Anssi Ylätalo

DEVELOPMENT OF PROCESS AND TOOLS FOR VULNERABILITY MANAGEMENT

Master's thesis
Master's Degree Programme in Cybersecurity

2019

South-Eastern Finland
University of Applied Sciences

South-Eastern Finland
University of Applied Sciences

Author (authors)	**Degree**	**Time**
Anssi Ylätalo	Master's Degree Programme in Cybersecurity	November 2019
Thesis title		
Development of process and tools for vulnerability management		49 pages 9 pages of appendices

Commissioned by

Atos IT Solutions and Services Oy

Supervisor

Vesa Kankare (Senior Lecturer / XAMK), Markku Roiha (Head of Operations / Atos)

Abstract

The primary objective of this thesis was to improve vulnerability management within cybersecurity domain by defining an unambiguous process to handle findings causing threat to commissioner’s local ICT infrastructure. Also, supporting tools were to be studied and implemented or suggested. The mentionable secondary objectives were to improve the situational awareness by giving visibility to a security posture, and to enhance the quality of asset management information.

The study was carried out as a research-assisted development project using the constructive project model where the strengths of a traditional linear project model and a spiral project model have been combined. The constructive development project emphasises interaction, participation and pedagogical way of working as primary methodology. The most important data gathering methods were observation, participation, documentation intake and online meetings. The research work was done on top of authors daily work, in co-operation with key resources from commissioner’s organization.

The primary result of the research was a definition and an implementation of a vulnerability management process, including process key roles, responsibilities, tasks and KPIs. The process was streamlined for efficiency and simplicity. The secondary result was an implementation of a virtual vulnerability scanner infrastructure and a build of a dashboard for vulnerability information on an existing log management system.

Vulnerability management is a key component in planning and implementing security controls and executing a risk assessment. The study suggests that having a formal process and proper tooling in place would improve risk management of an organisation, daily work efficiency and quality, situational awareness, and enhance asset data quality. The study showed that proper asset management is in a key role to execute the vulnerability management process successfully. Also, the study discovered that the process could be used for the remediation of deviations found by some other activities, like a compliancy scan. The improvement of tooling should continue with a SIEM system implementation on top of an existing log management system, and by enabling a credentialed vulnerability scanning.

Keywords
vulnerability management, risk assessment, cybersecurity, process, development project

CONTENTS

APPENDICES

ABBREVIATIONS

API	Application Programming Interface
BPMN	Business Process Modelling and Notation
CERT	Computer Emergency Response Team
CMDB	Configuration Management Database
CLI	Command Line Interface
CVSS	Common Vulnerability Scoring System
GCE	Greenbone Community Edition
GUI	Graphical User Interface
ICT	Information and Communication Technology
ITIL	Information Technology Infrastructure Library
ITSM	Information Technology Service Management
KPI	Key Performance Indicator
NCSC	National Cyber Security Centre
NVT	Network Vulnerability Test
SCAP	Security Content Automation Protocol
SEM	Security Event Management
SIEM	Security Information and Event Management
SIM	Security Information Management
SOAR	Security Orchestrations, Automation and Response
TCP	Transmission Control Protocol
TRR	Timely Remediation Rate
UDP	User Datagram Protocol
WIR	Weighted Intrusion Rate

1 INTRODUCTION

This thesis was made to improve local cybersecurity vulnerability management within commissioner’s organization. The vulnerability management in cybersecurity domain is not a new topic although it has nowadays become even more critical function as external and internal threats are constantly developing and evolving. The reasoning of vulnerability management is obvious. Exploitation of vulnerabilities by a threat introduces a risk to organization which vulnerability management attempts to mitigate. Thus, vulnerability management can be understood as a part of risk management. In bibliography, there are quite many guidelines and best practices how to implement the vulnerability management process per se. However, as vulnerability management has close relationship to risk management, their co-operation could be emphasized even more. Also, it is important to understand that vulnerability management is a business-driven process requiring skilled people, not just technology.

In this thesis, local vulnerability management has two meanings. Firstly, it can be understood as a local extension of commissioner’s global vulnerability management process which outputs the results of a global vulnerability scan to the local organization for remediation. Secondly, it refers to a process of handling the vulnerabilities of local ICT infrastructure which are out of the scope of commissioner’s global vulnerability management process and tools. Local ICT infrastructure includes assets, such as physical and virtual servers which have a management function, or which provide some other internal service. Also, virtualization platforms, network devices and firewalls are part of local ICT infrastructure. However, assets directly assigned to, or owned by a customer are excluded as they are handled either by the global vulnerability management process, or by a customer dedicated vulnerability management service.

While a vulnerability management process could have many external threat and vulnerability intelligence sources, like CERT and NCSC feeds, the focus of this thesis is in a handling of the vulnerabilities found by network scanning. Results of this thesis could be used to implement or enhance the process for compliancy management as well.

2 RESEARCH PROBLEM

In the following chapter, the research problem and the research questions of the thesis will be presented. Also, thesis objectives are discussed, and what secondary benefits should be attained by resolving the research problem.

2.1 Research questions

Lack of formal vulnerability management process lowers, or even hinders, understanding the security posture of an organization. Limited visibility introduces a risk of vulnerability exploitation and potential data breach. Regarding the commissioner's initial situation, the visibility was limited over the vulnerability status of assets which are out of scope of commissioner's global vulnerability management scanning. This was due to fact that no regular vulnerability scanning was in place, only random scans were done an on-demand basis. In addition, local processing of vulnerabilities found by the global vulnerability scan was handled with an informal process which resulted in problems, such as evaded responsibilities, and difficulties with the follow-up of remediation. These discovered problems can be presented as a single statement:

> “There is no unambiguous process in place for local vulnerability management.”

This statement would be the research problem of the thesis. To identify problem reasons, and to be able to eliminate those, the research problem has been transformed to following research questions:

- Why is a clear vulnerability management process important?
- What are pre-requisites for a vulnerability management process?
- What are key roles in a vulnerability management process?
- What are the main phases in a vulnerability management process?
- What are interfaces to other processes?
- What tools support the process best?

Finding answers to these questions, and thus eliminating the research problem, should lead to a successful future mode of operation.

2.2 Research objectives

The primary objective of the thesis can be divided into two topics, which are the process and the tooling. The first topic includes a design and an implementation of a process for local vulnerability management. The second topic involves the selection and implementation of proper tooling to support the process, and for local vulnerability scanning. Some pre-requisites and guidelines were given to process development:

- The process should be aligned with commissioner's global vulnerability management process, and with other interfaced processes.
- The process should be benchmarked with agreed KPIs.
- The process should be transferable to another country or business unit with minimal effort.

The secondary objectives, which are natural outcomes of reaching the primary objective, can be listed as follows:

- A good situational awareness is a key factor for successful decision making. Situational awareness should be improved as vulnerabilities would be reported frequently giving the management better visibility to security posture.
- Conducting tasks inside an organization without formal procedures can be inefficient and prone to errors. Co-operation within an organization should be enhanced once roles and responsibilities are clearly defined, and expectations are made clear.
- The up-to-date asset information is a key pre-requisite for the most ITIL-based processes. The requirement is emphasised in vulnerability management context as the asset owners will have a major role in a process. Asset information should be up-to-date after system discovery.
- Potential optimization and quality improvement of a patching process as it might be more focused and precise when driven by a vulnerability management process.

3 RESEARCH FRAMEWORK

In this chapter, the research methodology of the thesis will be discussed starting with a relationship between traditional research and research assisted development projects, continuing with a development project methodology and models. Finally, the chapter is concluded by addressing results and report importance.

3.1 Research assisted development

There are differences in a functional and a traditional research thesis. Salonen (2013, 5-6) has summarized them into three essential topics. Firstly, in a functional thesis, the outcome is a product while a research thesis usually introduces new information. Secondly, in a functional thesis, the other actors are involved and contributing in different phases of a project. In a research thesis, the fundamental actor is the researcher himself and the other actors are usually involved as a data source only. Thirdly, the phases of thesis work within a functional thesis are progressing with dialog and interaction between actors. This enables possibility for discussion, evaluation, feedback, and even change the course of the project.

In a research thesis, an interaction is mostly one-way data collection and exchange. An exception to this is an action research. However, a traditional research methodology cannot be applied directly to a functional thesis as a traditional research follows the strict rules and guidelines of academic frameworks and tradition. There are, nonetheless, many usable characteristics in traditional action and design research. In action research, researcher aims for a change as a participant and the realization of change is ensured. In design research, however, an active participation is not required, likewise the change does not necessarily take place (Kananen 2015, 57).

3.2 Data gathering

There are differences how the data is produced, and what are its criteria. In research assisted development, the production of information is guided by practical problems and questions. Information is produced in a real-life environment with the assistance of research methods (Toikko & Rantanen

2009, 22). Traditional research considers reliability and accuracy as information criteria. In a development project, the usability and practicality of information are more important (Sb. 156).

In development projects, the data gathering methods are mainly the same as research methodology within traditional research. However, in development project methodology is more flexible (Salonen 2013, 23). The most important data gathering methods in the thesis are observation, participation, documentation intake and online meetings. The sources would be commissioner’s existing documentation and key personnel. Also, a thesis diary will be used to record the progress of the project. On the implementation phase, literature would be reviewed for best practices, and to identify coherent patterns in existing implementations. The project participants and stakeholders are chosen key resources from commissioner’s organization who have either operative or management interest to vulnerability management.

3.3 Results

On the question of the results, Metsämuuronen (2001, 64) implies that using Denzin’s (1988) the triangulation method will result in more accurate and reliable information. In triangulation, the same problem is approached from different points of view using multiple research methods. The results of the thesis project will be analysed with a multi method approach. Firstly, the product of a project, a vulnerability management process will be analysed from each process role perspective with a simple before-after analysis. Also, the management point of view will be analysed. This analysis should identify actual changes, and all pros and cons of a process on a theoretical level.

In addition to investigating concrete changes, user experience could be investigated as well (Toikko & Rantanen 2009, 153). To evaluate the effect of a process, an anonymous survey will be set up for all participants to indicate their expectations and experience on the process. The results will be analysed to complement the results of concrete change analysis. Finally, the results of both sources will be cross analysed to find any correlation between changes and user experiences.

3.4 Report

One of the most important parts of a thesis is the report. According to Salonen (2013, 12), one of the biggest differences between research and development project is that project work is typically planned, implemented, and evaluated without a strict source-based system of concepts. Naturally, a project work has language, concepts and vocabulary of its own, but an extensive coverage of those with literature sources in a final report is not mandatory like in a traditional research (said book p. 12). The thesis research is implemented as a research assisted development project which emphasizes the development aspect. Therefore, from the results usability perspective of development activities, it is not mandatory to report a development project according to scientific research criteria. Nevertheless, a report of the development project should be more than an outcome of the project. The report should represent understanding about the development project in general, application to given area, and thesis writer’s personal erudition (Sb. p. 25).

4 THEORETICAL FRAMEWORK

In the following chapter, the key concepts and general theories related to the thesis will be presented. The concepts related to vulnerabilities and process development are covered in a comprehensive manner as they are the fundamentals for the research.

