On the assessment of Denial of Service vulnerabilities affecting smart home systems

Vulnerability scanning using OpenVAS

Bachelor's Thesis, 15 credits

Sebastian Andersson
Oliver Josefsson

Degree: Bachelor of Science, 180 credits
Field of Study: Computer Science
Program: Computer Systems Developer
Spring 2019
Final Seminar Date: 2019-05-31
Supervisor: Romina Spalazzese & Joseph Bugeja
Examiner: Helena Holmström Olsson

Abstract
IoT is an abbreviation of the term Internet of Things. The term describes everyday items such as light bulbs that are connected to the Internet. IoT is a field that is growing very quickly with some researchers and industry leaders predicting that there will be up to 200 billion connected IoT devices in the world by 2020. Many IoT devices are developed by smaller companies looking to capitalize on a specific need in the market. Because of this, the companies may favor launching a product as fast as possible which could mean that the devices may have not been adequately tested for different vulnerabilities.

The IoT and Smart Home market is currently experiencing rapid growth and all signs point towards that continuing in the future. This thesis focuses on testing for vulnerabilities to Denial of Service attacks in common-off-the-shelf IoT devices that can be found in a smart home environment. The purpose of this thesis is to create more knowledge about the vulnerabilities that can be found in Internet connected devices that are used daily.

This thesis includes experiments using OpenVAS, which is a vulnerability scanner developed by Greenbone Security used to test for vulnerabilities to Denial of Service attacks in IoT devices. The devices that are tested are Sony PlayStation 4, IKEA Trådfri Smart Lighting, Google Chromecast (First Generation), Apple TV (Third Generation) and D-Link DCS-930LB Wi-Fi IP-Camera. The firmware/software of all the devices are updated as of April of 2019.

The results of the conducted experiments show that all the tested devices besides Chromecast and IKEA Trådfri had vulnerabilities to Denial of Service attacks. PlayStation 4 was the device with the highest amount of vulnerabilities (9) and the vulnerability with highest possible severity (10.0). The effects of a Denial of Service attack range from an annoyance, when a gaming console is unavailable, to a security risk when an IP camera can be temporarily disabled.

Keywords
Internet of Things (IoT); IoT Vulnerability; Vulnerability Testing; OpenVAS; Availability; Denial of Service (DoS); Smart Home; Smart Lighting; IP Camera.

Acknowledgements
We would like to thank Romina Spalazzese and Joseph Bugeja for their invaluable help during the process of writing this thesis.

Table of Contents

1 Introduction

As time passes, an increasing number of things are getting connected to the Internet; ranging from toothbrushes to light bulbs to thermostats [1]. These "things" are collectively called "IoT" devices, IoT is an abbreviation for Internet of Things which describes everyday objects that are connected to the Internet [2][3].
The use of IoT devices and services in our surroundings are increasing and surveys are estimating that there will be up to 200 billion connected IoT devices around the world and a projected market share of $457 billion US dollars by 2020 [4-6].

Several scholars and industry professionals [8-16] states that security is one of the big challenges in IoT. The continuous increase in IoT devices requires extensive studies and testing to make sure that the devices are safe for consumers to use. The data that IoT devices store and send can be personal and private, and thus requires adequate security.
Banafa [9] claims that some IoT vendors are releasing products as fast as possible to beat their competition by releasing their innovative solutions before anyone else can, which can indicate that security is not always a main priority.

The focus of this thesis is testing IoT devices to examine how vulnerable they are to Denial of Service attacks and how severe the vulnerabilities are. Naik et al. [23] explains that one of the most common type of security attack is Denial of Service, which is a type of attack that aims to bring down a network, by sending a large amount of traffic to a device or a server. By performing a Denial of Service attack an attacker can cause unavailability in a device, the consequences of unavailability can range from annoyance if the device is for example a gaming console to security risks if the device is a camera that is used for home security.

The devices are scanned by using OpenVAS, which is a vulnerability scanning tool developed by Greenbone Security that can be used to test different Internet protocols. In this thesis, OpenVAS is used in combination with Kali Linux. Kali Linux is an operating system with several pre-installed ethical hacking and security tools. OpenVAS and Kali Linux has been used in several articles, two of them being Wang et al. [33] and Gordin et al. [34] where different security tools were compared and tested.

The thesis is structured in the following way, firstly an introduction to the different terms and the scenario that is used throughout the thesis. Secondly, research motivation, goal and question that the thesis aims to answer. Followed by that, a background to present previous studies regarding the subject of IoT and Smart Home security, Vulnerability testing and Denial of Service attacks. After the background, the method chapter introduces this thesis' experiment using OpenVAS and the subsequent Results chapter presents the results from the experiment. Following the Results chapter is the Analysis and Discussion where the results are examined, and the consequences are discussed. In the Conclusion and Future Work chapter, final thoughts are presented along with possible additions to the experiment conducted in this thesis.

1.1 IoT Security Challenges

One of the challenges that exists with IoT devices is security [7-10]. The reason behind this is the fact that the devices are always connected to the internet, which also means that the risks tied to Internet exposed systems are introduced; e.g., hackers. There have been stories involving hackers gaining control of Internet-connected cars and gaining access to IP cameras which enables the hacker to spy on users in the safety of their own homes. Many IoT devices suffer from security vulnerabilities, which could be partly attributed to the manufacturer's rush to deliver innovative devices without prioritizing proper security and vulnerability testing [8][9][11].

Some of the challenges that exist with IoT devices according to academics and industry professionals [10-16] are:

- The lack of processing power and memory that are available in IoT devices. Security approaches that rely on security algorithms would then be constrained and the devices would be unable to perform complex encryption.
- Applying device updates, including security patches as not all devices support over-the-air updates which would require the user to update the devices manually. Older devices might not receive patch updates as they are no longer supported by the manufacturer.
- Ensure high availability, as people rely more on IoT in their day-to-day life, developers need to make sure that the data in the IoT devices are always available.
- Detecting and managing vulnerabilities, using logs to identify if a system has been compromised or vulnerability testing to figure out flaws in the security of a system.

The focus of this thesis is on Denial of Service vulnerabilities in common-off-the-shelf IoT devices found in a smart home, as the increased reliance on IoT devices in our daily lives will require data that is accessible whenever we need it. If the data is unavailable when a person needs it, the consequences can range from annoyance that your PlayStation is not working, to fatal if your Internet-connected diabetes device stops working [17].

1.2 Smart Home

The smart home is according to Bugeja et al. [18] an environment, where heterogeneous electronic devices and appliances are networked together to provide smart services in a ubiquitous manner to individuals.
Jose et al. [80] explains that the concept of a smart homes expectations changes over time. As the Internet was introduced, the modern smart home includes combinations of Ubiquitous Computing devices, also known as IoT devices.
According to Ramlee et al. [81] and Vacher et al. [82] elderly and disabled can also benefit from the assistance that a smart home can provide by controlling connected devices through a computer, smartphone, tablet or by speech.

Rehman et al. [19] explains that the advantages of the Smart Home is the fact that the user's home becomes simplified. The simplification is because a user can control the lights, doors or cameras remotely, e.g., at work or while on vacation. The main concern that is mentioned by Rehman et al. [19] and by Siboni et al. [20] is security and the lack of security standards. Zhang et al. [21] and Rehman et al. [19] explains that the data that is being sent over the home network may be personal, private and sensitive and that some Smart Home systems and IoT devices may lack adequate security, as a hacker does not need to physically be inside the house to control the system.

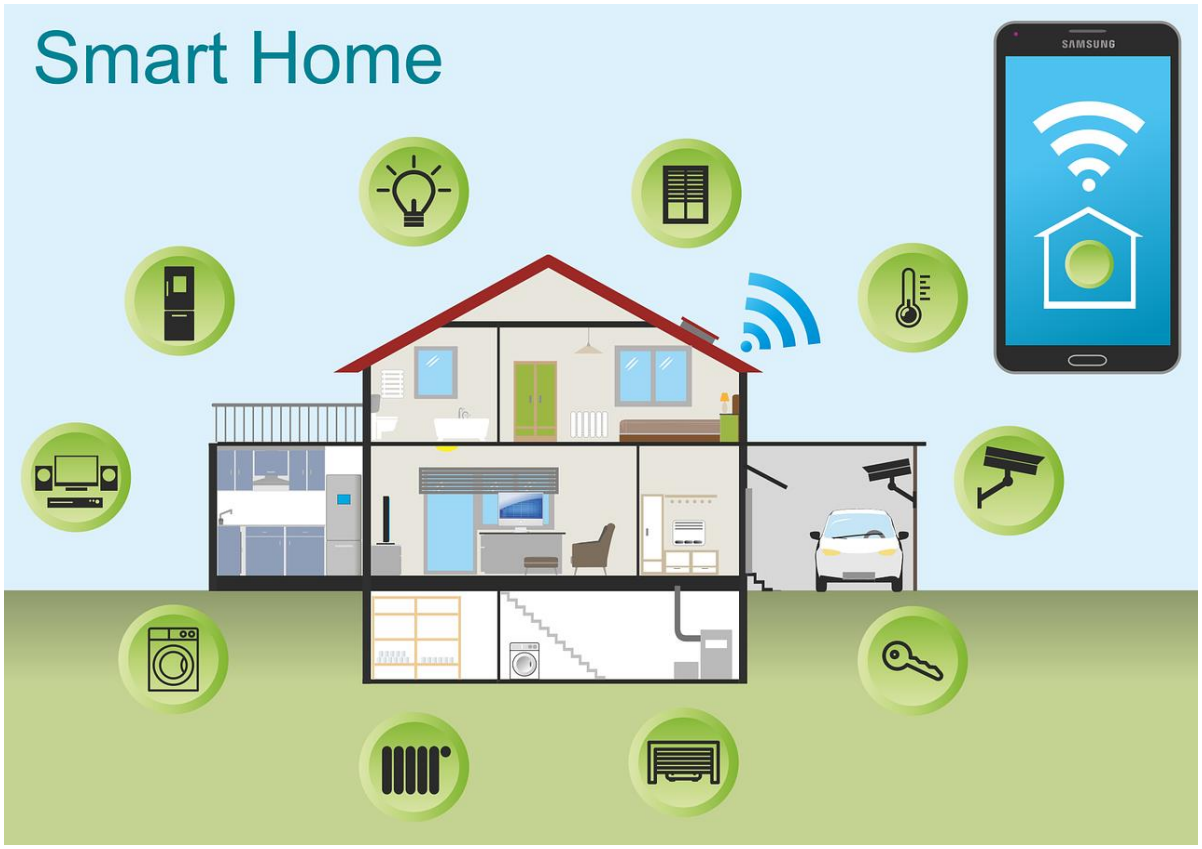


Figure 1. Smart Home with several connected devices, e.g., smart light bulbs, smart blinds, smart thermostat and an IP camera [22].

1.3 Security Models

Two of the existing information security models are the Confidentiality, Integrity and Availability (CIA) triad and the STRIDE model. STRIDE is an acronym for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of privileges.

The CIA triad [24] is a model that presents the three cardinal security goals of an information system, in this thesis specifically IoT devices.
Confidentiality aims to prevent unauthorized access to data. Integrity aims to prevent unauthorized modification of data. Availability aims to makes sure that constant authorized access to data is available.

The STRIDE model is another model that categorizes different threats. The STRIDE model is connected to the CIA triad, as most of the elements in STRIDE are a security risk to either confidentiality, integrity or availability [25].

The focus of this thesis is on the “Availability” element of the CIA Triad, which correlates to the “Denial of Service” category from the STRIDE model.

