# **Real-Time Internet of Things Security**

## (RIoTS)



By Eman Fatima Asifa Iqbal Shaheer Ahmad M Mustafa Zahoor

## Project Supervisor: Asst. Prof. Waleed Bin Shahid

Submitted to the faculty of Department of Electrical Engineering, Military College of Signals, National University of Sciences and Technology, Islamabad, in partial fulfillment for the requirements of B.E Degree in Electrical (Telecom) Engineering. June 2022 In the name of ALLAH, the Most benevolent, the Most Courteous

## **CERTIFICATE OF CORRECTNESS AND APPROVAL**

This is to officially state that the thesis work contained in this report

"Real-Time Internet of Things Security"

is carried out by

#### Eman Fatima, Asifa Iqbal, Shaheer Ahmad and M Mustafa Zahoor

under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Electrical (Telecom.) Engineering in Military College of Signals, National

University of Sciences and Technology (NUST), Islamabad.

Approved by

Supervisor Asst. Prof. Waleed Bin Shahid Department of IS, MCS

Date: \_\_\_\_\_

## **DECLARATION OF ORIGINALITY**

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

## ACKNOWLEDGEMENTS

Allah Subhan'Wa'Tala is the sole guidance in all domains.

Our parents, colleagues and most of all supervisor, Asst. Prof. Waleed Bin Shahid without your

guidance.

The group members, who through all adversities worked steadfastly.

## Plagiarism Certificate (Turnitin Report)

This thesis has 13% similarity index. Turnitin report endorsed by Supervisor is attached.

Eman Fatima 00000261278

Asifa Iqbal 00000284103

Shaheer Ahmad 00000278791

M Mustafa Zahoor 00000249296

Signature of Supervisor

## Abstract

The field of Internet of things and worldwide real time information exchange is gradually increasing. IoT devices and services are becoming more widely available. Threats and assaults against IoT devices and services are also on the rise as a result of their success. In the IoT world, cyber-attacks are nothing new, but as the IoT gets more deeply embedded in our lives and communities, it will be critical to step up and take cyber defense seriously.

Our safeguard everyday IoT devices being used by project aims to society. RIOTS will be installed at the gateway, allowing all data exchanged and created by smart IoT devices to pass through. Using pre-trained machine learning algorithms, it would be able to detect unusual entrance and egress traffic, as well as cyber-attacks like analyzing what information goes out of the network, and it captures the traffic and analysis that, sees if there are weak passwords that can be compromised, it checks whether devices are vulnerable to DoS attacks, whether remote code can be executed over them, whether the communication they are using with the rest of the internet is unencrypted or not, if information is being leaked and whether there are configuration issues that the attacker can exploit because even one of these internal IoT device is compromised then it will act as a pivot point to launch wider range of attacks on rest of the IoT network. By securing smart IoT devices, RIoTS can help assure data privacy and security.

*Keywords:* Internet of things, real time information, data privacy and security, *Cyber-attacks.* 

# **Table of Contents**

Chapter 1: Introduction	1
1.1 Chapter Overview	
1.2 Problem Statement	2
1.3 Objectives	4
1.4 Justification for the Selection of topic	6
1.5 Scope	6
1.6 Contributions	7
1.7 Chapter Summary	7
Chapter 2: Literature Review	
2.1. Chapter Overview	
2.2. Project Domain	
2.3. Background	
2.4. Literature Review	
2.4.1 Threat Modelling of Smart Light Bulb	
2.4.2 Smart Bulb using IoT	
2.4.3 Taxonomy of Security Attacks	
2.5. Research Based Results	
2.5.1 IP Camera	
2.5.1.1 Security Analysis	
2.5.2 Nest Thermostat	
2.5.3 Nike Feulband	
2.5.3.1 Vulnerabilities	15
2.5.3.2 Firmware Testing	15
2.5.3.3 Results	16
2.6. Conjecture	
2.7. Chapter Summary	

Chapter 3: Technical Specification	
3.1. Chapter Overview	
3.2. Network Diagram	
3.3. Methodology	
3.4. Stage 1: Hardware Procurement	21
3.4.1 Xiaomi Smart Bulb	21
3.4.2 IP Smart Camera	
3.4.3 Raspberry pi	
3.5. Chapter Summary	
1 2	

Chapter 4:Proposed Solution	25
4.1. Chapter Overview	26
4.2. Stage 2: Launching Attacks	26
4.2.1 Xiaomi Smart Bulb	27
4.2.1.1 ARP Poisoning	27
4.2.1.2 Traffic Analysis	30
4.2.2 IP Smart Camera	32
4.2.2.1 ARP Poisoning	32
4.2.2.2 Traffic Analysis	35
4.2.2.3 Password Extraction	37
4.2.2.4 Remote Access	
4.2.2.5 Denial of Service	41
4.3. Chapter Summary	43

Chapter 5: Remedial Measures	44
5.1. Chapter Overview	45
5.2. Stage 3: Remedial Measures after Analysis of attack	45
5.2.1 Setting up Raspberry	46
5.2.2 Securing Raspberry	47
5.2.3 Connecting Raspberry and Camera	49
5.3. Chapter Summary	
Chapter 6: Conclusion and Future Work	
6.1. Chapter Overview	53
6.2. Conclusion	53
6.3. Enhancments and Future Work	55
6.4. Chapter Summary	55
Refernces	

# List of Figures

Figure !-1: IoT Devices	3
Figure 2-1: IP Camera	12
Figure 2-2: Nest Thermostat	13
Figure 2-3: Nike Fuelband	14
Figure 3-1: RIoTS	19
Figure 3-2: Xiaomi Smart Bulb	21
Figure 3-3: IP Smart Camera	22
Figure 3-4: Raspberry pi	24
Figure 4-1: Ettercap Interface	27
Figure 4-2:Search for Hosts	28
Figure 4-3: List of Available Hosts	28
Figure 4-4: Adding devices as targets	29
Figure 4-5: ARP Poisoning	29
Figure 4-6:Wireshark Interface in linux	30
Figure 4-7: Communication between router and smart bulb	31
Figure 4-8: Ettercap Interface	32
Figure 4-9: Scan for Hosts	33
Figure 4-10: List of Hosts on network	
Figure 4-11: Adding Camera as target 1	34
Figure 4-12: Adding Router as target 2	34
Figure 4-13: ARP poisoning	34
Figure 4-14: Wireshark interface	35
Figure 4-15: Communication between Camera and Router	36
Figure 4-16: Password extraction via filter	37
Figure 4-17: Password in packet description	
Figure 4-18(a): Password in packet description	40
Figure 4-18(b): Password in packet description	41
Figure 4-19: Multiple Xerxes Windows in Operation	42
Figure 4-20(a): Response of app	42
Figure 4-20(b): App Unresponsive while Xerxes Active	43
Figure 5-1: Raspberry Installation	46
Figure 5-2: Raspberry Interface	43
Figure 5-3: Open Ports	47

Figure 5-4: Script	48
Figure 5-5: Connection of Raspberry with IP Camera	49
Figure 5-6: Script file to provide access to camera	50
Figure 5-7: IP Camera and Raspberry connected via Ethernet	50
Figure 5-8: Netwrok Scan	51
Figure 6-1:Security Risks	55

# **CHAPTER 1: INTRODUCTION**

# CHAPTER 01 INTRODUCTION

## 1.1. Chapter Overview

This chapter sketches a detailed introduction of RIoTS, accentuating the problem statement, objectives, scope and reason for the exception of the project under discussion.

#### **1.2.** Problem statement

The Internet of Things (IoT) is a network of physical objects with embedded electronics that can communicate and sense interactions with one another and the outside environment. In the coming years, IoT-based technology will provide improved levels of service, effectively transforming how people live their lives. Medicine, power, gene therapies, agriculture, smart cities, and smart homes are just a few of the fields where IoT has made a significant impact.

As the technology is evolving every day, people are relying more on the advanced smart devices such as smart watch, smart camera, baby monitors and multiple devices which are used in our daily lives. Little do they know that with increasing demand of these devices, the security considerations are overlooked and vulnerable IoT devices are being designed. This opens the door for adversaries, which often exploit such devices with little or no effort. Attacker can easily find the flaws and compromise the security of such devices. By compromising, the attacker gets a hold of the device and access to the information stored on it. The information can be exploited by any means and thus is a great threat.

Security managers are focused on preventing network and web attacks by preventing all types of penetration, but nothing is done to identify and mitigate the attacks in a single solution. By combining identification and mitigation, our project bridges this distance. With the number of attackers and the sophistication of their DoS, Scanning, and Web attacks rapidly increasing, it is more important than ever to monitor and record their activities and collect information that can help improve existing security tools.

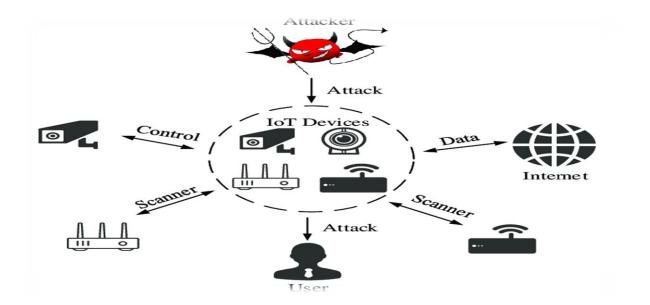


Figure 1-1 IoT Devices

## 1.3. Objectives

#### • Creation of a minor IoT system

An Internet of Things platform connects device sensors to data networks. It gives you access to the data that the backend application uses. An IoT platform enables developers to disperse apps, collect data remotely, secure connectivity, and monitor sensors. We intend to develop a limited IoT network consisting of the necessary components, interconnecting nodes and smart device(s) for being able to complete necessary tasks required of an IoT system.

#### • Launching a number of attacks on said system

Internet of things (IoT) connects devices remotely with the easy operability. Because of particular aspects of the underlying technology, threats against IoT systems and devices translate to higher security risks. With an IoT system in actuation we intend to launch cyber-attacks on said system to compromise normal functionality

Some common cyber-attack types are enlisted:

- Physical attacks: This sort of attack tampers with hardware components. The majority of gadgets are used in locations where they are vulnerable to physical attacks.
- Reconnaissance attacks: unauthorized discovery of service, or vulnerability detection and mapping. Scanning network ports, packet sniffers, traffic analysis, and making queries regarding

IP address information are all examples of reconnaissance attacks.

- 3. Denial-of-service (DoS): This kind of attack is an attempts to make a machine or network resource unavailable to its intended users. The majority of IoT devices are vulnerable to this attack because of low memory capabilities and limited computation resources
- 4. Access attacks unauthorized persons gain access to networks or devices to which they have no right to access. There are two types of access attack: physical access, whereby the intruder can gain access to a physical device. The second is remote access, where an intruder can remotely access internetconnected devices.
- Attacks on privacy: Because enormous amounts of data are easily accessible through remote access mechanisms, privacy protection in IoT has become increasingly challenging.
- Detection and remedial measures after analysis of attacks: Proposing an attack detection model for protection of IoT/smart devices from such attacks

## **1.4.** Justification for the Selection of Topic

The absence of a system that detects and mitigates attacks on an IoT device motivated us to create our handy device, which is easy to use, carry, configure and upgrade. Moreover, the proposed solution can easily be merged with any existing security solution and is not costly. It would not share any user data with the vendor, making it trustworthy and easily adaptable.

