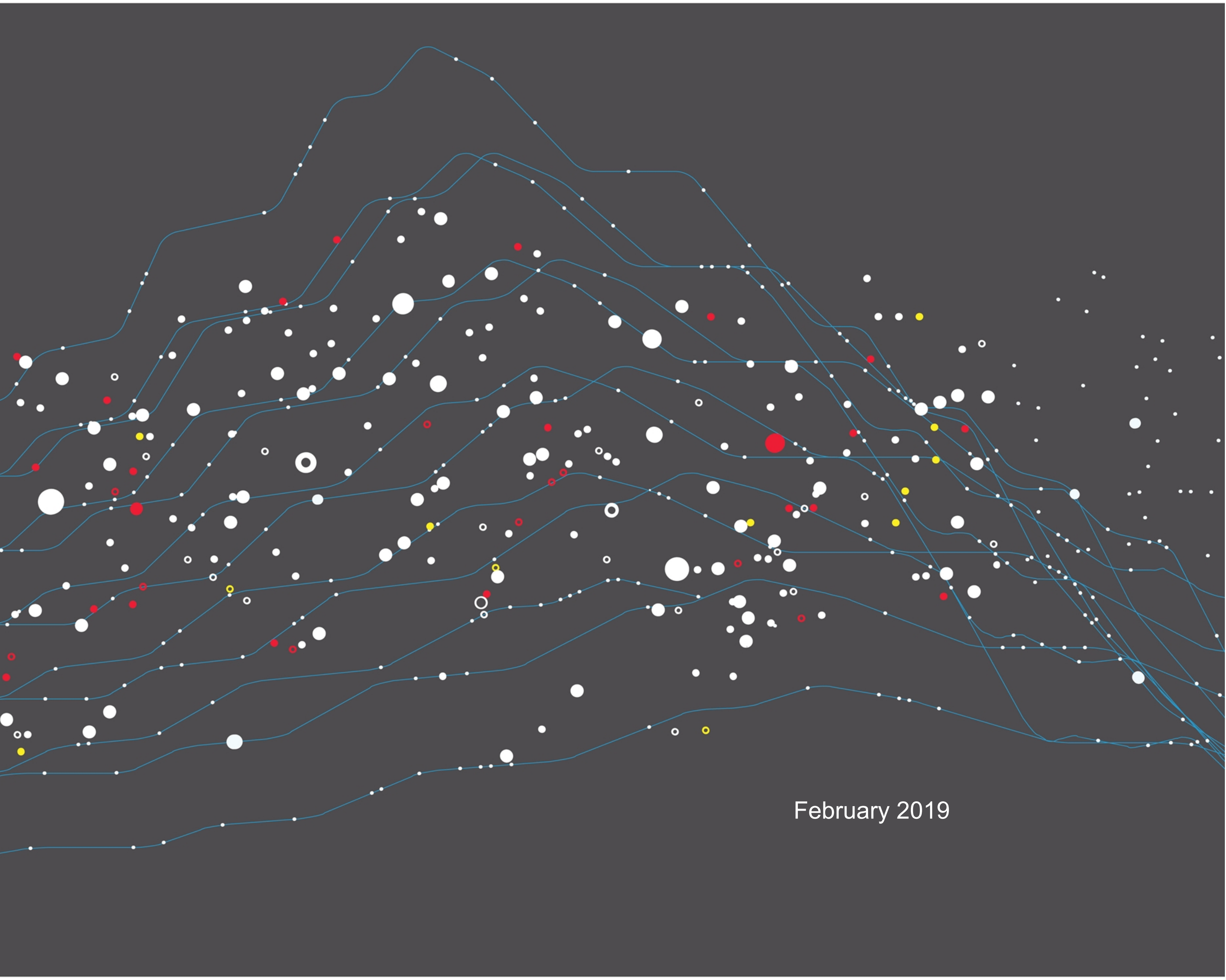
cisco
Preferred
Solution
Partner

RECOMMENDED PRACTICES GUIDE

F5 SSL Orchestrator and Cisco WSA

SSL Visibility with Serving Chaining for Advanced Threat Protection



February 2019

Table of Contents

Introduction

The Secure Sockets Layer (SSL) protocol and its successor, Transport Layer Security (TLS), have been widely adopted by organizations to secure IP communications, and their use is growing rapidly. While SSL provides data privacy and secure communications, it also creates challenges to devices in the security stack intended to inspect the encrypted traffic. In short, the encrypted communications cannot be seen as clear text and are passed through without inspection, becoming security blind spots. This creates serious risks for businesses: What if attackers are hiding malware inside the encrypted traffic?

Performing decryption of SSL/TLS traffic on security inspection devices with native decryption support is one answer, but this can tremendously degrade the performance of those devices. This performance concern becomes even more challenging given the demands of stronger, 2048-bit certificates.

An integrated F5 and Cisco solution solves these related SSL/TLS challenges. F5® SSL Orchestrator™ centralizes SSL inspection across complex security architectures, enabling flexible deployment options for decrypting and re-encrypting user traffic. It also provides intelligent traffic orchestration using dynamic service chaining and policy-based management. The decrypted traffic is then inspected by one or more Cisco Web Security Appliances (WSAs), which can prevent previously hidden threats and block exploits. This solution eliminates the blind spots introduced by SSL and closes any opportunity for adversaries. **SSL Orchestrator, with its ability to address HTTP proxy devices inside its decrypted inspection zone, allows the Cisco WSA to provide optimal security functionality while offloading SSL and complex orchestration to the F5 system**.

This guide provides an overview of the joint F5 and Cisco solution and describes different deployment modes, including service chain architectures and recommended practices.

The Integrated F5 and Cisco Solution

The F5 and Cisco integrated solution enables organizations to intelligently manage SSL while providing visibility into a key threat vector that attackers often use to exploit vulnerabilities, establish command and control channels, and steal data. Without SSL visibility, it is impossible to identify and prevent such threats at scale.

F5 SSL Orchestrator provides:

- **Multi-layered security**. To solve specific security challenges, security administrators are accustomed to manually chaining together multiple point products, creating a bare-bones “security stack” consisting of multiple services. A typical stack may include components like data leak prevention (DLP) scanners, web application firewalls (WAFs), intrusion prevention and/or detection systems (IPS and IDS), malware analysis tools, and more. In this model, all user sessions are provided the same level of security, as this daisy chain of services is hard-wired.
- **Dynamic service chaining**. Dynamic service chaining effectively breaks the daisy chain paradigm by processing specific connections, based on context provided by the security policy, that then allow specific types of traffic to flow through arbitrary chains of services. These service chains can include five types of services: layer 2 inline services, layer 3 inline services, receive-only services, ICAP services, and HTTP web proxy services.

- **Topologies**. Different environments call for different network implementations. While some can easily support SSL visibility at layer 3 (routed), others may require these devices to be inserted at layer 2. SSL Orchestrator can support all of these networking requirements with the following topology options:
  - Outbound transparent proxy
  - Outbound explicit proxy
  - Outbound layer 2
  - Inbound reverse proxy
  - Existing application
  - Inbound layer 2
- **Security policy**. The SSL Orchestrator security policy provides a rich set of context-aware methods to dynamically determine how best to optimize traffic flow through the security stack. Context can minimally come from a variety of source, destination, and traffic data.

The Cisco Web Security Appliance provides:

- **Web reputation and categorization** that is regularly updated by TALOS, one of the largest commercial threat research groups in the world, to assess web content and risk.
- **Anti-malware protections** that can identify known malicious files as well as analyze unknown files for hidden threats.
- **Automated monitoring and analysis** to quickly determine the scope of damage, remediate it, and bring operations back to normal.
- **Application visibility** that grants full control over web applications such as those included in the Microsoft Office 365 suite.
- **Granular policy options** that can block individual web objects and file types.
- **Automated traffic analysis** to scan all web traffic in real time for both known and new malware, using dynamic reputation and behavior-based analysis on all web content.

SSL visibility: How do we do it?

F5's industry-leading full-proxy architecture enables F5 SSL Orchestrator to install a decryption/clear-text zone between the client and web server, creating an aggregation (and, conversely, disaggregation) visibility point for security services. The F5 system establishes two independent SSL connections—one with the client and the other with the server. When a client initiates a TLS connection to the server, the F5 system intercepts and decrypts the client-encrypted traffic and steers it to a pool of security devices for inspection before re-encrypting the same traffic to the server. The returned response from the server to the client is likewise intercepted and decrypted for inspection before being sent on to the client.

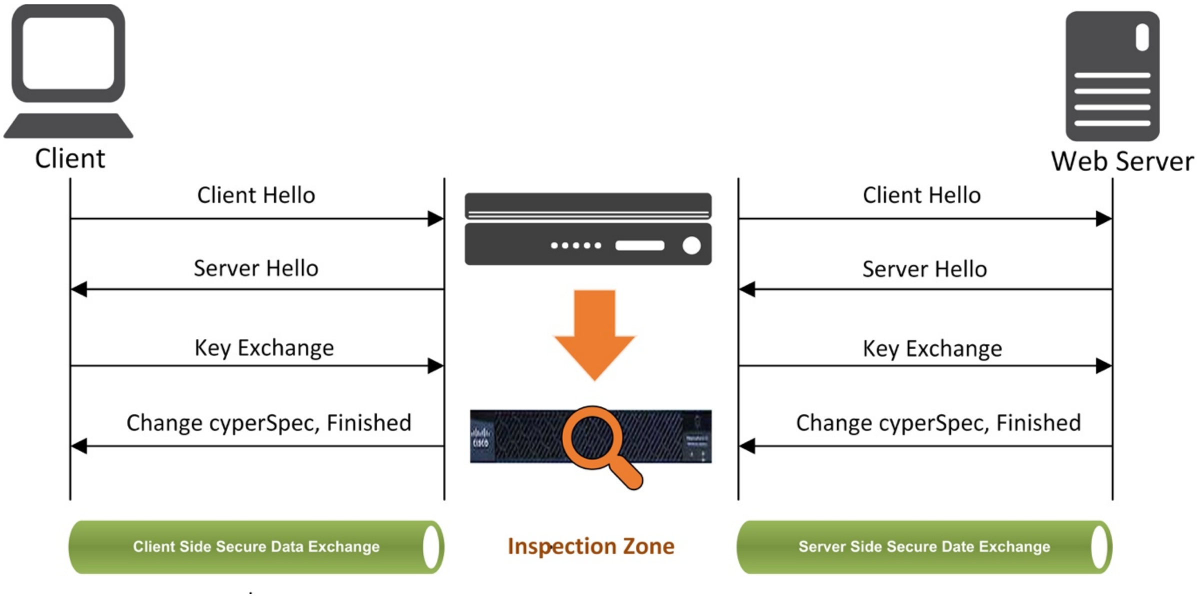

Client
Web Server
Client Hello
Server Hello
Key Exchange
Change cyperSpec, Finished
Client Hello
Server Hello
Key Exchange
Change cyperSpec, Finished
Client Side Secure Data Exchange
Inspection Zone
Server Side Secure Date Exchange

Figure 1: SSL Orchestrator creates a decryption/clear-text zone between the client and web server.

