



DEGREE PROJECT IN TECHNOLOGY,  
FIRST CYCLE, 15 CREDITS  
*STOCKHOLM, SWEDEN 2021*

# **Ethical Hacking of a Smart Fridge**

Evaluating the cybersecurity of an IoT device  
through gray box hacking

**MATEO FLOREZ**

**GABRIEL ACAR**



# **Ethical Hacking of a Smart Fridge**

**Evaluating the cybersecurity of an IoT device through gray box hacking**

MATEO FLOREZ

GABRIEL ACAR

Bachelors in Computer Science

Date: June 9, 2021

Supervisor: Fredrik Heiding

Supervisor: Robert Lagerström

Examiner: Pawel Herman

School of Electrical Engineering and Computer Science

Swedish title: Etisk Hackning av ett Smart Kylskåp



## **Abstract**

With the increasing popularity of Internet of Things (IoT) devices, a complete smart home is becoming more of a reality. Thus the security of said devices is becoming increasingly more important. Unsecured IoT devices could lead to potential consequences of societal proportion. Therefore there is an entire industry dedicated to mitigating such security threats and contributing to a sustainable society. The security of a Samsung Smart Refrigerator was evaluated in this thesis. The methodology implemented followed a gray box approach and consisted of initially creating a threat model, indicating potential vulnerabilities, and then using the said model to design penetration tests to assess the findings. It was concluded that the fridge was secure in the scope of this thesis. However, grounds for further research were discovered.

## Sammanfattning

Med den ökande populariteten för IoT-enheter (Internet of Things) blir ett komplett smart hem alltmer verklighet. Säkerheten för dessa enheter blir därför allt viktigare. Osäkrade IoT-enheter kan leda till potentiella konsekvenser av samhällelig omfattning. Därför finns det en hel industri som ägnar sig åt att mildra sådana säkerhetshot och bidra till ett hållbart samhälle. Säkerheten hos ett smart kylskåp från Samsung utvärderades i denna avhandling. Den metod som tillämpades följde en grå box-strategi och bestod av att först skapa en hot modell, ange potentiella sårbarheter och sedan använda nämnda modell för att utforma penetrationstester för att bedöma resultaten. Slutsatsen blev att kylskåpet var säkert inom ramen för denna avhandling. Det upptäcktes dock skäl för ytterligare forskning.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem Definition . . . . .	1
1.2	Objective . . . . .	1
1.3	Method . . . . .	2
1.4	Delimitation . . . . .	2
1.5	Related Work . . . . .	3
1.6	Outline . . . . .	3
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Internet of Things . . . . .	5
2.2	Cyber Attacks . . . . .	6
2.3	Penetration Testing . . . . .	7
2.4	Mitigating Security Risks and Threats . . . . .	9
2.5	Threat Modeling . . . . .	10
2.6	Ports . . . . .	11
2.7	Protocols . . . . .	11
<b>3</b>	<b>Methodology</b>	<b>13</b>
3.1	Penetration Testing Methodology . . . . .	13
<b>4</b>	<b>System Under Consideration: Samsung Family Hub</b>	<b>15</b>
4.1	Target of Attack - Introduced . . . . .	15
4.2	Samsung Tizen OS . . . . .	15
4.3	Mobile applications . . . . .	16
<b>5</b>	<b>Attack Surface Mapping</b>	<b>18</b>
5.1	Identified Components . . . . .	19
5.2	Architectural Overview . . . . .	20
5.3	Selected Threats . . . . .	24

<b>6</b>	<b>Penetration Testing</b>	<b>26</b>
6.1	Test 1: Sensitive Data Exposure . . . . .	26
6.2	Test 2: Security Misconfiguration (DoS) . . . . .	28
6.3	Test 3: DNS Response Flooding (DoS) . . . . .	30
6.4	Test 4: Broken Authentication . . . . .	34
<b>7</b>	<b>Results</b>	<b>38</b>
<b>8</b>	<b>Discussion</b>	<b>39</b>
8.1	Methodology . . . . .	39
8.2	Results . . . . .	39
8.3	Sustainability and Ethics . . . . .	40
<b>9</b>	<b>Conclusions and Future Work</b>	<b>42</b>
9.1	Conclusion . . . . .	42
9.2	Future Work . . . . .	42
	<b>Bibliography</b>	<b>42</b>



# Chapter 1

## Introduction

### 1.1 Problem Definition

Internet of Things (IoT) devices are becoming increasingly more popular. According to Statista research, the overall base of intelligent gadgets, such as smart TVs, smart locks, IP cameras, and their related services, will reach 75 billion units by the conclusion of 2025 [1]. As the popularity of IoT devices increases, so does the importance of security aspects surrounding these devices. One such device is a smart refrigerator. Refrigerators are necessary for every household, and smart refrigerators are considered the technologically advanced replacement of ordinary refrigerators. Thus it is plausible that many, if not all, future households will contain a smart fridge. Therefore, in order to guarantee the safety of one's house, it is necessary to identify all possible vulnerabilities of a smart fridge[2]. Finding unintended vulnerabilities can be done through iterative penetration testing (pen testing) of an IoT device.

There is currently no controlled global enactment on cybersecurity concerning IoT devices. However, domestic laws and authorities are pushing the issue of requesting better security regarding IoT devices. The need for increased security regulations regarding IoT devices is made clear as 2019 saw a 300% increase in cyber attacks in comparison to previous year[3]. The increase in popularity of IoT devices leads to a growing attack surface for cyberattacks; thus, the importance of securing IoT devices also increases.

### 1.2 Objective

The goal of this thesis is to assess some parts of the security of a smart fridge. The security of said IoT device will be tested through ethical hacking and the

process of penetration testing, where a set of cyber attacks will be attempted on the device. Should a vulnerability be discovered as a result of this project then it will be reported to the manufacturer primarily.

**Research question** The research question that this thesis answered was:

*Is the Samsung Family Hub smart fridge secure against cyber attacks?*

### 1.3 Method

This study was done in 4 parts:

1. The first part was a literature study aiming to increase the knowledge of penetration testing and cyber attacks, as well as answering questions such as "what is threat modeling?" and "what is a vulnerability assessment?".
2. The second part consisted of choosing a suitable methodology for the objective of the thesis.
3. The third part was to implement the chosen methodology and construct a threat model followed by a vulnerability assessment.
4. Lastly, the testing was conducted and the results documented.

### 1.4 Delimitation

Setting the scope of a penetration test is usually done together with the client. Since the client does not exist per se, parameters such as what kind of penetration test will be performed are decided solely by the authors.

The primary limitation was the hardware. It could not be accessed due to the damage the fridge would have encountered. Not having access to the hardware limited the extraction of Firmware for further examination and plausible extraction of valuable data, such as encryption keys and fingerprints used in protocols during communication.

Furthermore, indications of vulnerabilities mentioned under Section 6.1.4 suggest that the server the fridge communicates with, uses expired certificates, thus providing a potential entry point for exploitation, but, in accordance with Swedish legislation *Brottsbalken 4 kap. 9c §.*, we are not lawfully able to investigate these vulnerabilities without consulting Samsung.

Many potential vulnerabilities were disregarded due to time limitations that occurred since neither of us had previous knowledge in the field of penetration testing. Thus a more significant amount of the time available was spent on research rather than penetration testing.

An aspect of the system that was not considered part of this project's scope was the interaction of the fridge with other Samsung smart appliances, such as cameras. This aspect was disregarded due to limitations in resources and time.

## 1.5 Related Work

A thorough research was made into finding work that is related to this project.

The paper written by Lagerström and Wenjun (2019) answers the questions, "What is threat modeling?", and "What is state-of-the-art work in this field?".[4] The paper reviewed 176 articles and based on them produced a summary of state-of-the-art threat modeling methods. Thus this paper helped truly understand what threat modeling is and what it is used for.

In 2015, PenTestPartners performed a penetration test on DefCons Samsung Family Hub, RF28HMELBSR.[5] Their evaluation took place over a day and uncovered that the system was susceptible to a man in the middle attack. The attack allowed the intruder to access the users Gmail password when the system was updating its calendar with the users google calendar.

A penetration test revealing multiple vulnerabilities in the WebKit browser engine was conducted by Ajin Abraham in 2015.[6] Later in 2017, Dhiraj Mishra managed to conduct a same-origin policy bypass of Samsung web browsers versions earlier than 5.4.[7]

## 1.6 Outline

The thesis was divided into the following six chapters:

Chapter one is an introduction to the thesis which contains the nature of the problem to be examined. The objective as well as a mention of related work can also be found in this section.

Chapter two provides the background needed in order to understand the thesis. The target of attack for this thesis is introduced. Cyber attacks and some

known consequences of these are presented followed by possible countermeasures. Ports and different, relevant, protocols are explained.

Chapter three iterates through the methodology used in this project. The general approach is documented but a detailed attack method is given in a later section in the occasion of a successful attack.

Chapter four is a collection of the results of the examination. A threat traceability matrix is used to present the threat analysis as well as the results of implemented attacks.

Chapter five analyses and discusses the result and reflects over how well the thesis contributes to answering the previously stated problem.

Chapter six concludes the thesis and gives suggestions on what could be done differently for future work in the area.

# Chapter 2

## Background

### 2.1 Internet of Things

#### 2.1.1 Definition

*Internet of things (IoT)* devices are devices connected to diverse frameworks through the web. Among other things, these gadgets are used in various sectors, varying from the private sector to the industrial and commercial sector. IoT devices are utilized to facilitate and operate services.[8] IoT devices are devices such as smart TVs, smart locks, IP cameras, domestic assistants, and their related services.

