InteropNet Labs

4 in a Series

Cooking up 802.1X Successfully

Any 802.1X deployment requires that you acquire and install four basic ingredients.

1) **Supplicant software** or hardware runs on the device that requires authentication.
2) An 802.1X-compatible **network adapter** is also required in the end user's device.
3) The supplicant needs to talk to an **authenticator**, such as a wireless access point.
4) The actual authentication decision is made via an **authentication server**, which is normally a RADIUS server that has been extended to support EAP (Extensible Authentication Protocol).

If you want to know more about 802.1X, see our other white papers on 802.1X in the Interop iLabs.

To select these four parts properly, you first have to answer one major question: "**How am I going to authenticate the users?**" Start with that question, and then work your way from the backend out towards the user end.

1st: How am I going to authenticate the users?

In other words, what EAP authentication method are you going to choose? You can read more about these in our white paper "What are the EAP authentication methods?" Although there are dozens of methods, the four currently available standards-based implementations are: MD5, TLS (Transport Layer Security), TTLS (Tunneled TLS), and PEAP (Protected EAP).

MD5 is adequate for authentication only in some circumstances, for example where your primary concern is authentication and not eavesdropping. If you are going to go to the trouble of setting up an 802.1X deployment, you probably don't want to use MD5. The security there is hardly worth the effort.

Most network managers will prefer the added security (including encryption) that comes with TLS , TTLS or PEAP. The criteria for choosing between these schemes is simple: do you have user-based digital certificates, or are you willing to deploy certificates for all your users? If the answer is "no," then you can't use TLS. But if you are willing to build a PKI to help authenticate users, then TLS is the easiest choice. Certificates and PKI aren't the major nightmare they were when first introduced in 1993, but it's still not something you enter into lightly. But if you have PKI already rolled out, or were looking for a good pilot application, TLS authentication for wireless access is a good approach. Companies with all-inclusive Windows Active Directory deployments will find TLS very easy because of the high-level of built-in support for certificates which Microsoft has provided.

If you don't want to do certificates, or if you prefer token-based authentication or integration with a legacy user/password database (such as an LDAP server or Unix /etc/passwd database), then you want TTLS or PEAP for your authentication method. Our white paper on *Comparing PEAP and TTLS* will help you understand the choices and issues. If you use TTLS or PEAP, you also need to select an **inner** authentication method, such as MS-CHAP.

2nd: How do I pick an authentication server?

You don't have a huge number of options. There are less than a dozen authentication servers that support EAP at all. However, remember that RADIUS servers---the norm for EAP-based authentication---often proxy off each other. So, for example, you might have an EAP-aware server that doesn't do much more than re-transmit queries to some other existing server somewhere. For example, if you use token-based authentication already via RADIUS, you can add an EAP/RADIUS server which simply gateways between the 802.1X (EAP) world and your existing RADIUS server. You don't necessarily have to wait until your token-based authentication vendor enables their RADIUS server for EAP.

The things you should ask to help narrow the field of products are:

1) Does the server support my EAP authentication method and inner authentication method (for TTLS or PEAP)?
2) Does the server support my user database (*e.g.*, LDAP server or Windows 2000 user database)?
3) Does the server run on my platform of choice (typically Unix or Windows)?

The table below summarizes Authentication Servers we know about (as of August, 2002)

Vendor	EAP Authentication Methods Supported
Funk Software	MD5, LEAP, TLS, TTLS (PAP, CHAP, MD5), PEAP (MS-CHAP)
Interlink Networks	MD5, LEAP, SPEKE, TLS, TTLS (PAP, CHAP, MS-CHAP), PEAP (MS-CHAP)
Hewlett-Packard	MD5, LEAP, TLS, TTLS (PAP, CHAP, MD5), PEAP (MS-CHAP)
Microsoft	MD5, TLS, PEAP (MS-CHAP)
Meetinghouse	MD5, TLS, TTLS
Cisco Systems	MD5, LEAP, TLS, PEAP
FreeRadius	MD5, TLS

3rd: What supplicant should I use? Which client wireless cards?

The supplicant runs on the client workstation, so you have to find one that is both compatible with your operating system and supports the EAP authentication method you have chosen. Again, at least right now, you don't have a lot of choices.

Generally, supplicant software is vendor-independent and will work with any wireless card. The one exception is Cisco's supplicant, which, understandably, is only licensed for use with Cisco's cards. The table below should give you a good handle on what supplicants work with which operating systems and authentication methods.

Vendor	O/S Supported	EAP Authentication Methods Supported
Microsoft	2000/XP	MD5, TLS, PEAP, TTLS (with plugin)
Meetinghouse	98/ME/NT/2000/XP;MacOS X;Linux	MD5, TLS, PEAP, TTLS
Funk	98/ME/2000/XP	MD5, TLS, PEAP, TTLS
Free1X.ORG	Linux/BSD	TLS (without key setting)
Cisco	95/98/NT/2000/XP;MacOS;Linux	MD5, LEAP, TLS, PEAP

Wireless cards are pretty straightforward (in the Windows world), as long as your card supports the most recent NDIS specification in its driver, which is needed for some of the OID (Object Identifiers) definitions. All of the enterprise-oriented cards we tested worked fine, but some of the "no-name" or generic wireless cards may not be compatible. Test before committing to a large purchase.

4th: What Wireless Access Point (AP) should I use?

This is fairly simple. The wireless AP is not supposed to be very aware of the EAP conversation. It largely unpacks and repacks packets from the Supplicant to the Authentication Server. However, we have discovered two areas where there are incompatibilities. Some AP vendors have intentionally blocked MD5 authentication, as too insecure for wireless use. Others have blocked some of the newer authentication methods, such as EAP-TTLS and PEAP. Check with your AP vendor to be sure that the EAP method you selected is compatible---and run a quick test.

Finally: Putting it all together

Start by getting your EAP/RADIUS server(s) installed. This will probably take more time than anything, since this can be complex, particularly if you are authenticating against an LDAP database or some token-based system. If you have a RADIUS test client, make sure you test your server before going any further. Most commercial RADIUS servers ship with a small application you can use to test the authentication. This won't test the EAP path, but it's a good start.

Drop in an access point. Install your wireless client card. Verify that the AP works before committing any EAP shenanigans and your client can connect to the wired side of the network without problems.

Link the AP to the RADIUS server with a shared secret, and enable EAP authentication. Make sure you get your UDP port numbers straight---half the RADIUS products in the world use 1645; the other half use 1812.

Finally, get your supplicant going. Since you're probably dealing with Windows, be patient and be prepared to call the supplicant vendor for support. Most of these products are fairly new, and you can expect that the configuration GUI is going to be a little rough around the edges.

Once you get your supplicant talking to the wireless client card and authenticating via the access point to the EAP server, all that's left is documentation, training, and that little victory dance.