

# **SpyD: An Automated Testing Tool**



## **BACHELOR OF ENGINEERING IN INFORMATION TECHNOLOGY**

**By**

**Abhishek Pandey**

**Harshvardhan Dubey**

**Pratik Varma**

**Sumit Awinash**

**Under The Guidance Of**

**Prof. Reena Kothari**

Assistant Professor, Department of Information Technology



**Shree Rahul Education Society's (Regd.)**  
**SHREE L.R. TIWARI**  
**College of Engineering**

(Approved by AICTE, Government of Maharashtra and Affiliated to University of Mumbai)  
ISO 9001:2008 Certified.

Near Commissioner's Bungalow, Kanakia Park, Mira Road (E), Thane-401107, Maharashtra.

**Department of Information Technology  
2018-19**

**A Project Report On**  
**SpyD: An Automated Testing Tool**  
*Submitted to Mumbai University*



*In partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**  
**IN INFORMATION**  
**TECHNOLOGY**

**By**  
**Abhishek Pandey**

**Harshvardhan Dubey**

**Pratik Varma**

**Sumit Awinash**

**Under The Guidance Of**  
**Prof. Reena Kothari**

Assistant Professor, Department of Information Technology



Shree Rahul Education Society's (Regd.)  
**SHREE L.R. TIWARI**  
**College of Engineering**  
(Approved by AICTE, Government of Maharashtra and Affiliated to University of Mumbai)  
ISO 9001:2008 Certified.

Near Commissioner's Bungalow, Kanakia Park, Mira Road (E), Thane-401107, Maharashtra.

**Department of Information Technology**  
**2018-19**



Shree Rahul Education Society's (Regd.)  
**SHREE L.R. TIWARI**  
**College of Engineering**

(Approved by AICTE, Government of Maharashtra and Affiliated to University of Mumbai)  
ISO 9001:2008 Certified.

Near Commissioner's Bungalow, Kanakia Park, Mira Road (E), Thane-401107, Maharashtra.

**UNIVERSITY OF MUMBAI**

**CERTIFICATE**

This is to certify that the project titled “**SpyD: An Automated Testing Tool**” has been completed under our supervision and guidance by the following students:

**Abhishek Pandey**

**Harshvardhan Dubey**

**Pratik Varma**

**Sumit Awinash**

In the partial fulfillment of degree of Bachelor of Engineering in Information Technology branch as prescribed by the University of Mumbai during the academic year 2018-2019. The said work has been assessed and is found to be satisfactory.

**Signature of the Internal Examiner**

Name: Prof. Reena Kothari

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

**Signature of the External Examiner**

Name: \_\_\_\_\_

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

**Signature of the H.O.D.**

Name: Prof. Deepali Patil

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

**Signature of the Coordinator**

Name: Prof. Aarti Puthran

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

**Signature of the Principal**

Name: Dr. S. Ram Reddy

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

# **DECLARATION**

We do hereby declare that the work embodied in the project entitled “**SpyD: An Automated Testing Tool**” is the outcome of our original work under the guidance and supervision of **Prof. Reena Kothari**. This piece of work or any part of it has not been submitted previously for the award of any other degree, diploma, or other title to any other institution.

We also declare that this written submission represents our ideas in our own words and where others ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

**Abhishek Pandey**

Roll No.: 33

Exam Seat No.:

Date: The \_\_\_\_\_ April, 2019

**Harshvardhan Dubey**

Roll No.: 07

Exam Seat No.:

**Pratik Varma**

Roll No.: 67

Exam Seat No.:

**Sumit Awinash**

Roll No.: 02

Exam Seat No.:

## ACKNOWLEDGEMENT

A few sublime human experiences defy expressions of any kind, and a feeling of true gratitude is one of them. I, therefore, find words quite inadequate to express my indebtedness to my Guide **Prof. Reena Kothari** her virtuous guidance, encouragement and help throughout this work. Their deep insight into the problem and the ability to provide solutions has been immense value in improving the quality of project at all stages. This experience of working with them shall ever remain a source of inspiration and encouragement for me.

I express my thanks to **Prof. Deepali Patil**, HOD (IT), SLRTCE, Mira Road , for extending her support that she gave truly help the progression of the project work. I express my thanks to **Prof. Aarti Puthran**, Asst. Prof., SLRTCE, Mira Road for extending his support.

My sincere thanks to **Dr. S. B. Singh**, Director and **Dr. S. Ram Reddy**, Principal, SLRTCE, Mira Road for providing me the necessary administrative assistance in the completion of the work.

I am extremely grateful to the celebrated authors whose precious works have been consulted and referred in my project work. I also wish to convey my appreciation to my friends who provided encouragement and timely support in the hour of need.

Special thanks to my Parents whose love and affectionate blessings have been a constant source of inspiration in making this a reality.

All the thanks are, however, only fraction of what is due to Almighty for granting me an opportunity and the divine grace to successfully accomplish this assignment.

**Abhishek Pandey**

**Harshvardhan Dubey**

**Pratik Varma**

**Sumit Awinash**

## TABLE OF CONTENTS

Abstract.....		
List of Figures.....		

Chapter No.	Topic	Page No.
1.	INTRODUCTION 1.1 Description 1.2 Problem Formulation 1.3 Motivation 1.4 Proposed Solution 1.5 Scope Of the Project	1 2 3 3 4 4
2.	REVIEW OF LITERATURE 2.1 Literature Survey 2.1.1 Cybergrenade 2.1.2 Computer Network Security and Technology Research 2.2 Problem Statement	5 6 6 6 7
3.	SYSTEM ANALYSIS 3.1 Requirement 3.1.1 Functionality 3.1.2 Usability 3.1.3 Reliability & Availability 3.1.4 Performance 3.1.5 Security 3.1.6 Interfaces 3.2 Use Case of the proposed system	8 9 9 10 10 10 11 13
4.	ANALYSIS MODELING 4.1 Class Diagram 4.2 Sequence Diagram 4.3 Deployment Diagram 4.4 Data Flow Diagram	14 15 16 17 18
5.	DESIGN 5.1 Flowchart	20 21
6.	IMPLEMENTATION DETAILS 6.1 Implementation 6.2 Implementation Screenshot	23 24 25
7.	CONCLUSION 7.1 Conclusion 7.2 Future Scope	44 45 45
8.	REFERENCES	46
9.	PAPER PUBLISHED	48

## **ABSTRACT**

As people are realizing the importance of cyber security, the price for hacking tool such as Pineapple is increasing rapidly. One Pineapple device can cost up to Rs. 21000+Tax+Shipping charges. Due to high cost people do not prefer buying such a product which hamper their security as an organization. By building SpyD we can provide a similarly effective product at a very considerably lower price which will help the organization to manage the security. SpyD is a automated network penetration device/bot which will be able to perform all major operations through the means of an automated script with built in failsafe which can be performed by any professional ethical hacker. This will ensure that the routine jobs are performed more precisely as compared to a hired professional, thus, eliminating the threat of any man-made errors. Additionally, SpyD will automate the work of softwares like Nmap, Aircrack-NG, Karma, Tcpdump etc.

## LIST OF FIGURES

Figure No.	Name of Figure	Page No.
3.2	Use Case Diagram for SpyD	13
4.1	Class Diagram for SpyD	16
4.2	Sequence Diagram for SpyD	17
4.3	Deployment Diagram for SpyD	18
4.4	Data Flow Diagram for SpyD	19
5.1	Flowchart for SpyD	22
6.2.1	SpyD Stock User Interface	25
6.2.2	Custom Firmware for SpyD	25
6.2.3	Custom Firmware Being Flashed	25
6.2.4	SpyD Network Interfaces	26
6.2.5	Updating SpyD Repositories	27
6.2.6	Browsing /dev/ directory of SpyD	27
6.2.7	Installed Packages on SpyD	28
6.2.8	LuCI Wireless Network Interface of SpyD	29
6.2.9	Testing SpyD Network Connection using ping	29
6.2.10	Installing kmod-usb-storage package on SpyD	30
6.2.11	Installing kmod-fs-vfat on SpyD	30
6.2.12	Usb Device Detected on SpyD	30
6.2.13	sda & sda1 in /dev/ on SpyD	31
6.2.14	SpyD: Homepage	31
6.2.15	SpyD: User Mode	32
6.2.16	SpyD: Tester Mode	33
6.2.17	SpyD: Admin Mode	33
6.2.18	Directory	34
6.2.19	Service Scripts	34
6.2.20	MAC Changer	36
6.2.21	IPV6 Changer	37
6.2.22	OS Detection	37
6.2.23	Monitor WLAN	38
6.2.24	Open Port Detection	39
6.2.25	Fake Access point creation	40

6.2.26	Android Environment	51
6.2.27	Android Interface	52

# **Chapter 1**

# **Introduction**

# Chapter 1

## Introduction

This chapter will introduce the reader with SpyD, which is used as a penetration testing tool and a hacker device. It will light up the topics like the description of the project and the former formulation of the problem behind it as well as what motivated the makers of the project to take a decision to make this project and its related problem solutions and thus covering up the scope of the project.

### About

As people are realizing the importance of cyber security, the price for a testing and hacking tool such as Pineapple is increasing rapidly. One Pineapple device can cost a lot in Indian markets. Due to high cost people do not prefer buying such a product which hamper their security as an organization. By building SpyD we can provide a similarly effective and completely automated product at a very considerably lower price which will help the organization to manage the security and reduce human efforts.

By building SpyD we aim to provide a product similar to current pen test tools like PineAP at a very considerably lower price along with packet tracing which will help the organization to manage their security.

### 1.1 Description

SpyD is a follow up of the established Wi-Fi Pineapple and adds various functionalities like complete automation, packet monitoring, etc. The secondary objective is to lower the cost of the device to target smaller organizations and start-ups.

In this project we aim to overcome the shortcomings of conventional testing tools like rutabaga or PineAP that are need for a professional, high cost, manual work etc.

There are number of product available in the market with different vendors and it creates a bit confusion in the mind of consumer that which component to buy. Therefore it becomes necessary to design effective system to summarize the pros and cons of product characteristics so that consumer can quickly find their favorable product so as to develop an efficient and cost effective security testing tool. We have developed a system which performs

tests and automates the working, thus, removing a middle-man which was required to be a testing professional.

The proposed system will help the users or organizations to perform their tests locally and improve the network's efficiency. From a hacking point of view, they can perform hacks with the help of SpyD in a much better and feasible way.

Finally, a fully automated router/modem which is the SpyD will be available at the remote locations to access the internet and perform the desired tests to improve the network and eventually the system security.

## 1.2 Problem Formulation

Nowadays, the current penetration testing and hacking tools such as conventional Wi-Fi PineAP are being modified to be made more advanced due to current security needs and policies. This modification has led to a significant increase in the cost of such tools and more man power is required to carry out such tasks by professionals.

With our automated network penetration device we'll be able to perform major operations through the means of automated scripts with built-in failsafe which can be performed by any professional ethical hacker. This ensures that routine jobs are performed more precisely as compared to any hired professional, thus, eliminating the threat of man-made errors.

## 1.3 Motivation

The overwhelming demand of security in local and small organizations due to daily security attacks has motivated us to develop this project. In addition to bring the usability of testing tools has made us to think to develop this project so as to simplify the process of testing to make the efficient penetration testing tool for the end user.

So we wished to design a system that could help the user to perform automated tests and monitor packets with only sitting on a PC. Also we wish to give the user the features that will enable them to be at ease when attempting to gain access to networks with performing tests.

## 1.4 Proposed Solution

We aim to provide a system that is an automated network penetration device which will perform all major operations through the means of an automated script with built-in failsafe which can be performed by any professional hacker, thus, providing automation at a lower cost and eliminating any man made errors at the same time.

SpyD aims at providing an automated solution towards the previously used network scanners and packet sniffers like Nmap, Aircrack-NG, and Tcpdump.

## 1.5 Scope of the project

The project aims to ease the process of Network security testing, so it will be easy for the users to perform tests to check for vulnerabilities as per the requirement. In many organizations, the testing tools which are used are the conventional ones which are usually operated and worked upon by testing professionals. This isn't a feasible option for end users and the organizations that are relatively smaller than those organizations. Through this proposed system, the end users as well as small scale organizations can perform tests and attacks to check for any vulnerability in their system and remove those flaws. The project uses various firmwares from the significant domain Network security like OpenWRT, LuCI, and operating systems like Python, etc to perform attacks like Man-in-the-middle, Framereplay, and fake Frame generation. All these functions are clustered in one chipset connected to the network to perform tests or attacks based on user's requirements.

## **Chapter 2**

# **Review of Literature**

## Chapter 2

# Review of Literature

Here we will elaborate the aspects like the literature survey of the project and what all projects are existing and been actually used in the market which the makers of this project took the inspiration from and thus decided to go ahead with the project covering with the problem statement.

### 2.1 Literature Survey

#### 2.1.1 Cybergrenade: Automated Exploitation of Local Network machines via Single Board Computers by Anurag Akkiraju, David Gabay, Halim Burak Yesilyurt, Hidayet Aksu, Selcuk Uluagac

In the reference paper, a defensive cyber security framework device called Cyber grenade automating various penetration testing tools to sequentially exploit machines connected to a Single local network, all underneath a single application running on a Single-Board Computer (SBC). The paper further describes the advantage of the SBC's unique capabilities in a way that manual exploitation simply cannot match. Currently, while many SBCs are being used in research as exploitation tool-kits, the current state of automation of the processes associated with exploitation leaves much to be desired. While the reference paper describes the device Framework, it can be used as a guideline for future research automating the exploitation process. The Device allows tools such as Nmap, OpenVAS, and Metasploit tools to be automatically utilized under one framework. The Paper is divided into five sub parts

- I. Introduction
- II. Related Work
- III. System Design and Implementation
- IV. Experiments
- V. Conclusion

#### 2.1.2 Computer Network Security and Technology Research by Fan Yan, Yang Jianwen and Cheng Lin

The rapid development of computer network system brings both a great convenience and new security threats for users. Network security problem generally includes network system security and data security. Specifically, it refers to the reliability of the network

system, confidentiality, integrity, and availability of data information in the system. Network security problem exists through all the layers of the computer network, and the network security objective is to maintain the confidentiality, authenticity, integrity, dependability, availability, and audit-ability of the network. This paper introduces the network security technologies mainly in detail, including authentication, data encryption technology, firewall technology, intrusion detection system (IDS), antivirus technology and virtual private network (VPN). Network security problem is related to every network user, so we should put a high value on network security, try to prevent hostile attacks and ensure the network security.

The penetration testing devices available in foreign countries and Multinational companies are PineAP or rutabaga. These aren't feasible because of the necessity of a professional user/tester and the high cost and maintenance. This leads to the need of an alternative in the local market feasible for all users.

## **2.2 Problem Statement:**

Nowadays, the current penetration testing and hacking tools such as conventional Wi-Fi PineAP are being modified to be made more advanced due to current security needs and policies. This modification has led to a significant increase in the cost of such tools and more man power is required to carry out such tasks by professionals.

With our automated network penetration device we'll be able to perform major operations through the means of an automated script with built-in failsafe which can be performed by any professional ethical hacker. This ensures that routine jobs are performed more precisely as compared to any hired professional, thus, eliminating the threat of man-made errors.

# **Chapter 3**

# **System Analysis**

## Chapter 3

# System Analysis

System Analysis will cover the topics like the Functional, Non-Functional and the specific requirements of the project and touching all the software and the hardware requirements as well.

### 3.1 Requirements

#### 3.1.1 Functionality

##### 3.1.1.1 Provide comprehensive network details.

SpyD will display detailed information of the available access network.

##### 3.1.1.2 Provide Network Access

SpyD will provide detail network access to the user.

##### 3.1.1.3 Provide packet monitoring

SpyD will monitor the incoming/outgoing packets through the network.