#### 2.1.2 Security Concerns of IoT Devices

As the popularity of IoT devices increases [8], so do the security aspects surrounding these devices. According to Statista research[9], the whole introduced base of intelligent gadgets will reach 75 billion units by the conclusion of 2025.

Right now, there is no controlled universal enactment on cybersecurity concerning IoT devices. However, domestic laws and authorities are pushing the issue of requesting better security regarding IoT devices.[10]. The security perspective is not a solid incentive for distributors because of all the repercussions of being held responsible for possible vulnerabilities. Therefore the obligation is tilted more towards the consumer instead of the distributor. Thus, the responsibility of mitigating the possible security risk lies on consumers.

We all confront the challenge that these various devices have little or no security and represent the fastest-growing cyberattack target for organizations

worldwide, with cyberattacks up 300 percent in 2019 alone [3]. Cybercriminals exploit numerous vulnerabilities in smart devices and regularly utilize them to infiltrate whole systems (e.g. the Stuxnet Worm attack 2.2.2).

## 2.2 Cyber Attacks

### 2.2.1 Definition

There are plenty of definitions of cyberattack, the most popular and most cited coming from US government security expert Richard A. Clarke. Who defines cyber-wars as “actions by a nation-state to penetrate another nation’s computers or networks for the purpose of causing damage or disruption.”[11]. Most definitions are too broad and do not distinguish between a cyber-crime, cyber-attack and cyber-war. In this thesis, we embrace a narrow definition of cyber-attack, one implied to center consideration on the particular risk postured by cyber-technology. Cyberattack is therefore defined as attacking a computer or a computer-controlled system using the internet. A cyberattack is performed to damage, disrupt, or overload a system. Some attacks can cause damage if secret information falls into the wrong hands or leaks to the public (Information disclosure). For example, a so-called *Hacker* can *penetrate* a system only to expose its security flaws. Such exposure can be considered a warning bell or a significant threat, depending on its protracted facts.

### 2.2.2 Consequences of Cyber Attacks

#### Economic Impact

According to the report done by The Center For Strategic & International Studies (CSIS) in partnership with McAfee, cybercrime costs the world \$600 billion. The whole internet economy was in 2016 worth \$11.5 trillion or 15.5 percent of global GDP, the cost of cybercrime can be calculated to a 5 percent tax on growth [12], [13]. Ransomware is one type of attack that is rapidly being deployed and used against corporations, where the enterprises must pay a fee for the malware to be removed. \$209 million was paid in the first quarter of 2016 alone. This type of attack can spread through the network, IoT, and other devices without security[13].

#### The Stuxnet Worm Attack

In January 2010, inspectors from the International Atomic Energy Agency visiting a nuclear plant in Natanz, Iran, noted with bewilderment that centrifuges

used to enrich uranium were failing. Five months later, the same phenomenon was repeated in the country, but this time, the experts could detect the causes behind the malfunction: a malicious computer virus. The *worm* - now known as Stuxnet - took control of 1,000 machines involved in the production of nuclear materials and gave them instructions to self-destruct essentially. The Stuxnet Worm attack is the first recorder cyberattack that damaged "real world" infrastructure [14].

The virus searched for the programmable logic controllers (PLC) among the infected computers. PLCs are used to automate machine processes. After finding a PLC computer, the malware attack upgraded its code over the web and started sending damage-inducing instructions to the PC-controlled electro-mechanical gear. At the same time, the infection sent false signals to the central controller. Anybody overseeing the hardware would have had no sign of an issue until the hardware started to self-destruct[14].

## 2.3 Penetration Testing

The need for Penetration Testing, also known as Ethical Hacking or Pen Testing, grows with the increase of new products with diverse security. An Ethical Hacker, also referred to as a white-hat hacker, performs the same sort of attacks as a malicious hacker or black-hat hacker, but the intention of the attack differs. An Ethical Hacker performs the attacks with legal authorization to intrude on other systems with the intent of finding vulnerabilities through which a malicious hacker could attack[15].

### 2.3.1 IoT environment

When it comes to Penetration Testing, the IoT environment is considered more complex than traditional environments. Such an environment covers different architectures, operating systems, and communication protocols. Understanding the whole ecosystem, proper investigation of components, and developing a comprehensive assessment plan are the keys to successfully securing the IoT environment. An IoT environment mainly consists of the following components[16]:

- **Network:** An IoT environment runs and updates over a network, for example, the Internet, BLE, 4G, LTE, Zigbee, LoRA, WiFi, MQTT, 802.11.15.4.

- **Applications:** The applications in IoT manage the devices. The Web-App, The Mobile-App, can be web applications or mobile applications.
- **Firmware:** This is the software and operating system of the device.
- **Encryption:** Encryption protects communications and data stored on the device.
- **Hardware:** The hardware of the IoT device varies. Examples of hardware are, integrated circuits (IC), storage, JTAG, UART ports, sensors and cameras.

### 2.3.2 Types of Penetration Testing

A parameter that will affect the proceedings of the Penetration Testing methodology, more specific the threat model methodology, is what kind of pen-test is done.

#### White Box

In this type of test, pen-testers or Ethical Hackers have full know the system's internal workings and work with information that one or more employees can access within the organization. Such information could, for example, be source code and other critical information that is not disclosed to users outside the system.

With this preliminary information, the penetration test can accurately target its attack and discover what needs to be improved and reoriented. As it is a high volume of preliminary information, this type of Pen-test is generally carried out by members of the company's own IT team.

#### Black Box

In this type of testing, the penetration testers do not know the system's internal workings and work with the information they can get through their means, just as a hacker could. Given these characteristics, it will act in a highly similar way to that of cybercriminals without much information mapping.

#### Gray Box

Gray-box testing is a mixture of the two aforementioned box-testing strategies. Here the pentester may have access to and knowledge about the system. Gray-box penetration testers usually have some knowledge of the internals of the



network. This kind of testing aims to provide a more focused and efficient assessment of the network's security. Having prior access to information about the system leads to more focus on the assessment than spending time determining that information.[17].

Regardless of the type of penetration test. According to Gupta in the book aforementioned [16], the testing should consist of the four following phases:

1. Understanding Scope.
2. Attack Surface Mapping.
3. Vulnerability Assessment and Exploitation.
4. Documentation and Reporting.

## 2.4 Mitigating Security Risks and Threats

The Swedish government commissioned the National Defense Radio Establishment (FRA), the Swedish Armed Forces, The Swedish Civil Contingencies Agency (MSB), The Swedish Security Services, to together take preparatory measures and submit a proposal to serve as the base for the creation of a National Cyber Security Center in 2020. In parallel with this, in-depth regulatory cooperation was taking place aimed at promoting the task. As part of this, the authorities together with the Swedish Police Authority jointly produced a report "Cybersäkerhet i Sverige – hot, metoder, brister och beroenden"[18]. The purpose of the report was to jointly compile a situation picture that in a simple and accessible manner describes cybersecurity from a national perspective[18].

Based on the previously mentioned report, the same organizations established the following recommendations in order to counteract the vulnerabilities highlighted:

- **Ensure an ability to detect security incidents:** Acquire the ability to detect security incidents in the IT environment as early as possible. Monitor events in the IT environment with manual, technical, and automatic measures. Create security logs for monitoring, such logs should be protected against unauthorized access or alteration.
- **Upgrade software and hardware:** Replace obsolete hardware and software to mitigate vulnerabilities that have been exposed over time and to obtain the intended function and sufficient safety.

- **Install security updates as soon as possible:** Prioritize updating information systems exposed to the internet, business-critical, and those where vulnerabilities risk can be exploited. Aim to install security updates as soon as they are published[19].

## 2.5 Threat Modeling

*Threat modeling* is the process in which the penetration tester analyzes the different components and processes of a device to discover possible attack surfaces. During the process, the security and countermeasures of individual components, or lack thereof, are identified and analyzed. Using this information, it is then possible to rate the threat of found attack surfaces. Threat modeling thus allows focusing on threat mitigation more effectively. There are many different threat modeling methodologies, such as PASTA and STRIDE. In this thesis, STRIDE will be used.

### 2.5.1 STRIDE Threat Modeling

STRIDE is a mnemonic that represents many things that can go wrong with a system's security. It has its origin in Microsoft, where it was used to facilitate the reasoning of the possible threats that a system may face. In order to follow the STRIDE method, the system must be decomposed into different components. These components are analyzed to check if the system is susceptible to threats. Actions are then taken to mitigating the threats, and the process is repeated until reaching a comfortable state with the remaining threats. [20]. STRIDE describes the following stages:

- **[S]poofing Identity:** This category of threats indicates that a users should not be able to be impersonated by others in order to gain access to the system.
- **[T]ampering with Data:** Unauthorized tampering of something on disk, network, memory, or elsewhere.
- **[R]epudiation:** In this category, the goal is to establish a level of adequate monitoring of the actions carried out by users.
- **[I]nformation Disclosure:** Providing information to someone not authorized to access it. For example, the unsafe use of shared USB memory that facilitates obtaining credentials or obtaining a list of users.

- **[D]enial of Service:** Exhausting resources making a service or an application unavailable. Can be done by flooding a server exhausting the system's resources.
- **[E]levation of Privilege:** Allowing someone to do something they're not authorised to.

