F5 Networks

SSL Orchestrator 5.3 (14.1)

Lab Guide (Ravello Edition)

Northeast Region Customer Hands on Events

Participant Lab Guide

Solutions for
an application world.

Last Updated: *05.2019*

Table of Contents

WHAT IS THE F5 SSL ORCHESTRATOR?

F5 SSL Orchestrator (SSLO) provides an all-in-one appliance solution designed specifically to optimize the SSL infrastructure, provide security devices with visibility of SSL/TLS encrypted traffic, and maximize efficient use of that existing security investment. This solution supports policy-based management and steering of traffic flows to existing security devices, designed to easily integrate into existing architectures, and centralizes the SSL decrypt/encrypt function by delivering the latest SSL encryption technologies across the entire security infrastructure.

Multi-layered security
In order 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 (WAF), Intrusion Prevention and 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 allows 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

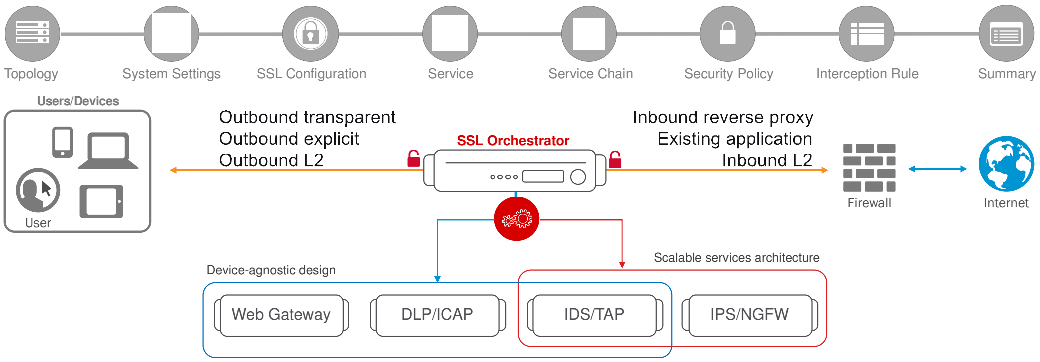
Different environments call for different network implementations. While some can easily support SSL visibility at layer 3 (routed), others may require these devices to be inserted at layer 2. SSL Orchestrator can support all of these networking requirements with the following topology options:

- Outbound transparent proxy
- Outbound explicit proxy
- Outbound layer 2
- Inbound reverse proxy
- Existing application
- Inbound layer 2

Security Policy

The SSLO 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 the following:

- Source and destination address/subnet
- URL filtering and IP intelligence - Subscriptions
- Host and domain name
- Destination port
- IP geolocation
- Protocol



SSL ORCHESTRATOR LAB ENVIRONMENT

The lab environment for this guide has provided some prerequisite settings that you should be aware of. These are provided to make the demo simpler. All of the following would need to be configured manually in another environment.

- **Client side VLAN and subnet are defined** – this is the VLAN that an internal client connects to for outbound traffic flows. SSLO does not define the client-side VLAN(s) and self-IP(s). A web server also exists on the client side VLAN to facilitate an inbound (reverse proxy) use case – external client to an internal set of websites.

- **Outbound side VLAN and subnet are defined** – this is the VLAN that traffic egresses from SSLO to the Internet gateway. SSLO does not define the server-side VLAN(s) and self-IP(s).

- **ICAP service VLAN and subnet are defined** – SSLO does not define the networking for this service type, so it has been pre-created in this lab.

- **CA certificate and private key are installed** – this is the CA certificate and private key that are used to re-issue (forge) remote server certificates to internal clients for outbound traffic flows.

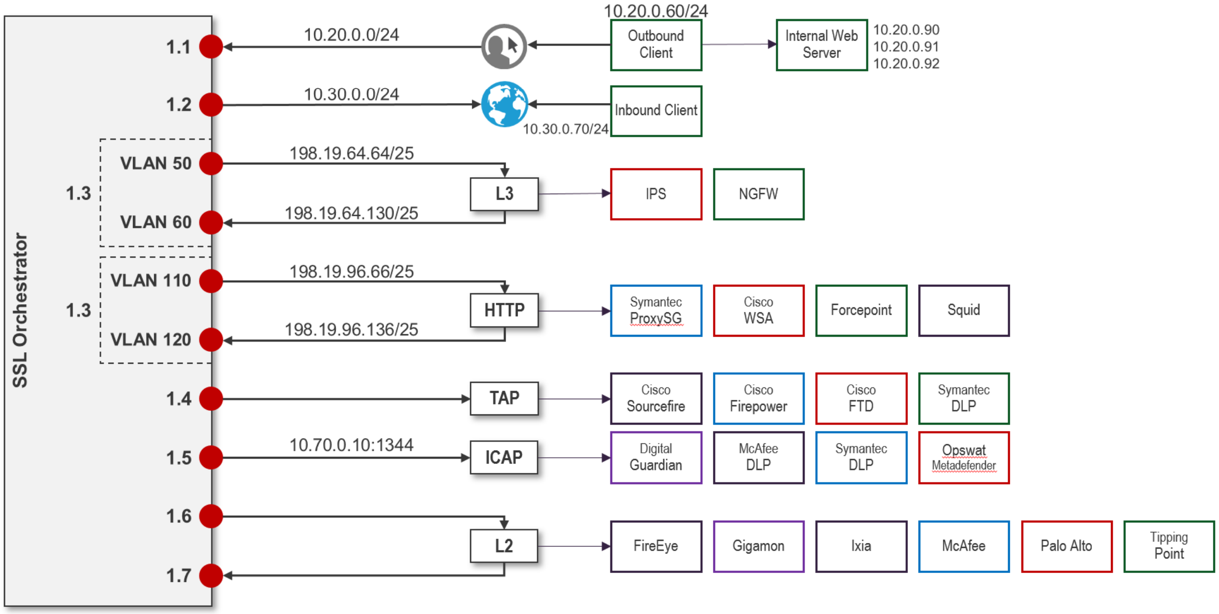
- **Server certificate and private key are installed** – for the inbound (reverse proxy) traffic flow use case, SSL traffic is terminated at the F5, and re-encrypted on the way to the internal application environment. A wildcard server certificate is installed to facilitate using any name under the "*.f5labs.com*" sub-domain.

Note: It is a security best practice to isolate security devices within the protected network enclaves provided by SSLO. You will often desire NOT to move or change existing security services. However, while possible with SSLO 4.0 and beyond, passing this decrypted traffic to points on an existing network architecture could create a provide multiple points of data exposure. Usernames, passwords, credit card numbers and other sensitive information could be exposed to other devices on that network. Each inline layer 3 security service definition includes an "Auto Manage" option. This option, enabled by default, provides internal network settings for security services to use, so that only the interface (and 802.1q VLAN tag as needed) is required to be defined for the inbound and outbound interfaces. Should customers opt to not follow security best practices, or simply need different networking settings, you can disable the Auto Manage option and define all the required inbound and outbound networking setting manually.

The following is a visual representation of this Ravello lab environment. The numbers inside the right edge of the SSL Orchestrator box indicate the port numbers assigned. The colored boxes to the right of the services indicate a few product examples of each respective service type.

SSL Orchestrator

Lab architecture (Ravello)



Note: ALL CONFIGURATION TASKS/ LABS BELOW ARE TO BE DONE USING RDP TO "OUTBOUND WIN 7 CLIENT" unless specifically asked to RDP to another host! YOUR PERSONAL DEVICE IS ONLY USED FOR RDP. YOUR INSTRUCTOR WILL PROVIDE THE DETAILS TO CONNECT/RDP TO THE HOST/S. Accept and Proceed through all warnings to connect.

Once you have the Remote Desktop established. Launch the Chrome Browser on the RDP host and connect to the SSLO WebUI at https://10.10.0.110 or use the bookmarked shortcut to start with the lab 1 below. Accept and Proceed through Certificate Warnings.

LAB 1 – CREATE A TRANSPARENT FORWARD PROXY SSLO

The majority of enterprise forward proxy configurations will involve a single F5 platform performing the SSL visibility task. The SSL Orchestrator has been designed with that principle in mind and performs robust security service chaining of security devices attached to a single appliance. SSL Orchestrator 5.0 now makes configuration of a single-box deployment simple and intuitive. Please follow the steps below to create a transparent forward proxy SSL Orchestrator configuration.

Step 1: Review the lab environment and map out the services and endpoints

Review the "SSL Orchestrator Lab Environment" section above or in the handout. This lab will attach one of each type of security service (HTTP, ICAP, L2, L3, TAP) to SSLO for an outbound forward proxy traffic flow. Afterwards, an internal client will be able to access remote (Internet) resources through SSLO, providing decrypted, inspectable traffic to the security services.

- The client is attached to a *10.20.0.0/24* network and is assigned the IP *10.20.0.60*. This network is attached to the BIG-IP 1.1 interface.

- The **L2 device** is an Ubuntu 14.04 LTS server configured to bridge its eth1 and eth2 interfaces. Its inbound VLAN (traffic to it) is attached to the BIG-IP *1.6* interface. Its outbound interface (traffic coming from it) is attached to the BIG-IP *1.7* interface.

- The **L3 device** is an Ubuntu 14.04 LTS server configured to route between its eth1.10 and eth1.20 (tagged) interfaces. Its inbound VLAN (traffic to it) is attached to the BIG-IP *1.3 (VLAN tag 50)* interface and has an IP of *198.19.64.64/25*. Its outbound interface (traffic coming from it) is attached to the BIG-IP *1.3 (VLAN tag 60)* interface and has an IP of *198.19.64.130/25*. Its default gateway is *198.19.64.245*, which will be a VLAN self-IP on the BIG-IP.

- The **TAP** device is an Ubuntu 14.04 LTS server configured with a single eth1 interface. That interface is attached to the BIG-IP *1.4* interface.

- The **DLP/ICAP** device is an Ubuntu 14.04 LTS server configured with a single eth1 interface. That interface is attached to the BIG-IP *1.5* interface and has an IP of *10.70.0.10 and listening on port 1344*. The box is running c-icap and Squid/Clamav.

- The **Explicit Proxy device** is an Ubuntu 14.04 LTS server configured with Squid. Its interfaces are eth1.30 and eth1.40 (tagged). Its inbound VLAN (traffic to it) is attached to the BIG-IP *1.3 (VLAN tag 110)* interface and has an IP of *198.19.96.66/25*. Its outbound interface (traffic coming from it) is attached to the BIG-IP *1.3 (VLAN tag 120)* interface and has an IP of *198.19.96.136/25*. Its default gateway is *198.19.96.245*, which will be a VLAN self-IP on the BIG-IP.

- The outbound network is attached to the BIG-IP *1.2* interface, in the *10.30.0.0/24* subnet, and has a gateway of *10.30.0.1*.

- **In the lab, client inbound, Internet outbound, and DLP VLANs and self-IPs are already created**.

Step 2: Fulfill the SSL Orchestrator prerequisites

There are a number of objects that SSL Orchestrator does not create and expects to exist before deploying the iApp. You must create the following objects before starting the iApp:

- **Import the CA certificate and private key** – in order to terminate and re-encrypt outbound SSL traffic, SSL Forward Proxy must re-issue, or rather "forge" a new server certificate to the client. In order to perform this re-issuance process, the BIG-IP must possess a certificate authority (CA) certificate and associated private key. *This lab environment already has a subordinate CA certificate and private key installed*.

- **Create the client inbound VLAN and self-IP** – create the VLAN and self-IP that connects the client to the BIG-IP. In this lab that's the *10.20.0.0/24* subnet and interface *1.1* on the BIG-IP. This lab environment already has this VLAN and self-IP created.

- **Create the Internet outbound VLAN and self-IP** – create the VLAN and self-IP that connects the BIG-IP to the outbound Internet router. In this lab that's the *10.30.0.0/24* subnet and interface *1.2* on the BIG-IP. *This lab environment already has this VLAN and self-IP created*.

