Srinivas Nayani

**DESIGNING SECURE SOLUTIONS FOR EMBEDDED SYSTEMS**

**TITLE OF THESIS**

First name Last name

Thesis

Term (e.g.Spring) year

Name of degree programme

Oulu University of Applied Sciences

abstract

Oulu University of Applied Sciences

Degree programme, option

Author(s):

Title of the bachelor’s thesis:

Supervisor(s):

Term and year of completion: Number of pages: x + x appendices

An abstract is a concise, independent presentation of the thesis. It is written when the thesis is finished. The abstract is always written with full sentences and with passive voice. The abstract introduces briefly the subject and objective of the thesis, methodology and the essential results.

The contents are usually divided into

1. The subject and objectives
2. The methodology, execution and progress
3. The results and conclusions.

The abstract should not be longer than a page written with line spacing 1.

Replace these instructions with your own text.

Keywords: x, x, x

(List 3–7 keywords that describe your thesis. Use, for example, keywords from the following controlled vocabulary thesauri:

MeSH <http://www.yso.fi/onto/mesh/conceptscheme>

Agriforest <http://www-db.helsinki.fi/agri/agrisanasto/Welcome_eng.html>

Helecon <http://helecon3.hkkk.fi/helevoc/?lang=eng&dbname=MIX>)

PREFACe

The preface tells, where and when the work has been done and who has ordered the work. It also introduces the role of the tutoring teacher and the client’s representative and the source of possible external help.

The thesis can include a preface, in which the customer, supervisors and other possible supporters and assistants are thanked. The preface can also include other thesis-related, interesting information, which would not otherwise be mentioned in the text. The preface is dated and signed.

Replace these instructions with your own text.

contents

[abstract 3](#_Toc446494280)

[PREFACe 4](#_Toc446494281)

[contents 5](#_Toc446494282)

[VOCABULARY 7](#_Toc446494283)

[1 introduction 8](#_Toc446494284)

[2 Drivers 9](#_Toc446494285)

[2.1 Impacts for a compromised security 9](#_Toc446494286)

[2.2 Second Driver 9](#_Toc446494287)

[2.2.1 First subtitle 10](#_Toc446494288)

[2.2.2 Second subtitle 10](#_Toc446494289)

[3 Security Attacks 11](#_Toc446494290)

[~~3.1 Tables~~ 11](#_Toc446494291)

[~~3.2 Figures~~ 12](#_Toc446494292)

[~~3.3 Formulas~~ 12](#_Toc446494293)

[3.4 Attack types 13](#_Toc446494294)

[3.4.1 Insider Attack 13](#_Toc446494295)

[3.4.2 Plain text attacks 13](#_Toc446494296)

[3.4.3 Cipher text attacks 13](#_Toc446494297)

[3.5 Classification of attackers 15](#_Toc446494298)

[3.6 Levels of difficult 15](#_Toc446494299)

[4 Techniques for Classificating vunerabilities 17](#_Toc446494300)

[4.1 Classification by SDLC 17](#_Toc446494301)

[4.2 Classification by genesis 17](#_Toc446494302)

[4.3 Classification by location in object models 17](#_Toc446494303)

[4.4 Classification by effected technology 17](#_Toc446494304)

[4.5 Classification by errors 17](#_Toc446494305)

[4.6 Classification by enabled attack scenario 17](#_Toc446494306)

[4.7 CLASP Classification 17](#_Toc446494307)

[5 Taxonomy on Atacks and vunerabilities 18](#_Toc446494308)

[5.1 PLOVER 18](#_Toc446494309)

[5.2 CWE 18](#_Toc446494310)

[5.3 CAPEC 18](#_Toc446494311)

[6 Security in legecy systems 19](#_Toc446494312)

[7 Drivers for security 20](#_Toc446494313)

[7.1 Impacts for a compromised security 20](#_Toc446494314)

[8 Design challenges 21](#_Toc446494315)

[9 Basic security concepts 22](#_Toc446494316)

[9.1 Symmetric Cryptography 22](#_Toc446494317)