4.1 Key concepts

4.1.1 Vulnerability

Vulnerability can be defined in several ways depending on the viewpoint. NIST (2012, 2-9) defines vulnerability as *weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source*. Abernathy & McMillian (2016, 77) also have a control perspective by defining vulnerability as *a weak or absent countermeasure*. Vulnerabilities in a cybersecurity domain are mostly resulted from software bugs or programming errors, misconfigurations, or intended features. Foreman (2010, 61) has recognised this by a definition stating that vulnerability is *weakness in the software or hardware that allows the use of the product beyond its design intent with an adverse effect on the software, system or data*.

4.1.2 Vulnerability management

Vulnerability management, VM for short, is a process, not a technology. (Foreman 2010, 181.)

There is a tight connection between vulnerability management and risk management. Foreman (2010, 205) suggests that vulnerability management is a segment of risk management by supplying the key information to the risk assessment process. Figure 1 presents how the vulnerability management is positioned to the risk management process.

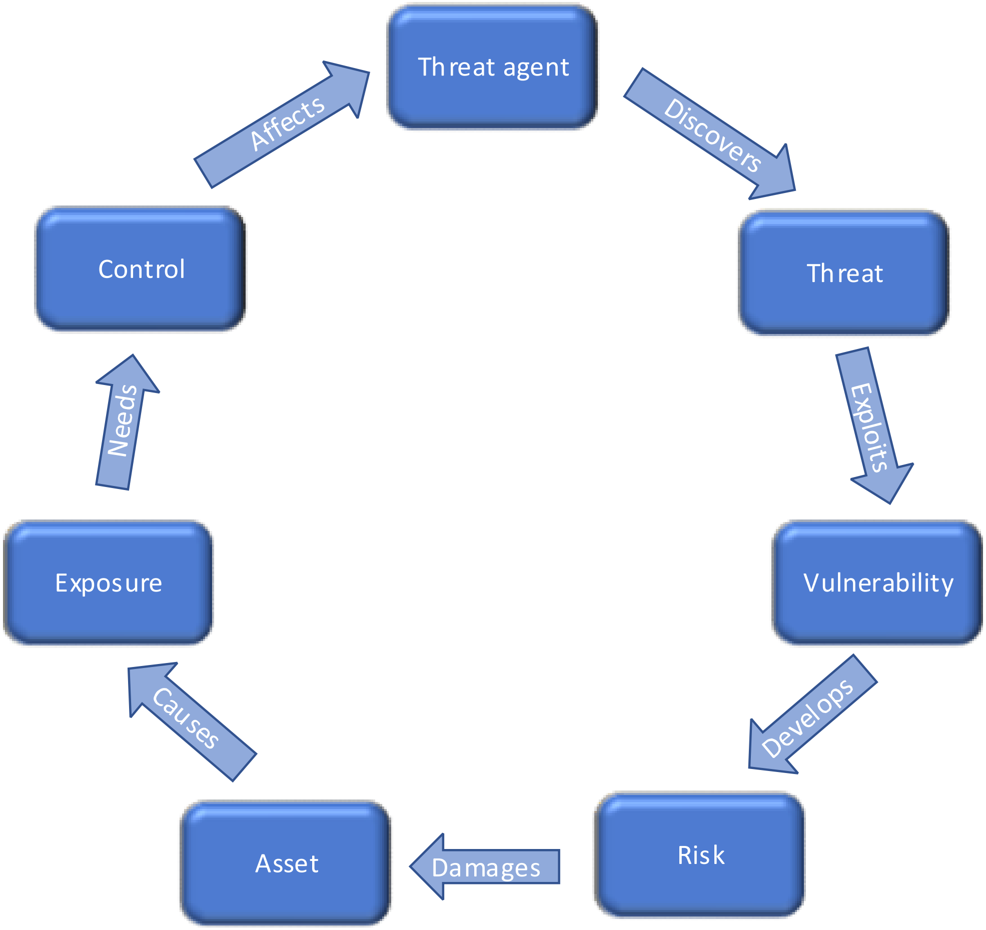


Figure 1. Security concept cycle (Abernathy & McMillian 2016)

A threat is introduced when vulnerability has been identified or exploited. The actor who identifies or exploits the vulnerability is called a threat agent. The amount of risk introduced by the vulnerability is dependent on the probability the threat agent will exploit the vulnerability and the impact of such event. The risk causes an asset being exposed to losses. To remediate such an exposure, the security controls are in place to provide a countermeasure against the threat agent. (Abernathy & McMillian 2016, 76-78.)

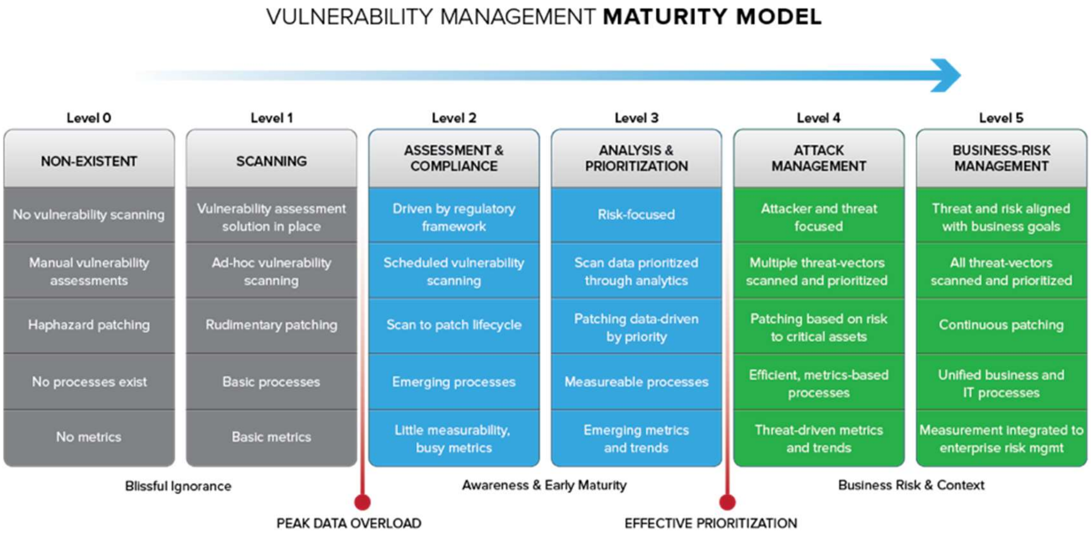
In general, vulnerability management and risk management do not only concentrate on ICT related assets or threats. Abernathy & McMillian (2016, 82-83) classify the threat agents in six categories; human, natural, technical, physical, environmental and operational. As far as vulnerabilities are concerned, Foreman (2010, 2) points out that those might exists also in strategies, economics, business processes, and supply chains. In Carnegie Mellon University's (2016, 4) model, assets are divided into four categories; people, information, technology and facilities. All these categories have different characteristics and vulnerabilities. Thus, the low-level vulnerability management process implementations can vary significantly. However, high-level processes are usually presented in a general level which makes them applicable to all domains. This thesis, however, concentrates on vulnerability management of ICT infrastructure assets belonging to a technology, that is, a cybersecurity domain where the most threat actors fall in a human, a technical or an operational category.

The vulnerability management process is most commonly described as a cyclic sequence of tasks where vulnerabilities are identified, and the risk caused by vulnerabilities is evaluated (Palmaers 2012, 2). Based on the risk evaluation, the vulnerabilities, or risks, should be addressed by one of the actions presented by Foreman (2010 ,1) and Abernathy & McMillian (2016, 85-86):

- Avoidance which prevents risk from taking place.
- Reduction where the effect of risk would decrease.
- Transfer where the risk is passed to a 3rd party.
- Acceptance where situation is left as is.

The actual risk response action would be dependent on the cost and organisations practical level of risk acceptance (Kohnke et al. 2016, 186-187).

The ability of an organisation to execute the vulnerability management can be also evaluated with a maturity model. Figure 2 shows a six-level model presented by Shanks (2015, 2), originally created by Core Security (2014, 3) as *The Threat and Vulnerability Management Maturity Model*. The model is based a traditional Carnegie Mellon Maturity model (Core Security 2014, 3).



Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
NON-EXISTENT	SCANNING	ASSESSMENT & COMPLIANCE	ANALYSIS & PRIORITIZATION	ATTACK MANAGEMENT	BUSINESS-RISK MANAGEMENT
No vulnerability scanning	Vulnerability assessment solution in place	Driven by regulatory framework	Risk-focused	Attacker and threat focused	Threat and risk aligned with business goals
Manual vulnerability assessments	Ad-hoc vulnerability scanning	Scheduled vulnerability scanning	Scan data prioritized through analytics	Multiple threat-vectors scanned and prioritized	All threat-vectors scanned and prioritized
Haphazard patching	Rudimentary patching	Scan to patch lifecycle	Patching data-driven by priority	Patching based on risk to critical assets	Continuous patching
No processes exist	Basic processes	Emerging processes	Measureable processes	Efficient, metrics-based processes	Unified business and IT processes
No metrics	Basic metrics	Little measurability, busy metrics	Emerging metrics and trends	Threat-driven metrics and trends	Measurement integrated to enterprise risk mgmt

Figure 2. Vulnerability management maturity model (Core Security 2014, cited in Shanks 2015)

The model could be interpreted that on the level zero and level one, no clear-cut vulnerability management program is in place. On level zero, no real actions are taking place other than some random tasks every now and then. On the first level, some vulnerability scanning, and remediation activities are occasionally done relying the existing ITIL processes. However, there is no systematic way of working to handle the findings as a whole. The model also suggests that it would be very hard to go above the level one without proper vulnerability management because the amount of data would simply become very difficult to be handled effectively. On the second level, the organisation has recognised the importance of vulnerability management and begins to introduce formal procedures. However, the process is not a risk- but a compliancy-driven, limiting the scanning and remediation to the scope of a regulation at hand. At this point, the organisation might also suffer from data exhaustion caused by inability for effective remediation activities. On level three, the process has become a risk-driven where the remediation actions are prioritised according to the criticality of assets and severity of vulnerabilities. Also, the metrics and KPIs would start to produce useful information regarding the process effectiveness. On the levels four and five, the process would turn even more threat, risk and asset criticality focused. The ultimate goal is to integrate a business strategy into the program by focusing entirely on the risk to the business and providing early warnings to the organization when threats and vulnerabilities pose risk to business performance. (Core Security 2014, 3-6.)

In general, the vulnerability management maturity model suggests that the vulnerability management capability of organisation will increase over time once an appropriate vulnerability management program is in place, the program is improved, and that the program will be executed in a rigorous manner. To achieve high maturity levels, an automation would be recommended to ensure consistency and task completion assurance. (Shanks 2015, 1.)