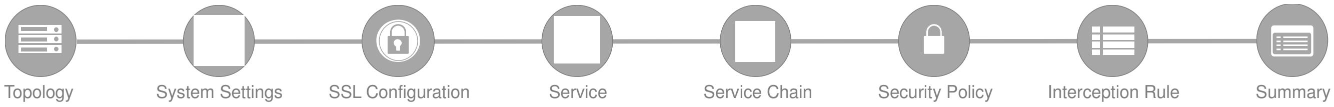
- **Create the DLP VLAN and self-IP** – if it is desired to isolate the DLP/ICAP device, create the VLAN and self-IP that connects the DLP device to the BIG-IP. In this lab that's the *10.70.0.0/24* subnet and interface *1.5* on the BIG-IP. The DLP security device is listening on *10.70.0.10* and ICAP is listening on port *1344*. *This lab environment already has this VLAN and self-IP created*.

- **Create the default internet route for outbound traffic** – the iApp provides an option to leverage a defined gateway pool or use the system default route. If a gateway pool is not used, they system route table will need to have a default route used to reach Internet destination. *We'll use a gateway pool defined within SSLO*.

As a general rule, avoid using names with dashes (ex. sslo-demo-1) while creating objects in SSL Orchestrator. Underscores (ex. sslo_demo_1) and camel-casing (ex. ssloDemo1) are preferred.

Step 3: Create the SSL Orchestrator deployment through Guided Configuration

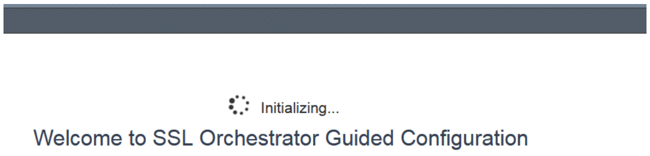
The SSL Orchestrator Guided Configuration presents a completely new and streamlined user experience. This workflow-based architecture provides intuitive, re-entrant configuration steps tailored to the selected topology.



The following steps will walk through the Guided Configuration (GC) to build a simple transparent forward proxy.

The following provides verbose details on each setting.

- **Initialization** – if this is the first time accessing SSLO in a new BIG-IP build, upon first access, GC will automatically load and deploy the built-in SSLO package. Click on **SSL Orchestrator>>Configuration.** In our labs the package is installed so it will initialize.

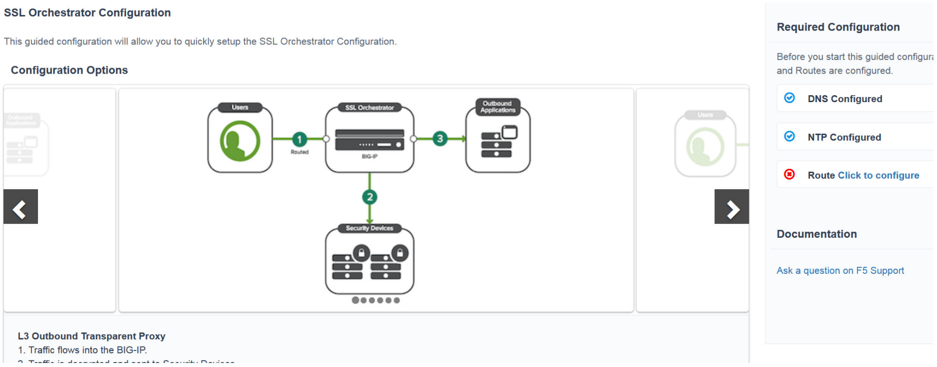


Initializing...

Welcome to SSL Orchestrator Guided Configuration

- **Configuration review and prerequisites** – take a moment to review the topology options and workflow configuration steps involved. No other configurations are required on this page, so click Next.

DNS and NTP device settings have already been defined in this lab.



SSL Orchestrator Configuration

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

Configuration Options

L3 Outbound Transparent Proxy

1. Traffic flows into the BIG-IP.

Required Configuration

Before you start this guided configura
and Routes are configured.

- DNS Configured
- NTP Configured
- Route Click to configure

Documentation

Ask a question on F5 Support

- **Topology Properties** – SSLO now creates discreet configurations based on the selected topology. For example, in previous versions of SSLO, a transparent and explicit forward proxy might be defined together. In SSLO 5.0, these are configured separately. 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 provides 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, 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:

  - **L3 Explicit Proxy** – this is the traditional explicit forward proxy.

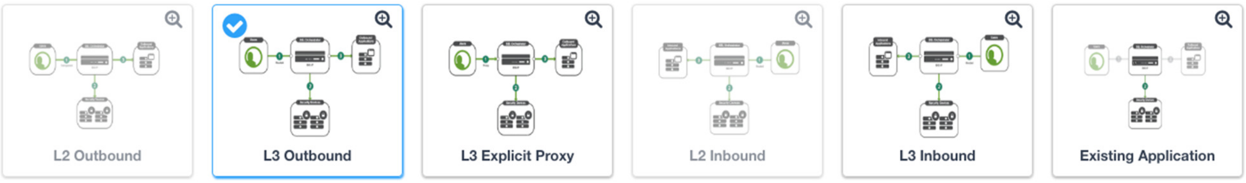
  - **L3 Outbound** – this is the traditional transparent forward proxy.

  - **L3 Inbound** – this is a reverse proxy “gateway” configuration. In its simplest form, this topology builds an SSLO environment designed to sit **in front of** another ADC or routed path. Advanced options allow it to define a pool for more directed traffic flow, but alone does not provide the same flexibility afforded a typical LTM reverse proxy virtual server. It also must perform re-encryption on egress. The primary use case for this topology is as a gateway SSL visibility solution, potentially sitting at a boundary edge in front of multiple internal ADC environments.

  - **L2 Inbound** – the layer 2 topology options insert SSLO as a bump-in-the-wire in an existing routed path, where SSLO 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 SSLO as a bump-in-the-wire in an existing routed path, where SSLO 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 LTM applications. Whereas the L3 Inbound topology provides an inbound gateway function for SSLO, Existing Application works with LTM virtual servers that already perform their own SSL handling and client-server traffic management. The Existing Application workflow proceeds directly to service creation and security policy definition, then exits with an SSLO-type access policy and per-request policy that can easily be consumed by an LTM virtual server.

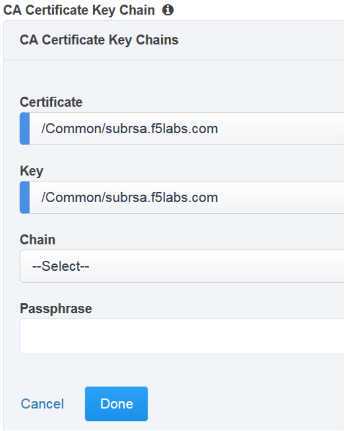


For this lab,

- **Name**: some name (ex. “**demo**”)
- **Protocol**: **Any** – this will create separate TCP, UDP and non-TCP/UDP interception rules. Make sure to pick ANY for Protocol.
- **IP Family**: IPv4
- **Topology**: L3 Outbound (Click on the icon)
- 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. For this lab, create a new SSL profile,
  - **Client-side SSL**
    - **Cipher Type** – cipher type can be a Cipher Group or Cipher String. If the former, select a previously-defined cipher group (from Local Traffic – Ciphers – Groups). If the latter, enter a cipher string that appropriately represents the client-side TLS requirement. For most environments, DEFAULT is optimal. For this lab, leave Cipher String selected.
    - **Certificate Key Chain** – the certificate key chain represents the certificate and private key used as the "template" for forged server certificates. While re-issuing 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 customers the opportunity to apply their own template private key, and optionally 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 BIG-IP system is installed. The pre-defined default.crt and default.key can be left as is.
    - **CA Certificate Key Chain** – an SSL forward proxy must re-sign, or "forge" remote server certificate to local clients using a local certificate authority (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. Click the pencil icon to Edit, then select subrsa.f5labs.com for both Certificate and Key, and click Done.

Make sure you only picked the subrsa.f5labs.com cert and key for the CA Certificate and Key Chain Section only as shown below


CA Certificate Key Chain

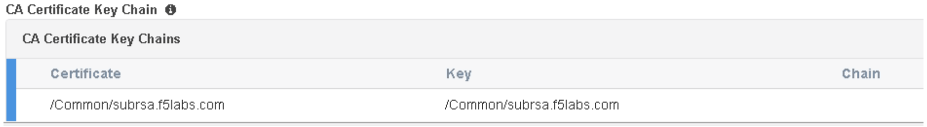
CA Certificate Key Chains

Certificate: /Common/subrsa.f5labs.com

Key: /Common/subrsa.f5labs.com

Chain: --Select--

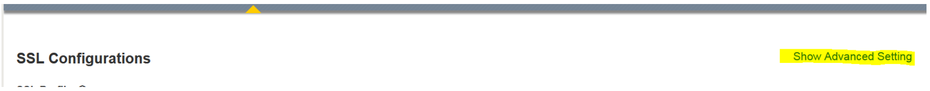
Passphrase:


CA Certificate Key Chain

CA Certificate Key Chains

Certificate	Key	Chain
/Common/subrsa.f5labs.com	/Common/subrsa.f5labs.com	

Click “Show Advanced Setting” towards the top right to see below options.


SSL Configurations

- **[Advanced] Bypass on Handshake Alert** – this setting allows the underlying SSL Forward Proxy process to bypass SSL decryption if an SSL handshake error is detected on the server side. It is recommended to leave this disabled (unchecked).

- **[Advanced] Bypass on Client Certificate Failure** – this 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. It is recommended to leave this disabled (unchecked).

The above two Bypass options can create a security vulnerability. If a colluding client and server can force an SSL handshake error, or force client certificate authentication, they can effectively bypass SSL inspection. It is recommended that these settings be left disabled.

- **Server-side SSL**

  - **Cipher Type** – cipher type can be a Cipher Group or Cipher String. If the former, select a previously-defined cipher group (from Local Traffic – Ciphers – Groups). If the latter, enter a cipher string that appropriately represents the server-side TLS requirement. For most environments, 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 SSL visibility product now performs all server-side certificate validation, in lieu of the client browser, and should therefore do its best to maintain the same industry security trends. BIG-IP ships with a CA certificate bundle that maintains a list of CA certificates common to the browser vendors. However, a more comprehensive bundle can be obtained from the F5 Downloads site. For this lab, the built-in ca-bundle.crt is already selected.

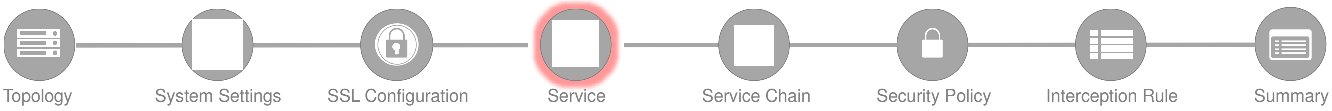
- **[Advanced] Expire Certificate Response** – SSLO performs validation on remote server certificates and can control what happens if it receives an expired server certificate. The options are **drop**, which simply drops the traffic, and **ignore**, which mirrors an expired forged certificate to the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

- **[Advanced] Untrusted Certificate Authority** – SSLO 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 forward proxy is to drop traffic on an untrusted certificate.

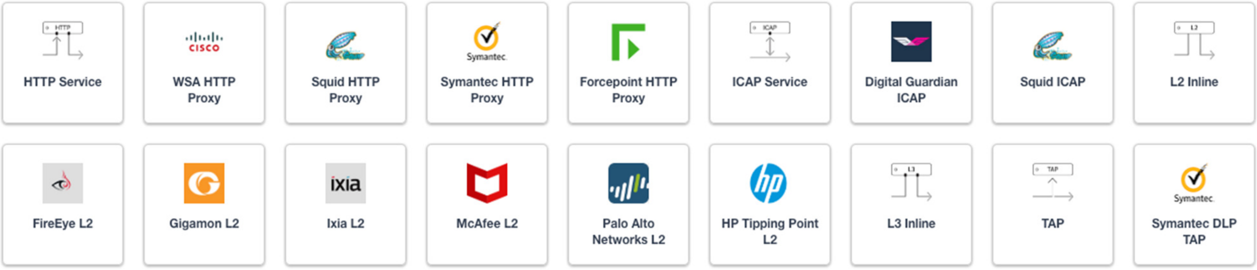
- **[Advanced] OCSP** – this setting selects an existing or can create a new OCSP profile for server-side Online Certificate Status Protocol (OCSP) and OCSP stapling. With this enabled, if a client issues a Status_Request message in its ClientHello message (an indication that it supports OCSP stapling), SSLO will issue a corresponding Status_Request message in its server-side TLS handshake. SSLO 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, SSLO 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 or can create a new CRL profile for server-side Certificate Revocation List (CRL) validation. With this enabled, SSLO attempts to match server certificates to locally-cached CRLs.