SSL orchestration using security service chains

As shown in Figure 1, SSL Orchestrator can load balance, monitor, and dynamically chain security services, including next-gen firewalls, DLP, IDS/IPS, web application firewalls, and antivirus/malware, by matching the user-defined policies to determine whether to bypass or decrypt and whether to send to one set of security services or another. This policy-based traffic steering capability allows for better utilization of the existing security services investment and helps to reduce administrative costs.

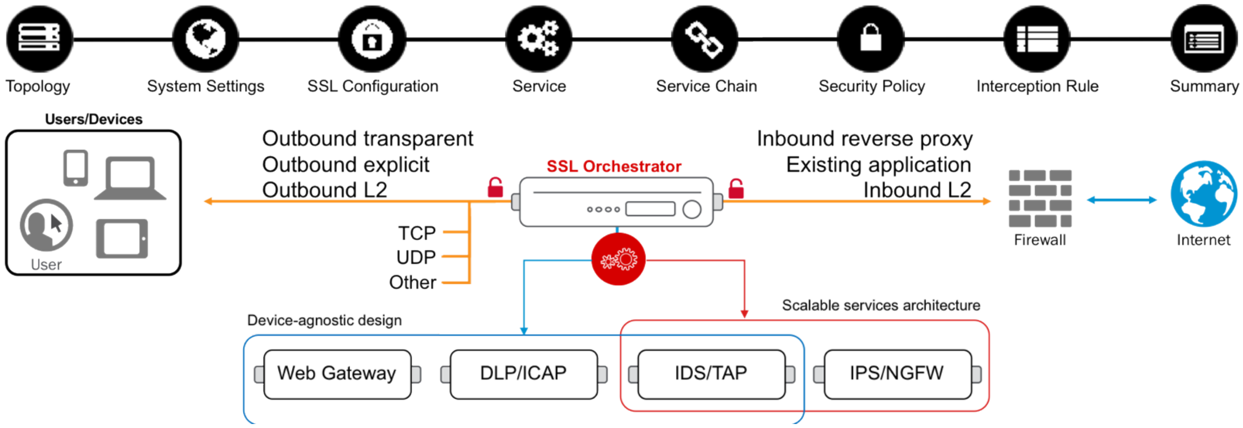


Figure 2: SSL Orchestrator enables dynamic service chaining based on user-defined policies.

SSL Orchestrator enables administrators to apply different service chains based on context derived from a powerful classification engine. That context can come from:

- Source IP/subnet
- Destination IP/subnet
- IP intelligence category

- IP geolocation
- Host and domain name
- URL filtering category
- Destination port
- Protocol

Deployment Planning

Careful advance consideration of deployment options can ensure an efficient and effective implementation of the integrated solution using F5 SSL Orchestrator and the Cisco WSA security system.

Sizing

The main advantage of deploying SSL Orchestrator in the corporate security architecture is that the wire traffic now can be classified as “interesting” traffic, which needs to be decrypted by SSL Orchestrator for inspection by Cisco WSA, and “uninteresting” traffic, which is allowed to pass through or be processed differently according to other corporate policy requirements. This selective steering of only the interesting traffic to the firewall system conserves its valuable resources (as it need not inspect the entire wire traffic), maximizing performance.

As a result, it is important to consider the entire wire traffic volume to calculate the appropriate F5 device size. The Cisco WSA system will require two interfaces on the F5 systems (or one 802.1q VLAN tagged interface) to allow traffic flow through logical inbound and outbound service interfaces.

Refer to the SSL Orchestrator Datasheet and consider the following factors when sizing the F5 system for the integrated solution:

- Port density
- SSL bulk encryption throughput
- System resources
- The number of security services and devices in service chain

Note: F5 SSL Orchestrator has no specific port density requirement. Layer 3 devices must be layer 3 adjacent (routable); layer 2 devices must be layer 2 adjacent (switched); and the F5 device supports 802.1q VLAN tagging so a single interface can be logically divided into multiple VLANs. Security devices can connect to SSL Orchestrator across a switched or routed architecture, so port density is expandable. The only significant requirement is that inline security devices (layer 2, layer 3, and HTTP devices) must have separate physical or logical inbound and outbound interfaces.

License components

The SSL Orchestrator solution supports two licensing modes: standalone and as an F5® BIG-IP® Local Traffic Manager™ (LTM) add-on.

Standalone software license mode

This option supports the following F5 platforms:

- i2800
- i5800
- i10800
- i11800
- i15800
- VE High Performance (HP – 8vCPU)
- VE High Performance (HP – 16vCPU)

The standalone option is suited for environments that need standalone security solutions and have no need to integrate with other F5 software functions. Standalone mode restricts the F5 platform to the following additional F5 software modules:

- **BIG-IP® Access Policy Manager® (APM)** for user authentication
- **F5® Secure Web Gateway (SWG) or URL Filtering** for URL categorization
- **F5® IP Intelligence** for IP reputation services

BIG-IP LTM add-on software license mode

This option supports all F5® iSeries and older F5 platforms and has no specific restrictions on additional F5 software modules (including the above software services). This option is suited for environments that need to deploy SSL Orchestrator on an existing F5 device or have other functions that must run on the same device.

Optional licensing options

In addition to the above licensing modes, the following may also be licensed:

- **A URL Filtering** subscription to use the URL category database
- **An F5 IP Intelligence subscription** to detect and block known attackers and malicious traffic
- **A network hardware security module (HSM)** to safeguard and manage digital keys for strong authentication

Traffic exemptions for SSL inspection

As noted, the F5 system can be configured to distinguish between interesting and uninteresting traffic for the purposes of security processing. Examples of uninteresting traffic (including those types that cannot be decrypted) to be exempted from inspection may include:

- Guest VLANs.
- Applications that use pinned certificates.
- Trusted software update sources like Microsoft Windows updates.

- Trusted backup solutions like a crash plan.
- Any lateral encrypted traffic to internal services to be exempted.

Administrators can also exempt traffic based on domain names and URL categories. The policy rules of the SSL Orchestrator system enable administrators to enforce corporate Internet use policies, preserve privacy, and comply with regulatory requirements. Traffic exemptions based on URL category might include bypasses (and thus no decryption) for traffic from known sources of these types of traffic:

- Financial
- Health care
- Government services

Certificate requirements

Certificate requirements depend on the direction of traffic flow.

- **Outbound traffic flow** (internal client to Internet): An SSL certificate and associated private key—preferably a subordinate certificate authority (CA)—on the F5 system are needed to issue certificates to the end host for client-requested external resources that are being intercepted. To ensure that clients on the corporate network do not encounter certificate errors when accessing SSL-enabled websites from their browsers, this issuing certificate must be locally trusted in the client environment.
- **Inbound traffic flow** (Internet client to internal applications): Inbound SSL orchestration is similar to traditional reverse web proxy SSL handling. At minimum, it requires a server certificate and associated private key that matches the host name external users are trying to access. This may be a single instance certificate or a wildcard or subject alternative name (SAN) certificate if inbound SSL orchestration is defined as a gateway service.

Architecture best practices

A number of best practices can help ensure a streamlined architecture that optimizes performance and reliability as well as security. F5 recommendations include:

- Deploy the F5 systems in a sync/failover device group (S/FDG), which includes the active standby pair, with a floating IP address for high availability (HA).
- Every Cisco WSA in the service pool must be dual-homed on the inward and outward VLANs with each F5 system in the device sync/failover device group.
- Further interface redundancy can be achieved using the Link Aggregation Control Protocol (LACP). LACP manages the connected physical interfaces as a single virtual interface (aggregate group) and detects any interface failures within the group.
- Unlike with some competing solutions, the F5 systems do not need physical connections to the Cisco WSA. All the F5 system requires is layer 3 reachability to steer traffic through the firewalls. In slow networks, however, we recommend deploying the services not more than one hop away.

Security best practices

SSL orchestration generally presents a new paradigm in the typical network architecture. Previously, client/server traffic passed encrypted to inline security services, which then had to perform their own decryption if they needed to inspect that traffic. In the F5/Cisco integrated solution using SSL Orchestrator, *all* traffic to a security device is decrypted—including user names, passwords, and social security and credit card numbers. It is therefore highly recommended that security services be isolated within a private, protected enclave defined by SSL Orchestrator. It is technically possible to configure SSL Orchestrator to send the decrypted traffic anywhere that it can route to, but this is a dangerous practice that should be avoided.

IP addressing

Keeping in mind the security best practice noted above, the recommended approach to integrating security devices is to physically move them to an isolated enclave of SSL Orchestrator. This generally requires re-addressing inline layer 3 security services. The following information assumes this approach, and the sample IP addresses represent a local/internal addressing scheme.

When the Cisco WSA is deployed as either a transparent or explicit proxy, we recommend configuring its IP addresses for connected inbound and outbound interfaces from private addressing subnets provided by SSL Orchestrator. The default subnets are derived from an RFC2544 CIDR block of 198.19.0.0, which improves security and minimizes the likelihood of address collisions.

For example, administrators can configure a Cisco WSA to use the IP address 198.19.96.10/25 on its inbound interface and 198.19.96.130/25 on its outbound interface. The table below explains the IP addresses that need to be configured when deploying multiple Cisco WSAs in a service pool.

<vendor>	Inbound IP	Outbound IP	Gateway
Cisco WSA 1	198.19.96.10/25	198.19.96.130/25	198.19.96.245
Cisco WSA 2	198.19.96.11/25	198.19.96.131/25	
Cisco WSA n	198.19.96.n/25	198.19.96.n/25	

The Cisco WSA would then default route back to SSL Orchestrator on a defined address in the outbound subnet. In the example above, that IP is 198.19.96.245.

Note: A /25 network (255.255.255.128) subdivides a regular /24 network into two subnets and is a simple way to maximize local protected IP addressing for security services. Below are the IP subnets and schemes that SSL Orchestrator uses by default, but any addressing scheme can be used.

198.19.96.1 — 198.19.96.126

198.19.96.129 — 198.19.96.254

Additionally, SSL Orchestrator does not source NAT (SNAT) for the client source address across inline layer 3 services. Inline layer 3 services must therefore also have a static route back to the inbound side of SSL Orchestrator for IPs/subnets that match the client-side network. See Figure 3 for an example.

