DEPLOYMENT GUIDE

SSL Insight Hardening and Best Practices Guide

List of Contents

A10 Networks® SSL Insight® is a complex technology which empowers users to decrypt traffic and enables their existing security infrastructure to inspect encrypted traffic. This guide consists of a set of A10 Networks' recommended best practices that users should follow in order to set up and use the SSL Insight technology in the best and most secure way.

***Note**: This guide is based on ACOS code version 4.1.0-P7. Other code versions are mentioned explicitly wherever applicable.*

AppCentric Templates

AppCentric Templates is a wizard-based configuration tool that enables organizations to apply best practices to their SSL Insight technology solutions, deploying and securing SSLi® with minimal effort.

A10 highly recommends the use of this configuration tool for the deployment and management of SSLi since these templates were developed with a focus on best practices. For that reason, most of the subsequent points can be easily configured via AppCentric Templates.

The AppCentric Templates for SSL Insight technology consist of four main sections:

Dashboard

The dashboard gives users a view of different statistics related to the current state of the system, including system status, CPU and memory usage, connection rate, traffic rate, bypassing statistics and other device information. Custom dashboards can also be created based on user preferences.

Wizard

The wizard provides users with a flow-based initial configuration of SSLi. Subsections include:

- **Topology** selection, where users can select a deployment topology based on their needs. This can either place the SSLi device in Layer 2 or Layer 3 mode.
- **Decryption** configuration, where users can set up the decryption side of SSLi.
- **Re-encryption** configuration, where users can set up the re-encryption side of SSLi.
- **Bypass Configuration** options, which are further divided into category list, domain list or IP list-based bypassing (this part is optional, based on user requirements).
- **Confirmation** section, which lets users confirm their configurations before applying them to the backend.

Configuration

The configuration section provides users with current configuration of the device as well as access to some advanced options. Any configurations applied using the Wizard will be visible here and can be modified based on user requirements. Most of the subsequent best practices can be configured using this section.

Troubleshooting

The troubleshooting section provides users with three different ways of troubleshooting any issues that they might encounter with the functionality of SSLi. These include:

- **System Diagnostics**, for basic system resource status check.
- **End to End**, for verification of functionality of different SSLi components based on traffic flow.
- **Advanced**, for more detailed troubleshooting outputs. This option is suitable for more advanced users with a detailed understanding of the system.

***Note**: For more information on how to use the AppCentric Templates, refer to the SSL Insight Proof of Concept Guide.*

Signing CA and Key

For SSL Insight technology to work, a certificate authority (CA) certificate and private key need to be installed onto the solution so that certificates can be successfully proxied. It is crucial for decryption solutions that these certificates and keys are handled in a safely configured environment, since misuse or mishandling of certificates and keys can compromise the overall security of the network.

A10 recommends that users take the following actions when generating and using these certificates and keys with SSLi:

- Generate a signing CA/key from your company root CA with 2K key and SHA-256. The industry has moved on from using 1K keys and SHA-128 because they can easily be compromised using brute force attacks.
- Renew the signing CA/key periodically.
- Protect the private key with a nontrivial password.

Hardware Security Modules (HSMs) and Certificate Signing Requests (CSR)

The SSL Insight deployment can be made more secure with the use of HSMs, therefore avoiding the tampering with private keys. SSL Insight support multiple Internal HSMs, and also supports integration with External Network HSMs i.e. Thales nShield.

Since keys cannot be exported from an HSM for use, a CSR is required for sending public keys as well as identification information e.g. common name, organization name etc.

The ability to generate self-signed CSRs has been added to the pki create command. The csr-generate options have been removed from the import commands. See the SLB CLI Commands for further information.

TLS Version and Cipher Suites

Since SSL Insight technology works based on a full proxy architecture, it has the power to influence the cipher suite selection procedure. This can help in strengthening the solution, since SSLi can influence the client and server to select newer, better and more secure cipher suites for encryption. This also gives SSLi the flexibility to renegotiate to different cipher suites of similar strength if one is not supported, avoiding network downtime.

A10 recommends that users take the following actions when dealing with cipher suites:

Disable SSLv3

SSLv3 should be disabled from both Inside and Outside SSL Insight instances (partitions/appliances). SSLv3 has a known vulnerability to the POODLE attack and is not secure anymore; therefore, using this can expose the network to serious threats. *Please note that SSLv3 is enabled by default and it must be disabled manually by users*. This can be done as follows:

Using the CLI

The CLI commands and procedures for enabling/disabling SSL/TLS versions are different for Inside and Outside SSL Insight instances. For the Inside SSL Insight instance, go to the Client SSL template settings and disable SSLv3.

```
slb template client-ssl template-name

disable-sslv3
```

Once done, add the Client SSL template to the Virtual Server’s port 443.

For the Outside SSL Insight instance, go to the Server SSL template settings,

```
slb template server-ssl template-name
```

In A10 Networks Advanced Core Operating System (ACOS®), SSLv3 is version 30, TLSv1.0 is 31, TLSv1.1 is 32 and TLSv1.2 is 33. At this point, enter the following command to disable support for SSLv3 by excluding it from the list of SSL/TLS versions supported.

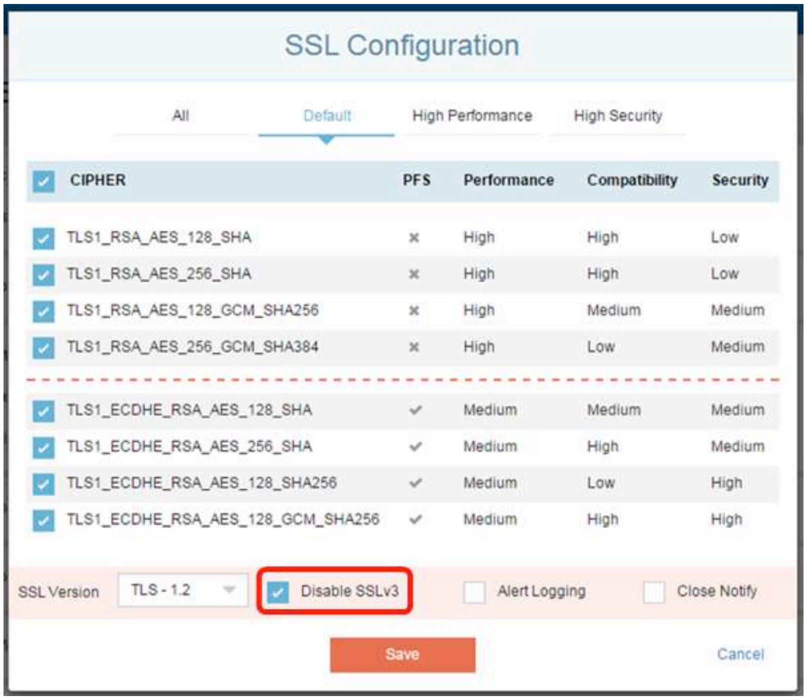
```
version 33 31
```

The above command essentially excludes version 30 from the list of SSL/TLS versions supported (which is SSLv3).

