



Security Research Division

Secure Socket Layer (SSL) Transparency

Gives Authorized Network Security and Management Components a “Window of Transparency” into Encrypted Communications

Overview

The SSL (Secure Sockets Layer) Transparency project, solves an important, quickly growing problem: While ubiquitous end-to-end encryption enables users to communicate securely, it denies access to network analysis and network security tools. An Intrusion Detection System (IDS) cannot scan encrypted traffic for attacks. Network General's Sniffer[®], for example, cannot display encrypted traffic to an administrator diagnosing network problems. Thus, end-to-end encryption is a mixed blessing. While the specific traffic is more secure, ironically the network on which it runs is less secure, as the tools meant to protect the network cannot function properly.

The objective of the SSL Transparency project is to permit authorized administrators and management tools to decrypt SSL protected traffic while keeping the traffic secure from other entities. SSL Transparency must maintain a high level of security for the end points of the SSL session

We chose to focus on SSL traffic because of its ubiquity: SSL protects the majority of secure web-based transactions, many are familiar with it, and SSL is widely implemented and heavily used.

Concept

In establishing an SSL session, the two end points negotiate encryption keys that are used to encrypt the data for that session. To decrypt the SSL data one must know the session keys, which by design are known only to the two end points.

SSL Transparency accomplishes its objective by securely extracting the session keys from a collaborating end-point and sharing the keys with network administration and security tools. The project designed a framework for sharing keys

that is independent of the collaborating end point (client or server), and independent of the specific network tool sharing the keys. The network administration and security tools use these keys to decrypt the traffic, giving them a “window of transparency” into the traffic, thus allowing them to do their job while keeping the traffic secure from other entities.

As SSL Transparency requires sharing encryption keys, another important objective of the project is to develop a secure, flexible key sharing policy and robust software to implement both key sharing and key sharing policy enforcement.

Approach

The SSL Transparency framework consists of three independent components: a key supplier, a key broker, and a key consumer. There can be many key suppliers and many key consumers, but usually there is only one key broker. The key supplier is a cooperating end point in the SSL communication. The supplier extracts the SSL encryption keys, encrypts them, and deposits them into the key broker. The key broker accepts keys from key suppliers and distributes keys to authorized key consumers. Since the keys stored in the broker are encrypted, the key broker merely collects and distributes opaque blobs of data to authorized consumers. Authorized key consumers request and receive SSL encryption keys for the specific stream of traffic they wish to decode.

Sharing of encryption keys is a very sensitive issue. Thus a strong and flexible policy was developed that defines the interactions among the components in the framework, especially the key suppliers. The framework and associated policy are intended to be deployed in a corporate or government setting that already has full authority over end user's machines, expects full control

This work was an internal research and development project funded by McAfee Research, which is now the Security Research Division of SPARTA.



Secure Socket Layer (SSL) Transparency

Gives Authorized Network Security and Management Components a "Window of Transparency" into Encrypted Communications

over end user's communication, and where personal privacy is less of an issue.

The policy defines when and who can supply, store, and consume keys. The main impact of the policy is on the key supplier and takes into account the nature of web-based SSL traffic. The key supplier can share keys for a specific traffic stream only when the following conditions are met:

- The machine on which the supplier resides is authorized to share keys. If the machine is sensitive, such as the personal desktop of a company CEO, then no keys will be shared.
- There are no privacy restrictions associated with the end points. Keys will not be shared if the transaction involves a web site, such as a personal banking site, containing sensitive data.

If the SSL session meets these requirements, the key supplier deposits the SSL encryption keys for that particular session into the key broker.

The policy defines which key consumer may request session keys from the key broker. This policy is enforced through cryptography. Each key consumer encrypts the session keys prior to depositing them in the key broker. If a consumer does not know how to unwrap the session keys, it cannot decrypt the SSL session. Thus, compromising the key broker does not compromise the security of an SSL session.

Initial Prototype

For the initial SSL Transparency prototype we implemented the framework on Windows 2000®. We implemented a minimal key broker, a single collaborating key supplier, the Internet Explorer® (IE), and a single key consumer, SnifferPro® 4.5.

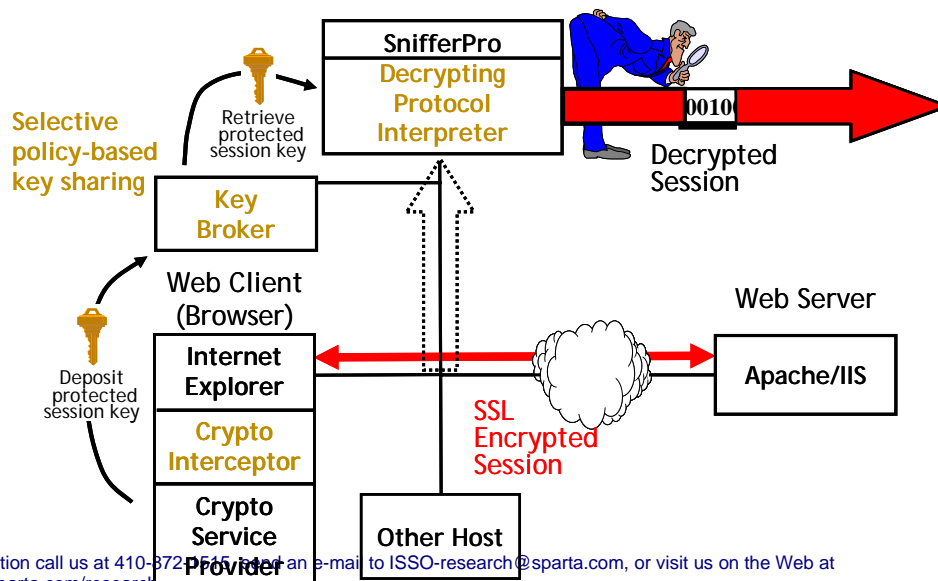
The prototype takes advantage of the cryptographic framework in the Windows operating systems (OS). The OS allows third-party plug-ins, known as Cryptographic Service Providers (CSPs), to handle cryptographic functions for applications. We insert a small shim CSP in place of the standard SSL CSP. This shim layer simply passes all calls to the standard CSP – unless it sees an SSL key being generated, in which case it makes a copy of the key, encrypts it, and then deposits the encrypted key in its key broker.

Future Work

There are a number of straightforward tasks to fully realize the potential of this project:

- Decrypt other streams, such as IPSEC.
- Use a public key infrastructure to protect and authorize components instead of the shared secrets used in the prototype. Key suppliers would encrypt keys for specific consumers.
- Develop new SSL key suppliers.
- Develop new key consumers, such as an IDS.

Architecture – After SSL Transparency



For more information call us at 410-372-0755, fax at 410-372-0756, or e-mail to ISSO-research@sparta.com, or visit us on the Web at <http://www.issosparta.com/research>.