Exinda How To Guide:

SSL Acceleration

Exinda ExOS Version 7.4.3

Copyright

Document Built on Friday, July 22, 2016 at 2:54 PM

Using this guide

Before using this guide, become familiar with the Exinda documentation system.

Documentation conventions

These documentation conventions apply across all of the Exinda documentation sets. All instances of the following may not appear in this documentation

Typographical conventions

- **bold** - Interface element such as buttons or menus. For example: Select the **Enable** checkbox.
- *italics* - Reference to other documents. For example: Refer to the *Exinda Application List*. Also used to identify in the various procedures the response the systems provide after applying an action.
- **>** - Separates navigation elements. For example: Select **File > Save**.
- `monospace text` - Command line text.
- `<variable>` - Command line arguments.
- `[x]` - An optional CLI keyword or argument.
- `{x}` - A required CLI element.
- `|` - Separates choices within an optional or required element.

Links

With the exception of the various tables of contents, all links throughout the documentation are **blue**. Most links refer to topics within the documentation, but there may be links that take you to web pages on the Internet. In this documentation we differentiate between these types of links by **underlining** only the external links.

Tips, Notes, Examples, Cautions, etc.

Throughout this manual, the following table styles are used to highlight important information:

- **Tips** include hints and shortcuts. Tips are identified by the light blub icon.

TIP

text

- **Notes** provide information that is useful at the points where they are encountered. Notes are identified by the pin and paper icon.

NOTE

Text

- **Important** notes provide information that is important at the point where they are encountered. Important notes are identified by the amber triangle.

IMPORTANT

Text

- **Cautions** provide warnings of areas of operation that could cause damage to appliances. Cautions are identified by the orange triange.

CAUTION

Text

- **Examples** are presented throughout the manual for deeper understanding of specific concepts. Examples are identified by a pale green background.

EXAMPLE

Text

- **Best Practices** are identified by the "thumbs-up" icon.

Best Practice:

It is a best practice to

Table of Contents

Chapter 1: SSL Acceleration

How SSL Protocol Acceleration Works

How SSL works

SSL is the standard protocol for establishing a secure encrypted link between a remote application server and the client Web browser on the local user computer. The SSL protocol secures each session link by automatically establishing connections on-demand using standards-based protocols, encryption techniques, and certificate exchange.

SSL encryption requires a certificate on the server to authenticate the identity of a server. A certificate is an electronic confirmation that you, as the owner of a public key, are who you claim to be, and that you hold the private key corresponding to the public key in the certificate.

You create this certificate by generating a certificate and sending a certificate signing request to a Certificate Authority (CA) using your public key. The CA checks with a registration authority to verify your identity and then signs and returns the certificate. You then upload the signed certificate and public key onto the server.

When a client browser visits a web site hosted on the server over HTTPS, the server offers the signed certificate and public key. The client browser verifies that the certificate is valid for the site that is being visited and that it has not expired. Then it will verify the chain of trust by looking at who has signed the certificate:

- If the certificate is a root-certificate, it will compare it against the ones shipped with the OS or browser.
- If it is a non-root-certificate, it will follow the chain of trust up each level until reaching a root-certificate.

Now the server has the private key and the client has the public key effectively creating a private encrypted tunnel that allows them to appropriately communicate by encrypting and decrypting the traffic between them. When the session is over, the connection is automatically terminated.

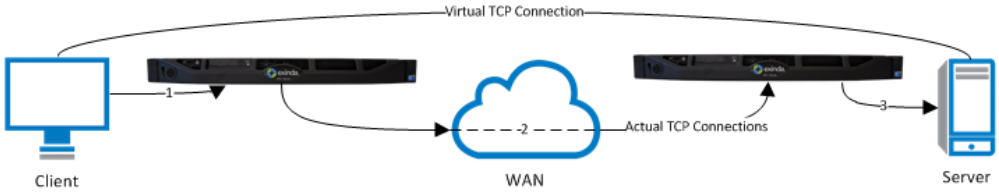
There are other certificate signing options

You can create a self-signed certificate, where the certificate has signed itself and therefore there is no chain of trust. The browser will issue a warning, telling the user that the site certificate cannot be verified. To continue, the user will have to confirm that they trust the site. When the browser visits this site again later, the warning will not be presented again since the user has already confirmed their trust of the site. An alternate use case, is the company that created the self-signed certificate can provide the certificate to the client users and tell them to load the certificate into their browsers. This is equivalent to confirming trust when the warning is shown. Using self-signed certificates is reasonable in situations where you want encryption but you do not need the third party verification, such as an internal system where you want your internal users to have password protection, however as the clients and the server are behind a firewall so you do not need the third party verification.

You can create your own self-signed CA certificate for signing other certificates. In this case, the certificates that your self-signed CA certificate signs will have no chain of trust. Similar to self-signed certificates, using your own self-signed CA certificate to sign other certificates is reasonable in situations where you want encryption but you do not need the third party verification. The difference is that a warning is shown to the client for each server when using self-signed certificates, whereas when using a self-signed CA certificate to sign multiple other certificates, the warning will only be shown once for all the certificates that were signed by the CA certificate, that is, once the client trusts a certificate that is signed by the self-signed CA certificate, the client automatically trusts all other certificates signed by that self-signed CA certificate.

