# The use of vulnerability data for risk assessment

JENNY MARTINSSON

MASTER´S THESIS

DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

# The use of vulnerability data for risk assessment

Jenny Martinsson
dic15jma@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisors:
Martin Hell
Felix Kruuse, Debricked
Henrik Tehler

Examiner: Thomas Johansson

June 11, 2021

Printed in Sweden
Tryckeriet i E-huset, Lund

# Abstract

Finding vulnerabilities in open source software is an important part of software security. Software security is in turn a vital part in risk management and making risk assessments. The purpose of this thesis is to help organisations make decisions about vulnerabilities they have in their software programs by helping them make their own risk assessment. Our research uses the Common Vulnerability Scoring System (CVSS), the Common Weakness Enumeration (CWE) and ISO-controls. Our research focused on the environmental score part of the CVSS score, and ways to derive the security requirement values were suggested. In the next step the modified base metrics were looked into and it was shown how easily they can be changed depending on what system they are used on. This was shown by comparing the CVSS score given by National Vulnerability Database (NVD) with the CVSS score given by the organisation Red Hat. The last part was to put the vulnerabilities in a larger risk perspective, where a connection was made between the vulnerabilities and the ISO-controls with the use of the CWEs connected to each vulnerability. Our conclusion shows that it is important to look at vulnerabilities from a larger risk perspective, and that our method can facilitate making continuous risk assessments in cybersecurity since a lot of the data can be reused in the future.

# Acknowledgements

First I want to give a huge thanks to my supervisors Martin Hell and Henrik Tehler who gave me continuous feedback throughout the entire process. They were a great sounding board to have and gave great ideas and advice when the thesis took a different turn than what was first planned.

I also want to sincerely thank Felix Kruuse at Debricked who always made time to talk when I had any questions. His insight about clients' needs and his technical knowledge was vital for making this thesis have a real usability.

Lastly I want to thank my family, especially my sister Linda and my partner Emil who was always there to give advice and encouragement when I got stuck. I will also give a special thanks to my cat Chilli, just for being a great companion during this process.

Thank you.

# Glossary

### CVE - Common Vulnerabilities and Exposures

A CVE is a way to identify a specific vulnerability by giving each vulnerability found a unique CVE ID.

### CVSS - Common Vulnerability Scoring System

A system which gives each CVE a score that measures a generic severity of the vulnerability.

### CWE - Common Weakness Enumeration

A CWE is a way to identify or describe a weakness by giving them a unique CWE ID.

### DoS - Denial of Service

Denial of Service is a specific cyber-attack which aims to make a resource unavailable to its intended users either temporarily or indefinitely.

### NPM - Node Package Manager

Can be used to help keep track of dependencies in a particular project.

### NVD - National Vulnerability Database

A database containing a list of all CVEs and their corresponding CVSS score as well as other useful information.

### Triage

Triage is a process to help decide the order of treatment.

# Table of Contents

# List of Figures

# List of Tables

Chapter 1

# Introduction

## 1.1 Background

In today's software development a lot of open source software is used to make it easier for developers in the sense that they do not have to reinvent the wheel. This saves time and money for the organisations using it. Open source code can however have issues with it in the form of software vulnerabilities. This might be because it is an older version and new security measures have been added afterwards, or it can be because it is impossible to write perfectly secure code and vulnerabilities will always be found eventually. This results in a lot of resources being spent on making security checks and updating to handle the software vulnerabilities.

To help with this there are organisations which have started to collect a list of all vulnerabilities found in software being used, so that it is publicly available what vulnerabilities exist. Two big organisations are MITRE and NVD (National Vulnerability Database) which have a close relationship with each other [1]. MITRE collects a list of all known vulnerabilities in the CVE (Common Vulnerabilities and Exposure) list. The list of CVEs is then used by NVD to calculate a CVSS (Common Vulnerability Scoring System) score which aims to give an overview of the severity of a vulnerability. This is used to help organisations make decisions about what to do about the vulnerabilities. A problem with the CVSS score however is that it is often used as a risk assessment [2] [3] [4], even though this is not the intended use as CVSS measures severity, not risk. This is something that even the official specification document clarified in their newest version, writing:

> The CVSS Specification Document has been updated to emphasize and clarify the fact that CVSS is designed to measure the severity of a vulnerability and should not be used alone to assess risk [5].

Risk management is another important objective in organisations. Risk assessment consists of risk identification, risk analysis and risk evaluation [6]. This is something that is similar to the CVEs and their CVSS scores. Vulnerabilities are identified and then analyzed in to give them a CVSS score as a start of an evaluation. This is however not a complete risk assessment, since many key parts of the risk management process are overlooked. The first step in a risk management process is to decide the goal or objective for the risk analysis. The next step is to figure out the scope and context. This is something that is not clarified in the CVSS base score, but the environmental score helps a bit with explaining the context.

### Debricked

This master thesis is written at the company Debricked. Debricked is a company whose product is used to handle vulnerabilities in dependencies, open source code and other libraries. Their clients upload their code repositories, and Debricked gives a list of all vulnerabilities found in the form of CVEs. To do this they scan various sources for information about vulnerabilities, licenses and health data. These include the National Vulnerability Database (NVD), NPM (Node Package Manager), security advisories and looking for their own vulnerabilities.

Debricked also gives a score of how severe each vulnerability is using the CVSS score. They are also working on their own scoring system, debAI, since the CVSS score is lacking. They also provide some solutions to the vulnerabilities by checking for: false positives, if the attack vector is relevant, if the vulnerable code is being used, and if there is any update available. If there is an update available it is important to check if there are any breaking changes. At this stage there are essentially five different alternatives: update, patch, figure out a workaround, pause, or triage. At the moment the client has to make their own evaluation at what decision they will take, by making their own risk assessment.

In this master thesis we want to further help organisations make decisions about vulnerabilities, by helping them make a risk assessment.

## 1.2 Purpose and Project Aims

Risk management and assessment is crucial in all organisations, not the least in cybersecurity. Making a risk assessment takes a lot of work and time, and can become very costly for an organisation. This might be a reason why so many people think of the CVSS score as a risk assessment even when it is not. Because of this we want to investigate if there is a way to help make risk assessments by originating from the CVSS score. We also want to help people understand the difference between risk and severity, and that the CVSS score can only be used to measure the severity of a vulnerability, and not as a risk assessment.

The outcome of this master thesis will contribute to how we can use the CVSS score and other information security properties to make both generic risk assessments, and also how it can be used to make specific risk assessments for individual clients.

### Purpose

The purpose of this thesis is to help organisations make decisions about vulnerabilities they have in their software programs by helping them make their own risk assessment.

### Project Aims

In this master thesis, we have two main project aims. Firstly we want to build an easy-to-use template for determining the environmental score in the CVSS, which is dependent on the environment and thus not given by NVD. Secondly we aim to investigate how to use the environmental score to build parts of a risk framework based on the ISO 27000 standards.

To do this we want clients to give us specific information which we will then use, together with the CVEs we get from a vulnerability tool and information about ISO standards and CWEs, to give clients an output they can use to help with their risk assessment. We will also look deeper into the CVSS score, and show how an environmental analysis can affect the outcome.

The main questions in this thesis are:

- How can we help clients calculate the environmental CVSS score for their vulnerabilities?
- How can we help clients assess each vulnerability?
- How can we help clients make their own risk assessment based on their environmental CVSS score and the ISO standards?

## 1.3 Approach

Our system model is shown in Figure 1.1. Here the input and output is defined, as well as showing which steps will be needed to be taken.

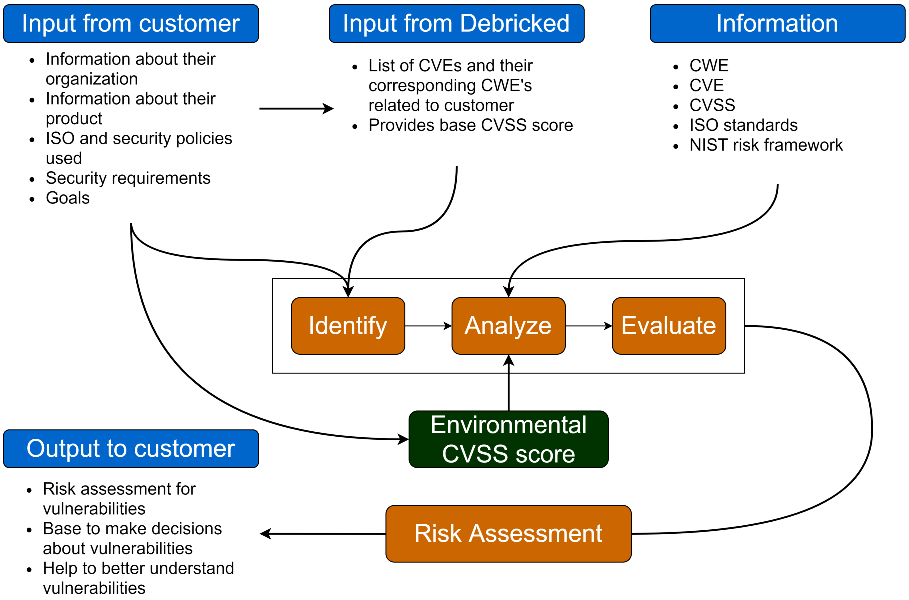
Input from customer
• Information about their organization
• Information about their product
• ISO and security policies used
• Security requirements
• Goals
Input from Debricked
• List of CVEs and their corresponding CWE's related to customer
• Provides base CVSS score
Information
• CWE
• CVE
• CVSS
• ISO standards
• NIST risk framework
Identify
Analyze
Evaluate
Environmental CVSS score
Output to customer
• Risk assessment for vulnerabilities
• Base to make decisions about vulnerabilities
• Help to better understand vulnerabilities
Risk Assessment

**Figure 1.1:** System model

As shown in Figure 1.1, we see what input is used in the different parts of a risk assessment, which consists of identifying, analyzing and evaluating risk. We will use input from the client and Debricked to *identify* what vulnerabilities exist for the client. The CVSS base score provides a first rough approximation about how severe the vulnerabilities can be. However, to find out the severity for our client, we first need to examine both the client's goals and policies to determine the scope and context that the risk assessment will cover. At this point we will make use of the CVSS environmental score process to help the clients put the vulnerabilities in their own context. For the *analysis* we will also use different frameworks and standards to help determine the severity, in both the client's context, and in an overall risk assessment. We especially focus on the controls in ISO 27002 and make connections between them and top CWEs, to help make a connection between ISO controls and vulnerabilities in software. Lastly we *evaluate* our results which can help clients make decisions about vulnerabilities as they now know how severe the vulnerabilities can be for them. Our goal is also to help people better understand vulnerabilities and what they mean in the context of the client. This can then be used by the client to prioritize the vulnerabilities and start a risk treatment.

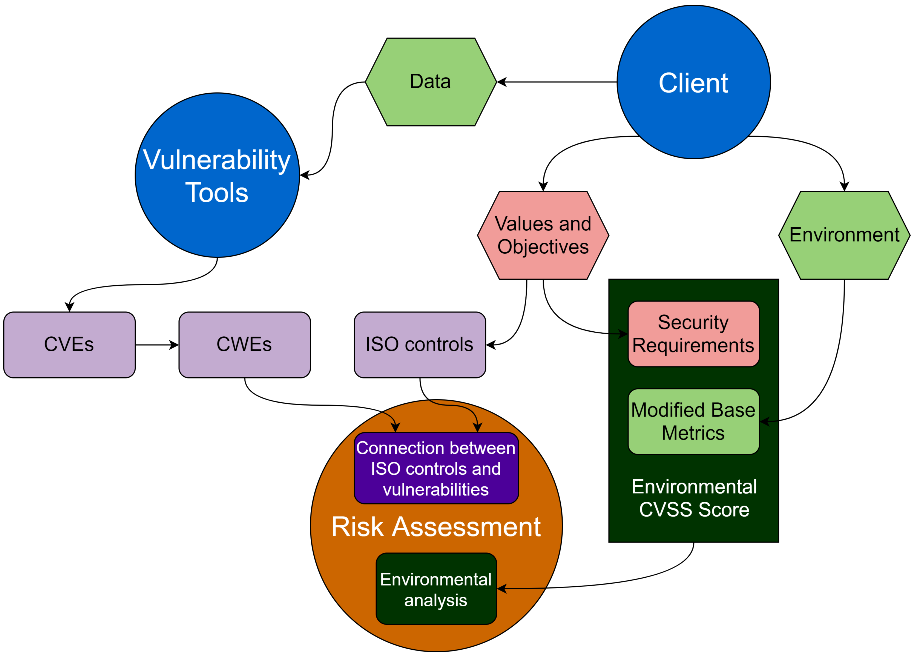

**Figure 1.2:** Overview

Figure 1.2 presents an overview of what this report consists of and how it fits together. The client gives their code to a software vulnerability tool, e.g. Debricked, who analyses it for software vulnerabilities and presents them in the form of identifiers, e.g. CVEs. Each CVE has one or more CWEs connected to them, which is given by NVD. The client also needs to provide information about their values and objectives to us, in a form of security requirements in their environmental analysis. Here they also provide us with any modified base metrics based on their environment. They also provide a more detailed overview of their goals and values in form of what ISO controls they use. The list of CWEs connected to their vulnerabilities, the ISO controls and the environmental score is then what we use to help them make a risk assessment for each vulnerability.

### 1.3.1 Outline

#### Chapter 1: Introduction

This chapter introduces the problem to the reader. It specifies the goals and objectives of this thesis, and what approach that was used. It also gives an outline of the report and mentions related work.

#### Chapter 2: Theoretical Background

The theoretical background explains the background to vulnerabilities in the form of CVEs, and also about the organisations behind it. It continues explaining the CVSS score, what it is used for and how to use it. It also gives definitions for some of the terms in this master thesis, explaining the different definitions that exist and which one is relevant in this thesis. Lastly it gives background to the ISO guidelines.

#### Chapter 3: Method

This chapter goes through the method used for the analysis. It explains why the approach was chosen and what steps were made to get the results. It also contains the steps made in each process.

#### Chapter 4: Environmental Analysis

In this chapter the results for the environmental analysis guidance are presented. It presents questions to help calculate the security requirements, and explains the difficulty and importance of modified base metrics.

#### Chapter 5: Risk Evaluation

This chapter continues the results with a focus on CWEs and ISO standards and how they can be used to help guide clients in making a risk assessment.

#### Chapter 6: Discussion

This chapter discusses the results and proposes ideas for where future work could focus.

#### Chapter 7: Conclusion

The last chapter of the report summarizes the findings and answers the main questions of the master thesis.

## 1.4 Related Work

There has been a lot of work relating to the CVSS score and its limitations.

Hamid and MacDermott [7] presented a modified version of the CVSS score which they named DVSS, dynamic vulnerability scoring system. In this they developed a dynamic database that took into account interaction between vulnerabilities. The score took into consideration intrinsic (or essential), time-based and ecological metrics. They also laid a focus on the attackers' needed privileges (none, user and root). This is related to our thesis, since it looks into how to add to the CVSS score. However, Hamid and MacDermott have more of a focus on how vulnerabilities interact with each other, and how that can change the vulnerabilities' severity when combined. This is something very interesting, but not covered in this thesis. It is definitely something that would be interesting to look further into, especially with connecting it to ISO controls when combined.

Doynikova and Kotenko [8] suggested several techniques for risk assessment based on the CVSS score and attack modeling. The techniques all used input data from security information and event management (SIEM) systems. They had both static and dynamic techniques. They performed a set of experiments with these techniques on different networks and attack sequences and the advantages and disadvantages of these techniques. Doynikova and Kotenko have a similar goal as us, but a different execution. They also want to use the CVSS score for risk assessments, but focus on attack modelling instead of using the environmental score and connecting vulnerabilities to ISO controls.

