Internet Security
SS 2004

Certificate Extensions and Attributes Supporting Authentication in PPP and Wireless LAN

Daniel Schwarz

Overview:

1. Introduction

The PKIX working group, responsible for the internet draft this work is all about, was established in 1995. The intention of that working group was to develop internet standards that are needed to support a X.509 based PKI (public key infrastructure).
X.509 is some kind of standard for authentication and defines the content of a digital certificate. The expression itself comes from the X.500 specification on directory services. The directory services serve as something comparable to an electronic phonebook, where enabled applications can lookup included entities. Each entity has an identifying record or certificate and the format of that certificate follows the recommendation X.509 of the ITU (=International Telecommunication Union -> globally coordinates telecommunication networks and services).
The PKIX group not only profiles ITU PKI standards, but also develops new standards apropos to the use of X.509 based PKIs in the internet.

2. Basics

2.1. PPP

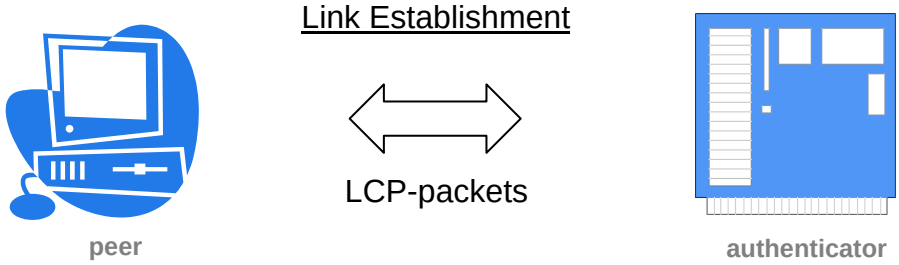
The Point-to-Point Protocol (PPP) is a standard method for communication between two hosts and is most commonly used for dial-up internet access. It is also used by some ISPs (Internet Service Providers) for DSL and cable modem authentication, in form of PPP over Ethernet.
PPP is part of the Layer 2 Tunnelling Protocol which is a core part of Microsoft's secure remote access solution for windows 2000 and beyond. It has also got an integrated error correction, so errors during transmission are automatically discovered. Another feature is the compression of the IP-header. Imagine Telnet and that it's possibly that there's only one letter transmitted with a message, so it would be nice if the IP-header is not 20 times as big as the message itself.
Very important is another part of PPP: the LCP (Link Configuration Protocol); it's responsible for the configuration, establishment and the clearing of a PPP-connection.
PPP evolved beyond its former and original use as a dial-up access method and is used all over the internet. One piece of PPP defines an authentication mechanism. With your dial-up internet access that's simply username and password. So PPP is used to identify the user at the other end of the line before giving him access.
In our times most enterprises want to do more for security than simply employing usernames and passwords for access, so a new authentication protocol, called the EAP (Extensible Authentication Protocol), was designed.

2.2. EAP

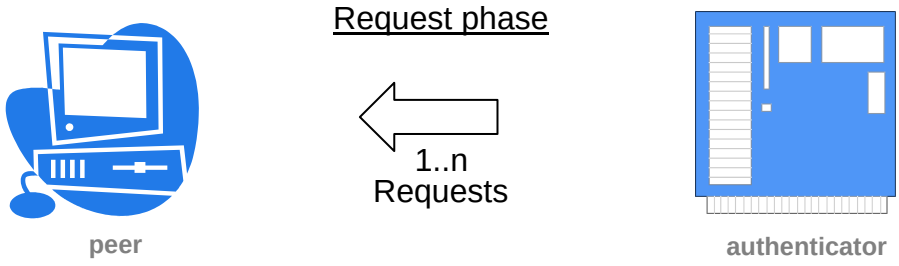
EAP sits inside of PPP's authentication protocol and provides a generalized framework for different authentication methods. It does not select a specific authentication method at the LCP-phase but rather postpones this until the authentication phase. So the end of the link requiring the authentication (the so called Authenticator) is able to request more information before determining the specific authentication method.
It is also permitted to use a "back-end" server which actually implements the various methods while the PPP authenticator merely passes through the authentication exchange.
The EAP protocol implements three important steps:

1) After the link establishment phase is completed, the Authenticator sends one or more requests to authenticate the Peer:

a)



