DEGREE PROJECT IN TECHNOLOGY,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2019

Weaknesses and risks of the Consumer Internet of Things

FABIENNE REITZ

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Weaknesses and risks of the Consumer Internet of Things

FABIENNE REITZ

Bachelor in Computer Science
Date: June 7, 2019
Supervisor: Robert Lagerström
Examiner: Örjan Ekberg
School of Electrical Engineering and Computer Science
Swedish title: Svagheter och risker inom Consumer Internet of Things

Abstract

The Consumer Internet of Things (CIoT) is a term to describe everyday items connected to the internet. The number of CIoT devices is growing rapidly and with it comes a number of security problems. One way to tackle these security issues is by learning from mistakes and to be aware of the risks at hand at both production and consumer level.

This report examines vulnerabilities from the years 2008-2018 in the National Vulnerability Database (NVD). With the Common Vulnerability Scoring System (CVSS) and the Common Weakness Enumeration (CWE) the following questions are answered: Which are the most common types of vulnerabilities in CIoT products, what risks do they pose and is there evidence of a connection between type of product and type of vulnerability?

The study found that the most common weaknesses were CWE-119, CWE-200 CWE-20 and CWE-264. However, the vulnerabilities of type CWE-119 turned out to be highly concentrated to Apple products and do not reflect the overall trends. The before mentioned weaknesses pose risks to users' confidentiality, integrity and the availability of the software (CIA). The CWEs with the greatest risk of exploitation were CWE-264 with the highest percentage of complete impact on the CIA attributes, and CWE-119 with lower percentage of complete impact but with far more occurrences. The study found no conclusive answer whether there is a connection between products and weaknesses, but an indication of a relation between certain CWEs and the company Apple. Further intensive and recurring studies should be conducted in the field.

Sammanfattning

Consumer Internet of Things (CIoT) är ett uttryck för vardagliga produkter med anslutning till internet. Antalet CIoT enheter ökar fort vilket har medfört ett antal olika säkerhetsutmaningar. Ett sätt att handskas med sådana säkerhetsproblem är att lära sig från tidigare misstag och att vara medveten om de involverade riskerna på både produktions- och konsumentnivå.

Denna rapport undersöker sårbarheter från åren 2008-2018 i National Vulnerability Database (NVD). Med hjälp av Common Vulnerability Scoring System (CVSS) och Common Weakness Enumeration (CWE) har följande frågor besvarats: Vilka är de vanligaste typerna av sårbarheter i CIoT, vilka risker medför dessa och finns det belägg för ett samband mellan typ av produkt och typ av sårbarhet?

Studien fann att de vanligaste svagheterna var CWE-119, CWE-200, CWE-20 och CWE-264. Sårbarheterna av typ CWE-119 visade sig dock vara ovanligt koncentrerade i Apple produkter och representerade inte fördelnigen i i allmänhet. De ovan nämnda svagheter utgör risker för användarnas konfidentialitet, integritet och mjukvarans tillgänglighet (CIA). CWE:na som utgör störst risk är CWE-264 som med högst sannolikhet visade total inverkan på CIA aspekterna och CWE-119 som trots lägre sannolikhet för total inverkan, var den oftast uppträdande. Studien fann inget definitivt svar för huruvida det existerar ett samband mellan produkter och svagheter, däremot en antydan om ett samband mellan särskilda svagheter och företaget Apple. Det behövs utförligare och periodiska studier på området i helhet.

Contents

Chapter 1

Introduction

The Internet of Things (IoT) is a broad term used to describe products which are connected to the Internet [1]. This could be anything from home lightening controlled by an app, to smart traffic management for entire cities. Some sources say that the number of IoT devices will increase by one billion in the coming year[1], whereas others claim the number will be as high as 4 billion[2]. Whether these assumptions are true or not, it is certain that the IoT has grown rapidly the last decade and does not show any tendencies to slow down [2].

Meanwhile, the fast growth of IoT has not come without consequences, shows a study conducted by Conti et al. [3]. The article's authors observe several security issues and concerns present in many IoT products. The challenges are, according to them, that the devices are wide spread and carry sensible data, which makes them harder to secure and principally important to protect respectively. There have even been some prominent cases of security breaches in IoT products, such as the infamous Mirai Bot[3], which launched an attack by gaining access to thousands of IoT devices and used them to attack popular websites' servers, causing the sites to crash.

We learn by this example that IoT devices are not only vulnerable to being attacked for the data they carry but also for other purposes. An article on CBC talks about the ability to hack baby nests to get a view inside the house, or hacking smart alarm systems to get access to the house through the front door[4]. This kind of data and access breaches through IoT products is dangerous for its users. Taking control of IoT products for other purposes might not endanger the user himself, but as in the example of the Mirai Bot, this can lead to huge

[1] https://iot-analytics.com/
[2] https://www.statista.com/
[3] https://www.theguardian.com/
[4] https://www.cbc.ca/

attacks on other, bigger networks.

This raises the question of how to prevent future security breaches. One way to tackle this problem is to be aware of common weaknesses and to better prepare the devices. As a product can have many vulnerabilities, not all might be discovered nor does the existence of a vulnerability imply that it will be exploited. However, being aware of the vulnerabilities in a product might help both the user and the producer to calculate and minimise the risks of an attack. This report will take a closer look at the Consumer Internet of Things (CIoT), which is a subclass that divides IoT devices into consumer and industrial IoT. CIoT products are items used by a single person or a household, such as the home lightening mentioned before.

The National Vulnerability Database (NVD)[5] is a government repository which provides publicly provided vulnerabilities for web pages, computers, mobile phones and IoT devices. NVD categorises and analyses these vulnerabilities with the Common Vulnerability Scoring System (CVSS)[6]. CVSS, which is a state of the art system that describes a vulnerability by its corresponding risks. NVD further makes use of the Common Weakness Enumeration (CWE)[7] to specify the type of vulnerability.

This study will use the NVD to extract information about the most common security breaches in CIoT-devices specifically.

1.1 Purpose

The aim of this study is to determine common weaknesses in CIoT devices, and compare similarities and differences in these. It is important to know the weaknesses and risks of a product and to be prepared for the consequences these might cause. The results of this study can help create awareness for all parties included, such as consumers, producers, programmers et cetera.