[9.2 Asymmetric cryptography 22](#_Toc446494318)

[9.3 Hashing algorithms 22](#_Toc446494319)

[9.4 Encryption algorithms 22](#_Toc446494320)

[10 open standards 23](#_Toc446494321)

[11 Protection mechanisms 24](#_Toc446494322)

[11.1 Hardware security 24](#_Toc446494323)

[11.2 Software security 24](#_Toc446494324)

[12 Conclusion 25](#_Toc446494325)

[13 References 26](#_Toc446494326)

[14 Appendix 28](#_Toc446494327)

[14.1 Some simple facts about designing secure systems 28](#_Toc446494328)

[REFERENCES 29](#_Toc446494329)

VOCABULARY

If the same abbreviations or entries for quantities, units or drawings are repeated in the thesis, they are listed in a vocabulary with explanations. Depending on the needs, the heading can be, for example, ABBREVIATIONS, SYMBOLS or TERMS. The list is written in an alphabetical order. The SFS standard 4600 and SI system are used in alphabetising.

Remove this page if you do not need it.

# introduction

Security should be treated as integral part of system which should be considered right from product design stage. It should be built into the system across multiple levels often referred as layered approach. It is not feasible or often viable to design totally a fool-proof system. So designers should rather focus on systems that can be difficult to compromise and reduce the risk to an acceptable level.

# Drivers

Replace these texts with your own.

When using the Heading 1 style, each main chapter will always start a new page. The following normal text or a subheading is separated from the main heading with two empty lines (automatically included in the heading styles). A subheading within a main chapter is preceded and followed by one empty line (automatically included in the heading styles). The main heading will be capitalised (automatically included in the Heading 1 style).

## Impacts for a compromised security

If you use subheadings, use at least two: if the subheading is 1.1, you also have to have subheading 1.2. Separate the heading number and text with an empty space. Do not use a full stop after the last digit of the subheading. A heading text continuing to the next line is aligned to the first letter, not the number. Headings should be short and informative. Headings are not clauses or questions.

Texts in chapters are aligned to the left: all lines start from the place without any indentation. The chapters are separated with an empty line. It is also possible to use justified text. The text is hyphenated. Remember that a chapter is longer than just one sentence. One chapter always contains one subject. It is recommended to vary the length of the chapters.

## Second Driver

Three levels of headings are usually enough. The numbering of the decimal grouping will be marked as follows:

* 5 MAIN HEADING (Heading 1)
* 5.1 Subheading (Heading 2)
* 5.1.1 Subtitle (Heading 3).

### First subtitle

Text here

### Second subtitle

Text here

Heading without numbering

If necessary, it is possible to also use unnumbered subheadings. They are not included in the table of contents, and the font size is 12.

# Security Attacks

~~Replace also these texts with your own.~~

~~You can improve the intelligibility and readability of the text with tables, diagrams and appendices. The tables and diagrams are independent and self-explanatory, and the text describes the essentials or conclusions presented in them. If you have a lot of diagrams and tables, place some of them in appendices. Do not present the same facts both as a diagram and a table. Usually, it is not worth using a diagram or a table to present one or two facts. Leave an empty line before and after diagrams and tables. Also, leave an empty line between the title and the diagram or the table.~~

~~Number the diagrams and tables consecutively, both separately. All those, which are not tables, are diagrams. The term diagram is used, for instance, for photos, maps and drawings. Refer to each diagram or table in the preceding text. Use a leading text before the diagram or table, do not discuss them directly after the heading or title.~~

## ~~Tables~~

~~The tables should be as clear and self-explanatory as possible. Use titles in rows and columns to organise the contents of the tables. Number the tables. The titles should clearly state the subject of the table. The title is placed over the table. The word~~ *~~TABLE~~* ~~is written in capital letters and in italic. The name of the table is also written in italics. You can use previously published tables, too. In this case, place the source at the end of the title (table 1). Align the title and the table similarly with the body text; the length should also be similar. Use borders and coloured shading with consideration in order to improve the clarity of the table and cells. If necessary, footnotes can be placed under the table.~~

