Aalto University
School of Electrical Engineering
Degree Programme in Communications Engineering

Mari Nikkarinen

English Name PLACE-HOLDER
English Subtitle PLACE-HOLDER

Master's Thesis
Espoo, English Date PLACE-HOLDER

DRAFT! — January 21, 2018 — DRAFT!

Supervisors: English Supervisor PLACE-HOLDER
Advisor: English Instructor PLACE-HOLDER

Aalto University
School of Electrical Engineering
Degree Programme in Communications Engineering

ABSTRACT OF
MASTER'S THESIS

Author:	Mari Nikkarinen		
Title:	English Name PLACE-HOLDER English Subtitle PLACE-HOLDER		
Date:	English Date PLACE-HOLDER	**Pages:**	18
Major:	English Professorship PLACE-HOLDER	**Code:**	Code PLACE-HOLDER
Supervisors:	English Supervisor PLACE-HOLDER		
Advisor:	English Instructor PLACE-HOLDER		
!FIXME **The abstract is written last.** FIXME!			
Keywords:	English Keywords PLACE-HOLDER		
Language:	English		

Aalto University
School of Electrical
Engineering

Aalto-yliopisto
Sähkötekniikan korkeakoulu
Degree Programme in Communications Engineering

ABSTRACT OF
MASTER'S THESIS

Author: Tekijä	Mari Nikkarinen		
Työn nimi: Finnish Name PLACE-HOLDER Finnish Subtitle PLACE-HOLDER			
Päiväys:	Finnish Date PLACE-HOLDER	**Sivumäärä:**	18
Pääaine:	Finnish Professorship PLACE-HOLDER	**Koodi:**	Code PLACE-HOLDER
Valvojat:	Finnish Supervisor PLACE-HOLDER		
Ohjaaja:	Finnish Instructor PLACE-HOLDER		
!Fixme **Abstrakti kirjoitetaan viimeisenä.** Fixme!			
Asiasanat:	Finnish Keywords PLACE-HOLDER		
Kieli:	Englanti		

Acknowledgements

!Fixme **Acknowledge some people.** Fixme!

Espoo, English Date PLACE-HOLDER

Mari Nikkarinen

Abbreviations and Acronyms

	!Fixme **Any used acronyms** Fixme!
acronym	explanation

Contents

Chapter 1

Introduction

!Fixme

- **Introduction of security landscape**
- **Explanation of point of view taken**
- **Pointing out the importance of threat analysis**
- **Short overview of OneID and equivalent systems**

Fixme!

Chapter 2

Background

2.1 Threat Analysis

!FIXME **I'll write about "early history" when I find sources.** FIXME!

Baskerville [1] separates the different threat analysis methods used up until 1993 into three generations: the first generation starting from 1972, the second generation from 1981 onward and the third generation that was introduced in 1988.

The first generation relies heavily on check-lists composed by authorities in the field. The systems these first-generation methods were used on were much simpler than the ones in use now, and security analysis consisted of choosing the best option of a limited list of known components, instead of the wealth of options that developers currently have. They do not expect the analyst to have deep knowledge, as independent analysis is not needed. It was also more focused on hardware than software. [1]

The second generation came when the systems got too complicated for the first generation's check-list method. It relies on partitioning the system into smaller components and then coming up with a solution that matches the functional requirements of each component. Secong generation methods are more complex, and the analysts need a higher degree of training, but does not rely on a set solution set and can be used in much more complex systems. [1]

Third generation relies on more abstraction compared to the second generation. Instead of partitioning the system into components like int second generation, the third generation relies on building abstract models of the systems, and using them as an aid in the analysis. These methods are most useful when designing the system, and the amount of training is even higher than in second generation. On the other hand, third generation solutions are

more flexible and should lead to less conflict between security and usability. [1]

Gerber and Von Solms [2] also divide the types of risk analysis done in the past into three different time periods.

The first era that they call the Computer-centric era is before the early 1980's, corresponding to a time before what Baskerville [1] calls the first generation. According to Gerber and Von Solms [2], a company's business didn't depend on computers and computer security then as it does now. Assets were easy to protect, as the protection could be done with physical controls like locks on the doors and threats could be found using simple check lists.

The second era, or the IT-centric era, lasted from the early 1980's to the early 1990's, which corresponds to what Baskerville [1] called the first and second generations. The businesses were increasingly dependent on computers, and security became more and more important. The identifying of the assets became an issue, as they were not physically located in the same place any more, and sometimes were not even physical. This lead to first using impact values, which depend on subjective guesses on impacts of threats, and then to qualitative techniques using tables with impact values and the probability of threats. [2]

The third era, called the Information-Centric era, started in the 1990's. In the Information-Centric era businesses depend on information and computers, and risk analysis is no longer enough as legal and business requirements become more and more important. This has lead to a trend of moving away from pure risk analysis, and into mixed risk and security requirements analysis, taking into account the unique risks and requirements each system and business has. [2]

2.1.1 STRIDE

STRIDE is a threat modelling technique that was developed by Loren Kohnfelder and Praerit Garg at Microsoft. It is based on going through types of threats that can be found in a system with the help of the mnemonic "STRIDE", which stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege. [5]

Spoofing is the attacker or a system is pretending to be someone or something they are not.