### 1.5. Scope

To establish an indigenous secure anonymous network for Pakistan on a relatively smaller scale that makes use of

• Hardware – Raspberry Pi 3.0 (with LCD Display)

• Software

- Windows 10

– PyCharm 2018.2.2

– Raspbian

- Linux

– Ubuntu

• Networks - Socket programming, TCP/IP - Threading/ Multi-threadin

This project might be used in various disciplines like e-commerce, sensitive organizations, the healthcare sector, and the entire security sector of the country since it allows for the identification and prevention of attacks in real-time. It can reduce compromised privacy in banks and secure other confidential information, such as stocks and client personal data, in several multinational corporations and offices.

The project's primary purpose is to supplement and incorporate existing security tools while also providing resistance to attacker evasion. As a result, we hope to use this project to combine our theoretical expertise with practical experiences to improve our network security. This project aims to create a complete security solution with its machine learning detection system and a raspberry gateway to analyze attacks.

### **1.6.** Contributions

This project was created primarily to protect networks against DoS, Scanning, and Web attacks by real-time identification and mitigation of said attacks. Providing insight into the attackers' activities and allowing researchers to understand the attackers' strategies better. As a result, they can integrate and improve the existing architecture.

#### **1.7.** Chapter Summary

The current chapter elucidates a comprehensive overview of the project under discussion, thus explaining the introductory details, including scope, problem definition and reason for selecting the project RIoTS.

# **CHAPTER 2: LITERATURE REVIEW**

# CHAPTER 02 LITERATURE REVIEW

### 2.1. Overview

This chapter deals with comprehensive details of attacks on IoT devices and solutions offered worldwide, their limitations and the uniqueness of our proposed solution.

#### 2.2. Project Domain

As the world progresses towards automation and smart cities in the internet age, nearly everyone is connected to the internet via Wi-Fi Access points. Smart IoT devices are now everywhere. They make our life easier by conversing, sharing information, and doing vital tasks.. As devices are easy to use, their increasing demand lessened the focus on securing devices. This leads to privacy and security concerns for businesses, governments and smart cities. The necessity to defend these smart devices from cyber-attacks is critical.

#### 2.3. Background

Advancements in digital communication, leading to increased security and privacy concerns, have urged researchers to develop different tools and solutions so far. A secure IoT device has always been required to preserve the communication among local wifi networks. Usually, a raspberry gateway is used to serve the need. Still, the problem lies in the fact that smart devices provide extensive anonymity but cannot encrypt the data for the users. On the contrary, RIoTS successfully secure real-time information but does not hide the identities of communicating parties. For IoT devices, the best solution up to date provides both security and anonymity. Still, it is slack, prohibited in some countries and cannot be availed by intelligence agencies and the Pak military because of surveillance and scrutiny. The elucidation of these problems resides in an endemic network that can provide complete security and anonymity to the web traffic.

#### 2.4. Literature Review

With billions of IoT devices, applications, and services currently in use and more on the way, it's clear that the Internet of Things is here to stay. For very same reason, IoT security is paramount. Cyberattacks can be launched using poorly secured IoT devices and services, compromising sensitive data and endangering individual users' safety. This literature review aims to gain an understanding relevant to our project, Real-Time Internet of Things Security.

#### 2.4.1. Threat modelling of Smart Light Bulb

The first paper which we studied is Threat modelling of Smart Light Bulb. In this paper, the author mainly focuses on the security and possible attacks related to CoAP and Zigbee protocols, and the method used to identify threats is the threat model. The AD Tool (Attack-Defense trees) is chosen to model the threats. The attacks mentioned in this paper are Worm attacks, Hacking Smart Light bulbs using Bluetooth, and DDoS attacks called Botnet.

### 2.4.2. Smart bulb using IoT

The second paper studied is the smart bulb using IoT. It can be controlled by using the internet. In this paper, a smart battery charger is designed to continuously provide a power supply to the bulb without any stop. This project combines a solar power inverter, a wireless power transfer idea for power generation, and a home automation system based on the Internet of Things (IoT).

#### 2.4.3. Taxonomy of Security Attacks

The third paper studied is Taxonomy of Security Attacks. In this paper, there are several types of attacks on IoT, such as Spoofing/Altering/Replay Routing attacks, Denial of Service (DoS) attacks, Sybil attacks, and node capture attacks in IoT. There are some limitations mentioned in this research paper, like it outlines the various attacks on appropriate security measures.

### 2.5. Research Based Results

#### 2.5.1. IP Camera

A case study of a web IP camera will discover the existing security models in the camera and try to find out the further vulnerabilities of this IP camera. In smart cities or smart homes, all the devices connected via the internet face the maximum threat problems. Web IP camera is employed outside buildings and homes for protection. It can be monitored through android phones.



Figure 2-1 IP Camera

## 2.5.1.1. Security Analysis

Three security flaws were found while studying IP camera.

- First it is not using secure channel for communication, all sensitive data was transferred in clear text without using any encryption techniques.
- Second is RTSP URL used to stream data was brute forcible as it a common RTSP URL.
- Third is IP camera account credentials are stored in clear text and not encrypted within the mobile application.

Any attacker can get access to the camera provided that the camera is connected to the internet and gets video information just by having its IP address. Attackers can even scan the internet to find open RTSP ports and try common URLs which this camera uses to broadcast its video information.

## 2.5.2. Nest Thermostat

Nest Thermostat is a smart device that uses learned behaviour to manage air conditioning (HVAC) units. The thermostat has a motion..sensor that detects whether users are present at the installed site, as well as a WiFi module that allows it to connect to the user's home or business network and interact with the Nest Cloud, allowing it to be controlled remotely. It also has a ZigBee..module that allows it to communicate with other Nest devices



Figure 2-2 Nest Thermostat

Steps taken to compromise the device:

- Setting system \_boot [5] pin to be pulled high.
- Trojan Injection.
- Custom Kernel generation with debugging capabilities.

## 2.5.3. Nike Fuelband

The Nike comes with a normal USB port that may be used to charge and sync your device. This connector can also be used to update the device's firmware, although it does not offer external access to the BOOT0 pin. As such, the device must be opened. The fact that the microcontroller is built as a Ball Grid Array (BGA) further complicates the situation, as there is no direct access to the BOOT0.



Figure 2-3 Nike Feulband

External reads and writes are locked against the internal flash by the microprocessor, effectively isolating the device's firmware from the outside

world. This protection was not employed on Nike. As a result, an attacker with physical access to the device can freely modify the contents of flash.

#### 2.5.3.1. Vulnerabilities

The circuit board's traces must then be followed in order to come across a test point that indirectly exposes the pin in issue. Following this procedure, we were able to detect the BOOT0 pin indirectly, which was then driven to logic 1 using a 100 V resistor linked to..VDD. This allowed us to gain access to the alternate..boot mechanism and take advantage of the device's lack of read/write security. Using standard ST Microelectronics development tools, communication..over USB with the STM32.was achieved, and the device's firmware..was obtained. We began modifying the device's firmware once we had it in our hands.

### 2.5.3.2. Firmware Testing

The most basic alteration is a string replacement, which involves locating a string in the program that is displayed at some time and replacing it with something different. After making the update, the device's modified firmware was written to it, only to discover that normal functioning had vanished. Further testing revealed that the failure to compute the right CRC for the image was to blame. The check failed because the image had been altered.

#### 2.5.3.3. Results

A closer look into the disassembled firmware image revealed that it used the STM32 microcontroller's CRC engine to validate its authenticity by comparing the result of the checksum. CRC computation against a stored value. This value could be changed because it was located within the image. The patched firmware was delivered to the device and verified to work after adding the necessary checksum.Boot0 pin not accessible.

- Select a different boot mechanism.
- Take advantage of the device's lack of read and write protection.Communication over USB was achieved.
- The device's firmware was obtained.
- String replacement.
- The disassembled firmware image.
- This value was found within the image itself.

## 2.6. Conjecture

Since all the above-mentioned anonymity solutions have stated issues and complications which have made them unreliable and risky, thus urging the need of an effective and innovative solution that will corroborate the provision of utter anonymity and security, thus promising the internet users with the solution of all their reservations..regarding cyber threats and cyber vulnerabilities.

## 2.7. Chapter Summary

The above chapter incorporated the details of different tools used today in order to attain security. It comprised of comprehensive study of smart devices and the issues that occur at the cost of their employment. It also includes the concerns addressed by the different companies.

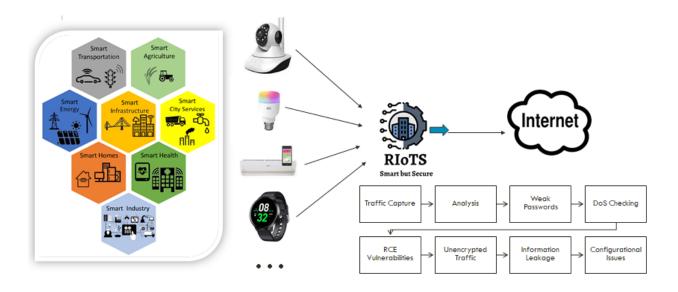
# CHAPTER 3: TECHNICAL SPECIFICATIONS

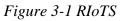
# CHAPTER 03 TECHNICAL SPECIFICATION

## 3.1. Overview

This chapter covers the project's technical requirements as well as the specifics of the project's key steps as it progresses.

## 3.2. Network Diagram





RIoTS would work as depicted in the network diagram above. In a general environment IoT devices communicate with their routers gateway towards the internet. These devices are communicating valuable & critical user data without check, hence inviting attackers to take advantage of this fact

So RIoTS works in a way that it ensures real-time IoT security, the solution essentially sits behind these devices and the network gateway in order to analyze the information goes out of the network, and how it captures the traffic and analyses that, sees if there are weak passwords that can be compromised, it checks whether devices are vulnerable to DoS attacks, whether remote code can be executed over them, whether the communication their using with the rest of the internet is unencrypted or not, if information is being leaked and whether there are configuration issues that the attacker can exploit because even one of these internal IoT device is compromised then it will act as a pivot point to launch wider range of attacks on rest of the IoT network.

## **3.3.** Methodology

Our solution can basically be broken down into three stages. First would consist of the procurement of IoT based hardware which would be in-use in society and would a great threat if compromised. Second would be to devise ways of compromising these devices in order to bring forward their flaws and eventually shut them down. Lastly to produce ways to counter the security flaws found before and implement them in real-time.

