Conversations between group members concerning the problem.
- Formal write-up by the group leader where the violation is detailed, specific actions required from the offender in question to "make up for" the incident, as well as information on further consequences for repeated and/or continued violation of the guidelines/rules.
- Conversation between group and/or group leader, the offender and guidance counselor.
- Formal decision on exclusion from the group. All parties involved shall be notified. Exclusion can not take place in the last 20 days leading up to the final deadline.

## Signatures



May-Liss Amundsen



Marius Ødegård Lindvall



Steinar Vrenne



Martin Wahl

## **Appendix F**

### **Project agreement**

## Prosjektavtale

mellom NTNU Fakultet for informasjonsteknologi og elektroteknikk (IE) på Gjøvik (utdanningsinstitusjon), og

**Kongsberg Gruppen** (oppdragsgiver), og

**May-Liss Amundsen, Marius Ødegård Lindvall, Steinar Vrenne og Martin Wahl** (student(er))

Avtalen angir avtalepartenes plikter vedrørende gjennomføring av prosjektet og rettigheter til anvendelse av de resultater som prosjektet frembringer:

1. Studenten(e) skal gjennomføre prosjektet i perioden fra **11/1-2021** til **20/5-2021**.

Studentene skal i denne perioden følge en oppsatt fremdriftsplan der NTNU IE på Gjøvik yter veiledning. Oppdragsgiver yter avtalt prosjektbistand til fastsatte tider. Oppdragsgiver stiller til rådighet kunnskap og materiale som er nødvendig for å få gjennomført prosjektet. Det forutsettes at de gitte problemstillinger det arbeides med er aktuelle og på et nivå tilpasset studentenes faglige kunnskaper. Oppdragsgiver plikter på forespørsel fra NTNU å gi en vurdering av prosjektet vederlagsfritt.

2. Kostnadene ved gjennomføringen av prosjektet dekkes på følgende måte:
  - Oppdragsgiver dekker selv gjennomføring av prosjektet når det gjelder f.eks. materiell, telefon, reiser og nødvendig overnatting på steder langt fra NTNU i Gjøvik. Studentene dekker utgifter for ferdigstillelse av prosjektmateriell.
  - Eiendomsretten til eventuell prototyp tilfaller den som har betalt komponenter og materiell mv. som er brukt til prototypen. Dersom det er nødvendig med større og/eller spesielle investeringer for å få gjennomført prosjektet, må det gjøres en egen avtale mellom partene om eventuell kostnadsfordeling og eiendomsrett.
3. NTNU IE på Gjøvik står ikke som garantist for at det oppdragsgiver har bestilt fungerer etter hensikten, ei heller at prosjektet blir fullført. Prosjektet må anses som en eksamensrelatert oppgave som blir bedømt av intern og ekstern sensor. Likevel er det en forpliktelse for utøverne av prosjektet å fullføre dette til avtalte spesifikasjoner, funksjonsnivå og tider.
4. Alle beståtte bacheloroppgaver som ikke er klausulert og hvor forfatteren(e) har gitt sitt samtykke til publisering, kan gjøres tilgjengelig via NTNUs institusjonelle arkiv NTNU Open.

Tilgjengeliggjøring i det åpne arkivet forutsetter avtale om delvis overdragelse av opphavsrett, se «avtale om publisering» (jfr Lov om opphavsrett). Oppdragsgiver og veileder godtar slik offentliggjøring når de signerer denne prosjektavtalen, og må evt. gi skriftlig melding til studenter og instituttleder/fagenhetsleder om de i løpet av prosjektet endrer syn på slik offentliggjøring.

Den totale besvarelsen med tegninger, modeller og apparatur så vel som programlisting, kildekode mv. som inngår som del av eller vedlegg til besvarelsen, kan vederlagsfritt benyttes til undervisnings- og forskningsformål. Besvarelsen, eller vedlegg til den, må ikke nyttes av NTNU til andre formål, og ikke overlates til utenforstående uten etter avtale med de øvrige parter i denne avtalen. Dette gjelder også firmaer hvor ansatte ved NTNU og/eller studenter har interesser.