## 2.6 Ports

A port is used as a communications endpoint in IoT devices. By using ports, communication between client and host can be established using different protocols. Each port has a port number and is identified with it. The port number is always associated with the IP address of the host and the protocol used for communicating. Short-lived ports, called Ephemeral ports, are often found in ports 49152 to 65535[21].

## 2.7 Protocols

### 2.7.1 IP - Internet Protocol

**TCP** or Transmission Control Protocol is an internet protocol in charge of informing the destination of the data, allowing secure connections. The main characteristic of the TCP protocol is that it is a connection-oriented protocol. In order to establish a connection between client and server, it is necessary to establish a previous connection with said server.

This previous connection is called a 3-way handshake and consists of the client (the one that initiates the connection) sends an SYN message to the server (the one that receives the connection). Subsequently, the server will send an SYN-ACK message, indicating that it can start sending information. Finally, the client sends an ACK- package indicating that it has received it correctly. Now all the information is being sent between the client and server in a bidirectional way. [22].

### 2.7.2 HTTP - Hypertext Transfer Protocol

*HTTP*, Hypertext Transfer Protocol, is the name of a protocol that allows us to request data and resources, such as HTML documents. It is the basis of any data exchange on the World Wide Web and a client-server structure protocol. Meaning that a data request is initiated by the element that will receive the

data (the client), usually a Web browser. Thus, a complete web page results from the union of different sub-documents received, such as, for example, a document that specifies the layout style of the web page Cascading Style Sheets (CSS), the text, images, videos, scripts.

HTTP is a protocol based on the client-server principle. Requests are sent by an entity, the user, or a proxy at one's request. Each request is sent to a server, which manages and responds to it. There are several intermediaries between each request and response, usually called proxies, which perform different functions, such as gateways or caches. [23]

### 2.7.3 SSL/TLS - Security protocols

SSL stands for Secure Sockets Layer, a protocol for keeping Internet connections secure by ensuring confidentiality, data integrity, and authentication of information sent between two systems. The TLS (Transport Layer Security) protocol is just an updated and more secure version of SSL[24].

SSL / TLS works by binding identities of devices such as websites and companies to cryptographic key pairs via digital documents called X.509 certificates. Each key pair consists of a private key and a public key. The private key is kept secure, and the public key can be distributed mainly via a certificate.

The relation between the private and public key pairs means that it is possible to use the public key to encrypt a message that the possessor of the private key can only decrypt. In addition, the holder of the private key can use the key to sign other digital documents, e.g., web pages, and anyone with the public key can verify this signature.

The most common and well-known use of SSL / TLS is secure web browsing via HTTPS protocol. A properly configured public HTTPS Website contains an SSL / TLS certificate signed by a publicly trusted certification authority (CA). Users visiting an HTTPS website can be sure of:

- **Authenticity:** The server presenting the certificate has a private key that matches the public key in the certificate.
- **Integrity:** Documents signed by the certificate (e.g., web pages) have not been changed in transit by a Man in the middle.
- **Encryption:** Communication between the client and the server is encrypted.

# Chapter 3

## Methodology

### 3.1 Penetration Testing Methodology

The implemented Penetration Testing methodology followed is based upon two sources, the first source being the hacking guides posted on the NSE Lab homepage [25] and the second a book, “The IoT Hacker’s Handbook - A Practical Guide to Hacking the Internet of Things.”[16] Primarily based on the delimitations mentioned in section 1.4 a Gray box testing methodology was followed. This stage aims to understand the scope of the penetration test and any other constraints and limitations.

#### 3.1.1 Information Gathering

As a first step in the penetration testing process, information on ports and the communication of the device was gathered. A network scan of the target device was performed to discover the various ports that are open and what type of services/protocols are running. The scan was performed with the network scanning tool Nmap. Nmap is a powerful tool that allows us to see the different open ports, the services running and, in some cases, perform additional exploitation.[26].

Furthermore, when it comes to gathering information about the devices communication, Ettercap[27] was used to perform a Man-in-the-Middle attack. Ettercap allows for techniques such as ARP poisoning. It works by tricking the client into sending network traffic through the attacker’s machine (man in the middle), which enabled us to *Sniff*, (capture and monitor), all the network packets throughout the network with the help of the tool Wireshark.[28]

To further examine the device, a vulnerability scanning program, Nessus,

was used [29]. Nessus [29] is a vulnerability scanning program for various operating systems. It consists of a daemon or devil, *nessusd*, that scans a system in search for known vulnerabilities. In regular operation, Nessus starts by scanning ports with Nmap or its port scanner to find open ports and then tries various exploits to attack it.

### 3.1.2 Attack Surface Mapping

In accordance with Gupta [16], the following steps were taken in order to create the attack surface map of the target device:

#### **List all the components present in the target group**

The various components were identified and represented in table 5.1 The components were assigned an ID and described.

#### **Prepare an architecture diagram & Label the components and the communication flows between them**

Based upon the previous step, a data flow diagram was created with the different components along with the various communication protocols and channels. The tool used for this step was *diagrams.net*. (Figure 5.1)

#### **Identify attack vectors for each component an the communication or protocol used**

In this step, the data flow protocols and applications were analyzed in order to disclose exposed entry points. Every entry point was given an ID and a description and documented in Table 5.2.

#### **Categorize the attack vectors based on the varying criticality**

Based on the information in the previous step the actual threats were identified. To do so, different tool were utilized. One of the tools was the *STRIDE* model, and the library of attacks *OWASP Top Ten*.

### 3.1.3 Penetration Testing Rapport and Documentation

The process of pen testing the intelligent fridge was continuously documented in a table presenting the nature and success of each test, a threat traceability matrix.

## **Chapter 4**

# **System Under Consideration: Samsung Family Hub**

This chapter introduces the target of this security evaluation, the Samsung Family Hub Side by Side Fridge, and presents its functions.

### **4.1 Target of Attack - Introduced**

The target to be evaluated in this thesis is the Samsung Family Hub Side by Side Fridge, specifically the model RS68N8941SL, and will continuously be referred to as the Target of Attack (ToA). The main difference between an ordinary fridge and a smart fridge is the fact that a smart fridge is connected to the cloud[30]. This connection enables the fridge to feature several additional functions over the traditional ones. The ToA has features such as a camera inside the fridge that allows the user to see the contents through a smart phone and tag items with expiry dates. Another feature is a 21.5 inch touch display on the outside of the fridge door with several installed applications. These applications include an internet browser and Spotify. The ToA allows many of these functions to be controlled through a smartphone and supports WiFi and Bluetooth communication. The operating system(OS) on which the device runs is a Linux based OS, developed by Samsung, called Tizen.

### **4.2 Samsung Tizen OS**

The Samsung Tizen operating system is, as mentioned, an operating system based on the Linux kernel. It is used for multiple Samsung devices and comes

in multiple different profiles, each adapted for the use of their respective device[31]. Tizen comes with multiple core services, and one such feature is its security system. The built-in security is responsible for access control, certificate management and secure application distribution[32]. Applications are run with non-root privileges and their access to the file-system is therefore restricted. The primary software used for security in Tizen is Simplified Mandatory Access Control in Kernel (SMACK). SMACK is a Linux security module used to control the access rights of different processes[33]. In Tizen SMACK is implemented with a three-domain policy. The policy consists of having three groups of labels or domains. These domains are often: Floor, System, and User. Objects have free access to other objects in the same domain but restricted access to cross-domain objects, although there are exceptions. In order to reduce the load on SMACK, Tizen also uses Cynara. Cynara works as a security control point. System services use Cynara to check whether or not to grant access rights to resources for clients.

Developing applications require both an author signature and a distributor signature in order to work on Tizen. The store distributor stores the privileges of different applications. However, the application permissions are registered on a manifest that is stored locally, only accessible to system daemons. Additionally, Tizen uses a Content Secure Policy (CSP) which provides the system with a countermeasure towards various attacks such as XSS, injections, click-jacking, or other forms of malicious injections. [34]

## 4.3 Mobile applications

Samsung has released two different applications for connecting your phone to the smart fridge, SmartThings and Samsung Family Hub. SmartThings is used for many of Samsung's IoT devices and can be used for administrative functions, while Samsung Family Hub allows the user to access some of the applications on the fridge. Both of the mobile applications are available on both iOS and android and can use WiFi or mobile provider network to communicate with the smart fridge.[35]

**SmartThings** As an additional feature for some of their smart products, Samsung launched a mobile application available on both iOS and android, SmartThings. This mobile application can be used to regulate the temperature of the fridge, view the contents of the fridge through your phone, update the software of the fridge, and more.[35]



**Samsung Family Hub** Samsung Family Hub features such as sharing ones calendar with the calendar app in the fridge, create and edit to-do lists, getting and sending memos from and to the fridge, sending and receiving whiteboard messages, sending photos to the fridge, creating and editing shopping-lists, and viewing the insides of the fridge as well as tagging items with expiry dates.[35]

# **Chapter 5**

## **Attack Surface Mapping**

The Attack Surface Mapping (or Threat modeling) methodology used is described under section 3.3.1. Here follows the attack surface mapping of the device in consideration.

