WPA Passive Dictionary Attack Overview

TakehiroTakahashi

This short paper presents an attack against the Pre-Shared Key version of the WPA encryption platform and argues the need for replacement.

What is WPA?

The WPA standard is a subset of the 802.11i wireless security standard intended to address the cryptographic shortcomings of Wired Equivalent Protocol (WEP). WPA comes in two forms: per-user based security designed for enterprises, and a pre-shared key mode designed for consumers. While the former utilizes a RADIUS server to ensure per user keying, the latter greatly simplifies deployment for home and SOHO users by having a master key (based on a pass phrase) for the wireless LAN. WPA and 802.11i are necessary because WEP has known weaknesses, poor key manageability, and lacks simplicity needed among home users for deployment.

Types of WPA

Enterprise Mode - Per-user authentication based protocol with the combination of 802.1x security framework, authentication server, TKIP key management and Michael integrity checking aimed for enterprise use. The 802.1x provides administrators with a variety of security implementations to establish authentication in which RADIUS is the de-facto. TKIP and Michael offer per-packet key mixing, a message integrity check and a re-keying mechanism with efficiency.

Consumer Mode - Pre-shared key (PSK) based protocol with the combination of Pre-Shared Key, TKIP key management and Michael integrity checking aimed for home use. Simplicity of deployment is of primary concern.

Mechanism

Essential WPA tasks in Consumer Mode:

1. associating with the access point (AP)
2. authentication and distribution of the PMK (Pair-wise Master Key)
3. creation and installation of the PTK (Pair-wise Transient Key) based on PMK
4. integrity check
5. a successful wireless session using TKIP based on PTK

Vulnerability

The PSK version of WPA suffers from an offline dictionary attack because of the broadcasting of information required to create and verify a session key. In WPA, the PMK (master key) is produced by running a special function on a pre-shared pass phrase and an SSID. Both the host and the AP use this PMK, along with MAC addresses and nonces, in order to create the PTK (session key) and install it on both sides. The following is pseudo code for the creation of the PMK and the PTK, where PBKDF2 and PRF-512 are key generating algorithms based on keyed hashes.

PMK = PBKDF2(passphrase, ssid, ssidLength, 4096, 256)
PTK = PRF-512(PMK, “Pairwise key expansion”, Min(AP_Mac, Client_Mac) | |
Max(AP_Mac, Client_Mac) | | Min(ANonce, SNonce) | | Max(ANonce, SNonce))

The PMK is generated by inputting the string of the pass phrase, SSID, and the SSID length into the PBKDF2 algorithm, which is set to hash 4096 times and generate a value of 256 bits. Since the SSID is easily recoverable, it should be noted that only the pass phrase would have to be guessed in order to determine the valid PMK. Furthermore, in the generation of the PTK for cracking purposes, only the PMK needs to be determined since all other fields can be trivially discovered; the 1st step in the 4 way handshake provides ANonce and AP_MAC while the 2nd step provides SNonce and Client_MAC, and the signature of the PTK just generated. The PTK consists of 4 keys: Key Confirmation Key (KCK), Key Encryption Key (KEK), Temporal Key 1, and Temporal Key 2. After receiving the 1st packet of the 4way handshake traffic, the client generates the PTK and runs MD5 hash function on the KCK and the EAP packet to be sent. This hash is then added to the EAP packet and sent over the network as the 2nd step. Now, an intruder can utilize the

hash portion of this packet and match it with the hash result of his guessed PTK and collected EAP packet; the correctly guessed pass phrase produces the same signature. Hence the intruder, by passively sniffing two of the EAPOL packets, can begin an offline dictionary attack. The availability of the attack is not constrained to the state of users as long as there exits an active session within the network. A well known disassociation attack can be used to trigger a re-association between the host and the AP at which time the attacker can gather the necessary packets.

Analysis

A tool called wpa_attack has been prepared to challenge this vulnerability. The effectiveness of the attack is directly proportional to the entropy of the pass phrase chosen. Because a detailed analysis of heuristics deployed in a typical password cracking tool is beyond the scope of this paper, any experimental results are not provided here. However, the software extends "Password Cracking Library" (PCL) and does provide an easily configurable interface allowing flexibility in implementing heuristics.

A study of password choice observed in UNIX system administrators supports the relative feasibility of the tool: nearly 40% of administrator's passwords consisted of a word or combination of words [1]. It is a fairly reasonable assumption that less technically inclined users are likely to choose low entropy pass phrases at a higher rate.

Conclusion

The PSK version of WPA is rather self-contradicting. The nature of a 'pass phrase' is likely to constrain the entropy of the pass chosen, leaving a great opportunity for a tool like wpa_attack to automate the process of effective off-line password cracking. Hence, the Wi-Fi alliance recommends a pass phrase longer than 20 characters, a requirement unlikely to be executed in practice by its target audience. In the end, the PSK version of WPA does not provide what is expected for end-users: easy and secure wireless connectivity. A clear resolution to this issue can be found in the project called "tinyPEAP", named after the protocol it uses for 802.1x authentication phase. In short, tinyPEAP is a small self contained authenticator utilizing RADIUS and PEAP/MSCHAPv2, allowing developers to implement enterprise level security within consumer level products. Even more

importantly, it also provides users with a straight forward setup and uses familiar username and password based credentials for authentication. A user with tinyPEAP enabled devices can enjoy the highest level of the wireless security while easily managing the backend through graphical interfaces (i.e. Linksys WRT54G version implements a web interface).

References

[1] M. Bishop, “Improving System Security via Proactive Password Checking”, 1994

[2] R. Moskowitz, ““PSK as the Key Establishment Method”, 2003

4 Way Handshake