5. Besvarelsens spesifikasjoner og resultat kan anvendes i oppdragsgivers egen virksomhet. Gjør studenten(e) i sin besvarelse, eller under arbeidet med den, en patentbar oppfinnelse, gjelder i forholdet mellom oppdragsgiver og student(er) bestemmelsene i Lov om retten til oppfinnelser av 17. april 1970, §§ 4-10.
6. Ut over den offentliggjøring som er nevnt i punkt 4 har studenten(e) ikke rett til å publisere sin besvarelse, det være seg helt eller delvis eller som del i annet arbeide, uten samtykke fra oppdragsgiver. Tilsvarende samtykke må foreligge i forholdet mellom student(er) og faglærer/veileder for det materialet som faglærer/veileder stiller til disposisjon.
7. Studenten(e) leverer oppgavebesvarelsen med vedlegg (pdf) i NTNUs elektroniske eksamenssystem. I tillegg leveres ett eksemplar til oppdragsgiver.
8. Denne avtalen utferdiges med ett eksemplar til hver av partene. På vegne av NTNU, IE er det instituttleder/faggruppeleder som godkjenner avtalen.
9. I det enkelte tilfelle kan det inngås egen avtale mellom oppdragsgiver, student(er) og NTNU som regulerer nærmere forhold vedrørende bl.a. eiendomsrett, videre bruk, konfidensialitet, kostnadsdekning og økonomisk utnyttelse av resultatene. Dersom oppdragsgiver og student(er) ønsker en videre eller ny avtale med oppdragsgiver, skjer dette uten NTNU som partner.
10. Når NTNU også opptrer som oppdragsgiver, trer NTNU inn i kontrakten både som utdanningsinstitusjon og som oppdragsgiver.
11. Eventuell uenighet vedrørende forståelse av denne avtale løses ved forhandlinger avtalepartene imellom. Dersom det ikke oppnås enighet, er partene enige om at tvisten løses av voldgift, etter bestemmelsene i tvistemålsloven av 13.8.1915 nr. 6, kapittel 32.
12. Deltakende personer ved prosjektgjennomføringen:

NTNUs veileder (navn): **Jia-Chun Lin**

Oppdragsgivers kontaktperson (navn): **David Lee Andersen**

Student(er) (signatur): Martin Wall dato 19/1-21

Marius O. Lindahl dato 14/1-21

Steinar Vrenne dato 14/1-21

Maybiss Andersen dato 16/1-21

Oppdragsgiver (signatur): Thomas R. Andersen dato 26/1-21

Signert avtale leveres digitalt i Blackboard, rom for bacheloroppgaven.  
Godkjennes digitalt av instituttleder/faggrupeleder.

Om papirversjon med signatur er ønskelig, må papirversjon leveres til instituttet i tillegg.  
Plass for evt sign:

Instituttleder/faggrupeleder (signatur): \_\_\_\_\_ dato \_\_\_\_\_





# Appendix G

## Meeting minutes

### G.1 Jan 14: Supervisor meeting

**Date** 2021-01-14

**Time** 15:05

**Supervisor** Jia-Chun Lin

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

Digital meeting on Teams where the project supervisor presents a “to do / to know” list containing deadlines and tasks, followed by questions from the project group.

#### G.1.1 Presentation key points

- The “to do / to know” list will be uploaded in Teams.
- The project plan is to be sent to the supervisor before February 1 (not submitted in BB).
- The 1st draft (due March 31) should contain the structure and chapters of the thesis, but the thesis does not have to be finished. It must be sent to the supervisor, and the group will get feedback on it.
- Every week the group must send a status report to the supervisor one day before our meeting. Details of what this report should contain can be found in the “to do / to know” list.

#### G.1.2 Questions from the project group

1. **Is there a preference of what repository the group should use for configuration files etc. (Bit-Bucket, GitHub, GitLab, Selfhosted)?**

Answer: No. The group can choose this together with the client

2. **Should the group write the log book/journal (including for example what technology the group use and why, etc.) individually or as a group?**

Answer: The status report that the group send to the supervisor weekly before every meeting can be considered as a journal. The group should use a shared folder/repositories for loose files etc

3. **Should the group include the client to the project guidance teams channel?**

Answer: Unless there’s a special reason, the meetings should be held separate.

4. **Should the group fill in “vår dato” and “vår referanse” in the project agreement?**

Answer: No.

## G.2 Jan 14: Group decisions

**Date** 2021-01-14

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.2.1 Group decisions

Project agreement:

- Needs to be signed digitally by all members as soon as possible. Uploaded in Nextcloud.

Project plan:

- Scrum-ban will be used as development method.
- Group meetings will be every Wednesday, Thursday, and Friday from 10:00
- Group meetings will start with a status meeting / daily scrummeeting.
- The first group meeting of the week will start with a weekly scrum meeting.
- Group rules decided.

Other:

- The thesis will be written in English.
- The thesis will be written in reStructuredText file format.
- Martin and May-Liss are responsible for booking group rooms.
- Timesheets for every group member must be filled in Nextcloud.

### G.2.2 Other info

- The computers are on their way from KDA to Steinar.

## G.3 Jan 20: Client meeting

**Date** 2021-01-20

**Time** 10:00

**Client participants** David Lee Andersen, Erlend Hammer

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

Digital meeting on Teams

### G.3.1 Decisions and info

- The project contract is sent to the client via email.
- The invites to the Teams group are sent to the client's KM mail addresses.

### G.3.2 Questions from the project group

1. **What repository should the group use for the code? For example Bit Bucket.**

Answer: The client will limit the use of internal information in the project, so that there will be no confidential information in the report/project. Because of this, the group can choose what repository they want. The reason for why the group chose this repository needs to be included in the report.

2. **When is the client's easter break?**

Answer: Week 13

3. **Should the project include user tests?**

Answer: User tests are not necessary. The client wishes that the group give them a demo when it is ready. This may be achieved by finding a way to screen share while using cubes. This is "nice to have".

