Wi-Fi Protected Access (WPA) –PSK (Phase Shift Keying) Key Cracking Using AIRCRACK-NG

Sheikh Md. Rabiul Islam

Abstract: The IEEE 802.11i standard formally replaces Wired Equivalent Privacy (WEP) and the other security features of the original IEEE 802.11 standard. The IEEE 802.1x Port-Based Network Access Control standard is an optional method for authenticating 802.11 wireless clients. In this paper used the Aircrack-ng software for cracking the WPA pre-shared keys (PSK).The experiment with a simple ASCII keys and a complex hexa-decimal keys to check if the keys could be cracked.

Keywords: WPA, WEP, TKIP, ASCII, PTK, GTK, PSK.

1. Introduction

Wi-Fi Protected Access (WPA) is an interim standard adopted by the Wi-Fi Alliance to provide more secure encryption and data integrity while the IEEE 802.11i standard was being ratified. The WPA supports authentication through 802.1x (known as WPA Enterprise) or with a pre-shared key (known as WPA Personal), a new encryption algorithm known as the Temporal Key Integrity Protocol (TKIP). Aircrack-ng is software which is used to crack the data which has been captured and to perform this cracking; the attacker should get a four way handshake.
There are two variants to this protocol

- ENTERPRISE: Requires an IEEE 802.1x authentication server, which will distribute different keys to different users and different keys to the same user.
  PERSONAL: WPA provides security to the SOHO (small office/home office) user. WPA uses the pre-shared key method to create the PMK used to initialize the TKIP (Temporal Key Integrity Protocol) encryption process. The pre-shared key is previously configured in the access point and all nodes. The passphrase may be from 8 to 63 printable ASCII characters or 64 hexadecimal digits (256 bits) .In PSK mode, security depends on the strength and privacy of the passphrase: the weak passphrase users typically select are vulnerable to password cracking attacks (off line attack based on dictionary).

Sheikh Md. Rabiul Islam, is currently pursuing Ph.D. degree program in Information science & Engineering under Faculty of Education, Science, Engineering & Technology in University of Canberra, Australia, E-mail: Sheikh.Islam@canberra.edu.au

2. The Baseline Experiment Scenario

In the diagram below, brief explanation of the practical work setup is explained. Two clients are used in this experiment in which client machine 1 running "*airodump*" software in the monitor mode and the client machine 2 is connected to the network having SSID: "NCG" [1].

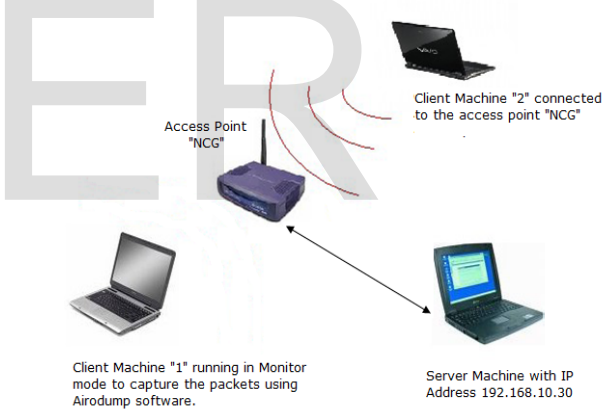


Figure 1: System Setup showing two machines connected to Access Point "NCG"

For performing the WPA cracking, first the CISCO access point should be configured to support the WPA-PSK. This can be done by setting the following parameters: In the Client Authentication Key Management section, select the Key Management as Mandatory from the list and the checkbox for WPA should be selected. In the WPA Pre-Shared Key section, It can provide the string as an ASCII character or Hexa-Decimal [1].

3. The Four Way Handshake Process

Every client/station connects to the access point; they exchange the Extensible Authentication Protocol over LAN

(EAPOL) key to verify the identities. After the authentication is over, the access point starts the four way handshake.