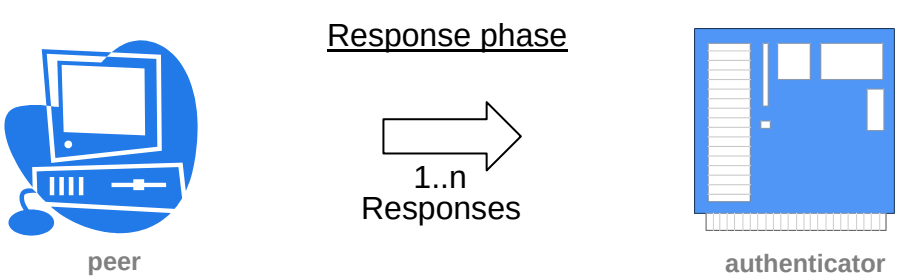
b)



The request has a type field to indicate what is being requested. Typically, the Authenticator will send an initial identity request followed by one or more requests for authentication information. But this initial identity request is not required and may be bypassed in cases where identity is presumed (e.g. for dedicated dial-ups, …).
Examples for these request types are the identity, a MD5-challenge, one-time passwords, generic token card, etc.

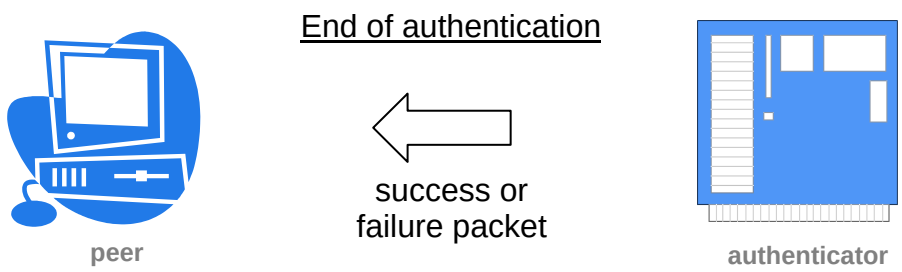
2) The Peer sends back a response packet in reply to each request. As with the request packet, the response packet contains a type field which corresponds to the type field of the request.

c)



3) The Authenticator ends the authentication phase with a “success” or “failure” packet.

d)



One advantages of the EAP protocol is that you can use multiple authentication mechanisms without having to pre-negotiate a particular one during the LCP phase. Also certain devices do not necessarily have to understand each request type and may be able to simply act as a “pass-through agent” for some kind of “back-end” server on a host. These “pass-through agents” only need to look for the success/failure code to terminate the authentication phase.
On the other hand there are also some disadvantages: the PPP implementation needs to be modified (the addition of a new authentication type to the LCP is required) and the focus is set on authenticating a Peer to an Authenticator. The problem is that the Peer doesn’t request any authentication from the Authenticator; this is solved with the EAP-TLS protocol (chapter 4.2.).

2.3. 802.1x

Said in simply words, 802.1x is a standard for passing EAP messages over a wired or wireless LAN. These EAP messages are packed in Ethernet frames without using PPP. 802.1x enables authenticated access to all IEEE 802 media (Ethernet, Token Ring, 802.11 WLAN, …). RADIUS (Remote Authentication Dial-In User Service) support is optional but it is expected that many IEEE 802.1x Authen-

ticators will function as RADIUS clients. 802.1x is used in situations where other protocols than TCP/IP are needed or the overhead and complexity of using PPP is undesirable.
802.1x defines three important terms:
a) Supplicant: user or client that wants to be authenticated
b) Authentication Server: actual server doing the authentication (typically a RADIUS server)
c) Authenticator: device in between (such as a WLAN access point)

The Authenticator can be simple and dumb and that's why it's ideal for WLAN access points because they only need little memory and processing power. The protocol in 802.1x is called EAP encapsulation over LANs (EAPOL). There are different modes operation; different possibilities how this protocol can be working. The most common one acts as follows:

1) The Authenticator send an EAP-Request/Identity packet to the Supplicant as soon as it detects that the link is active (e.g. the WLAN-client has connected to the access point).
2) The Supplicant sends back a Response and the Authenticator acts as some kind of "pass-through" device and the Response goes on to the Authentication Server.
3) The Authentication Server answers with several challenges such as a token password system; then the Authenticator unpacks this from IP and repackages it into EAPOL and sends it to the Supplicant. Different authentication methods will vary this message and the total number of messages. EAP supports client-only authentication and strong mutual authentication which would be appropriate for the wireless case. More about that in the chapter about EAP-TLS.
4) The Supplicant sends the several replies to the challenges.
5) The Authentication Server sends a Success message and now the Authenticator allows access to the LAN, possibly restricted based on attributes that came back from the Authentication Server. For example, the Authenticator might switch the Supplicant to a particular virtual LAN.
6) The Supplicant is able to get access.