## G.4 Jan 20: Daily scrum meeting

**Date** 2021-01-20

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.4.1 Goals for the day

- Try to start the computers from the client with Qubes OS.
- Finish the Gantt diagram first draft.
- Write a status report to the project supervisor.
- Mail NTNU regarding skyhigh access for Splunk.
- Decide on the amount of "homework" to read through regarding Qubes OS documentation.

## G.5 Jan 20: Weekly scrum meeting

**Week** 3

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

## G.5.1 Goals for the week

- Work on the background and literaturesurvey.
- Work on the tool survey.
- Set up splunk instance in openstack.
- Make the gantt diagram in latex.
- Finish the project plan.

## G.5.2 Decisions

- Every other Wednesday, when we group has a meeting with the client, we will meet at 09:30.
- “Homework” for the group will be published in the Discord channel “qubes-os-homework”.

## G.6 Jan 21: Supervisor meeting

**Date** 2021-01-21

**Time** 14:00

**Supervisor** Jia-Chun Lin

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

Digital meeting on Teams where the project group presents the progress and plan for the weekend gets feedback from the project supervisor.

### G.6.1 Key points and questions

1. **Are there any relevant or similar bachelor theses that we can read through, for example theses about assessing possibilities of a system?**

Answer: The group should search with keywords such as “security assessment” or “pentest article” keyword to find relevant articles talking about this. Should be able to find some examples of theoretical descriptions on how to precede. Can use the articles we find as references in the paper.

2. **Feedback on Gantt diagram**

Answer: The group does not need detailed time periods of when to write what part of the thesis. Instead, it should be parted into two parts: first draft and final thesis. The LaTeX style of the Gantt chart looks good.

3. **Feedback on the project plan**

Answer: The implementation plan is unclear (only the milestones are described). It should be clear *why* the client wants the group to do the project, make different zones etc. “What do we want to deliver?” “What is our final product?” “What would this system look like, and what functions will it have?” This needs to be answered in the report. Study ISO 27000 series and NIST-800 (security requirements) and come up with a plan on how to implement it in the way that meets the requirements (safe enough to be used by Kongsberg).

Task until next meeting:

- How are we going to assesstheplatform, how to testit (pentesting etc.).
- Find a systematic procedure to follow. Try to find papers with keywords such as “assessment” and “evaluation” and find out how other people did evaluations/assessments of systems.

- Post in teams if we find other articles / theses if we want feedback as to whether it is relevant.

Other: Send the finished project plan to project supervisor (not Blackboard).

## G.7 Jan 21: Daily scrum meeting

**Date** 2021-01-21

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.7.1 Goals for the day

- Read through the project plan.
- Set up Bitbucket.
- Meeting with the project supervisor.
- Finish Gantt diagram.
- Finish and deliver the project plan.

## G.8 Jan 22: Daily scrum meeting

**Date** 2021-01-22

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.8.1 Goals for the day

- Send an explanation of the requirements to the project supervisor.
- Update result goals
- Finish the implementation part of the project plan.
- Finish and deliver the project plan.
- Make a structure in Bitbucket for appendixes etc.

## G.9 Jan 27: Weekly scrum meeting

**Date** 2021-01-27

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.9.1 Goals for the day

- Describe the tasks and how we plan to implement them.
- Make an architecture drawing of the system.
- Send the architecture drawing to the client.
- Deliver the project plan (deadline 31.01)
- Read Qubes documents posted in the qubes-os-homework discord channel.

## G.10 Jan 28: Supervisor meeting

**Date** 2021-01-28

**Time** 14:00

**Supervisor** Jia-Chun Lin

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

Digital meeting on Teams, weekly supervisor meeting with feedback. The group presents the project progress from the previous week and plans for the following week.

### G.10.1 Group presentation on key points

- The project plan has been delivered. Needs to be revised after feedback.
- Presenting the first draft of the architecture drawing and explaining the zones and internet / VPN connections and servers.
  - Waiting for an answer from the client about their current workflow concerning malware and the VPN connection to the servers in the red zone.
- Problems:
  - Encryption requirements: This requirement seemed like the client wanted to know how to encrypt each VM, but it has been cleared up that they want to know how they could trust the separation of the VMs.