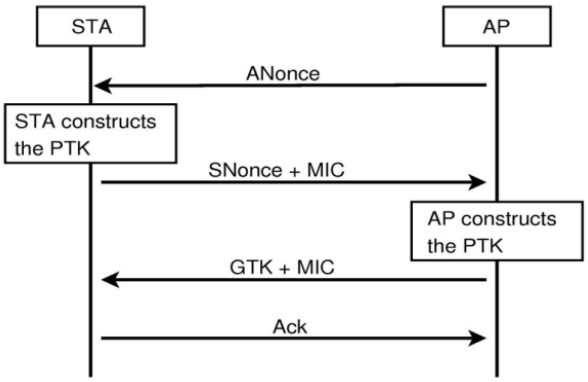


Figure 2: The schematic figure showing the four way handshake.

The access point (AP) initiates the conversation by selecting the random number ANonce to the client. The client calculates the PTK, and derives temporal keys, and then it sends SNonce and the MIC (Message Integrity Code) as a response to the access point. Upon receiving the response message, the access point calculates the PTK and derived temporal keys. Then the access point sends the GTK (Group Transient Key) which is in the encrypted form, along with the MIC. When the client receives this message, the MIC is checked to ensure that the access point knows the PMK and has correctly calculated the Pair wise Transient Key and the derived temporal keys. If the client confirms that the calculated PTK and derived temporal key is correct, it acknowledges back (confirmation message) to the access point which finishes the whole process of handshake.

After getting the four way handshaking, can use the software Aircrack-ng, which is used to crack or guess the Pre-Shared Key by capturing and analyzing the four way handshake.

4. Performance The Cracking Tests

To start the tests, experimented with five different ASCII passphrase and one hexadecimal character. In simulation, first gave an eight characters simple dictionary word to perform this test. When we tried to give the hexadecimal character, we were unable to change the PSK as the access point was not supporting it, so we gave a simple 64 character hexadecimal PSK to perform the test. In order to crack the WPA-PSK key, we need a four way handshake, so for performing this we used a tool called Airodump-ng which is used to capture the packets. For running this tool, we configured the client PC in a monitor mode with a same channel of access point[1].

In a machine started running aircrack tool by running the command,*airodump-ng –channel 6 eth0 –w /root/dump*

Where "channel 6" represents the channel for the wireless network and -w psk is the file name prefix for the file which will contain the IVs. The dumped (.cap) files can be made usable to read by running aircrack-ng tool.

5. Performing Offline Cracking Of Psk

After getting the .cap file format, we started the offline dictionary-based attack on the client machine 1.We downloaded the dictionary from the site ftp://ftp.ox.ac.uk/pub/wordlists . All the files which are downloaded from the site are in zipped format, so we used the unzip tool to extract it, after extracting to make all files into a single text file we run the command,

*$ cat * > NEWDICTIONARY1*

The dictionary consists of around two million words/phrases which contain the database of languages of different countries, dictionary words, etc., from the command prompt, when viewed the file using vi editor, vim *NEWDICTIONARY1* we found that it contains around more than two million words.

"NEWDICTIONARY1"[converted] 2646385L, 50554717C

As we had combined the lot of dictionaries and other database names, there might be some repetitions will be there. To remove the duplicates words/phrases after combining all the files, we used *sort NEWDICTIONARY1 | uniq > NEWDICTIONARY-sorted*

To capture the four way hand shake we used the command:

Aircrack-ng –a 2 –w NEWDICTIONARY1dump-01.cap

a) -w NEWDICTIONARY1.TXT is the name of the dictionary file. We need to specify the full path if the file is not located in the same directory.

b) *.cap is name of group of files containing the captured packets.

The first test was done to the .cap file which used 8 passphrase characters "defaults" key, and the result is as below.

