

# **HARDWARE BASED NETWORK SCANNING DEVICE**



By  
CSUO ABDULLAH ZULFIQAR  
GC MUAZ TARIQ  
GC MUHAMMAD BILAL YOUNUS

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 BE Degree in Electrical  
(Telecommunication) Engineering

JUNE, 2018

## CERTIFICATE OF CORRECTNESS AND APPROVAL

It is certified that the work contained in this thesis “Hardware Based Network Scanning Device ” was carried out by Abdullah Zulfiqar, Muaz Tariq and Muhammad Bilal Younus under the supervision of Lecturer Waleed Bin Shahid, for the partial fulfillment of Degree of Electrical (Telecommunication) Engineering, is correct and approved.

Approved by

---

(Asst Prof. Waleed Bin Shahid)

Department of Information Security Project

Directing Staff (DS)

Military College of Signals (MCS, NUST)

Dated: \_\_\_\_ July, 2018

## **ABSTRACT**

Network Security plays a vital role in modern communication systems. Surveillance is the demonstration of finding targets and creating techniques important to assault those objectives effectively. In order to accomplish this we need to collect a lot of information e.g. Physical location of the target, Data about users at the facility, Administrative shortcuts, Operating systems, Network structure etc. Nowadays networks are being analyzed using various network scanning tools (devices) but for every operation the devices are supposed to be a part of network. This can be a hurdle in an era of vast artificial intelligence. These purpose built devices are costly as well. Our brainstorming results in designing a specific cheap and remote Network Scanning Device running pre-defined attacks using Raspberry Pi and Kali Linux (OS) scanning and probing tools. This device will be controlled by a server over a remote connection. The gadget can be used inline or as a totally independent in network. After performing the pre-defined attacks, it will summarize the entire operation in a short report. However, the user will have the complete control to include or exclude the attacks for a specific test.

## **DECLARATION**

No bit of this work displayed in this exposition has been submitted in help of another honor or capability either at this foundation or somewhere else.

## **DEDICATION**

"Allah Almighty for His incalculable endowments

Teachers and companions for their assistance

What's more, our dearest guardians for their supplications and support”

## **ACKNOWLEDGEMENT**

Praise is to Almighty Allah Who guided and enabled us to undertake this project.

We would like to thank our project supervisor Lecturer Waleed Bin Shahid for his assistance, supervision and generous support throughout our Final Year Project. We would like to extend our gratitude towards Lecturer Narmeen Shafqat of Department of Information Security and Lecturer Aimen Aakif of Department of Electrical Engineering, Military College of Signals (MCS, NUST) for their help and guidance throughout this time period.

We would like to thank lab staff, our seniors, friends and all those who have helped us, either directly or indirectly, in the completion of this project.

## TABLE OF CONTENTS

<b>1. INTRODUCTION</b> .....	1
1.1 Overview .....	1
1.2 Problem Statement .....	1
1.3 Approach .....	1
1.4 Objectives.....	2
<b>2. BACKGROUND</b> .....	3
2.1 Existing Literature.....	3
2.2 Use of Existing Literature .....	4
<b>3. REQUIREMENTS AND SPECIFICATIONS</b> .....	5
3.1 Raspberry Pi .....	5
3.2 Linux-Back Track (Kali Linux) .....	6
<b>4. METHODOLOGY</b> .....	10
4.1 Basic Network Scanning .....	10
4.1.1 Establishing Goal .....	10
4.1.2 Reconnaissance .....	10
4.1.3 Exploitation.....	10
4.1.4 Taking Control .....	11
4.1.4 Pivoting .....	11
4.1.5 Reporting.....	11
4.2 Remote Network Scanning.....	11
<b>5. ATTACKING TOOLS</b> .....	14
5.1 Kismet .....	14
5.2 Aircrack-ng.....	14
5.3 Nmap .....	14
5.4 Setoolkit (social engineering toolkit).....	16
5.5 Dsniff.....	17
5.6 Zenmap.....	18
5.7 Wireshark .....	18
5.8 Ettercap.....	19
<b>6. SCOPE</b> .....	21
<b>7. FUTURE WORK</b> .....	22

<b>8. REFERENCES.....</b>	<b>23</b>
<b>9. APPENDIX A .....</b>	<b>25</b>
<b>10. APPENDIX B .....</b>	<b>31</b>



## LIST OF ILLUSTRATIONS

Figure 1:TCP/IP Model.....	4
Figure 2: Raspberry Pi 3 .....	6
Figure 3: USB to Ethernet Converter.....	6
Figure 4: Network Diagram .....	12
Figure 5: Raspberry Pi and C&C Server.....	13
Figure 6: Nmap .....	15
Figure 7: Output file for Nmap .....	16
Figure 8: Setoolkit.....	17
Figure 9: Zenmap .....	18
Figure 10: Wireshark .....	19
Figure 11: Ettercap.....	20
Figure 12: Ettercap Results .....	20
Figure 13: Automation Script .....	32
Figure 14: Automation Script.....	32

## **LIST OF ABBREVIATIONS**

OS	Operating System
OSI	Open Systems Interconnection
Wi-Fi	Wireless Fidelity
IP	Internet Protocol
GHz	Giga Hertz
LAN	Local Area Network
GB	Giga Bytes
RAM	Random Access Memory
USB	Universal Serial Bus
GPIO	General Purpose Input Output
HDMI	High Definition Multimedia Interface
mm	millimeter
SD	Storage Device
FHS	File system Hierarchy Standard
ARM	Advanced RISC Machines
RISC	Reduced Instruction Set Computer
C&C	Command and Control
GUI	Graphical User Interface
WPA	Wireless Protected Access

PSK	Pre-Shared Key
WEP	Wired Equivalent Privacy
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
BSD	Berkeley Software Distribution
ICMP	Internet Control Message Protocol
DNS	Domain Name Server
TOS	Type of Service

# 1. INTRODUCTION

## 1.1 Overview

Network Security is major concern in modern communication systems. In order to assure a network as secured a lot of scanning and attacking operations are performed on it. These operations result in indicating all the vulnerabilities and loopholes from where a network can be intruded. By overcoming those, we can make sure a particular network is secured.

## 1.2 Problem Statement

Nowadays network scanning or monitoring attacks are performed to check the security of a network but there are various issues in these attacks. Firstly the network scanning device itself should be a part of that Network. Secondly these devices available are costly or the process itself is expensive in nature. Also if the target network is distant from the attacker, it will be highly expensive as we'll have to move closer to the target in order to perform such attack, resulting in extra expense as well. Live attacks need a person to monitor them thoroughly for the entire period of time and then indicate the shortcomings in the network.

## 1.3 Approach

A hardware device (Raspberry Pi) having internet connection and can access external storage. Software applications are required to perform reconnaissance; scanning, sniffing and enumeration will be installed on the device. A report on loopholes and vulnerabilities in the network/systems(s) will be generated. Different latest attacks (embedded in meta-

split etc.) will be launched on the network/system(s) to perform the above mentioned tasks. A thorough report will be generated at the end of the process. The device will be used for both offensive and defensive purposes. It will solve the problems of cost, management and management of reports.

## 1.4 Objectives

This project is based on the concepts of Computer & Communication Networks and Network Security. One main objective of this project is to make a network scanning device which should go undetected in a network. This is because detection of such device can cause a huge problem during intelligence or spying applications. A network scanning suite will be designed for Raspberry Pi with attacks embedded in it. The device should also be customizable in terms of embedded attacks. For a specific network, user should be able to add or remove attacks. Our main focus is to provide a network scanning device for military, intelligence, banking and other general applications at low cost.

## 2. BACKGROUND

### 2.1 Existing Literature

- With the progress of advancement, everything now-a-days is on the web. We approach our data everywhere. Affiliations are working in frameworks. There is cover and intra various leveled data correspondence. With this movement, the stress of data security has also extended. To check data security, one needs to perform strike without any other person orchestrate. There are affiliations which give these organizations. By and by there is a resistance among those affiliations. Essential concern is incurred significant injury. Which association gives organizations easily, it is contracted.
- TCP/IP show changed over the 7 layers of OSI show into 5 layers which are immediately depicted as

#### **Layer 5**

##### **Application**

Indicates how a specific application utilizes a system.

