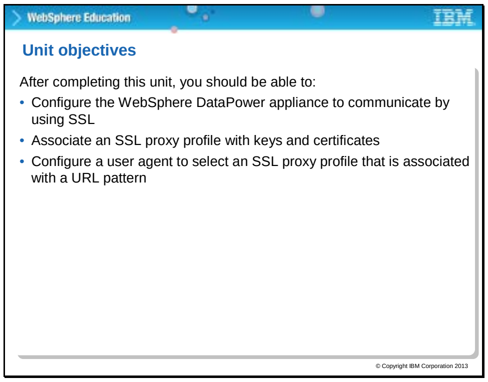
Slide 1

WebSphere Education

Securing connections by using SSL

Slide 2



WebSphere Education

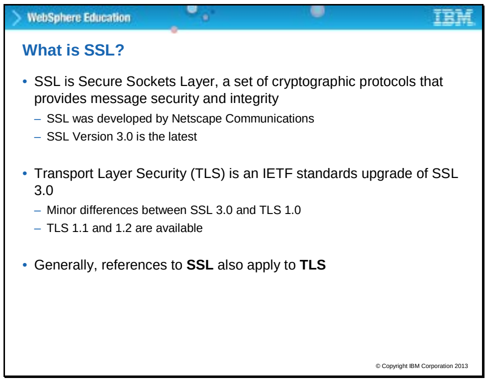
Unit objectives

After completing this unit, you should be able to:

- Configure the WebSphere DataPower appliance to communicate by using SSL
- Associate an SSL proxy profile with keys and certificates
- Configure a user agent to select an SSL proxy profile that is associated with a URL pattern

Solving security problems
The previous unit covered the three security problems of confidentiality, integrity, and non-repudiation, and how the security problems can be addressed by using keys, encryption, digital signatures, and certificates. In this unit, you see how to combine all three to create secure communications by using a protocol that is called SSL, or secure sockets layer. This protocol goes back to the 1990s and developed by Netscape. More recently a new protocol is introduced called transport Layer Security, or TLS. SSL 3 is similar to TLS 1. This unit covers SSL.

Slide 3



WebSphere Education

IBM

What is SSL?

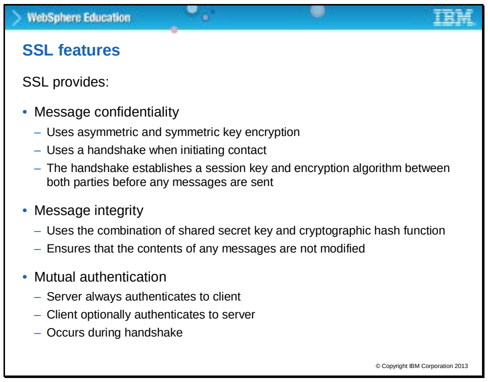
- SSL is Secure Sockets Layer, a set of cryptographic protocols that provides message security and integrity
  - SSL was developed by Netscape Communications
  - SSL Version 3.0 is the latest
- Transport Layer Security (TLS) is an IETF standards upgrade of SSL 3.0
  - Minor differences between SSL 3.0 and TLS 1.0
  - TLS 1.1 and 1.2 are available
- Generally, references to **SSL** also apply to **TLS**

What is SSL?

SSL is Secure Sockets Layer, a set of cryptographic protocols that provides message security and integrity. Netscape Communications developed SSL. SSL Version 3.0 is the latest

IETF is Internet Engineering Task Force: www.ietf.org.

Slide 4



WebSphere Education

SSL features

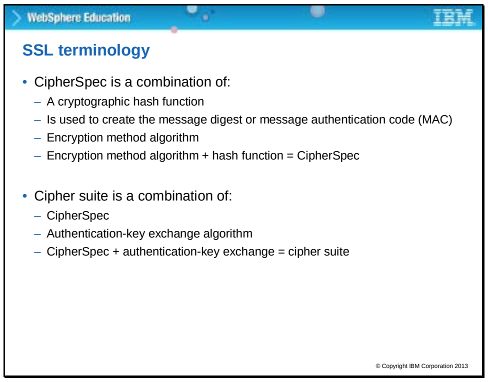
SSL provides:

- Message confidentiality
  - Uses asymmetric and symmetric key encryption
  - Uses a handshake when initiating contact
  - The handshake establishes a session key and encryption algorithm between both parties before any messages are sent
- Message integrity
  - Uses the combination of shared secret key and cryptographic hash function
  - Ensures that the contents of any messages are not modified
- Mutual authentication
  - Server always authenticates to client
  - Client optionally authenticates to server
  - Occurs during handshake

SSL features

SSL communication starts out as unsecured communication. The first step is what is called a handshake. The handshake is where the client calls the server and requests a secure session. The server always provides authentication information to the calling client, but it might also require the client to provide authentication as well. The process is known as mutual authentication.

Slide 5



WebSphere Education

SSL terminology

- CipherSpec is a combination of:
  - A cryptographic hash function
  - Is used to create the message digest or message authentication code (MAC)
  - Encryption method algorithm
  - Encryption method algorithm + hash function = CipherSpec

- Cipher suite is a combination of:
  - CipherSpec
  - Authentication-key exchange algorithm
  - CipherSpec + authentication-key exchange = cipher suite

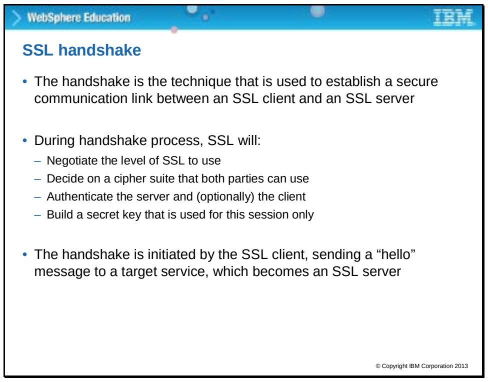
SSL terminology

Here are a couple of terms that appear several times during this unit.