○ Click Save & Next.



- **Services List** – the Services List page is used to define security services that attach to SSLO. The 5.0 SSLO Guided Configuration now 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 “generic” security services. Depending on screen resolution, it may be necessary to scroll down to see additional services.



This lab will create one of each type of security service. Click Add Service, then either select a service from the catalog and click Add, or simply double-click the service to go to its configuration page.

- **Inline layer 2 service** – select the FireEye Inline Layer 2 service from the catalog and click Add, or simply double-click the FireEye Inline Layer 2 service, or any other Inline Layer 2 service in the catalog.

  - **Name** – provide a unique name to this service (example "FireEye").

  - **Network Configuration** – paths define the network interfaces that take inspectable traffic to the inline service and receive traffic from the service. Click Add.

    - **Ratio** – inline security services are natively load balanced, so this setting defines a ratio, if any for the load balanced pool members. Enter 1.

    - **From BIGIP VLAN** – this is the interface taking traffic to the inline service. Select the Create New option, enter a unique name (ex. FireEye_in), select the F5 interface connecting to the inbound side of the service, and add a VLAN tag value if required. For this lab, select interface 1.6.

    - **To BIGIP VLAN** – this is the interface receiving traffic from the inline service. Select the Create New option, enter a unique name (ex. FireEye_out), select the F5 interface connecting to the outbound side of the service, and add a VLAN tag value if required. For this lab, select interface 1.7.

    - Click Done.

  - **Service Action Down** – SSLO also natively monitors the load balanced pool of security devices, and if all pool members fail, can actively bypass this service (**Ignore**), or stop all traffic (**Reset**, **Drop**). For this lab, leave it set to Ignore.

  - **Enable Port Remap** – this setting allows SSLO to remap the port of HTTPS traffic flowing across this service. This is advantageous when a security service defines port 443 traffic as encrypted HTTPS and natively ignores it. By remapping HTTPS traffic to, say, port 8080, the security service will inspect the traffic. For this lab, enable (check) this option and enter a port value 8080.

  - **iRules** – SSLO now 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 these iRules must not be used to control traffic flow (ex.

pools, nodes, virtuals, etc.), but rather should be used to view/modify application layer protocol traffic. For example, an iRule assigned here could be used to view and modify HTTP traffic flowing to/from the service. Additional iRules are not required, however, so leave this empty.

    - Click Save then Click Add Service to add the next service

- **Inline layer 3 service** – select the Generic Inline Layer 3 service from the catalog and click Add, or simply double-click the Generic Inline Layer 3 service.

    - **Name** – provide a unique name to this service (example "IPS").

    - **IP Family** – this setting defines the IP family used with this layer 3 service. Leave it set to IPv4.

    - **Auto Manage Addresses** – when enabled the Auto Manage Addresses 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. It is recommended to leave this option enabled (checked).

    - **To Service Configuration** – the "To Service" defines the network connectivity from SSLO to the inline security device.

        - **To Service** – with the Auto Manage Addresses option enabled, this IP address will be pre-defined, therefore the inbound side of the service must match this IP subnet. With the Auto Manage Addresses option disabled, the IP address must be defined manually. For this lab, leave the 198.19.64.7/25 address intact.

        - **VLAN** – select the Create New option, provide a unique name (ex. IPS_in), select the F5 interface connecting to the inbound side of the service, and add a VLAN tag value if required. For this lab, select interface 1.3 and VLAN tag 50.

    - **Service Down Action** – SSLO also natively monitors the load balanced pool of security devices, and if all pool members fail, can actively bypass this service (**Ignore**), or stop all traffic (**Reset**, **Drop**). For this lab, leave it set to Ignore.

    - **Security Devices >> L3 Devices** – this defines the inbound-side IP address of the inline layer 3 service, used for routing traffic to this device. Multiple load balanced IP addresses can be defined here. Click Add, enter 198.19.64.64, then click Done.

    - **From Service Configuration** – the "From Service" defines the network connectivity from the inline security device to SSLO.

        - **From Service** – with the Auto Manage Addresses option enabled, this IP address will be pre-defined, therefore the outbound side of the service must match this IP subnet. With the Auto Manage Addresses option disabled, the IP address must be defined manually. For this lab, leave the 198.19.64.245/25 address intact.

        - **VLAN** – select the Create New option, provide a unique name (ex. IPS_out), select the F5 interface connecting to the outbound side of the service, and add a VLAN tag value if required. For this lab, select interface 1.3 and VLAN tag 60.

    - **Enable Port Remap** – this setting allows SSLO to remap the port of HTTPS traffic flowing across this service. This is advantageous when a security service defines port 443 traffic as encrypted HTTPS and natively ignores it. By remapping HTTPS traffic to, say, port 8181, the security service will inspect the traffic. For this lab, enable (check) this option and enter a port value value 8181.

    - **Manage SNAT Settings** – SSLO now defines an option to enable SNAT (source NAT) across an inline layer 3/HTTP service. The primary use case for this is horizontal SSLO scaling, where independent SSLO devices are scaled behind a separate load balancer but share the same inline layer 3/HTTP services. As these devices must route back to SSLO, there are now multiple SSLO devices to route back to. SNAT allows the layer 3/HTTP device to know which SSLO sent the packets for proper routing. SSLO scaling also requires that the Auto Manage option be disabled, to provide separate address spaces on each SSLO. For this, leave it set to None.

    - **iRules** – SSLO now 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 these iRules must not be used to control traffic flow (ex. pools, nodes, virtuals, etc.), but rather should be used to view/modify application layer protocol traffic. For example, an iRule assigned here could be used to view and modify HTTP traffic flowing to/from the service. Additional iRules are not required, however, so leave this empty.

    - Click Save then Click Add Service to add the next service

- **Inline HTTP service** – an inline HTTP service is defined as an explicit or transparent proxy for HTTP (web) traffic. Select the WSA HTTP Proxy service from the catalog and click Add, or simply double-click the WSA HTTP Proxy service, or any other HTTP Proxy service in the catalog.

    - **Name** – provide a unique name to this service (example "Proxy").

    - **IP Family** – this setting defines the IP family used with this layer 3 service. Leave it set to IPv4.

    - **Auto Manage Addresses** – when enabled the Auto Manage Addresses 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. It is recommended to leave this option enabled (checked).

    - **Proxy Type** – this defines the proxy mode that the inline HTTP service is in. For this lab, set this option to Explicit.

    - **To Service Configuration** – the "To Service" defines the network connectivity from SSLO to the inline security device.

  - **To Service** – with the Auto Manage Addresses option enabled, this IP address will be pre-defined, therefore the inbound side of the service must match this IP subnet. With the Auto Manage Addresses option disabled, the IP address must be defined manually. For this lab, leave the 198.19.96.7/25 address intact.

  - **VLAN** – select the Create New option, provide a unique name (ex. Proxy_in), select the F5 interface connecting to the inbound side of the service, and add a VLAN tag value if required. For this lab, select interface 1.3 and VLAN tag 110.

- **Service Down Action** – SSLO also natively monitors the load balanced pool of security devices, and if all pool members fail, can actively bypass this service (**Ignore**), or stop all traffic (**Reset**, **Drop**). For this lab, leave it set to Ignore.

- **Security Devices >> HTTP Proxy Devices** – this 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 198.19.96.66 for the IP Address, and 3128 for the Port, then click Done.

- **From Service Configuration** – the "From Service" defines the network connectivity from the inline security device to SSLO.

  - **From Service** – with the Auto Manage Addresses option enabled, this IP address will be pre-defined, therefore the outbound side of the service must match this IP subnet. With the Auto Manage Addresses option disabled, the IP address must be defined manually. For this lab, leave the 198.19.96.245/25 address intact.

  - **VLAN** – select the Create New option, provide a unique name (ex. Proxy_out), select the F5 interface connecting to the outbound side of the service, and add a VLAN tag value if required. For this lab, select interface 1.3 and VLAN tag 120.

- **Manage SNAT Settings** – SSLO now defines an option to enable SNAT (source NAT) across an inline layer 3/HTTP service. The primary use case for this is horizontal SSLO scaling, where independent SSLO devices are scaled behind a separate load balancer but share the same inline layer 3/HTTP services. As these devices must route back to SSLO, there are now multiple SSLO devices to route back to. SNAT allows the layer 3/HTTP device to know which SSLO sent the packets for proper routing. SSLO scaling also requires that the Auto Manage option be disabled, to provide separate address spaces on each SSLO. For this, leave it set to 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. For this lab, leave it disabled (unchecked).

- **iRules** – SSLO now 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 these iRules must not be used to control traffic flow (ex.

pools, nodes, virtuals, etc.), but rather should be used to view/modify application layer protocol traffic. For example, an iRule assigned here could be used to view and modify HTTP traffic flowing to/from the service. Additional iRules are not required, however, so leave this empty.

    - Click Save then Click Add Service to add the next service.

- **ICAP service** – an ICAP service is an RFC 3507-defined service that provides some set of services over the ICAP protocol. Select the Squid ICAP service from the catalog and click Add, or simply double-click the Squid ICAP service, or any other ICAP service in the catalog.

    - **Name** – provide a unique name to this service (example "ICAP").

    - **IP Family** – this setting defines the IP family used with this layer 3 service. Leave it set to IPv4. It will be grayed out.

    - **ICAP Devices** – this defines the IP address of the ICAP service, used for passing traffic to this device. Multiple load balanced IP addresses can be defined here. Click Add, enter 10.70.0.10 for the IP Address, and 1344 for the Port, and then click Done.

    - **ICAP Headers** – select either **Default** or **Custom** to specify additional ICAP headers. To add custom headers, select Custom, otherwise leave as Default.

    - **OneConnect** – the F5 OneConnect profile improves performance by reusing TCP connections to ICAP servers to process multiple transactions. If the ICAP servers do not support multiple ICAP transactions per TCP connection, do not enable this option. For this lab, leave the OneConnect setting enabled.

    - **Request URI Path** – this is the RFC 3507-defined URI request path to the ICAP service. Each ICAP security vendor will differ with respect to request and response URIs, and preview length, so it is important to review the vendor's documentation. In this lab, enter /squidclamav.

    - **Response URI Path** – this is the RFC 3507-defined URI response path to the ICAP service. Each ICAP security vendor will differ with respect to request and response URIs, and preview length, so it is important to review the vendor's documentation. In this lab, enter /squidclamav.

    - **Preview Max Length(bytes)** – this defines the maximum length of the ICAP preview. Each ICAP security vendor will differ with respect to request and response URIs, and preview length, so it is important to review the vendor's documentation. A zero-length preview length implies that data will be streamed to the ICAP service, similar to an HTTP 100/Expect process, while any positive integer preview length defines the amount of data (in bytes) that are transmitted first, before streaming the remaining content. The ICAP service in this lab environment does not support a complete stream, so requires a modest amount of initial preview. In this lab, enter 524288.

    - **Service Down Action** – SSLO also natively monitors the load balanced pool of security devices, and if all pool members fail, can actively bypass this service (**Ignore**), or stop all traffic (**Reset**, **Drop**). For this lab, leave it set to Ignore.

        - **HTTP Version** – this defines whether SSLO sends HTTP/1.1 or HTTP/1.0 requests to the ICAP service.