Houmb et al [9] presents a CVSS Risk Level Estimation Model, which computes the overall risk level of a system, based on frequency and impact estimates which were derived from a rearranged version of the CVSS attributes. The model is implemented as a Bayesian Belief Network (BBN) which has its limitations since the construction of the BBN has a big part in the outcome of the risk analysis, and it is therefore important that it is constructed correctly. This model also has the same goal as us, but with a very different focus. They use all parts of the CVSS score, both the temporal metrics and environmental metrics, to create a BBN. We instead keep the CVSS score as our severity rating and put it in a bigger context, i.e. how it connects to ISO controls.

Allodi et al [10] discusses how the base metrics of the CVSS score can vary depending on who did the calculations. In Allodi et al's [10] paper they perform an experiment using computer science students to calculate a CVSS score for specific CVEs in a specific environment. They conclude that "Our experimental results indicate that contextual security assessments do not scale well with complexity of the environment even when limited to simple tasks involving the identification of 'segmented' areas of a network. This may lead to high error rates in the assessment, ultimately resulting in decreased overall security and compliance to regulation". Allodi et al's work is related to our work in Section 4.2 on the modified base metrics. They look at the difference between several individual assessors' scores, whereas we look at the scores given by two big organisations, NVD and Red Hat.

Gallon [11] instead discussed the impact the security requirements have on the CVSS score. However, since the paper was written in 2010, they use an earlier version for the CVSS score, and the environmental score is therefore calculated differently than now. Gallon [11] concluded that the distribution of the CVSS score differed a lot when modifying the integrity requirement, in comparison with the confidentiality and availability requirements. They also concluded that the Adjusted Impact formula, which is used to calculate the new CVSS score with the environmental score taken into account, did not match the distribution of IT vulnerabilities thus leading to the opposite effect one would want from the environmental vector. They did however only use the security requirements part of the environmental score. Gallon's research has a different focus than ours, but has a very interesting perspective which shows that the security requirements have different impact on the severity. This is related to our work in Section 4.1.

Li, Xi and Zhao [12] discuss the distribution of the CVSS environmental score version 2, to see what final scores they could get for the CVSS score. The CVSS v2 environmental score was divided into three parts, Collateral Damage Potential, Target Distribution and Security Requirements. They concluded that "Based on these studies, network administrators can have a high-level view with the impact of environmental metrics, thus determine the priority of processing different vulnerabilities and take appropriate mitigation measures to ensure network security." Their work is similar to Gallon's work, and is also related to our section about security requirements.

Chapter 2

# Theoretical Background

## 2.1 Vulnerability Databases and Organizations

There are several different databases when it comes to software vulnerabilities, the biggest ones being National Vulnerability Database (NVD) and the one provided by MITRE Corporation.

MITRE is a non profit organization which works together with governments and industries to, for one thing, improve and adapt solutions against cybersecurity threats through awareness, resiliency and adopting new concepts [13]. As a part of this they provide us with a list of common software and hardware weakness types, called CWE (Common Weakness Enumeration) [14]. In connection to this they also host CVE Records (Common Vulnerabilities and Exposures). These are vulnerabilities registered by different CNAs (CVE Numbering Authorities) which consist of organisations and researchers from all over the world. These CNAs provide us with the vulnerabilities they are aware of and give them a CVE ID and description of the vulnerability or exposure, so that it is possible to track it [15]. The CVE is made to give a standardized identifier and provide a reference point for data exchange in the cybersecurity community, to help discover these potential gaps in the security coverage [16].

The National Institute of Standards and Technology (NIST) is an organisation that is part of the U.S. Department of Commerce. The mission of NIST is to promote innovation and industrial competitiveness [17].

The National Vulnerability Database (NVD) is a product of the NIST Computer Security Division [18]. NVD is in close contact with MITRE and the CVE Program to make a collection of all CVEs. NVD is built upon the CVE List and takes all the information from the CVE Records and adds to that information, providing enhanced information for each CVE [1]. The NVD analysis results in a vulnerability score using CVSS, vulnerability types using CWE as well as other important metadata. The NVD does not actively perform vulnerability testing and instead rely on vendors, third party security researchers and vulnerability coordinators to provide the information that is needed for the analysis [18].

FIRST stands for Forum of Incident Response and Security Teams and is responsible for the Common Vulnerability Scoring System and its improvement, among other things [19]. They provide the specification document for the different versions of the CVSS, both for the current version 3.1, and all the archived documents for the older versions. They also provide a user guide and examples to help organisations calculate their CVSS score [5].

## 2.2 CVEs and CWEs

Common Vulnerabilities and Exposures (CVE) and Common Weakness Enumeration (CWE) are two concepts used to explain vulnerabilities in software. The CWE is an overall view of the different types of vulnerabilities there are, whereas the CVE lists all the specific vulnerabilities found so far, where each of them have their own CVE number, and each is in a specific CWE category.

### 2.2.1 CVE

The goal behind the CVE program is to 'identify, define, and catalog publicly disclosed cybersecurity vulnerabilities'. Each vulnerability gets its own CVE Record [16]. The CVE Record contains descriptive data about the vulnerability with a specific ID. They also contain which CNA they were provided from, and other information that needs to be provided according to CVE Numbering Authority (CNA) Rules [20].

NVD gives each CVE a CVSS score, which gives information about the vulnerability and how it can affect you. It also provides the connection to the CWE, listing the connected CWE-IDs for each CVE, often provided by NIST. It also provides a list of 'known affected software configurations', and referenced to 'advisories, solutions and tools' which can provide updates done in patches which solves the vulnerability, issue tracking and other information that can be useful.

### 2.2.2 CWE

The Common Weakness Enumeration (CWE) is a community-developed list of software and hardware weakness types. It is there to help developers and security practitioners to describe and discuss weaknesses in a common language [14]. It is also there to help check for weaknesses in existing products, evaluate the coverage of tools targeting these weaknesses, set a baseline standard for weakness identification, mitigation and prevention efforts and lastly to prevent vulnerabilities prior to deployment [14]. The CWE list also has different ways to navigate the hierarchy, by focusing on a specific CWE VIEW, e.g. software development (View ID 699). This view then supplies the different categories that exist, and the CWEs listed under that category [21].

Each CWE entity typically contains an ID, a name explaining the weakness, a description of relationships to other CWEs, applicable platforms the weakness can appear, common consequences, likelihood of exploit and examples that demonstrate the weakness to name some of the things it contains.

## 2.3 CVSS Score

CVSS, Common Vulnerability Scoring System, is an open framework used to communicate the characteristics and the severity of vulnerabilities in software [22]. They use three different metric groups: Base, Temporal and Environmental. The base metric is divided into exploitability metrics and impact metrics. The exploitability metrics are divided into attack vector, attack complexity, privileges required and user interaction. The impact metrics are confidentiality, integrity and availability. There is also the 'scope' metric. These metrics are then used to give an overall CVSS score. The Base metric is what is most often referred to when talking about the CVSS score, as it gives a basic overview of the exploitability and impact. When combined with the temporal and environmental score however, it gives a more accurate score of the exploits and impact to the specific environment. The temporal score is divided into Exploit Code Maturity, Remediation Level and Report Confidence which go into more of the specifics around the CVE, where it shows likelihood of the vulnerability being attacked, if there are any solutions and how good they are, and also the background information around the vulnerability and how accurate it may be. Lastly there is the Environmental Score, which is the focus of this master thesis. The Environmental Score consists of three security requirements, and modified base metrics. The security requirements are divided into Confidentiality, Availability and Integrity. This is something that is connected to the company's values and what is important to protect. The modified base metrics are derived just as with the original base metrics, but adapted to the system requirements of the company's project [22].

### 2.3.1 Base Metrics

Below, the different base metrics are listed, together with the different values they can take [22]. If the modified base metric is the same as the base metric, the score is not affected.

#### Exploitability Metrics

**Attack Vector:** How the vulnerability can be exploited. The score increases the more remote an attacker can be.

- **Network:** Remotely exploitable. The vulnerable component is bound to the network stack, and can be accessed from the entire Internet.
- **Adjacent:** The attack needs to be launched from the same shared physical network to exploit the vulnerability.
- **Local:** The attack can only be exploited locally (e.g. via keyboard), remotely via, e.g. SSH or via User Interaction.
- **Physical:** The attack requires the attacker to physically touch or manipulate the vulnerable component.

**Attack Complexity:** How difficult it is to exploit the vulnerability. If there are any conditions beyond the attackers control. The score increases the less complex the attack is.

- **Low:** Specialized conditions do not exist. An attacker can expect repeatable success.
- **High:** A successful attack depends on conditions beyond the attacker's control. Requires the attacker to invest in some measurable amount of effort in preparation or execution against the vulnerable component before a successful attack can be expected.

**Privileges Required:** What level of privileges is needed for an attacker to successfully exploit the vulnerability. The score increases the less privileges are required.

- **None:** The attacker is unauthorized prior to attack, and therefore does not require any access to settings or files to carry out an attack
- **Low:** The attacker requires privileges that provide basic user capabilities that could normally affect only settings and files owned by a user or cause an impact only to non-sensitive resources.
- **High:** The attacker requires privileges that provide significant control over the vulnerable component, (e.g. administrative).

**User Interaction:** If there is a need for any user interaction, besides the attacker, for the attack to succeed. The score increases if no interaction is needed.

- **None:** The vulnerable system can be exploited without any interaction from any user.
- **Required:** Successful exploitation of this vulnerability requires a user to take some action before the vulnerability can be exploited.

### Scope

**Scope:** If a successful attack impacts a component other than the vulnerable component. The score increases if the scope is changed.

- **Unchanged:** An exploited vulnerability can only affect resources managed by the same security authority.
- **Changed:** An exploited vulnerability can affect resources beyond the security scope managed by the security authority of the vulnerable component.

### Impact Metrics

**Confidentiality:** Measures the impact to the confidentiality of the information resources managed by the vulnerable component. Confidentiality refers to limiting information access and disclosure to only authorized users, as well as preventing access by, or disclosure to, unauthorized ones. The score increases the higher the loss.

- **None:** There is no loss of confidentiality within the impacted component.

- **Low:** There is some loss of confidentiality. Access to some restricted information is obtained, but the attacker does not have control over what information is obtained, or the amount or kind of loss is limited.
- **High:** There is total loss of confidentiality, resulting in all resources within the impacted component being divulged to the attacker. Alternatively, access to only some restricted information is obtained, but the disclosed information presents a direct, serious impact.

**Integrity:** Measures the impact to integrity of the vulnerable component. Integrity refers to the trustworthiness and veracity of information, i.e. how can the attacker modify the data. The score increases the higher the loss.

- **None:** There is no loss of integrity within the impacted component.
- **Low:** Modification of data is possible, but the attacker does not have control over the consequence of a modification, or the amount of modification is limited.
- **High:** There is a total loss of integrity, or a complete loss of protection. For example, the attacker is able to modify any/all files protected by the impacted component. Alternatively, only some files can be modified, but malicious modification would present a direct, serious consequence to the impacted component.

**Availability:** Measures the impact to the availability of the vulnerable component. Availability refers to the accessibility of information resources, so attacks that consume network bandwidth, processor cycles, or disk space all impact the availability of an impacted component. The score increases the higher the impact.

- **None:** There is no impact to availability within the impacted component.
- **Low:** Performance is reduced or there are interruptions in resource availability. The resources in the impacted component are either partially available all of the time, or fully available only some of the time.
- **High:** There is total loss of availability, resulting in the attacker being able to fully deny access to resources in the impacted component.

### 2.3.2 CVSS Vector String

The Base Metrics, as described above, can be summarized in a vector string for an easier overview. The vector string is a compressed representation of the values used to derive the score [22]. The vector string begins with the CVSS version, and continues with the different metrics in the same order as seen in Table 2.1 with a '/' between each metric and a ':' between the metric and the value. An example of a vector string can be

CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:L/I:L/A:N [22].

**Table 2.1:** List of all base metrics and their abbreviated forms, together with the possible values they can take

| **Base Metric** | **Possible Values** |
|---|---|
| Attack Vector (AV) | Network (N), Adjacent (A), Local (L), Physical (P) |
| Attack Complexity (AC) | Low (L), High (H) |
| Privileges Required (PR) | None (N), Low (L), High (H) |
| User Interaction (UI) | None (N), Required (R) |
| Scope (S) | Unchanged (U), Changed (C) |
| Confidentiality (C) | None (N), Low (L), High (H) |
| Integrity (I) | None (N), Low (L), High (H) |
| Availability (A) | None (N), Low (L), High (H) |

## 2.4 Environmental Analysis

The environmental analysis is something that can be used to give a more accurate view of the vulnerabilities' severity on your own system [22]. The environmental score consists of two main parts, the security requirements and the modified base metrics. The security requirements focus on the companies' goals and values, and let you express them in wide terms for the three categories confidentiality, integrity and availability. This is the first step to making the CVSS score more personalised and accurate for the vulnerabilities in your own environment [22]. The second step is the modified base metrics. These are there to physically determine if the vulnerability affects your system in the same way as the generic system. This takes into account the network setup, privileges, extra security in your setup etc. It also looks into the impact, and takes into account if there even is a possibility to affect the confidentiality, integrity or availability in your system [22].

### 2.4.1 Security Requirements

The security requirements exist for you to express what values you have as a company, and what you feel is important to protect. It also handles how sensitive the data which the company is storing is. The security requirements are the Confidentiality Requirement, Integrity Requirement and Availability Requirement. The security requirements have no effect if the corresponding impact metric in the base metrics is set to 'None' [22].

**Confidentiality:** The Confidentiality Requirement is mainly based on the classification level of the data that is stored or used on the target system. It also takes into account the encryption level of the data at rest. Only data that is being consumed or processed should be taken into consideration when assessing the Confidentiality attribute [5].

**Integrity:** The Integrity Requirements for the system focus on the importance of the accuracy of the data it stores or uses [5].

**Availability:** The Availability Requirement is based on the up-time requirements and redundancy of the device or the applications hosted by the device [5].

#### Values

Each security requirement has four different possible options, 'Not Defined', 'Low', 'Medium' and 'High'.

**Low:** The 'Low' option lowers the severity of the impact from the vulnerability, and should be picked in a situation where the loss of the specific security requirement is likely to only have a limited adverse effect [22].

**Medium:** The 'Medium' option does not impact the base score, and is the default value. It should be picked in a situation where the loss of the specific security requirement is likely to have a serious adverse effect [22].

**High:** The 'High' option increases the severity, and should be picked in a situation where the loss of the specific security requirement is likely to have a catastrophic adverse effect [22].

**Not Defined:** The 'Not Defined' option gives the same result as the 'Medium' option, but also tells us there is insufficient information to make a decision [22].

### 2.4.2 Modified Base Metrics

The modified base metrics are derived the same ways as the base metric, see Section 2.3.1, except it is checked against a specific system. This can mean that what was once on an 'Attack Vector: Network', can be changed to 'Modified Attack Vector: Adjacent' if the vulnerable component is deployed on a secure network, unavailable from the rest of the internet, for example with the help of a firewall [22].

## 2.5 Relevant Definitions

Risk is a word that is used in a lot of different ways and can have several different meanings depending on the context. Risk is often used when talking about severity, consequences or probability of something happening. However, risk is so much more than that. Below are some examples of different organisations definitions of risk, and other related terms.