#### **Layer 4**

##### **Transport**

Indicates how to guarantee solid transport of information.

#### **Layer 3**

##### **Network/Internet**

Indicates packet arrange and directing.

#### **Layer 2**

##### **Data Link**

Indicates how to guarantee solid transport of information

## Layer 1

### Physical

Indicates the fundamental system hardware.

Layer #	Layer Name	Protocol	Protocol Data Unit	Addressing
5	Application	HTTP, SMTP, etc...	Messages	n/a
4	Transport	TCP/UDP	Segments/ Datagrams	Port #s
3	Network or Internet	IP	Packets	IP Address
2	Data Link	Ethernet, Wi-Fi	Frames	MAC Address
1	Physical	10 Base T, 802.11	Bits	n/a

*Figure 1:TCP/IP Model*

- Attacks can be implanted in a gadget for auto activity. Once introduced and started, testing gadget can perform assaults consequently.
- Raspberry Pi can be controlled remotely finished a remote association (Wi-Fi/4G Dongle) by any PC.

## 2.2 Use of Existing Literature

- If we could control our gadget remotely, it could diminish our cost. What's more, if this gadget is a Raspberry Pi, this cost will diminish to simply \$50.
- If we see the TCP/IP display, Network Layer allocates IP deliver to the gadget. Taking a shot at information connect layer won't allocate IP deliver to the address. This property can be utilized to influence a gadget to go undetected in a system. This is vital in knowledge based applications and spying.

## 3. REQUIREMENTS AND SPECIFICATIONS

### 3.1 Raspberry Pi

Raspberry Pi 3 is the device which we will be using in this project. Following are the specifications of Raspberry Pi:

- A 1.2GHz 64-bit quad-centre ARMv8 CPU
- 802.11n Wireless LAN
- 1GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5mm sound jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card space
- Price: \$35

Availability of Ethernet Port and Wi-Fi card make us able to test both wired and wireless networks. For in-line connection, USB port will be converted to Ethernet port using standard USB to Ethernet converter.





*Figure 2: Raspberry Pi 3*



*Figure 3: USB to Ethernet Converter*

### 3.2 Linux-Back Track (Kali Linux)

Kali Linux is a Debian-based Linux transport went for front line Network Scanning and Security Auditing. Kali contains a few hundred devices went for different data security assignments, for example, Network Scanning, Forensics and Reverse Engineering.

Kali Linux was discharged on the thirteenth March, 2013 as an entire, through and through revamp of Back Track Linux, holding fast totally to Debian advancement gauges.

**In excess of 600 Network Scanning devices included:** Ensuing to researching each instrument that was joined into Back Track, we killed a mind boggling number of contraptions that either essentially did not work or which copied particular devices that gave the same or identical handiness. Motivations behind eagerness on what's combined are on the Kali Tools site.

**Free (as in lager) and dependably will be:** Kali Linux, as Back Track, is absolutely complimentary and reliably will be. You will never anytime need to pay for Kali Linux.

**Open source Git tree:** We are centered around the open source change show and our progression tree is available for all to see. Most of the source code which goes into Kali Linux is available for any person who needs to change or alter packs to suit their specific needs.

**FHS consistent:** Kali holds quick to the File system Hierarchy Standard, empowering Linux customers to easily discover parallels, reinforce records, libraries, et cetera.

**Colossal remote gadget support:** A common staying point with Linux transports has been kept up for remote interfaces. We have gathered Kali Linux to help the best number of remote contraptions as we can, enabling it to run truly on a wide game

plan of rigging and making it incredible with various USB and assorted remote gadgets.

**Custom piece, fixed for injection:** As Network Scanners, the progression amass frequently needs to do remote examinations, so our bit has the latest mixture patches included.

**Created in a protected environment:** The Kali UNIX framework group is shaped of a minor low horde of individuals United Nations office region unit the sole one's trusty to submit bundles and move with the stores, the majority of that is finished misuse various secure conventions.

**GPG marked bundles and stores:** Each package in Kali Linux is set apart by each individual designer who created and submitted it, and the vaults in this manner sign the groups as well.

**Multi-dialect bolster:** Each bundle in Kali Linux is separate by every individual fashioner who made and submitted it, and the vaults along these lines sign the gatherings too.

**Totally customizable:** We totally fathom that few out of every odd individual will agree with our framework decisions, so we have made it as straightforward as plausible for our more gutsy customers to re-try Kali Linux to their adoring, the separation down to the part.

**ARMEL and ARMHF bolster:** Since ARM-based single-board structures like the Raspberry Pi and Beagle Bone Black, among others, are winding up ceaselessly customary and humble, we grasped that Kali ARM support would ought to be as solid as we could supervise, with altogether working foundations for both ARMEL and ARMHF systems. Kali Linux is open on a wide accumulation of ARM contraptions and has ARM stores energized with the mainline scattering so instruments for ARM are empowered in conjunction with whatever is left of the stream.

## 4. METHODOLOGY

### 4.1 Basic Network Scanning

Following is the basic methodology which is employed while doing Network Scanning:

#### 4.1.1 Establishing Goal

Perceiving openings in security, affiliation can recognize the gap of the structure security and companions can develop an action expect to diminish the hazard with the help of penetration test.

To discover new threats, Network Scanning measures will help the relationship with finding the new risks. To revolve around inside security resources, A Penetration test and its security examination empower the relationship to focus inside security resources. To meet authoritative compliances, to find weakest association, Penetration test and security survey will assist the firm with locating the weakest association in their stunning structure and it will give standard security to each and every normal component. Give endorsement input, Penetration test pass on endorsement feedback to business substances and security structure that lead the relationship to reduce the risk in the execution.

#### 4.1.2 Reconnaissance

Observation is the demonstration of social affair fundamental information or knowledge on your objective. The information is assembled keeping in mind the end goal to better arrangement for your assault. Observation can be performed effectively (implying that you are straightforwardly contacting the objective) or latently (implying that your recon is being performed through a delegate).

#### 4.1.3 Exploitation

Subsequent to finding the vulnerabilities, we endeavor to abuse those vulnerabilities to rupture the framework and its security. For the Exploitation we utilize distinctive system and programming that are suggested for exploitative reason and are unreservedly accessible. In this stage, Network Scanner will attempt to recognize exercises for the different vulnerabilities found in the past stage.

#### 4.1.4 Taking Control

Keeping up get to requires making the strides engaged with having the capacity to be tenaciously inside the objective condition so as to assemble however much information as could reasonably be expected and work from the host for promote methods .The aggressor remain stealth in this stage, to not get captured while utilizing the host condition.

#### 4.1.4 Pivoting

Turning is the novel strategy of using a case (furthermore suggested as a 'plant' or 'strong balance') to have the ability to "move" around inside a framework. Basically using the chief exchange off to allow and even guide in the deal of other for the most part hard to achieve systems.

#### 4.1.5 Reporting

Uncovering stage is the last stage in the invasion test methodology. Declaring stage is to a great degree basic stage and this report will cover both organization and particular perspectives, give separated information about all revelations, figures with true blue graphs. Framework Scanner will give suitable presentation of the vulnerabilities and its impact to the matter of the goal affiliation.

## 4.2 Remote Network Scanning

Following will be the approach which will be utilized in our Remote Network Scanning Toolkit:

- The Network Scanning Toolkit will be introduced inline or independent in a system to be tried. Along these lines, it will catch the majority of the data trade and will infiltrate into the system. It will chip away at Layer 2 of correspondence demonstrate which is Data Link Layer.
- Remote Network Scanning Toolkit will be executed on Raspberry Pi and controlled through secure shell by a Command and Control Server.
- Linux will be the stage utilized for Testing.
- Once started, it will perform security appraisal of system utilizing pre-characterized abuses.
- At the end, a short report will be produced.

