InteropNet Labs

8 in a Series

Inner Authentication Methods

by Matthew Gast

Recent work on wireless LAN security is based on the IEEE's 802.1X standard, which allows network administrators to build secure wireless LANs networks by requiring user authentication prior to network access. 802.1X is based on the IETF's Extensible Authentication Protocol (EAP). EAP, which is specified in RFC 2284, is an authentication framework that allows the use of many different authentication types on a link.

Tunneled Authentication

If EAP is the foundation of wireless network security, Transport Layer Security (TLS) is the cornerstone. TLS, which is specified in RFC 2246, is based on the Secure Socket Layer (SSL) protocol, and has the same objective: establishing a secure authenticated and encrypted channel across untrusted networks. TLS can be used to secure wireless networks in two ways: (1) as an authentication protocol itself, using digital certificates for authentication, and (2) to establish a secure tunnel for transmission of other legacy authentication protocols.

When TLS is used directly as the authentication protocol, it is used within EAP (EAP-TLS). EAP-TLS, which is specified in RFC 2716, uses digital certificates for authentication. While digital certificates are considered to be a secure method of strongly authenticating both clients and servers, they can require a fairly hefty public key infrastructure (PKI) at the enterprise level and significant changes to each client system. For this reason, most organizations have been slow to move to digital certificates for user authentication.

Rather than fully deploy PKI, organizations continue to use older authentication systems, either based on usernames and static passwords, or token-based authentication systems such as Secure Computing's SafeWord or RSA's SecurID. To maximize the existing investment in these authentication systems (which are commonly called "legacy authentication methods"), many organizations would like to extend these systems for use in the wireless environment. Tunneling these legacy methods inside TLS addresses is required because legacy authentication methods do not help the client identify the server, and since wireless environments are particularly easy for Man in the Middle (MITM) attacks, server authentication is a critical requirement.

When TLS is used to protect a legacy method, the combined protocol has two steps. In the first step, TLS is used to establish an "outer" protective tunnel, which identifies the server to the client using digital certificates. The outer tunnel is then used in the second step, when a legacy authentication protocol is run through the TLS tunnel. The tunneled method is frequently called the "inner" authentication protocol. The TLS tunnel brings two important characteristics. It offers the message integrity of TLS, ensuring that no outsider can interfere with the authentication, and it offers the encryption of TLS, ensuring that no outsider can eavesdrop on the authentication method. Tunneled TLS (TTLS) and Protected EAP (PEAP) are the two proposed protocols for providing tunneled authentication with 802.1X. You can learn more about TTLS and PEAP in our white paper “TTLS and PEAP Comparison.” The major difference between the two is that PEAP requires that the inner protocol also be an EAP protocol, while TTLS does not.

Inner Authentication Methods

Password Authentication Protocol (PAP). PAP was originally specified for use with PPP in RFC 1334. PAP transmits the user name and password across the network unencrypted. As such, it should not be used over a network that does not provide privacy protection. PAP is not an EAP method, and is only supported by TTLS. PAP can be used with existing network logon systems as well as token card servers. PAP is selected by network managers who wish to one-way encrypt passwords. If passwords are not reversibly encrypted, then the only way for a client (supplicant, in 802.1X terms) to prove its identity is to pass the actual password to the authentication server. For example, some LDAP databases store passwords so that they can be written and compared, but can never be read. Another example where PAP is needed is in Unix /etc/passwd files, which are one-way encrypted. In both TTLS and PEAP, the TLS tunnel “protects” the password, so passing it in the clear is not insecure.

Challenge Handshake Authentication Protocol (CHAP). Like PAP, CHAP was originally designed for use with PPP. It is specified in RFC 1994. The authentication server challenges the client, and the client proves that it is in possession of the shared secret by successfully responding to the challenge. CHAP is not an EAP method, and is only supported by TTLS. CHAP (and methods similar to CHAP, such as MS-CHAP and MS-CHAP-V2) is selected by network managers who are concerned about passing the password between the client and authentication server. With CHAP, the passwords must be available in clear text both at the client and the authentication server end. (You may also see this referred to as “reversibly encrypted.”) The security factors

which drive network managers to use CHAP (and CHAP-like methods) are not relevant in the world of 802.1X. However, network managers who have built authentication infrastructures using challenge-response authentication protocols (such as CHAP) will want to use the same protocol in their wireless environment.