        - **ICAP Policy** – an ICAP policy is a pre-defined LTM CPM policy that can be configured to control access to the ICAP service based on attributes of the HTTP request or response. ICAP processing is enabled by default, so an ICAP CPM policy can be used to disable the request and/or response ADAPT profiles.

        - Click Save Click Add Service to add the next service

    - **TAP service** – a TAP service is a passive device that simply receives a copy of traffic. Select the Cisco Sourcefire TAP service from the catalog and click Add, or simply Double-click the Cisco Sourcefire TAP service, or any other TAP service in the catalog.

        - **Name** - provide a unique name to this service (example "TAP").

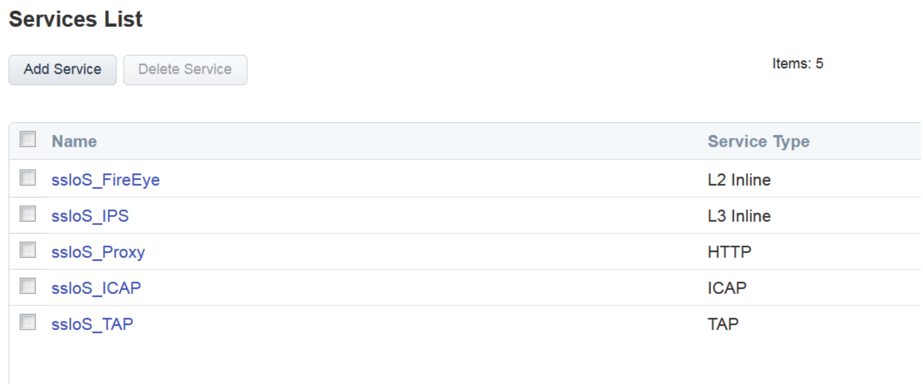
        - **Mac Address** – for a tap service that is not directly connected to the F5, enter the device's MAC address. For a tap service that is directly connected to the F5, the MAC address does not matter and can be arbitrarily defined. For this lab, enter 12:12:12:12:12:12.

        - **VLAN** – this defines the interface connecting the F5 to the TAP service. Click Create New and provide a unique name (ex. TAP_in).
        - **Interface** – select the 1.4 interface.

        - **Enable Port Remap** – this setting allows SSLO to remap the port of HTTPS traffic flowing to this service. For this lab, leave the option disabled (unchecked).

        - Click Save.

          You should have five services defined as below

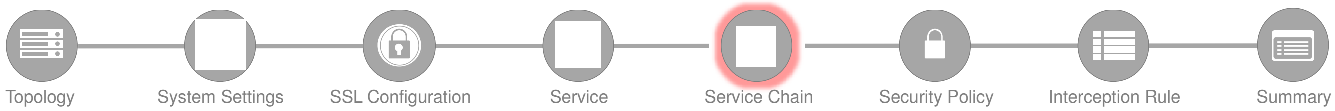


Services List

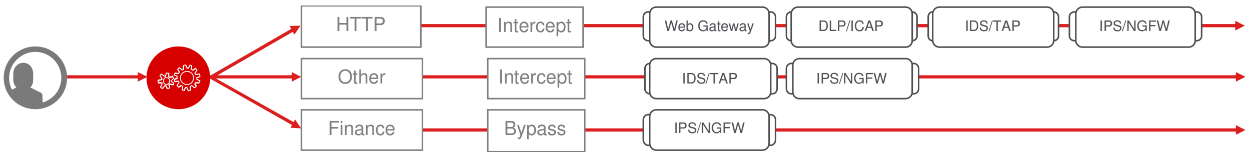
Items: 5

☐	Name	Service Type
☐	ssloS_FireEye	L2 Inline
☐	ssloS_IPS	L3 Inline
☐	ssloS_Proxy	HTTP
☐	ssloS_ICAP	ICAP
☐	ssloS_TAP	TAP

    - Click Save & Next.

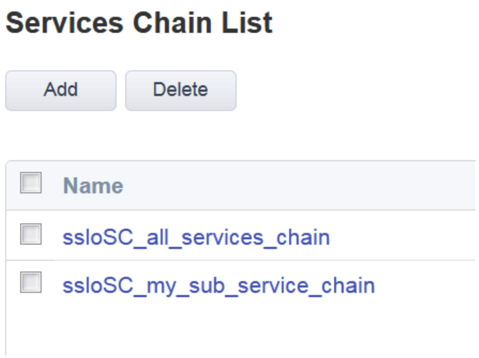


- **Service Chain List** – service chains are arbitrarily-ordered lists of security devices. Based on environmental requirements, different service chains may contain different re-used sets of services, and different types of traffic can be assigned to different service chains. For example, HTTP traffic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



  - Click Add to create a new service chain containing all of the security services.
    - **Name** – provide a unique name to this service (ex. “all_services_chain”).
    - **Services** – you can select any number of desired service and move them into the **Selected Service Chain Order** column, optionally also ordering them as required. In this lab, select all of the services.
    - Click Save.
  - Click Add to create a new service chain for just the L2 (ex. FireEye) and TAP services.
    - **Name** – provide a unique name to this service (ex. “my_sub_service_chain”).
    - **Services** – select the inline layer 2 (ex. FireEye) and TAP services and move them to the **Selected Service Chain Order** Column.
    - Click Save.

    You should have two service chains as below:



Services Chain List

Add Delete

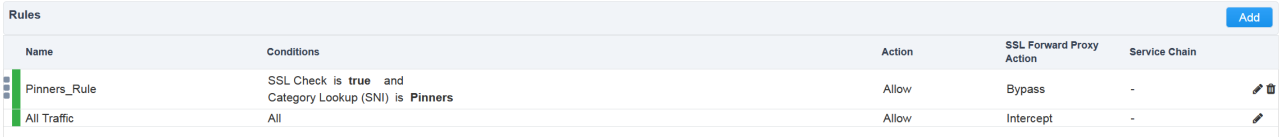
□	Name
□	ssloSC_all_services_chain
□	ssloSC_my_sub_service_chain

  - Click Save & Next.

Topology
System Settings
SSL Configuration
Service
Service Chain
Security Policy
Interception Rule
Summary

- **Security Policy** – security policies are the set of rules that govern how traffic is processed in SSLO. The "actions" a rule can take include,
    - Whether or not to allow the traffic
    - Whether or not to decrypt the traffic
    - Which service chain (if any) to pass the traffic through

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



Rules

Add

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

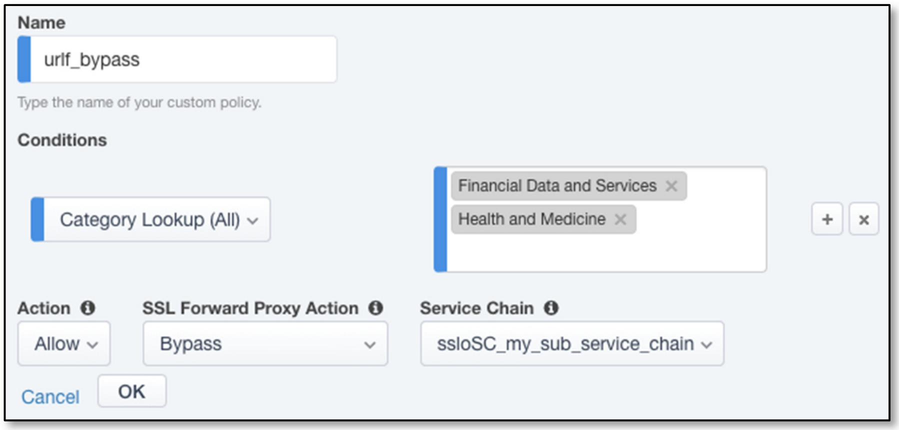
In the background, SSLO maintains these security policies as visual per-request policies. If traffic processing is required that exceeds the capabilities of the rule-based user interface, the underlying per-request policy can be managed directly.

Note that once the per-request policy is manipulated, the rules-based interface can no longer be used.

For the lab, create an additional rule to bypass SSL for "Financial Data and Services" and "Health and Medicine" URL categories.

- Click Add to create a new rule.
    - **Name** – provide a unique name for the rule (ex. "urlf_bypass").
    - **Conditions**
        - **Category Lookup (All)** – add Financial Data and Services and Health and Medicine.

          The Category Lookup (All) condition provides categorization for TLS SNI, HTTP Connect and HTTP Host information.
    - **Action** – select Allow.
    - **SSL Forward Proxy Action** – select Bypass.
    - **Service Chain** – select the my_sub_service_chain service chain.
    - Click OK.



Name

urlf_bypass

Type the name of your custom policy.

Conditions

Category Lookup (All): Financial Data and Services, Health and Medicine

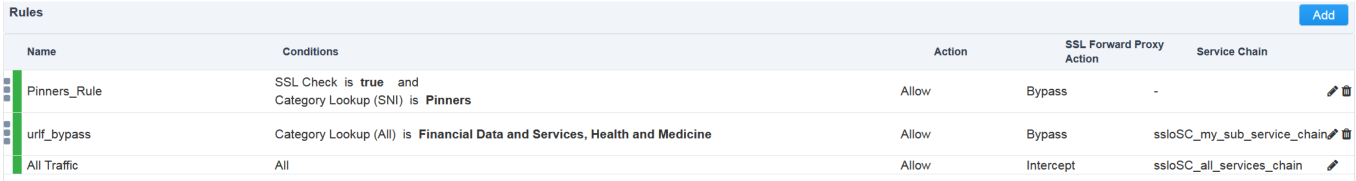
Action: Allow

SSL Forward Proxy Action: Bypass

Service Chain: ssloSC_my_sub_service_chain

Notice in the list of rules that the **All Traffic** rule intercepts but does not send traffic to a service chain. For the lab, edit this rule to send all intercepted traffic to a service chain.

- Click the pencil icon to edit this rule.
- Service Chain – select the service chain containing all of the services.
- Click OK.



Rules

Name	Conditions	Action	SSL Forward Proxy Action	Service Chain
Pinners_Rule	SSL Check is **true** and Category Lookup (SNI) is **Pinners**	Allow	Bypass	-
urlf_bypass	Category Lookup (All) is **Financial Data and Services, Health and Medicine**	Allow	Bypass	ssloSC_my_sub_service_chain
All Traffic	All	Allow	Intercept	ssloSC_all_services_chain

Click Save & Next.

Topology
System Settings
SSL Configuration
Service
Service Chain
Security Policy
Interception Rule
Summary

- **Interception Rule** – interception rules are based on the selected topology and define the "listeners", analogous to LTM virtual servers, that accept and process different types of traffic (ex. TCP, UDP, other). The resulting LTM virtual servers will bind the SSL settings, VLANs, IPs, and security policies created in the topology workflow.
  - **Ingress Network (VLANs)** – this defines the VLANs through which traffic will enter. For a transparent forward proxy topology, this would be a client-side VLAN. Select client-net and move it to the selected column.
  - **L7 Interception Rules** – FTP and email protocol traffic are all "server-speaks-first" protocols, and therefore SSLO 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 SSLO. For this lab we will not select any,
  - Click Save & Next.

- **Egress Setting** – traffic egress settings are now defined per-topology and manage both the gateway route and outbound SNAT settings.

  - **Manage SNAT Settings** – enables per-topology instance SNAT settings. For this lab, select Auto Map.

  - **Gateways** – enables per-topology instance gateway routing. Options are to use the system default route, to use an existing gateway pool, or to create a new gateway. For this lab, select Create New.

  - **IPv4 Outbound Gateways** – when creating a new gateway, this section provides the ratio and gateway address settings.

  - **Ratio** – multiple gateway IP addresses are load balanced in an LTM pool, and the ratio setting allows SSLO to proportion traffic to the gateway members, as required. A ratio on 1 for all members evenly distributes the load across them. For this lab, select 1.