As discussed earlier, vulnerability management is a key component in planning and implementing adequate security controls and executing a risk assessment. Exploitation of vulnerability by a threat results in a risk to the organization. An effective vulnerability management process reduces the amount of vulnerabilities, thus, reducing risk to the organization (Carnegie Mellon University 2016, 5). In other words, vulnerability management would help organizations to understand its weaknesses, and take appropriate actions based on correct information.

4.1.3 Vulnerability scanning

Vulnerability scanning is a process where special software is used to discover vulnerabilities or evaluate the security configuration of a network, a computing system or an application (Foreman 2010, 99; Center of Internet Security 2018, 12). A vulnerability scanning system can be used to centralize and automate the continual testing process for vulnerabilities (Abernathy & McMillian 2016, 522). The scanning software can be installed on top of the most operating systems, or it can be used as a software or hardware appliance where a vendor provides a turn-key solution for the scanner installation and maintenance (Foreman 2010, 64; Greenbone Networks 2018b, 4-5). A vulnerability scanner can also be provided as a SaaS service from a vendor or from a public cloud provider, like the most modern web application vulnerability scanners (OWASP Foundation 2019). Alternatively, an agent-based framework could be deployed where a small piece of software would be installed on each device to perform the vulnerability scan and possibly other security related tasks as well (Foreman 2010, 66-70).

The vulnerability scanning can be divided into two main types which are a credentialed scan and a non-credentialed scan. Those are synonyms for an authenticated and an unauthenticated scan respectively (BeyondTrust 2019). Foreman (2010, 86) refers them also as a black or white box testing. The non-credentialed scan examines the targets by probing them from the network side, thus, revealing their weaknesses and flaws from the attacker point of view. The credentialed scan uses the pre-configured privileged credentials to log in to the target and execute vulnerability tests inside the target. In most cases, this would result in a more detailed and valuable information about the configuration and potential weaknesses. In general, vulnerability found by a non-credentialed scan represents a greater threat than if the same vulnerability was found by a credentialed scan. This is because in the latter case an attacker would have to gain access to the target system prior to be able to exploit the vulnerability. (Sb.)

Another method of categorizing the ways of vulnerability scanning is based on the use case. An external scan would examine the services exposed to the Internet, like company’s website or some extranet service for customers. When the scanning is targeted to the services inside a corporate network, the scan would be classified as an internal scan. The third use case would be an environmental vulnerability scan. These scans are based of the specific technology environment where the company operates, for example mobile or IoT devices. (BeyondTrust 2019; RedLegg 2019.)

Figure 3 shows the typical steps performed by the scanner in a non-credentialed scan. After the scan configuration has been loaded, a host discovery is performed to find targets that are alive, that is, ones that are responding to requests in the first place. Then, a port scan would be initiated against each alive target to find all TCP and UDP ports that have an application listening. Once all listening ports have been identified, the scanner connects to them and tries to detect the application and all the information related to it. These findings are used for vulnerability analysis on the last step. The scanner also tries to identify the operating system of the target. This would be also a valuable piece of information in the last step which is the vulnerability analysis. (Infosec 2019.)

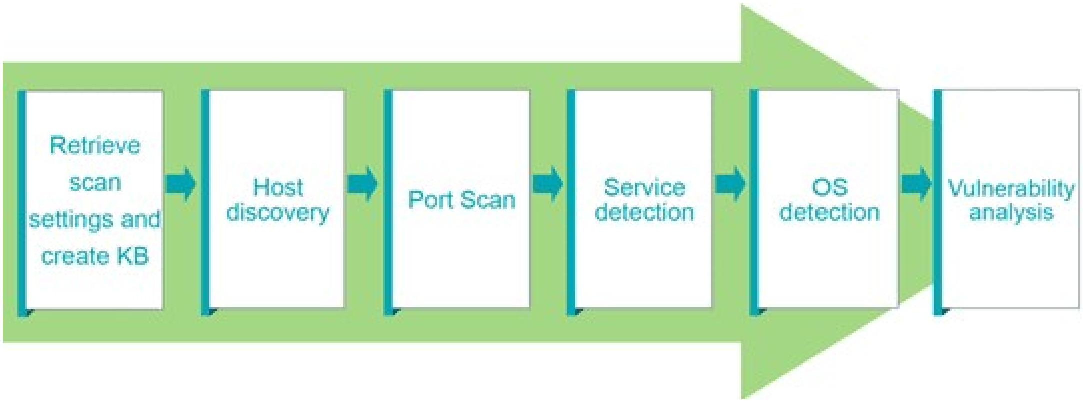


Figure 3. Nessus non-credentialed scan steps (Infosec 2019).

Also, it is important to understand that a vulnerability scan is a snapshot of the environment at that time. The situation might change quickly if new devices would be introduced, new application would be installed, or even just by a configuration change. Thus, it is very important to schedule regular scanning to reduce the exposure time (Palmaers 2012, 4).

The vulnerability scanning should be differentiated from the penetration testing due to fact that a non-intrusive vulnerability scan should only discover and report the vulnerabilities, whereas a penetration test also tries to exploit them in order to break into a target system (Foreman 2010, 308-309). However, a vulnerability scan can also try to exploit the findings to highlight the risk and impact. This is known as an intrusive scanning. This method, however, should be used with caution as it might cause disruption to the targeted services (RedLegg 2019).

4.1.4 SIEM

Security Information and Event Management, SIEM, combines two areas; Security Information Management (SIM) which makes the log collection, analysis and reporting; and Security Event Management conducting the monitoring, alerting and correlation (imperva 2019). Also, advanced SIEM systems are able to categorize events automatically, thus improving normalisation, analysis and correlation significantly (Abernathy & McMillian 2016, 462). According to Gartner (2019a), the essential requirements for a SIEM system would be extensive log and event management capabilities, log analysis and correlation capabilities, incident management, and reporting.

As discussed above, the correlation has been identified as one of the most important and powerful features of a modern SIEM system. Traditionally, the correlation has been based on rules which compare incoming events with predefined relationships between entities to identify anomalies. The rules are groups of conditions which would raise an alert once they are met. Creating such rules would require an existing knowledge about the steps that should be detected. Thus, such rule-based detection is becoming inefficient as the attacks are converting to more complex and advanced. The best use cases for the rule-based detection are a real-time monitoring of known threats, a compliancy monitoring and a signature-based threat detection. (Exabeam, 2019.)

The next generation SIEM systems introduce a capability known as modelling which allows to analyse user’s, or asset’s behaviour and to add context in the analysis. The additional context allows not only to determine what users or assets are doing, but also to identify themselves, their roles, and determine their normal actions. In other words, the models are used to detect deviations from the baseline behaviour. The best use cases for the models are to detect the behaviour-based anomalies, data exfiltration attempts, zero-day threats and lateral movement. (Sb.)

Nowadays, SIEM solutions might be complemented by a Security Orchestration, Automation and Response (SOAR) solution. Gartner (2019b) defines the SOAR as a digital workflow format enabling organizations to collect inputs monitored by the security operations team and to define incident analysis and response procedures. In other words, a SOAR solution would enable security teams to automatically gather the context for alerts, and automatically or manually respond to alerts by utilising the best suiting playbooks or such against the threat. At its best, a SOAR system will provide a full security incident response lifecycle beginning with detection and qualification, continuing with triage, escalation, enrichment and containment, and finally completing the cycle with remediation. Also, one of the key benefits of a SOAR system is to reduce the resolution time of the breach. Used on top of a SIEM solution, a SOAR system would also improve the incident response by automating and orchestrating the routine and tedious tasks. This would free valuable human resources to more productive tasks. (Tillyard, 2019.)

4.1.5 Asset

ISO 55000:2014 (2014) defines an asset as *an item, a thing or an entity that has potential or actual value to an organization*. In ICT or cybersecurity context, an asset is a piece of software or hardware which should be continuously inventoried, tracked and corrected to make sure that only authorised assets exist (Center of Internet Security 2018, 6-8). As this thesis is referring to cybersecurity vulnerability management context, this would mean all devices that are in the scope of vulnerability scan.

4.1.6 Process

Fundamentally, a process can be defined as a chain of inter-connected events (Tuurala, 2010). In industrial engineering and management, a process consumes resources to create added value for clients (Martinsuo & Blomqvist 2010, 1). Davenport (1992) defines a process as a structure of action which can be described as *a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs.* Essential characteristics of process-like thinking are: A systematic way of thinking, customer focus, objective orientation, concentration on added value, and using feedback to systematically improve the process (Martinsuo & Blomqvist 2010, 3).

Processes within a company can be categorized as core processes and supporting processes. Core processes always have a connection to an external client, whereas supporting processes are internal processes supporting the core processes. Another way of a process categorization is in dividing processes in the main processes and subprocesses which constitute a hierarchy with different levels (said book p. 4). The objective of the thesis is to create an internal process that could be executed independently, and as a sub-process that supports commissioner’s global vulnerability management process and risk assessment in general.

Business Process Model and Notation (BPMN) is a de-facto standard of process modelling which provides a notation and model for business processes, and an interchange format. The latest version has been formally published by ISO as the 2013 edition standard: ISO/IEC 19510. (Trisotech, 2019.)

4.2 General theory

4.2.1 Vulnerability management strategy

Development of a vulnerability management process should start by defining a strategy where scope, goals, objectives, and priorities of vulnerability management function are defined (Carnegie Mellon University 2016, 4). Foreman (2010, 256-258) refers to this as a charter development where he also emphasizes the business case and importance of the default assumptions. The business case should focus on a risk management practices, like the impact of loss, and reputation of a company, instead of trying to quantify the probabilities of vulnerability exploitation. The default assumptions may be high-level statements if there is uncertainty at the time strategy is created. However, the specific statements will elucidate the expectations better. (Sb. p. 259.)

For the actual deployment of a vulnerability management process, Foreman (2010, 268-269) suggest two possible approaches. First is a basic strategy where deployment starts with a small scope and extends over time once confident and knowledge grows. Second approach is a risk-based strategy. The most critical assets are identified and deployed first. With this approach, it is proposed (said book p. 270) that deployment should start from the largest, most critical data centre. As the system support teams are usually collocated with the critical systems, the monitoring and problem response would be more efficient, and major issue handling would be faster. Palmaers (2012, 8) recommends starting with small scope to prevent the stakeholders being overwhelmed by vulnerability information from big amount of systems. He also suggests that organization's risk tolerance is an important factor when deciding about vulnerability management scope (Sb. p. 9).

4.2.2 Vulnerability management policies

According to Foreman (2010, 54), any security or compliance initiative should be supported by a policy. While the existing company security policies should cover the most general requirements, new policies should be introduced specifically for vulnerability management. These should include, but not to limit to: (Sb. p. 55).

- Prioritisation of vulnerabilities. As assets are not equal in terms of criticality, it is vital to decide what gets an attention first.
- Valuation of assets. Each asset should have a relative value representing its criticality which could be used in the prioritisation process.
- Remediation timely limitations. Depending on vulnerability severity and an asset criticality, a deadline should be set.

4.2.3 Development project

A development project can be presented with different models. Traditionally, it has been presented as a linear model where process phases follow each other subsequently from the beginning of a project to the end. This model, however, might simplify the process too much compared with the real life.

