WIRELESS

WORKSHOP

Crash Course: The ABCs of WPA2 Security

Learn How Wi-Fi Protected Access 2 Works

Our WPA2 primer tells you how to use this standard to secure your network and how it handles encryption and access control. It also addresses why it's safer than previous standards.

Jan 27, 2006 - By Frank Bulk

Looking for more secure Wi-Fi? WPA2 (Wi-Fi Protected Access 2) gives wireless networks both confidentiality and data integrity, two terms not previously associated with Wi-Fi.

Security, of course, has long been the trade-off with Wi-Fi. Early wireless networks leaned heavily on VPNs to provide Layer 3 security, which--aside from the additional overhead of encapsulation and the challenges of roaming, quality of service, client support and scalability--left the IP network vulnerable to attacks. The Layer 2-based WPA2 better protects the network.

But WPA2 alone can't provide enterprise security: Combining WPA2 with the IEEE 802.1X port-based authentication protocol for access control should eliminate most security worries. This won't protect you from rogues, denial-of-service attacks or interference, but it will ensure secure wireless communication.

The Wi-Fi Alliance's WPA2 security spec is a major improvement over WEP (Wired Equivalent Privacy), the security standard in IEEE's original 802.11 (for more on WEP, see "WEP: Old-School Security for Wi-Fi" at). WEP was susceptible to attacks and poorly implemented by vendors, and never took off in the enterprise. WEP's weaknesses and the ease with which they've been exploited led to the 802.11i standard, which was approved and published in 2004. The Wi-Fi Alliance created WPA, a subset of the draft version 802.11i, and later, WPA2, which provided stronger security than the first version of WPA.

Standards
Click to enlarge

WPA came with support for TKIP (Temporal Key Integrity Protocol), which uses the RC4 cipher, and it can be implemented in software with just a driver or firmware update. Keys are rotated frequently, and the packet counter prevents packet replay or packet re-injection attacks. WPA provides integrity checking using MIC (Message Integrity Code), sometimes nicknamed "Michael." Although this checksum method can be attacked with brute-force methods, network traffic is halted automatically for a minute and the session keys reset if a WPA-based access point detects more than one TKIP MIC failure within 60 seconds, so the risks are minimal.

WPA2, meanwhile, uses a new encryption method called CCMP (Counter-Mode with CBC-MAC in another window

Protocol), which is based on AES (Advanced Encryption Standard), a stronger encryption algorithm than RC4.

Both WPA and WPA2 include two authentication modes: personal and enterprise. WPA2-Personal generates a 256-bit key from a plain-text pass phrase, sometimes called a PSK, or preshared key. The PSK (as well as the Service Set Identifier and SSID length) form the mathematical basis for the PMK (pairwise master key) that's used to initiate a four-way handshake and generate the PTK (pairwise transient key)--or session key--between the wireless user device and access point. WPA2-Personal, like static WEP, poses challenges in key distribution and maintenance, making it a fit for small offices but not the enterprise.

WPA2-Enterprise, meanwhile, addresses concerns regarding distributing and managing static keys, and controls access on a per-account basis by tying in to most organizations' authentication services. This mode requires credentials, such as a user name and password, a certificate or a one-time password, and authentication occurs between the station and central authentication server. The access point or wireless controller monitors the connection and directs authentication packets to the appropriate authentication server, typically a RADIUS server. The framework for this is 802.1X, which supports user and machine authentication with port-based control that works for both wired switches and wireless access points.

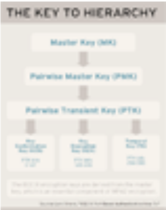
Three major components of 802.1X authentication are supplicant, authenticator and authentication server.

The 802.1X specification describes the supplicant as the device requesting access to the network, usually a laptop or mobile device, but in practice it's software on that device that initiates and responds to 802.1X commands (see "Supplicants: Pleading for Access").

The authenticator--typically an access point, but in a centralized AP architecture, it may reside on the switch/controller--authenticates the client to the network. This device processes requests from the supplicant, and leaves the network interface blocked unless directed by the authentication server to unblock it.

The authentication server, meanwhile, receives and processes the authentication request. It usually is a RADIUS server, but it's not just any RADIUS server--it must be compatible with the supplicant's EAP (Extensible Authentication Protocol) types (for more on EAP, see "D-EAP-en Your Understanding").