~~TABLE 1. The thermal loss capacity of a heating system in outdoor temperatures of –25 ˚C…–10 ˚C (1, p. 23)~~

|  |  |
| --- | --- |
| **~~Section~~** | **~~Thermal loss capacity [W]~~** |
| ~~Boiler~~ | ~~3,000~~ |
| ~~Piping~~ | ~~6,198~~ |
| ~~Accumulator~~ | ~~5,717~~ |
|  |  |
| ~~In total~~ | ~~14,915~~ |

## ~~Figures~~

~~Align the figure and its title with the body text. The title~~ *~~FIGURE~~* ~~is placed below and written in italics, as is the name of the figure. The source of a referenced figure is placed in brackets after the title. Avoid dark colours. Use coloured graphics when the colours are necessary in order to understand the diagram. (Figure 1.)~~

~~~~

~~FIGURE 1. A flexible claw clutch (2, p. 368)~~

## ~~Formulas~~

~~Formulas are numbered and the quantities presented in them are explained. The numbers of the formulas are aligned to the right on the same line as the formula itself. In text, they are referenced with a number. Variables and quantities are written in italics; measures are written in a normal style. (Formula 1.) Chemical formulas can be presented as figures, which are numbered and headlined normally.~~

~~The impulse of a torque is calculated with formula 1 (3, p. 93).~~

*~~K = Mt FORMULA 1~~*

*~~K~~* ~~= impulse of torque (kgm~~~~2~~~~/s)~~

*~~M~~* ~~= torque of strength (Nm)~~

*~~t~~* ~~= time of influence of the torque (s)~~

## Attack types

### Focused attack

Focused attacks are highly focused on particular or specific kind of systems or environment or ecosystem. This kind of attack has no limitation on time, money and resources. Most practical examples of this kind of attacks are to target defence installations, penetrating enemy communication lines etc.. Most recent example of this kind of attack is building of “stuxnet” whose is targeted to attack only Siemens systems in Iran.

### Cryptanalytic attacks

Cryptanalytics is a study of techniques to unravel the meaning of encrypted text without access to secret keys. Cryptanalysis techniques are used to decrypt the ciphered text without really accessing the encryption keys. Doing a cryptanalysis requires working knowledge of system and knowing internals of cryptography which in practice means uncovering the secret key. These attacks are briefly classified as plain and cipher text attacks.

#### Known plain text attack

In this kind of attack, attacker will have access to at least one pair of plain text and corresponding cipher text which are not explicitly chosen and act as inputs for further analysis. These plain texts are usually obtained via eavesdropping or from parities who already possess encryption key. The results are used to break rest of the encryption in the system by tracing out the secret key.

#### Chosen plain text attack

In this kind of attack, attacker feeds in pre-chosen text into the cipher after which analyses the result and in worst case can figure out secret key.

Two forms of chosen plain text attacks are batch chosen plain text attack and adaptive plain text attack. Batch chosen plain text chooses all the plain texts before it analyses the ciphered text where in adaptive chosen text attack a cryptanalyst requests for additional cipher texts after analysing the results of previous cipher operations.[5] .

Chosen-plain text attacks are more powerful compared to other plain text attacks as the attackers gets hold of many copies of chosen plain and cipher text pairs through which chances of success is multiplied.

#### Known cipher text attacks

Under known cipher text attacks, attacker has access to bunch of cipher texts mostly obtained either by eavesdropping or stealing. Also here the attacker will not have access to more cipher texts or will not have luxury of choosing cipher text or neither can produce more. This is certainly one of the weakest attacks as the attacker will have nothing to work against other than few cipher texts in hand.

#### Chosen cipher text attacks (CCA)

In chosen cipher text attack, attacker will be able to produce clear text from set of pre-selected ciphered text messages from decryption oracle.[3]

Chosen cipher text attacks can be adaptive or non-adaptive. Under non adaptive cipher text attacks, attacker choses certain cipher texts in advance for decrypting them. The clear texts obtained are not used for next cipher operations. Adaptive cipher text attacks are context based and employ by changing the text input obtained as a result of previous operations into cipher.[1,3]

#### Lunchtime Attack

This is also referred as midnight attack or CCA1 attack . This is a kind of attack targeted by attackers when the owner or user of the system is away or often when system is not logged in. This is with idea that system is vulnerable and is often less or no resistance when there is no active user which otherwise will be more challenging to penetrate. During this time, the attacker will generate a pre-chosen cipher text quarries which are valid until period of time after which penetrating will be increasingly difficult or attacker should show improved ability for achieving his objectives.[2]