Once done, add the Server SSL template to the Virtual Server's port 8080.

Using the AppCentric Templates

Disabling SSLv3 using the AppCentric Templates is easy. Simply navigate to **AppCentric Templates > SSL Insight > Configuration > Decryption > SSL Configuration** and select the Disable SSLv3 option.



☑ CIPHER	PFS	Performance	Compatibility	Security
☑ TLS1_RSA_AES_128_SHA	×	High	High	Low
☑ TLS1_RSA_AES_256_SHA	×	High	High	Low
☑ TLS1_RSA_AES_128_GCM_SHA256	×	High	Medium	Medium
☑ TLS1_RSA_AES_256_GCM_SHA384	×	High	Low	Medium
☑ TLS1_ECDHE_RSA_AES_128_SHA	✓	Medium	Medium	Medium
☑ TLS1_ECDHE_RSA_AES_256_SHA	✓	Medium	High	Medium
☑ TLS1_ECDHE_RSA_AES_128_SHA256	✓	Medium	Low	High
☑ TLS1_ECDHE_RSA_AES_128_GCM_SHA256	✓	Medium	High	High

SSL Version: TLS - 1.2 ☑ Disable SSLv3 □ Alert Logging □ Close Notify

Figure 1. Disabling SSLv3 support on Inside SSL Insight instance

Limit Cipher Suites

Cipher suites should be limited to the ones below, as recommended by A10, and PFS ciphers should be preferred to non-PFS suites by using higher priority values.

- `TLS1_RSA_AES_128_SHA`
- `TLS1_RSA_AES_256_SHA`
- `TLS1_RSA_AES_128_GCM_SHA256`
- `TLS1_RSA_AES_256_GCM_SHA384`
- `TLS1_ECDHE_RSA_AES_128_SHA priority 10`
- `TLS1_ECDHE_RSA_AES_256_SHA priority 10`
- `TLS1_ECDHE_RSA_AES_128_SHA256 priority 10`
- `TLS1_ECDHE_RSA_AES_128_GCM_SHA256 priority 10`
- `TLS1_ECDHE_ECDSA_AES_128_GCM_SHA256  priority 10`

- `TLS1_ECDHE_ECDSA_AES_128_SHA256 priority 10`
- `TLS1_ECDHE_ECDSA_AES_256_GCM_SHA384 priority 10`
- `TLS1_ECDHE_ECDSA_AES_256_SHA priority 10`

Using CLI

If the configuration is generated using the AppCentric templates, the Cipher Template should exist, as it is created as a default. However, if the complete configuration is generated using the CLI or the ACOS GUI, then the Cipher Template would have to be created. This can be done by using the following commands:

For the Inside SSL Insight instance:

```
slb template cipher template-name
```

At this point, desired cipher suites can be added to this list. Examples can be seen above.

Once the Cipher Template is created, it should be added to the Client SSL template, under port 443 on the Virtual Server.

For the Outside SSL Insight instance:

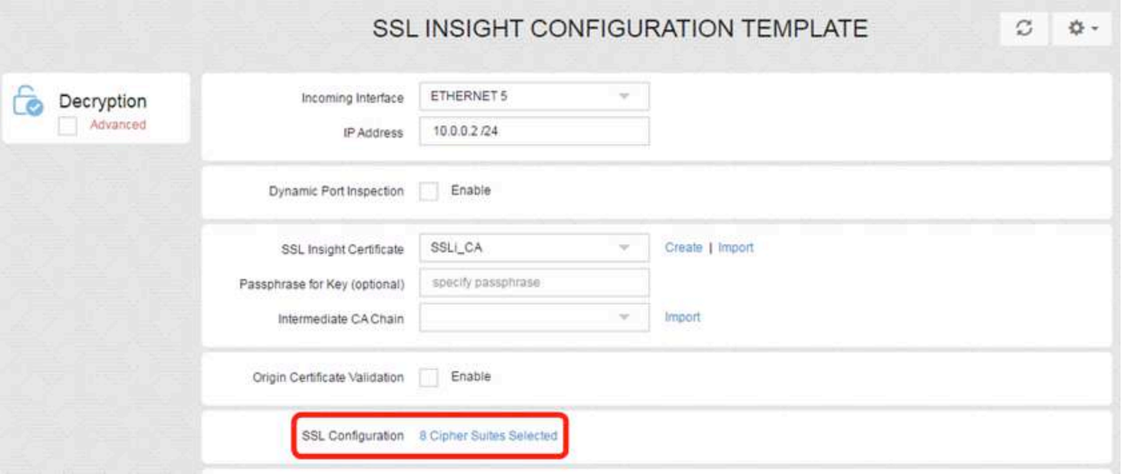
```
slb template cipher template-name
```

At this point, desired cipher suites can be added to this list. Examples can be seen above.

Once the Cipher Template is created, it should be added to the Server SSL template, under port 8080 on the Virtual Server.

Using the AppCentric Templates

To deploy SSL Insight technology will ensure that the best cipher suites are selected. If the user desires to further cut down or expand the cipher suite list, it can be done by navigating to **AppCentric Templates > SSL Insight > Configuration > Decryption > SSL Configuration**, for the Inside SSL Insight instance, and selecting the desired cipher suites from the available list.



