SpyD: An automated security tool

[1]Abhishek Pandey, [2]Harshvardhan Dubey, [3]Pratik Varma, [4]Sumit Awinash, [5]Prof. Reena Kothari
[1]B.E., [2] B.E., [3] B.E., [4] B.E., [5]M. Tech
[1]Information Technology,
[1]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 be 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.

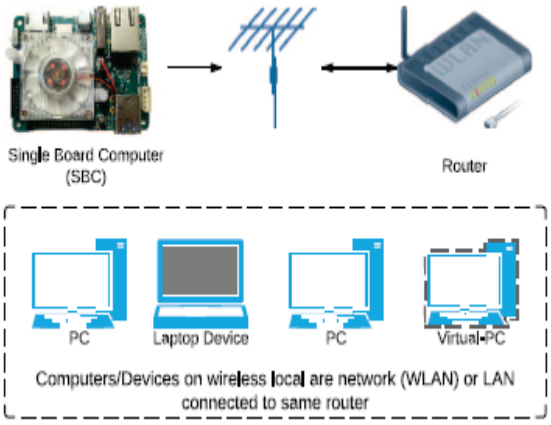

Fig. 1. SpyD system flow

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.

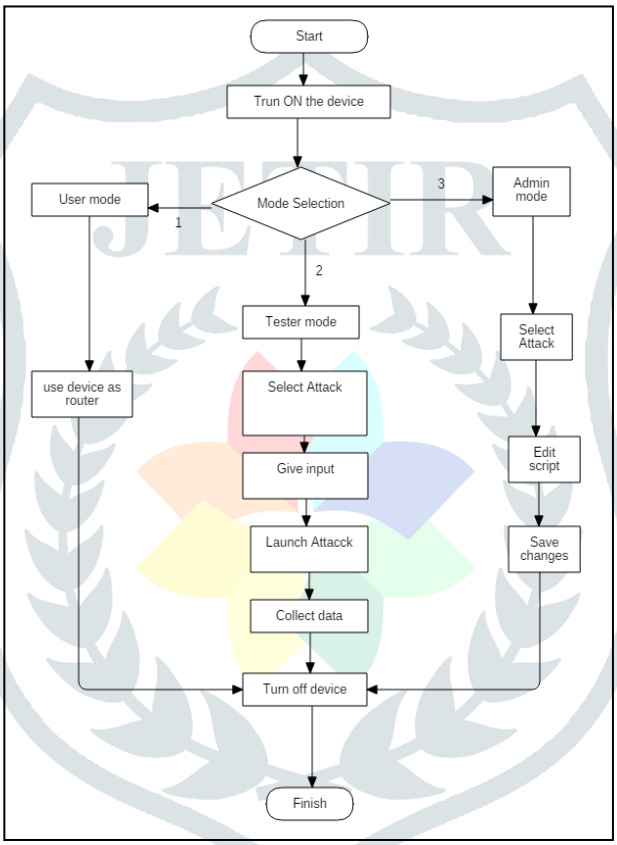

Fig. 2. Architecture of SpyD

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 andassessment", 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 15thInternationalSymposium 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. Packet, 2015.

[5] Y. Hu, D. Sulek, A. Carella, J. Cox, A. Frame, and K. Cipriano, "Employing miniaturized computers for distributed vulnerability assessment," in 11th 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