[5] https://nvd.nist.gov/
[6] https://www.first.org/
[7] https://cwe.mitre.org/

1.2 Research Question

The research questions will be:

- Which are the most common weaknesses in CIoT products?
- What risks do the most common weaknesses pose?
- Are type of IoT product and type of weakness connected?

1.3 Scope

The term Internet of Things, more precisely defined in section 2.1, is very broad and may include any number of digital devices. IoT is commonly divided into Consumer IoT (CIoT) and Industrial IoT (IIoT). This report will only take into account the CIoT devices. CIoT can further be divided into the strict sense of the term IoT, which includes modern phones (such that can connect to the internet) and computers or laptops. Since these are more established and more expensive products than the general CIoT device they usually do not encounter the same security issues as other CIoT devices [4]. Hence, this report will exclude products such as phones and computers.

1.4 Outline

This paper is divided into six chapters. The first being an introduction to the topic and a definition of the report's purpose and scope. The second chapter deals with definitions of terms and systems used throughout the report and presents studies related to the field. In the third chapter the approach and techniques used to get results are explained and motivated. In chapter 4 the answers to all three of the research questions are given in one sub section each. The following chapter 5 discusses the methods, results, limitations and suggestions for the future and the last chapter summarises the results in a few sentences.

Chapter 2

Background

This chapter will introduce common definitions used in the remainder of the report, it will explain important systems and specifications used to answer the research questions and it will disclose what related studies have found on the report's topic.

2.1 Internet of Things

International Telecommunication Union [1] defines an IoT device as:

> With regard to the Internet of things, this [a thing] is an object of the physical world (physical things) or the information world (virtual things), which is capable of being identified and integrated into communication networks.

By this definition, any thing that can be connected to the internet is an IoT device. This report will from now on use the term accordingly, with respect to the limitations given in section 1.3.

2.2 Confidentiality, Integrity and Availability

The aspects of confidentiality, integrity and availability, commonly known as the CIA triad, are generally used to describe and precise IT-security. A paper by Hansen, Jensen, and Rost [5] defines the three features as follows:

> Confidentiality addresses the need for secrecy, i.e. the non-disclosure of certain information to certain entities within the IT system in

> consideration. Integrity expresses the need for reliability and non-repudiation, regarding a given piece of information, i.e. the need for processing unmodified, authentic, and correct data. As an important subset of such data, identity-related information is needed in authentic way to perform access control operations. Availability represents the need of data to be accessible, comprehensible, and processable in a timely fashion.

In this report the terms *CIA* and *security triad* will be used interchangeably with the definition above.

2.3 Vulnerabilities

NVD[1] defines a vulnerability as follows:

> A weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability.

This definition exemplifies that an error in the software, if exploited, will be a violation of one of the forms mentioned above (CIA). Important to note is, that no evidence of such a breach must exist, the mere threat of one happening is sufficient. A bug, on the other hand, which by definition also is a software error, would generally not harm CIA. A weakness is a type specific and reoccurring error, which leads to a vulnerability in a product. A vulnerability on the other hand only represents one error in one device.

2.3.1 Common Vulnerability Scoring System

Vulnerabilities' severity is demonstrated on a numerical scale called the Common Vulnerability Scoring System (CVSS)[2]. A CVSS score is a number from 0 to 10, where 10 is a critical vulnerability and 0 is a low (or nonexistent) vulnerability. The score is calculated taking into consideration six aspects of the vulnerability. The Access Vector, Access Complexity and Authentication being the first three and an impact metric for confidentiality, integrity and availability each.

[1] https://nvd.nist.gov/
[2] https://www.first.org/

Access Vector

The Access Vector describes from where the product is vulnerable. The possible values for the Access Vector are *Local, Adjacent Network* and *Network*. If a product is only locally vulnerable an attacker must gain access to the system either physically or with a local account. Naturally, *Local* is the least severe value for the Access Vector. For the value *Adjacent Network* an attacker must have access to the broadcast or collision domain of the desired system. The value *Network* means the product can be attacked from afar, without immediate connection to the network.

Access Complexity

The Access Complexity describes how difficult it would be for an attacker to exploit the product, given access to the system. The possible values for the Access Complexity are *High, Medium* and *Low*. If a product lacks complex verification or privilege security it scores *Low*. In that case an attacker will not need any specialised access or skills to exploit the product. The values *Medium* and *High* require somewhat more and full specialisation access conditions, respectively. Hence, the higher the Access Complexity the lower the vulnerability score.

Authentication

The Authentication metric describes how often an attacker must authenticate himself before being able to exploit the system. It is notable that this metric does not take into consideration how elaborate the authentication step is, but only the number of times it would be required. The possible values for Authentication are *Multiple, Single* and *None*, where Multiple denotes any system where two or more authentications are required. Single and None simply stand for one and no authentication respectively. The fewer authentications are demanded the higher the vulnerability score is.

Confidentiality, Integrity and Availability Impact

These three separate metrics measure the potential impact of the loss of confidentiality, integrity and availability each. The metrics have the same possible values, being *None, Partial* and *Complete*. The values are rather self explanatory, for example partial impact on availability would mean reduced performance, while a complete impact on availability would mean no access to the system at all.

2.4 Common Weakness Enumeration

The Common Weakness Enumeration(CWE) specification provides codes for universal types of software weaknesses. Every CWE code has a unique description linked to it, describing in what part the error has occurred and what problems it might cause. The CWEs are tree structured, meaning that some CWEs are categories which further consist of lower level CWEs. The lower level a CWE is located at, the more specific the CWE generally is. In Appendix A a table with all CWEs and their descriptions which are mentioned in this report are displayed.

2.5 Related Work

CWE, in cooperation with SANS, has conducted investigations into which are the 25 most common software errors [6], with much the same purpose as this report. In their published paper from 2010 they strive to educate both programmers, managers, consumers and teachers to ensure fewer vulnerabilities and higher awareness. The results show 25 different CWEs, with CWE-79 (Improper Neutralization of Input During Web Page Generation) and CWE-89 (Improper Neutralization of Special Elements used in an SQL Command) on top. In contrast to this report the paper does not focus on any specific products but an overall analysis.