SSL INSIGHT CONFIGURATION TEMPLATE

Decryption

□ Advanced

Incoming Interface: ETHERNET 5

IP Address: 10.0.0.2 /24

Dynamic Port Inspection □ Enable

SSL Insight Certificate: SSLi_CA

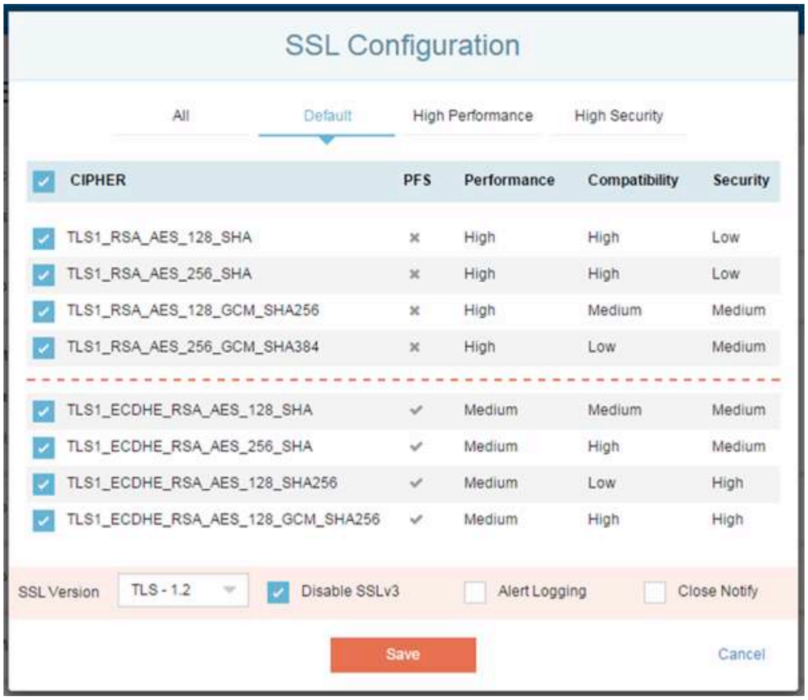
Passphrase for Key (optional): specify passphrase

Intermediate CA Chain:

Origin Certificate Validation □ Enable

SSL Configuration: 8 Cipher Suites Selected

Figure 2. Cipher suite selection for Inside SSL Insight instance (A)



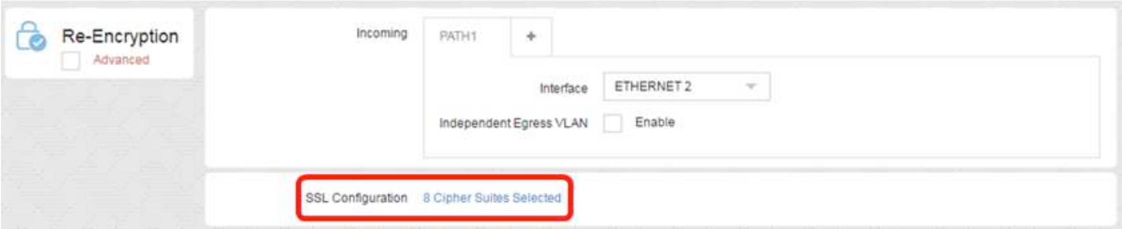
SSL Configuration

☑ CIPHER	PFS	Performance	Compatibility	Security
☑ TLS1_RSA_AES_128_SHA	✗	High	High	Low
☑ TLS1_RSA_AES_256_SHA	✗	High	High	Low
☑ TLS1_RSA_AES_128_GCM_SHA256	✗	High	Medium	Medium
☑ TLS1_RSA_AES_256_GCM_SHA384	✗	High	Low	Medium
☑ TLS1_ECDHE_RSA_AES_128_SHA	✓	Medium	Medium	Medium
☑ TLS1_ECDHE_RSA_AES_256_SHA	✓	Medium	High	Medium
☑ TLS1_ECDHE_RSA_AES_128_SHA256	✓	Medium	Low	High
☑ TLS1_ECDHE_RSA_AES_128_GCM_SHA256	✓	Medium	High	High

SSL Version: TLS - 1.2 ☑ Disable SSLv3 □ Alert Logging □ Close Notify

Figure 3. Cipher suite selection for Inside SSL Insight instance (B)

For the Outside SSL Insight instance, navigate to AppCentric Templates > SSL Insight > Configuration > Re-Encryption > SSL Configuration and select the desired cipher suites from the available list.



Re-Encryption
□ Advanced

Incoming: PATH1 +

Interface: ETHERNET 2

Independent Egress VLAN: □ Enable

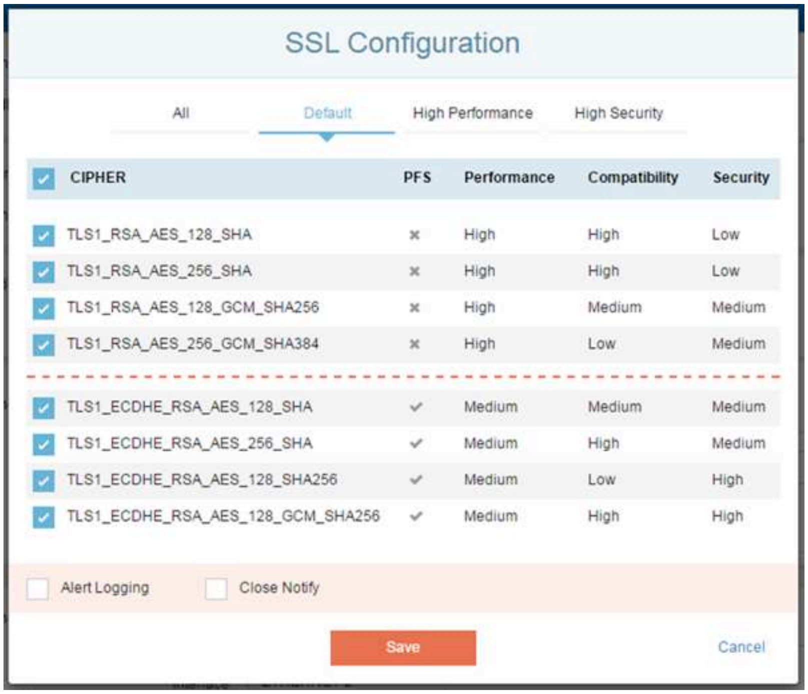
SSL Configuration: 8 Cipher Suites Selected

Figure 4. Cipher suite selection for Outside SSL Insight instance (A)

The default cipher suites are always applied once the configuration is generated using the AppCentric Templates.