In the case where there are multiple virtual hosts on a single server, a Server Name Indication (SNI) is used to indicate what virtual hostname the client is attempting to connect to during the handshaking process. This allows a single server to present individual certificates for each of its multiple secure websites without requiring all of the sites to use the same certificate.

How Exinda accelerates the SSL protocol



For SSL acceleration, a server-side Exinda appliance and a client-side appliance is put in line for this SSL traffic. The traffic between these appliances are accelerated. The benefits that can be gained by generic application acceleration on encrypted data are limited. For example, the Exinda WAN Memory technology achieves higher reduction on clear text rather than encrypted data. However, the SSL acceleration feature is designed to overcome these limitations by transparently decrypting accelerated traffic, performing the relevant application acceleration techniques such as TCP Acceleration and WAN Memory, then re-encrypting the traffic again. This means Exinda can apply all application acceleration technologies to the traffic as if it were clear text, while still maintaining SSL connections.

The server-side appliance will act on behalf of the client in the communication between the appliance and the server and the client-side appliance will act on behalf of the server for communication between the client and the appliance. In order to decrypt and re-encrypt the traffic, the Exinda appliances must have access to the appropriate certificate and public key for each server that clients will communicate with over SSL. Furthermore, the Exinda appliances must be configured to know which servers can receive traffic that is SSL accelerated. These servers are defined by IP address and port, certificate, and other details. Only traffic to servers that are explicitly configured in this way is SSL accelerated. If the server is hosting multiple virtual hosts, when defining the server, you can define an acceleration server for each of the virtual hosts by specifying the SNI virtual host so that the virtual host name is presented during the handshake process with the appliance.

NOTE

If you upload the appropriate certificates and configure SSL Acceleration Servers on the server-side appliance, the appliance will use the Exinda acceleration community feature to push these certificates and server configurations to the other appliances in the community. Configurations that have been pushed to the remote appliances will appear in the Remote SSL Acceleration Servers list on the Optimization > SSL page.

By default, the Exinda appliances are pre-loaded with several root CA certificates. The site-specific certificates will be loaded onto the appliances by the user (or distributed using the community feature). When the client attempts to access the website, during the handshake, the appliance sends to the client all of the certificates in the chain of trust.

NOTE

If you are concerned about any decrypted data on the Exinda appliance, then you can choose to use storage disk encryption.

To configure SSL acceleration

1. Configure SSL certificates and private keys (or configure SSL CA certificates and private keys) to use for SSL acceleration.
2. Configure the servers to use for SSL acceleration.
3. Create Optimizer policies that allow SSL traffic to be accelerated.
4. If you're concerned about decrypted data on the appliance, then enable storage disk encryption.

How to tell if SSL Acceleration is working?

1. Ensure that the SSL Acceleration Server is ok by seeing its status on the SSL server configuration page. There is a green checkmark next to each SSL Acceleration Server that has a good state.
2. On the real-time conversations page, turn on Show Policies and ensure the SSL traffic that you are interested is accelerated. If the traffic shows in a gold band, then it is processed by an accelerated policy. If the traffic has a lock icon, then it is being SSL accelerated.

   If your traffic is not accelerating and you need to trouble-shoot, try the following:

1. Check that the polices seem to be configured properly and that they are in the proper order. You want to ensure that another policy earlier in the tree is not capturing your desired traffic.

2. Check the SSL Acceleration Server details. Ensure you are using correct spelling, etc. More troubleshooting help for disabled SSL Acceleration Servers is offered in the Configure SSL Acceleration Servers section.
3. Check that the Exinda community feature has distributed the certificates and SSL Acceleration Server configuration properly to your appliance. They will appear as "remote"

Managing Certificates and CA Certificates

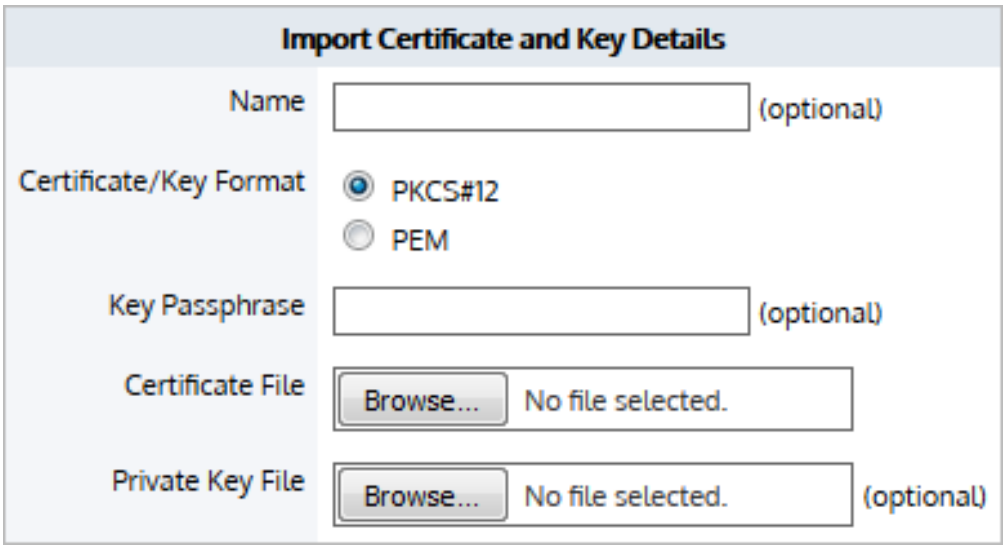
When accelerating encrypted traffic, the Exinda appliance transparently decrypts the traffic, performs the relevant application acceleration techniques, such as TCP Acceleration, WAN Memory, or Edge Cache caching, and then re-encrypts the traffic. The Exinda appliances must have access to the appropriate certificates or certificate authority (CA) certificates, and the public keys to decrypt and re-encrypt the traffic. You can import a certificate or a CA certificate, or generate a self-signed certificate or CA certificate.