In short, availability means that authorized people should have reliable access to the data. To prevent availability, a hacker may execute a Denial of Service (DoS) attack to make the device unavailable. According to Minoli et al. [24], to ensure availability, it's important to keep systems updated to the latest version, and to maintain the hardware by replacing faulty devices [26].

1.4 Denial of Service Attacks

An article written by Kolias et al. [27] focuses on the vulnerabilities in many IoT devices and how they can be vulnerable to different types of attacks. The attack mentioned in the article is called a Denial of Service attack. Singh et al. [10] mentions that there is not always adequate protection against these types of attacks in IoT devices.

Denial of Service attack is a single-source attack that exploits vulnerabilities in the services that the host offers. According to Rizvi et al. [28] and Hussain et al. [29] a malicious user floods the host with requests, which makes the service unavailable for the intended users. Cloudflare [30] also explains that the primary focus of a DoS attack is to oversaturate the capacity of a targeted machine, resulting in denial-of-service to additional requests. An example of a Denial of Service attack according to industry professionals [31] is a flood attack, which floods a system with traffic, either by sending packets or requests. R. Zerbari et al. [32] mentions SYN Flood Attack as an example of Denial of Service attack, which exploits a vulnerability in the TCP connection process:

1. The attacker sends a request to connect to a server's port.
2. The server starts a connection process, but the attacker does not complete the connection, instead uses this vulnerability to lock up the port of the server.

By sending a "flood" of these requests, overwhelming the server by occupying the ports thus making other connection requests unavailable.

1.5 Vulnerability Testing

Wang et al. [33] explains vulnerability testing as the process of using one computer to look for weaknesses in another computer or network. Vulnerability scans enables security experts to find weaknesses in a system and then fix those weaknesses, while an attacker may use that weakness to attack the system instead.
Vulnerability scanning tools generate a detailed report of vulnerabilities and how severe they are which enables a security expert to prioritize which weaknesses to address first. Some examples of vulnerability scanning tools that [33] and Gordin et al. [34] mention is OpenVAS, Nessus and NMAP.

Parizi et al. [35] mention that vulnerability testing of modern software applications is an undervalued topic and that these tests are vital during the development of an application. According to Sedaghat et al. [83] experts believe that the best way to improve security is to analyze previous mistakes and prevent the same mistakes from reoccuring. Parizi et al. [35] and Sedaghat et al. [83] both mention that security assessment cannot be done all at once but should instead be done throughout the development process.

According to Sedaghat et al. [83] the process of integrating security into the development process starts with informing development groups about existing and new vulnerabilities. Sedaghat et al. [83] also mentions that vulnerability scanning tools can be used after an application has been developed, to present the vulnerabilities to security professionals in a more interpretable way.

1.6 Smart Home Scenario - The Living Room

The scenario that this thesis uses is the living room and the difference devices that can be found in the living room.

Most living rooms have a TV, either a smart TV or a TV connected to a smart device that enables streaming from the Internet, e.g., a Chromecast or an Apple TV. The Chromecast was selected because 55 million have been sold around the world [36]. A gaming console, the reason the PlayStation 4 was chosen is because it is the most sold gaming console of the current generation [37]. Smart lighting, because of its practicality and its growing popularity. An IP camera, as it is an easy way to increase security in a home.

The effects of a Denial of Service attack to each of the devices are analyzed in the Analysis chapter.

PlayStation 4
The PlayStation 4 is a game console developed by Sony. The console connects to a TV through the HDMI port. The PlayStation 4 connects to the local network by Wi-Fi or by ethernet cable.

Smart Lighting
Smart lighting is lighting that can be controlled by using an application on a Smartphone. Some examples of smart lighting are Philips Hue and IKEA Trådfri [38].

Chromecast (First Generation)
Google Chromecast is a device that connects to the HDMI port of a TV. The user then connects the Chromecast to the local network through Wi-Fi. If the user and the Chromecast is connected to the same network, the user can then stream content from applications like YouTube on their phone, tablet or computer directly to the TV [39].

Apple TV (Third Generation)
Like the Chromecast, Apple TV is a device that connects to the HDMI port of a TV. The user then connects the device to the local network through Wi-Fi or by ethernet cable. Once the device is connected to the Internet, the user can use the remote that comes with the device to scroll through a list of applications, for example Netflix [40].

IP Camera
An IP camera is a digital video camera that can be used to send and receive data through a computer network. The images captured by the camera can be shown on a smartphone or a computer [11][41][42].

Figure 2. The Living Room, showing the different devices present in the scenario.

1.7 Research Objectives

This section provides the motivation behind this thesis, along with the goals and the research question this thesis aims to answer.

1.7.1 Research Motivation

The motivation behind the research of this thesis is to create more awareness of the security vulnerabilities that exist in everyday IoT devices. Because of the big increase in IoT devices that is projected by industry professionals [4-6] in the coming years the issue can become a bigger problem as the number of devices increases as well as the diversity of the devices and their suppliers. Besides this, with more devices being automated and connected to the Internet, ranging from devices for recreational use like a gaming console to health-related devices such as inhalers or automated insulin pens. Vulnerability testing needs to be done as the consequences of an unavailability in some devices could according to Sândescu et al. [43] and Econsultancy [17] in extreme cases even lead to devastating outcomes, such as loss of life.

1.7.2 Research Goal

The goal of this thesis was to show the amount of vulnerabilities to Denial of Service attacks that exists in everyday IoT devices and how severe they are. The experiment is partly based on an experiment that was conducted by Tekeoðlu et al. [42], where OpenVAS was used to find vulnerabilities in IP cameras. According to Wang et al. [33] and Gordin et al. [34] which compared several vulnerability tools, they both concluded that OpenVAS is a well-designed scanning tool that is also free to use.

1.7.3 Research Question

The question that this thesis aims to answer is the following:

> *To what extent are IoT devices in a smart home environment vulnerable to Denial of Service attacks?*

The question is relevant because the amount of IoT devices are increasing according to industry professionals [4-6], and since security is not always the highest priority during the development of these devices [8][9][11]. Other than this, there are several articles that include both IoT and Denial of Service, but not specifically focusing on Denial of Service attacks to common IoT devices that can be found in a traditional home.

2 Background

The literature review is used to get an understanding of which studies and articles have been done in the field. There have been a lot of articles regarding IoT and Smart Home security, Vulnerability scanning and Denial of Service individually, but with a gap in articles specifically regarding Denial of Service attacks to IoT and Smart Home devices. The articles that were found ranged from explanations of IoT applications by Sethi et al. [45], to investigating the vulnerabilities in Smart Home cameras by Bugeja et al. and Tekeoðlu et al. [11][84][42]. Parizi et al. [35] and Sedaghat et al. [83] both mention the importance of integrating vulnerability scanning into the software development process. Other articles ranged from comparing different vulnerability scanners by Wang et al. [33] and Gordin et al. [34], to articles inventing new types of security protocols and solutions by Liu et al. [7] and Singh et al. [10].

The chapter is structured in the following way, firstly articles of studies regarding IoT and Smart Home security will be introduced, followed by articles about Vulnerability testing and lastly Denial of Service attacks.

2.1 IoT and Smart Home Security

Miraz et al. [85] states that the term Internet of Things (IoT) includes electronic devices of varying sizes and capabilities, the thing these devices has in common is that they all are connected to the Internet. Sândescu et al. [43] describes the Internet of Things as an architecture where assets and services are exchanged. Sândescu et al. [43] also explains that the development process of IoT devices mostly focus on functionality and that security can be an afterthought. Banafa [9] claims that IoT vendors might try to release their innovative devices before their competition, therefore not prioritizing security testing.

Khan et al. [86] mentions that IoT devices that are not updated to the latest software version can be vulnerable to security and privacy risks. Which will be further elaborated on in subsequent chapters.

Both Jose et al. [80] and Bugeja et al. [84] explains that the smart home is a developing term, and that some of the earliest attempts at making a home “smart” dates back to the 1970’s. Since then the term “smart home” has developed, and with the introduction of Internet, Internet connected (IoT) devices such as lighting systems, cameras, and other home appliances can be remotely controlled through a variety of communication protocols.
This is also mentioned by Ramle et al, [81] and Vacher et al. [82], which states that smart devices enable elderly and disabled to have more control of their home and the connected devices with the help of a smartphone, computer or a tablet.

But with the introduction of Internet, security and privacy challenges are also introduced to the smart home. Dorri et al. [87] explains that today’s IoT devices that generate, process and exchange data are attractive targets for cyber-attacks. Many of the devices are lightweight and low energy, which means that the devices must dedicate the computing power to the core application, and the traditional security methods can then be too computer power intensive to execute.
The data that is sent between the connected devices can be sensitive, private or personal, Rehman et al. [19], Zhang et al. [21] and Dorri et al. [87] explains that IoT devices and smart home systems which may lack appropriate security, gives the

hacker a chance to take control of the connected devices remotely and extract the data or to attack the devices to cause unavailability.

2.2 Vulnerability Testing

Vulnerability testing is according to Wang et al. [33] the process of using one computer or tool to look for weaknesses in another device, computer or network.
About Vulnerability Scanners and Vulnerability Testing, Vernotte [88] mentions, that as the Internet increases in size and becomes more complex, it becomes increasingly hard to secure all the transactions that occur every millisecond around the world.
Doupé et al. [89] states that the biggest Internet and software security issue is web application vulnerabilities, which can also be applied to IoT devices that also uses web servers and applications to communicate.
Vernotte [88] claims that vulnerability scanners suffer from a sensible number of false positives and false negatives and that a more structured approach to testing is required.
Having a structured approach combined with an agile approach to vulnerability testing throughout the development of software is also echoed by Parizi et al. [35] and Sedaghat et al. [83] who mention that developers cannot afford to believe that the initial security requirements are perfect or impenetrable. Parizi et al. [35] claims that as the development process continues, the number of components in the system generally increase, along with the amount of possible vulnerabilities that exist in those components.
Parizi et al. [35] mentions that vulnerability testing is an undervalued topic, and by integrating continuous vulnerability testing in the development of software applications, allowing the developers according to learn from previous mistakes and prevent the same mistakes from happening again.
Parizi et al. [35] and Sedaghat et al. [83] both mention that security assessment cannot be done all at once but should instead be done throughout the development process. Parizi et al. [35] claims that a lot of software engineers lack the proper knowledge of security vulnerabilities, which along with the big increase in Internet-connected devices, e.g., IoT devices could lead to an increasing amount of security flaws in everyday devices, such as Smart TV, IP cameras or baby monitors.
An article by Bugeja et al. [11][84] claims that hackers found 700 connected baby monitors streaming babies asleep in their cribs and 73,000 IP Cameras that streamed their surveillance footage live on the Internet. Based on this, the assumption could be made that IP cameras are vulnerable to different types of attacks.
Tekeoðlu et al. [42] mentions a vulnerability scanning tool called OpenVAS, with which the authors conducted an experiment to find vulnerabilities in IP-cameras. The results from the experiment showed that the camera was vulnerable to Denial of Service attacks.

After the development of an application is concluded, vulnerability scanning tools can according to Sedaghat et al. [83] be used to present the vulnerabilities that exists in the application to security experts in a digestible manner.
An article by Wang et al. [33] compares several tools, specifically OpenVAS, Nessus and NMAP for the sake of finding a vulnerability scanner that could be used for the security courses at Columbia State University. The conclusion that Wang et al. [33] reached was that OpenVAS was the best option out of the three, because the program was well designed and free. Gordin et al. [34] also lists three vulnerability scanning tools which according to the authors offers the best results. The three tools mentioned are OpenVAS, Nessus and Metasploitable, and the conclusions the

authors reached was even with OpenVAS being free, the results from the scans were well organized and detailed.