Route	Example
Client-side network	10.20.0.0/25
SSL Orchestrator service inbound self	198.19.96.7/25
SSL Orchestrator service outbound self	198.19.96.245/25

Figure 3: Sample routing for inline layer 3 services

In this sample, the following Unix/Linux route command would create a static route for traffic originating from 10.20.0.0/25 to flow back to 198.19.96.7:

```
route add -net 10.20.0.0/24 gw 198.19.96.7
```

Initial Setup

Initial setup includes configuration of the Cisco WSA and setup of SSL Orchestrator. Once these steps are complete, proceed to configuration for the chosen deployment scenario.

Configure Cisco WSA prerequisites

Before the WSA can receive traffic from SSL Orchestrator, there are a few basic configurations that must be completed. Any and all licenses should be applied, and the System Setup Wizard (SSW) should be completed. The SSW will step through configuration of the system hostname, Domain Name Servers (DNS), Network Time Protocol Servers (NTP), and time zone, along with many other settings, including the IP address, subnet mask, and hostname for the management interface (M1). Instructions for additional interface configuration appear later in this guide.

Configure SSL Orchestrator prerequisites

Before SSL Orchestrator configuration can begin, a few prerequisites need to be addressed:

Define client side and outbound side VLANs and self-IPs

For SSL orchestration in a layer 3 (routed or explicit proxy) topology, the F5 system must be configured with appropriate client-facing and outbound-facing VLANs and self-IPs. The VLANs define the connected interfaces, and the self-IPs define the respective IPv4 and/or IPv6 subnets.

1. Log in to the F5 system, and under the **Network** menu, configure the client side and outbound side VLANs and self-IPs appropriately.

Import a CA certificate and private key

For SSL orchestration in an outbound traffic topology, a local CA certificate and private key are required to re-sign the remote server certificates for local (internal) clients. For an inbound traffic topology, remote clients terminate their TLS sessions at the F5 system, so it must possess the appropriate server certificates and private keys.

1. Log in to the F5 system, and under **System—Certificate Management**, import the required CA and/or server certificates and private keys.

With these prerequisites complete, proceed to the second part of configuration, where the system is finalized for SSL Orchestrator.

Configuring SSL Orchestrator/WSA Integration

SSL Orchestrator is configured to send decrypted traffic to an inline Cisco WSA. SSL Orchestrator handles both decryption and re-encryption of HTTPS traffic, with an inspection zone installed between the ingress and egress. Decrypted traffic is steered to a service pool of Cisco WSA devices. Administrators can also deploy the F5 system as a device sync/failover device group (including an HA pair) with a floating IP address for high availability.

The Cisco WSA can be configured as either a transparent proxy or explicit proxy inside the inspection zone.

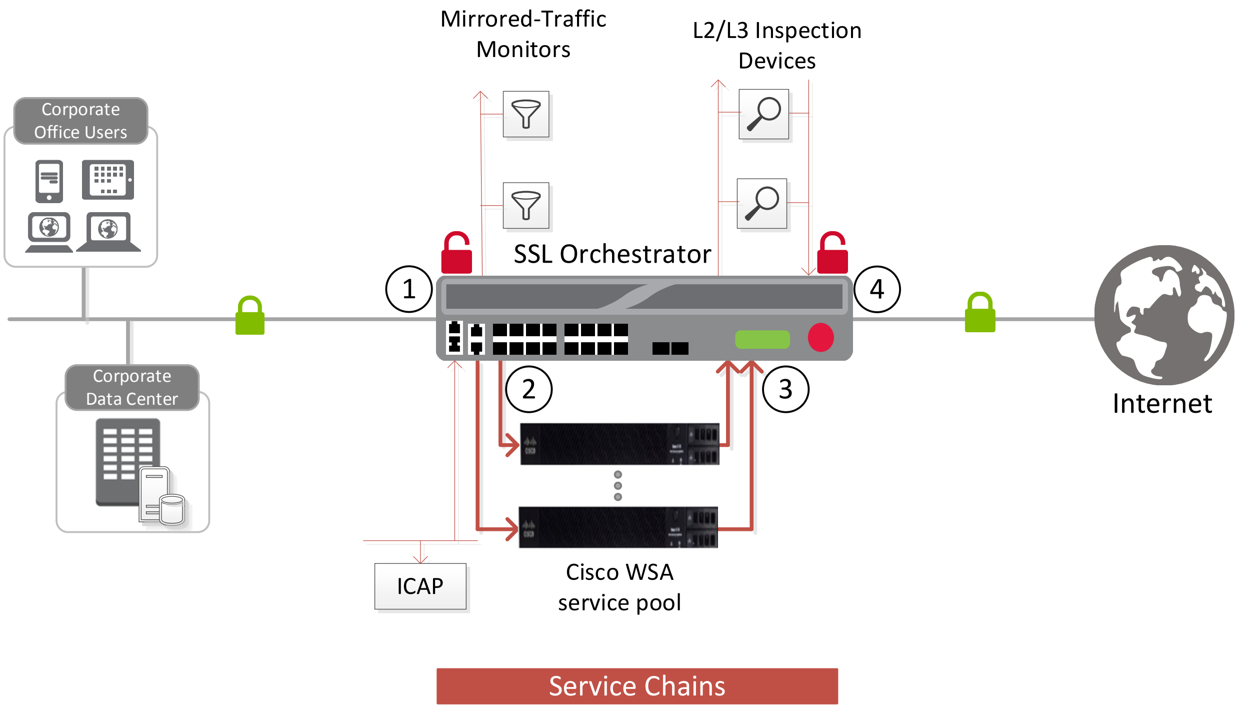


Figure 4: Traffic flow for the integrated solution

How traffic flows in this deployment:

1. Client traffic arrives at the ingress side of the F5 system, where it is classified and interesting HTTPS traffic is decrypted as part of the SSL handling process.
2. SSL Orchestrator steers the decrypted traffic through the load balanced Cisco WSA service pool as part of a service chain that potentially includes multiple types of security services.
3. The HTTP traffic is inspected by the Cisco WSA services for any hidden threats before sending that traffic back to the F5 system.
4. The F5 system orchestrates the decrypted traffic through other services in the chain before it aggregates and re-encrypts the traffic, which is then routed to the next destination.

Note: Inside the SSL Orchestrator inspection zone, all traffic passing to the Cisco WSA devices is unencrypted HTTP. It is a security best practice to protect this device and the network between it and the F5 device. It is therefore recommended that the Cisco WSA devices be moved to the secure enclave created by SSL Orchestrator. This brings with it a few architectural changes.

If the Cisco WSA was previously installed in the network to service client traffic as either a transparent or explicit proxy, SSL Orchestrator must now fulfill that role:

- If the Cisco WSA was configured as an **explicit proxy**, SSL Orchestrator must then be configured as an explicit proxy and clients must communicate with it as their explicit proxy gateway.
- If the Cisco WSA was configured as a **transparent proxy**, SSL Orchestrator must be configured as a transparent proxy and routed client traffic must now pass through it.

Cisco WSA devices inside the inspection zone can be explicit or transparent, irrespective of the proxy mode of SSL Orchestrator.

If the Cisco WSA device was previously handling explicit proxy user authentication, SSL Orchestrator must now fulfill this role. F5 BIG-IP Access Policy Manager (APM) can be provisioned to provide the same authentication functionality.

Configure the Cisco WSA

Following are the minimum requirements to configure a Cisco WSA device for integration with SSL Orchestrator. Please refer to Cisco documentation for additional product-specific information.

Configure interfaces

The Cisco WSA devices must be physically or logically two-armed. In other words, each device must have separate physical or logical inbound and outbound interfaces. This can either be two separate physical interfaces or a single 802.1q VLAN tagged interface (a single interface with two tagged VLANs). The recommended method is to use the P1 interface for inbound traffic and the P2 for outbound traffic.

The M1 interface should be reserved for management traffic, while the P1 interface and optionally the P2 interface are defined for data plane traffic.

1. To complete the interface configuration in the WSA GUI, navigate to **Network > Interfaces > Edit Settings...** and set the relevant parameters described below.

Setting	User Input
Interfaces	Configure the P1 and P2 interface addressing according to the deployment. For example, if using the **Auto Manage** option in the HTTP service settings in SSL Orchestrator, enter an address in the 198.19.96.0/25 subnet for P1 and in the 198.19.96.245/25 subnet for P2. Otherwise, use appropriate addresses per the SSL Orchestrator configuration.
Separate Routing for Management Services	This setting should be enabled to limit management services to the M1 interface only. In this configuration, services such as SSH, HTTP/S, and FTP which are used to administer the WSA are not allowed unless they ingress on the M1 interface.
Appliance Management Services	This option defines which services are restricted by the previous setting.

Configure VLANs (Optional)

If two separate data plane interfaces are not available (P1 and P2), it is also possible to configure the single P1 interface with two VLANs. This is best done from the command line interface using SSH. From the Cisco WSA CLI, perform the following actions to define the P1 (inbound) and P2 (outbound) interfaces:

1. Ensure that the interfaces are enabled and mapped correctly. This will list the active interfaces and their layer 2 (MAC) addresses

   ```
   []> etherconfig
   []> media
   ```

2. Create a VLAN.

   ```
   [] > etherconfig
   [] > vlan
   ```

   Assuming none already have been created:

   ```
   [] > new
   ```

3. Enter a VLAN tag ID for the interface (for example, “34”). This is an arbitrary 802.1q VLAN tag ID number. Enter any reasonable value below 4096.

4. Enter the name or number of the ethernet interface you wish to bind to; select the number of the P1 interface.

5. Perform the same action for a second VLAN on the same interface.

6. Create an IPv4 address for the new VLANs.

   ```
   []> interfaceconfig
   []> new
   ```

7. Select the number of the first VLAN from the presented list.

8. For the question, **Would you like to configure an IPv4 address for this interface (y/n),** enter **Y**.

9. Enter an IPv4 address. Assuming the best practice security, this will be an address defined for use with SSL Orchestrator (see IP Addressing above). For example, when using the **Auto Manage** option in SSL Orchestrator’s HTTP service settings, enter an address in the 198.19.96.0/25 subnet (ex. 198.19.96.60). Otherwise, use an appropriate address that will be in a different subnet than the P2 outbound interface.

10. For **Netmask**, enter the appropriate subnet mask for the defined IPv4 address (for example “24”, “255.255.255.0”, or “0xffffff00”). Again, if following best practice security with the /25 subnet examples above, enter “25” or “255.255.255.128”.