In this chapter, we will discuss first stage that is hardware procurement. Rest two stages will be discussed in further chapters

## 3.4. Stage 1: Hardware Procurement



## 3.4.1 Xiaomi Smart Bulb

Figure 3-2 Xiaomi Smart Bulb

## **Specifications**:

- Service Life:Approx.25,000 h
- On/off : 125000 Cycles
- Operating Temperature:-10°C to 40°C
- Operating Humidity:0–85% RH
- Operating Frequency:2412–2472 MHz

• Maximum Output Power:17.41 dBm

## 3.4.2 IP Smart Camera



Figure 3-3 IP Smart Camera

## **Specifications:**

• Brand	V380
• Shape	Dome(Indoor)

2 MP

Camera Resolution

Camera Technology	Network/IP/Wireless
Camera Range	10 to 20 m
• Max. Image Resolution	1280 x 720
• Minimum Order Quantity	5

## 3.4.3 Raspberry Pi

The last hardware utilized in the project is Raspberry Pi 3.0. It is a portable and rechargeable device which is password protected. This module performs all the necessary processing required for security of IoT device. It will act as a gateway to the devices and detect the attacks in real time. Raspberry Pi is also responsible for the filtration of malicious URLs. Raspberry Pi 3 is basically a development aboard in PI series. It can be contemplated as a solitary board computer which works on Linux OS. The board, in addition of having several characteristics, also has a tremendous processing speed, making it appropriate for modern applications. The Raspberry Pi hardware has advanced through numerous versions which highlight distinctions in peripheral-device provision and memory capacity.

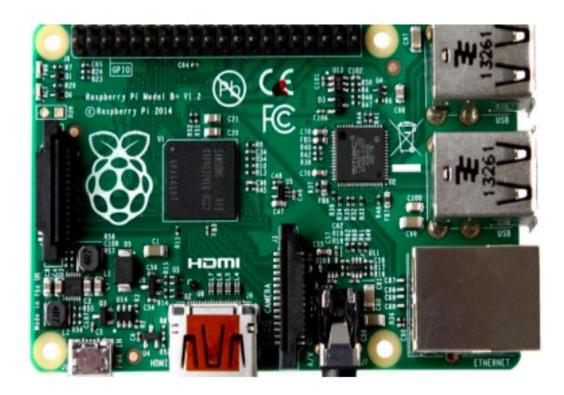


Figure 3-4 Raspberry pi

## **3.5. Chapter Summary**

This chapter gives insight to the methodology used for the project. It gives the description and specifications of hardware devices used in our project.

# **CHAPTER 4: PROPOSED SOLUTION**

# CHAPTER 04 PROPOSED SOLUTION

## **4.1** Chapter Overview

The following chapter explicates the entire technical working of the project under discussion, including launching of attacks and its countermeasures part. It describes the modes in which the solution operates and tells about the problems which RIoTS resolved.

#### 4.2 Stage 2: Launching Attacks

Now we will discuss in detail the attacks launched on our procured devices.

#### 4.2.1 Xiaomi Smart Bulb:

#### 4.2.1.1 ARP Poisoning:

We start by starting up the Ettercap tool in Kali Linux, which we will use to perform a man-in-the-middle attack on the Smart bulb. Ettercap is an all-in-one solution for man-in-the-middle attacks. It has live connection sniffing, on-the-fly content filtering, and many other cool features. It provides various network and host analysis functions, as well as active and passive protocol dissection.



Figure 4-1Ettercap Interface

• Connect to the network that the smart bulb is on and click on 'scan for hosts' from the drop-down menu. A search will pop up and automatically search for all available hosts on the connected network

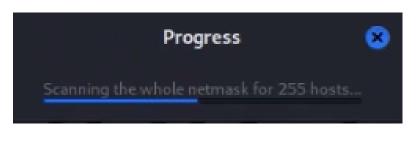


Figure 4-2 Search for hosts

• All the scanned hosts can be viewed by clicking on 'show hosts' from the drop-down menu. All hosts will be displayed with their IP addresses and MAC addresses.

NUM         NUM <th>e a a</th> <th></th> <th></th> <th>E</th> <th>tercap</th> <th><b>3 8 : 0 0</b></th>	e a a			E	tercap	<b>3 8 : 0 0</b>
Memory         Mode Adde State Sta	Second and second					
Deter Hot         Addto Target 1           Deter Hot         Addto Target 1						
1/148.151       6/4.47/CK0128         1/148.154       0/074/DE54.468         1/148.155       0/074/DE54.468         1/148.155       0/074/DE54.468         1/148.155       0/074/DE54.468         1/148.155       0/074/DE54.468         1/148.157       0/074/DE54.468         1/148.158       0/074/DE54.468         1/148.157       0/061/DB58/28         1/148.158       0/061/DB58/28         1/148.158       0/061/DB58/28         1/148.157       0/061/DB58/28         1/148.158       0/061/DB58/28         1/148.151       0/061/DB58/28         1/148.151       0/061/DB58/2						
1048.123       3070/1056A.484         1048.123       SOSHE / 2007.680         1048.123       SOSHE / 2007.680         1048.123       ILD/1696A.221         1048.123       ILD/1696A.221         1048.123       ILD/1696A.221         1048.123       ILD/1696A.221         1048.124       SOSHE / 2007.000         1048.125       ILD/1696A.221         1048.126       ILD/1696A.221         1048.127       ILD/1696A.224         1048.128       ILD/1696A.24707         1048.128       ILD/1696A.24707         1048.128       ILD/1696A.244						
1048.135       90581/2007.0150         1268.135       6054809FFC48         1268.137       14019494CA12         1268.138       6054809FFC48         1268.137       14019494CA12         1268.138       6054909FFC48         1268.137       14019494CA12         1268.137       1601947CA1009         1268.137       70721008420         1268.138       0044000CA10         1268.138       004500CA4000						
1268.123       SC-38 80 97 FC 43         1268.123       SC-38 80 97 FC 43         1268.123       SD-36 57 C-40         1268.124       SD-36 57 C-40         1268.125       SD-36 57 C-40         1268.126       Addro 300 C-41         1268.127       SD-36 57 C-40         1268.128       ID-40 000 C-41						
2146.18.3         65.48.09#FC48           2126.18.3         65.48.09#FC48           2126.18.3         65.48.09#FC48           2126.18.37         160.158/CA1C           2126.18.37         FC1088.02.6           1126.18.7         FC1088.02.6           1126.18.8         FC1488.02.6           1126.18.9         FC1488.02.6           1126.18.9         FC1488.02.7           FC168.18.10         FC148.02.7	192.168.18.26					
1268.18.37         10.169.967.0-7.21C         "           1268.18.37         10.268.16.37         00.267.266.0-7.00           1268.18.37         00.266.0-7.00         -           1268.18.37         00.246.0-7.00 DP         -           1268.18.37         00.241.00 DP         -           1268.18.37         0.221.01 ST/TOTO         -           1268.18.37         0.221.01 ST/TOTO         -           1268.18.37         0.272.01 ST/TOTO         -           1268.18.37         0.272.01 ST/TOTO         -           1268.18.37         0.272.01 ST/TOTO         -           1268.18.30         0.477.05 ST/TOTO         -           1268.18.30         C.574.00 ST/TOTO         -           1268.18.31         Addto 22255.02         -         -           1268.18.31         C.558.00 SF/TOTO         -         -           1268.18.31         C.558.00 SF/TOTO         -         -           1268.18.13         C.558.00 SF/TOTO <td< td=""><td>192.168.18.35</td><td></td><td></td><td></td><td></td><td></td></td<>	192.168.18.35					
108.18.54       80.857 EX.30.08         108.18.57       FOE:10.08.86 26         108.18.70       CA2664740.09         108.18.71       CA2664740.09         108.18.72       1044002CA10         108.18.84       C541594707         108.18.87       C573190.0846         108.18.84       C541594707         108.18.87       AC27395557A         108.18.87       C57390.0846         108.18.87       C747800A8672         208.18.10       A0480278555         C108.18.07       C747800A8672         208.18.10       C45451980         C10*       C0767C207F88         C0767C207F88       C0767C307F88         C0756C207F88       C0767C307F88         C0756C207F88       C0767C307F88         C0756C207F88       C0767C307F88         C0756C207F88       C0767C307F88         C0756C207F88       C0756C207F88         C0756C207F88       C0756C207F88 <td>192.168.18.37</td> <td></td> <td></td> <td></td> <td></td> <td></td>	192.168.18.37					
126.8.157       625:108.86/26         126.8.157       0CA266/47.00.09         126.8.157       0CA266/47.00.09         126.8.158       104:000CA1:10         126.8.158       0.54:10.94:47.07         126.8.158       4C-63:7:10.09.46         126.8.158       4C-63:7:10.01.46         126.8.158       4C-63:7:10.01.46         126.8.158       4C-63:7:10.01.46         126.8.158       4C-63:7:10.01.46         126.8.159       Ad-40:10:29.255.55         126.8.159       Ad-40:10:20.1         126.8.159       Ad-40:10:20.1         126.8.159       Ad-40:10:20.1         126.8.159       Ad-40:10:20.1         126.8.159       C-5A:80:9F; C:1A         126.8.159       C-5A:80:9F; C:1A         126.8.159       C-5A:80:9F; C:1A         126.8.159       Addto Target 1         126.8.159       C-5A:80:9F; C:1A	192.168.18.54					
126.12.7       0.52.66.24.30.09         126.12.7       2411.45.17.00C         126.12.8.12.8       0.144.000C.A1.0         126.12.8.12.8       0.54.19.04.27.7         126.12.8.12.8       0.54.19.04.27.7         126.13.8.3       0.54.19.04.27.7         126.13.8.3       1.64.05.29.55.5         126.13.02       A46.05.29.55.5         126.13.03       6.54.09.27.7.8         126.13.04       C.76.7.20.04.67.7         126.13.03       C.76.7.20.7.8         0.75.43.04.06.45.7       C.76.7.20.7.8         126.13.03       C.76.7.20.7.7.8         0.75.43.04.06.06.45       211.45.31.78.0C         Androidocal       C.76.7.20.7.7.8         0.75.43.04.06.06.45       211.45.31.78.0C         126.13.103       C.76.7.20.7.7.8         0.75.43.04.06.06.45       211.45.31.78.0C         126.13.13       C.76.7.20.7.7.8         0.75.43.04.06.06.45       211.45.31.78.0C         126.13.13       C.76.7.20.77.8         0.75.43.04.06.06.45       211.45.31.78.0C         Androidocal       AddtoTarget1         126.14.11.31       C.76.7.20.77.80.78.78.78.78.78.78.78.78.78.78.78.78.78.	192.168.18.57					
1246.17.5       241/1 453.17.8.0.C       Andreid local         1246.8.18.4       0.544.150.47.0.7       1         1246.8.18.7       422.17.1.8.13.4.4       0.544.150.47.0.7         1246.8.18.7       422.17.1.8.13.7.7.8.7       1         1246.8.18.8       1.657.554.88.4.8.8       1         1246.8.19.0       1.667.554.88.4.8.8       1         1246.8.19.0       1.667.554.88.4.8.8       1         1246.8.19.0       1.667.555.55       1         1246.8.19.10       6.468.12.00       1         1246.8.19.10       6.468.12.00       1         1246.8.19.10       6.468.12.00       1         1246.8.19.10       6.57.50.0F.88       1         1246.8.19.10       6.55.80.09.F.FC.1A       AddtoTarget1         1246.8.19.10       AddtoTarget1       AddtoTarget2	192.168.18.71					
1248.188       104.000C.41.10         1248.188       0.5441504.470         1248.18.9       4.52719.557.00         1248.18.9       4.52719.557.00         1248.18.9       4.52719.557.00         1248.18.9       4.52719.557.00         1248.18.9       4.52719.557.00         1248.18.9       4.567.53.08.48         1248.18.00       A.640529.555.5         1248.18.10       A.640529.555.5         1248.18.10       A.640529.555.5         1248.18.10       A.640529.555.5         1248.18.10       A.640529.555.5         1248.18.10       A.640529.555.5         1248.18.10       A.640529.557.6         1248.18.10       A.640529.557.5         1248.18.10       A.640529.557.5         1248.18.10       A.640529.557.5         1248.18.10       A.640529.557.5         1248.18.13       A.640529.557.5         1	192.168.18.76					
146.18.87       A227.94557.A0         1266.18.97       AC637.190.08.46         1268.18.03       AC67.05.184.86         1268.18.02       A6405295555         1268.18.03       TCA7.800.08.46         1268.18.00       FCA7.800.08.46         1268.18.01       FCA7.800.08.46         1268.18.01       FCA7.800.08.46         1268.18.01       FCA7.800.08.46         05.742.94.07.86       Ad406.86.72         1268.18.01       GC5.50.67.88         05.752.94.07.86       Ad406.00.48         1268.18.13       GC5.80.97.FC.1         05.752.94.07.86       Ad406.73.97         1268.18.13       GC5.80.97.FC.1         05.752.94.07.86       Ad406.73.97         1268.18.13       GC5.80.97.FC.1         05.752.94.07.86       Ad406.73.97         1268.18.13       GC5.80.97.FC.1         05.752.94.07.95       Ad406.73.97         1268.18.13       Ad406.73.97         1269.	192.168.18.83					
106.19.95       476.97.90.08.46         106.19.95       1AC F054.59.486         1026.18.90       1AC F054.59.486         1026.18.91       TAC F054.59.486         1026.18.91       CoFF62.55.67.88         1027.56.81.50.12       AddtoTarget1         AddtoTarget2	192.168.18.84	OC:54:15:D4:47:07				
1268.18.98       1.47.094.38.486         1268.18.90       A448.05593555         1268.18.100       F63.186.47.78         1268.18.101       F65.250.FF.88         10:1       C0F6.250.FF.88         10:1       C0F5.250.FF.88         10:1       C0F5.250.FF.21A	192.168.18.87					
1168.18.102       A48.05929555.05         1168.18.102       TX-78/DD/A66/52         1168.18.103       A64.08248.07.78         1168.18.103       A64.08248.07.78         0.01       COF 6C2.05/FE.88         0.0276.48.20.05.05       Addto Target 1         Addto Target 1       Addto Target 2	192.168.18.95	4C:63:71:9D:D8:46				
146.18.103       7C.78         146.8.18.103       Ac4480248F7.C0         0c3       C6FC.250FF.88         105.754a3.60364567       Ac4480248F7.C0         126.8.18.103       C6FC.250FF.88         105.754a3.60364567       Ac4480248F7.C0         126.8.18.103       C6FC.250FF.88         105.754a3.60364567       Ac4480248F7.C0         1268.18.103       C6FC.250FF.88         1268.18.103       C6FSA8099F.FC.1A	192.168.18.98	1A:CF:D5:43:B4:BE				
1164.18.10       6.45.18.12.46.47.76         0.2166.18.11.3       Adv4686.46.86.7C         0.01       COF6C.15.0FF.88         00:756.43.84.086.4048       24.11.45.31.78.0C         Android October       Android October         1168.18.131       6C.5A.80.89F.FC1A    Delete Host        Addto Target 1    Addto Target 2        Addto Target 1    Addto Target 2        Addto Target 1	192.168.18.102	A4:48:D5:29:55:5C				
L166 L8 113       Add to Target 1       Add to Target 2         Delete Host       Add to Target 1       Add to Target 2         Implified with Generating up Understands for 55 hosts.       Second	192.168.18.103	7C:A7:B0:DA:86:F2				
0:1     C0F6/230/F984       0:1:5/62-34/03964-0408     24114/53.178.0.C       Android local       1:6.8.18.13     6C.5A.80.9F.F.C.1A         Delete Host     Addro Target 1         Addro Target 2    complete Host	192.168.18.110					
Delete Host Add to Target 1 Add to Target 2 Ad	192.168.18.113					
Delete Hoot     Add to Target 1     Add to Target 2       no scripti were specified, not starting uf ong blande string	fe80::1					
Delete Host Add to Target 1 Add to Target 2 no scripto were specified, net starting upl ting Undired solfting domizing 35 hosts for scanning noigtine whole entrationation for 55 hosts						
noscripts were specified, not starting upt ting Unified suffing domizing 25 hosts for scanning noing the whole extensis for 255 hosts	192.168.18.151	6C:5A:B0:9F:FC:1A				
noscripts were specified, not starting upt ting Unified suffing domizing 25 hosts for scanning noing the whole extensis for 255 hosts						
noscripts were specified, not starting upt ting Unified suffing domizing 25 hosts for scanning noing the whole extensis for 255 hosts						
noscripts were specified, not starting upt ting Unified suffing domizing 25 hosts for scanning noing the whole extensis for 255 hosts						
noscripts were specified, not starting upt ting Unified suffing domizing 25 hosts for scanning noing the whole extensis for 255 hosts						
noscripts were specified, not starting upt ting Unified suffing domizing 25 hosts for scanning noing the whole extensis for 255 hosts						
noscripts were specified, not starting upt ting Unified suffing domizing 25 hosts for scanning noing the whole extensis for 255 hosts						
noscripts were specified, not starting upt ting Unified suffing domizing 25 hosts for scanning noing the whole extensis for 255 hosts						
no scripts were specified, net starting upl ting Walled wallfing domizing DSF hosts for scanning		Delete	Host	Add to	Target 1	Add to Target 2
nning the whole netmask for 255 hosts						
nning the whole netmask for 255 hosts	Randomizing 255 hosts fo	r scanning				
ests added to the hosts list	Scanning the whole netm	ask for 255 hosts				
	0 hosts added to the hos	ts list				

