IBM Software Group

IBM HTTP Server - Certificates and the Secure Sockets Layer (SSL) - session#2

Robert Boretti
Advisory Software Engineer

WebSphere® Support Technical Exchange

ON DEMAND BUSINESS

Today's Agenda

- Learn How to…
  - configure the **Secure Sockets Layer** within IBM® HTTP Server 6.x.

Client ←--SSL--→(443)IHS

- Specifically Learn..
  - what **directives** are required to enable an SSL website in the web server's *configuration file* (httpd.conf)
  - how to enable "**multiple**" SSL *virtualhosts* (websites)

Today's Agenda (continued..)

- how to enable SSL **client authentication**
- how to *limit encryption* at **128bit** or **higher** for optimal security
- how to **automatically** *redirect* non-SSL clients over SSL

- Brief overview..
  - SSL within the **WebSphere® HTTP plugin**

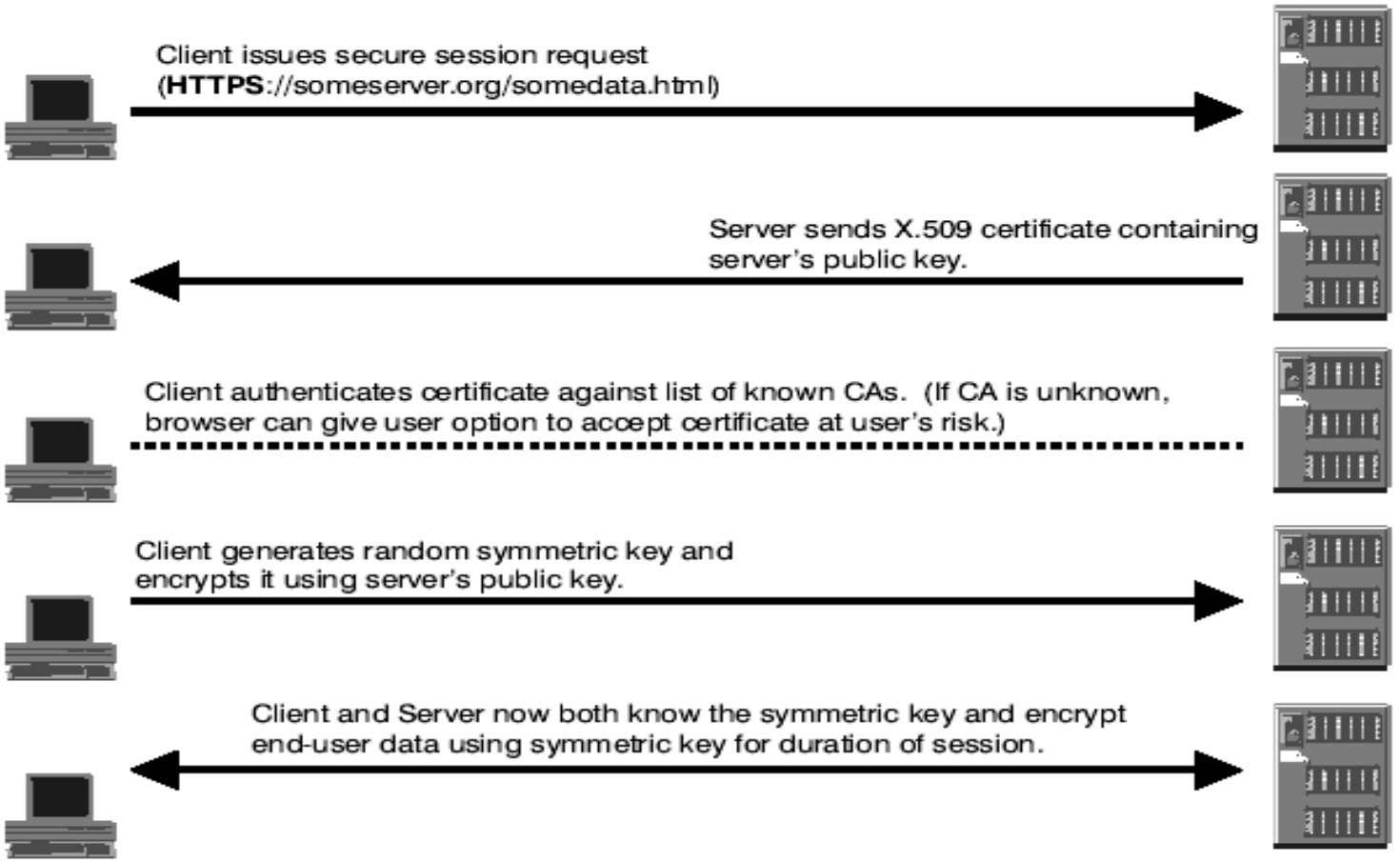
First Things First..

