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](#_Toc448513313)

[PREFACe 4](#_Toc448513314)

[contents 5](#_Toc448513315)

[VOCABULARY 7](#_Toc448513316)

[1 introduction 8](#_Toc448513317)

[2 Drivers 9](#_Toc448513318)

[2.1 Impacts for a compromised security 9](#_Toc448513319)

[2.2 Second Driver 9](#_Toc448513320)

[2.2.1 First subtitle 10](#_Toc448513321)

[2.2.2 Second subtitle 10](#_Toc448513322)

[3 Security Attacks 11](#_Toc448513323)

[~~3.1 Tables~~ 11](#_Toc448513324)

[~~3.2 Figures~~ 12](#_Toc448513325)

[~~3.3 Formulas~~ 12](#_Toc448513326)

[3.4 Attack types 13](#_Toc448513327)

[3.4.1 Focused attack 13](#_Toc448513328)

[3.4.2 Cryptanalytic attacks 13](#_Toc448513329)

[3.4.3 Network attacks 15](#_Toc448513330)

[3.5 Classification of attackers 16](#_Toc448513331)

[3.6 Levels of difficult 17](#_Toc448513332)

[3.7 Techniques for Classificating vunerabilities [7] 17](#_Toc448513333)

[3.7.1 Classification by SDLC (Software Development LifeCycle) 17](#_Toc448513334)

[3.7.2 Classification by attackers objective 18](#_Toc448513335)

[3.7.3 Classification by attacks by their location in OSI model and their origin 18](#_Toc448513336)

[3.7.4 Classification by effected technology 19](#_Toc448513337)

[3.7.5 Classification by errors 19](#_Toc448513338)

[3.7.6 Classification by enabled attack scenario 20](#_Toc448513339)

[3.7.7 CLASP Classification 20](#_Toc448513340)

[3.8 Popular Dictionaries for Attack Taxonomy 21](#_Toc448513341)

[3.8.1 PLOVER 22](#_Toc448513342)

[3.8.2 CWE 22](#_Toc448513343)

[3.8.3 CVE 22](#_Toc448513344)

[3.8.4 CAPEC 22](#_Toc448513345)

[3.9 Attack vectors 23](#_Toc448513346)

[4 product Design solutions 24](#_Toc448513347)

[4.1 Product housing 24](#_Toc448513348)

[4.1.1 External Interfaces 24](#_Toc448513349)

[4.1.2 Anti-tamper mechanisms 25](#_Toc448513350)

[4.2 PCB design and routing 27](#_Toc448513351)

[4.3 Memory and bus protection 27](#_Toc448513352)

[5 Security in legecy systems 29](#_Toc448513353)

[6 Drivers for security 30](#_Toc448513354)

[6.1 Impacts for a compromised security 30](#_Toc448513355)

[7 Design challenges 31](#_Toc448513356)

[7.1 Resource limitations 31](#_Toc448513357)

[7.2 Deployment challenges 31](#_Toc448513358)

[7.3 Memory constraints 31](#_Toc448513359)

[7.4 No One solution or silver bullet 31](#_Toc448513360)

[8 open standards 32](#_Toc448513361)

[9 Device security 33](#_Toc448513362)

[9.1 Hardware security 33](#_Toc448513363)

[9.2 Software security 33](#_Toc448513364)

[9.2.1 Multiple Independent layered security (MILS) 33](#_Toc448513365)

[9.2.2 Secure boot 37](#_Toc448513366)

[9.2.3 Role of Operating systems 37](#_Toc448513367)

[9.2.4 Application level security 37](#_Toc448513368)

[10 Conclusion 38](#_Toc448513369)

[11 References 39](#_Toc448513370)

[12 Appendix 41](#_Toc448513371)

[12.1 Some simple facts about designing secure systems 41](#_Toc448513372)

[REFERENCES 42](#_Toc448513373)

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

Outsiders are external attackers exploiting certain weakness in the system who has certain knowledge of the system.

Class2: Knowledgeable insiders

Insider attackers are the one who have needed technical expertise as well as access to specialized tools to break into the system.

Class3: Funded Organizations

Organizations fund attacks with focused objectives. Here attackers have in-depth knowledge ,no restriction on funding and are equipped with highly specialized tools to break open the systems.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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[13]

|  |  |  |
| --- | --- | --- |
| **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.