2.3 Denial of Service

In an article written by Soliman et al. [90] they cite CN-CERT cyber threats and tendencies 2017 [91] which claims that the amount of Denial of Service attacks is increasing rapidly, which Nagesh et al. [92] also states. Kominos et al. [93], Soliman et al. [90] and Nagesh et al. [92] explains Denial of Service as an attack that renders a device or a system unavailable to its legitimate users which is conducted by a single computer (Denial of Service) or multiple computers (Distributed Denial of Service).
Nagesh et al. [92] claims that a Denial of Service attack can be launched against both web servers and networks and that the impact can vary from minor inconveniences to serious consequences. In the scope of this thesis, a minor inconvenience being the unavailability of a gaming console or smart lighting to huge security risks with an IP camera used for home surveillance being unavailable.
Sândescu et al. [43] and Econsultancy [17] claims that in extreme cases unavailability in some devices could cause physical harm or even loss of life if the affected device is used for medical purposes.

3 Method

The method chapter introduces the different research methods that were used to gather information about Denial of Service vulnerabilities in IoT devices by reading scientific articles pertaining to the specific subject. The reason behind doing this is to collect information about what types of studies has been done in the field and how the studies were conducted.
The purpose of this chapter is to explain the different scientific methods that are used in this thesis and the reasoning behind using them. Firstly, explaining the literature review method and secondly the experiment method.
The aim of this thesis is to show how many vulnerabilities that exist in different devices that can be found in a traditional living room as shown in Figure 2. To achieve this several articles were studied to gather information about prior studies and results in the field, and to understand which tools that were used in these articles.

3.1 Literature Review Description

The IEEE [46] and ACM [47] databases were used to find peer-reviewed research articles with information pertaining to the search phrases below. After several research articles were found, the articles' content was analyzed to confirm that the information was relevant to the subject. This process was then repeated to narrow down the number of relevant articles pertaining to the subject.
The results were chosen based on the relevance to the subject, with a focus on the date and the amount of citations of the article or paper.

3.1.1 Search phrases

Below are the search phrases that were used while searching for information in the IEEE [46] and ACM [47] databases.

"IoT"
The results that are chosen about the topic of IoT are used to introduce IoT; the definition and functionalities of these devices.

"Vulnerability testing"
To gather information about what vulnerability testing is, what scientific studies has been done and which tools has been used in these studies.

"Vulnerability Testing + IoT"
Since the thesis is focused on vulnerability testing IoT devices that can be found in a Smart Home scenario, the focus was on finding articles that included similar types of devices and tests.

"Denial of Service"
Since the thesis is focused on Denial of Service attacks, the focus was on finding articles that included similar attacks.

“Smart Home”
Because of the Smart Home scenario as shown in Figure 2, the focus was solely on finding relevant articles to Smart Home and which devices that can be included in such a setting.

“PlayStation 4”
Sony’s PlayStation 4 is part of the scenario that is introduced as shown in Figure 2.

“Smart Lighting”
Smart Lighting, specifically IKEA Trådfri is part of the scenario that is introduced in Figure 2.

“Chromecast” & “Apple TV”
Media players, specifically Google Chromecast and Apple TV are part of the scenario that is introduced in Figure 2.

“IP Camera”
The IP camera D-Link DCS-930L is part of the scenario that is introduced.

3.2 Experiment Description

The information retrieved from scientific methods are generally either qualitative or quantitative. Qualitative data is according to Oates [44, pp. 266] images, words and audio gathered from interviews. Quantitative data is, also according to Oates [44, p 245] data based on numbers generated by experiments which then are analyzed using tables or graphs. Oates [44, pp. 127] explains an experiment as a strategy that investigates cause and effect relationships, aiming to prove or disprove a link between a factor and the outcome. An experiment is then designed to prove or disprove a hypothesis and then the results are observed. Based on this an experiment is the selected method, which generates quantitative data, as the research question requires measurable data.

3.3 Method Discussion

The reason behind doing a literature review is according to Oates [44, pp. 72] to make sure that the topic is worthwhile, the research is not just repeating the work done by someone else and that the researcher has created some new knowledge.

As previously mentioned, the data collected by experiment is considered quantitative according to Oates [44, p 245], which is fitting to the research question this thesis aims to answer:

To what extent are IoT devices in a smart home environment vulnerable to Denial of Service attacks?

To examine the extent or severity of a vulnerability the data, the data must be measurable.

The reason behind conducting an experiment with OpenVAS is firstly because it has been used in multiple prior studies, some examples of studies are Wang et al. [33], Gordin et al. [34] and Tekeoðlu et al. [42]. Secondly, because OpenVAS shows the result in a measurable form, based on a severity scale of 0.0-10.0, thus allowing the results to be presented in a clear and intuitive way.

For this thesis it was decided to conduct an experiment on different types of IoT devices that can be found in a traditional home. To limit the number of devices to scan a scenario is created, which limits the devices to only focus on ones that generally can be found in a living room with a collection of entertainment devices.
The reason behind doing this experiment is to increase the knowledge about the vulnerabilities that could exist in the IoT devices present in someone's home.
The decision was made to conduct an experiment on common IoT devices that generally could be found in homes, thus being more relevant to a "regular" person.

Once it was decided which devices to test, we then wanted to find tools freely available on the Internet. Most tools that were found were behind a subscription or a one-time cost. The tool that was selected was OpenVAS, which is a vulnerability assessment software available to install on the Kali Linux operating system. Both Kali Linux and OpenVAS are free to download which enables people to reproduce the same experiment that is conducted in this thesis without having to pay a fee or a subscription.

To collect data regarding vulnerabilities in IoT devices we firstly had to find IoT devices that we could test. The devices were selected based on what we considered a realistic living room scenario of a traditional home and what was available for us to use. Traditional home in this case a home that has a living room, which could include some or all these devices.

All the devices are shown in the scenario introduced in Figure 2.
PlayStation 4, as it is the most sold gaming console of the current generation [37]. A Chromecast or an Apple TV to be able to stream content from the Internet (Netflix,HBO etc.). A smart lighting solution, in this case the IKEA Trådfri and an IP camera, in this case the D-Link DCS930LB1.

4 Experiment

CN-CERT cyber threats and tendencies 2017 [91] along with Nagesh et al. [92] claims that the amount of Denial of Service attacks is increasing rapidly.
Which in correlation with the increase in IoT devices could become a larger issue and therefore something that needs to be studied further.

Based on the information that was gathered from the literature review, different tools for vulnerability testing were mentioned. Both Wang et al. [33] and Gordin et al. [34] have written articles comparing different vulnerability scanners, the scanners introduced and compared in these articles are OpenVAS, Nessus, Nmap and Metasploitable.

Another example where OpenVAS is used, is a study done by Tekeoðlu et al. [42] which used OpenVAS to scan an IP Camera for vulnerabilities, similar to what this thesis aims to achieve.
Based on the information gathered from these articles, along with the fact that OpenVAS is free, available for anyone to use and suitable for the intended experiment it was decided that this was the tool that would be used to conduct the experiment in this thesis.

The devices that are scanned in this thesis is a PlayStation 4, IKEA Trådfri Smart Lighting, Google Chromecast, Apple TV and a D-Link IP Camera. The software versions, local IP and the different "roles" are as shown in Table 1. The scans used for this experiment are using a standard installation configuration compiled by Offensive Security [48].

The setup for the experiment is based on the study done by Liang et al. [49] and consists of a computer that acts as a host for Kali Linux, which is running on VirtualBox. OpenVAS, installed on Kali Linux acting as the "attacker", which uses a router on a local network to simulate attacks on the different devices, listed as victims in Table 1 and as shown in Figure 3. The scans were done on a local network.

Device	**Software/Firmware Version**	**IP Address**	**Role**
PC	Windows 10	192.168.0.10	Server
Kali Linux (VM)	2019,1	192.168.0.10	Attacker
Compal Router	5.510.5.11	192.168.0.1	Local Network Gateway
PlayStation 4	6.50	192.168.0.13	Victim
IKEA Trådfri	1.4.15	192.168.0.4	Victim
Chromecast	1.32.14470	192.168.0.9	Victim
Apple TV	7.2.2	192.168.0.7	Victim
D-Link IP Camera	2.17.03	192.168.0.29	Victim

Table 1. A list of the devices, which software or firmware version on the devices, the IP address of the device and the role in the scenario, e.g., attacker or victim.

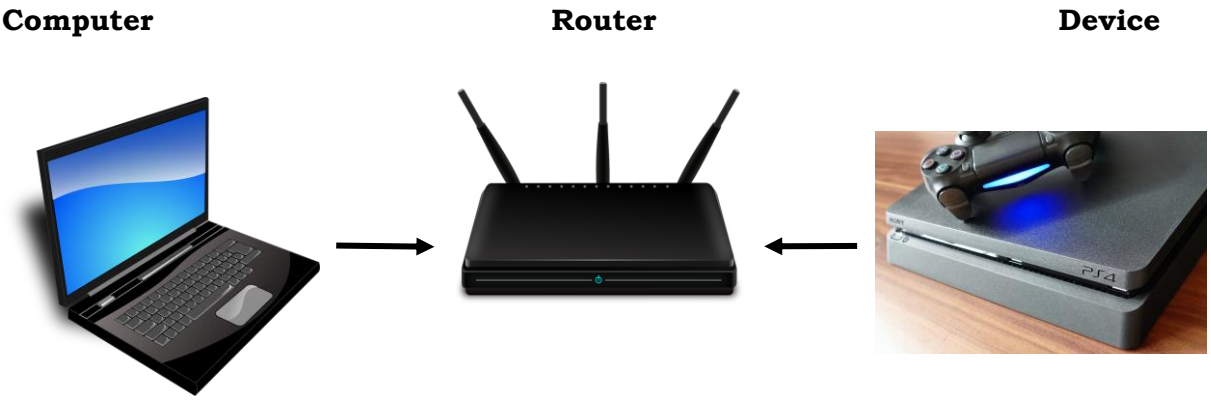


Figure 3. A computer with Kali Linux and OpenVAS installed, scanning for vulnerabilities in a PlayStation 4 over a local network.

The initial idea of the experiment is to find general vulnerabilities in IoT devices, without focusing on a specific type. To find vulnerabilities, five OpenVAS scans are completed on each of the devices that were introduced in the scenario. It became evident after a multitude of scans across several devices that the most common vulnerability was availability attacks, specifically Denial of Service (DoS) attacks. The effects of a Denial of Service attack to each of the devices will be presented in the Results chapter, along with possible solutions for the problems based on information in OpenVAS. The experiment conducted in this thesis can be applied to most IoT devices on a local network.

The hypothesis of this experiment is that consumer IoT devices are vulnerable to Denial of Service attacks, which is based on the information gathered from the literature review.

4.1 Tools

The software tools used to conduct the experiment are the following:

VirtualBox
VirtualBox is a program developed by Oracle that allows the user to install additional operating systems as applications on a single computer. The operating systems are called “Guest Operating Systems (OS)” and enables the use of operating systems such as Linux or Mac OS on a computer with Microsoft Windows installed as the main operating system or vice versa. The program was released in January 2007 and the latest version release at the time of writing this thesis was released on the 28th of January 2019, which is the version that is used in this experiment [50].

Kali Linux
Kali Linux is an operating system that allows users to conduct security and vulnerability testing. Kali comes with hundreds of tools that focuses on Penetration Testing, Security research, Computer Forensics and Reverse Engineering. The program was released in 2013 and the latest release was version 2019.1 which was released the 18th of February 2019, this is the version that is used in this experiment. The Kali Linux project is being supported by Offensive Security, which is according to their website, Industry-Leading Online Penetration Testing Training and Certification for Information Security Professionals. Kali Linux is free and available for anyone to use [51-53].