### G.10.2 Going through the report

- When mentioning that malware should be unable to run malware other places, it should be mentioned that it is common to run the malware when performing the analysis.
- Be more precise:
  - Use the word “malware sample” instead of “data” if the data referred to is a malware sample.
  - When saying “allow results to be forwarded”, specify where it should be forwarded.
  - Write “malware analysis”, not just “analysis”.
  - When mentioning files, specify what type of files, malware samples etc.
- Instead of writing “improving the productivity” explain how: Cut cost on machines? Reduce time spent on moving between machines?
- Use more academic, professional language (e.g. the word “installed” should be changed).
- Use the same word for when mentioning the same thing:

- Machines (or computers/laptops) are more specific than hardware/analysis platform. If hardware and machines is the same thing, use the same word each time. “Hardware configurations” could be changed to “several different types of laptops” or “laptops with different hardware”.
- Are “various analysis tools” and “applications” the same thing? If so be consistent with the name.
- Use the word “platform” instead of “Standalone installation”.
- Use one or two sentences to describe what Splunk is (in text right after the term is ok).
- Field of study: Describe the content that is relevant to the project and why. Describe how this field will be used in the project. Reference the sources.
- Central management: Explain more about what this is. The term is a bit misleading. Might mention pushing configurations etc.
- Risks: Do not write “see above”, better to copy paste.
- Gantt chart: Use the word “platform” not “system”.

### G.10.3 Questions from the project group

#### 1: Should the report be written in personal or objective form?

Answer: It does not matter; it can be mixed.

### G.10.4 Discussing the project

The ultimate goal for the project/platform is to allow for the client to run the malware analysis separately, at the same time allowing them to have business applications on the same hardware.

The group should find out how to make a solution that any company can use. A platform that is more general.

- Can a general method be made, where a company can state their specifications and the configuration can automatically be suggested or realized?
- For example: The client can state how many zones they want and what functions are required for each zone (business, malware etc). Given the number of zones and functionality, then the system is able to automatically provide a suggestion for configuration of the zones. This may be automatically configured instead of manually.
- Can other hypervisors than Qubes OS do the same?
  - Which hypervisors have similar capabilities to Qubes OS?
  - Could companies enter their preferred hypervisor in our solution and get a configuration suggestion?
  - Can the platform and central management etc. be used with other types of hypervisors?
  - Each type of hypervisor might need its own specific configurations.
  - May have to focus on a specific type of hardware.
  - Find out how different types of hypervisors separate machines and applications.
- If the group choose to focus on Qubes OS, good arguments need to be provided. Why should companies need/choose to use Qubes?
- The group can provide own ideas, and do not need to use these suggestions. But it is recommended to do more than only manual configuration.
- The project plan might need to be revised again.

## G.10.5 Other

- In the thesis, words/names should be described in text, as well as having a separate page of glossary (list of terminology).

## G.11 Jan 28: Daily scrum meeting

**Date** 2021-01-28

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.11.1 Goals for the day

- Make a power point for the meeting with the project supervisor.
- Work on the architecture drawing.
- Start setting up Qubes with zones.

### G.11.2 Decisions

- Research more hypervisors and find out if there are alternatives to Qubes OS.
- Write a chapter on why we chose the hypervisor compared to other hypervisors.
- Make a system that is able to automatically provide a suggestion for configuration based on input from the user.

## G.12 Feb 03: Weekly scrum meeting

**Week** 5

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.12.1 Goals for the week

- Continue tool survey and background/literature survey.
- Hypervisors: What are the limitations of different hypervisors and - Write first draft for the thesis. - Note all sources and date.
- Investigate separation of VM and host: - Write first draft for the thesis. - Note all sources and date.
- Designing architecture: - Describe the architecture drawing and the zone connections.



## G.13 Feb 05: Daily scrum meeting

**Date** 2021-02-05

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.13.1 Status

- Windows: Installing windows 10 with the Qubes tools included is challenging. Have not yet tried windows 7.
- Investigate different hypervisors: More investigation is needed.

### G.13.2 Goals and tasks for the day

- Send Status report to supervisor.
- Install windows 7 with Qubes tools.
- Work on architectural drawing.
- Investigate different hypervisors.
- Investigate separation of VM and host.

## G.14 Feb 10: Weekly scrum meeting

**Week** 6

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.14.1 Goals for the week

- Finish investigation of hypervisors
- Finish investigation of separation of VM and host
- Finish architecture drawing
- Tool survey: - Start writing about other tools
- **Base configuration in Qubes:**
  - Set up zones
  - Network rules
  - Test the functionalities

## G.15 Feb 11: Supervisor meeting

**Date** 2021-02-11

**Time** 14:00

**Supervisor** Jia-Chun Lin

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

Digital meeting on Teams, weekly supervisor meeting with feedback and discussion. The group presents the project progress from the previous week and plans for the following week.

### G.15.1 Investigation of different hypervisors

- The group have found hypervisors similar to Xen, and some alternatives to Qubes:
  - Hysolate (limited to windows OS)
  - PolyXene (perhaps still in development? lack of documentation).
- The investigation of hypervisors can be a separate chapter before the description of zones etc.
  - Explain the different types of hypervisors.
  - Explain how you decided to assess different hypervisors.
  - List the investigated hypervisors. Describe the advantages and disadvantages.
    - The list should contain the most popular options and the options with the most fitting functionalities (perhaps less known options).
    - Can rule out some if they do not provide the same functionalities or security.
    - Convince the reader why you ended up with your decision.
    - Qubes is suggested by the client. Maybe Qubes is more mature than other options.