11. For the question, **Would you like to configure an IPv6 address for this interface (y/n)?** enter **N**.

12. Enter a **Hostname**.

13. Perform steps 7 through 12 for the second VLAN, starting from the [ ]> new command. This time, select the number of the second VLAN from the presented list. Follow the same IP addressing instructions as above, except that the IPv4 address will be in a separate subnet. If following security best practice and using the **Auto Manage** option in SSL Orchestrator’s HTTP service settings, when the inbound address is in the 198.19.96.0/25 subnet, then the outbound interface will be in the 198.19.96.129/25 subnet. Enter an IP address between 198.19.96.130 and 198.19.96.254. Otherwise use an appropriate address that will be in a different subnet than the P1 inbound interface.

14. Commit these settings.

```
[] > commit
```

Configure routes

The Cisco WSA devices will require two routes:

- **A gateway route to send data plane traffic outbound**. This is an IP address on SSL Orchestrator facing the outbound side of the WSA.
- **A static return route**. SSL Orchestrator does not source NAT (SNAT) traffic across inline security devices by default, so the source address passing through the WSA will be foreign and will need a static return route to define the path back to the F5 system on the inbound side of the WSA.

These can be configured using either the GUI or the command-line interface (CLI). To configure using the GUI, navigate to **Network > Routes** and add the appropriate route to the **Data** routing table.

To configure using the CLI, follow these steps:

1. Configure the gateway route.

```
[] > setgateway
```

2. Select **IPv4** from the list of options.

3. Select **Data Default Gateway** from the list of options.

4. Enter a new default gateway. Enter an address that is in the outbound side interface/VLAN subnet. For example, if using the auto-managed IP subnets, you might use an address of 198.19.96.244.

5. Configure the static return route.

```
[] > routeconfig
```

Select **IPv4** from the list of options.

6. Select **Data** from the list of options.

```
[] > new
```

7. Enter a **Name** for the route. Enter any arbitrary name.

8. Enter the destination IPv4 address to match. CIDR addresses such as 192.168.42.0/42 are also allowed.
9. Enter the client-side subnet. This is the source-side subnet that the WSA will see. For example, if the client-side subnet is 10.20.0.0/24, enter this value directly.
10. Enter the gateway IP address for traffic to **<subnet>**. This will be the self-IP on the F5 system on the inbound side of the WSA. This will be defined as part of SSL Orchestrator configuration. When using the auto-managed option, this might be an address in the 198.19.96.0/25 subnet (ex. 198.19.96.1).
11. Commit these settings.

```
[] > commit
```

Note: In addition to the client and default routes, we recommend manually configuring various system services to use the management routing table to access the Internet. The following services can be manually configured to use the management routing table:

- External URL feeds
- AMP file reputation and analysis
- Updates and upgrades
- DNS
- Active Directory

If these services use the data routing table, they will egress through SSL Orchestrator and will require additional configuration on that device to allow the connections.

Configure DNS

DNS settings are minimally required to allow the Cisco WSA devices to talk to remote (home base) services for engine updates. They are also required if the WSA is configured as an explicit proxy.

There are generally two options for DNS: passing through either the management interface (M1) or the data plane outbound interface (P2 or the second VLAN on P1). In WSA version 11 and up, it is possible to explicitly configure which routing table to use. When the solution is configured for data plane egress, SSL Orchestrator must be further configured to allow this traffic out. For this reason, the recommended practice is to limit DNS traffic to the M1 interface.

1. To configure DNS in the WSA GUI, navigate to **Network > DNS**. Identify the relevant settings below:

Setting	User Input
Primary DNS Server	Configure the desired DNS resolver and any alternative resolvers which are required. Optionally, configure the WSA to use the Internet Root DNS servers.
Secondary DNS Servers	Optionally configure any additional servers to be used.
Routing Table for DNS Traffic	This option is available in WSA version 11 and up. If it is available, the best practice is to enable this setting to prevent the need for additional configuration on SSL Orchestrator to allow this traffic.

Alternately, to configure DNS using the CLI, reference the steps below:

1. Enter DNS configuration:

```
[] > dnsconfig
[] > setup
```

2. For the question, **Do you want the Gateway to use the Internet's root DNS servers or would you like it to use your own DNS servers?** choose the appropriate setting for your network:

- 1. Use Internet root DNS servers.
- 2. Use own DNS cache servers.

```
[2]>
```

3. Choose the routing table to use. Choose option 2 if available.

- 1. Data
- 2. Management

```
[2]>
```

4. Enter the number of seconds to wait before timing out reverse DNS lookups, or accept the default.

```
[20]>
```

5. Enter the minimum TTL in seconds for DNS cache. The default is 1800; lower this setting to 300 to accommodate low TTL records.

```
[1800]>
```

6. Enter the number of failed attempts before considering a local DNS server offline. We recommend accepting the default, 100.

```
[100]>
```

7. Enter the interval in seconds for polling an offline local DNS server. Again, accept the default, 5 seconds.

```
[5]>
```

8. The system will indicate what local DNS cache servers are currently being used. This will reflect the server address configured using the System Setup Wizard.

```
1. Priority: 0  192.168.0.100
```

9. Commit these settings.

```
[] > commit
```

Configure the web proxy service

Please refer to the appropriate Cisco WSA documentation for more detailed information on configuring WSA. The following are the minimal settings required for integration with SSL Orchestrator. Configuration of the web proxy will be performed from the WSA UI.

In this scenario, the Cisco WSA can be configured as either a transparent or explicit web proxy.

1. In the Cisco WSA UI, under **Security Services > Web Proxy**, confirm or create the following settings:

Setting	User Input
HTTP Ports to Proxy	SSL Orchestrator passes unencrypted HTTP traffic to the WSA, but by default it does not alter the original destination ports (80 for HTTP and 443 for HTTPS). This option defines the ports that SSL Orchestrator will use when sending that traffic to the WSA. Traditionally an explicit proxy listens on port 3128 or 8080, and a transparent proxy HTTP listens on port 80. These ports are arbitrary, but to match the default configuration of SSL Orchestrator, these can be changed to 80 and 443.
Caching	SSL Orchestrator does not interfere in any way with WSA's caching functions.
Proxy Mode	When in **Transparent** mode, the proxy (WSA) can accept for transparent and explicit forward connections. When in **Forward** mode, only explicit forward connections are supported. To allow for the WSA to authenticate clients, this setting must be set to **Transparent**. *The rest of this guide assumes a transparent proxy configuration from the WSA perspective.*
IP Spoofing	This mode enables WSA to source outbound traffic from the original client source address. In other words, if **Disabled**, the traffic leaving the WSA will be sourced from the outbound side IP address of the WSA itself. If **Enabled**, the traffic leaving the WSA will be sourced from the original client source address. SSL Orchestrator can work with either setting irrespective of any other devices in the service chain.

Note: SSL Orchestrator does not source NAT (SNAT) traffic to inline services by default, so X-Forwarded-For headers are generally not required. The WSA device will receive the true client source address in the packets.

Authentication (optional)

The WSA may optionally be used to authenticate users to enforce web access policies and to allow for web tracking and reporting based on Active Directory (AD) users and groups. This is not required, and therefor this section may be skipped if WSA authentication is not part of the deployment.

Enforcing authentication in the WSA when integrated with SSL Orchestrator requires a transparent proxy configuration. When clients initially send an unauthenticated request to a web site, the WSA will respond with an HTTP redirect response that directs the client to its own hostname. Upon following the redirect, the client is then challenged for authentication. Once authentication has been completed, the client is then redirected back to the original URL.

Because the WSA only receives HTTP traffic from SSL Orchestrator, it will always redirect the client to the original URL over HTTP when authentication has completed. If the original request was sent over HTTPS and the site does not accept connections over HTTP, the client request will fail. To account for this, an F5® iRule® must be created to re-write the redirect URL string sent to the client from the WSA. This is addressed as part of SSL Orchestrator configuration later in this guide.

Because a typical deployment of SSL Orchestrator will utilize multiple WSA appliances for load balancing and high availability, specific steps are required to ensure that when the client follows the redirect for authentication of the request, it is delivered to the same WSA to which the initial web request was made. This is accomplished using a dedicated virtual IP (VIP) and CARP load balancing in SSL Orchestrator configuration, which is also addressed later in this guide.

The redirect hostname configured on the WSA is the name that will be used for authentication redirects. This hostname should resolve to the VIP that is configured on SSL Orchestrator for direct authentication between the clients and the WSA.

1. Configure this hostname using the WSA GUI by navigating to **Network > Authentication**. Please refer to the WSA user guide documentation regarding domain authentication configuration, which is outside the scope of this guide.

If Kerberos is required in the deployment, additional steps are required to employ the key tab feature on the WSA. Please refer to specific documentation regarding this feature, which also is outside the scope of this guide.

Configure SSL Orchestrator

A Cisco WSA device is configured as an HTTP service in SSL Orchestrator. This configuration will focus on the traditional outbound (forward proxy) use case. Once logged into the F5 System, navigate to the **SSL Orchestrator** menu and review the environment.

Create the SSL Orchestrator deployment using guided configuration

SSL Orchestrator's guided configuration presents a completely new and streamlined user experience. This workflow-based architecture provides intuitive, re-entrant configuration steps tailored to the selected topology.

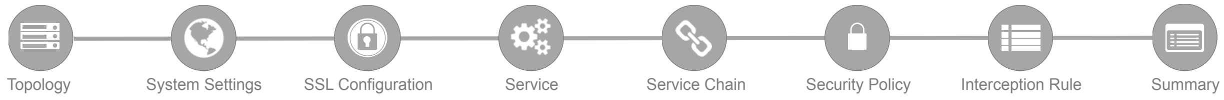


Figure 5: The SSL Orchestrator guided configuration workflow

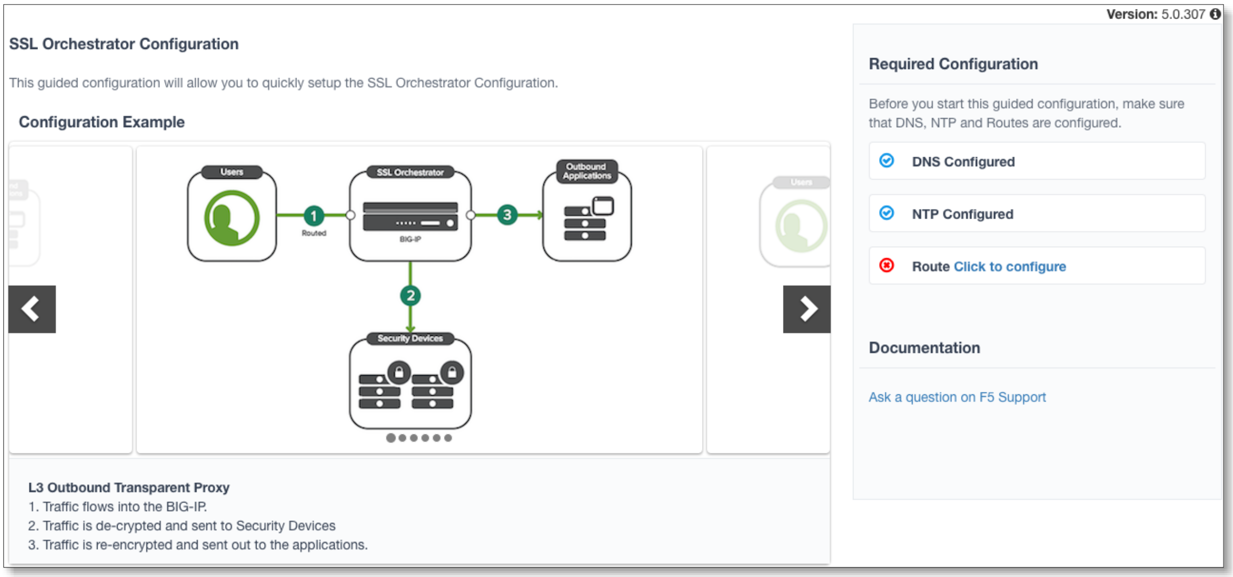
The following steps will walk through the guided configuration to build a simple transparent forward proxy.