OpenVAS
OpenVAS is a free full-featured vulnerability scanner that allows the user to scan for vulnerabilities in IP-addresses and ports. OpenVAS is developed by Greenbone Networks, the program was released in 2009 and the latest version GVM-10 was released 5th of April 2019 [54].

4.2 Settings

The settings used in this experiment are based on the guide from Kali Linux. The guide shows a step by step process of how to set up and use OpenVAS [55].

OpenVAS setup
The port list used in the scans is the *All TCP and Nmap 5.51 top 100 UDP* [56] and it searches through all the 65535 TCP (Transmission Control Protocol) ports, together with the top 100 UDP (User Datagram Protocol) ports [57].

The Scan Config used for each scan is the *Full and very deep ultimate* [58] which, based on port list, searches through each port with several Network Vulnerability Tests (NVT). Each NVT belongs to a "family", e.g., "Brute force attacks", "Buffer overflow" and "Denial of Service", and in total, *Full and very deep ultimate* contains 62 families with 49653 NVTs combined.

4.3 Observations

After the experiments are concluded the results are collected and observed. The research question; To what extent are IoT devices in a smart home environment vulnerable to Denial of Service attacks? requires measurable data. The tool that is used to conduct the experiment and to gather the data (OpenVAS) includes a severity scale from 0.0 - 10.0, 0.0 being the least severe and 10.0 being the most severe. Severity measures how critical a vulnerability is, based on different variables, these are explained in the following chapter.

5 Results

The chapter is structured in the following way, firstly, the different terms are explained followed by the results of the scans of each device.
The vulnerability scans are done multiple times for each device, this to make sure that the results are consistent and not just a random occurrence.
After a scan is completed, it will list the vulnerabilities (if any were found), with basic information of what type of vulnerability, solution type, severity, quality of detection, host and on which location the vulnerability is found.

Vulnerability:
Displays the name of the vulnerability found on the specific IP address (host). If the user clicks on the vulnerability, it will give further information about the vulnerability and which solutions that exists if there are any available.

Solution type:
This column shows an icon if there are any solutions available, there is five different icons based on different solutions. The icons will give a hint about the solution types.

 **Workaround:** There might be a configuration or solution to bypass the vulnerability.
 **Mitigation:** A configuration, using external devices or access controls might be available that may reduce the risk of the vulnerability, although they might not be authored by the affected product.
 **Vendor-Fix:** An official fix that will fully solve the vulnerability is available.
 **None-Available:** There is no fix currently available.
 **WillNotFix:** There is no fix available, and there will not be one either. The product might have been abandoned or deprecated [59].

Severity:
Shows a value on a 0.0 to 10.0 scale, based on how critical the vulnerability is, calculated by OpenVAS CVSS calculator [60][61].

The calculator considers the factors of Access vector, which is categorized into three different types, local, adjacent and network. Access complexity ranging from high to low. Authentication, which can be either multiple, single or none. Confidentiality, which is the information access, ranges from none to complete. Integrity is the modification access that can be none, partial or complete. Availability is the accessibility of resources, and is either none, partial or complete. The factors are described in Table 2.

Factors	Values		
Access Vector	Local, the attacker must have physical access to the system	Adjacent Network, the attacker must have access to collision domain or the broadcast	Network, the attacker can exploit the vulnerability remotely
Access Complexity	High, specialized access conditions. e.g., attacker must have elevated privileges	Medium, access conditions exist, attacker is limited to some level of authorization	Low, access conditions do not exist.
Authentication	Multiple, the attacker is required to authenticate two or more times	Single, requires the attacker to log in to the system	None, authentication not required
Confidentiality	None, exploited vulnerability will not affect confidentiality	Partial, access to read some files, but cannot select which ones	Complete, the attacker has full read access to all the system files
Integrity	None, exploited vulnerability will not affect integrity	Partial, the attacker can modify system files, but cannot select which ones	Complete, the attacker can modify all the system files
Availability	None, exploited vulnerability will not affect Availability	Partial, the attacker can reduce the performance for availability	Complete, the attacker can deny access to all resources

Table 2. A brief explanation of each factor and their corresponding value, both of which have an impact on risk severity.

The severity is the calculated base score from the vulnerability, which is calculated by the exploitability of the vulnerability and their impact.

To calculate the exploitability, the values of Access Vector, Access Complexity and Authentication is multiplied by 20.

$$Exploitability = 20 * Access\,Vector * Access\,Complexity * Authentication$$

The Impact value is calculated by the values of Confidentiality, Integrity and Availability multiplied by 10.41.

$$Impact = 10.41 * (1 - (1 - Confidentiality) * (1 - Integrity) * (1 - Availability))$$

The base score is then calculated in the following way:

$$Base\,Score = ((0.6 * impact) + (0.4 * Exploitability) - 1.5) * f(impact)$$

The function $f(impact)$ is 0 if Confidentiality, Integrity and Availability is set to "None". Otherwise if any of Confidentiality, Integrity or Availability is "Partial" or "Complete", the value is a constant value of 1.176 [62].

As an example, if Confidentiality, Integrity and Availability is all set to "None", the Severity score will be 0.0, even if the values of Access Vector, Access Complexity and Authentication showing that the device is vulnerable to attacks. Although, if either of Confidentiality, Integrity or Availability is set to Partial or Complete, the Severity score increases.

Table 3 shows three different examples of vulnerabilities with different severity levels based on the above-mentioned calculations.

CVSS Calculator	Example 1	Example 2	Example 3
Access Vector	Local (0.395)	Adjacent (0.646)	Network (1.0)
Access complexity	Low (0.71)	High (0.35)	Medium (0.61)
Authentication	Multiple (0.45)	None (0.704)	Single (0.56)
Confidentiality	None (0.0)	Partial (0.275)	Complete (0.660)
Integrity	Partial (0.275)	Complete (0.660)	Complete (0.660)
Availability	Complete (0.660)	Complete (0.660)	Complete (0.660)
Result	5.0 (medium)	6.5 (Medium)	8.5 (High)

Table 3. Represents three examples with different values of each category using the CVSS calculator. Each example also shows the constant values in parenthesis of the selected value of each category that is used to calculate the base score.

The severity rating is based on the Common Vulnerability Scoring System (CVSS) [63]. In OpenVAS the ratings are set to Low (0.0 - 3.9), Medium (4.0 - 6.9) and High (7.0 - 10.0) [64].

The Quality of Detection (QoD):
Shows a value for each result which ranges from 0 to 100%, the higher the percentage, the higher is the reliability of the detection. In OpenVAS there is a default minimum value set to 70%, which means that results below 70% is filtered out and are not listed [64].

Host:
Lists which IP address the vulnerability is found on.

Location:
Displays which port, and if the vulnerability was found on a TCP or UDP port.

Summary:
Information about the vulnerability according to OpenVAS.

Solution:
The solution that OpenVAS suggests for the vulnerability.

The following pages includes the results of the vulnerability scans conducted on the devices, in the following order:

1. PlayStation 4
2. IKEA Trådfri
3. Chromecast
4. Apple TV
5. D-Link IP Camera

5.1 PlayStation 4

Vulnerability	Severity	QoD	Host	Location	Created
Linksys WRT54G DoS	10.0 (High)	99%	192.168.0.13	9295/tcp	Wed Apr 10 13:45:15 2019
Mongoose Webserver Content-Length Denial of Service Vulnerability	7.8 (High)	99%	192.168.0.13	9295/tcp	Thu Apr 11 18:17:34 2019
HTTP Windows 98 MS/DOS device names DOS	7.5 (High)	99%	192.168.0.13	9295/tcp	Fri Apr 5 12:17:14 2019
Format string on HTTP method name	6.9 (Medium)	99%	192.168.0.13	9295/tcp	Wed Apr 10 13:44:16 2019
Webseal denial of service	5.0 (Medium)	99%	192.168.0.13	9295/tcp	Fri Apr 5 12:15:35 2019
Jigsaw webserver MS/DOS device DoS	5.0 (Medium)	99%	192.168.0.13	9295/tcp	Fri Apr 5 12:15:50 2019
HTTP unfinished line denial	5.0 (Medium)	99%	192.168.0.13	9295/tcp	Wed Apr 10 13:44:37 2019
mod_access_referer 1.0.2 NULL pointer dereference	5.0 (Medium)	99%	192.168.0.13	9295/tcp	Thu Apr 11 18:18:07 2019
Mereo 'GET' Request Remote Buffer Overflow Vulnerability	5.0 (Medium)	99%	192.168.0.13	9295/tcp	Thu Apr 11 18:18:48 2019

Figure 4. The results of All TCP top 100 UDP search on the PlayStation 4 after five tests had been completed.

PlayStation 4 results

The results gathered from the vulnerability scan of the PlayStation 4 showed that there were several vulnerabilities as shown in Figure 4 and Table 4.

Vulnerability	Severity	Summary	Solution
Linksys WRT54G Denial of Service	10.0	It is possible to freeze the remote web server by sending and empty GET request	Upgrade firmware
Mongoose Webserver Content-Length Denial of Service	7.8	Successful exploitation will let the remote unauthenticated attackers to cause a denial of service or possibly execute arbitrary code	No known solution. Options is to remove or replace the product with a newer one
HTTP Windows 98 MS/DOS device names Denial of Service	7.5	It is possible to freeze or reboot Windows by reading a MS/DOS device through HTTP using file name like `CON\CON`, `AUX.htm/AUX`	Upgrade your system or use a server that filters those types of names out
Format string on HTTP method name	6.9	The remote web server seems to be vulnerable to a format string attack on the method name	Upgrade your software
Webseal denial of service	5.0	The remote web server dies when an URL ending with %2E is requested	Upgrade server or firewall
Jigsaw web server MS/DOS device DOS	5.0	It is possible to crash the Jigsaw web server by requesting /servlet/con 30 times	Upgrade your software
HTTP unfinished line Denial	5.0	It is possible to crash the web server by sending an unfinished line without a return carriage at the end of the line.	Upgrade your web server
mod_access_referer 1.0.2 NULL point dereference	5.0	The remote web server may be using a mod_access_referer apache module which contains a NULL pointer dereference bug which can be used for DoS attacks	Try another access control module, as the current one has not been updated for a long time
Mereo 'GET' Request Remote Buffer Overflow Vulnerability	5.0	Mereo fails to perform adequate boundary checks on user-supplied input before copying it to an insufficiently sized memory buffer	No known solution. Options is to remove or replace the product with a newer one

Table 4. Showing the vulnerabilities and severity of the vulnerabilities found in the PlayStation 4 with a summary and suggested solution.

Remarks about PlayStation 4
Compared to the other devices that were tested, the results from the PlayStation 4 were the ones that stood out the most. The results showed that the amount of vulnerabilities (9) were the highest out of every tested device. The worst vulnerability (10.0) was the most severe vulnerability found out of every tested device and the highest possible severity in according to the scale in OpenVAS.

The vulnerability with 10.0 severity, *Linksys WRT54G*, is mostly known to affect the Linksys routers (especially the WRT54G model) was also found to be present in the PlayStation 4 console.

The vulnerability with 7.5 severity pertaining to *HTTP Windows 98 MS/DOS device names*, affects Windows operating systems. The PlayStation 4 is using Orbis OS, which is an operating system based on FreeBSD which is not related to Windows, but the vulnerability was still found by OpenVAS.

Many of the vulnerability solutions suggest an update to the firmware, but all the tests were done on the PlayStation 4 firmware version 6.50 (the latest version at the time of the scan in April of 2019).

5.2 IKEA Trådfri Smart Lighting

IKEA Trådfri results
The results showed that IKEA Trådfri did not have any vulnerabilities according to our experiment.