However, users can select different cipher suites based on specific requirements.

High Performance cipher suites will provide for faster functionality but at the risk of slightly reduced security. On the other hand, High Security cipher suites are more focused on the security and therefore performance is a secondary consideration. These options help users optimize their network and SSLi solutions based on specific needs.



☑ CIPHER	PFS	Performance	Compatibility	Security
☑ TLS1_RSA_AES_128_SHA	×	High	High	Low
☑ TLS1_RSA_AES_256_SHA	×	High	High	Low
☑ TLS1_RSA_AES_128_GCM_SHA256	×	High	Medium	Medium
☑ TLS1_RSA_AES_256_GCM_SHA384	×	High	Low	Medium
☑ TLS1_ECDHE_RSA_AES_128_SHA	✓	Medium	Medium	Medium
☑ TLS1_ECDHE_RSA_AES_256_SHA	✓	Medium	High	Medium
☑ TLS1_ECDHE_RSA_AES_128_SHA256	✓	Medium	Low	High
☑ TLS1_ECDHE_RSA_AES_128_GCM_SHA256	✓	Medium	High	High

☐ Alert Logging ☐ Close Notify

Figure 5. Cipher suite selection for Outside SSL Insight instance (B)

If performance optimization is "a must," you might prefer RSA key exchange for the Inside SSL Insight instance, and Perfect Forward Secrecy (PFS) for the Outside SSL Insight instance. In the AppCentric Templates, the PFS ciphers are marked with a check mark (✓), while the non-PFS ciphers are marked with a cross mark (✖) in the PFS column as shown in Figure 5.

Origin CA Validation

To validate the origin of a certificate, SSL Insight technology can be configured by using a public CA bundle. This CA bundle should be taken from a trusted source and should be installed in both the Inside and Outside SSL Insight instances. A10 includes a Mozilla CA bundle in ACOS to be used for this functionality.

Currently, origin certificate validation can be configured on the Inside SSL Insight instance using both the CLI and the AppCentric Templates. For the Outside SSL Insight instance, only CLI can be used.

Using the AppCentric Templates

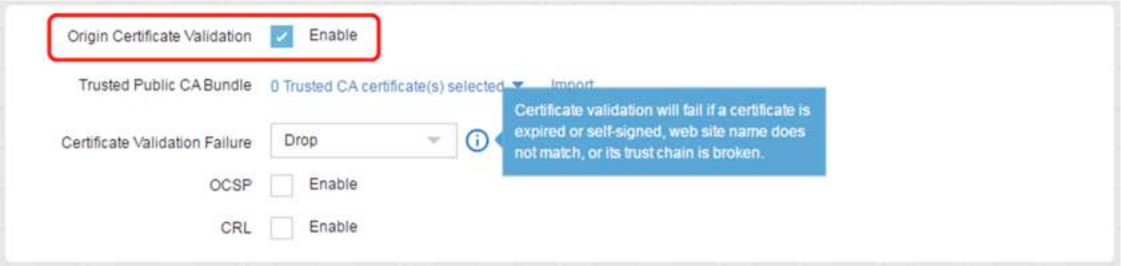
Navigate to AppCentric Templates > SSL Insight > Configuration > Decrypt and check the Origin Certificate Validation option. This opens a sub-menu which can be used to import a Trusted Public CA Bundle.

The Certificate Validation Failure option should be set to the Drop option, which means if a certificate is found to be self-signed or expired, and the website name doesn't match or the trust chain is broken, the connection can be dropped.

If a self-signed certificate needs to be allowed, you have two choices:

- If the websites belong to your organization, you may opt in to bypass inspection using the Domain List based Bypass Configurations. However, the preferred way is to have websites issued a regular certificate from your root CA.
- Otherwise, create an alternate signing key with a CA that is not trusted by browsers and use it to sign origin self-signed certificates; browsers and API clients will be able to see at least a broken certificate chain. To make this work, you have to disable CA validation on the Outside SSL Insight instance.

***Note**: There is no way to set a separate policy for sites belonging to the organization versus external sites with self-signed certs. The rule applies to all websites equally.*



Origin Certificate Validation ☑ Enable

Trusted Public CA Bundle: 0 Trusted CA certificate(s) selected ▾ Import

Certificate Validation Failure: Drop

Certificate validation will fail if a certificate is expired or self-signed, web site name does not match, or its trust chain is broken.

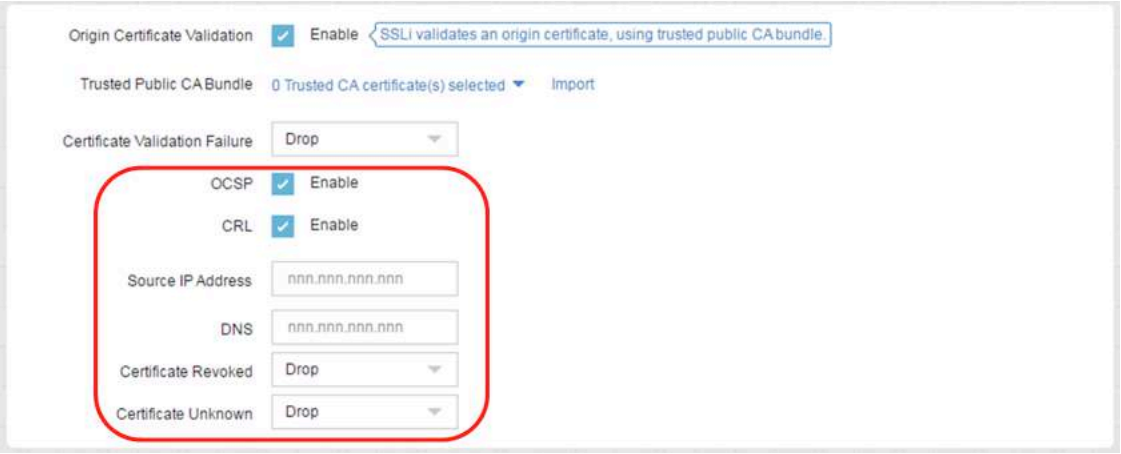
OCSP □ Enable

CRL □ Enable

Figure 6. Enabling origin certificate validation

The above settings are used to set up a basic certification validation check. However, if more advanced control and security is required, then Online Certificate Status Protocol (OCSP) and Certificate Revocation Lists (CRL) validation for certificate revocation checks should also be enabled on the Inside SSL Insight instance.