## 5.1 Identified Components

Table 5.1: Components of the system under consideration

ID	Component	Description
1	Samsung Refrigerator	The Smart fridge has a built in display that the user can interact with. The display runs Tizen 4.0 operative systems, which allows for the system to have applications. The fridge has two built in cameras inside that serve for monitoring.
2	Web application	The web application consist of the local browser, Samsung internet Browser.
3	Mobile application	As mentioned in section 4.3, there are two applications. These applications can be used to upload pictures, edit settings such as temperature, and more.
4	Built in Cameras	The built in cameras provide a view of the fridges inside only when the fridges doors are closed. The cameras take a pictures that the an authorized user can see access via the mobile application.
5	Cloud services	The cloud services, <i>Samsung/Smart Things Cloud</i> are used to update the state of the device.
6	Network	The devices Web communications are done with WiFi via the built in WiFi Module CWAP210M.
7	Firmware	The Fridges runs on the Tizen OS which in turn uses Linux kernel 4.9.

## 5.2 Architectural Overview

This section consists of two parts: the data flow diagram of the components and their communication or protocols.

### 5.2.1 Decomposing The Samsung Smart Fridge System

The system under consideration was decomposed to better understand the various component and the data flow between them. We used the following tools for extracting the different communication protocols between the various components (HTTP, TCP): Nmap, Wireshark, and Ettercap. The tools were used as described in section 3.1.1. The extracted information is represented as a diagram, figure 5.1.

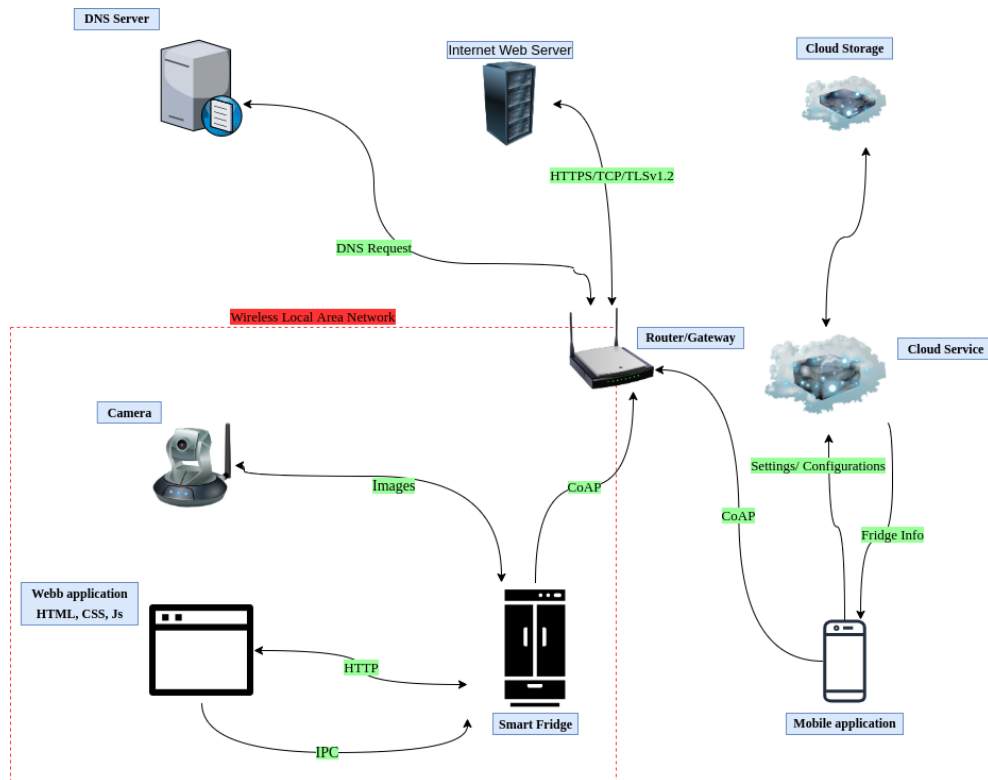


Figure 5.1: Attack surface map for Samsung Smart Fridge

Further inspecting of the diagram in Figure 5.1, and all the technical specifications of the various devices led to a better view of the systems vulnerable entry points. The different entry points were documented (Table 5.2), each given an ID and a description.

Table 5.2: Entry points.

ID	Entry points	Description
1	Samsung Refrigerator	The device has a web browser and a mobile application. The device utilizes a cloud service to perform firmware updates and for storage. Various open ports of the device have been found and analyzed.
2	Web interface (Samsung internet browser)	The web interface uses TLS 1.2 encryption protocol when communicating with TCP protocol. TLS 1.2 allows secure web (HTTPS) connections.
3	Mobile application	The mobile application is used to oversee the fridge and its content with the built-in cameras' help. The application can also be used to apply configurations on the fridge remotely. To set up the application, one must be in the same network as the fridge and have credentials in the form of a <i>Samsung Account</i> . The data flow from the application to the fridge takes place over Samsung's cloud services.

Continuation of Table 5.2		
ID	Entry points	Description
4	Cloud services	The fridges communication with the cloud services is encrypted with TLS 1.2 protocol. The cloud services act as an intermediary between the mobile application and the fridge, allowing exchange of information about the state of the fridge or information about changes made via the mobile application.
5	Network	All connections to the web occur over WiFi 2.412-2.462 GHz, via the built in WiFi Module CWAP210M, and use the IEEE 802.11 protocol.
6	Firmware	Downloaded applications can use the stock firmware to control the devices behavior if authorized by the devices access control.

### 5.2.2 Identifying Threats

With different entry points identified, the next step was identifying threats and plausible targets. Based upon the *STRIDE* approach, Table 5.3 was made presenting potential threats and exploits.

**5.2.2.1 STRIDE Threats**

Table 5.3: STRIDE Treats.

Threat Type	Threats
Spoofing	<ul style="list-style-type: none"><li>• Spoofing of user interaction with web application</li><li>• Spoofing of user interaction with mobile applications</li></ul>
Tampering	<ul style="list-style-type: none"><li>• Alter the packages in transit</li><li>• Load malicious files to the device</li><li>• Tampering with the the firmware to perform unauthorized actions.</li></ul>
Repudiation	<ul style="list-style-type: none"><li>• Disable logging of mobile and web applications</li></ul>
Information disclosure	<ul style="list-style-type: none"><li>• Leak credentials</li><li>• Interception of image traffic from the cameras to the server, web/mobile application</li></ul>

Continuation of Table 5.3	
Threat Type	Threats
Denial of Service (DoS)	<ul style="list-style-type: none"> <li>• Exhaust legitimate requests</li> </ul>
Elevation of privilege	<ul style="list-style-type: none"> <li>• Elevation from low level user to a more privileged user</li> </ul>

## 5.3 Selected Threats

In the following section the threats were analyzed based on their impact and their probability of success. Based on aforementioned criteria, time, and resources, threats were filtered and a subset was selected for further testing.

The following entry points were excluded:

- **Firmware:** The Samsung Smart Fridge firmware is not available for download from any source. Sniffing Over The Air (OTA) while the device is updating is not an option as the device is up to date thus does not require updating. Extracting the firmware via the hardware is not an option due to fact that the device would have to be damaged beyond repair, and this method goes beyond the scope of the pentest. The only option left is to reverse the application, more specific the mobile application *Smart Things*, but due to time limitations this is was not tried.
- **Web interface:** The web interface was excluded from further testing due to the nature of the scope (See section 1.4).

After analyzing the results from the Nessus scan, one vulnerability surfaced. The fridge could potentially be susceptible to a DoS attack because it responds to DNS requests. Furthermore, after performing an ARP poisoning attack with the Ettercap tool, and then sniffing the communications from the fridge to different destinations with Wireshark, we discovered that the cloud server website uses outdated certificates. This indicates that the security patches are not up to date, and potential vulnerabilities can be found,



which makes it plausible to mount a Man-in-the-middle attack between the cloud server and the device. Also, after performing various scanning with Nmap, port 17654/TCP was open. This port is in the range of static ports and can potentially be exploited through a DoS attack.

Finally, the threats chosen for further evaluation were:

**From Nessus:**

1. DNS Response Flooding (DoS)

**From OWASP top ten:**

1. Sensitive Data Exposure (Information disclosure)
2. Broken authentication
3. Injection (Tampering)
4. Security Misconfiguration (DoS)

# Chapter 6

## Penetration Testing

### 6.1 Test 1: Sensitive Data Exposure

This exploit comes from the OWASP Top Ten.

#### 6.1.1 Background

Data sent and received over the internet between devices (Server & Client) can be intercepted (Sniffed) by third parties. Therefore the data must be encrypted/hashed to protect its content from a malicious attacker. The encryption and decryption process can differ depending on what medium the data is transferred through and what protocols the devices or services use, e.g., Section 2.7.3

#### 6.1.2 Method

As mentioned before, a way for a malicious attacker to gain access to sensitive information such as credentials, sensitive emails is to *Sniff* the packet traffic in a network. The OWASP Top Ten presents the following questions to mitigate sensitive data exposure:

1. Is any data transmitted in clear text?
2. Are any old or weak cryptographic algorithms used either by default or in older code?

To intercept the packages sent and received by the device, an ARP poisoning attack was made with the tool

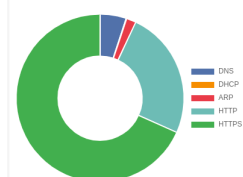


Figure 6.1: A-packets - Protocols

Ettercap as mentioned in Section 3.1.5. This attack enabled us to determine the communication protocols used by complementing the procedure with Wireshark to enable capturing and monitoring of the sent packets.

Interactions with the device were done while the communication was intercepted to get as much traffic to analysis as possible. The packets collected were saved as a pcap file from Wireshark, then uploaded to a program called *A-packets*. This program analyses the different protocols used.

