WI-FI NETWORK'S PASSWORD STRENGTH ANALYSIS USING BACKTRACK TOOLS

[1]P RAGHU RAM, [2]D SINDHURA, [3]OSMAN SHAREEF

M.Tech - Computer networks and information security, JNT University Hyderabad
[2]Software Engineer, Infosys, Hyderabad, [3]IT Analyst Serco Global Services
Email: [1]raghuram369@gmail.com, [2]sindhura919@gmail.com, [3]osmanshareef.contact@gmail.com

Abstract—Wireless networks are most common type of networks found in today`s communication. Advantage of these wireless networks (IEEE 802.11) is that physical connection is not required for being a part of a network and at a same time no physical access is required to crack and penetrate such a network. The three main security algorithms which are implemented for the security of this protocol (WIFI) are WEP, WPA and WPA2. In this paper we discuss an attack on the WIFI networks to crack its password and gain unauthorized access to the network. Aircrack tool which is a part of the Backtrack OS, is used to perform attack on the wireless network. Based on the time taken by the tool to crack the network we try to determine the strength of the password and describe a format for good password based on the results. After gaining access into a network we make use of another penetration testing tool, Metasploit to break a web server remotely.

Keywords— Aircrack, Backtrack, Password, Metasploit, WEP, WIFI, WPA2

I. INTRODUCTION

WI-FI is a very well prevalent technology that allows an electronic device to exchange data wirelessly using radio waves over a computer network, including high-speed Internet connections. The Wi-Fi alliance defines Wi-Fi as any "wireless local area network (WLAN) products that are based on the Institute of Electrical and Electronics Engineers' (IEEE) 802.11 standards".[1]
Wi-Fi can be less secure than wired connections such as the Ethernet as an intruder does not need a physical connection. Web pages that use SSL are secure but unencrypted internet access can easily be detected by intruders. In our ages, most of the providers give out protection. Unfortunately most of these Wi-Fi boxes apply WEP crypting by default if we activate the wireless.

It is likely known that this protection has passed deadline...

weak and easily crackable. A small hour is enough to crack a 128 bytes WEP key (packets capture + crack) and barely more for a 256 bytes key with Aircrack.
Because of this, Wi-Fi has adopted various encryption technologies. The early encryption was WEP, it was proved that it could be easily broken.
Higher quality protocols (WPA, WPA2) were added later. An optional feature added in 2007, called Wi-Fi Protected Setup (WPS) had a serious flaw that allowed an attacker to recover the router's password. There are two basic types of vulnerabilities associated with WLANs: those caused by poor configuration and those caused by poor encryption.
Poor configuration causes many vulnerabilities. Wireless networks are often put into use with no or insufficient security settings. With no security settings – the default configuration – access is obtained simply by association. With insufficient security settings as cloaking and/or MAC address filtering, security can be easily cracked.
Poor encryption causes the remaining vulnerabilities. Wired Equivalent Privacy (WEP) is defective and can be defeated in several ways. Wi-Fi Protected Access (WPA) and Cisco's Lightweight Extensible Authentication Protocol (LEAP) are vulnerable to dictionary attacks.
In this paper we will see how a tool can used for cracking the password of the WIFI network and how metasploit can be used to break into a web server in that network. The tool which can be used for this purpose is Aircrack-ng. The techniques used are discussed below.

II. AIRCRACK

Aircrack-ng is a tool which runs on Windows and Linux operating systems, and can be used to crack WEP and WPA-PSK keys. It makes use of Pychkine-Tews-Weinmann and KoreK attacks, both of which are statistical methods. These are more efficient than the traditional FMS attack. Aircrack-ng consists of several components. Airmon-ng configures the wireless network card. Airodump-ng captures the frames. Aireplay-ng generates traffic. Aircrack-ng does the cracking, using the data collected by Airodump-ng. Finally, Airdecap-ng decrypts all packets that were captured. Thus, Aircrack-ng is the name of the suite and also of one of the components. We will discuss WEP key cracking using Aircrack and backtrack terminal.