# product Design solutions

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

## Product housing

Typical attack hard point in this case would be product enclosure and attack vector where attackers will make an effort to open the casings and access to internal circuitry and components. Product designers needs to prevent an easy access to the product internals by concealing the access points through which the device can be opened. Some of the safe enclosure techniques currently in use are

* Designing product with a single piece outer shell.
* Using high melting point glues where ever needed.
* Designing in such a way that opening of device needs complete destruction.
* In addition to above, leave surface points for device accessibility for service personal who can open the device with specialized tools.

### External Interfaces

External interfaces like proprietary connectors, Ethernet, RS232, USB, Firewall, JTags, Wireless (802.11) and Bluetooth are vital lifeline for the product to outside world for proper functioning of device. In addition to carrying the usual device operations these interfaces also aid in regular maintenance tasks, on-field diagnostic procedures and field programming.

External interfaces are always attractive targets for hackers as root of the attack tree can originate from here. Attackers indulge in probing, sniffing, flooding and push malformed packets through these interfaces. Product designers should be able to defeat all possible attacks that originate from these external interfaces. Designers should keep in mind below points while interfaces are designed.

* Device shall transmit only non-sensitive and public information in clear text format.
* Encrypt all sensitive and confidential information that has to be exchanged over interfaces.
* Adequate protection inside the device to defeat spoofing, malformed packets. For ex , Hardening of OS, strong firewall etc..
* Strict no obfuscation policy, attackers are always ahead of designers and can uncover them easily.
* All the diagnostic ports , backdoor interfaces and JTAG connecters to be removed from production software images. Closing them by employing techniques like blowing resisters and fuses is not the best approach as they can be reopened by attackers.

### Anti-tamper mechanisms

Attackers try to gain physical access to device by tampering to know confidential information and know working internals device. Some of the tamper mechanisms worth considering by product designers are

#### Tamper resistance

Tamper resistance relies on restricting physical access to devices. Devices shall be housed and constructed with specialized tamper resistance materials and mechanisms like hardened steel enclosures, usage of one way screws and epoxy coating materials, tight airflow channels, usage of security bits, encapsulate the circuit and critical components to prevent intentional probing and tampering due to environment hazards.

Products with are well protected housing would require partial if not complete destruction of device to open up. It is also essential for designers to lay emphasis on such a design that will leave a visible and clear evidence if a tamper attempt is made.

#### Tamper evidence

Tamper evidence mechanisms are designed to ensure that visible evidence or trails left for an attempted break in. These mechanisms do not protect the device or the confidential data there in but to raise awareness that there has been an attack on the system. Some of the widely used mechanisms are brittle packages, crazed aluminum and polished packages, bleeding paints and holographic tapes. All the above mechanisms will leave a definite very well evident cracks or damages on the surface that are hard to be unnoticed.

#### Tamper detection [12]

Tamper detection can be done by installing relevant sensors which detect the tamper and trigger the relevant response mechanisms. Some of the widely used tamper detection mechanisms that are built into devices are

* Switches that detect mechanical movement.
* Sensors that can detect and inform external environmental changes.
* Closed circuit and cables that are wrapped in and around the device for detecting a attempted break, tamper or modification.
* Monitoring for the changes in voltages, clock frequencies from and to the chips.

#### Tamper response

Systems should trigger shutdown and disable themselves in response to tamper detection. They should be designed with capability to log and generate forensic data for further analysis post tamper detection that can be accessed by service personal. Designers should also consider techniques such as erasing the sensitive data in response to tamper detection. Products that house confidential data, keys should protect the data by resorting to erasing methods. Confidential data is either stored in RAM or ROM. Erasing random access memory is relatively easier and accomplished by dropping the voltage levels which will effectively clear the memory contents. In case of ROM, a ROM overwrite may be needed. There are even cases where system employ to ultimate mechanism of physical destruction by shorting the circuits and rendering device inoperable.

Further care has to be taken these tamper response mechanisms does not trigger in case of unintentional actions, accidentally or by environmental factors.

## PCB design and routing

Enough precautions should be taken while designing circuitry boards for not letting an easy access to components like FPGA , processors and memories. Easy access to these components and circuitry can help attackers in reverse engineering the product. Designers should look into advanced chip packaging technologies while they design PCBs for ex : COB (Chip on Board) packaging, CIB (Chip In Board) packaging, Ball Grid Array packaging. Product should demark or erase all the chip markings either by black topping or using other methods like small sander or etching. Failure to remove the chip markings will leave potential hints to attackers to learn about the chips behavior and their usage by referring their datasheets.

