IMPLEMENTATION AND DETECTION OF FAKE ACCESS POINT ATTACKS IN OPEN WIRELESS NETWORKS

Rahil Sharma[*]

Abstract-- A **rogue access point** is a wireless access point that has either been installed on a secure company network without explicit authorization from a local administrator or has been created to allow a hacker to conduct a man in the middle attack. Rogue access points of the first kind can pose a security threat to large organizations with many employees, because anyone with access to the premises can install (maliciously or non-maliciously) an inexpensive wireless router that can potentially allow access to a secure network to unauthorized parties.

Problem of fake access points can lead to serious vulnerabilities in the network. Here in this paper first we explore how to successfully perform this type of attack followed by detection of fake access points from the real ones .Research on this topic reveals that passive monitoring technique can be used to scan the wireless network without even letting other devices to know that scanning is going on. In this way air traffic can easily be sniffed, which raises many security risks and threats related to sniffing the air traffic in wireless network that can easily reveal the personal identity of the end user who is using the wireless network. We have considered Man-in-the-Middle scenario to implement Fake Access Point where the end user has to compromise its security in order to gain access to the network resources. We have also proposed and developed solutions to detect the Fake Access Point in the network for two particular scenarios where attacker may or may not have internet connectivity.

[*] B Tech CSE, VIT Vellore

I . INTRODUCTION

In this paper we discuss the threats Fake Access Point pose to open wireless networks.

A wireless network can be attacked in many ways. One such way is by creation of unauthorized Fake Access Point (AP) in the network. A malicious attacker can sniff the beacon frames of authorized AP. He then extracts Medium Access Control (MAC) Address, SSID and BSSID of the authorized AP and puts it into his own frame. Thus, newly created AP now acts as a duplicate AP of the authorized AP. Once this fake AP is set up, users of authorized AP think that fake AP is the real one and connect to it. Now, the attacker can launch various attacks on these clients. Fake APs are dangerous than a direct attack on the network. This is due to following reasons.

i. When different clients try to connect to the authorized AP, they use *election algorithms* that are purely based on signal strength. If the attacker manages to increase its signal strength, he can easily lure the clients into using fake AP instead of authorized one. Once connected , attacker can make sure that client never uses authorized AP again by keeping its signal strength at constant level. Thus, client is now virtually quarantined from the rest of the network of authorized AP without the knowledge of his elimination.

ii. Beacons sent by fake AP have their fields copied from beacons of authorized AP. As a result, security system can also be fooled as it thinks that packets are coming from authorized AP itself. Thus fake AP evades its detection. It is difficult to detect the presence of fake AP in a wireless network. It is more difficult to exactly identify the device and block it .

iii. These fake access points can be easily created using beacon sniffers and packet injectors which only need a wireless Network Interface Card (NIC) which can work in monitor mode.

There are many ways in which the attacker uses to avoid detection. An experienced attacker will spoof the MAC address to match a legitimate AP. The attacker can also set power, channel, and SSID on the rogue AP to limit its effective coverage area, which decreases the chances of the rogue AP being detected.

II . RELATED RESEARCH

For detection of fake access points we use following methods:

Wired Based Detection Methods

David Buretto [14] proposed a TCP Fingerprinting approach which examines the subtle differences in how a target responds to various specially crafted packets in hope of determining the OS of the target system. We look for packet fields that are unique for different operating systems.The advantage of this method lies with simple starting a tool called Nmap that scans the entire network and gathers the analysed data for Rogue APs. The disadvantage lies with long running time for entire large network. As the act of determining the OS is intrusive, noisy and is likely to look like suspicious activity to Intrusion Detection System(IDS)s, it can't be considered one of best method .The other disadvantage is that TCP fingerprinting is not 100% accurate. Nmap uses the information to make a guess at the OS but there are bound to be times when it gives false positives or false negatives.

S Vishal[15] proposed that SNMP scanning can also be used to detect rogue APs. SNMP scanning is similar to TCP fingerprinting but instead of using the differences in the TCP/IP network stack it uses information obtained by using the SNMP protocol. The advantages of this method are similar to that of TCP fingerprinting.

Matt Jabocs[16] proposed packet sniffing for detecting rogue APs. Here a device is configured to run in promiscuous mode , analyzes the packets and examines the Ethernet headers to check that the MAC addresses are authorized MAC addresses. Another method is to compare MAC and ARP entries, and look for ports with multiple connections or to compare MAC and ARP entries and look for popular WLAN vendor MAC addresses. An example of a tool that to monitor MAC addresses is Arpwatch. Arpwatch is a tool that monitors Ethernet activity and keeps a database of Ethernet/IP address pairings. It also reports certain changes via email. The advantage of this method is that it is a continuous process and it is constantly monitoring the network for unauthorized MAC addresses. Once a rogue AP is connected to the network it is detected once it transmits any data.The disadvantage is that sniffing the traffic may be considered intrusive since it requires that all the traffic needs to be monitored.

Wireless Based Detection Methods

T Kim[17] discuses the active probing method as a way of detecting fake access points.The active probing method uses Probe Request Frames on each channel where it is able to detect wireless activity. When an AP comes within range of the client and receives a probe request frame it typically responds with a probe response frame containing the network ESSID.

T Kim[17] also proposes RF monitoring as a way of detecting fake access points .RF monitoring is a completely passive method of wireless LAN discovery known as radio frequency monitoring (RFMON). A client with a wireless card that is configured in RFMON mode enables the capture of all RF signals on the channels to which it is configured to listen. This method detects fake access points by monitoring for any suspicious frames.

Other techniques that use a combination of above methods are:

1)Detecting Rouge Access Point by Analyzing Traffic Characteristics of the network[1]

2) Hidden Markov Model[2]

3) Wireless IDS/WIPS(Wireless Intrusion Prevention System)

The first approach is useful in heterogeneous networks ie. combining both wireless and wired networks.It is implemented in two phases.

Phase 1

We detect the difference in wireless and ethernet by analysing the traffic.

Phase 2

Here the rouge access points are detected.When a attacker gains access to an unauthorized WLAN host who is using a rouge access point, he does a port scanning operation first to find open ports and the services running on them.Accordingly the vulnerabilities exist in the system.This creates a large traffic on a particular port. This phase depends on two threshold parameters based on straight- access and crossing-access attempts. Main disadvantage of this method is that it generates a lot of false positives.

The second approach is somewhat similar to the first one.

Phase 1

We detect traffic from both the real and fake access points.

Phase 2

We plan to know the inter arrival times for packets both from real and fake access points.The model aims at differentiating the real and fake access points based on inter arrival times of the frames (this can be found in timestamp field of the access point).Main disadvantage of this method is its dependability on the hardware.

Using Wireless Intrusion Prevention System/Wireless Intrusion Detection System to detect Rouge Access Points:

A wireless intrusion detection system monitors radio spectrum to find unauthorized devices . The system immediately alerts the system admin when a rouge access point is detected . It detects Fake Access Points by comparing MAC IDs of connecting devices.

Underlying Hardware Organization of Wireless Intrusion Prevention System

- **Sensors** — These devices scan the WiFi spectrum for packets and send data back to servers.
- **Server** — These devices analyze the packets sent by sensors and detect rouge access points.
- **Console** — It provides console for reporting and administration.