On the Certificates tab, you can import normal certificates and you can generate untrusted self-signed certificates. Note that the normal certificates may be trusted Certificate Authority (CA)-signed certificates or self-signed certificates. In the **Certificates and Keys** table, you can see a list of all the (non-CA) certificates available on the appliance. You can show, delete, or export any of these certificates.

On the CA Certificates tab, you an import CA certificates and you can generate untrusted self-signed CA certificates. By importing CA certificates, the appliance can offer the entire chain of trust to clients when performing an SSL handshake. In the **CA Certificates and Keys** table, you can see a list of all the CA certificates available on the appliance. You can show, delete, or export any of these certificates.

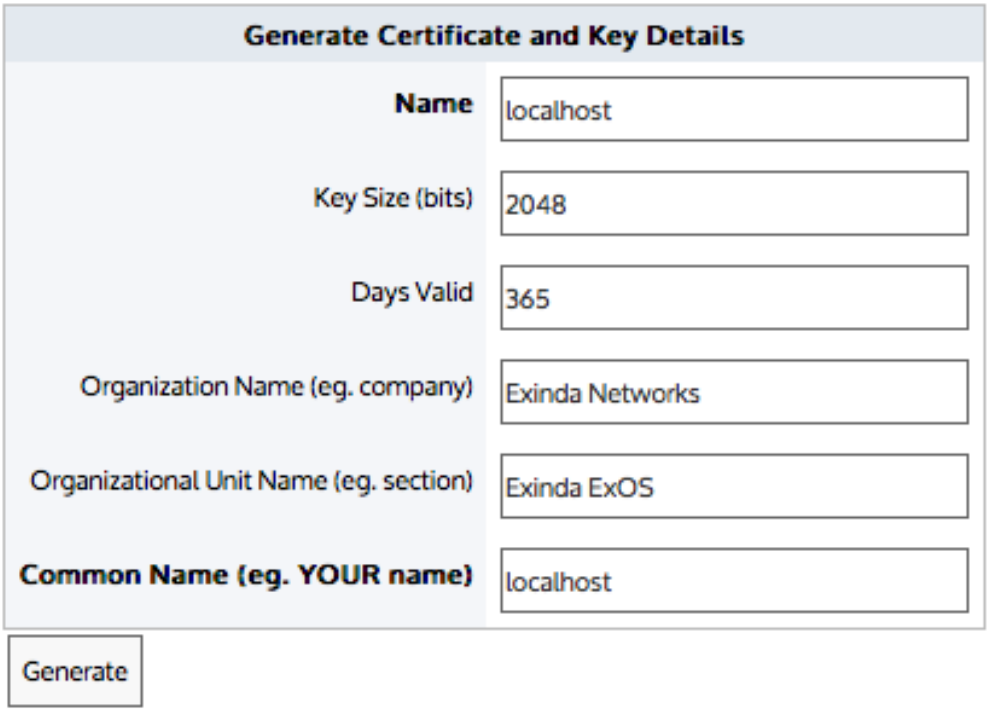
NOTE

Certificates and keys are stored securely on the Exinda appliance. It is not possible to export or view the private key once it has been imported. If you lose the configuration or need to migrate the configuration to another appliance, you must manually load the private key again.



NOTE

The interface for importing both Certificates and CA Certificates is the same.



NOTE

The interface for generating both Certificates and CA Certificates is the same.

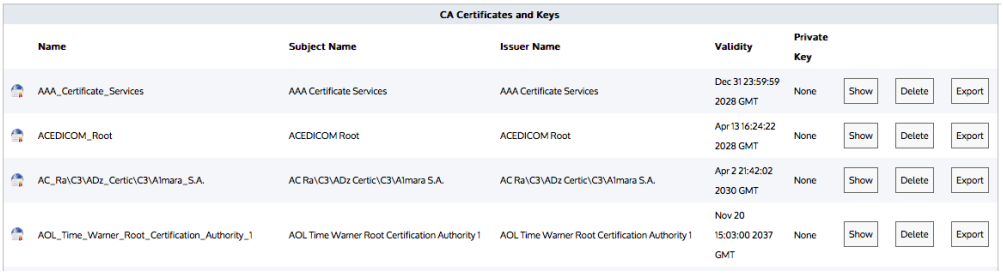


Figure - List of CA Certificates and Keys (the list is similar on the Certificates tab for Certificates and Keys)

Where do I find these settings?