2.4. X.509

X.509 is an ITU standard for PKI and specifies, among other things, standard formats for public key certificates. It's part of the hierarchical X.500 standard and thus assumes a strict hierarchical system of certificate authorities (CAs) for issuing the certificates. This is in contrast to "web of trust" models, like PGP, where everyone may sign keys of others. The X.500 system has never been fully implemented, so the IETF's PKIX have made updates to the standard in order to make it work with the more loose organization of the Internet.
X.509 usually refers to the X.509 v3 certificate specified in RFC 2459. In the X.509 system a CA issues a certificate binding a public key to a particular name. However as no real implementation of this standard exists, the binding is more usually betweens a public key and an email-address.

Now let's go to the certificate extensions. The extensions defined for X.509 v3 certificates provide methods for associating additional attributes with users or public keys. It's also allowed for communities to define private extensions to carry information unique to those communities.
Each extension in a certificate system is designated as either **critical** (system *must* reject the certificate if it doesn't recognize the extension) or **non-critical** (system *may* ignore the extension).

The "key usage" extension defines the purpose of the key contained in the certificate, e.g. encipherment. The usage restriction might be employed when a key that could be used for more than one operation is to be restricted. And this extension should be marked as critical.

The "extended key usage" extension indicates on or more purposes for which the certified public key may be used. It's used in addition or in place of the basic purpose indicated in the key usage extension. It may be marked as critical or non-critical.

If a certificate contains both a "key usage" extension and an "extended key usage" extension then both extensions *must* be processed independently and the certificate *must* only be used for a purpose consistent with both extensions. If there is no purpose consistent with both extensions, then the certificate *must not* be used for any purpose.

3. Certificate extensions and attributes supporting authentication in PPP and WLAN

3.1. EAP extend key usage values

Two new values are defined in the PKIX draft:
a) id-kp-eapOverPPP OBJECT IDENTIFIER ::= { id-kp 13 }
Indicates that the certified public key is appropriate for use with EAP in the PPP environment
b) id-kp-eapOverLAN OBJECT IDENTIFIER ::= { id-kp 14 }
Indicates that the certified public key is appropriate for use with EAP in the LAN environment

Inclusion of both values indicates that the certified public key is appropriate for use in either of the environments. It may be critical or non-critical.

3.2. WLAN SSID Public Key Certificate Extension

This extension is defined in the draft and is always non-critical. It contains a list of SSID's (Service Set Identifiers) also called Network Name. A SSID could be up to 32 characters and is configured in the

access point of a WLAN and also on all clients that want access to this network. This character string always stands unencrypted at the beginning of each packet.
If more than one certificate include an extended key usage extensions indicating that the certified public key is appropriate for use with the EAP in LAN environment then the list of SSIDs *may* be used to select the correct certificate for authentication in a particular WLAN.
It's a fact that SSIDs are unmanaged, so the same SSID can appear in different certificates that are intended to be used with different WLANs. You have to get some input about the right combination or another option is the trail-and-error method ("try the combinations until success"). By maintaining a cache of access point MAC addresses or authentication server identities with which the certificate has successfully authenticated, user involvement can be minimized.

3.3. WLAN SSID Attribute Certificate Attribute

If a public key certificate does not contain the WLAN SSID certificate extension you can use a so-called attribute extension. It acts the same way as the extension and is structure similar to a public key structure. The main difference is that the attribute certificate doesn't contain a public key. An attribute certificate may contain attributes that specify group membership, roles or stuff like that. A public key certificate can be considered to be like a passport: it identifies the holder, lasts for a long time and should not be trivial to obtain. An attribute certificate is more like an entry visa: it's typically issued by a different authority and does not last for a long time.
The attribute certificate contains a list of SSIDs and can be used to select the correct certificate.

4. EAP & 802.1x

4.1. EAPOL (802.1x)

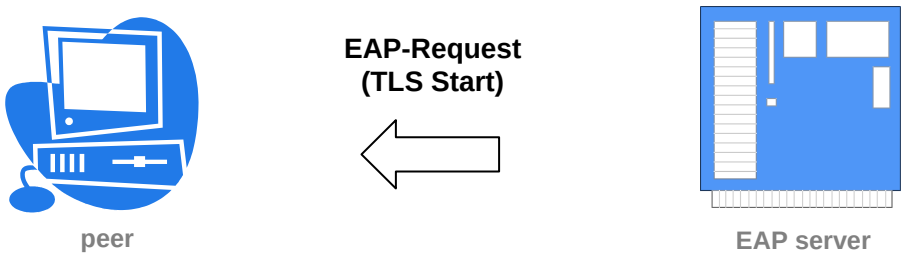
As heard in chapter 2.3 the authentication server sends several challenges to the supplicant and the supplicant sends an answer to each challenge. The existing problem is the missing mutual authentication. Therefore you can use the EAP-TLS to authenticate both the supplicant and the authentication server to each other.