### G.15.2 Problems

- Windows 10 installation does not allow having one VM to one specific task (“seamless mode”). Have to discuss with the client if this a problem or not.
- LaTeX has a maximum number of write files, and too many files written. Found a package on the LaTeX site that seem to have fixed the problem.

### G.15.3 Questions from the supervisor

- What OS should be on the VMs?
  - Attempt to make linux, windows 10 and (if the client needs it) windows 7.
  - Ask the client what linux they use, and what tools are relevant.
- Are the zones tied to each VM? If there are several applications on the same VM, which application belongs to which zone?
  - Zones represent the trust levels.
  - Zones are not forced on the application level, and only represent which VMs are trusted or untrusted.
  - Two VMs can share applications even if the VMs are in different zones.
    - Each VM is built on a template and has an overlay on top of that template. The template is read only. The application runs in the VM based on the template, but the VM cannot change the template. Can only write/store in the user home directory of the VM.
  - You can limit what applications the different zones can use.
  - You can also limit what applications the different VMs can use.
  - Zones have colours so you can always see which zone the VM is in.
  - There is a menu in Qubes where you can choose zone and each zone has different applications you can choose to open in that zone.

## G.15.4 Other

- Possibly related work<sup>3</sup>
  - Look for related work and describe why this project is different from them. Reference to it if it is similar to this project.
- When writing citations, add the date that you got the website.

## G.16 Feb 11: Daily scrum meeting

**Date** 2021-02-11

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.16.1 Status

- Architecture drawings almost done.
- Investigation of separation between VM and host in progress
- Investigation of different hypervisors in progress
- Windows 7 and 10 installed on Qubes.

### G.16.2 Goals and tasks for the day

- Write text to the architecture drawings.
- Continue investigation of separation between VM and host.
- Finish investigation of different hypervisors.
- Meeting with supervisor.

### G.16.3 Decisions

- Investigation separation of VM/host should include VM storage encryption (ref. conversation with the client on Teams channel: Kongsberg - Multipurpose platform for security analysts).

## G.17 Feb 17: Weekly scrum meeting

**Week** 7

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

---

<sup>3</sup> <https://orange.cyberdefense.com/global/blog/sensepost/reevaluating-qubes-os-as-a-pentesting-platform/>

## G.17.1 Tasks for the week

- Tool survey
  - Splunk
  - Active Directory/FreeIPA
  - VPN
  - Salt Stack/Ansible/Puppet
- Separation of VM and host
  - General description
  - VM storage
  - VM communication (QREXEC)
- Firewall rules and networking
- Extract logs to SIEM
  - Set up Splunk.
- Webcam and sound
  - Prioritized last of these tasks.
  - If this is not done before week 9 it will be prioritized after the optional features.

## G.18 Feb 18: Supervisor meeting

**Date** 2021-02-18

**Time** 14:00

**Supervisor** Jia-Chun Lin

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

Digital meeting on Teams, weekly supervisor meeting with feedback and discussion. The group presents the project progress from the previous week and plans for the following week.

### G.18.1 Group presentation key points

- Tasks for next week:
  - Make decisions regarding what tools will be used (Splunk, Active Directory/FreeIPA, VPN, SaltStack/Ansible/Puppet) and why.
  - Start using Splunk and extract to SIEM.
- Almost on track. Some tasks from last week are still not finished.
- Problem with audio and USB in Windows in Qubes:
  - There are no simple solutions out of the box, but there are some open-source tools to get around this.
  - Qubes uses USB over IP (Qrexec), but this is not compatible with Windows.
  - To get audio to work in Windows, it is possible to assign the PCI device to the Windows VM. But only one VM can use it at a time. This means that Windows VMs cannot play audio at the same time. But it is possible to create a virtual speaker device that forwards the audio through a Linux VM.

- Receiving audio is complicated. There are some templates in c++ but the group would have to code a driver to share the microphone between VMs.

## G.18.2 Questions from the supervisor

1. **What machines should be used for surfing the web?**

Answer: Any. Linux or even Windows 10 can be used. Any OS with a modern browser should work for this purpose.

2. **Are the laptops powerful enough to host all the zones?**

Answer: Depends on the machine. The Ryzen machine would probably work, but some of them might be too slow. Hardware recommendations will be included in the report.

3. **What is the next step in the project?**

Answer: Two laptops have enough installed OSes to set up the zones/domains. At least one machine will be used to configure the firewall and make it work. This will provide a rough blueprint that can be used to start working on the central management system.

## G.18.3 Questions from the project group

1. **Are there any updates on the presentation date?**

Answer: Around 8th or 9th of June but need a final confirmation.

2. **When is the deadline of the reflection note?**

Answer: Same as the thesis. This also needs a final confirmation.

3. **Should the square brackets in the citations be before or after period?**

Answer: Before.

