Supporting virtual servers with Server Name Indication

CERTIFIED DELIVERY

Server Name Indication lets you operate more than one SSL-protected service per IP address. BY THORSTEN FISCHER

Web users and developers are equally devoted to the goal of keeping attackers from sniffing online orders, logging credit card numbers, and plundering user accounts. Fortunately, the introduction of Secure Socket Layer (SSL [1]), a protocol for encrypted data transmission and reliable identification, helps prevent this horror scenario by offering a means for protecting sensitive web activities such as online banking.

The https protocol integrates SSL with http for secure web communication. With the use of encryption parameters that are negotiated with the use of the DNS name of the server, https establishes a secure connection.

This approach works very well when only one DNS name is associated with the IP address; however, it creates a problem for anyone wanting to run virtual servers with different names on a single address.

The client contacts the SSL server through the specified IP address and announces that it wants to encrypt. The server confirms the request, presents a certificate, and proposes a combination of algorithms supported by both server and client. If the options are to the client's liking, the client accepts the certificate. The partners handle all subsequent http requests via the encrypted channel (Figure 1).

One of the parameters typically transmitted with the certificate is the DNS name for the website, which is added to the *CommonName* field. X.509v3 certificates enter this information to the *CN:* attribute. After receiving the certificate, the client sends the http request. The request also contains the name of the server to which it is addressed. This means that, if multiple websites use the same IP address, the only site that can handle secure communications is the one specifically named by the certificate.

The client is not the only element that can cause confusion because of an ambiguous request. For example, if the provider runs multiple https services on a single IP address, the server also has a major problem: After establishing the

Table 1: Supported Programs

Program	Role	As of Version	Comment
Apache	Server	2.0.55 and 2.2	Modules are available for both *mod_gnutls* and *mod_openssl*. The Apache developers offer patches.
Lighttpd	Server	1.4.18 and 1.5	Requires a patch that depends on OpenSSL [4].
Firefox	Client	2.0	TLS must be enabled.
Opera	Client	8.0	TLS 1.1 must be enabled.
Internet Explorer	Client	7.0 Beta 2	Only on Windows Vista; not on Windows XP.
Konqueror	Client	3.5.1	Requires OpenSSL 0.9.8f; see KDE bug #122433.

secure connection, the server must identify which private key to use to process the client's encrypted request.

To do so, the server must evaluate the *Host:* header in the http request. Of course, the request – including the header – is encrypted until the server determines which private key to use.

The recent emphasis on virtual computing and the need for web hosters and other service providers to conserve IP addresses have added urgency to this problem. Fortunately, the Internet Engineering Task Force's (IETF's) successor to SSL, the Transport Layer Security (TLS) protocol, provides a solution through the Server Name Indication (SNI) extension, which is described in RFC 4366.

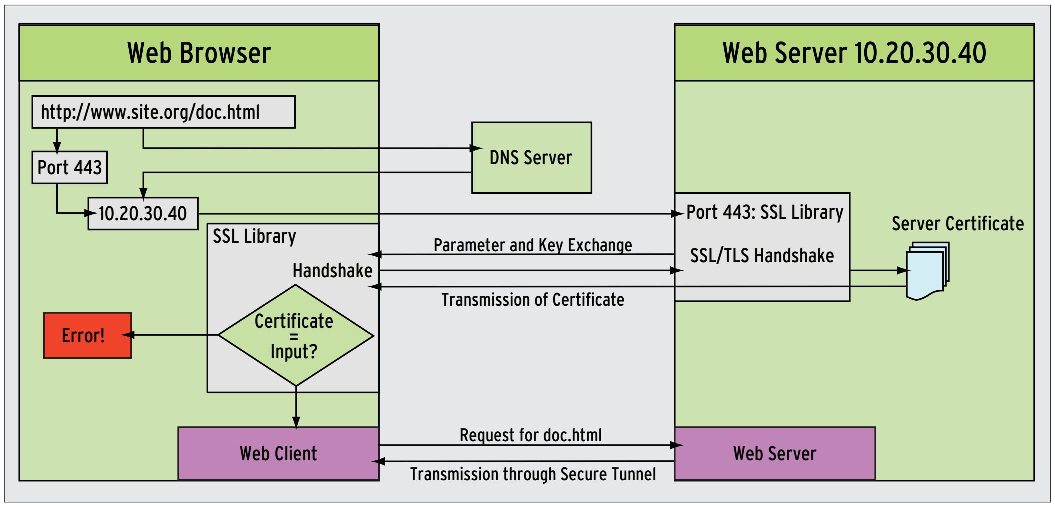

Figure 1: Several steps are required between entering a URL in your browser and transmitting secure web content. The client uses a DNS server to find the IP address for the name in the URL. The client receives a certificate from the IP address as part of the SSL session. Only if the certificate is valid and the name matches the name in the request will SSL start to transfer content.

To solve the problem of operating multiple virtual servers on a single IP address, clients must have the ability to specify the name with which they want to communicate at the time an SSL connection is established.

Conventional Secure Socket Layer doesn't offer this option, but the SNI extension supports the transmission of additional data at the handshake phase [2]. To be more precise, an optional field can be transmitted at the TLS *ClientHello*

step. The client can use this field to specify the name of the partner it wants to talk to. When the server then transmits the correct certificate at the next step, the client knows which private key to use for the ensuing communications (see Figure 2).

For Server Name Indication to work, both the client-side and server-side software must support the method. The accompanying sidebar titled "TLS Library Extensions" summarizes the support currently offered by the Gnutls, OpenSSL, and NSS (Network Security Services) libraries.

