Wireless security with 802.1x and PEAP

Steve Riley
Senior Consultant, MCS Trustworthy Computing Services
13 January 2003

Wireless LAN authentication

Wireless networks present a number of security challenges—not only to authentication but to also to data privacy and integrity. Because the air isn't a controllable physical medium, there's no way physically to prevent attempted interception or interjection of data. Extremely strong authentication and encryption are requirements for any secure wireless network.

Certain flaws[1] in the implementation of RC4 encryption in WEP (wired equivalent privacy), the encryption method used in 802.11 wireless networks, permit an attacker to determine the encryption key and gain access to a wireless network after capturing some data and passing it through commonly-available key generation programs[2]. Various standards bodies are working to develop an alternative to WEP. For now, however, a common way to work around the problem is to deploy a technology that rotates encryption keys at intervals that prevent an attacker from gathering sufficient data to crack a key. An authentication protocol called 802.1x, originally intended for wired LANs, works quite well for this purpose, too.

Traditionally, access to network ports—whether wired or wireless—was controlled by using MAC access lists. MAC-based access control doesn't scale well in large networks because it requires significant administrative workload—each time a user gets a new computer or replaces a network card, the MAC access list on the switch or wireless access point requires updating.

802.1x

802.1x[3] is a port-based access control method defined by the IEEE that provides a better way to control access to network ports. The specification allows for flexibility in choice of authentication methods; the most common approach for wireless is to use EAP[4] (extensible authentication protocol), a framework for specifying particular authentication methods. The method is chosen by the supplicant and the authenticator at authentication time.

An EAP client ("supplicant") contacts an authenticator (a wireless access point or a VPN server, for example), which challenges the client for authentication information. The authenticator receives this information and passes it to an authentication server (usually a RADIUS server) for validation. At this point, because the supplicant hasn't yet been authenticated, no other communications from the supplicant are allowed. The authentication server validates the logon request and returns either an accept or reject message to the supplicant. If the logon is accepted, the authentication server generates a WEP key specifically for that supplicant and sends it through the access point to the supplicant; the supplicant is allowed access to the network behind the authenticator.

802.1x authentication, whether using passwords with PEAP or client certificates with EAP-TLS, requires a certificate on the authentication server. This eliminates spoofed access points and authentication servers, another WEP vulnerability. The authentication server presents its certificate to the supplicant; the supplicant can verify the validity of this certificate and know that it's communicating with authorized access points and authentication servers.

---

[1] "Security of the WEP Algorithm," http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html

[2] For example, *WEPCrack*, http://sourceforge.net/projects/wepcrack/

[3] "802.1x – Port Based Network Access Control," http://www.ieee802.org/1/pages/802.1x.html

[4] RFC 2284, "PPP Extensible Authentication Protocol," http://www.ietf.org/rfc/rfc2284.txt

Understand that EAP itself does not specify any particular encryption mechanism for the exchange of authentication information. Various extensions to EAP address this problem and are described below.

Logging on to a Windows domain

In a Windows network, there are actually two logons that occur. When a computer is booted, the computer itself logs onto the domain using a machine account. Performing machine authentication allows the domain to authenticate the device which is necessary for execution of machine group policies and software installation settings. At the end of this phase, the user logon dialog appears, and the user logs on with his/her credentials. Now the user has authenticated to the domain and user group policies will execute.

An access token containing the domain's SID, the user's SID, and SIDs for all groups the user belongs to is created and cached on the local computer; each time the user accesses a resource, the token is presented and the resource compares the SIDs to its access control list to determine whether the user is authorized. Note that caching this credential is allowed only if the computer is a member of the same domain as the user; if the computer is a member of a different or no domain, the token isn't cached. The user will be prompted for his/her ID, password, and domain each time he/she accesses a domain resource.

Requirements for wireless authentication

This two-logon sequence needs to occur regardless of the physical network type—wired or wireless. Because machine logon and user logon are identical processes, the implementation of EAP in Windows has been designed to accommodate this requirement. Two flavors of EAP are available in Windows XP: EAP-TLS and EAP-MD5; a third is available in Windows XP service pack 1: PEAP (protected EAP).

- ***EAP-TLS*[5].** TLS (SSL) is the encryption method used to protect the EAP exchange. In Windows, TLS uses certificates for machine logon and for user logon. The computer first authenticates to the network using a machine certificate; this allows for computer policies in Active Directory and SMS packages to be applied before a user logs on. When a user then does log on, he/she authenticates with a user certificate whose subject alternate name contains the user's Active Directory UPN. At this point, the wireless connection is completely authenticated, and user policies are applied. A public key infrastructure is required for EAP-TLS.