## G.19 March 3: Weekly scrum meeting

Week 9

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.19.1 Tasks for the week

- Splunk: Write in thesis (Steinar)
- AD: Try to set up and write in thesis (May-Liss)
- Background: Write in thesis (Marius)
- VPN: Working, write about in thesis (Martin)

## G.20 March 10: Supervisor meeting

**Date** 2021-03-10

**Time** 14:00

**Supervisor** Jia-Chun Lin

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

Digital meeting on Teams, weekly supervisor meeting with feedback and discussion. The group presents the project progress from the previous week and plans for the following week.

### G.20.1 Qubes demo

In the demo the group show: - The Qubes manager, an overview of all VMs (called qubes)

- Icon that is white and barely visible means it is an app VM (seamless mode).
- Icon that is one screen means it is a standalone VM (like Windows 10).
- Icon that is double screen means it is a template.
  
- Starting a disposable surfing VM based on the surfing template. The VM is isolated from all other VMs and deleted when the window is closed.
- Starting a Windows 10 VM. Windows 10 has to run as a standalone machine. There is no integration for having one VM for one specific task (“seamless mode”) on Windows 10.
- VPN proxy VMs that acts as the network provider for this Windows 10 VM. All traffic from the VM goes through the proxy VM.
- Windows 7 has integration for seamless mode.

### G.20.2 Questions from the supervisor about the demo:

1. **Are all the VMs displayed in the manager requested by the client? Or is it just for testing?**  
Answer: Some of them are things the client wants us to test, for example different kinds of linux VMs. Some of them are default (for example all the system ones). Some of them we made for testing.
2. **How do you allocate what application the VM runs?**  
Answer: Demo: How to see available applications in the VM and choose which to use.
3. **The configurations will be the same on all 4 laptops?**  
Answer: All the qubes (VMs) on the laptops will be the same, synchronized with SaltStack, but there will be some differences in configurations because of the different hardware.
4. **Do you encounter any performance issues or delay when opening applications?**  
Answer: Yes there are some issues, and some bad performance when using Windows. Both compatibility and performance issues and minimum resource requirements should be suggested in the report.

## G.20.3 Weekly report

### Active Directory

There have been some problems connecting to the client's Azure Active Directory. The client has tried to change some settings, but this did not help. We sent them an email today but no answer yet.

### VPN

Most VPNs work the same way, and we have no requirements to the VPN other than to test if VPN works with Qubes. We chose OpenVPN because it is easy to set up servers that can be used for testing and because we were most familiar with OpenVPN. We find no trustworthy source to reference why we think OpenVPN is secure.

Response from supervisor: The group should have some reasons for the decision. This could be:

Tool popularity and group members familiar with the tool. The report should also show readers that there are other tools available, and briefly mention the most popular ones.

### Splunk

The client uses Splunk so it is natural that we choose this tool in our project. The reason behind the choice should be given in the report. Not sure if the client wants this information to be exposed in the report.

Response from supervisor: We will discuss what to do after we have an answer from the client. There are different ways to address this issue.

### Schedule

About one week behind schedule. Planned to be done with configurations before easter, so we have some buffer time.

### Thesis

We are taking notes and writing in the thesis along the way. For some parts of the thesis, the person working with the technology is the one writing about it in the thesis.

### Questions from the supervisor

1. **How much testing do you still need in the rest of the project?**

Answer: The big things that are left are SaltStack, geolocation and 2 factor authentication.

2. **How will you show the testing and configurations in the report?**

Answer: Code snippets, SaltStack config files, descriptions of methodology and results from testing along the way.

## Other

The group will invite the supervisor to the bitbucket repository, and also send the PDF (because some special tools are needed to build the PDF).

### G.20.4 Regarding presentation

May or may not be digital presentation. Prepare for both. Make sure to practice live demonstrations beforehand and have video recording as backup. Find out how to make the presentation smooth and make sense, how can we present in an understandable way? Make figures, give a high-level introduction to Qubes (Example: Start by showing the application menu, not the qube manager). Everyone should wear headphones with microphone.

## G.21 March 17: Client meeting

**Date** 2021-03-17

**Time** 14:30

**Client participants** David Lee Andersen, Erlend Hammer

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.21.1 Splunk

The group has permission from the client to write in the thesis that the client uses Splunk. This can make it easier to explain why Splunk is used in the project.

The client has had an earlier assessment of different SIEMs where the Splunk was gave the best results. In regard to rules, alarms, security etc. Splunk is in a different league then the other SIEMs that were assessed, but it is more expensive. This assessment is not public, but the client will find out if there is some of the information that could be shared with the group.

Isolated for the simple solution that the project group is working on, Splunk might be a bit much. It would be nice if the group can also write about what they would have chosen. Both for extracting the logs and also an overall assessment of what should be logged and what we can get out of it from a more tactical perspective.

Splunk apps we could have a look at: Splunk TA, Windows event code security analysis.

The client will send an email with a google drive link with some files from Splunk workshops.