Sensitive components and circuitry lines should be concealed by using Epoxy material around them. Care should be taken during PCB production not to leave test points visibly open and every attempt should be made to obfuscate the trace paths and critical lines into inner layers of PCB. Even electromagnetic emissions from product can act as potential attack points which attackers can monitor to determine secret information. So every step needs to be taken to minimize the emissions by installing appropriate shielding and taking care of unprotected I/O buss from ESDs. A well designed power lines and grounding can reduce noise levels and emissions. All unused GPIOs should be either disabled or to be fixed to predetermined state.

## Memory and bus protection

Designers should be aware that address and control bus lines are prone to probing and traffic is not secure. Designers need to design their systems to perform secure operations either inside SoC or components that shall not use the unsecure bus in the system. Memory devices like RAM and ROM are known to be unsecure. Even though systems are designed with proper tamper detection and response mechanisms like complete wipe out or erasing of data, but there is high chance that traces of data can still be being left out. Although the product supports security fuses, boot-block protection mechanisms, these can be bypassed by die-attacks and chip-decapping as attackers can reproduce and recreate cryptographic keys or remove security bits. Designers should also take steps to limit the time the secure data can be stored in memory, should erase information once their need is done.

# 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

## Resource limitations

## Deployment challenges

## Memory constraints

## No One solution or silver bullet

# open standards

# Device security

## Hardware security

## Software security

### Multiple Independent layered security (MILS)

Legacy secure systems were designed around secure kernel and trusted computing base , with the idea that all the security decisions and the security enforcement mechanisms are an integral part of the TCB[15]. This methodology can increase the complexity of TCB and its footprint as designers tend to pump in more and more logic there thus creating issues for maintainability. From design paradigm, MILS architecture stands better compared to legacy one as it forces designers to follow a structured and modular approach.

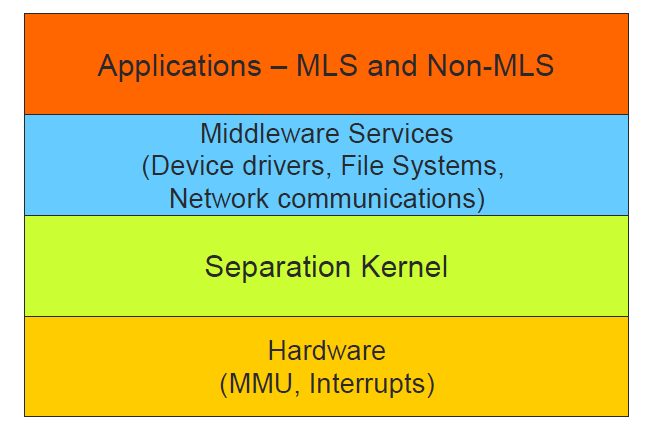
Multiple Independent Levels of Security (MILS) is a security architecture based on the concepts of separation and controlled information flow implemented by separation mechanisms that support both untrusted and trusted code.

This approach provides applications with mechanisms to control, manage and force their security policies in a manner that enforcement mechanisms are always invoked, mechanisms are non-by-passable, evaluable, and tamperproof [15]. Under MILS architecture, privileged mode processing is isolated from under privileged user data or applications. MILS architecture is targeted towards mission critical operations where failure cannot be even thought of.

Main ideas behind designing MILS systems.

* Follow a proactive approach while designing secure solutions rather than being reactive to prevent and employee in damage control exercises.
* Reduce the footprint of secure critical code dramatically , the lesser is better.
* Increase and isolate the critical components and operations in the system.

**Conceptual VIEW OF MILS layers**



(https://www.acsac.org/2005/case/thu-130-taylor.pdf)

#### Separation kernel (SK)

Separation kernels are usually small code bases (4k) that execute close to hardware with primary objective to isolate execution environments for all partitions. They act as base layer interacting with hardware, effectively enforcing complete data separation and information flow control in a SoC by providing time and storage partition between secure and unsecure operations.

Four main objectives for secure kernel are data isolation, highly controlled information flow, data sanitization, damage limitation.