Microsoft CHAP (MS-CHAP). MS-CHAP was designed by Microsoft to offer similar functionality to CHAP, but with enhanced functionality for Windows systems. Unlike CHAP, MS-CHAP does not require that the shared secret be stored in cleartext at both ends of the link. MS-CHAP calls for a particular one-way cryptographic hash of the password to be used to store the password on the server. Because the client knows the hash method used by the server, it can reproduce it, effectively creating a "matching" password on both ends. This matching password can then be used in a challenge/response handshake authentication, with the client proving that it knows the password, or, more precisely, the hashed value of the password. MS-CHAP is proprietary to Microsoft, but is documented in RFC 2433. MS-CHAP is useful in environments where Microsoft authentication databases are used. However, there are particular issues with MS-CHAP that make it undesirable from a security point of view; it should only be used by network managers who must support very old Microsoft clients, such as Windows 95/98. MS-CHAP is not an EAP method, and is only supported by TTLS.

Microsoft CHAP version 2 (MS-CHAP-V2). MS-CHAP has several identified security vulnerabilities. MS-CHAP-V2, which was initially introduced with Windows 2000 and documented in RFC 2759, addressed the shortcomings of MS-CHAP by eliminating the weak encoding of passwords for older clients, providing mutual authentication, and improving keying and key generation. MS-CHAP-V2 is widely supported by Microsoft clients, and is commonly supported and used as an inner authentication method with PEAP. Network managers who have their authentication databases stored in Microsoft Windows will want to use MS-CHAP-V2. MS-CHAP-V2 is defined both as a PPP method, which means it can be used "as-is" within TTLS, and as an EAP method, which means it can be used as "EAP-MS-CHAP-V2" tunneled within both TTLS and PEAP.

EAP-MD5 Challenge (EAP-MD5). EAP-MD5 was standardized along with EAP in RFC 2284. Its basic structure is similar to that of CHAP. Like CHAP, it requires that the passwords on both ends be available. Unlike CHAP, however, it is an EAP method, and can be used with either TTLS or PEAP. EAP-MD5 will not be useful to network managers as a tunneled authentication method; support for MS-CHAP-V2 is more common (see table below). It is more likely that EAP-MD5 will be used in non-wireless environments outside of TTLS or PEAP.

EAP-Generic Token Card (EAP-GTC). EAP-GTC was standardized along with EAP in RFC 2284. Like PAP, EAP-GTC allows the exchange of cleartext authentication credentials across the network. However, because the token card password, if properly used, is not vulnerable to a replay attack, EAP-GTC can be used by itself, and does not need to be tunneled inside of TTLS or EAP. Nevertheless, EAP-GTC should be used inside of TTLS or PEAP to provide server authentication in the wireless environment. EAP-GTC has also been proposed as a potential solution to a different problem. There is no standardized EAP method which is simply "username+password," which means that network managers who have one-way encrypted password databases (or compare-only password databases) cannot use EAP for authentication. The GTC method does provide a way to move a simple username and password from client to server using an EAP method, so it can be used to provide a PAP-like authentication method. Naturally, if EAP-GTC is used to transport reusable passwords, it must be used inside a tunnel for protection and server authentication. EAP-GTC can be used with both TTLS and PEAP. Network managers will want to use EAP-GTC if they have token cards, or as a workaround to the lack of a PAP-like authentication method within the panoply of EAP methods.

Protocol Support

TTLS Supplicants

	PAP	CHAP	MS-CHAP	MS-CHAP-V2	EAP-MD5	EAP-GTC	EAP-TLS
Funk	√	√	√	√	√	√	√
Meetinghouse	√	√	√	√	√		
Alfa & Ariss	√						

PEAP Supplicants

	EAP-MS-CHAP-V2	EAP-MD5	EAP-TLS	EAP-GTC	EAP-SIM
Microsoft	√		√		
Cisco				√	√
Meetinghouse	√		√	√	
Funk	√	√		√	