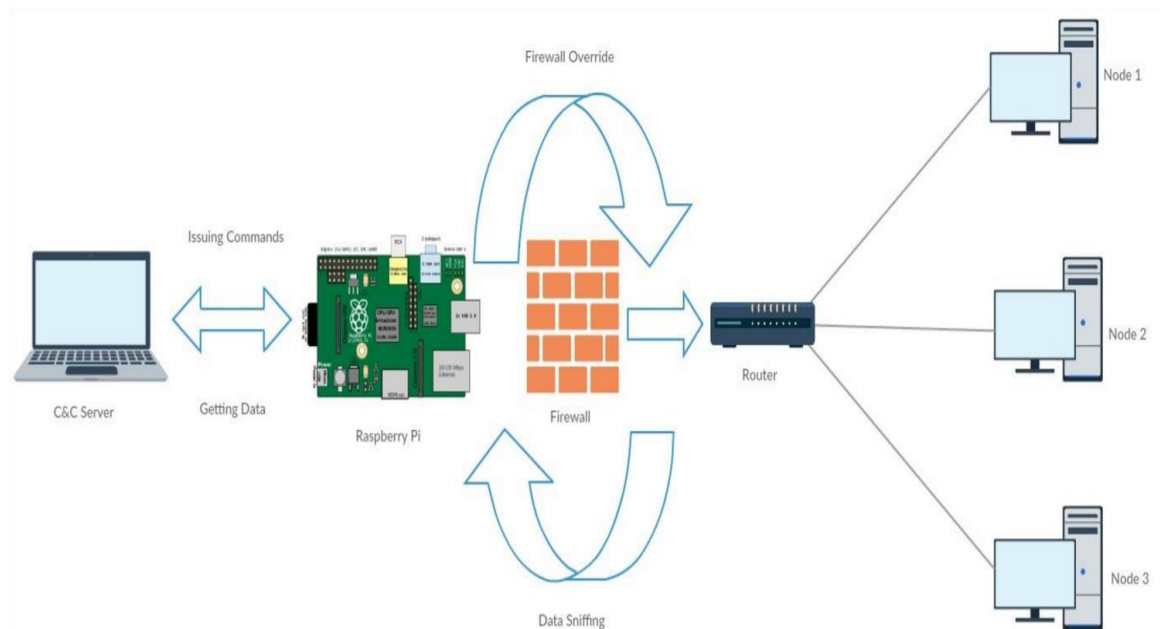


Figure 4: Network Diagram

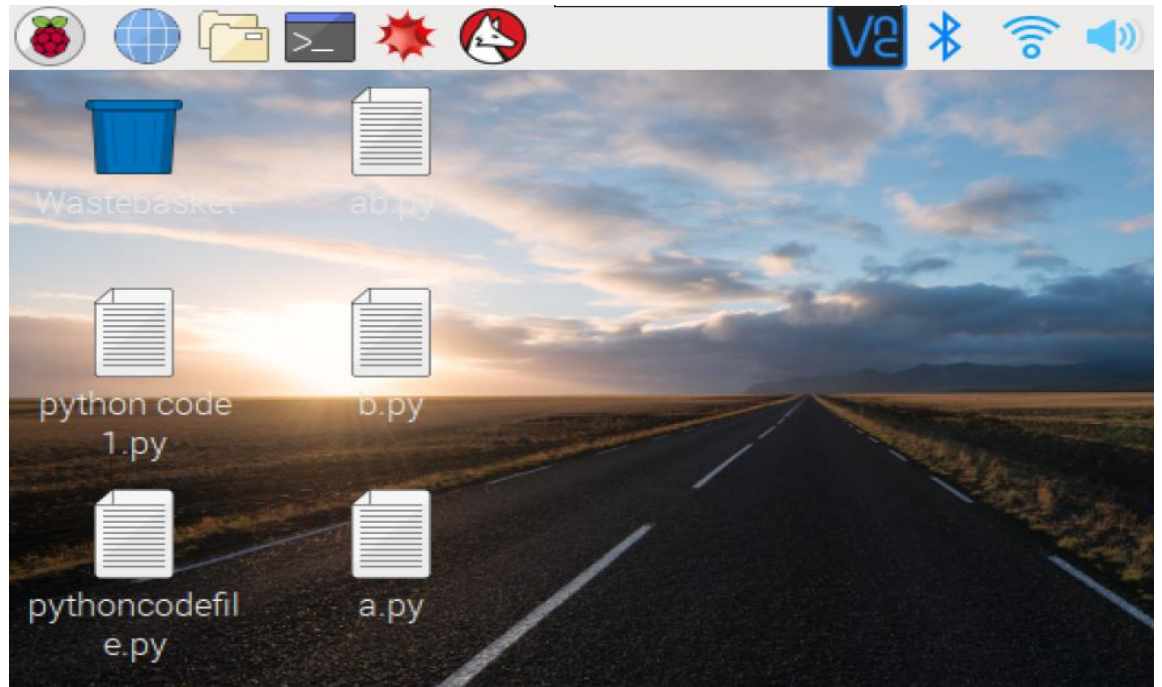


Figure 5: Raspberry Pi and C&C Server



## 5. ATTACKING TOOLS

Following attacking tools have been embedded in the Network Scanning suite for Raspberry Pi. These attacks are customizable for a given test.

### 5.1 Kismet

Kismet is an open source remote system analyzer running under the Linux frameworks. It can identify any 802.11 a/b/g remote systems around it. 802.11 a/b/g conventions are WLAN (Wireless Local Area Network) principles. Kismet distinguishes arrangements by inactively sniffing giving it the focal points to find the "shrouded" remote systems and acting naturally imperceptible. The kismet program is made by a server called "kismet\_server" and a customer "kismet\_client" which can associate with numerous servers.

### 5.2 Aircrack-ng

Aircrack-ng is a total group of devices to survey Wi-Fi organize security. All apparatuses are summoning line which takes into account substantial scripting. It works principally Linux yet in addition Windows, OS X and so on.

It centers around various zones of Wi-Fi security:

**Monitoring:** Packet capture and sending out data to content documents for further processing by external devices.

**Attacking:** Replay assaults, validation, counterfeit passageways and others through parcel infusion.

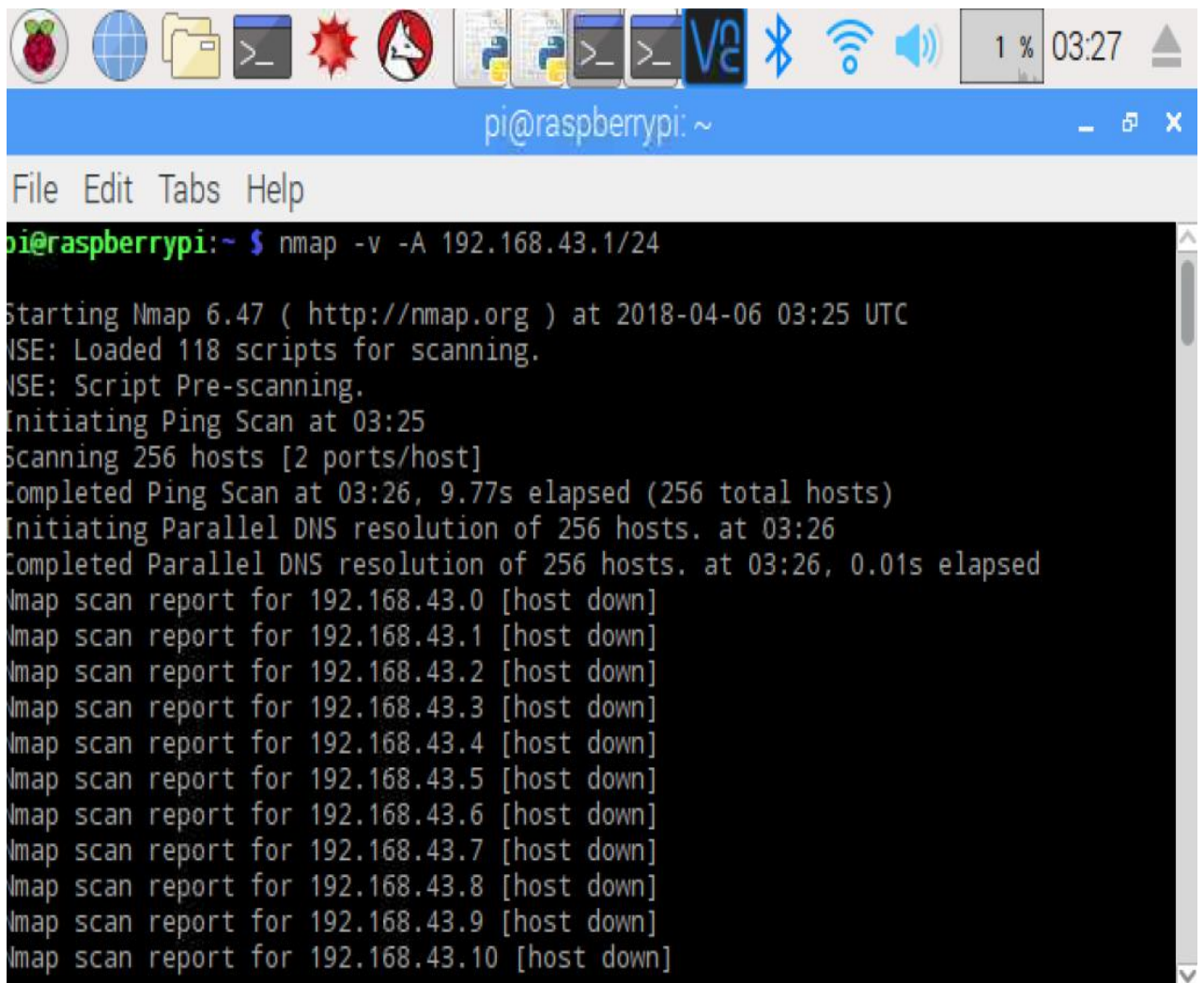
**Testing:** Testing Wi-Fi cards and driver capacities (clip and mix).

**Cracking:** WEP and WPA PSK (WPA 1 and 2).

### 5.3 Nmap

Nmap ("Network Mapper") is a free and open source utility for deal with examination and security researching. Nmap utilizes harsh IP bundles in novel approaches to manage

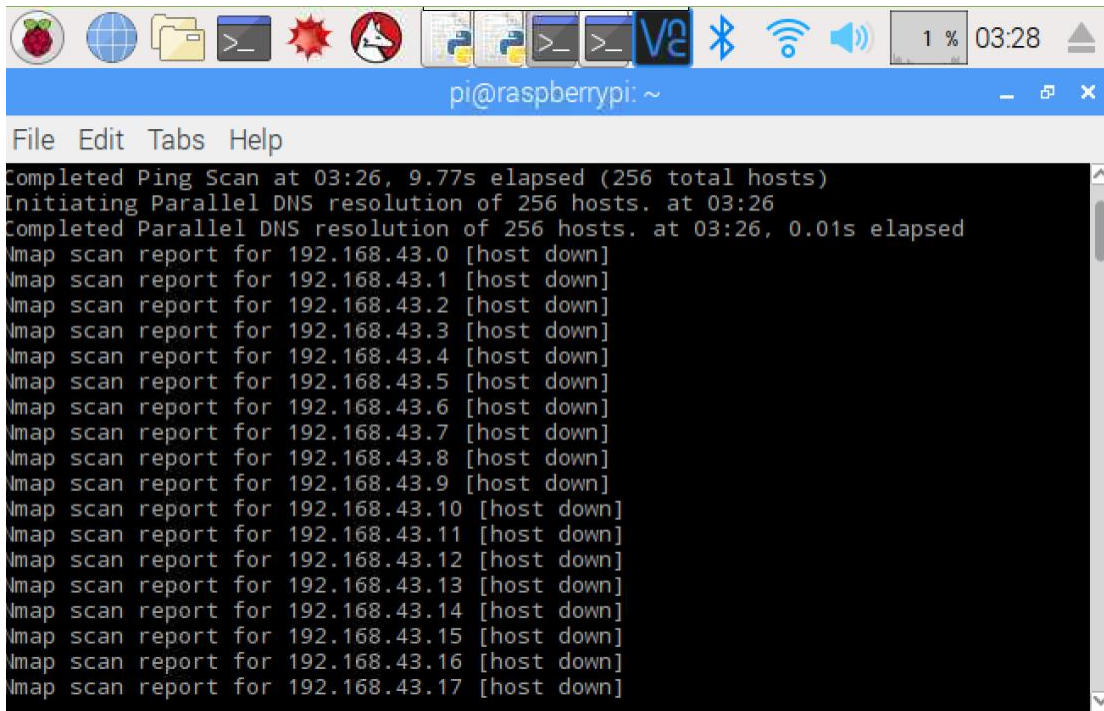
understand what has are open on the structure, what associations (application name and shape) those hosts are propelling, what working frameworks (and OS renditions) they are running, and what kind of package channels/firewalls are being used. It was proposed to quickly break down wide systems, in any case works fine against single hosts. Nmap keeps running on all basic PC working structures, and both support and graphical structures are accessible.



The image shows a terminal window on a Raspberry Pi. The window title is "pi@raspberrypi: ~". The terminal output shows the execution of the command "nmap -v -A 192.168.43.1/24". The output includes the following text:

```
pi@raspberrypi:~ $ nmap -v -A 192.168.43.1/24
Starting Nmap 6.47 ( http://nmap.org ) at 2018-04-06 03:25 UTC
NSE: Loaded 118 scripts for scanning.
NSE: Script Pre-scanning.
Initiating Ping Scan at 03:25
Scanning 256 hosts [2 ports/host]
Completed Ping Scan at 03:26, 9.77s elapsed (256 total hosts)
Initiating Parallel DNS resolution of 256 hosts. at 03:26
Completed Parallel DNS resolution of 256 hosts. at 03:26, 0.01s elapsed
Nmap scan report for 192.168.43.0 [host down]
Nmap scan report for 192.168.43.1 [host down]
Nmap scan report for 192.168.43.2 [host down]
Nmap scan report for 192.168.43.3 [host down]
Nmap scan report for 192.168.43.4 [host down]
Nmap scan report for 192.168.43.5 [host down]
Nmap scan report for 192.168.43.6 [host down]
Nmap scan report for 192.168.43.7 [host down]
Nmap scan report for 192.168.43.8 [host down]
Nmap scan report for 192.168.43.9 [host down]
Nmap scan report for 192.168.43.10 [host down]
```

Figure 6: Nmap



```
pi@raspberrypi: ~  
File Edit Tabs Help  
Completed Ping Scan at 03:26, 9.77s elapsed (256 total hosts)  
Initiating Parallel DNS resolution of 256 hosts. at 03:26  
Completed Parallel DNS resolution of 256 hosts. at 03:26, 0.01s elapsed  
Nmap scan report for 192.168.43.0 [host down]  
Nmap scan report for 192.168.43.1 [host down]  
Nmap scan report for 192.168.43.2 [host down]  
Nmap scan report for 192.168.43.3 [host down]  
Nmap scan report for 192.168.43.4 [host down]  
Nmap scan report for 192.168.43.5 [host down]  
Nmap scan report for 192.168.43.6 [host down]  
Nmap scan report for 192.168.43.7 [host down]  
Nmap scan report for 192.168.43.8 [host down]  
Nmap scan report for 192.168.43.9 [host down]  
Nmap scan report for 192.168.43.10 [host down]  
Nmap scan report for 192.168.43.11 [host down]  
Nmap scan report for 192.168.43.12 [host down]  
Nmap scan report for 192.168.43.13 [host down]  
Nmap scan report for 192.168.43.14 [host down]  
Nmap scan report for 192.168.43.15 [host down]  
Nmap scan report for 192.168.43.16 [host down]  
Nmap scan report for 192.168.43.17 [host down]
```

Figure 7: Output file for Nmap

#### 5.4 Setoolkit (social engineering toolkit)

Social engineering, concerning data security, infers mental control of individuals into performing activities or revealing riddle data. A sort of affirmation trap with a definitive target of data get-together, extortion, or framework get to, it separates from a conventional "con" in that by and large one of different means in a more diverse mutilation plot.

