WHITE PAPER

WPA3 AND ENHANCED OPEN: NEXT GENERATION WI-FI SECURITY

aruba
a Hewlett Packard Enterprise company

TABLE OF CONTENTS

TODAY'S WI-FI SECURITY PROBLEMS

Work on the protocols that became WPA2 began in 2001 by the IEEE 802.11i Task Group. At the time there was no "Wi-Fi," and 802.11 was not pervasive as it is today. It was largely a first hop technology for offices, and Wi-Fi interfaces took the form of PCMCIA cards (i.e. PC Cards) plugged into laptops. In addition, wireless access points (APs) were stand-alone devices (no controller-based architectures existed yet) that had small, limited CPUs which could not do complex cryptographic operations.

The IEEE 802.11i standard was finally ratified in 2004 with two defined modes of operation: one that used a pre-shared key (PSK) to authenticate a simple handshake; and 802.1X/EAP which offloaded the authentication work to a third-party server.

Once work on IEEE 802.11i was finished, the Wi-Fi Alliance certified implementations under the WPA2 banner. If an implementation achieved WPA2 certification, it was almost guaranteed that it would work with other WPA2-certified devices. The PSK mode in IEEE 802.11i became known as WPA2-Personal (or sometimes WPA2-PSK), and the 802.1X/EAP mode in IEEE 802.11i became known as WPA2-Enterprise.

That was over 15 years ago (a lifetime in Internet years) and now IEEE 802.11i is starting to show signs of age.

Issues with PSK Mode

As soon as it was released, PSK mode was acknowledged to be susceptible to attack. To keep the cryptographic operations of the AP down to a minimum, the secret key used in the simple, lightweight authentication handshake was directly based on the pre-shared key. This opened the mode up to an off-line dictionary attack where an attacker sees the simple handshake execute over the air, and then takes copies of the handshake messages and goes off-line – trying every password imaginable until it finds one that can validate the handshake messages.

This is not as onerous as it might sound because most passwords in use today are typically one of several thousand, so the amount of compute power necessary to discover the password was easily within the reach of any moderate attacker. Furthermore, since it is an off-line attack, the work could be farmed out to others. So even with strong passwords it was only a matter of time before the attacker succeeds.

In fact, that is exactly what has happened: arrays of FPGAs each programmed to do this specific attack have been used to run through hundreds of thousands of PSKs per second, thereby making even long and somewhat complex PSKs vulnerable to attack.

The problem is that APs did not have the compute power necessary to implement a strong and secure protocol at the time the protocol was developed – so the onus of security was placed on the users. For 802.11i's PSK mode to be used securely, it was necessary to utilize long, complex, mixed case PSKs with numbers, letters and special characters. But the more complex the PSK, the harder it is to manage and the lower the probability that it was entered correctly.

The human element in PSK management puts an effective upper boundary on the complexity that is possible in an 802.11i PSK-managed network. Hence, an upper boundary on the security that network will end up having which will affect all users and devices.

Issues with 802.1X/EAP

To keep the APs from having to do too much work and still allow for strong, cryptographic authentication to be achieved with IEEE 802.11i, the 802.1X/EAP mode of operation was defined. In general, this uses a standalone server, separate from the client and AP, which speaks EAP (Extensible Authentication Protocol) and authenticates itself to the client, and optionally authenticates the client. Once the EAP authentication exchange negotiates a shared secret called the Pairwise Master Key (PMK), the key is sent from the EAP server to the AP which performs the lightweight authentication handshake.

The first EAP method, LEAP, was woefully insecure and it was immediately determined that the Transport Layer Security (TLS) protocol should be leveraged inside EAP to facilitate a more secure connection. This resulted in PEAPv0, PEAPv1, and TTLS, of which all protocols use TLS to do authentication and key establishment.

Authentication is a two-step process with these EAP methods where the server authenticates itself to the client using TLS, and then through the secure TLS tunnel the client authenticates itself to the server, usually via a username and password.