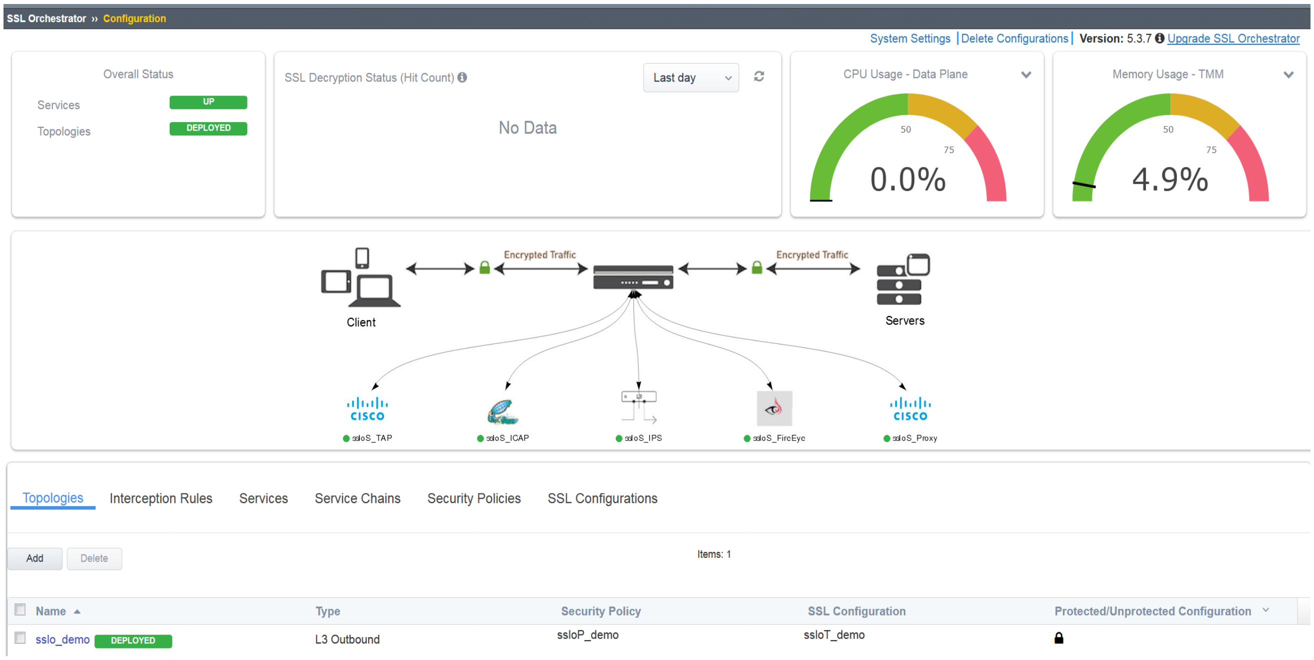
  - **Address** – this is the next hop gateway IP address. For this lab, enter 10.30.0.1.

  - Click Save & Next.

Topology
System Settings
SSL Configuration
Service
Service Chain
Security Policy
Interception Rule
Summary

- **Summary** – the summary page presents an expandable list of all of the workflow-configured objects. To expand the details for any given setting, click the corresponding arrow icon on the far right. To edit any given setting, click the corresponding pencil icon. Clicking the pencil icon will send the workflow back to the selected settings page.
  - When satisfied with the defined settings, click Deploy.

Upon successfully deploying the configuration, SSL Orchestrator will now display a **Dashboard** view:



Overall Status

Services UP

Topologies DEPLOYED

SSL Decryption Status (Hit Count)

No Data

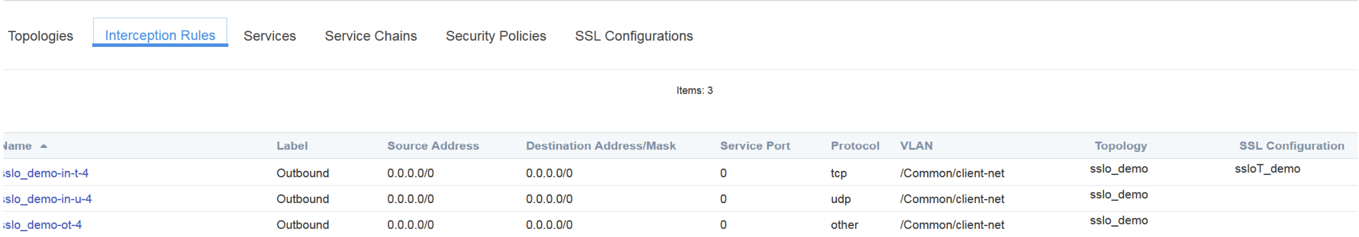
CPU Usage - Data Plane 0.0%

Memory Usage - TMM 4.9%

Items: 1

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

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



Topologies | Interception Rules | Services | Service Chains | Security Policies | SSL Configurations

Items: 3

Name ▲	Label	Source Address	Destination Address/Mask	Service Port	Protocol	VLAN	Topology	SSL Configuration
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	
sslo_demo-ot-4	Outbound	0.0.0.0/0	0.0.0.0/0	0	other	/Common/client-net	sslo_demo	

In the above,

- The **-in-t-4** listener defines normal TCP IPv4 traffic.
- The **-in-u-4** listener defines normal UDP IPv4 traffic.
- The **-ot-4** listener defines normal non-TCP/non-UDP IPv4 traffic.
- The **-ftp**, **-ftps**, **-pop3**, **-smtp25** and **-smtp587** listeners create paths for each respective protocol. You will not see those as we didn't select those services.

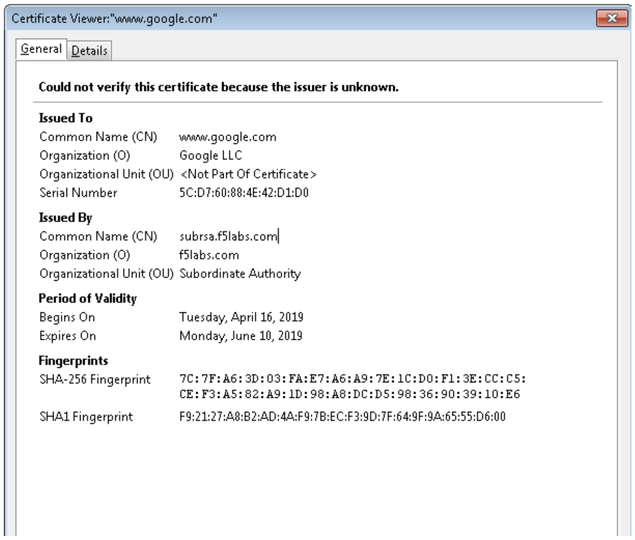
This completes the configuration of SSL Orchestrator as a transparent forward proxy. At this point an internal client should be able to browse out to external (Internet) resources, and decrypted traffic will flow across the security services.

Step 4: Test the solution

To test the deployed solution, use the following options:

- **Server certificate test**

  Open a browser (or a new tab in the Chrome window) on the client system (the RDP host you are working in) and navigate to any remote HTTPS site, for example, 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 configured in SSLO. This confirms that the SSL forward proxy functionality enabled by SSL Orchestrator is working correctly.



Could not verify this certificate because the issuer is unknown.

Issued To

Common Name (CN)	www.google.com
Organization (O)	Google LLC
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	5C:D7:60:88:4E:42:D1:D0

Issued By

Common Name (CN)	subrsa.f5labs.com
Organization (O)	f5labs.com
Organizational Unit (OU)	Subordinate Authority

Period of Validity

Begins On	Tuesday, April 16, 2019
Expires On	Monday, June 10, 2019

Fingerprints

SHA-256 Fingerprint	7C:7F:A6:3D:03:FA:E7:A6:A9:7E:1C:D0:F1:3E:CC:C5: CE:F3:A5:82:A9:1D:98:A8:DC:D5:98:36:90:39:10:E6
SHA1 Fingerprint	F9:21:27:A8:B2:AD:4A:F9:7B:EC:F3:9D:7F:64:9F:9A:65:55:D6:00

- **Decrypted traffic analysis on the security services**

  Depending on the type of security service, it may be easier to log into the console shell and run a tcpdump capture on the inbound or outbound interface, to tail its capture logs, or to log into its management UI and capture analytics. A tcpdump capture usually requires root or sudo access.

  Let's check if we see data on the TAP device. Use Putty to ssh to 10.10.0.4 or use saved session option and load TAP. Login as "student" with the credentials provided. Use the below command for tcpdump. 10.20.0.60 is the IP address of the client machine.

```
sudo tcpdump -lnni eth1 -Xs0 host 10.20.0.60
```

Provide password for student. Browse to an SSL site from the RDP hosts browser eg: https://www.cnn.com and notice that the TAP device is receiving traffic unencrypted. Use CTRL+C to stop the tcpdump.

Let's check another security device. Perform a tcpdump on the IPS device to observe the decrypted clear text traffic. Use Putty to ssh to 10.10.0.3 or use saved session option and load IPS. Login as "student" with the credentials provided. Use the below command for tcpdump. 10.20.0.60 is the client machine and since we are doing port remap the unecrypted traffic will arrive on port 8181

```
sudo tcpdump –lnni eth5.50 -Xs0 host 10.20.0.60 and port 8181
```

Provide password for student. Browse to an SSL site from the RDP hosts browser eg: https://www.cnn.com and notice that the IPS device is receiving traffic unencrypted. Use CTRL+C to stop the tcpdump. Close out (exit) all putty windows.

This completes Lab 1. You have successfully deployed SSLO as an OUTBOUND Transparent Proxy providing visibility to various security devices in the service chain for actionable insight.

LAB 2 – CREATE A GATEWAY REVERSE PROXY SSLO

SSL Orchestrator generally defines inbound traffic flows with a “gateway” architecture. That is, SSLO is designed to sit in front of a separate ADC/load balancer or routed path, and not directly in front of applications, though it is technically possible to support a “single instance” listener going to a single pool of resources. This lab will be re-using the security services created in the first lab to create a single inbound “gateway” service SSLO configuration.

This lab will consist of an abbreviated set of steps, as some of the objects created in Lab 1 (services and service chains) will be fully re-usable here. If any of these objects have not been created, please review Lab 1 for more detailed configuration instructions.

Step 1: Review the lab diagram and map out the services and endpoints

Specifically, note that in this lab there is a web server on the internal network (the client’s network in this case) that external users want to get to. An external client desktop exists on the external/outbound network, that accesses these resources through SSLO.

- The external client is attached to a *10.30.0.0/24* network and is assigned the IP *10.30.0.70*. This network is attached to the BIG-IP 1.2 interface.

- The web server is an Ubuntu 14.04 LTS server configured with Apache2 and PHP5, and listens on three addresses:
  - 10.20.0.90
  - 10.20.0.91
  - 10.20.0.92

  Each instance includes a simple Apache2 text page that also shows which site was accessed.

- In lieu of a separate DNS server in the lab, the external client has static /etc/hosts entries that map the above addresses to the following URLs, respectively:
  - test0.f5labs.com
  - test1.f5labs.com
  - test2.f5labs.com

- A wildcard (*.f5labs.com) server certificate and private key have been installed on the SSL Orchestrator.

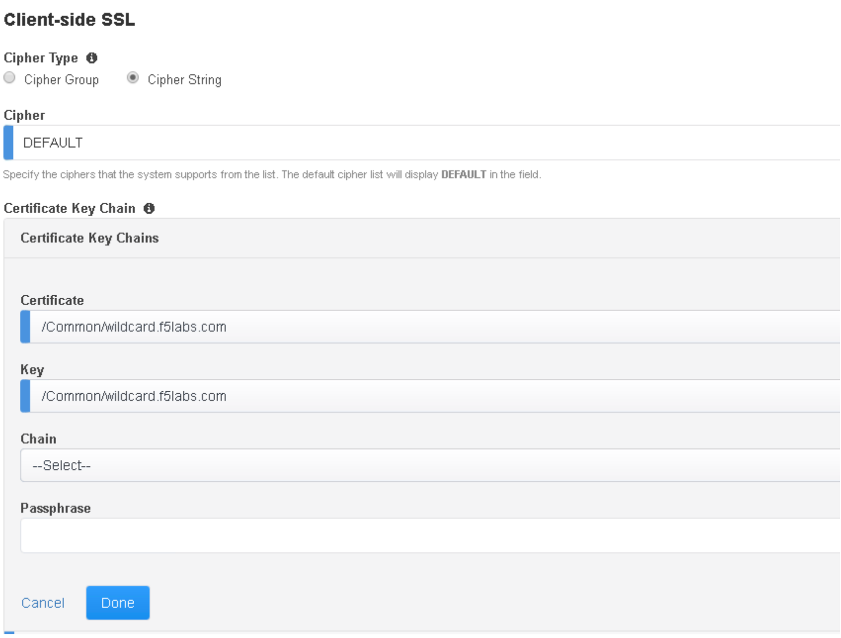
The external client has two options for accessing the internal websites: via wildcard (0.0.0.0/0) gateway, and direct IP listener. This lab will explore the gateway (wildcard) option below.

