BIG-IP® System: SSL Administration

Version 13.0.0

Table of Contents

About SSL Administration on the BIG-IP System

About SSL administration on the BIG-IP system

TheBIG-IP® system offers a robust set of features for managing SSL traffic. With the BIG-IP system, you can:

- Manage digital certifi cates on BIG-IP systems for secure communication with other BIG-IP systems on the network.
- Manage digital certificates on the BIG-IP system for secure communication with client and server systems on the network.
- Manage SSL profiles to offload client authentication and encryption/decryption tasks from the target server. When offloading SSL tasks for a server, the BIG-IP system can optimize and manipulate the data in user-defined ways before sending the data on to the target server.

Device Certificate Management

About BIG-IP device certificates and keys

Before BIG-IP® systems can exchange data with one another, they need to exchange device certificates, that is, digital certificates and keys used for secure communication. For example, multiple BIG-IP systems might need to verify credentials before communicating with each other to collect performance data over a wide area network, for global traffic management.

A default device certificate and key are located in these directories on the BIG-IP system:

Device certificate file

`/config/httpd/conf/ssl.crt/server.crt`

Device key file

`/config/httpd/conf/ssl.key/server.key`

Note: *The BIG-IP system offers a certificate management user role for managing digital certificates on the BIG-IP system.*

Device certificate requirements

BIG-IP® devices use SSL certificates for authentication and communication among BIG-IP devices on the network. For this authentication and communication between BIG-IP devices to function properly, you should be aware of the following:

- Device certificates must reside in the correct locations on each BIG-IP system.
- Device certificates must be valid and must not be expired.
- BIG-IP device group members require unique device certificates that you must maintain and renew independently.
- You must manage device certificates for any BIG-IP® DNS (previously Global Traffic Manager™) deployment.
- You must manage device certificates for any BIG-IP Application Acceleration Manager™ (AAM®) symmetric deployment.
- For BIG-IP DNS deployments and AAM symmetric deployments, if you update or renew device certificates after they have expired, you must ensure that you copy the new certificates to the remote BIG-IP devices. BIG-IP devices exchange device certificates when running these scripts:

```
bigip_add (BIG-IP DNS and AAM)
big3d_install (BIG-IP DNS only)
```

About trusted device certificates

The BIG-IP® system uses a trusted device certificate or a certificate chain to authenticate another system. For example, a BIG-IP system running BIG-IP® DNS might send a request to a Local Traffic Manager™ system. In this case, the Local Traffic Manager system receiving the request checks its trusted device certificate or certificate chain to authenticate the request.

BIG-IP device certificate management

There are several tasks you can perform to manage device certificates on the BIG-IP® system.

Task list

Importing a device certificate

You can use the Configuration utility to import a device certificate from a management workstation.

1. From the Main tab, click **System** > **Certificate Management** > **Device Certificate Management** > **Device Certificate**.
2. Click **Import**.
3. From the **Import Type** list, select **Certificate**.
4. For the **Certificate Source** setting, select **Upload File** and browse to select the certificate to upload.
5. Click **Import**.

Renewing a device certificate

You can use the Configuration utility to renew a device certificate that has expired.

1. On the Main tab, click **System** > **Certificate Management** > **Device Certificate Management** > **Device Certificate**.
2. Click **Renew**.
3. Modify or retain the device certificate properties.
4. Click **Finished**.

Exporting a device certificate

You can use the Configuration utility to export a device certificate to a management workstation.

1. On the Main tab, click **System** > **Certificate Management** > **Device Certificate Management** > **Device Certificate**.
2. Click **Export**.
3. Click **Download server.crt** to export a copy of the device certificate to the management workstation.

Importing a device certificate/key pair

You can use the Configuration utility to import a device certificate/key pair from a management workstation.

1. On the Main tab, click **System** > **Certificate Management** > **Device Certificate Management** > **Device Key**.
2. Click **Import**.
3. From the **Import Type** list, select **Certificate and Key**.
4. For the **Certificate Source** setting, click **Upload File**.
5. For the **Key Source** setting, click **Upload File**.
6. Click **Import**.

SSL Certificate Management

Supported certificate/key types

The BIG-IP® system supports multiple cipher suites when offloading SSL operations from a target server on the network. The BIG-IP system can support cipher suites that use these algorithms:

- Rivest Shamir Adleman (RSA)
- Elliptic Curve Digital Signature Algorithm (ECDSA)
- Digital Signature Algorithm (DSA)

When you generate a certificate request or a self-signed certificate, you specify the type of private key, which determines the specific signing or encryption algorithm that is used to generate the private key.

Note: *On the BIG-IP system, limits on SSL transactions per second (TPS) with RSA cipher suites vary according to key size.*

About RSA certificates

RSA (Rivest Shamir Adleman) is the original encryption algorithm that is based on the concept of a public and a private key. When a public site attempts to communicate with a device such as the BIG-IP® system, the device sends the site a public key that the site uses to encrypt data before sending that data back to the device. The device uses its private key associated with the public key to decrypt the data. Only the private key can be used to decrypt data encrypted with the public key.

The RSA encryption algorithm includes an authentication mechanism.

Note: *On the BIG-IP system, limits on SSL transactions per second (TPS) with RSA cipher suites vary according to key size.*

About DSA certificates

DSA (Digital Signature Algorithm) uses a different algorithm for signing key exchange messages than that of RSA. *DSA* is paired with a key exchange method such as Diffie-Hellman or Elliptical Curve Diffie-Hellman to achieve a comparable level of security to RSA. Because DSA is generally endorsed by federal agencies, specifying a DSA key type makes it easier to comply with new government standards, such as those for specific key lengths.

About ECDSA certificates

When creating certificates on the BIG-IP® system, you can create a certificate with a key type of ECDSA (Elliptic Curve Digital Signature Algorithm). An *ECDSA key* is based on Elliptic Curve Cryptography (ECC), and provides better security and performance with significantly shorter key lengths.

For example, an RSA key size of 2048 bits is equivalent to an ECC key size of only 224 bits. As a result, less computing power is required, resulting in faster, more secure connections. Encryption based on ECC is ideally suited for mobile devices that cannot store large keys. The BIG-IP system supports both the prime256v1 and secp384r1 curve names.

About SSL certificate management

You can obtain a certificate for the BIG-IP system by using the BIG-IP® Configuration utility to generate a certificate signing request (CSR) that can then be submitted to a third-party trusted certificate authority (CA). The CA then issues a signed certificate.

In addition to requesting CA-signed certificates, you can create self-signed certificates. You create self-signed certificates primarily for testing purposes within an organization.

When you install the BIG-IP software, the application includes a default self-signed certificate. The BIG-IP system also includes a default CA bundle certificate. This certificate bundle contains certificates from most of the well-known CAs.

Note: *To manage digital certificates for the BIG-IP system, you must have a role of Certificate Manager, Administrator, or Resource Administrator assigned to your BIG-IP user account.*

Creating a self-signed digital certificate

If you are configuring the BIG-IP® system to manage client-side HTTP traffic, you perform this task to create a self-signed certificate to authenticate and secure the client-side HTTP traffic. If you are also configuring the system to manage server-side HTTP traffic, you must repeat this task to create a second self-signed certificate to authenticate and secure the server-side HTTP traffic.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**. The Traffic Certificate Management screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Self**.
5. In the **Common Name** field, type a name.

   This is typically the name of a web site, such as `www.siterequest.com`.
6. In the **Division** field, type your department name.
7. In the **Organization** field, type your company name.
8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.

    This name is embedded in the certificate for X509 extension purposes.

    By assigning this name, you can protect multiple host names with a single SSL certificate.
14. From the **Security Type** list, select **NetHSM**.
15. From the **Key Type** list, **RSA** is selected as the default key type.
16. From the **Size** list, select a size, in bits.
17. Click **Finished**.

Requesting a certificate from a certificate authority

You perform this task to generate a certificate signing request (CSR) that can then be submitted to a third-party trusted certificate authority (CA).

Note: *F5 Networks recommends that you consult the CA to determine the specific information required for each step in this task.*

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**.
   The Traffic Certificate Management screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Certificate Authority**.
5. In the **Common Name** field, type a name.
   This is typically the name of a web site, such as `www.siterequest.com`.
6. In the **Division** field, type your department name.
7. In the **Organization** field, type your company name.
8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.
    This name is embedded in the certificate for X509 extension purposes.
    By assigning this name, you can protect multiple host names with a single SSL certificate.
14. In the **Challenge Password** field, type a password.
15. In the **Confirm Password** field, re-type the password you typed in the **Challenge Password** field.
16. From the **Key Type** list, select a key type.
    Possible values are: **RSA**, **DSA**, and **ECDSA**.
17. From the **Size** or **Curve Name** list, select either a size, in bits, or a curve name.
18. If the BIG-IP system contains an internal HSM module, specify a location for storing the private key.
19. Click **Finished**.
    The Certificate Signing Request screen displays.
20. Do one of the following to download the request into a file on your system.
    - In the **Request Text** field, copy the certificate.
    - For **Request File**, click the button.
21. Follow the instructions on the relevant certificate authority web site for either pasting the copied request or attaching the generated request file.
22. Click **Finished**.
    The Certificate Signing Request screen displays.

The generated certificate signing request is submitted to a trusted certificate authority for signature.

About SSL file import

You can import several types of SSL files onto the BIG-IP system.

Importing a certificate signed by a certificate authority

Before performing this task, confirm that a digital certificate signed by a certificate authority (CA) is available.

You can install an SSL certificate signed by a CA by importing a certificate that already exists on the hard drive of the management workstation. You can import a private key, a certificate or certificate bundle, or an archive.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**. The Traffic Certificate Management screen opens.
2. Click the **Import** button.
3. From the **Import Type** list, select **Certificate**.
4. For the **Certificate Name** setting, do one of the following:
   - Select the **Create New** option, and type a unique name in the field.
   - Select the **Overwrite Existing** option, and select a certificate name from the list.
5. For the **Certificate Source** setting, do one of the following:
   - Select the **Upload File** option, and browse to the location of the certificate file.
   - Select the **Paste Text** option, and paste the certificate text copied from another source.
6. Click **Import**.

After you perform this task, the SSL certificate that was signed by a CA is installed.

Importing an SSL key

You can use the BIG-IP® Configuration utility to import an SSL key onto the BIG-IP system from another location.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**. The Traffic Certificate Management screen opens.
2. Click the **Import** button.
3. From the **Import Type** list, select **Key**.
4. For the **Key Name** setting, do one of the following:
   - Select the **Create New** option, and type a unique name in the field.
   - Select the **Overwrite Existing** option, and select a certificate name from the list.
5. For the **Key Source** setting, do one of the following:
   - Select the **Upload File** option, and browse to the location of the key file.
   - Select the **Paste Text** option, and paste the key text copied from another source.
6. In the **Password** field, type the password associated with the import source.
7. from the **Security Type** list, select a security type.
8. Click **Import**.

After you perform this task, the BIG-IP system imports the specified key.

Importing a PKCS-formatted file

You can use the BIG-IP® Configuration utility to import file onto the BIG-IP system that is in Public Key Cryptography Standards (PKCS) number 12 format.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**. The Traffic Certificate Management screen opens.
2. Click the **Import** button.
3. From the **Import Type** list, select **PKCS 12 (IIS)**.
4. For the **Certificate Name** setting, type a certificate name.
5. For the **Certificate Source** setting, click **Browse** and locate the source file.
6. In the **Password** field, type the password associated with the import source.
7. from the **Security Type** list, select a security type.
8. Click **Import**.

After you perform this task, the BIG-IP system imports the specified PKCS 12-formatted file.

Importing an archive file

You can use the BIG-IP® Configuration utility to upload an archive file onto the BIG-IP system.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**. The Traffic Certificate Management screen opens.
2. Click the **Import** button.
3. For the **Upload Archive File** setting, click **Browse** and select the file to be imported.
4. Click the **Load** button.

After you perform this task, the BIG-IP system uploads an archive file onto the BIG-IP system.

Exporting an SSL certificate

You perform this task to export an SSL certificate to another device.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**. The Traffic Certificate Management screen opens.
2. Click the name of the certificate you want to export. The General Properties screen displays.
3. Click **Export**. The Certificate Export screen displays the contents of the certificate in the **Certificate Text** box.
4. To obtain the certificate, do one of the following:
   - Copy the text from the **Certificate Text** field, and paste it as needed into an interface on another system.
   - At the **Certificate File** option, click **Download filename** where filename is the name of the certificate file, such as `mycert.crt`.

Viewing a list of certificates on the system

You can perform this task to view a list of existing digital certificates on the BIG-IP® system.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**. The Traffic Certificate Management screen opens.
2. In the Name column, view the list of certificates on the system.

Digital SSL certificate properties

From the BIG-IP® Configuration utility, you can see the properties of the SSL digital certificates you have installed on the BIG-IP® system.

Property	Description
Certificate	The name of the certificate.
Content	The type of certificate content, for example, Certificate Bundle or Certificate and Key.
Common name	The common name (CN) for the certificate. The common name embedded in the certificate is used for name-based authentication. The default common name for a self-signed certificate is `localhost.localdomain`.
Expiration date	The date that the certificate expires. If the certificate is a bundle, this information shows the

Property	Description
	range of expiration dates that apply to certificates in the bundle.
Organization	The organization name for the certificate. The organization name embedded in the certificate is used for name-based authentication. The default organization for a self-signed certificate is `MyCompany`.

About certificate bundle management

You can use the bundle manager to automatically update and install certificate authority (CA) bundles on the system from two sources: local certificate file objects and remote URL resources. By using the **Include Bundles** and **Include URLs** options, you can combine CA certificates from various sources to create a new, customized CA bundle. You can also use the **Exclude Bundles** and **Exclude URLs** options to remove certain CA certificates from the resulting CA bundle file. The newly created or modified CA bundle file is installed as a certificate-file-object on the system and used as a trusted CA bundle by other modules.

In addition, you can set the update frequency of the CA bundle, or use a web proxy for downloading the remote URL resources. By default, a newly created CA bundle manager does not create or update the managed CA bundle object. Exceptions are if the CA bundle manager has a positive update interval or is explicitly told to do so since you have set the **Update Now** option.

Creating a new certificate bundle

You can create a new certificate authority (CA) bundle, and specify bundles and URLs to include or exclude. You can also set the update frequency of the CA bundle, or use a web proxy for downloading the remote URL resources.

Note: *The resulting bundle file will be named the same as the bundle manager object.*

Note: *By default, a newly created CA bundle manager does not create or update the managed CA bundle object unless the CA bundle manager has a positive* ***Update Interval*** *or is explicitly told to do so by the* ***Update Now*** *option.*

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management** > **Bundle Manager List**.

   The Bundle Manager List screen opens.
2. Click **Create**.
3. From the **Include Bundles Available** list, select the certificate file objects to include for generating a new CA bundle.
4. In the **Include URLs** field, type the URL where remote CA bundles reside, and click **Add** to include that for generating the new CA bundle.

   Only HTTPS URLs are allowed in the **Include URLs** fields.
5. From the **Exclude Bundles Available** list, select the certificate file objects to exclude from the new CA bundle.
6. In the **Exclude URLs** field, type the URL where remote CA bundles reside, and click **Add** to exclude it from the new CA bundle.

   Only HTTPS URLs are allowed in the **Exclude URLs** fields.

7. In the **Update Interval** field, type the number of days at which to refresh the remote CA bundles at the URLs.

   Note: *The default value is set to **0** and indicates that the generated CA bundle is not dynamically updated.*

8. If you want the CA bundle manager to immediately refresh its generated CA bundle from all its sources and recalculate its certificate contents, select the **Update Now** check box.

   Note: *The default value is disabled.*

9. From the **Trusted CA-Bundle** list, select the CA bundle that this CA bundle manager will use to download remote CA bundles in the include and exclude URLs.
10. In the **Proxy Server** field, type the host name or IP address of the proxy server for accessing remote URL resources.

    Note: *Only HTTP proxy is supported. You may optionally prepend* `http://` *to the host name or IP address.*

11. In the **Proxy Server Port** field, type the port number of the proxy server for accessing remote URL resources.

    Note: *The default is **3128**.*

12. In the **Download Timeout** field, specify the timeout period, in seconds, to download the remote CA bundles from the URLs.

    The value range is from 1 to 3600 (1 hour) seconds.

    Note: *The default value is **8** seconds.*

13. Click **Finished**.

The system installs a generated CA bundle file as a certificate-file-object on the system to be used as a trusted CA bundle by other modules.

Modifying an existing certificate bundle

You can use the bundle manager to modify an existing certificate authority (CA) bundle.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management** > **Bundle Manager List**.

   The Bundle Manager List screen opens listing all existing CA bundles and their name, update interval, proxy server, trusted CA-bundle, and partition/path details.

2. From the **Bundle Manager List**, click the name of the CA bundle that you want to modify.
   The Properties screen opens showing the selected CA bundle general properties and configuration details
3. Select the **Update Now** check box if you want the bundle to be updated.
4. Modify any of the configuration details needed, and click **Update**.

The system updates the selected CA bundle's configuration with the modified configuration details.

Deleting an existing certificate bundle

You can use the bundle manager to delete an existing certificate authority (CA) bundle.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management** > **Bundle Manager List**.

   The Bundle Manager List screen opens listing all existing CA bundles and their name, update interval, proxy server, trusted CA-bundle, and partition/path details.
2. Select the check box next to the name of the CA bundle that you want to delete.
3. Click **Delete**.

Note: *You can also delete a CA bundle on the Properties screen by clicking* ***Delete*** *at the bottom of the screen.*

Note: *Deleting the CA bundle manager does not delete the managed CA bundle file object. You should delete the CA bundle file object separately or you might receive an error message indicating that your managed CA bundle file object is referenced by a CA bundle manager.*

This deletes the selected CA bundle from the system.

SSL Traffic Management

About SSL offload

When you want the BIG-IP system to process application traffic over SSL, you can configure the system to perform the SSL handshake that destination servers normally perform. This ability for the BIG-IP system to offload SSL processing from a destination server is an important feature of the BIG-IP system.

The most common way to configure the BIG-IP system is to create a Client SSL profile, which makes it possible for the BIG-IP system to decrypt client requests before sending them on to a server, and encrypt server responses before sending them back to the client.

Within a Client SSL profile specifically, you can specify multiple certificate/key pairs, one per key type. This enables the system to accept all types of cipher suites that a client might support as part of creating a secure connection. The system then decrypts the client data, manipulates any headers or payload according to the way that you configured the Client SSL profile, and by default, sends the request in clear text to the target server for processing.

For those sites that require enhanced security on their internal network, you can configure a Server SSL profile. With a Server SSL profile, the BIG-IP system re-encrypts the request before sending it to the destination server. When the server returns an encrypted response, the BIG-IP system decrypts and then re-encrypts the response, before sending the response back to the client.

About client-side and server-side SSL profiles

You can manage the way that the BIG-IP system processes SSL application traffic by configuring two types of SSL profiles: A Client SSL profile, a Server SSL profile, or both. These profiles affect the way that the system manages SSL traffic passing through the system.

When you configure Client SSL or Server SSL profiles and assign them to a virtual server, the BIG-IP system offloads SSL processing from the destination server. This offloading not only conserves resource on destination servers, but enables the BIG-IP system to customize SSL traffic processing according to your configuration specifications.

Creating a custom Client SSL profile

After you have built the cipher string that you want the BIG-IP to use to negotiate client-side SSL connections, you create a custom Client SSL profile. You create the profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and encrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads these decryption/encryption functions from the destination server. When you perform this task, you can specify multiple certificate key chains, one for each key type (RSA, DSA, and ECDSA). This allows the BIG-IP system to negotiate secure client connections using different cipher suites based on the client's preference.

Note: *At a minimum, you must specify a certificate key chain that includes an RSA key pair. Specifying certificate key chains for DSA and ECDSA key pairs is optional, although highly recommended.*

---

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client SSL profile list screen opens.

2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.
   The settings become available for change.
6. For the **Certificate Key Chain** setting, click **Add**.
   a) From the **Certificate** list, select a certificate name.

      This is the name of a certificate that you installed on the BIG-IP® system. If you have not generated a certificate request nor installed a certificate on the BIG-IP system, and the BIG-IP system is not part of a device service clustering (DSC) configuration, you can specify the name of the existing certificate named `default`.

      Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default certificate name, and ensure that this same certificate name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*

   b) From the **Key** list, select the name of the key associated with the certificate specified in the previous step.

      This is the name of a key that you installed on the BIG-IP® system. If you have not installed a key on the BIG-IP system, and the BIG-IP system is not part of a device service clustering (DSC) configuration, you can specify the name of the existing key named `default`.

      Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default key name, and ensure that this same key name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*

   c) From the **Chain** list, select the chain that you want to include in the certificate key chain.

      A certificate chain can contain either a series of public key certificates in Privacy Enhanced Mail (PEM) format or a series of one or more PEM files. A certificate chain can contain certificates for Intermediate certificate Authorities (CAs).

      Note: *The default self-signed certificate and the default CA bundle certificate are not appropriate for use as a certificate chain.*

   d) For the **Passphrase** field, type a string that enables access to SSL certificate/key pairs that are stored on the BIG-IP system with password protection.