First, there is the CipherSpec. The two sides must agree on what hash function is used to create the message digest, and then which type of algorithm is used to encrypt the exchange. These two things together are called the CipherSpec.

The next piece to know is how the keys are going to be exchanged. The key exchange, together with the CipherSpec, is called the cipher suite.

Slide 6



WebSphere Education

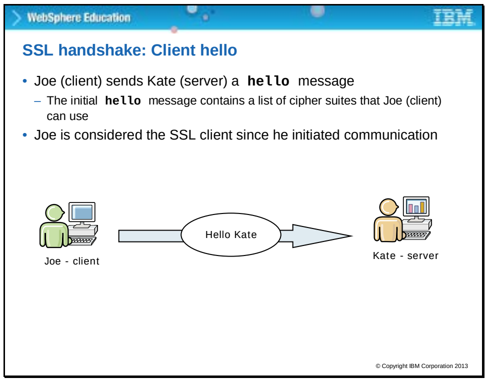
SSL handshake

- The handshake is the technique that is used to establish a secure communication link between an SSL client and an SSL server
- During handshake process, SSL will:
  - Negotiate the level of SSL to use
  - Decide on a cipher suite that both parties can use
  - Authenticate the server and (optionally) the client
  - Build a secret key that is used for this session only
- The handshake is initiated by the SSL client, sending a “hello” message to a target service, which becomes an SSL server

SSL handshake

As already mentioned, secure communication starts out as clear, or unsecured, communication. The handshake says openly, “I want to talk secretly with the server”. The server indicates what level of SSL it can provide, and what cipher suite it is willing to use. At the same time, the server authenticates itself by sending a certificate, and might ask the client to reply with its own authentication. Finally, a secret key can be constructed, and the communication moves from openness to secrecy. The next few slides review these steps in some detail.

Slide 7



WebSphere Education

SSL handshake: Client hello

- Joe (client) sends Kate (server) a **`hello`** message
  - The initial **`hello`** message contains a list of cipher suites that Joe (client) can use
- Joe is considered the SSL client since he initiated communication

Joe - client

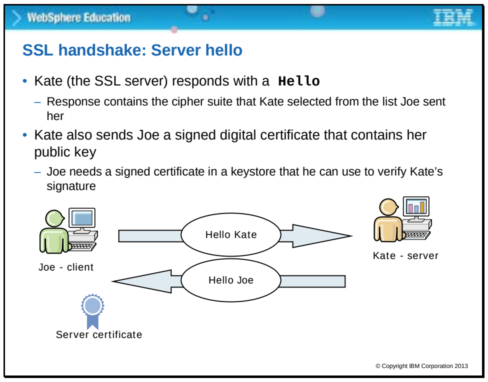
Hello Kate

Kate - server

SSL handshake: client hello

In the next few slides, Joe is the client that calls the server Kate. Communication starts when Joe sends an open ‘hello’ message to the server. Joe sends a list of any cipher suites the client can work with; that is to say, hash function, encryption method algorithm, and authentication-key exchange algorithm. The client includes some random data with the hello. The process is covered in more detail later.

Slide 8



WebSphere Education

SSL handshake: Server hello

- Kate (the SSL server) responds with a `Hello`
  - Response contains the cipher suite that Kate selected from the list Joe sent her
- Kate also sends Joe a signed digital certificate that contains her public key
  - Joe needs a signed certificate in a keystore that he can use to verify Kate's signature

Hello Kate

Hello Joe

Joe - client

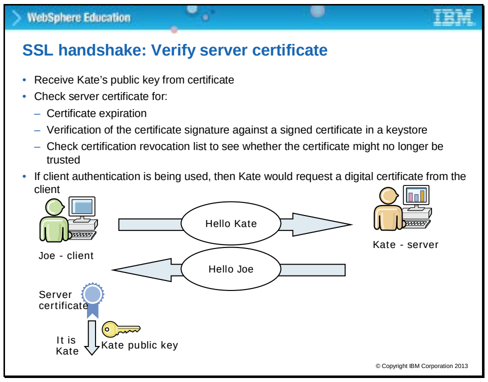
Kate - server

Server certificate

SSL handshake: server hello

The server receives the message and looks at the list of cipher suites. The message is compared to the cipher suites supported by the server. Whichever is the most secure suite that both sides can support is chosen. The hello response that is sent back to the client contains the cipher suite and a digital certificate. The authenticity of the certificate is verified by the client by comparing it to what the client already has on their machine.

Slide 9



WebSphere Education

SSL handshake: Verify server certificate

- Receive Kate’s public key from certificate
- Check server certificate for:
  - Certificate expiration
  - Verification of the certificate signature against a signed certificate in a keystore
  - Check certification revocation list to see whether the certificate might no longer be trusted
- If client authentication is being used, then Kate would request a digital certificate from the client

Joe - client

Hello Kate

Kate - server

Hello Joe

Server certificate

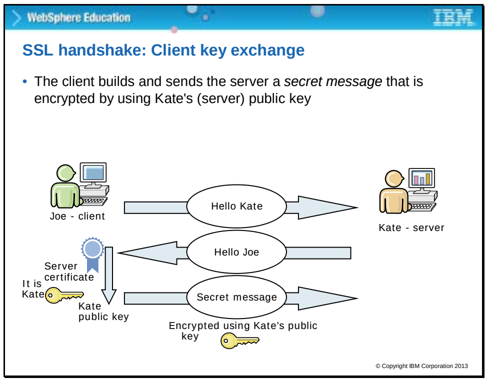
It is Kate

Kate public key

SSL handshake: verify server certificate

The client checks things such as the expiration date of the certificate, whether it is on a revocation list, and whether the signature matches the signature that the client has. The server might also ask the client to send a digital certificate for mutual authentication. So the client would respond with their own certificate and the server would follow the exact same steps that the client took: expiration verification, signature verification, and revocation verification.