Every AP sends out about 10 so called beacon frames a second. These packets contain the following information:

- Name of the network (ESSID)
- If encryption is used (and what encryption is used; pay attention, that may not be always true just because the AP advertises it)
- What MBit data rates are supported
- Which channel the network is on

If we want to connect to a wireless network, there are some possibilities. In most cases, Open System Authentication is used.

Open System Authentication:
Ask the AP for authentication.

1. The AP answers: OK, you are authenticated.
2. Ask the AP for association
3. The AP answers: OK, you are now connected.

This is the simplest case, BUT there could be some problems if you are not legitimate to connect:

- WPA/WPA2 is in use, you need EAPOL authentication. The AP will deny you at step 2.
- Access Point has a list of allowed clients (MAC addresses), and it lets no one else connect. This is called MAC filtering.
- Access Point uses Shared Key Authentication, you need to supply the correct WEP key to be able to connect.

III. CRACKING WEP KEYS

Prior to looking for networks, we must put our wireless card into “monitor mode”. Monitor mode is a special mode that allows our PC to listen to every wireless packet. This monitor mode also allows us to optionally inject packets into a network.

Aircrack provides us with a wealth of options:

- -a [mode 1 or 2] 1=WEP, 2=WPA-PSK
- -e [essid] target selection network ID
- -b [bssid] target access point's MAC
- -q enable quiet mode
- -w [path] path to a dictionary word list (WPA only)
- -n [no. bits] WEP key length (64, 128, 152 or 256)
- -f [fudge no.] defaults are 5 for 64 bit WEP and 2 for 128 bit WEP

To put your wireless card into monitor mode:
airmon-ng start

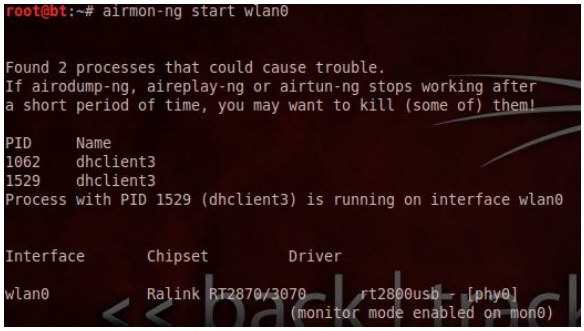


Fig 1: monitor mode for Wi-Fi adapter

Then, start airodump-ng to look out for networks:
Airodump-ng <interface name> airodump-ng hops from channel to channel and shows all access points it can receive beacons from. Channels 1 to 14 are used for 802.11b and g (in US, they only are allowed to use 1 to 11; 1 to 13 in Europe with some special cases; 1-14 in Japan). Channels between 36 and 149 are used for 802.11a. The current channel is shown in the top left corner.
After a short time some APs and (hopefully) some associated clients will show up. Because of the channel hopping we won't capture all packets from your target net. So we want to listen just on one channel and additionally write all data to disk to be able to use it for cracking.

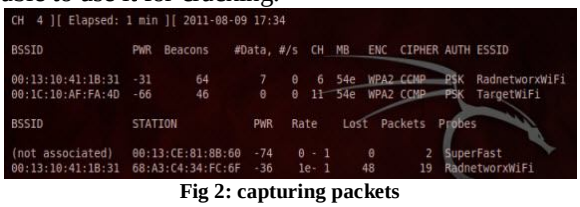


Fig 2: capturing packets

airodump-ng -c 11 --bssid 00:01:02:03:04:05 -w dump <interface name>
With the -c parameter you tune to a channel and the parameter after -w is the prefix to the network dumps written to disk. The”--bssid” combined with the AP MAC address limits the capture to the one AP. The ”--bssid” option is only available on new versions of airodump-ng. Before being able to crack WEP we will usually need between 40 000 and 85 000 different Initialization Vectors (IVs). Every data packet contains an IV. IVs can be re-used, so the number of different IVs is usually a bit lower than the number of data packets captured.
So we will have to wait and capture 40K to 85K of data packets (IVs). If the network is not busy it will take a very long time. Often you can speed it up a lot by using an active attack (=packet replay). [2]

aireplay-ng -1 0 -a (bssid) -h 00:11:22:33:44:55 -e (essid) (interface)
aireplay-ng -3 -b (bssid) -h 00:11:22:33:44:55 (interface)