Aircrack-ng 1.0 beta1
[00:14:43] 100394 keys tested (150.54 k/s)
KEY FOUND! [Defaults]
Master Key: 50 58 FF 76 A1 E6 CC 19 98 29 24 E7 48 BD BC C1
60 89 2F BB 49 EB EE DE DB 85 7F C4 AA 4C D0 04
Transient Key: 7E 91 42 FF CA B9 B4 07 C8 5A 24 D0 76 84 84 8F
DD 70 5C BF E1 0E 52 4B 44 A9 A5 5C 01 8F E3 A5
69 CA 58 5C 8E C1 79 A8 28 C8 5F A9 00 8B 00 B2
2A 02 E1 28 EB 57 5E 44 33 47 09 10 5E 2C FB F9
EAPOL HMAC:32 6C E7 C3 29 A9 F8 8B 24 19 90 20 FA 65 42 CE

From the above results, we had cracked the PSK key "defaults" for which it took 100394 keys tested from the dictionary file for a time of 14 minutes and 43 seconds with an average speed of 150.54 keys per second. One of the

drawbacks of using the dictionary is that, all the words in the dictionary are in lower case and if we give the key in the upper case the dictionary will not be able to find that word even it was in the lower case. To overcome this drawback we can convert all the lower case words in the dictionary to upper case, but following this method will be the dictionary file will get larger and it will take more time to crack the keys.

We also experimented with key like "DEFAULTS" in such a way to use the new dictionary which has been converted from lower case to the upper case as mentioned above. We obtained the following result as.

Aircrack-ng 1.0 beta1
[00:29:46]100394 keys tested (52.65k/s)
KEY FOUND! [DEFAULTS]
Master Key: F5 F7 F1 F1 F3 A2 37 59 63 CC F1 C1 88 DC 63 55
8F C5 BD 1B 8C F9 0B FA F2 B6 6A B8 83 72 1E 27
Transient Key: 0A CE 2D 7C 11 AB 46 BB BF DD A0 D2 0F 4A 67 4C
57 28 DF E3 AB 38 82 A8 D0 E8 58 9B 12 36 6A DB
17 08 C5 CD 72 3A 93 7C 7D 72 0B A4 82 1D 74 89
89 74 70 46 C6 51 04 AC CE B8 FB 7B 0F D4 D9 78
EAPOL HMAC:0D EF 8C 13 3C E3 3D D8 49 29 96 75 3C 00 05 72

To convert all the lower case words/phrases in the dictionary file to upper case and vice versa, we used the tr (translation) tool with the following command,

tr '[:lower:]' '[:upper:]' < NEWDICTIONARY-sorted > NEWDICTIONARY-uppercase

tr '[:upper:]' '[:lower:]' < NEWDICTIONARY-sorted > NEWDICTIONARY-lowercase

The drawback of this approach is that the dictionary's size will get larger by roughly a factor of 3 (not exact since not all entries are alphabet words/phrases), which in turn will force aircrack-ng to perform more tests and requires more time. Another drawback in using the dictionary is that the dictionary will not have any special characters, letters etc. We conducted an experiment by giving the PSK key as "D1apE@i$" and result is shown as below.

Aircrack-ng 1.0 beta1
[02:28:50]575198 keys tested (99.84 k/s))
Current passphrase: zyzzyvas
Master Key :AB 09 4C 47 52 26 96 B7 6C 1E AB 6B C5 9F 67 CD
E6 15 97 A1 B7 5E 69 F2 2A A9 70 D6 E6 A2 33 B9
Transcient Key :E5 FA 2D 35 74 BB 51 50 73 16 3C EB 9E 71 05 A8
99 3B E1 75 95 90 73 1E 81 31 F6 F9 D6 FD 8E 33
8A 41 7C 15 22 8C 62 02 5A 24 64 E6 B8 91 59 3D
43 7A 2F 23 6E 8D 2C FE 55 17 E2 B1 C9 A8 34 94
EAPOL HMAC : 87 8D 3A 9A 8C D0 5F C5 A9 A4 79 F4 30 68 A1 88
Passphrase not in dictionary
Quitting aircrack-ng...