Figure 1-3 List of available hosts

- Add the smart bulb as target 1 by manually searching for its IP address from the list.
- Add the router as target 2 by manually searching for the gateway IP from the list.

192.168.18.1	C0	
192.168.18.16	E4	Add to Target 1
192.168.18.24	30	Add to Target 2
192.168.18.25	50	Delete host

Figure 4-4 Adding devices as targets

Begin ARP poisoning by selecting it from the drop-down menu. ARP poisoning is successful, and your device will be the man-in-the-middle between the smart bulb and the router.

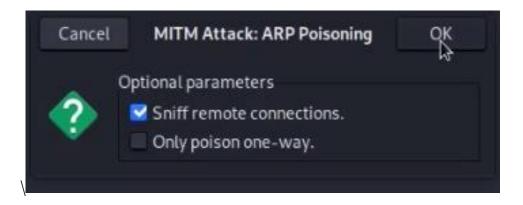


Figure 4-5ARP poisoning

#### **4.2.1.2.** Traffic Analysis

Startup Wireshark tool in Kali Linux and select the option with an active internet connection. Wireshark is the world's most popular and widely used network protocol analyzer. It is the de facto (and frequently de jure) standard across many commercial and non-profit organisations, government agencies, and educational institutions because it allows you to observe what is occurring on your network at a microscopic level. The development of Wireshark is still going strong because to voluntary contributions from networking experts all over the world, and it is a continuation of Gerald..Combs' 1998 effort.

		The Wireshark Network Analyzer	000
	re <u>A</u> nalyze <u>Statistics</u> Telephony <u>W</u> ireless <u>T</u> ools <u>H</u> elp		
	🖹 🙆 Q E > A E > 📰 📰 🖬 🖬 🖬 🖬		
Apply a display filter <ctrl< th=""><th>-1&gt;</th><th></th><th>□··</th></ctrl<>	-1>		□··
	Welcome to Wireshark		
	Open		
	/home/kali/Webcam_Analysis_2.pcapng (5598 KB)		
	/home/kali/Webcam_Analysis.pcaping (3598 KB)		
	A strandstark from a defension of the defension of the start of the A start of the start of t		
	Capture		
	_using this filter:		All interfaces shown -
	eth0		
	any H*		
	Loopback: lo . bluetooth-monitor .		
	nflog .		
	nfqueue . dbus-system		
	dbus-session		
	Cisco remote capture: ciscodump		
	DisplayPort AUX channel monitor capture: dpauxmon .     Random packet generator: randpkt		
	systemd Journal Export: sdjournal		
	User's Guide · Wiki · Questions and Answers · Mailing Lists		
	You are running Wireshark 3.4.9 (Git v3.4.9 packaged as 3.4.9-1).		
Ready to load or capture		No Packets	Profile: Default

Figure 4-6 Wireshark interface in linux

Now all the communication between the smart bulb and the router will pass through your device. Enter the IP address of the smart bulb in the filter available above so only the relevant communication is displayed and is visible for analysis.

