InteropNet Labs

3 in a series

What are Your EAP Authentication Options?

After you've decided to use 802.1X authentication for your wireless network, you have to make one of the most difficult and important decisions regarding its deployment. You must decide which EAP authentication mechanism you will use.

Although EAP supports a bunch of authentication methods, only four are commonly used. They are: MD5, a one-way authentication of supplicant to network using passwords; Cisco's proprietary username-based LEAP; TLS, which uses PKI-issued (public key infrastructure) digital certificates for strong mutual authentication; and, TTLS and PEAP, which combine server-side certificates with some other authentication such as passwords.

Not every supplicant supports every authentication method defined in 802.1X. Not every RADIUS/EAP server supports every method. And not every access point supports all methods. Your choice of EAP authentication method, then, drives everything else in your network.

Summary of Common EAP Authentication Methods

Method	Description of Most Common Implementation	Authentication Attributes	WEP Key Generated?	Deployment Difficulty	Wireless Security
MD5	Challenge-based password	One-way authentication	NO	Easy	Poor
TLS	Certificate-based two-way authentication	Mutual authentication	YES	Hard	Best
TTLS and PEAP	Server authentication via certificates; client via other method	Mutual authentication; identity hiding	YES	Moderate	Better

What is the MD5 Authentication Method?

The MD5 authentication method is the simplest one available to wireless LAN users, and support is required in the EAP standard. However, the insecurity of MD5 in a wireless environment is so blatant, that some wireless vendors have chosen not to allow MD5 as an authentication method. You may choose to use MD5 in a wired 802.1X environment, but we strongly recommend that you do **not** use MD5 in a wireless environment. It provides poor security. With MD5 authentication, the authenticator sends a challenge to the supplicant: some string, along with a serial number. The supplicant proves it knows the password by hashing the challenge, the string, and the password together and then sending the information back.

Challenge-based authentication schemes, like MD5, were designed to counter the insecurity of schemes like PAP (Password Authentication Protocol), which actually send the username and password in the clear across the wire. With MD5 (or CHAP in traditional PPP), the password doesn't pass across the wire. Instead, the supplicant "proves" that it knows the password. There are three reasons that MD5 is inappropriate wireless authentication.

First, MD5 requires that user passwords be stored in a way that lets the authenticator get at the original plain-text password. You'll sometimes hear this referred to as "reversibly encrypted." This opens up the possibility of someone **other** than the authentication server getting access to the file of passwords.

Secondly, MD5 **only** authenticates the supplicant. It does nothing to authenticate the authenticator, the wireless access point. Since wireless is especially vulnerable to impersonation, this is the major problem. Whereas impersonating a dial-up access server on the other end of a phone line is fairly difficult, impersonating wireless just means getting within a couple hundred feet of the supplicant. This lack of mutual authentication is the reason some wireless vendors have chosen not to allow MD5.

Thirdly, MD5 does not create a WEP session key. Ideally, immediately after authentication, the wireless client and access point would jump into WEP-encrypted communications, which reduces the risks of eavesdropping, impersonation, or data corruption by a hostile attacker. Other authentication methods, such as TLS and TTLS, support this but MD5 does not and therefore this limits its usefulness in the wireless world.

What is the TLS (Transport Layer Security) Authentication Method?

EAP-TLS is an IETF-standardized authentication method based on the same protocol used for secure Web traffic via the SSL (Secure Sockets Layer) protocol. SSL, as initially developed by Netscape Corp. for use with its Web browsers and servers, is the protocol used by Web browsers and servers to negotiate an encrypted connection. When you use an https:// link, http-over-SSL is invoked and authentication takes place, automatically.

There are very few differences between SSL version 3 (which is supported by all current versions of Web browsers) and TLS. Most people think of SSL (or TLS) largely in terms of the result: an encrypted session between your browser and the Web server. But as part of setting up the session, SSL starts with an authentication phase, and that's what is being used in any EAP-TLS operation.

TLS authentication within EAP is very simple. You take the TLS session establishment dialog between the supplicant and the authentication server and pack each TLS message inside of an EAP-TLS packet. When the TLS authentication dialog succeeds, the authenticator is informed and access to the network is granted.

Although TLS has actually set up an encrypted channel between the authentication server and the supplicant, this channel is not used. The supplicant wants to talk to the authenticator---typically an access point---not the authentication server. So, the keying material created during the TLS session establishment is sent by the authentication server to the authenticator using a RADIUS message. Then, the supplicant (which already knows the TLS-established secret key) and authenticator use that key for WEP encryption.

In EAP-TLS, certificates are used to authenticate the authentication server to the supplicant, and to authenticate the supplicant to the authentication server. The authentication server starts by sending its digital certificate to the supplicant. The most common authentication used today on the Web with SSL is one-way authentication - a server sends its certificate to your browser to prove its identity. However, with TLS-EAP, you are more interested in mutual authentication so that you can protect your network against man-in-the-middle attacks. Because the certificates are sent over the air, EAP-TLS does not hide the identity of clients from eavesdroppers.

The advantages of EAP-TLS make it a preferred authentication method. Both wireless client and access points are strongly authenticated using digital certificates. As a side effect, a per-session WEP key is set up, and the client can be re-authenticated and re-keyed as often as needed without inconveniencing the end user at all.

The problem with EAP-TLS is that it requires that clients hold digital certificates. While many enterprises have deployed a PKI infrastructure to handle certificates, many other enterprises are not ready for that. Some have also decided that other authentication systems, such as token-based authentication, align more closely with their business models and security policies.

What are the TTLS (Tunneled TLS) and the PEAP (Protected EAP) Methods?

Enterprise users that want the security of TLS, but have legacy authentication methods or token-based authentication methods will probably choose TTLS or PEAP for their EAP method.

EAP-TTLS and PEAP are essentially an extension to EAP-TLS. EAP-TTLS authentication uses certificates and EAP-TLS to authenticate the server only and establish an encrypted tunnel. Then, within that tunnel, the client authenticates to the server, typically using a simpler method, such as username/password or token card. EAP-TTLS and PEAP maintain similar security properties to TLS like mutual authentication and a shared secret for session WEP key. In fact, EAP-TTLS and PEAP are actually better than EAP-TLS in at least one way. An eavesdropper can't even see the user identity, because it is sent only after the TLS tunnel is established.

EAP-TTLS and PEAP accomplish this by packing another authentication protocol inside of the TLS tunnel when it comes time to authenticate the user. That's where the "Tunneled" in "Tunneled TLS" comes from. With TTLS, the network manager has the option of using a very simple tunneled authentication protocol, such as clear-text passwords or challenge-response passwords, or a more advanced technique, such as token-based authentication. With PEAP, there are fewer options: the tunneled authentication method is EAP itself, meaning that you can only use an EAP-defined method for authentication. Some network managers who wish to use simple password authentication won't like PEAP, because there is no EAP method to support a simple username/password.

The main drawback of EAP-TTLS and PEAP is that the IETF has not yet standardized which approach is best for environments where digital certificates are not available. The lack of a standard makes choosing one of these two approaches now a little risky: you could end up in a dead end.