The enunciation "social working" as an appearing of mental control of a human, is also connected with the humanistic frameworks, at any rate its utilization has

gotten on among PC and data security masters.

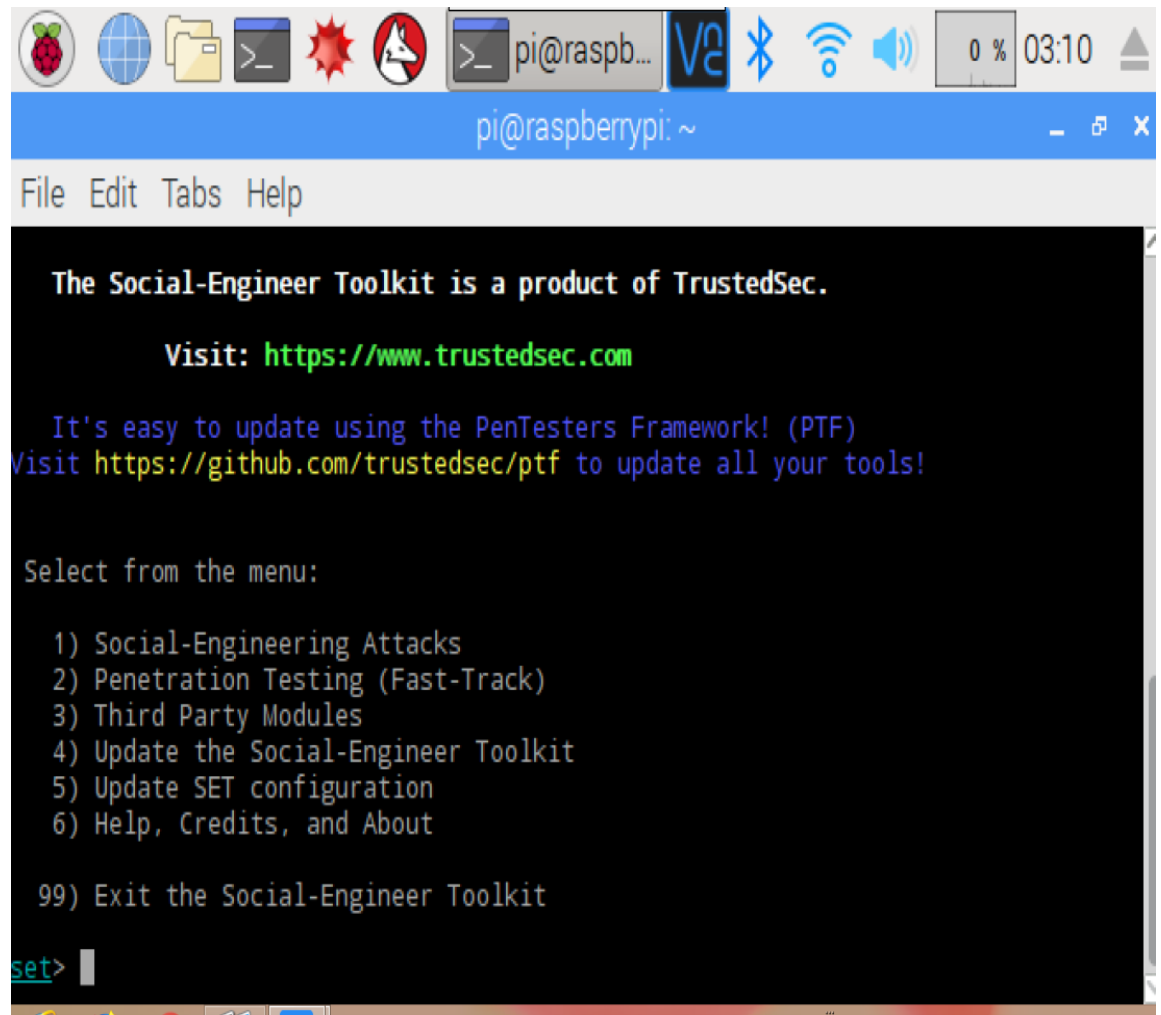


Figure 8: Setoolkit

## 5.5 Dsniff

The capacity to get to the crude parcels on a system interface, has for quite some time been an imperative device for framework and system directors. For troubleshooting purposes usually accommodating to take a gander at the system movement last possible minute level to perceive what is precisely being transmitted. Dsniff, as the name infers, is a system sniffer yet intended for testing of various sort.

## 5.6 Zenmap

Zenmap is the master graphical UI (GUI) for the Nmap Security Scanner. It is a multi-stage, free and open-source application wanted to make Nmap fundamental for tenderfoots to utilize while giving moved highlights to experienced Nmap clients. A critical piece of the time utilized breadths can be spared as profiles to make them simple to run more than once. A charge maker gifts shrewd making of Nmap orchestrate lines. Yield results can be spared and seen later. Spared yields can be emerged from each other with perceive how they change. The inevitable results as of late yields are anchored in an open database.

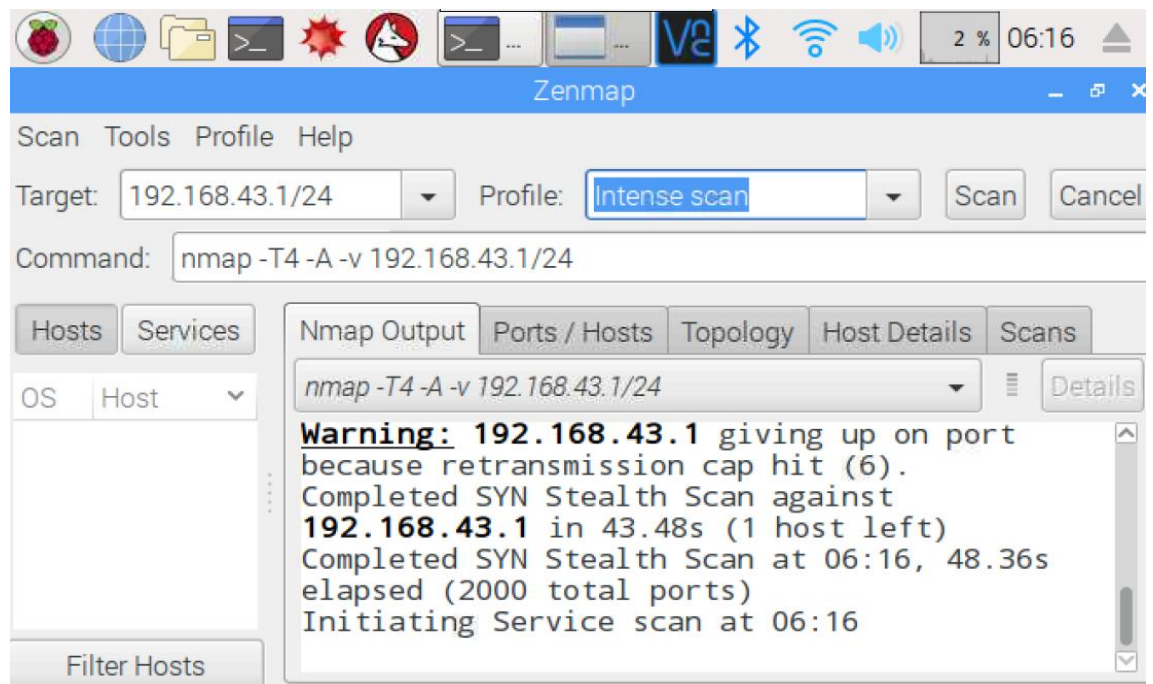


Figure 9: Zenmap

## 5.7 Wireshark

Wireshark is pervasive advancement analyzer which works in Deep examination of various customs, with all the more being joined dependably, for Live catch and isolated examination, for Standard three-sheet apportion, for Captured organize information can be investigated by techniques for a GUI, or through the TTY-mode TShark utility, Rich VoIP examination and Capture records compacted with gzip can be decompressed on the

fly..