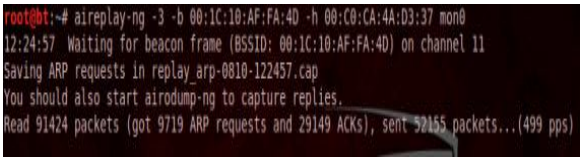


Fig 3: Capturing ARP requests

Here we're creating router traffic to capture more throughput faster to speed up our crack.
aircrack-ng -b (bssid) (file name-01.cap).
If we do not get enough data, Aircrack will fail and tell you to try again with more. If it succeeds, the WEP key appears on the screen.

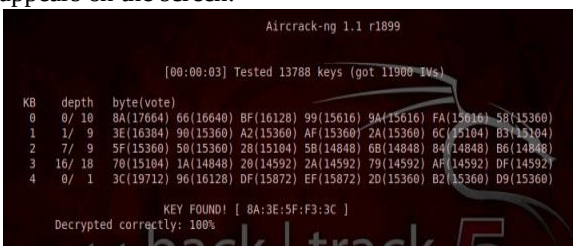


Fig 4: Password retrieval

WPA and WPA2 keys can also be recovered making use of the packet injection technique and dictionary attacks. Discussing WPA and WPA2 cracking is beyond the scope of this paper and hence is avoided.[3]

IV. WORKING OF AIR CRACK

Multiple techniques are combined to crack the WEP key:

- FMS (Fluhrer, Mantin, Shamir) attacks - statistical techniques
- Korek attacks - statistical techniques
- Brute force

When using statistical techniques to crack a WEP key, each byte of the key is essentially handled individually. Using statistical mathematics, the possibility that a certain byte in the key is correctly guessed goes up to as much as 15% when the right initialization vector (IV) is captured for a particular key byte. Essentially, certain IVs "leak" the secret WEP key for particular key bytes. This is the fundamental basis of the statistical techniques. [5]
By using a series of statistical tests called the FMS and Korek attacks, votes are accumulated for likely keys foreach key byte of the secret WEP key. Different attacks have a different number of votes associated with them since the probability of each attack yielding the right answer varies mathematically. The more votes a particular potential key value accumulates, the more likely it is to be correct. For each key byte, the screen shows the likely secret key and the number of votes it has accumulated so far. Needless to say, the secret key with the largest number of votes is most likely correct but is not guaranteed. Aircrack-ng will subsequently test the key to confirm it. If we use a fudge factor 2, it takes the votes of the most possible byte, and checks all other possibilities which are at least half as possible as this one on a brute force basis. The larger the fudge factor, the more possibilities aircrack-ng will try on a brute force basis. As the fudge factor gets larger, the number of secret keys to try goes up tremendously and consequently the elapsed time also increases. Therefore with more available data, the need to brute force, which is very CPU and time intensive, can be minimized.
For cracking WEP keys, a dictionary method is also included. For WEP, you may use either the statistical method described above or the dictionary method, not both at the same time. With the dictionary method, you first create a file with either ASCII or hexadecimal keys. A single file can only contain one type, not a mix of both. This is then used as input to aircrack-ng and the program tests each key to determine if it is correct. WEP keys can be entered in hexadecimal or ASCII. The following table describes how many characters of each type is required in our files

WEP key length in bits	Hexadecimal Characters	Ascii Characters
64	10	5
128	26	13
152	32	16
256	58	29