      This setting is optional. For added security, the BIG-IP system automatically encrypts the pass phrase itself. This pass phrase encryption process is invisible to BIG-IP® system administrative users.

   e) Click **Add**.
7. Click **Add** and repeat the process for all certificate key chains that you want to specify. At a minimum, you must specify an RSA certificate key chain.

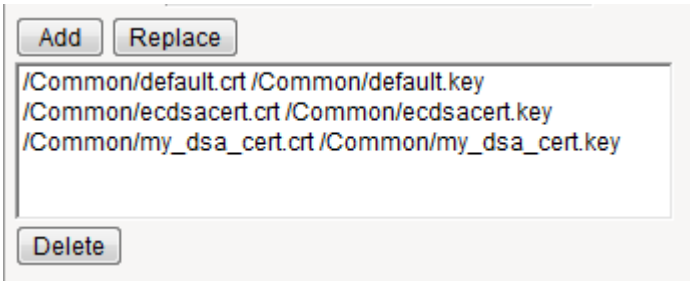
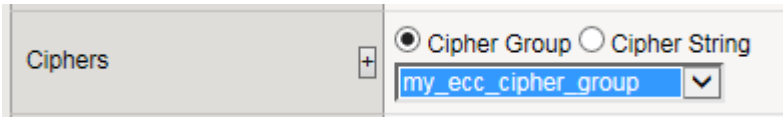


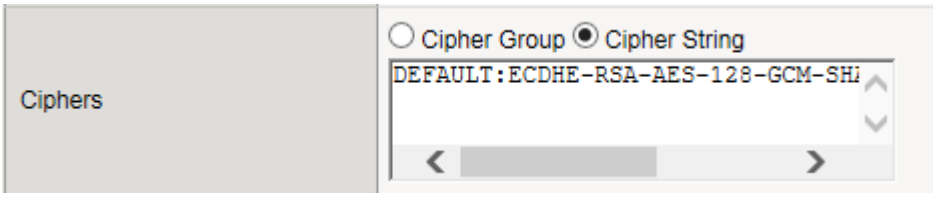
Figure 1: Sample configuration with three key types specified

The result is that all specified key chains appear in the text box.

8. For the **OCSP Stapling** setting, select the check box.
   This setting is optional. To enable OCSP stapling, you must first create an OCSP Stapling profile.
9. For the **Notify Certificate Status to Virtual Server** setting, select the check box.
   This setting is optional.
10. For the **Ciphers** setting, specify a cipher group or cipher string by choosing one of these options.

Note: *If you specified an ECDSA certificate key chain in the* ***Certificate Key Chain*** *setting, you must include the cipher string* `ECDHE_ECDSA` *in the cipher group or cipher string that you specify in the* ***Ciphers*** *setting. (At a minimum, you should specify a cipher group or string such as* `DEFAULT:ECDHE_ECDSA`*.) This is necessary to ensure successful cipher negotiation when the BIG-IP system is offered an ECDSA-based certificate only.*

Option	Description
Cipher Group	Select an existing cipher group from the list when you want to use a system-defined or custom cipher group to define the ciphers that the BIG-IP system uses for negotiating SSL connections. Here's an example of the `Ciphers` setting where we've selected a custom cipher group that we created earlier. 
Cipher String	Type a cipher string in the box if you want to manually specify a cipher string instead of selecting a cipher group. For security and performance reasons, consider following these recommendations: • Always append ciphers to the `DEFAULT` cipher string. • Type a cipher string that includes the ECC key type, because its shorter length speeds up encryption and decryption while still offering virtually the same level of security. • Disable ADH ciphers but also include the keyword `HIGH`. To do this, just include both `!ADH` and `:HIGH` in your cipher string. • For AES, DES, and RC4 encryption types, make sure you specify the DHE key exchange method. DHE uses *Forward Privacy*, which creates a key that it throws away after each session so that the same session key never gets used twice. When you use DHE, make sure that the SSL private key isn't being shared with a monitoring system or a security device like an intrusion detection or prevention system. Also, diagnostic tools like `ssldump` won't work when you're using Forward Secrecy.

Option	Description
	• Disable EXPORT ciphers by including `!EXPORT` in the cipher string. • If you can live with removing support for the SSLv3 protocol version, do it. This protocol version is not secure. Simply include `:!SSLv3` in any cipher string you type. Here's an example of the `Ciphers` setting where we have opted to manually type the cipher string `DEFAULT:ECDHE-RSA-AES-128-GCM-SHA256:!ADH:!EXPORT:HIGH`: 

11. Configure all other profile settings as needed.
12. Click **Finished**.

After performing this task, you can see the custom Client SSL profile in the list of Client SSL profiles on the system.

You must also assign the profile to a virtual server.

Creating a custom Server SSL profile

With a Server SSL profile, the BIG-IP® system can perform decryption and encryption for server-side SSL traffic.

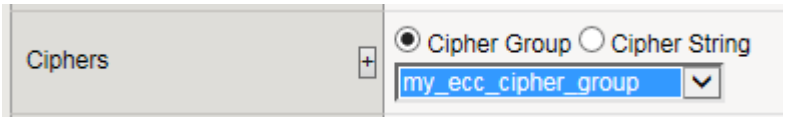
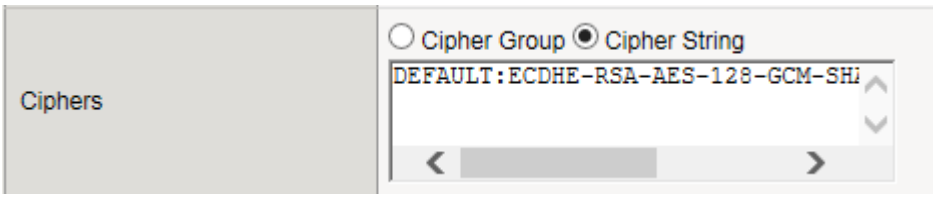
1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The Server SSL profile list screen opens.
2. Click **Create**.
   The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **serverssl** in the **Parent Profile** list.
5. From the **Configuration** list, select **Advanced**.
6. Select the **Custom** check box.
   The settings become available for change.
7. From the **Certificate** list, select the name of an SSL certificate on the BIG-IP system.

   Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default certificate name, and ensure that this same certificate name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*
8. From the **Key** list, select the name of an SSL key on the BIG-IP system.

   Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default key name, and ensure that this same key name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*
9. In the **Pass Phrase** field, select a pass phrase that enables access to the certificate/key pair on the BIG-IP system.

10. From the **Chain** list, select the name of an SSL chain on the BIG-IP system.
11. For the **Ciphers** setting, specify a cipher group or cipher string by choosing one of these options.

Note: *If you specified an ECDSA certificate key chain in the* ***Certificate Key Chain*** *setting, you must include the cipher string* `ECDHE_ECDSA` *in the cipher group or cipher string that you specify in the* ***Ciphers*** *setting. (At a minimum, you should specify a cipher group or string such as* `DEFAULT:ECDHE_ECDSA`*.) This is necessary to ensure successful cipher negotiation when the BIG-IP system is offered an ECDSA-based certificate only.*

Option	Description
Cipher Group	Select an existing cipher group from the list when you want to use a system-defined or custom cipher group to define the ciphers that the BIG-IP system uses for negotiating SSL connections. Here's an example of the `Ciphers` setting where we've selected a custom cipher group that we created earlier. 
Cipher String	Type a cipher string in the box if you want to manually specify a cipher string instead of selecting a cipher group. For security and performance reasons, consider following these recommendations: • Always append ciphers to the `DEFAULT` cipher string. • Type a cipher string that includes the ECC key type, because its shorter length speeds up encryption and decryption while still offering virtually the same level of security. • Disable ADH ciphers but also include the keyword `HIGH`. To do this, just include both `!ADH` and `:HIGH` in your cipher string. • For AES, DES, and RC4 encryption types, make sure you specify the DHE key exchange method. DHE uses *Forward Privacy*, which creates a key that it throws away after each session so that the same session key never gets used twice. When you use DHE, make sure that the SSL private key isn't being shared with a monitoring system or a security device like an intrusion detection or prevention system. Also, diagnostic tools like `ssldump` won't work when you're using Forward Secrecy. • Disable EXPORT ciphers by including `!EXPORT` in the cipher string. • If you can live with removing support for the SSLv3 protocol version, do it. This protocol version is not secure. Simply include `:!SSLv3` in any cipher string you type. Here's an example of the `Ciphers` setting where we have opted to manually type the cipher string `DEFAULT:ECDHE-RSA-AES-128-GCM-SHA256:!ADH:!EXPORT:HIGH`: 

12. Select the **Custom** check box for **Server Authentication**.
13. Modify the settings, as required.
14. Click **Finished**.

To use this profile, you must assign it to a virtual server.

Assigning SSL profiles to a virtual server

The final task in the process of implementing SSL profiles is to assign the SSL profile to a virtual server. If the relevant virtual server does not yet exist, you can assign the SSL profile (or profiles) to the virtual server when you create it.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of a virtual server.
3. From the **Configuration** list, select **Advanced**.
4. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created and move the name to the **Selected** list.
5. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL profile you previously created and move the name to the **Selected** list.
6. Click **Update** to save the changes.

Support for multiple key types

For client-side traffic specifically, you can configure a Client SSL profile to specify multiple certificate key chains on the BIG-IP® system, one for each key type: RSA, DSA, and ECDSA. By configuring a Client SSL profile with different digital certificates and keys, the system can accept all types of cipher suites that clients might request as part of creating a secure connection. The system supports OCSP stapling for all certificate and key types.

Important: *To ensure successful negotiation, the BIG-IP system requires you to specify an RSA-based certificate key chain at a minimum, to accommodate any RSA-based ciphers that the client presents. However, F5 Networks highly recommends that you also specify DSA and ECDSA certificate key chains.*

About OCSP stapling

When you create a Client SSL profile, you can specify Online Certificate Status Protocol (OCSP) stapling to improve the certification response time. *OCSP stapling* is when a TLS server (acting as OCSP client) asks the OCSP server for a valid revocation status of its TLS certificate ahead of time and "staples" the signed OCSP response to the TLS handshake. The TLS client sees the stapled OCSP response and verifies the signature, thus validating the TLS server's certificate and eliminating the round trip at the client for fetching the certificate status. It also helps protect the identity of the client.

By default, OCSP stapling is disabled. You can can create an OCSP stapling profile and then enable it from within a Client SSL profile. There is no default OCSP Stapling profile, so you must create one that specifies the parameters you want to use. For example, the default OCSP stapling profile setting is to use a DNS resolver to fetch the OCSP response, and you must specify the DNS resolver to use. Alternatively, you can choose to use a proxy server to fetch the OCSP response, and then you must specify the proxy server pool.

Creating an OCSP stapling profile

Ensure that you configure a proxy server pool or a DNS resolver.

When you create an OCSP stapling profile and assign it to a client SSL profile, you speed up the time it takes for the client to get the certificate revocation status of the BIG-IP® system.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management** > **OCSP>**.
2. Click **Create**.
   The new OCSP Stapling Profile screen opens.
3. In the **Name** field, type a unique name for the OCSP stapling profile.
4. For the **Use Proxy Server** check box, select one of the following options:

Option	Description
Select	If you want the BIG-IP system to use the **Proxy Server Pool**. Use when there are one or more servers that can proxy an HTTP request to an external server and fetch the response.
Clear (default)	If you want to use the **DNS Resolver**. Use when the OCSP responder can be reached on one of the BIG-IP interfaces.

5. In the **Proxy Server Pool/DNS Resolver** field, select the proxy server pool/DNS resolver used for fetching the OCSP response.
6. In the **Trusted Certificate Authorities** field, select the name of the file containing a trusted Certificate Authority (CA) certificate used to sign the responder's certificate.
7. In the **Trusted Responders** field, select the name of a certificate to use to verify the response from the OCSP responder.
8. In the **Responder URL** field, type the name of a URL that will override the OCSP responder URL obtained from the certificate's AIA extension. This must be an HTTP or HTTPS-based URL.
9. In the **Signer Certificate** field, select a certificate corresponding to the key used for signing the OCSP request.
10. In the **Signer Key** field, select a key to use to sign an OCSP request.
11. In the **Signer Key Passphrase** field, type the passphrase of the key used to sign an OCSP request.
12. In the **Sign Hash** field, select the hash algorithm used to sign an OCSP request. The default is **SHA256**.

    Note: *This is not the algorithm used in the certificate itself. It is what the OCSP responder will use when validating the request.*

    ---

13. In the **Timeout** field, type a time interval for the BIG-IP system to wait before dropping the connection to the OCSP responder.
14. In the **Clock Skew** field, type a value for the maximum tolerable absolute difference between the clocks of the responder and the BIG-IP system.
15. In the **Status Age** field, type a value for the maximum allowed lag time in the OCSP response that the BIG-IP system accepts. If you type `0`, the validation is skipped. The default value is **86400**.
16. In the **Cache Timeout** field, select a value that specifies the lifetime of the OCSP response. The default is **Indefinite**, indicating that the response validity period takes precedence.
17. In the **Cache Error Timeout** field, type a value for how long a BIG-IP system will cache an error response.
18. In the **Options** field, if necessary, select the **Strict Responder Certificate Checking** check box for the system to check the responder's certificate for the OCSP signing extension.
19. Click **Finished**.

After you create an OCSP profile, make sure you enable the **OCSP Stapling** setting from within the relevant Client SSL profile.

About BIG-IP cipher support

The BIG-IP® system supports a large set of cipher suites that you can choose from to build the cipher string used for security negotiation.

Supported cipher suites include various combinations of encryption algorithms and authentication mechanisms, including RSA (Rivest Shamir Adleman), DSA (Digital Signature Algorithm), and ECDSA (Elliptic Curve Digital signature Algorithm).

The system includes a default cipher string named `DEFAULT`, which contains a subset of the cipher suites that the BIG-IP system supports.

Glossary of cipher-related terms

This list defines terms for cipher-related features that the BIG-IP ® system supports.

Cipher suite
A combination of authentication, encryption, message authentication code (MAC), and key exchange algorithms that the BIG-IP system can offer to a client and server system when negotiating security for an SSL network connection.

Cipher string
A string that contains the cipher suite or suites that the BIG-IP system can use to negotiate security for an SSL connection.

Cipher rule
A named BIG-IP configuration object that contains a cipher string. The BIG-IP system offers several pre-built cipher rules. Examples are `f5-default`, `f5-ecc`, and `f5-secure`, which represent the cipher strings `DEFAULT`, `ECDHE:ECDHE_ECDSA`, and `ECDHE:RSA:!SSLV3:!RC4:!EXP:!DES`, respectively.

Cipher group
A named BIG-IP configuration object that specifies one or more cipher rules, with instructions for how you want the BIG-IP system to apply those rules to an SSL connection. The BIG-IP system offers several pre-built cipher groups, such as `f5-default`, `f5-ecc`, and `f5-secure`, where each cipher group includes one corresponding pre-built cipher rule. You can use a pre-built cipher group, or you create a custom cipher group.

Cipher audit
The actual cipher suites that the BIG-IP system can offer to a client or server system as part of negotiating an SSL connection, after resolving any mismatches.

About the DEFAULT cipher suite

Some of the cipher suites that the BIG-IP system supports are included in a default cipher string named `DEFAULT`.

The `DEFAULT` cipher string appears as the default value in the **Ciphers** setting of the Client SSL and Server SSL profiles.

Important: *We strongly recommend that you use the default cipher string and, for added security, append other cipher suites to it.*

View cipher suites in cipher string DEFAULT

Before you use this command, confirm that your BIG-IP® user account grants you access to the advanced shell.

Follow these steps to display the list of cipher suites currently included in cipher string `DEFAULT`, for both client-side and server-side SSL connections.

1. Using an SSH client application such as PuTTY, access the advanced shell on the BIG-IP system.
2. At the system prompt, for client-side negotiation, type the command `tmm --clientciphers DEFAULT`. For server-side negotiation, type `tmm --serverciphers DEFAULT`.

   For example, here's a partial list of the cipher suites currently included in the `DEFAULT` cipher suite for client-side traffic:

```
login as: admin
Password:
Last login: Wed Oct 26 13:26:00 2016 from 172.18.42.59
[admin@bigip11:Active:Standalone] ~ # tmm --clientciphers DEFAULT
      ID  SUITE                          BITS PROT    METHOD  CIPHER    MAC     KEYX
 0: 49199  ECDHE-RSA-AES128-GCM-SHA256     128  TLS1.2  Native  AES-GCM   SHA256  ECDHE_RSA
 1: 49171  ECDHE-RSA-AES128-CBC-SHA        128  TLS1    Native  AES       SHA     ECDHE_RSA
 2: 49171  ECDHE-RSA-AES128-CBC-SHA        128  TLS1.1  Native  AES       SHA     ECDHE_RSA
 3: 49171  ECDHE-RSA-AES128-CBC-SHA        128  TLS1.2  Native  AES       SHA     ECDHE_RSA
 4: 49191  ECDHE-RSA-AES128-SHA256         128  TLS1.2  Native  AES       SHA256  ECDHE_RSA
 5: 49200  ECDHE-RSA-AES256-GCM-SHA384     256  TLS1.2  Native  AES-GCM   SHA384  ECDHE_RSA
 6: 49172  ECDHE-RSA-AES256-CBC-SHA        256  TLS1    Native  AES       SHA     ECDHE_RSA
 7: 49172  ECDHE-RSA-AES256-CBC-SHA        256  TLS1.1  Native  AES       SHA     ECDHE_RSA
 8: 49172  ECDHE-RSA-AES256-CBC-SHA        256  TLS1.2  Native  AES       SHA     ECDHE_RSA
 9: 49192  ECDHE-RSA-AES256-SHA384         256  TLS1.2  Native  AES       SHA384  ECDHE_RSA
10:   156  AES128-GCM-SHA256               128  TLS1.2  Native  AES-GCM   SHA256  RSA
11:    47  AES128-SHA                      128  TLS1    Native  AES       SHA     RSA
12:    47  AES128-SHA                      128  TLS1.1  Native  AES       SHA     RSA
13:    47  AES128-SHA                      128  TLS1.2  Native  AES       SHA     RSA
14:    47  AES128-SHA                      128  DTLS1   Native  AES       SHA     RSA
15:    60  AES128-SHA256                   128  TLS1.2  Native  AES       SHA256  RSA
16:   157  AES256-GCM-SHA384                256  TLS1.2  Native  AES-GCM   SHA384  RSA
17:    53  AES256-SHA                      256  TLS1    Native  AES       SHA     RSA
18:    53  AES256-SHA                      256  TLS1.1  Native  AES       SHA     RSA
19:    53  AES256-SHA                      256  TLS1.2  Native  AES       SHA     RSA
20:    53  AES256-SHA                      256  DTLS1   Native  AES       SHA     RSA
21:    61  AES256-SHA256                   256  TLS1.2  Native  AES       SHA256  RSA
22:    65  CAMELLIA128-SHA                 128  TLS1    Native  CAMELLIA  SHA     RSA
23:    65  CAMELLIA128-SHA                 128  TLS1.1  Native  CAMELLIA  SHA     RSA
24:    65  CAMELLIA128-SHA                 128  TLS1.2  Native  CAMELLIA  SHA     RSA
25:   132  CAMELLIA256-SHA                 256  TLS1    Native  CAMELLIA  SHA     RSA
26:   132  CAMELLIA256-SHA                 256  TLS1.1  Native  CAMELLIA  SHA     RSA
28: 49195  ECDHE-ECDSA-AES128-GCM-SHA256    128  TLS1.2  Native  AES-GCM   SHA256  ECDHE_ECDSA
29: 49161  ECDHE-ECDSA-AES128-SHA           128  TLS1    Native  AES       SHA     ECDHE_ECDSA
30: 49161  ECDHE-ECDSA-AES128-SHA           128  TLS1.1  Native  AES       SHA     ECDHE_ECDSA
31: 49161  ECDHE-ECDSA-AES128-SHA           128  TLS1.2  Native  AES       SHA     ECDHE_ECDSA
32: 49187  ECDHE-ECDSA-AES128-SHA256        128  TLS1.2  Native  AES       SHA256  ECDHE_ECDSA
33: 49196  ECDHE-ECDSA-AES256-GCM-SHA384    256  TLS1.2  Native  AES-GCM   SHA384  ECDHE_ECDSA
34: 49162  ECDHE-ECDSA-AES256-SHA           256  TLS1    Native  AES       SHA     ECDHE_ECDSA
35: 49162  ECDHE-ECDSA-AES256-SHA           256  TLS1.1  Native  AES       SHA     ECDHE_ECDSA
36: 49162  ECDHE-ECDSA-AES256-SHA           256  TLS1.2  Native  AES       SHA     ECDHE_ECDSA
37: 49188  ECDHE-ECDSA-AES256-SHA384        256  TLS1.2  Native  AES       SHA384  ECDHE_ECDSA
38:   158  DHE-RSA-AES128-GCM-SHA256        128  TLS1.2  Native  AES-GCM   SHA256  EDH/RSA
39:    51  DHE-RSA-AES128-SHA              128  TLS1    Native  AES       SHA     EDH/RSA
40:    51  DHE-RSA-AES128-SHA              128  TLS1.1  Native  AES       SHA     EDH/RSA
41:    51  DHE-RSA-AES128-SHA              128  TLS1.2  Native  AES       SHA     EDH/RSA
42:    51  DHE-RSA-AES128-SHA              128  DTLS1   Native  AES       SHA     EDH/RSA
43:   103  DHE-RSA-AES128-SHA256           128  TLS1.2  Native  AES       SHA256  EDH/RSA
44:   159  DHE-RSA-AES256-GCM-SHA384        256  TLS1.2  Native  AES-GCM   SHA384  EDH/RSA
45:    57  DHE-RSA-AES256-SHA              256  TLS1    Native  AES       SHA     EDH/RSA
46:    57  DHE-RSA-AES256-SHA              256  TLS1.1  Native  AES       SHA     EDH/RSA
47:    57  DHE-RSA-AES256-SHA              256  TLS1.2  Native  AES       SHA
48:    57  DHE-RSA-AES256-SHA              256  DTLS1   Native  AES       SHA
49:   107  DHE-RSA-AES256-SHA256           256  TLS1.2  Native  AES       SHA25
50:    69  DHE-RSA-CAMELLIA128-SHA         128  TLS1    Native  CAMELLIA  SHA
51:    69  DHE-RSA-CAMELLIA128-SHA         128  TLS1.1  Native  CAMELLIA  SHA
52:    69  DHE-RSA-CAMELLIA128-SHA         128  TLS1.2  Native  CAMELLIA  SHA
53:   136  DHE-RSA-CAMELLIA256-SHA         256  TLS1    Native  CAMELLIA  SHA
54:   136  DHE-RSA-CAMELLIA256-SHA         256  TLS1.1  Native  CAMELLIA  SHA
55:   136  DHE-RSA-CAMELLIA256-SHA         256  TLS1.2  Native  CAMELLIA  SHA
56: 49170  ECDHE-RSA-DES-CBC3-SHA           168  TLS1    Native  DES       SHA
57: 49170  ECDHE-RSA-DES-CBC3-SHA           168  TLS1.1  Native  DES       SHA
58: 49170  ECDHE-RSA-DES-CBC3-SHA           168  TLS1.2  Native  DES       SHA
```

What is a cipher group?