Slide 10



WebSphere Education

SSL handshake: Client key exchange

- The client builds and sends the server a *secret message* that is encrypted by using Kate's (server) public key

Joe - client

Hello Kate

Kate - server

Hello Joe

Server certificate

It is Kate

Kate public key

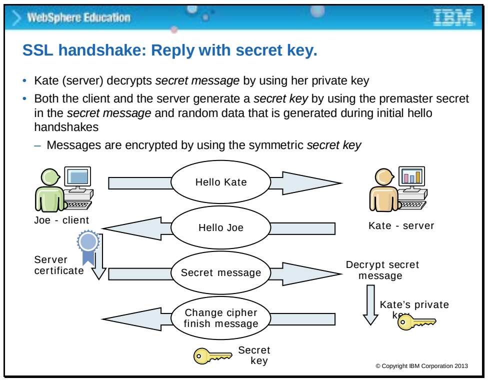
Secret message

Encrypted using Kate's public key

SSL handshake: client key exchange

The certificate the server sent to the client includes the public key, and so the client is now in a position to switch from clear communication to secret communication. Using the public key, Joe can send an encrypted message to the server. The server can decrypt this message with the private key. Now both sides have a message that is passed secretly and so cannot be read outside of this communication.

Slide 11



WebSphere Education

SSL handshake: Reply with secret key.

- Kate (server) decrypts *secret message* by using her private key
- Both the client and the server generate a *secret key* by using the premaster secret in the *secret message* and random data that is generated during initial hello handshakes
  - Messages are encrypted by using the symmetric *secret key*

Joe - client

Hello Kate

Kate - server

Hello Joe

Server certificate

Secret message

Decrypt secret message

Kate's private key

Change cipher finish message

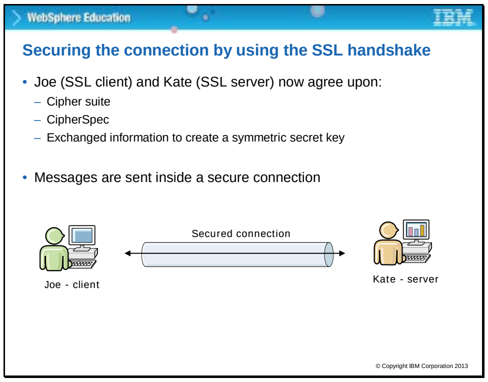
Secret key

© Copyright IBM Corporation 2013

SSL handshake: reply with secret key

The secret message, together with the random data that was sent right back at the beginning of this exchange of messages, is now used to algorithmically set up a secret key. The server sends a message with this new key, and the client responds by using the new key. If both sides can understand the exchange, they know that a secure session is set up.

Slide 12



WebSphere Education

IBM

Securing the connection by using the SSL handshake

- Joe (SSL client) and Kate (SSL server) now agree upon:
  - Cipher suite
  - CipherSpec
  - Exchanged information to create a symmetric secret key
- Messages are sent inside a secure connection

Secured connection

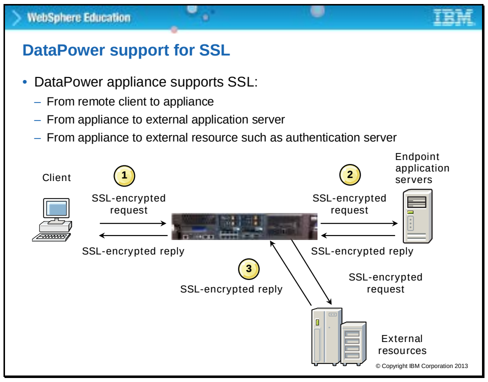
Joe - client

Kate - server

SSL handshake secured

Both sides now use the same secret key, and so they are using symmetric keys. The initial secure exchange was done by using asymmetric keys.

Slide 13


WebSphere Education

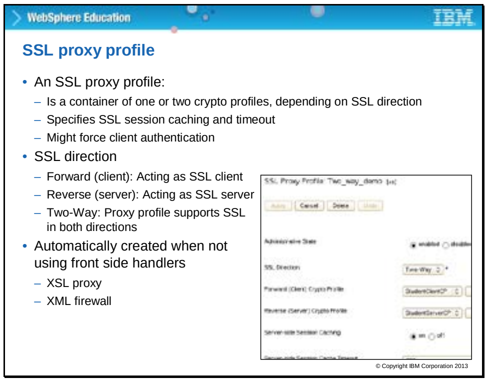
DataPower support for SSL

- DataPower appliance supports SSL:
  - From remote client to appliance
  - From appliance to external application server
  - From appliance to external resource such as authentication server

DataPower support for SSL

DataPower supports SSL both as a server and as a client. As server, it is DataPower that is called from some client, and DataPower sends its authentication certificate. As a client, DataPower requests a secure session from either a back-end application server or some other resource such as an LDAP server.