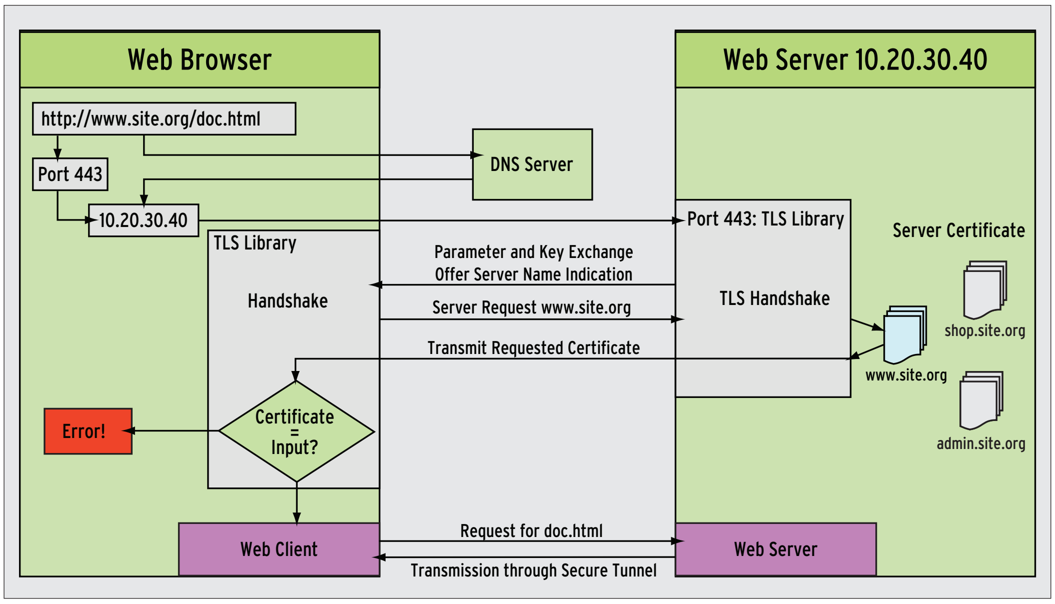

Figure 2: Server Name Indication (SNI) relies on a Transport Layer Security (TLS) extension. The client passes in the required server name in the handshake phase so that the server can respond with a matching certificate.

For the most part, Server Name Indication is transparent to the user. Table 1 gives an overview of clients and servers that are supported. Opera was the first browser to support Server Name Indication in version 8.0, and Firefox introduced support in version 2.0. Both browsers require the user to enable Transport Layer Security explicitly, but most security experts recommend the use of this setting anyway.

Internet Explorer version 7.0 or newer also supports Server Name Indication, but only for Vista, not for Windows XP. If you would like to test the SNI capabilities of your browser, you can surf to the Transport Layer Security Server Name Indication test site [3].

TLS Library Extensions

If you want to use SNI, you need a library that supports it. Three of the popular libraries offering SNI support include Gnutls by the Free Software Foundation, NSS from the Mozilla project, and the widespread OpenSSL.

Version 0.5.10 of the Gnutls library introduced SNI support in 2002 in the form of separate server- and client-side functions. All a developer needs to do is call *gnutls_server_name_set()* client-side and *gnutls_server_name_get()* server-side to write SNI-capable applications. This explains why the Apache *mod_gnutls* module can handle SNI.

OpenSSL only recently introduced SNI support with version 0.9.8f of the TLS extensions. Macros such as *SSL_set_tlsext_host_name()* and functions such as *SSL_get_servername()* are available. The Apache *mod_ssl* module can also handle this.

The NSS library, which is used by Firefox, is not quite as advanced. NSS 3.11.1 supports client-side SNI, but server-side SNI is not yet supported. The project roadmap explicitly states that version 3.12 will not support SNI.

Security Matters

Server Name Indication offers many benefits for virtual server environments, but as with any powerful tool, it is important to proceed carefully.

If you perform web server security testing, keep in mind that an incorrect configuration can provide a number of attack vectors. For example, an attacker could use the Server Name Indication extension to guess generic virtual server names hiding behind the shared IP address.

If the web server receives a request with a modified *Host:* header, following, say, a generic name such as *intranet*, it might serve up the content for this site. The http *CONNECT* method can sometimes be tricked in a similar way to establish illegitimate internal proxy connections.

Server Name Indication obviously does not deliberately publish internal documents on insecure networks, but it does give administrators another opportunity to make configuration errors. After all, the web server is expected to handle requests for content from websites with different names.

Conclusion

Server Name Indication solves a problem that the developer community has brought down on its own head in the rush for rapid growth. An extension of the Transport Layer Security standard now gives website operators the ability to bind multiple certificates to a single IP address. This prerequisite is important with regard to running multiple, secure virtual servers on a single IP address.

Providers can save money and resources by doing so, but administrators should take care whenever offering multiple web servers on a single operating system instance. ■

INFO

[1] "The SSL Protocol Version 3.0" by A. Freier, P. Karlton, and P. C. Kocher, March 1996, *http://wp.netscape.com/eng/ssl3/draft302.txt*

[2] RFC 4366, "Transport Layer Security (TLS) Extensions": *http://www.ietf.org/rfc/rfc4366.txt*

[3] Kaspar Brand Server Name Indication test site: *https://sni.velox.ch*

[4] SNI patch for Lighttpd: *http://trac.lighttpd.net/trac/ticket/386*