Configuration of 802.1X/EAP is difficult and assumes special knowledge that your average Wi-Fi user does not have. For instance, they typically cannot define an "inner EAP method" or an "anonymous identity" – let alone even know what the value should be in those fields. So, 802.1X/EAP deployments are really only done when a skilled IT department is able to provision each and every client prior to connecting to the network.

The big issue with 802.1X/EAP is due to its numerous options. It is possible to connect with a configuration that superficially looks secure—hashing with SHA256, or encrypting with AES and 128-bit keys—but actually it ends up considerably less secure due to other parameters outside the control of the end-user. It is possible to negotiate AES-CCM-128 at association, but you end up with a key exchange that results in a key with approximately 60- to 80-bits of security.

For example, negotiating any of the following TLS cipher suites inside of EAP will end up producing a symmetric key that is not suitable for any non-deprecated cipher in 802.11:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_RC4_128_MD5
- TLS_DH_DSS_WITH_AES_256_CBC_SHA

This is due to the fact that SHA or MD5 is the hash function used or RC4 is the cipher being used. Also, using a TLS cipher suite that performs an RSA key exchange with a certificate that has a 1024-bit RSA key will result in a symmetric key that is not suitable for any non-deprecated cipher in 802.11. The problem is that the client has no programmatic control over what TLS cipher suites get negotiated inside of EAP.

This problem is exacerbated by the common sense urge to ensure there would be no connectivity issues due to incompatible cipher suite negotiation by negotiating down to the most common denominator, which unsurprisingly is not the most secure.

Problems Addressing Popular Use Cases

As mentioned above, IEEE 802.11i was conceived before Wi-Fi became ubiquitous. Coffee shops, hotels, and restaurants did not offer Wi-Fi and if anyone was anticipating the deployments we see today, they were not taking part in the development of IEEE 802.11i.

When Wi-Fi started taking off, integrated radio technology replaced PCMCIA adaptors in laptops. Soon after, it became a necessity to embed Wi-Fi radios in every mobile device. People expected it wherever they went and its presence or absence influenced their decision to enter an establishment or leave. In order to attract and retain customers, proprietors would install an AP and offer free Internet service in their premises.

But these people were not in the business of providing Internet service. They were selling coffee, scones, beer or a big, juicy steak. They didn't know much about Wi-Fi security and, frankly, didn't want to know. All they knew was they were not going to purchase, configure, and maintain an EAP server and ask their users to try and configure 802.1X/EAP!

The only other option was PSK mode. Attempting to provide each customer with a unique PSK was out of the question so the PSK had to be shared among all users. To facilitate Internet access and free up the staff to deal with the business at hand and not perform IT tasks, the PSK was then made public. In fact, the popular practice today is writing the PSK out on a chalk board or menu for all to see – it was the only option available to meet this use case.

Unfortunately, this practice is completely insecure. Since the PSK is written up there on the chalkboard an attacker does not need to perform dictionary attack! The PSK is known and they can easily capture the simple, lightweight handshake that the client and AP must engage in. The PSK is then used to determine the encryption keys being used by the client and AP.

Every frame can be decrypted, modified, replayed, and frames can be forged. Additionally, as the attacker knows the PSK, it is trivial to create a rogue AP that attracts clients and is then able to intercept all traffic sent from and to a client. Shared and public PSK's effectively afford the same security as an Open network.

Another popular deployment model that was not served by the two modes of WPA2 is that of a captive portal. This is for deployments in which more of an Internet Service Provider (ISP) approach is most commonly used. Clients connect to the AP and then are redirected to a server which can then ask for Terms and Conditions to be acknowledged, require the user to watch a video, or use a credit card, to obtain Internet access. Once the user has satisfied the captive portal server workflow, traffic will no longer be redirected and the user is given access to the Internet.

Since these captive portal deployments use a third-party server to handle user validation, and such validation does not require any prior provisioning of client devices, the interaction between the client and AP is done in the clear. The client does "Open" 802.11 Authentication and Association, and frames sent between a client and AP are not secure.