- ***EAP-MD5.*** MD5 uses CHAP, a challenge-response process for the user authentication portion. Unlike TLS, MD5 has no ability to create an encrypted session between the authenticator and the supplicant; therefore, password hashes are transmitted in the clear. For this reason, EAP-MD5 is specifically disabled as an authentication method for 802.1x wireless.

- ***PEAP*[6].** Similar to EAP-TLS, PEAP protects the authentication exchange carried inside EAP. But unlike EAP-TLS, PEAP doesn't require the use of TLS (although TLS is certainly allowed); any secure authentication exchange mechanism can be used. In Windows, PEAP supports MS-CHAPv2 in addition to TLS. PEAP-MSCHAPv2 uses machine and user credentials for authentication. MSCHAPv2 uses rotating keys to encrypt both machine and user password hashes. Certificates are still required for the authentication server (IAS) and are used for server-to-client authentication; credentials, carried inside MSCHAPv2, are used for client-to-server authentication. Machine and user certificates are no longer necessary.

[5] RFC 2176, "PPP EAP-TLS Authentication Protocol," http://www.ietf.org/rfc/rfc2176.txt

[6] "Protected EAP Protocol," ftp://ftp.rfc-editor.org/in-notes/internet-drafts/draft-josefsson-pppext-eap-tls-eap-05.txt

More on protected EAP with passwords and MS-CHAPv2

PEAP is available with the following combination of products:

- Windows XP service pack 1 on the clients
- Windows .NET Server on the IAS (RADIUS) server—.NET RC1 includes PEAP

The IAS Server can belong to a Windows 2000 domain—Windows .NET Server isn't required on the domain controllers.

When a wireless client boots, the client first performs an 802.11 association. But because the client hasn't yet authenticated, the access point blocks the client—it has no ability communicate on the network behind the access point. Next, PEAP uses the machine account to authenticate to the wireless access point; if the logon is valid, the RADIUS server generates a unique WEP key and sends it to the client. The computer logs onto the domain and can access only those network resources allowed by the computer account. The user then logs on with his/her credentials; PEAP uses the credentials to authenticate the user for wireless access and completes the logon. User policies are finally applied, and the client has regular access to the network.

If a user's password expires between logons, PEAP will properly handle the situation. Because the computer can still log onto the domain, PEAP authenticates the machine, then prompts the user to change his/her password. As with a standard LAN connection, the password change is communicated to the domain controller, the cached credentials are updated, and the user is logged on.

Like any EAP-based 802.1x authentication method, PEAP eliminates the need to hardcode WEP keys in the wireless clients. Because PEAP relies on RADIUS to authenticate both the computer and the user, PEAP follows the 802.1x normal practice of expiring WEP keys periodically (the expiration interval is configurable) and forcing new WEP keys if a roaming user moves between access points. The periodic rotation of WEP keys is currently the only work-around that overcomes flaws in WEP's implementation of RC4 encryption.

802.1x support is required on the wireless access points. Note that this is simply generic EAP support; the AP is unaware of the particular kind of EAP negotiated between the client and the RADIUS server.

Configuring wireless LAN authentication

Several steps are required to deploy PEAP-based authentication for wireless:

- Install Windows .NET Server, including IAS (Internet authentication services)
- Obtain a computer certificate for each IAS server
- Register each IAS server in Active Directory
- Configure logging on each IAS server
- Add the wireless access point as a RADIUS client and configure each AP for EAP
- Create a remote access policy for wireless access
- Configure XP clients for PEAP authentication

Certificates

A computer certificate is required on each IAS server. This certificate must be issued from an authority that clients trust—that is, the authority's public root must be in the client's trusted root certificate store. The recommended approach is to configure automatic machine enrollment in the default domain policy. Each computer that logs onto the domain will

receive a computer certificate. Alternately, using the certificates MMC snap-in, individual computers can be enrolled with computer certificates.

Registering IAS with AD

IAS servers must be registered with AD so that IAS can query the user account database. This process also adds the IAS server to the domain's pre-existing *RAS and IAS Servers* security group. Domain administrator privileges are required. See the help for step-by-step instructions.

IAS logging

Logging is a critical component of any authentication infrastructure. IAS can log to a text file and to a SQL Server database. Text files are appropriate for test; production IAS servers should log to a SQL Server database instead. See the help for step-by-step instructions.

Adding the WAP as a RADIUS client

RADIUS clients must be added to the IAS configuration. For each wireless access point:

- Add its IP address
- The **Client-vendor** is **Microsoft**
- Enter the correct shared secret—all WAPs can use the same shared secret, or each WAP can have its own
- EAP requires the use of message authenticator attributes

Configuring security on the WAP. The WAP must support 802.1x authentication to work with PEAP (if an AP works now with LEAP, it will work with PEAP, too; the access point itself isn't involved in the selection of the particular EAP method). In the AP's security settings, enter the IP address of the IAS server, enable EAP authentication, and enter the RADIUS shared secret. Also configure a WEP key—this key is used only when the AP sends broadcast information to all wireless clients; it isn't used for individual wireless clients associated with the AP.