Note: SSL Orchestrator sends all traffic through an inline layer 3 or HTTP device in the same direction – entering through the inbound interface. Therefore, the layer 3 device will not be able to correctly route both outbound (forward proxy) and inbound (reverse proxy) traffic at the same time in our labs. **Do not use a service chain with L3 Service or HTTP Proxy in it for our labs.**

Step 2: Configure an L3 inbound SSLO deployment through Guided Configuration

In this scenario, an SSLO L3 inbound listener is configured as a gateway service. It will listen on a wildcard VIP (0.0.0.0/0), ((other option specific subnet (vs. a dedicated single IP)) and terminate inbound TLS traffic flows via wildcard or subject alternative name (SAN) certificate. Follow the L3 Inbound topology workflow to build this solution. In the SSL Orchestrator dashboard view, select the Topologies tab (bottom) and click Add.

- **Configuration review and prerequisites** – take a moment to review the topology options and workflow configuration, then click Next.
- **Topology Properties**
    - **Name**: provide some name (ex. "inbound")
    - **Protocol**: TCP
    - **IP Family**: IPv4
    - **Topology**: select L3 Inbound
    - Click Save & Next
- **SSL Configuration** – an inbound topology requires different SSL settings.
    - Click Show Advanced Setting
    - **Client-side SSL**
        - **Cipher Type**: Cipher String
        - **Cipher String**: DEFAULT
        - **Certificate Key Chain** – the certificate key chain represents the certificate and private key of an endpoint server instance (the target of a remote client's request). In a gateway-mode configuration, this would typically be a wildcard of Subject Alternative Name (SAN) certificate in the event the SSLO inbound listener was intended to service multiple sites. In this lab a wildcard certificate has been provided. Select the pencil icon to edit, then select the wildcard.f5labs.com certificate and private key and click Done. Make sure you are adding/editing the cert/key in the Certificate Key Chain section as indicated below.



Client-side SSL

Cipher Type
○ Cipher Group ● Cipher String

Cipher
DEFAULT

Specify the ciphers that the system supports from the list. The default cipher list will display **DEFAULT** in the field.

Certificate Key Chain

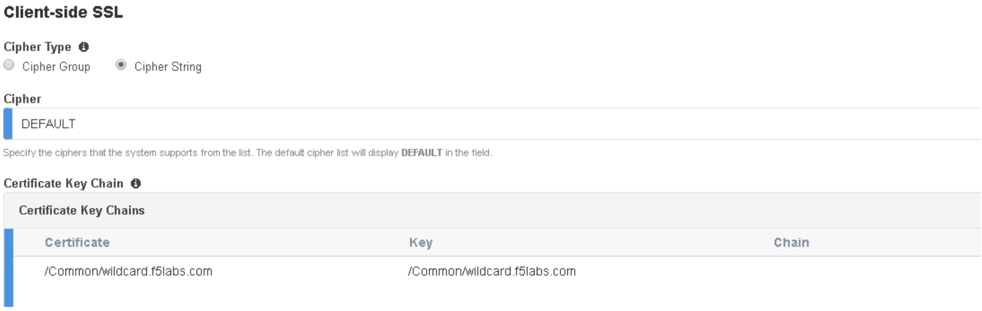
Certificate Key Chains

Certificate
/Common/wildcard.f5labs.com

Key
/Common/wildcard.f5labs.com

Chain
--Select--

Passphrase



Client-side SSL

Cipher Type
○ Cipher Group ● Cipher String

Cipher
DEFAULT

Specify the ciphers that the system supports from the list. The default cipher list will display **DEFAULT** in the field.

Certificate Key Chain

Certificate Key Chains

Certificate	Key	Chain
/Common/wildcard.f5labs.com	/Common/wildcard.f5labs.com	

- **Server-side SSL**
  - **Cipher Type**: Cipher String
  - **Cipher String**: DEFAULT
  - **Trusted Certificate Authority** – as an inbound solution, the server-side SSL would be pointing to internal servers. While definitely possible to perform validation against internal server certificates, it is likely less important to do so. Leave this setting as is.
  - **Expire Certificate Response** – Assuming no internal certificate validation is needed, the default **drop** setting will cause the connection to fail, so set this to Ignore.
  - **Untrusted Certificate Authority** – Assuming no internal certificate validation is needed, the default **drop** setting will cause the connection to fail, so set this to Ignore.
  - **[Advanced] OCSP** – Assuming no internal certificate validation is needed, any OCSP configuration will cause the connection to fail, so leave this as is.
  - **[Advanced] CRL** – Assuming no internal certificate validation is needed, any CRL configuration will cause the connection to fail, so leave this as is.

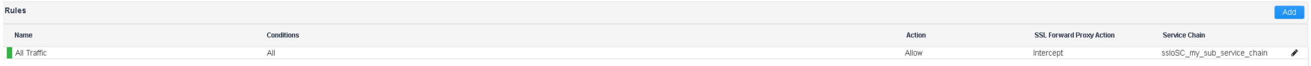
  - Click Save & Next.

- **Services List** – the same services can be leveraged here, so simply click Save & Next.

- **Service Chain List** – the same service chains can be leveraged here, so simply click Save & Next.

- **Security Policy** – the security policy requirements are specific to each organization, though an inbound security policy would likely be less complex than an outbound policy.

  Remove (click the trash can) the built-in "Pinners_Rule", edit (click the pencil icon) the "All Traffic" policy to add the service chain "my_sub_service_chain" with the L2 and TAP services (only), click OK and click Save & Next.



Rules

Name	Conditions	Action	SSL Forward Proxy Action	Service Chain
All Traffic	All	Allow	Intercept	ssloSC_my_sub_service_chain

- **Interception Rule** – here is where a gateway-mode topology and the existing application topology generally differ. Where an explicit application topology "bolts onto" an existing application that performs its own SSL management (SSL offload), traffic management (pools) and traffic intelligence (iRules, profiles), the gateway-mode SSLO topology provides a single, generic entry point for potentially multiple applications, and would sit in front of another ADC or routing device. This is mostly useful when an SSL visibility device must sit closer to the outer edge of an environment, and/or when the SSL visibility product "owner" does not otherwise own the applications or ADC(s).

  It is possible to configure an L3 Inbound topology configuration with a single target IP address and port and destination pool (targeted mode). However, an L3 Inbound topology must re-encrypt the inbound traffic. In this lab we will just use the Gateway mode.

  - **Gateway mode** – interception rule listening on a wildcard IP, port 443, with a wildcard or SAN certificate. Clients route through SSLO.

    - Hide Advanced Setting. Make sure you are NOT viewing advanced settings.

    - **Source Address**: 0.0.0.0/0 (this is default)

    - **Destination Address/Mask**: 0.0.0.0/0

    - **Port**: 443
    - **VLANs**: outbound (this is the server-side VLAN in our labs). Move this to selected column.

    - **[Protocol Settings] L7 Profile Type** – this setting enables or disables HTTP processing. Set to HTTP (default)

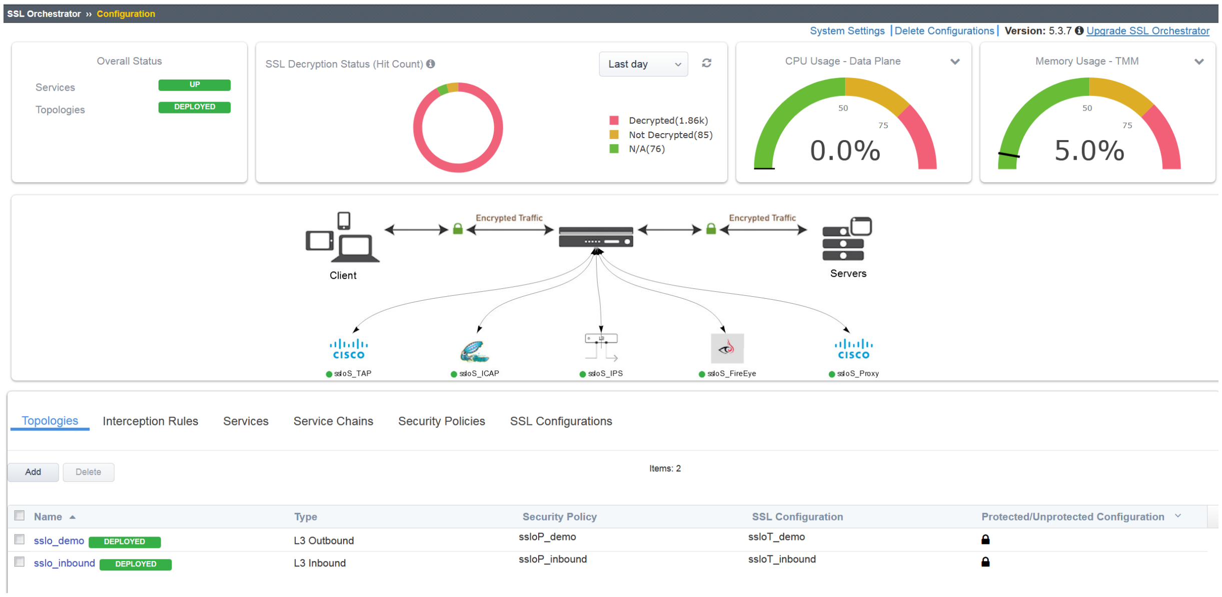
    - **[Protocol Settings] L7 Profile** – if the above option is set to HTTP, this option selects a specific HTTP profile. Set to /Common/http.

    Click Save & Next

- **Egress Settings** – traffic egress settings are now defined per-topology and manage both the gateway route and outbound SNAT settings.
  - **Manage SNAT Settings** – enables per-topology instance SNAT settings. For this lab, select Auto Map.
  - **Gateways** – enables per-topology instance gateway routing. Options are to use the system default route, to use an existing gateway pool, or to create a new gateway. For this lab, select Default Route.

    Click Save & Next

- **Summary** – the summary page presents an expandable list of all of the workflow-configured objects. To expand the details for any given setting, click the corresponding arrow icon on the far right. To edit any given setting, click the corresponding pencil icon. Clicking the pencil icon will send the workflow back to the selected settings page.
  - When satisfied with the defined settings, click Deploy. After successful deployment you will see a new topology in the dashboard view as below.



SSL Orchestrator » Configuration

System Settings | Delete Configurations | Version: 5.3.7 Upgrade SSL Orchestrator

Overall Status

Services	UP
Topologies	DEPLOYED

SSL Decryption Status (Hit Count)

Last day

- 🟥 Decrypted(1.86k)
- 🟨 Not Decrypted(85)
- 🟩 N/A(76)

CPU Usage - Data Plane: 0.0%

Memory Usage - TMM: 5.0%

Topologies | Interception Rules | Services | Service Chains | Security Policies | SSL Configurations

Add | Delete

Items: 2

Name	Type	Security Policy	SSL Configuration	Protected/Unprotected Configuration
☐ sslo_demo DEPLOYED	L3 Outbound	ssloP_demo	ssloT_demo	🔒
☐ sslo_inbound DEPLOYED	L3 Inbound	ssloP_inbound	ssloT_inbound	🔒

- **Testing** – RDP to the inbound Win7 client. Use credentials provided. For gateway-mode testing, the lab's inbound desktop client includes static Hosts entries that match the real IPs of the internal web server,
  - test0.f5labs.com = 10.20.0.90
  - test1.f5labs.com = 10.20.0.91
  - test2.f5labs.com = 10.20.0.92

  and a static persistent route that points 10.20.0.0/24 traffic to the BIG-IP outbound (external) VLAN self-IP (10.20.0.100). Open chrome browser and browse to https://test0.f5labs.com and/or https://test1.f5labs.com and/or https://test2.f5labs.com

- **Decrypted traffic analysis on the security services**

  Depending on the type of security service, it may be easier to log into the console shell and run a tcpdump capture on the inbound or outbound interface, to tail its capture logs, or to log into its management UI and capture analytics. A tcpdump capture usually requires root or sudo access.