Initialization

If this is the first time accessing SSL Orchestrator in a new BIG-IP build, upon first access, the guided configuration will automatically load and deploy the built-in SSL Orchestrator package.

Configuration review and prerequisites

1. Take a moment to review the topology options and workflow configuration steps involved.
2. (Optional). Satisfy any of the **DNS**, **NTP** and **Route** prerequisites from the page in Figure 6. Keep in mind, however, that aside from NTP, the guided configuration will provide an opportunity to define DNS and route settings later in the workflow.
3. No other configurations are required on this page, so click **Next**.



Version: 5.0.307

SSL Orchestrator Configuration

This guided configuration will allow you to quickly setup the SSL Orchestrator Configuration.

Configuration Example

L3 Outbound Transparent Proxy
1. Traffic flows into the BIG-IP.
2. Traffic is de-crypted and sent to Security Devices
3. Traffic is re-encrypted and sent out to the applications.

Required Configuration

Before you start this guided configuration, make sure that DNS, NTP and Routes are configured.

✓ DNS Configured

✓ NTP Configured

☒ Route Click to configure

Documentation

Ask a question on F5 Support

Figure 6: As an option, prerequisites can be configured from the initial guided configuration screen.

Topology properties

SSL Orchestrator now creates discreet configurations based on the selected topology. An explicit forward proxy topology will ultimately create an explicit proxy listener and its relying transparent proxy listener, but the transparent listener will be bound only to the explicit proxy tunnel. If a subsequent transparent forward proxy topology is configured, it will not overlap the existing explicit proxy objects.

The **Topology Properties** page includes the following options:

- The **Protocol** option presents four protocol types:
  - **TCP**. This option creates a single TCP wildcard interception rule for the L3 inbound, L3 outbound, and L3 explicit proxy topologies.
  - **UDP**. This option creates a single UDP wildcard interception rule for L3 inbound and L3 outbound topologies.
  - **Other**. This option creates a single any protocol wildcard interception rule for L3 inbound and L3 outbound topologies. This option is typically used for non-TCP/UDP traffic flows.
  - **Any**. This option creates the TCP, UDP and non-TCP/UDP interception rules for outbound traffic flows.
- The **SSL Orchestrator Topologies** option page presents six topologies. See Figure 7.
  - **L3 Explicit Proxy**. This is the traditional explicit forward proxy.
  - **L3 Outbound**. This is the traditional transparent forward proxy.
  - **L3 Inbound**. This is a reverse proxy configuration.
  - **L2 Inbound**. The layer 2 topology options insert SSL Orchestrator as a bump-in-the-wire in an existing routed path, where SSL Orchestrator presents no IP addresses on its outer edges. The L2 inbound topology provides a transparent path for inbound traffic flows.

- **L2 Outbound**. The layer 2 topology options insert SSL Orchestrator as a bump-in-the-wire in an existing routed path, where SSL Orchestrator presents no IP addresses on its outer edges. The L2 outbound topology provides a transparent path for outbound traffic flows.
- **Existing Application**. This topology is designed to work with existing BIG-IP LTM applications. Whereas the L3 inbound topology provides an inbound gateway function for SSL Orchestrator, the **Existing Application** option works with BIG-IP LTM virtual servers that already perform their own SSL handling and client-server traffic management. The existing application configuration workflow proceeds directly to service creation and security policy definition, then exits with an SSL Orchestrator-type access policy and per-request policy that can easily be consumed by a BIG-IP LTM virtual server.

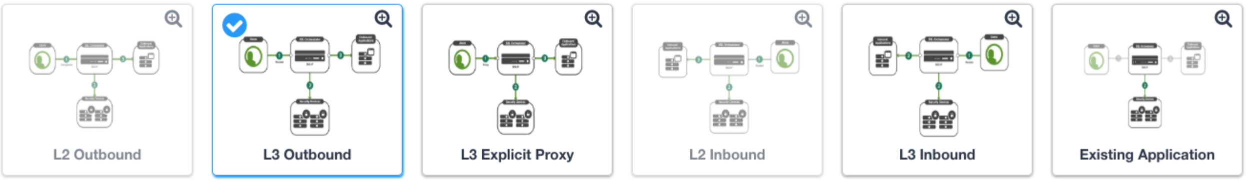


Figure 7: The six topology options

1. To create a typical outbound SSL visibility solution, enter a **Name**.
2. Select the **Any** protocol, which will create separate TCP, UDP and non-TCP/UDP interception rules.
3. Select **IPv4** for the **IP Family**.
4. Select the **L3 Outbound** or **L3 Explicit Proxy Topology**.
5. Click **Save & Next**.

SSL Configurations

This page defines the specific SSL settings for the selected topology, in this case a forward proxy, and controls both client-side and server-side SSL options. If existing SSL settings are available (from a previous workflow), it can be selected and re-used. Otherwise the **SSL Configurations** page creates new SSL settings for this workflow.

1. Under **Client-side SSL**, select the appropriate option for each of the following:

- **Cipher Type**. The cipher type can be a **Cipher Group** or a **Cipher String**. For the former, select a previously defined cipher group from **Local Traffic > Ciphers > Groups**. For the latter, enter a cipher string that appropriately represents the client-side TLS requirement. For most environments, the **Default** is optimal.
- **Certificate Key Chain**. The certificate key chain represents the certificate and private key used as the template for forged server certificates. While reissuing server certificates on-the-fly is generally easy, private key creation tends to be a CPU-intensive operation. For that reason, the underlying SSL forward proxy engine forges server certificates from a single defined private key. This setting gives administrators the opportunity to apply their own template private key, and optionally to store that key in a FIPS-certified HSM for additional protection. The built-in default certificate and private key uses 2K RSA and is generated from scratch when the F5 system is installed. Click **Add**, select **default.crt** and **default.key**, and click **Done**.

- **CA Certificate Key Chain**. An SSL forward proxy must re-sign or “forge” remote server certificate to local clients using a local CA certificate, and local clients must trust this local CA. This setting defines the local CA certificate and private key used to perform the forging operation. Assuming a local CA certificate and key were imported in the initial setup, select them here.

  Note: SSL settings minimally require an RSA-based template and CA certificates but can also support elliptic curve digital signature algorithm (ECDSA) certificates. In this case, SSL Orchestrator would forge an EC certificate to the client if the TLS handshake negotiated an ECDHE_ECDSA cipher. To enable EC forging support, add both an EC template certificate and key, and an EC CA certificate and key.

- **[Advanced] Bypass on Handshake Alert**. This advanced setting allows the underlying SSL forward proxy process to bypass SSL decryption if an SSL handshake error is detected on the server side. We recommend leaving this **Disabled**.

- **[Advanced] Bypass on Client Certificate Failure**. This advanced setting allows the underlying SSL forward proxy process to bypass SSL decryption if it detects a certificate request message from the server, as in when a server requires mutual certificate authentication. We recommend leaving this **Disabled**.

Note: These two advanced bypass options can create a security vulnerability. If a colluding client and server can force an SSL handshake error or a client certificate authentication, they can effectively bypass SSL inspection. This is why we recommended leaving these settings disabled.

2. Under **Server-side SSL**, select appropriate settings from the following options:

- **Cipher Type**. The cipher type can be a **Cipher Group** or **Cipher String**. For the former, select a previously defined cipher group from **Local Traffic > Ciphers > Groups**. For the latter, enter a cipher string that appropriately represents the server-side TLS requirement. For most environments, the **Default** is optimal.

- **Trusted Certificate Authority**. Browser vendors routinely update the CA certificate stores in their products to keep up with industry security trends and to account for new and revoked CAs. In the SSL forward proxy use case, however, the F5 product now performs all server-side certificate validation in lieu of the client browser. SSL Orchestrator ships with a CA certificate bundle that maintains a list of CA certificates common to the browser vendors. A more comprehensive bundle can (and should) be obtained from the F5 downloads site and selected here. Otherwise, it is safe to select the built-in **ca-bundle.crt**.

- **[Advanced] Expire Certificate Response**. SSL Orchestrator performs validation on remote server certificates and can control what happens if it receives an expired server certificate. The options are **Drop**, which simply drops the traffic, and **Ignore**, which mirrors an expired forged certificate to the client. The default and recommended behavior for the forward proxy use case is to **Drop** traffic on an expired certificate.

- **[Advanced] Untrusted Certificate Authority**. SSL Orchestrator performs validation on remote server certificates and can control what happens if it receives an untrusted server certificate, based on the **Trusted Certificate Authority** bundle. The options are **Drop**, which simply drops the traffic, and **Ignore**, which allows the traffic and forges a good certificate to the client. The default and recommended behavior for the forward proxy use case is to **Drop** traffic with an untrusted certificate.

- **[Advanced] OCSP**. This advanced setting selects an existing OCSP profile or can create a new one for server-side Online Certificate Status Protocol (OCSP) and OCSP stapling. With this setting **Enabled**, if a client issues a Status_Request message in its ClientHello message (an indication that it supports OCSP stapling), SSL Orchestrator will issue a corresponding Status_Request message in its server-side TLS handshake. SSL Orchestrator will then forge the returned OCSP stapling response back to the client. If the server does not respond with a staple but contains an Authority Info Access (AIA) field that points to an OCSP responder URL, SSL Orchestrator will perform a separate OCSP request. The returned status is then mirrored in the stapled client-side TLS handshake.
- **[Advanced] CRL**. This setting selects an existing CRL profile or can create a new one for server-side Certificate Revocation List (CRL) validation. With this setting **Enabled**, SSL Orchestrator attempts to match server certificates to locally-cached CRLs.