A study by Williams et al. [7] researches vulnerabilities in CIoT with the help of Shodan API, a search engine for connected devices and Nessus, a vulnerability scanning software. Their results show that the most common weaknesses are:

- devices running outdated versions of MiniUPnP, a network discovery and communication protocol
- problems related to the Simple Network Management Protocol(SNMP)

These vulnerabilities can lead to breaches of all three CIA aspects and even contribute to allowing an attacker to get access to other devices in the network.

Another study, by Välja, Korman, and Lagerström [8], examines CVSS scores and CWEs from vulnerabilities found in the NVD database. In contrast to this report the study focuses on embedded systems in power networks. The results of the study show that the most common weakness are CWE-20, followed by CWE-310 and tied for third place were CWE-119 and CWE-399, the descriptions of which can be seen in Table 2.1.

CWE	Description
CWE-20	Improper input validation
CWE-119	Improper restriction of operations within the bounds of a memory buffer
CWE-310	Cryptographic Issues
CWE-399	Resource Management Errors

Table 2.1: Common CWEs found in software Study and their descriptions

Chapter 3

Methods

This section will describe how the research questions were answered. As the work was divided into parts, this chapter follows a particular order: first relevant products and vendors were identified, second the database was browsed and filtered with the before acquired products and vendors and third the final data was analysed and arranged for illustration.

3.1 Identifying products and vendors

This first part is based on studying literature. To identify relevant products of CIoT this report used a study on consumer attitudes and knowledge about IoT devices [9]. The report found that the most common IoT product owned by its respondents was the connected TV, also known as smart TV. Further findings disclosed wearable technology (i.e. health trackers, smart watches or head phones) and home control systems as the most popular categories of IoT products owned [9].

Identifying relevant vendors and companies in CIoT was done by examining a study on vulnerabilities by the IoT Security Foundation [10]. The aim of the study was not further relevant to this report, however it identified a large set of companies active on the IoT market and listed their respective products. For the purpose of this report, the selection of companies was too broad and redundant. Therefore a selection was made by cross referencing the most commonly owned IoT products, mentioned above, with the table of companies and their products. This procedure led to a list of the following Companies and products:

Company	Product
Amazon	TV
Apple	TV, Wearables
Asus	Router, Home control System
Google	Home control System
Motorola	Baby monitor, Router
Philips	Wearables, Router
Roku	Router
Samsung	TV
Sony	Camera
Xiaomi	Router

Table 3.1: Companies and their products used in the report

3.2 Browsing and filtering data

The second part was mainly practical.The NVD's database was chosen for this project because of its shown accuracy of vulnerability ratings [11]. NVD's database is accessible, free of charge, for anyone. The data for each year (2002-2019) can be downloaded as JSON or XML zip files. In this report the data from year 2008 until 2018 was collected to ensure the inclusion of both old and new IoT products. Older vulnerabilities are not taken into consideration as the term and idea of IoT was not common before 2008. Further the data for 2019 was still changing frequently and was not taken into account either. For the filtering of the database python was used, with which vulnerabilities with vendors contained in the list above were extracted. These vulnerabilities were further filtered by eliminating those where the product was either a mobile phone or a computer.

From the given data set some information was deemed applicable, others not. An example of a detail that was not taken into consideration, is the *version* of a specific product. This choice was made because the loss of this information would not affect the results of the research question, since it only is interested in the type of product, not any specific product version. Further this report only compares the CVSS version 2.0 scores, excluding its successor the CVSS version 3.0. This decision is based on the report by Johnson et al. [11], which examines the CVSS version 2.0's correctness and trustworthiness. The report, finding the scoring system sufficiently accurate, chose to only consider

version 2 as this was the most commonly used version in their test data.

3.3 Analysing and illustrating data

For the last part the relevant data for each vulnerability were: vendor, product name, CVSS 2.0 score, CWE and the CIA Impact. The corresponding information for each vulnerability was extracted to an Excel sheet to easier handle the data and create diagrams and tables of it.

The answers to the research questions are presented differently, depending on the question being quantitative or qualitative. The first question (Which are the most common vulnerabilities in CIoT products?) is clearly quantitative and is therefore described with tables and diagrams made in Excel. The information needed was all unique CWEs and the matching amount of times each of them occurred.

The second research question (What risks do the most common vulnerabilities pose?) was answered by eliminating all but the four most common CWEs from question one. For each of the four CWEs and each of the three CIA triad's aspects the amount of entries with *Complete, Partial* and *None* were counted and lastly divided by the total amount of entities of the according CWE. The results of these calculations were then visualised with a diagram. To fully answer the research question additional information from the CWE homepage was included about the CWEs in question.

To answer the third research question (Are type of IoT product and type of vulnerability connected?) the vulnerabilities were sorted by the given categories and the amount of each CWE was counted for each of the categories. CWEs with no record for the given product category were omitted to facilitate the view of the diagrams.

Chapter 4

Results

This section will present the results for the three research questions one at a time. Further analysis of the results can be found in the next chapter labelled Discussion. The results are built upon 238 vulnerabilities. The distribution of amount of vulnerabilities per company can be seen in Table 4.1 below.