EAP traffic is exchanged between the client (supplicant) and AP (authenticator) over the Layer 2 EAPoL (EAP over LAN) protocol. The supplicant doesn't have Layer 3 connectivity to the RADIUS server: When the AP receives EAP traffic from the client, it converts it to the appropriate RADIUS request and then passes it to the RADIUS server for processing. If the supplicant encrypts the data, the authenticator can't inspect the contents of the request, but can extract from the response attributes such as the client's VLAN assignment.



The Key To Hierarchy
Click to enlarge in another window

After 802.1X authentication, the client receives the master key (MK) from the authentication server. The master key is tied to that authentication session. From the MK, the same primary master key (PMK) is generated on both the client and the authentication server. The authenticator--in this case an access point--receives the PMK from the authentication server through a predefined RADIUS attribute. Once the client and access point possess the PMK, the client and AP generate the pair-wise transient key (PTK) without actually exchanging it. This is possible over a four-way handshake, which eliminates a successful man-in-the-middle attack.

WPA2's PTK comprises three types of keys. They are the Key Confirmation Key (KCK), which is used to check the integrity of an EAPOL-Key frame (used in the MIC), the Key Encryption Key (KEK), which encrypts the GTK, and the Temporal Keys (TK), which secure data traffic.

All wireless devices associated with an access point must be able to decrypt the broadcast and multicast traffic. They do so with the same group key, or GTK. If the AP changes the GTK because it was compromised, for example, the AP issues a replacement key using a simpler two-way handshake with the KEK encrypting the GTK.

Because this entire process of client authentication to the RADIUS server can take up hundreds of milliseconds (if not seconds) when a device is roaming from one AP to another, it's unacceptable for Wi-Fi phones or streaming applications on laptops. So most enterprise wireless products have 802.11i features that help minimize roaming latency--preauthentication and PMK caching.

Preauthentication lets a mobile client authenticate with other APs in its vicinity while remaining associated with its primary AP. With PMK caching, a roaming client need not fully reauthenticate over 802.1X when it returns "home."

WPA2 is built around AES, which has replaced DES and 3DES as the de facto industry encryption standard. The computationally intensive AES requires hardware assistance, something not always in older WLAN equipment.

WPA2 uses CBC-MAC (Cipher Block Chaining Message Authentication Code) Protocol for authentication and integrity, and CTR (Counter Mode) to encrypt the data and MIC. WPA2's MIC is similar to a checksum and provides data integrity for the nonchangeable fields in the 802.11 header, unlike WEP and WPA. This prevents packet replay from being exploited to decrypt the packet or compromise cryptographic information.

MIC calculation uses a 128-bit IV (initialization vector). The IV is encrypted with AES and the temporal key, producing a 128-bit result. The algorithm then performs an exclusive OR on that result and the next 128 bits of data. The result of this calculation is encrypted with AES and the TK, and then an exclusive OR is performed on that and the next 128 bits of data. The last step is repeated until all 128 blocks in the 802.11 payload are exhausted. At the end of the operation, the first 64 bits are used to produce the MIC.

Protocols Compared
Click to enlarge in another window

The counter-mode encryption algorithm encrypts the data and the MIC. The algorithm begins with a 128-bit counter preload similar to the MIC IV, but uses a counter value initialized to 1 instead of a data length. So a different counter is used to encrypt each packet.

The first 128 bits are encrypted using AES and the TK, producing a 128-bit result, and an exclusive OR is performed on that result. The first 128 bits of data produce the first 128-bit encrypted block. The counter preload value increases incrementally and is encrypted with AES and the data encryption key. Then an exclusive OR is performed on that and the next 128 bits of data.

The last step is repeated until all the 128-bit blocks have been encrypted. Then the final counter value is set to 0 and encrypted using AES and XORed with the MIC. The result is appended to the encrypted frame.

Once the MIC is calculated using CBC-MAC, the data and MIC are encrypted. That information is prefixed with an 802.11 header and the CCMP packet number field, appended with the 802.11 trailer, and then sent out.

WPA2 decryption works in reverse. The counter value is derived from the same algorithm used in the encryption. That value and the encrypted portion of the 802.11 payload are decrypted with the counter mode decryption algorithm and the TK, which results in the decrypted data and MIC. The data then goes through the CBC-MAC algorithm to recalculate the MIC. If the values don't match, the packet is dropped. If they do, the decrypted data is sent up the network stack and to the client.