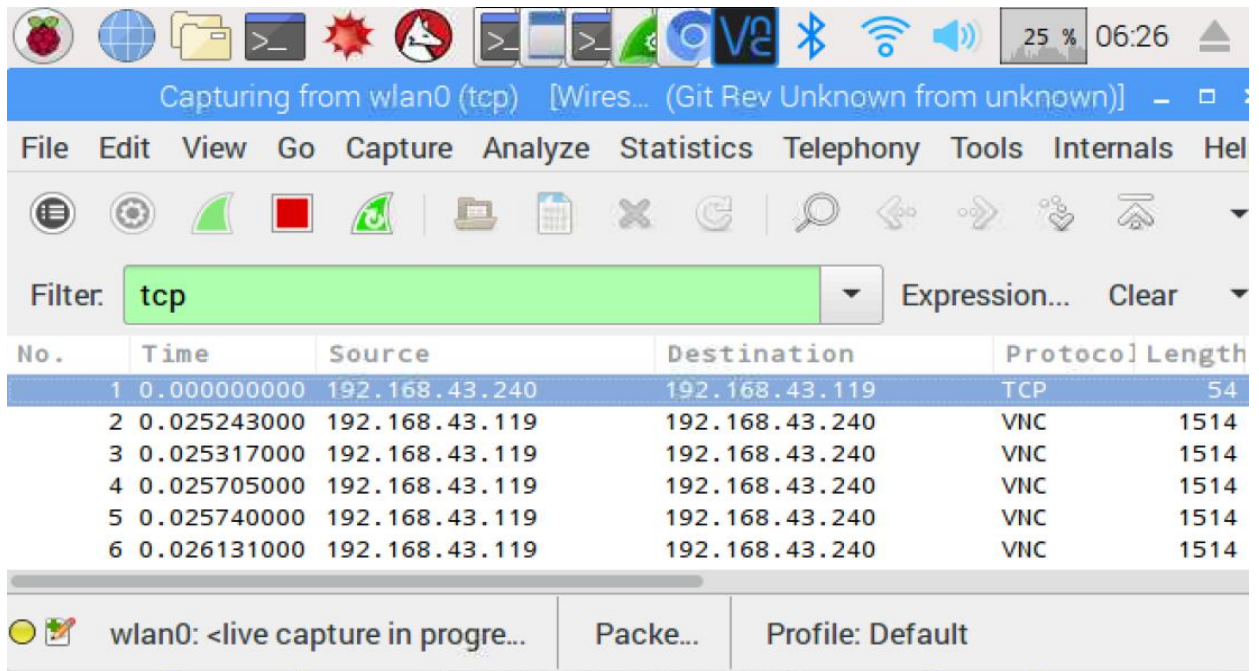


Figure 10: Wireshark

## 5.8 Ettercap

Ettercap is a flexible system control instrument. It utilizes its capacity to effectively perform man-in-the-center (MITM) assaults in an exchanged LAN condition as the platform for huge numbers of its different capacities. Once has embedded itself amidst an exchanged association, it can catch and look at all correspondence between the two casualties has, and thusly exploit different highlights.