Tampering means changing something in data the attacker should not be able to change.

Repudiation is claiming not to have done something, or to have done something. This can be by bypassing or tampering with the logs, or on a

business layer by making claims about what has and hasn't been done.

Information Disclosure refers to providing information or data to an attacker or a third person who should not have access to it.

Denial of Service is an attack type where a system is tricked into using using its resources with illegitimate claims, usually to the point where legitimate service requests no longer go through.

Elevation of privilege means giving the attacker or someone else rights to do something they should not be able to do.

!FIXME **Maybe add examples? Might be useful to make this into a list or a table.** FIXME!

2.1.1.1 STRIDE at Microsoft

Howard and Lipner [3] outline the threat analysis progress used at Microsoft. It starts with defining use cases and determining the scope of the system that will be analysed. They also recommend that the reasoning behind the scope is documented, as changes in the system can lead to the need to have a different scope.

After that the dependencies of the system are gathered. [3] This also covers the hardware and the operating system or systems the program or programs run on. Situations where dependencies are proprietary or black boxes from the point of view of the user need to be taken into account.

The third thing that is done is defining security assumptions. [3] Incorrect assumptions can lead to large issues, for example if the operating system or the hardware is incorrectly assumed to be safe. This is also important because if the assumptions are not explicitly defined they will still implicitly exist in the background.

Then the developers write security notes for the use of developers who depend on the product and users. [3] This includes writing down what security implications have been introduced by external dependencies, for example any ports that have been opened for outside access.

After all this has been done, as many data flow diagrams as are needed are drawn based on the system. [3] The DFDs usually consist of entities, processes, data storages, data flows and external entities that have access to the system. Trust boundaries, or places where data is moved from one level or privilege to another, are also shown in DFDs. The diagrams are also usually drawn on several levels, to show processes from the user's and the program's point of view.

Threat types based on STRIDE are then determined, and what threats are relevant to each system element is identified.

When threat types have been determined, which of them is a risk to which element of the system is determined. [3] Threat-tree patterns can be used at this point. These threats are then documented.

The risk each of them poses to the system usually also is calculated based on the chance that the attack will occur and the damage the attack would pose to the company and to the system. [3] This helps the prioritization of the resolving of the threats. Unfortunately there is a measure of uncertainty in determining how likely each attack is, and for this reason Howard and Lipner [3] recommend using threat levels instead of raw numbers in prioritization.

Finally, the risks are mitigated, for example by fixing any issues found. Minimizing the risk that an attack will happen or removing the feature if it turns out to be too risky are also options.

According to Scandariato et al. [4], computer science students using STRIDE for the first time were able to find 1.8 threats per hour, were able to find 64-69% of all threats and overlooked 19-24% of threats.

2.1.2 Exploit Chain Analysis

!Fixme **Research this!** Fixme!

2.1.3 Assurance Cases

!Fixme **Research this!** Fixme!

2.1.4 Attack trees

!Fixme **Research this!** Fixme!

2.1.5 Threat Analysis at F-Secure

!Fixme

- **The flow of analysis on F-Secure**
- **Might contain secret material while writing, start only after text has been moved to F-Secure servers**

Fixme!

2.2 Authentication and Authorization systems

!Fixme

- **Some history again**
- **Common flows in systems**
- **Examples of different kinds**
- **Importance of good system and its security**

Fixme!

2.2.1 OneID

!Fixme

- **Introduction to OneID as a system**
- **Enough architecture to build the analysis on later**
- **Might contain secret material while writing, start only after text has been moved to F-Secure servers**

Fixme!

Chapter 3

Research Question

!Fixme

- **The threat analysis and threat assessment flows that are done and should be done in industry contexts, and with regards to OneID in particular.**
- **What security flaws happen and what are especially dangerous in identity management systems, especially in the context of the human factor.**
- **Add evaluation criteria!**

Fixme!

Chapter 4

Analysis

Chapter 5

Evaluation

!Fixme **How well did it go gets put here.** Fixme!

Chapter 6

Conclusions

!Fixme Wrap-up here. Basically the whole thing in a nutshell. Written at the end. Fixme!

Bibliography

[1] Richard Baskerville. Information systems security design methods: implications for information systems development. *ACM Computing Surveys (CSUR)*, 25(4):375–414, 1993.

[2] Mariana Gerber and Rossouw Von Solms. From risk analysis to security requirements. *Computers & Security*, 20(7):577–584, 2001.

[3] Michael Howard and Steve Lipner. *The security development lifecycle*, volume 8. Microsoft Press Redmond, 2006.

[4] Riccardo Scandariato, Kim Wuyts, and Wouter Joosen. A descriptive study of microsoft's threat modeling technique. *Requirements Engineering*, 20(2):163–180, 2015.

[5] Adam Shostack. *Threat modeling: Designing for security*. John Wiley & Sons, 2014.

Appendix A

First appendix

!Fixme **Any appendices here.** Fixme!