Company	Amount of Vulnerabilities
Amazon	7
Apple	144
Asus	40
Google	1
Motorola	2
Philips	34
Roku	1
Samsung	1
Sony	5
Xiaomi	3

Table 4.1: Companies and the amount of unique vulnerabilities found for each of them

4.1 Which are the most common weaknesses in CIoT products?

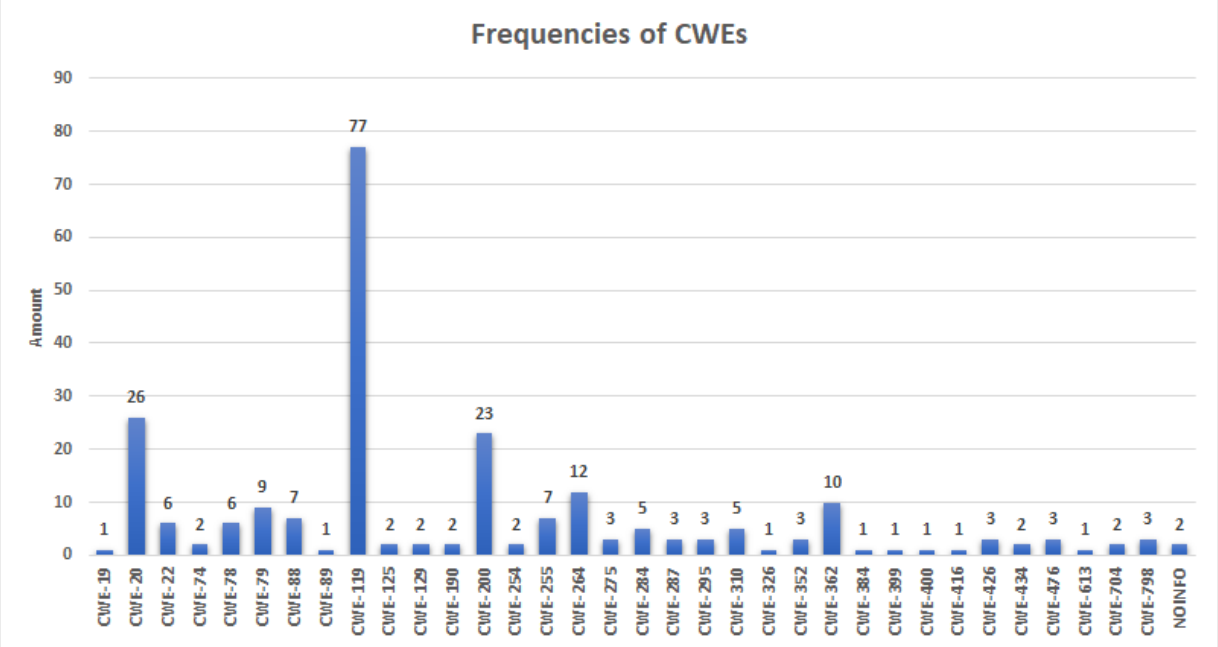


Figure 4.1: Frequencies of CWEs

As can be seen in Figure 4.1 above, the most common weakness is CWE-119 with 77 reported vulnerabilities. On second place is CWE-20 with 26 vulnerabilities, the third most common weakness is CWE-200 with 23 vulnerabilities and the fourth most common weakness is CWE-264 with 12 vulnerabilities. Descriptions for all CWEs involved can be found in Appendix A.

From Table 4.2, we can tell the distribution of the most common CWEs and companies. Unsurprisingly, Apple stands for most of the weaknesses for each of the most common CWEs. For CWE-119 73 out of the 76 vulnerabilities were found in Apple products, which corresponds to 95%. Because this number is higher than any of the corresponding numbers of the other top CWEs (which were 73% for CWE-20, 52% for CWE-200 and 66% for CWE-264), this report will carefully discuss the reliability of CWE-119 as the most common weakness in the discussion chapter.

	CWE-20	CWE-119	CWE-200	CWE-264
Amazon	0	0	0	0
Apple	19	73	12	8
Asus	4	2	5	0
Google	0	0	1	0
Motorola	0	0	0	0
Philips	3	1	3	4
Roku	0	0	1	0
Samsung	0	0	0	0
Sony	0	1	0	0
Xiaomi	0	0	1	0

Table 4.2: The frequencies of the top CWEs for each company

4.2 What risks do the most common weaknesses pose?

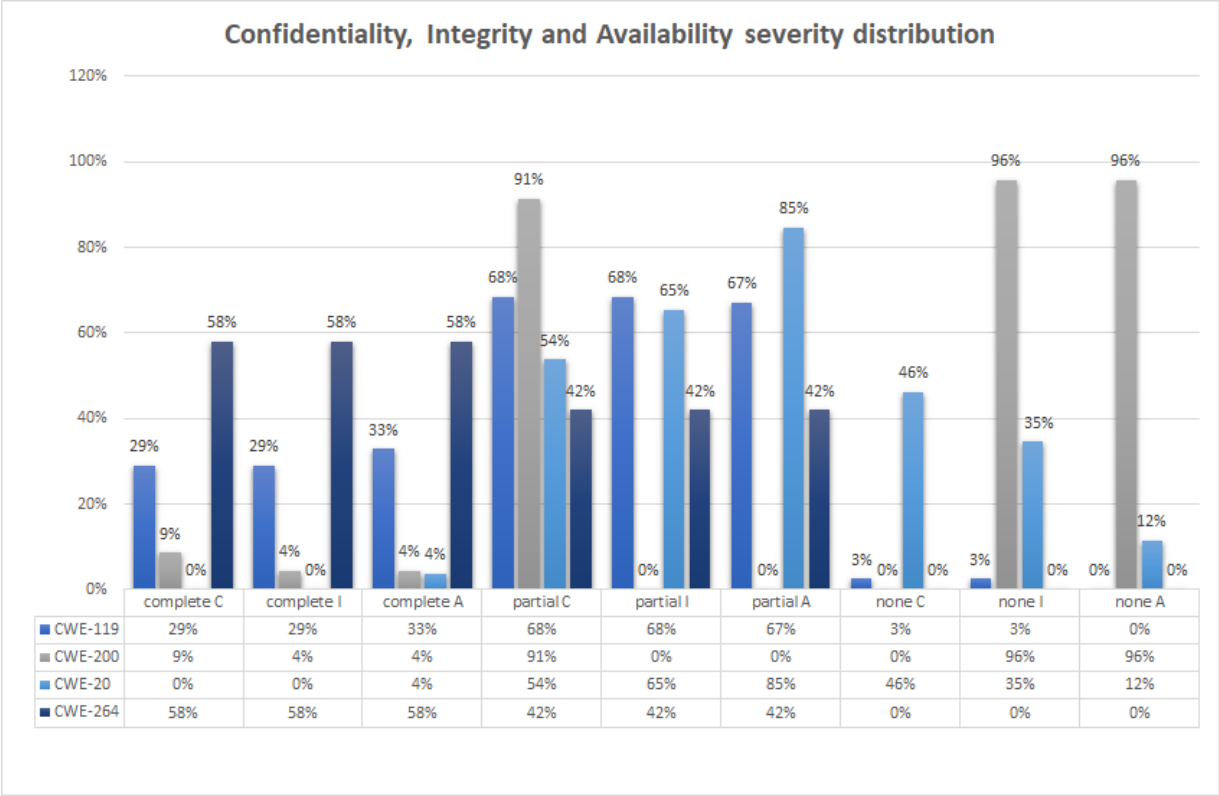
The weaknesses which occurred the most often were:

- CWE-119: Improper restriction of operations within the bounds of a memory buffer
- CWE-20: Improper input validation
- CWE-200: Information exposure
- CWE-264: Permissions, Privileges, and Access Controls

4.2.1 CWE-119

CWE-119 leads to the possibility of reading or writing to a location in the memory that is outside the boundary of the buffer. This may lead to consequences such as execution of arbitrary code, erasing of other memory data, overwriting of existing security code or even direct access to sensitive information. Depending on the exact occurrence this means that an CWE-119 vulnerability can lead to breaches within all three CIA categories.

As can be seen in Figure 4.2 CWE-119 vulnerabilities' confidentiality and integrity impact was complete 29% of the time, partial 68% of the time and not affected the remaining 3%. Whereas the availability was affected completely 33%, partial 67% of the time and never not affected.



	complete C	complete I	complete A	partial C	partial I	partial A	none C	none I	none A
CWE-119	29%	29%	33%	68%	68%	67%	3%	3%	0%
CWE-200	9%	4%	4%	91%	0%	0%	0%	96%	96%
CWE-20	0%	0%	4%	54%	65%	85%	46%	35%	12%
CWE-264	58%	58%	58%	42%	42%	42%	0%	0%	0%

Figure 4.2: Distribution of severity of the CIA triad in CWEs 119, 20, 200 and 264

4.2.2 CWE-20

CWE-20 occurs when input is not checked properly, which can lead to an altered execution of the code, or crashing the system. As an inconsistent input could let an attacker get access to all of the code this vulnerability may also cause breaches within all three of the security aspects of the CIA.

Figure 4.2 illustrates how there was no data confirming complete impact of either confidentiality or integrity. In 4% of the time there was a complete impact on availability and 85% of times partial impact. Furthermore we can see that 54% and 65% of times there was partial confidential impact respectively partial impact of integrity.

4.2.3 CWE-200

CWE-200 may lead to the software exposing sensitive information by accident or revealing information that would enable an attack on the software. Because the severity of the exposure heavily depends on the type of information that is exposed, it varies greatly.

This vulnerability can obviously lead to a breach of confidentiality if the exposed information is of a sensitive nature. However, exposing part of the program which enables attacks may cause any kind of breach of confidentiality, integrity or availability.

In Figure 4.2 we can observe that the risk of CWE-200 having complete impact on confidentiality is 9% and 91% partial impact. On the other hand the risk for integrity and availability breaches was complete in only 4% of the cases each and none the remaining times.

4.2.4 CWE-264

CWE-264 is a CWE category consisting of several other CWEs. CWEs in this category deal with improper management of ownership and access control, or privilege issues. These weaknesses are tightly connected to security mechanisms and as there is a variety of different vulnerabilities that can take form within this category all three CIA features could be exploited.

The 12 occurrences of CWE-264 had complete impact on confidentiality, integrity and availability 7 out of the 12 times each (58%) and partial impact the remaining 5 times each (43%). These numbers can also be observed in Figure 4.2.

4.3 Are type of IoT product and type of weakness connected?

The product categories used were TV, wearables, router and home control system. The remaining products used in this paper, presented in Table 3.1 had too few of its kind to draw conclusions. For the below product categories some show equal distribution and frequencies of CWEs as the original distribution in Figure 4.1, others show some deviation which may be interpreted as dependent on the product. The topic will be thoroughly discussed in the next Chapter.

4.3.1 Tvs

The TV products category was the biggest with 88 vulnerabilities. We can observe, by comparing Figure 4.3 and Figure 4.1, that this category follows a rather similar distribution as the original one. Unsurprisingly, we can see that the majority of CWEs were of type 119 and 20. All instances of CWE-88 are present in TVs, whereas CWE-200 is slightly underrepresented in contrast to the original curve.

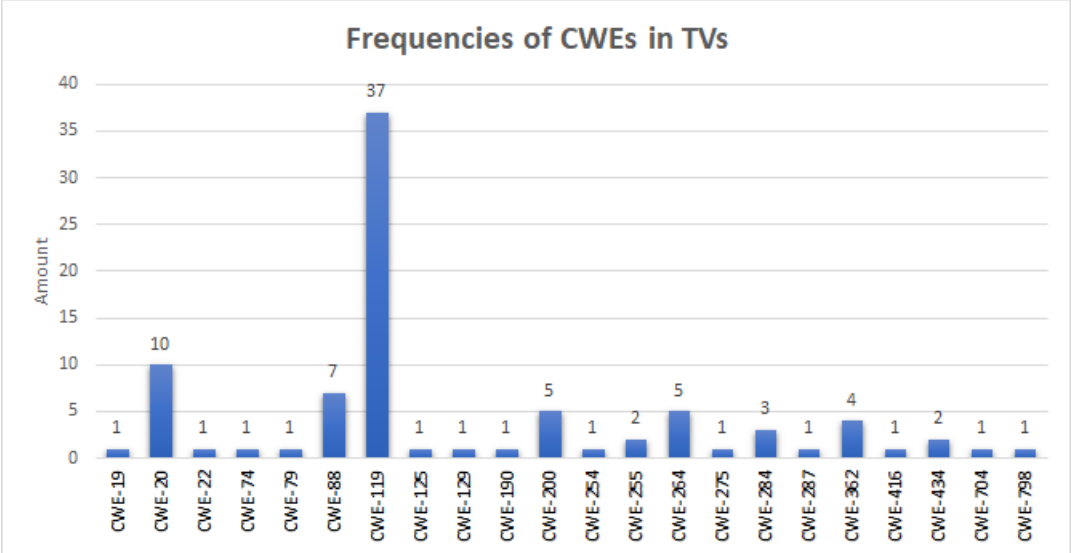


Figure 4.3: Frequencies of CWEs in TVs

4.3.2 Home Control Systems

Home Control System was the smallest product category with 15 instances recorded. In the diagram in Figure 4.4 one can observe that problems of type CWE-200 are the most common within Home Control Systems. Out of the 21 occurrences of CWE-200, five are connected to this product type which point to a connection between the two. With regard to the small amount of data in this group the amount of the other CWEs does not with confidence resolve the question of a relation between home control systems and CWEs 22, 78, 79, 254 or 255.