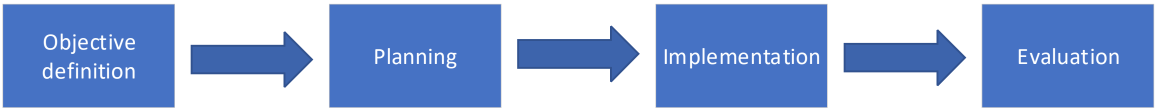


Figure 4. Linear project model (Toikko & Rantanen 2009; Salonen 2013).

In a spiral model, a development process can be understood as a continuous cycle. In this model, the results are evaluated repeatedly in new cycles (Toikko & Rantanen 2009, 64-70).

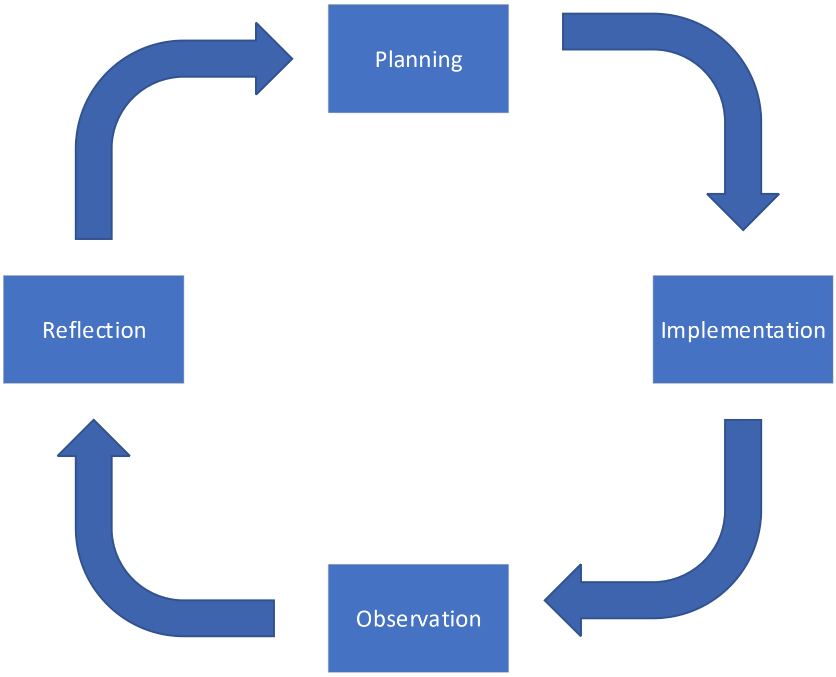


Figure 5. Spiral project model (Toikko & Rantanen 2009; Salonen 2013).

The model selected for the thesis is a *constructive model* (Salonen 2013) where the strengths of a spiral model and a traditional linear model have been combined.

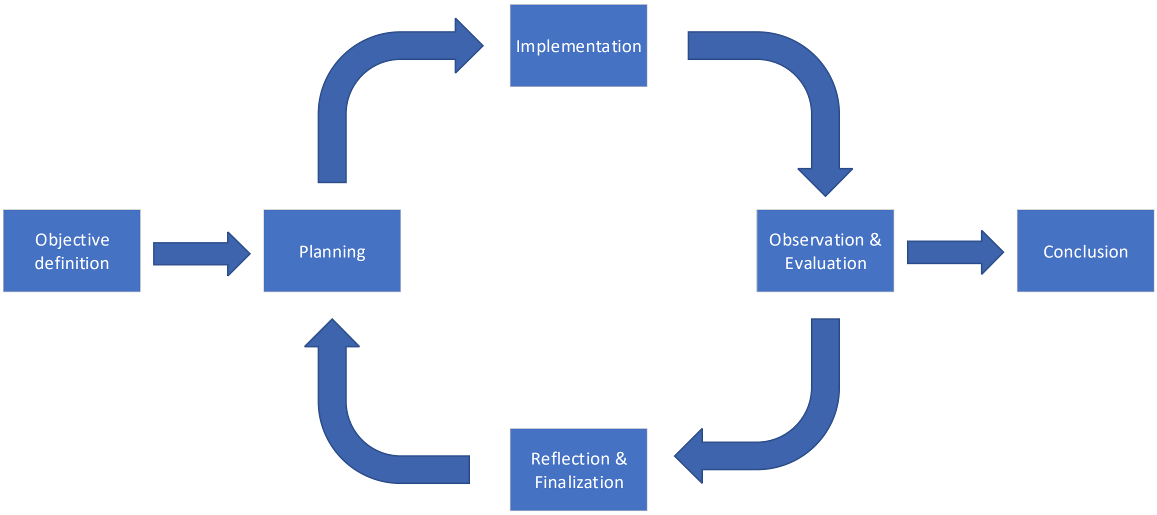


Figure 6. Constructive project model (Salonen 2013).

As discussed earlier, the phases in a linear model are timely sequential, according to a predefined plan, the model cannot necessarily address variables that might change the project direction. In the constructive model, the objectives are set first, like in a traditional model. After this phase, the spiral model is used for planning and implementation. The outcome will then be presented and published. The constructive model attempts to overcome the working challenges caused by personal, cultural, and social features. To accomplish these goals, the constructive model emphasizes interaction, participation and pedagogical way of working. In practice, that means pausing, evaluation, reflection, possible revise, and interactive debate. The model has a built-in incremental way of working, which acknowledges that everything cannot be planned beforehand. Instead, tasks will become clearer over time and they might even be changed or adjusted. (Salonen 2013, 13-14).

Development activities are, most of all, a social process which assumes people’s active participation and interaction (Toikko & Rantanen 2009, 89). The early involvement of key stakeholders ensures that all requirements and interests are covered (Sb. 90). Inclusiveness also helps to overcome politics, fear, and the natural tendency of people to resist change (Foreman 2010, 59). Thus, the proper introduction of a new process, roles and tools would increase people’s commitment, reduce friction during implementation, and clarify expectations.

4.2.4 Vulnerability management process development

In reference literature, there seem to be varying views on how the vulnerability management process should be designed. Many authors, like Palmaers (2013) and Perraudeau (2009), have a practical approach with precise and perceptible steps how to design and implement the process. In most cases, these documents are focused on vulnerability management of cybersecurity, thus, making them useful for this thesis. On the other hand, there are publications by authors like Kohnke et al. (2016) and Carnegie Mellon University (2016), that have a theoretical approach which makes them applicable to vulnerability management process development for any domain.

All reference material related to vulnerability management proposes that it should be an ongoing process running in cycles. Figure 7 presents an example stance how a vulnerability management process could look like.

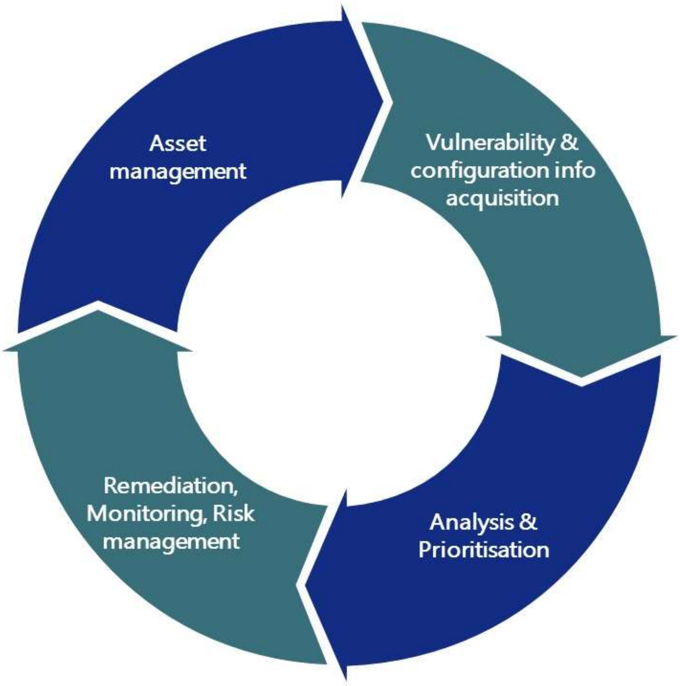

Asset management
Vulnerability & configuration info acquisition
Analysis & Prioritisation
Remediation, Monitoring, Risk management

Figure 7. Vulnerability management circle (Foresight Cyber).

In literature, there are some nuances about what should be included in a vulnerability management process. Nevertheless, following items have been identified as the key elements to implement a successful vulnerability management process:

1. All assets in scope must be identified, categorized and prioritized. This requirement may require extra effort in terms of host discovery, ownership finding, CMDB update and risk assessment. (Brackin 2002, 4-5; Perraudeau 2009; Gerberding 2018; Center of Internet Security 2018, 6-8).
2. Process roles and flow must be properly communicated to all stakeholders and proper training arranged. Expectations must be clearly presented. Also, sponsorship from management is vital for a process to succeed. (Carnegie Mellon University 2016, 19).
3. Discovered vulnerabilities must be prioritized based on relevancy, asset priority, associated thread and risk. Risk is assessed by evaluating what is the asset’s importance to business, and what impact an availability disruption would introduce. (Brackin 2002, 8-10; Perraudeau 2009; Carnegie Mellon University 2016, 21-22; Lynch 2015).
4. The remediation phase must be monitored, and it must be completed in a beforehand agreed timeframe which depends on asset’s priority. The remediation task is a joint undertaking between security officer, asset

owners and support teams. Security officer and support teams analyse vulnerabilities from their perspective and give recommendations to asset owners about possible remediation methods. The asset owner makes final decision based on recommendations and his own analysis. If vulnerability cannot by remediated, it must be handled via an exception process. (Perraudeau 2009; Palmaers 2012, 3).

5. Remediation should be verified. This is to make sure that vulnerabilities have been removed and no new vulnerabilities have been introduced. All findings during process cycle should be recorded and analysed for improvement activities. (Perraudeau 2009; Carnegie Mellon University 2016, 24; Center of Internet Security 2018, 11).

5 RESEARCH RESULTS

In this chapter, the results of the thesis will be presented. Firstly, the strategy and additionally created supporting policies are described. Then, the developed process and tooling applied will be presented. Finally, the people and their participation will be discoursed.

5.1 Strategy

The following strategy for local vulnerability management was defined. The strategy consists of setting the objective, scope and defining the default assumptions and the methods.

5.1.1 Objective

The objective was derived from the original research problem. The objective is to have a well-designed vulnerability management process and supporting tooling defined and implemented.

5.1.2 Scope

The strategic target scope was declared to cover all commissioner’s devices attached to network in Finland. This would include:

- Physical and virtual servers

- Network devices and firewalls
- Virtualization platforms

Out of scope:

- Internet facing systems (covered by global scan)
- Any customer device or system
- Workstations

On tactical level, the basic deployment strategy (Foreman 2010, 268) was selected. This means starting with smaller scope, monitor progress, and then extend gradually. The initial deployment was divided in three approximately same sized phases. The order of phases was decided by asset criticality, starting with low criticality.