The downside is that the captive portal is typically engaged in a cryptographic exchange with the client prior to giving the client Internet access, but all of the cryptographic state is thrown away when the client finishes the captive portal validation. Anyone who is in proximity of the client and AP can also forge a de-authentication frame, kick the user off the network, and assume the MAC address of the client in order to steal Internet access.

Clearly, the above are use cases that are poorly served by WPA2.

WPA3 TO THE RESCUE

Each of the problems described above were discovered over the years and an effort was made to design a suitable protocol that would help address each issue. Eventually, a suitable family of protocols were defined that would become a new Wi-Fi Alliance certification program. Hence, WPA3.

Addressing the PSK Problem

Moore's Law states that the number of transistors on an integrated circuit will double every two years. This has led some to infer that either the size of a transistor decreases by half or the computing power doubles. The effect of Moore's Law on APs means that an AP's computing power used to perform strong cryptographic operations doubles every couple of years. This, coupled with more efficient elliptic curve cryptography, means protocols performing strong cryptographic authentication and key exchange protocols can be run on the AP. Eventually, it became time to address the PSK problem.

The Simultaneous Authentication of Equals (SAE) protocol was added in the late 2000s to the IEEE 802.11s (Mesh Networking) standard. IEEE 802.11s was certified in 2012. SAE is an instantiation of the dragonfly key exchange which performs a password-authenticated key exchange using a zero knowledge proof—each side proves it knows the password without exposing the password, or any password-derived data.

By using a zero knowledge proof, an attacker is no longer able to witness a single exchange and go off-line to crack the PSK. The only way an attacker can learn whether a guess of the password is correct or not, is to actively engage in SAE—one guess per active attack. With SAE, all those arrays of FPGAs and those rainbow tables dedicated to cracking passwords are all worthless.

The implication of a zero knowledge proof means that SAE can be used with passwords that have traditionally been referred to as weak. WPA2-PSK is a weak protocol and compensates by putting the burden of security on the one place it shouldn't belong—the users. Requirements were placed on passwords such that they needed to be double digit in length, mixed case, including numbers and special characters, etc.

This created a situation in which the password was hard to remember and entering it correctly was prone to mistakes. Of course, this resulted in passwords being written down somewhere which completely defeated their purpose. With SAE, the requirement on passwords is only that they are hard to guess – for example, picking a number between 1 and 10,000,000. If this password was used with WPA2-PSK, it would take a few seconds of an off-line dictionary to determine the password. But if the same password is used with SAE, it would take approximately 5,000,000 active attacks before the probability even became 0.5. Active attacks are simple to detect and mitigate.

802.1X Consistency

WPA3 introduces a new configuration option for 802.1X/EAP called CNSA (Commercial National Security Algorithms). CNSA was defined by the United States National Security Agency (NSA) to protect secret and top-secret data on government and military networks. Due to the fact that CNSA affords consistent security without the ability to misconfigure, it is being adopted by enterprises that have strong security requirements – like financial institutions.

CNSA establishes a suite of cryptographic algorithms that all afford roughly the same level of protection: SHA384 for hashing, NIST's p384 elliptic curve for key establishment and digital signatures, and AES-GCM-256 for data encryption and authentication. With CNSA, the EAP method must be EAP-TLS and the negotiated TLS cipher suite must exclusively use cryptographic algorithms from the CNSA suite.

What this means is that deploying CNSA insures that no 802.1X/EAP misconfigurations are possible. It is not possible to mix-and-match algorithms in an insecure manner and there are no possibilities for incompatibility or cipher downgrades – and this dramatically simplifies network deployment.

Enhanced Open: Securing Open Networks

Coffee shops and other public venues want a simple way to provide customers with some semblance of security. Open networks exhibit well-known problems, so they had no choice but to use WPA2-PSK with a shared and public PSK. Now there's a new solution that provides more security than a shared and public PSK—Wi-Fi CERTIFIED Enhanced Open™ with Opportunistic Wireless Encryption (OWE).