Go to **Configuration > System > Certificates**.

To learn more about SSL and Exinda's SSL Acceleration

Go to How SSL Protocol Acceleration Works.

To import a certificate

In the **Import Certificate and Key Details** section:

1. Select the **Import Certificate** radio button.
2. (Optional) Type a **Name** for the certificate. If no name is specified, the filename of the certificate is used.

   Private keys are stored separately from certificates, and are automatically named the same as the certificate, with '_key' appended to the end.
3. Select the **Certificate/Key Format**.
   - **PKCS#12**—Format used when the certificate and key are stored together, and usually have extensions such as `.pfx` and `.p12`..
   - **PEM**—Common format for certificates issued by Certificate Authorities. PEM certificates usually have extentions such as `.pem`, `.crt`, `.cer`, and `.key`. If PEM format is selected, an additional upload field is exposed so that the private key can be uploaded with the certificate.
4. If the key is password protected, in the **Key Passphrase** field type the password.
5. In the **Certificate File** field, click **Choose File** and navigate to the file to be uploaded to the Exinda appliance.

6. If the PEM format is selected, the private key must be uploaded. In the **Private Key File** field, click **Choose File** and navigate to the private key file.
7. Click **Import**.
   *The certificates are displayed in the **Certificates and Keys** table on the Certificates tab or CA Certificates and Keys table on the CA Certificates tab. From the tables the contents of a certificate can be viewed, or the certificate can be deleted or exported.*

To generate a self-signed certificate

To encrypt SSL traffic that passes through the network without requiring the traffic to be signed, a self-signed certificate needs to be generated.

In the **Import Certificate and Key Details** section:

1. Select the **Generate Certificate** radio button.
2. Type a **Name** for the certificate.
3. In the **Key Size** field, specify the number of bits to use when encrypting the contents of the certificate.
4. Specify how many days the certificate is valid for.
5. Type the name of the organization and the name of the area that will be using this certificate.
6. In the **Common Name** field, type the name of the person issuing the certificate.
7. Click **Generate**.
   After the certificate has been created, it appears in the list of certificates on the Certificates tab.

To display the contents of a certificate

View the contents of an SSL certificate to see the owner of the certificate, information on the issuer of the certificate, and the time period the certificate is valid.

1. In the **(CA) Certificates and Keys** table, locate the certificate in the list, and click **Show**.
2. To return to the list of certificates, click the **Back** button below the table.

To export a certificate

If an SSL certificate is only available on one appliance, export the certificate so it can be imported onto the other Exinda appliances on the network.

1. In the **(CA) Certificates and Keys** table, locate the certificate in the list, and click **Export**.
2. Select the format for the exported certificate.
3. Click **Save**.
   The certificate is downloaded onto the computer accessing the Exinda Web UI.

To delete a certificate

Delete an SSL certificate from the Exinda appliance when it expires, or becomes invalid.

1. In the **(CA) Certificates and Keys** table, locate the certificate in the list, and click **Delete**.
2. In the confirmation dialog, click **OK**.
   The certificate is deleted.

View all certificates and private keys

The All Certificates tab displays a list of all Certificate Authority certificates, self-signed certificates, and all base certificates included on the Exinda appliance. This is the combination of the certificate lists on the Certificates tab and the CA Certificates.

Where do I find these settings?

Go to **Configuration > System > Certificates > All Certificates**.

To display the contents of a certificate

View the contents of an SSL certificate to see the owner of the certificate, information on the issuer of the certificate, and the time period the certificate is valid.

1. In the **All Certificates and Keys** table, locate the certificate in the list, and click **Show**.
2. To return to the list of certificates, click the **Back** button below the table.

To export a certificate

If an SSL certificate is only available on one appliance, export the certificate so it can be imported onto the other Exinda appliances on the network.

1. In the **All Certificates and Keys** table, locate the certificate in the list, and click **Export**.
2. Select the format for the exported certificate.
3. Click **Save**.
   The certificate is downloaded onto the computer accessing the Exinda Web UI.

To delete a certificate

Delete an SSL certificate from the Exinda appliance when it expires, or becomes invalid.

1. In the **All Certificates and Keys** table, locate the certificate in the list, and click **Delete**.
2. In the confirmation dialog, click **OK**.
   The certificate is deleted.

Configure SSL Acceleration Servers

SSL Acceleration provides acceleration of SSL encrypted TCP sessions by intercepting SSL connections to configured servers by decrypting these sessions, performing acceleration techniques, and then re-encrypting them. Only traffic to the configured servers is SSL accelerated. Any SSL traffic that the Exinda appliance sees that does not belong to a configured server is ignored.