  Let’s check if we see data on the TAP device. Use Putty (on the outbound RDP Win7 machine) to ssh to 10.10.0.4 or use saved session option and load TAP. Login as “student” with the credentials provided. Use the below command for tcpdump. 10.20.0.90 is the IP address of one of the web server/application

```
sudo tcpdump -lnni eth1 -Xs0 host 10.20.0.90
```

  Provide password for student. Browse to https://test0.f5labs.com on the Inbound Win7 Client RDP hosts browser and notice that the TAP device is receiving traffic unencrypted. Use CTRL+C to stop the tcpdump.

  Close out (exit) the putty window.

  This completes Lab 2. You have successfully deployed SSLO as an INBOUND Reverse Proxy (gateway mode) providing visibility to various security devices in the service chain for actionable insight.

Lab 3 – Create an explicit forward Proxy SSLO

SSL Orchestrator creates discreet, non-overlapping interception rules (listeners) based on the selected topology. For example, the explicit forward proxy workflow minimally creates an explicit proxy listener and relying transparent proxy listener attached to the explicit proxy tunnel. If a separate transparent proxy workflow was created, the resulting listener would not conflict with or overlap the existing transparent proxy listener. Therefore, assuming a transparent forward proxy already exists from Lab 1, the following workflow will create a separate set of non-overlapping listeners to satisfy an explicit forward proxy use case.

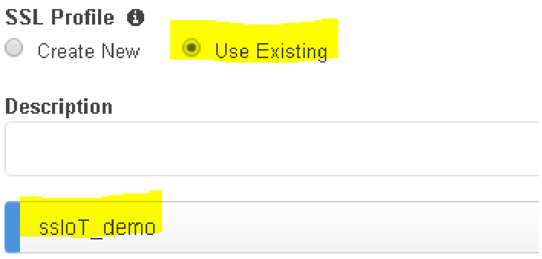
This lab will consist of an abbreviated set of steps, as all of the objects created in Lab 1 (SSL settings, services, service chains and security policies) will be fully re-usable here. If any of these objects have not been created, please review Lab 1 for more detailed configuration instructions.

Step 1: Review the lab diagram and map out the services and endpoints

This lab uses the exact same environment, so SSL settings, services, service chains and security policy will be re-used.

Step 2: Configure an explicit proxy SSLO deployment through Guided Configuration

- **Configuration review and prerequisites** – In the SSL Orchestrator dashboard view, select the Topologies tab (bottom) and click Add take a moment to review the topology options and workflow configuration, then click Next.
- **Topology Properties**
  - **Name**: provide some name (ex. "explicit")
  - **Protocol**: TCP
  - **IP Family**: IPv4
  - **Topology**: select L3 Explicit Proxy
  - Click Save & Next
- **SSL Configurations** – the existing outbound SSL settings from Lab 1 can be re-used here.
  - **SSL Profile**: Use Existing, select existing outbound SSL settings.



SSL Profile

○ Create New ● Use Existing

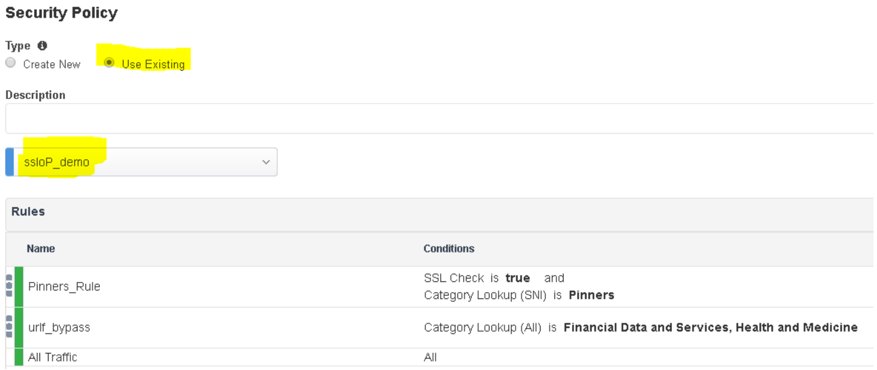
Description

ssloT_demo

  - Click Save & Next

Whenever repurposing a topology setting, a warning will appear, “There are other configuration items that are referencing this item. Editing this item will affect the referencing ones mentioned below”. Click OK to acknowledge.

- **Services List** – there are no new services to create.
  - Click Save & Next
- **Service Chain List** – there are no new service chains to create.
  - Click Save & Next
- **Security Policy** – the existing outbound Security Policy from Lab 1 can be re-used here.
  - **Type**: Use Existing, select existing outbound SSL settings.



Security Policy

Type
○ Create New ● Use Existing

Description

sslоP_demo

Rules

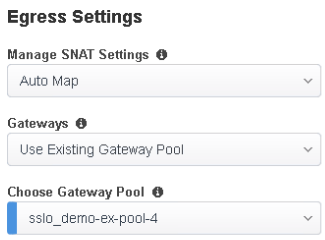
Name	Conditions
Pinners_Rule	SSL Check is **true** and Category Lookup (SNI) is **Pinners**
urlf_bypass	Category Lookup (All) is **Financial Data and Services, Health and Medicine**
All Traffic	All

  - Click Save & Next

Whenever repurposing a Security Policy setting, a warning will appear, “There are other configuration items that are referencing this item. Editing this item will affect the referencing ones mentioned below”. Click OK to acknowledge.

- **Interception Rule** – an explicit proxy requires a unique IP address and port listener.
  - **IPV4 Address**: 10.20.0.150
  - **Port**: 3128
  - **Access Profile**: if enabling explicit proxy authentication, select an existing SWG-Explicit access profile here. Leave None.
  - **VLANs**: client-net. Move it to the selected column.
  - Click Save & Next
- **Egress Setting** – traffic egress settings are now defined per-topology and manage both the gateway route and outbound SNAT settings.

- **Manage SNAT Settings** – enables per-topology instance SNAT settings. For this lab, select Auto Map.
- **Gateways** – enables per-topology instance gateway routing. Options are to use the system default route, to use an existing gateway pool, or to create a new gateway. For this lab, select Use Existing Gateway Pool, then select the "-ex-pool-4" gateway pool.



Egress Settings

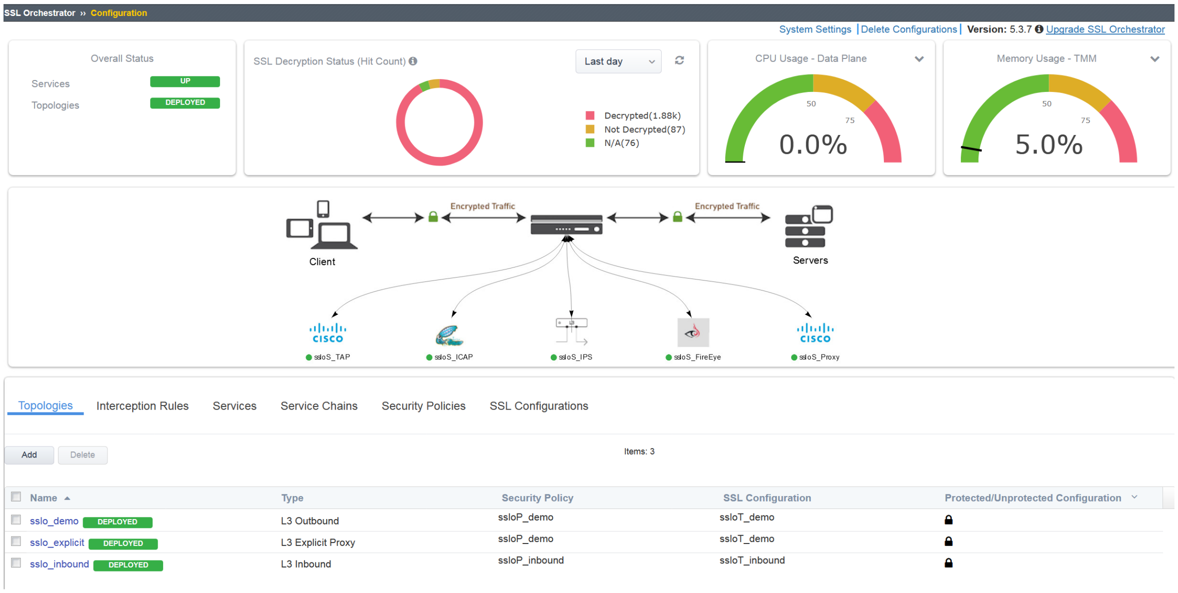
Manage SNAT Settings: Auto Map

Gateways: Use Existing Gateway Pool

Choose Gateway Pool: sslo_demo-ex-pool-4

- Click Save & Next

- **Summary** – the summary page presents an expandable list of all of the workflow-configured objects. To expand the details for any given setting, click the corresponding arrow icon on the far right. To edit any given setting, click the corresponding pencil icon. Clicking the pencil icon will send the workflow back to the selected settings page.
  - When satisfied with the defined settings, click Deploy. After successful deployment the dashboard will show the new topology as below.



Version: 5.3.7 Upgrade SSL Orchestrator

Services: UP

Topologies: DEPLOYED

🟥 Decrypted(1.88k)
🟨 Not Decrypted(87)
🟩 N/A(76)

CPU Usage - Data Plane: 0.0%

Memory Usage - TMM: 5.0%

Items: 3

Name	Type	Security Policy	SSL Configuration	Protected/Unprotected Configuration
sslo_demo DEPLOYED	L3 Outbound	ssloP_demo	ssloT_demo	🔒
sslo_explicit DEPLOYED	L3 Explicit Proxy	ssloP_demo	ssloT_demo	🔒
sslo_inbound DEPLOYED	L3 Inbound	ssloP_inbound	ssloT_inbound	🔒

This completes Lab 3. You have successfully deployed SSLO as an OUTBOUND Explicit Proxy providing visibility to various security devices in the service chain for actionable insight. Additionally, you just used the same SSLO for both Transparent and Explicit proxy deployments.

LAB 4 – CREATE AN SSLO FOR EXISTING APPLICATIONS

SSL Orchestrator defines an existing application as a typical reverse proxy LTM virtual server, performing its own SSL handling and traffic management. The Existing Application SSLO topology therefore only needs to create the components that this virtual server can consume, specifically the services, service chains, and security policy. The Existing Application SSLO workflow skips SSL management and interception rules, and ultimately produces an SSLO-type per-request policy that can be attached to an existing LTM virtual server.

This lab will consist of an abbreviated set of steps, as all of the relevant objects created in Lab 1 (services, service chains and security policies) will be fully re-usable here. If any of these objects have not been created, please review Lab 1 for more detailed configuration instructions.