Comparison of Defense Techniques

Techniques	Cost	Features	Requirements
Detect rogues AP with sensor	High cost to install for wide network	Detects MITM attacks by analyzing packets	Large number of sensors
Rogue AP Protection System Based On Radius Authentication Server	High cost to installation and maintenance costs	Radius authentication server can communicate with the AP through a secure	Difficult to detect rogue AP on open environment
Detection technique using wired and wireless networks	High cost to installation and maintenance costs	Get information from wired and wireless network	Difficult to detect rogue AP on open environment. Gives a lot of false positives.
Detection using timestamp method	Hight cost of hardware and computation power needed	Detects Fake AP based on clock skews	Requires a lot of hardware cost and processing power.

III . Proposed Solution

As it is difficult to detect RAP's using the MAC address detection method as it gives a lot false results . New approach should be to weed out spoofed MAC IDs using advanced finger printing . Here we analyze the unique signatures exhibited by every device .The signatures emitted by each wifi devices are compared against the known signatures of authorized devices .Thus we easily weed out the spoofed device.But this method also fails when the attacker limits its area of coverage .As the sensors cant be installed everywhere .The sytem admin here should do manual rouge access point detection .The admin needs to make sure that the tool he uses for rouge access point detection also incorporates the use of an active probing component. This is necessary because the attackers generally turn off their BSSID to "cloak" themselves.

IV . Implementation Scheme

For carrying out this attack we will be using a linux distribution based on Ubuntu- Backtrack 5.To do the attacks we will use parts of the Aircrack-ng suite (wireless auditing tools),Ettercap(MITM tool) ,sslstrip,dnsspoof etc.

We will be running backtrack 5 in a virtual box(this will act as the attacker) on windows host .The windows host should have a internet connection .To perform various steps in the attack we will need a wireless network card that supports monitor mode and has packet injection capabilities .

In monitor mode the card is not associated with a network and passively listens for packets. A card supporting packet injection is used to send specially handcrafted network packets into the Ether .We are using Alfa AWUS036H USB adapter .Here we are focusing on setting up a fake ap to make clients/victims connect to it in place of an real AP .The fake AP will broadcast the real APs SSID and by providing a stronger signal we make clients attach to the fake AP .

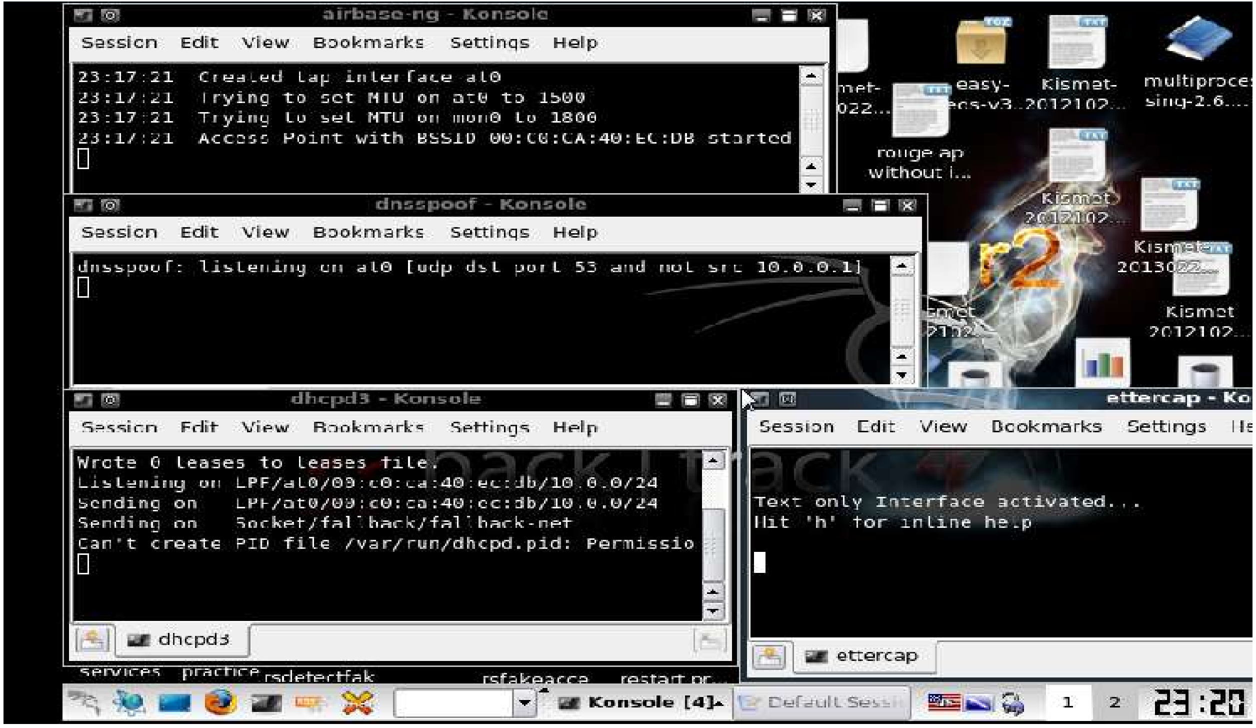
Pseudocode of Fake AP

```
Step 1.0 Initiate the setup for Fake AP Implementation.

Step 1.1 Scan wireless spectrum for Victim MAC ID and SSID.

Step 2.1 Perform the check for wireless card for monitor mode support:

  If ( Monitor mode support is found ) do

{

     Step 2.1.1 Spoof the MAC ID of the wireless card.

     Step 2.1.2 Put the card in monitor mode.

     Step 2.1.3 Create Fake Access Point using Airbase module.

     Step 2.1.4 Start DHCP server and configure it accordingly.

     Step 2.1.5 Bring the tapping interface up.

     Step 2.1.6 Configure IPTABLES and enable IP Forwarding.
```

```
        Step 2.1.7 Run monitoring software like Wireshark,Ettercap.
}
else
{
 The wireless card does not support monitor hence the attack is not supported by the hardware.
ReScan or Perform step 1.1.


}
```

Basic steps involved in performing this attack :