Slide 14



WebSphere Education

SSL proxy profile

- An SSL proxy profile:
  - Is a container of one or two crypto profiles, depending on SSL direction
  - Specifies SSL session caching and timeout
  - Might force client authentication
- SSL direction
  - Forward (client): Acting as SSL client
  - Reverse (server): Acting as SSL server
  - Two-Way: Proxy profile supports SSL in both directions
- Automatically created when not using front side handlers
  - XSL proxy
  - XML firewall

SSL Proxy Profile: Two_way_demo [up]

Administrative State ● enabled ○ disabled

SSL Direction Two-Way *

Forward (Client) Crypto Profile StudentClientCP

Reverse (Server) Crypto Profile StudentServerCP

Server-side Session Caching ● on ○ off

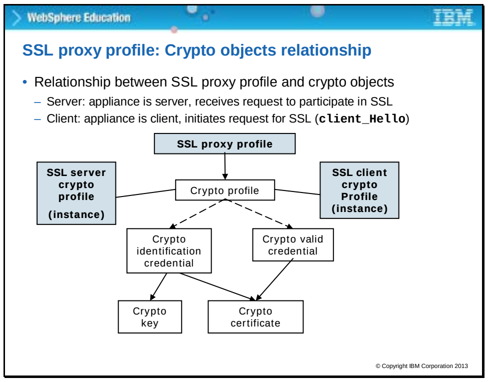
Server-side Session Cache Timeout

SSL proxy profile

An SSL proxy profile Is a container of one or two crypto profiles, depending on SSL direction. The profile specifies the SSL session caching and timeout, and might also force client authentication.

The SSL direction is one of three states; 1) Forward (client): Acting as SSL client, 2) Reverse (server): Acting as SSL server, and 3) Two-Way: Proxy profile supports SSL in both directions.

Slide 15



WebSphere Education

SSL proxy profile: Crypto objects relationship

- Relationship between SSL proxy profile and crypto objects
  - Server: appliance is server, receives request to participate in SSL
  - Client: appliance is client, initiates request for SSL (`client_Hello`)

SSL proxy profile

SSL server crypto profile (instance)

Crypto profile

SSL client crypto Profile (instance)

Crypto identification credential

Crypto valid credential

Crypto key

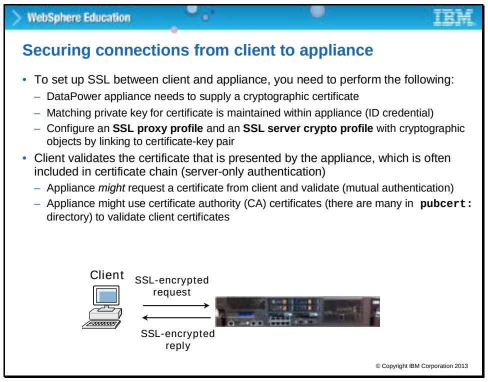
Crypto certificate

SSL Proxy profile: Crypto objects relationship

In the previous unit, you briefly considered the crypto profile and examined the objects that are associated with it. Here is the diagram that was built up in the previous unit. Review the different parts.

There is the crypto identification credential, which links the private key with its certificate – and therefore with the matching public key. There is also a crypto validation credential that lists all the certificates on the appliance that can be used to validate a received certificate. Both the crypto identification credential and the crypto validation credential are associated in a crypto profile. As seen in the previous unit, the crypto profile object points to one or more of these underlying objects, depending on the type of communication: is DataPower the server, or is DataPower the client, and is mutual authentication involved

Slide 16



WebSphere Education

Securing connections from client to appliance

- To set up SSL between client and appliance, you need to perform the following:
  - DataPower appliance needs to supply a cryptographic certificate
  - Matching private key for certificate is maintained within appliance (ID credential)
  - Configure an **SSL proxy profile** and an **SSL server crypto profile** with cryptographic objects by linking to certificate-key pair
- Client validates the certificate that is presented by the appliance, which is often included in certificate chain (server-only authentication)
  - Appliance *might* request a certificate from client and validate (mutual authentication)
  - Appliance might use certificate authority (CA) certificates (there are many in **`pubcert:`** directory) to validate client certificates

Client

SSL-encrypted request

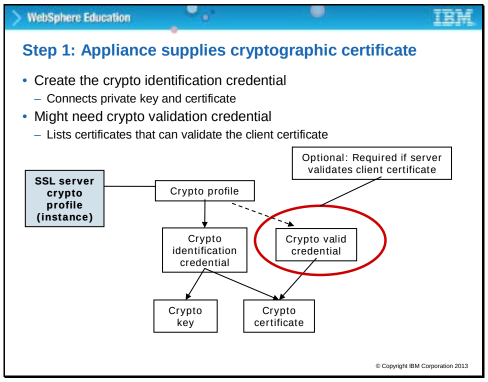
SSL-encrypted reply

© Copyright IBM Corporation 2013

Securing connections from client to appliance

If DataPower is the server, it serves a digital certificate back to the calling client and tracks which private key is associated with that certificate. As you saw, if mutual authentication is required, DataPower simultaneously asks the client for a certificate, and then compare it with the certificates available on the appliance.