##### 3.1.1.4 Provide creating of pseudo access point

SpyD will create a pseudo access point that will exploit the vulnerability of the victim.

##### 3.1.1.5 Provide packet injection.

SpyD allows pseudo frame generation that is injected into the network.\

##### 3.1.1.6 Provide penetration testing.

SpyD allows thorough self-evaluation of its current network to find vulnerabilities.

##### 3.1.1.7 Provide various attacks.

SpyD allows attacks like Man in the Middle (MITM), Arp Spoofing, etc.

### **3.1.1.8 Generate Reports Related to Packets.**

SpyD will generate detailed reports for specific packets.

### **3.1.2 Usability**

#### **3.1.2.1 Graphical User Interface**

SpyD provides the Luci web interface for further development

Luci Web Interface is a Graphical User Interface provided by SpyD

### **3.1.3 Reliability & Availability**

SpyD is reliable since it monitors packets flowing in the network actively and perform operations as specified by the user by using standardized algorithm used by professional ethical hackers to test the network.

### **3.1.4 Performance**

SpyD is a device that has to be physically connected or be within proximity of the network.

The performance will depend on the hardware / network of the device.

### **3.1.5 Security**

#### **3.1.5.1 Data Transfer**

The device performs data transfer between files present at the multiple nodes within the connected network in an encrypted manner. So, no eavesdropper can intercept the data transfer between the nodes.

#### **3.1.5.2 Data Storage**

SpyD uses extRoot to store data on the device. The device storage capability depends on the model and version of SpyD, where the minimum capacity will be 16 Giga Bytes.

### 3.1.6 Interfaces

There are many types of interfaces as such supported by our SpyD namely; User Interface, Software Interface and Hardware Interface.

The protocol used shall be HTTP, FTP, SMTP, and TCP/IP.

The Port number used will be varied as per the requirement.

#### 3.1.6.1 User Interfaces

The device is accessible via a command line interface that is either via “ssh” or “telnet”. In the latest version access via telnet is not possible due to encryption hence ssh is preferred. On linux ssh is used and on windows puTTy is used.

#### 3.1.6.2 Hardware Interfaces

CPU	Ram	Flash	Network	USB	Serial	JTag
Atheros AR9331@400MHz	32MiB	4MiB	1 x 100MBit	1 x 2.0	Yes	No

- SoC: Atheros AR9330 rev 1
- 802.11 b/g/n 150 Mbps
- Powered via USB B-Mini (5 Volts)
- Tiny form factor
- 5.7 cm x 5.7 cm PCB
- 6.7 cm x 7.4 cm x 2.2 cm case
- RJ45 I/O port

#### 3.1.6.3 Software Interfaces

LuCI is installed as a 'meta package' which installs several other packages by having these defined as a dependency. Notably, it installs the uHTTPd web server, configured for use with LuCI. The dependent packages are the following (see the LuCI technical reference for more information):

- uhttpd
- uhttpd-mod-ubus
- luci-mod-admin-full
- luci-theme-bootstrap
- luci-app-firewall

- luci-proto-core
- luci-proto-ppp
- libiwinfo-lua

In case you want to use uHTTPd for the web interface there is little configuration necessary as uHTTPd is configured with CGI to make LuCI work with the Lua interpreter. By default this is organized as follows. By default /www is the standard document root. Thus, by requesting this docroot (by pointing your browser to the devices IP address) an index file such as index.html is searched for (per uHTTPdsettings). The file /www/index.html (installed with LuCI) is prepared such that when requested, it redirects you to /cgi-bin/luci, which is the default CGI gateway for LuCI. This is just a script, which basically calls Lua at /usr/bin/lua. uhttpd is configured by default to load pages as CGI in the /cgi-bin path, and thus starts serving these pages with the /cgi-bin/luci script.

It is also possible to run LuCI with Lua as an embedded process. uhttpd supports this; see the corresponding section of the uHTTPd Web Server Configuration article on the UCI configuration of uhttpd.

#### **3.1.6.4 Communications Interfaces**

LuCI web interface is used for the same.

### 3.2 Use case diagram of the proposed system

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. There are three actors which is customer, system, and the authentication system. The customer can sign up and login which will be authenticated by authentication system and can add items and will get final base rating for assembled System.

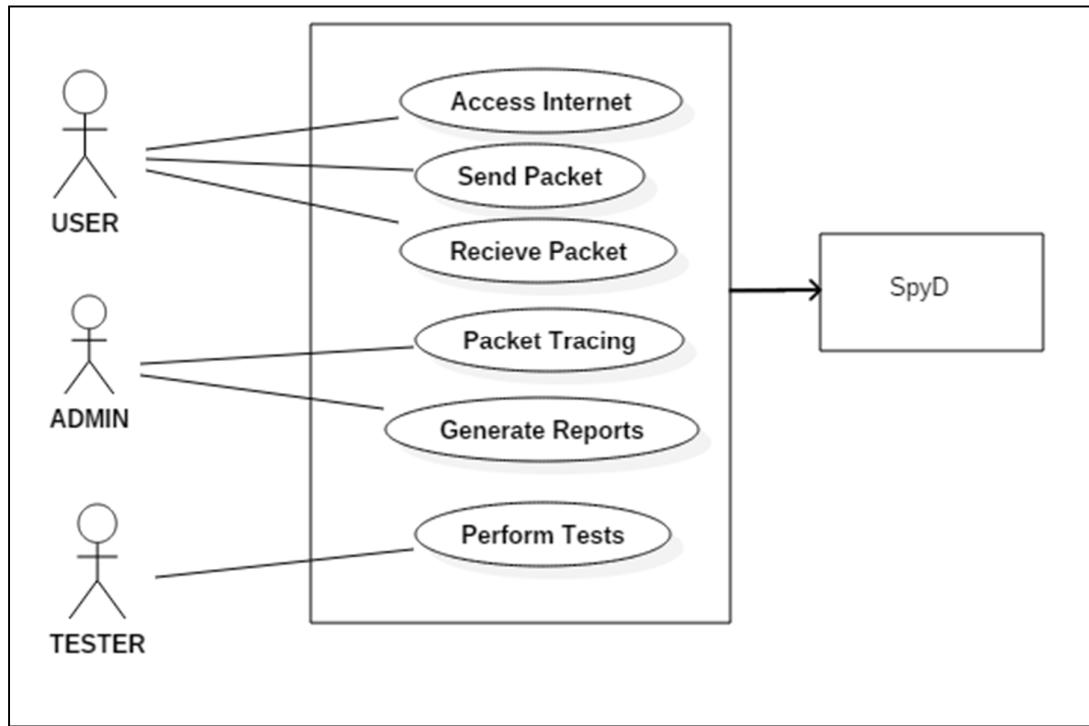


Figure 3.2: Use Case diagram for SpyD

# **Chapter 4**

# **Analysis Modeling**

## Chapter 4

# Analysis Modeling

All the aspects of the proposed system will be covered in this chapter in the diagrammatic manner and provides the detailed manner of the system.

### 4.1 Class Diagram

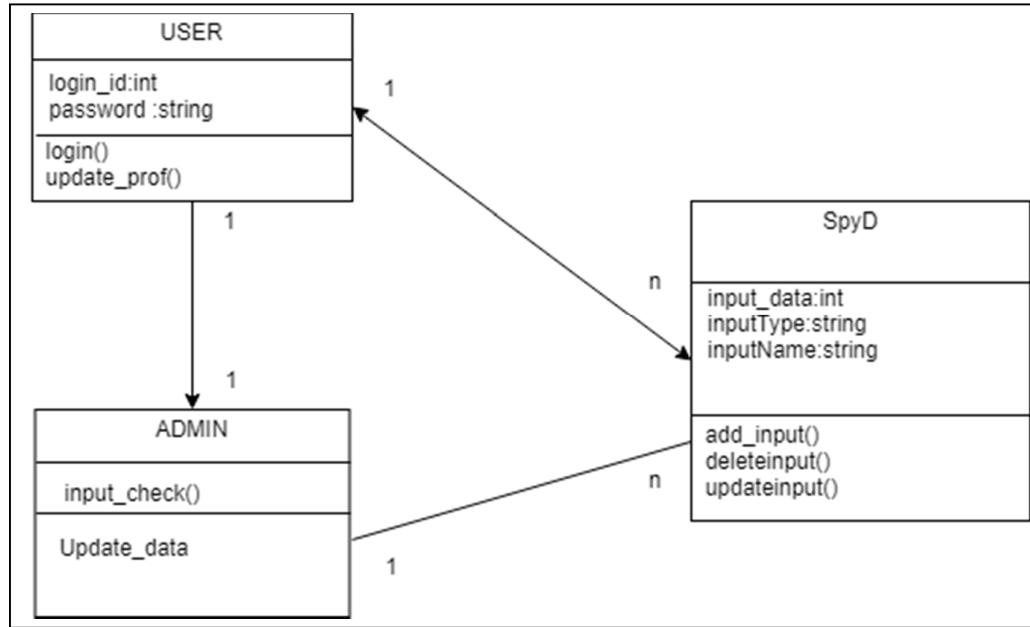
The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a *structural diagram*.

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages.

So the purpose of the class diagram can be summarized as:

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

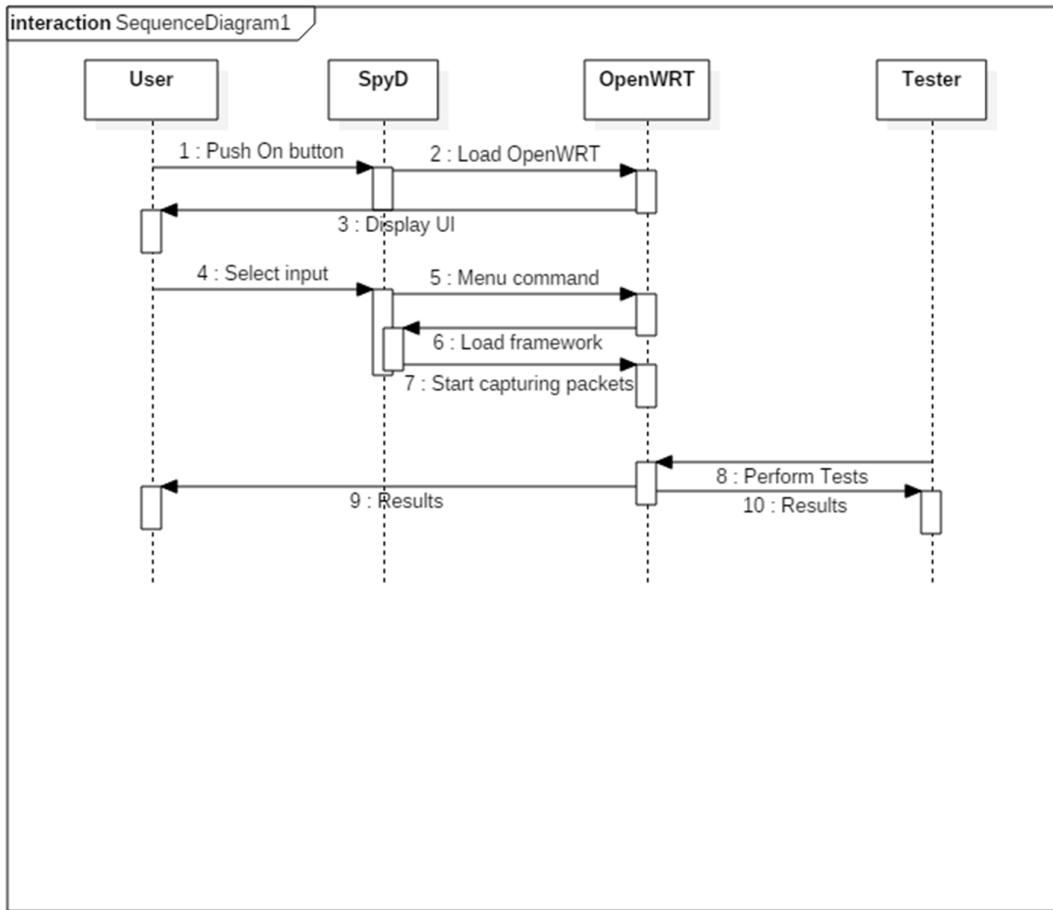
Here we have three classes viz. user , Admin and SpyD device .The user will have login details and would be able to search for the operation he wants to perform .Each operation have its unique ID, name, and functionality. The task of administrator is to keep on monitoring the system and updating the system catalog. Finally the SpyD will return the desired output.



## 4.2 Sequence Diagram

The Sequence Diagram models the collaboration of objects based on a time sequence. It shows how the objects interact with others in a particular scenario of a usecase. With the advanced visual modeling capability, we can create complex sequence diagram in few clicks. Besides, VP-UML can generate sequence diagram from the flow of events which you have defined in the use case description.

Sequence diagram below demonstrates the interaction between various entities of SpyD system like user, SpyD, OpenWRT, and Tester in a sequential manner.



to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

The name *Deployment* itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related.