## G.21.2 VPN

- The group has tested that it is possible to connect to a VPN client and ping it.
- Is there any solution that VPN should be tested against?
- The client prefers solutions with more framework than OpenVPN. Maybe try Cisco AnyConnect that NTNU uses.
- Can test with WireGuard. It does not support some things like DHCP and username/password but use a public/private key. If this key ends up in the wrong hands there will be no indication of what machine it is used from, except for IP address.
- How can we be 100% sure that the person who uses the key is the right person, and ensure the integrity of the key? How to uncover that some one retrieves the router VM and starts it on a different machine?
- Could use a rolling key that could be rolled out with Puppet/Ansible/SaltStack. Or use some technology that makes the user authenticate after connecting to the VPN (zero trust). Cannot be done automatically because this could be replicated by cloning the disk. Could use a 2-factor authentication with phone, YubiKey or code generator. This could be configured in the proxy VM.
- At least the group can write about WireGuard and how this could be a solution. Can also write about other solutions, what challenges the solutions have and why it will or will not work for our project.

## G.21.3 RAM

- The VPN VMs run on debian with no graphical interface, so they do not need that much RAM. Can use for example iperf3 to check the RAM usage and include it in the report.
- Let the client know if more RAM is needed for the testing.
- Find out the physical memory of the laptops and send the client an email about it.

## G.22 March 24: Supervisor meeting

**Date** 2021-03-24

**Time** 14:00

**Supervisor** Jia-Chun Lin

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

Digital meeting on Teams, weekly supervisor meeting with feedback. The group presents the project progress from the previous week and plans for the following week.

### G.22.1 Weekly report

#### Achievements

- Splunk is written about in the thesis.
- Connected to Active directory (but cannot join the VM).
- There are some limitations with screen sharing. The workaround is using a capture card and sending the signal back into the VM. This only works with USB access that the Windows VM in Qubes

does not yet have. It is possible to screenshare from within a Windows VM, but then you cannot share anything outside of that one VM you are sharing from.

- The group will advise the client not to bother with it until the GUI VM is finished by the Qubes team.

## Problems

Salt stack and admin VMs allow some of the VMs in Qubes to make other VMs. One of the machines in the Qubes laptop will control some of the other VMs and give them the correct configurations. When doing this, rules have to be included in the Qubes policy system: In dom0 there is a folder holding files that refer to specific actions that other VMs can ask for permission to take (for example start another VM etc.). In order to allow one machine to create others, it should be possible to add some lines in the policies, but currently there are some problems doing this.

Connecting the Windows VM to AD works, but when trying to log in it does not work, and only gives a spinning wheel.

Advice: The group should try their best to fix the problems. If it does not work explain why, and write what solutions were tried and what does not work. The settings on the client's end, it is out of the control of the group, so the attempts to connect the VMs should be documented, and the unsolved issues should be explained. Mentioning what the client did is not necessary, but the connection attempt and result should be documented.

## Discussion topics

- We are using a lot of terminology in the report, where does the limit go for what we should define?
- Your report is a bit special. if it is not related to our thesis, we don't have to explain, ex. don't have to explain raspberry pi, have to explain active directory, vm, hypervisor, vpn, SIEM etc.
- Terminologies and abbreviations that we talk about in our report should be defined and described, but terms that are a part of that description does not need to be recursively defined.
- If you think the term is not very common, then you need to explain it. Firewall is common, IP address is common. VM, hypervisor, splunk are not common. Will read through and say if there is something we need to explain.

## Plans for next week

- Work on SaltStack.
- Work on finishing the identity management.
- Write about windows integration.
- Report writing and structuring.

The 1st thesis draft deadline is the 1st of April. Upload it to the Teams folder and send a message.

## G.22.2 Progress

- Azure AD can be used for MFA for the windows VM, and hopefully this might be done very soon. MFA with YubiKey is not done.
- SaltStack is ongoing.
- In order to do the risk assessment, the group needs an understanding of the whole platform, so this should be done last. It is definitely doable with the time that is left.
- Platform evaluation is not done, but found out what machines work and definitely do not work, but not yet the specific ram etc.
- The task about “VM storage encryption” has shown to be the same task as “separation of VM and host”. Qubes is designed to separate this.
- Approximately one week behind schedule.

## G.22.3 Other

- The group has invited the supervisor to the git repository, and the report is uploaded to Teams.
- If the group wants feedback on a specific thing, upload the PDF file in teams and send a message to the supervisor and she will take a look.
- No meeting next week in the easter. The meeting on 7. April, will be spent going through the feedback on the thesis first draft.
- Could have a “test” presentation some time near the project end where the supervisor can give suggestions on the presentation.

## G.23 March 31: Client meeting

**Date** 2021-03-31

**Time** 10:00

**Client participants** David Lee Andersen

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.23.1 Active Directory

The group have managed to join the Windows VM to AD.

#### Questions from the project group

1. **How many VMs can we join to AD?**

Answer: Unsure.

## G.23.2 Splunk

The client got help from an external company to decide on Splunk as their SIEM, but they cannot share the report. If seen as an isolated case in our project, other solutions might be better and cheaper. But the client's complexity makes Splunk worth it for them. The project group should write that there are alternatives, but that the client has chosen Splunk.