Creating a wireless access policy

An IAS remote access policy allows wireless users to connect to the network. Delete the existing default remote access policies, and then create a new one for wireless:

- Add one policy condition with **NAS-port-type** matching two values: **Wireless – IEEE 802.11** and **Wireless – other**
- Grant remote access permission

In the policy, configure this profile:

- Deselect all authentication methods
- Configure an EAP method—the EAP type is **Protected EAP**; the certificate is the computer certificate obtained earlier; fast reconnect is enabled
- Allow only **Strongest encryption (MPPE 128 bit)**
- Modify the attributes (**Advanced** tab)—add **Ignore-user-dialin-properties** and set to **True**; delete **Framed-protocol**

Because this policy is for wireless access, not VPN or RAS access, the dialin properties of a user account should be ignored.

Configuring XP clients for PEAP

Windows XP's zero configuration for wireless makes using PEAP very easy. Service pack 1 is required for PEAP. After applying SP1:

- Open the properties of the wireless NIC
- Select the SSID of the PEAP-enabled wireless network
- Modify the authentication method to use PEAP; turn on fast reconnect
- Optionally, choose to validate the IAS server's certificate. . Choosing this will instruct the client to validate the entire certificate chain, including any intermediate certificate authorities.

Note that a wireless client must already be a member of the domain. Non-members will be denied access; furthermore, it isn't possible to join the domain over wireless. A wired connection is necessary for domain joins.

Certificates and smart cards

Deploying PEAP solves an immediate need—to implement secure wireless without using certificates. It also overcomes a problem in Cisco's proprietary LEAP protocol: its inability to process password changes. PEAP has the benefit of not requiring changes in user behavior. Nevertheless, even over PEAP, passwords are still plagued with all their attendant problems: people forget them, people write them down, people share them, people try to circumvent password complexity policies. Passwords are a low-trust method of authentication.

At some point in time, new or changed business processes might dictate that higher levels of trust in authentication are required. Certificates provide this additional level of trust. A digital certificate binds a private key to a strongly-authenticated identity; certificates replace IDs and passwords and are associated with Active Directory user accounts. The trust that a certificate asserts is only as strong as the authentication procedure used in the enrollment process. Physical appearance with photo identification before a certificate officer is, of course, a very strong procedure. Incorporating certificate issuance into the new-hire process is also quite strong.

For some time, though, managing digital certificates has been very difficult. Even though a certificate asserts the identity of a human user, the certificate itself is stored on a computer's hard drive. For users who move from computer to computer, porting the digital certificate has proven all but impossible. This one problem has been the failure of many PKI deployments. So for a number of reasons, storing certificates on computers is not preferred:

- A certificate asserts the identity of a person; storing this on a computer (rather than the person) weakens the "something you possess" factor because a stolen computer contains the stolen certificate.
- Managing certificates stored on computers is very difficult; users will need to remember to export them whenever operating systems are reinstalled or upgraded.
- Computer-stored certificates are impossible to manage when users move from computer to computer.

Smart cards are the logical answer to this problem. Smart cards accompany humans, not computers. Smart cards are also (should be) stored separately from the computer itself. When a user wants to log on to a network, the user removes his/her smart card from a wallet or purse, inserts it into the smart card reader, and logs on. Windows 2000 and XP fully support smart cards for network access. Because the keys and certificates are stored on the card rather than the computer's hard drive, it's very easy for a user to move to another machine: the user simply removes the smart card (which automatically

forces a log off), goes to another computer, inserts the card, and logs on. At no time are user credentials stored on the local machines.

Certificates on smart cards combine the high trust of stronger authentication with the ease of mobility users expect. But the cards themselves come with certain social issues that require thought. Despite this now being the only way to access network resources, smart cards have no stored value, and therefore aren't treated with the same care as a credit card, a public transit pass, or even an employee identification card. A common mistake users make is to leave their smart cards in their desks or to carry them in the same briefcases as their laptops. Smart cards should always be kept on the person when not in use. One way to encourage this is to roll out new employee ID badges that are also smart cards. Users now don't have to deal with separate cards, they're accustomed to carrying ID badges in wallets or purses, and the smart card now has a value attached to it. Indeed, this is the approach Microsoft has taken. Over the course of one year, all 55,000 employee ID cards are being replaced with combined smart cards/proximity badges.