- What is SSL and How does the handshake work?
  - the **secure sockets Layer** is an *encryption protocol* used to..
    - **encrypt** sensitive data *between* the client and the IBM HTTP Server
    - **prevent** *third parties* from *reading* the data even if it is intercepted in route
    - **enable** clients the ability to *authenticate* the *identity* of the web server

(continued..)

- for **data** between a client and web server to be *secured* over SSL, the two parties must first **exchange** *private* and *public* keys
- the *method* of exchanging *public* and *private* keys is commonly referred to as the **handshake** (see diagram on next slide)

(continued..)


Client issues secure session request
(HTTPS://someserver.org/somedata.html)
Server sends X.509 certificate containing
server's public key.
Client authenticates certificate against list of known CAs. (If CA is unknown,
browser can give user option to accept certificate at user's risk.)
Client generates random symmetric key and
encrypts it using server's public key.
Client and Server now both know the symmetric key and encrypt
end-user data using symmetric key for duration of session.

Next, Let's Get Busy..

- what **directives** are required to enable an SSL website in the web server's *configuration* file (httpd.conf)?
- how can I enable "**multiple**" SSL *virtualhosts* (websites)?
- how do I enable SSL **client authentication**?
- how can I limit *encryption* at **128bit** or *higher* for optimal security?
- what is needed to "**automatically**" *redirect* non-SSL clients over SSL?

what directives are required to enable an SSL website in the web server's configuration file (httpd.conf)?

- Create ahead of time..
  - per the previous session, a **keyfile** (e.g. key.kdb) will need to be created along with a **server certificate**, using the *IBM Key Management* Utility
- Define an *ipaddress:port* you want to use for SSL by adding a ***Listen*** directive
  - example: Listen 123.45.67.89:443

(continued..)

- Next, **add** or **uncomment** the SSL *loadmodule*
  - LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
- Now, *create* a **virtualhost** stanza with the *ipaddress:port* you have designated for SSL along with the following directives
  - ***ServerName -*** defines the website name for this virtualhost
  - ***SSLEnable*** - enables this virtualhost for secure communication
  - ***SSLClientAuth None*** - Indicates that client authentication is disabled. This means that the client browser is not required to pass a client certificate during the handshake. Most sites do not require client authentication.

(continued..)

- example: *basic* virtualhost configuration for SSL

```
<VirtualHost 123.45.67.89:443>
ServerName robo.raleigh.ibm.com
SSLEnable
SSLClientAuth None
</VirtualHost>
```

(continued..)

- Finally, you need to add some *additional* directives which can be placed **outside** the *virtualhost* stanza
  - ***SSLDisable* -** In the Global Scope; indicates SSL is disabled outside the virtualhost
  - ***KeyFile* -** In the Global Scope; points to the key database file that contains the personal server certificates required by the browser during an SSL handshake
  - ***SSLV2Timeout* -** Sets the timeout for SSL Version 2 session IDs
  - ***SSLV3Timeout* -** Sets the timeout for SSL Version 3 session IDs

(continued..)

- Putting it all together..