4.2. EAP-TLS – mutual authentication

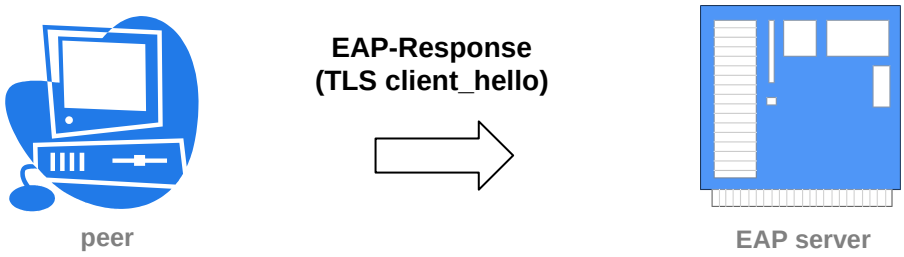
The following example is a common way how EAP-TLS is used in pratice:
The steps a) and b) are not described in detail. They simply handle the EAP-Request for identity (step a) and the EAP-Response (steps b) from the Peer.
From this point the Authenticator in the middle may act as a pass-through device, with the EAP packets received from the Peer being encapsulated for transmission to a RADIUS server or backend security server. To make it more simple on the following pictures there will be only an "EAP-server" instead of Authenticator + EAP-backend-server.

Let’s go on to the next steps where the EAP-TLS is starting.

c)

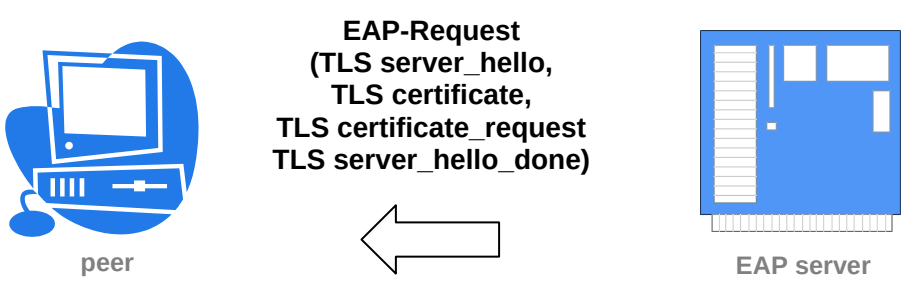


d)

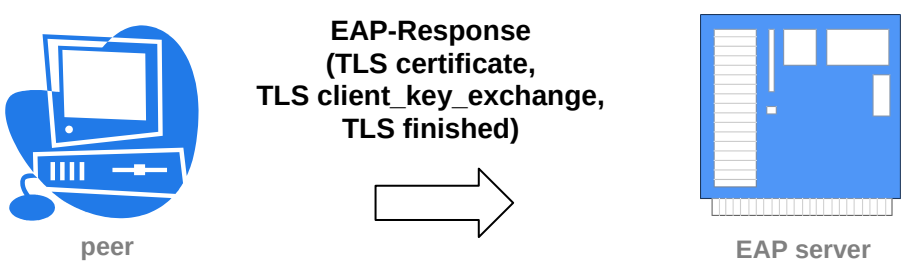


Step c) shows the start of the EAP-TLS protocol. An EAP-Request packet with EAP-Type set to EAP-TLS is sent. The starting-bit is set and it contains no data.
In step d) the Peer responds to the EAP server. The data field of this packet will encapsulate one or more TLS records in TLS record layer format, containing a TLS client_hello handshake message with the client’s TLS version number, a session ID, a random number and a set of ciphersuites supported by the client.

e)

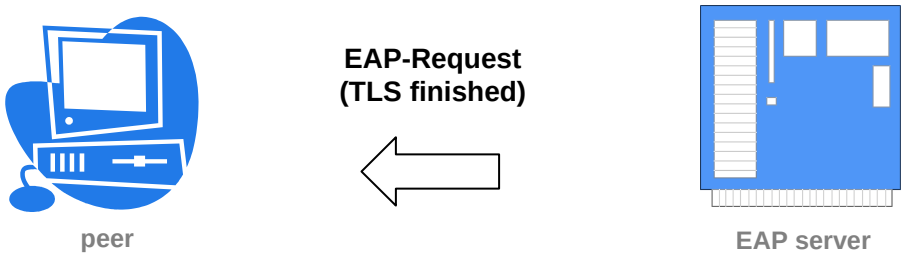


f)

