WPA PSK Crackers: Loose Lips Sink Ships

By Lisa Phifer

http://www.wi-fiplanet.com/tutorials/article.php/3667586/WPA-PSK-Crackers-Loose-Lips-Sink-Ships.htm (Back to article)

Wi-Fi Protected Access (WPA) protects wireless data by applying encryption, integrity checks, and sequencing. The first version of WPA patched around mistakes in the old, broken Wired Equivalent Privacy (WEP), while WPA2 started from a clean slate to deliver more robust, efficient security.

Either version of WPA can stop wireless eavesdropping -- with one big caveat. The encryption keys upon which they depend must never be disclosed to outsiders. That's where PSK crackers come in.

Can You Keep a Secret?

WPA and WPA2 are best when used with 802.1X Port Access Control for per-user authentication and per-session key delivery. Known as WPA-Enterprise or WPA2-Enterprise, this approach was designed for business networks with the staff and resources required to support RADIUS-based authentication. Every new session gets its own fresh random key, used for a relatively short time. While that doesn't make key cracking completely impossible, it substantially reduces that risk.

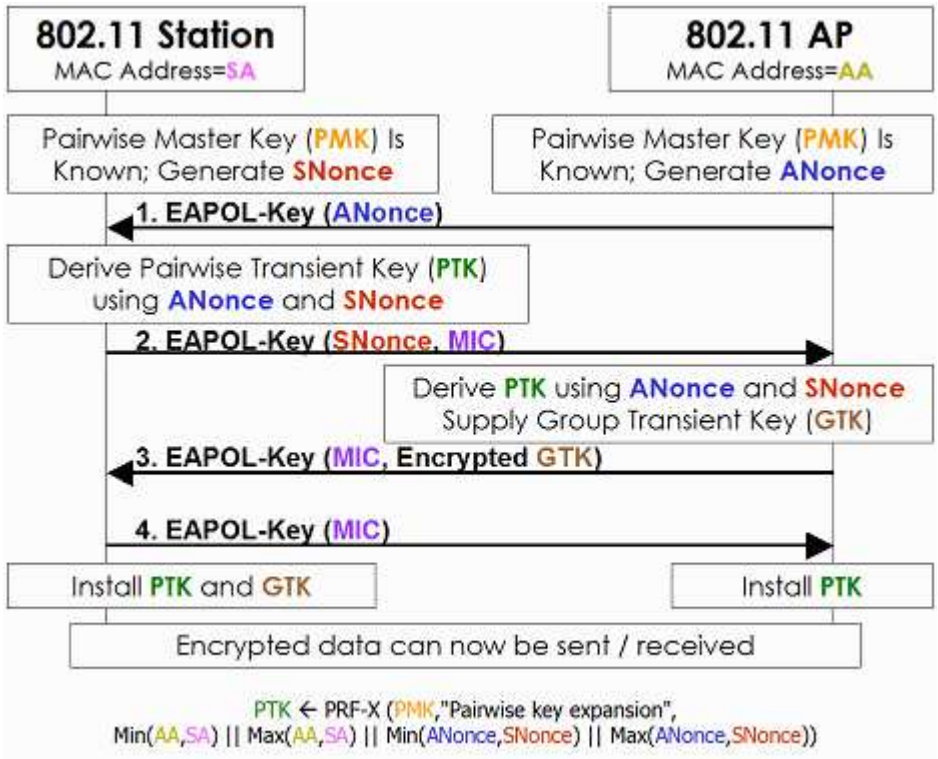
Since home networks don't generally have RADIUS servers, a simpler option also exists: Pre-Shared Keys (PSKs). Known as WPA-PSK, WPA-Personal or WPA2 Personal, this approach authenticates everyone using the WLAN with the same secret passphrase, configured into the Access Point (AP). But, unlike those old WEP keys, PSKs are not encryption keys -- they are the starting point for deriving per-station (client) encryption keys. (For how to set up WPA at home, see WPA-PSK: Step-by-Step.)

Unfortunately, the way in which WPA/WPA2 encryption keys are generated and delivered makes it easy for an attacker to try to guess your WLAN's PSK. Once an outsider has the PSK, he can steal service or decrypt data sent by legitimate users on your network.