5.1.3 Business case

No business case was calculated at this time, nor it was required by the original assignment.

5.1.4 Assumptions

The following high- and low-level statements were assumed by default. They would support the vulnerability management process and tooling implementation by expressing the expectations:

- Management support for the project would be available.
- The required human resources would be available until the project completion.
- Computing capacity will be available for vulnerability scanner infrastructure.
- The log management system is able to ingest and index the vulnerability scanner logs.
- The vulnerability scan should take place at least monthly, preferably fortnightly.

5.1.5 Methods

The selected method of scanning was a non-credentialed scan which only tests vulnerabilities that are exposed to network. One of the reasons for this approach was the ease to get started, and to avoid an additional task to set up the scanning credentials for the targets. Another purpose was not to cause a potential disturbance or performance degradation in an early stage of the program. However, a credentialed scan where tests are executed inside targets should be possible in the future, and functionality should be supported by selected tooling.

5.2 Policies

The primary requirement for vulnerability management comes from commissioner's global ICT security policies. To enforce the remediation activities, following additional policies were introduced. These policies are to ensure that all pre-requisites for successful vulnerability management are in place. The first and the second policy were defined to support and emphasise the key roles of the new vulnerability management process. The rest of the policies were defined according to Foreman (2010, 54-55) to make the remediation a priority. None of these policies overrides global policies in case of conflict.

1. All assets must have an owner defined. Asset owners are responsible for risk assessment and remediation plans.
2. All assets must have a support group defined. Support group, or system engineers are responsible for remediation implementation. They also have an important role in remediation planning.
3. All assets must have criticality or relative value assigned. This information would be used for the prioritization of remediations.
4. Vulnerabilities must be prioritized based on severity and asset criticality. This will give a guideline for the order of remediation activities should take place.
5. Dead-line for remediation or risk acceptance must be defined. The vulnerability classification is based on Common Vulnerability Scoring System (CVSS) version 2 (Mell et al. 2007). This model is compatible with commissioner's global vulnerability classification and it should be supported by selected vulnerability scanning software. The remediation timelines for each level of scale were adapted from commissioner's security patch policy (see Appendix 1).

5.3 Process

The most important objective of the thesis was to define an unambiguous process for local vulnerability management. The following key topics for such a process were identified:

- Roles
- Input & output
- Flow
- Interfaces to other processes
- Key performance indicators

The process planning begun by creating a simple concept map by adopting Palmaers' (2012) process roles to identify all parties that should be involved, what are their responsibilities, and how they should interact with each other. This gave a starting point to define roles and process further.

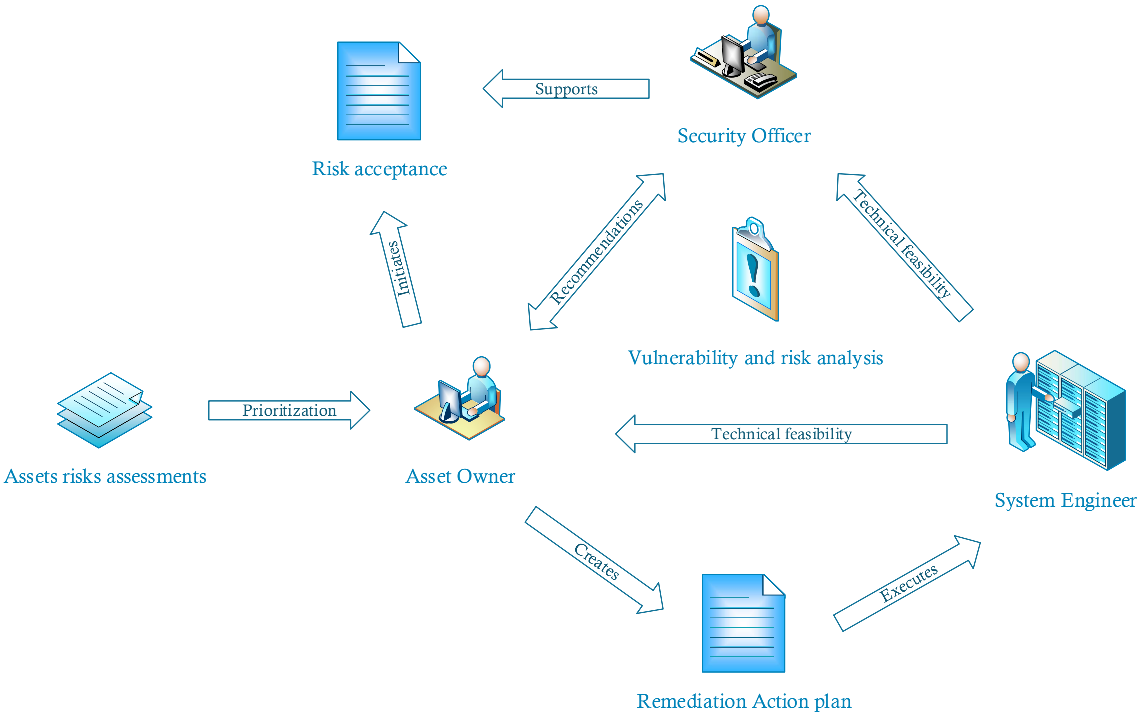


Figure 8. Concept map for vulnerability management process

Instead of four roles, like Palmaers (2012) has presented in his vulnerability management process, three main roles were identified during conceptual mapping. These were the Security Officer role, the Asset Owner role and the System Engineer role. The fourth role by Palmaers (2012), the security engineer, was combined with the Security Officer role in order to simplify the process and keep roles to minimum. Also, it was noticed that roles have not only functional responsibilities during the process, but also general level obligations like asset management, risk assessment, or scan scope definition.

5.3.1 Security officer role

The first role to scrutinise is the Security Officer (SO) role. The Security Officer is responsible for the process itself and its execution. Also, the Security Officer contributes to the risk management and vulnerability remediation activities supporting the decision making. The detailed general tasks and responsibilities of Security Officer role in were defined as:

- Process ownership, including but not limited to, process development, improvement, and revision. The process should be maintained and adjusted according to changing operational environment.
- Process governance. The progress of remediation should be monitored and intervened if not completed in a timely manner.
- Process training and information sharing. The people should be involved as much as possible to boost the commitment. Also, new processes should be trained properly prior to roll-out.
- Local vulnerability scan scope definition (if results not coming from the global vulnerability management process). Each vulnerability scan should be limited to the pre-defined set of targets. These could be entire networks, single hosts, or even single application.
- Assist asset owners with risk assessment activities. Besides being an integral part of risk management, the risk assessment supports asset management which helps the prioritisation of remediation activities.

The detailed operational tasks and responsibilities of Security Officer role were defined as:

- Local vulnerability scan planning and scheduling (if the results are not coming from the global vulnerability management process). The vulnerability scan should take place on a beforehand agreed time frame.
- Participation in analysis of vulnerabilities and associated risk. Once the vulnerability scan result has been received, Security Officer should immediately triage whether high severity vulnerabilities exist that would require extra attention.

- Recommendation of remediation tactics. Based on vulnerability and risk assessment, Security Officer should give recommendations to asset owners how to proceed with remediation.
- Initiation of rescan. Once remediation tasks have been implemented, a rescan should take place to confirm that vulnerabilities have been mitigated. If rescan fails, the results should be sent to the asset owner to revise the remediation plan.
- Initiation of an exception process. If vulnerability cannot be mitigated or risk reduced by additional controls, the risk should be accepted. This should be done via an exception process. In an exception process a risk document will be created which all parties will sign. This is to ensure that liabilities and responsibilities have been fully understood and agreed.

5.3.2 Asset owner role

The second role is the Asset Owner (AO). The Asset Owner should be treated as the most important role accountable for the remediation of vulnerabilities would be carried out one way or another. Having the best knowledge of the asset at hand, the Asset Owner should be able to decide and plan the remediation based on the risk and the consultancy received from other stakeholders of the process. The detailed general tasks and responsibilities of Asset Owner role were defined as:

- Management level responsibilities of ICT assets in vulnerability scan scope. Asset Owners should be aware of applications, application connections or interfaces, and general architecture. This would enable the Asset Owner to make proper decisions regarding to vulnerability remediation activities.
- Assessing risk for assets with the help of Security Officer. As stated earlier, the risk assessment results are used during remediation planning to prioritize the actions.

The detailed operational task and responsibilities of Asset Owner role were defined as:

- Analyse vulnerabilities based on context, relevance and risk, and then create a prioritisation. Relevant vulnerabilities should be forwarded to

System Engineer team for technical feasibility evaluation. As the Asset Owner has the best knowledge of the environment, the applicability of vulnerabilities should be evaluated with less effort. Asset Owner will also get input from security officer to support the remediation planning.

- Create a remediation plan. The remediation plan includes basically three risk treatment options:
    - Mitigation
    - Compensation
    - Acceptance

  Risk mitigation means fixing the vulnerability by applying a software patch, or even removing the vulnerable item. Risk compensation typically means narrowing down the effects of vulnerability by introducing additional controls. These could include, but not limit to, changing configuration or limiting access to vulnerable service. If vulnerability cannot be mitigated or risk reduced by additional controls, the risk should be accepted. This should be done via an exception process which is handled by Security Officer. The risk exception process makes sure that all parties acknowledge and accept the risk.
- Revise the remediation plan. After the remediation tasks have been implemented, a rescan will take place. If rescan fails, the remediation plan should be corrected to cover failures.

5.3.3 System engineer role

The third role is the System Engineer (SE). This role is typically a delivery or support team of a certain technical domain, such as a Linux server team or a network team. The System Engineer provides technical expertise to support the decision making. The detailed general tasks and responsibilities of System Engineer role were defined as:

- Daily operations. These are typical system administration tasks like configuration changes, applying patches, or installing applications.

The detailed operational task and responsibilities of System Engineer role were defined as:

- Analyse vulnerabilities from technical perspective. As System Engineers have the best technical knowledge of the assets, they should be able to evaluate the applicability of reported vulnerabilities.
- Provide recommendations for remediation. System Engineers should investigate the technical feasibility of mitigation actions, and then give proper recommendations for the remediation.
- Implement the remediation actions. Once the remediation plan has been created by the Asset Owner, it should be executed by System Engineers. If implementation fails, Asset Owner should be notified with proposal how to revise the plan. After successful completion, Security Officer should be notified to initiate a rescan.

5.3.4 Input

The input of a process can be seen as an external stimulus required to initiate the process (Tuurala, 2010). For this vulnerability management process, the input is a list of vulnerabilities that needs remediation. The list can be a result from a vulnerability scan, or from other means that are able to identify vulnerabilities. Also, input could consist of a compliancy scan results which are not necessary vulnerabilities but findings that would need similar treatment.