Time	Source	Destination		Length Info
	14 192.168.18.149	161.117.7.124	TCP	182 57125 - 80 [PSH, ACK] Seq=1 Ack=1 Win=5296 Len=128[Malformed Packet]
	41 192.168.18.149	161.117.7.124	TCP	150 57125 - 80 [PSH, ACK] Seq=129 Ack=1 Win=5296 Len=96 [TCP segment of a reassembled PDU]
	62 192.168.18.149	161.117.7.124	TCP	86 57125 - 80 [PSH, ACK] Seq=225 Ack=1 Win=5296 Len=32 [TCP segment of a reassembled PDU]
	81 192.168.18.149	161.117.7.124	TCP	182 57125 → 80 [PSH, ACK] Seq=257 Ack=1 Win=5296 Len=128
	01 192.168.18.149	161.117.7.124	TCP	86 57125 → 80 [PSH, ACK] Seq=385 Ack=1 Win=5296 Len=32 [TCP segment of a reassembled PDU]
	21 192.168.18.149	161.117.7.124	TCP	150 57125 → 80 [PSH, ACK] Seq=417 Ack=1 Win=5296 Len=96
	41 192.168.18.149 60 192.168.18.149	161.117.7.124	TCP	86 57125 → 80 [PSH, ACK] Seq=513 Ack=1 Win=5296 Len=32 [TCP segment of a reassembled PDU]
	74 192.168.18.149	161.117.7.124	TCP	182 57125 → 80 [PSH, ACK] Seq=545 Ack=1 Win=5296 Len=128 182 [TCP Retransmission] 57125 → 80 [PSH, ACK] Seq=1 Ack=1 Win=5296 Len=128
	37 192.168.18.149	161.117.7.124	TCP	150 [TCP Retransmission] 57125 → 80 [PSH, ACK] Seq=129 Ack=1 Win=5296 Len=128
	22 192.168.18.149	161.117.7.124	TCP	86 [TCP Retransmission] 57125 → 80 [PSH, ACK] Seq=225 Ack=1 Win=5296 Len=32
	06 192,168,18,149	161 .117 .7 .124	TCP	182 [TCP Retransmission] 57125 → 80 [PSH, ACK] Seq=257 Ack=1 Win=5296 Len=128
	44 192,168,18,149	161.117.7.124		86 [TCP Retransmission] 57125 → 88 [PSH, ACK] Seq=385 Ack=1 Win=5296 Len=32
35: 182 byt	es on wire (1456 bi	ts), 182 bytes captu	red (1456 bits	a) on interface eth0, id 0
				0e:34:8d (08:00:27:0e:34:8d)
		2.168.18.149, Dst: 1		
mination Cont				
	rol Protocol. Src P	ort: 5/125. Dst Port	: 80. Sea: 1.	Ack: 1. Len: 128
		Port: 57125, Dst Port	: 80, Seq: 1,	Ack: 1, Len: 128
rtext Transfe	r Protocol	ort: 5/125, Dst Port	: 80, Seq: 1,	Ack: 1, Len: 128
formed Packet	r Protocol	ort: 5/125, Dst Port	: 80, Seq: 1,	Ack: 1, Len: 128
rtext Transfe formed Packet 08 00 27 0e :	r Protocol : HTTP] 34 8d b4 60 ed 6b -	75 76 88 60 45 80	'4* kuv	E
rtext Transfe formed Packet 08 60 27 0e : 06 a8 60 cc i	r Protocol : HTTP] 34 8d b4 60 ed 6b 5 90 00 ff 06 3e 55 6	75 76 08 00 45 00 c0 a8 12 95 a1 75	'4. 'kuv >U	E
rtext Transfe formed Packet 08 00 27 0e : 00 a8 00 cc : 07 7c df 25 i	r Protocol : HTTP] 34 8d b4 60 ed 6b 39 00 Ff 06 3e 55	75 76 08 00 45 00 c0 a8 12 95 a1 75 c3 a4 c2 44 59 18	'4. kuv >U  %-P > :L1	E
rtext Transfe formed Packet 08 00 27 0e : 00 08 00 cc : 07 7c df 25 : 14 b0 91 70 i	rr Protocol : HTTP] 34 8d b4 60 ed 6b 30 00 ff 06 3e 55 30 50 00 60 20 3e	75 76 68 60 45 80 C0 a8 12 95 a1 75 SC 3a 4c 44 59 18 SO 60 66 06 00 15 37	'4. kuv >U  % P > :Lt P !1	Е- 
08 00 27 0e : 00 a8 00 27 0e : 00 a8 00 cc : 07 7c df 25 : 14 b0 91 70 i 4d 62 61 b8 :	rr Protocol : HTTP] 34 8d b4 60 ed 6b 30 00 ff 66 3e 55 30 50 00 60 20 3e 30 50 00 60 20 3e 30 50 00 60 20 3e 30 50 00 60 90 78 1	75 76 98 60 45 90 c0 a8 12 95 a1 75 c8 3a 4c 44 50 18 00 60 60 09 15 37 c8 85 bc 14 c6 2 20	'4 kuv >U  %P > :L( p !1 ba x	E
08 00 27 00 3 09 00 27 00 3 00 00 00 77 00 12 14 00 91 70 44 02 01 00 17 16 1a a4 4	rr Protocol : HTTP] 34 8d b4 60 ed 6b 30 00 ff 06 3e 55 30 50 00 60 20 3e	75 76 98 90 45 80 69 88 12 95 61 75 86 88 46 44 59 15 80 98 98 90 01 13 7 86 88 19 14 67 20 16 65 15 14 63 3a	'4. kuv >U  % P > :Lt P !1	E
08 00 27 0e : 00 08 00 27 0e : 00 08 00 cc 07 7c df 25 14 b0 91 70 14 02 01 b0 i fc 1a a4 47 03 b5 69 e1 3	rr Protocol : HTTP] 34 8d b4 60 ed 6b 30 00 ff 06 3e 55 30 50 00 40 20 3e 30 00 69 00 20 3e 30 00 21 00 80 80 00 21 00 80 80 00 25 76 30 30 00 57 76 30 50 00 57 76 30 50 00 57 76 30 50 00 57 76 57 76 50 50 00 57 50 50 00 50 50 00 57 50 50 00 57 50 50 00 50 50 000 50 00 50 50 000 5	75 75 98 00 45 90 0 08 12 95 01 75 08 08 00 11 37 90 09 00 11 37 51 6f 55 bf a6 3a 51 6f 55 bf a6 3a 51 6f 75 bf 76 35	'4 kuv >U  %P > U p !1 ba x 6f }Qoe	E
08         00         27         00         3           00         a8         00         cat         00         a8         00         cat         00         a8         00         cat         00         a8         00         cat         4         00         cat         a7         cd         cat         cat	rr Protocol : HTTP] 34 8d b4 60 ed 6b 30 90 6f 96 3e 55 30 50 96 9c 20 3e 30 50 96 9c 20 3e 30 50 96 9c 20 3e 30 50 96 78 30 97 8 36 56 17 37 44 4 32 eb 56 3f 77 37 42 37 66 46 36 66 a	75 75 98 00 45 00 60 88 12 95 61 75 80 80 00 15 37 80 00 98 00 15 37 81 6f 65 5f 63 38 74 16 75 15 73 74 1f 73 15 75 99 74 1f 73 15 75 99 74 1f 73 15 75 99	'4 `kuv >U p 11 ba x 6f }00e 1+z D ) ⊕ 0 w7 +7h ]j ]	E- pp- 
orest         Transfe           formed         Packet           00         80         02         02           00         88         00         cc           00         70         cd ff 25         14           01         77         cd ff 25         14           02         61         84         16           14         62         61         86           15         14         65         91         76           16         18         63         65         64           16         4         35         35         16           16         42         35         35         16	rr         Protocol           ::         HTTP)           34         8d         b4         60         ed         6b           30         00         0ff         66         3e         55           30         00         ff         66         20         3e           30         00         ff         66         3e         55           30         00         ff         66         3e         55           30         00         ff         66         3e         57           30         00         ff         66         3e         57           32         37         81         ff         77         3e         37         66         4n         54         66         96         19	75 76 88 60 45 80 60 83 22 95 61 75 63 34 62 45 60 13 80 80 80 60 15 33 63 86 85 14 65 76 35 80 86 85 76 35 80 76 14 56 76 35 81 76 31 57 99 9 61 67 67 07 60 49 94 32 64 33 96 62 80	'-4 - kuv >U p !1 ba x 6f }Qoe 1 + z D - 3 + 7h ]J - 1 + 7h ]J - 1 + S&v5	E
est         Transfe           formed         Packet           00         88         00         ct           00         88         00         ct           00         7         ct         7         ct           14         b0         91         76         dt         ct           44         02         01         b8         01         dt         <	st 8d b4 66 ed 6b           34 8d b4 66 ed 6b           36 8d b4 66 ed 6b           36 90 67 66 36 25 55           36 90 60 67 66 36 20 3e           36 90 60 62 35           36 90 60 80 78           36 90 60 80 78           36 90 61 80 80 80 78           36 94 73 35 60 76           37 66 46 77 37           37 66 47 73 37           37 66 47 73 37           37 66 47 73 74           37 66 47 73 74           37 66 47 73 74           37 66 74 76 47           38 61 88 97 17	75 76 88 00 45 80 00 88 12 95 61 75 80 08 00 15 37 80 00 80 00 15 37 51 6f 65 5f 86 38 81 f 4 38 7 35 74 1f 3 15 75 99 74 31 75 99 74 39 60 49 43 54 43 96 62 84	'4 kuv >U p !1 >: ba Gf }00 1 + z 0 1 + z 0 1 + 7h ]3 − #S&v5 x 0 y 2kv5 x 0 y 2kv5	E
orest         Transfe           formed         Packet           00         80         02         02           00         88         00         cc           00         70         cd ff 25         14           01         77         cd ff 25         14           02         61         84         16           14         62         61         86           15         14         65         91         76           16         18         63         65         64           16         4         35         35         16           16         42         35         35         16	st 8d b4 66 ed 6b           34 8d b4 66 ed 6b           36 8d b4 66 ed 6b           36 90 67 66 36 25 55           36 90 60 67 66 36 20 3e           36 90 60 62 35           36 90 60 80 78           36 90 60 80 78           36 90 61 80 80 80 78           36 94 73 35 60 76           37 66 46 77 37           37 66 47 73 37           37 66 47 73 37           37 66 47 73 74           37 66 47 73 74           37 66 47 73 74           37 66 74 76 47           38 61 88 97 17	75 76 88 00 45 80 00 88 12 95 61 75 80 08 00 15 37 80 00 80 00 15 37 51 6f 65 5f 86 38 81 f 4 38 7 35 74 1f 3 15 75 99 74 31 75 99 74 39 60 49 43 54 43 96 62 84	'-4 - kuv >U p !1 ba x 6f }Qoe 1 + z D - 3 + 7h ]J - 1 + 7h ]J - 1 + S&v5	E
est         Transfe           formed         Packet           00         88         00         ct           00         88         00         ct           00         7         ct         7         ct           14         b0         91         76         dt         ct           44         02         01         b8         01         dt         <	st 8d b4 66 ed 6b           34 8d b4 66 ed 6b           36 8d b4 66 ed 6b           36 90 67 66 36 25 55           36 90 60 67 66 36 20 3e           36 90 60 62 35           36 90 60 80 78           36 90 60 80 78           36 90 61 80 80 80 78           36 94 73 35 60 76           37 66 46 77 37           37 66 47 73 37           37 66 47 73 37           37 66 47 73 74           37 66 47 73 74           37 66 47 73 74           37 66 74 76 47           38 61 88 97 17	75 76 88 00 45 80 00 88 12 95 61 75 80 08 00 15 37 80 00 80 00 15 37 51 6f 65 5f 86 38 81 f 4 38 7 35 74 1f 3 15 75 99 74 31 75 99 74 39 60 49 43 54 43 96 62 84	'4 kuv >U p !1 >: ba Gf }00 1 + z 0 1 + z 0 1 + 7h ]3 − #S&v5 x 0 y 2kv5 x 0 y 2kv5	E
ext.         Transfe           formed         Packet           90         86         90         27         96           90         87         76         67         72         97           14         b9         76         17         25         14         b9         76         17         16         16         44         12         15         59         16         16         44         52         10         16         16         44         52         16         16         45         40         12         15         59         16         16         42         53         16         45         40         12         13         15         59         16         42         53         16         42         33         13         16         42         35         16         14         23         15         16         42         33         16         42         33         13         16         42         35         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16         16	st 8d b4 66 ed 6b           34 8d b4 66 ed 6b           36 8d b4 66 ed 6b           36 90 67 66 36 25 55           36 90 60 67 66 36 20 3e           36 90 60 62 35           36 90 60 80 78           36 90 60 80 78           36 90 61 80 80 80 78           36 94 73 35 60 76           37 66 46 77 37           37 66 47 73 37           37 66 47 73 37           37 66 47 73 74           37 66 47 73 74           37 66 47 73 74           37 66 74 76 47           38 61 88 97 17	75 76 88 00 45 80 00 88 12 95 61 75 80 08 00 15 37 80 00 80 00 15 37 51 6f 65 5f 86 38 81 f 4 38 7 35 74 1f 3 15 75 99 74 31 75 99 74 39 60 49 43 54 43 96 62 84	'4 kuv >U p !1 >: ba Gf }00 1 + z 0 1 + z 0 1 + 7h ]3 − #S&v5 x 0 y 2kv5 x 0 y 2kv5	E

Figure 4-7 Communication between router and smart bulb

## 4.2.2 IP Smart Camera

### 4.2.2.1 ARP Poisoning

We start of by starting up the Ettercap tool in Kali Linux which we will use to perform man-in-the-middle attack on the Smart Camera.