### 2.5.1 Risk

#### ISO 31000

In ISO 31000 they define **risk** as "Effect of uncertainty on objectives". They also clarify that an **effect** is "a deviation from the expected" and that **risk** is often expressed in the terms of "risk sources, potential events, their consequences and their likelihood" [6].

They have also defined **risk management** as "coordinated activities to direct and control an organization with regard to *risk*".

### SRA

The Society for Risk Analysis (SRA) glossary contains several definitions of risk. They start with 7 qualitative definitions of risk, and one interpreting the ISO definition of risk [23].

1. Risk is the possibility of an unfortunate occurrence
2. Risk is the potential for realization of unwanted, negative consequences of an event
3. Risk is exposure to a proposition (e.g., the occurrence of a loss) of which one is uncertain
4. Risk is the consequences of the activity and associated uncertainties
5. Risk is uncertainty about and severity of the consequences of an activity with respect to something that humans value
6. Risk is the occurrences of some specified consequences of the activity and associated uncertainties
7. Risk is the deviation from a reference value and associated uncertainties

SRA also has definitions for risk analysis, risk assessment and risk evaluation, which are all part of this master thesis. However, they do not define risk identification and they do not include it in their risk analysis, instead starting with risk assessment and moving on to risk characterization, risk communication and risk management.

### NIST

NIST also has several definitions of **risk**, one being [24]:

> A measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence.

### Our definition

In this master thesis we use a combination of different definitions of risk, especially ISO 31000 and NIST since they both are very similar in their definitions but expressed in different ways. Our main definition is: “A measure of the extent to which an entity is threatened by a potential circumstance or event.”

## 2.5.2 Vulnerability

When talking about vulnerabilities in this master thesis, we talk about vulnerabilities in software.

### FIRST

The organisation FIRST define **vulnerability** as [5]:

> A weakness or flaw in the functional behavior of a vulnerable computational component (software or hardware) that can be exploited, resulting in a negative impact to the Confidentiality, Integrity, and/or Availability of an impacted component.

FIRST also define **vulnerable** as [5]:

> A component is *vulnerable* if it contains a weakness or flaw that can be exploited, given the necessary conditions and/or exposure.

### SRA

SRA has also defined **vulnerability**, in three ways [23]:

- The degree to which a system is affected by a risk source or agent
- The degree to which a system is able to withstand specific loads
- Vulnerability is risk conditional on the occurrence of a risk source/agent. [...] vulnerability is uncertainty about and severity of the consequences, given the occurrence of a risk source

### MITRE

MITRE has in connection to CWE defined **vulnerability** as [25]:

> an occurrence of a weakness (or multiple weaknesses) within a product, in which the weakness can be used by a party to cause the product to modify or access unintended data, interrupt proper execution, or perform incorrect actions that were not specifically granted to the party who uses the weakness.

However in connection to the CVEs, MITRE has a definition similar to FIRST's definition of **vulnerability**, it being [20]:

> A flaw in a software, firmware, hardware, or service component resulting from a weakness that can be exploited, causing a negative impact to the confidentiality, integrity, or availability of an impacted component or components.

### Our definition

In this master thesis we use the definition given by FIRST, since the vulnerabilities mentioned in this report are all connected to vulnerabilities in code. Their definition is also what we will use when talking about CVEs, since a CVE is just an identifier referencing a specific vulnerability. This definition of a vulnerability is closer to the ISO 31000 definition of a **risk source**, being "element which alone or in combination has the potential to give rise to risk".

### 2.5.3 Weakness

#### FIRST

The organisation FIRST define **weakness** as [5]:

> An error in software or hardware implementation, code, design, or architecture that, depending on exposure, could be exploited by an attacker.

#### MITRE

MITRE instead defines **weakness** as [25]:

> a type of mistake that, in proper conditions, could contribute to the introduction of vulnerabilities within that product. This term applies to mistakes regardless of whether they occur in implementation, design, or other phases of a product lifecycle.

### 2.5.4 Severity

Severity is a very important word in this master thesis. It is also something often mixed up with risk, as seen in the case of the CVSS base score. However, the only of the previously mentioned sources giving a definition of severity was the society for risk analysis (SRA). FIRST talks a lot about it in connection to the CVSS score, since it measures severity, but it never gives their definition of it.

#### SRA

SRA defined **severity** as: "The magnitude of the damage, harm, etc." where they define **harm** as "Physical or psychological injury or damage" and **damage** as "Loss of something desirable" [23].

## 2.6 Guidelines and Frameworks

#### NIST

The organisation NIST (National Institute of Standards and Technology) has a risk management framework for information systems and organisations which works as a guide for how to better reduce cybersecurity risks [26]. In this framework they go through the fundamentals when it comes to managing security and privacy risk and also the process of how to execute these risk management framework tasks [26].

#### ISO 27000 series and ISO 31000

The ISO 27000 series consists of a set of guidelines for ISMS - Information security management systems. It is divided into four parts of standard families, vocabulary standard, requirement standard, guideline standard and sector-specific guideline standards. The first gives an overview of the 27000 series and terminology around ISMS. The second standard family consists of standards specifying requirements for general ISMS but also certification, audits and sector specific requirements. The general guidelines consist of several standards for different parts of the ISMS, from general guidelines to guidelines about monitoring, analyzing, measuring and evaluating. Lastly the sector specific guidelines describe guidelines for different types of organisations, from telecommunication to managing personal data in clouds [27].

ISO 27002 is a document in the ISO 27000 series which lists best practice controls that should be used for achieving information security. These controls should be seen as a guidance when selecting and implementing controls for one's own business [27]. The document is divided into several categories, all with their own objectives. The controls are listed under each category with a definition, implementation guidance and general information about the use of the control [28].

ISO 31000 focuses more on the general/common approach to risks and risk management. They focus on uncertainty's effect on goals. It contains the principles of risk management, followed by risk management frameworks for different parts of an organisation, from leadership and commitment, to design and integration. It lastly goes through the risk management process, going through scope and context but also risk assessment and risk treatment [6].

Chapter 3

# Method

This chapter covers the method used for the analysis. It starts with the method used for establishing the security requirement questions, and continues with the method used for comparing the CVSS score. It also brings up the approach used to get the top most common CWEs, and how we got the technical impacts. It follows by explaining the method used for comparing the CWEs over the years. It ends with explaining our approach to the ISO controls and how and why they were chosen. The chapter explains why the method was chosen and what steps were made to get our results, which helps to create reproducibility.

## 3.1 Environmental Analysis

In chapter 4 we present a way to help clients calculate their environmental score for each vulnerability. This is an important part in this master thesis, since the environmental CVSS score is a basis for our risk assessment which will be brought up more in later chapters.

The plan was to create a template for clients where they can calculate their own environmental score. To do this we had to find out what parameters would be needed to calculate an environmental CVSS score and what input would be needed from clients and what input could be taken from other places, e.g. the NVD (National Vulnerability Database).

The CVSS base score can be taken from the NVD database, among other places, which lists all CVEs with their corresponding CVSS score. In some cases there is no CVSS score yet, and the CVE is then listed without it. What is needed for an environmental score is security requirements and modified base metrics.

### Security Requirements

The security requirements are completely dependent on the values and requirements of the client's business. To find out what to set your security requirements to, one can follow the guide on first.org [5]. What we did was to use this as a base, and formulate questions that are easier for the client to answer. The questions are all based on one or more of the examples on first.org. Our next step was to find a way to easily answer these questions, which was done by formulating flowcharts and asking the most important questions first. The most important questions were in this case the questions that would result in a specific outcome regardless of what the answer was to the other questions. A flowchart was then created with questions for each security requirement to easier find out which level the security requirement should be at.

Each question was also explained in detail, to clarify the context and if it was even applicable for the client's organisation.

### Comparing CVSS Scores

Determining the modified base metrics CVSS score can be difficult, especially without a specific scenario in mind. We illustrate how the base metrics can differ when different evaluators calculate the CVSS score. In this case we picked NVD and the organisation Red Hat, which is also a CVE Numbering Authority (CNA).

Showing the differences in CVSS scores can help clients with their own modified base metrics, as they can see in which cases they have been changed and apply it on their own system. It also shows how stable the CVSS score is depending on the system, as a big difference can mean it is very system dependent whereas the same score might mean it is not.

When comparing CVSS scores all the vulnerabilities from 2020 listed on NVD were chosen. This list of CVEs were easily downloaded from the NVD web page in a json file format. After that all of these vulnerabilities were checked if they had a CVSS score, and if they did their CVSS score was selected together with their CVE ID and their CVSS metrics.

The same thing was performed for Red Hat by using their REST API and picking the same elements as mentioned above. After that every metric in NVD was compared with its metric in Red Hat, and if any of the metrics differed they were added to a list. This list then got exported to an excel document with the CVE ID, CVSS score and CVSS metrics for both NVD and Red Hat.

Lastly tools in Excel were used to determine which database had given a higher score for each metric, and counting how many of each CVSS metric was higher/lower for Red Hat.

## 3.2 CWE

### Top 50 CWEs

In chapter 5 we want to connect ISO controls to CWEs. To manage the scope we chose to only use the top 50 CWEs of 2020, since they cover most of the CWEs. To get the top 50 CWEs of 2020 the json file mentioned above was used, i.e. the json file listing all CVEs from NVD for 2020. Here instead was the CVE ID selected and the corresponding CWE ID and CWE name for each CVE. Tools in excel were used to count every instance of each CWE to see how many CVEs had a specific CWE. We then removed the CVEs that did not have a CWE (NVD-CWE-noinfo) assigned to it, and the ones without a specified CWE (NVD-CWE-Other). Lastly a list was printed out with the top 50 CWE-IDs and their respective count.

### Technical Impacts

The technical impacts are yet another guide to help understanding CVEs, since the CWE specifies what technical impacts are affected. It also shows the scope of the CWEs, which can be relevant when it comes to the impact metrics in the CVSS score. This helps give a better understanding to what the vulnerability is about, and shows us which technical impacts are most diverse, showing the different number of CWEs connected to each impact. For the technical impacts we went through the complete document of CWE Version 4.3 and looked up the different kinds of technical impacts that were available by checking the impacts for each CWE. After getting a list of all different technical impacts we did a word search for the technical impact with a space before and after to remove incorrect mentions of impacts. After that the number of times they appeared was noted, and we then went through all the occasions where it was mentioned to check that it was not written in other places, i.e. given an incorrect count. This was however done manually, so it is possible that we might have added a few extra mentions of some of the technical impacts.

### CWE comparison through the years

The world of computer science changes a lot over a few years, and therefore we want to know if the outcome of this master thesis will be usable in the upcoming years. To do this we looked at the change in CWEs over the last 3 years, by looking at all CVEs from 2018, 2019 and 2020 and checking how many of the CVEs used a specific CWE. The same system as for the top 50 CWEs was used, but now included the same for both 2019 and 2018. After having a complete list of all the CWEs for 2020, 2019, and 2018 and their respective count each year, we checked which ones were unique for each year, and which ones had been mentioned in the other years but were not in this year's list. The outcome can be seen in appendix A, where only the one with a difference is listed. We also noted which ones were new since previous years, and which ones just skipped a year to reappear again.

## 3.3 ISO Controls

In this section we looked at the different controls in ISO 27002 (Information technology - Security techniques - Code of practice for information security controls). This document contains controls for all parts of information technology, not just the technical parts. Therefore the document was sifted through, and only the controls relevant to this master thesis were picked out. Our selection process started out with removing anything not related to software, as hardware and physical criteria were beyond our scope. We then went through the remaining ISO controls and picked out only the ones that specifically mentioned software controls, and not just general things linked to software. We chose the controls that could clearly be linked to software vulnerabilities. The reason for this is that connecting ISO controls to vulnerabilities is the first step to connecting them to CWEs. Lastly they were compared to CWE definitions to see if there were any kind of connections between them, and if the ISO control was too general it was rejected.

When going through the controls in ISO 27002 we picked out the ones we found most relevant for this master thesis project, i.e. the ones connected to software vulnerabilities. This was done by first looking over the categories, and removing the ones that did not have a direct connection to software, e.g. 'Human resource security', 'Supplier relationships', 'Compliance' and 'Information security incident management'. These categories were also sifted through, to check that there was no direct connection to software. The controls were all read through in these categories and keywords were selected that could be related to software, e.g. 'Access control', 'Malware', 'Network', 'Cryptography', 'Logging', 'System' and 'Development'. This process was then iterated several times, starting with a huge list of controls and removing controls that felt unnecessary or irrelevant. This process was not very scientific or reproducible, and there is a chance that some controls that were left out could have been relevant. This is an obvious error source that should be kept in mind.

The reason for picking out ISO controls is to help make a connection between vulnerabilities and ISO controls. This can then be used to help put vulnerabilities in a bigger risk management perspective, as the ISO controls are made to help fulfill your requirements.

## 3.4 Connecting CWEs to ISO Controls

After choosing the most relevant controls, we looked at connecting them to the top 50 CWEs of 2020. The reason for this is that the top 50 CWEs covered most CWEs in 2020, and this can therefore be used as a reference list, to see if the CWE breaks any of the ISO controls. Here we began going through all CWEs and sifted out all CWEs that were too specific to be connected to an ISO, e.g. a CWE connected to specific languages, or a specific framework. Each CWE was then checked against all the ISO controls listed, and noted if the control applied to that specific CWE.

### 3.4.1 Example

In Section 5.1.1 the chosen ISO controls are listed. Below we will show our thought process behind three CWEs and why we thought they were connected to the specific ISO controls.

#### CWE-326: Inadequate Encryption Strength

This CWE has the description:

> The software stores or transmits sensitive data using an encryption scheme that is theoretically sound, but is not strong enough for the level of protection required.

It has the Technical Impacts 'Bypass Protection Mechanism' and 'Read Application Data' connected to it. This CWE is very specific, the encryption is not strong enough. This is directly connected to the 'Cryptography' category which we included two ISO-controls from. 10.1.1 goes into the level of encryption, which is the main problem in this CWE. 10.1.2 is more specific with key management which is not necessarily connected to this CWE, but it can not be excluded, since the encryption may fail because of the key management.

The ISO-controls connected to it were therefore set to 10.1.1, and 10.1.2.

### CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

CWE-200 has the description

> The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.

and the Technical Impact 'Read Application Data'. This CWE is not as straightforward as CWE-326, since it is connected to both sensitive data and authorization. The ISO-controls chosen for this CWE were: 9.2.3, 9.2.5, 9.2.6, 9.3.1, 9.4.1, 12.1.4, 13.2.3, 14.1.1 and 14.1.2.

Authorization is strongly connected to privilege management which is why it is connected to 9.2.3, 9.2.5 and 9.2.6 which are all connected to privilege management. It continues with 9.3.1 and 9.4.1 which are both connected to access management. The control 12.1.4 is chosen since it brings up unauthorized access if environments are not separated. The controls 13.2.3, 14.1.1 and 14.1.2 also talk about unauthorized access, which of course applies to this CWE.

This is why the above ISO-controls were chosen. The reason for not including other similar ISO controls, e.g. 9.2.1 and 9.2.2 is because these two only mention basic user access rights, which are not connected to CWE-200 since it does not apply to sensitive information.

### CWE-269: Improper Privilege Management

This CWE has the description:

> The software does not properly assign, modify, track, or check privileges for an actor, creating an unintended sphere of control for that actor.

and has the Technical Impact 'Gain Privileges or Assume Identity'. This CWE is clearly connected to privilege management, and therefore the ISO controls connected to access control. The controls 9.1.1, 9.2.2, 9.2.3, 9.2.5 and 9.2.6 goes through access control policies, access rights and the need for access rights to be restricted, reviewed and kept up to date with the current policies. The controls 9.4.1 and 9.4.5 go into the implementation of access rights, and advise what information should be limited. Lastly 14.2.6 brings up the need for a secure development environment, which is connected to access control and therefore privilege management.