1)Creating a Access Point



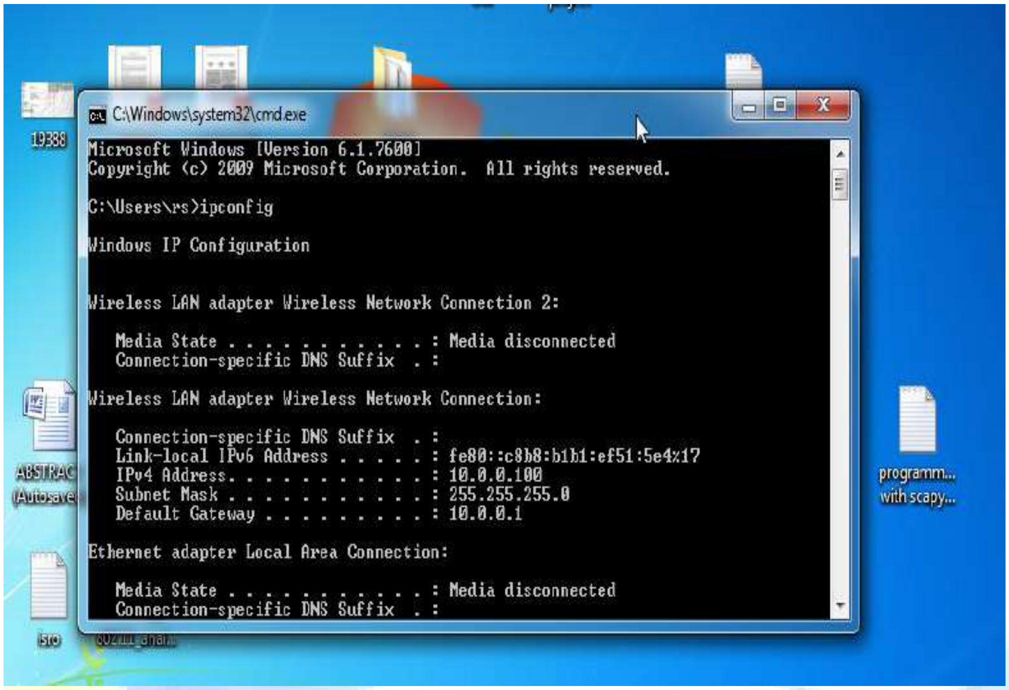
```
23:17:21  Created tap interface at0
23:17:21  Trying to set MTU on at0 to 1500
23:17:21  Trying to set MTU on mon0 to 1800
23:17:21  Access Point with BSSID 00:C0:CA:40:EC:DB started
```

```
dnsspoof: listening on at0 [udp dst port 53 and not src 10.0.0.1]
```

```
Wrote 0 leases to leases file.
Listening on LPF/at0/00:c0:ca:40:ec:db/10.0.0/24
Sending on   LPF/at0/00:c0:ca:40:ec:db/10.0.0/24
Sending on   Socket/fallback/fallback-net
Can't create PID file /var/run/dhcpd.pid: Permissio
```

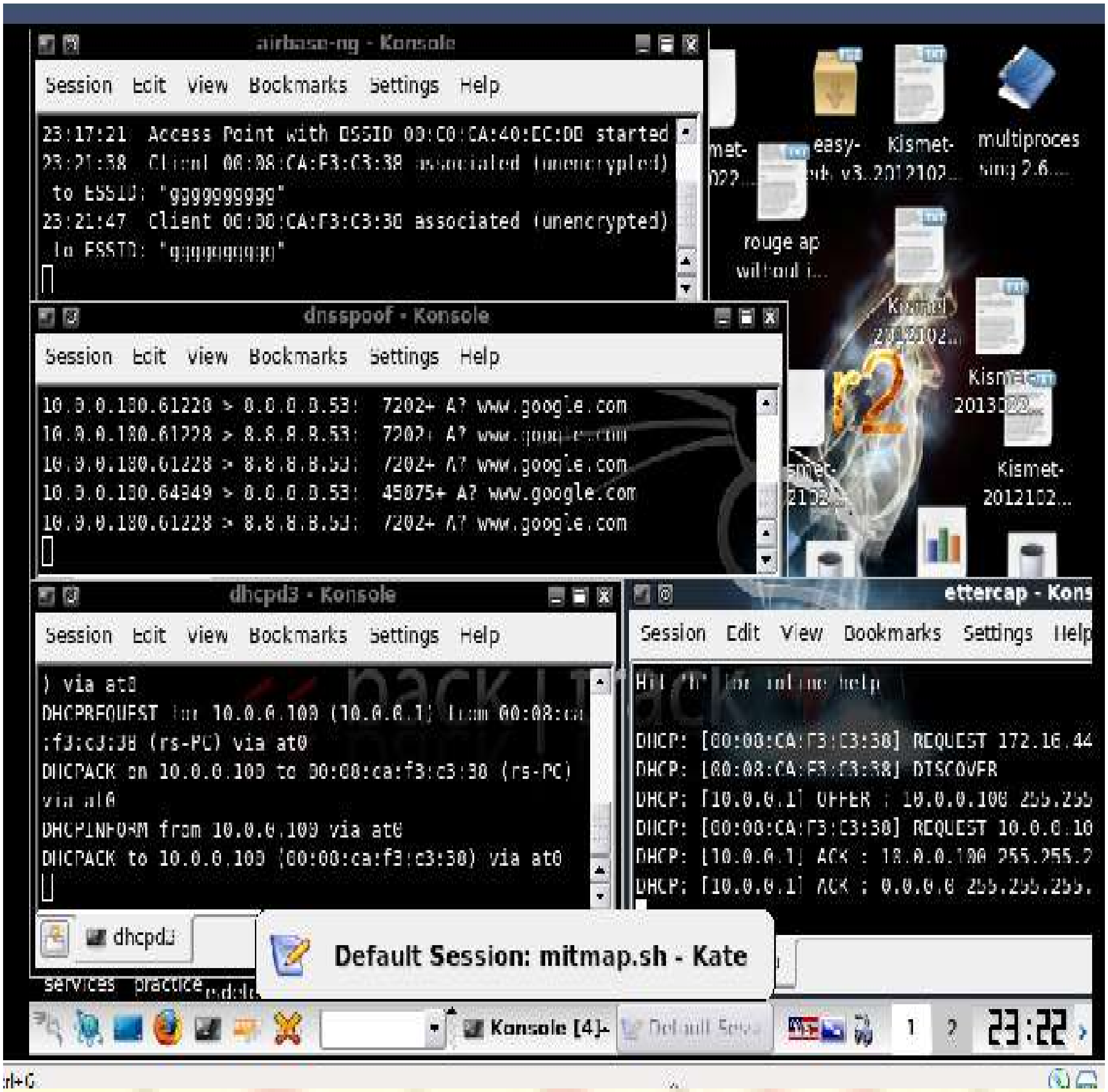
```
Text only Interface activated...
Hit 'h' for inline help
```

2)Victim connects and is assigned an IP address



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\rs>ipconfig

Windows IP Configuration


Wireless LAN adapter Wireless Network Connection 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wireless Network Connection:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::c8b8:b1b1:ef51:5e4%17
   IPv4 Address. . . . . . . . . . . : 10.0.0.100
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.0.1

Ethernet adapter Local Area Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

3) Client With Mac Id 00:08:ca:f3:c3:38 Associates With Our Access Point And Is Assigned A Ip Address Of 10.0.0.100



```
23:17:21  Access Point with BSSID 00:C0:CA:40:EC:DD started
23:21:38  Client 00:08:CA:F3:C3:38 associated (unencrypted)
 to ESSID: "gggggggggg"
23:21:47  Client 00:00:CA:F3:C3:30 associated (unencrypted)
 to ESSID: "gggggggggg"
```

```
10.0.0.100.61228 > 8.8.8.8.53:  7202+ A? www.google.com
10.0.0.100.61228 > 8.8.8.8.53:  7202+ A? www.google.com
10.0.0.100.61228 > 8.8.8.8.53:  7202+ A? www.google.com
10.0.0.100.64349 > 8.8.8.8.53:  45875+ A? www.google.com
10.0.0.100.61228 > 8.8.8.8.53:  7202+ A? www.google.com
```