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so

Listen 123.45.67.89:443

<VirtualHost 123.45.67.89:443>
        ServerName robo.raleigh.ibm.com
        SSLEnable
        SSLClientAuth None
</VirtualHost>

SSLDisable
KeyFile "c:/program files/ibm http server/key.kdb"
SSLV2Timeout 100
SSLV3Timeout 1000
```

how can I enable “multiple” SSL virtualhosts (websites)?

- Enabling **additional** websites listening on the *same port* (e.g. 443) requires..
  - a *unique* **ipaddress**
- In addition, the following *directives*..
  - **SSLServerCert -** Specifies the labelname of the certificate in the key database file that must be passed to the client browser during an SSL handshake. This is required when you have multiple certificates stored in the same key database file

    OR
  - a *unique* **KeyFile** *inside* each SSL virtualhost

(continued..)

- Example: multiple SSL virtualhosts

```
Listen 123.45.67.89:443

<VirtualHost 123.45.67.89:443>
ServerName robo.raleigh.ibm.com
SSLEnable
SSLServerCert robo
SSLClientAuth None
</VirtualHost>

Listen 123.45.67.90:443

<VirtualHost 123.45.67.90:443>
ServerName robo2.raleigh.ibm.com
SSLEnable
SSLServerCert robo2
SSLClientAuth None
</VirtualHost>

SSLDisable
KeyFile "c:/program files/ibm http server/key.kdb"
SSLV2Timeout 100
SSLV3Timeout 1000
```

(continued..)

- Another example: multiple SSL virtualhosts

```
Listen 123.45.67.89:443

<VirtualHost 123.45.67.89:443>
ServerName robo.raleigh.ibm.com
SSLEnable
KeyFile "c:/program files/ibm http server/key.kdb"
SSLClientAuth None
</VirtualHost>

Listen 123.45.67.90:443

<VirtualHost 123.45.67.90:443>
ServerName robo2.raleigh.ibm.com
SSLEnable
KeyFile "c:/program files/ibm http server/key2.kdb"
SSLClientAuth None
</VirtualHost>

SSLDisable
SSLV2Timeout 100
SSLV3Timeout 1000
```

how do I enable SSL client authentication?

- In an IBM HTTP Server environment, enabling SSL *client authentication* can be accomplished using the **SSLClientAuth** directive as follows..

```
<VirtualHost 123.45.67.89:443>
ServerName robo.raleigh.ibm.com
SSLEnable
SSLServerCert robo
SSLClientAuth Required
</VirtualHost>
```

how can I limit encryption at 128bit or higher for optimal security?

- Background:
  - the *encryption level* in SSL is determined by the **cipher specification** used during the secure transaction
  - by default, IBM HTTP Server has a **built-in list** of *cipher specifications* to use for communicating with clients over Secure Sockets Layer (SSL)

(continued..)

- the *actual* cipher specification that is *used* for a particular client connection is selected from those which are **supported** by both IBM HTTP Server and the client
- some cipher specifications provide a **weaker level** of *security* than others, and might need to be *avoided* for security reasons

(continued..)

- The **SSLCipherSpec** directive can be used to limit encryption to *128bit ciphers* or *higher*. If the client supports any of the ciphers listed, the handshake will succeed, if not, the handshake will fail

  example of SSLv3 ciphers:

  - **3A** - SSL_RSA_WITH_3DES_EDE_CBC_SHA

    (Triple-DES SHA (168-bit))
  - **35** - SSL_RSA_WITH_RC4_128_SHA RC4

    (SHA (128-bit))
  - **34** - SSL_RSA_WITH_RC4_128_MD5 RC4

    (MD5 (128-bit))

(Continued..)

- Example of using **SSLCipherSpec**

<VirtualHost 123.45.67.89:443>
ServerName robo.raleigh.ibm.com
SSLEnable
SSLServerCert robo
SSLClientAuth None
SSLCipherSpec 34
SSLCipherSpec 35
SSLCipherSpec 3A
</VirtualHost>

(Continued..)

- In addition, **SSLCipherRequire** and **SSLCipherBan** can be used to limit access to particular *resources* based on the *cipher* which was used.
  - However, these directives should be used inside ***directory*** or ***location*** stanzas only. See example below..