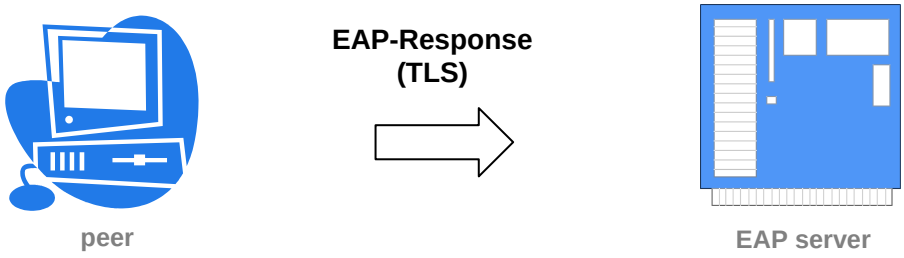


Step e) shows the next EAP-request packet. The data field of this packet will encapsulate one or more TLS records. These will contain a TLS server_hello exchange message, followed by TLS certificate, certificate_request and server_hello_done. The server_hello handshake again contains a TLS version number, another random number, a session ID and a ciphersuite. The session ID has to be the same as the ID from the client.
In step f) the EAP-Response is displayed. In this example the certificate message contains a signature public key (such as RSA). In this case a TLS server_key_exchange handshake message must also be included to allow the key exchange take place.

g)



h)



The steps g) and h) show the final “TLS finished” messages to indicate that the client_key_exchange key handshake messages are exchanged.
With the success-message in the next step the peer can gain access.

i)

4.3. EAP-Alternatives

1. EAP-MD5:

EAP-MD5 lets a RADIUS server authenticate LAN stations by verifying an MD5 hash of each user's password. This is reasonable choice for trusted Ethernets where there is low risk of outsiders sniffing or active attack. EAP-MD5 is not suitable for public Ethernets or wireless LANs because outsiders can easily sniff station identities and password hashes, or masquerade as access points to trick stations into authentication with them instead of the real deal (so called "man-in-the-middle"-attack)

2. LEAP (Lightweight EAP):

Cisco's solutions goes a notch beyond EAP-MD5 by requiring mutual authentication and deliviering keys used for WLAN encryption. Mutual authentication reduces the risk of access point masquerading. However, station identities and passwords remain vulnerable to attackers armed with sniffers and dictionary attack tools. LEAP is mostly attractive to organizations that use Cisco access points and cards and want to modestly raise the security bar.

3. EAP-TTLS and PEAP:

These two solutions have been proposed to simplify 802.1x development. Both require certificate-based authentication only for the RADIUS server. In addition an extensible set of different user authentication methods is offered. Organizations that have not yet issued certificates to every station and don't want to just for 802.1x can use Windows Logins and passwords instead. RADIUS servers that support EAP-TTLS and PEAP can check LAN access requests with Windows Domain controllers, Active Directories, and other existing user databases. From a sniffing perspective, these options are just as strong as EAP-TLS. However user-passwords are still more likely to be guessed, shared, or disclosed through social engineering than client-side certificates.

5.Conclusion

EAP-TTLS and PEAP are not yet finalized and are still only existing as Internet Drafts and the other options EAP-MD5 and LEAP are very simple but not that safe. So EAP & 802.1x is currently the best way to protect your WLAN via the EAP protocol but it has a huge effort with the administration of public keys for the users.

Sources:

RFC 3770 - Certificate Extensions and Attributes Supporting Authentication in Point-to-Point Protocol (PPP) and Wireless Local Area Networks (WLAN) (2004)
http://www.ietf.org/rfc/rfc3770.txt

RFC 3280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (2002)
http://www.ietf.org/rfc/rfc3280.txt

RFC 2284 - PPP Extensible Authentication Protocol (EAP) (1998)
http://www.ietf.org/rfc/rfc2284.txt

RFC 2716 - PPP EAP TLS Authentication Protocol (1999)
http://www.ietf.org/rfc/rfc2716.txt

RFC 3580 - IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines (2003)
http://www.ietf.org/rfc/rfc3580.txt

Gespann aus 802.1x und EAP authentifiziert auf Port-Ebene Torwächter an jedem Port (2002)
http://www.networkcomputing.de/index.php?p=heft/solutions/sl-2002/sl_1902234.htm

Sicherheit im Wireless-LAN (2002)
http://www.networkcomputing.de/index.php?p=heft/solutions/sl-2002/sl_0902_48.htm

Axel Sikora - Sicherheit im WLAN (2002)
http://www.tecchannel.de/hardware/928/index.html

Markus Nispel (Enterasys Networks) – EAP - Extensible Authentication Protocol
http://www.enterasys.com/de/products/whitepapers/EAP_Artikel.pdf