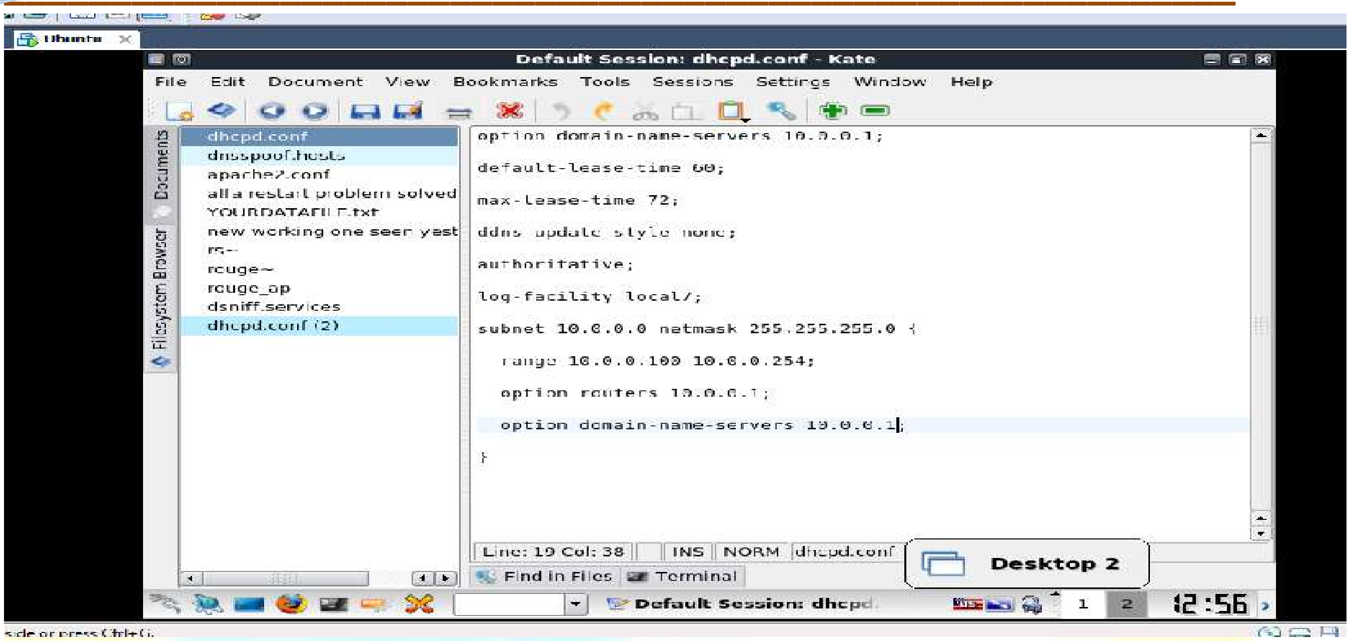
```
) via at0
DHCPREQUEST for 10.0.0.100 (10.0.0.1) from 00:08:ca
:f3:c3:38 (rs-PC) via at0
DHCPACK on 10.0.0.100 to 00:08:ca:f3:c3:38 (rs-PC)
via at0
DHCPINFORM from 10.0.0.100 via at0
DHCPACK to 10.0.0.100 (00:08:ca:f3:c3:38) via at0
```

```
Hit 'h' for inline help

DHCP: [00:00:CA:F3:C3:30] REQUEST 172.16.44
DHCP: [00:08:CA:F3:C3:38] DISCOVER
DHCP: [10.0.0.1] OFFER : 10.0.0.100 255.255
DHCP: [00:00:CA:F3:C3:30] REQUEST 10.0.0.10
DHCP: [10.0.0.1] ACK : 10.0.0.100 255.255.2
DHCP: [10.0.0.1] ACK : 0.0.0.0 255.255.255.
```

4)DHCP configuration file

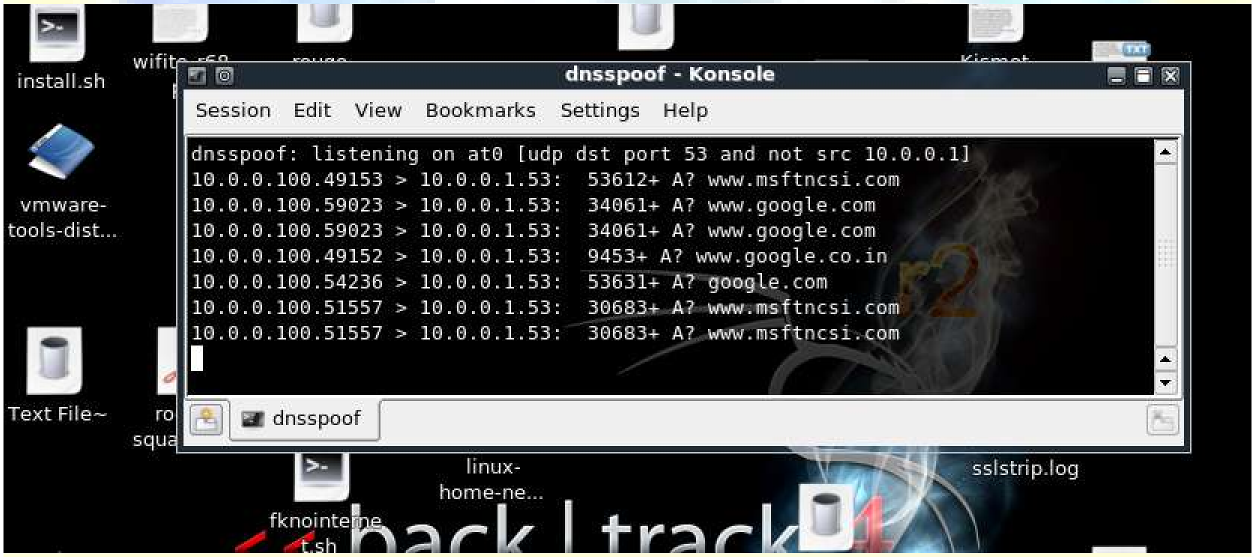
We need dhcp to assign ip addresses to the clients connecting to the fake ap.Here "***subnet 10.0.0.0 netmask 255.255.255.0***" specifies the subnet all the hosts will belong to. eg here it means if we want to assign IP addresses from the subnet 10.0.0.0 with a mask of 255.255.255.0 .

"range 10.0.0.1 10.0.0.254" specifies a range of IP addresses within the subnet that will be assigned to any machine requesting an address.