Both PC Card and USB readers are available; to avoid having to install additional drivers or cryptographic service providers, choose either of these readers (whose drivers and CSPs are included in Windows):

- Gemplus GemSAFE
- Schlumberger Cryptoflex

Smart cards are best deployed in groups. An administrator, with a smart card enrollment station, should enroll (and optionally activate) say 50 users' smart cards and then notify these users that their cards are ready. Each user should physically appear to receive his/her smart cards; don't deliver them via inter-company mail. After the first group of users has installed their readers and verified that smart card access works, the next group of users should be processed.

Activation involves assigning a PIN to the smart card. If the administrator performing the enrollment also activates the card, the administrator will know the user's PIN until the user decides to change the it; however, the user needs to perform no work before using the smart card for VPN authentication. If users will do their own activation, it's best to have a dedicated "activation computer" in the same location where the pick up their cards. A user inserts the card in the computer's reader, starts the activation program, and assigns a PIN of his/her choice to the card.

No special readers are required for enrollment or activation—any smart card reader can be used for these purposes.

Adding smart cards and certificates to the wireless network. Most deployments of PEAP will use EAP-MSCHAPv2 for password-based computer and user authentication to the network. PEAP also can use EAP-TLS, the specification for certificate-based authentication. When smart cards are eventually deployed, simply add EAP-TLS as an EAP type to the existing PEAP access policy in IAS. If computer and user certificates are available, PEAP will negotiate EAP-TLS; if the certificates aren't available, PEAP will negotiate EAP-MSCHAPv2. Once all users have received smart cards, EAP-MSCHAPv2 can be removed from the list of EAP types.

Certificate revocation lists are mandatory

Authenticating users with certificates (whether stored on smart cards or hard drives) and using the RADIUS authentication provider results in a logon sequence that performs a number of steps. For certificate-based authentication, a digital signature is executed by the client and verified by the server in the following manner:

1. The server sends the client a hash to sign; the server retains a copy of this hash
2. The client signs the hash:
    a. Encrypt the hash with the client's private key right there on the client
    b. Bundle the encrypted hash and the client's certificate into a PKCS#7 blob

    c. Send the blob to the server
3. The server verifies the signature:
    a. Decrypt the encrypted hash with the public key found in the certificate
    b. Does the decrypted hash match the original copy retained by the server? It should
    c. Is the certificate valid?
        i. Was the certificate issued by a CA that the server trusts? (build the certificate chain up to the trusted root)
        ii. Check whether the certificate has been tampered with (this is a cryptographic operation, not a boolean)
        iii. Check the validity period of certificate
        iv. Check whether the certificate has been revoked; AD will automatically check CRLs.
4. Assuming all is good, log the user onto the domain:
    a. Read the certificate's subject alternate name (SAN)—which contains a Windows AD UPN
    b. Contact the domain controller for the domain in the SAN
    c. Find the account whose UPN matches the certificate's SAN
    d. Obtain a Kerberos credential and supply it to the client

This entire communication chain is secured with TLS, following the specifications for EAP-TLS and EAP-TLS over RADIUS. Between RRAS and RADIUS, EAP-TLS is encapsulated inside RADIUS. The RADIUS server receives this, decapsulates the EAP-TLS, and performs the logon process as described previously. If the logon sequence succeeds, the RADIUS server sends an ACCESS-ACCEPT to the client, still within the EAP-TLS session. If it fails, RADIUS sends an ACCESS-REJECT.

A basic thing to understand. At this point, now, the server is satisfied that the client who executed the signature used its own private key—because the corresponding public key in the certificate was able to decrypt the hash—and that the CA is the one issuing DC certificates. Observe that trust in the certificate logon process rests entirely on the trust in the CA and the validity of the certificate.

Now consider the case in which a user's laptop is stolen. Even though the user receives a new laptop with a new certificate, that stolen certificate still exists. Without a certificate revocation list (CRL) to check validity, there's no way to prevent that certificate from logging on:

1. The stolen computer still has the private key and can use it to sign the hash
2. The stolen certificate will still be bound into the blob
3. The DC will successfully decrypt the hash
4. The DC will check the CRL—but if there’s no CRL, the user will get logged on

While Active Directory does store a copy of a user’s certificate and public key in the directory, this is used only for verifying e-mail signatures, not for authentication. There’s no explicit reference or link between an account and a certificate—a user’s own certificate doesn’t appear in **Certificate mappings** on the user’s account.

Unlike IDs and passwords, where trust is in the DC itself, trust lies elsewhere when using certificate logon. That trust is in the ability to verify whether a certificate is still valid—and the only way to do that is to have a CRL. This is the way X.509 is designed.

Therefore, *a CRL is a requirement for certificate-based logon to RRAS.* Additional information is available in the help and in the Windows 2000 Resource Kit. See http://www.microsoft.com/windows2000/en/server/help/sag_CS_CertRevoke.htm and http://www.microsoft.com/windows2000/techinfo/reskit/en-us/distrib/dscj_mcs_eako.asp for more information.