Slide 17



WebSphere Education

Step 1: Appliance supplies cryptographic certificate

- Create the crypto identification credential
  - Connects private key and certificate
- Might need crypto validation credential
  - Lists certificates that can validate the client certificate

Optional: Required if server validates client certificate

SSL server crypto profile (instance)

Crypto profile

Crypto identification credential

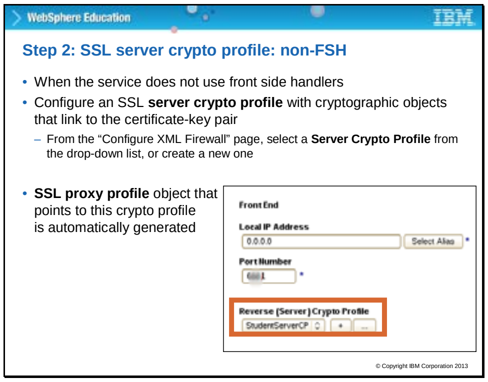
Crypto valid credential

Crypto key

Crypto certificate

Step 1: Appliance supplies cryptographic certificate
This diagram sums up the objects that are required when the appliance acts as a server. The service has an SSL crypto profile instance that points to the crypto identification credential, which in turn points to the associated private key and certificate (which includes the public key). If mutual identification is the case, the service on the appliance also requires the crypto validation credential to validate the certificate that is sent by the client.

Slide 18



Step 2: SSL server crypto profile: non-FSH

- When the service does not use front side handlers
- Configure an SSL **server crypto profile** with cryptographic objects that link to the certificate-key pair
  - From the “Configure XML Firewall” page, select a **Server Crypto Profile** from the drop-down list, or create a new one
- **SSL proxy profile** object that points to this crypto profile is automatically generated

Front End

Local IP Address: 0.0.0.0 Select Alias *

Port Number: [illegible]1 *

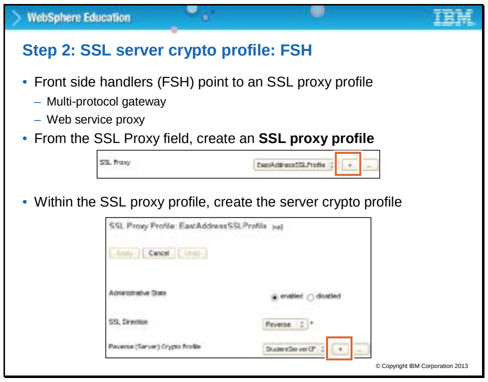
Reverse (Server) Crypto Profile: StudentServerCP

Step 2: SSL server crypto profile: non-FSH

When the service does not use front side handlers, one must configure an SSL server crypto profile with cryptographic objects that link to the certificate-key pair.

The SSL proxy profile object that points to this crypto profile is automatically generated.

Slide 19



Step 2: SSL server crypto profile: FSH

- Front side handlers (FSH) point to an SSL proxy profile
  - Multi-protocol gateway
  - Web service proxy
- From the SSL Proxy field, create an **SSL proxy profile**

SSL Proxy: EastAddressSSLProfile

- Within the SSL proxy profile, create the server crypto profile

SSL Proxy Profile: EastAddressSSLProfile [up]

Administrative State: ● enabled ○ disabled

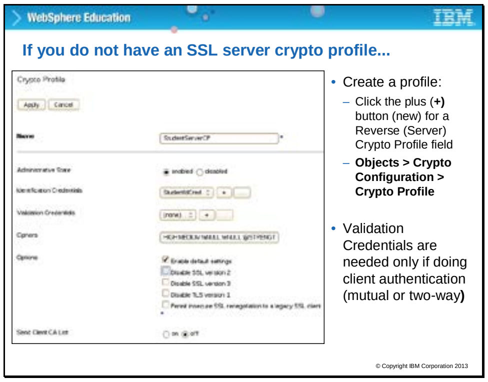
SSL Direction: Reverse *

Reverse (Server) Crypto Profile: StudentServerCP

Step2 SSL server crypto profile front side handler

The Front side handlers (FSH) points to an SSL proxy profile.

Slide 20



If you do not have an SSL server crypto profile...

Crypto Profile

Apply | Cancel

Name: StudentServerCP *

Administrative State: ● enabled ○ disabled

Identification Credentials: StudentIdCred | + | ...

Validation Credentials: (none) | + | ...

Ciphers: HIGH:MEDIUM:!aNULL:!eNULL:@STRENGTH

Options:
☑ Enable default settings
☐ Disable SSL version 2
☐ Disable SSL version 3
☐ Disable TLS version 1
☐ Permit insecure SSL renegotiation to a legacy SSL client

Send Client CA List: ○ on ● off

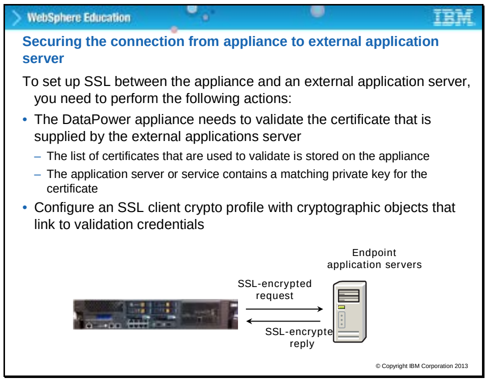
- Create a profile:
  - Click the plus (**+**) button (new) for a Reverse (Server) Crypto Profile field
  - **Objects > Crypto Configuration > Crypto Profile**
- Validation Credentials are needed only if doing client authentication (mutual or two-way**)**

If you do not have an SSL server crypto profile

So what does the dialog look like if you type the plus button as shown on the previous slide? There are many fields to complete! You need to give the server credentials, in other words, you state which certificate you are using. You indicate the trusted certificates that the client might present in the case of mutual identification. You choose which level of SSL of TLS you want to disable, if any. There is also an advanced tab for if you are using an intermediate certificate, or if you want to use a CRL, a certificate revocation list.