By configuring the SSL Acceleration Server, you are specifying:

- The location of the server (IPv4 address and port)
- The SNI (Server Name Indication) which is the hostname of a virtual host when multiple secure web-sites are hosted on a single host where you want each website to use its own certificate
- Which certificate is used to re-encrypt the traffic
- Which certificate is used to authenticate the traffic and what type of validation to perform using that certificate
- If any CA validation is chosen, then you can choose whether to check if that CA certificate is still valid or whether it has been revoked.

NOTE

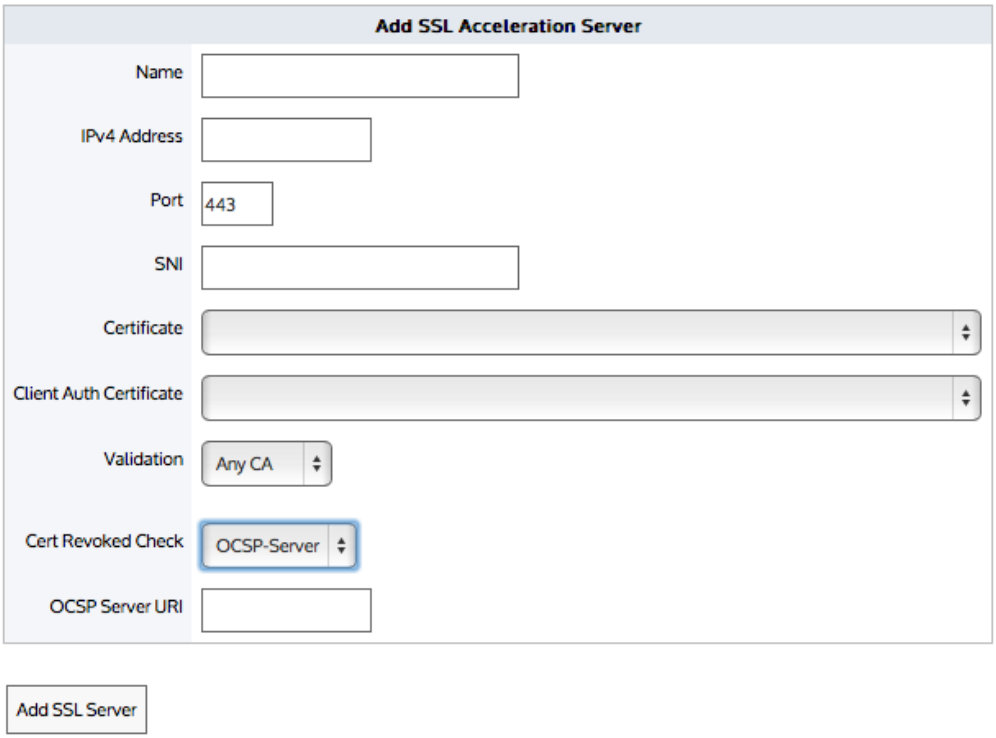
If the revocation check cannot be done or the certificate has been revoked, then the SSL Acceleration Server is disabled. If the OCSP Responder is offline, the server is disabled. The appliance periodically tests the connection and re-enables the server when it is back up. If the OCSP response verification fails or if the certificate has been revoked, then the connection is reset and the server is disabled.

NOTE

If there are any problems with the certificate or key associated with a configured SSL server (e.g., a missing key, or an expired certificate), then SSL Acceleration ignores that traffic until the issue is resolved. The traffic may still be accelerated, just not SSL-accelerated.

NOTE

The SSL Acceleration service uses port 8018 to communicate between Exinda Appliances. Please ensure this port is open for proper functionality



IMPORTANT

Before a server with an SNI extension can be added to the Exinda appliance, the server must be added to the appliance without the SNI extension. A server without an SNI extension can be used as a fallback in event that the client is unable to process the SSL certificate with SNI. A server with the same IP address and port number can be added to the appliance by specifying a unique SNI extension for each server.

IMPORTANT

A server cannot be deleted if another server with the same IP address and port number, and an SNI extension has been configured on the Exinda appliance. Servers with SNI extensions must be deleted before the server can be deleted.

Where do I find these settings?

Go to **Configuration > System > Optimization > SSL**.

To configure an SSL Acceleration server

1. In the **Add SSL Acceleration Server** area, type a name for the server or application you wish to enable for SSL Acceleration.
2. Type the **IPv4 Address** of the server running the SSL enabled application.
3. Type the **Port** number running the SSL enabled application on the server.
4. If the server has multiple SSL certificates with a Server Name Indication (**SNI**) specified, type the SNI extension in the field.

   The server (without an SNI) must be added before the server with the same IP and port number and an SNI can be added.
5. Select the **Certificate** to use for re-encryption of the SSL session.