The overriding technique is capture as much data as possible. That is the single most important task. The number of initialization vectors (IVs) that you need to determine the WEP key varies dramatically by key length and access point. Typically we need 250,000 or more unique IVs for 64 bit keys and 1.5 million or more for 128 bit keys. Clearly a lot more for longer key bit lengths. Then there is luck. There will be times that the WEP key can be determined with as few as 50,000 IVs although this is rare. Conversely, there will be times when we will need multiple millions of IVs to crack the WEP key. The number of IVs is extremely hard to predict since some access points are very good at eliminating IVs that lead the WEP key.
If aircrack-ng determines the key, it is presented to you in hexadecimal format. The length will vary based on the WEP bit key length used. See the table above which indicates the number of hexadecimal characters for the various WEP key bit lengths.
Airodump-ng is used for packet capturing of raw 802.11 frames and is particularly suitable for collecting WEP IVs (Initialization Vector) for the intent of using them with aircrack-ng. If you have a GPS receiver connected to the computer, airodump-ng is capable of logging the coordinates of the found access points.

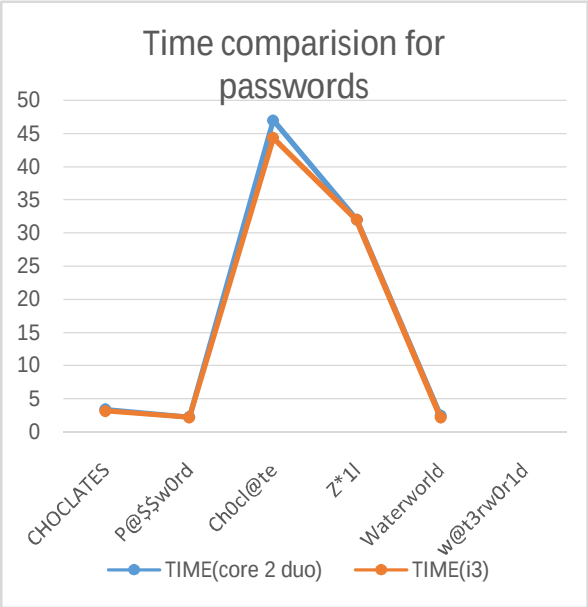
V. PASSWORD CRACKING TIMES

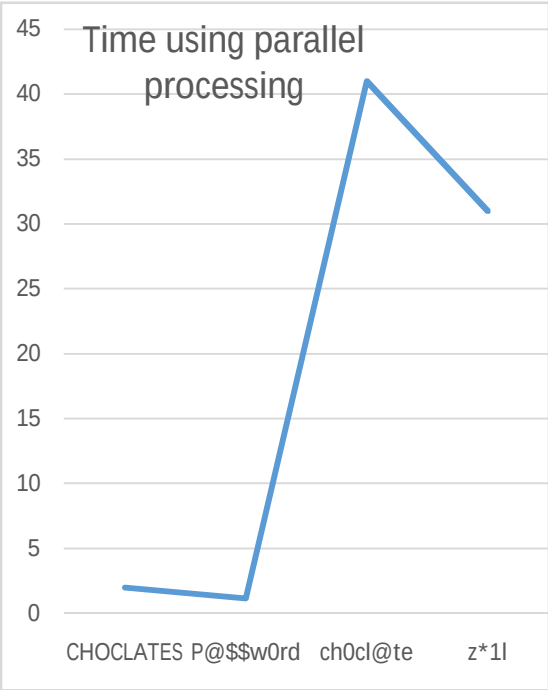
Using the above method, certain passwords are assigned to Wi-Fi routers and we tried to crack them. Passwords chosen were random and used to test

different formats of passwords. This observation is done using different processors to test the dependency on processing capabilities of a pc.

Password	Time(core 2 duo)	Time(i3)
CHOCLATES	3min 42 secs	3min 2 secs
P@$$w0rd	2min 23 secs	2min 20 secs
Ch0cl@te	47mins	44mins 32 secs
Z*1l	32mins	32mins
Waterworld	2mins 45 secs	2mins 21 secs
w@t3rw0r1d	Unable to crack	Unable to crack

Table 1: Comparison of time taken to crack passwords using different processors





Passwords mentioned in the above table are completely random. Based on the above observation we can make certain analysis. Cracking time is certainly dependent on the processing capability of the system. Dictionary words like 'CHOCLATES' took least time as they are cracked by dictionary attacks which is the most common attach. The same word with a minute change in its letters took a whole lot of time as it is not a dictionary word. Another observation which can be made from the above table is that cracking time is no way dependent on the length of the password. Dictionary word like CHOCLATES which is 9 letters in length took about 4mins while "z*1l"which is 4 letters long took 32mins.