#### Adaptive cipher text attack

Adaptive cipher text attack is also referred as CCA2 attack and is stronger in nature compared to CCA1. This attack relies on approach to select cipher dynamically at runtime when ever attacker is posed of challenge.

This is interactive based attack where attacker sends stream of ciphered texts to be decrypted and subsequent ciphered texts are choosing depending on responses from the system.[4]

One increasing order from weakness to strength, above attacks can be sorted as known cipher text attack, known plain text attack, chosen plain text attack, chosen cipher text attack.

### Network attacks

In this world of ever increasing networking, systems have become very attractive targets for external attacks through network. Networking attacks most rely on monitoring, spoofing or masquerading of network traffic.

#### Passive attacks

Passive attack monitors the unprotected or weekly encrypted communication between two nodes for capturing authentication information or passwords which can be passed on to parties who would or has ability to compromise the system.

#### Active attacks

Active attackers penetrate the system by circumventing the security and breaking the protection existing systems . They can cause undesired effects by executing their malicious code and injecting viruses or Trojan horses. Active attacks can have varied effects right from minor to bringing down the whole system or network especially if the attack is on servers.

#### Insider attack

These are the attacks that are perpetrated from inside the organization or persons who have genuine access to the system. Insider attacks come from disloyal persons, persons with malicious intent, dissatisfied employees from an organization.

#### Phishing attack

Phishing attacks are attacks where attacker will design fake almost identical web sites through which they direct users to login with credentials. These credentials will be recorded and used by attackers to log into proper websites that can result in stealing of vital information or can even result in financial frauds if the target is banking sites.

#### Hijack and spoof attacks

Attacker can hijack the communication sessions and disconnect the one of the node. Under a hijacked session, other connected party is still under the impression that as if it is communicating with original party and can still pass some vital private or secret information.

Under spoof attacks, attackers change the source address of network packets by which they packets are disguised as coming from other valid sources in an effort to circumvent firewall.[6]

## Classification of attackers

Class1: Clever Outsiders

Class2: Knowledgeable insiders

Class3: Funded Organizations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resource | Hacker class  (Class1) | Academic (Class 2) | Organized (Class 3) | Government (Class 4) |
| Time | Limited | moderate | Large | Large |
| Budget | < $1000 | $10K – $100k | >$100k | Unknown |
| Creativity | Varies | High | Varies | Varies |
| Detectability | High | High | Low | Low |
| Target | Challenge | Publicity | Money | Varies |
| Number | Many | Moderate | Few | Unknown |
| Organized | No | No | Yes | Yes |
| Release Info | Yes | Yes | Varies | No |

## Levels of difficult

TABLE 1. Attack difficult

|  |  |  |
| --- | --- | --- |
| **Level** | **Name** | **Description** |
| 1 | None | Primitive ,No special tools or skill needed |
| 2 | Intent | Minimal skills needed to compromise the system |
| 3 | Common tools | Technically competent, Can be dealt with tools available in market |
| 4 | Unusual tools | Can be compromised with tools can be available to most people |
| 5 | Special tools | With specialized tools and with expertize only available in universities or government |
| 6 | In Laboratory | Major effort needed, Only available to few facilities in the world. |

## Techniques for Classificating vunerabilities [7]

Attacks can be classified based on certain parameters like their objectives, impacts, Origin ,phase of introduction, technology employed and exploitability etc.. Classification of attacks and vulnerabilities can be help us to understand the tools, remedies , approaches that can help us contain , trace and overcome these vulnerabilities.

### Classification by SDLC (Software Development LifeCycle)

Vulnerabilities can be classified based on the phase of the software development life-cycle they usually creep in. Some of the popular lifecycle phases that have higher chance to see system vulnerabilities are design , testing ,deployment and maintenance phases.