The reason that only these controls were chosen is because they are directly connected to access control and privilege management. However controls concerning authorization and authentication, which are also connected to access control, are not included. This is because they are only connected in the sense that access control needs to be implemented already for authorization and authentication to work.

Chapter 4

# Environmental Analysis

In this chapter we want to present our findings to help calculate the environmental score. The purpose of an environmental analysis is to understand the vulnerabilities from the business's point of view. This is also what is needed to start making a risk assessment, since it all starts from the business's point of view. To make a risk assessment we need to understand how the vulnerability can affect the business, and based on that look at the risks concerning the vulnerabilities. This part continues with the risk identification done by finding the vulnerabilities, and now checking if they are applicable to the system.

## 4.1 Security Requirements

### 4.1.1 Determining the requirements

In the figures 4.1, 4.2 and 4.3 are a few questions that can help clients figure out what their security requirements are. If they are unsure about any of the questions, it might be good to set 'Not defined' as an answer for that specific security requirement. However if they are sure about the answer to any of the questions resulting in a high outcome the security requirement should be set to 'High', even though they may be unsure about any of the other questions. Figure 4.1 shows a flowchart for determining the confidentiality requirement metric, with the three different outcomes, 'High', 'Medium' and 'Low'. Figure 4.2 instead presents a flowchart for the integrity requirement metric, with the same outcomes as mentioned above. Lastly there is Figure 4.3 which shows how to determine the availability requirement metric with the same outcomes as mentioned above.

#### Confidentiality

- **Do you store highly sensitive data in your application?**

  Highly sensitive data include personal information and can be, e.g. financial account information, passwords, or social security number. If highly sensitive data is being stored higher security is needed, since the information can be very damaging if leaked. There are a lot of regulations when it comes to sensitive data, e.g. GDPR. If these regulations are not followed there will be negative consequences, e.g. fines or bans [29].

- **Do you store login credentials without encryption?**

  Without encryption refer to credentials stored in plain text, not using salt, or using out-dated hash algorithms. If, e.g. SSO is being used the identity provider is the one storing the login credentials which means this question can be ignored. Since users often use the same login credentials on different sites this information can be very sensitive and should be stored securely.

- **Can your data be openly shared with the public?**

  This may vary between different organisations, but the information shared with the public should never include sensitive data, information about network equipment or information used to make business decisions. If all the data on the system can be shared with the public or if the sensitive data is stored in a different system, the system will not need a high confidentiality requirement.

- **Does your application store data about your network equipment, e.g. routing or forwarding tables?**

  If the application stores any data about network equipment, either at rest or in motion, the answer should be yes. Data about their network equipment can let attackers get access to the network topology around the routing table. This gives the attacker information about their communication, which can be sensitive information.

- **Is your sensitive data encrypted at rest, i.e. is your inactive data encrypted?**

  Sensitive data is, as mentioned before, personal information. Data at rest refers to inactive data that is often protected by, e.g. a firewall. Active data, or data in motion, is instead data being accessed frequently, e.g. through a database that is accessed via applications [30]. Often data that is not used frequently can be forgotten, but it can still contain highly sensitive information. If it is not encrypted and the protection gets breached, the data will be available to the attacker.

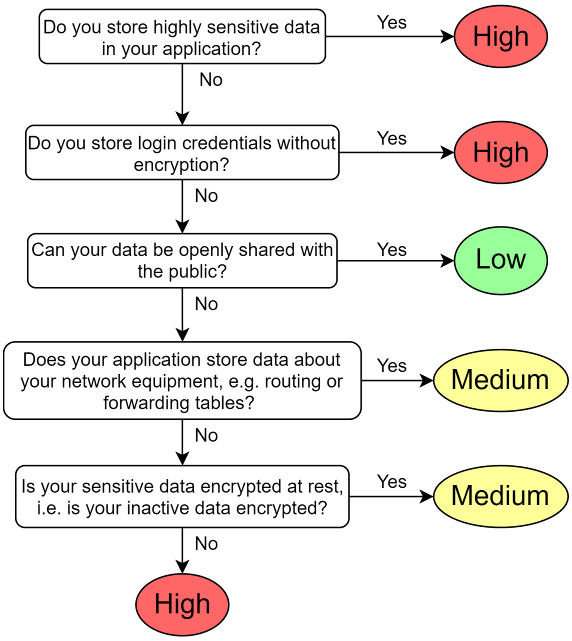
Do you store highly sensitive data in your application?
Yes
High
No
Do you store login credentials without encryption?
Yes
High
No
Can your data be openly shared with the public?
Yes
Low
No
Does your application store data about your network equipment, e.g. routing or forwarding tables?
Yes
Medium
No
Is your sensitive data encrypted at rest, i.e. is your inactive data encrypted?
Yes
Medium
No
High

**Figure 4.1:** Confidentiality requirement

Integrity

- **Does your application contain monetary transactional data, e.g. payments or other funds?**

  This only applies if you do not use an outside source or third party system for your transactional data. The consequences if the integrity is impacted during monetary transactions can be very severe, since when information is modified the transactions can go to other parties than the intended, and the payment sum can be changed.

- **Does your application contain personally identifiable information, i.e. data that can be used to identify a person, e.g. social security number or bank account information?**

  Personally identifiable information is sensitive data and if it is stored in the application, whether it is protected or not, the answer should be yes. The alteration of sensitive data can have huge consequences, e.g. transactions or identity theft, and it is very important that this data is correct.

- **Does your application contain data used to make health decisions?**

  This is connected to sensitive information and personal data. The alteration of this data can have severe consequences that affect people's health and can even lead to death, e.g. giving out the wrong dose of medication or giving medication which the patient may be allergic to. There are also special rules with data connected to making health decisions as set by, e.g. FDA.

- **Does your application store information about your firewalls?**
  If data about their firewalls is stored anywhere on their system, the answer should be yes. Firewalls are often used to restrict connectivity to sensitive areas [31]. Integrity can be negatively impacted if the data about their firewalls get out since they contain information about what traffic is allowed, and if this gets edited the firewall is no longer secure and outside forces can get access to the system.
- **Does your application contain data used for making business or risk management decisions? How severe is the impact of the decisions made from this data?**
  Any data concerning business or risk decisions that should not be publicly available. Depending on the severity of the impact from these decisions, it tells us how severe the consequences could be if they got out. Data used for making decisions can compromise the integrity if it is made public, since this may impact the way you make decisions.

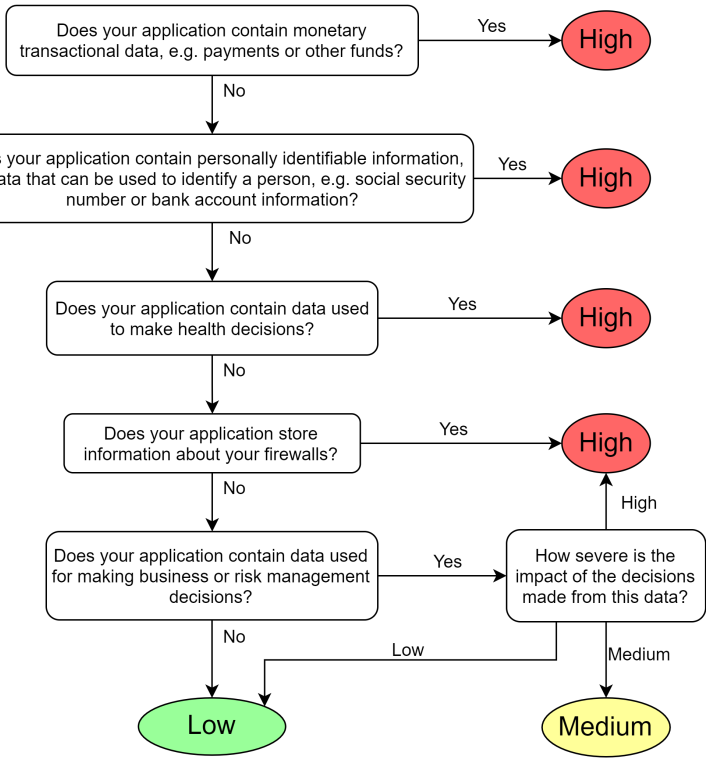
Does your application contain monetary transactional data, e.g. payments or other funds?
Yes
High
No
Does your application contain personally identifiable information, i.e. data that can be used to identify a person, e.g. social security number or bank account information?
Yes
High
No
Does your application contain data used to make health decisions?
Yes
High
No
Does your application store information about your firewalls?
Yes
High
No
High
Does your application contain data used for making business or risk management decisions?
Yes
How severe is the impact of the decisions made from this data?
No
Low
Medium
Low
Medium

**Figure 4.2:** Integrity requirement

### Availability

- **Does your application require rapid response times due to transactional purposes?**

If their application includes any transactions that are not using a third party system, the answer should be yes. Any system which requires rapid response times will have a 'High' need for availability. The reason that rapid response times are connected to transactional purposes is because each transaction process needs to be completed before the next one since the system otherwise can be abused. If the response time would be too long this would limit the system's capacity which means you could lose out on business.

- **Do you have full capacity redundancy?**

  With full capacity redundancy their system has a complete backup that will run if the main system is taken down. If they have full capacity redundancy their system is less sensitive against availability attacks, since the consequences will diminish.

- **Do you use clustered devices?**

  Clustered devices use multiple systems that all are assigned the same task so that if one of the systems fail, another will be used instead and keep up the availability [32]. As with full capacity redundancy the consequences will diminish if exposed to attacks on availability, since they still let the system run.

- **How long downtime is acceptable?**

  Downtime refers to when the system is unavailable, offline or not operational, i.e the period of time where the system fails to perform its primary function [33]. If the system is low traffic and not crucial, a long downtime can make only a small impact on the system. Therefore availability attacks will not give very severe consequences. However, if it is a very crucial part of the system availability attacks can be very impactful even for a very short downtime.

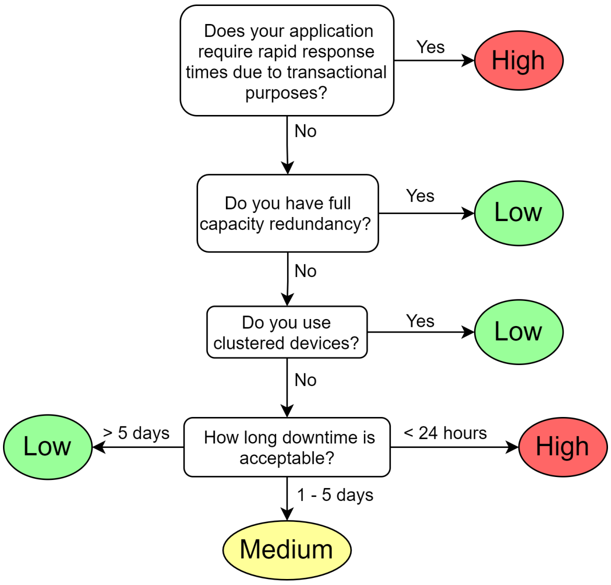
Does your application require rapid response times due to transactional purposes?
Yes
High
No
Do you have full capacity redundancy?
Yes
Low
No
Do you use clustered devices?
Yes
Low
No
How long downtime is acceptable?
> 5 days
Low
< 24 hours
High
1 - 5 days
Medium

**Figure 4.3:** Availability requirement

### 4.1.2 Discussion

The above questions will in most cases be enough to give a simple but accurate score for each security requirement, as they are based on the CVSS user guide [5]. However, these questions could benefit from being extended by covering more specific cases and outliers. As we can see there are not many different outcomes for the security requirements, and they can be divided into below average requirements, average requirements, and above average requirements. These scores are then used as a multiplier in a function for calculating the complete environmental score. However, the security requirement does not affect the score if the corresponding impact metric is 'None'.

It is also interesting to see what has been said in the literature about this. In the paper 'On the Impact of Environmental Metrics on CVSS Scores' Gallon [11] discussed the impact the security requirements have on the CVSS score. In his paper he concluded that the integrity requirement had a much higher effect on the final score than the confidentiality and availability requirement. This is something that is important to have in mind, that depending on your environment and values, each requirement can have a different weight from each other. This also shows that the security requirements do not present a completely accurate evaluation of the severity of a vulnerability.

## 4.2 Modified Base Metrics

The modified base metrics can vary a lot depending on who did the calculation, which is shown in Allodi et al's [10] paper where they show the difficulty in setting modified base metrics. What we want to show is that even the base metric can be different, depending on who did the calculation. This can be seen by checking different vendors' calculated CVSS score which sometimes differ from the CVSS score made by the NVD. One vendor with a big collection of CVEs and vulnerabilities is Red Hat, and even though they often have the same CVSS score as what NVD has calculated, they sometimes differ a lot. This is because the NVD evaluates it on a larger scale, whereas Red Hat looks for a more specific usage with their own product.

Red Hat has a lot fewer CVEs than NVD since they only include the ones affecting their product. During 2020 NVD listed 14865 vulnerabilities, which each has a CVE identifier, whereas Red Hat only listed 1992 of these CVEs during 2020.

When looking at the different CVEs on NVD from 2020 and comparing them to the Red Hat CVSS score, we derived 646 CVEs where at least one base metric differed. In Table 4.1 the base metrics that differed the most are listed, together with how many times Red Hat deemed the metric more severe, and when NVD deemed them more severe. 440 of them were considering at least one of the exploitability metrics, whereas the remaining 206 only differed in the impact metrics. Out of the 1992 taken from Red Hat, 646 of them differed from the score NVD had given them, which is approximately 32% which had a different score. This of course means that 68% of the CVEs had the same score. However, it still gives us an overview of the inconsistency of a CVSS score. It shows that the base score is not something to completely base one's decision making on, at least not without doing an environmental analysis first.

**Table 4.1:** Differences in CVSS score between Red Hat and NVD

| | **Number of Differences** | **Red Hat more severe** | **NVD more severe** |
|---|---|---|---|
| **Attack Vector** | 69 | 35 | 34 |
| **Attack Complexity** | 213 | 18 | 195 |
| **Privileges Required** | 108 | 60 | 48 |
| **User Interaction** | 88 | 26 | 62 |
| **Scope** | 126 | 76 | 50 |
| **Confidentiality Impact** | 201 | 121 | 80 |
| **Integrity Impact** | 199 | 73 | 126 |
| **Availability impact** | 159 | 88 | 71 |

As seen in Table 4.1, the attack complexity was estimated as 'High' (less severe) a lot more often according to Red Hat, whereas NVD often deemed the attack complexity as 'Low' (more severe). It is also the metric differing the most out of all the CVSS metrics, which means that it can be more important to check it out for yourself.

Because of these differences in metrics, the importance of making an environmental analysis of the CVSS score becomes very clear. However, it is difficult to give a step by step solution to follow in this case, as there are so many different factors which affect the scoring.

### 4.2.1 Comparison

The main difference between the CVSS score given by NVD and the one given by Red Hat is that NVD is trying to set it in the worst case scenario, where we do not know anything about the system. Red Hat instead gives a score based on what products use it and how it is built [34].

Below are some examples where the CVSS score has differed in any of the CVSS metrics.

#### CVE-2020-10379

The description for CVE-2020-10379 is:

> In Pillow before 7.1.0, there are two Buffer Overflows in libImaging/TiffDecode.c.