Remarks about IKEA Trådfri
Based on the information that was gathered, the results showed that the IKEA Trådfri was safe from Denial of Service attacks. This could indicate that either the device is safe, or the tools that were used to conduct this experiment could not find the vulnerabilities present in the device. The two results that were found, were scans used to identify the operating system on the scanned device, with both having a severity grade of 0.0.

5.3 Google Chromecast (First generation)

Vulnerability	Severity	QoD	Host	Location	Created
SSL/TLS: Certificate Signed Using A Weak Signature Algorithm	4.0 (Medium)	80%	192.168.0.9	9000/tcp	Sun Apr 7 14:30:47 2019

Figure 5. The results of All TCP top 100 UDP search on the Chromecast after five tests had been completed.

Vulnerability	Severity	Summary	Solution
SSL/TLS: Certificate Signed Using A Weak Signature Algorithm	4.0	The remote service is using a SSL/TLS certificate in the certificate chain that has been signed using a cryptographically weak hashing algorithm	Servers that use SSL/TLS certificates signed with a weak SHA-1 hashing algorithm need to obtain new SHA-2 signed SSL/TLS

Table 5. Showing the vulnerabilities and severity of the vulnerabilities found in the Chromecast with a summary and suggested solution.

Remarks about Chromecast

The scans of the Chromecast gave only one result in total, and it had a severity value of 4.0 as shown in Figure 5 and Table 5.

The vulnerability that was found was not in the scope of denial of service but revealed that the Chromecast uses the outdated hashing algorithm SHA-1 which was released in 1995 and has been vulnerable to attacks since 2005. The solution in OpenVAS also mentions that both Microsoft and Google have warned users since 2017 that web sites using SHA-1 is not secure according to PCWorld [65][66].

5.4 Apple TV (Third generation)

Vulnerability	Severity	QoD	Host	Location	Created
Mongoose 'Content-Length' HTTP Header Remote Denial Of Service Vulnerability	5.0 (Medium)	100%	192.168.0.7	5000/tcp	Wed Apr 10 22:34:53 2019
Jigsaw webserver MS/DOS device DoS	5.0 (Medium)	99%	192.168.0.7	7100/tcp	Wed Apr 10 22:37:59 2019

Figure 6. The results of All TCP top 100 UDP search on the Apple TV after five tests had been completed.

Vulnerability	Severity	Summary	Solution
Mongoose 'Content-Length' HTTP Header Remote Denial of Service Vulnerability	5.0	Mongoose is prone to remote Denial of Service attacks, as it fails to handle specially crafted input	No known solution. Options is to remove or replace the product with a newer one
Jigsaw web server MS/DOS device Denial of Service	5.0	It is possible to crash the Jigsaw web server by requesting `/servlet/con` 30 times	Upgrade your software

Table 6. Showing the vulnerabilities and severity of the vulnerabilities found in the Apple TV with a summary and suggested solution.

Remarks about Apple TV

The results from Apple TV showed two types of Denial of Service vulnerabilities as shown in Figure 6 and Table 6. The first, *Mongoose* has no known solution, other than upgrade to a newer release. The *Jigsaw* vulnerability solution is to upgrade the software, which is a similar vulnerability to what was also found in the vulnerability scan for the PlayStation 4.

5.5 D-Link DCS-930LB Day Wi-Fi Camera

Vulnerability	Severity	QoD	Host	Location	Created
LiteServe URL Decoding DoS	9.3 (High)	99%	192.168.0.29	443/tcp	Wed Apr 10 19:29:00 2019
Polycom ViaVideo denial of service	5.0 (Medium)	99%	192.168.0.29	443/tcp	Wed Apr 10 19:30:40 2019

Figure 7. The results of All TCP top 100 UDP on the D-Link DCS-930LB Day Wi-Fi Camera after five tests had been completed.

Vulnerability	Severity	Summary	Solution
LiteServe URL Decoding DoS	9.3	The remote web server dies when an URL consisting of a long invalid string of % is sent	Upgrade your server or firewall
Polycom ViaVideo DoS	5.0	The remote web server locks up when several incomplete web requests are sent, and the connections are kept open	Upgrade your web server

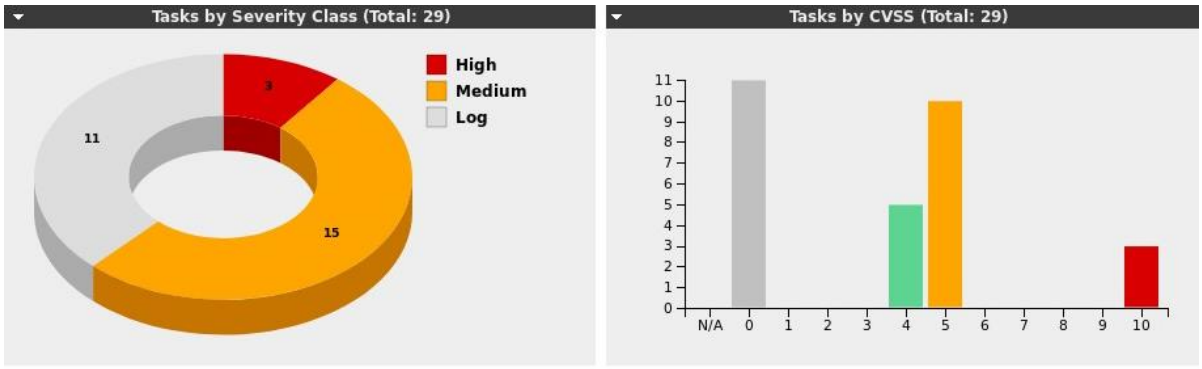
Table 7. Showing the vulnerabilities and severity of the vulnerabilities found in the Apple TV with a summary and suggested solution.

Remarks about D-Link IP Camera

The results of the scan on the D-Link IP camera showed two different types of vulnerabilities to Denial of Service attacks as shown in Figure 7 and Table 7. The vulnerabilities could lead to a temporary outage of the camera, and the consequences of an attacker being able to disable the camera could lead to security risks, such as breaking and entering.

5.6 Summary of the Results

All the devices tested besides the IKEA Trådfri were shown to be susceptible to Denial of Service attacks, as shown in Figure 8 and 9.
The results of the 30 scans that were done showed that the PlayStation 4 had the highest number of vulnerabilities and the vulnerability with the highest severity (10.0).
The D-Link IP camera had the second most severe vulnerability (9.3).
The results from the Chromecast showed that the only vulnerability that existed was the fact that the device was using an outdated hashing algorithm.
Apple TV showed that there were vulnerabilities with medium severity (5.0).
The scans on IKEA Trådfri did not find any vulnerabilities, which could be due to there not being any vulnerabilities, or because OpenVAS could not find them.



Name	Status	Reports		Severity	Trend	Actions
		Total	Last			
1. Playstation4 All TCP top 100 UDP (Full and very deep ultimate / All TCP top 100 UDP)	Done	5 (5)	Apr 17 2019	10.0 (High)		
2. IKEA Tradfri All TCP top 100 UDP (Full and very deep ultimate / All TCP top 100 UDP)	Done	5 (5)	Apr 11 2019	0.0 (Log)		
3. Chromecast All TCP top 100 UDP (Full and very deep ultimate / All TCP top 100 UDP)	Done	5 (5)	Apr 16 2019	4.0 (Medium)		
4. AppleTV All TCP top 100 UDP (Full and very deep ultimate / All TCP top 100 UDP)	Done	5 (5)	Apr 12 2019	0.0 (Log)		
5. D-link 930 All TCP top 100 UDP (Full and very deep ultimate / All TCP top 100 UDP)	Done	5 (5)	Apr 11 2019	9.3 (High)		

Figure 8. OpenVAS. The top graphs showing the number of vulnerabilities found sorted in categories, log (meaning the vulnerabilities found were not relevant, e.g., tests to detect which OS a device is running), medium (between 4.0 and 6.9) and high (between 7.0 and 10.0). The bottom list showing the different devices that were scanned for vulnerabilities, their status, the date of the scan and the most severe vulnerability found.

Figure 9 shows the total amount of vulnerabilities that were found in the devices. The results show that the PlayStation 4 was the device with both the most vulnerabilities (9), with D-Link IP Camera and Apple TV following, with two vulnerabilities each. Neither the results Google Chromecast nor from IKEA Trådfri shows that either of the device has any vulnerabilities to Denial of Service attacks.

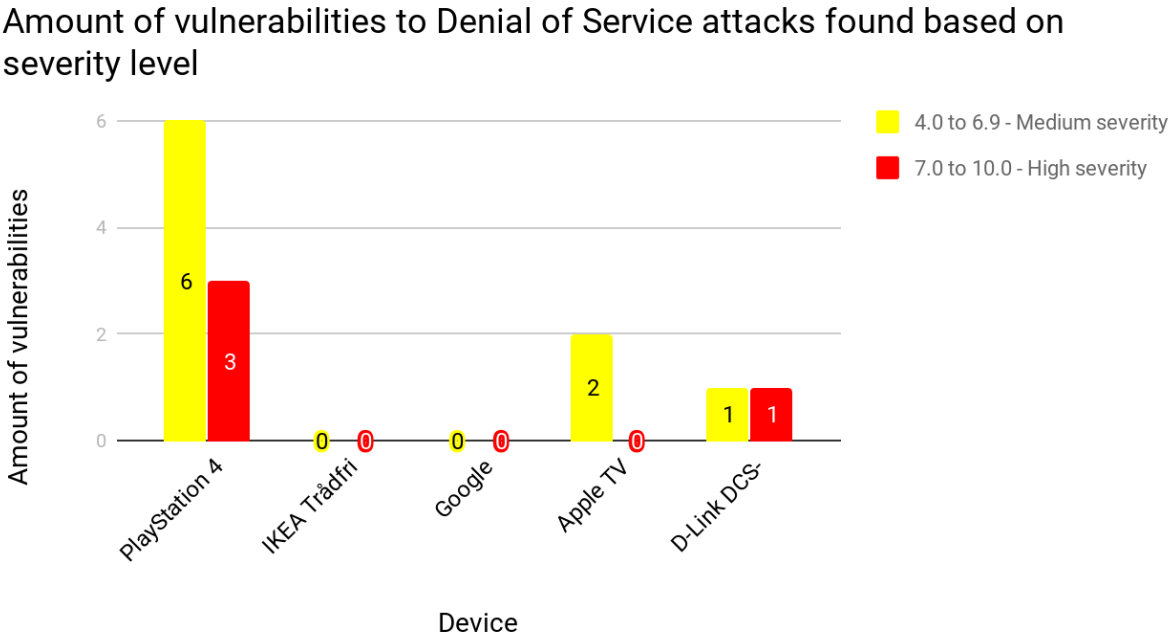


Figure 9. Showing all the vulnerabilities to Denial of Service attacks found in the devices based on severity level (4.0 to 6.9 categorized as “medium” severity and 7.0 to 10.0 categorized as “high” severity).

6 Analysis and Discussion

The purpose of this chapter is to analyze and discuss the results that were gathered from the experiment and, in some of the cases to find solutions to some of the vulnerabilities that were found.

6.1 Analysis

The analysis will be structured in the following way, firstly the most severe vulnerabilities of the individual devices are examined and evaluated. Secondly, a summary of the results from all the devices and the consequences the vulnerabilities may have, at the end of the chapter.

PlayStation 4
The vulnerability with severity 10.0, Linksys WRT54G DoS generally affects Linksys WRT54G Routers, but according to OpenVAS it also affects the PlayStation 4. The explanation given by Security Space [67], it is possible to freeze the remote web server by sending an empty GET request.