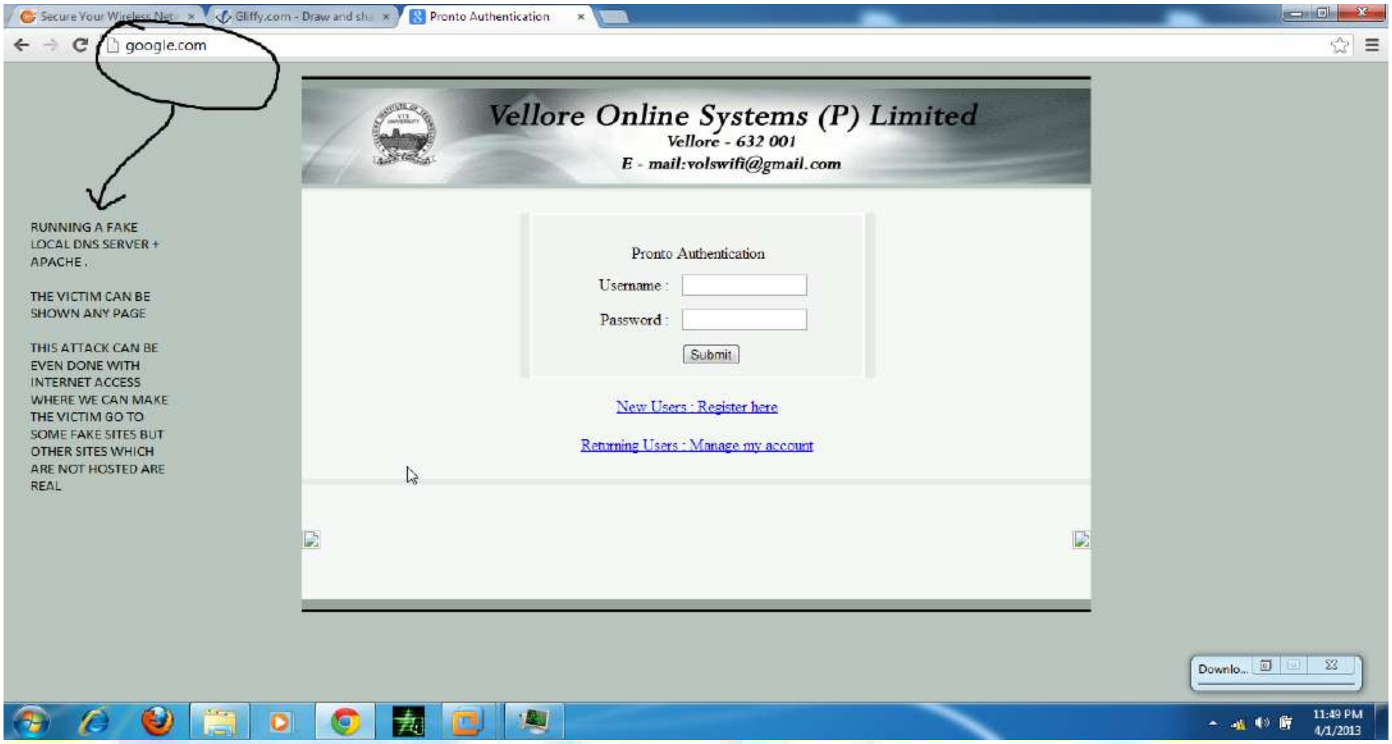
```
option domain-name-servers 10.0.0.1;

default-lease-time 60;

max-lease-time 72;

ddns-update-style none;

authoritative;

log-facility local7;

subnet 10.0.0.0 netmask 255.255.255.0 {

  range 10.0.0.100 10.0.0.254;

  option routers 10.0.0.1;

  option domain-name-servers 10.0.0.1;

}
```

5)Using Dnsspoof to resolve “google.com” to site that is running on our local server.



```
dnsspoof: listening on at0 [udp dst port 53 and not src 10.0.0.1]
10.0.0.100.49153 > 10.0.0.1.53:  53612+ A? www.msftncsi.com
10.0.0.100.59023 > 10.0.0.1.53:  34061+ A? www.google.com
10.0.0.100.59023 > 10.0.0.1.53:  34061+ A? www.google.com
10.0.0.100.49152 > 10.0.0.1.53:  9453+ A? www.google.co.in
10.0.0.100.54236 > 10.0.0.1.53:  53631+ A? google.com
10.0.0.100.51557 > 10.0.0.1.53:  30683+ A? www.msftncsi.com
10.0.0.100.51557 > 10.0.0.1.53:  30683+ A? www.msftncsi.com
```

6)Fake Page Hosted on our Local Apache Server



Vellore Online Systems (P) Limited
Vellore - 632 001
E - mail:volswifi@gmail.com

Pronto Authentication

Username :

Password :

Submit

New Users : Register here

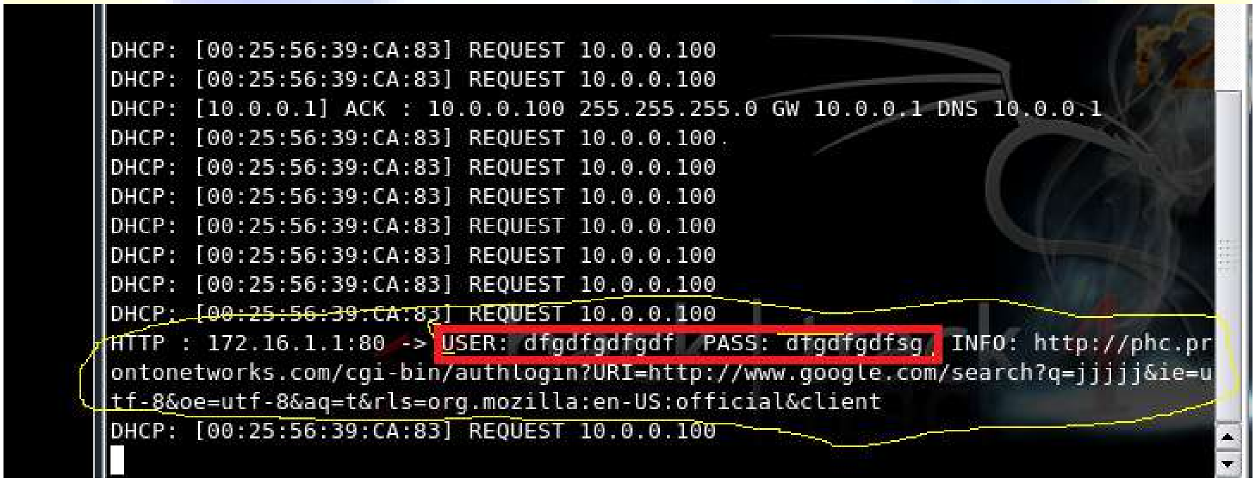
Returning Users : Manage my account

RUNNING A FAKE LOCAL DNS SERVER + APACHE.