There is a second way that you can create the crypto profile, by going to the navigation bar and drilling down through objects, then crypto configuration, then crypto profile. Here you can specify the id credential and if you need it, the validation credential. Next, you can find your crypto profile in the drop-down list of the service wizard.

Slide 21



WebSphere Education

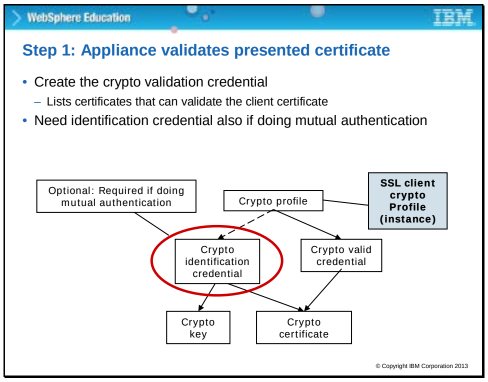
Securing the connection from appliance to external application server

To set up SSL between the appliance and an external application server, you need to perform the following actions:

- The DataPower appliance needs to validate the certificate that is supplied by the external applications server
  - The list of certificates that are used to validate is stored on the appliance
  - The application server or service contains a matching private key for the certificate
- Configure an SSL client crypto profile with cryptographic objects that link to validation credentials

Securing the connection from appliance to external application server
The other possibility is that DataPower is itself the client and is making a request to another server. In this case, DataPower validates the certificate that is presented by the server it is calling to. DataPower requires a client crypto profile object to do this communication. The client crypto profile object is a part of the SSL proxy profile object.

Slide 22



WebSphere Education

Step 1: Appliance validates presented certificate

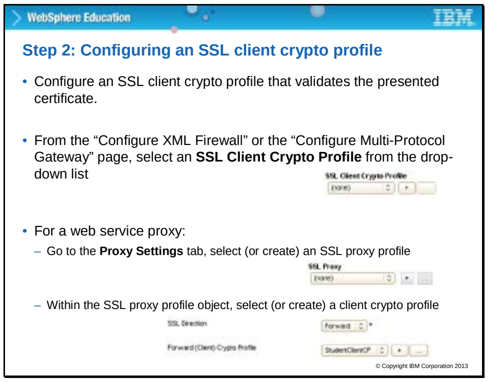
- Create the crypto validation credential
  - Lists certificates that can validate the client certificate
- Need identification credential also if doing mutual authentication

Optional: Required if doing mutual authentication
Crypto profile
SSL client crypto Profile (instance)
Crypto identification credential
Crypto valid credential
Crypto key
Crypto certificate

Step 1: Appliance validates presented certificate

Here schematically is what is involved, and you see immediately that it is the same scenario as for mutual identification when DataPower is the server. The external machine presents a certificate, and DataPower validates it against its validation credential that holds the list of available certificates.

Slide 23



Step 2: Configuring an SSL client crypto profile

- Configure an SSL client crypto profile that validates the presented certificate.
- From the “Configure XML Firewall” or the “Configure Multi-Protocol Gateway” page, select an **SSL Client Crypto Profile** from the drop-down list

SSL Client Crypto Profile: (none) + ...

- For a web service proxy:
  - Go to the **Proxy Settings** tab, select (or create) an SSL proxy profile

SSL Proxy: (none) + ...

  - Within the SSL proxy profile object, select (or create) a client crypto profile

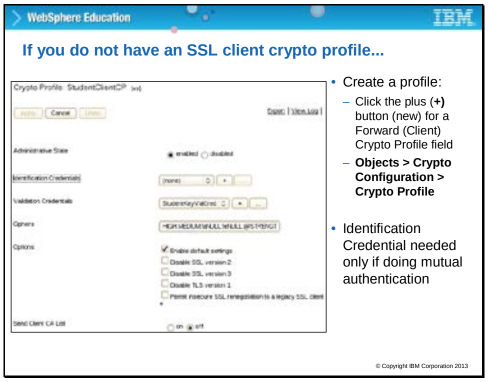
SSL Direction: Forward *

Forward (Client) Crypto Profile: StudentClientCP + ...

Step 2: Configuring an SSL client crypto profile

You can configure the SSL client crypto profile on the main page of the service wizard. You find it under the back-end setup. As for the server crypto profile, you can either make your choice from the drop-down list (and then modify the choice if necessary), or ask for the server crypto profile. Click the plus button and define a new client crypto profile.

Slide 24



WebSphere Education

If you do not have an SSL client crypto profile...

Crypto Profile: StudentClientCP [up]

Apply | Cancel | Undo — Export | View Log |

Administrative State: ● enabled ○ disabled

Identification Credentials: (none) | + | ...

Validation Credentials: StudentKeyValCred | + | ...

Ciphers: HIGH:MEDIUM:!aNULL:!eNULL:@STRENGT

Options:
- ☑ Enable default settings
- □ Disable SSL version 2
- □ Disable SSL version 3
- □ Disable TLS version 1
- □ Permit insecure SSL renegotiation to a legacy SSL client

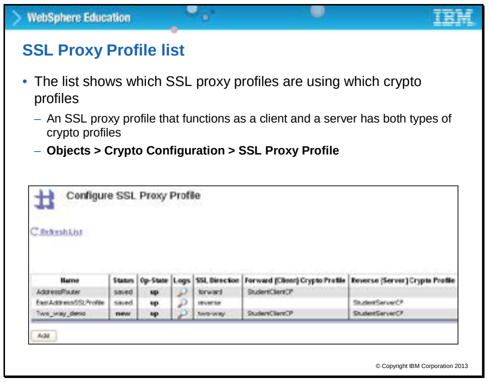
Send Client CA List: ○ on ● off

- Create a profile:
  - Click the plus **(+)** button (new) for a Forward (Client) Crypto Profile field
  - **Objects > Crypto Configuration > Crypto Profile**
- Identification Credential needed only if doing mutual authentication

Step 3: Verify SSL client proxy profile settings

Remember that you saw the profile for the server crypto profile. You can examine the SSL proxy profile that was automatically generated by opening the crypto object from the left navigation bar. The SSL direction is forward; DataPower is the client that is sending the request to a back-end server. When DataPower was the server, the direction here was reverse. The direction can also be two-way if two-way is for DataPower to act both as client and as server.