This vulnerability is connected to the 'CWE-120: Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')' which can "be used to execute arbitrary code, which is usually outside the scope of a program's implicit security policy. This can often be used to subvert any other security service." and also "generally lead to crashes. Other attacks leading to lack of availability are possible, including putting the program into an infinite loop." [35].

- CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:H - **Red Hat**
- CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H - **NVD**

Above the scoring vectors are displayed, and they show that the Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR) and User Interaction (UI) are different.

The Attack Vector is assessed to 'Network' for Red Hat, whereas it was 'Local' according to NVD. This is interesting, since Red Hat deems it more severe than NVD. The reason for this is probably that Red Hat uses the vulnerable component on an unprotected network, which has direct access to the internet. NVD instead assesses that the vulnerable component would be lying behind a secure network, like a VPN (Virtual Private Network) and firewalls, which protects against attacks from the rest of the internet.

The reason for the difference in Attack Complexity is probably because the way Red Hat is using the vulnerable component they have an extra layer of protection which increases the complexity of the attack, whereas NVD creates a more generalised score.

Once again, Red Hat has a more specific view of the vulnerability. For Privileges Required they probably set it as 'Low', as anyone with access to the resource would need a basic user account with basic privileges. NVD instead set it as 'None', since they assume no privileges are required, but instead they show the need for User Interaction.

The User Interaction is actually more severe according to Red Hat, which says it is not required, whereas NVD presumes it is required. This is often connected to the Privileges Required, since the need for User Interaction means you get the privileges of that user. This may be why Red Hat set the privileges required as ’Low’, as the basic user, whereas NVD set it as ’None’ while they count on a User Interaction which of course have some basic privileges.

### CVE-2020-14390

This CVE has had two different descriptions which both start with:

> A flaw was found in the Linux kernel in versions from 2.2.3 through 5.9.rc5. When changing screen size, an out-of-bounds memory write can occur leading to memory corruption or a denial of service.

and is now ended with

> Due to the nature of the flaw, privilege escalation cannot be fully ruled out.

whereas NVD previously ended it with

> This highest threat from this vulnerability is to system availability.

It is connected to ’CWE-787: Out-of-bounds Write’ which can “result in corruption of data, a crash, or code execution” [36].

- CVSS:3.1/AV:L/AC:H/PR:H/UI:N/S:U/C:H/I:H/A:N - **Red Hat**
- CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:U/C:L/I:L/A:H - **NVD**

Here the most interesting difference in these vector strings are the three impact metrics, Confidentiality (C), Integrity (I) and Availability (A). Memory corruption is connected to both Confidentiality and Integrity. Integrity since it is possible to write to, i.e. change, the information which is a breach of integrity. Confidentiality since it is possible to get the information which is being overwritten which is a breach of confidentiality. The difference between High and Low in these cases are hard to decide, since it depends on what part of the system is vulnerable and what information is on there. Red Hat has also specified that privilege escalation is possible, which can lead to accessing data that is normally protected. This might be why they have a higher rating than NVD. The denial of service (DoS) is connected to Availability since it leads to the system crashing or not responding. According to NVD this was the biggest threat, with a score of High. Red Hat however has said the impact to availability is none in this case. The reason for this is probably that the way Red Hat is using the vulnerable part in the Linux kernel makes it not possible to perform a DoS attack.

Chapter 5

# Risk Evaluation

In this chapter we try to further help with the steps needed to make a risk assessment. The risk identification has already been done by going through the CVEs and checking if they are actually applicable to the system's environment. The Environmental Score is a good start to the risk analysis, since it takes into account the company's goals, objectives and values as well as putting them in the actual environment. However, the environmental score is still not a complete risk assessment, it is just a more accurate CVSS score, i.e. it measures severity. Going back to our definition of risk we know it has two main parts, "the adverse impacts that would arise if the circumstance or event occurs" and "likelihood of occurrence". The environmental analysis has given us the impact in the form of severity, but we also need to look at the likelihood. However, we will not do any likelihood estimations, since that is out of scope for this report, but instead we will help identify potential events that may damage what the customer considers worthy of protection. This can then be used as the first step for future likelihood calculations. This is why we continue with this chapter, where we go onto connecting the vulnerabilities via their CWEs to the controls of ISO 27002 to show how these vulnerabilities can affect the organization. It is important to remember that even what we use as security controls can have gaps in its security.

The main goal of this master thesis is, as mentioned, to help clients make a risk assessment based on the CVSS score, the environmental score, and our connection between the ISO controls and CWEs. In this chapter we want to remind everyone about the connection between CVEs and CWEs. The reason we use CWEs instead of CVEs is because they are less specific and easier to connect to risks. The CVEs are very specific for a certain type of code, system and usage which makes it hard to draw any real conclusions, since there is such a huge number of very similar CVEs which would all fall under the same risk type. The CWEs with their already existing connection to CVEs are our gateway to connect ISO controls to CVEs, and hence a connection to the environmental CVSS score. ISO controls are set up as an easy way to minimize risks in your environment, by following the controls they set up. Our connection can show which ISO controls need to be focused on, and if any specific control is extra vulnerable. This will help with prioritization during the future safety work and risk treatment.

## 5.1 ISO controls and CWEs

In this section we look at the ISO controls in ISO 27002 and compare them with the top 50 CWEs in 2020, to see if there is a connection between them. We will look at the top CWEs and show how they can be used to find risks. The top CWEs are taken from the list of CVEs in 2020, making it possible for us to make a connection between the CVEs and the ISO controls, by using the CWEs as a bridge.

### 5.1.1 ISO controls

Because of the difficulty in making an environmental analysis, we have also investigated how the controls in ISO 27002 can be connected to vulnerabilities.

ISO 27002 is a standard for information technology, security techniques and code of practice for information security controls. These controls are all very important for information technology and security, but not all are relevant for this study, since we focus on the technical vulnerabilities.

There are several different categories in ISO 27002, where we found the most important chapters to be 8: Asset Management, 9: Access Control, 10: Cryptography, 12: Operations security, 13: Communication Security, and 14: System Acquisition, development and maintenance. These all have a clear connection to vulnerabilities in software, either directly like with the access control, or more indirectly by bringing up the need for information to be classified correctly.

#### Asset management

8.2.1 This control mandates the need for information to be classified. It should in part be classified in terms of sensitivity to unauthorised exposure or modification.

8.2.3 This control talks about the need for asset handling to be implemented in accordance with the information classification scheme.

#### Access control

9.1.1 This control goes through the access control policy that should be implemented by the organization. It goes through different policies that should be accounted for, including security requirements for their business application, consistency between access rights and information classification policies, management of access rights, and segregation of access control roles.

9.2.1 This control describes the need for a formal user registration and deregistration process that should be implemented to enable access rights.

9.2.2 This control continues with the user access rights process, and the need for both assigning and revoking access rights to all users.

9.2.3 This control brings up the allocation and use of privileged access rights, which should be restricted and controlled.

9.2.4 This control describes the process for management of secret authentication information of users.

9.2.5 This control talks about the importance of the reviewing of access rights at regular intervals, and the logging of changes in privilege allocations.

9.2.6 This control goes through the need for adjustment and removal of access rights.

9.3.1 This control brings up the importance of keeping information confidential, the change of secret authentication information when compromised, password security, not sharing user's secret authentication information.

9.4.1 This control continues with the implementation of access right management. Limiting information in outputs, controlling access rights of users: read, write, delete, execute, controlling access rights of other applications, controlling which data can be accessed by specific users.

9.4.2 This control describes secure log-on procedures. Logging of unsuccessful and successful attempts, protection against brute force attacks, not transmitting confidential information in clear text, termination of inactive sessions.

9.4.3 This control goes into the specifics of password management systems, enforcing quality passwords, the storing of passwords separately from the application system, storing and transmitting passwords in a protected form, e.g. hash.

9.4.5 This control brings up the need for restriction of the access control to the program source code.

### Cryptography

10.1.1 This control describes the need for cryptographic controls, and how they should be developed and implemented. It brings up the required level of protection based on risk assessments, key management, roles and responsibilities. Goes through its impact on confidentiality, integrity, authenticity, authentication and non-repudiation.

10.1.2 This control goes into the specifics of key management. Generating keys, storing keys, public key certificates, destroying keys and logging key management related activities.

### Operations security

12.1.4 This control talks about the separation of environments. This means that the development, testing, and operational environments should be separated. This is to reduce the risks of unauthorized access or changes made in the operational environment.

12.2.1 This control brings up the importance of malware protection. It suggests ways to do this, e.g. white-listing, blacklisting, logging information, isolating environments, and ongoing reviews of critical software.

12.4.2 This control goes into the need for protection of log information.

12.6.2 This control describes the need for restrictions on software installation.

### Communications security

13.1.2 This control talks about security on networks. It brings up the need for authentication, encryption, secured connections, restricted access and network connection controls.

13.2.1 This control goes into information transfer policies and procedures. It brings up the need for protecting transferred information from interception, copying, modification, mis-routing and destruction. It also brings up the need for detection against malware, and the use of cryptography.

13.2.2 This control brings up the need for procedures to ensure traceability and non-repudiation in communication. It mentions the need for the minimum technical standards to be used for packaging and transmission, the need for cryptography, and that technical standards for recording and reading information and software is followed.

13.2.3 This control goes into the need for protecting messages from unauthorized access, modification or denial of service, corresponding to the classification scheme adopted by the organization. It also brings up the need for ensuring correct addressing and transportation of the message, the reliability and availability of the service, and stronger levels of authentication controlling access from publicly accessible networks.

### System acquisition, development and maintenance

14.1.1 This control brings up the need for different requirements in new or enhanced information systems. It mentions the authentication requirements, access provisioning and authorization process, required protection regarding the impact metrics, and also logging and monitoring.

14.1.2 This control talks about secure application services on public networks. They bring up the importance of authentication, authorization, integrity requirements, confidentiality requirements, verification, and avoidance of loss or duplication of transaction information;

14.1.3 This control goes through what is needed for protecting application services transactions. It brings up the need for electronic signatures, that the user's secret authentication information for all parties are valid and verified, that the transaction remains confidential, the communication should be encrypted and use secure protocols. It also mentions that the storage should be located outside of public accessible environments and the need for an end-to-end certificate/signature management process.

14.2.1 This control brings up the importance of a secure development policy. It mentioned the security of the development environment, the security in the software development methodology and the use of secure coding guidelines for each programming language used. It also brings up the need for secure repositories, security in the version control, required application security knowledge, and finally it brings up the need for the developers' capability of avoiding, finding and fixing vulnerabilities.

14.2.6 This control talks about the importance of a secure development environment. It brings up the need for segregation, access control, the moving of data to and from the environment, monitoring changes and considering the sensitivity of data.

### 5.1.2 CWE

#### Top 50 2020

Most CVE has a CWE connected to it. Going through all CVEs of 2020, we could see that 1817 out of 14865, i.e. 12% of the CVEs did not have a CWE connected to them yet. Out of the remaining 13048 the top 5 CWEs consisted of 38% (4957/13048) of the CVEs. The top 50 CWEs consisted of 90.5% of all CVEs with a CWE, and we therefore decided to focus on only the top 50, instead of the total of 187 CWEs connected to CVEs in 2020. In Table 5.1 we can see the top 50 CWEs in 2020 together with their number of occurrences. In Table C.1 in Appendix C the top 50 CWEs are listed together with their name.

**Table 5.1:** Top 50 most common CWEs in 2020

| CWE ID | No. |
|---|---|
| CWE-79 | 1670 |
| CWE-269 | 1017 |
| CWE-20 | 841 |
| CWE-200 | 791 |
| CWE-787 | 638 |
| CWE-125 | 526 |
| CWE-119 | 515 |
| CWE-89 | 373 |
| CWE-22 | 336 |
| CWE-416 | 307 |
| CWE-78 | 301 |
| CWE-352 | 295 |
| CWE-287 | 285 |
| CWE-74 | 255 |
| CWE-863 | 246 |
| CWE-120 | 240 |
| CWE-400 | 239 |
| CWE-276 | 219 |
| CWE-434 | 190 |
| CWE-522 | 183 |
| CWE-862 | 161 |
| CWE-190 | 146 |
| CWE-502 | 135 |
| CWE-476 | 133 |
| CWE-798 | 130 |
| CWE-732 | 115 |
| CWE-295 | 108 |
| CWE-94 | 103 |
| CWE-918 | 100 |
| CWE-306 | 96 |
| CWE-611 | 87 |
| CWE-601 | 84 |
| CWE-362 | 82 |
| CWE-77 | 79 |
| CWE-427 | 62 |
| CWE-319 | 58 |
| CWE-917 | 58 |
| CWE-312 | 54 |
| CWE-327 | 54 |
| CWE-532 | 54 |
| CWE-59 | 54 |
| CWE-326 | 50 |
| CWE-347 | 48 |
| CWE-843 | 47 |
| CWE-426 | 46 |
| CWE-668 | 44 |
| CWE-613 | 42 |
| CWE-755 | 41 |
| CWE-444 | 40 |
| CWE-311 | 39 |

### Technical Impacts

All CWEs are listed by the MITRE organisation in CWE version 4.3. This is the complete list of all Common Weakness Enumeration, which is a community-developed list of software and hardware weakness types.

There are 916 CWEs in the latest version of the CWE list. Out of these there are 22 different technical impacts they are divided into, not including the category 'Other'.

Each CWE can have one or more technical impacts. In Table 5.2 is a list of the technical impacts, which were mentioned in total 1674 times, with 182 of these being 'Read Application Data', which was mentioned the most.

**Table 5.2:** Technical impacts and the number of times they appear in the CWE list v4.3, and their primary scope

| Count | Technical Impact | Scope |
|---|---|---|
| 182 | Read Application Data | Confidentiality |
| 177 | Bypass Protection Mechanism | Access Control |
| 156 | Execute Unauthorized Code or Commands | Integrity |
| 140 | Gain Privileges or Assume Identity | Access Control |
| 117 | DoS: Crash, Exit, or Restart | Availability |
| 115 | Unexpected State | Integrity |
| 110 | Modify Memory | Integrity |
| 104 | Modify Application Data | Integrity |
| 87 | Read Memory | Confidentiality |
| 79 | Read Files or Directories | Confidentiality |
| 66 | Modify Files or Directories | Integrity |
| 50 | Alter Execution Logic | Integrity |
| 49 | Quality Degradation | Other |
| 40 | DoS: Resource Consumption (CPU) | Availability |
| 37 | Reduce Maintainability | Other |
| 35 | DoS: Resource Consumption (Other) | Availability |
| 31 | Hide Activities | Non-Repudiation |
| 26 | DoS: Resource Consumption (Memory) | Availability |
| 26 | Reduce Reliability | Other |
| 21 | DoS: Instability | Availability |
| 19 | Reduce Performance | Other |
| 7 | DoS: Amplification | Availability |

However, this does not mean that 'Read Application Data' is the most common technical impact, it is just the technical impact with the most number of CWEs connected to it. A CWE can have more than one technical impact connected to it, and it is hard to determine which technical impact is the most common. This is because the technical impacts can be relevant in different scenarios. For example a Denial of Service (DoS) impact and Execute Unauthorized Code or Commands can be used in very different scenarios, so the CWE being exploited does not necessarily exploit both of these technical impacts in the same scenario, i.e for a specific CVE.

The technical impacts and their scope can also be helpful when it comes to dealing with specific vulnerabilities. The scope can tell us what technical impacts we might encounter, and if they are relevant for our security requirements or that they can be more or less overlooked.