### 6.1.3 Results

**Is any data transmitted in clear text?** Yes, communication via both HTTP and HTTPS was found and, as mentioned earlier, HTTP communication lack encryption. By analyzing how the device communicates with the server, it became clear that the device utilizes *ephemeral/dynamically allocated* ports. The ports used were in the uppermost range of ports and the system uses a Linux kernel.

**Are any old or weak cryptographic algorithms used either by default or in older code?** No, TLS v1.2 is used for HTTPS communications.

### 6.1.4 Discussion

Having the communication unencrypted leaves the system vulnerable to MITM attacks. When performing sensitive interactions with the system, such as logging in or registering a new user, the TLS 1.2 encryption protected the data. While the version of TLS is not the latest, TLS v1.2 is considered to be secure. As the fridge was up to date, communication of the fridge updating its firmware was unattainable. However, other than that, the system properly encrypted sensitive information and was considered safe against sniffing attacks.

As mentioned, the ports used for the communication to the server are *ephemeral* ports, meaning that the device does not receive any types of request but rather uses *polling* to update its current state. This implies that the device does not get called by any server but rather utilizes self initiated connections.

Furthermore, the server site that the fridge communicates via does not have a valid SSL certificate. The site is available over HTTP without TLS encryption. This accessibility can potentially make the system vulnerable to MITM attacks to map the communications between the device (Fridge) and the back-end API.

## 6.2 Test 2: Security Misconfiguration (DoS)


### 6.2.1 Background

When scanning the device with Nmap multiple open ports were found. All of those ports were in the uppermost range of the ports, meaning they were *ephemeral* ports, except for port 17654/TCP as mentioned in Section 5.3. These kinds of ports could be vulnerable to *TCP SYN flood* attacks.

A *TCP SYN flood* attack takes advantage of the TCP protocol described in Section 2.7.3. To create a denial of service, an attacker takes advantage of the fact that the server will respond with one or more SYN / ACK packets upon receipt of an initial SYN packet, waiting for the last step to establish communication. The attacker sends a large volume of SYN packets to the targeted server, often with spoofed IP addresses. The server then responds to each connection request and leaves a port open, ready to respond. While the server waits for the last ACK packet, which never arrives, the attacker sends new SYN packets. The arrival of each new SYN packet causes the server to hold a new open port connection temporarily. After all available ports have exhausted, the server can no longer function effectively.

### 6.2.2 Method

The tool used for the *flood* attack is *hping3*. Port 17654/TCP was targeted with *hping3* as demonstrated in Figure 6.2 . While the attack was performed, the target machine was monitored by being pinged via the machine's public IP address (not within the local network). By pinging the machine, we can monitor the response time. An increase in the ping response time would indicate that the attack is exhausting in the system. Also, during the attack, attempts to interact with the target machine were done to witness the potential impact.



```
→ sudo hping3 --syn --rand-source --flood -p 17654 194.168.53.134
```

Figure 6.2: hpin3 command

### 6.2.3 Results

As the *flooding* attack began, a significant increase in the *ping* response time could be observed.

```

→ ~ ping 130.237.53.252
PING 130.237.53.252 (130.237.53.252) 56(84) bytes of data.
64 bytes from 130.237.53.252: icmp_seq=1 ttl=51 time=25.8 ms
64 bytes from 130.237.53.252: icmp_seq=2 ttl=51 time=31.8 ms
64 bytes from 130.237.53.252: icmp_seq=3 ttl=51 time=32.2 ms
64 bytes from 130.237.53.252: icmp_seq=4 ttl=51 time=27.6 ms
64 bytes from 130.237.53.252: icmp_seq=5 ttl=51 time=31.3 ms
64 bytes from 130.237.53.252: icmp_seq=57 ttl=51 time=42.6 ms
64 bytes from 130.237.53.252: icmp_seq=61 ttl=51 time=59.0 ms
64 bytes from 130.237.53.252: icmp_seq=66 ttl=51 time=140 ms
64 bytes from 130.237.53.252: icmp_seq=133 ttl=51 time=51.5 ms
64 bytes from 130.237.53.252: icmp_seq=136 ttl=51 time=41.0 ms
64 bytes from 130.237.53.252: icmp_seq=153 ttl=51 time=49.1 ms
64 bytes from 130.237.53.252: icmp_seq=154 ttl=51 time=29.8 ms
64 bytes from 130.237.53.252: icmp_seq=155 ttl=51 time=36.2 ms
64 bytes from 130.237.53.252: icmp_seq=156 ttl=51 time=45.5 ms
64 bytes from 130.237.53.252: icmp_seq=157 ttl=51 time=41.2 ms
64 bytes from 130.237.53.252: icmp_seq=158 ttl=51 time=32.6 ms
64 bytes from 130.237.53.252: icmp_seq=159 ttl=51 time=22.6 ms
64 bytes from 130.237.53.252: icmp_seq=160 ttl=51 time=30.1 ms
64 bytes from 130.237.53.252: icmp_seq=161 ttl=51 time=36.0 ms
64 bytes from 130.237.53.252: icmp_seq=162 ttl=51 time=28.4 ms
64 bytes from 130.237.53.252: icmp_seq=163 ttl=51 time=33.7 ms
64 bytes from 130.237.53.252: icmp_seq=164 ttl=51 time=34.8 ms
64 bytes from 130.237.53.252: icmp_seq=165 ttl=51 time=32.9 ms
64 bytes from 130.237.53.252: icmp_seq=166 ttl=51 time=39.4 ms
64 bytes from 130.237.53.252: icmp_seq=167 ttl=51 time=59.8 ms
64 bytes from 130.237.53.252: icmp_seq=168 ttl=51 time=38.1 ms
64 bytes from 130.237.53.252: icmp_seq=169 ttl=51 time=56.4 ms
64 bytes from 130.237.53.252: icmp_seq=170 ttl=51 time=33.8 ms
64 bytes from 130.237.53.252: icmp_seq=171 ttl=51 time=29.9 ms
64 bytes from 130.237.53.252: icmp_seq=172 ttl=51 time=33.0 ms

```

Figure 6.3: *ICPM* response times of the target device.

As observed in Figure 6.3, the response time *spiked* at some points, we can even observe timeouts by examine the `icmp_seq` column, it makes big jumps, e.g. from `icmp_seq = 5` to `icmp_seq = 57`. This yielded in slow response times, and legitimate request got delayed responses.

### 6.2.4 Discussion

It is essential to notice that although the response times and legitimate requests from the target device got delayed responses, the target device did not seize to respond, and the target device did not suffer from significant repercussions - e.g. total freeze or reboot. Although the consequences of the attack were not a significant concern, it is recommended to add congestion monitoring

mechanisms such as firewalls to mitigate this aspect further.

## 6.3 Test 3: DNS Response Flooding (DoS)

This exploit is based upon the Nessus vulnerability scan performed on the target device from within the same network (WLAN).

### 6.3.1 Background

MEDIUM

Multiple Vendor DNS Response Flooding Denial Of Service

>

**Description**

The remote DNS server is vulnerable to a denial of service attack because it replies to DNS responses.

An attacker could exploit this vulnerability by spoofing a DNS packet so that it appears to come from 127.0.0.1 and make the remote DNS server enter into an infinite loop, therefore denying service to legitimate users.

**Solution**

Contact the vendor for an appropriate upgrade.

**See Also**

<http://www.nessus.org/u?a04dcb96>

**Output**

```
Nessus sent the following response data :
0x00: F9 E1 81 80 00 01 00 00 00 01 00 00 03 77 77 77 .....www
0x10: 06 67 6F 6F 67 6C 65 03 63 6F 6D 00 00 10 00 01 .....google.com....
0x20: C0 10 00 06 00 01 00 00 00 3C 00 26 03 6E 73 31 .....<.s.nsl
0x30: C0 10 09 64 6E 73 2D 61 64 6D 69 6E C0 10 16 40 .....dns-admin...@
0x40: 93 90 00 00 03 84 00 00 03 84 00 00 07 08 00 00 .....
0x50: 00 3C .....<

And the DNS server replied with the following response :
0x00: F9 E1 81 82 00 01 00 00 00 00 00 00 03 77 77 77 .....www
0x10: 06 67 6F 6F 67 6C 65 03 63 6F 6D 00 00 10 00 01 .....google.com....
0x20: C0 10 00 06 00 01 00 00 00 3C 00 26 03 6E 73 31 .....<.s.nsl
0x30: C0 10 09 64 6E 73 2D 61 64 6D 69 6E C0 10 16 40 .....dns-admin...@
0x40: 93 90 00 00 03 84 00 00 03 84 00 00 07 08 00 00 .....
0x50: 00 3C .....<
```

Port	Hosts
53 / udp / dns	192.168.53.134

Figure 6.4: Nessus- Basic Network Scan

The Domain Name System (DNS) can be seen as the *phonebook* of the Internet. When a device searches for the domains, e.g. google.com, through a web browser, DNS then converts the domain name into an IP address, (google.com = 8.8.8.8), so that the browser can proceed to load Internet resources. A DNS attack is an exploit in which a hacker takes advantage of the vulnerabilities in the domain name system (DNS).

The Nessus scan results flagged for potential DNS server vulnerabilities since the server replies to DNS responses. This makes it possible for an attacker to exploit this vulnerability by *spoofing* a DNS packet so that the packet source appears to be from the device itself (IP 127.0.0.1). By doing this,

the attacker could potentially force the DNS server to enter an *infinite loop* and therefore exhausting the server resulting in legitimate requests being denied, thus resulting in a denial of service attack.