Slide 25



SSL Proxy Profile list

- The list shows which SSL proxy profiles are using which crypto profiles
  - An SSL proxy profile that functions as a client and a server has both types of crypto profiles
  - **Objects > Crypto Configuration > SSL Proxy Profile**

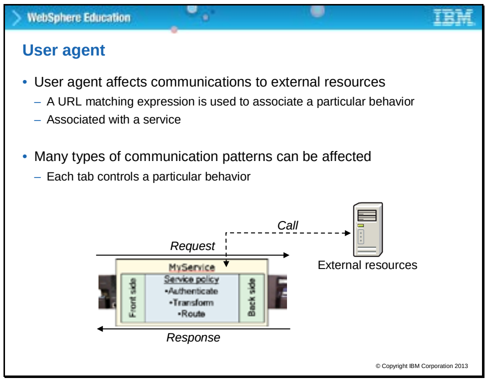
Configure SSL Proxy Profile

Name	Status	Op-State	Logs	SSL Direction	Forward (Client) Crypto Profile	Reverse (Server) Crypto Profile
AddressRouter	saved	up	🔍	forward	StudentClientCP	
EastAddressSSLProfile	saved	up	🔍	reverse		StudentServerCP
Two_way_demo	new	up	🔍	two-way	StudentClientCP	StudentServerCP

SSL Proxy Profile list

The screen capture is what the SSL proxy profile list can look like. Here there are three SSL proxy profiles that are already defined, each one with a different direction. For the first two, the domain configuration is saved already. The third one is created, but not yet saved to the domain configuration. All three are operational. The direction is indicated, and the crypto profile that is being used. Notice that all three SSL proxy profiles use the same crypto profile. Remember that the crypto profile is the top-level object that references the crypto identification credential. The profile is used when DataPower is acting as a server, and also the crypto validation credentials, used when DataPower is acting as a client.

Slide 26



WebSphere Education

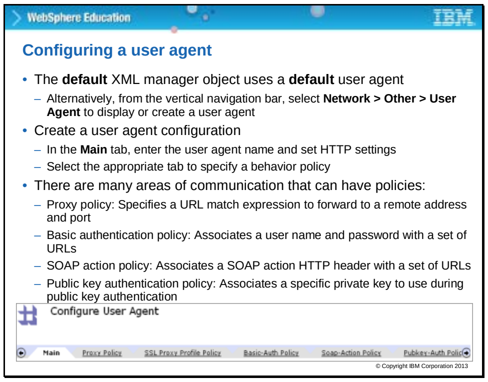
User agent

- User agent affects communications to external resources
  - A URL matching expression is used to associate a particular behavior
  - Associated with a service
- Many types of communication patterns can be affected
  - Each tab controls a particular behavior

User agent

The user agent is a helper object that looks at the URLs that are being called and if it finds a match to some expression it knows, it does something on behalf of the service. It is on the back-end of the appliance, the server side. In the example on this slide, you are looking at the user agent that is configured to instigate an SSL session.

Slide 27



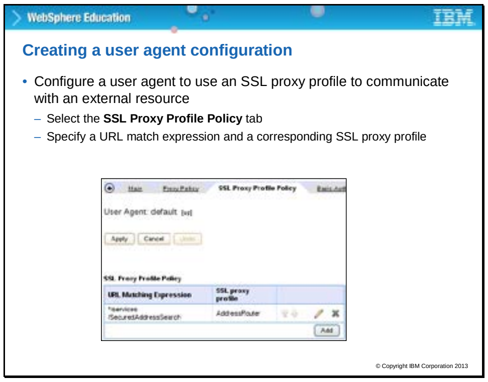
Configuring a user agent

- The **default** XML manager object uses a **default** user agent
  - Alternatively, from the vertical navigation bar, select **Network > Other > User Agent** to display or create a user agent
- Create a user agent configuration
  - In the **Main** tab, enter the user agent name and set HTTP settings
  - Select the appropriate tab to specify a behavior policy
- There are many areas of communication that can have policies:
  - Proxy policy: Specifies a URL match expression to forward to a remote address and port
  - Basic authentication policy: Associates a user name and password with a set of URLs
  - SOAP action policy: Associates a SOAP action HTTP header with a set of URLs
  - Public key authentication policy: Associates a specific private key to use during public key authentication

Configuring a user agent

The XML manager has a default user agent that is associated with it, and you can create your own user agent for specific requirements. You can set the HTTP request headers and a few other things on the main tab. Then, associate a number of different policies with the user agent. The policies are set up so that if this address is a particular URL, the service runs a specific type of invocation. Policies include proxy policy, basic authentication policy, SOAP action policy, or public key authentication policy. There are 10 different tabs for the different policies.