5.3.5 Output

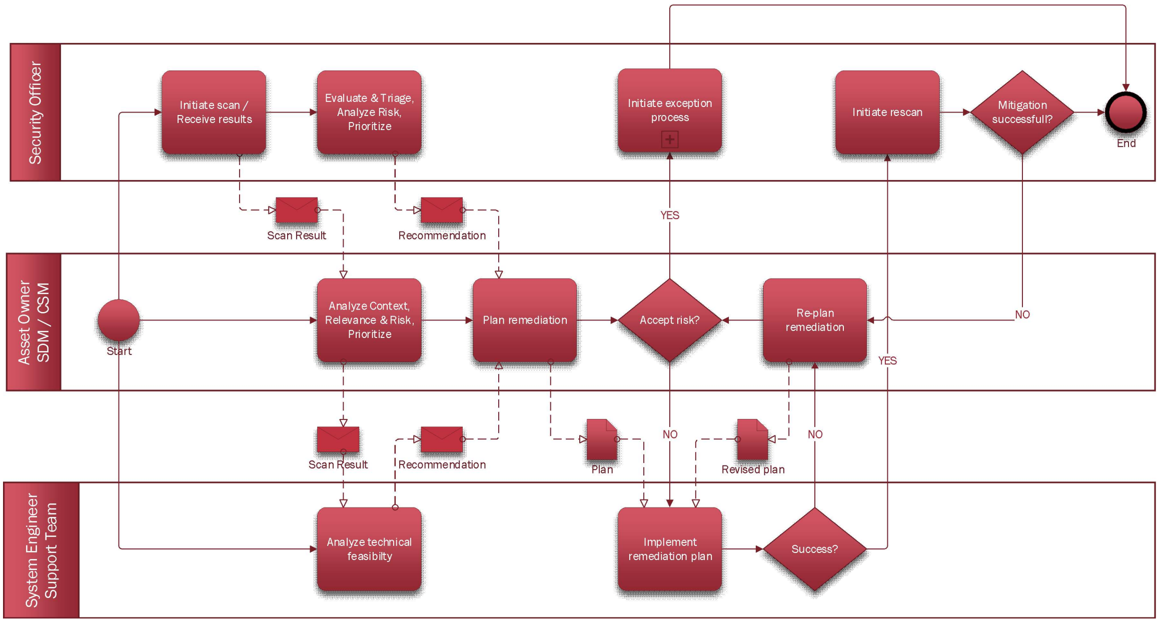
Tuurala (2010) uses the term output to refer to a consequence of a process which materializes as a product, a by-product, or a change in certain condition or state. The desirable result of the vulnerability management process is to have all vulnerabilities handled one way or another in a timely manner. This would produce an added-value to asset owners, and ultimately to the whole organization, by reducing a risk of vulnerability exploitation and data breach. The output is intangible, but it is, nonetheless, measurable with proper performance indicators.

5.3.6 Interfaces

During process execution, an external function might be required. The function can be another process, or some other part of value chain the process is part of (Martinsuo & Blomqvist 2010, 9). The vulnerability management process has one interface to a sub-process for an exception handling. If vulnerability cannot be mitigated, or the effect reduced, an exception must be created and recorded for risk acceptance. It is worth mentioning that also a change management process could be invoked during the remediation activities. This, however, has no direct interface to the vulnerability management process due to fact that the change management should be handled within the remediation planning task of the vulnerability management process. This approach was chosen to simplify the vulnerability management process presentation and functionality.

5.3.7 Steps

As mentioned earlier, the starting point for planning process steps was Palmaers' (2012) process model. He has divided and presented his model in separate phases which makes it detailed and comprehensive. However, as the objective was to create a streamlined process, only the mandatory steps and functions were to be included. Also, Palmaers (2012) has included the initial vulnerability scan in his process. In our case, this would be unnecessary as the scan results would be an external input to the process. The process steps were presented as BPMN flowchart where all three key roles have their own swim lane (Figure 9).



Security Officer
Initiate scan / Receive results
Evaluate & Triage, Analyze Risk, Prioritize
Initiate exception process
Initiate rescan
Mitigation successful?
End
Scan Result
Recommendation
YES
Asset Owner SDM / CSM
Start
Analyze Context, Relevance & Risk, Prioritize
Plan remediation
Accept risk?
Re-plan remediation
NO
YES
Scan Result
Recommendation
Plan
NO
Revised plan
NO
System Engineer Support Team
Analyze technical feasibilty
Implement remediation plan
Success?

Figure 9. The vulnerability management process.

Process flow can be described:

1. The process is started by Security Officer who initiates a vulnerability scan and receives the results. Alternatively, the result is received from the global vulnerability scan. Asset Owners and System Engineers are made aware of results.
2. Security Officer, Asset Owner and System Engineer will analyse results from their competence point of view. Security Officer and System Engineer will give Asset Owner recommendations about remediation options.
3. Asset Owner plans remediation by either
    a. creating a remediation plan and passing it to System Engineer for execution. This might involve other ITSM processes like change management.
    b. accepting the risk. This is indicated back to Security Officer who initiates an external exception process. This process exists.
4. If the vulnerability was to be mitigated, System Engineer would implement the remediation plan and either
    a. indicates failure to Asset Owner who revises the remediation plan, or accepts risk
    b. indicates success to Security Officer who initiates a rescan to confirm mitigation.
5. If mitigation activities were not successful, Asset Owner is informed and again revise the remediation plan or accept the risk. Otherwise process exits.

5.3.8 KPIs

The primary objective of measuring performance, or effectiveness of a process, is to enhance process control and continuous improvement. A good measuring system does not strain the process itself, and it has only a few measurement items which concentrate on the pertinent. (Martinsuo & Blomqvist 2010, 16.)

Following key performance indicators were defined for process measurement (see Appendix 2):

- Weighted Intrusion Rate (WIR)

- Timely Remediation Rate (TRR)

The WIR is commissioner’s standard KPI for vulnerability management. It is a weighted ratio of an amount of vulnerabilities in a given population. It will be calculated per scan targets in a subnet, and per total number of scan targets. The TRR is a ratio of vulnerabilities remediated by meeting the time objective compared with an amount of vulnerabilities to be remediated. The TRR was developed to be able to measure the timely effectiveness of the process.

5.4 Tooling

Besides vulnerability management process definition, another objective of the project was to introduce tooling that supports the process implementation in real life practice. The tooling consists of two domains: Vulnerability scanner and Security Information and Event Management (SIEM) system.

5.4.1 Vulnerability scanner

The chosen product for vulnerability scanning was Greenbone Community Edition (GCE from now), formerly known as Open Vulnerability Assessment System (OpenVAS from now). As the OpenVAS, or GCE, has been used in commissioner’s environment in the past, and people already have experience with it, there was no need to look for alternatives within this project. GCE is an open source version of Greenbone’s commercial product Greenbone Security Manager (Greenbone Networks 2018b, 2). However, the whole scanner architecture was re-designed to get more agile and distributed solution (see Appendix 3).

The scanning architecture is based on GCE’s master-slave configuration (Greenbone Networks 2018a, 213). In this configuration, master holds all configurations, scan results, and provides a web-based user interface for administrative tasks. Master gets the updated vulnerability data from Greenbone Community Feed which is the free version of their security feed (Greenbone Networks 2018b, 3). The feed consists of NVT (Network Vulnerability Tests) data, SCAP (Security Content Automation Protocol) data, and CERT (Computer Emergency Response Team) data (Greenbone

Networks 2018c). Master acts also as a security feed proxy for slave scanners.

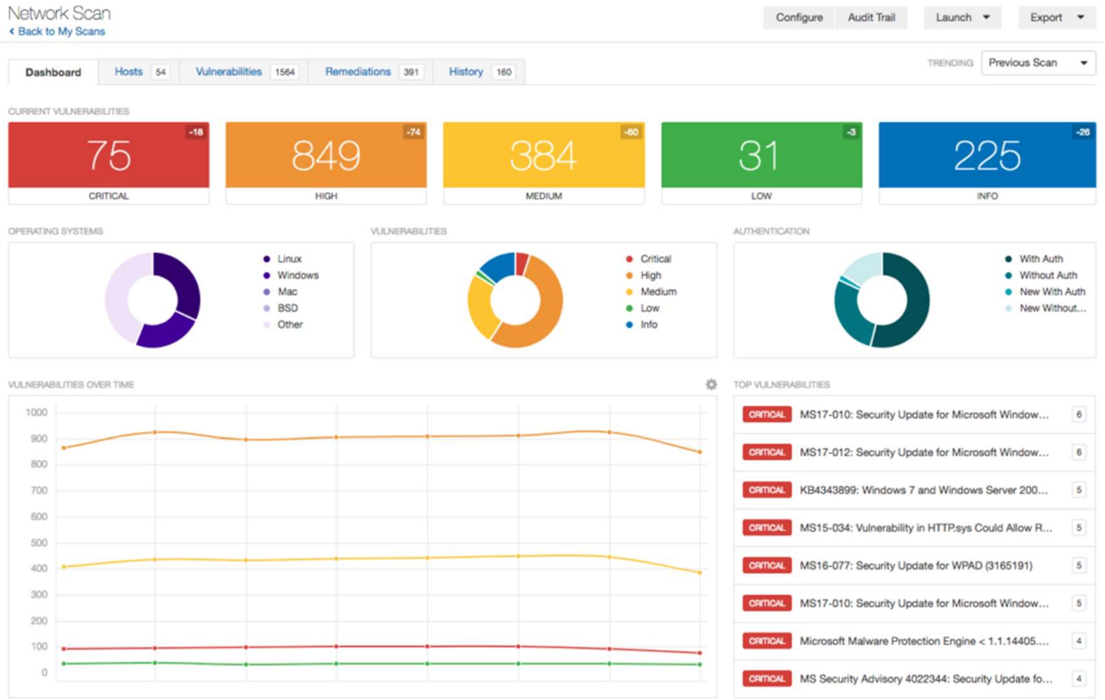
The slave scanners are deployed in the target networks on-demand basis. They execute the actual scanning tasks based on scheduling and commands delivered by master. The slave scanners are virtual machines in commissioner’s private cloud which makes them easily installable and removeable with a single command (see Appendix 4). The deployment could be also automated by using API calls from the master. Command and control traffic between the master and the slaves is encrypted.

Once the scan results have been received from slave scanners, the master sends them to a log management system for further analysis and remediation. The results can also be browsed and analysed by using the master scanner’s GUI.

5.4.2 SIEM

Commissioner has been using a software product “Splunk Enterprise” as a log management solution for years. Thus, it was a natural choice to feed in the vulnerability scan results as well to be able to enrich the data for correlation with other related events, and to be able to create a dashboard for situational awareness.