#### Changes over the years

We are soon going to connect the ISO controls to different CWEs. However, because we want this thesis project result to be usable for a while, we want to see the difference in CWEs over the years, to see if they change a lot, or if they will be very different in the next few years.

Checking the data from NVD for 2018, 2019 and 2020 we could see that there was a slight variation of the order of some of the CWE, but most of them had similar positions in the list. What we could notice was that in the bottom of the list there were a few new CWEs that had not been used in previous years. There were 40 new CWEs in 2020, in a total of 183 CWEs. Out of these 40 CWEs, they were connected to 74 CVEs, where one CWE had 27 CVEs connected to it, whereas 33 of the 40 CWEs only had 1 CVE connected to it. Looking at 2019 there were 9 unique that year, and in 2018 there were 6 unique. We could also see that some CWEs were new from 2019, and some that were in 2018 and 2019 did not exist in 2020. A table of this data can be seen in Table A.1.

Since 2018, 54 CVEs are using CWEs that are no longer used. That is 36 CWEs that have not been used since 2018. Since 2018, 108 CVEs have a new CWE. That is 49 new CWEs since 2018. Out of these 108 CVEs, 74 of them are new CWE since 2019, i.e. 40 CWEs.

What we can see from this data is that there might be a small trend in using more specific CWEs, since there was a total of 155 CWEs used in 2018, 164 CWEs used in 2019 and 183 CWEs used in 2020.

What we can conclude from this is that it does not seem to be a very high risk that the CWEs will become obsolete, at least not the ones that are connected to a high number of CVEs. The top 50 CWEs of 2020 are all used in the previous years, where 44/50 were top 50 in 2019, and 41/50 were top 50 in 2018. This makes it seem like the top CWEs are pretty steady. Hopefully this means that the ISO controls to CWEs can still be used in the next few years, with a minor need for changes.

## 5.2 Connecting CWEs to ISO controls

### 5.2.1 CWEs with ISO controls

When going through the ISO controls, most of them are not connected to software security, but instead risk management as a whole. This means that a lot of the controls are not relevant, but also that some of the controls are not enough. In

Table 5.3 the above ISO controls are connected to the top 50 CWEs in 2020. The table is missing some CWEs, which will be discussed in Section 5.2.2. To check what the CWEs are about, a list of the top 50 CWE IDs and their name can be found in Table C.1 in Appendix C.

**Table 5.3:** ISO controls connected to CWEs

| **ISO** | **CWE** |
|---|---|
| 8.2.1 | CWE-532, CWE-732 |
| 8.2.3 | CWE-532, CWE-732 |
| 9.1.1 | CWE-269 |
| 9.2.1 | CWE-862, CWE-732 |
| 9.2.2 | CWE-269, CWE-276, CWE-862, |
| 9.2.3 | CWE-269, CWE-200, CWE-276, CWE-295 |
| 9.2.4 | CWE-287, CWE-522, CWE-312 |
| 9.2.5 | CWE-269, CWE-200, CWE-276 |
| 9.2.6 | CWE-269, CWE-200 |
| 9.3.1 | CWE-200, CWE-287, CWE-522, CWE-798, CWE-319, CWE-532, CWE-311, CWE-295, CWE-312 |
| 9.4.1 | CWE-269, CWE-200, CWE-276, CWE-668, CWE-732, CWE-312 |
| 9.4.2 | CWE-20, CWE-287, CWE-522, CWE-862, CWE-319, CWE-311, CWE-312 |
| 9.4.3 | CWE-522, CWE-798, CWE-319, CWE-532, CWE-311 |
| 9.4.5 | CWE-269, CWE-276, CWE-862 |
| 10.1.1 | CWE-522, CWE-798, CWE-319, CWE-532, CWE-311, CWE-326, CWE-327, CWE-347, CWE-295, CWE-312 |
| 10.1.2 | CWE-798, CWE-319, CWE-311, CWE-326, CWE-347, CWE-295 |
| 12.1.4 | CWE-668, CWE-200, CWE-434 |
| 12.2.1 | CWE-434, CWE-502, CWE-532 |
| 12.4.2 | CWE-532, CWE-862 |
| 12.6.2 | CWE-434, CWE-276 |
| 13.1.2 | CWE-287, CWE-522, CWE-862 |
| 13.2.1 | CWE-862, CWE-798, CWE-319, CWE-311, CWE-295 |
| 13.2.2 | CWE-798, CWE-319, CWE-311 |
| 13.2.3 | CWE-200, CWE-287, CWE-522, CWE-798, CWE-319, CWE-532, CWE-311, CWE-862, CWE-863 |
| 14.1.1 | CWE-200, CWE-287, CWE-306, CWE-863, CWE-522, CWE-862, CWE-532 |
| 14.1.2 | CWE-200, CWE-287, CWE-863, CWE-522, CWE-862, CWE-532, CWE-306, CWE-502, CWE-732, CWE-295, CWE-312 |
| 14.1.3 | CWE-287, CWE-522, CWE-862, CWE-502, CWE-798, CWE-319, CWE-311, CWE-295 |
| 14.2.1 | CWE-20, CWE-22, CWE-74, CWE-78, CWE-400, CWE-190, CWE-475, CWE-94, CWE-362, CWE-77, CWE-427, CWE-59, CWE-842, CWE-426, CWE-613 |
| 14.2.6 | CWE-269, CWE-276, CWE-862, CWE-755 |

Looking at Table 5.3 we can see that a few of the controls have a lot more CWEs connected to them than others, the top five being 14.2.1 (15), 14.1.2 (11), 10.1.1 (10), 9.3.1 (9) and 13.2.3 (9). The top one here is probably the top since it has such a broad control, i.e. 'secure development policy'. This however does not tell us much, since it does not specify what a secure development policy is. The following controls are also very broadly defined, which tells us that the controls could probably be a bit more specific to more clearly show us if they are fulfilled or not.

### 5.2.2 Other

Some CWEs are too specific to match to one of the ISO 27002 controls. These will be listed here, with descriptions of why they did not get assigned to an ISO.

#### Memory related

**Table 5.4:** Memory related CWEs

| **CWE ID** | **Name** |
|---|---|
| CWE-787 | Out-of-bounds Write |
| CWE-125 | Out-of-bounds Read |
| CWE-119 | Improper Restriction of Operations within the Bounds of a Memory Buffer |
| CWE-416 | Use After Free |
| CWE-843 | Access of Resource Using Incompatible Type ('Type Confusion') |

The above CWEs in Table 5.4 are all related to memory allocation and other memory related things. This is however very language specific, and are only relevant if languages like C, C++ or Assembly are used.

### Web based

**Table 5.5:** Web related CWEs

| **CWE ID** | **Name** |
|---|---|
| CWE-79 | Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') |
| CWE-352 | Cross-Site Request Forgery (CSRF) |
| CWE-918 | Server-Side Request Forgery (SSRF) |
| CWE-611 | Improper Restriction of XML External Entity Reference |
| CWE-444 | Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling') |
| CWE-601 | URL Redirection to Untrusted Site ('Open Redirect') |
| CWE-613 | Insufficient Session Expiration |

In Table 5.5 the CWEs listed are all related to web programming. These are all very serious weaknesses, but they do not have a strong connection to any of the ISO controls since they are too specific. These are however related to some controls listed by NIST in their Special Publication 800-53 [37].

### Other CWEs

**Table 5.6:** CWEs without a specific category

| **CWE ID** | **Name** |
|---|---|
| CWE-89 | Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') |
| CWE-476 | NULL Pointer Dereference |

The CWEs in Table 5.6 are too specific for the ISO controls. CWE-89 is database related, whereas CWE-476 can be from bad coding practice or other flaws.

## 5.2.3 Conclusion

What we can see is that even the few ISO controls that were actually connected to the software vulnerabilities regarding this thesis, were still not enough to cover the top 50 CWEs. Being ISO certified is something a lot of businesses want, and it is a sort of stamp of approval that everything is being done correctly. However, since there is a lack of details when it comes to cybersecurity, it does not necessarily mean that something that is ISO certified is actually secure. It is especially important for the business themselves to understand this, so that they know that there are still vulnerabilities and risks in their organisation. One way to help is to use other frameworks in addition to ISO, for example the NIST security controls for information systems, (NIST SP 800-53), called 'Security and Privacy Controls for Information Systems and Organizations' [37]. This is a great complement to the ISO 27002 controls, with more in depth controls for, e.g. Access control, authentication, cryptography, secure development and risk assessment to name a few.

NIST also provides their own risk management framework, which can complement the ISO 27000 series and the ISO 31000, called 'Risk Management Framework for Information Systems and Organisation', (NIST SP 800-37) [26]. This has more of a focus on cybersecurity than the more general ISO controls previously mentioned.

## 5.3 Usability

In this section we show how our findings can be used to help clients make their own risk assessment.

The first thing to do is look over the entire risk management process. It consists of three major parts, determining the scope, the risk assessment and the risk treatment. In this thesis we focus on the risk assessment part, which consists of the risk identification, risk analysis and risk evaluation. However, all steps are still vital for any risk management process and it is important to remember to do all the steps. The second thing to remember is that risk management is not a one time operation, it is something that needs to be done continuously throughout the entire project.

When following our methods it is assumed that the scope and context is already decided by the client and we can start with the risk assessment. Our method also relies on that all vulnerabilities connected to the clients application has been provided, e.g. via an outside source like Debricked. Our next step is to make the environmental analysis. This is something that will be easier to do after each time, since the future evaluations can use previous data for simplifying the environmental analysis process. The security requirements will not differ much unless big changes are made, and for the modified base metrics it is easy to see how they have been estimated earlier and if those estimations seemed correct. This will lead to the environmental analysis being faster and more accurate for each time it is being done.

The next step is to continue evaluating these vulnerabilities and also connecting them to something more concrete, i.e. the ISO-controls. The way we do this is by using the CWEs connected to the vulnerabilities and connecting the CWEs to the ISO-controls. This data is what our method results in. The results can be used in the risk management process, both in the risk evaluation and to help make decisions for the risk treatment. To further show how this research can be used, an example is done in Section 5.4.

## 5.4 Example

In this section we want to present an example of how to use our proposed method as an aid when making a risk assessment.

### 5.4.1 Background

In ISO 27002 the control 9.3.1 has 'password vault' as one example for storing secret authentication information [28]. They also go into specific controls for password management systems in the control 9.4.3. This is of course a great way to protect your information, but that does not mean it is risk free. In this example we will look at a password manager, and show that there can still be vulnerabilities in the software we use to protect ourselves. We chose to look at the password manager 1Password.

### 5.4.2 Model

The first step in our risk assessment model is to look at the vulnerabilities found. These can be given to us by an external source, e.g. Debricked, which list all the CVEs that impact the client. In this case we searched for 1Password related CVEs on NVD and got six vulnerabilities, as listed in Table 5.7. These vulnerabilities each have their CVSS score, as provided by a vendor or organisation like NVD. This score can show us a first priority about the vulnerabilities and determine if we feel the need to look at all vulnerabilities or focus on a few specific CVEs. For a more comprehensive analysis, we need to look at the environmental score. The 6th vulnerability, CVE-2012-6369, only has an old CVSS score from version 2.0, and will therefore not be included in the Environmental Analysis.

**Table 5.7:** Vulnerabilities found in 1Password with their CVSS score and associated CWEs

| No. | CVE-ID | CVSS score | CWEs |
|---|---|---|---|
| 1 | CVE-2021-26905 | 6.5 | CWE-522 |
| 2 | CVE-2020-10256 | 9.8 | CWE-335 |
| 3 | CVE-2014-3753 | 5.5 | CWE-200 |
| 4 | CVE-2018-19863 | 5.5 | CWE-532 |
| 5 | CVE-2018-13042 | 5.9 | CWE-20 |
| 6 | CVE-2012-6369 | N/A | CWE-79 |

#### Environmental Score

The environmental score has two parts, the security requirements and the modified base metrics. Starting with the security requirements we need to know more about the organisation. 1Password's main objective is to store and manage passwords. However this is not the only thing they do, they also act as an authenticator, store copies of important documents, and save credit card details, to name a few [38].

Looking at the Confidentiality Requirement the first question asks if they store highly sensitive data. Since it is a password vault, they do store highly sensitive data, i.e. passwords and user data, which means their Confidentiality Requirement will be set to High. The other questions are now superfluous, since the outcome will always be 'High'.

The Integrity Requirement will also be set to 'High' since the data they store can be connected to personally identifiable information in the form of e.g. bank account data. Both Confidentiality and Integrity is also something that is very important to them, as mentioned in their security design document [39].

Lastly there's the Availability Requirement. 1Password uses Amazon Web Services (AWS) as their hosted services. AWS provides several availability zones [40] which leads to full capacity redundancy if any of the zones would go down. Since 1Password is not immediately connected to transactional purposes the Availability Requirement can be set as 'Low' since they do have full capacity redundancy.

**Table 5.8:** Estimated Values for 1Password's Security Requirements

| Security Requirement | Value |
|---|---|
| Confidentiality Requirement (CR) | High (H) |
| Integrity Requirement (IR) | High (H) |
| Availability Requirement (AR) | Low (L) |

Calculating an updated CVSS score is now possible, and can be done by using the calculator provided by FIRST [41]. The security requirements are summarized in Table 5.8. These give us a new and partially improved CVSS score when calculated which can be seen in Table 5.9.

**Table 5.9:** Updated CVSS score after applying the Security Requirements part of the Environmental Score

| No. | CVSS Base Score vector string | Score |
|---|---|---|
| 1 | CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N | 8.3 |
| 2 | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H | 9.8 |
| 3 | CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N | 7.3 |
| 4 | CVSS:3.0/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N | 7.3 |
| 5 | CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H | 4.1 |

Our next step is to find out if there are any modified base metrics. Calculating the modified base metrics is very system specific and the steps to solve this are not included in this report since it is beyond our scope. Instead we encourage the user to look at the CVSS score done by different vendors. In our report above we picked Red Hat, but since 1Password is not connected to Red Hat they have not made a CVSS analysis themselves. However we did find some examples done by vulDB, another big vulnerability database. They are not as forthcoming with their numbers however, but we did find their CVSS vector for two of the above vulnerabilities.

**Table 5.10:** The CVSS vector strings for CVE-2021-26905 by the NVD and vulDB databases

| DB | CVSS Base Score vector string | Score |
|---|---|---|
| NVD | CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N | 6.5 |
| vulDB | CVSS:3.0/AV:A/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:L | 5.7 |

There are a few differences in the CVSS vector strings for NVD and vulDB as seen in Table 5.10. The first one is that the Attack Vector went from 'Network' at NVD to 'Adjacent' in vulDB. This might be because vulDB assumes that the affected component is deployed on a secure network behind a firewall. This is however only speculation since vulDB does not disclose the background to their decisions. Looking at the impact metrics instead, vulDB changed the Confidentiality impact from 'High' to 'Low', but also changed the other two impact vectors from 'None' to 'Low'.

**Table 5.11:** The CVSS vector strings for CVE-2020-10256 by the NVD and vulDB databases

| DB | CVSS Base Score vector string | Score |
|---|---|---|
| NVD | CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H | 9.8 |
| vulDB | CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N | 3.7 |

There are two big differences in the CVSS vector strings in Table 5.11. The first one is that vulDB deems the Attack Complexity to be High whereas NVD sets it as Low. This turns down the score from 9.8 to 8.1. The second big thing is the impact metrics. As we can see NVD deemed the impact to be high in all the three impact metrics, whereas vulDB only thought that Confidentiality was impacted, and to a low degree.