A *cipher group* contains a list of cipher rules, and the instructions that the BIG-IP® system needs for building the cipher string it will use for security negotiation. The instructions tell the system which cipher rules to include in the string, and how to apply them (allow, disallow, and so on, and in what order).

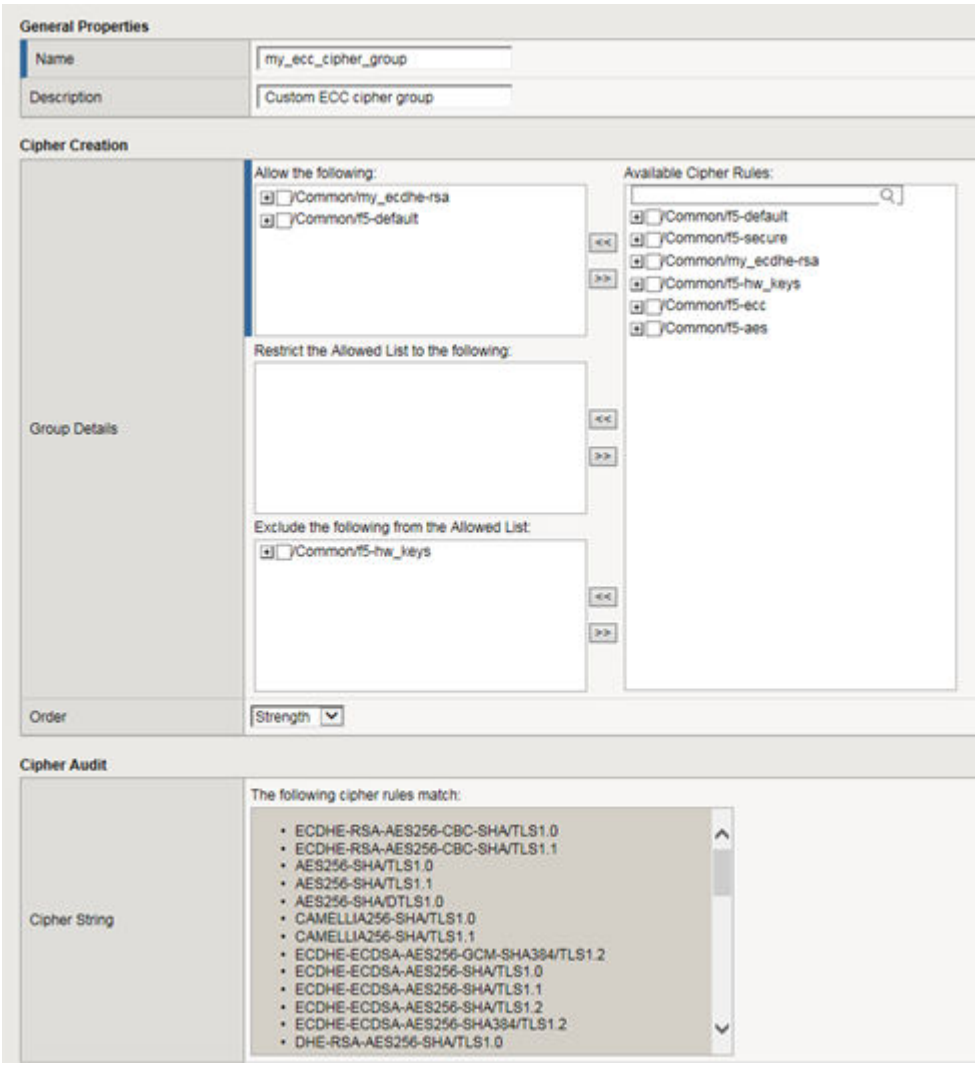
Pre-built cipher groups

The BIG-IP system offers a few pre-built cipher groups that you can choose from to use as is to build your final cipher string, However, it's common to create your own custom cipher group instead.

Custom cipher groups

This illustration shows an example of a custom cipher group. Using this cipher group, the BIG-IP system builds the final cipher string using a user-created custom cipher rule named `/Common/my_ecdhe_rsa` and the pre-built cipher rule `/Common/f5-default`.

Notice that the system will exclude from the string any cipher suites defined in the pre-built cipher rule `/Common/f5-hw_keys`.



Also notice that the cipher group displays a preview of the final cipher string after the instructions are applied.

What is a cipher rule?

A *cipher rule* is a partial cipher string, with a name, that contains one or more cipher suites. You can combine these cipher rules to create a custom cipher group, which the BIG-IP® system uses to build the final cipher string that the BIG-IP system will use for SSL negotiation with client and server systems.

An example of a cipher rule might be one that specifies only cipher suites using a particular bulk encryption algorithm or a particular key exchange algorithm.

Pre-built cipher rules

The BIG-IP system offers a set of pre-built cipher rules, with names containing the prefix `f5-`. This table lists these cipher rules and the cipher strings they represent.

Table 1: Pre-built cipher rules and their contents

Cipher rule name	Associated cipher string
`f5-aes`	AES
`f5-default`	DEFAULT
`f5-ecc`	ECDHE:ECDHE_ECDSA
`f5-hw_keys`	ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES256-CBC-SHA:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-SHA256:DHE-RSA-AES256-SHA:ECDH-RSA-AES256-GCM-SHA384:ECDH-RSA-AES256-SHA384:ECDH
`f5-secure`	ECDHE:RSA:!SSLV3:!RC4:!EXP:!DES

Custom cipher rules

If none of the pre-built cipher rules contains the cipher suites you need, you can create your own cipher rules to include in a custom cipher group. You can also combine your own cipher rules with pre-built ones.

Here's an example of a custom cipher rule that you can create for an Elliptic Curve cipher suite:

Best practices for BIG-IP cipher strings

For security and performance reasons, consider the following recommendations:

- Always append cipher suites to the `DEFAULT` cipher string.
- Include a cipher string that specifies the ECC key type, because its shorter length speeds up encryption and decryption while still offering virtually the same level of security.
- Disable ADH ciphers but also include the keyword `HIGH`. To do this, just include both `!ADH` and `:HIGH` in your cipher string.
- For AES, DES, and RC4 encryption types, make sure you specify the DHE key exchange method. DHE uses *Forward Privacy*, which creates a key that it throws away after each session so that the same session key never gets used twice. When you use DHE, make sure that the SSL private key isn't being shared with a monitoring system or a security device like an intrusion detection or prevention system. And by the way, diagnostic tools like `ssldump` won't work when you're using Forward Secrecy.
- Disable EXPORT ciphers by including `!EXPORT` in the cipher string.

- If you can live with removing support for the SSLv3 protocol version, do it. This protocol version is unsecure. Simply include `:!SSLv3` in any cipher string you build.

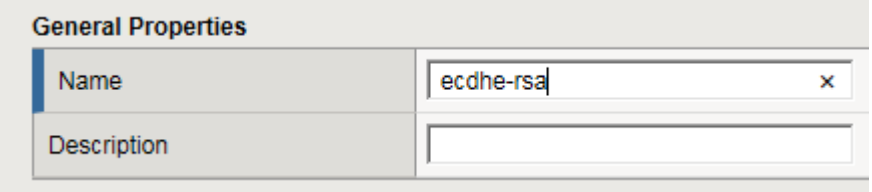
Create partial cipher strings to include in a custom cipher string

When you create your own cipher rules for a custom cipher group, the BIG-IP® system can build a cipher string that includes or excludes the cipher suites you need for negotiating SSL connections.

1. On the Main tab, click **Local Traffic** > **Ciphers** > **Rules**.
   The screen displays a list of pre-built cipher rules.
2. Click **Create**.
3. In the **Name** field, type a name for the cipher rule.

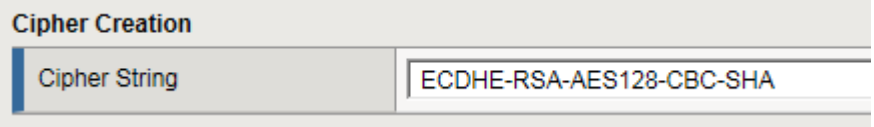
   Note: *Never include the prefix `f5-` in a cipher rule name. This prefix is reserved for pre-built cipher rules only.*

   For example:



4. In the **Cipher String** field, type a cipher string that represents one or more cipher suites.

   For example:



5. Click **Finished**.

The cipher rule now appears within any custom cipher group, in the list of available cipher rules.

Build a custom cipher string

Before starting this task, make sure you've confirmed the need to create a custom cipher string instead of using a pre-built cipher group.

You build a final, custom cipher string by creating a cipher group. A *cipher group* contains the cipher rules and instructions that the BIG-IP® system needs for building the cipher string it will use for security negotiation with a client or server system.

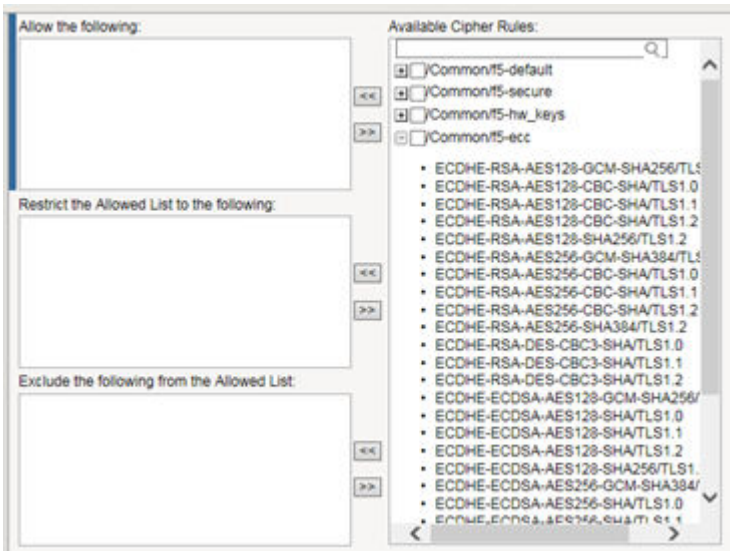
1. On the Main tab, click **Local Traffic** > **Ciphers** > **Groups**.
   The screen displays a list of pre-built cipher groups.
2. Click **Create**.
3. In the **Name** field, type a name for the cipher group.

   Note: *Never include the prefix `f5-` in a cipher rule name. This prefix is reserved for pre-built cipher groups only.*

4. If you created any custom rules, then in the Cipher Creation area of the screen in the **Available Cipher Rules** list, verify that the custom rules appear in the list.

5. For each cipher rule in the **Available Cipher Rules** list, click the plus sign to view the cipher suites included in the rule.

   For example, this shows the cipher suites included in the pre-built cipher rule named `/Common/f5-ecc`.



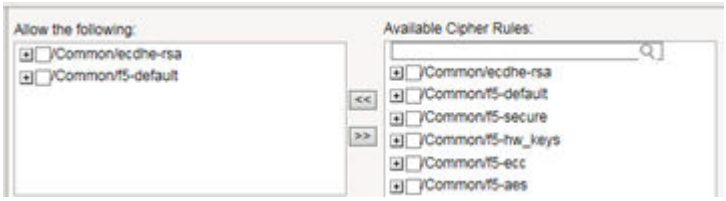
6. In the **Available Cipher Rules** list, select the boxes for the cipher rules you want to allow for negotiating security for SSL connections.

   Important: *We strongly recommend that you select the cipher rule* `/Common/f5-default`*, and for added security, select other cipher rules, too.*

   Here's an example of a list of available cipher rules that you might see within a cipher group. Notice that we've selected both a pre-built cipher rule and a custom cipher rule:

7. Move the selected cipher rules to the **Allow the following** box.

   Here we see that we're instructing the BIG-IP system to allow, during security negotiation, the cipher suites contained in the selected cipher rules:



8. Again from the **Available Cipher Rules** list, select the boxes for the cipher rules you want to restrict the allowed cipher rules to when negotiating security for SSL connections.
9. Using the Move button, move the selected cipher rules to the **Restrict the Allowed list to the following** box.
10. If you want to exclude any cipher rules from the allowed list, then from the **Available Cipher Rules** list, select the boxes for the rules you want to exclude.

11. Using the Move button, move the selected cipher rules to the **Exclude the following from the Allowed list** box.
12. From the **Order** list, select the order that you want the BIG-IP system to use when negotiating SSL connections.

    The choices are: **Default**, **Speed**, **Strength**, **FIPS**, and **Hardware**.
13. In the Cipher Audit area of the screen, view the cipher string that the BIG-IP system will construct based on the selections you made in the previous steps.
14. Click **Finished**.

After you complete this task, the BIG-IP system has a custom cipher group that the BIG-IP system will use to build the final cipher string.

About Elliptic Curve encryption

The BIG-IP system supports Elliptic Curve Cryptography (ECC). Because Elliptic Curve key sizes are significantly smaller than those of other key types, ECC is ideally suited for smaller, mobile devices for which key storage is an issue. On the BIG-IP system, ECC works with the SSL offload feature.

About Elliptic Curve cipher support

The BIG-IP system supports multiple ciphers that use Elliptic Curve Cryptography (ECC) encryption with Diffie-Hellman key exchange. On the BIG-IP system, EC is used with DHE to establish the shared secret; however, the subsequent bulk encryption of data cannot be done with any ECC-based algorithm and must be done using conventional crypto algorithms such as AES and 3DES. For example, a typical Elliptic Curve cipher is: ECDHE-RSA-AES128-CBC-SHA.

The specific ECC ciphers that the BIG-IP system supports are:

- ECDHE-RSA-*
- ECDHE-ECDSA-*
- ECDH-ECDSA-*

Because ECC with Diffie-Hellman does not include a mechanism for digitally signing handshake messages, the RSA or DSA algorithms are used to digitally sign the handshake messages to thwart Man-in-the-Middle attacks. For example, an ECDHE-ECDSA-* cipher suite uses the ECC DSA certificate specified in the Client SSL profile to digitally sign the handshake messages.

Note: *Elliptic Curve ciphers with DSA are not included in the* ***DEFAULT*** *cipher suite.*

Specifying the use of Elliptic Curve ciphers

Use this task to modify an existing Client SSL profile to enable support for Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) key exchange.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client** or **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The Client SSL or Server SSL profile list screen opens.
2. In the Name column, click the name of the profile you want to modify.
3. Select the **Custom** check box.
   The settings become available for change.
4. For the **Ciphers** setting, click **Cipher String** and type `DEFAULT:ECDHE`.
5. Click **Update**.

After you perform this task and assign the profile to a virtual server, the BIG-IP system uses the ECDHE key exchange method to establish secure communication with the relevant client or server.

Viewing ECDH key exchange statistics

You can use the Traffic Management Shell (tmsh) to view statistics about the use of Elliptic Curve Diffie-Hellman ciphers in SSL negotiation.

1. Access the system prompt on the BIG-IP system.
2. From the BIG-IP system prompt, type `tmsh show ltm profile client-ssl` *`profile_name`*`|` grep ECDH.
   An example of a name for a profile that specifies DHE ciphers is `my_ecdh_profile`.

After you type this command, the BIG-IP system displays output such as the following. In this example, the profile statistics show that ECDH with RSA certificates has been used six times:

```
[root@server35:Active:Standalone] config # tmsh show ltm profile client-ssl myclientssl
| grep ECDH
  Ephemeral ECDH w/ RSA Certs                    6
```

Figure 2: Sample profile statistics for key exchange method

About Diffie-Hellman Ephemeral key exchange

The BIG-IP® system supports the Diffie-Hellman Ephemeral key exchange method, as well as other Diffie-Hellman variations. A *Diffie–Hellman key exchange method* is an alternative to RSA key exchange and allows the client and the BIG-IP system to establish a shared secret session key to use for communication.

About DHE cipher support

Because Diffie-Hellman key exchange methods do not include authentication, use of Diffie-Hellman Ephemeral (DHE) requires that it be paired with an authentication mechanism. The DHE ciphers that the BIG-IP system supports are:

- DHE-RSA-* (Diffie-Hellman Ephemeral-RSA)
- DHE-DSS-* (Diffie-Hellman Ephemeral-DSS)
- ECDHE-RSA-* (Elliptic Curve Diffie-Hellman Ephemeral-RSA)
- ECDHE-ECDSA-* (Elliptic Curve Diffie-Hellman Ephemeral-DSA)

Note: *For DHE, the* `DEFAULT` *cipher suite includes Elliptic Curve cipher suites only. DHE ciphers for RSA and DSS encryption are not included.*

Viewing a list of supported DHE ciphers

Before using this command, confirm that your user account grants you access to the advanced shell.

You perform this task when you want to display a specific set of ciphers that the BIG-IP system supports.

1. Access the advanced shell on the BIG-IP system.
2. At the system prompt, type the command `tmm --clientciphers` *`ciphers`*.
   a) For example, to see a list of DHE+DES ciphers, type `tmm --clientciphers DHE:DHE_DSS`
   The BIG-IP system displays the list of all DHE+DES ciphers that the BIG-IP system supports:

```
      ID  SUITE                          BITS PROT    METHOD  CIPHER  MAC
0:    21  DHE-RSA-DES-CBC-SHA             64  SSL3    Native  DES     SHA
1:    21  DHE-RSA-DES-CBC-SHA             64  TLS1    Native  DES     SHA
2:    21  DHE-RSA-DES-CBC-SHA             64  TLS1.1  Native  DES     SHA
3:    21  DHE-RSA-DES-CBC-SHA             64  TLS1.2  Native  DES     SHA
```

Figure 3: Supported DHE+DES ciphers on the BIG-IP system

b) To see a list of ECDHE ciphers, type `tmm --clientciphers ECDHE:ECDHE_ECDSA`.
The BIG-IP system displays the list of all ECDHE ciphers that the BIG-IP system supports:

```
      ID  SUITE                              BITS PROT    METHOD  CIPHER  MAC      KEYX
 0: 49200  ECDHE-RSA-AES256-GCM-SHA384       256  TLS1.2  Native  AES-GCM  SHA384   ECDHE_RSA
 1: 49192  ECDHE-RSA-AES256-SHA384           256  TLS1.2  Native  AES      SHA384   ECDHE_RSA
 2: 49172  ECDHE-RSA-AES256-CBC-SHA          256  TLS1    Native  AES      SHA      ECDHE_RSA
 3: 49172  ECDHE-RSA-AES256-CBC-SHA          256  TLS1.1  Native  AES      SHA      ECDHE_RSA
 4: 49172  ECDHE-RSA-AES256-CBC-SHA          256  TLS1.2  Native  AES      SHA      ECDHE_RSA
 5: 49170  ECDHE-RSA-DES-CBC3-SHA            192  TLS1    Native  DES      SHA      ECDHE_RSA
 6: 49170  ECDHE-RSA-DES-CBC3-SHA            192  TLS1.1  Native  DES      SHA      ECDHE_RSA
 7: 49170  ECDHE-RSA-DES-CBC3-SHA            192  TLS1.2  Native  DES      SHA      ECDHE_RSA
 8: 49199  ECDHE-RSA-AES128-GCM-SHA256       128  TLS1.2  Native  AES-GCM  SHA256   ECDHE_RSA
 9: 49191  ECDHE-RSA-AES128-SHA256           128  TLS1.2  Native  AES      SHA256   ECDHE_RSA
10: 49171  ECDHE-RSA-AES128-CBC-SHA          128  TLS1    Native  AES      SHA      ECDHE_RSA
11: 49171  ECDHE-RSA-AES128-CBC-SHA          128  TLS1.1  Native  AES      SHA      ECDHE_RSA
12: 49171  ECDHE-RSA-AES128-CBC-SHA          128  TLS1.2  Native  AES      SHA      ECDHE_RSA
13: 49196  ECDHE-ECDSA-AES256-GCM-SHA384     256  TLS1.2  Native  AES-GCM  SHA384   ECDHE_ECDSA
14: 49188  ECDHE-ECDSA-AES256-SHA384         256  TLS1.2  Native  AES      SHA384   ECDHE_ECDSA
15: 49162  ECDHE-ECDSA-AES256-SHA            256  TLS1    Native  AES      SHA      ECDHE_ECDSA
16: 49162  ECDHE-ECDSA-AES256-SHA            256  TLS1.1  Native  AES      SHA      ECDHE_ECDSA
17: 49162  ECDHE-ECDSA-AES256-SHA            256  TLS1.2  Native  AES      SHA      ECDHE_ECDSA
18: 49160  ECDHE-ECDSA-DES-CBC3-SHA          192  TLS1    Native  DES      SHA      ECDHE_ECDSA
19: 49160  ECDHE-ECDSA-DES-CBC3-SHA          192  TLS1.1  Native  DES      SHA      ECDHE_ECDSA
20: 49160  ECDHE-ECDSA-DES-CBC3-SHA          192  TLS1.2  Native  DES      SHA      ECDHE_ECDSA
21: 49195  ECDHE-ECDSA-AES128-GCM-SHA256     128  TLS1.2  Native  AES-GCM  SHA256   ECDHE_ECDSA
22: 49187  ECDHE-ECDSA-AES128-SHA256         128  TLS1.2  Native  AES      SHA256   ECDHE_ECDSA
23: 49161  ECDHE-ECDSA-AES128-SHA            128  TLS1    Native  AES      SHA      ECDHE_ECDSA
24: 49161  ECDHE-ECDSA-AES128-SHA            128  TLS1.1  Native  AES      SHA      ECDHE_ECDSA
25: 49161  ECDHE-ECDSA-AES128-SHA            128  TLS1.2  Native  AES      SHA      ECDHE_ECDSA
```

Figure 4: Supported ECDHE ciphers on the BIG-IP system

Specifying the use of Diffie-Hellman ciphers

Use this task to modify an existing Client SSL profile to enable support for Diffie-Hellman key exchange.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client** or **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The Client SSL or Server SSL profile list screen opens.
2. In the Name column, click the name of the profile you want to modify.
3. Select the **Custom** check box.
   The settings become available for change.
4. To specify DHE ciphers:
   a) From the **Configuration** list, select **Advanced**.
   b) For the **Ciphers** setting, click **Cipher String** and type `DHE:DHE_DSS`.
5. Click **Update**.