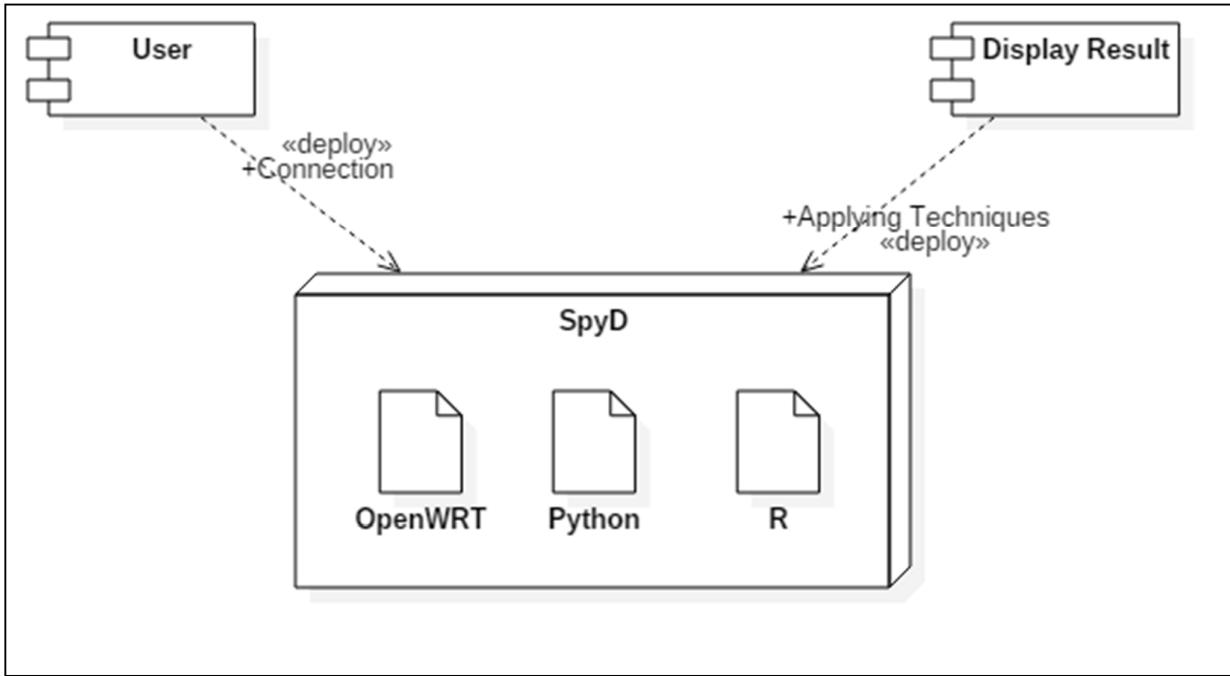


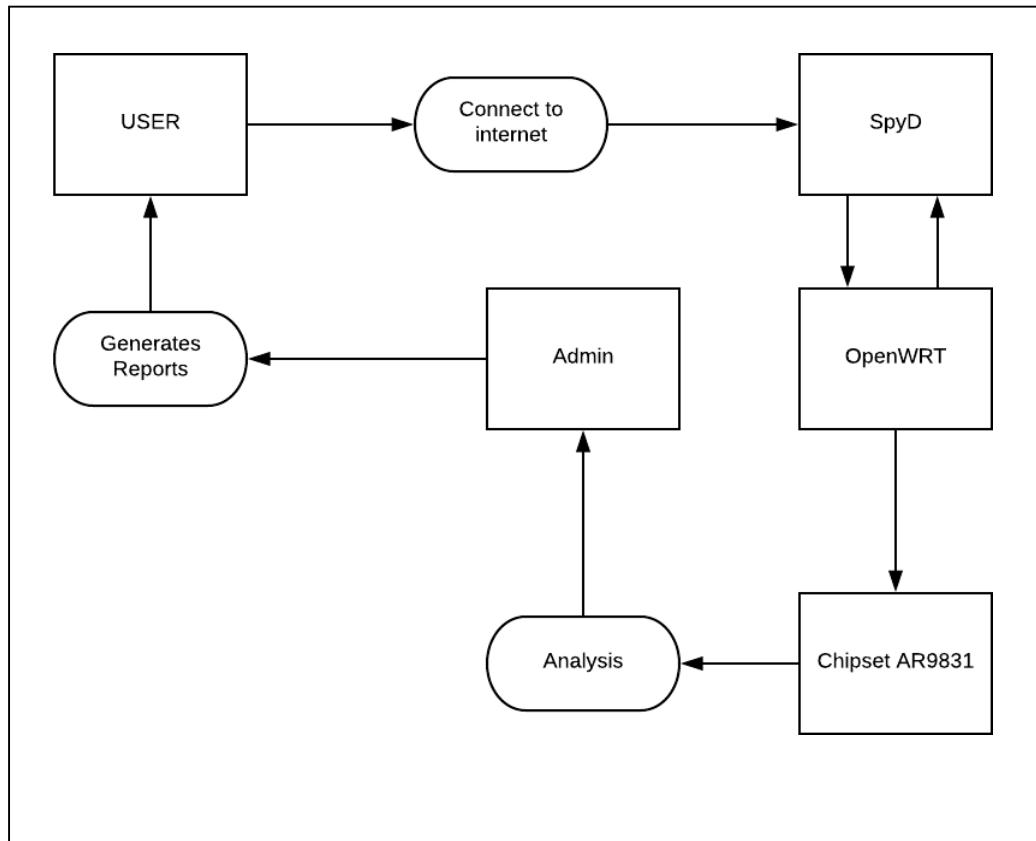
Figure 4.3: Deployment diagram for SpyD

#### 4.4 Data flow diagram of the system

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated.

A two-dimensional diagram that explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output.

Individuals seeking to draft a data flow diagram must (1) identify external inputs and outputs, (2) determine how the inputs and outputs relate to each other, and (3) explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspect.



# **Chapter 5**

# **Design**

# Chapter 5

## Design

Design will elaborate the step by step flow of SpyD based on user Benchmark thus giving up the detailed information as to the basic flow of the system.

### 5.1 Flowchart

Flowcharts are used in designing and documenting complex processes or programs. Like other types of diagrams, they help visualize what is going on and thereby help the people to understand a process, and perhaps also find flaws, bottlenecks, and other less-obvious features within it. There are many different types of flowcharts, and each type has its own repertoire of boxes and notational conventions. The two most common types of boxes in a flowchart are:

- A processing step, usually called *activity*, and denoted as a rectangular box
- A decision usually denoted as a diamond.

A flowchart is described as "cross-functional" when the page is divided into different swim lanes describing the control of different organizational units. A symbol appearing in a particular "lane" is within the control of that organizational unit. This technique allows the author to locate the responsibility for performing an action or making a decision correctly, showing the responsibility of each organizational unit for different parts of a single process.

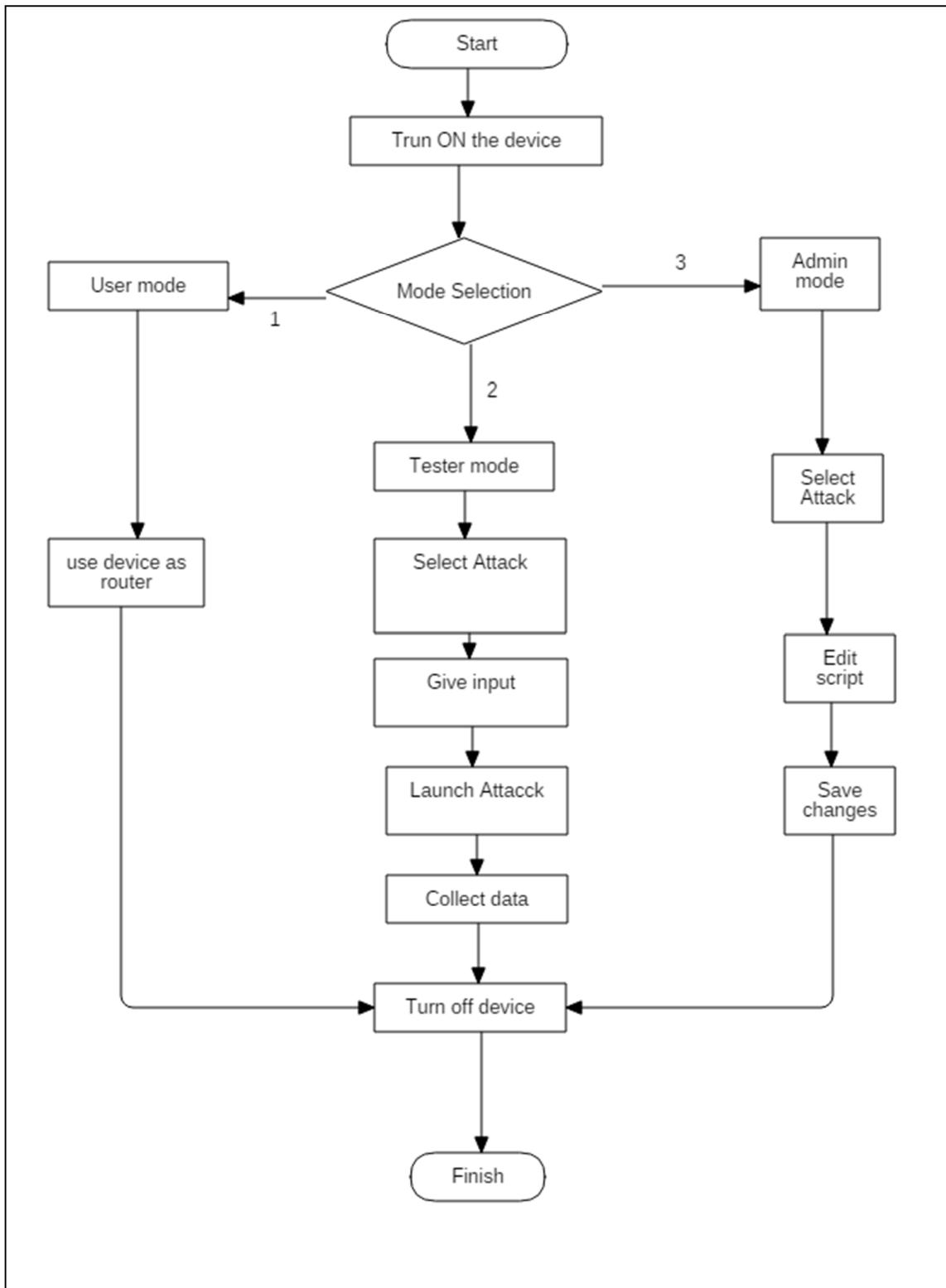


Figure 5.1: Flowchart for SpyD

# **Chapter 6**

# **Implementation Details**

## Chapter 6

# Implementation Details

This chapter will give all the Implementation details and methods used to implement the Application in brief.

### 6.1 Implementation

We have designed following algorithm which would be followed throughout the implementation phase of this system

#### PenSec Algorithm

---

1. Start
2. Turn on the device
3. Select Mode of operation
  - a: User mode
    - Use device as a router
  - b: Tester mode
    - i: Select Attack
      - Man in the middle Attack
      - Frame replay attack
      - Fake frame generation attack
      - Create fake access point
      - More attacks will be added
    - ii: Give appropriate input with respect to attack
    - iii: launch attack
    - iv: Collect appropriate data
    - v: Finish or go back to step i.
  - c: Admin mode
    - i: Select Attack
      - Man in the middle Attack
      - Frame replay attack
      - Fake frame generation attack
      - Create fake access point
      - More attacks will be added
    - ii: Make changes in script as per requirement
    - iii: Finish or go back to step i.
4. Exit selected mode
5. Turn off the device or Change user mode
6. Stop