Let's Shake on It

A PSK is a 256-bit value, known to every device in the WLAN. That PSK is usually generated by combining the WLAN's name (Service Set Identifier, SSID) with a passphrase (an ASCII string, 8-63 characters.) If you have ever used Windows XP to connect to a WPA-Personal WLAN, you have been prompted to enter a WPA passphrase.

When using WPA or WPA2, every station is permitted to associate with the AP. The AP or station has the option to start 802.1X authentication, exchanging Extensible Authentication Protocol (EAP) messages to verify user/server identities. After authentication (if any), the AP kicks off a four-way handshake (see figure below) to derive the keys for this session. The handshake must be completed before any encrypted data can actually be exchanged between this station and AP.



Four-way key handshake

What is actually happening during this handshake?

- The AP and each station need an individual Pairwise Transient Key (PTK) to protect unicast communication between them. To derive a different PTK for each AP/station combo, a Pairwise Master Key (PMK) is fed into an algorithm, along with MAC address and two values, ANonce and SNonce. Messages #1 and #2 in the figure above show how the AP and station manage to derive the same PTK without ever sending it over the air.
- The AP also generates a Group Transient Key (GTK) to protect all broadcast and multicast communication. Because every station on the WLAN needs that same GTK to decrypt broadcast/multicast frames, the AP sends the current GTK in message #3 of the handshake. To prevent eavesdropping, the GTK is encrypted with the PTK.
- To stop these handshake messages from being forged, messages #2 through #4 carry a Message Integrity Code (MIC). Each MIC is generated by hashing a specified part of the message, then encrypting that hash with the PTK.

This four-way handshake occurs whenever you connect to a WLAN using WPA or WPA2. It also occurs periodically thereafter, whenever the AP decides to refresh transient keys. End users may not be aware of this handshake -- and administrators may not really care about all of these gory details.

But WPA PSK crackers do.

What if Someone Hears You?

In WLANs that use WPA-Enterprise or WPA2-Enterprise, every session starts from a different PMK, delivered during 802.1X authentication. However, in WLANs using WPA-Personal or WPA2-Personal, there's no 802.1X, so the station and AP must use a value they both already know. What value do they both know? The PSK, of course.

WPA-PSK crackers like aircrack, KisMAC, and coWPAtty try to guess the PSK by capturing and analyzing the four-way handshake messages spelled out above. For example, the following output was generated by coWPAtty:

```
$ cowpatty -r capturefile -s soho-psk -f dictionaryfile
cowpatty 2.0 - WPA-PSK dictionary attack.

Collected all necessary data to mount crack against passphrase.
Starting dictionary attack.  Please be patient.
key no. 1000: apportion
key no. 2000: cantabile
key no. 3000: contract
key no. 4000: divisive
The PSK is "secretsecret".
4089 passphrases tested in 112.55 seconds:  36.33 passphrases/second
```

Here, coWPAtty uses a supplied dictionary file to analyze a packet capture file containing (at least) the four-way handshake exchanged during someone else's successful connection to the "soho-psk" WLAN. Recording handshake messages from an active WLAN is easily accomplished with a wireless capture program like Wireshark or Kismet. Those who get impatient can use a frame injector like Airjack to force legitimate users to reconnect.

CoWPAtty searches the capture file and extracts a four-way handshake for the named WLAN. It then extracts all of the values of interest from that handshake: AP and station MAC addresses, those SNonce and ANonce values, and message #4's payload and MIC.

Now, coWPAtty starts trudging through the supplied dictionary file, trying out words as possible passphrases, trying to come up with the right PSK. The right PSK will be one that, when used as a PMK with all of these observed values, ends up generating a matching MIC. This is shown in the following extra-verbose coWPAtty output:

```
$cowpatty -vv -r capturefile -s soho-psk -f veryshortlist
cowpatty 2.0 - WPA-PSK dictionary attack.

Collected all necessary data to mount crack against passphrase.
AA is:
         0020 a64f 31e4                           . .O1.

SPA is:
         000c 41da f2e7                           ..A...

snonce is:
         ed12 afbd a8c5 8305 0032 e5b5 2953 82d2  .........2..)S..
         7956 fd58 4a63 43ba fe49 135f 2695 2a0f  yV.XJcC..I._&.*.


anonce is:
         477b a8dc 6d7e 80d0 1a30 9d35 891d 868e  G{..m~...0.5....
         b82b cc3b 5d52 b5a9 a42c 4cb7 fd34 3a64  .+.;]R...,L..4:d


keymic is:
         f3a0 f691 4e28 a2df 1030 61a4 1ee8 3878  ....N(...0a...8x


eapolframe is:
         0103 005f fe01 0900 0000 0000 0000 0000  ..._............
         0200 0000 0000 0000 0000 0000 0000 0000  ................
         0000 0000 0000 0000 0000 0000 0000 0000  ................
         0000 0000 0000 0000 0000 0000 0000 0000  ................
         0000 0000 0000 0000 0000 0000 0000 0000  ................
         00f3 a0f6 914e 28a2 df10 3061 a41e e838  .....N(...0a...8
         7800                                     x. 00
              .

Starting dictionary attack.  Please be patient.
Testing passphrase: secretsecret
Calculating PMK for "secretsecret".
Calculating PTK with collected data and PMK.
Calculating hmac-MD5 Key MIC for this frame.

The PSK is "secretsecret".
```

