March 2005

PGP® White Paper

Transport Layer Security (TLS) & Encryption: Complementary Security Tools

PGP™

Table of Contents

Introduction

Transport Layer Security (TLS) and content encryption can both be used to secure email communications. TLS can only be used to secure part of the path an email message takes from sender to recipient, however, and it does not secure the portion of that path on which most security breaches occur. Nevertheless, TLS is a popular approach to securing email traffic across the Internet. Some security “experts” have even suggested that enterprises using TLS to secure their email traffic do not need to encrypt email message content. This PGP White Paper addresses this myth, describes the differences between TLS and content encryption, and summarizes why both are required in a comprehensive email security solution.

History of Transport Layer Security

TLS began life in 1994 as the Secure Socket Layer (SSL) feature of the Netscape Web browser. SSL was developed to solve the two problems most industry watchers then believed would limit the use of the Internet for electronic commerce:

- The need to provide Web browsers a way to verify that users were actually communicating with the website with which they intended to communicate. Given the recent spate of phishing attacks, this concern was clearly prescient.
- The need to perform secure transactions over an inherently insecure network, the public Internet. To do this, Netscape concluded it needed a way of encrypting the network traffic from its Web browser to the Web servers it wanted to sell to enterprises for e-commerce applications.

Netscape developed SSL to address both of these issues while making two key architectural assumptions about the way SSL would be used:

- All network traffic secured by SSL would be point-to-point, with no intermediate “hops.” This assumption is generally valid for Web traffic, but is not generally true for email.
- The data transmitted over an SSL link only needs to be encrypted while in transit between the Web browser and Web server. The assumption is that the Web browser user will protect the information (credit card number, user name, password, etc.) before typing it into the browser, and that the Web server physically and logically secures that information once it is received.

Although SSL was designed to encrypt Web traffic, it sits between the application layer and the transport layer of a network solution and can also be used to secure Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP), and other types of application data. Netscape released three versions of SSL between 1994 and 1996, and SSL 3.0 became the basis for the IETF-approved TLS 1.0 specification. There are some minor differences between TLS and SSL that prevent them from being interoperable; however, TLS can also “back down” to SSL when required to communicate with an application that is not fully TLS-enabled. In fact, the TLS specification allows two applications attempting to establish a TLS link to back down to older versions of SSL that contain known vulnerabilities.

For procedural reasons, TLS is not and will not likely ever be a proper superset of SSL functionally. TLS does not, for example, provide for the use of the Skipjack encryption algorithm used by U.S. military and intelligence agencies. Nor does TLS provide for certificate-less communication as SSL does. Consequently, SSL in all its current and prior incarnations will probably continue to be in use for a long time.

TLS Operating Overview

The TLS protocol is made up of two components:

- **The TLS Handshake Protocol** – used to initiate a TLS session. The Handshake Protocol allows two TLS-capable applications to negotiate which version of the TLS protocol will be used for secure communication, which encryption algorithm will be used to encrypt the data, and allows the initiating application or client to authenticate the recipient application or server. It is this authentication step that allows a Web browser to determine if the Citibank server it thinks it has contacted is actually the Citibank Web server.
- **The TLS Record Protocol** – used to transmit the client's data securely to the server application. The Record Protocol is also used to encapsulate the TLS Handshake Protocol, although discussion if this functionality exceeds the scope of this paper.

Setup of a TLS session that involves both client and server authentication occurs as follows:

Client Commands	Server Commands
1. Client contacts the server requesting a TLS session be established and includes in that request the version(s) of TLS and SSL that can be used, ciphers supported by the client, session ID, and other required parameters.	
	2. The server responds with the version of TLS/SSL that will be used, the cipher algorithm to be used, and some random data that is used to generate a session key. The server then sends the client its Certificate (which contains a public key) and an indication that client authentication is to be performed.
3. The client attempts to authenticate the server Certificate, indicates success or failure to the server, and if successful, indicates that all further communication will be encrypted. The client also forwards its certificate to the server.	
	4. The server authenticates the client Certificate and acknowledges that all further communication will be encrypted.
5. The client generates a session key using the random data previously provided by the server, encrypts it to the server's public key, and sends it to the server.	
	6. The server decrypts the session key using its private key and acknowledges receipt of the session key.

At this point, both the client and the server may use the session key to encrypt messages to one another until the session is complete.

Issues with TLS & Email

There is no denying that TLS is a good solution to the problems it was originally designed to solve. It is not, however, a very good way to secure email message traffic because email inherently violates the two previously mentioned design assumptions upon which TLS is based. TLS is also susceptible to a common type of cryptographic attack known as "Man in the Middle," which exploits the "back down" functionality cited earlier.

"Man in the Middle" Attack

The TLS Handshake Protocol requires the two communicating servers agree on the algorithms and parameters to be used to secure the communications link. Although TLS will always attempt to use the most secure combination of parameters, it is designed to "back down" to less-secure and/or older protocols and algorithms when required to establish a secure session. A "Man in the Middle" can manipulate the protocol to convince a TLS-enabled mail server to use one of these less-secure configurations either by asserting the more-secure algorithms aren't supported or by blocking the ports required to use them. Once the less-secure algorithms and parameters are in use, the attacker can then exploit known weaknesses in what are typically older protocols.