### 6.3.2 Method

To reaffirm the findings from the Nessus scan, Nmap was used to find and analyze the open port (port 53/UDP) flagged by the Nessus scan.

For further testing we used the tool *Scapy* to do a simple DNS packet query. While sending the query, the traffic was captured to later be examined and visualized using Wireshark. Then the exploit code shown in listing 6.1 was used to send recursive random queries to the target device in the attempt to exhaust the service.

Listing 6.1: PoC Exploit code

```
import sys
from scapy.all import *

top_level = ".ch"
domain = "192.168.53.134" # enter the DNS server IP
cnt = 10000 # enter the count of request to be send
dns_server = "192.168.53.134" # enter the same DNS server IP
test = "." + domain + top_level
for i in range(0,cnt):
    s = RandString(RandNum(1,8)).decode("utf-8")
    s1 = s.lower()
    q = s1 + test
    #print (i,q)
    sr1(IP(dst=dns_server)/UDP(sport=RandShort())/DNS(rd=1,qd = DNS
```

### 6.3.3 Results

Image 6.5 shows the nmap scan of the port 53, the port is shown to be closed, the scan does show that the port 53 is in fact running DNS service.

```

→ ~ sudo nmap -sV --script vuln -p53 192.168.53.134

Starting Nmap 7.60 ( https://nmap.org ) at 2021-05-18 17:45 CEST
Nmap scan report for 192.168.53.134
Host is up (0.067s latency).

PORT      STATE SERVICE VERSION
53/tcp    closed domain
MAC Address: 88:57:1D:02:1C:0C (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.20 seconds

```

Figure 6.5: Nmap scan of port 53

```

→ ~ sudo scapy3
INFO: Can't import matplotlib. Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: IPython not available. Using standard Python shell instead.
AutoCompletion, History are disabled.

aSPY//YASa
apyyyyCY////////YCa
sY////////YSpcs scpCY//Pp
ayp ayyyyyySCP//Pp syY//C
AYAsAYYYYYYYY//Ps cY//S
pCCCCY//p cSSps y//Y
SPPPP//a pP//AC//Y
A//A cyP//C
p//Ac sC//a
P//Y/Cpc A//A
sccccp//pSP//p p//Y
sY////////y caa S//P
cayCyayP//Ya pY/Ya
sY/PsY//Y/Cc aC//Yp
sc sccaCY//PCypaapyCP//YSs
spCPY////////YPSps
ccaacs

Welcome to Scapy
Version 2.4.5
https://github.com/secdev/scapy
Have fun!
Craft me if you can.
-- IPv6 layer

>>> res = sr1(IP(dst="192.168.53.134")/DNS(rd=1,qd=DNSQR(qname="www.google.com")))
Begin emission:
Finished sending 1 packets.
.*
Received 2 packets, got 1 answers, remaining 0 packets

```

Figure 6.6: Scapy - DNS query

The above image shows a simple DNS packet with the query "www.google.com". This packet was then monitored using Wireshark as seen in image 6.7.



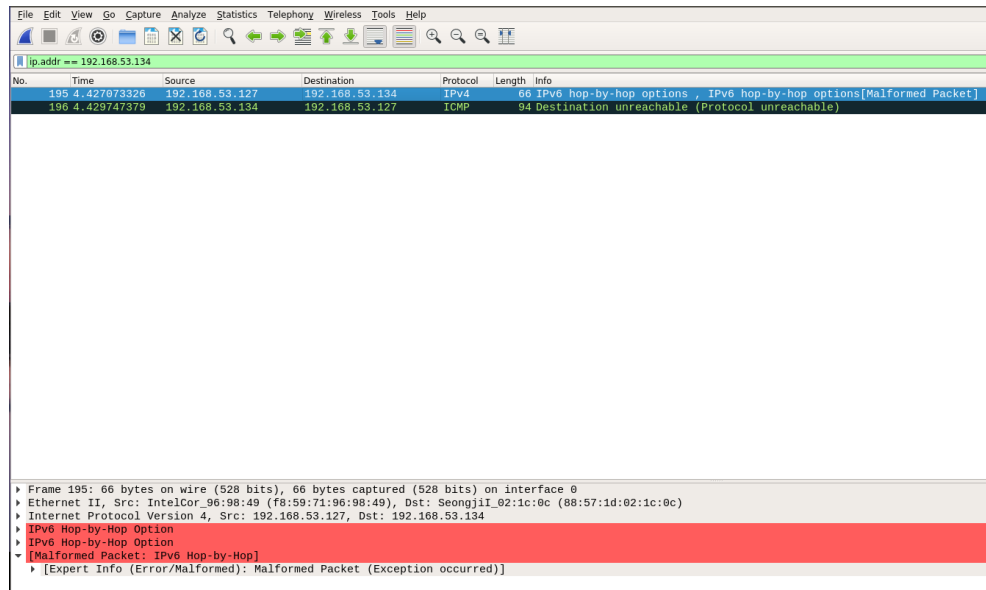
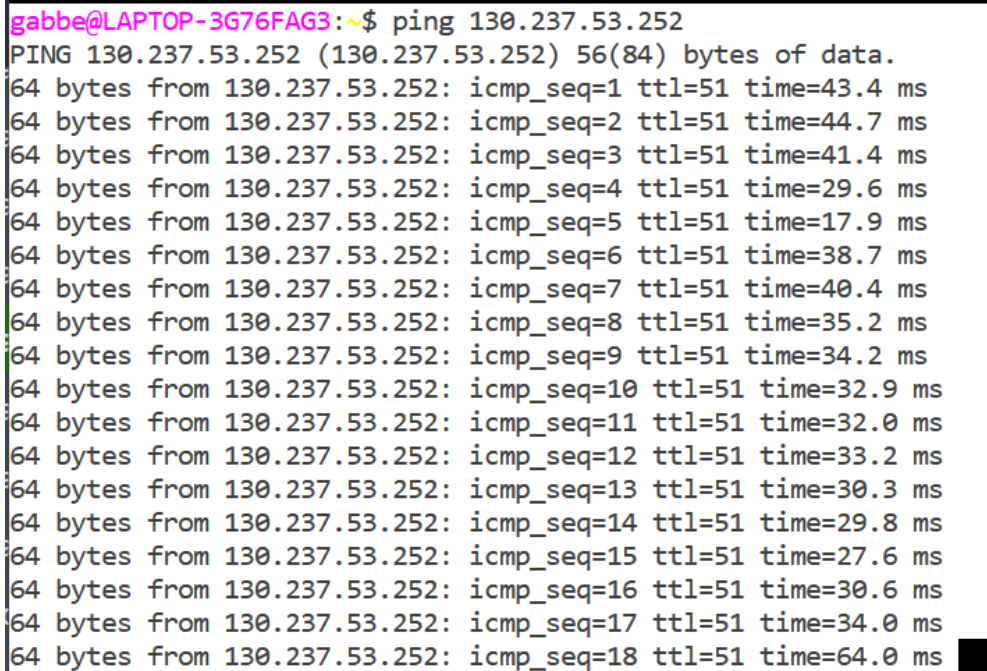


Figure 6.7: Wireshark monitoring of DNS packet

We can see that the server target device responds with the same query "www.google.com".

The exploit code was executed, and queries were sent and monitored via Wireshark. The target device response times were monitored via ping. It can be observed on the ping that the device was not affected by the attack, and the response times were the same before and during the attack.

A terminal window with a black background and white text. The prompt is 'gabbe@LAPTOP-3G76FAG3:~\$'. The command entered is 'ping 130.237.53.252'. The output shows a successful ping with 56(84) bytes of data. It then lists 18 responses, each showing '64 bytes from 130.237.53.252: icmp\_seq=X ttl=51 time=Y ms' where X ranges from 1 to 18 and Y shows varying round-trip times.

```
gabbe@LAPTOP-3G76FAG3:~$ ping 130.237.53.252
PING 130.237.53.252 (130.237.53.252) 56(84) bytes of data.
64 bytes from 130.237.53.252: icmp_seq=1 ttl=51 time=43.4 ms
64 bytes from 130.237.53.252: icmp_seq=2 ttl=51 time=44.7 ms
64 bytes from 130.237.53.252: icmp_seq=3 ttl=51 time=41.4 ms
64 bytes from 130.237.53.252: icmp_seq=4 ttl=51 time=29.6 ms
64 bytes from 130.237.53.252: icmp_seq=5 ttl=51 time=17.9 ms
64 bytes from 130.237.53.252: icmp_seq=6 ttl=51 time=38.7 ms
64 bytes from 130.237.53.252: icmp_seq=7 ttl=51 time=40.4 ms
64 bytes from 130.237.53.252: icmp_seq=8 ttl=51 time=35.2 ms
64 bytes from 130.237.53.252: icmp_seq=9 ttl=51 time=34.2 ms
64 bytes from 130.237.53.252: icmp_seq=10 ttl=51 time=32.9 ms
64 bytes from 130.237.53.252: icmp_seq=11 ttl=51 time=32.0 ms
64 bytes from 130.237.53.252: icmp_seq=12 ttl=51 time=33.2 ms
64 bytes from 130.237.53.252: icmp_seq=13 ttl=51 time=30.3 ms
64 bytes from 130.237.53.252: icmp_seq=14 ttl=51 time=29.8 ms
64 bytes from 130.237.53.252: icmp_seq=15 ttl=51 time=27.6 ms
64 bytes from 130.237.53.252: icmp_seq=16 ttl=51 time=30.6 ms
64 bytes from 130.237.53.252: icmp_seq=17 ttl=51 time=34.0 ms
64 bytes from 130.237.53.252: icmp_seq=18 ttl=51 time=64.0 ms
```

Figure 6.8: Ping of the device under attack from outside the local network

### 6.3.4 Discussion

As can be seen in figure 6.8, the device showed no impact as a result of the attack. Thus it is safe to assume that the device has countermeasures implemented and is secure against this kind of attack.