Another experiment is conducted to check the crack ability of a password based on the distance of attacking machine in our case, back track machine from the access point. The observation is shown in table below.

Password	TIME(dist-10m)	TIME(dist-25m)
CHOCLATES	3min 42 secs	5min 39 secs
P@$$w0rd	2min 23 secs	4mins 21secs
Ch0cl@te	47mins	Unable to crack
Z*1l	32mins	Unable to crack

Table 2: Comparison between the times based on the distance between AP and attacker

Certain observations can be made from the above table. Cracking times depend to a great extent on the distance between attacking machine and the access point. This is because the attacking machine needs to capture the packets flowing through the access points and then need to inject requests. The closer the machine is to the access point easier it is to inject ARP requests.

VI. PASSWORD CRACKING USING PARALLEL COMPUTING

Above mentioned experiments are now conducted by making a network of 3 computers all with Intel core 2 duo processor, ram of 2gb. This is to ensure compatibility between the machines. Now the cracking is done by splitting the task between the three machines. In case of dictionary attack, the task is easily done as the words are split between the machines for comparison and also since the processing capability is also improved non dictionary attacks also should take comparatively less time.

PASSWORD	TIME
CHOCLATES	2mins
P@$$w0rd	1min 18 secs
Ch0cl@te	41mins
Z*1l	31mins

Table 3: Time taken to crack password using parallel computing

We can observe that dictionary words and common words like p@$$w0rd took comparatively less times but there is not much of a difference between the cracking times of non-dictionary words as it mainly depends on the strength of ARP poisoning.

CONCLUSION

Wi-Fi network is most popular in today's communication world. Wi-Fi popularity has led to the question of its security. There are many exploits that can be performed once an attacker is inside our network. Hence it is essential toprevent attackers from gaining access to the wireless network. One way entry into a Wi-Fi network is through access point`s password. Hence the strength of such a password plays a crucial role.

Sixteen characters of password is more than sufficient, if they are randomly generated using a cryptographic-function. If we use lower-case, upper-case, and digits, and if we generate it randomly, then a 16-character password has 95 bits of entropy. That is more than sufficient. 12 characters should be sufficient as it gives us 71 bits of entropy for security against all of the attacks that attackers might try to attack our password.Once the password is 12 characters or longer, the password is extremely unlikely to be the weakest link in our system. Therefore, there's not much point choosing a longer password. [4]

Along with the password its usability is also very important. From Table 1 we have seen that short password strings making use of complex letters are difficult to crack when compared to the long dictionary passwords. If we make the security mechanism too hard to use, people will get annoyed and may be more reluctant to use it in the future, which isn't good. Hence any password which is 12 characters in length and is randomly generated should be good enough as we have seen that few passwords couldn't be cracked using the mentioned method.

REFERENCES

[1] Practical attacks against WEP and WPA - Martin Beck and Erik Tews(http://dl.aircrack-ng.org/breakingwepandwpa.pdf)
[2] Using Fluhrer, Mantin, and Shamir Attack to Break WEP-Stubblefield, A. Ioannidis, J. and Rubin (http://download.aircrack-ng.org/wiki-files/doc/using_FMS_attack.pdf)
[3] B. Schneier. Applied Cryptography - Protocols, Algo-rithms, and Source Code in C . John Wiley & Sons, Inc., 1994
[4] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. In Eighth Annual Workshop on Selected Areas in Cryptography , Toronto, Canada, Aug. 01.http://download.aircrack-ng.org/wiki-files/doc/aircrack_reverse_engineer.pdf
[5] W. A. Arbaugh. An Inductive Chosen Plaintext Attack Against WEP and WEP2, 2001

★★★