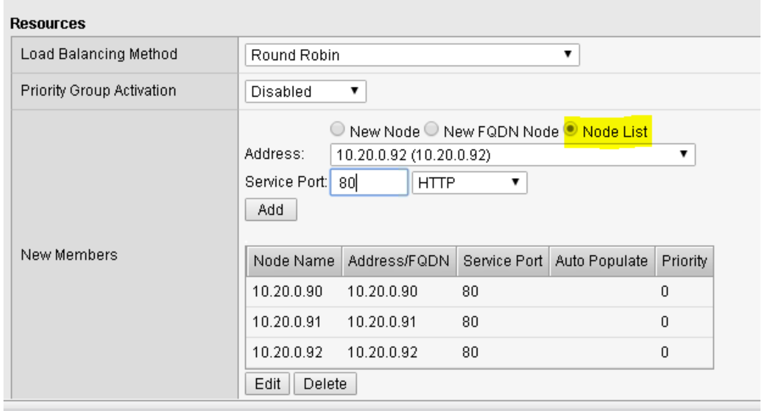
Step 1: Review the lab diagram and map out the services and endpoints

This lab uses the exact same environment, so SSL settings, services, service chains and security policy will be re-used.

Step 2: Create an LTM application

For the lab, create a simple LTM application,

- **Create a pool** – use one (or multiple) of the internal webserver IPs and select port 80. Go to **Local Traffic>>Pools>>Pool List** and click Create. Name the pool "existing-app-pool". Leave other values as default and add the following three members. Under Resources you can click Node List and add these with port 80. Once all three are added click Finished.
  - 10.20.0.90:80
  - 10.20.0.91:80
  - 10.20.0.92:80



Resources

Load Balancing Method: Round Robin

Priority Group Activation: Disabled

New Members: ○ New Node ○ New FQDN Node ● Node List

Address: 10.20.0.92 (10.20.0.92)

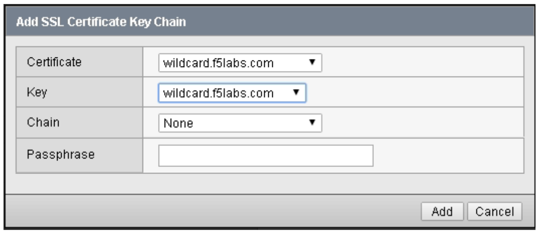
Service Port: 80 HTTP

Add

Node Name	Address/FQDN	Service Port	Auto Populate	Priority
10.20.0.90	10.20.0.90	80		0
10.20.0.91	10.20.0.91	80		0
10.20.0.92	10.20.0.92	80		0

Edit Delete

- **Create a client SSL profile** – use the wildcard.f5labs.com certificate and private key. Go to **Local Traffic>>Profiles>>SSL>>Client** and click Create with name "existing_app_clientssl"
  Click the checkbox all the way to the right of "**Certificate Key Chain**" and click **Add.** Chose the wildcard.f5labs.com cert and key and then click Add and then Finished.



Add SSL Certificate Key Chain

Certificate	wildcard.f5labs.com
Key	wildcard.f5labs.com
Chain	None
Passphrase	

- **Create an LTM virtual server** – use the following basic settings. Go to **Local Traffic>>Virtual Servers>>Virtual Servers List** and click Create with the name "existing_app_vs". Leave other settings default.
  - **Destination Address/Mask**: 10.30.0.205
  - **Service Port**: 443
  - **HTTP Profile (Client)**: http
  - **SSL Profile (Client)**: wildcard.f5labs.com SSL profile (Move existing_app_clientssl to the selected column)
  - **VLANs and Tunnel Traffic**: Drop down to "Enabled on" and then move outbound VLAN to selected column.
  - **Source Address Translation**: Auto Map
  - **Default Pool**: existing-app-pool

  Click Finished.

- **Test access to the LTM virtual server** – the webserver should be accessible via HTTPS request to the LTM virtual server. RDP to the Inbound Win7 Client.
  - A hosts file entry pointing 10.30.0.205 to existingapp.f5labs.com is already created on your test machine so to test access open a browser and browse to https://existingapp.f5labs.com. The certificate is a wildcard, so any *.f5labs.com hostname would also work.

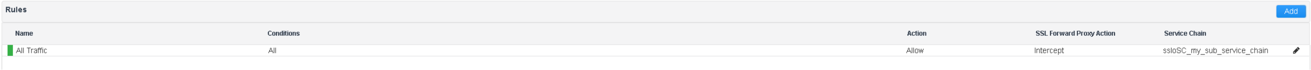
Step 3: Configure an Existing Application deployment through Guided Configuration

- **Configuration review and prerequisites** – In the SSL Orchestrator dashboard view, select the Topologies tab (bottom) and click Add take a moment to review the topology options and workflow configuration, then click Next.

- **Topology Properties**
  - **Name**: provide some name (ex. "existing_app_1")
  - **IP Family**: IPv4
  - **Topology**: select Existing Application
  - Click Save & Next

- **Services List** – there are no new services to create.
    - Click Save & Next
- **Services Chain List** – there are no new service chains to create.
    - Click Save & Next
- **Security Policy** – the security policy requirements are specific to each organization, though an inbound security policy would likely be less complex than an outbound policy.

  Remove the built-in “Pinners_Rule”, edit the “All Traffic” policy to add the service chain “my_sub_service_chain” with the L2 and TAP services (only), click OK and then click Save & Next.

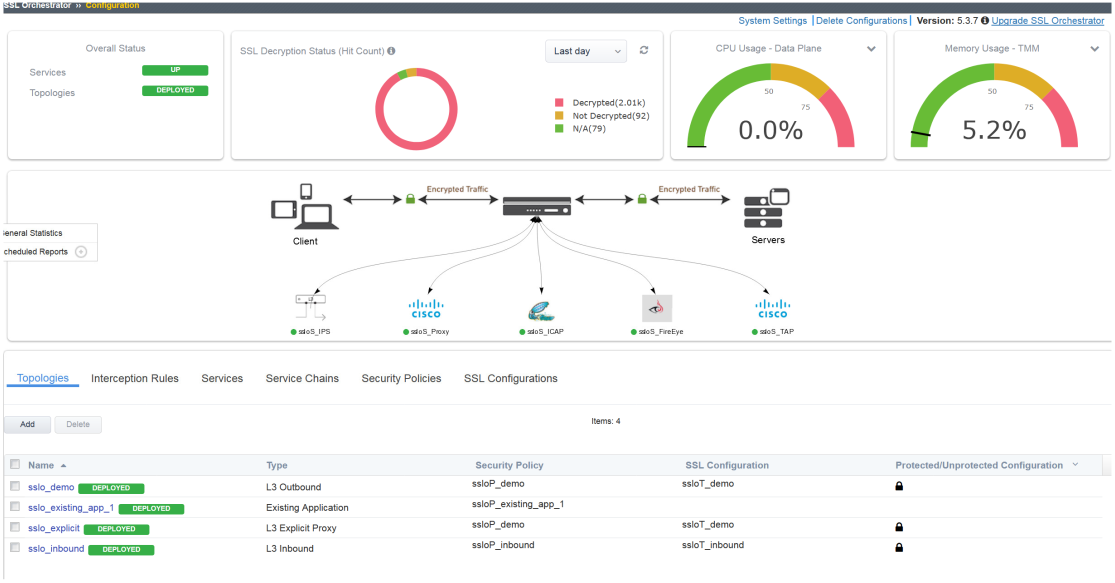


Rules

Add

Name	Conditions	Action	SSL Forward Proxy Action	Service Chain
All Traffic	All	Allow	Intercept	ssloSC_my_sub_service_chain

- **Summary** – the summary page presents an expandable list of all of the workflow-configured objects. To expand the details for any given setting, click the corresponding arrow icon on the far right. To edit any given setting, click the corresponding pencil icon. Clicking the pencil icon will send the workflow back to the selected settings page.
    - When satisfied with the defined settings, click Deploy. After successful deployment the Dashboard will show the new topology.



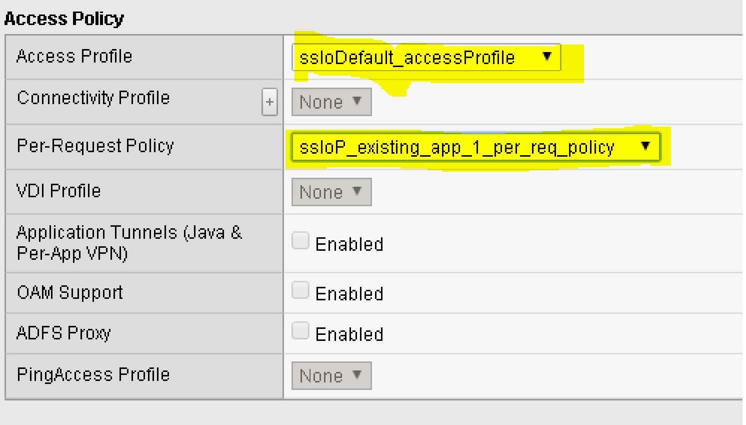
Items: 4

Name	Type	Security Policy	SSL Configuration	Protected/Unprotected Configuration
sslo_demo DEPLOYED	L3 Outbound	ssloP_demo	ssloT_demo	🔒
sslo_existing_app_1 DEPLOYED	Existing Application	ssloP_existing_app_1		
sslo_explicit DEPLOYED	L3 Explicit Proxy	ssloP_demo	ssloT_demo	🔒
sslo_inbound DEPLOYED	L3 Inbound	ssloP_inbound	ssloT_inbound	🔒

Step 4: Attach the SSLO objects to an existing LTM application

The Existing Application topology workflow produces a single SSLO per-request policy. To attach this to the LTM virtual server, edit the virtual server properties. Go to **Local Traffic>>Virtual Servers>>Virtual Servers List** and click on the “existing_app_vs”. Assign the profiles in the Access Policy section as below:

- **Access Policy (Access Profile**): attach the single “ssloDefault_accessProfile”.
- **Access Policy (Per-Request Policy)**: attach the existing application per-request policy.



Access Policy

Access Profile	ssloDefault_accessProfile ▼
Connectivity Profile	None ▼
Per-Request Policy	ssloP_existing_app_1_per_req_policy ▼
VDI Profile	None ▼
Application Tunnels (Java & Per-App VPN)	□ Enabled
OAM Support	□ Enabled
ADFS Proxy	□ Enabled
PingAccess Profile	None ▼

Click Update

- **Decrypted traffic analysis on the security services**

  Depending on the type of security service, it may be easier to log into the console shell and run a tcpdump capture on the inbound or outbound interface, to tail its capture logs, or to log into its management UI and capture analytics. A tcpdump capture usually requires root or sudo access.

  Let’s check if we see data on the TAP device. Use Putty (on the outbound RDP Win7 machine) to ssh to 10.10.0.4 or use saved session option and load TAP. Login as “student” with the credentials provided. Use the below command for tcpdump. 10.30.0.70 is the IP address of our client machine accessing the existing application.

```
sudo tcpdump -lnni eth1 -Xs0 host 10.30.0.70
```

Provide password for student. Browse to https://existingapp.f5labs.com on the Inbound Win7 Client RDP hosts browser and notice that the TAP device is receiving traffic unencrypted. Use CTRL+C to stop the tcpdump.

Close out the putty window.

This completes Lab 4. You have successfully deployed SSLO as an INBOUND inspection device for an existing application providing visibility to various security devices in the service chain for actionable insight.

APPENDIX – RESOURCES

The following are some additional resources.

SSLO Overview : https://www.f5.com/content/dam/f5/corp/global/pdf/products/ssl-orchestrator-overview.pdf

SSLO Datasheet : https://www.f5.com/pdf/products/ssl-orchestrator-datasheet.pdf

Decreasing Security TCO : https://www.f5.com/pdf/products/decreasing-security-total-cost-of-ownership-with-f5-ssl-orchestrator.pdf

Recommended Practices Guides

FireEye : https://www.f5.com/pdf/solution-center/f5-sslo-5-1-integration-with-fireeye-nx-recommended-practices-guide.pdf

Symantec DLP : https://www.f5.com/pdf/solution-center/f5-ssl-orchestrator-and-symantec-dlp-ssl-visibility-and-content-adaptation.pdf

Palo Alto NGFW : https://www.f5.com/pdf/solution-center/f5-ssl-orchestrator-and-palo-alto-networks-next-gen-firewall-solution.pdf

Cisco WSA : https://www.f5.com/pdf/solution-center/f5-ssl-orchestrator-and-cisco-wsa-recommended-practices.pdf