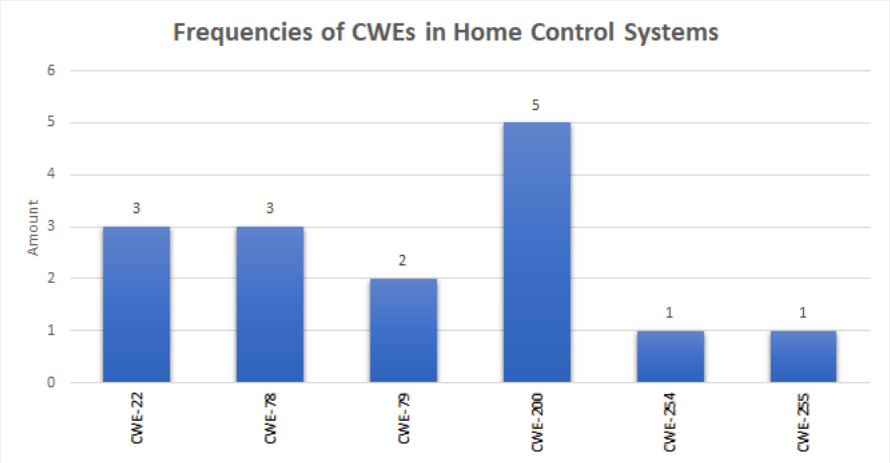


Figure 4.4: Frequencies of CWEs in Home Control Systems

4.3.3 Wearables

In the product category *Wearables* 62 vulnerabilities were found. Out of these, a large portion were of type CWE-119 and CWE-20. A slightly higher percentage of CWE-264 and CWE-362, in comparison to the original distribution, can be observed in the below diagram Figure 4.5.

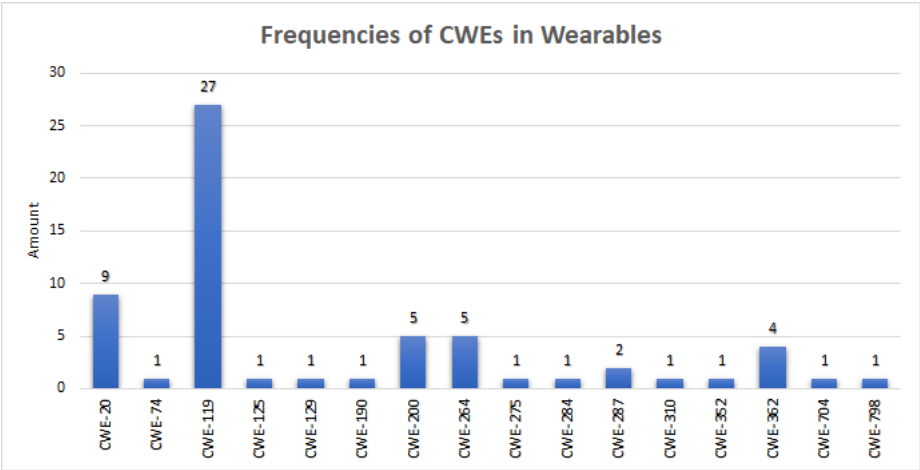


Figure 4.5: Frequencies of CWEs in Wearables

4.3.4 Routers

Out of the 51 instances of routers in the data, CWE-20 is the most common, although with small margin. The distribution, shown in Figure 4.6 is the least

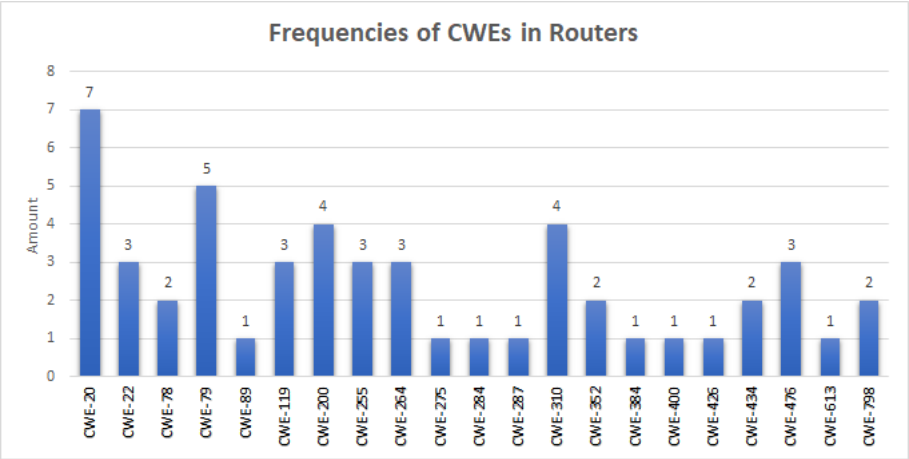


Figure 4.6: Frequencies of CWEs in Routers

similar to the original distribution and shows several common CWEs. After CWE-20 comes CWE-79 (Improper Neutralization of Input During Web Page Generation) with 5 occurrences. CWE-200 and CWE-310 tie for third most common CWE with 4 instances each. It is interesting to note that the occurrence of CWE-119 is unusually low in this example, compared to the above categories and the original distribution.

Chapter 5

Discussion

This chapter will discuss topics such as the results, related work, limitations of the study and future recommendations.