Deciding to further investigate or not is up to the company, but for this example we chose to go with a mix of the scores, with a main focus on NVD's vector string. In the case where there is no middle option between NVD's and vulDB's score the NVD score will be picked. If there is an alternative between vulDB's and NVD's metric score that alternative will be chosen. With this the only change would be the Integrity metric and Availability metric, which would be changed to 'Low' giving us the string CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:L. However, this is not the correct way to decide the modified base metrics, and further investigations should be done for a more accurate score. The new complete Environmental CVSS score can be calculated once again with the help of FIRST's calculator [41].

**Table 5.12:** Old and Updated CVSS Score

| No. | CVE-ID | Base CVSS Score | Environmental CVSS Score |
|---|---|---|---|
| 1 | CVE-2021-26905 | 6.5 | 6.0 |
| 2 | CVE-2020-10256 | 9.8 | 9.7 |
| 3 | CVE-2014-3753 | 5.5 | 7.3 |
| 4 | CVE-2018-19863 | 5.5 | 7.3 |
| 5 | CVE-2018-13042 | 5.9 | 4.1 |

The Security Requirements and Modified Base Metrics give us an Environmental CVSS score which is more accurate for our system than the Base CVSS score. The new scores can be seen in Table 5.12.

### Risk Evaluation

Our next step is to further analyze and evaluate these vulnerabilities. Here we can further sift between the vulnerabilities, if we notice that the Environmental CVSS score is low, and therefore not a priority. In this case there are only six vulnerabilities, and we will therefore look at all of them. What we do now is that we list all CWEs for each CVE we have and check them against Table B.1 in Appendix B. The CWEs for these vulnerabilities are listed in Table 5.7. A summary of the CWEs and ISO controls are listed in Table 5.13

**Table 5.13:** 1Passwords CWEs and their related ISO-controls

| No. | CWEs | ISO controls |
|---|---|---|
| 1 | CWE-522 | 9.2.4, 9.3.1, 9.4.2, 9.4.3, 10.1.1, 13.1.2, 13.2.3, 14.1.1, 14.1.2, 14.1.3 |
| 2 | CWE-335 | Not included in top 50 CWEs |
| 3 | CWE-200 | 9.2.3, 9.2.5, 9.2.6, 9.3.1, 9.4.1, 12.1.4, 13.2.3, 14.1.1, 14.1.2 |
| 4 | CWE-532 | 8.2.1, 8.2.3, 9.3.1, 9.4.3, 10.1.1, 12.2.1, 13.4.2, 13.2.3, 14.1.1, 14.1.2 |
| 5 | CWE-20 | 9.4.2, 14.2.1 |
| 6 | CWE-79 | Too specific (Cross-site Scripting) |

At this point we have gathered a lot of data which can be used for future risk assessments. The Environmental Score gives a good idea of how important these vulnerabilities will be to fix, and together with the connection to ISO-controls there is a clear ground to build one's decisions from. It also helps with prioritizing the risk management, since it shows that there now is a control that is vulnerable. It also shows us that the previous risk assessment the organisation has made can be wrong, since they probably use a password manager as a protection, thinking that will be enough. This new data shows that it is vulnerable, and that one of their ISO controls are no longer secure and therefore not necessarily fulfilled. This information can now be used to reassess the current risk assessment for the entire organisation. It is also a good base for any future risk management decisions, since it is something that will keep evolving with the client as they can use their old data to evaluate future vulnerabilities.

Chapter 6

# Conclusion

## 6.1 Project Aims

The three questions of this master thesis were

1 How can we help clients calculate the environmental CVSS score for their vulnerabilities?

2 How can we help clients assess each vulnerability?

3 How can we help clients make their own risk assessment based on their environmental CVSS score and the ISO standards?

### 6.1.1 Question 1

The environmental CVSS score has two major parts, the security requirements and the modified base metrics. We therefore had two different parts to help with. For the security requirements we put forward a few questions for each requirement to help determine what level each requirement should be at. We also gave descriptions for each question to help understand what was being asked. For the modified base metrics we did not have something as straightforward as for the security requirements, but instead gave some background to why and how to use modified base metrics. We showed the difference between Red Hat and NVD in their CVSS scores, and gave some examples where they differed and why that could be.

### 6.1.2 Question 2

The second question is strongly connected to the first, since the environmental score is a large part in assessing vulnerabilities. However, at this stage we also make the connection to the ISO-controls to help put the vulnerability in a wider context. To do this the CWEs connected to each vulnerability was used and a table was created to help match the CWEs to the ISO-controls and thereby connect the vulnerabilities to ISO-controls.

### 6.1.3 Question 3

The last question is the main question of this thesis, since it brings up that our goal is to help clients make their own risk assessment. This question is also built on the results from the previous two questions, as their data is what can be used when making a risk assessment. The data provided are an environmental score and a connection between each vulnerability and the ISO-controls concerning software security. This data gives both a quantitative severity score, which can easily be used when comparing vulnerabilities, and qualitative connection to ISO-controls which can be used to easier understand how the vulnerabilities affect you and guide what needs to be in focus for your future safety work. Both the quantitative data and the qualitative data gives an idea of what the consequences could be and how severe the consequences could be if left untreated.

### 6.1.4 Purpose

The objective of this master thesis was to:

> help organisations make decisions about vulnerabilities they have in their software programs by helping them make their own risk assessment.

We have tried to help clients understand vulnerabilities and the CVSS score and how it can be used to help prioritize the vulnerabilities. Our next step was to help make risk assessments easier by focusing on these vulnerabilities and how they are connected to the bigger picture. A complete risk management process consists of determining the scope, making a risk assessment, and finally the risk treatment. This master thesis should not be used as a guide to make a risk assessment, but instead as support used when making your own risk assessment. Risk management is something that needs to be worked with continuously, and this thesis gives a way to look at the data you get when making a risk assessment and how to use it for future risk assessment.

What we have tried to show with this master thesis is that it is important to look at vulnerabilities from a larger risk perspective, and that analyzing the vulnerabilities can help make continuous risk assessments easier.

## 6.2 Discussion

Parts of our work is corroborated by previous research, mentioned in Section 1.4. The main one being Allodi et al's paper about calculating modified base metrics. Here they show when non-experts were tasked to assess the modified base metrics for a fake scenario, which was then compared to some experts [10]. In our research we instead chose to look at two different vendors, which are both well trained in determining the CVSS score, but which both do not have a specific scenario in mind and instead a more general approach to the base metrics. There are also similarities between our research and the paper from Doynikova and Kotenko where they combined using the CVSS score with other risk assessment techniques [8]. Even though their approach is different from our connection between vulnerabilities and ISO-controls, it is a sign of the need to go outside cybersecurity metrics when making a risk assessment.

One thing our research could have benefited from is to have the model tested on outside clients. At the moment the research is just theoretical and a practical viewpoint would have been an interesting addition. This would be interesting to explore and with more time and resources this should be something to focus on.

## 6.3 Future Work

Cybersecurity and Risk management are two areas which get more important every day. With more and more things going online, cyber threat is one of the biggest global risk sources right now according to the World Economic Forum [42].

There has been a lot of previous work around assessing vulnerabilities, as mentioned in Section 1.4. There they analyze the current CVSS score and propose new scoring systems or use the CVSS score together with other techniques to make risk estimations. In our report we used the CVSS score and its environmental analysis together with the ISO-controls to help put the vulnerabilities in a larger risk perspective. However, there is a lot more research that can be done with this method, especially by using other frameworks like NIST's 'Security and Privacy Controls for Information Systems and Organizations', NIST Special Publication 800-53 [37] and their risk framework 'Risk Management Framework for Information Systems and Organisation', NIST Special Publication 800-37 [26]. These frameworks have a lot more focus on cybersecurity and risks than ISO 27002, and can therefore be easier connected to CWEs and vulnerabilities.

Another interesting thing to focus on would be going more into the modified base metrics and how to determine them. At the moment there is not a lot of publicly available information about how the decisions are made when determining the base metrics, except for a few examples and suggestions at first.org [43]. With more time and resources the derivation of the base metrics could be made clearer, and it could help create an easy model to determine the modified base metrics.

Another focus in the CVSS score can be the focus on the Temporal Metrics, since they give an idea of the current exploitability opportunities and if there are any existing patches [22]. This is definitely an area that would increase the accuracy of the severity rating from the CVSS score, and which would be an excellent addition to our research.

Other connections between cybersecurity and risk are also interesting and need future research. In Hubbard and Seieresen's book 'How to Measure Anything in Cybersecurity Risk' [44] they provide quantitative risk assessment methods in cybersecurity while also criticizing popular risk scores.

# Bibliography

[1] (2020). Cve and nvd relationship, MITRE, [Online]. Available: https://cve.mitre.org/about/cve_and_nvd_relationship.html (visited on 2021-03-01).

[2] G. Ollmann. (2019). Stop using cvss to score risk, Wired Business Media, [Online]. Available: https://www.securityweek.com/stop-using-cvss-score-risk (visited on 2021-05-27).

[3] C. Robinson. (2019). Why cvss does not equal risk: How to think about risk in your environment, Red Hat, [Online]. Available: https://www.redhat.com/en/blog/why-cvss-does-not-equal-risk-how-think-about-risk-your-environment (visited on 2021-05-27).

[4] R. Campagna. (2020). 5 reasons to stop using cvss scores to measure risk, Balbix, [Online]. Available: https://www.balbix.com/blog/5-reasons-to-stop-using-cvss-scores-to-measure-risk/ (visited on 2021-05-27).

[5] (2020). Common vulnerability scoring system version 3.1: User guide, FIRST, [Online]. Available: https://www.first.org/cvss/v3.1/user-guide (visited on 2021-02-27).

[6] "Risk management – Guidelines," International Organization for Standardization and the International Electrotechnical Commission, ISO/IEC 31000, 2018.

[7] T. Hamid and Á. MacDermott, "A methodology to develop dynamic cost-centric risk impact metrics," Dec. 2015, pp. 53–59. DOI: 10.1109/DeSE.2015.9.

[8] E. Doynikova and I. Kotenko, "Cvss-based probabilistic risk assessment for cyber situational awareness and countermeasure selection," in *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, 2017, pp. 346–353.

[9] S. H. Houmb, V. N. L. Franqueira, and E. A. Engum, "Quantifying security risk level from cvss estimates of frequency and impact," *J. Syst. Softw.*, vol. 83, no. 9, pp. 1622–1634, Sep. 2010, ISSN: 0164-1212. [Online]. Available: https://doi.org/10.1016/j.jss.2009.08.023.

[10] L. Allodi, S. Biagioni, B. Crispo, K. Labunets, F. Massacci, and W. Medeiros dos Santos, "Estimating the assessment difficulty of cvss environmental metrics: An experiment," Nov. 2017, pp. 23–39, ISBN: 978-3-319-70003-8. DOI: 10.1007/978-3-319-70004-5_2.

[11] L. Gallon, "On the impact of environmental metrics on cvss scores," in *2010 IEEE Second International Conference on Social Computing*, 2010, pp. 987–992. DOI: 10.1109/SocialCom.2010.146.

[12] H. Li, R. Xi, and L. Zhao, "Study on the distribution of cvss environmental score," in *2015 IEEE 5th International Conference on Electronics Information and Emergency Communication*, 2015, pp. 122–125. DOI: 10.1109/ICEIEC.2015.7284502.

[13] (2021). Capabilities: Cybersecurity, MITRE, [Online]. Available: https://www.mitre.org/capabilities/cybersecurity/overview (visited on 2021-02-27).

[14] (2020). About cwe, MITRE, [Online]. Available: https://cwe.mitre.org/about/index.html (visited on 2021-02-27).

[15] (2021). Cve numbering authorities, MITRE, [Online]. Available: https://cve.mitre.org/cve/cna.html (visited on 2021-02-27).

[16] (2021). About cve, MITRE, [Online]. Available: https://cve.mitre.org/about/index.html (visited on 2021-02-27).

[17] (2021). About nist, NIST, [Online]. Available: https://www.nist.gov/about-nist (visited on 2021-03-01).

[18] (2021). National vulnerability database: General information, NIST, [Online]. Available: https://nvd.nist.gov/general (visited on 2021-03-01).

[19] (2020). Common vulnerability scoring system sig, FIRST, [Online]. Available: https://www.first.org/cvss/ (visited on 2021-02-27).

[20] (2020). Cve terminology, MITRE, [Online]. Available: https://cve.mitre.org/about/terminology.html (visited on 2021-03-01).

[21] (2020). Cwe list version 4.4, MITRE, [Online]. Available: https://cwe.mitre.org/data/index.html (visited on 2021-03-26).

[22] (2020). Common vulnerability scoring system version 3.1, FIRST, [Online]. Available: https://www.first.org/cvss/v3.1/specification-document (visited on 2020-09-11).

[23] "Society for Risk Analysis Glossary," Society for Risk Analysis, Glossary, 2018.

[24] (2012). Risk cnssi 4009, NIST, [Online]. Available: https://doi.org/10.6028/NIST.SP.800-30r1 (visited on 2021-04-21).

[25] (2018). Cwe glossary, MITRE, [Online]. Available: https://cwe.mitre.org/documents/glossary/index.html (visited on 2021-04-21).

[26] (2018). Risk management framework for information systems and organisation, NIST, [Online]. Available: https://doi.org/10.6028/NIST.SP.800-37r2 (visited on 2020-09-11).

[27] "Information technology – Security techniques – Information security management systems – Overview and vocabulary," International Organization for Standardization and the International Electrotechnical Commission, ISO/IEC 27000, 2018.

[28] "Information technology – Security techniques – Code of practice for information security controls," International Organization for Standardization and the International Electrotechnical Commission, ISO/IEC 27002:2017, 2017.

[29] (2018). The eu general data protection regulation (gdpr), IT Governance, [Online]. Available: https://www.itgovernance.eu/sv-se/eu-general-data-protection-regulation-gdpr-se (visited on 2021-05-01).

[30] (2020). What is encryption at rest, and why is it important for your business? BrightlineIT, [Online]. Available: https://brightlineit.com/encryption-at-rest-important-business/ (visited on 2021-04-21).

[31] (2009). Guidelines on firewalls and firewall policy, NIST, [Online]. Available: https://doi.org/10.6028/NIST.SP.800-41r1 (visited on 2021-05-01).

[32] (2021). Computer cluster, SUSE, [Online]. Available: https://susedefines.suse.com/definition/computer-cluster/ (visited on 2021-04-21).

[33] (2021). Downtime, SUSE, [Online]. Available: https://susedefines.suse.com/definition/downtime/ (visited on 2021-04-21).

[34] (2020). Security flaws and cvss rescore process with nvd, Red Hat, [Online]. Available: https://www.redhat.com/en/blog/security-flaws-and-cvss-rescore-process-nvd (visited on 2021-04-23).

[35] (2021). Cwe-120: Buffer copy without checking size of input ('classic buffer overflow'), MITRE, [Online]. Available: `https://cwe.mitre.org/data/definitions/120.html` (visited on 2021-04-23).

[36] (2021). Cwe-787: Out-of-bounds write, MITRE, [Online]. Available: `https://cwe.mitre.org/data/definitions/787.html` (visited on 2021-04-23).

[37] (2021). Security and privacy controls for information systems and organizations, NIST, [Online]. Available: `https://doi.org/10.6028/NIST.SP.800-53r5` (visited on 2021-03-29).

[38] (2021). 1password does more than just store passwords, AgileBits Inc, [Online]. Available: `https://1password.com/why-1password/` (visited on 2021-05-09).

[39] (2021). 1password security design, AgileBits Inc, [Online]. Available: `https://1password.com/files/1Password-White-Paper.pdf` (visited on 2021-05-09).