## 6.2 Implementation Screenshots

### Flashing the Custom Firmware on SpyD

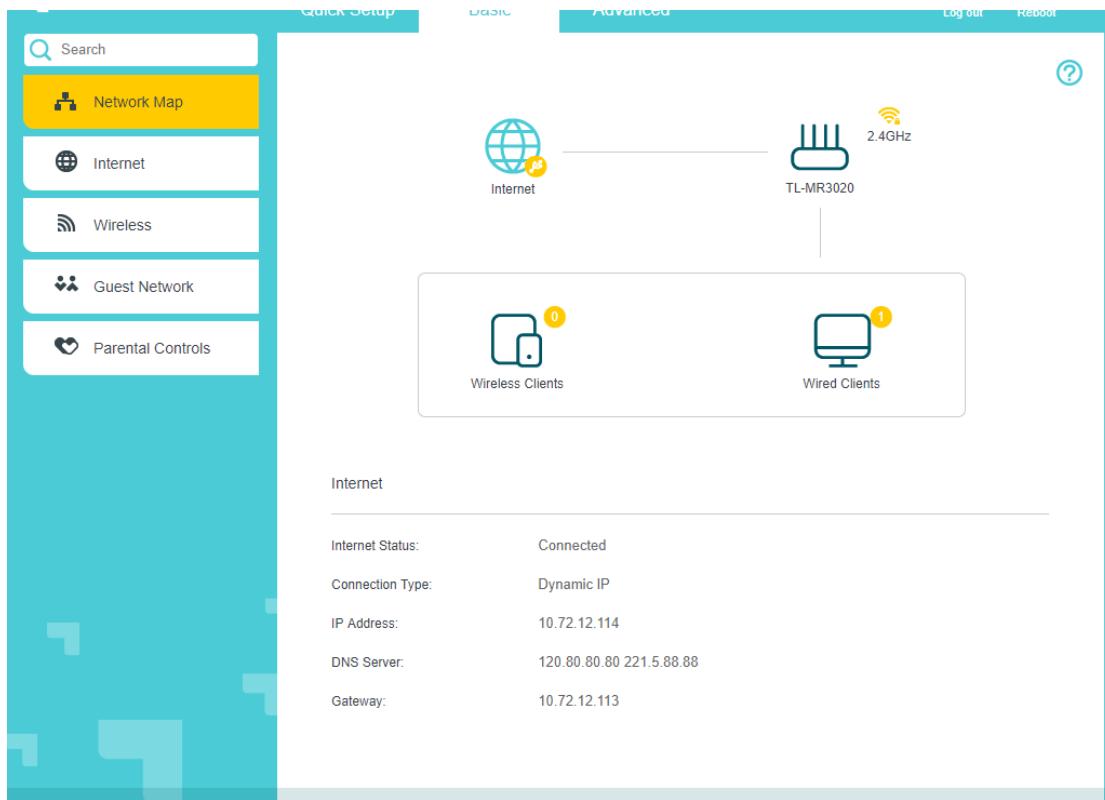


Figure 6.2.1: SpyD Stock User Interface



Figure 6.2.2: Custom Firmware for SpyD

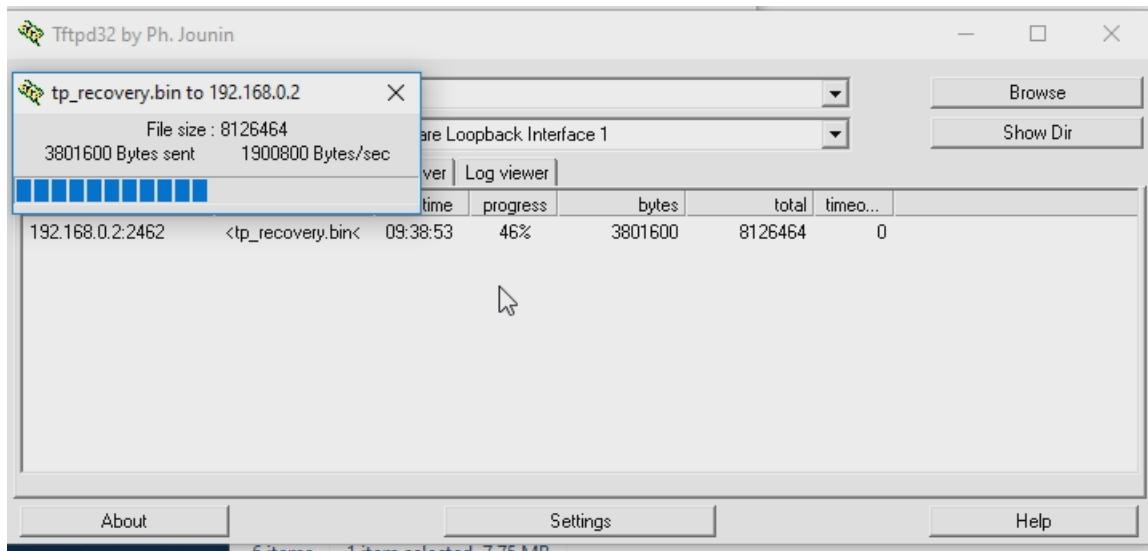


Figure 6.2.3: Custom Firmware Being Flashed On SpyD

The following screenshot shows all the network interfaces available by using the command ifconfig -a

```

root@Pulpstone:~# ifconfig -a
br-lan Link encap:Ethernet HWaddr AC:84:C6:42:40:82
          inet addr:192.168.1.1 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::ae84:c6ff:fe42:4082/64 Scope:Link
             inet6 addr: fdff:9d61:2626:1/60 Scope:Global
                UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                RX packets:67917 errors:0 dropped:0 overruns:0 frame:0
                TX packets:39763 errors:0 dropped:0 overruns:0 carrier:0
                collisions:0 txqueuelen:1000
                RX bytes:23130602 (22.0 MiB) TX bytes:14392756 (13.7 MiB)

eth0      Link encap:Ethernet HWaddr AC:84:C6:42:40:82
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:68779 errors:0 dropped:9 overruns:0 frame:0
          TX packets:40693 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:24138981 (23.0 MiB) TX bytes:14586609 (13.9 MiB)
          Interrupt:5

eth0.2    Link encap:Ethernet HWaddr AC:84:C6:42:40:83
          inet6 addr: fe80::ae84:c6ff:fe42:4083/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:484 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:163912 (160.0 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:4239 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4239 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:328462 (320.7 KiB) TX bytes:328462 (320.7 KiB)

wlan0     Link encap:Ethernet HWaddr AC:84:C6:42:40:82
          inet addr:192.168.43.41 Bcast:192.168.43.255 Mask:255.255.255.0
          inet6 addr: fe80::ae84:c6ff:fe42:4082/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:28637 errors:0 dropped:0 overruns:0 frame:0
          TX packets:45647 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11048390 (10.5 MiB) TX bytes:22573330 (21.5 MiB)

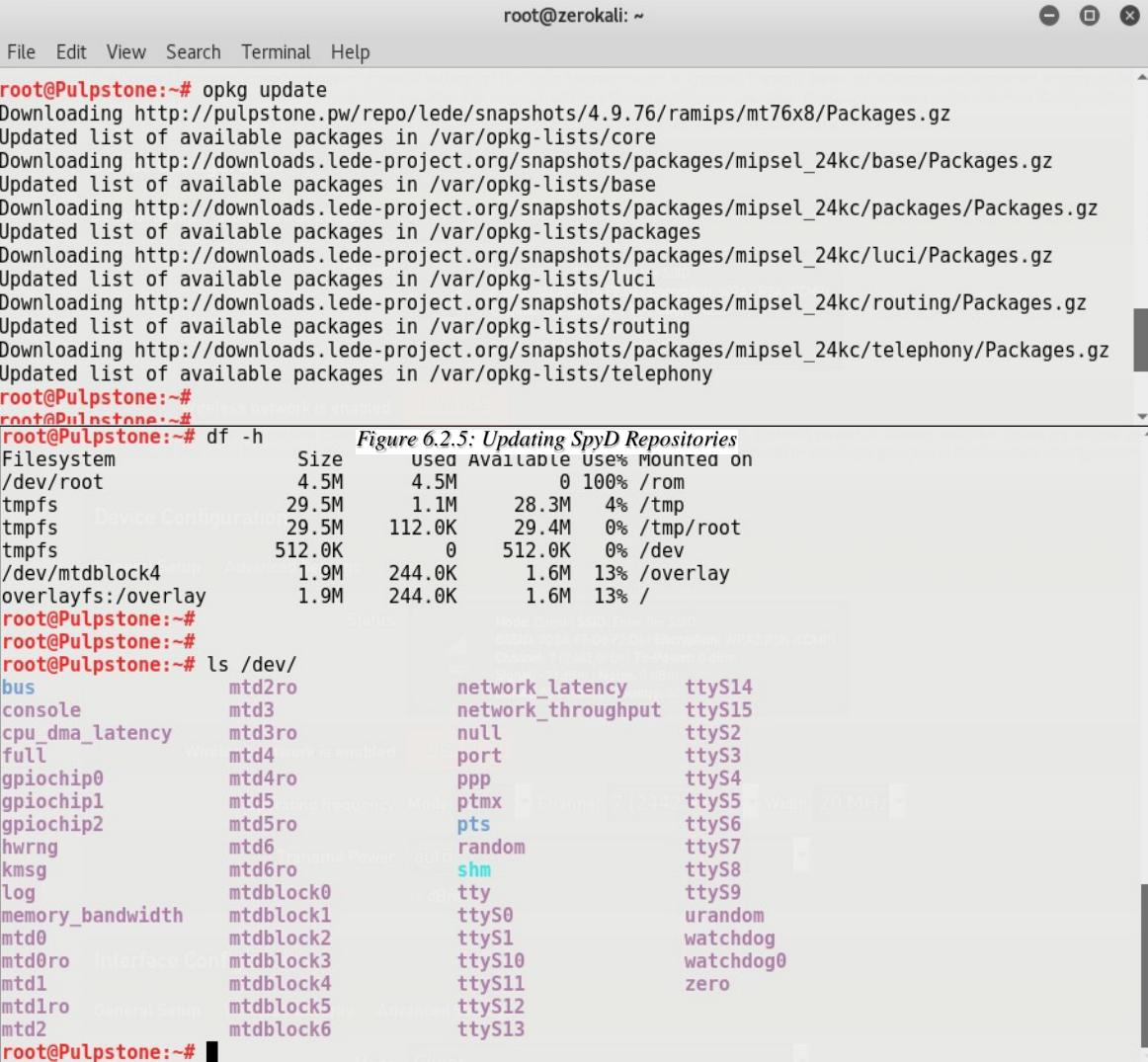
wlan0-1   Link encap:Ethernet HWaddr AE:84:C6:42:40:82
          inet6 addr: fe80::ac84:c6ff:fe42:4082/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:761 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:166184 (162.2 KiB)

root@Pulpstone:~# 

```

Figure 6.2.4: SpyD Network Interfaces

Updating SpyD repositories by using the command ‘opkg update’



The terminal window shows the root user performing two tasks:

- Updating SpyD Repositories:** The user runs the command `opkg update`. The output shows the download of several package lists from various URLs, including `http://pulpstone.pw/repo/lede/snapshots/4.9.76/ramips/mt76x8/Packages.gz` and `http://downloads.lede-project.org/snapshots/packages/mipsel_24kc/base/Packages.gz`.
- Browsing /dev/ directory:** The user runs `ls /dev/`. The output lists numerous device nodes, including `mtd2ro`, `network_latency`, `ttyS14`, `console`, `mtd3`, `network_throughput`, `ttyS15`, `cpu_dma_latency`, `mtd3ro`, `null`, `ttyS2`, `full`, `mtd4`, `port`, `ttyS3`, `gpiochip0`, `mtd4ro`, `ppp`, `ttyS4`, `gpiochip1`, `mtd5`, `ptmx`, `ttyS5`, `gpiochip2`, `mtd5ro`, `pts`, `ttyS6`, `hwrng`, `mtd6`, `random`, `ttyS7`, `kmsg`, `mtd6ro`, `shm`, `ttyS8`, `log`, `mtdblock0`, `tty`, `ttyS9`, `memory_bandwidth`, `mtdblock1`, `ttyS0`, `urandom`, `mtd0`, `mtdblock2`, `ttyS1`, `watchdog`, `mtd0ro`, `mtdblock3`, `ttyS10`, `watchdog0`, `mtd1`, `mtdblock4`, `ttyS11`, `zero`, `mtd1ro`, `mtdblock5`, `ttyS12`, and `mtd2`, `mtdblock6`, `ttyS13`.

*Figure 6.2.5: Updating SpyD Repositories*

We see the device directory. ‘df’ command displays the amount of disk space available on the file system containing each file name argument. Here the ‘-h’ option prints the size in powers of 1024 that is in the form of megabytes.

The screenshot shows the SpyD LuCI web interface. The top navigation bar has tabs for 'Actions' and 'Configuration'. Below this, a progress bar indicates 'Free space: 87% (1.64 MB)'. A modal dialog box is open, prompting 'Download and install package:' with an 'OK' button. A search bar labeled 'Filter:' has a 'FIND PACKAGE' button next to it.

**Status**

Under the 'Status' section, there are two tabs: 'Installed packages' (selected) and 'Available packages'. The 'Installed packages' table lists the following packages:

	Package name	Version
Remove	3ginfo	pulpstone-repacked-20..114
Remove	3ginfo-qmisignal	pulpstone-repacked-20..114
Remove	3ginfo-text	pulpstone-repacked-20..114
Remove	base-files	183-r5847-3efc4d1
Remove	blkid	2.30.2-2
Remove	block-mount	2018-01-13-18090d97-1
Remove	busybox	1.27.2-3
Remove	chat	2.4.7-12
Remove	comgt	0.32-30
Remove	comgt-hso	1.0
Remove	comgt-ncm	0.32-30
Remove	dnsmasq	2.79-2
Remove	dropbear	2017.75-5
Remove	eudev	3.2-1
Remove	firewall	2018-03-20-5cdf15ee-2
Remove	fstools	2018-01-13-18090d97-1
Remove	fwtool	1
Remove	hostapd-common	2017-08-24-c2d4f2eb-6
Remove	ip6tables	1.6.2-1

*Figure 6.2.7: Installed Packages on SpyD*

LuCI web interface is used to check the storage available as well a list down all the installed packages.

It also gives us the option to search for packages and install them. Other packages are available at ‘Available Packages’

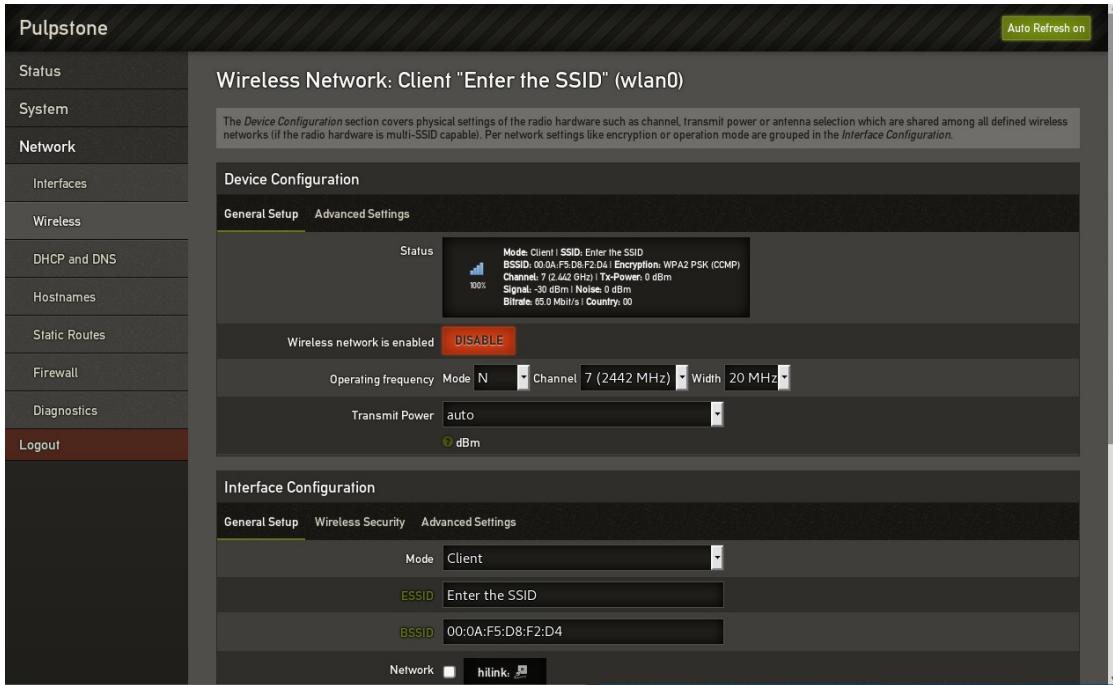
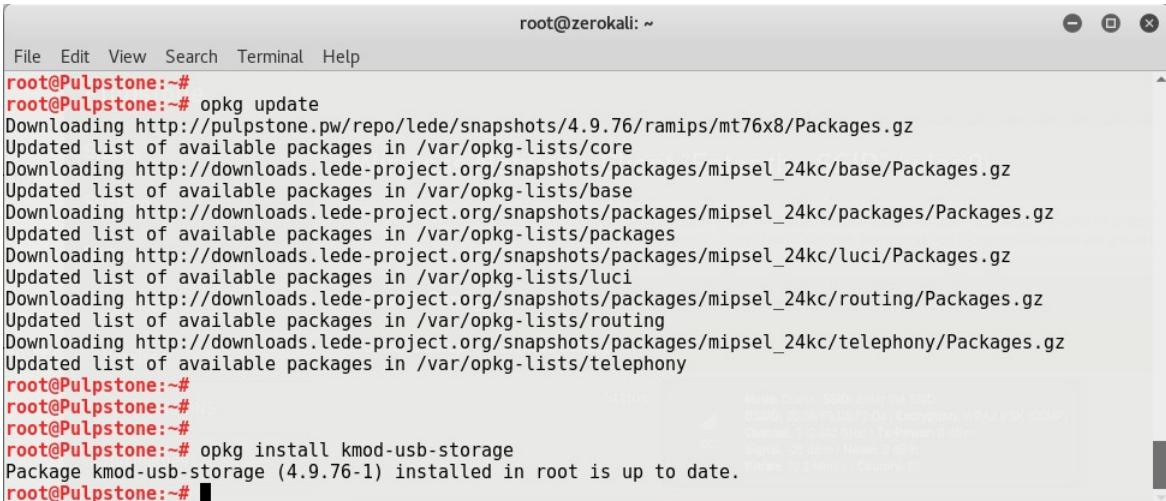


Figure 6.2.8: LuCI Wireless Network Interface of SpyD

```
root@zerokali: ~
File Edit View Search Terminal Help
root@Pulpstone:~#
root@Pulpstone:~# ping google.com
PING google.com (172.217.161.14): 56 data bytes
64 bytes from 172.217.161.14: seq=0 ttl=51 time=54.766 ms
64 bytes from 172.217.161.14: seq=1 ttl=51 time=50.336 ms
64 bytes from 172.217.161.14: seq=2 ttl=51 time=54.704 ms
64 bytes from 172.217.161.14: seq=3 ttl=51 time=54.058 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 50.336/53.466/54.766 ms
root@Pulpstone:~#
root@Pulpstone:~#
root@Pulpstone:~#
root@Pulpstone:~#
root@Pulpstone:~#
```

Figure 6.2.9: Testing SpyD Network Connection using ping

SpyD is successfully connected to “Enter the SSID”. It works in dual channel that is spyD is connected to a network as well as providing one. This is possible due to its multiple interfaces.



```

root@Pulpstone:~#
root@Pulpstone:~# opkg update
Downloading http://pulpstone.pw/repo/lede/snapshots/4.9.76/ramips/mt76x8/Packages.gz
Updated list of available packages in /var/opkg-lists/core
Downloading http://downloads.lede-project.org/snapshots/packages/mipsel_24kc/base/Packages.gz
Updated list of available packages in /var/opkg-lists/base
Downloading http://downloads.lede-project.org/snapshots/packages/mipsel_24kc/packages/Packages.gz
Updated list of available packages in /var/opkg-lists/packages
Downloading http://downloads.lede-project.org/snapshots/packages/mipsel_24kc/luci/Packages.gz
Updated list of available packages in /var/opkg-lists/luci
Downloading http://downloads.lede-project.org/snapshots/packages/mipsel_24kc/routing/Packages.gz
Updated list of available packages in /var/opkg-lists/routing
Downloading http://downloads.lede-project.org/snapshots/packages/mipsel_24kc/telephony/Packages.gz
Updated list of available packages in /var/opkg-lists/telephony
root@Pulpstone:~#
root@Pulpstone:~#
root@Pulpstone:~# opkg install kmod-usb-storage
Package kmod-usb-storage (4.9.76-1) installed in root is up to date.
root@Pulpstone:~#