Most of the latest enterprise wireless systems support WPA2 or are upgradable to it. But if you don't have an authentication or RADIUS server that supports the requisite EAP types, you'll have to pull together the elements to do so. And you probably have a few laptops and PC cards that don't support WPA2 because they lack the necessary AES encryption hardware. Sometimes a firmware and/or driver upgrade will activate that functionality.

Another challenge is getting WPA2 to embedded or small form-factor devices such as PDAs, Wi-Fi phones, barcode scanners and wireless print servers. These devices tend to lag in security features due to integration challenges and their infrequent replacement lifecycle.

You can create a separate SSID with WEP or WPA on a separate VLAN with limited, controlled and monitored access to your network. An example is Wi-Fi phones that support only WEP or WPA-PSK: Because they need to communicate only with the VoIP infrastructure, you should restrict them from accessing the general corporate network. Of course, voice calls are still susceptible to decryption, and it might make sense to wait for handsets that support some form of 802.1X.

Supporting WPA2 on your existing desktops and laptops isn't always easy. If the type of EAP you're using is not supported by the wireless station's OS, you can use the supplicant provided on your wireless card's drive or install, configure and manage a third-party supplicant. If you can't convert all your users in short order, you can overlay your system with a new SSID that uses WPA2 or mixed-mode encryption. Then you can convert your devices to WPA2 by location, for instance.

Either way, Wi-Fi is ready for prime time when it comes to enterprise security. WPA2 provides encryption and data integrity, and when used with 802.1X authentication, you get complete link-level security.

Frank Bulk is a contributing editor to Network Computing. He works for a telecommunications company based in the Midwest. Write to him at fbulk@nwc.com.

Wired Equivalent Privacy (WEP) provides encryption using a simple algorithm based on the RC4 stream cipher. RC4 made sense as the encryption algorithm for wireless when the first 802.11 standard was developed because only minimal computational cycles could be assigned for security: RC4 was simple and fast.

A known shared key, 40 or 104 bits in length, is used with a 24-bit IV (initialization vector), yielding a 64- or 128-bit WEP key to encrypt the data. One critical weakness was the use of a "short" IV. Within a few hours of heavy traffic, the same IV will generate a collision, which is a cryptographic term for "repeat." The repetition of the IV combined with different payloads means that with enough captured packets, the WEP key can be identified. The WEP key can be used to associate to the once-secure network, as well as decrypt any past or future traffic within range of a capturing device. Some active attacks, such as those that inject traffic, make it possible to capture traffic for only a few minutes and decrypt the keys even faster (see www.tomsnetworking.com/Sections-article111-page1.php and www.securityfocus.com/infocus/1814), making WEP merely a nuisance for hackers and therefore painfully vulnerable.

WEP also provides integrity checking through an ICV (integrity check value), but the CRC-32 algorithm is well-known and the value easy to manipulate. WEP doesn't explicitly support any authentication, though it can run within a 802.1X context. Even with dynamic key rotation, WEP is still vulnerable to frame manipulation because of its lack of a strong integrity check and decryption.

Back in the heyday of the wireless Wild West, enterprise networks were open and free. SSIDs were considered passwords. Once organizations realized that SSIDs could be easily sniffed, they started using WEP. But even with WEP, the clients and users still were not uniquely identified and everyone shared the same key.

Now that WPA2 should be the de facto Layer 2 security standard for the enterprise, user or token-based authentication is required before encryption keys are given and access granted. The client-side interface into this authentication is called a supplicant--the entity that seeks to be authenticated by an authenticator, which is typically an access point. Supplicants contain a vendor's implementation of EAP types and security standards, so they are closely tied to the encryption capabilities of the card and driver. If a card doesn't support AES, for example, there will be no support for WPA2 using CCMP within the supplicant.

Microsoft Windows 2000 and XP, when properly patched, include a reasonably complete supplicant that ties tightly into the operating system. Windows XP supports EAP-TLS, PEAPv0/MSCHAPv2, and PEAPv0/EAP-TLS. MacOS X 10.3 offers a wider range of support: EAP-TLS, EAP-TTLS, LEAP, PEAP, and EAP-MD5.