```
<VirtualHost 123.45.67.89:443>
ServerName robo.raleigh.ibm.com
SSLEnable
<Location /mystuff/resume.html>
SSLCipherRequire 34
SSLCipherRequire 35
SSLCipherRequire 3A
</Location>
</VirtualHost>
```

What is needed to “automatically” redirect non-SSL clients over SSL?

- In an IBM HTTP Server environment, there is a **simple** way to redirect *incoming clients* from non-SSL to SSL using mod_rewrite

```
LoadModule rewrite_module    modules/mod_rewrite.so

RewriteEngine on
RewriteCond %{SERVER_PORT} =80
RewriteRule ^(.*) https://%{SERVER_NAME}%{REQUEST_URI }
```

Last, But Not Least..

- SSL within the WebSphere HTTP plug-in

Client ←--SSL--→(443)IHS | plug-in←--SSL--→(9443)WebSphere

  - What is required in the WebSphere HTTP plug-in's configuration file (plugin-cfg.xml)?
  - What is required in the WebSphere HTTP plug-in's keyfile (plugin-key.kdb)?

What is required in the WebSphere HTTP plug-in's configuration file (plugin-cfg.xml)?

- Background:
  - the *HTTP WebSphere plug-in* running in the IBM HTTP Server acts as **client** to WebSphere.
  - the HTTP WebSphere plug-in communicates over SSL to the backend *WebSphere Application Server* port (e.g 9443)
  - since the plug-in is a client, the SSL between the **plug-in←---ssl---→(9443)WebSphere**, is *separate* from the SSL between the **client←---ssl--→(443)IHS**

(continued..)

- There are **two requirements** for the *HTTP plug-in client* to be able to *parse* an *incoming* request received on *port 443* and to then establish a *connection* over SSL to *port 9443 (plugin-cfg.xml)*

  - 1. **virtualhost** defined as **:443* or *hostname:443*

    example:

    <VirtualHostGroup Name="default_host">

    <VirtualHost Name="*:443"/>

    <VirtualHost Name="*:80"/>

    </VirtualHostGroup>

(continued..)

- 2. an **HTTPS transport** defined with the *WebSphere SSL port* and *plugin-key.kdb* used to store the certificate authority signer certificates. Also, the client certificates if *SSL mutual authentication* is enabled in WebSphere

(continued..)

- Example: HTTPS transport (plugin-cfg.xml)

```
<Transport Hostname="robo.raleigh.ibm.com" Port="9443"
   Protocol="https">
<Property Name="keyring" Value="C:\Program Files\WebSphere\etc\plugin-
   key.kdb"/>
<Property Name="stashfile" Value="C:\Program Files\WebSphere\etc\plugin-
   key.sth"/>
</Transport>
```

What is required in the WebSphere HTTP plug-in's keyfile (plugin-key.kdb)?

- As indicated previously, the *WebSphere HTTP plug-in* is a **client**, therefore, similar to a *browser*, the following must be true..
  - The plugin must be able to **authenticate** and **trust** the *identity* of the server it is connecting too over SSL. In this case WebSphere
  - The plugin must be able to **pass** a *client certificate* to WebSphere if *client mutual authentication* is enabled in WebSphere.

(continued..)

- This is where the plugin-key.kdb comes in.
  - This keyfile is where the plugin stores it's list of trusted *certificate authority signer certificates* used to **validate** the *server certificate* passed by WebSphere during the handshake. These CA signer certificates are stored under the "signer certificates" section of the keyfile
  - This keyfile also is used to hold a certificate under the "personal certificates" section that can be **passed** to WebSphere as a *client certificate* if needed.

Let's Not Forget..

- Additional SSL information not covered today
  - What is session id cache?

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/ihs/ihs/cihs_ensessid.html

  - Getting started with the cryptographic hardware for SSL

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_cryptossl.html

  - SSL with LDAP

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/cihs_sslandldap.html

  - SSL Certificate Revocation List

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/cihs_crlinssl.html

  - Reverse Proxy with SSL

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.ihs.doc/info/ihs/ihs/tihs_revprox.html

Additional WebSphere Product Resources

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/
- Learn about other upcoming webcasts, conferences and events: http://www.ibm.com/software/websphere/events_1.html
- Join the Global WebSphere User Group Community: http://www.websphere.org
- Access key product show-me demos and tutorials by visiting IBM Education Assistant: http://www.ibm.com/software/info/education/assistant
- View a Flash replay with step-by-step instructions for using the Electronic Service Request (ESR) tool for submitting problems electronically: http://www.ibm.com/software/websphere/support/d2w.html
- Sign up to receive weekly technical My support emails: http://www.ibm.com/software/support/einfo.html

Questions and Answers