After you perform this task and assign the profile to a virtual server, the BIG-IP uses the DHE key exchange method to establish secure communication with the relevant client or server.

Viewing DHE key exchange statistics

You can use the Traffic Management Shell (tmsh) to view statistics about the use of Diffie-Hellman ciphers in SSL negotiation.

1. Access the system prompt on the BIG-IP system.
2. From the BIG-IP system prompt, type `tmsh show ltm profile client-ssl` *profile_name*. An example of a name for a profile that specifies DHE ciphers is `my_dhe_profile`.

After you type this command, the BIG-IP system displays output such as the following. In this example, the profile statistics show that Diffie-Hellman Ephemeral (DHE) with RSA certificates has been used once:

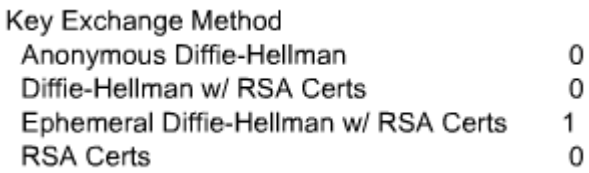


Figure 5: Sample profile statistics for key exchange method

About Perfect-Forward-Privacy

The Diffie-Hellman Ephemeral (DHE) key exchange method provides Perfect Forward Privacy (PFP). With standard Diffie-Hellman, multiple key exchanges all use the same session key, which can compromise security. By contrast, DHE uses *PFP*, which generates a disposable key per session and thereby ensures that the same session key is never used twice. No key remains to be disclosed, and if the private key of the server is discovered, past communication remains secure.

Supported Diffie-Hellman variations

The BIG-IP system supports all three Diffie-Hellman key exchange methods. They are:

Diffie-Hellman Ephemeral (DHE)
Diffie-Hellman Ephemeral uses temporary public keys. The authenticity of a temporary key can be verified by checking the digital signature included in the key exchange messages. The key exchange messages are signed using either the RSA or DSA algorithms, depending on the cipher being used. For example, DHE-RSA uses RSA to sign the key exchange messages. DHE includes Perfect Forward Secrecy (PFS), which means that a compromise of the system's long-term signing key does not affect the privacy of past sessions. Like FIPS, DHE prevents private key disclosure.

Diffie-Hellman (DH)
Diffie-Hellman embeds the system's public parameter in the certificate, and the CA then signs the certificate. That is, the certificate contains the Diffie-Hellman public-key parameters, and those parameters never change.

Anonymous Diffie-Hellman (ADH)
Anonymous Diffie-Hellman uses DH, but without authentication. The keys used in the exchange are not authenticated, resulting in keys being susceptible to security attacks.

Client and server certificate authentication

There are several settings that you can configure on an SSL profile to manage client-side SSL authentication.

Requirement for a client certificate

You can cause Local Traffic Manager™ to handle authentication of clients or servers in certain ways. For client-side processing, the possible behaviors are:

Ignore
Ignore a certificate (or lack of one) and therefore never authenticate the client. The **Ignore** option is the default, and when used, causes any per-session authentication setting to be ignored.

Require
Require a client to present a valid and trusted certificate before granting access.

Request
Request and verify a client certificate. In this case, the SSL profile always grants access regardless of the status or absence of the certificate.

Warning: *If you are using the LDAP-based client authorization feature, use of the* ***Request*** *or* ***Ignore*** *values can sometimes cause a connection to terminate.*

Tip: *The* ***Request*** *value works well with the header insertion feature. Configuring the SSL profile to insert client certificate information into an HTTP client request, and to authenticate clients based on the* ***Request*** *option, enables the BIG-IP® system or a server to then perform actions such as redirecting the request to another server, sending different content back to the client, or performing client certificate or session ID persistence.*

For server-side processing, the possible behaviors are:

Require
Require a server to present a valid and trusted certificate before granting access.

Ignore
Ignore a certificate (or lack of one) and therefore never authenticate the server. The **Ignore** value is the default setting, and when used, causes any per-session authentication setting to be ignored.

Frequency of authentication

You can configure an SSL profile to require authentication either once per SSL session (once), or once upon each subsequent re-use of an SSL session (always). The default setting for this option is once.

Whether you set this value to once or always depends on your application. A well-designed web application should need to verify a certificate only once per session. F5 recommends for performance reasons that you use the default setting (once) whenever possible.

You can modify the SSL profile to require authentication not only once per session, but also upon each subsequent re-use of an SSL session.

Certificate chain traversal depth

You can use the **Certificate Chain Traversal Depth** setting of an SSL profile to configure the maximum number of CA certificates (intermediate CA certificates and/or root CA certificates) that can be traversed in the certificate chain. The default value is `9`. If a longer chain is provided, and the client has not been authenticated within this number of traversals, client or server certificate verification fails.

Any certificates installed as part of a CA certificate bundle are trusted. All certificates sent by the peer are not trusted, and the BIG-IP system rejects v1/v2 intermediate certificates presented by the peer. The process of certificate chain building starts from a leaf certificate and ends when the issuer CA certificate is trusted.

Trusted certificate authorities

For client-side and server-side SSL processing, you can use the **Trusted Certificate Authorities** setting on an SSL profile to configure an SSL profile to verify certificates presented by a client or a server. You

can specify a client trusted CAs file name, which the BIG-IP® system then uses to verify client or server certificates. If you do not configure a trusted CAs file, the system uses a default file.

The trusted CAs file that you specify for certificate verification contains one or more certificates, in Privacy Enhanced Mail (PEM) format. Built manually, this file contains a list of the client or server certificates that the SSL profile will trust. If you do not specify a trusted CAs file, or the specified trusted CAs file is not accessible to the BIG-IP system, the system uses the default file name.

Advertised certificate authorities

For client-side profiles only, if you intend to configure the SSL profile to require or request client certificates for authentication, you will want the profile to send to clients a list of CAs that the BIG-IP® system is likely to trust.

This list, known as the *Client Certificate CA file*, is different from the client Trusted CAs file. This is because, in some cases, you might have a client that does not possess a valid client certificate, in which case you might not want to reveal the actual list of CAs that the BIG-IP system trusts. The client certificate CA file solves this problem by allowing the BIG-IP system to advertise a list of CAs that is different from the actual client trusted CAs file configured as part of certificate verification.

Although the contents of the Client Certificate CA file can differ from that of the Client Trusted CAs file, it is best, for compatibility reasons, to set the **Advertised Certificate Authorities** setting to match the **Trusted Certificate Authorities** setting. This is because modern browsers might not permit SSL session negotiation to proceed if the peer that requests the client certificate does not provide a list of trusted CAs.

Note: *The maximum size of native SSL handshake messages that Local Traffic Manager™ allows is 14304 bytes. Consequently, if the SSL handshake is negotiating a native cipher and the total length of all messages in the handshake exceeds this byte threshold, the handshake can fail. Although typical use does not cause message length to exceed this threshold, we recommend that when configuring a Client SSL profile to request or require client certificates, you avoid specifying large numbers of certificates through the* ***Advertised Certificate Authorities*** *setting. This minimizes the number of certificates that must be exchanged during a Client SSL handshake.*

Name-based authentication

For server-side profiles only, Local Traffic Manager™ supports name-based authentication, which guards against man-in-the-middle attacks. When you configure the Authenticate Name setting for a server-side profile, Local Traffic Manager checks the name against the Common Name (CN) listed in the certificate that the target server presents to the BIG-IP® system. If the name attribute that you specify does not match the CN in the server certificate, Local Traffic Manager closes the connection. An example of a CN is `www.f5.com`.

Certificate revocation

The **Certificate Revocation List (CRL)** setting of an SSL profile allows Local Traffic Manager™ to use CRLs to check revocation status of a certificate prior to authenticating a client or server.

Important: *CRL files can become outdated, and might need to be updated as often as every day, or as seldom as every 30 days. If your CRL file is out-of-date, Local Traffic Manager rejects all certificates, both valid and invalid. For this reason, it is important to keep your CRL files up-to-date at all times. You can do this by accessing the CRL in the* `/config/ssl/ssl.crl` *directory and then using the* `openssl crl` *command. For more information, see* `http://www.openssl.org/docs/`.

As an alternative to using CRLs, you can use the Online Certificate Status Protocol (OCSP) feature, which ensures up-to-date information on certificate revocation status.

Additional SSL Profile Configuration Options

Options

OpenSSL supports a set of SSL options and defect workarounds. You can enable these workarounds and options as settings of an individual client-side or server-side SSL profile. The default value for the **Options** setting is **Options List**. Retaining the default value enables one option, which is **Don't insert empty fragments**. You can then enable other options that appear in the **Available Options** list.

Important: *For security reasons, when you enable the Proxy SSL setting, the BIG-IP® system automatically disables the* ***Don't insert empty fragments option****. Disabling this option when Proxy SSL is enabled guards against a particular type of cryptographic attack.*

Note that when configuring protocol versions, you must ensure that the protocol versions configured for the BIG-IP system match those of the system's peer. That is, protocol versions specified in the client-side SSL profile must match those of the client, and protocol versions specified in the server-side SSL profile must match those of the server. Thus, for both client-side and server-side SSL connections, you can specify the protocol versions that you do not want the BIG-IP system to allow.

Note: *F5 Networks recommends that, at a minimum, you specify protocol version SSLv2 as invalid.*

Workarounds and other SSL options

This table lists and describes the possible workarounds and options that you can configure for an SSL profile.

SSL Attribute	Description
Cipher server preference	When the BIG-IP® system chooses a cipher, this option uses the server's preferences instead of the client preferences. When this option is not set, the SSL server always follows the client's preferences. When this option is set, the SSLv3/TLSv1 server chooses by using its own preferences. Due to the different protocol, for SSLv2 the server sends its list of preferences to the client, and the client always chooses the cipher.
Don't insert empty fragments	This option disables a countermeasure against a SSL 3.0/TLS 1.0 protocol vulnerability affecting CBC ciphers. These ciphers cannot be handled by certain broken SSL implementations. This option has no effect for connections using other ciphers. This is the default value for the Options list. ***Note:*** *For security reasons, this option is not available when you enable the Proxy SSL setting.*
Ephemeral RSA	This option uses ephemeral (temporary) RSA keys when doing RSA operations. According to the specifications, this is only done when an RSA key can only be used for signature operations (namely under export ciphers with restricted RSA key length). By setting this option, Local Traffic Manager™ always uses ephemeral RSA keys. This option breaks compatibility with the SSL/TLS specifications and can lead to interoperability problems with clients, and we therefore do not recommend it. You

SSL Attribute	Description
	should use ciphers with EDH (ephemeral Diffie-Hellman) key exchange instead. This option is ignored for server-side SSL.
Microsoft session ID bug	This option handles a Microsoft® session ID problem.
Netscape CA DN bug workaround	This option handles a defect regarding system instability. If the system accepts a Netscape® browser connection, demands a client cert, has a non-self-signed CA that does not have its CA in Netscape, and the browser has a certificate, then the system crashes or hangs.
Netscape challenge bug	This option handles the Netscape challenge problem.
Netscape demo cipher change bug workaround	This option deliberately manipulates the SSL server session resumption behavior to mimic that of certain Netscape servers (see the Netscape reuse cipher change bug workaround description). We do not recommend this option for normal use and it is ignored for server-side SSL processing.
Netscape reuse cipher change bug workaround	This option handles a defect within Netscape-Enterprise/2.01, only appearing when connecting through SSLv2/v3 then reconnecting through SSLv3. In this case, the cipher list changes. First, a connection is established with the RC4-MD5 cipher list. If it is then resumed, the connection switches to using the DES-CBC3-SHA cipher list. However, according to RFC 2246, (section 7.4.1.3, cipher_suite) the cipher list should remain RC4-MD5. As a workaround, you can attempt to connect with a cipher list of DES-CBC-SHA:RC4-MD5 and so on. For some reason, each new connection uses the RC4-MD5 cipher list, but any re-connect ion attempts to use the DES-CBC-SHA cipher list. Thus Netscape, when reconnecting, always uses the first cipher in the cipher list.
No SSL	Do not use the SSL protocol.
No SSLv2	Do not use the SSLv2 protocol.
No SSLv3	Do not use the SSLv3 protocol.
No session resumption on renegotiation	When Local Traffic Manager performs renegotiation as an SSL server, this option always starts a new session (that is, session resumption requests are only accepted in the initial handshake). The system ignores this option for server-side SSL processing.
No TLS	Do not use the TLS protocol.
No TLSv1	Do not use the TLSv1 protocol.
Microsoft big SSLV3 buffer	This option enables a workaround for communicating with older Microsoft® applications that use non-standard SSL record sizes.
Microsoft IE SSLV2 RSA padding	This option enables a workaround for communicating with older Microsoft® applications that use non-standard RSA key padding. This option is ignored for server-side SSL.
Passive close	Specifies that the SSL filter helps prevent packets from getting into the TCP half-closed state by waiting for a connection shutdown from the server. This is a workaround for HTTP/1.0 and HTTP/0.9 clients that send an HTTP request followed by a FIN, which immediately closes the connection for server-SSL-only proxies. Instead of closing immediately, the proxy waits for the server to close.
PKCS1 check 1	This debugging option deliberately manipulates the PKCS1 padding used by SSL clients in an attempt to detect vulnerability to particular SSL server vulnerabilities. We

SSL Attribute	Description
	do not recommend this option for normal use. The system ignores this option for client-side SSL processing.
PKCS1 check 2	This debugging option deliberately manipulates the PKCS1 padding used by SSL clients in an attempt to detect vulnerability to particular SSL server vulnerabilities. We do not recommend this option for normal use. The system ignores this option for client-side SSL processing.
Single DH use	This option creates a new key when using temporary/ephemeral DH parameters. You must use this option if you want to prevent small subgroup attacks, when the DH parameters were not generated using strong primes (for example, when using DSA-parameters). If strong primes were used, it is not strictly necessary to generate a new DH key during each handshake, but we do recommend this. You should enable the Single DH use option whenever temporary/ephemeral DH parameters are used.
SSLEAY 080 client DH bug workaround	This option enables a workaround for communicating with older SSLeay-based applications that specify an incorrect Diffie-Hellman public value length. This option is ignored for server-side SSL.
SSL Ref2 reuse cert type bug	This option handles the SSL re-use certificate type problem.
TLS D5 bug workaround	This option is a workaround for communicating with older TLSv1-enabled applications that specify an incorrect encrypted RSA key length. This option is ignored for server-side SSL.
TLS block padding bug workaround	This option enables a workaround for communicating with older TLSv1-enabled applications that use incorrect block padding.
TLS rollback bug workaround	This option disables version rollback attack detection. During the client key exchange, the client must send the same information about acceptable SSL/TLS protocol levels as it sends during the first hello. Some clients violate this rule by adapting to the server's answer. For example, the client sends an SSLv2 hello and accepts up to SSLv3.1 (TLSv1), but the server only understands up to SSLv3. In this case, the client must still use the same SSLv3.1 (TLSv1) announcement. Some clients step down to SSLv3 with respect to the server's answer and violate the version rollback protection. This option is ignored for server-side SSL.

ModSSL methods

You can enable or disable ModSSL method emulation. You enable ModSSL method emulation when the OpenSSL methods are inadequate. When you enable this setting, you can then write an iRule, using the `HTTP::header insert_modssl_fields` command, which inserts some of the ModSSL options as headers into HTTP requests.

ModSSL options for use with iRules

This table lists the options that you can insert into an HTTP request.

Header Type	Header Name and Format	Description
Certificate status	SSLClientCertStatus: [status]	The status of the client certificate. The value of [status] can be NoClientCert, OK, or Error. If status is NoClientCert,

Header Type	Header Name and Format	Description
		only this header is inserted into the request. If status is Error, the error is followed by a numeric error code.
Certificate version	SSLClientCertVersion: [version]	The version of the certificate.
Certificate serial number	SSLClientCertSerialNumber: [serial]	The serial number of the certificate.
Signature algorithm of the certificate	SSLClientCertSignatureAlgorithm: [alg]	The signature algorithm of the certificate.
Issuer of the certificate	SSLClientCertIssuer: [issuer]	The issuer of the certificate.
Certificate validity dates	SSLClientCertNotValidBefore: [before] SSLClientCertNotValidAfter: [after]	The validity dates for the certificate. The certificate is not valid before or after the dates represented by `[before]` and `[after]`, respectively.
Certificate subject	SSLClientCertSubject: [subject]	The subject of the certificate.
Public key of the subject	SSLClientCertSubjectPublicKey: [key]	The type of public key type. The allowed types are RSA ([size] bit), DSA, or Unknown public key.
The certificate itself	SSLClientCert: [cert]	The actual client certificate.
MD5 hash of the certificate	SSLClientCertHash: [hash]	The MD5 hash of the client certificate.

SSL session cache size and timeout

You can configure timeout and size values for the SSL session cache. Because each profile maintains a separate SSL session cache, you can configure the values on a per-profile basis.

SSL session cache size

You can specify the maximum size of the SSL session cache. The default value for the size of the SSL session cache is 262144 entries. A value of `0` disallows session caching.

SSL session cache timeout

You can specify the number of usable lifetime seconds of negotiated SSL session IDs. The default timeout value for the SSL session cache is `3600` seconds. If you specify a timeout value, valid values are integers greater than or equal to 1.

Clients attempting to resume an SSL session with an expired session ID are forced to negotiate a new session.

Warning: *If the timeout value for the client-side SSL session cache is set to zero, the SSL session IDs negotiated with that profile's clients remain in the session cache until the cache is filled and the purging of entries begins. Setting a value of zero can introduce a significant security risk if valuable resources are*

available to a client that is reusing those session IDs. It is therefore common practice to set the SSL session cache timeout to a length of time no greater than 24 hours, and for significantly shorter periods.

Alert timeout

You can specify the duration in seconds that the BIG-IP® system waits while trying to close an SSL connection, before the connection is reset. The default timeout value for this setting is 10 seconds.

Handshake timeout

You can specify the amount of time in seconds that the BIG-IP® system spends attempting to perform an SSL handshake. The default timeout value for this setting is 10 seconds.

Renegotiation of SSL sessions

Long-lived connections are susceptible to man-in-the-middle attacks. To prevent such attacks, you can force the BIG-IP® system to renegotiate SSL sessions, based on either time period or application size. You can also force the BIG-IP system to terminate an SSL session after receiving a specified number of records.

More specifically, you can control, on a per-connection basis, the way that the system responds to mid-stream SSL reconnection requests. When you enable renegotiation, the system processes mid-stream SSL renegotiation requests. When you disable renegogiation, the system either terminates the connection or ignores the request, depending on the system configuration. By default, renegotiation is enabled.