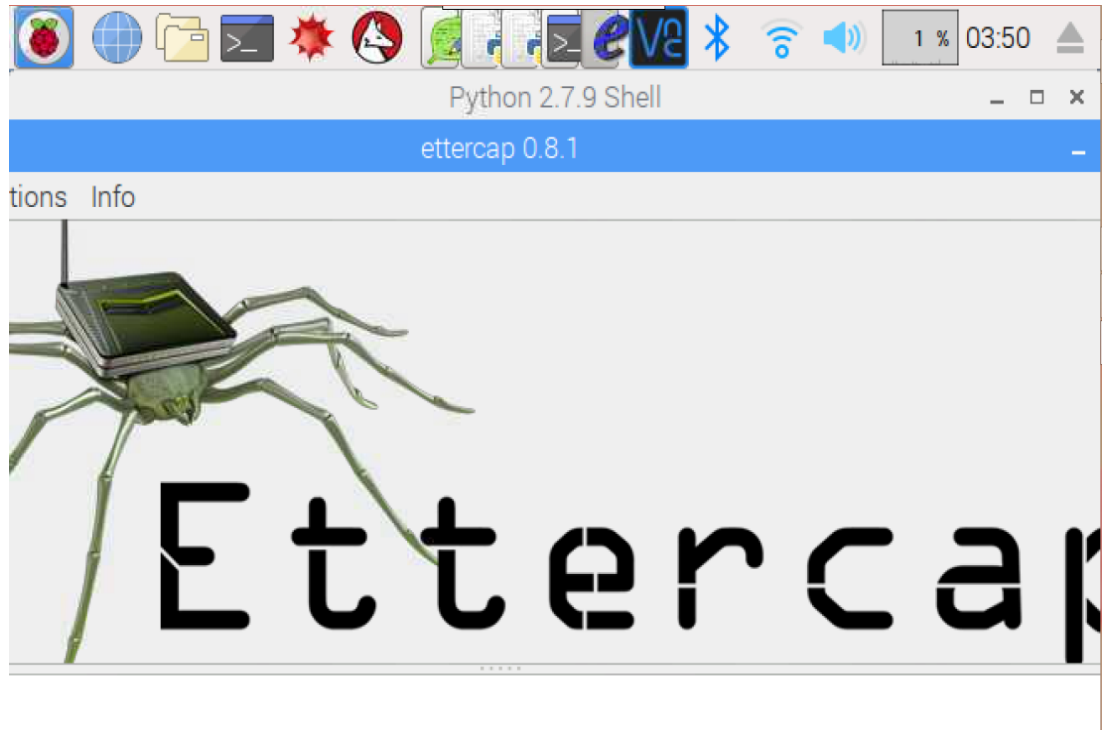


Figure 11: Ettercap

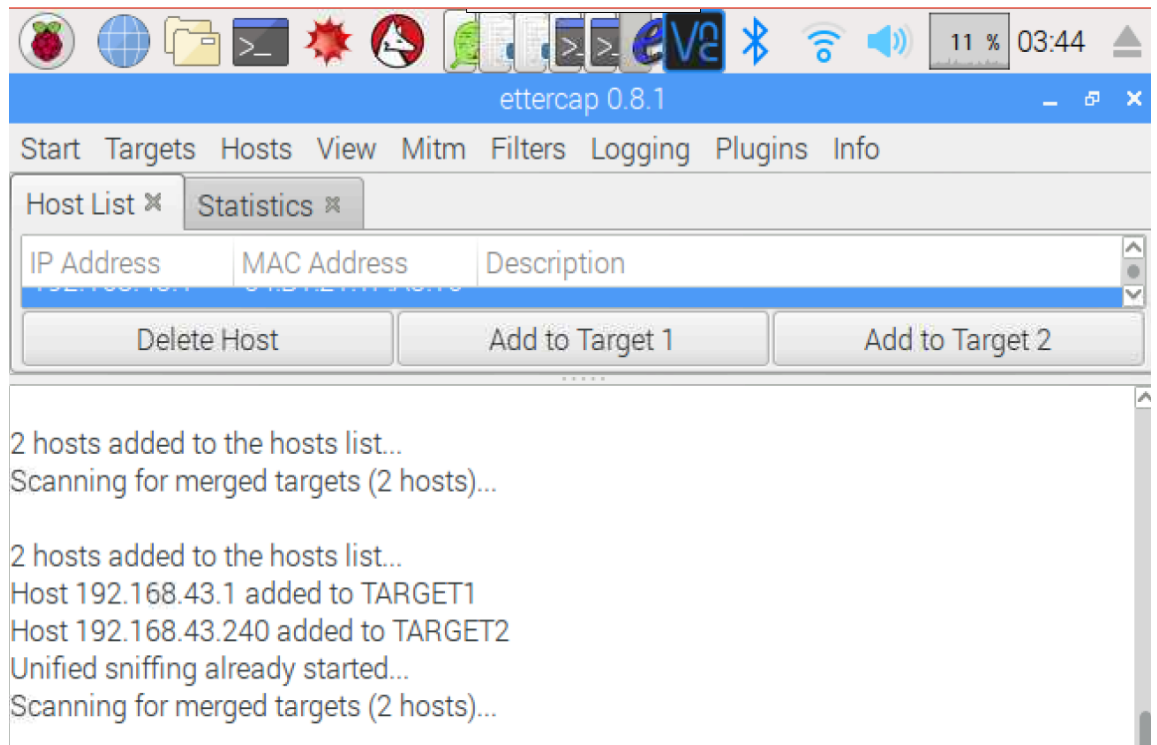


Figure 12: Ettercap Results

## 6. SCOPE

- **Testing Security Controls:** Penetration test tells us whether our system or network is exploitable or not.
- **Ensure System Security:** If we see any flaw in our system security, we can overcome that flaw in order to ensure system security.
- **Prevention from Data Breach:** Overcoming security flaws will prevent breach into our data by the hacker.
- **Banking and other networks:** Banking networks are most threatened by hackers. Testing of banking networks will ensure their security.
- **Spying:** One of fundamental extensions is spying of adversary's systems. Raspberry Pi is an extremely savvy gadget. It can be shrouded anyplace in the wires of system. System Scanner will be at home checking adversary's system.



## **7. FUTURE WORK**

This venture takes care of the issue of expense and development of labor. Once introduced in arrange, this gadget needs to dispatch assaults physically. This can be mechanized by composing a content by which gadget would have the capacity to naturally check, identify, endeavor and report the system vulnerabilities.

## **8. REFERENCES**

The Blueprint to Hacking by Cyber Punk Architects (2016)

Raspberry Pi User Guide, its authors are Eben Upton and Gareth Halfacree

# APPENDICES

## 9. APPENDIX A

### Python tools Installation

```
# Install Python tools
```

```
Echo "[+] Installing Perl/Python tools to /raspberry pi..." cp -a src/raspberry pi/ /  
chown -R root: root /raspberry pi/ chmod +x /raspberry pi/Cisco-auditing-  
tool/CAT chmod +x /raspberry pi/easy-creds/easy-creds.sh chmod +x /raspberry  
pi/goohost/goohost.sh chmod +x /raspberry pi/lbd/lbd.sh chmod +x /raspberry  
pi/sslstrip/sslstrip.py echo "[+] Perl/Python tools installed in /raspberry pi."
```

### SET Installation

```
# Install SET echo "[+] Installing latest SET framework to /raspberry pi..." git  
clone https://github.com/trustedsec/social-engineer-toolkit/ /raspberry pi/set/ cd  
src/pexpect-2.3/ python setup.py install cd ../.. Echo "[+] SET framework installed  
in /raspberry pi."
```

### NMAP Installation

```
#Install NMAP echo "Installing Nmap to /raspberry pi..."  
#First writes sudo su in the terminal to give root privileges or write sudo before  
every command  
Sudo su  
apt-get update  
apt-get upgrade  
apt-get installs Nmap  
echo" Nmap is installed in /raspberry pi."
```

## ZENMAP Installation

```
#Install ZENMAP echo "Installing Zenmap to /raspberry pi..."  
#First writes sudo su in the terminal to give root privileges or write sudo before  
every command  
Sudo su  
apt-get update  
apt-get upgrade  
apt-get installs Zenmap  
echo" Zenmap is installed in /raspberry pi."
```

## Wireshark Installation

```
#Install WIRESHARK echo "Installing Wireshark to /raspberry pi..."  
http://anonsvn.wireshark.org/wireshark/trunk/wireshark  
sudo su  
apt-get installs libtool  
apt-get installs bison  
apt-get installs flex  
./autogen.sh  
apt-get install libgtk-3-dev  
./configure  
make  
  
echo" Wireshark is installed in /raspberry pi."
```

## Ettercap Installation

```
#Install Ettercap echo "Installing Ettercap to /raspberry pi..."
```

```
Sudo apt-get install python-gtk2-dev libnet1-dev cmake flex libcap0.8-dev  
libncurses5-dev
```

```
Git clone https://github.com/Ettercap/ettercap.git
```

```
echo" Ettercap is installed in /raspberry pi."
```

## Kismet Installation

```
#Install Kismet echo "Installing Kismet to /raspberry pi..."
```

```
#!/bin/bash
```

```
#Install build tools and dependencies
```

```
apt-get -y update
```

```
apt-get -y install cmake libusb-1.0-0-dev make gcc g++ libbluetooth-dev  
libncurses5-dev libnl-dev \
```

```
pkg-config libpcap-dev python-numpy python-pyside python-qt4
```

```
#Get and build libbtbb
```

```
cd~
```

```
wgethttps://github.com/greatscottgadgets/libbtbb/archive/2015-10-  
R1.tar.gz -O libbtbb-2015-10-R1.tar.gz
```

```
tar xf libbtbb-2015-10-R1.tar.gz
```

```
cd libbtbb-2015-10-R1
```

```
mkdir build
```

```
cd build
```

```
cmake
```

```
Make
```

```
make install
```

```
Ldconfig
```

### **#Get and build Ubertooth tools**

```
cd~  
wget  
https://github.com/greatscottgadgets/ubertooth/releases/download/2015-10-R1/ubertooth-2015-10-R1.tar.xz -O ubertooth-2015-10-R1.tar.xz  
tar xf ubertooth-2015-10-R1.tar.xz  
cd ubertooth-2015-10-R1/host  
mkdir build  
cd build  
cmake  
Make  
make install  
Ldconfig
```

### **#Get and build Kismet and Ubertooth plugin**

```
cd~  
wget http://www.kismetwireless.net/code/kismet-2016-01-R1.tar.xz  
tar xf kismet-2016-01-R1.tar.xz  
cd kismet-2016-01-R1  
ln -s ../ubertooth-2015-10-R1/host/kismet/plugin-ubertooth ./  
./configure  
make deb  
make && make plugins  
make suidinstall  
make plugins-install  
#Set writeinterval  
sed -i"s/writeinterval=300/writeinterval=10/g" /usr/local/etc/kismet.conf
```

Echo” Kismet is installed in /raspberry pi.”

## Aircrack-ng Installation

#Install Aircrack-ng echo “Installing Aircrack-ng to /raspberry pi...”

### # Install the ssl development libraries

```
step_1(){  
    sudo apt-get -y install libssl-dev  
}
```

### # Download aircrack and compile

```
step_2(){  
    wget http://download.aircrack-ng.org/aircrack-ng-1.2-beta1.tar.gz  
    tar -zxvf aircrack-ng-1.2-beta1.tar.gz  
    cd aircrack-ng-1.2-beta1  
    Make  
    sudo make install  
}
```

### # Update aircrack

```
step_3(){  
    airodump-ng-oui-update  
}
```

### # Install IW

```
step_4(){  
    sudo apt-get -y install iw  
}
```

### # Start the software

```
final(){  
    airmon-ng start wlan0  
    airodump-ng mon0  
}
```

### # Call the functions

```
step_1  
step_2  
step_3  
step_4  
Final
```

□

echo” Aircrack-ng is installed in /raspberry pi.”