The Mongoose 'Content-Length' HTTP Header Remote Denial of Service Vulnerability found in PlayStation 4 has the next highest severity of 7.8. Exploit-db [68] shows an example how this vulnerability can be exploited, by sending a specially crafted GET request with a content-length of "`-2147483648`", which is equal to the minimal value of an integer [69]. By successfully sending this GET request, the web server will crash, and the service will be unavailable.

HTTP Windows 98 MS/DOS device names DOS is the last of the three vulnerabilities with severity above 6.9 (high severity). Security Focus [70] explains that the vulnerability can be exploited in several ways, an example on how to remotely exploit the vulnerability is to add "`/con/con`" at the end of the web service address, e.g., `http://target.example/con/con`.

All the vulnerabilities found in the PlayStation 4 were on port 9295, which is the port that enables "Remote Play" in PlayStation 4. Remote Play is a feature that enables the user to control their PlayStation 4 with their computer [71][72].

Based on the results of the PlayStation 4 scan, a secondary test was done after disabling Remote Play and updating to firmware version 6.51, which was released after the initial tests had been conducted. This was done to see if the update and the disabling of Remote Play solved the vulnerabilities.

The results of the secondary test showed that several vulnerabilities had been solved, as shown in Figure 10. The total number of vulnerabilities (9) and the severity of the vulnerabilities (highest being 10.0) that were found with Remote Play enabled was the highest out of the tested devices. After Remote Play was disabled the amount of vulnerabilities decreased to five, with the highest severity of those vulnerabilities being 6.9 compared to 10.0 with Remote Play enabled as shown in Figure 10, Figure 11 and Table 8.

Vulnerability	Severity	QoD	Host	Location	Created
Format string on HTTP method name	6.9 (Medium)	99%	192.168.0.13	9295/tcp	Mon Apr 22 14:20:42 2019
Webseal denial of service	5.0 (Medium)	99%	192.168.0.13	9295/tcp	Mon Apr 22 14:21:26 2019
HTTP negative Content-Length DoS	5.0 (Medium)	99%	192.168.0.13	9295/tcp	Tue Apr 23 11:20:59 2019
HTTP unfinished line denial	5.0 (Medium)	99%	192.168.0.13	9295/tcp	Tue Apr 23 14:18:15 2019
Jigsaw webserver MS/DOS device DoS	5.0 (Medium)	99%	192.168.0.13	9295/tcp	Tue Apr 23 11:21:50 2019

Figure 10. Result of scan of PlayStation 4 conducted after Remote Play was disabled.

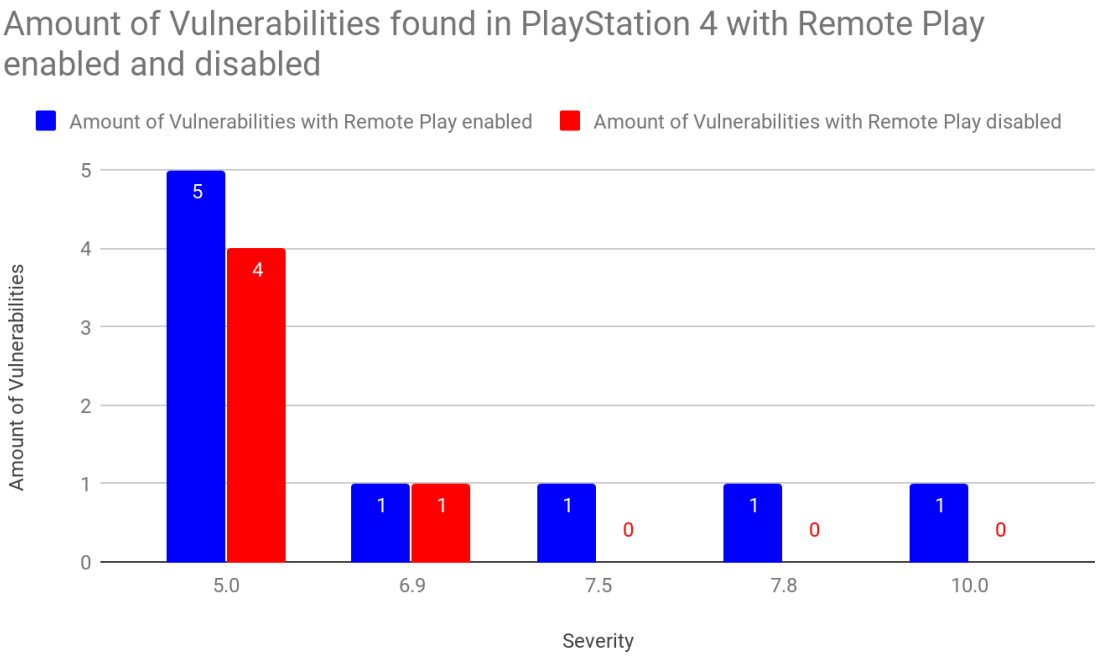


Figure 11. Showing the number of vulnerabilities found in the PlayStation 4 with and without Remote Play enabled.

Vulnerability	Severity	Remote Play Enabled	Remote Play Disabled
Linksys WRT54G Denial of Service	10.0	Yes	No
Mongoose Webserver Content-Length Denial of Service	7.8	Yes	No
HTTP Windows 98 MS/DOS device names Denial of Service	7.5	Yes	No
Format string on HTTP method name	6.9	Yes	Yes
Webseal denial of service	5.0	Yes	Yes
Jigsaw web server MS/DOS device DOS	5.0	Yes	Yes
HTTP unfinished line Denial	5.0	Yes	Yes
mod_access_referer 1.0.2 NULL point dereference	5.0	Yes	No
Mereo 'GET' Request Remote Buffer Overflow Vulnerability	5.0	Yes	No
HTTP negative Content-Length DoS	5.0	No	Yes

Table 8. Showing the specific vulnerabilities and whether they are present with PlayStation 4 Remote Play enabled or disabled (or both).

IKEA Trådfri

No vulnerabilities were found while using OpenVAS, but that does not necessarily mean that the device is safe. Some of the other tools mentioned in this thesis, e.g., Nessus or NMAP can be used to scan the device instead of only using OpenVAS. By using other tools, vulnerabilities might be found.

Chromecast

The vulnerability found in the Chromecast was an outdated signature algorithm (SHA-1) which has been vulnerable to hacking since 2005. The suggested solution to this vulnerability is to update to a newer security algorithm, e.g., SHA-2.
The vulnerability was located on TCP port 9000, but there were no articles with further information on the specific port, neither was there solution to close the port to enable further testing.

Apple TV

Two vulnerabilities were found on the Apple TV, the first being Mongoose 'Content-Length' HTTP Header Remote Denial of Service Vulnerability. Exploit-db [68] shows an example how this vulnerability can be exploited, which is by sending a specially crafted GET request with a content-length of "`-2147483648`", which is equal to the minimal value of an integer [69]. By successfully sending this GET request, the web server will crash, and the service will be unavailable. The vulnerability also states that there has not been a solution for over a year, and there might not be one either in the future.

The Jigsaw Webserver MS/DOS device DoS Vulnerability has the solution to upgrade to a newer software release. Security Focus [73] explains that this vulnerability can be exploited by sending requests for '`/servlet/con`' multiple times. Each request that is made permanently reduces the number of available server threads as there is no timeout function to cancel the request.

When the initial experiment was conducted the Apple TV was using software version 7.22 which was the latest update at the time (released December 6, 2017, scanned April 10, 2019). Apple released a new security update (7.3) on May 13, 2019 [74]. According to Apple, the update solves three possible vulnerabilities in Bluetooth and Wi-Fi connections, but Apple does not disclose, discuss or confirm security issues until an investigation has been conducted and there is a patch or new release available.

Connection	Impact	Description	CVE ID
Bluetooth	A remote attacker may cause an unexpected application termination or arbitrary code execution	An input validation issue existed in Bluetooth. This issue was addressed with improved input validation.	CVE-2017-14315
Wi-Fi	An attacker within range may be able to execute arbitrary code on the Wi-Fi chip	A memory corruption issue was addressed with improved memory handling.	CVE-2017-9417
Wi-Fi	An attacker within range may be able to execute arbitrary code on the Wi-Fi chip	A stack buffer overflow was addressed through improved input validation.	CVE-2017-6975

Table 9. Showing the security vulnerabilities that were fixed by the update to software version 7.3 [74].

The device was then scanned after updating to software version 7.3 using the exact same settings as the initial scan, to see if the vulnerabilities had been fixed. The results showed that none of the vulnerabilities found in software version 7.2.2 could be found in the updated software version 7.3.

D-Link IP camera

Both vulnerabilities in the D-Link IP camera enables a hacker to disable the camera temporarily, which could lead to big security risks if the camera is used for home surveillance.

The LiteServe URL Decoding DoS vulnerability was published on several security websites and vulnerability databases [75][76] some dating back to September of 2002, without there being a possible solution to the problem besides upgrading the firewall according to tenable.com [77].
The Polycom ViaVideo DoS vulnerability was added to vulnerability database websites in 2003 with the only possible solution available is to upgrade the web server. Both these vulnerabilities are as shown in Table 7.

Summary
The scans were repeated several times to make sure that the results were able to be reproduced, and it became clear that scans with the same settings sometimes had different results. Because of this all the results from five scans (per device) are shown.

OpenVAS, which was used to scan for vulnerabilities is a free software available for anyone to download which could mean that the scan is not as good as some of the paid alternatives, such as Nessus. But even with free software, the results clearly showed that 3 out of 5 devices had vulnerabilities to Denial of Service attacks. It is possible that more vulnerabilities could be found if the same devices are tested using another tool e.g., Nessus, Metasploitable or Nmap.

The vulnerabilities that were found in PlayStation 4, Chromecast and Apple TV enables a hacker to temporarily disable the devices. As these devices are mostly used for entertainment and recreational purposes the effects of their unavailability are not necessarily a security risk, but instead an annoyance. Regardless of this, vulnerabilities in devices that can cause unavailability should not exist as people generally want their devices to work when they want and need them.

The vulnerabilities found in the D-Link IP camera allows a hacker to temporarily disable the device. If the camera is used for security purposes, the risks and consequences can be grave if a hacker can disable the camera [42][11].

Based on the hypothesis in the thesis, which is that IoT devices are vulnerable to Denial of Service attacks, the results clearly showed that to be true in three out of the five tested devices. This fact does not necessarily show that three out of five IoT devices are unsafe universally, instead showing that to be true in this specific case. The devices tested in this thesis are ones that can be found in a home but does not necessarily reflect every home, as some homes have fewer devices and some have more. Some of the devices are not the newest or most current hardware version, which could also reflect on a realistic home where having the newest devices might not always be a priority.

6.2 Discussion

This thesis focuses only on Denial of Service attacks, and not DDoS attacks, which is an abbreviation for Distributed Denial of Service, which means it is a Denial of Service attack conducted by a cluster of computers [27]. The relevance is because IoT devices are becoming more common, and more people are relying on them daily. Because of this reliance, availability of the devices will become increasingly important as the consequences of temporary outage can lead to security risks and, in extreme cases health risks [9][17].

Based on previous studies done by Wang et al. [33], Gordin et al. [34] and Tekeoðlu et al. [42] using the settings employed by Liang et al. [49]. This was combined with information gathered from the literature review which indicated that OpenVAS was a suitable tool for examining vulnerabilities even though there were other options, such as Nessus, Nmap or Metasploitable [33][34].

There seems to be a general understanding by industry professionals and scholars that the security in IoT devices is not at as good as it should be according to industry professionals [9] and scholars such as Rauscher et al. [78]. But, as it is evident by the big increase in the amount of IoT devices, one can argue that consumers does not seem to think the lack of security is such a big problem as the experts do based on the annual increase in Internet connected devices [4-6].
With the limited security that some IoT devices may have, combined with the big increase of those devices it could lead to an increasing amount of problems. If the problem becomes big enough, and people start realizing that some of their devices might not be as safe as they think, the IoT and Smart Home market might take a hit as people become more cautious with buying devices that might compromise their privacy and security.