Sessions based on a time period

You can specify the number of seconds from the initial connect time that the system renegotiates an SSL session. The options are a number you specify, indefinite, and default. The default is indefinite, meaning that you do not want the system to renegotiate SSL sessions. Each time the session renegotiation is successful, essentially a new connection is started. Therefore, the system attempts to renegotiate the session again in the specified amount of time following the successful session renegotiation. For example, setting the **renegotiate period** to **3600** seconds triggers session renegotiation at least once an hour.

Sessions based on application data size

You can force Local Traffic Manager™ to renegotiate an SSL session after the specified number of megabytes of application data have been transmitted over the secure channel. The default value for this setting is `Indefinite`.

Maximum record delay

You can force Local Traffic Manager™ (LTM®) to terminate an SSL session after receiving the specified maximum number of delayed SSL records. If LTM receives more than the specified number of delayed SSL records, the system closes the connection. The default value for this setting is **Indefinite**.

Secure renegotiation

The Secure Renegotiation setting specifies the method of secure renegotiation for SSL connections. The default value for the Client SSL profile is Require; the default value for the Server SSL profile is Require Strict. If your configuration does not require secure SSL renegotiation, set this value to Request. The possible values for this setting are:

Request
Specifies that the system requests secure renegotiation of SSL connections.

Require
Specifies that the system requires secure renegotiation of SSL connections. In this mode, the system permits initial SSL handshakes from clients, but terminates renegotiations from unpatched clients.

Require Strict
Specifies that the system requires strict secure renegotiation of SSL connections. In this mode, the system refuses new SSL connections to unsecure servers and terminates existing SSL connections to unsecure servers.

Maximum reneogtiations

You can specify the maximum number of SSL records per minute that the system can receive before renegotiating an SSL session. After receiving this number of SSL records, the system closes the connection. This setting applies to client profiles only. By default, the system receives five records per minute before reneogiating the session.

Maximum aggregate renegotiations

You can specify a maximum number of aggregate SSL renegotiations, to ensure that the configured per-flow maximum renegotiation rate limit is enforced. Specifying a maximum number of aggregate renegotiations prevents the opening of new connections as a way to bypass the per-flow limit. Allowed values are from `0` through `0xFFFFFFFF`.

Server name

The **Server Name** setting in an SSL profile specifies the name of the specific domain from which the client requests a certificate. This setting supports a feature known as TLS Server Name Indication (TLS SNI), used when a single virtual IP server needs to host multiple domains.

For example, suppose that the BIG-IP® system needs to host the two domains `domain1.com` and `domain2.com`, on the same HTTP virtual server. Each domain has its own server certificate to use, such as domain1.crt and domain2.crt, and each has different security requirements.

To ensure that the BIG-IP system presents the correct certificate to the browser, you enable SNI, which sends the name of a domain as part of the TLS negotiation. This, in turn, enables the BIG-IP system to select this domain rather than waiting to read the domain name in the request header.

To enable SNI, you configure the Server Name and other TLS-related settings on an SSL profile, and then assign the profile to a virtual server.

Note that the wildcard character (*) is supported within any domain name that you specify.

Default SSL Profile for SNI

When you enable the **Default SSL Profile for SNI** setting on an SSL profile, you are specifying that this is the default SSL profile to use when the client provides either no Server Name Indication (SNI) extension, or provides a non-matching SNI extension.

When assigning multiple SSL profiles to a single virtual server, you can enable this setting on one Client SSL profile only and one Server SSL profile only.

Require Peer SNI Support

If you enable the **Require Peer SNI Support** setting on an SSL profile, the domain name of the peer must match the domain name that you specify in the **Default SSL Profile for SNI** field.

Unclean SSL shutdowns

In an *unclean shutdown*, underlying TCP connections are closed without exchanging the required SSL shutdown alerts. However, you can disable unclean shutdowns and thus force the SSL profile to perform a clean shutdown of all SSL connections by configuring this setting.

This feature is especially useful with respect to the Internet Explorer browser. Different versions of the browser, and even different builds within the same version of the browser, handle shutdown alerts differently. Some versions or builds require shutdown alerts from the server, while others do not, and the SSL profile cannot always detect this requirement or lack of it. In the case where the browser expects a shutdown alert but the SSL profile has not exchanged one (the default setting), the browser displays an error message.

By default, this setting is enabled, which means that Local Traffic Manager™ performs unclean shutdowns of all SSL connections.

Strict Resume

You can configure Local Traffic Manager™ to discontinue an SSL session after an unclean shutdown. By default, this setting is disabled, which causes Local Traffic Manager (LTM®) to resume SSL sessions after an unclean shutdown. If you enable this setting, LTM does not resume SSL sessions after an unclean shutdown.

About session tickets

To enhance system performance, you can enable the use of session tickets, a TLS extension defined in RFC 5077. The use of session tickets is an alternative to the standard session caching mechanism that systems such as the BIG-IP system typically use to resume sessions.

When you enable this feature, the BIG-IP system, acting as a server to terminate SSL connections, sends a special message to the client as part of the SSL handshake. This message includes a *session ticket*, which contains complete session state information. Sending the session state information to the client removes the need for the BIG-IP system to maintain a server-side cache for storing session information. With session tickets, the entire session state is remembered by the client.

The session state information in the ticket includes the master secret negotiated between the client and the BIG-IP system, as well as the cipher suite used.

Generic alerts

For security reasons, when sending an SSL alert message, the BIG-IP® system sends a generic `handshake failure` message with an alert code of 40, with no detailed information. This is the default behavior.

If you want SSL alert messages to include the specific reason for the failure, you can disable the **Generic Alerts** setting. In this case, when an SSL failure occurs, the system sends an alert message with a specific numeric code. For example, an alert message due to a certificate revocation would show a specific code of 48 instead of the generic code of 40.

Acceptance of non-SSL connections

You can configure Local Traffic Manager™ to accept connections that are not SSL connections. In this case, connections pass through the BIG-IP® system in clear-text format. By default, this setting is disabled.

SSL sign hash

For a Client SSL profile, this setting specifies the hash algorithm that the Client SSL profile uses to sign Server Key Exchange messages and advertises in a Certificate Request message when requesting a client certificate.

For a Server SSL profile, this setting specifies the certificate signature hash algorithm or algorithms that the Server SSL profile advertises in the ClientHello signature_algorithms extension and uses in the Certificate Verify handshake message when using a client certificate.

About SSL handshake limits

A reboot or reset action can sometimes produce an excessive number of SSL handshakes, which can impact normal BIG-IP® system operation. To prevent this from happening, you can use the **Max Active Handshakes** setting on a Client SSL or Server SSL profile to limit the number of concurrent handshakes.

When the number of active SSL handshakes pertaining to an SSL profile reaches the specified limit, the system terminates the most recent SSL handshake, and the BIG-IP system displays a message that the specified handshake limit has been reached. The system also sends an alert message to other members of the device group.

The default setting is **Indefinite**, which means that there is no limit on the number of active SSL handshakes that the system allows.

About dynamic record sizing

Dynamic record sizing is a TLS performance enhancement, designed to improve application response by preventing bottlenecks caused by the buffering of TLS record fragments. Without dynamic record sizing, a record that spans multiple TCP packets can cause buffering, forcing the TCP receiver to wait for all of

the TCP packets to arrive before constructing the original-sized record (typically 16 KB). Other causes of buffering can include packet loss, packet reordering, or throttling. The result is that the browser is left to deal with significant bottlenecks that affect performance.

With dynamic record sizing, the system dynamically adjusts the size of TLS records based on the state of the connection. For example, if a connection is idle for awhile, it might make sense for the system to ensure a single TLS record per packet, where the size of the TLS record is the TCP maximum segment size (MSS). For connections in another state, such as large application streams, it might still make sense for a TLS record to span multiple packets.

To specify dynamic record sizing, you log in to the BIG-IP ®Configuration utility screen, and locate the screen for creating a Client SSL profile. Then enable the **Allow Dynamic Record Sizing** check box.

About the maximum record size

The **Maximum Record Size** setting on a Client SSL profile defines the maximum size possible for a TLS record. If you enable dynamic record sizing (for performance enhancement), then the maximum record size you set is the largest size that the system can use for a TLS record when that size is needed. If dynamic record sizing is disabled (the default value), then all record size is static, and the size can be up to the maximum record size you specify.

To set the maximum record size, you log in to the BIG-IP ®Configuration utility screen and locate the screen for creating a Client SSL profile. Then type a value for the **Maximum Record Size** setting. The default value in kilobytes (KB) is **16384**.

SSL Persistence

SSL persistence

SSL persistence is a type of persistence that tracks SSL sessions using the SSL session ID, and it is a property of each individual pool. Using SSL persistence can be particularly important if your clients typically have translated IP addresses or dynamic IP addresses, such as those that Internet service providers typically assign. Even when the client's IP address changes, BIG-IP system® still recognizes the session as being persistent based on the session ID.

You might want to use SSL persistence and source address affinity persistence together. In situations where an SSL session ID times out, or where a returning client does not provide a session ID, you might want the BIG-IP system to direct the client to the original node based on the client's IP address. As long as the client's simple persistence record has not timed out, the BIG-IP system can successfully return the client to the appropriate node.

Criteria for session persistence

For most persistence types, you can specify the criteria that the BIG-IP® system uses to send all requests from a given client to the same pool member. These criteria are based on the virtual server or servers that are hosting the client connection. To specify these criteria, you configure the **Match Across Services**, **Match Across Virtual Servers**, and **Match Across Pools**settings contained within persistence profiles. Before configuring a persistence profile, it is helpful to understand these settings.

Note: *For the Cookie persistence type, these global settings are only available the Cookie Hash method specifically.*

Creating an SSL persistence profile

You create an SSL persistence profile when you want to customize the way that the BIG-IP®system persists SSL traffic.

Important: *The BIG-IP system includes a default SSL persistence profile named* `ssl`*. If you do not need to customize the way that the system persists SSL traffic, you can skip this task. Instead, simply use the* ***Default Persistence Profile*** *setting on the relevant virtual server to specify the default* `ssl` *profile.*

1. On the Main tab, click **Local Traffic** > **Profiles** > **Persistence**.
   The Persistence profile list screen opens.
2. Click **Create**.
   The New Persistence Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Persistence Type** list, select **SSL**.
5. For the **Parent Profile** setting, confirm that **ssl** appears.
6. Select the **Custom** check box.
7. Configure settings as needed.
8. Click **Finished**.

The custom SSL persistence profile now appears in the persistence profiles list.

After creating a persistence profile, you must assign the profile to the relevant virtual server.

Managing Client-Side HTTP Traffic Using a CA-Signed RSA Certificate

Overview: Managing client-side HTTP traffic using a CA-signed RSA certificate

When you want to manage HTTP traffic over SSL, you can configure the BIG-IP® system to perform the SSL handshake that target web servers normally perform.

A common way to configure the BIG-IP system is to enable client-side SSL, which makes it possible for the system to decrypt client requests before sending them on to a server, and encrypt server responses before sending them back to the client. In this case, you need to install only one SSL key/certificate pair on the BIG-IP system.

This implementation uses a certificate signed by an RSA certificate authority (CA) to authenticate HTTP traffic.

Task summary

To implement client-side authentication using HTTP and SSL with a certificate signed by a certificate authority, you perform a few basic configuration tasks.

Task list

Requesting an RSA certificate from a certificate authority

You can generate a request for an RSA digital certificate and then copy or submit it to a trusted certificate authority for signature.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**. The Traffic Certificate Management screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Certificate Authority**.
5. In the **Common Name** field, type a name.

   This is typically the name of a web site, such as `www.siterequest.com`.
6. In the **Division** field, type your department name.
7. In the **Organization** field, type your company name.
8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.

    This name is embedded in the certificate for X509 extension purposes.

    By assigning this name, you can protect multiple host names with a single SSL certificate.

14. In the **Challenge Password** field, type a password.
15. In the **Confirm Password** field, re-type the password you typed in the **Challenge Password** field.
16. From the **Key Type** list, select **RSA**.
17. From the **Size** list, select a key size, in bits.
18. Click **Finished**.
    The Certificate Signing Request screen displays.
19. Do one of the following to download the request into a file on your system.
    - In the **Request Text** field, copy the certificate.
    - For **Request File**, click the button.
20. Follow the instructions on the relevant certificate authority web site for either pasting the copied request or attaching the generated request file.
21. Click **Finished**.
    The Certificate Signing Request screen displays.

The generated RSA certificate request is submitted to a trusted certificate authority for signature.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP® system to manage HTTP traffic.

Note: *Other HTTP profile types (HTTP Compression and Web Acceleration) enable you to configure compression and cache settings, as required. Use of these profile types is optional.*

---

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **HTTP**.
   The HTTP profile list screen opens.
2. Click **Create**.
   The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

After you have built the cipher string that you want the BIG-IP to use to negotiate client-side SSL connections, you create a custom Client SSL profile. You create the profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and encrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads these decryption/encryption functions from the destination server. When you perform this task, you can specify multiple certificate key chains, one for each key type (RSA, DSA, and ECDSA). This allows the BIG-IP system to negotiate secure client connections using different cipher suites based on the client's preference.

Note: *At a minimum, you must specify a certificate key chain that includes an RSA key pair. Specifying certificate key chains for DSA and ECDSA key pairs is optional, although highly recommended.*

---

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client SSL profile list screen opens.
2. Click **Create**.

The New Client SSL Profile screen opens.

3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.
   The settings become available for change.
6. For the **Certificate Key Chain** setting, click **Add**.
   a) From the **Certificate** list, select a certificate name.

      This is the name of a certificate that you installed on the BIG-IP® system. If you have not generated a certificate request nor installed a certificate on the BIG-IP system, and the BIG-IP system is not part of a device service clustering (DSC) configuration, you can specify the name of the existing certificate named `default`.

      Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default certificate name, and ensure that this same certificate name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*

   b) From the **Key** list, select the name of the key associated with the certificate specified in the previous step.

      This is the name of a key that you installed on the BIG-IP® system. If you have not installed a key on the BIG-IP system, and the BIG-IP system is not part of a device service clustering (DSC) configuration, you can specify the name of the existing key named `default`.

      Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default key name, and ensure that this same key name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*

   c) From the **Chain** list, select the chain that you want to include in the certificate key chain.

      A certificate chain can contain either a series of public key certificates in Privacy Enhanced Mail (PEM) format or a series of one or more PEM files. A certificate chain can contain certificates for Intermediate certificate Authorities (CAs).

      Note: *The default self-signed certificate and the default CA bundle certificate are not appropriate for use as a certificate chain.*

   d) For the **Passphrase** field, type a string that enables access to SSL certificate/key pairs that are stored on the BIG-IP system with password protection.

      This setting is optional. For added security, the BIG-IP system automatically encrypts the pass phrase itself. This pass phrase encryption process is invisible to BIG-IP® system administrative users.

   e) Click **Add**.
7. Click **Add** and repeat the process for all certificate key chains that you want to specify. At a minimum, you must specify an RSA certificate key chain.

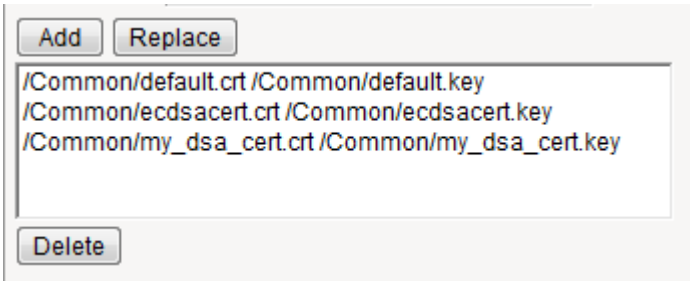
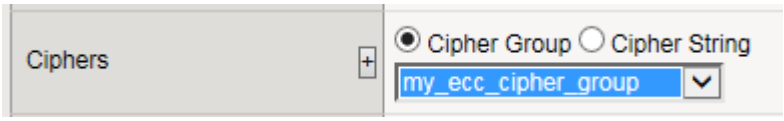


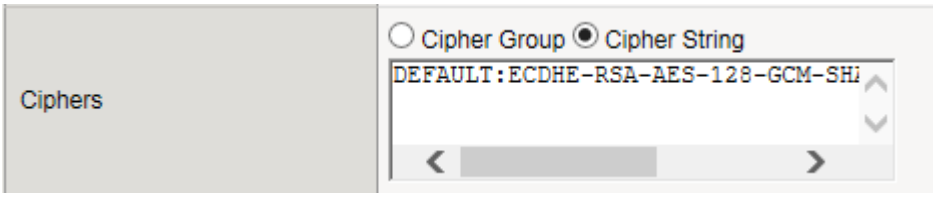
Figure 6: Sample configuration with three key types specified

The result is that all specified key chains appear in the text box.

8. For the **OCSP Stapling** setting, select the check box.
   This setting is optional. To enable OCSP stapling, you must first create an OCSP Stapling profile.
9. For the **Notify Certificate Status to Virtual Server** setting, select the check box.
   This setting is optional.
10. For the **Ciphers** setting, specify a cipher group or cipher string by choosing one of these options.

Note: *If you specified an ECDSA certificate key chain in the* ***Certificate Key Chain*** *setting, you must include the cipher string* `ECDHE_ECDSA` *in the cipher group or cipher string that you specify in the* ***Ciphers*** *setting. (At a minimum, you should specify a cipher group or string such as* `DEFAULT:ECDHE_ECDSA`*.) This is necessary to ensure successful cipher negotiation when the BIG-IP system is offered an ECDSA-based certificate only.*

Option	Description
Cipher Group	Select an existing cipher group from the list when you want to use a system-defined or custom cipher group to define the ciphers that the BIG-IP system uses for negotiating SSL connections. Here's an example of the `Ciphers` setting where we've selected a custom cipher group that we created earlier. 
Cipher String	Type a cipher string in the box if you want to manually specify a cipher string instead of selecting a cipher group. For security and performance reasons, consider following these recommendations: • Always append ciphers to the `DEFAULT` cipher string. • Type a cipher string that includes the ECC key type, because its shorter length speeds up encryption and decryption while still offering virtually the same level of security. • Disable ADH ciphers but also include the keyword `HIGH`. To do this, just include both `!ADH` and `:HIGH` in your cipher string. • For AES, DES, and RC4 encryption types, make sure you specify the DHE key exchange method. DHE uses *Forward Privacy*, which creates a key that it throws away after each session so that the same session key never gets used twice. When you use DHE, make sure that the SSL private key isn't being shared with a monitoring system or a security device like an intrusion detection or prevention system. Also, diagnostic tools like `ssldump` won't work when you're using Forward Secrecy.

Option	Description
	• Disable EXPORT ciphers by including `!EXPORT` in the cipher string. • If you can live with removing support for the SSLv3 protocol version, do it. This protocol version is not secure. Simply include `:!SSLv3` in any cipher string you type. Here's an example of the `Ciphers` setting where we have opted to manually type the cipher string `DEFAULT:ECDHE-RSA-AES-128-GCM-SHA256:!ADH:!EXPORT:HIGH`: 

11. Configure all other profile settings as needed.
12. Click **Finished**.

After performing this task, you can see the custom Client SSL profile in the list of Client SSL profiles on the system.

You must also assign the profile to a virtual server.

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor and move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) Type an IP address in the **Address** field.
   b) Type `80` in the **Service Port** field, or select **HTTP** from the list.
   c) (Optional) Type a priority number in the **Priority** field.
   d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating a virtual server for client-side HTTP traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTP traffic over SSL.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *The IP address you type must be available and not in the loopback network.*
5. In the **Service Port** field, type `443`, or select **HTTPS** from the list.
6. From the **HTTP Profile** list, select the HTTP profile that you previously created.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created and move the name to the **Selected** list.
8. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
9. Click **Finished**.

After performing this task, the virtual server appears in the Virtual Server List screen.

Implementation results

After you complete the tasks in this implementation, the BIG-IP® system can authenticate and decrypt HTTP traffic coming from a client system, using an RSA digital certificate. The BIG-IP system can also re-encrypt server responses before sending them back to the client.

Managing Client-Side HTTP Traffic Using a CA-Signed Elliptic Curve DSA Certificate

Overview: Managing client-side HTTP traffic using a CA-signed, ECC-based certificate

When you configure the BIG-IP® system to decrypt client-side HTTP requests and encrypt the server responses, you can optionally configure the BIG-IP system to use the Elliptic Curve Digital Signature Algorithm (ECDSA) as part of the BIG-IP system's certificate key chain. The result is that the BIG-IP system performs the SSL handshake usually performed by target web servers, using an ECDSA key type in the certificate key chain.

This particular implementation uses a certificate signed by a certificate authority (CA).

Task summary

To implement client-side authentication using HTTP and SSL with a certificate signed by a certificate authority, you perform a few basic configuration tasks.

Task list

Requesting a signed certificate that includes an ECDSA key

You can generate a certificate that includes an Elliptic Curve Digital Signature Algorithm (ECDSA) key type, and then copy it or submit it to a trusted certificate authority for signature.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**. The Traffic Certificate Management screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Certificate Authority**.
5. In the **Common Name** field, type a name.

   This is typically the name of a web site, such as `www.siterequest.com`.
6. In the **Division** field, type your department name.
7. In the **Organization** field, type your company name.
8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.

    This name is embedded in the certificate for X509 extension purposes.