This type of attack can be mitigated by using care in configuring a TLS-enabled server, but this requirement places a burden on IT management to review the TLS configuration regularly to ensure nothing has been altered deliberately or inadvertently by IT technical staff. Worse, it means that enterprises depending solely on TLS for email security must depend on *all* of their trading partners to exercise similar diligence to prevent email transmissions from being breached by this sort of attack.

TLS: Point-to-Point Security in a Store-and-Forward Environment

The first core design assumption of TLS that is broken by email is that TLS's communication security is point-to-point. The SMTP protocol used by all email systems globally is designed as a store-and-forward transfer mechanism. The reasons for this design are beyond the scope of this paper, but the most important benefit of this approach is that mail goes through even if one of the email servers along its intended route is unavailable.

Therefore, for a comprehensive TLS-based email security system to be implemented, TLS must be provisioned on all links across which the email is expected to flow. This approach is so problematic that no one even attempts it. Instead, TLS-based email security systems focus on the link between a sending enterprise's outbound email server and the recipient's inbound gateway. Thus, the mail is only secured while it is traversing the public Internet on that one "hop." The problem with this approach is that it secures the link on which the least number of security breaches occur. It is a widely accepted tenet of enterprise information security that up to 70% of all breaches occur inside the enterprise's firewall[1] when email is still moving unprotected by any type of encryption or other security. The only way to ensure the security of email from its point of origin to its final destination is to use a content encryption solution such as PGP® Desktop or PGP® Universal.

[1] *CSI/FBI Computer Crime & Security Survey*, 2003

The other consequence of attempting to secure email using TLS is that this approach assumes email *always* takes its intended path to the recipient. If an email server or mail transport agent (MTA) fails or the network link between the sending and receiving MTAs fail, the email could follow almost any route across the Internet to its destination, completely circumventing the TLS security system. Although this problem can be mitigated by configuring the sending MTA to hold all email until the intended network path and/or receiving MTA are again available, it means that key business transactions may be delayed and that outbound email is exposed to anyone with access to the sending MTA where they're staged.

TLS: Protects Data in Motion Only

The second TLS design assumption violated by email is that data need only be protected while it is in motion. This is a good assumption when securing information being submitted by a Web browser to a Web server, but it carries potentially tragic consequences in the case of email. A quick check of the headers of any email reveals that it resides on at least six different systems in its lifespan:

- Sender's desktop or laptop system (or mail server, depending on the configuration)
- Sending enterprise gateway MTA
- Recipient enterprise gateway MTA
- Recipient enterprise mail server
- End user recipient's desktop or laptop system

The foregoing is the minimum number of times (five) an email is stored on a client or server system. In many cases, it may also be received, stored, and resent by transparent email proxy servers within the sender's or recipient's Internet Service Provider (ISP) infrastructure. A complete and effective email security solution protects messages both in motion and at rest. No deployment of TLS, regardless of configuration, can achieve this level of security.

Comprehensive Email Security Using TLS & Encryption

The only way to completely secure email is to use both TLS and a content encryption solution such as PGP Desktop or PGP Universal. Encrypting email content guarantees end-to-end security and ensures that email traversing the public Internet is not subject to the sort of attacks to which TLS is vulnerable.

However, TLS does have a role even when an email content encryption solution is used. PGP Universal does, in fact, use TLS for certain email client communication when no desktop encryption functionality is available or when PGP Universal is deployed in a cluster configuration. PGP Universal handles these TLS connections transparently and requires no administrator intervention to ensure their integrity. PGP Universal's usage of TLS is also consistent with the design assumptions cited above and is not subject to TLS's inherent vulnerabilities.

There is also one application of secure email that requires use of both content encryption and TLS when email is crossing the public Internet. In certain (rather rare) cases, it is desirable to encrypt both the message content and the message headers that contain the To, From, and Subject lines of the message. These applications are typically limited to military and national intelligence deployments. In these extremely sensitive applications, the combination of both content encryption and TLS while the email is traversing the public Internet is the best solution.

TLS clearly plays an important role in securing sensitive data on the Internet. Given its relative simplicity and low cost, it is also a simple way to secure email traffic. As this paper has shown, however, TLS is a vulnerable and incomplete approach for securing real-world enterprise email content. To achieve this mission-critical objective, content encryption solutions such as PGP Universal are the preferred solution.

PGP Corporation
3460 West Bayshore Road
Palo Alto, CA 94303 USA
Tel: +1 650 319 9000
Fax: +1 650 319 9001
Sales: +1 877 228 9747
Support: www.pgpsupport.com
www.pgp.com

The information described in this document may be protected by one or more U.S. patents, foreign patents, or pending applications.

PGP is a registered trademark of PGP Corporation. Product and brand names used in the document may be trademarks or registered trademarks of their respective owners. Any such trademarks or registered trademarks are the sole property of their respective owners.

The information in this document is provided “as is” without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

This document could include technical inaccuracies or typographical errors.

All strategic and product statements in this document are subject to change at PGP Corporation's sole discretion, including the right to alter or cancel features, functionality, or release dates.

Changes to this document may be made at any time without notice.