```

Figure 6.2.10: Installing kmod-usb-storage package on SpyD

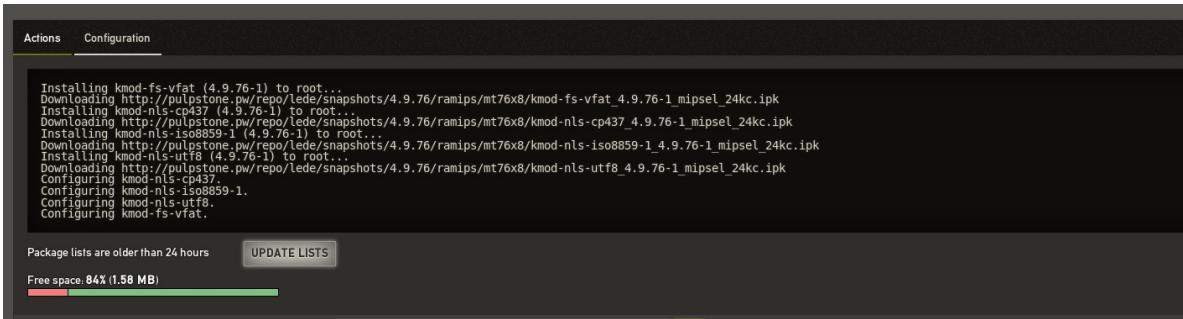
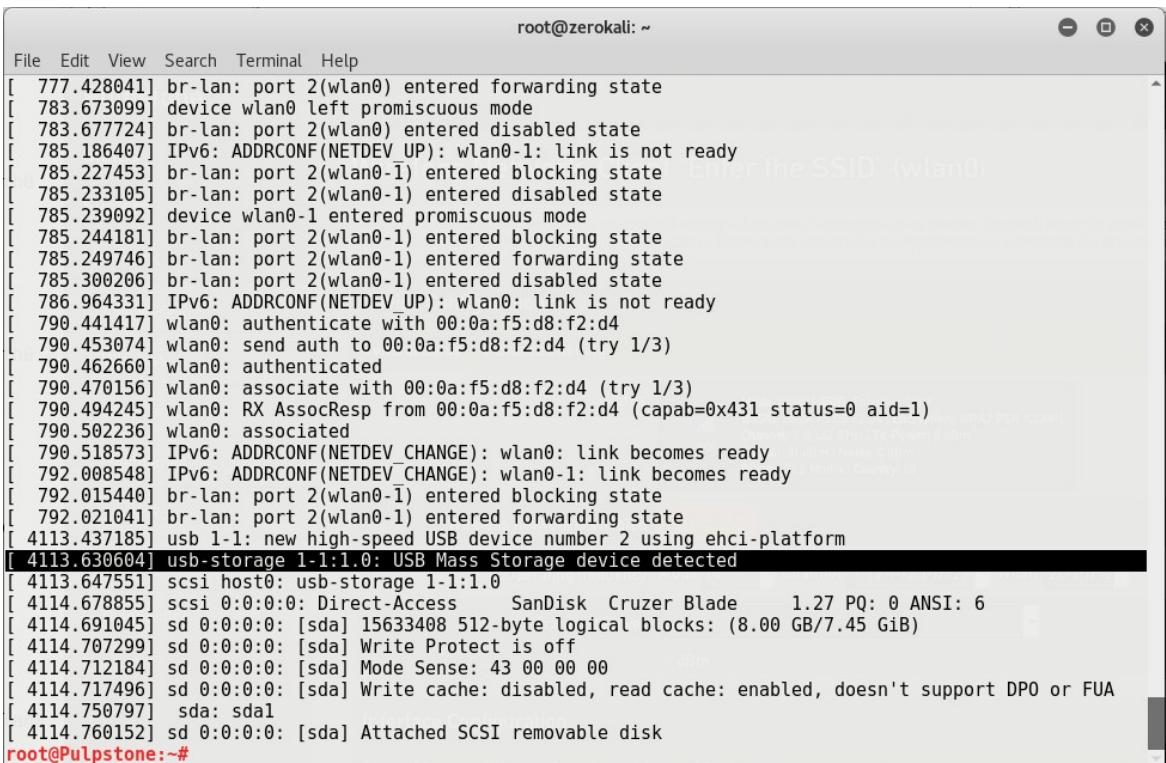


Figure 6.2.11: Installing kmod-fs-vfat on SpyD



```

root@zerokali:~#
File Edit View Terminal Help
[ 777.428041] br-lan: port 2(wlan0) entered forwarding state
[ 783.673099] device wlan0 left promiscuous mode
[ 783.677724] br-lan: port 2(wlan0) entered disabled state
[ 785.186407] IPv6: ADDRCONF(NETDEV UP): wlan0-1: link is not ready
[ 785.227453] br-lan: port 2(wlan0-1) entered blocking state
[ 785.233105] br-lan: port 2(wlan0-1) entered disabled state
[ 785.239092] device wlan0-1 entered promiscuous mode
[ 785.244181] br-lan: port 2(wlan0-1) entered blocking state
[ 785.249746] br-lan: port 2(wlan0-1) entered forwarding state
[ 785.300206] br-lan: port 2(wlan0-1) entered disabled state
[ 786.964331] IPv6: ADDRCONF(NETDEV UP): wlan0: link is not ready
[ 790.441417] wlan0: authenticate with 00:0a:f5:d8:f2:d4
[ 790.453074] wlan0: send auth to 00:0a:f5:d8:f2:d4 (try 1/3)
[ 790.462660] wlan0: authenticated
[ 790.470156] wlan0: associate with 00:0a:f5:d8:f2:d4 (try 1/3)
[ 790.494245] wlan0: RX AssocResp from 00:0a:f5:d8:f2:d4 (capab=0x431 status=0 aid=1)
[ 790.502236] wlan0: associated
[ 790.518573] IPv6: ADDRCONF(NETDEV CHANGE): wlan0: link becomes ready
[ 792.008548] IPv6: ADDRCONF(NETDEV CHANGE): wlan0-1: link becomes ready
[ 792.015440] br-lan: port 2(wlan0-1) entered blocking state
[ 792.021041] br-lan: port 2(wlan0-1) entered forwarding state
[ 4113.437185] usb 1-1: new high-speed USB device number 2 using ehci-platform
[ 4113.630604] usb-storage 1-1:1.0: USB Mass Storage device detected
[ 4113.647551] scsi host0: usb-storage 1-1:1.0
[ 4114.678855] scsi 0:0:0:0: Direct-Access      SanDisk Cruzer Blade      1.27 PQ: 0 ANSI: 6
[ 4114.691045] sd 0:0:0:0: [sda] 15633408 512-byte logical blocks: (8.00 GB/7.45 GiB)
[ 4114.707299] sd 0:0:0:0: [sda] Write Protect is off
[ 4114.712184] sd 0:0:0:0: [sda] Mode Sense: 43 00 00 00
[ 4114.717496] sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[ 4114.750797] sda: sda1
[ 4114.760152] sd 0:0:0:0: [sda] Attached SCSI removable disk
root@Pulpstone:~#

```

Figure 6.2.12: Usb Device Detected on SpyD

```
File Edit View Search Terminal Help
root@Pulpstone:~#
root@Pulpstone:~#
root@Pulpstone:~#
root@Pulpstone:~#
root@Pulpstone:~# ls /dev
bus          mtd1           mtdblock1      random        ttyS2
console      mtd1ro         mtdblock2      sda          ttyS3
cpu_dma_latency mtd2          mtdblock3      sdal         ttyS4
full         mtd2ro         mtdblock4      shm          ttyS5
gpiochip0    mtd3           mtdblock5      tty          ttyS6
gpiochip1    mtd3ro         mtdblock6      ttyS0        ttyS7
gpiochip2    mtd4           mtdblock7      network_latency  ttyS1        ttyS8
hwrng        mtd4ro         mtdblock8      network_throughput  ttyS10       ttyS9
kmsg         mtd5           mtdblock9      null         ttyS11       urandom
log          mtd5ro         mtdblock10     port          ttyS12       watchdog
memory_bandwidth mtd6          mtdblock11     ppp           ttyS13       watchdog0
mtd0         mtd6ro         mtdblock12     ptmx          ttyS14       zero
mtd0ro       mtdblock0      mtdblock13     pts           ttyS15

root@Pulpstone:~#
```

Figure 6.2.13: *sda* & *sda1* in */dev* on SpyD

USB storage is installed on SpyD. This will provide the storage needed for attacks, scripts, files, etc.

The main script (home) of the device interface is as follows:



*Figure 6.2.14: SpyD: Homepage*

Code:

```
#!/bin/bash
clear
echo "
-----#
# #####      ######
# # ##### # # # #
##### # ##### # # # #
# #####      # # #
# # #      # # #
##### #      # ######
-----"

echo ""
echo "Please select the SpyD Operating Mode"
1: User Mode
2: Tester Mode
3: Admin Mode
4: Restart Device
5: Exit"
echo ""
read -p "Enter your Choice : " mode
case $mode in
1)sh ./UserMode.sh
;;
2)sh ./TesterMode.sh
;;
3)sh ./AdminMode.sh
;;
4)reboot
;;
5)exit
;;
*)echo "Please enter the correct option"
;;
esac
```

User mode:

```
root@SpyD: ~/SpyD
Enter your Choice : 1

#####
# # #
# # #
#####
# # #
# # #
#####
# # #
# # #
#####
# # #
# # #

1. Open A Browser and GoTo https://192.168.1.1/cgi-bin/luci
2. GoTo Network > Wireless
3. Scan Wireless Networks > Select Network > Enter PassPhrase
4. Save And Apply > Enjoy

Interface      Chipset      Driver
wlan0          Unknown       mt76_wmac - [phy0]
mon0           Unknown       mt76_wmac - [phy0] (removed)
mon1           Unknown       mt76_wmac - [phy0]
wlan0-1         Unknown       mt76_wmac - [phy0]
IEEE           Unknown       Unknown (MONITOR MODE NOT SUPPORTED)
802.11          Unknown       Unknown (MONITOR MODE NOT SUPPORTED)

root@SpyD: ~/SpyD
802.11          Unknown       Unknown (MONITOR MODE NOT SUPPORTED)
Mode:Master      Unknown       Unknown (MONITOR MODE NOT SUPPORTED)
Tx-Power=18      Unknown       Unknown (MONITOR MODE NOT SUPPORTED)
dBm             Unknown       Unknown (MONITOR MODE NOT SUPPORTED)

br-lan          no wireless extensions.

wlan0           IEEE 802.11  ESSID:off/any
                Mode:Managed Access Point: Not-Associated Tx-Power=18 dBm
                RTS thr:off   Fragment thr:off
                Encryption key:off
                Power Management:off

mon1            IEEE 802.11  Mode:Monitor Tx-Power=18 dBm
                RTS thr:off   Fragment thr:off
                Power Management:off

wlan0-1          IEEE 802.11  Mode:Master Tx-Power=18 dBm
                RTS thr:off   Fragment thr:off
                Power Management:off

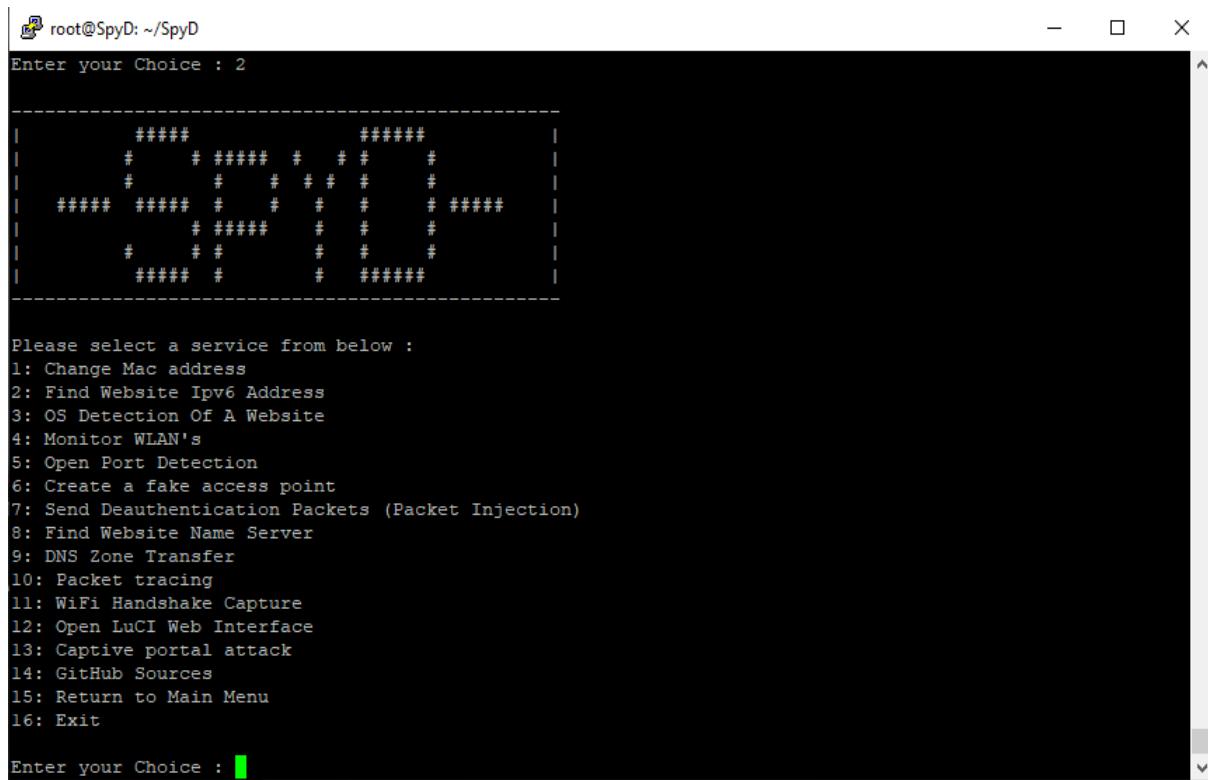
eth0            no wireless extensions.

lo              no wireless extensions.

Exit ? (1/0) : 1
```

Figure 6.2.15: SpyD: User Mode

Tester mode:



root@SpyD: ~/SpyD

```
Enter your Choice : 2

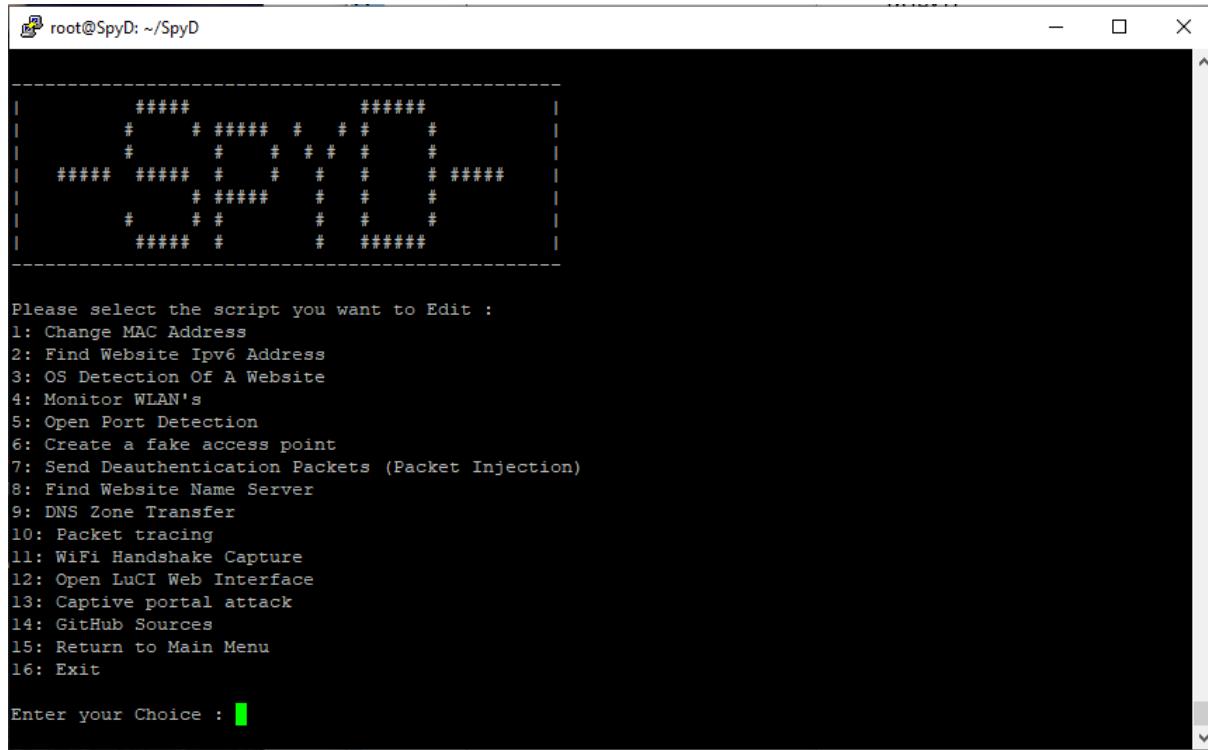
| #####      ##### |
| #  # ##### #  # #  # |
| #  #  #  #  #  #  #  # |
| ##### ##### #  #  #  # |
| #  # ##### #  #  #  # |
| #  #  #  #  #  #  #  # |
| ##### ##### #  #  #  # |

Please select a service from below :
1: Change Mac address
2: Find Website Ipv6 Address
3: OS Detection Of A Website
4: Monitor WLAN's
5: Open Port Detection
6: Create a fake access point
7: Send Deauthentication Packets (Packet Injection)
8: Find Website Name Server
9: DNS Zone Transfer
10: Packet tracing
11: WiFi Handshake Capture
12: Open LuCI Web Interface
13: Captive portal attack
14: GitHub Sources
15: Return to Main Menu
16: Exit

Enter your Choice : [green prompt]
```