    By assigning this name, you can protect multiple host names with a single SSL certificate.
14. In the **Challenge Password** field, type a password.
15. In the **Confirm Password** field, re-type the password you typed in the **Challenge Password** field.

16. From the **Key Type** list, select **ECDSA**.
17. From the **Curve Name** list, select **prime256v1** or **secp384r1**.
18. Click **Finished**.
    The Certificate Signing Request screen displays.
19. Do one of the following to download the request into a file on your system.
    - In the **Request Text** field, copy the certificate.
    - For **Request File**, click the button.
20. Follow the instructions on the relevant certificate authority web site for either pasting the copied request or attaching the generated request file.
21. Click **Finished**.
    The Certificate Signing Request screen displays.

The generated certificate is submitted to a trusted certificate authority for signature.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP® system to manage HTTP traffic.

Note: *Other HTTP profile types (HTTP Compression and Web Acceleration) enable you to configure compression and cache settings, as required. Use of these profile types is optional.*

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **HTTP**.
   The HTTP profile list screen opens.
2. Click **Create**.
   The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

You create a custom Client SSL profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and decrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads these authentication and decryption/encryption functions from the destination server. When you perform this task, you specify a certificate key chain that includes Elliptic Curve Digital Signature Algorithm (ECDSA) as the key type.

Note: *In addition to specifying an ECDSA certificate key chain, you must also specify an RSA key chain. Specifying an RSA key chain is a minimum requirement for all Client SSL profiles that you configure.*

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client SSL profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.
   The settings become available for change.

6. For the **Certificate Key Chain** setting, click **Add**.
   a) From the **Certificate** list, select the name of a certificate with a key of type ECDSA.

      This is the name of a certificate that you installed on the BIG-IP® system. If you have not generated a certificate request nor installed a certificate on the BIG-IP system, and the BIG-IP system is not part of a device service clustering (DSC) configuration, you can specify the name of the existing certificate named `default`.

      Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default certificate name, and ensure that this same certificate name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*

   b) From the **Key** list, select the name of an ECDSA key.

      This is the name of a key that you installed on the BIG-IP® system. If you have not installed a key on the BIG-IP system, and the BIG-IP system is not part of a device service clustering (DSC) configuration, you can specify the name of the existing key named `default`.

      Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default key name, and ensure that this same key name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*

   c) From the **Chain** list, select the chain that you want to include in the certificate key chain.

      A certificate chain can contain either a series of public key certificates in Privacy Enhanced Mail (PEM) format or a series of one or more PEM files. A certificate chain can contain certificates for Intermediate certificate Authorities (CAs).

      Note: *The default self-signed certificate and the default CA bundle certificate are not appropriate for use as a certificate chain.*

   d) For the **Passphrase** field, type a string that enables access to SSL certificate/key pairs that are stored on the BIG-IP system with password protection.

      This setting is optional. For added security, the BIG-IP system automatically encrypts the pass phrase itself. This pass phrase encryption process is invisible to BIG-IP® system administrative users.

   e) Click **Add**.
7. Click **Add** and repeat the process for all certificate key chains that you want to specify. At a minimum, you must specify an RSA certificate key chain.

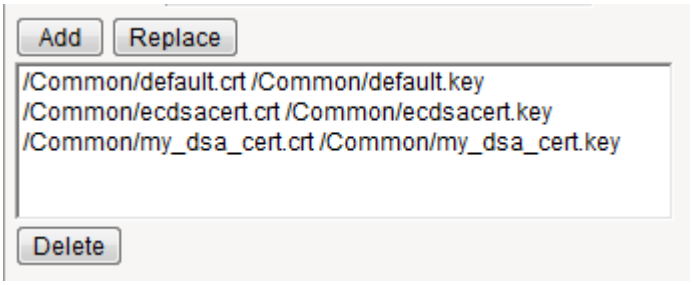


Figure 7: Sample configuration with three key types specified

   The result is that all specified key chains appear in the text box.
8. To specify ECDHE ciphers:
   a) From the **Configuration** list, select **Advanced**.
   b) In the **Ciphers** field, type `ECDHE`.
9. Configure all other profile settings as needed.

10. Click **Finished**.

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor and move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) Type an IP address in the **Address** field.
   b) Type `80` in the **Service Port** field, or select **HTTP** from the list.
   c) (Optional) Type a priority number in the **Priority** field.
   d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating a virtual server for client-side HTTP traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTP traffic over SSL.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *The IP address you type must be available and not in the loopback network.*

   ---
5. In the **Service Port** field, type `443`, or select **HTTPS** from the list.
6. From the **HTTP Profile** list, select the HTTP profile that you previously created.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created and move the name to the **Selected** list.

8. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
9. Click **Finished**.

After performing this task, the virtual server appears in the Virtual Server List screen.

Implementation results

After you complete the tasks in this implementation, the BIG-IP® system encrypts client-side ingress HTTP traffic using an SSL certificate key chain. The BIG-IP system also re-encrypts server responses before sending the responses back to the client.

The certificate in the certificate key chain includes an Elliptic Curve Digital Signature Algorithm (ECDSA) key and certificate.

Managing Client- and Server-Side HTTP Traffic Using a CA-Signed Certificate

Overview: Managing client and server HTTP traffic using a CA-signed certificate

One of the ways to configure the BIG-IP system to manage SSL traffic is to enable both client-side and server-side SSL termination:

- *Client-side SSL termination* makes it possible for the system to decrypt client requests before sending them on to a server, and encrypt server responses before sending them back to the client. This ensures that client-side HTTP traffic is encrypted. In this case, you need to install only one SSL key/certificate pair on the BIG-IP system.
- *Server-side SSL termination* makes it possible for the system to decrypt and then re-encrypt client requests before sending them on to a server. Server-side SSL termination also decrypts server responses and then re-encrypts them before sending them back to the client. This ensures security for both client- and server-side HTTP traffic. In this case, you need to install two SSL key/certificate pairs on the BIG-IP system. The system uses the first certificate/key pair to authenticate the client, and uses the second pair to request authentication from the server.

This implementation uses a CA-signed certificate to manage HTTP traffic.

Task summary

To implement client-side and server-side authentication using HTTP and SSL with a CA-signed certificate, you perform a few basic configuration tasks.

Task list

Requesting a certificate from a certificate authority

You perform this task to generate a certificate signing request (CSR) that can then be submitted to a third-party trusted certificate authority (CA).

Note: *F5 Networks recommends that you consult the CA to determine the specific information required for each step in this task.*

---

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**. The Traffic Certificate Management screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Certificate Authority**.
5. In the **Common Name** field, type a name.

   This is typically the name of a web site, such as `www.siterequest.com`.
6. In the **Division** field, type your department name.
7. In the **Organization** field, type your company name.
8. In the **Locality** field, type your city name.

9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.

    This name is embedded in the certificate for X509 extension purposes.

    By assigning this name, you can protect multiple host names with a single SSL certificate.
14. In the **Challenge Password** field, type a password.
15. In the **Confirm Password** field, re-type the password you typed in the **Challenge Password** field.
16. From the **Key Type** list, select a key type.

    Possible values are: **RSA**, **DSA**, and **ECDSA**.
17. From the **Size** or **Curve Name** list, select either a size, in bits, or a curve name.
18. If the BIG-IP system contains an internal HSM module, specify a location for storing the private key.
19. Click **Finished**.
    The Certificate Signing Request screen displays.
20. Do one of the following to download the request into a file on your system.
    - In the **Request Text** field, copy the certificate.
    - For **Request File**, click the button.
21. Follow the instructions on the relevant certificate authority web site for either pasting the copied request or attaching the generated request file.
22. Click **Finished**.
    The Certificate Signing Request screen displays.

The generated certificate signing request is submitted to a trusted certificate authority for signature.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP®system to manage HTTP traffic.

Note: *Other HTTP profile types (HTTP Compression and Web Acceleration) enable you to configure compression and cache settings, as required. Use of these profile types is optional.*

---

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **HTTP**.
   The HTTP profile list screen opens.
2. Click **Create**.
   The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

After you have built the cipher string that you want the BIG-IP to use to negotiate client-side SSL connections, you create a custom Client SSL profile. You create the profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and encrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads

these decryption/encryption functions from the destination server. When you perform this task, you can specify multiple certificate key chains, one for each key type (RSA, DSA, and ECDSA). This allows the BIG-IP system to negotiate secure client connections using different cipher suites based on the client's preference.

Note: *At a minimum, you must specify a certificate key chain that includes an RSA key pair. Specifying certificate key chains for DSA and ECDSA key pairs is optional, although highly recommended.*

---

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client SSL profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.
   The settings become available for change.
6. For the **Certificate Key Chain** setting, click **Add**.
   a) From the **Certificate** list, select a certificate name.

      This is the name of a certificate that you installed on the BIG-IP® system. If you have not generated a certificate request nor installed a certificate on the BIG-IP system, and the BIG-IP system is not part of a device service clustering (DSC) configuration, you can specify the name of the existing certificate named `default`.

      Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default certificate name, and ensure that this same certificate name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*

      ---

   b) From the **Key** list, select the name of the key associated with the certificate specified in the previous step.

      This is the name of a key that you installed on the BIG-IP® system. If you have not installed a key on the BIG-IP system, and the BIG-IP system is not part of a device service clustering (DSC) configuration, you can specify the name of the existing key named `default`.

      Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default key name, and ensure that this same key name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*

      ---

   c) From the **Chain** list, select the chain that you want to include in the certificate key chain.

      A certificate chain can contain either a series of public key certificates in Privacy Enhanced Mail (PEM) format or a series of one or more PEM files. A certificate chain can contain certificates for Intermediate certificate Authorities (CAs).

      Note: *The default self-signed certificate and the default CA bundle certificate are not appropriate for use as a certificate chain.*

      ---

   d) For the **Passphrase** field, type a string that enables access to SSL certificate/key pairs that are stored on the BIG-IP system with password protection.

      This setting is optional. For added security, the BIG-IP system automatically encrypts the pass phrase itself. This pass phrase encryption process is invisible to BIG-IP® system administrative users.

   e) Click **Add**.

7. Click **Add** and repeat the process for all certificate key chains that you want to specify. At a minimum, you must specify an RSA certificate key chain.

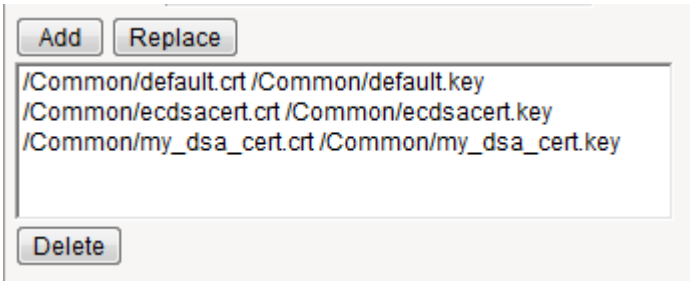
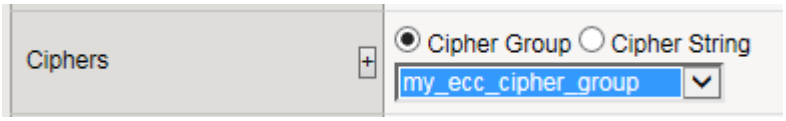


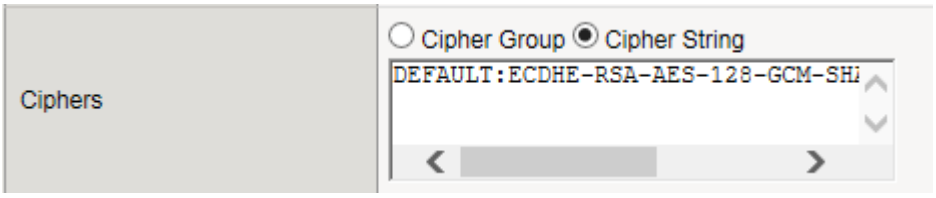
Figure 8: Sample configuration with three key types specified

The result is that all specified key chains appear in the text box.

8. For the **OCSP Stapling** setting, select the check box.
   This setting is optional. To enable OCSP stapling, you must first create an OCSP Stapling profile.
9. For the **Notify Certificate Status to Virtual Server** setting, select the check box.
   This setting is optional.
10. For the **Ciphers** setting, specify a cipher group or cipher string by choosing one of these options.

Note: *If you specified an ECDSA certificate key chain in the* ***Certificate Key Chain*** *setting, you must include the cipher string* `ECDHE_ECDSA` *in the cipher group or cipher string that you specify in the* ***Ciphers*** *setting. (At a minimum, you should specify a cipher group or string such as* `DEFAULT:ECDHE_ECDSA`*.) This is necessary to ensure successful cipher negotiation when the BIG-IP system is offered an ECDSA-based certificate only.*

Option	Description
Cipher Group	Select an existing cipher group from the list when you want to use a system-defined or custom cipher group to define the ciphers that the BIG-IP system uses for negotiating SSL connections. Here's an example of the `Ciphers` setting where we've selected a custom cipher group that we created earlier. 
Cipher String	Type a cipher string in the box if you want to manually specify a cipher string instead of selecting a cipher group. For security and performance reasons, consider following these recommendations: • Always append ciphers to the `DEFAULT` cipher string. • Type a cipher string that includes the ECC key type, because its shorter length speeds up encryption and decryption while still offering virtually the same level of security. • Disable ADH ciphers but also include the keyword `HIGH`. To do this, just include both `!ADH` and `:HIGH` in your cipher string. • For AES, DES, and RC4 encryption types, make sure you specify the DHE key exchange method. DHE uses *Forward Privacy*, which creates a key that it throws away after each session so that the same session key never gets used twice. When you use DHE, make sure that the SSL private key isn't being shared with a monitoring system or a security device like an intrusion

Option	Description
	detection or prevention system. Also, diagnostic tools like `ssldump` won't work when you're using Forward Secrecy. • Disable EXPORT ciphers by including `!EXPORT` in the cipher string. • If you can live with removing support for the SSLv3 protocol version, do it. This protocol version is not secure. Simply include `:!SSLv3` in any cipher string you type. Here's an example of the `Ciphers` setting where we have opted to manually type the cipher string `DEFAULT:ECDHE-RSA-AES-128-GCM-SHA256:!ADH:!EXPORT:HIGH`: 

11. Configure all other profile settings as needed.
12. Click **Finished**.

After performing this task, you can see the custom Client SSL profile in the list of Client SSL profiles on the system.

You must also assign the profile to a virtual server.

Creating a custom Server SSL profile

With a Server SSL profile, the BIG-IP® system can perform decryption and encryption for server-side SSL traffic.

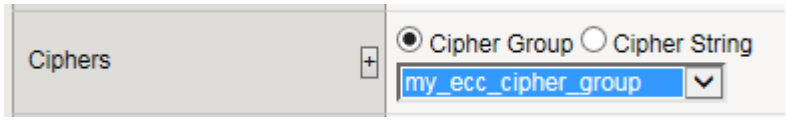
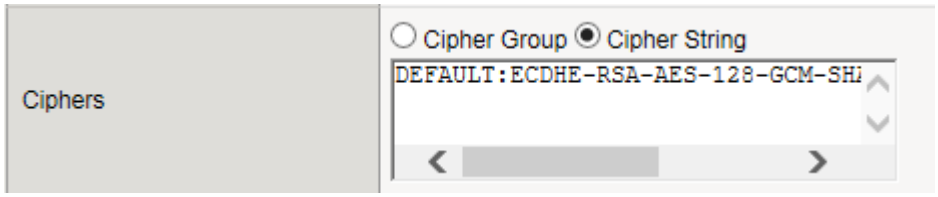
1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The Server SSL profile list screen opens.
2. Click **Create**.
   The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **serverssl** in the **Parent Profile** list.
5. From the **Configuration** list, select **Advanced**.
6. Select the **Custom** check box.
   The settings become available for change.
7. From the **Certificate** list, select the name of an SSL certificate on the BIG-IP system.

   Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default certificate name, and ensure that this same certificate name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*
8. From the **Key** list, select the name of an SSL key on the BIG-IP system.

   Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default key name, and ensure that this same key name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*

9. In the **Pass Phrase** field, select a pass phrase that enables access to the certificate/key pair on the BIG-IP system.
10. From the **Chain** list, select the name of an SSL chain on the BIG-IP system.
11. For the **Ciphers** setting, specify a cipher group or cipher string by choosing one of these options.

Note: *If you specified an ECDSA certificate key chain in the* ***Certificate Key Chain*** *setting, you must include the cipher string* `ECDHE_ECDSA` *in the cipher group or cipher string that you specify in the* ***Ciphers*** *setting. (At a minimum, you should specify a cipher group or string such as* `DEFAULT:ECDHE_ECDSA`*.) This is necessary to ensure successful cipher negotiation when the BIG-IP system is offered an ECDSA-based certificate only.*

Option	Description
Cipher Group	Select an existing cipher group from the list when you want to use a system-defined or custom cipher group to define the ciphers that the BIG-IP system uses for negotiating SSL connections. Here's an example of the `Ciphers` setting where we've selected a custom cipher group that we created earlier. 
Cipher String	Type a cipher string in the box if you want to manually specify a cipher string instead of selecting a cipher group. For security and performance reasons, consider following these recommendations: • Always append ciphers to the `DEFAULT` cipher string. • Type a cipher string that includes the ECC key type, because its shorter length speeds up encryption and decryption while still offering virtually the same level of security. • Disable ADH ciphers but also include the keyword `HIGH`. To do this, just include both `!ADH` and `:HIGH` in your cipher string. • For AES, DES, and RC4 encryption types, make sure you specify the DHE key exchange method. DHE uses *Forward Privacy*, which creates a key that it throws away after each session so that the same session key never gets used twice. When you use DHE, make sure that the SSL private key isn't being shared with a monitoring system or a security device like an intrusion detection or prevention system. Also, diagnostic tools like `ssldump` won't work when you're using Forward Secrecy. • Disable EXPORT ciphers by including `!EXPORT` in the cipher string. • If you can live with removing support for the SSLv3 protocol version, do it. This protocol version is not secure. Simply include `:!SSLv3` in any cipher string you type. Here's an example of the `Ciphers` setting where we have opted to manually type the cipher string `DEFAULT:ECDHE-RSA-AES-128-GCM-SHA256:!ADH:!EXPORT:HIGH`: 

12. Select the **Custom** check box for **Server Authentication**.

13. Modify the settings, as required.
14. Click **Finished**.

To use this profile, you must assign it to a virtual server.

Creating a pool to manage HTTPS traffic

You can create a pool (a logical set of devices, such as web servers, that you group together to receive and process HTTPS traffic) to efficiently distribute the load on your server resources.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, assign **https** or **https_443** by moving it from the **Available** list to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Use the **New Members** setting to add each resource that you want to include in the pool:
   a) In the **Address** field, type an IP address.
   b) In the **Service Port** field type `443` , or select **HTTPS** from the list.
   c) (Optional) Type a priority number in the **Priority** field.
   d) Click **Add**.
8. Click **Finished**.

The HTTPS load balancing pool appears in the Pool List screen.

Creating a virtual server for client-side and server-side HTTPS traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTP traffic over SSL.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

   Note: *The IP address you type must be available and not in the loopback network.*

   ---
5. Type `443` in the **Service Port** field, or select **HTTPS** from the list.

6. For the **HTTP Profile** setting, verify that the default HTTP profile, **http**, is selected.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created and move the name to the **Selected** list.
8. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL profile you previously created and move the name to the **Selected** list.
9. Click **Finished**.

The virtual server now appears in the Virtual Server List screen.

Implementation results

After you complete the tasks in this implementation, the BIG-IP® system ensures that SSL authentication and encryption occurs for both client-side and server-side HTTP traffic. The system performs this authentication and encryption according to the values you specify in the Client SSL and Server SSL profiles.

Implementing SSL Forward Proxy on a Single BIG-IP System

Overview: SSL forward proxy client and server authentication

With the BIG-IP® system's *SSL forward proxy* functionality, you can encrypt all traffic between a client and the BIG-IP system, by using one certificate, and to encrypt all traffic between the BIG-IP system and the server, by using a different certificate.

A client establishes a three-way handshake and SSL connection with the wildcard IP address of the BIG-IP system virtual server. The BIG-IP system then establishes a three-way handshake and SSL connection with the server, and receives and validates a server certificate (while maintaining the separate connection with the client). The BIG-IP system uses the server certificate to create a second unique server certificate to send to the client. The client receives the second server certificate from the BIG-IP system, but recognizes the certificate as originating directly from the server.

Important: *To enable SSL forward proxy functionality, you can either:*

- Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the SSL Forward Proxy settings.
- Create new Client SSL and Server SSL profiles and configure the SSL Forward Proxy settings.

Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable SSL forward proxy functionality.

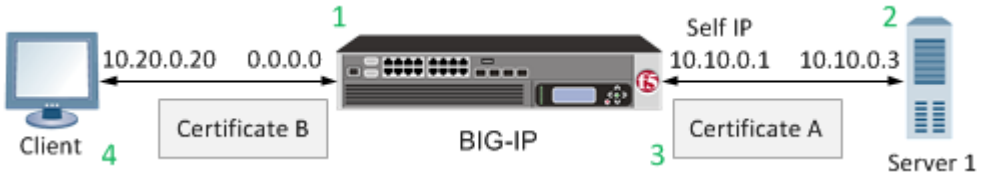


Figure 9: A virtual server configured with Client and Server SSL profiles for SSL forward proxy functionality

1. Client establishes three-way handshake and SSL connection with wildcard IP address.
2. BIG-IP system establishes three-way handshake and SSL connection with server.
3. BIG-IP system validates a server certificate (Certificate A), while maintaining the separate connection with the client.
4. BIG-IP system creates different server certificate (Certificate B) and sends it to client.