Figure 4-8 Ettercap Interface

Connect to the network that the smart camera is on and click on 'scan for hosts' from the drop-down menu. A search will pop up and automatically search for all available hosts on the connected network.

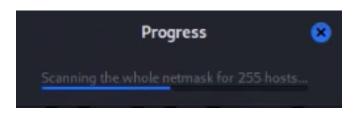


Figure 4-9 Scan for Hosts

All the scanned hosts can be viewed by clicking on 'show hosts' from the drop-down

menu. All hosts will be displayed with their IP addresses and MAC addresses.

e • • •			Ettercap 0.8.3.1 (EB)		0 0 : 0 0
IP Address	MAC Address	Description			
192.168.18.1	C0:F6:C2:5D:FF:B8				
	E4:A4:71:C8:DE:28				
192.168.18.24	30:07:4D:E6:AA:68				
192.168.18.25	50:58:C2:CD:CB:50				
192.168.18.26	3E:A4:8B:77:6B:E9				
192.168.18.35	6C:5A:80:9F:FC:48				
192.168.18.37	14:D1:69:8C:A2:1C				
192.168.18.54	80:86:F2:EA:30:08				
192.168.18.57	F0:EE:10:BB:86:26				
	DC:A2:66:47:4D:D9				
192.168.18.76	24:11:45:51:7B:DC	Android.local			
192.168.18.83	10:44:00:DC:A1:1D				
192.168.18.84					
192.168.18.87	A2:27:19:E5:57:A9				
192.168.18.95	4C:63:71:9D:D8:46				
192.168.18.98	1A:CF:D5:43:B4:BE				
192.168.18.102	A4:4B:D5:29:55:5C				
192.168.18.103	7C:A7:B0:DA:86:F2				
192.168.18.110	E4:B3:18:EA:47:78				
192.168.18.113	A6:46:BE:48:EF:CD				
fe80::1	C0:F6:C2:5D:FF:B8				
	0d8 24:11:45:51:7B:DC	Android.local			
192.168.18.151	6C:5A:80:9F:FC:1A				
	Delete	Host	Add to Target 1	Add to Target	2
Lua: no scripts were spec Starting Unified sniffing.	ified, not starting up!				
Randomizing 255 hosts fo icanning the whole netm 10 hosts added to the hos	ask for 255 hosts				
nosts added to the hos	California (California)				

Figure 4-10 List of Hosts on network

• Add the smart camera as target 1 by manually searching for its IP address from the list.

192.168.18.103	7C:A7:B0:DA:80	5:F2
192.168.18.110	E4:B3:18:EA:4	Add to Target 1
192.168.18.113	A6:46:BE:4B:E	Add to Target 2
fe80::1	C0:F6:C2:5D:F	Delete host

Figure 4-11 Adding Camera as target 1

• Add the router as target 2 by manually searching for the gateway IP from the list.

192.168.18.1	C0_	
192.168.18.16	E4	Add to Target 1
192.168.18.24	30	Add to Target 2
192.168.18.25	50	Delete host

Figure 4-12 Adding Router as target 2

Begin ARP poisoning by selecting it from the drop-down menu. ARP poisoning is successful, and your device will be the man-in-the-middle between the smart camera and the router.

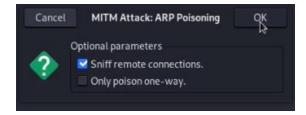


Figure 4-13 ARP poisoning

## 4.2.2.2 Traffic Analysis

Startup Wireshark tool in Kali Linux and select the option which has an active internet connection.

4		The Wireshark Network Analyzer	
	e <u>A</u> nalyze <u>S</u> tatistics Telephony <u>W</u> ireless <u>T</u> ools <u>H</u> elp		
	🗟 🙆 Q 🗧 Ə R 🔶 Ə 🖉 🖉 🖉 🖉		
Apply a display filter <ctrl- <="" th=""><th></th><th></th><th><b>□</b>••</th></ctrl->			<b>□</b> ••
	Welcome to Wireshark		
	Open		
	/home/kali/Webcam_Analysis_2.pcapng (5598 KB) /home/kali/Webcam_Analysis.pcapng (1998 KB)		
	hinning kany webbanikilanysis-praking (1220 k.b)		
	Capture		
	using this filter: I Enter a capture filter		* All interfaces shown *
	eth0 b		
	Loopback: lo		
	nflog		
	nfqueue		
	dbus-session		
	Cisco remote capture: ciscodump     DisplayPort AUX channel monitor capture: dpauxmon		
	Random packet generator: randpkt		
5 C	systemd Journal Export: sdjournal		
	Learn		
	User's Guide · Wiki · Questions and Answers · Mailing Lists		
	You are running Wireshark 3.4.9 (Git v3.4.9 packaged as 3.4.9-1).		
E. Bertstelander unter		No Packets	Profile: Default
Ready to load or capture		No Packets	Profile: Default

Figure 4-14 Wireshark interface

Now all the communication between the smart camera and the router will pass through your device. Enter the IP address of the smart camera in the filter available above so only the relevant communication is displayed and is visible for analysis.

Apply a display filter _ <cti< th=""><th>d. Is ¥</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th><b></b>+</th></cti<>	d. Is ¥										<b></b> +
lo. Time	Source	Destination	Protocol	Length Info							
38 1.249715629		192,168.18.183			37#11 443 - 38600	[ACK] Seg=1 Ack	-8668 Win=3373 L	en=0 TSval=3485967:	66 TSecr=429494		
39 1.324136026		192.168.18.1	ICMP		equest id=0x0000,						
40 1.329759135		192.168.18.1	ICMP		equest id=0x0000						
41 1.332374344		192.168.18.103	ICMP		eply id=0x0000			in 40)			
42 1.337692126 43 1.536500327		192.168.18.103 239.255.255.25	ICMP 8 SSDP	98 Echo (ping) r 217 M-SEARCH * HT		seq=65280/255,	tt1=64				
44 1.632128573		8.219.10.35	SSL	1466 Continuation							
45 1 634080489		8.219.10.35	TCP			3 [ACK] Seg=868	5 Ack=1 Win=1661	5 Len=1400 TSval=43	94949985 TSecr=	3485966798	
46 1.635843439		8.219.10.35	SSL	1866 Continuation							
47 1.641736322		8.219.10.35	TCP					n=16616 Len=1000 T		TSecr=3485967166	
48 1.740585272		192.168.18.103	TCP					658 TSecr=429494990			
49 1.741670568 50 1.748520733		192.168.18.193 192.168.18.183	TCP TCP					Len=0 TSval=3485967 566 TSecr=429494993		49995	
- 51 1.754671051		192.168.18.103						Len=0 TSval=3485967		49938	
0010 00 34 3c fd 40 6 0020 12 67 01 bb 96 c	58 c0 f6 c2 5d ff b8 50 34 06 23 a6 08 d0 58 bb 9d e9 84 1e ed 58 bb 9d e9 84 1e ed 59 91 91 88 6a cf c7	0 0a 23 c0 a8 1 cb 31 80 10 7 9e 5f ff ff	4< 0 4 # 4 9 5								
E eth0: <live capture="" in<="" p=""></live>	progress>							kets: 51 - Displayed: 51 (1			Profile: Defa

Figure 4-15 Communication between Camera and Router

## 4.2.2.3 Password Extraction

For password extraction enter the custom filter "udp.dstport == 7050 " in the filter and wait for the user to login to the smart camera from his app. The packet carrying the username and password will be captured by Wireshark

Tune         Source         Destitution         Protoci         Length Mp           2027.7663907100         0.215.10.3         192.1661.8.1.03         UOP         173.46148 - 7650 Leng130           2027.7663977100         0.215.10.3         192.1661.8.1.03         UOP         173.46148 - 7650 Leng130           2027.7663972505         0.215.10.3         192.1661.8.1.03         UOP         173.46149 - 7650 Leng130           277.7643972505         0.210.10.35         192.1661.8.1.03         UOP         173.46149 - 7650 Leng130
2027.7.649737010 0.229.10.35 102.100.18.10.3 UUP 170.6540 - 7050 Lem:30 272 7.649972505 0.229.10.35 102.100.15 102.15 102.100.15 102.15 10
272 7.64390505 8.229.10.35 192.108.18.103 UOP 178 46140 - 7050 Lem=36 273 7.640712365 8.219.10.35 192.108.18.103 UOP 178 46140 - 7050 Lem=36 Frame 262: 178 bytes on wire (1424 bits), 178 bytes captured (1424 bits) on interface eth8, 10 0 Ethernet II, SrC: Huawelle_Sdfffbd (cd5'foc.250.fffbd, Dst: PostBologu_be:20:00 (08:60:27.be:20:00) Internet Protocol, Version 4, Sp. 05: 192.108.100 USE Datagram Protocol, Src Port: 46140, Dst Port: 7050 Data (136 bytes) 00 00 00 00 00 00 00 00 00 00 00 00 00
273 7.649712865       8.219.10.35       192.108.18.103       UDP       178 46140 - 7090 Lem=130         Frame 262: 178 bytes on wire (1424 bits), 178 bytes captured (1424 bits) on interface eth0, id 0
Frame 262: 178 bytes on wire (1424 bits), 178 bytes captured (1424 bits) on interface eth8, 10 0 Ethernet II, Src: Huawelle_Sdiff:b8 (68:0f6cc2:Sdiff:b8), Dst: PostCompu_De:28:60 (68:68:27:be:28:60) Internet Protocol Version 4, Src: 8.219.10.35, Dst: 102.108.10.103 User Datagram Protocol, Src Port: 46140, Dst Port: 7050 Data (136 bytes) 68 60 27 be 28 60 c6 f6 c2 5d ff b8 68 60 45 14
Ethernet II, Srči Hummelle_Sdifft b8 (68/60 c2: 5difft b8 (68/60 c2 c3 bit fb: 10.00 b
Ethernet II, Srči Hummelle_Sdifft b8 (68/60 c2: 5difft b8 (68/60 c2 c3 bit fb: 10.00 b
Ethernet II, Srči Hummelle_Sdifft b8 (68/60 c2: 5difft b8 (68/60 c2 c3 bit fb: 10.00 b
10 09 44 c <sup>2</sup> 57 49 09 35 11 77 49 80 00 58 23 c9 88 · 149 5 · 14 <sup>2</sup> · 4 <sup>2</sup> ·
10 09 44 c <sup>2</sup> 57 49 09 35 11 77 49 80 00 58 23 c9 88 · 149 5 · 14 <sup>2</sup> · 4 <sup>2</sup> ·
10 09 44 c <sup>2</sup> 57 49 09 35 11 77 49 80 00 58 23 c9 88 · 149 5 · 14 <sup>2</sup> · 4 <sup>2</sup> ·
200 12 47 b4 3c 1b 8a 80 99 9b a2 a7 b2 3c 1b 8a 80 99 9b a2 a7 b3 3c 1b 8a 80 99 9b a2 a7 b3 3c 1b 8a 80 99 9b a2 a7 b3 3c 1b 8a 80 95 7b a2 a7 b3 backer
330 66 90 95 30 65 16 85 35 55 72 90 96 96 96 96 96 90 9 410 60 99 96 66 96 96 96 99 96 96 96 96 96 96
60 00 00 00 00 00 67 6b 68 6f 73 74 65 6c 2e 31 32 33 gkho stel.123
a0 e0 00 e0 00 e0 e0 e0 e0 e0 e0 e0 e0 e0
b0 09 00
Image: Strate

Figure 4-16 Password Extraction via filter

The username and password of the user that accessed the camera will be available in the packet description.