Figure 6.2.16: SpyD: Tester Mode

Admin mode:



root@SpyD: ~/SpyD

```
Enter your Choice : 2

| #####      ##### |
| #  # ##### #  # #  # |
| #  #  #  #  #  #  #  # |
| ##### ##### #  #  #  # |
| #  # ##### #  #  #  # |
| #  #  #  #  #  #  #  # |
| ##### ##### #  #  #  # |

Please select the script you want to Edit :
1: Change MAC Address
2: Find Website Ipv6 Address
3: OS Detection Of A Website
4: Monitor WLAN's
5: Open Port Detection
6: Create a fake access point
7: Send Deauthentication Packets (Packet Injection)
8: Find Website Name Server
9: DNS Zone Transfer
10: Packet tracing
11: WiFi Handshake Capture
12: Open LuCI Web Interface
13: Captive portal attack
14: GitHub Sources
15: Return to Main Menu
16: Exit

Enter your Choice : [green prompt]
```

Figure 6.2.17: SpyD: Admin Mode

Implementation files and the main module directory:

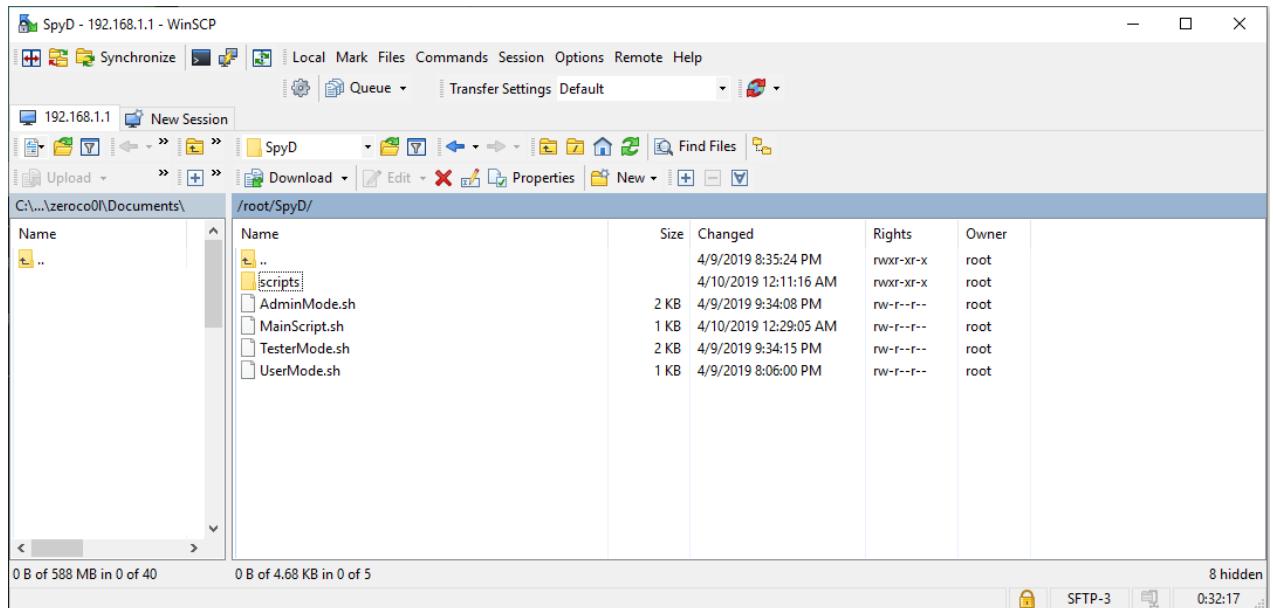


Figure 6.2.18: Directory

Service Scripts:

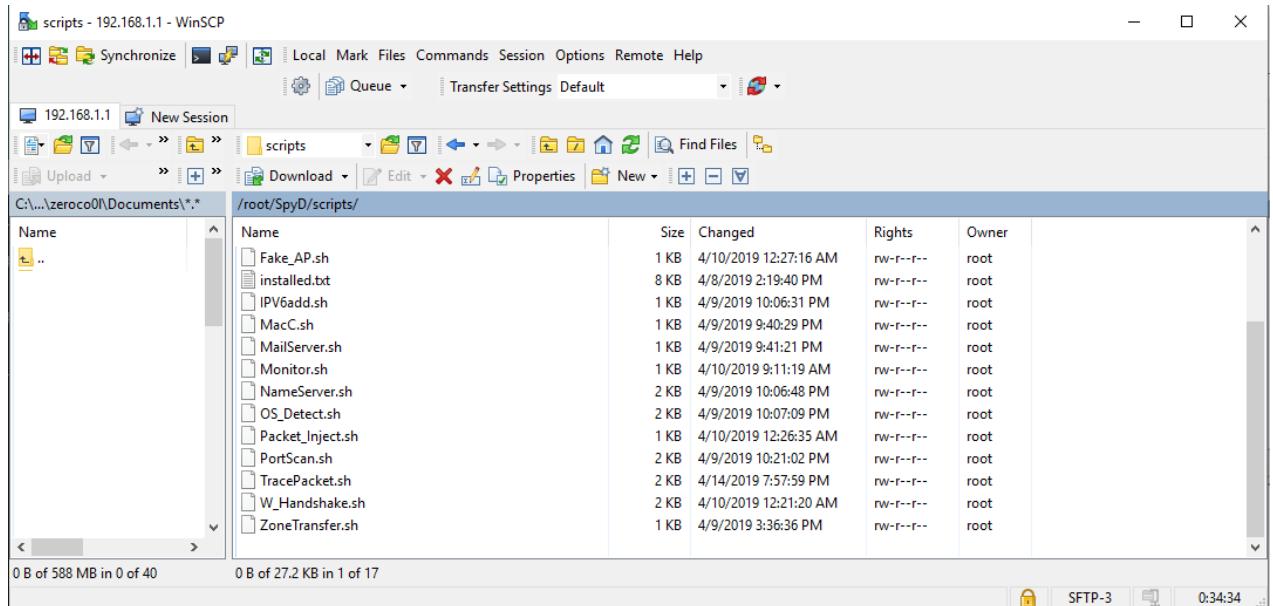
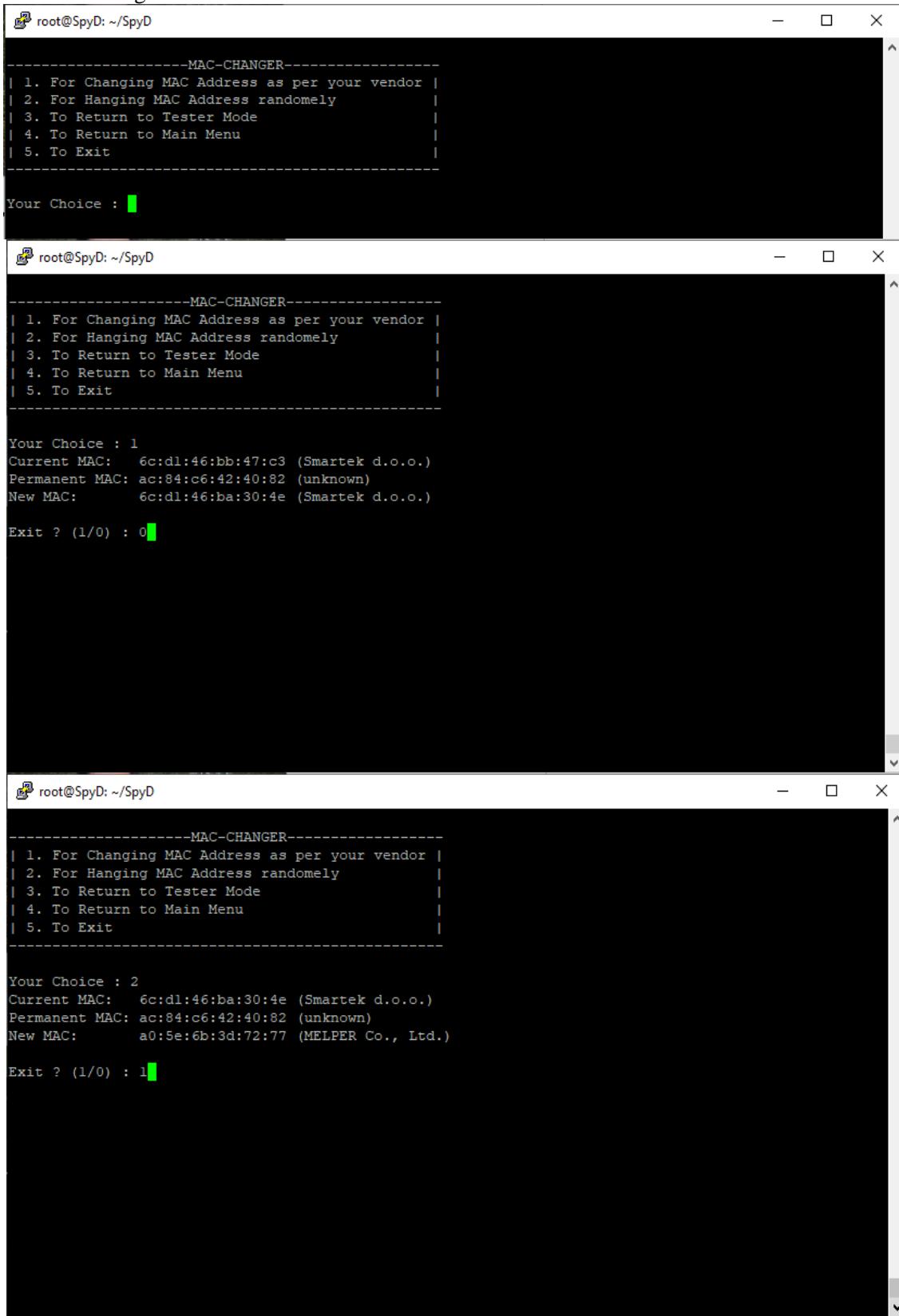


Figure 6.2.19: Service Scripts

## Some of the Services Provided

## 1. MAC Changer



The figure consists of three vertically stacked terminal windows, each titled "root@SpyD: ~/SpyD". Each window displays a menu for the MAC Changer service. The menu options are:

```
-----MAC-CHANGER-----
| 1. For Changing MAC Address as per your vendor |
| 2. For Hanging MAC Address randomly           |
| 3. To Return to Tester Mode                  |
| 4. To Return to Main Menu                   |
| 5. To Exit                                |
```

Below the menu, the user is prompted for their choice:

```
Your Choice : [input]
```

The first window shows the initial menu. The second window shows the menu followed by the current MAC address (6c:dl:46:bb:47:c3), permanent MAC address (ac:84:c6:42:40:82), and new MAC address (6c:dl:46:ba:30:4e). The third window shows the menu followed by the current MAC address (6c:dl:46:ba:30:4e), permanent MAC address (ac:84:c6:42:40:82), and new MAC address (a0:5e:6b:3d:72:77).

Figure 6.2.20: MAC Changer

## 2. Finding IPV6 Address of a Website

The figure consists of two vertically stacked terminal windows, both titled "root@SpyD: ~/SpyD".

The top window displays a menu with the following options:

```
-----IPV6-ADDRESS-----
| 1. Find IPV6 Address of website
| 2. To Return to Tester Mode
| 3. To Return to Main Menu
| 4. To Exit
```

Below the menu, the text "Your Choice : " is followed by a cursor.

The bottom window shows the user selecting option 1 and entering "www.google.com" as the website address. It then displays the results of the search:

```
-----IPV6-ADDRESS-----
| 1. Find IPV6 Address of website
| 2. To Return to Tester Mode
| 3. To Return to Main Menu
| 4. To Exit

Your Choice : 1
Enter Website (www.example.com) : www.google.com
Server:      127.0.0.1
Address:     127.0.0.1#53

Non-authoritative answer:
www.google.com has AAAA address 2404:6800:4009:805::2004

Exit ? (1/0) : 1
```

Figure 6.2.21: IPV6 Changer

## 3. OS Detection of a Website

A single terminal window titled "root@SpyD: ~/SpyD".

The window displays a menu with the following options:

```
-----OS-DETECTION-----
| 1. Find OS of a website Politely
| 2. Find OS of a website Secretly
| 3. Find OS of a website Aggressively
| 4. To Return to Tester Mode
| 5. To Return to Main Menu
| 6. To Exit
```

Below the menu, the text "Your Choice : " is followed by a cursor.

Figure 6.2.22: OS Detection

## 4. Monitor WLANs

The figure consists of three vertically stacked terminal windows, each with a title bar showing the session as 'root@SpyD: ~/SpyD'.

**Top Terminal:**

```
-----MONITOR-WLANS-----
| 1. To Monitor WLANs           |
| 2. To Return to Tester Mode   |
| 3. To Return to Main Menu     |
| 4. To Exit                   |
-----
```

Your Choice : [ ]

**Middle Terminal:**

```
Usage: ps
Show list of processes

      w      Wide output
ps: unrecognized option: a
BusyBox v1.28.4 () multi-call binary.

Usage: ps
Show list of processes

      w      Wide output

Interface      Chipset      Driver
wlan0          Unknown       mt76_wmac - [phy0]
               (monitor mode enabled on mon0)
mon1           Unknown       mt76_wmac - [phy0]
wlan0-1         Unknown       mt76_wmac - [phy0]
IEEE            Unknown       Unknown (MONITOR MODE NOT SUPPORTED)
802.11          Unknown       Unknown (MONITOR MODE NOT SUPPORTED)
Mode:Master     Unknown       Unknown (MONITOR MODE NOT SUPPORTED)
Tx-Power=18     Unknown       Unknown (MONITOR MODE NOT SUPPORTED)
dBm             Unknown       Unknown (MONITOR MODE NOT SUPPORTED)

Scan Networks for (seconds) : 15 [ ]
```

**Bottom Terminal:**

```
CH 13 ][ Elapsed: 12 s ][ 2019-04-09 23:46

BSSID          PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID
98:DE:D0:FE:CC:BA -30      145        874    0  4  54e. WPA2 CCMP   PSK  zeroco01 - NTWK

BSSID          STATION      PWR  Rate     Lost   Frames Probe
(not associated) 64:A2:F9:C7:1D:A2 -88    0 - 1      1      2  WR1505N3-ECB2
(not associated) DA:A1:19:4C:5E:6F -84    0 - 1      1      2
(not associated) DA:A1:19:E0:70:9C -87    0 - 1      0      2
(not associated) DA:A1:19:C0:B5:56 -88    0 - 1      1      2
(not associated) A0:32:99:3F:C8:33 -88    0 - 1      0      1  Bljp-cmFmaXFkdw
(not associated) DA:A1:19:C6:5E:34 -88    0 - 1      0      1
98:DE:D0:FE:CC:BA 04:B1:67:C0:50:BD -43  0e- 6      0      871

Caught signal 14 (SIGALRM). Please contact the author!
Exit ? (1/0) : [ ]
```

Figure 6.2.23: Monitor WLAN

## 5. Open Port Detection

The figure consists of three vertically stacked terminal windows, each titled "root@SpyD: ~/SpyD".