Typical examples of vulnerabilities in design phase can include wrong choice of OSS components , protocol, algorithms , using of untested reusable libraries or code that may not be really fool proof. Often vulnerabilities in design phases are easy to exploit and difficult to plug.

During maintenance phase ,Systems can be prone to vulnerabilities that come along with improper bug fixes and components that aren’t updated on timely and priority basis. In case system is using OSS or reusable closed sources, system integrators need to update the system with latest fixes from upstream , leaving them unmaintained can expose holes in the system makes the system an attractive target for exploit. Vulnerabilities that creep in due to coding errors, relaxed compiler settings, bad design of APIs are few that evolve during implementation phase. Some of these can be addressed by employing more stricter process approach using code analysis tools and fine tuning the compiler optimization options.

In most cases , software and hardware will be tailor made for relaxing certain hard Classic examples of vulnerabilities creeping in testing phase can be debug holes either software or access points in hardware that can

### Classification by attackers objective

One of the most prominent approach is to enumerate the attacks is by attackers objectives like gaining root access and higher privilages, creating denial of service, stealing confidential and sensitive data, malicious code execution, Integrity and security policy violation.

### Classification by attacks by their location in OSI model and their origin

One approach in classifying the vulnerabilities is by deciding where exactly they appear in 7 layered OSI reference model. In practice this means to segregate the vulnerability into one of the seven baskets (Application, presentation, session, network, link and physical layer). Attacks can also be enumerated depending on their origin location in network like local system, intranet (ethernet network), internet , wireless network.

This kind of classification may not be appropriate at all times as many of the times vulnerabilities can fall between layers and hence difficult to segregate on this layered approach.For example , it is not often easy to decide on whether the vulnerability is exactly in OS or application layer or both as contention results which is right and wrong.

### Classification by effected technology

Systems get vulnerable though holes created by string exploits and buffer overflows which are not unusual in C language. Likewise systems can be prone to vulnerable to attacks exploiting meta characters vulnerabilities like LDAP and SQL Injection that can happen with database languages.

Systems that run code with memory leaks, malicious code are vulnerable to resource exhaustion if the code can block the system resources and thus can result in DoS especially in the case of embedded systems where resources are limited. Classic resource hungry operations are TCP SYN Flooding, improper handling of resources which is resource hungry.

This kind of technology classification may not suit for vulnerabilities that spawn across technologies not just limited to one.

### Classification by errors

System can be made vulnerable as a result of improper code design and programmatic errors that creep in during implementation phase. To quote a few are, double free memory, executing content from malicious memory locations or locations program either did not allocate of have any control on.

At times ,unclosed holes that are left in production code to accommodate debugging for diagnostic purposes can spell trouble and can be exploited by attackers.

### Classification by enabled attack scenario

Vulnerabilities can also be classified based on precise type of attack scenarios. As seen above, Denial of Service is an effect that can happen due to multiple reasons like, memory leaks or buffer overflows etc.. So it makes sense to classify the vulnerabilities on the nature of attack scenario rather than based on effects.

Examples of attack scenarios are Cryptographic attacks , network attacks , Secure storage attacks , Software attacks ,entropy attacks, Malicious code execution. Techniques employed to execute these attacks can be chosen-known/cipher text attacks, compromised boot sequence, string exploits.

### CLASP Classification

CLASP (Comprehensive light weight application security process) enumerates vulnerabilities based on software events and conditions that are responsible for the vulnerability.

#### Range and type errors

Generic range type errors are errors due to buffer overflow , stack and heap overflow , integer overflow , truncation errors, signed and unsigned errors, Integer coercion errors, unchecked indexing of arrays, NULL character misplacing for string buffers , NULL pointer dereferencing , usage of freed memory, format string , code injection into data areas of memory.