Slide 28



Creating a user agent configuration

- Configure a user agent to use an SSL proxy profile to communicate with an external resource
  - Select the **SSL Proxy Profile Policy** tab
  - Specify a URL match expression and a corresponding SSL proxy profile

User Agent: default [up]

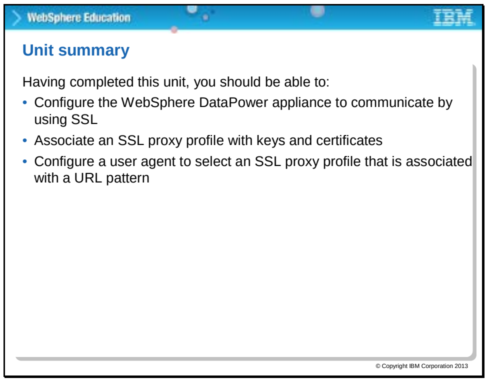
SSL Proxy Profile Policy

URL Matching Expression	SSL proxy profile
*/services /SecuredAddressSearch	AddressRouter

Create a user agent configuration

Here is a typical example. This one is the SSL proxy profile tab of the user agent configuration wizard. It shows that any URL ending with “/resource” is associated with the SSL proxy profile called AddressRouter.

Slide 29

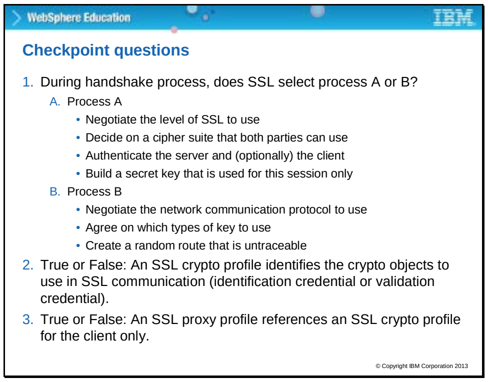


Unit summary

Having completed this unit, you should be able to:

- Configure the WebSphere DataPower appliance to communicate by using SSL
- Associate an SSL proxy profile with keys and certificates
- Configure a user agent to select an SSL proxy profile that is associated with a URL pattern

Slide 30



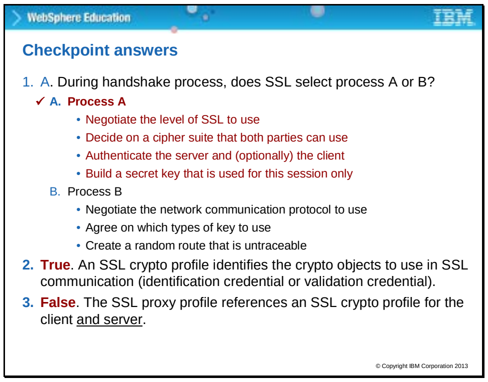
WebSphere Education

IBM

Checkpoint questions

1. During handshake process, does SSL select process A or B?
   - A. Process A
     - Negotiate the level of SSL to use
     - Decide on a cipher suite that both parties can use
     - Authenticate the server and (optionally) the client
     - Build a secret key that is used for this session only
   - B. Process B
     - Negotiate the network communication protocol to use
     - Agree on which types of key to use
     - Create a random route that is untraceable
2. True or False: An SSL crypto profile identifies the crypto objects to use in SSL communication (identification credential or validation credential).
3. True or False: An SSL proxy profile references an SSL crypto profile for the client only.

Slide 31



Checkpoint answers

1. A. During handshake process, does SSL select process A or B?

✓ **A. Process A**

- Negotiate the level of SSL to use
- Decide on a cipher suite that both parties can use
- Authenticate the server and (optionally) the client
- Build a secret key that is used for this session only

B. Process B

- Negotiate the network communication protocol to use
- Agree on which types of key to use
- Create a random route that is untraceable

2. **True**. An SSL crypto profile identifies the crypto objects to use in SSL communication (identification credential or validation credential).
3. **False**. The SSL proxy profile references an SSL crypto profile for the client and server.

Slide 32

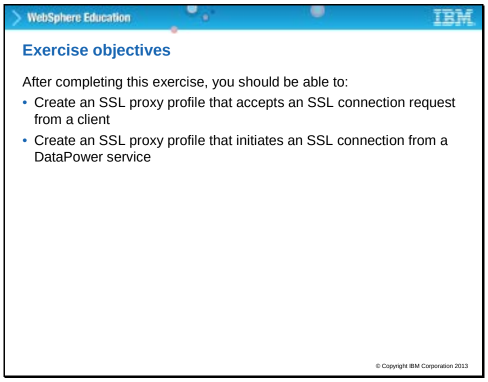
WebSphere Education

IBM

Exercise

Securing connections by using SSL

Slide 33



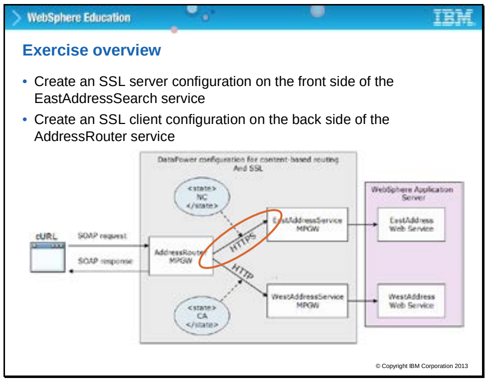
WebSphere Education

Exercise objectives

After completing this exercise, you should be able to:

- Create an SSL proxy profile that accepts an SSL connection request from a client
- Create an SSL proxy profile that initiates an SSL connection from a DataPower service

Slide 34



Exercise overview

- Create an SSL server configuration on the front side of the EastAddressSearch service
- Create an SSL client configuration on the back side of the AddressRouter service