**Top Terminal:**

```
-----PORT-SCANNER-----
| 1. For Scanning Port of Known IP Address |
| 2. For Scanning Port Of a Website          |
| 3. To Return to Tester Mode                |
| 4. To Return to Main Menu                  |
| 5. To Exit                                |

Your Choice : [green prompt]
```

**Middle Terminal:**

```
-----PORT-SCANNER-----
| 1. For Scanning Port of Known IP Address |
| 2. For Scanning Port Of a Website          |
| 3. To Return to Tester Mode                |
| 4. To Return to Main Menu                  |
| 5. To Exit                                |

Your Choice : 1

Please enter the IP to Scan :15.23.15.22
Please enter the port number :80

Host: 15.23.15.22 ()      Status: Down
# Nmap done at Tue Apr  9 23:49:54 2019 -- 1 IP address (0 hosts up) scanned in 4.83 seconds
Exit ? (1/0) : [green prompt]
```

**Bottom Terminal:**

```
-----PORT-SCANNER-----
| 1. For Scanning Port of Known IP Address |
| 2. For Scanning Port Of a Website          |
| 3. To Return to Tester Mode                |
| 4. To Return to Main Menu                  |
| 5. To Exit                                |

Your Choice : 2

Enter Website (www.example.com) : www.slrte.in
www.slrte.in has an address of slrtce.in.
206.221.182.74
Enter The Port To Scan : 80

Host: 206.221.182.74 (in4.fastwebhost.com)      Status: Up
Host: 206.221.182.74 (in4.fastwebhost.com)      Ports: 80/open/tcp//http///
Host: 206.221.182.74 (in4.fastwebhost.com)      Status: Up
Host: 206.221.182.74 (in4.fastwebhost.com)      Ports: 80/open/tcp//http///
# Nmap done at Tue Apr  9 23:51:25 2019 -- 2 IP addresses (2 hosts up) scanned in 4.86 seconds
Exit ? (1/0) : [green prompt]
```

Figure 6.2.24: Open Port Detection

## 6. Create a Fake Access Point

```
root@SpyD: ~/SpyD
-----
|-----FAKE-ACCESS-POINT-----|
| 1. To Create a Fake Access Point |
| 2. To Return to Tester Mode   |
| 3. To Return to Main Menu    |
| 4. To Exit                   |
-----
Your Choice : [
```

```
root@SpyD:~/SpyD
CH 6 ][ Elapsed: 8 s ][ 2019-04-09 23:54

BSSID          PWR  Beacons    #Data, #/s   CH   MB     ENC   CIPHER AUTH ESSID
AE:84:C6:42:40:82 -1        0       16    2   4   -1   WPA           <length: 0>
98:DE:D0:FE:CC:BA -32      101      12    0   4  54e. WPA2 CCMP   PSK  zerocool - NTWK

BSSID          STATION          PWR  Rate     Lost    Frames  Probe
(not associated) DA:A1:19:54:20:03 -77   0 - 1     0       3
(not associated) DA:A1:19:C4:98:71 -84   0 - 1     0       1
(not associated) DA:A1:19:9F:8B:E9 -84   0 - 1     0       4
AE:84:C6:42:40:82 00:0A:F5:D8:F2:D4 -18   0 - 6     0      27
98:DE:D0:FE:CC:BA 04:B1:67:C0:50:BD -38   0e- 6     0      15
```

```
root@SpyD: ~/SpyD
          BSSID      PWR  Beacons    #Data, #/s   CH   MB     ENC   CIPHER AUTH ESSID
AE:84:C6:42:40:82  -1        0       39      0   4    -1     WPA                <length: 0>
98:DE:D0:FE:CC:BA -32      194       25      0   4  54e.   WPA2   CCMP    PSK  zeroCo01 - NTWK

          BSSID      STATION           PWR  Rate     Lost    Frames  Probe
(not associated) DA:A1:19:4F:76:D3 -43    0 - 1      1        14 DIRECT-dz-S, Gothwal 903
(not associated) DA:A1:19:54:20:03 -77    0 - 1      0        3
(not associated) DA:A1:19:C4:98:71 -84    0 - 1      0        1
(not associated) DA:A1:19:9F:8B:E9 -84    0 - 1      0        4
(not associated) DA:A1:19:BA:D1:5C -87    0 - 1      0        1
AE:84:C6:42:40:82 00:0A:F5:D8:F2:D4 -18    0 - 1e     1        60
98:DE:D0:FE:CC:BA 04:B1:67:C0:50:BD -38    0e - 6      0        15

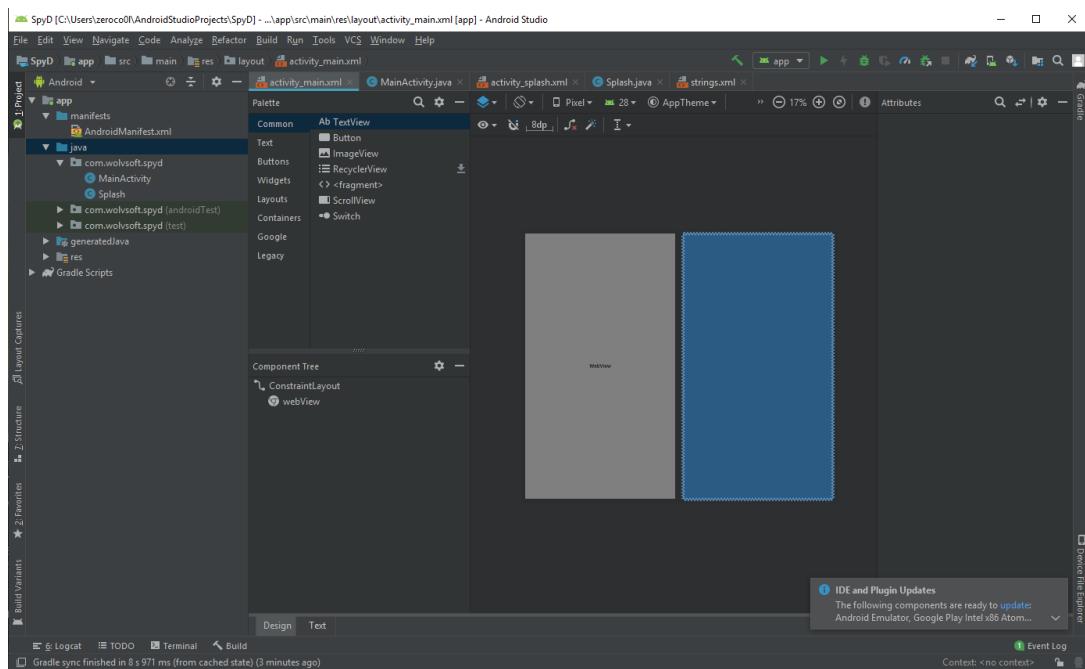
Caught signal 14 (SIGALRM). Please contact the author!

Select Channel of Fake access Point : 4
Please Enter the name of the fake access point : xorro
23:55:11 Created tap interface at0
23:55:11 Trying to set MTU on at0 to 1500
23:55:11 Trying to set MTU on mon0 to 1800
23:55:11 Access Point with BSSID AC:84:C6:42:40:82 started.
Error: Got channel -1, expected a value > 0.
```

*Figure 6.2.25: Fake access point creation*

## Android application

### 1. activity\_main.xml



### 2. activity\_splash.xml

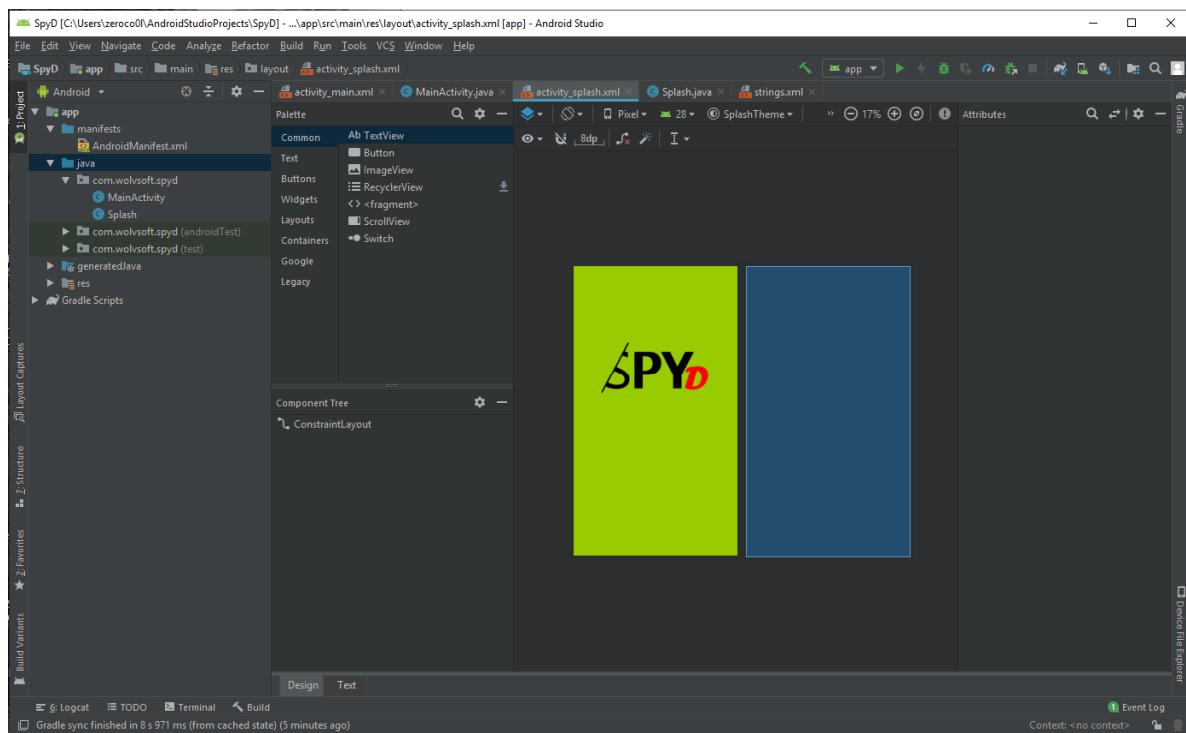
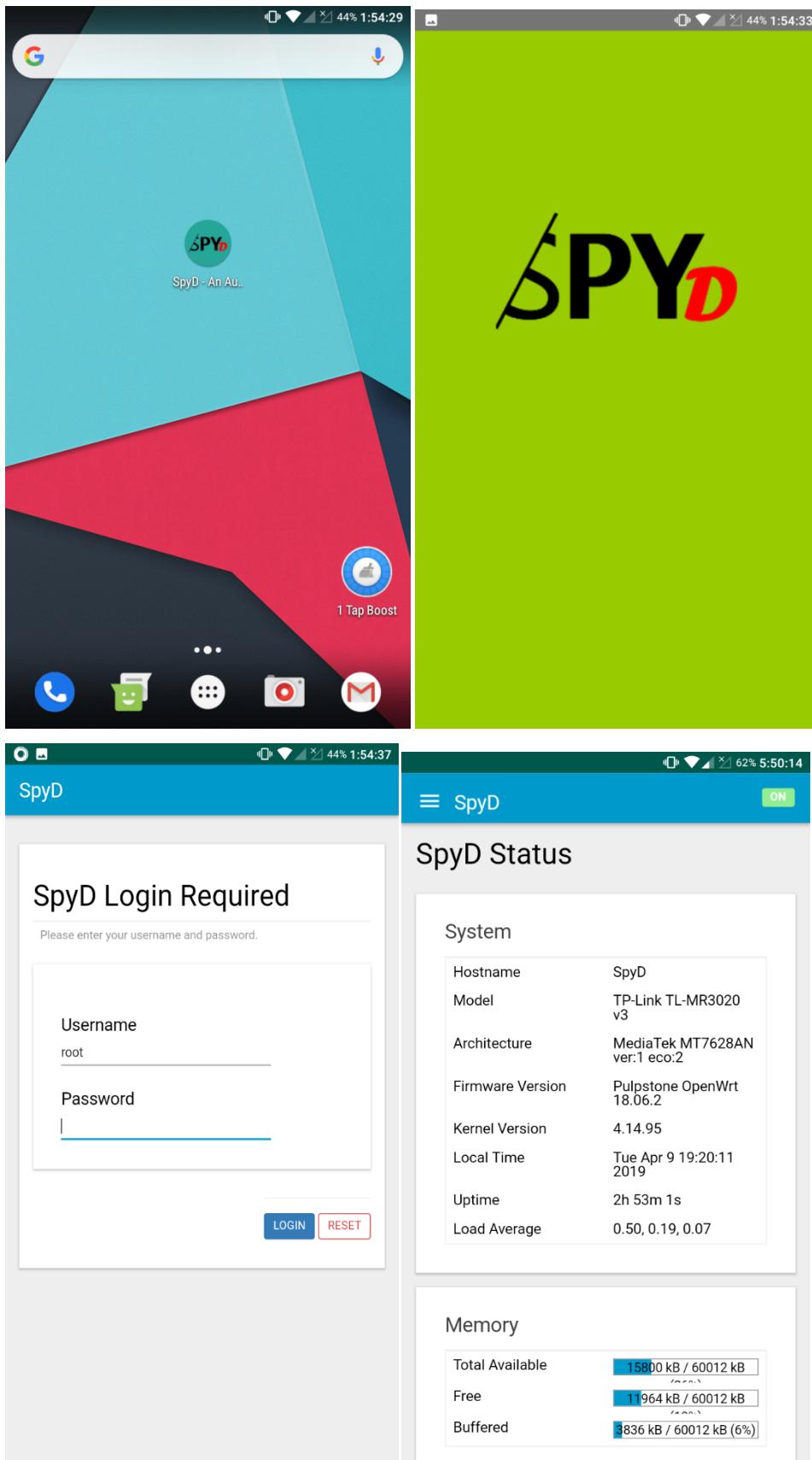


Figure 6.2.31:Android Environment

#### 4. Application



## Implementation Details

## SpyD: An Automated Security Tool

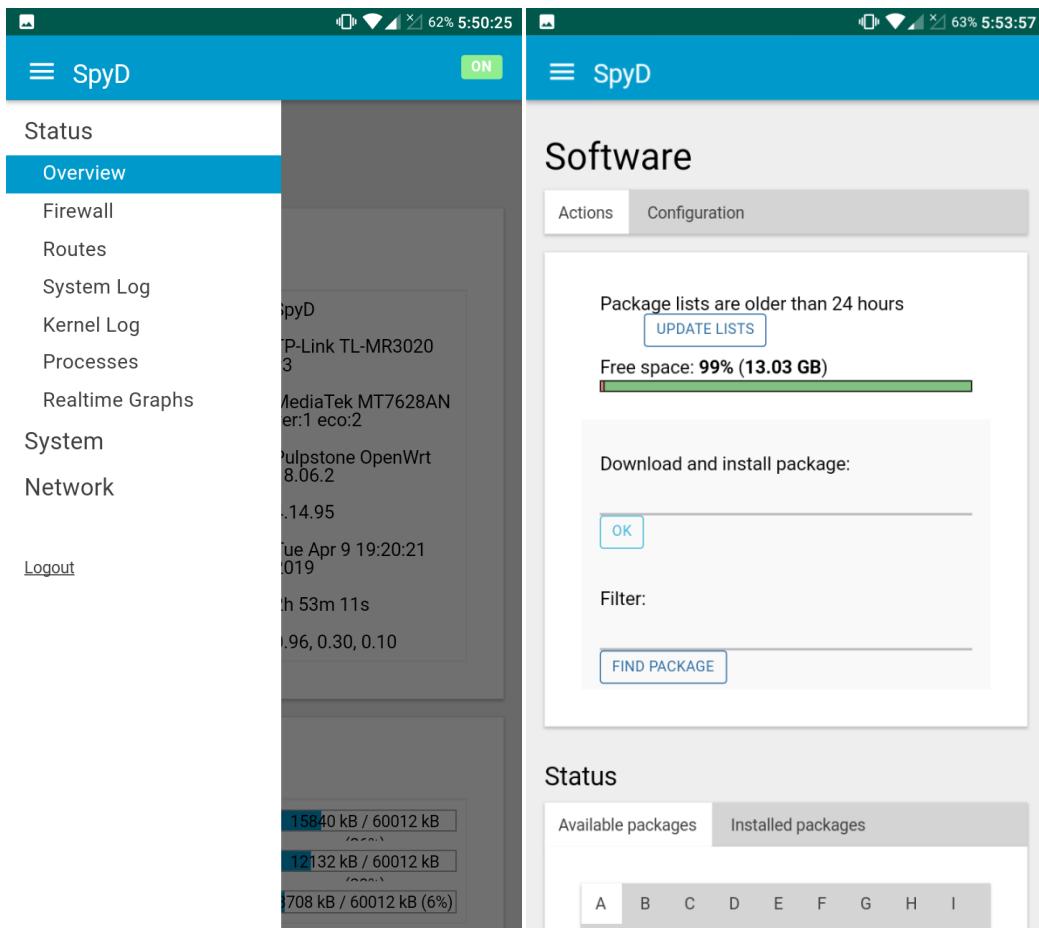


Figure 6.2.33:Android Interface

# **Chapter 7**

# **Conclusion**

## Chapter 7

# Conclusions

### 7.1 Conclusion

Internet usage is increasingly becoming important as more and more users access the Internet, and many users are using the Internet to express and share their opinions. Thus our Goal is to find the favorable or interesting way to provide security at a lower level as to provide security to each user.