Task summary

To implement SSL forward proxy client-to-server authentication, as well as application data manipulation, you perform a few basic configuration tasks. Note that you must create both a Client SSL and a Server SSL profile, and enable the SSL Forward Proxy feature in both profiles.

Task list

Creating a custom Client SSL forward proxy profile

You perform this task to create a Client SSL forward proxy profile that makes it possible for client and server authentication while still allowing the BIG-IP® system to perform data optimization, such as decryption and encryption. This profile applies to client-side SSL forward proxy traffic only.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client SSL profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. From the **SSL Forward Proxy** list, select **Advanced**.
6. Select the **Custom** check box for the SSL Forward Proxy area.
7. Modify the SSL Forward Proxy settings.
   a) From the **SSL Forward Proxy** list, select **Enabled**.
   b) From the **CA Certificate** list, select a certificate.

   Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default certificate name, and ensure that this same certificate name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*

   c) From the **CA Key** list, select a key.

   Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default key name, and ensure that this same key name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*

   d) In the **CA Passphrase** field, type a passphrase.
   e) In the **Confirm CA Passphrase** field, type the passphrase again.
   f) In the **Certificate Lifespan** field, type a lifespan for the SSL forward proxy certificate in days.
   g) (Optional) From the **Certificate Extensions** list, select **Extensions List**.
   h) (Optional) For the **Certificate Extensions List** setting, select the extensions that you want in the **Available extensions** field, and move them to the **Enabled Extensions** field using the **Enable** button.
   i) Select the **Cache Certificate by Addr-Port** check box if you want to cache certificates by IP address and port number.
   j) From the **SSL Forward Proxy Bypass** list, select **Enabled**.
   Additional settings display.
   k) From the **Bypass Default Action** list, select **Intercept** or **Bypass**.

   The default action applies to addresses and hostnames that do not match any entry specified in the lists that you specify. The system matches traffic first against destination IP address lists, then source IP address lists, and lastly, hostname lists. Within these, the default action also specifies whether to search the intercept list or the bypass list first.

   Note: *If you select **Bypass** and do not specify any additional settings, you introduce a security risk to your system.*

8. Click **Finished**.

The custom Client SSL forward proxy profile now appears in the Client SSL profile list screen.

Creating a custom Server SSL forward proxy profile

You perform this task to create a Server SSL forward proxy profile that makes it possible for client and server authentication while still allowing the BIG-IP® system to perform data optimization, such as decryption and encryption. This profile applies to server-side SSL forward proxy traffic only.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The Server SSL profile list screen opens.
2. Click **Create**.
   The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list select **serverssl**.
5. Select the **Custom** check box for the Configuration area.
6. From the **SSL Forward Proxy** list, select **Enabled**.
7. Click **Finished**.

The custom Server SSL forward proxy profile now appears in the Server SSL profile list screen.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

Note: *You must create the pool before you create the corresponding virtual server.*

---

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click **<<** to move the monitor to the **Active** list.

   Tip: *Hold the Shift or Ctrl key to select more than one monitor at a time.*

   ---
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.
   c) In the **Service Port** field, type a port number, or select a service name from the list.
   d) (Optional) In the **Priority** field, type a priority number.
   e) Click **Add**.
8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server for client-side and server-side SSL traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage application traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For a network, in the **Destination Address** field, type an IPv4 or IPv6 address in CIDR format to allow all traffic to be translated.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `0.0.0.0/0`, and an IPv6 address/prefix is `::/0`.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL forward proxy profile you previously created, and using the Move button, move the name to the **Selected** list.

   Important: *To enable SSL forward proxy functionality, you can either:*

   - Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the SSL Forward Proxy settings.
   - Create new Client SSL and Server SSL profiles and configure the SSL Forward Proxy settings.

   Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable SSL forward proxy functionality.
7. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL forward proxy profile you previously created, and using the Move button, move the name to the **Selected** list.

   Important: *To enable SSL forward proxy functionality, you can either:*

   - Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the SSL Forward Proxy settings.
   - Create new Client SSL and Server SSL profiles and configure the SSL Forward Proxy settings.

   Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable SSL forward proxy functionality.
8. Assign other profiles to the virtual server if applicable.
9. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
10. Click **Finished**.

The virtual server now appears in the Virtual Server List screen.

Implementation result

After you complete the tasks in this implementation, the BIG-IP® system ensures that the client system and server system can authenticate each other independently. After client and server authentication, the

BIG-IP system can intelligently decrypt and manipulate the application data according to the configuration settings in the profiles assigned to the virtual server.

Implementing Proxy SSL on a Single BIG-IP System

Overview: Direct client-server authentication with application optimization

When setting up the BIG-IP® system to process application data, you might want the destination server to authenticate the client system directly, for security reasons, instead of relying on the BIG-IP system to perform this function. Retaining direct client-server authentication provides full transparency between the client and server systems, and grants the server final authority to allow or deny client access.

The feature that makes it possible for this direct client-server authentication is known as *Proxy SSL*. You enable this feature when you configure the Client SSL and Server SSL profiles.

Note: *To use this feature, you must configure both a Client SSL and a Server SSL profile.*

Without the Proxy SSL feature enabled, the BIG-IP system establishes separate client-side and server-side SSL connections and then manages the initial authentication of both the client and server systems.

With the Proxy SSL feature, the BIG-IP system makes it possible for direct client-server authentication by establishing a secure SSL tunnel between the client and server systems and then forwarding the SSL handshake messages from the client to the server and vice versa. After the client and server successfully authenticate each other, the BIG-IP system uses the tunnel to decrypt the application data and intelligently manipulate (optimize) the data as needed.

Task summary

To implement direct client-to-server SSL authentication, as well as application data manipulation, you perform a few basic configuration tasks. Note that you must create both a Client SSL and a Server SSL profile, and enable the Proxy SSL feature in both profiles.

Before you begin, verify that the client system, server system, and BIG-IP® system contain the appropriate SSL certificates for mutual authentication.

Important: *The BIG-IP certificate and key referenced in a Server SSL profile must match those of the server system.*

As you configure your network for Proxy SSL, keep in mind the following considerations:

- Proxy SSL supports only the RSA key exchange. For proper functioning, the client and server must not negotiate key exchanges or cipher suites that Proxy SSL does not support, such as the Diffie-Hellman (DH) and Ephemeral Diffie-Hellman (DHE) key exchanges, and the Elliptic Curve Cryptography (ECC) cipher suite. To avoid this issue, you can either configure the client so that the ClientHello packet does not include DH, DHE, or ECC; or configure the server to not accept DH, DHE, or ECC.
- Proxy SSL supports only the NULL compression method.

Task list

Creating a custom Server SSL profile

You perform this task to create a Server SSL profile that makes it possible for direct client-server authentication while still allowing the BIG-IP® system to perform data optimization, such as decryption and encryption. This profile applies to server-side SSL traffic only.

Important: *The certificate and key that you specify in this profile must match the certificate/key pair that you expect the back-end server to offer. If the back-end server has two or more certificates to offer, you must create a separate Server SSL profile for each certificate and then assign all of the Server SSL profiles to a single virtual server.*

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The Server SSL profile list screen opens.
2. Click **Create**.
   The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **serverssl** in the **Parent Profile** list.
5. From the **Certificate** list, select a relevant certificate name.
6. From the **Key** list, select a relevant key name.
7. For the **Proxy SSL** setting, select the check box.
8. From the **Configuration** list, select **Advanced**.
9. Modify all other settings, as required.
10. Choose one of the following actions:
    - If you need to create another Server SSL profile, click **Repeat**.
    - If you do not need to create another Server SSL profile, click **Finished**.

All relevant Server SSL profiles now appear on the SSL Server profile list screen.

Creating a custom Client SSL profile

You perform this task to create a Client SSL profile that makes it possible for direct client-server authentication while still allowing the BIG-IP system to perform data optimization, such as decryption and encryption. This profile applies to client-side SSL traffic only.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client SSL profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **clientssl** in the **Parent Profile** list.
5. For the **Proxy SSL** setting, select the check box.
6. From the **Configuration** list, select **Advanced**.
7. Modify all other settings, as required.
8. Click **Finished**.

The custom Client SSL profile appears in the Client SSL profile list screen.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

Note: *You must create the pool before you create the corresponding virtual server.*

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.

2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click **<<** to move the monitor to the **Active** list.

   Tip: *Hold the Shift or Ctrl key to select more than one monitor at a time.*
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.
   c) In the **Service Port** field, type a port number, or select a service name from the list.
   d) (Optional) In the **Priority** field, type a priority number.
   e) Click **Add**.
8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server for client-side and server-side SSL traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage application traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Destination Address/Mask** field, type an address, as appropriate for your network.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the custom Client SSL proxy profile you previously created, and using the Move button, move the name to the **Selected** list.

   Important: *To enable proxy SSL functionality, you can either:*
   - Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the Proxy SSL settings.
   - Create new Client SSL and Server SSL profiles and configure the Proxy SSL settings.

Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable proxy SSL functionality.

7. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the custom Server SSL proxy profile you previously created and move the name to the **Selected** list.

Important: *To enable SSL proxy functionality, you can either:*

- Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the Proxy SSL settings.
- Create new Client SSL and Server SSL profiles and configure the Proxy SSL settings.

Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable SSL proxy functionality.

8. Assign other profiles to the virtual server if applicable.
9. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
10. Click **Finished**.

The virtual server now appears in the Virtual Server List screen.

Implementation result

After you complete the tasks in this implementation, the BIG-IP® system ensures that the client system and server system can initially authenticate each other directly. After client-server authentication, the BIG-IP system can intelligently decrypt and manipulate the application data according to the configuration settings in the profiles assigned to the virtual server.

Securing Client-Side SMTP Traffic

Overview: Securing client-side SMTP traffic

You can add SSL encryption to SMTP traffic quickly and easily, by configuring an SMTPS profile on the BIG-IP® system. *SMTPS* is a method for securing Simple Mail Transport Protocol (SMTP) connections at the transport layer.

Normally, SMTP traffic between SMTP servers and clients is unencrypted. This creates a privacy issue because SMTP traffic often passes through routers that the servers and clients do not trust, resulting in a third party potentially changing the communications between the server and client. Also, two SMTP systems do not normally authenticate each other. A more secure SMTP server might only allow communications from other known SMTP systems, or the server might act differently with unknown systems.

To mitigate these problems, the BIG-IP system includes an SMTPS profile that you can configure. When you configure an SMTPS profile, you can activate support for the industry-standard STARTTLS extension to the SMTP protocol, by instructing the BIG-IP system to either allow, disallow, or require STARTTLS activation for SMTP traffic. The STARTTLS extension effectively upgrades a plain-text connection to an encrypted connection on the same port, instead of using a separate port for encrypted communication.

This illustration shows a basic configuration of a BIG-IP system that uses SMTPS to secure SMTP traffic between the BIG-IP system and an SMTP mail server.

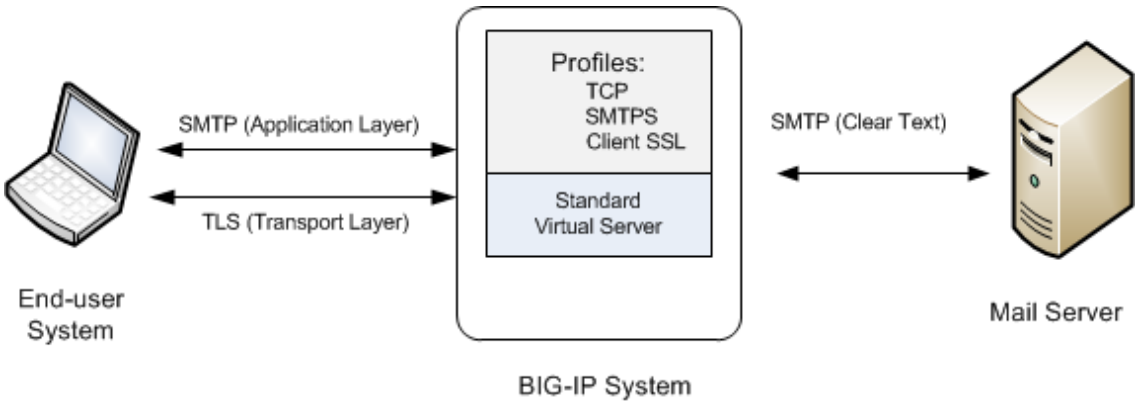


Figure 10: Sample BIG-IP configuration for SMTP traffic with STARTTLS activation

Task summary

To configure the BIG-IP ®system to process Simple Mail Transport Protocol (SMTP) traffic with SSL functionality, you perform a few basic tasks.

Task list

Creating an SMTPS profile

This task specifies that STARTTLS authentication and encryption should be required for all client-side Simple Mail Transport Protocol (SMTP) traffic. When you require STARTTLS for SMTP traffic, the BIG-IP® system effectively upgrades SMTP connections to include SSL, on the same SMTP port.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **SMTPS**.
   The SMTPS profile list screen opens.
2. Click **Create**.
   The New SMTPS Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select the **Custom** check box.
5. From the **STARTTLS Activation Mode** list, select **Require**.
6. Click **Finished**.

The BIG-IP system is now required to activate STARTTLS for all client-side SMTP traffic.

Creating a Client SSL profile

You create a Client SSL profile when you want the BIG-IP® system to authenticate and decrypt/encrypt client-side application traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client SSL profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. Configure all profile settings as needed.
4. Click **Finished**.

After creating the Client SSL profile and assigning the profile to a virtual server, the BIG-IP system can apply SSL security to the type of application traffic for which the virtual server is configured to listen.

Creating a virtual server and load-balancing pool

You use this task to create a virtual server, as well as a default pool of Simple Mail Transport Protocol (SMTP) servers. The virtual server listens for, and applies SSL security to, client-side SMTP application traffic. The virtual server then forwards the SMTP traffic on to the specified server pool.

Note: *Using this task, you assign an SMTPS profile to the virtual server instead of an SMTP profile. You must also assign a Client SSL profile.*

---

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Destination Address/Mask** field, type an address, as appropriate for your network.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.

5. In the **Service Port** field, type `25` or select **SMTP** from the list.
6. From the **Configuration** list, select **Basic**.
7. For the **SSL Profile (Client)** setting, in the **Available** box, select a profile name, and using the Move button, move the name to the **Selected** box.
8. From the **SMTPS Profile** list, select the SMTPS profile that you previously created.
9. In the Resources area of the screen, for the **Default Pool** setting, click the **Create (+)** button.
   The New Pool screen opens.
10. In the **Name** field, type a unique name for the pool.
11. In the Resources area, for the **New Members** setting, select the type of new member you are adding, then type the information in the appropriate fields, and click **Add** to add as many pool members as you need.
12. Click **Finished** to create the pool.
    The screen refreshes, and reopens the New Virtual Server screen. The new pool name appears in the **Default Pool** list.
13. Click **Finished**.

After performing this task, the virtual server applies the custom SMTPS and Client SSL profiles to incoming SMTP traffic.

Implementation result

After you have created an SMTPS profile and a Client SSL profile and assigned them to a virtual server, the BIG-IP system listens for client-side SMTP traffic on port 25. The BIG-IP system then activates the STARTTLS method for that traffic, to provide SSL security on that same port, before forwarding the traffic on to the specified server pool.

Securing Client-Side and Server-Side LDAP Traffic

Overview: Securing LDAP traffic with STARTTLS encryption

You can configure STARTTLS encryption for Lightweight Directory Access Protocol (LDAP) traffic passing through the BIG-IP® system. *LDAP* is an industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network.

You configure the BIG-IP system for STARTTLS encryption by configuring Client LDAP and Server LDAP profiles to activate the STARTTLS communication protocol for any client or server traffic that allows or requires STARTTLS encryption.

Normally, LDAP traffic between LDAP servers and clients is unencrypted. This creates a privacy issue because LDAP traffic often passes through routers that the servers and clients do not trust, resulting in a third party potentially changing the communications between the server and client. Also, two LDAP systems do not normally authenticate each other. A more secure LDAP server might only allow communications from other known LDAP systems, or the server might act differently with unknown systems.

To mitigate these problems, the BIG-IP system includes two LDAP profiles that you can configure. When you configure a Client LDAP or Server LDAP profile, you can instruct the BIG-IP system to activate the STARTTLS communication protocol for any client or server traffic that allows or requires STARTTLS encryption. The *STARTTLS* protocol effectively upgrades a plain-text connection to an encrypted connection on the same port (port 389), instead of using a separate port for encrypted communication.

This illustration shows a basic configuration of a BIG-IP system that activates STARTTLS to secure LDAP traffic between a client system and the BIG-IP system, and between the BIG-IP system and an LDAP authentication server.

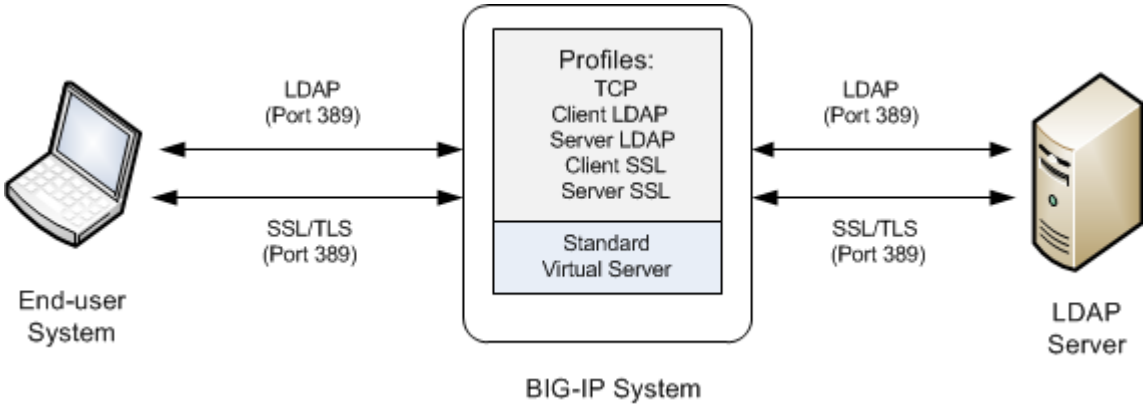


Figure 11: Sample BIG-IP configuration for LDAP traffic with STARTTLS activation

Task summary

To configure the BIG-IP ®system to process Lightweight Directory Access Protocol (LDAP) traffic with TLS encryption, you perform a few basic tasks.

Task list

Creating a Client LDAP profile

You perform this task to specify the condition under which the BIG-IP system should activate STARTTLS encryption for client-side traffic destined for a specific virtual server.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **Client LDAP**.
   The Client LDAP list screen displays.
2. Click **Create**.
   The New Client LDAP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, retain the default value, **clientldap**.
5. Select the **Custom** check box.
6. From the **STARTTLS Activation Mode** list, select a value:

Value	Description
Allow	This value activates STARTTLS encryption for any client-side traffic that allows, but does not require, STARTTLS encryption.
Require	This value activates STARTTLS encryption for any client-side traffic that requires STARTTLS encryption. All messages sent to the BIG-IP system prior to STARTTLS activation are rejected with a message stating that a stronger authentication mechanism is required.
None	This value refrains from activating STARTTLS encryption for client-side traffic. Note if you select this value, that you optionally can create an iRule that identifies client-side traffic that requires STARTTLS encryption and then dynamically activates STARTTLS for that particular traffic.

7. Click **Finished**.

After you perform this task, the Client LDAP profile appears on the Client LDAP list screen.

Creating a Server LDAP profile

You perform this task to specify the condition under which the BIG-IP system should activate STARTTLS encryption for server-side traffic destined for a specific virtual server.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **Server LDAP**.
   The Server LDAP list screen displays.
2. Click **Create**.
   The New Server LDAP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, retain the default value, **serverldap**.
5. Select the **Custom** check box.
6. From the **STARTTLS Activation Mode** list, select a value:

Value	Description
Allow	This value activates STARTTLS encryption for server-side traffic that allows, but does not require, STARTTLS encryption. In this case, the BIG-IP system only activates STARTTLS for server-side traffic when the BIG-IP system has activated STARTTLS on the client side and the client has acknowledged the activation.

Value	Description
Require	This value activates STARTTLS encryption for any server-side traffic that requires STARTTLS encryption. In this case, the BIG-IP system activates STARTTLS when a successful connection is made.
None	This value refrains from activating STARTTLS encryption for server-side traffic. Note that if you select this value, you can optionally create an iRule that identifies server-side traffic that requires STARTTLS encryption and then dynamically activates STARTTLS for that particular traffic.

7. Click **Finished**.

After you perform this task, the Server LDAP profile appears on the Server LDAP list screen.

Creating a custom Client SSL profile

After you have built the cipher string that you want the BIG-IP to use to negotiate client-side SSL connections, you create a custom Client SSL profile. You create the profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and encrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads these decryption/encryption functions from the destination server. When you perform this task, you can specify multiple certificate key chains, one for each key type (RSA, DSA, and ECDSA). This allows the BIG-IP system to negotiate secure client connections using different cipher suites based on the client's preference.