Kernel conceals and denies access to application data and memory contents between partitions. State of executions in current partition will not effect state of executions in other partitions and vice versa. In practical scenario, it might not be desirable to achieve total isolation between partitions as minimal communication across them might be needed. For such situations, secure kernel defines some secure authorized channels for inter-partition communications through which where data sharing can happen. Kernel takes responsibility for relinquishing shared resources and data clean up (buffers, processor register, memory allocations) once they expire on longevity. In the event of security breach or an errant behaviour Secure kernel (Sk) limits the damage limitation in the event of a security breach or an errant behaviour from any application by separating the address spaces between partitions. Secure kernel also cater to all partitions by enforcing bounds on shared resources and guaranteeing minimum processing times , interrupt servicing times and access to resources.

#### Device Drivers

MILS architecture stipulates kernel to be small, having minimum footprint in order to be fully evaluated. In any typical microkernel architecture, other needed OS services are typically included in operating system running as either as separate threads or processes but not in kernel space like any other monolithic kernel. But in MILS architecture , these services are confined to play in address spaces of individual partitions or relegated to separate shared partitions if they service shared resources like communication across partitions but are mediated by separation kernel policy. When the device is categorized as private and un-sharable , access to device is restricted to other partitions other than to partition where the device is configured to access. Their general access via MMUs needs to be curtailed down (memory mapped I/Os to communicate with devices) for private configured devices.

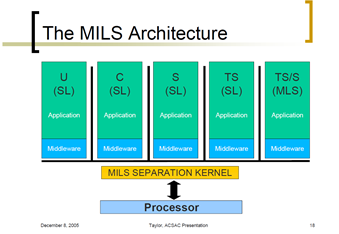
#### Hardware support

Secure kernel needs support of underlying hardware for effectively executing partitioning , control information flow and isolate system resources. SKPP (secure kernel protection profile) mandates certain functional requirements from hardware which secure kernel shall operate on.

Support from hardware MMU (Memoy Management Unit ) is needed for seperation kernel to isolate memory address spaces from various partitions. Processors should provide ability for seperation kernel need to grant configuration access for memory layouts partitions in the system. Processor SDK needs to provide privellaged instruction set that can be only executed by seperation kernel along with mechanism to transfer execution control to seperation kernel in the event of execution of any high privillaged operation or invalid instruction from any partitions. Underlying hardware need to provide automicity support for critical operations like partition swapings and memory layout changes made by seperation kernel. Processor should also provide seperation kernel with options to restrict or configure access of i/o perpherials to specific partitions.

#### Middleware services

Middleware layers sits on top of seperation kernel providing services to applications and services in a perticular partition. Some of the exmples are inter-core communications, event and diagnostic logging, resource sharing and allocation. In addition to above , middleware layer is also responsible to end to end security and communication policy by labeling , filtering and controlling message/information flow. Middleware services can also be designed for authorised communication channels between specfic partitiions dictated by usecase requirements.These services should address solutions from end to end across multiple partions and cores rather than confinment to single process , partition or core.



(https://www.acsac.org/2005/case/thu-130-taylor.pdf)

## Trusted execution environment (TEE)

### TPM

### Secure trusted boot

### Static root of trust

### Dynamic root of trust

## Secure boot

## Role of Operating systems

## Application level security

#### Cryptography

#### Network 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. <http://www.blackhat.com/presentations/bh-asia-04/bh-jp-04-pdfs/bh-jp-04-grand/bh-JP-04-grand.pdf>
9. <https://cwe.mitre.org/about/sources.html>
10. <https://cwe.mitre.org/documents/sources/PLOVER.pdf>
11. <https://cwe.mitre.org/about/faq.html#A.2>
12. <http://maurice.aarts.info/papers/tamper_resistance_evidence.pdf>
13. <https://www.blackhat.com/presentations/bh-usa-04/bh-us-04-grand/grand_embedded_security_US04.pdf>
14. <https://pdfs.semanticscholar.org/0e00/5dc438c192621dc1dbcbf44e17ef7a159e1f.pdf>
15. <https://www.acsac.org/2005/case/thu-130-taylor.pdf>
16. <http://www.omg.org/news/meetings/workshops/RT_2005/02-2_Uchenick>
17. <http://security.hsr.ch/mse/projects/2011_Root_of_Trust_for_Measurement.pdf>

# 

# 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.