## G.23.3 Progress

### Must have

#### *Finished:*

- Base configuration
- Firewall rules and networking
- Investigate separation of VM and host
- Ship host logs to SIEM

#### *Not started:*

- Risk assessment

### Nice to have

#### *Finished:*

- Connection to identity service

#### *Ongoing:*

- Investigate possibility of central management - This is possible. Ongoing proof of concept in Salt-Stack.
- Computer configuration will be written about in the report.

## G.24 April 7: Supervisor meeting

**Date** 2021-04-07

**Time** 14:00

**Supervisor** Jia-Chun Lin

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

Digital meeting on Teams, weekly supervisor meeting with feedback. The meeting consists of going through the thesis first draft feedback.

### G.24.1 General comments

- Summary is very different from other groups.
- The introduction is overall very good.
- Check that paragraphs are not too short or too long.
- Find some pictures to describe type 1 and type 2 hypervisor.

### G.24.2 Citations/references

- After mentioning an organization etc. put the official website as citation. All footnotes should be moved to in-text citations. In the corresponding reference a format needs to be followed with title, date, etc. Can use IEEE format for example.
- No need to provide page numbers when different chapters are mentioned in the text. But provide a hyperlink to the chapter. No need to provide the name of the chapter, just the number. For example: “explained in detail in chapter 4”.
- If an abbreviation is not used immediately after, it is better to just use the whole name and not the abbreviation. Use the abbreviation when it is mentioned again right after. If it is much text between the first mention of the name and abbreviation, the reader might forget what it meant.
- There are two ways of writing sentences that appear in another article. One way is to use quotations, but this is not the best way. A more appropriate way is to rewrite the original sentence into your own.
- The same citation can be used two times in a row if there is some text between where it is used. But for example “WannaCry ransomware” should be cited with what it is, for example by the Wikipedia page.
- “In more details below” it would be better to specify where.

### G.24.3 Related work

This is meant to describe competing methods that are trying to do the same thing. Mention how people are doing the same thing with different approaches. There may be too few related works to have this chapter. Investigation of the security of Qubes fits more background work than related work. Privacy platform for journalists might be related work, could write that their purpose is to configure Qubes for better privacy, however we are targeting to provide secure separate zones.

But if there is not much to mention here, this chapter should not be here. If the size of the chapter is not too short, it can be kept. If the chapter is removed, there is no need to mention why and the examiner will not reduce the grades because of this.

### G.24.4 Requirements specification

The requirements that the client gave are very “loose”. They could not be so specific, because the project is based on a lot of investigation of different solutions to find out what works and if it works at all. The client wants to see if it is possible or not. The requirements were not to make it work, but to see if it works or not.

In this case the group could skip the requirements specification chapter. However, clearly mention what the client wishes the final platform to be. This should be written in the goals chapter. Write the goals in more detail, could be bullet points with goals and sub-goals. Summarize the vision the client has.

## G.24.5 Technical design

- Before introducing networking etc, the overall architecture of the platform should be provided. Can for example use the picture for type 1 hypervisor to make a picture describing the platform.
- Some descriptions for the illustrations in the networking chapter should be more specific. The colours should be described more clearly. Figure 1.5 should be right after the chapter 1.5 text, and figure 5.2 should be right after the chapter 5.1.1 text.
- Pictures should always be right after the corresponding text.
- The specifications of the laptops should be mentioned.
- Networking could be its own chapter. Don't need technical design chapter.

## G.24.6 Implementation

- Investigating hypervisors could be moved to its own chapter because it is quite long. Should be chapter 4 before technical design (networking). Clearly mention in the thesis structure why we use this set-up.
- Rewrite the Qubes section.
- Spend more space describing Qubes OS after chapter 4 (investigation). Mention the details of Qubes OS, what it is, and how it looks by default. How many domains are created as default, and how does it look before it was used in this project.
- SIEM could be an independent chapter because it is very long.
- Configuration management could be an independent chapter if it becomes very long.
- Windows in Qubes OS could be an independent chapter.
- IAM could be an independent chapter.
- Clarify the difference between the OpenVPN and PiVPN. Should not have a subsection for each of these because it is quite short. Implementation for VPN could be integrated with the networking chapter.
- No need to clarify that these new chapters are a part of the implementation stage, "implementation" does not need to be its own title. Can say something like "in the following chapters we will address each of the solutions". Better to create a separate chapter for each topic where the topic is introduced, and the implementation is shown.

## G.24.7 Development process

This chapter can be after implementation chapters. This should be how the group performed this project as a group. Should mention how the group worked together, what tools were used to communicate (not in the same location due to covid). Can mention if some things did not work in the group. Example: Meetings, time management, development method (scrumban) etc.

## G.25 April 7: Weekly scrum meeting

Week 14

**Group participants** Marius Ødegård Lindvall, Steinar Vrenne, Martin Wahl, May-Liss Amundsen

### G.25.1 Tasks for the week

- Improve the report and make changes after receiving feedback.
- SaltStack
- Report writing