The experiment conducted in this thesis tested five devices that realistically can be found in homes around the world, scanned using free software available for anyone to use. In three out of five devices, vulnerabilities were found that can enable an attacker to temporarily disable the devices completely by conducting a Denial of Service attack to flood the device with requests. The fact that these vulnerabilities exist in devices manufactured and released by some of the biggest companies in the world (Sony, Google, Apple) begs to question, what about devices created by smaller companies with less capital and resources? Are the devices released by smaller companies more likely to be more or less secure? The increase in devices that has been going on for a number of years can be partly attributed to the bigger companies releasing more products, but similarly a huge number of those devices are released by smaller companies trying to find their spot in the growing IoT and Smart Home market.

The vulnerabilities that were found by OpenVAS in this thesis could easily have been found by the manufacturers during the development process, but that seems to not be the case as they had not been corrected and some will not be corrected at all according to the vendors, as shown in Table 4. Denial of Service or other types of availability attacks might not be the biggest threat as it pertains to a gaming console or a device used for streaming content to a television, but these vulnerabilities should not exist in a product released to the public, especially by enormous companies.

The vulnerability scans conducted in this thesis could easily be done by anyone wanting to test their own devices over a local network, and future additions to the

field of study could be done by using other devices, or other tools to see if the results are similar to the ones presented in this thesis.

One of the solutions that decreases the amount of vulnerabilities is to always keep the IoT devices updated to the latest firmware or software version, which is echoed by Khan et al. [86]. This fact was evident by a second vulnerability scan on the Apple TV. After the initial scan conducted in this thesis Apple released an update to the third generation of Apple TV as shown in Table 9, which removed all the vulnerabilities that were found in the initial scans of the Apple TV. Sony also released an update to PlayStation 4 which in combination with disabling Remote Play removed several of the most severe vulnerabilities that were found in the initial scan of the PlayStation 4, as shown in Figure 11.

7 Conclusions and Future Work

The purpose of this thesis is to present the vulnerabilities to Denial of Service attacks that exist in IoT devices found in smart homes.

After the experiment had been conducted and the results were examined, it was clear that there were vulnerabilities to Denial of Service attacks in three out of the five devices that were tested. Which in this specific scenario and case indicated that majority of the devices had vulnerabilities. The devices that were scanned were using the most recent software or firmware version at the time when the initial scan was conducted (April 2019), which indicates that the vulnerabilities currently exists in the devices that could be found in many people's (smart) homes. With the big increase in IoT devices [4-6], this could become an increasingly big issue if security continues being an afterthought in the development of the devices.

All of devices that were tested in this thesis are not the most current hardware version, which could be argued to be a realistic situation based on what devices are actually present in people's homes, as not everyone will buy the newest version of Chromecast or Apple TV every release year.

To add to the results of this thesis, the following additions could be made:

1. The most current version of the devices can be tested, e.g., Apple TV 4K or the newest Chromecast (fourth generation).
2. Testing other devices using similar methods.
3. Testing similar devices using different tools, such as Nessus, Nmap or Metasploitable which could present different results.
4. Each device could also be tested by conducting Denial of Service attacks to confirm that the vulnerabilities exist in practice.
5. To expand the work done in this thesis more types of Denial of Service attacks could be tested for.
6. Testing for other vulnerabilities related to Denial of Service, such as DDoS (Distributed Denial of Service).

The Greenbone Community Feed was used in this thesis, which is the free version of OpenVAS. OpenVAS also offers a paid version called Greenbone Security Feed. The difference between the paid and unpaid version is that the paid version scanner includes more Network Vulnerability Tests that is specially targeted for enterprise environments [79]. By using the paid version, more vulnerabilities could be found which might be something to consider in future work.

8 References

[1] D. Albright, “15 Examples of Internet of Things Technology in Use Today”, *Beebom,* February 2017. [Online] Available: https://beebom.com/examples-of-internet-of-things-technology/ [Accessed 2019-02-25].

[2] C. McClelland, “What is IoT? - A Simple Explanation of the Internet of Things”, *IoT for all*, January 2019. [Online] Available: https://www.iotforall.com/what-is-iot-simple-explanation/ [Accessed 2019-04-03].

[3] About the Internet of Things, “About the Internet of Things (IoT)”, *About the Internet of Things.* [Online] Available: http://www.abouttheinternetofthings.com/about-iot/ [Accessed 2019-03-12].

[4] Gartner Inc., “Gartner Says 8.4 Billion Connected ‘Things’ Will Be in Use in 2017, Up 31 Percent From 2016”, 2018. [Online] Available: https://www.gartner.com/newsroom/id/3598917 [Accessed 2019-04-03].

[5] Ericsson, ”Internet of Things forecast”, *Ericsson*. [Online] Available: https://www.ericsson.com/en/mobility-report/internet-of-things-forecast [Accessed: 2019-03-11].

[6] L. Columbus, “2017 Roundup Of Internet Of Things Forecasts,” *Forbes,* December 2017. [Online] Available: https://www.forbes.com/sites/louiscolumbus/2017/12/10/2017-roundup-of-internet-of-things-forecasts/#41a5fe1e1480 [Accessed: 2019-03-20].

[7] C. Liu, Y. Zhang and H. Zhang, "A Novel Approach to IoT Security Based on Immunology," *2013 Ninth International Conference on Computational Intelligence and Security*, Leshan, 2013, pp. 771-775.

[8] Bugeja Joseph, Vogel Bahtijar, Jacobsson Andreas, and Varshney Rimpu. "IoTSM: An End-to-end Security Model for IoT Ecosystems." *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2019, pp. 267-272.

[9] A. Banafa, “Three Major Challenges Facing IoT”, *IEEE*, March 2017. [Online] Available: https://iot.ieee.org/newsletter/march-2017/three-major-challenges-facing-iot.html [Accessed 2018-04-18].

[10] S. Singh and N. Singh, "Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce," *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2015, pp. 1577-1581.

[11] J. Bugeja, D. Jönsson and A. Jacobsson, "An Investigation of Vulnerabilities in Smart Connected Cameras," *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2018, pp. 537-542.

[12] A. Gerber, “Top 10 IoT security challenges”, *IBM,* November 2017. [Online] Available: https://developer.ibm.com/articles/iot-top-10-iot-security-challenges/ [Accessed: 2019-03-19].

[13] D. Roe, “7 Big Problems with the Internet of Things”, *CMSWiRE*, February 2018. [Online] Available: https://www.cmswire.com/cms/internet-of-things/7-big-problems-with-the-internet-of-things-024571.php [Accessed: 2019-04-10].

[14] C. Chen, Z. Zhang, S. Lee and S. Shieh, "Penetration Testing in the IoT Age" in *Computer*, vol. 51, no. 4, 2018, pp. 82-85.

[15] G. Baldini, A. Skarmeta, E. Fourneret, R. Neisse, B. Legeard and F. Le Gall, "Security certification and labelling in Internet of Things," *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, 2016, pp. 627-632.

[16] A. Boudguiga *et al.*, "Towards Better Availability and Accountability for IoT Updates by Means of a Blockchain," *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2017, pp. 50-58.

[17] Econsultancy, "10 examples of the Internet of Things in healthcare", *Econsultancy,* February 2019. [Online] Available: https://econsultancy.com/internet-of-things-healthcare/ [Accessed: 2019-04-02].

[18] Bugeja, Joseph, Andreas Jacobsson, and Paul Davidsson. "Smart Connected Homes." Internet of Things A to Z: Technologies and Applications. Wiley, 2018, pp. 359-384.

[19] S. ur Rehman and V. Gruhn, "An approach to secure smart homes in cyber-physical systems/Internet-of-Things," *2018 Fifth International Conference on Software Defined Systems (SDS)*, 2018, pp. 126-129.

[20] S. Siboni *et al.*, "Security Testbed for Internet-of-Things Devices," in *IEEE Transactions on Reliability*, vol. 68, no. 1, March 2019. pp. 23-44.

[21] Z. Zhang, M. C. Y. Cho and S. Shieh, "Emerging Security Threats and Countermeasures in IoT", 2015, *In Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security (ASIA CCS '15)*. 2015, pp.1-6.

[22] Pixaline, "Smart Home", *Pixabay*. [Online] Available: https://pixabay.com/illustrations/smart-home-house-technology-2005993/ [Accessed: 2019-04-03].

[23] S. Naik and V. Maral, "Cyber security — IoT," *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2017, pp. 764-767.

[24] D. Minoli, K. Sohraby and J. Kouns, "IoT security (IoTSec) considerations, requirements, and architectures," *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2017, pp. 1006-1007.

[25] P. Benjamin, "Demystifying STRIDE Threat Models" *DEV,* December 2018. [Online] Available: https://dev.to/petermbenjamin/demystifying-stride-threat-models-230m [Updated 2019-01-16; Accessed: 2019-03-23].

[26] Mozilla, "Confidentiality, Integrity and Availability", *Mozilla*. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/Security/Information_Security_Basics/Confidentiality,_Integrity,_and_Availability [Accessed: 2019-04-03].

[27] C. Kolias, G. Kambourakis, A. Stavrou and J. Voas, "DDoS in the IoT: Mirai and Other Botnets," in *Computer*, vol. 50, no. 7, 2017, pp. 80-84.

[28] S. Rizvi, A. Kurtz, J. Pfeffer and M. Rizvi, "Securing the Internet of Things (IoT): A Security Taxonomy for IoT," *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 163-168.

[29] A. Hussain, J. Heidemann and C. Papadopoulos, "A framework for classifying denial of service attacks", *In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03).* ACM, 2003, pp. 99-110.

[30] Cloudflare, "What is a denial-of-service attack", *Cloudflare.* [Online] Available: https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/ [Accessed: 2019-04-13].

[31] S. Weisman, "What are Denial of Service (DoS) attacks? DoS attacks explained", *Norton.* [Online] Available: https://us.norton.com/internetsecurity-emerging-threats-dos-attacks-explained.html [Accessed 2019-04-11].

[32] R. R. Zebari, S. R. M. Zeebaree and K. Jacksi, "Impact Analysis of HTTP and SYN Flood DDoS Attacks on Apache 2 and IIS 10.0 Web Servers," *2018 International Conference on Advanced Science and Engineering (ICOASE)*, 2018, pp. 156-161.

[33] Y. Wang and J. Yang, "Ethical Hacking and Network Defense: Choose Your Best Network Vulnerability Scanning Tool," *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2017, pp. 110-113.

[34] I. Gordin, A. Graur, A. Potorac and D. Balan, "Security assessment of OpenStack cloud using outside and inside software tools," *2018 International Conference on Development and Application Systems (DAS)*, 2018, pp. 170-174.

[35] R. M. Parizi, K. Qian, H. Shahriar, F. Wu and L. Tao, "Benchmark Requirements for Assessing Software Security Vulnerability Testing Tools," *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 2018, pp. 825-826.

[36] T. Collins, "Google has sold 55 million Chromecast devices" *Cnet,* October 2017. [Online] Available: https://www.cnet.com/news/google-has-sold-55-million-chromecast-and-chromecast-built-in-devices/ [Accessed: 2019-04-22]

[37] B. Gilbert, "The PlayStation 4 remains the world's most popular gaming console, with over 91 million sold", *Business Insider*, January 2019. [Online] Available: https://nordic.businessinsider.com/playstation-4-lifetime-sales-2019-1?r=US&IR=T [Accessed: 2019-04-22].