#### Environmental errors

Environmental errors can be as a result of resources exhaustion (for ex: sockets , kernel objects , file descriptors, memory), execution of untrusted code and data, system variable manipulations (for ex , system paths , library paths ..etc) , spoofing of system events, failure to protect secure data and keys, TRNG generation failure and insufficient entropy for PRNG.

#### Synchronization and timing errors

Situations leading to synchronization and timing errors are race conditions in code (unlocking code via kernel objects) , race condition in signal handlers, improper references for symbolic names which change at runtime , failure to drop user privileges at right times soon after task is accomplished, leaking sensitive information through error messages , time to check and time to use errors (for example , resources can change their state between a window of time lag between their validation and actual usage.

#### Protocol Errors

Protocols errors are one that usually arises out of protocol, algorithm errors that are as a result of improper use or wrong choices. Such vulnerabilities that are from failure to check for certification expiration and revocation, key exchange without proper authentication, failure to encrypt communication, failure to do integrity check where ever needed, usage of hardcoded and stored passwords or keys, trusting certain IP address or range of IPs that can be spoofed easily, using of broken, week or risker cryptographic algorithms, improper usage of OSS components and failure to protect confidential and sensitive data.

#### Generic Errors

Errors that are enumerated based their generic nature are improper error and exception handling, improper break and jump instructions in code, ignoring return values from functions ,uninitialized variables, failure to free unused resources and memory and unintentional assignment when comparison two values etc.

## Popular Dictionaries for Attack Taxonomy

MITRE a government funded non profit organization that publishes and controls standards to be used by community. Below are the popular dictionaries of publicly known information security vulnerabilities and exposures maintained by MITRE.

### PLOVER

PLOVER is a primary list of working examples intended for researchers that lists over 1400 real world vulnerabilities by their CVE IDs organised as a conceptual framework. This framework offers a platform for discussion for further analysis and describing them in further detailed manner. PLOVER is targeted for those who are engaged in vulnerabilities analysis in an effort to understand and communicate them in more abstract level.[9]

### CWE

CWE stands for Common Weakness Enumeration which essentially deals with underlying software weakness in general but not specific to particular instance in system. Vunerabilites can emerge from weakness and may have potential for getting targeted. Goal of CWE is to educate the programmers or system designers to For example ,[CWE-367](https://cwe.mitre.org/data/definitions/367.html) is time to check and time to use weakness spotted in software but not limited to any specific component or instance in any system.

### CVE

CVE stands for Common Venerability Enumeration which precisely describes a certain pin-pointed instance in system through which exploits can happen.For example [CVE-2015-7547](http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-7547) , points to specific venerable instance in eglibc for an attack.

### CAPEC

While CWE is a list of software weakness types, [Common Attack Pattern Enumeration and Classification (CAPEC™)](http://capec.mitre.org/) is a list of the most common methods attackers use to exploit vulnerabilities resulting from CWEs. Used together, CWE and CAPEC provide understanding and guidance to software development personnel of all levels as to where and how their software is likely to be attacked, thereby equipping them with the information they need to help them build more secure software.[10]

## Attack vectors

Attack vectors are typical routes via which an attacker can gain access and there after exploit the vulnerabilities in the system in order to achieve his objective. Typical attack vectors are eves dropping, brute force attacks, injecting crafted packets, reverse engineering, fabrication by counterfeit assets of a product. Attack vectors are also refereed as attack types.

# Safe Design solutions

Let us discuss here about mechanical design approaches that should br considered for bringing out a secure product.

# 

# Security in legecy systems

Replace also these texts with your own.