For both of these options, a subsequent sub-menu is opened which should be filled out according to the user's network settings. These settings include: a Source IP Address, which will be used by SSLi to dynamically make TCP connections with the OCSP resources; a DNS server, which is required to look for the OCSP servers that SSLi will be using; and actions to be taken if a certificate is revoked or its identity is unknown.



Origin Certificate Validation ☑ Enable — SSLi validates an origin certificate, using trusted public CA bundle.

Trusted Public CA Bundle: 0 Trusted CA certificate(s) selected ▾ Import

Certificate Validation Failure: Drop

OCSP ☑ Enable

CRL ☑ Enable

Source IP Address: nnn.nnn.nnn.nnn

DNS: nnn.nnn.nnn.nnn

Certificate Revoked: Drop

Certificate Unknown: Drop

Figure 7. Enabling OCSP and CRL

Using the CLI

To enable certificate validation and configure the Inside SSL Insight instance to drop connections when certificate validation fails, the following CLI command can be used:

```
slb template client-ssl template-name
forward-proxy-verify-cert-fail-action drop
forward-proxy-trusted-ca default_ca_bundle
```

If stronger security options are required to verify a revoke status for a certificate, OCSP and CRL should be configured on the Inside SSL Insight instance.

SSL Inspection Policy

SSL Insight technology can be configured with a number of different inspection policies that determine the way the solution works. These policies can be used to either strengthen the security of the overall solution or improve performance, based on the customer's needs. Support for configuring these policies has been added to the AppCentric Templates, which makes it easy for users to apply these settings and modify the way SSLi interacts with network traffic. However, this document focuses on the security of the solution; therefore, we will look at how each of these policies should be modified and applied, in accordance with A10 Networks' recommended best practices.

Google QUIC Protocol

Enable inspection of QUIC protocol (Google) by blocking UDP 80/443.

Google introduced an experimental protocol called Quick UDP Internet Connection (QUIC), used by Google Chrome web browsers. The protocol primarily uses encrypted UDP, working like SSL/TLS but reducing connection and transport latency. SSLi, by default, cannot decrypt UDP traffic.

In order to inspect QUIC traffic, UDP traffic is blocked using access control lists (ACLs), which forces Chrome to establish connections using TCP. SSLi can then inspect this traffic in the normal way.

Using the CLI

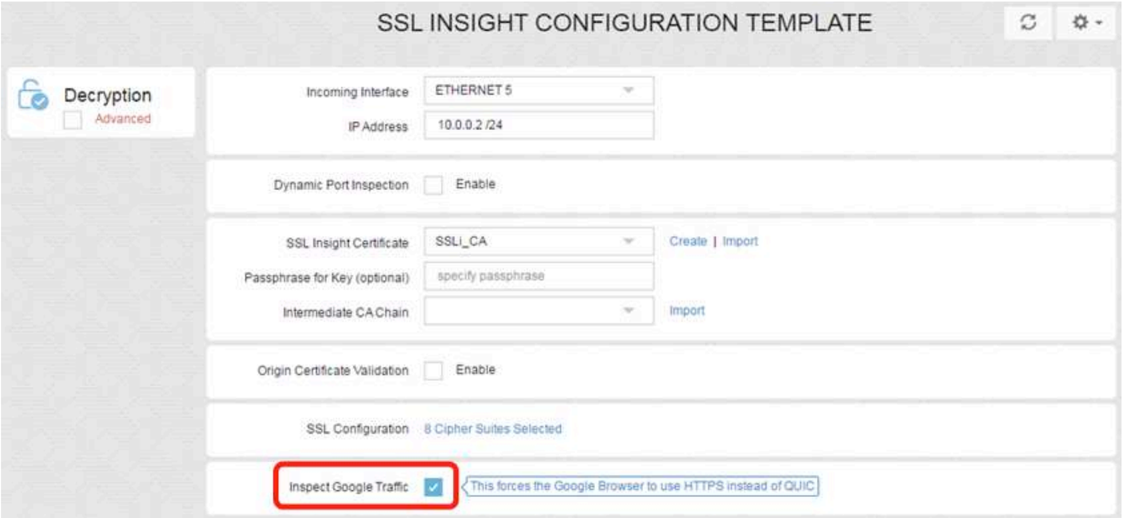
To enable QUIC protocol traffic inspection, the following ACL commands should be applied:

```
access-list ACL-name deny udp any any eq 80
access-list ACL-name deny udp any any eq 443
access-list ACL-name permit ip any any
```

Once created, these ACLs should be applied to the Ingress interface of the SSL Insight setup i.e. the client facing interface of SSL Insight that will receive encrypted traffic from the client.

Using the AppCentric Templates

Enabling inspection for Google's QUIC protocol using the AppCentric Templates is easy. Simply navigate to AppCentric Templates > SSL Insight > Configuration and select the Inspect Google Traffic option.



SSL INSIGHT CONFIGURATION TEMPLATE

Decryption
□ Advanced

Incoming Interface: ETHERNET 5
IP Address: 10.0.0.2 /24

Dynamic Port Inspection □ Enable

SSL Insight Certificate: SSLi_CA Create | Import
Passphrase for Key (optional): specify passphrase
Intermediate CA Chain: Import

Origin Certificate Validation □ Enable

SSL Configuration: 8 Cipher Suites Selected

Inspect Google Traffic ☑ This forces the Google Browser to use HTTPS instead of QUIC

Figure 8. Enabling inspection of Google QUIC traffic

Certificates Not in Cache

Drop a connection if *a certificate is not in cache* instead of bypassing inspection. This can be seen as a strict added security feature that may be used if the user does not trust any new websites being visited. The functionality can currently be configured using the CLI (AppCentric Templates support is not available at the moment).

Using the CLI

To drop a connection using a certificate that doesn't exist in the cache, the Client SSL template will be used.

```
slb template client-ssl template-name

forward-proxy-cert-not-ready-action reset
```

Once the above commands are entered, the Client SSL template should be applied to port 443 of the Virtual Server on the Inside SSL Insight instance.