## 6.4 Test 4: Broken Authentication

This exploit comes from the OWASP top ten.

### 6.4.1 Background

A natural approach for malicious attackers are attacks on session management. This is due to the current nature of access control in general. The implementation of access control is difficult and attackers can use libraries with a multitude of common username and password combinations, as well as tools for automated brute force attacks. Furthermore, it is not uncommon for users to compromise their credentials in one way or another.

### 6.4.2 Method

In order to log on to the SmartThings app, one must use a Samsung account. Thus it was the Samsung login page that was examined. The OWASP top ten presents nine questions for evaluation of a system's vulnerability to broken authentication attacks [36]. From these, the following were selected for investigation.

1. Does the application allow for automated attacks?
2. Does the application allow default, weak, or well-known passwords?
3. Does the application use weak password recovery processes, such as knowledge based answers?
4. Does the application make use of multi-factor authentication?
5. Does the application expose session IDs in the URL?
6. Does the application rotate session IDs after successful login?

### 6.4.3 Results

The following results were observed.

**Does the application allow for automated attacks?** After attempting to log in with the wrong password seven times in one session, the account was locked. In order to regain access to the account, the password was reset. After gaining access to the account once more, another attempt was made. After entering the wrong password another four times in a row, the account was this time locked for 30 minutes, and the application gave the message "sign-in limit reached". Therefore the application is protected against automated attacks such as brute force.

**Does the application allow default, weak, or well-known passwords?** As the device itself is not password locked, there are no default passwords to the application, instead a password must be chosen upon account creation. The password must fulfill certain criteria in order to be accepted as a valid password. The criteria are the following:

1. The password must be at least 8 characters long.

2. The password must contain at least one of the following:

- A letter.
- A number.
- A special case letter.

The application can therefore be considered to have requirements that ensure a secure password in accordance with NIST 800-63 B's guidelines for Memorized Secrets[37]. However the NIST guidelines also state that passwords chosen by a user should be compared to a "black list" of weak passwords such as "Password1!". A new account was created using the password "password1!" successfully. Thus the application was assumed to not have any black listed passwords and any password fulfilling the aforementioned criteria acceptable.

**Does the application make use of multi-factor authentication?** The application supports two-step authentication. Users are able to activate two-step authentication if wanted, it is however not obligatory. The two-step authentication uses either email or a phone as the second step.

**Does the application use weak password recovery processes, such as knowledge based answers?** The application uses an email address and a password to identify a user. In order to recover the password, a user must enter the email address associated with the account, upon which a password reset link will be sent to that email. Should a user forget their email address, this is also recoverable. In order to recover the email associated with the account, the user must enter their full name and date of birth. The application responds with the first three letters of the email, followed by asterisks instead of the remaining letter. Thus should the email be "HelloWorld@gmail.com", the application would give "Hel\*\*\*\*\*@gmail.com". The number of asterisks correctly represents the number of remaining letters. Thus the application does not use knowledge based answers for the password recovery process.

**Does the application expose session IDs in the URL?** The application does not expose the session IDs in the URL.

**Does the application rotate session IDs after successful login?** The application uses different session IDs for the login page and actual application. When first accessing the login page, session IDs are saved as cookies. Upon

successful login, different session IDs are saved as cookies. When the user logs out and is sent to the login screen, a new session ID is used. When a new user logs in, the session IDs is again replaced with a new one. Thus the application rotates session IDs after successful login.

#### **6.4.4 Discussion**

The tests showed that the application was protected and upon brute force attempts, locked the account. The application fulfilled OWASPs recommendations for security against attacks on broken authentication. The only recommendation that was not fulfilled was to prohibit use of common or simple passwords, such as "password1!". However attempting to gain access by using a library of commonly used passwords has a low probability of success due to the limit of login attempts before the account being locked.

# Chapter 7

## Results

This chapter presents the findings of this thesis documented in a threat traceability matrix. The traceability matrix can be found in figure 7.1 and contains the summarized results for all performed penetration tests.

Samsung Smart Fridge						
Surface	Component	Attack	Attack/ vulnerability family	Method (exploit/payload/process)	Result	Notes
Network	PCAP	Sensitive Data Exposure	Sensitive Data Exposure	ARP Poisoning	All data encrypted	
Network	Open TCP port	Denial Of Service	Denial Of Services	TCP SYN Flood	Little to no impact	
Network	DNS service	Spoofing Attack	Denial Of Services	DNS Response Flooding	No impact	
Cloud	Login Page	Broken Authentication	Broken Authentication	Checking if OWASP standards apply	Standards applied	
Cloud	Password Retrieval Page	Broken Authentication	Broken Authentication	Checking if OWASP standards apply	Standards applied	
Cloud	Samsung Account	Broken Authentication	Broken Authentication	Checking if OWASP standards apply	Most standards applied	Common passwords allowed

Figure 7.1: A threat traceability matrix illustrating our results

# Chapter 8

## Discussion

In the following section, the reliability, validity, and generalizability of the findings are debated, and a discussion on the security of the tested device alongside the answer to the research question previously stated in the scope of the thesis.

### 8.1 Methodology

The methodology followed gave a clear structure to the pentest and was reasonably easy to follow along. However, the methodology spent a significant amount of time on threat modeling. With none of us having prior experience with threat modeling, it became easy to get stuck in details, leading to less time spent on other aspects, such as the exploitation phase, where some tasks could have been examined more in-depth. Despite these limitations, enough vulnerabilities were properly tested and documented in order to motivate a conclusion.

### 8.2 Results

Based on both the information gathered to create a valid threat model and the results from each task derived from said threat model, the system's security can be considered secure in the scope of this thesis. However, this thesis did not address all entry points or potential threats due to the different limitations mentioned throughout the report.

The first performed task, section 6.1, showed that the system properly encrypts user data when communicating with other sources. In doing so, it successfully protects said data from being disclosed by any unauthorized third

party. Furthermore, the encryption protocol used is up to date. Thus it is improbable that a third party can decrypt the data.

The following test, section 6.2, showed that the system under consideration is overall safe against Denial of Service (DoS) attacks on ports examined in this thesis. The attempted attack did not have any significant impact on the system. However, the manufacturer can improve this aspect of the system by, e.g., adding rate limitations.

The third attempted attack, section 6.3, was a DNS Response Flooding attack, based on a Nessus scan's findings. Even though the vulnerability scan by Nessus showed the system to be vulnerable against DNS Response Flooding, after further investigation and attempted exploitation, the system proved to be secure against such attacks.

Lastly, the systems security measures against attacks trying to access authorized users accounts was analyzed, section 6.4. The analysis was based on OWASPs criteria of a system secure against Broken Authentication. The system fulfilled all but one of the examined criteria and was found to be secure against automated attacks such as brute force attempts. The one criteria not fulfilled was a criteria stating not to allow simple passwords such as "password1!". This is primarily exploitable when paired with an automated attack using a database of common and simple passwords. As such attacks are protected against by a limit to how many login attempts are allowed, failing this criteria was not assessed as a vulnerability.

## 8.3 Sustainability and Ethics

The methodology followed throughout the thesis was based upon Ethical hacking guides combined with the Universities guidelines, which follows ethical and lawful guidelines. Although some of the findings discussed in section 6.1.4 might suggest the possibility for further vulnerabilities and exploitation of the target device, Samsung has earlier taken immediate mitigating actions whenever vulnerabilities have surfaced, as suggested under section 1.5. These mitigation actions have prevented exploitation's of Samsung devices that otherwise would have caused potential societal and or economical damage, e.g. if an exploit led to a device used in a larger botnet attack such as The Mirai Malware Attack[38]. Also we strictly followed Swedish law and legislation that states, that it is illegal to hack something without permission.<sup>1</sup> As IoT de-

---

<sup>1</sup>[https://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/brottsbalk-1962700\\_sfs-1962-700](https://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/brottsbalk-1962700_sfs-1962-700)



vices are becoming a big part of our societal development, the security of said devices is essential to maintain societal and economic sustainability. Maintaining the security of IoT devices decreases the risk of their recruitment to botnets and potential data disclosure.

# Chapter 9

## Conclusions and Future Work

### 9.1 Conclusion

Granting the security surrounding an IoT device not easy, but it is crucial. At first glance, a Smart Refrigerator may seem harmless, but it may take part in bigger cybercrimes and attacks if exploited in the right way by the wrong hands. Furthermore, to answer the research question, the penetration test and vulnerability assessment conducted in this thesis concluded that the system under consideration was secure in the scope of this thesis. However, potential vulnerabilities that require further investigation were discovered.

### 9.2 Future Work

While gathering information about the target device, we found that the server, which the device communicates for state updates, is missing or has expired SSL/TLS certificates. Thus, the device may not validate the certificates, thus making the device vulnerable to Man-in-The-Middle attacks.

Another approach that should be considered is investigating how the device interacts with other Samsung smart devices, as these can be linked together. Connecting the fridge with other devices, such as smart cameras allows the user to access these devices through the hub on the fridge. This aspect was not explored due to the lack of such devices. This could potentially open up to several new exploits in the case of a discovered vulnerability.