THE VICTIM CAN BE SHOWN ANY PAGE

THIS ATTACK CAN BE EVEN DONE WITH INTERNET ACCESS WHERE WE CAN MAKE THE VICTIM GO TO SOME FAKE SITES BUT OTHER SITES WHICH ARE NOT HOSTED ARE REAL

7)Using script to do MITM Attacks using ETTERCAP .



```
DHCP: [00:25:56:39:CA:83] REQUEST 10.0.0.100
DHCP: [00:25:56:39:CA:83] REQUEST 10.0.0.100
DHCP: [10.0.0.1] ACK : 10.0.0.100 255.255.255.0 GW 10.0.0.1 DNS 10.0.0.1
DHCP: [00:25:56:39:CA:83] REQUEST 10.0.0.100
DHCP: [00:25:56:39:CA:83] REQUEST 10.0.0.100
DHCP: [00:25:56:39:CA:83] REQUEST 10.0.0.100
DHCP: [00:25:56:39:CA:83] REQUEST 10.0.0.100
DHCP: [00:25:56:39:CA:83] REQUEST 10.0.0.100
DHCP: [00:25:56:39:CA:83] REQUEST 10.0.0.100
DHCP: [00:25:56:39:CA:83] REQUEST 10.0.0.100
HTTP : 172.16.1.1:80 -> USER: dfgdfgdfgdf  PASS: dfgdfgdfsg  INFO: http://phc.pr
ontonetworks.com/cgi-bin/authlogin?URI=http://www.google.com/search?q=jjjjj&ie=u
tf-8&oe=utf-8&aq=t&rls=org.mozilla:en-US:official&client
DHCP: [00:25:56:39:CA:83] REQUEST 10.0.0.100
```

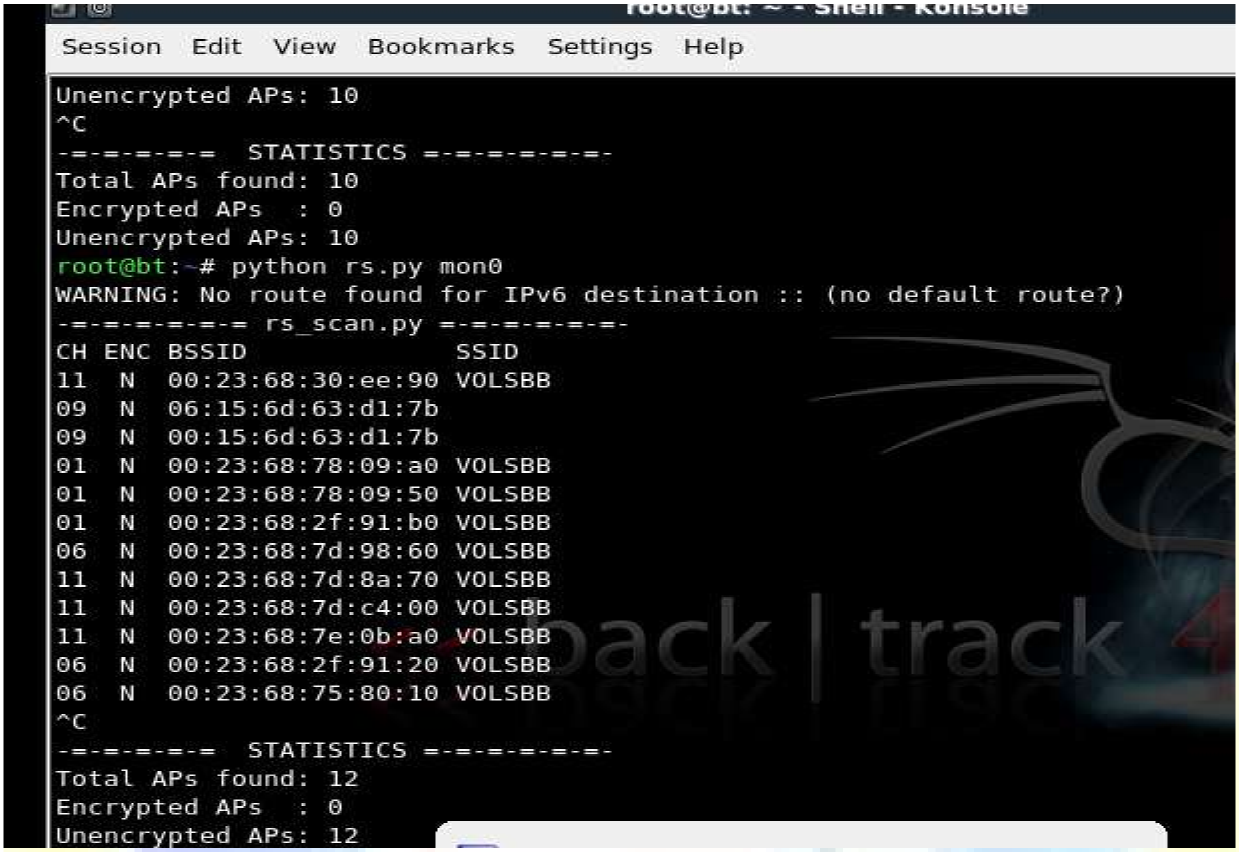
V . Detection Scheme

Pseudocode of Detecting Fake AP

```
Step 1.0 Initiate the setup for Fake AP detection.

Step 1.1 Scan wireless spectrum for MAC ID and SSID.

Step 2.1 Perform the check for wireless card for monitor mode support:

            If ( Monitor mode support is found ) do

      {

      Step 2.1.1 Spoof the MAC ID of the wireless card.

      Step 2.1.2 Put the card in monitor mode.

      Step 2.1.3 Sniff packets and discover network access points.

     Step 2.1.4 Discover hidden access points from data extracted from beacon  frames.

      Step 2.1.5 Collect ESSIDs ,MAC IDs ,security bit information and store them for  later use.

      Step 2.1.6 Perform ongoing intrusion detection using harvested information.


      }

else

{

 The wireless card does not support monitor mode hence detection of fake access point  is not supported by the hardware.

ReScan or Perform step 1.1.

}
```

For detection of Fake APs we have designed a wireless scanner that monitors radio frequencies repeatedly and reports if a Fake AP is found.

Results of Wifi Scanning


root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

```
Unencrypted APs: 10
^C
-=-=-=-=-=  STATISTICS =-=-=-=-=-=-
Total APs found: 10
Encrypted APs  : 0
Unencrypted APs: 10
root@bt:~# python rs.py mon0
WARNING: No route found for IPv6 destination :: (no default route?)
-=-=-=-=-=-= rs_scan.py =-=-=-=-=-=-
CH ENC BSSID             SSID
11  N  00:23:68:30:ee:90 VOLSBB
09  N  06:15:6d:63:d1:7b
09  N  00:15:6d:63:d1:7b
01  N  00:23:68:78:09:a0 VOLSBB
01  N  00:23:68:78:09:50 VOLSBB
01  N  00:23:68:2f:91:b0 VOLSBB
06  N  00:23:68:7d:98:60 VOLSBB
11  N  00:23:68:7d:8a:70 VOLSBB
11  N  00:23:68:7d:c4:00 VOLSBB
11  N  00:23:68:7e:0b:a0 VOLSBB
06  N  00:23:68:2f:91:20 VOLSBB
06  N  00:23:68:75:80:10 VOLSBB
^C
-=-=-=-=-=  STATISTICS =-=-=-=-=-=-
Total APs found: 12
Encrypted APs  : 0
Unencrypted APs: 12
```