If the operating system doesn't offer the supplicant you need or want, you can turn to the client software that comes with your wireless card, but you'll have to give up Windows zero-config (that could be a good or bad thing!). Cisco has an excellent supplicant in its ACU and ADU applications (different client software for different cards) but is slightly biased toward its proprietary EAP types. Intel, which has lately dominated the built-in wireless market with its Centrino-enabled laptops, offers its Intel ProSET software. It has a tight relationship with Cisco via the CCX (Cisco Compatible Extensions) program, so Intel supports all of Cisco's EAP types. Most of the other client card vendors have licensed their supplicant support from a third-party vendor, such as Meetinghouse, so expect strong EAP support even if the GUI doesn't look the same.

If you have a heterogeneous set of client cards or if you need enterprise management of your wireless configuration that isn't available through Windows domain policies, you'll need to turn to third-party supplicant and RADIUS vendors, such as Meetinghouse and Funk Software (now owned by Juniper). They offer well-rounded EAP support, custom-built and automated client and certificate distribution, and cross-platform support, but they come with a fat price tag.

There are also a range of open-source supplicants -- wpa_supplicant, Open1X, SecureW2, WIRE1x, and Xsupplicant. They offer a wide range and mixture of operating system and EAP support, so you'll need to look carefully.

It's easy to drown in acronyms when examining the different EAP types. We're here to help.

First, most EAP types include two parts: an outer and an inner authentication type, separated by a forward slash--such as PEAPv0/EAP-MSCHAPv2. The outer authentication type usually establishes a protected communications channel with the RADIUS server at the access point. Once that's established, the inner authentication type passes the user's credentials to the RADIUS for authentication.

There are five EAP standards certified by the Wi-Fi Alliance-- EAP-TLS, EAP-TTLS/MSCHAPv2, PEAPv0/EAP-MSCHAP-v2, PEAv1/EAP-GTC, and EAP-SIM--but there are some variants that deserve further explanation. Some are appended with a slash that indicates the inner authentication type.

Although not explicitly stated in the standards draft, EAP-TLS (Transport Layer Security) is the de facto EAP type for 802.1X. This is the same TLS that has replaced SSL in Web browsers to secure Web transactions using certificates. It's also the original EAP authentication protocol and arguably the most secure, but suffers from the requirement of server- and client-side certificates. Despite the PKI maintenance challenge, this EAP type is available across most platforms and in almost all supplicants.

EAP-TTLS adds a twist to the previous EAP type by creating a tunnel in which to exchange user credentials without the use of client-side certificates and the risk of a man-in-the-middle attack. It supports inner authentication types such as PAP point or MSCHAPv2. While the use of server-side certificates eliminates the burden of deploying a full PKI, acceptance has been limited. It was developed early on by supplicant and RADIUS vendor Funk Software (now owned by Juniper) and Certicom, but Microsoft and Cisco dealt that standard a blow when they introduced PEAP to the market, and neither their supplicants nor RADIUS servers support EAP-TTLS. The most common variant is EAP-TTLS/MSCHAPv2, which means that the password would need to be stored in some kind of MSCHAPv2 authentication store found, naturally, in Microsoft directories.

The next EAP type is PEAP, which comes in two versions. The first is the Microsoft favorite, PEAPv0/EAP-MSCHAPv2. Just like EAP-TTLS, it doesn't require client-side certificates, but more important, it's freely available on Windows 2000 and XP as well as Cisco's supplicant. Microsoft also provides IAS (Internet Authentication Services, Microsoft's RADIUS server) for free in Windows 2000 and 2003.

The second version is used by Cisco: PEAPv1/EAP-GTC. This one does not work with Microsoft's IAS, but is supported by the other popular RADIUS servers and supplicants and, of course, in Cisco's authenticator, ACS and wireless clients.

Cisco's two other EAP types are LEAP (Lightweight EAP) and EAP-FAST. LEAP is similar to EAP-TTLS but does not require a certificate. The disadvantage is that LEAP type requires Cisco access points for their keying material. Although LEAP isn't natively supported in the Microsoft Windows supplicant, it is found in Cisco's clients and Intel's Centrino line. The biggest problem with LEAP is that the user name is transmitted in the clear during the authentication exchange, and Joshua Wright's asleap attack, available for more than a year, has proven that dictionary-based attacks on passwords are trivial.

Cisco subsequently introduced EAP-FAST, which is considered much more secure and is available in the public domain, but only a few supplicants and RADIUS servers support it.