5.1 Data selection

The data, on which the results are based, rely on the 238 vulnerabilities that were found in the NVD using the methods presented in chapter 3. There may be vulnerabilities which are not reported to the NVD of the products used in the report and there may be CIoT products which were not found in the NVD, but could have been used in this report. CIoT, or IoT in general, is a newly introduced category of products and many objects that belong to this category may not be categorised as such, since they still have another purpose. For example, a router will most likely be described as a router rather than an IoT product, since that describes its purpose. Further, the number of IoT products, as mentioned in the introduction, grow extensively and in a lot of different product categories. This makes it incredibly hard to count or distinguish them. As for the reporting to NVD it is arguable whether cheap CIoT products have a high enough priority to be checked for vulnerabilities. Considering a product worth 10 USD, a company might not be willing to engage money and time in the testing and securing of that product.

5.2 Interpretation of results

The results show that the most common problem within CIoT devices is connected to handling and restricting memory buffers. CWE-119 was by far the

most frequent of CWEs, but seeing Table 4.2, one can doubt the result's reliability. If the vulnerabilities of Apple were discarded the remaining vulnerabilities would surely not indicate CWE-119 as the most common weakness. Since CWE-119 only has 4 occurrences from other companies than Apple, and the total number of vulnerabilities without Apple are 94 this would, by calculations in Table 4.2, make CWE-119 only tied for the third most common weakness. On the other hand, it is difficult to motivate a disregard of more than half of the data for this study, which is why Apple cannot we taken out of the equation.

However, when sorted by category, CWE-119 was clearly more common in TVs and wearables than in the other categories. One explanation might be that these two categories had the most data to back it up, and the fewer the number of items in a category, the more uncertain the results are. Another explanation could be that the majority of vulnerabilities reported in the categories TVs and wearables were produced by Apple, which again point to the fact that CWE-119 is very common in their products but not necessarily in others. Yet, in Figure 4.6, we see that CWE-119 is fairly common in the router category, which has no items belonging to Apple. Ultimately, this report cannot conclusively say whether CWE-119 is the most common CWE in CIoT products or the most common CWE in TVs and wearables, only that it is the most common CWE in the investigated Apple products.

Comparing the results of this report to the study by Välja et al. mentioned in section 2.5, we can find similarities. For one, CWE-119 is fairly common in their results too, implicating that the results in this report perhaps show contemporary trends. It is arguable whether the results resemble each other because they are taken from the same database, or because the products investigated might be of a similar nature and encounter the same issues. Earlier the impact of the company Apple on this report was discussed, comparing to the study on embedded systems none of Apple's products were used and no other products overlap with the ones used in this report. Interestingly the study came across much the same difficulties as this report, such as skewed distribution among the vendors and problems with acquiring enough data to confidently report results.

Assuming CWE-119 is generally the most common CWE in CIoT products, what conclusions can we draw? First off, this weakness can lead to both intended and unintended errors and breaches. This means that no harm must be intended but a mistake such as overwriting of data could happen to any user. Practically this means that a user's data or files could be overwritten and lost. These kinds of errors are less severe than if instead an attacker were to

discover the weakness and purposely tried to execute malicious code. In order to do this, an attacker would need advanced knowledge in programming, but could on the other hand pass without authentication or physical access. In this scenario a user's files, data, even the complete product could be corrupted and either destroyed or used against the user. Meanwhile these kinds of problems must be solved at the implementation phase, knowledge and awareness of it being a common weakness might help lower the number of occurring cases. Another solution would be to use programming languages or libraries which do not allow such errors, although these decisions seldom are up to the programmer and perhaps must be taken at a higher level.

Assuming CWE-119 is wrongly represented in this study, and if we were to disregard the Apple products, CWE-200 would be the most common weakness. This can be calculated by Table 4.2, where we see that CWE-200 has 11 occurrences in companies other than Apple whereas the other CWEs all have less than that. Vulnerabilities of the CWE-200 type unintentionally expose information of either sensitive nature or information which can lead to access to the system. Sensitive data might be exposed to any user, irrelevant of technical background. Practically this could mean that for example, an user's e-mail address were to be revealed to an unauthorised user, or in a less severe scenario some information would be revealed that the user does not understand or know anything about. Information unmasking parts of the system, are on the other hand likely to be dangerous if exposed to an attacker with knowledge of programming. Exposure of information unfortunately is not as easy to fix as CWE-119, it is not language specific and can occur in any program. To ensure that these mistakes do not happen as easily, precautionary steps must be taken in the architecture phase of a project. The people working in this phase should aim for separation of privilege. That way, there will be safe areas and areas which require close attention to privileges and programmers should be able to easily navigate through them.

It is interesting to compare the results of the study mentioned i section 2.5 by CWE/SANS. The study showed CWE-79 and CWE-89 to be the most common vulnerability types. This report in contrast, found CWE-79 to be tied for fifth most common and CWE-89 only appeared once. As CWE's and SANS's study was conducted on vulnerabilities of all kinds of products, whereas this report only focused on CIoT devices, the differences might be explained by that. Another reason for the diversity between the studies might be the time frames, which barely overlap. The CWE/SANS study was run on data up to 2010, while this study used data from 2008-2018. It is possible that CWEs have developed over time and new problems have become prominent, which

were unheard of more than 10 years ago.

The risks deriving from the vulnerabilities given in section 4.1 are shown in section 4.2. Since the results of the second research question heavily depend on the data for the first research question, we already know it is skewed towards CWE-119. It is important to note that the foundation for the numbers of CWE-200 and CWE-20 were only 23 and 26 data points respectively. CWE-264 had only 12 data points to rely on to answer this question. Ideally, these amounts should have been higher to give more accurate and reliable results. Taking this into consideration, it is still interesting to note that as much as 58% of these instances had complete impact on all three CIA principles. This number is much higher than the around 30% for CWE-119 and the even lower percentage of less than 10% for CWE-200 and CWE-20. We can observe that it is most common for CWE-200 to have impact on confidentiality, which was found partial 91% of times, while integrity and availability had no impact 96% of times each. Generally seen, CWE-119 seems to be the greatest threat, considering the amount of data points which lie behind the 29-33% of complete impacts. Nevertheless, CWE-264 must be taken into account, since its high percentage of complete impact make it a critical error whenever it does occur.