OWE is an alternative to Open networks. It has the same work-flow and the same user requirements. Basically, click on the available network and get connected. To the user, an OWE network looks just like an Open network (with no padlock symbol), but the advantage is that it's encrypted. OWE performs an unauthenticated Diffie-Hellman when the client associates to the AP. The result of that exchange is a key known only to two entities in the entire world, the client and the AP. That key can be used to derive keys to encrypt all management and data traffic sent and received by the client and AP.

While an unauthenticated Diffie-Hellman is technically insecure, it actually provides a higher level of security than a shared and public PSK with WPA2-PSK—basically the "password on the chalkboard" approach to network access. Because the PSK is public, everyone in earshot of the AP can figure it out – and because its shared, everyone uses the same PSK.

The implications are that any user can impersonate the AP (the client can't authenticate the AP) and the AP has no idea who is connecting (the AP can't authenticate the client); basically the shared and public PSK mode is completely unauthenticated, just like in OWE. But with OWE, the Diffie-Hellman exchange will give a truly pairwise and unique key to the client and AP – which means no one else can eavesdrop the connection. It is not possible for an attacker to decrypt, forge, modify, or replay any frame sent between the client and AP.

With a shared and public PSK, an attacker knows the PSK (everybody does!) and can therefore determine the encryption keys used by the client and AP by just passively observing the 4-way handshake. OWE provides a higher level of security to public venue deployments than WPA2 can ever offer.

For captive portal deployments, OWE offers security where none existed before. In these environments, the client and AP do an OWE exchange before the captive portal kicks in. All frames, including those being redirected to the captive portal, will be protected by the pairwise and unique keys derived from the OWE exchange. The captive portal can do its authorization business—forcing the user to click on Terms and Conditions, make the user watch a video, or obtain credit card information for Internet access—comfortable in the knowledge that the air is encrypted.

Once the captive portal has authorized the client, the AP can allow the client's traffic onto the Internet and retain the keys established by OWE at association time. The captive portal has authorized a user identified by a MAC address and the keys OWE established are identified by the same MAC address. Since OWE includes management frame protection, it is not possible for an attacker to forge de-authentication frames to kick a valid user off the network and steal its MAC address.

SUMMARY

WPA3 and Enhanced Open represent a long overdue evolution for Wi-Fi security. The Internet and how it's used has changed considerably since WPA2 was released, and the problems and issues associated with it have come to the forefront. WPA3 addresses the shortcomings of WPA2 and also addresses use cases that WPA2 could not.

An important part of WPA3 is that security is increased while complexity is not. Typically increases in security are accompanied by increases in complexity, which makes security harder to obtain and implement. The advantage of using WPA3 is that there are no changes in workflows or usage, no new steps to go through or caveats to remember. OWE looks just like open networks we're all used to – click and connect. And, WPA3-SAE looks just like WPA2-PSK, where you enter a password and connect. Lastly, CNSA removes the possibility of misconfiguration while leaving the 802.1X/EAP workflow alone.

REFERENCES

- Harkins, D. and W. Kumari, "Opportunistic Wireless Encryption", RFC 8110, March 2017
- IEEE 802.11-2016, https://standards.ieee.org/standard/802_11-2016.html, December 2016
- Harkins, D., "The Dragonfly Key Exchange", RFC 7664, November 2015
- US National Security Agency, "NSA Suite B Cryptography", January 2009
- Wi-Fi Alliance, "Device Provisioning Protocol Technical Specification" v0.2.8, December 2017
- Harkins, D. "The Public Key Exchange", draft-harkins-pkex-05, January 2018
- Stejano, F, and A. Ross, "The Resurrecting Duckling", Lecture Notes in Computer Science, vol. 1796. Springer, Berlin, Heidelberg, 1999
- IEEE 802.11ai-2016, "Amendment 1: Fast Initial Link Setup", 2016

WP_WPA3EnhancedOpen_110118 a00059003enw