Finally, not all entry points were investigated in this thesis. Thus future work may be interested in exploring other entry points.

# Bibliography

- [1] I. of Things (IoT) connected devices installed base worldwide from 2015 to 2025. (2021), [Online]. Available: %5Curl%7Bhttps://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/%7D (visited on 05/19/2021).
- [2] D. Eidenskog and F. Kamrani, "Internet of Things En IT-säkerhets mardröm. (Swedish)", *Strategisk Utblick 7: Närområdet och nationell säkerhet*, pp. 57–63, 2017. doi: [http://goteborgsforsvar.se/wp-content/uploads/2017/10/http-webbrapp.ptn\\_.foi\\_.se-pdf-c602efd1-93c9-4f2e-88a6-b7ad648561d6.pdf](http://goteborgsforsvar.se/wp-content/uploads/2017/10/http-webbrapp.ptn_.foi_.se-pdf-c602efd1-93c9-4f2e-88a6-b7ad648561d6.pdf).
- [3] F-Secure, "Attack landscape h2 2019: An unprecedented year for cyber attacks", 2020. doi: <https://blog-assets.f-secure.com/wp-content/uploads/2020/03/04101313/attack-landscape-h22019-final.pdf>.
- [4] W. Xiong and R. Lagerström, "Threat modeling – a systematic literature review", *Computers Security*, vol. 84, pp. 53–69, 2019, issn: 0167-4048. doi: <https://doi.org/10.1016/j.cose.2019.03.010>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167404818307478>.
- [5] P. Venda, *Hacking defcon 23's iot village samsung fridge*, Aug. 2015. [Online]. Available: <https://www.pentestpartners.com/security-blog/hacking-defcon-23s-iot-village-samsung-fridge>.
- [6] A. Abraham, "Hacking Tizen: The OS of Everything.", 2015. doi: <https://conference.hitb.org/hitbsecconf2015ams/wp-content/uploads/2015/02/WHITEPAPER-Hacking-Tizen-The-OS-of-Everything.pdf>.

- [7] “Common vulnerabilities and exposures”, Jan. 2017. [Online]. Available: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-17692>.
- [8] Oracle, *What is the internet of things (iot)?*, Jan. 2020. [Online]. Available: <https://www.oracle.com/internet-of-things/what-is-iot.html>.
- [9] L. S. Vailshery, *Number of iot devices 2015-2025*, Nov. 2016. [Online]. Available: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>.
- [10] A. Fernandez, *New iot security regulations: What you need to know*, Feb. 2020. [Online]. Available: <https://securityboulevard.com/2020/01/new-iot-security-regulations-what-you-need-to-know-2/>.
- [11] P. W. Singer and A. Friedman, in *Cybersecurity and cyberwar: what everyone needs to know*. Oxford University Press, 2014.
- [12] *Digital Spillover Measuring the true impact of the Digital Economy*. Oxford Economics, Huawei, 2017. [Online]. Available: <https://www.huawei.com/minisite/gci/en/digital-spillover/index.html>.
- [13] J. Lewis, *Economic Impact of Cybercrime-No Slowing Down*. Feb. 2018. [Online]. Available: <https://csis-website-prod.s3.amazonaws.com/s3fs-public/publication/economic-impact-cybercrime.pdf>.
- [14] D. Kushner, *The real story of stuxnet*, 2013. [Online]. Available: <https://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet>.
- [15] S. Patil, A. Jangra, M. Bhale, A. Raina, and P. Kulkarni, “Ethical hacking: The need for cyber security”, in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, 2017, pp. 1602–1606. doi: 10.1109/ICPCSI.2017.8391982.
- [16] A. Gupta, in *The Iot hacker’s handbook: a practical guide to hacking the internet of things*. Apress, 2019, pp. 20–21. [Online]. Available: [http://www.ime.cas.cn/icac/learning/learning\\_3/201907/P020190724586712846107.pdf](http://www.ime.cas.cn/icac/learning/learning_3/201907/P020190724586712846107.pdf).

- [17] H. Poston, *What are black box, grey box, and white box penetration testing?*, May 2021. [Online]. Available: <https://resources.infosecinstitute.com/topic/what-are-black-box-grey-box-and-white-box-penetration-testing/>.
- [18] *Cybersäkerhet i sverige – hot, metoder, brister och beroenden 2020 (Swedish)*. 2020. [Online]. Available: <https://www.sakerhetspolisen.se/download/18.f2735ce171767402ba202/1591164566288/Rapport-Cybersakerhet-Hot-Metoder-Brister.pdf>.
- [19] *Cybersäkerhet i Sverige – Rekommenderade säkerhetsåtgärder 2020 (Swedish)*. 2020. [Online]. Available: <https://www.msb.se/contentassets/fe72c449466e4017bd76787762ab9dc5/rapport-cybersakerhet-i-sverige-2020---rekommenderade-sakerhetsatgarder.pdf>.
- [20] N. Shevchenko, T. A. Chick, P. O’Riordan, T. Patrick Scanlon, and C. Woody, *shevchenko\_chick\_riordan\_patrickscanlon\_woody*2018. 2018. [Online]. Available: <https://apps.dtic.mil/sti/pdfs/AD1084024.pdf>.
- [21] B. A. Forouzan and S. C. Fegan, *Data communications and networking*. McGraw-Hill Higher Education, 2007.
- [22] R. T. Braden, *Requirements for Internet Hosts - Communication Layers*, RFC 1122, Oct. 1989. doi: 10.17487/RFC1122. [Online]. Available: <https://rfc-editor.org/rfc/rfc1122.txt>.
- [23] H. Nielsen, J. Mogul, L. M. Masinter, R. T. Fielding, J. Gettys, P. J. Leach, and T. Berners-Lee, *Hypertext Transfer Protocol – HTTP/1.1*, RFC 2616, Jun. 1999. doi: 10.17487/RFC2616. [Online]. Available: <https://rfc-editor.org/rfc/rfc2616.txt>.
- [24] S. S. Team, *What is ssl?*, Apr. 2021. [Online]. Available: <https://www.ssl.com/faqs/faq-what-is-ssl/>.
- [25] KTH, *Hacking guides*. [Online]. Available: [https://nse.digital/pages/guides/hacking\\_guides.html](https://nse.digital/pages/guides/hacking_guides.html).
- [26] *Nmap security scanner*, 2020. [Online]. Available: <https://nmap.org/>.
- [27] *Ettercap*. [Online]. Available: <https://www.ettercap-project.org/>.

- [28] Wireshark, *What is wireshark?* [Online]. Available: [https://www.wireshark.org/docs/wsug\\_html\\_chunked/ChapterIntroduction.html#ChIntroWhatIs](https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroWhatIs).
- [29] *Nessus product family*, May 2021. [Online]. Available: <https://www.tenable.com/products/nessus>.
- [30] M. Silverio-Fernández, S. Renukappa, and S. Suresh, *What is a smart device? - a conceptualisation within the paradigm of the internet of things*, May 2018. [Online]. Available: <https://doi.org/10.1186/s40327-018-0063-8>.
- [31] *Tizen operating system*. [Online]. Available: <https://www.tizen.org/about>.
- [32] S. Saxena. [Online]. Available: <https://www.tizen.org/sites/default/files/tizen-architecture-linuxcollab.pdf>.
- [33] K. D. Community, *Smack*. [Online]. Available: <https://www.kernel.org/doc/html/v4.14/admin-guide/LSM/Smack.html>.
- [34] *Web runtime*, 2021. [Online]. Available: <https://docs.tizen.org/application/web/tutorials/web-runtime/>.
- [35] Samsung, *Samsung refrigerator manual*, May 2018.
- [36] *A2:2017-broken authentication*, 2017. [Online]. Available: [https://owasp.org/www-project-top-ten/2017/A2\\_2017-Broken\\_Authentication](https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication).
- [37] N. Lefkovitz and J. Danker, *Nist special publication 800-63b*, Jun. 2017. [Online]. Available: <https://pages.nist.gov/800-63-3/sp800-63b.html#appA>.
- [38] J. Biggs, *Hackers release source code for a powerful ddos app called mirai*, Oct. 2016. [Online]. Available: [https://techcrunch.com/2016/10/10/hackers-release-source-code-for-a-powerful-ddos-app-called-mirai/?guccounter=1&guce\\_referrer=aHR0cHM6Ly91bi53aWtpcGVkaWEub3JnLw&guce\\_referrer\\_sig=AQAAACRH49rCM1CnQ1r3K56X-VIqqK2hgYKg1\\_E9Lx72yUwG8LVXqH\\_FopdJzrJTWrpbslDnKJKiKfV0rBxh0fQM2Iw-oHgSlzla9lMFVHvBSU0ohwiAjGRvOSkEo4IYrgGZj2k6W](https://techcrunch.com/2016/10/10/hackers-release-source-code-for-a-powerful-ddos-app-called-mirai/?guccounter=1&guce_referrer=aHR0cHM6Ly91bi53aWtpcGVkaWEub3JnLw&guce_referrer_sig=AQAAACRH49rCM1CnQ1r3K56X-VIqqK2hgYKg1_E9Lx72yUwG8LVXqH_FopdJzrJTWrpbslDnKJKiKfV0rBxh0fQM2Iw-oHgSlzla9lMFVHvBSU0ohwiAjGRvOSkEo4IYrgGZj2k6W).