***Note**: This feature is available in ACOS versions 4.1.0-P8 onwards.*

Disable SSLi Fail-Safe

SSLi has a functionality to bypass traffic when the origin certificate fetch fails. This causes the traffic in such cases to go through, encrypted and uninspected. For added security, users should *disable fail-safe* to prevent traffic from being bypassed when origin certificate fetches fail.

Using the CLI

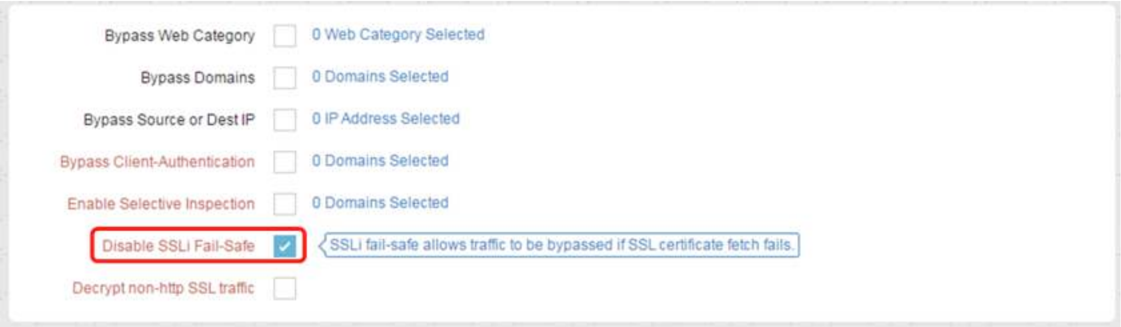
The following commands on the CLI can be used to disable the SSLi Fail-Safe option:

```
slb template client-ssl template-name

forward-proxy-failsafe-disable
```

Once the above changes have been made to the Client SSL template, the template should be added to port 443 of the Virtual Server on the Inside SSL Insight instance.

Using the AppCentric Templates

Navigate to **AppCentric Templates > SSL Insight > Configuration > Decryption > Advanced** and check the Disable SSLi Fail-Safe option.



Bypass Web Category □ 0 Web Category Selected

Bypass Domains □ 0 Domains Selected

Bypass Source or Dest IP □ 0 IP Address Selected

Bypass Client-Authentication □ 0 Domains Selected

Enable Selective Inspection □ 0 Domains Selected

Disable SSLi Fail-Safe ☑ SSLi fail-safe allows traffic to be bypassed if SSL certificate fetch fails.

Decrypt non-http SSL traffic □

Figure 9. Disabling SSLi Fail-Safe option

Non-HTTP SSL Traffic

Enable decryption of non-HTTP SSL traffic to make sure that users can inspect encrypted traffic that may use another application on port 443. By default, the SSLi device will drop non-HTTP requests that are sent on an HTTPS port.

Using the CLI

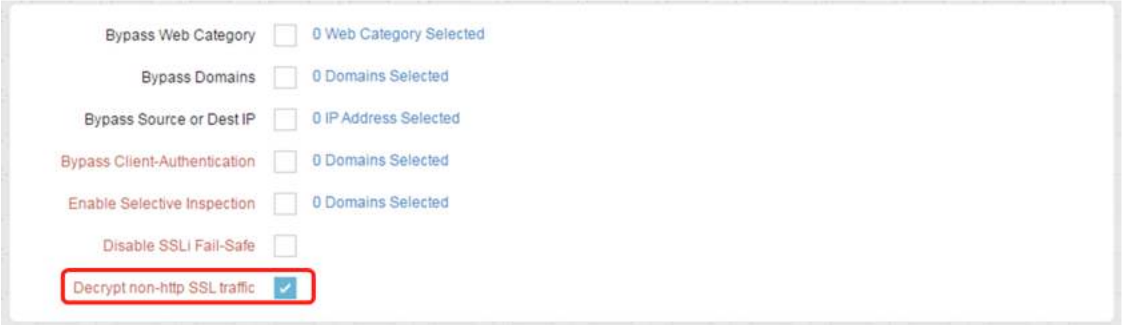
For this option to work, a user needs to add an HTTP template to the configuration on both the Inside and Outside SSL Insight instances. These templates will forward all non-HTTP traffic to the service group that is responsible for decrypting the traffic, i.e., the service group with port 8080 for the Inside and port 443 for the Outside SSL Insight instance.

```
slb template http template-name

non-http-bypass service-group service-group-with-port-8080
```

After creating the template, it should be added to the Virtual Server, under vport 8080 for the Inside and vport 443 for the Outside SSL Insight instances.

Using the AppCentric Templates

Navigate to AppCentric Templates > SSL Insight > Configuration > Decryption > Advanced and check the Decrypt non-http SSL traffic option.



Bypass Web Category □ 0 Web Category Selected
Bypass Domains □ 0 Domains Selected
Bypass Source or Dest IP □ 0 IP Address Selected
Bypass Client-Authentication □ 0 Domains Selected
Enable Selective Inspection □ 0 Domains Selected
Disable SSLi Fail-Safe □
Decrypt non-http SSL traffic ☑

Figure 10. Decrypting non-HTTP SSL traffic

Domain List Based Bypassing

If Domain List based bypassing is used, users should use the “exact” or “ends with” option to match to an apex domain name. This makes sure that users do not accidentally bypass any malicious websites that might otherwise go through if “contains” or “starts with” matching is used.

The easiest way of using these options is with the use of AppCentric Templates. If the configuration is being generated using the Wizard, these settings can be accessed by navigating to AppCentric Templates > SSL Insight > Wizard > Bypass Configuration > Bypass Domain List.

If an existing configuration is being modified, navigate to AppCentric Templates > SSL Insight > Configuration > Decryption > Bypass Domains.