Note: *At a minimum, you must specify a certificate key chain that includes an RSA key pair. Specifying certificate key chains for DSA and ECDSA key pairs is optional, although highly recommended.*

---

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client SSL profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.
   The settings become available for change.
6. For the **Certificate Key Chain** setting, click **Add**.
   a) From the **Certificate** list, select a certificate name.

      This is the name of a certificate that you installed on the BIG-IP® system. If you have not generated a certificate request nor installed a certificate on the BIG-IP system, and the BIG-IP system is not part of a device service clustering (DSC) configuration, you can specify the name of the existing certificate named `default`.

      Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default certificate name, and ensure that this same certificate name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*

      ---

   b) From the **Key** list, select the name of the key associated with the certificate specified in the previous step.

      This is the name of a key that you installed on the BIG-IP® system. If you have not installed a key on the BIG-IP system, and the BIG-IP system is not part of a device service clustering (DSC) configuration, you can specify the name of the existing key named `default`.

Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default key name, and ensure that this same key name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*

c) From the **Chain** list, select the chain that you want to include in the certificate key chain.

A certificate chain can contain either a series of public key certificates in Privacy Enhanced Mail (PEM) format or a series of one or more PEM files. A certificate chain can contain certificates for Intermediate certificate Authorities (CAs).

Note: *The default self-signed certificate and the default CA bundle certificate are not appropriate for use as a certificate chain.*

d) For the **Passphrase** field, type a string that enables access to SSL certificate/key pairs that are stored on the BIG-IP system with password protection.

This setting is optional. For added security, the BIG-IP system automatically encrypts the pass phrase itself. This pass phrase encryption process is invisible to BIG-IP® system administrative users.

e) Click **Add**.

7. Click **Add** and repeat the process for all certificate key chains that you want to specify. At a minimum, you must specify an RSA certificate key chain.

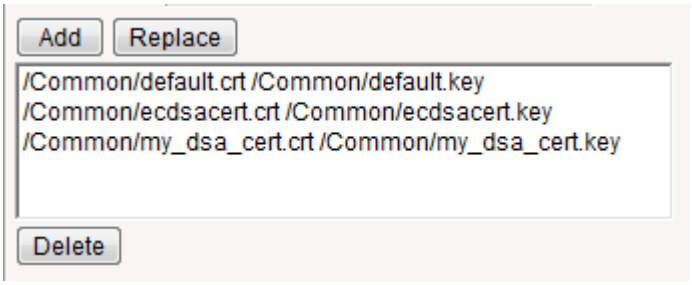


Figure 12: Sample configuration with three key types specified

The result is that all specified key chains appear in the text box.

8. For the **OCSP Stapling** setting, select the check box.

This setting is optional. To enable OCSP stapling, you must first create an OCSP Stapling profile.

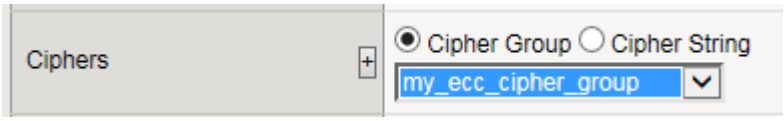
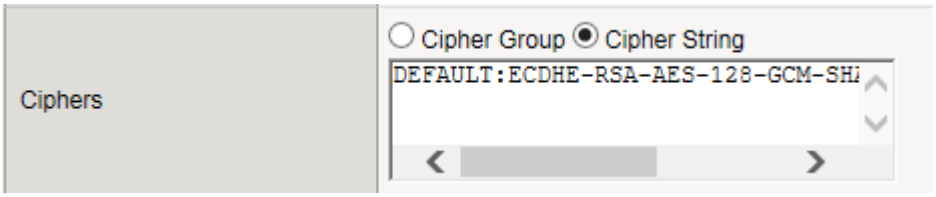
9. For the **Notify Certificate Status to Virtual Server** setting, select the check box.

This setting is optional.

10. For the **Ciphers** setting, specify a cipher group or cipher string by choosing one of these options.

Note: *If you specified an ECDSA certificate key chain in the* ***Certificate Key Chain*** *setting, you must include the cipher string* `ECDHE_ECDSA` *in the cipher group or cipher string that you specify in the* ***Ciphers*** *setting. (At a minimum, you should specify a cipher group or string such as* `DEFAULT:ECDHE_ECDSA`*.) This is necessary to ensure successful cipher negotiation when the BIG-IP system is offered an ECDSA-based certificate only.*

Option	Description
Cipher Group	Select an existing cipher group from the list when you want to use a system-defined or custom cipher group to define the ciphers that the BIG-IP system uses for negotiating SSL connections. Here's an example of the `Ciphers` setting where we've selected a custom cipher group that we created earlier.

Option	Description
	
Cipher String	Type a cipher string in the box if you want to manually specify a cipher string instead of selecting a cipher group. For security and performance reasons, consider following these recommendations: • Always append ciphers to the `DEFAULT` cipher string. • Type a cipher string that includes the ECC key type, because its shorter length speeds up encryption and decryption while still offering virtually the same level of security. • Disable ADH ciphers but also include the keyword `HIGH`. To do this, just include both `!ADH` and `:HIGH` in your cipher string. • For AES, DES, and RC4 encryption types, make sure you specify the DHE key exchange method. DHE uses *Forward Privacy*, which creates a key that it throws away after each session so that the same session key never gets used twice. When you use DHE, make sure that the SSL private key isn't being shared with a monitoring system or a security device like an intrusion detection or prevention system. Also, diagnostic tools like `ssldump` won't work when you're using Forward Secrecy. • Disable EXPORT ciphers by including `!EXPORT` in the cipher string. • If you can live with removing support for the SSLv3 protocol version, do it. This protocol version is not secure. Simply include `:!SSLv3` in any cipher string you type. Here's an example of the `Ciphers` setting where we have opted to manually type the cipher string `DEFAULT:ECDHE-RSA-AES-128-GCM-SHA256:!ADH:!EXPORT:HIGH`: 

11. Configure all other profile settings as needed.
12. Click **Finished**.

After performing this task, you can see the custom Client SSL profile in the list of Client SSL profiles on the system.

You must also assign the profile to a virtual server.

Creating a custom Server SSL profile

With a Server SSL profile, the BIG-IP® system can perform decryption and encryption for server-side SSL traffic.

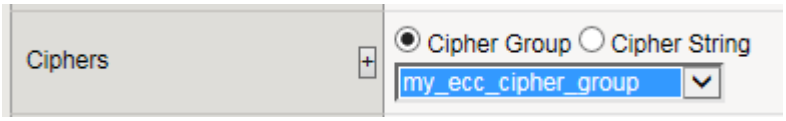
1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The Server SSL profile list screen opens.
2. Click **Create**.
   The New Server SSL Profile screen opens.

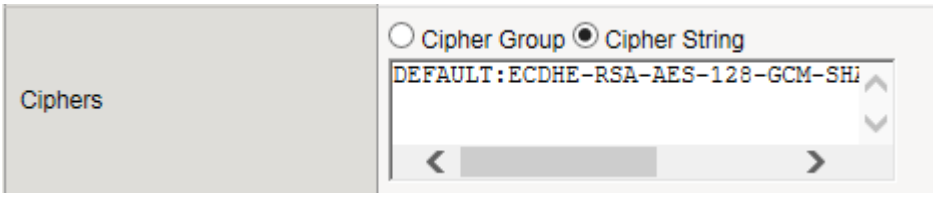
3. In the **Name** field, type a unique name for the profile.
4. Select **serverssl** in the **Parent Profile** list.
5. From the **Configuration** list, select **Advanced**.
6. Select the **Custom** check box.
   The settings become available for change.
7. From the **Certificate** list, select the name of an SSL certificate on the BIG-IP system.

   Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default certificate name, and ensure that this same certificate name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*
8. From the **Key** list, select the name of an SSL key on the BIG-IP system.

   Important: *If the BIG-IP system is part of a DSC Sync-Failover group, always select a non-default key name, and ensure that this same key name is specified in every instance of this SSL profile in the device group. Taking these actions helps to ensure that SSL handshakes are successful after a failover event.*
9. In the **Pass Phrase** field, select a pass phrase that enables access to the certificate/key pair on the BIG-IP system.
10. From the **Chain** list, select the name of an SSL chain on the BIG-IP system.
11. For the **Ciphers** setting, specify a cipher group or cipher string by choosing one of these options.

    Note: *If you specified an ECDSA certificate key chain in the* ***Certificate Key Chain*** *setting, you must include the cipher string* `ECDHE_ECDSA` *in the cipher group or cipher string that you specify in the* ***Ciphers*** *setting. (At a minimum, you should specify a cipher group or string such as* `DEFAULT:ECDHE_ECDSA`*.) This is necessary to ensure successful cipher negotiation when the BIG-IP system is offered an ECDSA-based certificate only.*

Option	Description
Cipher Group	Select an existing cipher group from the list when you want to use a system-defined or custom cipher group to define the ciphers that the BIG-IP system uses for negotiating SSL connections. Here's an example of the `Ciphers` setting where we've selected a custom cipher group that we created earlier. 
Cipher String	Type a cipher string in the box if you want to manually specify a cipher string instead of selecting a cipher group. For security and performance reasons, consider following these recommendations: • Always append ciphers to the `DEFAULT` cipher string. • Type a cipher string that includes the ECC key type, because its shorter length speeds up encryption and decryption while still offering virtually the same level of security. • Disable ADH ciphers but also include the keyword `HIGH`. To do this, just include both `!ADH` and `:HIGH` in your cipher string. • For AES, DES, and RC4 encryption types, make sure you specify the DHE key exchange method. DHE uses *Forward Privacy*, which creates a key that it throws away after each session so that the same session key never gets used

Option	Description
	twice. When you use DHE, make sure that the SSL private key isn't being shared with a monitoring system or a security device like an intrusion detection or prevention system. Also, diagnostic tools like `ssldump` won't work when you're using Forward Secrecy. • Disable EXPORT ciphers by including `!EXPORT` in the cipher string. • If you can live with removing support for the SSLv3 protocol version, do it. This protocol version is not secure. Simply include `:!SSLv3` in any cipher string you type. Here's an example of the `Ciphers` setting where we have opted to manually type the cipher string `DEFAULT:ECDHE-RSA-AES-128-GCM-SHA256:!ADH:!EXPORT:HIGH`: 

12. Select the **Custom** check box for **Server Authentication**.
13. Modify the settings, as required.
14. Click **Finished**.

To use this profile, you must assign it to a virtual server.

Creating a virtual server and load-balancing pool

You use this task to create a virtual server, as well as a default pool of LDAP servers. The virtual server then listens for and applies the configured STARTTLS activation to client-side or server-side LDAP traffic, or both. Part of creating this virtual server is specifying the names of any client-side and server-side LDAP and SSL profiles that you previously created.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Destination Address/Mask** field, type an address, as appropriate for your network.
   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or `2001:ed8:77b5:2:10:10:100:42/64`. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a `/32` prefix.
5. In the **Service Port** field, type `389` or select **LDAP** from the list.
6. From the **Configuration** list, select **Basic**.
7. For the **SSL Profile (Client)** setting, in the **Available** box, select a profile name, and using the Move button, move the name to the **Selected** box.
8. From the **Client LDAP Profile** list, select the Client LDAP profile that you previously created.
9. From the **Server LDAP Profile** list, select the Server LDAP profile that you previously created.
10. In the Resources area of the screen, for the **Default Pool** setting, click the **Create (+)** button.
    The New Pool screen opens.
11. In the **Name** field, type a unique name for the pool.

12. In the Resources area, for the **New Members** setting, select the type of new member you are adding, then type the information in the appropriate fields, and click **Add** to add as many pool members as you need.
13. Click **Finished** to create the pool.
    The screen refreshes, and reopens the New Virtual Server screen. The new pool name appears in the **Default Pool** list.
14. Click **Finished**.

After performing this task, the virtual server applies the custom LDAP and SSL profiles to ingress traffic.

Implementation result

After you have created the required LDAP and SSL profiles and assigned them to a virtual server, the BIG-IP® system listens for client- and server-side LDAP traffic on port 389. The BIG-IP system then activates the STARTTLS method for that traffic to provide SSL security on that same port, before forwarding the traffic on to the specified LDAP server pool.

Implementing External Cryptographic Server Offload with BIG-IP Systems

Overview: Implementing external cryptographic server offload

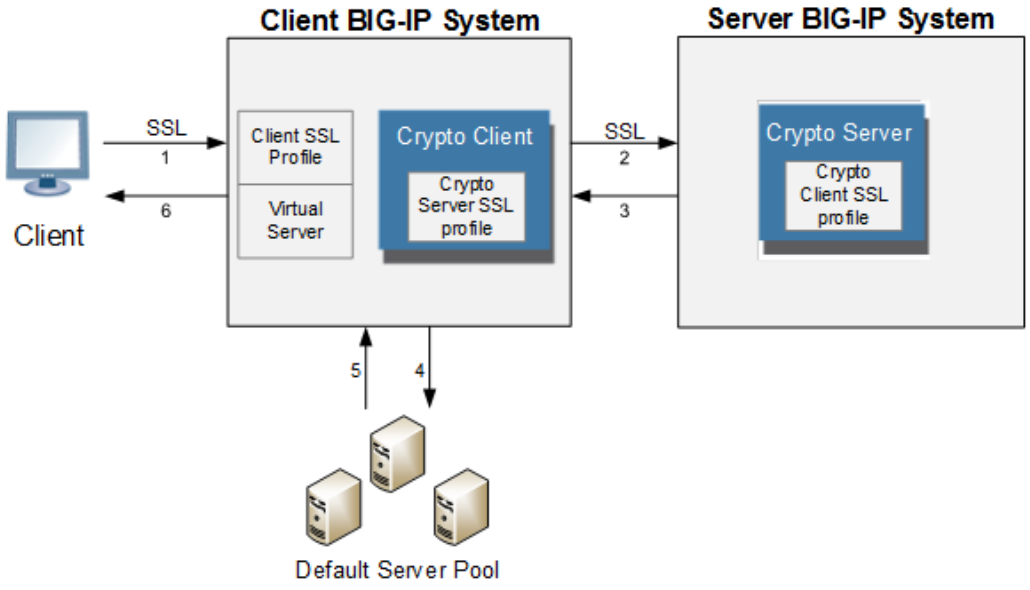
You can offload cryptographic operations to an external BIG-IP® system. For example, you can set up an LTM VE instance (the crypto client) to offload cryptographic operations, such as an RSA decryption operation for an SSL handshake, to an external BIG-IP system (the crypto server) that supports crypographic hardware acceleration.

In general, the setup process includes configuring a client BIG-IP system as a crypto client and a server BIG-IP system as a crypto server, and ensures secure communication between the end user, the crypto client, and the crypto server.

Important: *Both the crypto client and crypto server must be running BIG-IP software version 11.6.0 or later.*

Important: *Before you perform the tasks in this implementation, verify that each BIG-IP system has the default device certificate,* `default.crt`*, installed on it. For more information about device certificates, see BIG-IP® Digital Certificates: Administration.*

This illustration depicts an external cryptographic offload configuration.



The illustration shows the BIG-IP configuration objects that are required for implementing the external cryptographic server offload feature, as well as the flow of client traffic that occurs. In the illustration, one BIG-IP system includes a virtual server configured with the destination IP address for application traffic coming from a client system. Because the client traffic uses SSL, the BIG-IP system with the virtual server must include a standard Client SSL profile, which causes cryptographic functions to be offloaded from the selected destination server (pool member) to that BIG-IP system.

Once this BIG-IP system has assumed cryptographic functions from the destination server, the BIG-IP system can offload these functions to another BIG-IP system to handle the actual cryptographic processing. To enable the BIG-IP system to offload the cryptographic processing to another BIG-IP system, you must designate the two BIG-IP systems as a crypto client and crypto server, and you must

create an SSL profile on each system that is optimized for BIG-IP-to-BIG-IP cryptographic processing (a crypto-optimized Server SSL profile for the BIG-IP crypto client and crypto-optimized Client SSL profile for the BIG-IP crypto server).

Task summary

Creating a Client SSL profile on a client BIG-IP system

You create a Client SSL profile on a client BIG-IP® system to authenticate and decrypt/encrypt client-side application traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client SSL profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. Configure all profile settings as needed.
4. Click **Finished**.

After you create the Client SSL profile, you assign the profile to a virtual server. The BIG-IP® system can apply SSL security to the type of application traffic for which the virtual server is configured to listen.

Creating a pool on a client BIG-IP system

You can create a pool of servers on a client BIG-IP® system that you can group together to receive and process traffic.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) (Optional) In the **Node Name** field, type a name for the node portion of the pool member.
   b) In the **Address** field, type an IP address.
   c) In the **Service Port** field, type a port number, or select a service name from the list.
   d) (Optional) In the **Priority** field, type a priority number.
   e) Click **Add**.
5. Click **Finished**.

Creating a virtual server on a client BIG-IP system

A virtual server represents a destination IP address for application traffic on a client BIG-IP® system.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Destination Address** field, type the IP address in CIDR format.

   The supported format is address/prefix, where the prefix length is in bits. For example, an IPv4 address/prefix is `10.0.0.1` or `10.0.0.0/24`, and an IPv6 address/prefix is `ffe1::0020/64` or

2001:ed8:77b5:2:10:10:100:42/64. When you use an IPv4 address without specifying a prefix, the BIG-IP® system automatically uses a /32 prefix.

Note: *The IP address for this field needs to be on the same subnet as the external self-IP address.*

5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. In the Resources area of the screen, from the **Default Pool** list, select the relevant pool name.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created and move the name to the **Selected** list.

Creating a Server SSL profile on a client BIG-IP system

With a Server SSL profile, a client BIG-IP® system can perform decryption and encryption for server-side SSL traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The Server SSL profile list screen opens.
2. Click **Create**.
   The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **crypto-client-default-serverssl** in the **Parent Profile** list.
5. Modify the settings, as required.
6. Click **Finished**.

Creating a crypto client object on a client BIG-IP system

You can create a crypto client object to enable a BIG-IP® system to act as a crypto client for external cryptographic server offload.

1. On the Main tab, click **System** > **Crypto Offloading** > **Crypto Client**.
   The Crypto Client screen displays a list of crypto clients configured on the system.
2. Click **Create**.
3. In the **Name** field, type a unique name for the crypto client object.
4. In the **Address** field, type the IP address of the crypto server that you want to use for the crypto server object.
5. In the **Service Port** field, type a port number, or select a service name from the list.
6. In the **TCP Profiles** field, select **tcp**.
7. For the **SSL Profiles** setting, select the Server SSL profile that you previously created.

Creating a Client SSL profile on a server BIG-IP system

You create a Client SSL profile on a server BIG-IP® system to authenticate and decrypt/encrypt application traffic from the client BIG-IP system.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client SSL profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. Select **crypto-server-default-clientssl** in the **Parent Profile** list.
4. Configure all profile settings as needed.
5. Click **Finished**.

Creating a crypto server object on a server BIG-IP system

You can create a crypto server object to enable your BIG-IP® system to act as a crypto server for external cryptographic server offload.

1. On the Main tab, click **System** > **Crypto Offloading** > **Crypto Server**.
   The Crypto Server screen displays a list of crypto servers configured on the system.
2. Click **Create**.
3. In the **Name** field, type a unique name for the crypto server object.
4. In the **Address** field, type the IP address you want to use for the crypto server object.
5. In the **Service Port** field, type a port number, or select a service name from the list.
6. In the **TCP Profiles** field, select **tcp**.
7. For the **SSL Profiles** setting, select the Client SSL profile that you previously created.
8. (Optional) Using the **Crypto Client List** setting, add the crypto clients that can access the crypto server:
   a) In the **Address** field, type a crypto client self IP address.
   b) Click **Add**.

Verifying the crypto client and crypto server

After the client and server BIG-IP® systems have processed traffic, you can use `tmsh` to verify that the crypto client and crypto server systems are functioning properly.

1. Open the Traffic Management Shell (`tmsh`).
   ```
   tmsh
   ```
2. Verify that the crypto client is functioning.
   ```
   show sys crypto client <crypto_client_name>
   ```
   A summary similar to this example displays:

```
-------------------------
Sys::Crypto Client: crypto_client_name
-------------------------
  Received Packets      2
  Received Bytes       48
  Transmitted Packets   2
  Transmitted Bytes    40
```

3. Verify that the crypto server is functioning.
   ```
   show sys crypto server <crypto_server_name>
   ```
   A summary similar to this example displays:

```
-------------------------
Sys::Crypto Server: crypto_server_name
-------------------------
  Received Packets      2
  Received Bytes       40
  Transmitted Packets   2
  Transmitted Bytes    48
```

Legal Notices

Legal notices

Publication Date

This document was published on February 20, 2019.

Publication Number

MAN-0527-05

Copyright

Trademarks

For a current list of F5 trademarks and service marks, see *http://www.f5.com/about/guidelines-policies/trademarks*.

All other product and company names herein may be trademarks of their respective owners.

Patents

This product may be protected by one or more patents indicated at: *https://f5.com/about-us/policies/patents*.

Link Controller Availability

This product is not currently available in the U.S.

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a

residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Index

A

B

C

D

E

H

K

L

M

N

O

P

R

S

T

U

V