To have a quick overview in the vulnerability situation, a single pane of glass was created by using Splunk dashboard. Dashboards are views which can contain search boxes, fields, charts, tables, and lists (Splunk, 2019a). The idea what should be presented in the dashboard started to take a shape after some brainstorming. Obviously, the total amount of vulnerabilities by severity should be easily visible along with the key metrics. The amount of the most severe, and the most common vulnerabilities would help to understand if there are systematic problems in the environment. Naturally, the vulnerability details should be presented in a searchable and useful format to enable data export for reporting. The colour scheme and graphical appearance of the dashboard was designed as distinctly as possible with some influence from commercial dashboards like one from Purplesec (2019):



CRITICAL	MS17-010: Security Update for Microsoft Window...	6
CRITICAL	MS17-012: Security Update for Microsoft Window...	6
CRITICAL	KB4343899: Windows 7 and Windows Server 200...	5
CRITICAL	MS15-034: Vulnerability in HTTP.sys Could Allow R...	5
CRITICAL	MS16-077: Security Update for WPAD (3165191)	5
CRITICAL	MS17-010: Security Update for Microsoft Window...	5
CRITICAL	Microsoft Malware Protection Engine < 1.1.14405....	4
CRITICAL	MS Security Advisory 4022344: Security Update fo...	4

Figure 10. Network vulnerability assessment dashboard (Purplesec 2019)

The dashboard created presents following information (see Appendix 5).

- High severity, Medium severity, Low severity. A figure indicating amount of active vulnerabilities within each category.
- Vulnerabilities per Host. A figure presenting an average amount of vulnerabilities per a host.
- Weighed Intrusion Rate. A figure indicating WIR.
- Top CVSS. A gauge indicating the highest current CVS score.
- Average CVSS. A gauge indicating the average of current CVS score.
- Vulnerabilities by severity. A pie chart presenting the distribution of current vulnerability severities.
- Top 10 high vulnerabilities. A table showing the most frequent high severity vulnerabilities.
- Top 10 most common vulnerabilities. A table showing the most common vulnerabilities.
- Top 50 vulnerable hosts. A stacked vertical bar chart combining all current vulnerabilities per host classified by severity.
- Vulnerability details. A table presenting all details by current vulnerabilities enriched with available CMDB data. The table is searchable by
    - Asset criticality
    - Asset Fully Qualified Domain Name

    - Asset IP-address
    - Vulnerable port
    - Vulnerability severity
    - Free text
- Vulnerability history. A chart over time presenting the history of vulnerabilities.

In addition, it was investigated whether the vulnerability remediation workflow could be implemented by using Splunk. Pretty soon it came apparent that Splunk Enterprise alone cannot provide such a functionality. The vulnerabilities could be considered as incidents which have a certain lifetime and their status should be changeable. This was a problem because once an event has been indexed by Splunk, it can be changed no more (richgalloway, 2016).

After some investigations, we came across Splunk Enterprise Security add-on. It is a commercial product which basically turns the basic Splunk functionality into analytics driven SIEM solution (Splunk, 2019c). One of the functionalities of Splunk Enterprise Security is the Notable Events which are created by correlation searches, and their primary function is to provide an incident management functionality for security events (Splunk, 2019b). This was exactly the required feature. In addition, Splunk Enterprise Security provided many other features to improve the whole security posture significantly compared with plain log management functionality. The conclusion was to propose commissioner purchasing Splunk Enterprise Security. However, it was decided by commissioner to bundle Splunk Enterprise Security with other current security related acquisitions. Waiting for investment approval would have caused an extra delay to the thesis, therefore, the implementation of the remediation workflow with Splunk Enterprise Security remains a future development topic.

5.5 People

5.5.1 Participation

As was mentioned in the previous chapter, involvement of all stakeholders and information sharing are key elements for project success. Commissioner’s key stakeholders were introduced to the project in a very early phase. Many people from commissioner’s organization who will have a process role in future have been involved in asset management activities, gaining knowledge about the project.

5.5.2 Training

For training purposes, a presentation was created to cover:

- Background of project
- Scope and phasing
- Policies
- Process roles and responsibilities
- Process workflow
- Asset management
- Vulnerability ranking and prioritization
- KPIs

The training was arranged as an online session where participants had possibility to raise concerns and ask questions. Especially, the importance of then process roles and the asset management was emphasised.

6 DISCUSSION

In this final chapter, the results and findings during, and after, the research will be presented and reflected. Also, recommendations and possible development areas will be presented.

6.1 Research problem

The objective of the thesis was to create an unambiguous process for vulnerability management. The research problem was distilled into a statement: *There is no unambiguous process in place for local vulnerability management.* Seven research questions were created out of the research problem in order to find actual areas for development.

The first research question asks why it is important to have an unambiguous vulnerability management process in place. The most important aspect is that the risk of vulnerability exploitation would be reduced considerably as the found vulnerabilities would be remediated in controlled and timely manner. Another aspect would be to gain visibility over a security posture of the environment. This would be complemented by ability for detailed reporting. Also, commissioner’s security policies would require that a continuous and measurable vulnerability management process is in place. Failure to comply with this would result in an anomaly in an audit situation.

The second research question tries to understand what things should be considered prior to starting the implementation of the process. The first step would be to define a strategy and policies for the vulnerability management program. These would give a foundation and a framework for the further activities. The next step would be risk assessment based on up-to-date asset information. It would be very difficult to execute remediation activities without adequate knowledge about asset’s identity, connectivity, ownership, criticality and possible relationship to other assets. Also, a human aspect can be seen as a key success factor for implementation. People who will be affected by the new process will accept the change with less friction if they are allowed to participate in planning, and if they are informed well enough in time. Naturally, the information sharing and possibility for a training dictates also how the post implementation will succeed.

The next three research questions are handling the actual process and its internals. Firstly, we are looking at the key roles of the process. Three key roles were identified to group the essential tasks into proper aggregates. Like the second research question, also the role definition emphasizes the importance of asset management. The asset owner role plays the most important part in a process, starting from identifying and sharing the asset information, and concluding in being responsible for the remediation of vulnerabilities. The security officer role can be seen as a process initiator, and a facilitator who provides tools and context for other roles to work with. The system engineer role is clearly the supporting role which means providing input for remediation planning and then implementing the remediation according to plan. Nonetheless, both tasks are very important for the process success. As the number of roles was kept in bare minimum, the process is not getting too complex. All process roles have a clear segregation of the main tasks although all roles participate and contribute the remediation planning.

On the question of vulnerability management process progression, it can be split into three logical stages. Firstly, there is an analysis and planning phase where all input will be processed and converted into an actual remediation plan. Secondly, the implementation phase will take the plan into real world environment. Thirdly, the post implementation phase will finalise the process either by successful testing the remediation, or in case of failure, by putting the process back on the planning phase.

The only interface is with an exception process. However, it is a very important connection as it will exempt the asset owner from the responsibility of remediating the vulnerability any further. The exception process guarantees that a residual risk has been accepted by all appropriate parties and that a liability has been moved to a higher level in organisation.

The last research question asks about best suitable tooling for the process. Obviously, the process input, the results, should come from a scanning tool. This could be commercial, open source or in-house developed. Using an open source scanner, like in this case, is an economic way to achieve good result quality. Also, open source software can be customised according to the use case at hand. The downside is lack of commercial support. This is, however, compensated by the open source community to certain extend. Nevertheless, with open source tooling, local competencies are required. As far as the result are concerned, they are typically a list of findings, delivered in CSV- or XML-format. This makes a spreadsheet a possible tool for processing and distributing the results. However, for more flexible reporting and to create dashboards for situational awareness, a log management system should be used. As the findings could be understood as incidents, a tooling would be required for handling those. This could be an ITSM incident management tool or preferably a SIEM. For this, there are commercial and open source products available. In this thesis case, a natural choice for log management is to use commissioner’s existing Splunk log management system. However, as it lacks any incident handling capabilities, a commercial SIEM add-on, Enterprise Security, should be acquired.

6.2 Other findings

During the development, it became apparent that the process could be used not only for local vulnerability management, but for other areas as well where the similar mechanism of handling and remediating deviations would be needed. Such areas would be, for example, how to treat the findings of a global vulnerability scan, or a compliancy scan. After having the process in place for a short period of time, it was noticed that a formal way of working would benefit all parties involved. All stake-holders would know their role and responsibilities without confusion or overlap. Lack of hassle and frustration would not only boost people motivation, but it would actually improve performance as an effort is focused in a correct place. Also, managers would have a better visibility and confidence if the remediation of findings were completed in a formal and a timely manner. Although there is no measurable data available, the empirical remarks support these conclusions. Unfortunately, due to schedule issues, an anonymous end-user survey was not performed. To better understand the implications of rolling out a process, future studies could reserve more time for this. However, based on user feedback received on a daily work, it could be assumed that a user survey would have come up with a positive overall result.

6.3 Research evaluation and validity

Overall, even as there are several ways to implement a vulnerability management process, and multiple examples available, there is no silver bullet that would cover all the requirements in a single ready-made solution. First of all, the vulnerability management process should cover all requirements initially set by the commissioner, and by the thesis research problem. On the other hand, the process should be as light as possible which would make an implementation and roll-out easier, possibly in other countries as well. Combining an existing examples and knowledge with the requirements, a tailored vulnerability management process was successfully created in co-operation with commissioner’s key resources fulfilling the primary objective of the thesis. In general, after some searching, the answers to the questions were found quite easily although in many occasions this would require applying information from multiple sources. Most of the references are online sources which make them prone to a change or even deletion. This might affect the possibility to verify the references in the future, thus affecting on the validity of the research. However, all the key findings and result were backed up by multiple references. Hence, the results regarding to a vulnerability management process conception and creation is in line with the theoretical framework and the current best practices of the industry.

Setting up the vulnerability scanning environment and creating a single pane of glass dashboard was more challenging than just plain theoretical planning and design mainly due to fact that there was no complete reference architecture available. Also, some activities were dependent on someone else’s contribution to the project. This was, however, only a minor problem as a development process is people’s active participation and interaction, like the theory suggests. As there was no usable complete solution available for the vulnerability scanning infrastructure, a new, scalable solution based on virtual machines was introduced. The solution included in an image creation for a slave scanner, the image deployment from the commissioner’s private cloud portal, and slave scanner automatic integration with the master scanner. The available product documentation was used a certain extent, however, most of the work was done by applying an existing knowledge of the project resources. As discussed above, there is no known reference environment to compare. Therefore, the validity of the environment per se is difficult to be evaluated by any other means except a successful implementation. Nonetheless, all project resources have an extensive working history in their area of expertise. Thus, the outcome will follow the best practices where possible. As a matter of fact, the outcome could be evaluated as an innovative solution based on overall simplicity, yet a lot of technical details hidden under the hood. Also, the part of the vulnerability scanning infrastructure implementation was an integration to the log management system. This was done by utilising a built-in mechanism of the vulnerability scanner.

Once the scan results are sent to the log management system, they should be presented in an informative and a precise manner. For this, a dashboard was created presenting the key metrics from the latest scan results. As the author had earlier experience of creating searches and charts with the log management system, the dashboard was created with a reasonable effort.

The look and feel, the usability and informativeness of the dashboard are comparable to commercial products. Having that in mind and recalling that an improvement of a situational awareness was one of the project objectives, the implementation of the dashboard can be declared successful and valid.

Regarding to other secondary objectives, some of them will be reached only after a period of time. Especially, the asset information quality improvement might require considerable effort. The potential information sources might be scattered all over the organisation, or in the worst case, they have even left the company. However, being a mandatory pre-requisite for successful vulnerability management, this topic really should have the focus and the priority. In this context, the vulnerability management process execution can be seen as a driver that enforces proper asset management. Therefore, it could be summarised that the objective has not been fully reached yet but the journey has begun.