In section 4.3, the report tries to answer the question whether product and vulnerability are connected. The different numbers show results which can be interpreted to answer the question with both yes and no. In Figure 4.3 for example, we can see that CWE-119 is clearly over represented in comparison to the other CWEs. On the other hand, the distribution looks rather similar to the original CWE distribution in Figure 4.1. On contrast, the diagram in Figure 4.4 shows CWE-200 as the most common CWE but no clear connection to the distribution in Figure 4.1. One could also draw the conclusion that routers are usually affiliated with CWE-20 and home control systems with CWE-200, but more research is needed in the area. Nevertheless, from the results of this question and the table in section 4.1 the question arises whether the weaknesses are connected to a specific company rather than a product type.

It might be necessary to conduct more studies in the field, which could result in some interesting discoveries. If it were the case that weaknesses are connected to certain companies, the mitigation actions would perhaps look different to what will be done if they were affiliated with product types. For instance, Apple might want to improve overall programming and software architecture standards to revise common errors in their systems, rather than examining a single teams efforts in a product.

5.3 Limitations and future work

Because of time and resource limitations this study was constrained to the methods presented in chapter 3. Software such as Shodan API used in the study by Williams, mentioned in section 2.5 might be an easier alternative for checking larger amounts of data. It would be desirable to have a larger set of data for future studies since some parts of this report were impossible to conduct due to too few data points. The study by Williams did not encounter difficulties connected to skewed data either, which both this report and the study by Välja et al. did, therefore an approach combining the different methods might give another perspective. Another interesting topic would be to repeat this kind of investigation in set time periods and examine differences over time.

In the future it would be intriguing to compare studies of common weaknesses between CIoT and Industrial IoT products (IIoT). The differences in the risks the users are willing to take and what consequences these might lead to might differ between the groups and are interesting to investigate.

Chapter 6

Conclusions

The question of which are the most common weaknesses was answered with CWE-119, CWE-200, CWE-20 and CWE-264. However, further analysis showed that the amount of CWE-119 vulnerabilities discovered was highly common in Apple products, but less so in others. Therefore CWE-119 cannot confidently be claimed to be the most common vulnerability type in all CIoT products.

The question of what risks the most common weaknesses pose was answered with extracts from the CWE homepage and distributions of the CIA triad's severity scores. This shows that CWE-264 is the most dangerous as its percentage of complete impact on the CIA triad was the highest, on the other hand CWE-119 occurred a lot more often and still had a moderately high percentage of complete impact.

Lastly, the third research question could not conclusively decide whether there is a connection between products and weaknesses. The results show somewhat different distributions, although this might sooner relate weaknesses with companies rather than with product types.

Clearly, additional studies are needed in the field concerning differences over time, between CIoT and IIoT and with different methodological approaches.

Bibliography

[1] International Telecommunication Union. “Y.2060 Overview of the Internet of things”. In: *SERIES Y: Global information infrastructure, internet protocol aspects and next-generation networks* (2017), pp. 2–3.

[2] Thierer and Castillo. “Projecting the growth and economic impact of the Internet of Things”. In: *Journal of Economic Perspectives* (2015), pp. 4–6.

[3] Mauro Conti et al. “Internet of Things security and forensics: Challenges and opportunities”. In: *Future Generation Computer Systems* 78 (2018), pp. 544–546.

[4] Beale and Berris. “Hacking the Internet of Things: Vulnerabilities, dangers, and legal responses”. In: *Digitalization and the law* 16 (2018), p. 167.

[5] Hansen, Jensen, and Rost. “Protection Goals for Privacy Engineering”. In: 2015, pp. 159–166.

[6] Martin et al. “The 2010 CWE/SANS Top 25 Most Dangerous Software Errors”. In: ed. by Steve Christey. The MITRE Corporation. 2010, pp. 4–5.

[7] Williams et al. “Identifying vulnerabilities of consumer Internet of Things (IoT) devices: A scalable approach”. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). 2017, pp. 179–181.

[8] Välja, Korman, and Lagerström. “A Study on Software Vulnerabilities and Weaknesses of Embedded Systems in Power Networks”. In: 2017, pp. 47–52.

[9] Sruoginis and Shane. “The Internet of Things”. In: (2016), pp. 4, 10.

[10] IoT Security Foundation. “Understanding the Contemporary Use of Vulnerability Disclosure in Consumer Internet of Things Product Companies”. In: (2018), pp. 18–30.

[11] Pontus Johnson et al. “Can the Common Vulnerability Scoring System be Trusted? A Bayesian Analysis”. In: *IEEE Transactions on Dependable and Secure Computing* (2016), pp. 1–1.

Appendix A

CWE descriptions

CWE	Description
CWE-19	Data Processing Errors *
CWE-20	Improper input validation
CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path traversal')
CWE-74	Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection')
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
CWE-88	Argument Injection or Modification
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
CWE-119	Improper restriction of operations within the bounds of a memory buffer
CWE-125	Out-of-bounds Read
CWE-129	Improper Validation of Array Index
CWE-190	Integer Overflow or Wraparound
CWE-200	Information exposure
CWE-254	7PK - Security Features *
CWE-255	Credentials Management *
CWE-264	Permissions, Privileges, and Access Controls *
CWE-275	Permission Issues *
CWE-284	Improper Access Control

CWE-287	Improper Authentication
CWE-295	Improper Certificate Validation
CWE-310	Cryptographic Issues *
CWE-326	Inadequate Encryption Strength
CWE-352	Cross-Site Request Forgery (CSRF)
CWE-362	Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')
CWE-384	Session Fixation
CWE-399	Resource Management Errors *
CWE-400	Uncontrolled Resource Consumption
CWE-416	Use After Free
CWE-426	Untrusted Search Path
CWE-434	Unrestricted Upload of File with Dangerous Type
CWE-476	NULL Pointer Dereference
CWE-613	Insufficient Session Expiration
CWE-704	Incorrect Type Conversion or Cast
CWE-798	Use of Hard-coded Credentials

Table A.1: CWEs mentioned in the report and their descriptions found on cwe.mitre.org, descriptions marked with * are CWE categories

TRITA-EECS-EX-2019:390