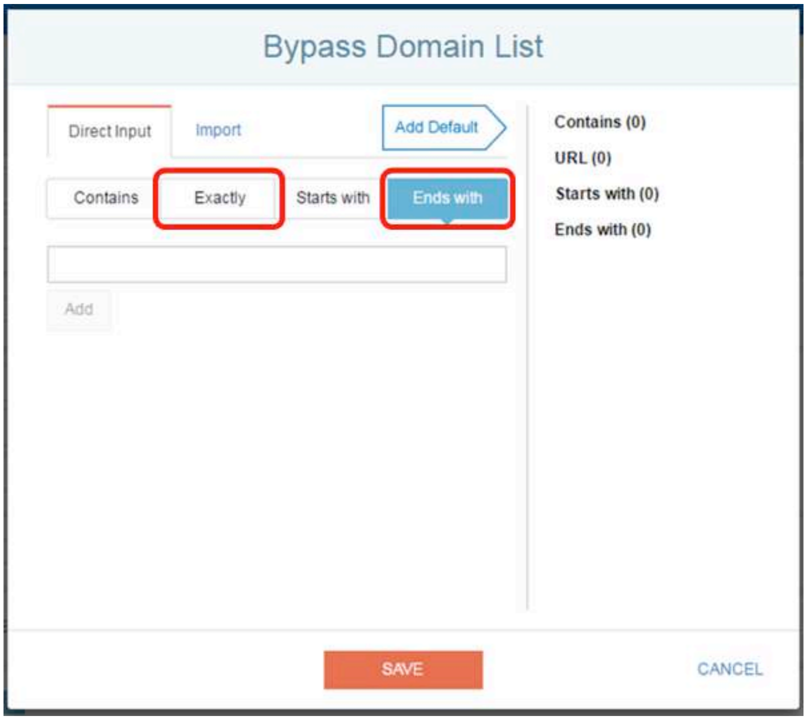


Figure 11. Domain List-based bypassing

Certificate Pinned Websites

There are certain desktop or mobile applications that use hard certificate pinning. This makes it impossible for decryption solutions like SSLi to decrypt such application traffic.

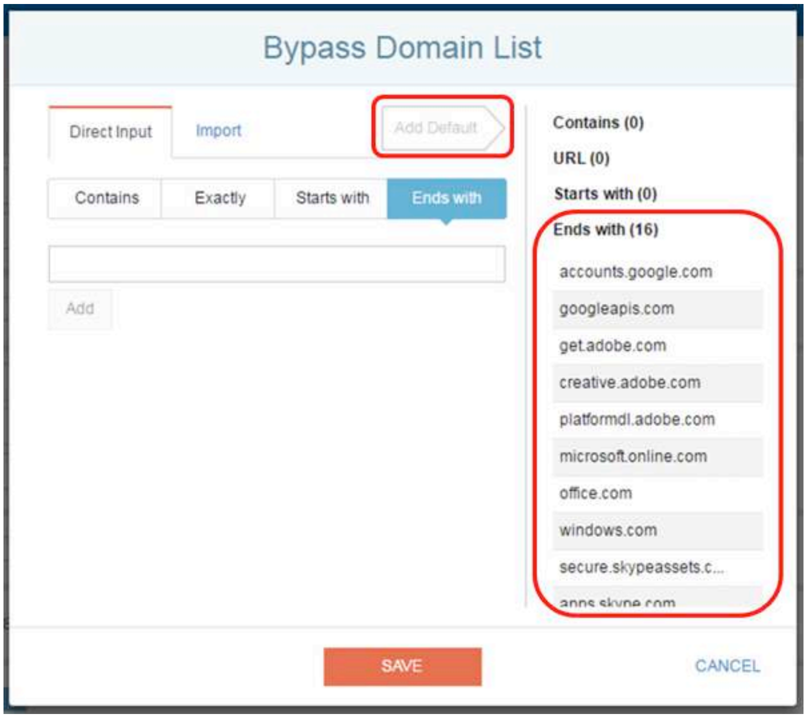
However, one important thing to note here is that decryption should be bypassed only if those applications are required for your business operation, and it has been positively confirmed that their certificates are pinned. In other words, it is a purely administrative decision that organizations have to think carefully about before deciding to bypass decryption.

To make configuration of this functionality easy, the A10 Networks research team has come up with a list of known websites and web applications that use hard certificate pinning. Users can bypass traffic to such websites using the AppCentric Templates' Domain List-based bypassing.

Using the AppCentric Templates

If the configuration is being generated using the Wizard, navigate to AppCentric Templates > SSL Insight > Wizard > Bypass Configuration > Bypass Domain List and click the Add Default button.

If an existing configuration is being modified, navigate to AppCentric Templates > SSL Insight > Configuration> Decryption > Bypass Domains and click the Add Default button.



Bypass Domain List

Direct Input | Import | Add Default

Contains | Exactly | Starts with | Ends with

Add

Contains (0)

URL (0)

Starts with (0)

Ends with (16)

- accounts.google.com
- googleapis.com
- get.adobe.com
- creative.adobe.com
- platformdl.adobe.com
- microsoft.online.com
- office.com
- windows.com
- secure.skypeassets.c...
- apps.skype.com

SAVE | CANCEL

Figure 12. Bypassing certificate pinning websites and web applications

Other Recommendations

All of the above recommended best practices are related to increasing the security of SSLi. However, this section is focused on general recommendations regarding the deployment and operations of SSLi.

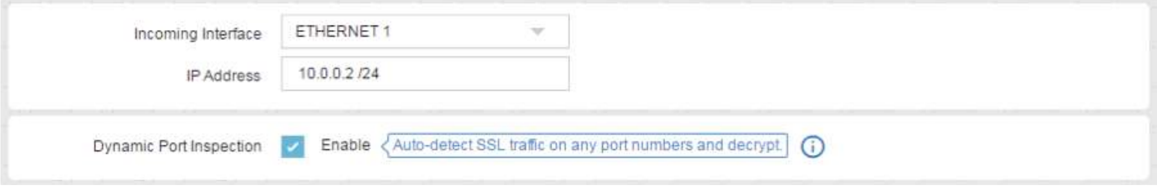
Dynamic Port Inspection

It is highly recommended that with static port SSLi (using TCP port 443), the dynamic port inspection is also configured. Dynamic port inspection makes sure that all encrypted traffic, regardless of the TCP port used, is decrypted and inspected by security devices. This feature uses the DSCP field in the IP header to mark if traffic is decrypted or not. It is important to ensure that the security devices connected to SSLi leave DSCP intact on egress.

Dynamic port inspection can be configured using both the CLI and the AppCentric Templates. The detailed steps for the CLI configuration can be found in the **ACOS SSL Insight Configuration Guide**.

Using the AppCentric Templates

To configure Dynamic Port Inspection using the AppCentric Templates, navigate to **AppCentric Templates > SSL Insight > Configuration > Decryption** and check the Dynamic Port Inspection option. This enables SSLi to start decrypting traffic that uses TCP port numbers other than 443.