From the above results it is clear that the dictionary we are using does not have any special characters which are in the key. The test was running for about two and half hours and after searching throughout the dictionary the process was halted and it quits the aircrack. For conducting this test, it has tested around 575198 keys at a rate of 99.84 keys per second.

As a last test of our experiment, we used the PSK keys with 64 characters hexadecimal key, which we were unable to crack the PSK key, as the key was unbreakable, when we run the aircrack software for more than an hour, we were unable to break the key. We got the result as follows:

Aircrack-ng 1.0 beta1
[02:27:57] 575194 keys tested (28.04 k/s))
Current Passphrase: zymotically
Master Key :D0 DD F9 A4 64 5A E6 22 77 08 D8 DB 79 8A D8 DE
EA C1 DE EB E2 5A 7A ED AA 88 83 BA 2B 53 4B A6
Transient Key: 54 28 63 1E 34 1C 3B B6 AA 14 BA 11 F3 EA 04 71
09 8A 32 49 63 30 27 5E 54 8F 57 72 C7 AD 55 1E
11 EC B8 29 03 7C 7D 4E 5E D9 4E 0F 0E 45 A7 17
C2 80 11 84 0A 29 48 1F 8B D8 32 67 60 22 43 90
EAPOL HMAC : 3B 9D A6 85 1F 60 29 97 53 33 C1 0D C9 21 2D 25
Passphrase not in dictionary.
Quitting aircrack-ng...

From the above results, we can see that for cracking this hexadecimal key it has took two hours twenty seven minutes and it has tested 5751944 keys with 28.04 keys per second. Hence, from the above tests it can be clearly seen that the probability of successful WPA-PSK key cracking depends on the quality of the dictionary being used. Since, for an eight character ASCII word, the aircrack has taken less time and in the case of the keys with special characters, hexadecimal numbers it takes more time and also the speed gets reduced.

6. Simulation Of The Test Result

In In this experiment, we tried with following key Passphrase which are listed in the below Table I.

Table I Results

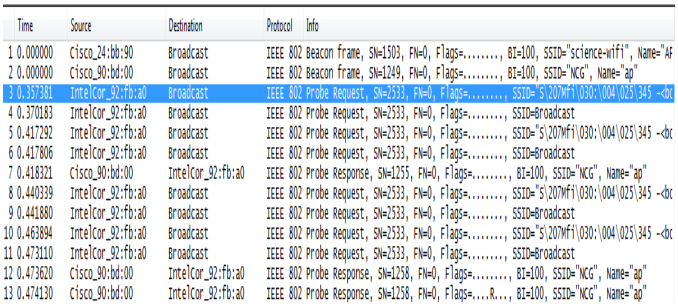
KEY	TYPE	LENGTH
Defaults	ASCII (Lower Case)	8 characters
DEFAULTS	ASCII (Upper Case)	8 characters

D1apE@i$	ASCII (With special characters)	8 characters
ILOVEMANAGEMENT	ASCII	15 characters
123456789012345 67890123456789 012345678901234 567890123456789 01234	Hexa-Decimal	64 characters

Hence, we can conclude that if the PSK keys are of longer key length with the combination of uppercase letters, lowercase letters, numbers and special symbols will make this kind of attack very difficult.

7. Measurement Analysis Of Four Way Handshaking Using Wireshark

When analyzing using the Wireshark, in the Figure.3 below, we found that the serial number 41 showing the Authentication and Association Request with Sequence Number 2533 and 2534 from IntelCor_92: fb: a0 to Cisco_90: bd: 00 for the access point NCG, but the probe request shows the source as IntelCor_92: fb: a0 and the destination as Broadcast with the same Sequence Number 2533 for some unknown
SSID="S\207Mfi\030:\004\025\345-<bogus>#\211\376\314#L\277\344\325\215\222"



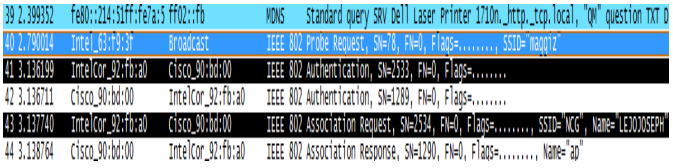


Figure. 3 Screen shot showing the Authentication Phase.