6.4 Future development

There were a few topics identified during the project for the future development. The most of them are regarding to the tooling. The topics are not in any particular order, however, the SIEM and an automation topic should have the priority.

As discussed earlier, to handle the vulnerability scan findings, or the security incidents properly, a Security Information and Event Management, or a SIEM system would be required. The vulnerability findings should be fed automatically into a SIEM system which would combine and correlate logs from various sources, and then raise alerts based on pre-defined criteria. Also, a tooling for workflows and security incident management for administrators should be available. This could be implemented with a SOAR system which would provide even more possibilities for security incident automation and response.

In addition, the existing vulnerability dashboard created in the log management system could be improved. A correlation to other events would be possible to enrich the output of existing searches with the vulnerability information. Also, various new metrics could be introduced to enhance the overall situational awareness. Particularly, a risk-based approach would help to concentrate on the most critical assets. This, however, would require proper asset information in place.

Another development topic is to investigate the possibility for a credentialed vulnerability scan. This would improve the scan coverage and quality considerably as the assets would be examined not only from the network side, but also from the inside. The most common and the most useful area of improvement would be to gain exact information about installed software and software patches. Comparing this information with an input provided by a vendor and other sources would give a more comprehensive view on faults the asset might have. The credentialed vulnerability scan could be used for inspecting a compliancy by checking whether a given set of configuration items, such as password settings are complying with the policies of an organisation.

Regarding the scanning infrastructure, an automation would be an obvious development theme. The provisioning of the slave scanners takes place in a private cloud portal. It is currently done interactively with a web browser. However, the portal has an application programming interface (API) which could be used to make the provisioning request from a command line by scripting. Also, the scanner master instance has a command line interface (CLI) which would allow to start scan tasks without logging in the web interface. The ultimate goal would be combining these two features allowing to automate the whole scanning process.

As explained earlier, an anonymous end user survey regarding process roll-out and usability was not carried out during the project phase. However, the survey would produce a valuable input of the humane side of the project. It might also reveal possible problems with the process itself that would not surface on daily work. Thus, implementing such a survey would be recommended in the future.

7 REFERENCES

Abernathy, R. & McMillian, T. 2016. CISSP Cert Guide. 2nd edition. Indianapolis: Pearson Education.

BeyondTrust. 2019. Vulnerability Scanning. WWW document. Available at: https://www.beyondtrust.com/resources/glossary/vulnerability-scanning [Accessed 7 November 2019]

Brackin, C. 2002. Vulnerability Management: Tools, Challenges and Best Practices. PDF document. Amended 29 August 2003. Available at: https://www.sans.org/reading-room/whitepapers/threats/vulnerability-management-tools-challenges-practices-1267 [Accessed 9 November 2018].

Carnegie Mellon University. 2016. CRR Supplement Resource Guide Volume 4 Vulnerability Management. PDF document. Available at: https://www.us-cert.gov/sites/default/files/c3vp/crr_resources_guides/CRR_Resource_Guide-VM.pdf [Accessed 4 November 2018].

Center of Internet Security. 2018. CIS Controls. PDF document. Updated 19 March 2018. Available at: https://www.cisecurity.org/cybersecurity-best-practices/ [Accessed 4 November 2019].

Core Security. 2014. The Threat & Vulnerability Management Maturity Model. PDF document. Available at: https://pdfs.semanticscholar.org/39b5/6bffb6385481043974cc9f0db743057cc37b.pdf [Accessed 3 November 2019]

Davenport, T. H. 1992. Process Innovation: Reengineering Work through Information Technology. Harvard Business Review Press.

Exabeam. 2019. White paper: Rules versus models in your SIEM. PDF document. Available at: https://www.exabeam.com/library/rules-versus-models-in-your-siem/ [Accessed 28 October 2019]

Foreman, P. 2010. Vulnerability Management. 1st Edition. Boca Raton FL: CRC Press.

Foresight Cyber. No date. Vulnerability management. WWW document. Available at: https://foresightcyber.com/vulnerability-management-details/ [Accessed 4 November 2018].

Gartner. 2019a. Security Information and Event Management (SIEM). WWW document. Available at: https://www.gartner.com/it-glossary/security-information-and-event-management-siem/ [Accessed 11 October 2019]

Gartner. 2019b. Security Orchestration, Automation And Response (soar). WWW document. Available at: https://www.gartner.com/en/information-technology/glossary/security-orchestration-automation-response-soar [Accessed 3 November 2019]

Gerberding, K. 2018. The Difference Between Vulnerability Assessments and Vulnerability Management. Blog. Updated 19 February 2018. Available at:

https://www.hitachi-systems-security.com/blog/difference-vulnerability-assessments-vulnerability-management/ [Accessed 4 November 2018].

Greenbone Networks. 2018a. Greenbone Security Manager with Greenbone OS 4. PDF document. Updated 4 April 2018. Available at: https://docs.greenbone.net/GSM-Manual/gos-4/en/GSM-Manual-GOS-4-en.pdf [Accessed 19 March 2019]

Greenbone Networks. 2018b. The Different Flavors of Greenbone's Technology. PDF document. Updated 5 February 2018. Available at: https://www.greenbone.net/wp-content/uploads/The-Different-Flavors-of-Greenbones-Technology-20180205.pdf [Accessed 19 March 2019].

Greenbone Networks. 2018c. CURRENT GREENBONE SECURITY FEED LIVE CONTENT. WWW document. Available at https://www.greenbone.net/en/greenbone-security-feed-live/ [Accessed 19 March 2019].

imperva. 2019. Security information and event management (SIEM). WWW document. Available at: https://www.imperva.com/learn/application-security/siem/ [Accessed 11 October 2019]

Infosec. 2019. A Brief Introduction to the Nessus Vulnerability Scanner. WWW document. Updated 26 July 2019. Available at: https://resources.infosecinstitute.com/a-brief-introduction-to-the-nessus-vulnerability-scanner/#gref [Accessed 7 November 2019]

ISO 55000:2014(en). 2014. Asset management — Overview, principles and terminology. Part 3.2: Terms relating to assets.

Kananen, J. 2015. ONLINE RESEARCH FOR PREPARING YOUR THESIS, A guide for conducting qualitative and quantitative research online. Jyväskylä: Authors & JAMK University of Applied Sciences.

Kohnke, A., Shoemaker, D. & Sigler, K. 2016. The Complete Guide to Cybersecurity Risks and Controls. Boca Raton FL: CRC Press.

Lynch, S. 2015. Vulnerability Management. Blog. Updated 30 October 2015. Available at https://resources.infosecinstitute.com/vulnerability-management/ [Accessed 4 November 2018].

Martinsuo, M. & Blomqvist, M. 2010. Prosessien mallintaminen osana toiminnan kehittämistä. PDF document. Tampere. Tampere University of Technology. Faculty of Business and Technology Management. Available at: https://tutcris.tut.fi/portal/files/2098668/prosessien_mallintaminen.pdf [Accessed 21 March 2019]

Mell, P, Romanosky, S. & Scarfone, K. 2007. A Complete Guide to the Common Vulnerability Scoring System Version 2.0. PDF document. Updated June 2007. Available at: https://www.first.org/cvss/cvss-v2-guide.pdf [Accessed 12 January 2019].

Metsämuuronen, J. 2001. Laadullisen tutkimuksen perusteet. Helsinki: International Methelp Ky

NIST. 2012. Guide for Conducting Risk Assessments. PDF document. Available at: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf [Accessed 4 November 2018].

OWASP Foundation. 2019. Vulnerability Scanning Tools. WWW document. Available at: https://www.owasp.org/index.php/Category:Vulnerability_Scanning_Tools [Accessed 7 November 2019]

Palmaers, Tom. 2012. Implementing a vulnerability management process. PDF-document. Available at: https://www.sans.org/reading-room/whitepapers/threats/implementing-vulnerability-management-process-34180 [Accessed 4 November 2018].

Perraudeau, E. 2009. Q&A: Vulnerability management. Blog. Updated 19 August 2009. Available at: https://www.helpnetsecurity.com/2009/08/19/qa-vulnerability-management/ [Accessed 4 November 2018].

Purplesec. 2019. Network Vulnerability Scanning And Assessment Services. WWW document. Available at: https://purplesec.us/services/network-vulnerability-scanning/ [Accessed 15 May 2019]

RedLegg. 2019. WHAT IS VULNERABILITY SCANNING, AND HOW DOES IT WORK?. BLOG. Updated 30 May 2019. Available at: https://www.redlegg.com/blog/what-is-vulnerability-scanning-and-how-does-it-work [Accessed 8 November 2019]

richgalloway. 2016. Is it possible to modify an indexed event?. WWW-document. Updated 10 June 2016. Available at: https://answers.splunk.com/answers/412085/is-it-possible-to-modify-an-indexed-event.html [Accessed 30 September 2019]

Salonen, K. 2013. NÄKÖKULMIA TUTKIMUKSELLISEEN JA TOIMINNALLISEEN OPINNÄYTETYÖHÖN, Opas opiskelijoille, opettajille ja TKI-henkilöstölle. Turku: Turun ammattikorkeakoulu.

Shanks, W. 2015. Building a Vulnerability Management Program – A project management approach. PDF document. Updated 21 May 2015. Available at: https://www.sans.org/reading-room/whitepapers/projectmanagement/building-vulnerability-management-program-project-management-approach-35932 [Accessed 23 August 2019]

Splunk. 2019a. About dashboards. WWW document. Available at: https://docs.splunk.com/Documentation/Splunk/7.2.6/SearchTutorial/Aboutdashboards [Accessed 29 April 2019]

Splunk. 2019b. Notable Event framework in Splunk ES. WWW document. Available at: http://dev.splunk.com/view/enterprise-security/SP-CAAAFA9 [Accessed 19 March 2019].

Splunk. 2019c. Security Information and Event Management. WWW document. Available at: https://www.splunk.com/en_us/cyber-security/siem-security-information-and-event-management.html [Accessed 30 September 2019]

Tillyard, J. 2019. The Difference Between SIEM and SOAR (Why Do I Need SOAR, If I Have SIEM?). WWW document. Updated 27 June 2019. Available at: https://www.dflabs.com/blog/the-difference-between-siem-and-soar-why-do-i-need-soar-if-i-have-siem/ [Accessed 1 November 2019]

Toikko, T. & Rantanen, T. 2009. Tutkimuksellinen Kehittämistoiminta. Tampere: Tampere University Press

Trisotech. 2019. BPMN Introduction and History. WWW document. Available at: https://www.trisotech.com/articles/bpmn-introduction-history [Accessed 29 April 2019]

Tuurala, T. 2010. Prosessi, prosessiorganisaatio ja prosessin ohjaus. WWW document. Updated 29 August 2010. Available at: http://www.kotiposti.net/tuurala/prosessit.htm [Accessed 22 March 2019]

LIST OF FIGURES