Incoming Interface	ETHERNET 1
IP Address	10.0.0.2 /24
Dynamic Port Inspection	☑ Enable Auto-detect SSL traffic on any port numbers and decrypt.

Figure 13 Dynamic Port Inspection

Wildcard vPorts

Wildcard vPorts can be used to streamline the flow of traffic through SSLi. It also allows for better statistics to be collected from the system. By default, the AppCentric Templates generate a total of four vPorts for the Inside and Outside partitions. These are as follows:

Inside	**Outside**
0 TCP	0 TCP
0 UDP	0 UDP
0 Others	0 Others
8080 TCP	443 TCP

A configuration like the one shown above can let SSLi forward non-encrypted traffic, or encrypted traffic that should be bypassed, to 0 TCP, 0 UDP and 0 Others ports while only the traffic that needs to be decrypted/re-encrypted can be forwarded ports 8080 TCP and 443 TCP.

Once these ports are created, they should be added to the wildcard virtual server on the relevant partition.

Management Interface for URL Classification Updates

The A10 URL classification service is powered by Webroot. The service requires internet connectivity to download the updates on a regular basis. These updates include URL classification databases which are then used by SSLi for configuring URL based bypassing and filtering. It is important that the URL classification service has uninterrupted internet connectivity so that it can access the updates cloud regularly. A10 recommends the use of the management interface on an SSLi device to be used for this purpose since it is isolated from the data plane and is safe from issues on data interfaces.

Using the CLI

Once the license for the URL classification is installed and the service is running, the following commands should be entered so that the service uses the management interface for downloading updates:

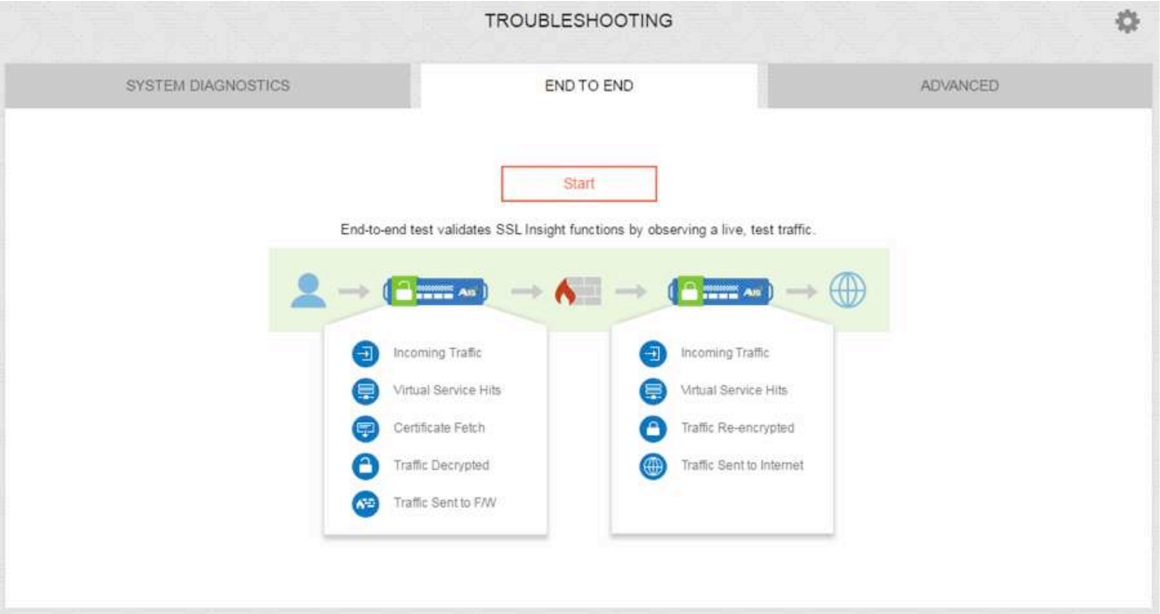
```
web-category

use-mgmt-port

enable
```

Troubleshooting SSLi Operations

A10 recommends the use of a simple HTTPS website to troubleshoot SSLi operations once the device has been configured. A10 has set up a simple test website https://sslitest.com that can be used for this purpose. Other examples may include https://httpbin.org

Using the AppCentric Templates

The AppCentric Templates provide a simple troubleshooting tool for this purpose in the Troubleshooting section. This tool uses the website https://sslitest.com for testing the SSLi operations. To use this tool, navigate to **AppCentric Templates > SSL Insight > Troubleshooting > End to End** and click the **Start** button. When prompted to access the website, do so using your client machine. Once done, click the **Done** button and you will be shown your system status.



End-to-end test validates SSL Insight functions by observing a live, test traffic.

Figure 14 End to End Troubleshooting

Summary

A10 Networks SSL Insight provides high performance visibility into SSL encrypted traffic. It empowers users to decrypt traffic and also enables their existing security infrastructure to inspect encrypted traffic. This guide offers a set of A10 Networks recommended best practices that users can deploy in order to set up and use SSL Insight technology in the most secure and optimal way.

About A10 Networks

A10 Networks is a leader in application networking, providing a range of high-performance application networking solutions that help organizations ensure that their data center applications and networks remain highly available, accelerated and secure. Founded in 2004, A10 Networks is based in San Jose, California, and serves customers globally with offices worldwide. For more information, visit:
www.a10networks.com

Corporate Headquarters

A10 Networks, Inc
3 West Plumeria Ave.
San Jose, CA 95134 USA
Tel: +1 408 325-8668
Fax: +1 408 325-8666
www.a10networks.com

Part Number: A10-DG-16164-EN-01
JUN 2017

Worldwide Offices

North America
sales@a10networks.com
Europe
emea_sales@a10networks.com
South America
latam_sales@a10networks.com
Japan
jinfo@a10networks.com
China
china_sales@a10networks.com

Hong Kong
hongkong@a10networks.com
Taiwan
taiwan@a10networks.com
Korea
korea@a10networks.com
South Asia
southasia@a10networks.com
Australia/New Zealand
anz_sales@a10networks.com

To discover how A10 Networks products will enhance, accelerate and secure your business, contact us at a10networks.com/contact or call to speak with an A10 sales representative.