Also we can see the EAPOL Key (a four way hand shakes) exchanged between the source Cisco_90: bd: 00 and destination IntelCor_92: fb: a0 in the Figure 4.

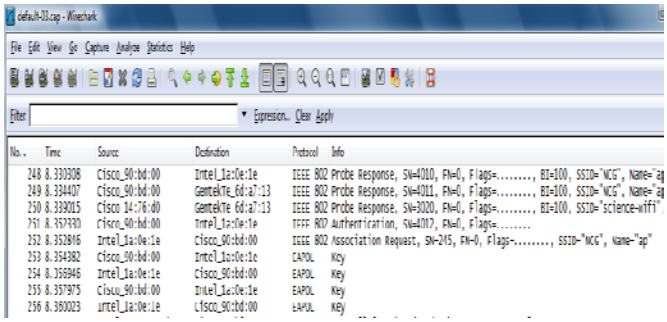


Figure.4 Screen shot showing the exchange of EAPOL Keys.

Figure. 5 & 6 represents the first pair and Figure. 7 & 8 represents the second pair which signifies the four "handshake" WPA packets. These are the four critical packets required by aircrack-ng to crack WPA using a dictionary.

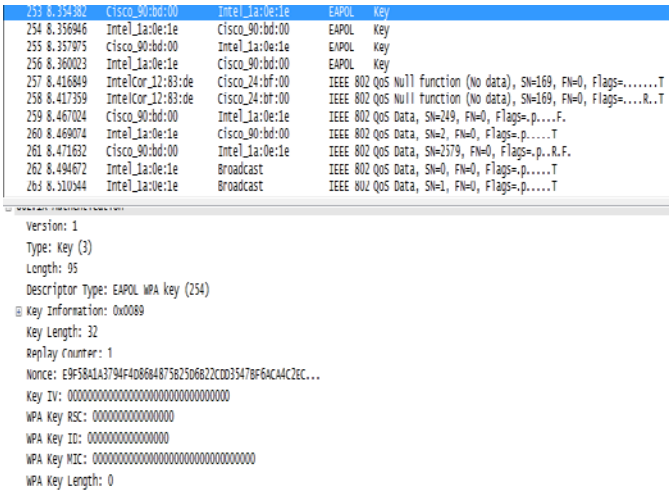


Figure.5 Screen shot of first pair of EAPOL Key.

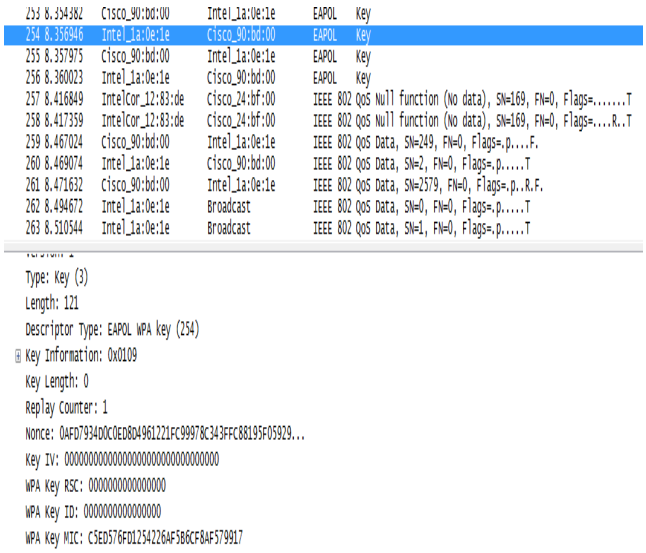


Figure. 6 Screen shot of first pair of EAPOL Key.