## Dsniff Installation

#Install Dsniff echo "Inst alling Dsniff to /raspberry pi..."

```
sudo apt-get update && apt-get upgrade
sudo apt-get install screen nodejsnodejs-legacy git npm
sudonpm install websocket
git clone https://github.com/samyk/poisonatp
Screen
sudo node backend_server.js
##piZero:
sudo apt-get update && apt-get upgrade
sudo apt-get -y install isc-dhcp-server dsniff screen nodejs git
git clone https://github.com/samyk/poisonatp
Update ws://YOUR.DOMAIN:1337 in backdoor.html to your public IP
ws://poisonatp.jgamblin.com:1337
#add this to /etc/network/interfaces:
auto usb0
    allow-hotplug usb0
    iface usb0 inet static
    address 1.0.0.1
    netmask 0.0.0.0

#Setup to run at boot:
sudo cp /home/pi/poisonatp/pi_startup.sh /etc/init.d/ &&sudochmod +x
/etc/init.d/pi_startup.sh
add this to /etc/rc.local right above exit 0:
/etc/init.d/pi_startup.sh &
#DHCP:
sudo cp -f dhcpd.conf /etc/dhcp/dhcpd.conf
add this to /etc/default/isc-dhcp-server:
INTERFACES="usb0"
```

□

echo" Dsniff is installed in /raspberry pi."

## 10. APPENDIX B

### Automation Code For Network Scanning On Raspberry Pi

```
import nmap

# initialize the port scanner

abdullahScan = nmap.PortScanner()

# scan localhost for ports in range 1-1000

abdullahScan. Scan('192.168.43.1/24', '1-1000')

print('/n/n/nDetails of Network: %s' % abdullahScan. Scan('192.168.43.1/24', '1-1000'))

print('\n\nSpecific Details\n')

# run a loop to print all the found result about the ports

for host in abdullahScan.all_hosts():

    print('\n\nHost : %s (%s)\n' % (host, abdullahScan[host].namesofhost()))

    print('\n\nState : %s\n' % abdullahScan[host].state())

        for proto in abdullahScan[host].all_protocols():

            print("")

            print('\n\nProtocol : %s\n' % proto)

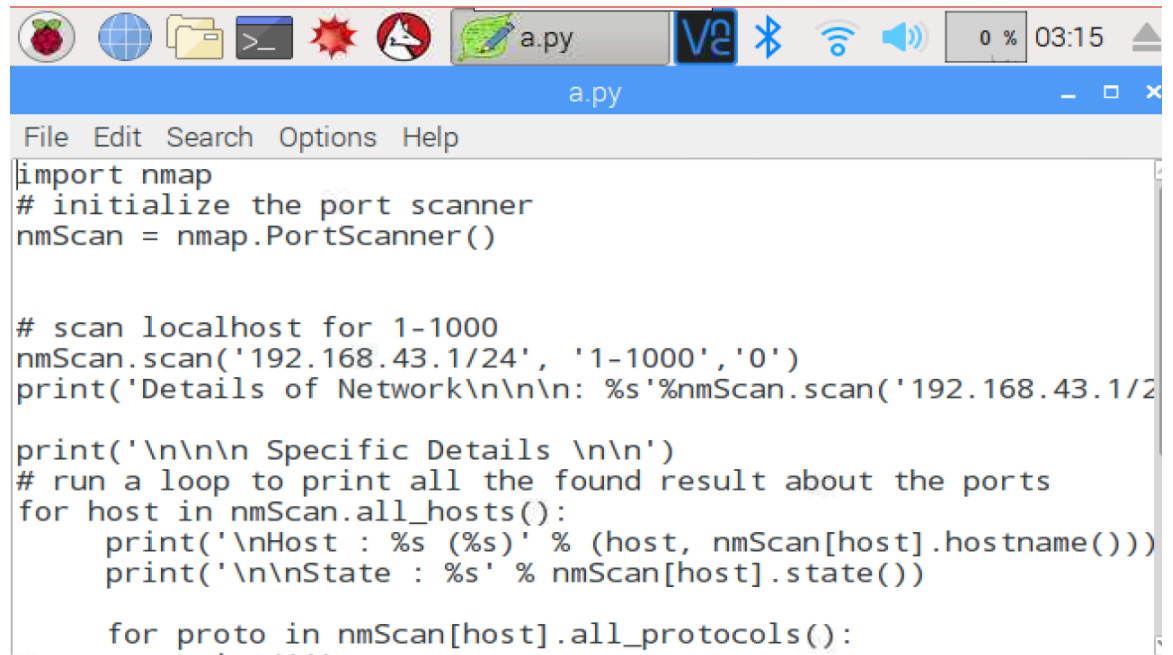
bilalport = abdullahScan[host][proto].keys()

bilalport.sort()

        for port in bilalport:

            print ('\n\nport : %s\tstate : %s\n' % (port,

abdullahScan[host][proto][port]['state']))
```



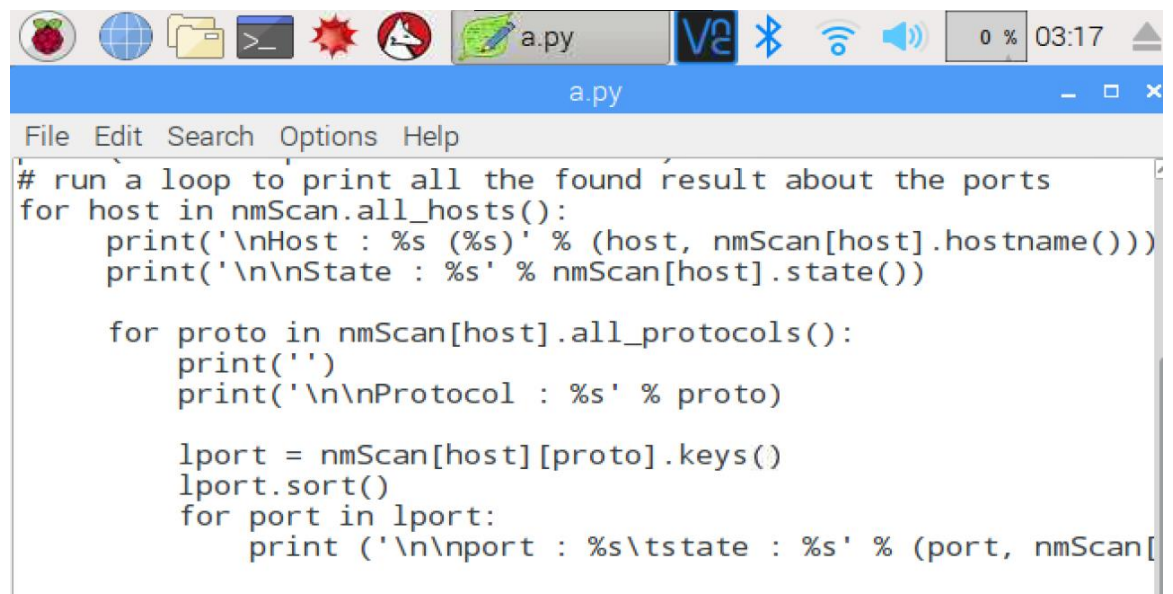
```
import nmap
# initialize the port scanner
nmScan = nmap.PortScanner()

# scan localhost for 1-1000
nmScan.scan('192.168.43.1/24', '1-1000', '0')
print('Details of Network\n\n\n: %s'%nmScan.scan('192.168.43.1/2

print('\n\n\n Specific Details \n\n')
# run a loop to print all the found result about the ports
for host in nmScan.all_hosts():
    print('\nHost : %s (%s)' % (host, nmScan[host].hostname()))
    print('\n\nState : %s' % nmScan[host].state())

    for proto in nmScan[host].all_protocols():
```

Figure 13: Automation Script



```
# run a loop to print all the found result about the ports
for host in nmScan.all_hosts():
    print('\nHost : %s (%s)' % (host, nmScan[host].hostname()))
    print('\n\nState : %s' % nmScan[host].state())

    for proto in nmScan[host].all_protocols():
        print('')
        print('\n\nProtocol : %s' % proto)

        lport = nmScan[host][proto].keys()
        lport.sort()
        for port in lport:
            print ('\n\nport : %s\tstate : %s' % (port, nmScan[
```

Figure 14: Automation Script