Making a Bad Situation Worse

A bare minimum PSK (8 lowercase letters) has 26^8 possible combinations, so even just trying all 208,827,064,576 of those possible passphrases would take far too long. Working from a dictionary file cuts that effort by just trying all 8-characters words. Huge password dictionaries are readily available for use with conventional Windows/UNIX password crackers like John the Ripper, and they can be fed into PSK crackers. But multi-pass hashing for every word in those files still takes time -- depending on SSID and PSK length, a lot of time. In the above example, coWPAtty tested just 36 passphrases per second.

Unfortunately, conventional password crackers know how to get around this problem using rainbow tables -- long lists of pre-computed password hashes. The latest version of coWPAtty can now use a variation on rainbow tables to speed PSK cracking by three orders of magnitude. For example, a 2006 Shmoocon demo

showed coWPAtty testing 18,000 passphrases per second using a pre-hashed WPA PSK lookup table.

Remember that WPA PSKs combine the passphrase with the WLAN's SSID. A pre-hashed PSK lookup table therefore depends on the target WLAN's name. Given enough time, storage and CPU, you can generate a table for any SSID using the "genpmk" program now included with coWPAtty. Although genpmk itself doesn't really take less time, the generated tables can be fed into coWPAtty to crack individual PSKs much, much faster. Or you could feed coWPAtty one of the pre-hashed WPA PSK lookup tables posted online, representing 170,000 words hashed against the top 1000 most common SSIDs.

How to Protect Yourself

At this point, you have probably noticed several vulnerabilities that WPA PSK crackers exploit. There is nothing you can do about some of those vulnerabilities (like the values exchanged during the four-way handshake). But there are several steps you can take to mitigate other vulnerabilities and protect your WLAN against WPA PSK crackers.

1. WPA PSKs are particularly easy to guess if you choose a passphrase that is composed of word(s) easily found in a password dictionary, or words that anyone might easily associate with you (e.g., your surname, your pet's name). The strongest passphrases are not words but randomly-generated strings that mix case, letters and numbers.
2. WPA PSKs that are too short are also much easier to guess. When configuring a passphrase, the IEEE 802.11i standard strongly recommends using *at least* 20 characters. If you can't come up with your own long random strings, try a random password generator (e.g., keygen, wlankeygen).
3. Some products let you configure actual 256-bit PSK values, rather than entering the ASCII passphrase used to generate the WPA PSK. In a few products, you can enter the PSK as a hex string. In others, pushing a button causes the AP and station to derive their own random PSK (e.g., Broadcom SecureEasySetup). In January, the Wi-Fi Alliance announced Wi-Fi Protected Setup (WPS), a certification program intended to simplify configuration -- including strong PSK generation. Just 17 products have been certified to date, but more will emerge.
4. Given the existence of published WPA PSK lookup tables for common SSIDs, it can also be helpful to give your WLAN a unique SSID. Using a common or default SSID increases the odds of a successful WPA PSK dictionary attack against your WLAN.
5. WPA PSK crackers can be avoided altogether by stepping up to WPA/WPA2-Enterprise. Small businesses should give this very serious consideration. Aside from crackers, passphrases have all the usual password drawbacks, like when workers share your passphrase or lose a configured laptop. If 802.1X sounds too hard, check out entry-level solutions (e.g., Witopia SecureMyWiFi, ZyXEL ZyAir B-1000).