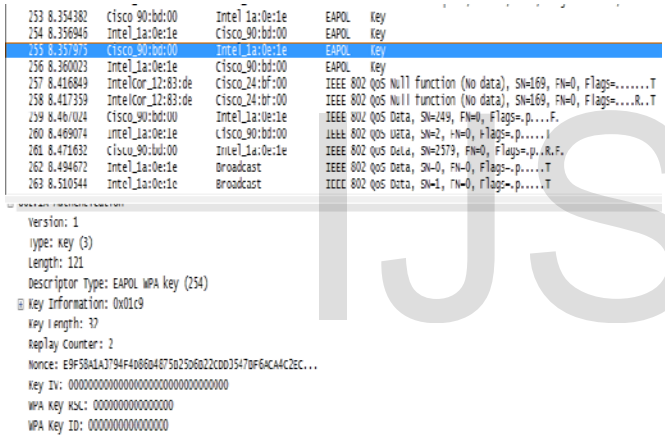


Figure. 7 Screen shot of second pair of EAPOL Key

From the above figures, it is clear that the AP initiates the four-way handshake by sending the first packet. The first pair of packets has a "replay counter" value of 1. The second pair has a "replay counter" value of 2. Packets with the same "replay counter" value are matching sets. This is why we have four EAPOL packets in our capture.

8. CONCLUSION

Wi-Fi Protected Access (WPA) overcomes the inherent flaws of early wireless networks. In the above experiment, we did a packet analysis on the 4 handshake signals captured and also to crack the WPA key. From the wireshark analysis of the captured packets, it's the AP that initiates the 4 way handshake. In this handshake the encrypted key is sent in parts and hence capturing this handshake signal, leads to a probability of the key being cracked, depending on the complexity of the set key password. But the key cracking is majority dependent on the passphrase contents in the dictionary. A simple ASCII word, which can be easily found in a dictionary, is easily cracked, whereas an inclusion of special symbols makes it more difficult for aircrack software to crack the key.

REFERENCES

[1] Ezedin Barka, Mohammed Boulmalf"On the Impact of Security on the Performance of LANs", JOURNAL OF OMMUNICATIONS, VOL. 2, NO. 4, JUNE 2007

[2] Trulove, J. "Build your own wireless LAN" .McGraw-Hill. Two Penn Plaza, New York. 2002.

[3] Shuaib,K. and Boulmalf M., "Co-existence of WLAN and WPAN Communication Systems" for the Handbook of Research in Mobile Business: Technical, Methodological and Social Perspectives" with IDEA group publishing (IGP), Hershey, PA, USA, April 28, 2006.

[4] Shuaib, K., Boulmalf M., Sallabi F. and Lakas A.," Performance Analysis: Co-existence of IEEE 802.11g with Bluetooth", Second IFIP International Conference on Wireless and Optical communication Networks, WOCN 2005, sponsored by IEEE. Dubai, March 6-9, 2005

[5] Khan, J and Khawaja, A. "Building Secure Wireless Networks with 802.11. Wiley Publishing, Inc. Indianapolis, Indiana, 2003.

[6] R.D.Vines. Wireless Security Essentials: Defending Mobile Systems from Data Piracy, Wiley Publishing Inc, 2002.

[7] Barken L., WEP Vulnerabilities—Wired Equivalent Privacy?, Dec 2003, retrieved on March, 2006,retrieved from http://www.informit.com/articles/article.asp?p=102230&seqNum=2

[8] Saleh M. and Al Khatib I., "Throughput Analysis of WEP Security in Ad Hoc Sensor Networks", The Second International Conference on Innovations in Information Technology (IIT'05), Dubai, 2005 .

[9] Agarwal K., Wang W.,"Measuring Performance Impact of Security Protocols in Wireless Local Area Networks", the Second International Conference on Broadband Networks. October, 2005. Boston, Massachusette, USA.