3. Click **Save & Next**.

Services List

The **Services List** page is used to define security services that attach to SSL Orchestrator. The guided configuration includes a services catalog that contains common product integrations. Beneath each of these catalog options is one of the five basic service types. The service catalog also provides a generic security service option. (Depending on the user's screen resolution, it may be necessary to scroll down to see additional service options.)

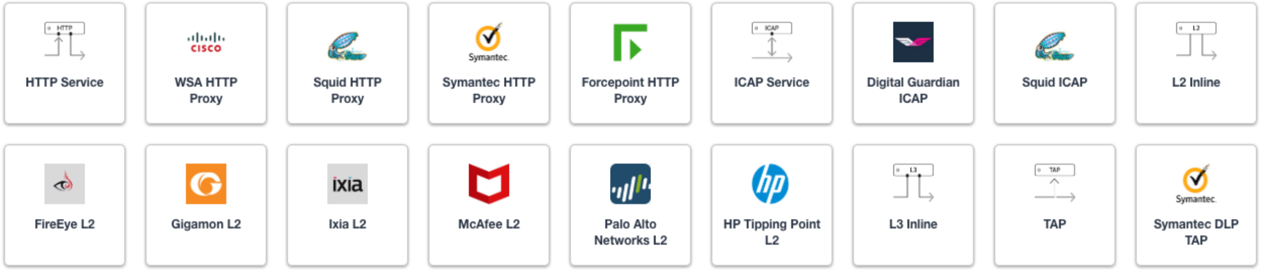


Figure 8: Sample service list options

1. To define the Cisco WSA service, either double-click it, or select it in the service catalog and click **Add**.
2. Enter a unique **Name** for this service (for example, "**WSA**").
3. Configure the following service details:

- **Auto Manage Addresses**. When **Enabled**, this setting provides a set of unique, non-overlapping, non-routable IP addresses to be used by the security service. If **Disabled**, the **To** and **From** IP addresses must be configured manually. We recommend leaving this option **Enabled** (selected).

Note: In environments where SSL Orchestrator is introduced to existing security devices, it is a natural tendency to not want to have to move these devices. And while SSL Orchestrator certainly allows this, not moving the security devices into SSL Orchestrator-protected enclaves runs the risk of exposing sensitive decrypted traffic, unintentionally, to other devices that may be connected to these existing networks. It is therefore *highly* recommended, and a security best practice, to remove integrated security devices from existing networks and place them entirely within the isolated enclave created and maintained by SSL Orchestrator.

- **Proxy Type**. This setting defines the proxy mode that the inline HTTP service is in. In **Explicit** proxy mode, the configuration will request the service's listening IP and proxy port. In **Transparent** proxy mode, the configuration will simply request the service's inbound interface IP address. To support WSA transparent proxy mode authentication, select **Transparent**.

4. Configure the **To Service**, which defines the network connectivity from SSL Orchestrator to the inline security device.

- When **Auto Manage Addresses** (above) is **Enabled**, this IP address will be predefined, and therefore the inbound side of the service must match this IP subnet. When **Auto Manage Addresses** option is **Disabled**, the IP address must be defined manually.
- **VLAN**. Select **Create New**, provide a unique name (for example, **WSA_in**), select the F5 interface connecting to the inbound side of the service, and add a VLAN tag value if required.

5. Configure the following service details:

- **Service Down Action**. SSL Orchestrator monitors the load balanced pool of security devices, and if all pool members fail, it can actively bypass this service (**Ignore**), or stop all traffic (**Reset**, **Drop**).
- **Security Devices**. This setting defines the inbound-side IP address of the inline HTTP service used for passing traffic to this device. Multiple load balanced IP addresses can be defined here. For a transparent proxy HTTP service, only an IP address is required. For an explicit proxy HTTP service, the IP address and listening port is required. Click **Add**, enter the service's inbound-side **IP Address** (the port value is explicit), and then click **Done**.

6. Configure the **From Service**, which defines the network connectivity from the inline security device to SSL Orchestrator.

- With the **Auto Manage Addresses** option **Enabled**, this IP address will be pre-defined, and therefore the outbound side of the service must match this IP subnet. With this option **Disabled**, the IP address must be defined manually.
- **VLAN**. Select **Create New**, provide a unique **Name** (for example, **WSA_out**), select the F5 interface connecting to the outbound side of the service, and add a VLAN tag value if required.

7. Continue to configure the following service details:

- **Enable Port Remap**. For an inline transparent proxy configuration, this setting defines the port that detected HTTPS traffic is remapped to. For the WSA, enter a remapping port of **80**.

- **Manage SNAT Settings**. This setting allows SSL Orchestrator to SNAT (source NAT) traffic to an inline service. This is especially useful in a load-balanced SSL Orchestrator scaling configuration but is not required in this use case. Leave the default setting, **None**.
- **Authentication Offload**. When an access authentication profile is attached to an explicit forward proxy topology, this option will present the authenticated username value to the service as an X-Authenticated-User HTTP header. To support WSA transparent proxy mode authentication, leave this set to **None**.
- **iRules**. SSL Orchestrator allows for the insertion of additional iRule logic at different points. An iRule defined at the service only affects traffic flowing across this service. It is important to understand, however, that such iRules must not be used to control traffic flow (for example, pools, nodes, or virtuals), but rather should be used to view or modify application layer protocol traffic. For example, an iRule assigned here could be used to view and modify HTTP traffic flowing to or from the service. Additional iRules are not required, however, so in general, leave this empty.

8. Click **Save & Next**.

Service chain list

Service chains are arbitrarily ordered lists of security devices. Based on environmental requirements, different service chains may contain different reused sets of services, and different types of traffic can be assigned to different service chains. For example, HTTP traffic may need to go through all of the security services, while non-HTTP traffic only goes through a subset of the services, and traffic destined to a financial service URL can bypass decryption and still flow through a smaller set of security services.

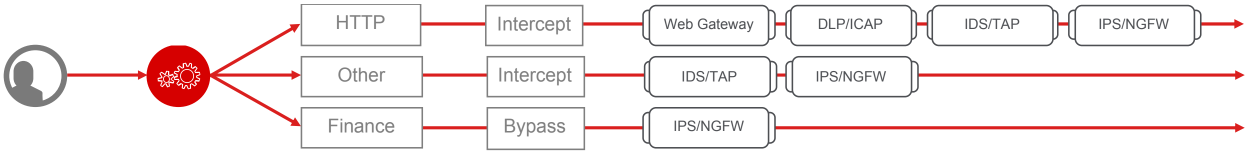


Figure 9: Different service chains for different types of traffic

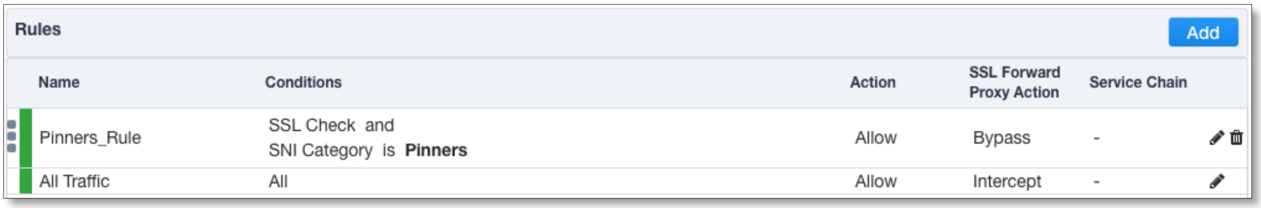
1. Click **Add** to create a new service chain containing all of the desired security services.
2. Provide a unique **Name** to this service (for example, **my_service_chain**).
3. Select any number of desired **Services** and move them into the **Selected Service Chain Order** column. Optionally, order them as desired. Select all of the services.
4. Click **Save & Next**.

Security policy

Security policies are the set of rules that govern how traffic is processed in SSL Orchestrator. The actions a rule can take include:

- Whether to allow the traffic.
- Whether to decrypt the traffic.
- Which service chain, if any, to pass the traffic through.

The guided configuration presents an intuitive, rule-based, drag-and-drop user interface for the definition of security policies.



Rules

Name	Conditions	Action	SSL Forward Proxy Action	Service Chain
Pinners_Rule	SSL Check and SNI Category is **Pinners**	Allow	Bypass	-
All Traffic	All	Allow	Intercept	-

Figure 10: Security policies and their associated actions

In the background, SSL Orchestrator maintains these security policies as visual per-request policies. If traffic processing is required that exceeds the capabilities of the rule-based user interface, the underlying per-request policy can be managed directly.

1. Create any security rules and associate actions as required, then click **Save & Next**.

Interception rules

Interception rules, which are based on the selected topology, define the "listeners," analogous to BIG-IP LTM virtual servers, that accept and process different types of traffic (such as TCP, UDP, or other). The resulting BIG-IP LTM virtual servers will bind the SSL settings, VLANs, IPs, and security policies created in the topology workflow.

The Interception Rules page includes default and custom options.

1. Select the outbound rule type:
    - For the transparent forward proxy topology workflow, the **Custom** option exposes source address, destination address and port, and layer 7 profile selections. These are not generally required for most outbound configurations, so keep the **Default** selection.
    - This option is not present for explicit proxy configurations.
2. Select the **Proxy Server Settings** as needed. This setting, which is present for an explicit proxy configuration, defines the SSL Orchestrator explicit proxy listening IP address and proxy port. For explicit proxy authentication, this section also allows for the selection of a BIG-IP APM SWG-explicit access policy.
3. Indicate the **Ingress Network (VLANs)**. This defines the VLANs through which traffic will enter. For a forward proxy topology, this would be a client-side VLAN.
4. Enable **L7 Interception Rules**. FTP and email protocol traffic are all "server-speaks-first" protocols, and therefore SSL Orchestrator must process these separately from typical client-speaks-first protocols like HTTP. This selection enables processing of each of these protocols, which create separate port-based listeners for each. As required, selectively **Enable** the additional protocols that need to be decrypted and inspected through SSL Orchestrator. (This option is not present for explicit proxy configurations.)
5. Click **Save & Next**.

Egress settings

The **Egress Settings** page defines the topology-specific egress characteristics.