   The certificates available here are those that are configured in the Certificate and Key page.
6. Select the **Client Auth Certificate** to authenticate sessions on the SSL server.
7. Select the type of validation to apply to the server's certificate.
   - **None**— SSL Acceleration accepts and processes the connection even if the server's SSL certificate is invalid or expired.
   - **Reject**— SSL Acceleration does not processes the connection under any circumstances. The connection is still accelerated, but is not SSL accelerated.
   - **Certificate**— SSL Acceleration accepts and processes the connection only if the server's certificate matches the specific certificate named in the Client Auth Certificate field. Otherwise, the connection is not processed.
   - **Any CA**— SSL Acceleration accepts and processes the connection if the server's certificate matches any CA certificate that is loaded on to the Exinda appliance.
   - **Any**— SSL Acceleration accepts and processes the connection if the server's certificate matches any certificate (CA or non-CA) that is loaded on to the Exinda appliance.
8. If **Certificate** is selected as the **Validation** type, select the certificate to validate against.
9. If **Any CA** or **Any** is selected as the **Validation** type, select the **Cert Revoked Check** type.
   - **None** — No check is performed. The client auth certificate is used regardless of whether the certificate is revoked or not.
   - **OCSP-AIA** — The Online Certificate Status Protocol (OCSP) Authority Information Access (AIA) check is performed. The method uses the location of the authority embedded in the certificate to check for the certificate's revocation status. Note that if the AIA location is not specified in the certificate when this option is chosen, then the certification revoke check will not happen.

- **OCSP-Server** — The Online Certificate Status Protocol (OCSP) check is performed. This method presents an **OCSP Server URI** field where you can type the location of the authority to check for the certificate's revocation status.

10. Click **Add SSL Server**.

    The servers are displayed at the top of the page, where they can be edited or deleted.

To edit an SSL Accelerated server

1. Locate the server in the **SSL Acceleration Servers** list, and click **Edit**.
2. Modify the settings for the server, and click **Apply Changes**.
   The settings for the server are changed.

To delete an SSL Accelerated server

1. Locate the server in the **SSL Acceleration Servers** list, and click **Delete**.

   Servers with SNI extensions must be deleted before the server with the same IP and port number (but without an SNI) can be deleted.

2. In the confirmation dialog, click **OK**.
   The server is deleted.

To troubleshoot a disabled SSL Acceleration Server

If the server is disabled, check the status message in the SSL Acceleration Servers list or Remote SSL Acceleration Servers list. The list will provide feedback on why the server is disabled. Perhaps the certificate validation failed or the OCSP validation failed.

To fix the problem, you can try relaxing the certificate validation a step at a time. For example, turn off OCSP validation and see what happens. Then turn off or broaden the certification validation, such as using ANY, or ANY-CA and see what happens. You can also use the openssl client to check the SSL handshake:

```
openssl s_client -state -msg -connection <ip:port> -ssl3 -showcerts

openssl s_client -connect <ip:host> -tls1 -showcerts -servername <server-name>
```

Create policies to accelerate SSL traffic

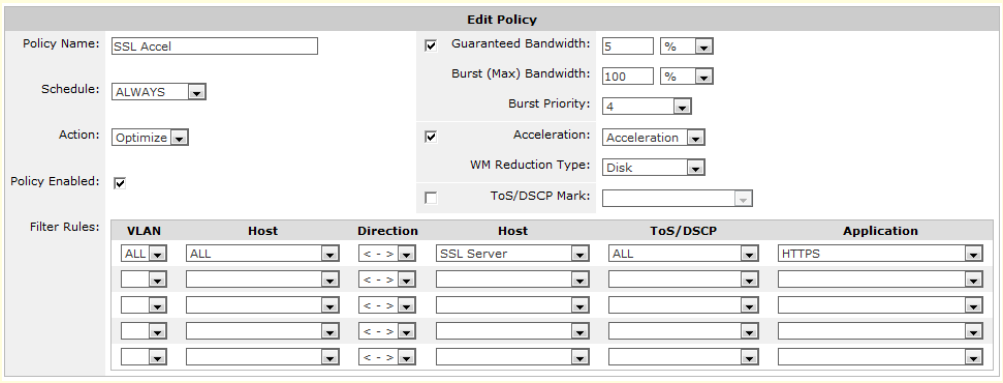
The default policies that are created as a result of running the policy configuration wizard captures SSL traffic in a QoS only policy, meaning no attempt is made to accelerate any SSL traffic by default. To accelerate SSL traffic, you need to create an acceleration policy for the SSL application server you want to accelerate. Any SSL traffic that matches an acceleration policy is passed to SSL Acceleration. If a valid certificate and key are configured for that SSL traffic, then SSL acceleration occurs.

To create a policy for accelerating an SSL application

1. Go to the Optimizer and create a new policy in the appropriate circuit & virtual circuit by clicking **Create new Policy...**
2. Specify the **Action** as **Optimize**.
3. Check the **Acceleration** checkbox.
4. Add a filter for the specified host and the specified SSL application.
5. Add the policy on all Exinda appliances.
6. Once the desired policies are in place on all Exinda appliances, restart the Optimizer.

E X A M P L E: Accelerate an SSL application

This policy needs to be placed above any other policy that generically captures SSL traffic in the policy tree.



Encrypt Disk Storage

SSL acceleration requires the SSL traffic to be decrypted and cached so that various acceleration techniques can be applied to the data. If you are concerned about this, then you can encrypt storage for WAN memory.

If the storage for WAN memory is encrypted, a green checkmark is shown in the Encrypted column.

Disk Storage Map.