[38] C. Marshall, "What is smart lighting? Everything you need to know for your connected home", *Techradar,* July 2017. [Online] Available: https://www.techradar.com/news/what-is-smart-lighting-everything-you-need-to-know-for-your-connected-home [Accessed 2019-03-25].

[39] M. Rouse, "Chromecast", *Tech Target.* [Online] Available: https://whatis.techtarget.com/definition/Chromecast [Accessed 2019-04-02].

[40] L. Goode, "Too Embarrassed to Ask: What Is Apple TV, Anyway?", *Recode,* March 2015. [Online] Available: https://www.recode.net/2015/3/27/11560732/too-embarrassed-to-ask-what-is-apple-tv-anyway [Accessed 2019-04-12].

[41] SafeSite Facilities, “Internet Protocol (IP) Cameras - How do They Work & What are the Benefits”, *SafeSite Facilities*. [Online]
Available: https://www.safesitefacilities.co.uk/knowledge-base/internet-protocal-cameras-how-do-they-work [Accessed: 2019-04-12].

[42] A. Tekeoðlu and A. S. Tosun, "Investigating Security and Privacy of a Cloud-Based Wireless IP Camera: NetCam," *2015 24th International Conference on Computer Communication and Networks (ICCCN)*, 2015, pp. 1-6.

[43] C. Săndescu, O. Grigorescu, R. Rughiniş, R. Deaconescu and M. Calin, "Why IoT security is failing. The Need of a Test-Driven Security Approach," *2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 2018, pp. 1-6.

[44] B.J. Oates. “Researching Information Systems and Computing”. SAGE Publications Ltd., 2006.

[45] P. Sethi and S. R. Sarangi, “Internet of Things: Architectures, Protocols, and Applications,” *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 1-25. [Online] Available: https://www.hindawi.com/journals/jece/2017/9324035/ [Accessed 2019-04-02].

[46] "IEEE Xplore Digital Library", Ieeexplore.ieee.org, 2019. [Online]. Available: https://ieeexplore.ieee.org/Xplore/home.jsp. [Accessed: 2019-03-02].

[47] "ACM Digital Library", Dl.acm.org, 2019. [Online]. Available: https://dl.acm.org/. [Accessed: 2019-03-02].

[48] Kali.org, “Configuring and Tuning OpenVAS in Kali Linux”, *Kali*. [Online] Available: https://www.kali.org/tutorials/configuring-and-tuning-openvas-in-kali-linux/ [Accessed: 2019-04-03].

[49] L. Liang, K. Zheng, Q. Sheng and X. Huang, "A Denial of Service Attack Method for an IoT System," *2016 8th International Conference on Information Technology in Medicine and Education (ITME)*, 2016, pp. 360-364.

[50] VirtualBox, “Welcome to VirtualBox.org”, *VirtualBox*. [Online] Available: https://www.virtualbox.org/ [Accessed 2019-03-08].

[51] Kali.org, “What is Kali Linux?”, *Kali*. [Online] Available: https://docs.kali.org/introduction/what-is-kali-linux [Accessed 2019-03-08].

[52] Dookie, “Kali Linux Blog”, *Kali*, February 2019. [Online] Available: https://www.kali.org/blog/ [Accessed 2019-03-08].

[53] Offensive Security, ”Offensive Security”, *Offensive Security*. [Online] Available: https://www.offensive-security.com/ [Accessed 2019-03-08].

[54] OpenVAS, “OpenVAS - Open Vulnerability Assessment System”. [Online] Available: http://openvas.org/ [Accessed 2019-03-08].

[55] Kali, “OpenVAS 8.0 Vulnerability Scanning”. *Kali*. [Online]. Available: https://www.kali.org/penetration-testing/openvas-vulnerability-scanning/ [Accessed 2019-04-17].

[56] Greenbone, “About Ports”, *Greenbone,* [Online] Available: https://docs.greenbone.net/GSM-Manual/gos-4/en/performance.html#about-ports [Accessed 2019-04-17].

[57] M. Rouse, “UDP (User Datagram Protocol)”, *Tech Target.* [Online]. Available: https://searchnetworking.techtarget.com/definition/UDP-User-Datagram-Protocol [Accessed 2019-04-03].

[58] Greenbone, “Scan Configuration”, *Greenbone.* [Online] Available: https://docs.greenbone.net/GSM-Manual/gos-3.1/en/scan_configuration.html [Accessed 2019-04-17].

[59] K. Charles, “OpenVAS Terms to Know”, *Security ORB,* June 2018. [Online] Available: https://www.securityorb.com/general-security/openvas-term-to-know/ [Accessed 2019-03-21].

[60] Greenbone, “CVSS Base Score Calculator”, *Greenbone.* [Online] Available: https://secinfo.greenbone.net/omp?cmd=cvss_calculator&token=guest [Accessed 2019-04-17].

[61] Greenbone, “VT Development”, *Greenbone*, September 2018. [Online] Available: https://community.greenbone.net/t/vt-development/226 [Accessed 2019-04-17].

[62] Greenbone, “CVSS Calculator”, *Greenbone.* [Online] Available: https://docs.greenbone.net/GSM-Manual/gos-4/en/vulnerabilitymanagement.html#cvss [Accessed 2019-04-17].

[63] NVD, “Vulnerability Metrics”, *NIST.* [Online] Available: https://nvd.nist.gov/vuln-metrics/cvss [Accessed 2019-04-17].

[64] Greenbone, “Glossary”, *Greenbone*. [Online] Available: https://docs.greenbone.net/GSM-Manual/gos-3.1/en/glossary.html [Accessed 2019-04-17].

[65] L. Constantin, “Stop using SHA1 encryption: It’s now completely unsafe, Google proves”, *PCWorld*. [Online] Available: https://www.pcworld.com/article/3173791/stop-using-sha1-it-s-now-completely-unsafe.html [Accessed: 2019-04-22].

[66] L. Constantin, “Microsoft finally bans SHA-1 certificates in Internet Explorer and Edge”, *PCWorld,* May 2017. [Online] Available: https://www.pcworld.com/article/3195921/microsoft-finally-bans-sha-1-certificates-in-internet-explorer-and-edge.html [Accessed 2019-04-22].

[67] Securityspace, *Securityspace* [Online] Available: http://www.securityspace.com/smysecure/catid.html?id=11941 [Accessed 2019-04-22].

[68] Exploit Database, “Mongoose 2.11 - ‘Content-Length’ HTTP Header Remote Denial of Service” [Online] Available: https://www.exploit-db.com/exploits/35158 [Accessed 2019-05-03].

[69] Microsoft, “Integer Limits”, *Microsoft*. January 2018. [Online] Available: https://docs.microsoft.com/en-us/cpp/cpp/integer-limits?view=vs-2019 [Accessed 2019-05-03].

[70] SecurityFocus,"Microsoft Windows MS DOS Device Name DoS Vulnerability", *SecurityFocus,* 2010, [Online] Available: https://www.securityfocus.com/bid/1043/discuss [Accessed; 2019-05-14].

[71] Playstation, "PS4 Remote Play", *Sony*. [Online] Available: https://remoteplay.dl.playstation.net/remoteplay/lang/en/index.html [Accessed 2019-04-17].

[72] G. Quadrio, A. Bujari, C. E. Palazzi, D. Ronzani, D. Maggiorini and L. A. Ripamonti, "Network analysis of the Sony Remote Play system," *2016 IEEE Symposium on Computers and Communication (ISCC)*, 2016, pp. 10-13.

[73] SecurityFocus, "Microsoft Windows MS DOS Device Name DoS Vulnerability", *SecurityFocus*. [Online] Available: https://www.securityfocus.com/bid/5258/discuss [Accessed: 2019-05-03].

[74] Apple, "About the security content of Apple TV Software 3.0", *Apple*. May 2019. [Online] Available: https://support.apple.com/en-us/HT210121 [Accessed 2019-05-15].

[75] SecuriTeam, "LiteServe URL Decoding DOS", *SecuirTeam*, 2002. [Online] Available: https://securiteam.com/windowsntfocus/6R00B2A60E/ [Accessed 2019-05-03].

[76] M. Arboi, "LiteServe URL Decoding DoS", *Vulners,* 2017. [Online] Available: https://vulners.com/openvas/OPENVAS:11155 [Accessed: 2019-05-07].

[77] Tenable, "LiteServe HTTP Service Malformed URL Decoding Remote DoS", *Tenable.* [Online] Available: https://www.tenable.com/plugins/nessus/11155 [Accessed: 2019-05-03].

[78] J. Rauscher and B. Bauer, "Safety and Security Architecture Analyses Framework for the Internet of Things of Medical Devices," *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2018, pp. 1-3.

[79] H. Poston "A brief Introduction to the OpenVAS Vulnerability Scanner", *Infosec,* October 2018. [Online] Available: https://resources.infosecinstitute.com/a-brief-introduction-to-the-openvas-vulnerability-scanner/#gref [Accessed: 2019-05-15].

[80] A. C. Jose and R. Malekian, "Improving Smart Home Security: Integrating Logical Sensing Into Smart Home," in *IEEE Sensors Journal*, vol. 17, no. 13, pp. 4269-4286, 2017.

[81] R. A. Ramlee, M. A. Othman, M. H. Leong, M. M. Ismail and S. S. S. Ranjit, "Smart home system using android application," *2013 International Conference of Information and Communication Technology (ICoICT)*, 2013, pp. 277-280.

[82] M. Vacher *et al.*, "The sweet-home project: Audio technology in smart homes to improve well-being and reliance," *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011, pp. 5291-5294.

[83] S. Sedaghat, F. Adibniya and M. Sarram, "The investigation of vulnerability test in application software," *2009 International Conference on the Current Trends in Information Technology (CTIT)*, 2009, pp. 1-5.

[84] J. Bugeja, A. Jacobsson and P. Davidsson, "On Privacy and Security Challenges in Smart Connected Homes," *2016 European Intelligence and Security Informatics Conference (EISIC)*, 2016, pp. 172-175.

[85] M. H. Miraz, M. Ali, P. S. Excell and R. Picking, "A review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT)," *2015 Internet Technologies and Applications (ITA)*, 2015, pp. 219-224.

[86] W. Z. Khan, M. Y. Aalsalem and M. K. Khan, "Five acts of consumer behavior: A potential security and privacy threat to Internet of Things," *2018 IEEE International Conference on Consumer Electronics (ICCE)*, 2018, pp. 1-3.

[87] A. Dorri, S. S. Kanhere, R. Jurdak and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017, pp. 618-623.

[88] A. Vernotte, "Research Questions for Model-Based Vulnerability Testing of Web Applications," *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*, 2013, pp. 505-506.

[89] Doupé, Adam & Cova, Marco & Vigna, Giovanni. (2010). Why Johnny Can't Pentest: An Analysis of Black-Box Web Vulnerability Scanners. Proc. DIMVA 2010. 6201. pp. 111-131.

[90] M. Soliman and M. A. Azer, "Web Application API Blind Denial of Service Attacks," *2018 14th International Computer Engineering Conference (ICENCO)*, 2018, pp. 249-253.

[91] CCN, "CYBER-THREATS AND TENDENCIES", *CNN*, 2017. [Online] Available: https://www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos/2249-ccn-cert-ia-16-17-cyberthreats-trends-2017-executive-summary-1/file.html
[Accessed: 2019-06-05].

[92] H. R. Nagesh and K. C. Sekaran, "Design and Development of Proactive Solutions for Mitigating Denial-of-Service Attacks," *2006 International Conference on Advanced Computing and Communications*, 2006, pp. 157-162.

[93] N. Komninos, E. Philippou and A. Pitsillides, "Survey in Smart Grid and Smart Home Security: Issues, Challenges and Countermeasures," in *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, 2014, pp. 1933-1954.