	08	00	27	be	20	60	co	f6	c2	5d	ff	<b>b</b> 8	08	00	45	14		+]E-
0010	00	a4	e7	57	40	00	35	11	77	dΘ	08	db	Θa	23	cΘ	a8	···W0.5·	W · · · · # · ·
	12	67	b4	3c	1b	8a	00	90	9b	a2	a7	00	00	00	64	00	• g < • • • •	d
	00	00	53	68	61	68	65	65	72	00	00	00	00	00	00	00	Shahee	r
	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
	00	00	00	00	67	6b	68	6f	73	74	65	6c	2e	31	32	33	···· gkho	stel.123
	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
	00	00	00	00	00	00	00	00	00	00	Θd	00	00	00	ΘΘ	00		
	00	00																

Figure 4-17 Password in packet description

#### 4.2.2.4 Remote Access

The username and the password obtained in password extraction will be essential for

remote access. We will use the following python script:

# import the opency library

import cv2

# define a video capture object

vid = cv2.VideoCapture("rtsp://admin:Admin123@192.168.18.19:554/11")

while:

# Capture the video frame

# by frame

ret, frame = vid.read()

print(ret)

# Display the resulting frame

cv2.imshow('frame', frame)

# the 'q' button is set as the

# quitting button you may use any

# desired button of your choice

if cv2.waitKey(1) & OxFF == ord('q'):

break

# After the loop release the cap object

vid.release()

# Destroy all the windows

cv2.destroyAllWindows()

After launching the script, a video link will be established, and you will see what the user

sees

	IDLE Shell 3.10.0*	-		X
le	Edit Shell Debug Options Window Help			_
	<pre>Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) AMD64)] on win32 Type "help", "copyright", "credits" or "license()" for more :</pre>		bit	(
>>		 		
	======================================	 	 	
	True			

Figure 4-18(a) Output of python script



Figure 4-18(b) Remote Access of camera on laptop

#### 4.2.2.5 Denial of Service

Denial of service is an interruption in an authorized user's access to a computer..network, typically caused by malicious intent. A DoS attack is possible using Xerxes, a tool in Kali Linux. The hacker The Jester (th3j35t3r) created Xerxes, an incredibly effective DoS tool for automating DoS attacks. It allows you to launch several separate attacks against multiple target sites without the need for a botnet. Startup multiple windows on Kali Linux with the command ./xerxes #targer IP [space] port no; where the target IP will be the camera with the port number being 554, which is permanently open.

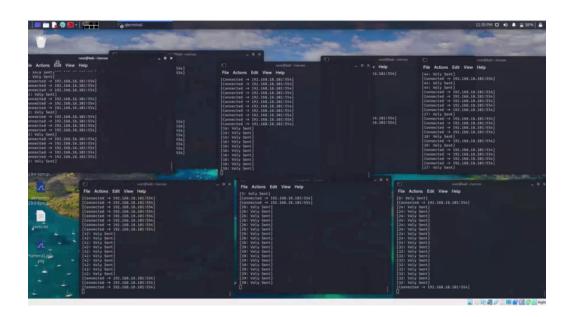


Figure 4-19 Multiple Xerxes Windows in Operation

After a while, the app connected to the smart camera will start lagging and will subsequently stop responding

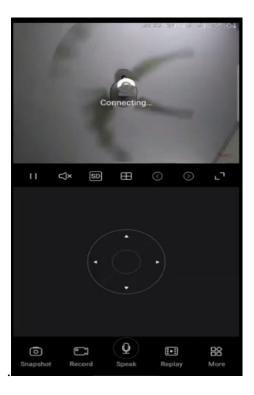


Figure 4-20(a) Response of app

Quitting and restarting the app will not help and it will continue to be non-responsive as long as the DoS is active using XerXes on Kali Linux.



Figure 4-20(b) App unresponsive while Xerxes active

## 4.3. Chapter Summary

We have successfully launched a variety of attacks such as DoS and ARP spoofing on our procured devices. The chapter gave detailed step by step of how we did it.

# **CHAPTER 5: REMEDIAL MEASURES**

# CHAPTER 05 REMEDIAL MEASURES

#### **5.1.** Chapter Overview

This chapter focuses on stage 3 of our project, in which we have to secure the procured devices. The details will be discussed on how to secure our smart devices from intruder.

#### 5.2. Stage 3: Remedial Measure after Analysis of attack

The main aspect of safeguarding devices is the raspberry pi. The Raspberry Pi Foundation, in collaboration with Broadcom, developed a series of miniature singleboard computers in the United Kingdom. The Raspberry Pi project was created with the goal of encouraging the teaching of basic computer science in schools and disadvantaged countries. What we will need to bring this into actuation is:

- Raspberry Pi
- Ethernet cable
- SD Card (4GB or greater)
- Micro-SD card reader
- Power supply for your Pi & a Micro USB cable USB
- Case for your Pi (optional)

#### 5.2.1. Setting up Raspberry Pi

To get Raspberry Pi OS to work on our Raspberry Pi Compute Module 4, we'll need to flash it first. First, use a micro-SD card reader to connect a micro-SD card to your computer, then download the Raspberry Pi Imager for your operating system. Open the Raspberry Pi Imager programme and select the newest version of the Raspberry Pi OS

#### under CHOOSE OS.

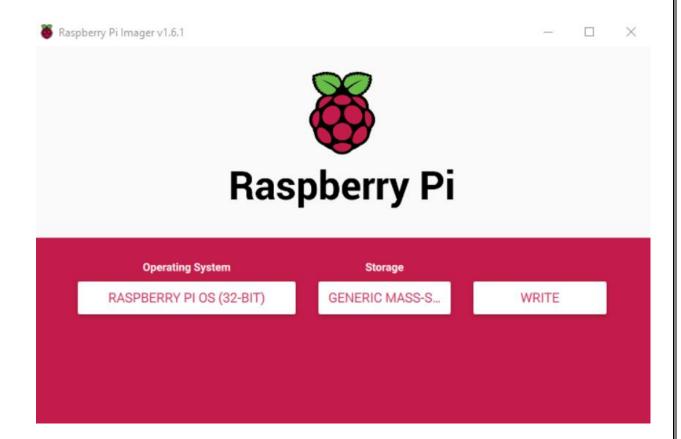


Figure 5-1 Raspberry Installation

Click CHOOSE STORAGE and select the connected micro-SD card and finally, click WRITE. The flashing should continue in a few minutes and writing will be complete.

#### 5.2.2. Securing Raspberry Pi

After all the necessary set-up is done and the raspberry pi is up and running (fig 2).



Figure 5-2 Raspberry Installation

Next comes to secure the raspberry pi from a network standing point. For this we close all the open ports including SSH access and RDP access. This can be done by in the following way:

1. List open ports:

netstat -Int

Active	Internet	S netstat -lnt connections (only servers end-Q Local Address	s) Foreign Address	State
tcp	Θ	0 0.0.0.0:53	0.0.0.0:*	LISTEN
tcp	Θ	0 0.0.0.0:22	0.0.0.0:*	LISTEN
tcp	Θ	0 127.0.0.1:631	0.0.0.0:*	LISTEN
tcp6	ø	0 :::53	111*	LISTEN
tcp6	0	0 :::22		LISTEN

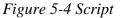
Figure 5-3 Open Ports

Filtered:

For tcp: netstat -lnt | grep LISTEN | awk '{ print ( \$4 ) }' | awk 'BEGIN{FS=":"} { print \$(NF) }' | sort -n | uniq

For udp: netstat -lnt | grep udp | awk '{ print ( \$4 ) }' | awk 'BEGIN{FS=":"} { print \$(NF) }' | sort -n | uniq





 Close all the unnecessary ports or in technical terms to "kill" the open ports using the following commands:

kill port already in use: kill -9 \$(lsof -i tcp:3000 -t)

kill a port: kill \$(lsof -ti:3000)

3. Change SSH default port and use SSH keys instead of passwords. SSH default

port is 22. Edit the SSH server configuration file:

sudo nano /etc/ssh/sshd\_config

Find this line:

#Port 22

Replace the port with the one you want to use, and make sure to uncomment the line:

Port 1111

Save and exit (CTRL+O, CTRL+X)

Restart you Server:

sudo service ssh restart

Then we generate a key on the computer, and then add it to the Raspberry Pi to allow a connection from the computer.

#### 5.2.3. Connecting Raspberry and Camera

Now, after securing the raspberry pi, the aim will be to cause the packets of the IP camera pass through the raspberry which will in its place interact with the internet. It will look something like this.

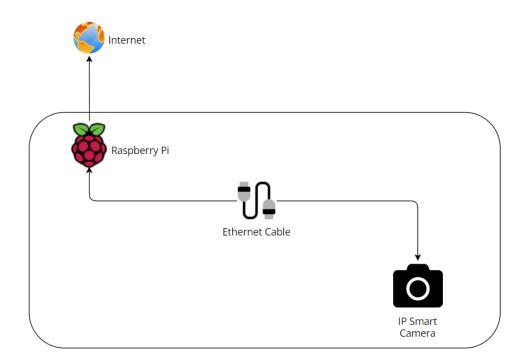


Figure 5-5 Connection of Raspberry with IP Camera

For this we will require an ethernet cable and connect it to the IP smart camera and he raspberry pi. However, this will not be enough to provide the IP camera with Internet. We will run the following script and that will allow us to provide internet access to the IP smart camera

	pi@raspberrypi	eth0: Configured 192.168.4.1/24
File Edit Tabs Help		wlan0: Associated with flat 6 wlan0: Configured 192.168.18
<pre>pi@raspberrypi:~ \$ sudo apt-ge Reading package lists Done Building dependency tree Reading state information D dnsmasq is already the newest 0 upgraded, 0 newly installed, pi@raspberrypi:~ \$ sudo nano / pi@raspberrypi:~ \$ sudo nano /</pre>	Done version (2.80-1 , 0 to remove an /etc/dhcpcd.conf tc/dnsmasq.conf	+rpt1+deb10u1). d 208 not upgraded. /etc/dnsmasq.conf.bak f

Figure 5-6 Script file to provide access to camera

With this done, the IP smart camera will connect to the server and start responding on the app

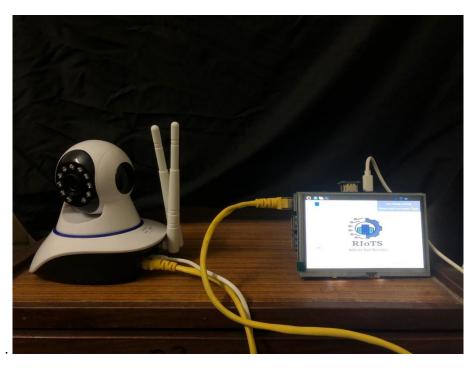


Figure 5-2 IP Camera and Raspberry connected via Ethernet

#### **5.3. Chapter Summary**

We secure the IP camera by using raspberry pi. In doing this, the raspberry pi is effectively acting as the stand in for the IP camera in the network and as a result keeping it safe from hackers. Scanning the network in any way does not show the IP smart camera as a host and hence keeping it safeguarded.