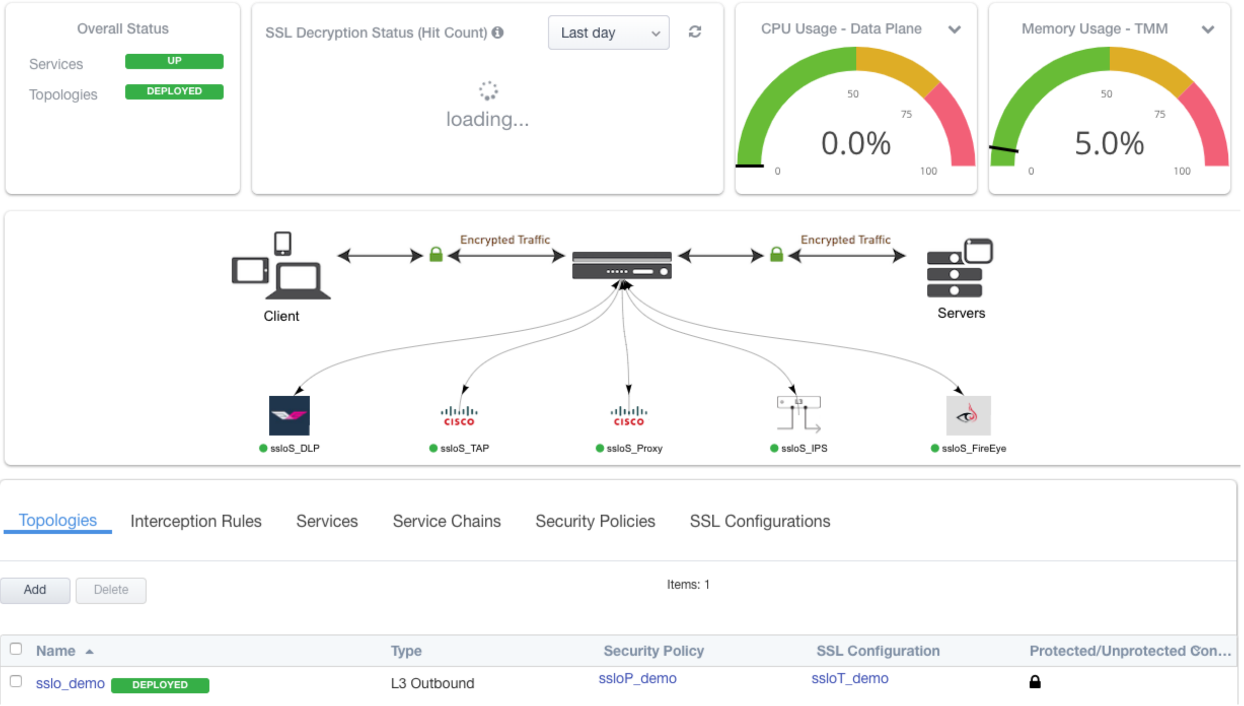
1. **Manage SNAT** settings. Define if and how source NAT (SNAT) will be used for egress traffic.
2. Define **Gateways**. This setting defines the next hop route for traffic. For an outbound configuration, this is usually a next hop upstream router.

Summary

The **Summary** page presents an expandable list of all of the workflow-configured objects.

1. To expand the details for any given setting, click the corresponding arrow icon on the far right.
2. To edit any given setting, click the corresponding pencil icon, which will send the workflow back to the selected settings page.
3. When the defined settings are correct, click **Deploy**.

Upon successful deployment of the configuration, SSL Orchestrator will display a dashboard. See Figure 11.



Overall Status

Services: UP

Topologies: DEPLOYED

SSL Decryption Status (Hit Count): Last day — loading...

CPU Usage - Data Plane: 0.0%

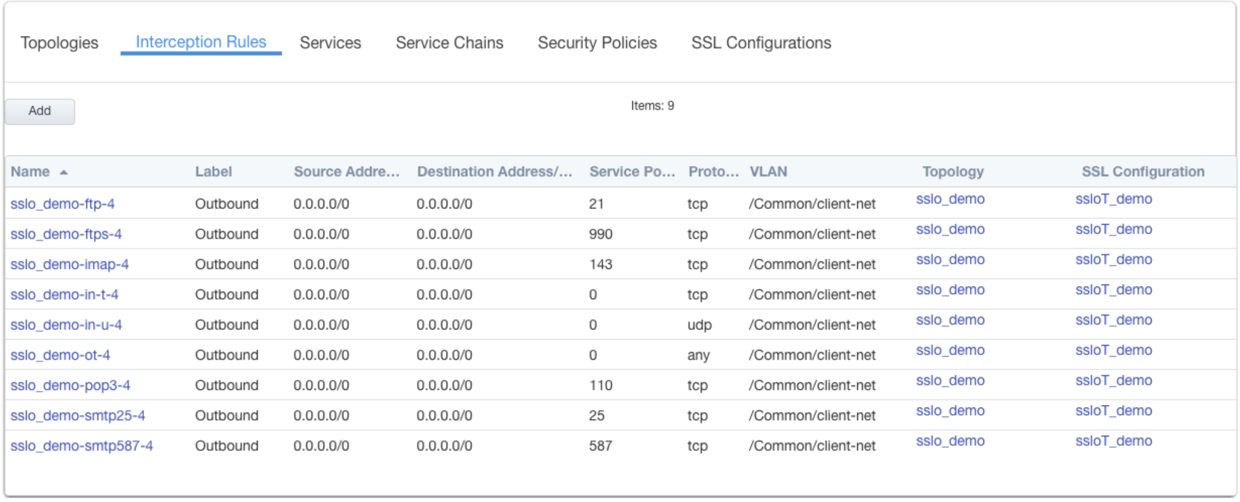
Memory Usage - TMM: 5.0%

Items: 1

	Name	Type	Security Policy	SSL Configuration	Protected/Unprotected Con...
☐	sslo_demo DEPLOYED	L3 Outbound	ssloP_demo	ssloT_demo	🔒

Figure 11: The deployment dashboard

The **Interception Rules** tab shows the listeners that were created per the selected topology.



Name ▲	Label	Source Addre...	Destination Address/...	Service Po...	Proto...	VLAN	Topology	SSL Configuration
sslo_demo-ftp-4	Outbound	0.0.0.0/0	0.0.0.0/0	21	tcp	/Common/client-net	sslo_demo	ssloT_demo
sslo_demo-ftps-4	Outbound	0.0.0.0/0	0.0.0.0/0	990	tcp	/Common/client-net	sslo_demo	ssloT_demo
sslo_demo-imap-4	Outbound	0.0.0.0/0	0.0.0.0/0	143	tcp	/Common/client-net	sslo_demo	ssloT_demo
sslo_demo-in-t-4	Outbound	0.0.0.0/0	0.0.0.0/0	0	tcp	/Common/client-net	sslo_demo	ssloT_demo
sslo_demo-in-u-4	Outbound	0.0.0.0/0	0.0.0.0/0	0	udp	/Common/client-net	sslo_demo	ssloT_demo
sslo_demo-ot-4	Outbound	0.0.0.0/0	0.0.0.0/0	0	any	/Common/client-net	sslo_demo	ssloT_demo
sslo_demo-pop3-4	Outbound	0.0.0.0/0	0.0.0.0/0	110	tcp	/Common/client-net	sslo_demo	ssloT_demo
sslo_demo-smtp25-4	Outbound	0.0.0.0/0	0.0.0.0/0	25	tcp	/Common/client-net	sslo_demo	ssloT_demo
sslo_demo-smtp587-4	Outbound	0.0.0.0/0	0.0.0.0/0	587	tcp	/Common/client-net	sslo_demo	ssloT_demo

Figure 12: Interception rules

This completes the configuration of SSL Orchestrator as a forward proxy. At this point an internal client should be able to browse out to external (Internet) resources, and decrypted traffic will flow across the security services.

Testing the Solution

Test the deployed solution using one of the following options:

Server certificate test

Open a browser on the client system and navigate to an HTTPS site such as https://www.google.com. Once the site opens in the browser, check the server certificate of the site and verify that it has been issued by the local CA set up on the F5 system. This confirms that the SSL forward proxy functionality enabled by SSL Orchestrator is working correctly.

Decrypted traffic analysis on the F5 system

Perform a TCP dump on the F5 system to observe the decrypted clear text traffic. This confirms SSL interception by the F5 device.

```
tcpdump -lnni eth<n> -Xs0
```

Decrypted traffic analysis on the Cisco WSA

1. From the web UI, navigate to **Help and Support > Packet Capture**.
2. Edit the packet capture settings, such as the network interface on which the packet capture runs, as required. Use one of the predefined filters, or create a custom filter with the use of any syntax that is supported by the Unix tcpdump command.
3. Click **Start Capture** to begin.
4. Click **Stop Capture** to end the capture.
5. Download the packet capture.

Additional Considerations

There are a few other configuration concepts that may need exploration when integrating a Cisco WSA device with SSL Orchestrator.

Creating service networks manually

As noted, the security best practice is to move security devices to SSL Orchestrator's protected network enclave. When that is done, SSL Orchestrator provides a set of internal network addresses that the security service can use. (This is the **Auto Manage** setting in the service definition.) Organizations that choose not to heed security best practices or otherwise need to create alternate addressing for the inline security service should follow the procedure below.

1. In defining the service, make sure **Auto Manage** is *not* selected.
2. Alter the **To Service** configurations by clicking **Create New** and providing a unique local **Name** (for example, **WSA_in**).
3. Under **VLAN**, click **Create New** and:

- Select the correct inbound-side interface (F5 to service).
- Enter a VLAN tag value as required.

4. Under **Network Settings**:

- Enter the SSL Orchestrator inbound-side self-IP (ex. 198.19.96.7)
- Enter the corresponding Netmask (ex. 255.255.255.128)

5. Alter the **From Service** configuration by selecting **Create New**.
6. Provide a unique local **Name** (for example, **WSA_out**).
7. Under VLAN, click **Create New** and:

- Select the correct inbound-side interface (service to F5).
- Enter a VLAN tag value as required.

8. Under **Network Settings**:

- Enter the SSL Orchestrator outbound-side self-IP (ex. 198.19.96.245).
- Enter the corresponding **Netmask** (ex. 255.255.255.128).

All other **To Service** and **From Service** settings should be the same as previously described in this guide.

Configuring external access for WSA DNS and engine updates

Interception rules are designed to process client-server traffic through SSL Orchestrator, and defined security services are (by best practice) isolated within the protected enclave created by SSL Orchestrator. Client-server traffic is signaled to allow SSL Orchestrator to track it through the service chains, and any traffic not signaled is blocked. This can be an issue when a security service itself needs to communicate with external resources, as this traffic would not carry a signal. For example, the Cisco WSA device may need to talk to a DNS server or reach out to subscription, licensing, and signature update services for updates. To enable this traffic flow, separate interception rule listeners can be defined to allow specific traffic from one of the service's IP addresses to go to specific destinations. In the following example, an interception rule listener is created to allow the Cisco WSA to communicate with external DNS (8.8.8.8).