[40] (2020). About hipaa compliance in 1password, AgileBits Inc, [Online]. Available: `https://support.1password.com/hipaa/` (visited on 2021-05-09).

[41] (2021). Common vulnerability scoring system version 3.1 calculator, FIRST, [Online]. Available: `https://www.first.org/cvss/calculator/3.1` (visited on 2021-05-11).

[42] "The global risks report 2019, 14th ed," World Economic Forum, Insight Report, 2019. [Online]. Available: `http://www3.weforum.org/docs/WEF_Global_Risks_Report_2019.pdf` (visited on 2021-05-14).

[43] (2019). Common vulnerability scoring system v3.1: Examples, FIRST, [Online]. Available: `https://www.first.org/cvss/examples` (visited on 2021-05-11).

[44] D. W. Hubbard and R. Seiersen, *How to Measure Anything in Cybersecurity Risk*. Hoboken, New Jersey: John Wiley & Sons, Inc, 2016, ISBN: 978-1-119-08529-4.

Appendix A

# Changes in CWEs over the year 2018-2020

**Table A.1:** Differences between the years, number of CVEs with a specific CWE. Red = unique for that year, Green = new since 2018, Yellow = Gone in 2020, Purple = Gone during 2019

| Placement/Year | **2020** | | **2019** | | **2018** | |
|---|---|---|---|---|---|---|
| **55** | CWE-1021 | 33 | CWE-19 | 40 | CWE-1236 | 24 |
| **57** | CWE-754 | 33 | CWE-310 | 39 | CWE-1188 | 21 |
| **64** | CWE-122 | 27 | CWE-88 | 34 | CWE-276 | 18 |
| **74** | CWE-617 | 23 | CWE-264 | 23 | CWE-285 | 14 |
| **87** | CWE-1236 | 14 | CWE-134 | 16 | CWE-113 | 8 |
| **88** | CWE-369 | 14 | CWE-320 | 16 | CWE-116 | 8 |
| **89** | CWE-776 | 14 | CWE-91 | 16 | CWE-185 | 8 |
| **90** | CWE-640 | 12 | CWE-116 | 15 | CWE-19 | 8 |
| **91** | CWE-284 | 11 | CWE-1187 | 15 | CWE-401 | 8 |
| **93** | CWE-121 | 10 | CWE-428 | 13 | CWE-682 | 8 |
| **96** | CWE-697 | 10 | CWE-552 | 10 | CWE-310 | 7 |
| **97** | CWE-134 | 9 | CWE-667 | 10 | CWE-358 | 7 |
| **100** | CWE-285 | 8 | CWE-354 | 9 | CWE-320 | 6 |
| **102** | CWE-252 | 7 | CWE-829 | 9 | CWE-123 | 5 |
| **105** | CWE-256 | 5 | CWE-693 | 8 | CWE-916 | 5 |
| **106** | CWE-338 | 5 | CWE-704 | 8 | CWE-172 | 4 |
| **108** | CWE-682 | 5 | CWE-93 | 8 | CWE-354 | 4 |
| **109** | CWE-704 | 5 | CWE-185 | 7 | CWE-388 | 4 |
| **110** | CWE-80 | 5 | CWE-193 | 7 | CWE-417 | 4 |
| **111** | CWE-131 | 4 | CWE-494 | 7 | CWE-538 | 4 |
| **114** | CWE-672 | 4 | CWE-284 | 6 | CWE-667 | 4 |
| **115** | CWE-693 | 4 | CWE-388 | 6 | CWE-681 | 4 |
| **116** | CWE-915 | 4 | CWE-399 | 6 | CWE-915 | 4 |
| **117** | CWE-916 | 4 | CWE-565 | 6 | CWE-93 | 4 |
| **118** | CWE-1187 | 3 | CWE-776 | 6 | CWE-184 | 3 |
| **119** | CWE-273 | 3 | CWE-189 | 5 | CWE-252 | 3 |
| **120** | CWE-288 | 3 | CWE-252 | 5 | CWE-335 | 3 |
| **121** | CWE-321 | 3 | CWE-254 | 5 | CWE-441 | 3 |

**Table A.1:** Differences between the years, number of CVEs with a specific CWE. Red = unique for that year, Green = new since 2018, Yellow = Gone in 2020, Purple = Gone during 2019

| Placement/Year | **2020** | | **2019** | | **2018** | |
|---|---|---|---|---|---|---|
| **123** | CWE-335 | 3 | CWE-331 | 5 | CWE-565 | 3 |
| **124** | CWE-385 | 3 | CWE-470 | 5 | CWE-670 | 3 |
| **125** | CWE-425 | 3 | CWE-672 | 5 | CWE-706 | 3 |
| **127** | CWE-459 | 3 | CWE-332 | 4 | CWE-909 | 3 |
| **129** | CWE-706 | 3 | CWE-131 | 3 | CWE-178 | 2 |
| **130** | CWE-73 | 3 | CWE-16 | 3 | CWE-273 | 2 |
| **132** | CWE-829 | 3 | CWE-275 | 3 | CWE-470 | 2 |
| **133** | CWE-208 | 2 | CWE-436 | 3 | CWE-662 | 2 |
| **134** | CWE-305 | 2 | CWE-749 | 3 | CWE-763 | 2 |
| **135** | CWE-358 | 2 | CWE-913 | 3 | CWE-922 | 2 |
| **136** | CWE-471 | 2 | CWE-21 | 2 | CWE-924 | 2 |
| **137** | CWE-61 | 2 | CWE-212 | 2 | CWE-118 | 1 |
| **138** | CWE-662 | 2 | CWE-417 | 2 | CWE-121 | 1 |
| **139** | CWE-669 | 2 | CWE-538 | 2 | CWE-212 | 1 |
| **140** | CWE-694 | 2 | CWE-662 | 2 | CWE-254 | 1 |
| **141** | CWE-114 | 1 | CWE-670 | 2 | CWE-255 | 1 |
| **142** | CWE-115 | 1 | CWE-838 | 2 | CWE-264 | 1 |
| **143** | CWE-117 | 1 | CWE-924 | 2 | CWE-297 | 1 |
| **144** | CWE-123 | 1 | CWE-99 | 2 | CWE-332 | 1 |
| **145** | CWE-1286 | 1 | CWE-117 | 1 | CWE-405 | 1 |
| **146** | CWE-130 | 1 | CWE-118 | 1 | CWE-407 | 1 |
| **147** | CWE-16 | 1 | CWE-172 | 1 | CWE-471 | 1 |
| **148** | CWE-170 | 1 | CWE-18 | 1 | CWE-642 | 1 |
| **149** | CWE-185 | 1 | CWE-184 | 1 | CWE-669 | 1 |
| **150** | CWE-197 | 1 | CWE-216 | 1 | CWE-707 | 1 |
| **151** | CWE-201 | 1 | CWE-23 | 1 | CWE-774 | 1 |
| **152** | CWE-23 | 1 | CWE-256 | 1 | CWE-838 | 1 |
| **153** | CWE-248 | 1 | CWE-297 | 1 | CWE-90 | 1 |
| **154** | CWE-255 | 1 | CWE-321 | 1 | CWE-913 | 1 |
| **155** | CWE-259 | 1 | CWE-371 | 1 | CWE-943 | 1 |
| **156** | CWE-261 | 1 | CWE-407 | 1 | | |
| **157** | CWE-266 | 1 | CWE-441 | 1 | | |
| **158** | CWE-270 | 1 | CWE-664 | 1 | | |
| **159** | CWE-279 | 1 | CWE-697 | 1 | | |
| **160** | CWE-299 | 1 | CWE-707 | 1 | | |
| **161** | CWE-303 | 1 | CWE-73 | 1 | | |
| **162** | CWE-334 | 1 | CWE-80 | 1 | | |
| **163** | CWE-342 | 1 | CWE-834 | 1 | | |
| **164** | CWE-349 | 1 | CWE-90 | 1 | | |
| **165** | CWE-350 | 1 | | | | |
| **166** | CWE-435 | 1 | | | | |

**Table A.1:** Differences between the years, number of CVEs with a specific CWE. Red = unique for that year, Green = new since 2018, Yellow = Gone in 2020, Purple = Gone during 2019

| Placement/Year | **2020** | | **2019** | | **2018** | |
|---|---|---|---|---|---|---|
| **168** | CWE-506 | 1 | | | | |
| **169** | CWE-507 | 1 | | | | |
| **170** | CWE-603 | 1 | | | | |
| **171** | CWE-684 | 1 | | | | |
| **173** | CWE-757 | 1 | | | | |
| **174** | CWE-759 | 1 | | | | |
| **175** | CWE-760 | 1 | | | | |
| **176** | CWE-805 | 1 | | | | |
| **177** | CWE-822 | 1 | | | | |
| **180** | CWE-87 | 1 | | | | |
| **181** | CWE-912 | 1 | | | | |
| **182** | CWE-920 | 1 | | | | |
| **183** | CWE-98 | 1 | | | | |

Appendix B

# List of CWEs connected to ISO controls

**Table B.1:** CWEs connected to ISO controls

| **CWE ID** | **ISO controls** |
|---|---|
| CWE-20 | 9.4.2, 14.2.1 |
| CWE-22 | 14.2.1 |
| CWE-59 | 14.2.1 |
| CWE-74 | 14.2.1 |
| CWE-77 | 14.2.1 |
| CWE-78 | 14.2.1 |
| CWE-94 | 14.2.1 |
| CWE-190 | 14.2.1 |
| CWE-200 | 9.2.3, 9.2.5, 9.2.6, 9.3.1, 9.4.1, 12.1.4, 13.2.3, 14.1.1, 14.1.2 |
| CWE-269 | 9.1.1, 9.2.2, 9.2.3, 9.2.5, 9.2.6, 9.4.1, 9.4.5, 14.2.6 |
| CWE-276 | 9.2.2, 9.2.3, 9.4.1, 9.4.5, 12.6.2. 14.2.6 |
| CWE-287 | 9.2.4, 9.3.1, 9.4.2, 13.1.2, 13.2.3, 14.1.1, 14.1.2, 14.1.3 |
| CWE-295 | 9.2.3, 9.3.1, 10.1.1, 10.1.2, 13.2.1, 14.1.2, 14.1.3 |
| CWE-306 | 14.1.1, 14.1.2 |
| CWE-311 | 9.3.1, 9.4.2, 9.4.3, 19.1.1, 10.1.2, 13.2.1, 13.2.2, 13.2.3, 14.1.3 |
| CWE-312 | 9.2.4, 9.3.1, 9.4.1, 9.4.2, 10.1.1, 14.1.2 |
| CWE-319 | 9.3.1, 9.4.2, 9.4.3, 19.1.1, 10.1.2, 13.2.1, 13.2.2, 13.2.3, 14.1.3 |
| CWE-326 | 10.1.1, 10.1.2 |
| CWE-327 | 10.1.1 |
| CWE-347 | 10.1.1, 10.1.2 |
| CWE-362 | 14.2.1 |
| CWE-400 | 14.2.1 |
| CWE-426 | 14.2.1 |
| CWE-427 | 14.2.1 |
| CWE-434 | 12.1.4, 12.2.1, 12.6.2 |
| CWE-502 | 12.2.1, 14.1.2, 14.1.3 |
| CWE-522 | 9.2.4, 9.3.1, 9.4.2, 9.4.3, 10.1.1, 13.1.2, 13.2.3, 14.1.1, 14.1.2, 14.1.3 |
| CWE-532 | 8.2.1, 8.2.3, 9.3.1, 9.4.3, 10.1.1, 12.2.1, 13.4.2, 13.2.3, 14.1.1, 14.1.2 |
| CWE-668 | 9.4.1, 12.1.4 |
| CWE-732 | 8.2.1, 8.2.3, 9.2.1, 9.4.1, 14.1.2 |
| CWE-755 | 14.2.1 |
| CWE-798 | 9.3.1, 9.4.3, 10.1.1, 10.1.2, 13.2.1, 13.2.2, 13.2.3, 14.1.3 |
| CWE-862 | 9.2.1, 9.2.2, 9.4.2, 9.4.5, 12.4.2, 13.1.2, 13.2.1, 13.2.3, 14.1.1, 14.1.2, 14.1.3, 14.2.6 |
| CWE-863 | 13.2.3, 14.1.1, 14.1.2 |

Appendix C

# List of top 50 CWE IDs and their name

**Table C.1:** IDs and name of the top 50 CWEs in 2020

| CWE ID | CWE Name |
|---|---|
| CWE-79 | Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') |
| CWE-269 | Improper Privilege Management |
| CWE-20 | Improper Input Validation |
| CWE-200 | Exposure of Sensitive Information to an Unauthorized Actor |
| CWE-787 | Out-of-bounds Write |
| CWE-125 | Out-of-bounds Read |
| CWE-119 | Improper Restriction of Operations within the Bounds of a Memory Buffer |
| CWE-89 | Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') |
| CWE-22 | Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') |
| CWE-416 | Use After Free |
| CWE-78 | Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') |
| CWE-352 | Cross-Site Request Forgery (CSRF) |
| CWE-287 | Improper Authentication |
| CWE-74 | Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection') |
| CWE-863 | Incorrect Authorization |
| CWE-120 | Buffer Copy without Checking Size of Input ('Classic Buffer Overflow') |
| CWE-400 | Uncontrolled Resource Consumption |
| CWE-276 | Incorrect Default Permissions |
| CWE-434 | Unrestricted Upload of File with Dangerous Type |
| CWE-522 | Insufficiently Protected Credentials |
| CWE-862 | Missing Authorization |
| CWE-190 | Integer Overflow or Wraparound |

**Table C.1:** IDs and name of the top 50 CWEs in 2020

| **CWE ID** | **CWE Name** |
|---|---|
| CWE-502 | Deserialization of Untrusted Data |
| CWE-476 | NULL Pointer Dereference |
| CWE-798 | Use of Hard-coded Credentials |
| CWE-732 | Incorrect Permission Assignment for Critical Resource |
| CWE-295 | Improper Certificate Validation |
| CWE-94 | Improper Control of Generation of Code ('Code Injection') |
| CWE-918 | Server-Side Request Forgery (SSRF) |
| CWE-306 | Missing Authentication for Critical Function |
| CWE-611 | Improper Restriction of XML External Entity Reference |
| CWE-601 | URL Redirection to Untrusted Site ('Open Redirect') |
| CWE-362 | Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') |
| CWE-77 | Improper Neutralization of Special Elements used in a Command ('Command Injection') |
| CWE-427 | Uncontrolled Search Path Element |
| CWE-319 | Cleartext Transmission of Sensitive Information |
| CWE-917 | Improper Neutralization of Special Elements used in an Expression Language Statement ('Expression Language Injection') |
| CWE-327 | Use of a Broken or Risky Cryptographic Algorithm |
| CWE-59 | Improper Link Resolution Before File Access ('Link Following') |
| CWE-532 | Insertion of Sensitive Information into Log File |
| CWE-312 | Cleartext Storage of Sensitive Information |
| CWE-326 | Inadequate Encryption Strength |
| CWE-347 | Improper Verification of Cryptographic Signature |
| CWE-843 | Access of Resource Using Incompatible Type ('Type Confusion') |
| CWE-426 | Untrusted Search Path |
| CWE-668 | Exposure of Resource to Wrong Sphere |
| CWE-613 | Insufficient Session Expiration |
| CWE-755 | Improper Handling of Exceptional Conditions |
| CWE-444 | Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling') |
| CWE-311 | Missing Encryption of Sensitive Data |

Printed by Tryckeriet i E-huset, Lund 2021

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2021-820
http://www.eit.lth.se