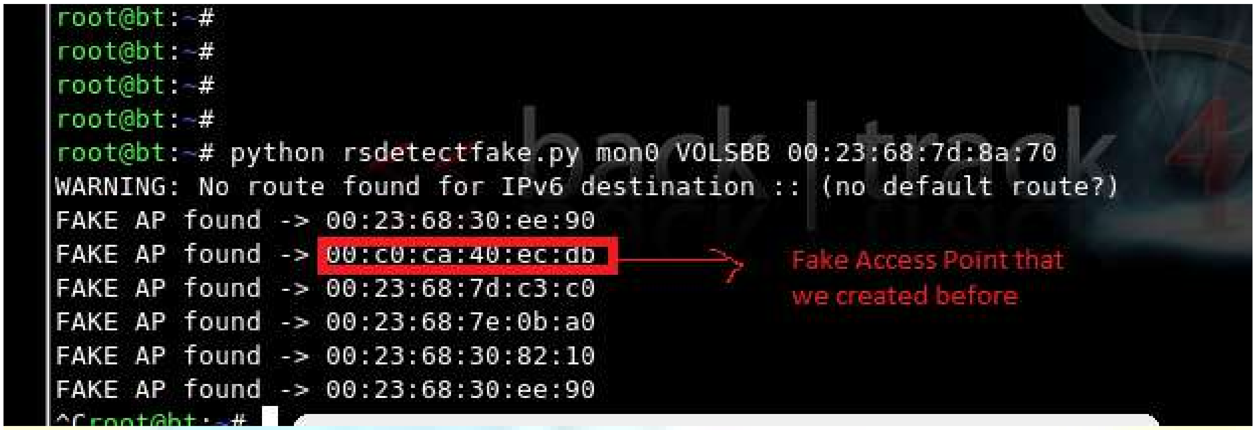
We have successfully tested the detection of Fake AP under two different scenarios.

Discovering fake access points by MAC ID filteration(results found in fig. 5.1)

A fake access point started with the same SSID as a corporate network poses a threat to the network. We can detect this by capturing packets and comparing their MAC addresses with authorized MAC addresses. The script takes an authorized access point with its base address and SSID and continues sniffing for beacon packets, while looking for the same SSID with a different base MAC. The script reports any and all access points located.

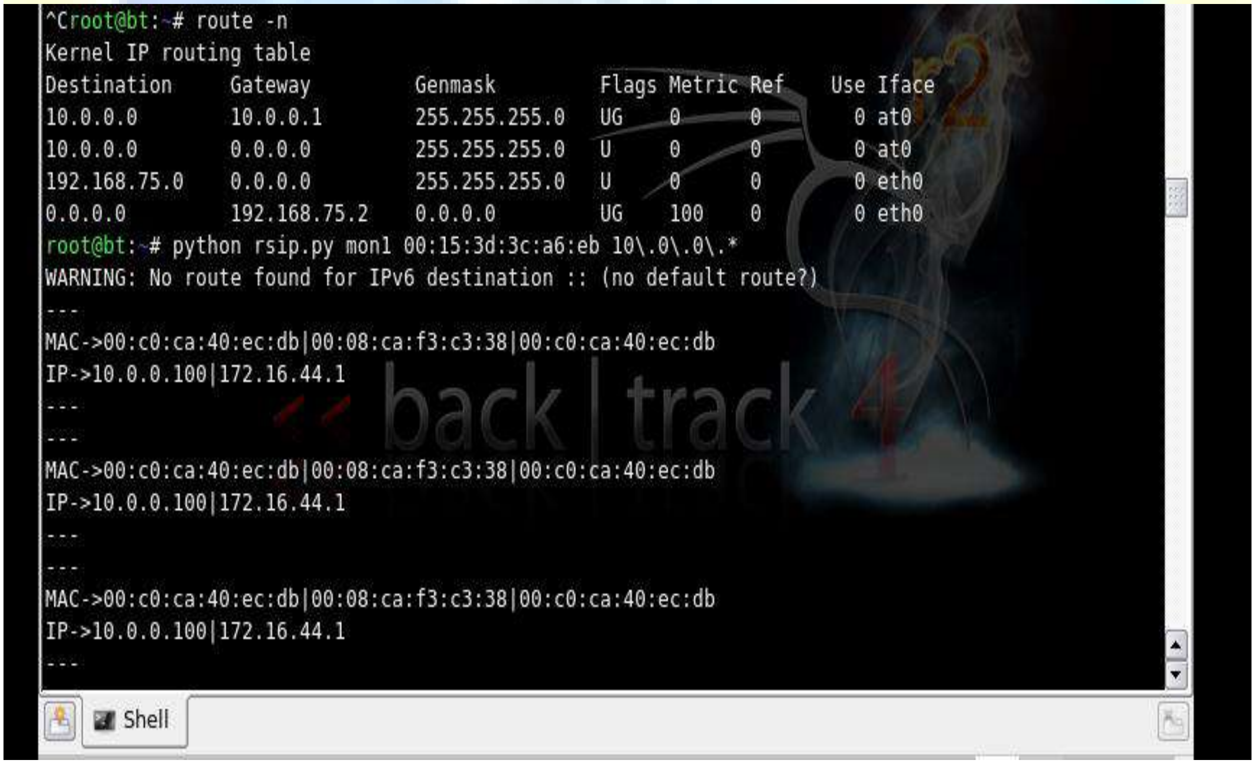
Discovering Rogue Access Points by local network usage(results found in fig. 5.2) If an unauthorized access point is deployed on your network, an administrator can find the traffic and exclude it from the network by capturing beacon packets or analyzing wireless IP traffic. For example, suppose your network is 10.0.0.1 and it consists of one access point with an address of 00:15:3d:3c:a6:eb. The objective is to track down any surrounding access points, other than this access point, that are accessing the 10.0.0.1 network.Here we make a access point named gggggggg .

The script captures all packets from the air and dissects the IP layer. A decision-making point is the source and destination IP address for the packets. If these packets are not part of an authorized access point defined by the MAC address, then they are reported. This can be a potential access point running on 10.0.0.1 We can verify its existence and traffic from the wire side once you notice intrusion to reduce false positives



```
root@bt:~#
root@bt:~#
root@bt:~#
root@bt:~#
root@bt:~# python rsdetectfake.py mon0 VOLSBB 00:23:68:7d:8a:70
WARNING: No route found for IPv6 destination :: (no default route?)
FAKE AP found -> 00:23:68:30:ee:90
FAKE AP found -> 00:c0:ca:40:ec:db
FAKE AP found -> 00:23:68:7d:c3:c0
FAKE AP found -> 00:23:68:7e:0b:a0
FAKE AP found -> 00:23:68:30:82:10
FAKE AP found -> 00:23:68:30:ee:90
^Croot@bt:~#
```

Fig. 5.1



```
^Croot@bt:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        10.0.0.1        255.255.255.0   UG    0      0        0 at0
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 at0
192.168.75.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.75.2    0.0.0.0         UG    100    0        0 eth0
root@bt:~# python rsip.py mon1 00:15:3d:3c:a6:eb 10\.0\.0\.*
WARNING: No route found for IPv6 destination :: (no default route?)
---
MAC->00:c0:ca:40:ec:db|00:08:ca:f3:c3:38|00:c0:ca:40:ec:db
IP->10.0.0.100|172.16.44.1
---
---
MAC->00:c0:ca:40:ec:db|00:08:ca:f3:c3:38|00:c0:ca:40:ec:db
IP->10.0.0.100|172.16.44.1
---
---
MAC->00:c0:ca:40:ec:db|00:08:ca:f3:c3:38|00:c0:ca:40:ec:db
IP->10.0.0.100|172.16.44.1
---
```

Fig. 5.2

VI . Testing Environment and Tools Used

Testing Environment :

Two Laptops containing Intel ® core TM 2 Duo CPU

RAM (Memory): - 2GB (Recommended) - 4GB

Network adaptors: - Alfa Aw series, Broadcom wifi

Platforms: - Linux, Ubuntu / Back track Linux 5.