1. Create a new L3 outbound **Topology** with the following properties:

- Provide a **Name** (for example, **wsa_outbound_dns**).
- **Protocol**: UDP
- **IP Family**: IPv4
- **Topology**: L3 Outbound

2. Click **Save & Next**.
3. No new services are required in the **Service List**, so click **Save & Next**.
4. No new **Service Chains** are required, so click **Save & Next**.
5. This **Security Policy** does not need any additional processing or categorization. We therefore recommend removing all rules except for the base **All Traffic** rule. Click **Save & Next**.
6. The **Interception Rule** for the service's outbound channel should be as specific as possible. For example, if an intercept rule were configured only with a destination IP, any legitimate client-side traffic to that IP address would incorrectly flow out through this service channel. The minimal recommendation is to ensure the service's outbound interception rule includes the service's source address, which would normally be the outbound-side interface IP. To do this:

- Select the **Custom** outbound rule type and enter a **Label** that describes this service outbound channel.
- Under **Source Address**, enter the outbound-side interface IP of the service, including its /32 CICR mask (for example, 198.19.0.138/32).
- Under **Destination Address/Mask**, enter the target service. In this case, the proxy needs to access Internet DNS at 8.8.8.8/32.
- Enter a **Port**. DNS is typically on port 53.
- Select the service's outbound-side **VLAN**.

7. Click **Save & Next**.
8. Under Egress Settings, define the topology-specific egress characteristics:

- Under **Manage SNAT**, define if and how source NAT (SNAT) should be used for egress traffic.
- Under **Gateways**, define the next hop route for traffic. For an outbound configuration, this is usually a next hop upstream router.

9. Click **Save & Next**.
10. On the **Summary** page, review the expandable list of all of the workflow-configured objects. Expand or edit the details as previously described.
11. When the defined settings are correct, click **Deploy**.

Note: While it is possible to pass this outbound service traffic through an SSL Orchestrator inspection zone, do not attempt to send the service-initiated traffic through a service chain that contains itself.

Configuring pass-through transparent proxy authentication to WSA

Cisco WSA provides transparent proxy authentication via an HTTP redirect mechanism to an on-box authentication service, which supports HTTP Basic authentication to LDAP, and Kerberos, NTLM, and Basic authentication to Active Directory. The mechanism normally works by redirecting the user to an authentication service URL on the WSA, where the user authenticates and is then redirected back through the transparent proxy. Session persistence can be maintained via IP address or a domain-based session cookie.

A summary of the transparent proxy authentication traffic flow:

Flow	Example
The client issues a normal HTTP request through the transparent proxy.	GET /foo Host: www.example.com
In the absence of an authenticated session, the proxy redirects the user to a separate authentication service URL.	HTTP/1.1 307 Proxy Redirect Location: https:// wsa.auth.url/<UID>/<IP>/<Original-URL>
The client is challenged and presents credentials to the authentication service.	<401-based authentication> Basic, NTLM, or Kerberos
If the client IP address is defined for session persistence (surrogacy), the authentication service simply redirects to the original URL, which flows through the transparent proxy.	GET /foo Host: www.example.com
If a session cookie is defined for surrogacy, the redirect from the authentication service contains a query string (iptac). The client redirects back through the transparent proxy with the iptac query string. The proxy redirects the client back to the original URL and sets a domain cookie based on the iptac value. The client redirects back to the original URL and sends the iptac cookie for all subsequent requests.	HTTP/1.1 307 Proxy Redirect Location: https://www.example.com/iptac-<ID> GET /iptac-<ID> Host: www.example.com HTTP/1.1 307 Proxy Redirect Location: https://www.example.com/foo Set-Cookie: iptac=<ID>, domain=.example.com GET /foo Host: www.example.com Cookie: <ID>

This logical flow presents two challenges when WSA is inside the SSL Orchestrator inspection zone:

- All traffic to the WSA is unencrypted HTTP, therefore the original URL in the authentication service redirect always includes an **HTTP://** URL. To address this challenge, a simple F5 iRule is required to rewrite the Original-URL value in the authentication service redirect as it leaves the WSA.

- The authentication service URL is defined within the WSA configuration and requires DNS to resolve to an IP. However, the IP address of the WSA inside the SSL Orchestrator inspection is non-routable and inaccessible to the client. To address this challenge, a separate F5 virtual server and pool are required to direct traffic to the WSA authentication service. Authentication service traffic should DNS resolve to this client-accessible IP on the F5, which then passes the traffic to the internal service.

The iRule and other configuration steps are detailed below.

WSA: Defining a WSA transparent proxy authentication realm

This guide does not expand on all of the different ways to configure transparent proxy authentication on Cisco WSA. Please see the official Cisco documentation for detailed options. The following steps approximate the configuration of an authentication realm:

1. Set up DNS to point to the Active Directory server regardless of the authentication type (Basic, Kerberos, or NTLM). The WSA must be able to access an Active Directory domain controller.
2. In the WSA UI, navigate to **Network > Authentication**. Click **Add Realm…** and configure authentication accordingly.
3. Click **Submit**.
4. Under **Network > Authentication**, click **Edit Global Settings…** and, at minimum, set the **Redirect Hostname** value. This is the URL that the transparent proxy will use to redirect to the authentication service. This address will need to be added to internal DNS to point to the F5 virtual server. Do not enable HTTPS for authentication, as this will be handled by the F5 virtual server.
5. Click **Submit**.
6. **Commit** the changes.

WSA: Add or update the WSA Identification Profile to enable transparent proxy authentication

To enable transparent proxy authentication, an **Identification Profile** must be defined accordingly.

1. In the WSA UI, navigate to **Web Security Manager > Identification Profiles**. **Add** a new identification profile, or simply edit the Global Identification Profile.
2. Under **Identification and Authentication**, select **Authenticate Users**.
3. Under **Authentication Realm**, select the previously created authentication realm.
4. Under **Authentication Surrogates**, select either **IP Address or Session Cookie**.

- **IP Address** manages client session persistence (surrogacy) by virtue of the client source address. Once authenticated, the client is not required to authenticate again until the session times out. Keep in mind, however, that IP address surrogacy is not effective in a NAT environment, where all clients can be coming to the proxy from the same source address. Otherwise, IP address is the more efficient of the two surrogacy options.

- **Session Cookie** manages client session persistence by virtue of domain cookies. The process is more completely described earlier, but essentially, for each new domain accessed (for example, *.google.com), the client is redirected to the authentication service, authenticates, is redirected back to the original host with an iptac query string, and then redirected again to the original full URL with a domain cookie. This method is more useful in NAT environments, but is considerably less efficient as it requires multiple redirects.

5. Submit and commit changes.

SSL Orchestrator: Create a pool to the WSA authentication service

With HTTPS authentication disabled in the WSA authentication realm settings, traffic to the authentication service defaults to port 80.

Note: The authentication service pool *must* point to the same IP addresses defined in the SSL Orchestrator service, but on port 80.

SSL Orchestrator: Create a client SSL profile for the WSA authentication service

Authentication traffic should still flow across an encrypted channel, so client SSL must be configured on the F5 system to decrypt to the WSA. The certificate and private key defined in the F5 client SSL profile should match the URL defined in the WSA authentication realm (**Redirect Hostname**).

SSL Orchestrator: Create a source address persistence profile

To enable authentication service requests and regular decrypted traffic to flow to the same WSA service for a single user, a persistence profile is required that maps the source address of the client. Create a source address affinity persistence profile, with **Match Across Virtual Servers** enabled.

SSL Orchestrator: Create a virtual server to the WSA authentication service

The F5 virtual server must minimally include:

- **Destination Address/Mask**—an IP address that matches the DNS resolution of the **Redirect Hostname** value.
- **Port**—443.
- **Client SSL profile**—the previously created client SSL profile.
- **VLANs and Tunnels**—the client-side VLAN.
- **Address Translation** —enabled.
- **Port Translation**—enabled.
- **Default Pool**—the previously created pool.
- **Default Persistent Profile**—the previously created source address affinity profile.

SSL Orchestrator: Create an iRule for the WSA authentication redirects

The iRule must be attached to the WSA service to catch and rewrite the authentication redirect responses coming from the WSA service.

1. In the SSL Orchestrator UI, under **Services**, click to edit the WSA service.
2. Under **Resources** at the bottom of the WSA service properties page, select the created iRule, then click **Save & Next**.
3. **Deploy** to save this configuration.
4. Edit the **WSATP** static variable in the RULE_INIT event to mirror the DNS name of the WSA authentication service.

```
when RULE_INIT {
  set static::WSATP "wsa.labs.f5demo.com"
}
when HTTP_RESPONSE {
  if { ( [HTTP::is_redirect] ) and ( [HTTP::header Location] starts_with "http://${static:::WSATP}"
) } {
    sharedvar ctx
    if { ( [info exists ctx(ptcl)] ) and ( $ctx(ptcl) eq "https" ) } {
      ## Handle HTTPS requests - first replace HTTP:// in auth URL with HTTPS
      set url [string map [list "http://${static:::WSATP}" "https://${static:::WSATP}"]
[HTTP::header Location]]
      ## Determine if WSA is sending the auth URL with a port value
      set hasPort [findstr ${url} "https://${static:::WSATP}" [string length
"https://${static:::WSATP}"] "/"]
      ## If the redirect contains a port, strip it off
      if { ${hasPort} ne "" } {
        set url [string map [list "http://${static:::WSATP}${hasPort}" "https://${static:::WSATP}"]
[HTTP::header Location]]
      }
      ## Replace HTTP:// with HTTPS:// in the original request URL
      set url [string map [list "http://$ctx(SNI)" "https://$ctx(SNI)"] ${url}]
      ## Replace the outbound Location redirect header
      HTTP::header replace Location ${url}
    } elseif { ( [info exists ctx(ptcl)] ) and ( $ctx(ptcl) eq "http" ) } {
      ## Handle HTTP request - first replace HTTP:// in auth URL with HTTPS
      set url [string map [list "http://${static:::WSATP}" "https://${static:::WSATP}"]
[HTTP::header Location]]
      ## Determine if WSA is sending the auth URL with a port value
      set hasPort [findstr ${url} "https://${static:::WSATP}" [string length
"https://${static:::WSATP}"] "/"]
      ## If the redirect contains a port, strip it off
      if { ${hasPort} ne "" } {
        set url [string map [list "http://${static:::WSATP}${hasPort}" "https://${static:::WSATP}"]
[HTTP::header Location]]
      }
      ## Replace the outbound Location redirect header
      HTTP::header replace Location ${url}
    }
  }
}
```

SSL Orchestrator: Attach the new iRule to the WSA service

1. Navigate to the SSL Orchestrator **Services** menu and click the WSA service.
2. Click the pencil icon to the right of the service to edit it.
3. At the bottom of the WSA service configuration, move the new authentication rule to the **Selected** column.
4. Click **Save & Next**.
5. Under **Service Chain**, click **Save & Next** again, then click **Deploy**.