In this project we propose a Penetration testing system where users/organizations can use various attacks to check whether their network is vulnerable against those. If the network is vulnerable, they can undertake necessary actions to avoid and remove those loopholes from the network. We use an OpenWRT based framework to add the functionalities to a chipset which can be employed locally at any site, and then perform tests or attacks using the provided predefined attacks. The experimental results show that the system is practical and the attacks are feasible.

This tool will ultimately help the users or small organizations to manage network security without hassles of going and researching the whole market thereby wasting a lot of money and time. Overall our project will give a way to perform tests at a local platform without any need of professionals, thus, improving the efficiency of organizations that are small scaled.

### 7.2 Future Scope

- Intrusion detection

By monitoring system activity through examining vulnerabilities in the system, the integrity of files and conducting an analysis of patterns based on already known attacks, SpyD, along with multiple additions like a session generator and tracker, will be able to detect intruders. It will also automatically monitor the Internet to search for any of the latest threats which could result in a future attack.

- Support a wide range of users with the use of a chipset with a higher computational speed.

# Chapter 8

# References

## References

- [1] Cybergrenade: Automated Exploitation of Local Network Machines via Single Board Computers. Available: <https://ieeexplore.ieee.org/document/8108803>
- [2] Computer Network Security and Technology Research by Fan Yan, Yang Jianwen2, Cheng Lin1 Available: <https://ieeexplore.ieee.org/document/7263569>
- [3] TP-Link TL-MR3020 OpenWRT flashing. Available:  
<https://wiki.openwrt.org/toh/tp-link/tl-mr3020>
- [4] K. Scarfone, M. Souppaya, A. Cody, and A. Orebaugh, “Technical guide to information security testing and assessment”, National Institute of Standards and Technology, Tech. Rep., 2008.
- [5] F. Palmieri, U. Fiore, and A. Castiglione, “Automatic security assessment for next generation wireless mobile networks,” Mobile Information Systems, vol. 7, no. 3, pp. 217–239, 2011.
- [6] F. Holik, J. Horalek, O. Marik, S. Neradova, and S. Zitta, “Effective penetration testing with metasploit framework and methodologies,” in 2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI), Nov. 2014, pp. 237–242. DOI:10.1109/CINTI.2014.7028682.
- [7] Y. Hu, D. Sulek, A. Carella, J. Cox, A. Frame, and K. Cipriano, “Employing miniaturized computers for distributed vulnerability assessment,” in 11<sup>th</sup> International Conference for Internet Technology and Secured Transactions (ICITST), Dec. 2016, pp. 57–61. DOI: 10.1109/ICITST.2016.7856666.

# **Chapter 9**

# **Paper Published**

# SpyD: An automated security tool

<sup>1</sup>Abhishek Pandey, <sup>2</sup>Harshvardhan Dubey, <sup>3</sup>Pratik Varma, <sup>4</sup>Sumit Awinash, <sup>5</sup>Prof. Reena Kothari

<sup>1</sup>B.E., <sup>2</sup>B.E., <sup>3</sup>B.E., <sup>4</sup>B.E., <sup>5</sup>M. Tech

<sup>1</sup>Information Technology,

<sup>1</sup>Mumbai University, Mumbai, India

*Abstract : As people are realizing the importance of cyber security, the price for hacking tool such as Pineapple is increasing rapidly. One Pineapple device can cost up to Rs. 13000+Tax+Shipping charges. Due to high cost people do not prefer buying such a product which hamper their security as an organization. By building SpyD we can provide a similarly effective product at a very considerably lower price which will help the organization to manage the security. SpyD is a automated network penetration device/bot which will be able to perform all major operations through the means of an automated script with built in failsafe which can be performed by any professional ethical hacker. This will ensure that the routine jobs are performed more precisely as compared to a hired professional, thus, eliminating the threat of any man-made errors. Additionally, SpyD will automate the work of softwares like Nmap, Aircrack-NG, Karma, Tcpdump etc.*

**Index Terms - Automation, Cyber security, Penetration testing.**

## I. INTRODUCTION

This chapter will introduce the reader with SpyD, which is used as a penetration testing tool and a hacker device. It will light up the topics like the description of the project and the former formulation of the problem behind it as well as what motivated the makers of the project to take a decision to make this project and its related problem solutions and thus covering up the scope of the project. SpyD is a follow up of the established Wi-Fi Pineapple and adds various functionalities like complete automation, packet monitoring, etc. The secondary objective is to lower the cost of the device to target smaller organizations and start-ups. In this project we aim to overcome the shortcomings of conventional testing tools like rutabaga or PineAP that are need for a professional, high cost, manual work etc.

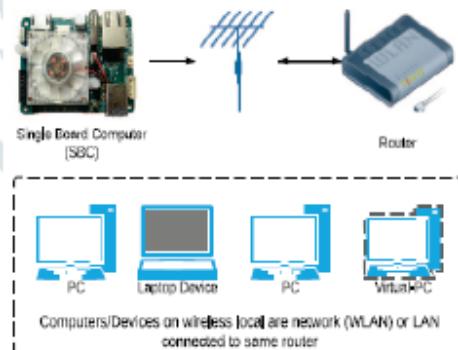


Fig. 1. SpyD system flow

## II. RELATED WORK

Great deals of researchers have worked on using SBCs as potential pentesting tool kits. However, there are very few researchers who have expounded on a way to automate the pentesting tools within a single SBC to take full advantage of the SBC's capabilities. Another work automates MSFconsole successfully in a similar way we end up automating it, by manipulating resource scripts. However, the caveat is that for their paper, it would only work in the case that they knew ahead of time which exploits to run, thereby not being as flexible in terms of our solution in which the resource script file sent with MSFconsole is truly procedurally generated each time we run the script. The system also does not focus on using these tools in tandem with SBCs, which we are focusing on with this paper. Other than these, various penetration testing tools are being used currently in the industries that provide a compelling level of security to the firm's infrastructure, data and other details. An example of such a system is the Wifi PineApple system that is a penetration testing device just like the SpyD to provide security to the systems by aiming at the vulnerabilities present and alerting the organizations about them.

### III. SYSTEM DESIGN AND IMPLEMENTATION

#### A. Architecture and Flow of SpyD

In SpyD, the Pulpstone framework is flashed over the stock MR.3020 chipset. Various tools and base modules are installed over the chipset's framework to provide further support. An extroot firmware is added to provide storage support to the Chipset that can be used to add modules of larger storage and processing requirements.

Design will elaborate the step by step flow of SpyD based on user Benchmark thus giving up the detailed information as to the basic flow of the system.

Flowcharts are used in designing and documenting complex processes or programs. Like other types of diagrams, they help visualize what is going on and thereby help the people to understand a process, and perhaps also find flaws, bottlenecks, and other less-obvious features within it. There are many different types of flowcharts, and each type has its own repertoire of boxes and notational conventions.

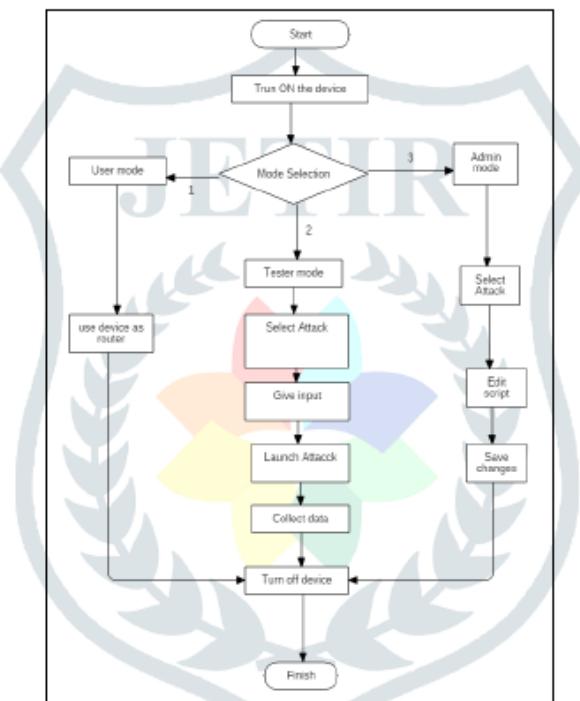


Fig. 2. Architecture of SpyD

As Kali Linux came with the 'Metasploit Penetration Testing Framework' built in, it was chosen to find and deliver exploits to the local machines using information given to it by the other two tools. All of the penetration-testing tools used were made generally to be manually used, with little, if any, support for automation, which presented a serious problem and demanded some unique solutions as we progressed.

#### B. Algorithm

1. Start
2. Turn on the device
3. Select Mode of operation
  - a: User mode
    - Use device as a router
  - b: Tester mode
    - i: Select Attack
      - Man in the middle Attack
      - Frame replay attack
      - Fake frame generation attack

- Create fake access point
- More attacks will be added
- ii: Give appropriate input with respect to attack
- iii: launch attack
- iv: Collect appropriate data
- v: Finish or go back to step i.
- c: Admin mode
  - i: Select Attack
    - Man in the middle Attack
    - Frame replay attack
    - Fake frame generation attack
    - Create fake access point
    - More attacks will be added
  - ii: Make changes in script as per requirement
  - iii: Finish or go back to step i.
- 4. Exit selected mode
- 5. Turn off the device or Change user mode
- 6. Stop

### C. Implementation

SpyD has a home interface that allows the user to select the mode they wish to operate in. The user may select the User mode or the Admin mode. The admin mode has various controls and privileges over normal users.

Following this, the user may select any one of the many available tests for the system. Listing 1, the user has selected to inject packets into a channel to test the vulnerability of the channel.

```
#!/bin/bash
#PACKET_INJECTIO
ifconfig wlan0 down
airmon-ng start wlan0
airodump-ng mon0
echo "Please select the channel in which you wish to inject the packets."
read CHANNEL
echo "Please enter the name of the fake a access point"
read AP_NAME
airbase-ng --essid $AP_NAME -c $CHANNEL mon0
```

Listing 1: Fake access point and packet injection

- With the use of the airbase -ng command, the \$CHANNEL channel is attacked.
- The \$AP\_NAME is a fake access point that is created to launch the attack from without getting traced to the original attacker.

If the user wants to test the vulnerability of the system against web port scanners, they can launch the Listing 2 script to attack the available web ports.

```
nmap -oG - $IP -p $PORT -vv > /Home/$IP.txt
```

Listing 2: Web port scanning

- The IP that is specified is the URL of the website.
- The PORT that is specified in the script is the ports that are to be checked for any vulnerability that may be present in the system.

To go into monitoring mode, the system executes another script that enables SpyD to monitor all the incoming and outgoing packets in the whole network, to and fro from all the access points and devices.

```
#!/bin/bash
#monitor the network
ifconfig wlan0 down
airmon-ng start wlan0
airodump-ng mon0
```

Listing 3: Packet monitoring

- The airmon-ng and airodump commands are used to assist the system in packet monitoring.

**D. Technologies used**

## 1. OpenWrt

OpenWrt is an open source project for embedded operating system based on Linux, primarily used on embedded devices to route network traffic. The main components are Linux, util-linux, musl, and BusyBox.

**E. Programming languages**

## 1. Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

## IV. APPLICATIONS

The project aims to ease the process of Network security testing, so it will be easy for the users to perform tests to check for vulnerabilities as per the requirement. In many organizations, the testing tools which are used are the conventional ones which are usually operated and worked upon by testing professionals. This isn't a feasible option for end users and the organizations that are relatively smaller than those organizations. Through this proposed system, the end users as well as small scale organizations can perform tests and attacks to check for any vulnerability in their system and remove those flaws. The project uses various firmwares from the significant domain Network security like OpenWRT, LuCI, and operating systems like Python, etc to perform attacks like Man-in-the-middle, Framereplay, and fake Frame generation. All these functions are clustered in one chipset connected to the network to perform tests or attacks based on user's requirements.

## V. CONCLUSION

Internet usage is increasingly becoming important as more and more users access the Internet, and many users are using the Internet to express and share their opinions. Thus our Goal is to find the favorable or interesting way to provide security at a lower level as to provide security to each user.

In SpyD we propose a Penetration testing system where users/organizations can use various attacks to check whether their network is vulnerable against those. If the network is vulnerable, they can undertake necessary actions to avoid and remove those loopholes from the network. We use an OpenWRT based framework to add the functionalities to a chipset which can be employed locally at any site, and then perform tests or attacks using the provided predefined attacks. The experimental results show that the system is practical and the attacks are feasible.

This tool will ultimately help the users or small organizations to manage network security without hassles of going and researching the whole market thereby wasting a lot of money and time. Overall our project will give a way to perform tests at a local platform without any need of professionals, thus, improving the efficiency of organizations that are small scaled.

## ACKNOWLEDGMENT

We would like to express our sincere gratitude to the teaching faculty at SLRTCE whose timely inputs and suggestions, helped in the completion of the project. We would also like to thank the Library of our college for allowing us to carry out our research. Finally, we are thankful for having been given this opportunity to learn something new about the world of technology.

## REFERENCES

- [1] K. Scarfone, M. Souppaya, A. Cody, and A. Orebaugh, "Technical guide to information security testing and assessment", National Institute of Standards and Technology, Tech. Rep., 2008.
- [2] F. Palmieri, U. Fiore, and A. Castiglione, "Automatic security assessment for next generation wireless mobile networks," Mobile Information Systems, vol. 7, no. 3, pp. 217–239, 2011.
- [3] F. Holik, J. Horalek, O. Marik, S. Neradova, and S. Zitta, "Effective penetration testing with metasploit framework and methodologies," in 2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI), Nov. 2014, pp. 237–242. DOI:10.1109/CINTI.2014.7028682.
- [4] J. Muniz, Penetration testing with raspberry pi. Packt, 2015.

- [5] Y. Hu, D. Sulek, A. Carella, J. Cox, A. Frame, and K. Cipriano, "Employing miniaturized computers for distributed vulnerability assessment," in 11<sup>th</sup> International Conference for Internet Technology and Secured Transactions (ICITST), Dec. 2016, pp. 57–61. DOI: 10.1109/ICITST.2016.7856666.
- [6] TP-Link TL-MR3020 OpenWRT flashing. Available: <https://wiki.openwrt.org/toh/tp-link/tl-mr3020>