Storage Configuration

Service	Status	Free		Size	Minimum	Encrypted	Operation		
cifs	available	39.92G	98%	40.73G	1024.00M	☒	Resize	Format	Encrypt
edge-cache	available	36.11G	85%	42.45G	1024.00M	☒	Resize	Format	Encrypt
monitor	available	40.85G	96%	42.45G	10.00G		Resize	Format	
users	available	974.62M	95%	1024.00M	512.00M		Resize	Format	
wan-memory	available	153.02G	98%	155.65G	5120.00M	☑	Resize	Format	No Encrypt
unallocated storage				0.00					
Total Available Storage:				**284.27G**					

Where do I find these settings?

Go to **Configuration > System > Setup > Storage**.

To encrypt WAN memory storage

Click the Encrypt button for wan-memory.

Appendix A: Ciphers supported in SSL acceleration

SSL Acceleration supports the following ciphers (encryption/decryption algorithms).

Protocol Key	Length	Cipher Name
SSLv3	256 bits	AES256-SHA
SSLv3	128 bits	AES128-SHA
SSLv3	168 bits	DES-CBC3-SHA
SSLv3	128 bits	RC4-SHA
SSLv3	128 bits	RC4-MD5
TLSv1	256 bits	AES256-SHA
TLSv1	128 bits	AES128-SHA
TLSv1	168 bits	DES-CBC3-SHA
TLSv1	128 bits	RC4-SHA
TLSv1	128 bits	RC4-MD5

If the client does not support any of these ciphers, the SSL connection is rejected.

If the server does not support any of these ciphers, it is automatically removed.

Appendix B: Host multiple secure websites on Windows Server 2012

On a corporate network, it may be necessary to have multiple secure websites being served from a single Windows server, on a single IP address. Previously, attempting to host multiple secure sites on a single IP address would cause certificate requests to be perceived as man-in-the-middle attacks, and the connections would be refused. IIS 8.0, available only on Windows Server 2012, introduces the Server Name Indication (SNI) extension which allows a hostname or domain name to be included in SSL certificate requests. With SNI, multiple secure websites can be served from a single IP address as the certificates requests for the sites include the SNI extension, allowing the correct certificate to be presented to the client browser.

To host multiple secure websites on Windows Server 2012, configure the websites to include the SNI extension in the connection requests.

1. Install IIS 8.0 on Windows Server 2012 (page 23)
2. Add sites to the web server (page 23).
3. Ensure the certificates required for the sites are available on the server.

   Depending on how your organization manages SSL certificates, this may involve generating a self-signed certificate or importing a certificate from a Certificate Authority. For instructions managing the certificates on the Windows Server, refer to the Microsoft help.
4. (Optional) If the site requires Server Name Indication (SNI), create a self-signed certificate that identifies the ID of the site. See Create self-signed certificates for each site requiring Server Name Indication (page 24).
5. Identify the certificate to be used by each website (page 24).
6. Export SSL certificates from Windows Server 2012 (page 25)
7. Managing Certificates and CA Certificates (page 10)
8. Configure SSL Acceleration Servers (page 15).

Install IIS 8.0 on Windows Server 2012

IIS 8.0 must be installed on the Windows server before certificates with Server Name Indicators (SNI) can be configured.

1. Open the **Server Manager**.
2. Select **Manage > Add Roles and Features**.
3. Select **Role-based or Feature-based Installation**. Click **Next**.
4. Select the appropriate server and click **Next**.
5. From the list of Server Roles, select **Web Server (IIS)**.
6. In the Add Roles and Features Wizard dialog, click **Add Features**.
7. Click **Next**.
8. Do not select any additional features, and click **Next**.
9. On the Web Server Role (IIS) information screen, click **Next**.
10. Accept the default role services, and click **Next**.
11. Review the selections, and click **Install**.
    When the IIS installation completes, the wizard reflects the installation status.
12. To exit the wizard click **Close**.

Add sites to the web server

Add sites that require SSL certificates with Server Name Indicators (SNI) to the IIS Manager to manage what certificates are used by each site.

1. In the **Server Manager**, and click **IIS**.
2. Right-click the server name, and select **Internet Information Services (IIS) Manager**.
3. Double-click the server name.
4. Right-click **Sites** and select **Add Website**.
5. Add the parameters for the website.
6. In the Binding area, ensure you type the host name of the server.
7. Click **OK**.
8. Repeat these steps for each secure website that will be available on the server.

Create self-signed certificates for each site requiring Server Name Indication

The SelfSSL tool is installed with IIS, and allows you to create self-signed certificates that include the ID of the site within the certificate.

1. In the **Internet Information Services (IIS) Manager**, click Sites and make note of the ID of each website using the self-signed certificate that has Requires Server Name Indication selected.
2. Open a command prompt and navigate to **C:\Program File (x86)\IIS Resources\SelfSSL**.
3. At the prompt type the parameters for the certificate, ensuring you specify the site ID for the site requiring Server Name Indication. For example:

NOTE

In the command, `/v` is the number of days the certificate is valid, `/s` is the ID of the site. Use the values that correspond to your site in the command.

```
selfssl.exe /N:CN=TEST.SITE.3 /K:1024 /V:<days-valid> /S:<site-ID> /P:443
```

The certificate is created.