Tools Used

All the tools that we have used are open source.

1)Aircrack-NG suite:

Aircrack-ng is a set of tools for auditing wireless networks.Aircrack-ng is an 802.11 WEP and WPA-PSK keys cracking program that can recover keys once enough data packets have been captured.Here we will be using it for putting card in monitor mode and for starting the fake access point.

2)Python-Scapy:

Scapy library is built in python language which is used for sniffing and dissecting 802.11 wireless frames. Scapy library enables user to send, manipulate and inject the wireless packets according to the requirements. It is a very powerful tool used for packet manipulation, and it is used to decode and fake packets of large number of protocols. There are many functions that can be achieved by using scapy. It can send and capture packets and match the request sent by the wireless devices. Scapy also performs scanning tasks especially passive scanning and unit testing using probe request. Attacking can also be done with network discovery. Scapy can remove and replace any part of the wireless network packet that can be used for war driving mechanisms like hping, arpspoof, arping, etc. As compared with other tools, scapy can easily handle other functionalities like injecting 802.11 frames and combing other techniques to complete a particular task. The two major tasks scapy usually performs are sending the packets and receiving their answers. It can also let you to define your own 802.11 frames and inject that in wireless network. It sends them in the network and waits for the answers from other network devices and then it matches the request with answers, finally returning back a list of packets.

3)Python 2.6

4)A wireless card that can go in monitor model(We are using Alfa AWUS036H for our prototype model).

5)Scapy dependencies-Other libraries like multiprocessor.py etc. which are not directly included in the python package.

6)ETTERCAP:

Ettercap is a comprehensive suite for man in the middle attacks. It features sniffing of live connections, content filtering on the fly and many other interesting tricks. It supports active and passive dissection of many protocols and includes many features for network and host analysis.

7)dnsspoof: dnsspoof forges replies to arbitrary DNS address / pointer queries on the LAN. This is useful in bypassing hostname-based access controls, or in implementing a variety of man-in-the-middle attacks.

8)Dhcp-3 server Ubuntu

9)Apache (LAMP server)

VII . Conclusions And Future Enhancements

Creation of Rogue Access Points poses threat to wireless networks.Though it is hard to get a victim to connect to your access point but this problem is solved by doing a DoS on the client machine to disassociate them from the AP.

In this project we have successfully identified various wireless vulnerabilities and security threats for the end users and finding solution to combat them.

We have successfully configured , created and installed a Fake Access Point in the network. We have successfully simulated Man in The Middle Attack and provided a solution to successfully diagnose the problem. Future work planned is to enhance the features for developing and implementing a complete framework that is able to all types of man in the middle attacks using fake access points and we would also like to add an independent location finding techniques to track down the location of the wireless devices with other parameters . We plan to include Timestamp or Clock Skews approach to detect Fake APs in the future. We plan to make a complete Intrusion Detect System(IDS) by implementing more detection programs for different attacks available on WLAN.

VIII . Acknowledgement

I have put in great efforts to successfully complete this project. However, it would not have been a success without the support and guidance rendered by the faculties and the University. I would like to thank Prof Harsh Arora, SCSE, VIT University for guiding me throughout the project. His guidelines and suggestions were of great help in achieving my milestones and deliverables.

IX . List of Abbreviations

AP	Access Point
BSS	Basic Service Set
BT	BackTrack Linux
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
MAC	Medium Access Control
NIC	Network Interface card
RF	Radio Frequency
RFMON	Radio Frequency Monitoring
SSID	Service Set Identifier
WEP	Wired Equivalency Protocol
WLAN	Wireless LAN
WNIC	Wireless Network Interface Card
PCMCIA	Personal Computer Memory Card International Association
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
MITM	Man In The Middle Attack

X . References

Standards

1. **IEEE Std. 829-1998 IEEE** *Standard for Software Test Documentation*
2. **IEEE Std. 830-1998 IEEE** *Recommended Practice for Software Requirements Specifications*
3. **IEEE Std. 1016-1998 IEEE** *Recommended Practice for Software Design Descriptions*
4. **IEEE Std. 802.11-1997 IEEE** *Also known as 802.11 legacy.Recommended standard for WLAN.*
5. **IEEE Std. 802.11b-1999 IEEE** *Is ammendement to WLAN standard 802.11-1997.*
6. **IEEE 802.11a-1999 or 802.11a** *Is an amendment to the 802.11 WLAN specifications. It was originally designed to support wireless communication in the unlicensed national information structure (U-NII) bands*

Journal Paper

[1] Gayathri Shivaraj, Min Song, Sachin Shetty , A Hidden Markov Model Based Approach to Detect Rogue Access Points , Department of Electrical and Computer Engineering, Old Dominion University .

[2] Sachin Shetty, Min Song , Rogue Access Point Detection by Analyzing Network Traffic Characteristics , Department of Electrical and Computer Engineering,

Old Dominion University

[3] Hirschmann Automation and Control GmbH , Rogue AP and Rogue Client Detection WLAN Access Point , 2010 .

[4] V. S. Shankar Sriram, G.Sahoo Krishna Kant Agrawal ,Detecting and Eliminating Rogue Access Points in IEEE-802.11 WLAN - A Multi-Agent Sourcing Methodology

[5] Jaemin Lee , Man-in-the-middle Attacks Detection Scheme on Smartphone using 3G network.

[6] Saurabh Vishal ,Scanned Wireless Network Setup Fake Access Point & its Detection

[7] Adam Maxwell (@catalyst256) , The Unofficial Guide to Scapy for Dummies

[8] Philippe Biondi and the Scapy community ,Scapy Documentation Release 2.1.0

[9] David Burreto , Using TCP fingerprinting for Fake AP detection

[10**]** S Vishal , Scanned Methods For Fake Access Points

[11] Matt Jacobs , Wifi Pineapple Mkv Sniffing Attacks

[12] T. Kim , Online Detection of Fake Access Points

Web Links:

[13] DHCP3-Server Retrieved from: https://help.ubuntu.com/community/dhcp3-server

[14] How things works: WLAN Technologies and security Mechanisms Retrieved from :

http://www.sans.org/reading_room/whitepapers/wireless/things-work-wlan-technologiessecurity-mechanisms_1301

[15]Backtrack Forums:http://www.backtrack-linux.org/forums/

[16]Wifi Planet : http://www.wi-fiplanet.com/tutorials/article.php/1564431

[17] Using nessus to discover rogue access points: http://blog.tenablesecurity.com/2009/08/using-nessus-to-discover-rogue-accesspoints.html

http://www.code.google.com/p/rsfakeap/