This thesis model contains a table of contents with the correct formatting. When using this model, you can update the table of contents by placing the mouse cursor on the first line of the table on the left margin. Then press F9 to open a selection list. Select **Update the entire table** and click OK.

If the table of contents was not updated correctly, check that your titles and headings have the following styles:

* Main chapter Heading 1
* First subheading Heading 2
* Second subheading Heading 3.

# Drivers for security

## Impacts for a compromised security

# Design challenges

# Basic security concepts

## Symmetric Cryptography

## Asymmetric cryptography

## Hashing algorithms

## Encryption algorithms

# open standards

# Protection mechanisms

## Hardware security

## Software security

# Conclusion

# References

1. <https://en.wikipedia.org/wiki/Chosen-ciphertext_attack>
2. <https://en.wikipedia.org/wiki/Chosen-ciphertext_attack#Lunchtime_attacks>
3. <http://www.tech-faq.com/known-ciphertext-attack.html>
4. <https://en.wikipedia.org/wiki/Adaptive_chosen-ciphertext_attack>
5. <https://en.wikipedia.org/wiki/Chosen-plaintext_attack>
6. <http://computernetworkingnotes.com/network-security-access-lists-standards-and-extended/types-of-attack.html>
7. <http://homes.cerias.purdue.edu/~pmeunier/aboutme/classes_vulnerabilities.pdf>
8. <https://cwe.mitre.org/about/sources.html>
9. <https://cwe.mitre.org/documents/sources/PLOVER.pdf>
10. <https://cwe.mitre.org/about/faq.html#A.2>

# 

# Appendix

## Some simple facts about designing secure systems

* Strive for simplicity
  + Designing complex secure systems can backfire as they can contain exploitable bugs.
* Do not go for untrusted or unnecessary security mechanisms.
* Minimize the trusted components in the system.
* Strive for total Isolation of secure operations from nonsecure ones.

REFERENCES

Remove the instruction texts before the actual list of references.

In the text sources are indicated as in-text references. The purpose of in-text references is to inform the reader whose text or thoughts are referred to and to provide the reader with an opportunity to verify the authenticity of the references and sources. The Copyright Act states that sources must be acknowledged. All in-text references must be found in the list of references. In-text references help the reader to find in the list of references the book, article or other source which the author refers to. (See examples and more thorough information on how to make in-text references in chapter 5.5 In-text references, p. 34-40 in Bachelor’s thesis instructions of Oulu UAS.)

Every source referenced in the thesis must also be listed in a list of references at the end of the thesis. The sources are arranged either in an alphabetical order or according to the number reference system. More information and examples can be found in the Bachelor’s thesis instructions of Oulu UAS in chapter 5.6 References, p. 40-46.

Below you can see an example of the reference list using the number reference system:

1. Kulha, Antti 2010. The effect of insulating the thermal system of a heating plant into the consumption of fuel. Oulu: Oulu University of Applied Sciences, Degree Programme in Building Services. A thesis.
2. Airila, Mauri – Ekman, Kalevi – Hautala, Pekka – Kivioja, Seppo – Kleimola, Matti – Martikka, Heikki – Miettinen, Juha – Niemi, Erkki – Ranta, Aarno – Rinkinen, Jari – Salonen, Pekka – Verho, Arto – Vilenius, Matti – Välimaa, Veikko 1995. Designing machinery parts. Juva: WSOY.

Technical schemas. 2000. Tampere: Tammertekniikka Oy.

**APPENDICES**

Appendices are meant for data, which seems necessary, but is not suitable to be included in the text. The appendices must also contain appropriate source references if they originate from sources outside the thesis.

Appendices can include, for example, a memorandum of initial data, tables, data sheets, drawings, diagrams, programme code listings and other illustrative material. If the appendix is not referenced in the text, it is redundant.

If you have more than five appendices, they are listed after the sources. If you do not have more than five appendices, they are listed in the table of contents.

Below you will find a model for a list of appendices.

APPENDICES

Appendix 1 Thesis initiation document

Appendix 2 An example of a multi-page appendix

Appendix 3 Appendix heading

Appendix 4 Appendix heading

Appendix 5 Appendix heading

Appendix 6 Appendix heading

THESIS INITIATION DOCUMENT

Author

Customer

Customer’s contact person and information

Title

Description

Objectives

Target schedule

Date and signatures

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.

An example of a multi-page appendix. The page numbering is placed automatically in the header, the number of the appendix has to be changed.