📉 🖿 🗏	🌢 ⊨ v   <u>1</u> 2 3 4   <u>@</u>		- Latin karat
🤮 🔳 લ 🕮		Ettercap 0.8.3.1 (EB)	0 🖲 : 🔹 🕷
IP Address	MAC Address Description		
92.168.18.1	C0:F6:C2:5D:FF:B8		
92.168.18.11	EA:45:D9:8A:22:F3		
92.168.18.20	1A:CF:D5:43:B4:BE		
92.168.18.35	6C:5A:B0:9F:FC:4B		
92.168.18.39	A0:57:E3:3C:17:48		
92.168.18.46	3C:A6:16:BF:BB:3F		
92.168.18.48	82:24:DA:9C:D9:CF		
92.168.18.61	B8:27:EB:54:D7:4D		
92.168.18.71	DC:A2:66:47:4D:D9		
92.168.18.76	86:CD:59:0D:B4:00		
92.168.18.78	76:38:32:50:CE:D7		
92.168.18.87	10:44:00:DC:A1:1D		
92.168.18.89	0C:54:15:D4:47:07		
92.168.18.92	06:8F:2A:C5:34:7B		
92.168.18.93	60:57:18:B6:B3:4F		
92.168.18.94	BA:75:DC:B9:A0:5C		
92.168.18.99	26:21:77:1A:FF:A3		
92.168.18.122	FC:77:74:6A:00:8B		
92.168.18.133	4A:37:04:13:41:2C		
92.168.18.134	E4:B3:18:EA:47:78		
80::1	C0:F6:C2:5D:FF:B8		
e80::38d3:5d77:2fff:	53ab 84:A6:C8:97:57:11 DESKTOP-PGJCH2R.local		
92.168.18.143	84:A6:C8:97:57:11		
92.168.18.151	6C:5A:B0:9F:FC:1A		
	Delete Host	Add to Target 1	Add to Target 2
ua: no scripts were sp	ecified, not starting up!		
arting Unified sniffin Indomizing 255 hosts	- i for scanning		flat 6 Internet access
canning the whole ne 2 hosts added to the h	tmask for 255 hosts osts list		VirtualBox Host-Only Network No internet access

Figure 5-3 Network Scan

As it can be seen the MAC address of the IP camera (7C:A7:B0:DA:86:F2) is not displaying upon scanning even though we are on the same network as the IP smart camera.

# CHAPTER 6: CONCLUSION AND ENHANCMENETS

# CHAPTER 06 CONCLUSION AND ENHANCEMENTS

#### **6.1.** Chapter Overview

The following chapter elucidates the enhancements and future work to be performed in order to render RIoTS more secure and advanced. It also concludes the whole document consists of RIoTS complete information.

#### **6.2.** Conclusion

According to the recently survey conducted in which about 21 organizations that consisted of jazz, Ufone MIS cell etc. We asked them have your organizational equipment involve IoT are secure or have you aware of any service that providing security to your organizational data. On which 92% answered No. Now this whole is a lot to accomplish ,so we in our limited scope, could get a hold of relatively limited and inexpensive IoT devices which actually act as a building block of smart cities.

Features	RIoTS
Real time information	$\checkmark$
Data privacy	$\checkmark$
IoT Device security	$\checkmark$
secrecy	$\checkmark$

When we talk about sustainable smart cities then encompass smart transportation, smart agriculture, smart homes. Now this whole is a lot to accomplish, so we in our limited scope, could get a hold of relatively limited and inexpensive IoT devices which actually act as a building block of these smart cities.

For instance, in the smart cities of Gwadar, Karachi and swat there are definitely cameras and lightening systems and in smart homes people have air conditioners, simply people wanting to smooth to better their lives. In a general environment these IoT devices communicate with their routers gateway towards the internet. Now they are communicating a lot valuable & critical user data without check, so our solution works in a way that it ensures real-time IoT security, the solution sits behind these devices and the network gateway in order to analyze what information goes out of the network, and how it works basically is that captures the traffic and analysis that, sees if there are weak passwords that can be compromised, it checks whether devices are vulnerable to DoS attacks, whether remote code can be executed over them, whether the communication their using with the rest of the internet is unencrypted or not, if information is being leaked and whether there are configuration issues that the attacker can exploit because even one of these internal IoT device is compromised then it will act as a pivot point to launch wider range of attacks on rest of the IoT network. Now this is the overall technical working of the device, and besides this RIoTS can work in a different fashion as well by just checking the security configurations, the passwords and encryption and all that stuff of the in its individuality without being applied to the network.

#### **6.3. Enhancements and Future Work**

- RIoTS meaning it can be applied on a whole network and also a device can be tested under the umbrella of RIoTS to reveal its weaknesses, it can analyze network traffic in real time, it covers wide range of attacks.
- This product requires basic technical knowledge with no extensive training required that the on-hand network administrator can handle it.
- RIoTS can work in a different fashion as well by just checking the security configurations, the passwords and encryption and all that stuff of the in its individuality without being applied to the network

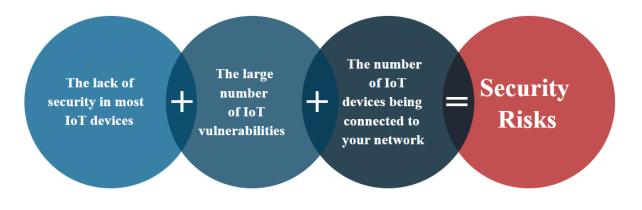


Figure 6-1 Security Risks

#### **6.4. Chapter Summary**

The current section concludes the thesis by implying the future work to be done in order to enhance the scope of our project and summarizes the entire documentary by shedding light on the core and important characteristics of the conversed project, RIoTS.

#### **REFERENCES:**

[1] ieeexplore.ieee.org/abstract/document/8697723

[2] ijariie.com/AdminUploadPdf/SMART\_BULB\_USING\_IOT\_ijariie9666.pdf

[3] ieeexplore.ieee.org/abstract/document/7804660

[4] Khalid Shahbar A. Nur Zincir-Heywood, "Benchmarking Two Techniques for IoT Classification, Flow Level and Circuit Level attacks", from Faculty of Computer Science, Dalhousie University, Halifax, Canada

[5] Jia Lingyu, Liu Yang, Wang Bailing, Liu Hongri, Xin Guodong, "A Hierarchical Classification Approach for Tor Anonymous Traffic analysis", in Proceedings of the 2017 9th IEEE International Conference on Software and Network security.

[6] Priya Mayank, A. K. Singh, "IoT devices Traffic Identification", in Proceedings of the 2017 7th International Conference on Communication Systems and Network Technologies.

[7] Khalid Shahbar, A.Nur Zincir Heywood, "Traffic Flow Analysis of IP camera".

[8] https://www.techrepublic.com/article/report-smart-bulbs-have-a-major-securityproblem/

[9] https://www.techrepublic.com/topic/security/

[10] https://www.techrepublic.com/topic/tech-and-work/

[11] https://www.researchgate.net/publication/344490696\_Smart\_Lamp\_or\_Security\_Camera\_Automatic\_Identification\_of\_IoT\_Devices

[12] Ibrahim Ghafir \*, Jakub Svoboda † and Vaclav Prenosil, "IOT-BASED
 MALWARE", \* Faculty of Informatics, Masaryk University† Institute of Computer
 Science, Masaryk University Brno, Czech Republic.

[13] https://opensource.com/resources/linux

[14] https://www.quora.com/What-is-socket-programming-a-socket-server-and-a-socket-client

[15] https://en.wikipedia.org/wiki/Windows\_10

[16] https://www.plesk.com/blog/various/what-is-linux/

[17] Khalid Shahbar, A.Nur Zincir Heywood, "Traffic Flow Analysis of TOR Pluggable Transports".

Riot	s Final			
ORIGIN	ALITY REPORT			
SIMILA	<b>3%</b> ARITY INDEX	<b>11%</b> INTERNET SOURCES	<b>7%</b> PUBLICATIONS	<b>0%</b> STUDENT PAPERS
PRIMAR	RY SOURCES			
1	WWW.ri	/erpublishers.co	m	2%
2	WWW.SC	rce		1%
3	digital.li	i <mark>brary.ucf.edu</mark>		1 %
4	Jadhav, Young H vulnera 2018 20	h Seralathan, Ta Jonathan Myers Ho Kim, Jeong No bility: A case stu Oth International ed Communicati , 2018	, Jaehoon Pau eyo Kim. "IoT s idy of a Web c Conference o	I Jeong, security amera", on
5	origin.g	eeksforgeeks.or	g	1%
6	WWW.CC	oursehero.com		1 %
7	raspber	rytips.com		1 %

8	Submitted to Higher Education Commission Pakistan Student Paper	<1%
9	www.honorbuy.com	<1%
10	iarjset.com Internet Source	<1%
11	Jeff Cicolani. "Beginning Robotics with Raspberry Pi and Arduino", Springer Science and Business Media LLC, 2021 Publication	<1%
12	dir.indiamart.com Internet Source	<1%
13	spectrum.library.concordia.ca	<1%
14	industrialinternetofthings.home.blog	<1%
15	dx.doi.org Internet Source	<1%
16	<b>ijisrt.com</b> Internet Source	<1%
17	onlinelibrary.wiley.com	<1%
18	open.library.ubc.ca	<1%

19	scholar.ppu.edu Internet Source	<1 %
20	samenacouncil.org	<1 %
21	udemycoupon.learnviral.com	<1 %
22	Mohamed Abomhara, Geir M. Koien. "Security and privacy in the Internet of Things: Current status and open issues", 2014 International Conference on Privacy and Security in Mobile Systems (PRISMS), 2014 Publication	<1 %
23	www.r4igold.co.za	<1 %
24	hdl.handle.net Internet Source	<1 %
25	mafiadoc.com Internet Source	<1 %
26	repository.wit.ie	<1 %
27	repository.nwu.ac.za	<1 %
28	Mohamed Abomhara, Geir M. K�ien. "Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and	<1 %

# Attacks", Journal of Cyber Security and Mobility, 2015

Publication

29	"Smart Sensors at the IoT Frontier", Springer Science and Business Media LLC, 2017 Publication	<1%

30 Orlando Arias, Jacob Wurm, Khoa Hoang, Yier Jin. "Privacy and Security in Internet of Things and Wearable Devices", IEEE Transactions on Multi-Scale Computing Systems, 2015 Publication

Exclude quotes	On Exclu	lude matches	Off
Exclude bibliography		idde materies	On