4. When prompted to replace the SSL settings for the site, type `Y`.
5. Modify the site to use the new certificate in the bindings. See "Identify the certificate to be used by each website" on page 24.

Identify the certificate to be used by each website

Specify the certificate that the secure website uses when receiving requests.

1. In the **Internet Information Services (IIS) Manager**, locate the site created in "Add sites to the web server" on page 23.
2. In the Actions list, select **Bindings**.
3. In the Type list, select **https** and click **Edit**.
4. Type a host name.
5. Select the appropriate SSL certificate.
6. If this site uses the same IP address as another secure site, select **Require Server Name Indication**.

7. To add the binding, click **OK**.
8. Click **Close**.
   The binding is added for the site.
9. Repeat this task for each site configured on the server.

Export SSL certificates from Windows Server 2012

Export the certificates from the Windows server so they can be imported onto the Exinda appliance.

1. In the **Server Manager**, and click **IIS**.
2. Right-click the server name, and select **Internet Information Services (IIS) Manager**.
3. Double-click **Server Certificates**.
4. Right-click the certificate, and select **Export**.
5. Specify the location where the exported certificate should be saved, and type a name for the certificate. Click **Open**.
6. Type and confirm the password required to use the certificate.
7. Click **OK**.
   The certificate is exported to the specified location.

Appendix C: Host multiple secure websites on Apache

On a corporate network, it may be necessary to have multiple secure websites being served from a single Apache server, on a single IP address. Previously, attempting to host multiple secure sites on a single IP address would cause certificate requests to be perceived as man-in-the-middle attacks, and the connections would be refused.

Configure the websites served up from Apache to include the SNI extension in the connection requests.

NOTE

Only Apache 2.2.12 and later and OpenSSL 0.9.8j and later have support for Server Name Indication (SNI).

SNI is not supported on Internet Explorer running on Windows XP.

1. Create all the secured sites on the Apache server.
2. Copy the certificate files for the secure sites onto the Apache server.

   Put the certificate files in the same location as the other certificates on the server. The certificates should be readable by the web server process only.
3. Enable SSL on Apache (page 27)
4. Specify the ports referenced by the virtual hosts (page 27)
5. Add a <VirtualHost> block for each secure site on the server (page 27)
6. Verify the secure server configuration (page 28)
7. Managing Certificates and CA Certificates (page 10)
8. Configure SSL Acceleration Servers (page 15)

Enable SSL on Apache

To use SSL on Apache, the mod_ssl module must be enabled.

1. To enable the mod_ssl module, type the following command:

```
sudo a2enmod ssl
```

Specify the ports referenced by the virtual hosts

A SSL web server must run on a different port than an unencrypted web server. The standard port for HTTPS traffic is 443, but any port number can be used. Apache will not accept incoming connections to any ports if they are not specified with a `Listen <port_number>` directive in the active configuration set.

1. Navigate to **/etc/apache2/conf.d** and open the **ports.conf** file in an editor.
2. Locate the `<IfModule mod_ssl.c>` block.
3. Ensure `Listen 443` is included in the block.
4. Add `NameVirtualHost *:443` to the block.
5. Save the configuration file.

Add a <VirtualHost> block for each secure site on the server

For each domain name or domain subset we want to support SSL for, a VirtualHost block must be declared. This block identifies the domain name to support connections for, and what Certificate or Key files to use for it.

1. Navigate to **/etc/apache2/sites-enabled** and open the folder for the secure site.
2. Open the **<site_name>.conf** file in an editor.
3. Add the <virtualhost> block for the secure server.

   The block will look similar to this:

```
<VirtualHost *:443>

   ServerName "secure2.example.com"

   ServerAdmin webmaster@example.com

   DocumentRoot /home/demo/public_html/secure1.example.com/public
```

```
    ErrorLog /home/demo/public_html/secure2.example.com/log/error.log

    LogLevel warn

    CustomLog /home/demo/public_html/secure2.example.com/log/access.log combined

    <Directory /home/demo/public_html/secure2.example.com/public>

       Options Indexes FollowSymLinks MultiViews

       AllowOverride None

       Order allow,deny

       allow from all

    </Directory>

    SSLEngine On

    SSLCertificateFile /var/www/certs/secure2.pem

    SSLCertificateKeyFile /var/www/keys/secure2.key

</VirtualHost>
```

Update the sample block to reflect the file locations on your Apache server, and ensure each block references the correct secure site.

4. Save the configuration file.

Verify the secure server configuration

It is always best to check your Apache config files for any errors before restarting, because Apache will not start again if your config files have syntax errors.

1. Check the Apache config files for errors, run the following command:

```
sudo apachectl configtest
```

2. Display the secure sites in a browser that supports SNI, and verify no errors are displayed.
3. Restart the Apache server to commit the configuration changes.

```
sudo apachectl stop

sudo apachectl start
```

4. After the server has restarted, run the command `sudo netstat -tnlp | grep 443` and verify that the server is listening on port 443.