LOADBALANCER

DEPLOYMENT GUIDE

Load Balancing WAF Gateway with Metaswitch EAS

Version 2.3.0

Table of Contents

1. Overview

1.1. Introduction to the WAF Gateway with Metaswitch EAS

Metaswitch and Loadbalancer.org have a long-standing partnership for the implementation of Metaswitch EAS. Whether deployed as hardware or virtualized, the Loadbalancer.org solution ensures Metaswitch EAS is highly available and highly secure.

The Loadbalancer.org appliance includes a fully integrated industry standard Web Application Firewall (WAF) by default. Although a wide number of 3rd party commercial hardware and virtual WAFs are currently available, these are typically not configured to meet the specific and ever-changing threats faced by communication service providers. Therefore, Metaswitch recommend that customers looking to enhance their network security upgrade to a Loadbalancer.org WAF Gateway.

Developed collaboratively with Metaswitch and based on real-world customer experience, the Loadbalancer.org solution explicitly addresses known threats in the Metaswitch installed base using custom WAF rules specifically developed by Loadbalancer.org to protect a Metaswitch EAS deployment. This is described in detail in this document.

1.2. Virtualized Deployments

In a virtualized environment, the Virtual EAS deployment topology used determines how the load balancer must be configured. It is essential to know which topology is in place or being planned before attempting to set up a load balancer. In this scenario, the Loadbalancer.org solution should be used to provide both load balancing and protection against security vulnerabilities.

1.3. Hardware Deployments

Hardware installs follow a different deployment architecture. In this scenario, a separate pair of dedicated Loadbalancer.org appliances should be installed to provide protection against security vulnerabilities, while leaving the existing production load balancers untouched. Contact Loadbalancer.org for assistance with deploying a hardware WAF gateway solution.

1.4. Non-Load Balanced Deployments

It is possible to add WAF gateways to a non-load balanced Metaswitch EAS deployment. In this way, even a simple deployment consisting of a single EAS server can benefit from the protection afforded by a WAF gateway solution.

2. About this Guide

This guide details the steps required to configure a Loadbalancer.org appliance with WAF gateways in a Metaswitch EAS environment utilizing Loadbalancer.org appliances. It covers the configuration of the Loadbalancer.org appliances and also any Metaswitch EAS configuration changes that are required to enable deploying WAF gateways in front of the Metaswitch services. This guide is applicable to all types of Metaswitch EAS deployments.

For more information about initial appliance deployment, network configuration and using the Web User Interface (WebUI), please also refer to the Administration Manual.

3. Loadbalancer.org Appliances Supported

Our **hardware 10G models and above** can be used for a WAF gateway with Metaswitch EAS deployment.

Any of our virtual models (provided that the virtual machine satisfies the minimum allocation specifications described in Sizing, Capacity, and Performance for a Virtual WAF Gateway Deployment) can be used for a WAF gateway with Metaswitch EAS deployment.

For full specifications of available models please refer to: https://www.loadbalancer.org/products/enterprise

4. Software Versions Supported

4.1. Loadbalancer.org Appliance

- V8.9.1 and later

> **Note** The screenshots used throughout this document aim to track the latest Loadbalancer.org software version. If you're using an older version, or the very latest, the screenshots presented here may not match your WebUI exactly.

4.2. Metaswitch EAS

- All versions

5. Sizing, Capacity, and Performance for a Virtual WAF Gateway Deployment

The Loadbalancer.org appliances can be deployed as **virtual appliances**.

For deployments up to 250,000 subscribers, your virtual host should be allocated a minimum of 8 vCPUs, 16 GB of RAM, and 8 GB of disk storage.

This specification will support the following bandwidth and connection thresholds:

- Internet → EAS bandwidth: 100 Mbit/s
- Internet → EAS packets/s: 70,000 pkts/s
- EAS → Internet bandwidth: 700 Mbit/s
- EAS → Internet packets/s: 55,000 pkts/s
- Concurrent connections: 380,000 connections

For larger deployments, your Metaswitch support representative will give you details of the expected load on your load balancers based on your predicted usage profile.

6. Using Loadbalancer.org WAF Gateways with Metaswitch EAS

> **Note** It's highly recommended that you have a working Metaswitch EAS environment first before implementing any load balancer appliances.

6.1. The EAS Services that can be Protected

The Loadbalancer.org WAF solution for Metaswitch EAS is designed to protect:

- the CommPortal login page;
- (optionally) the SIP provisioning services (SIP-PS and PPS) on the EAS nodes.

These are the "services" described throughout this document, in the context of "the services to be protected by the WAF gateway".

6.2. Conceptual Overview

For each service provided by a Metaswitch deployment that needs to be protected by putting a WAF gateway in front of it, up to three linked elements need to be created:

- TLS/SSL termination VIP (only required for services handling encrypted traffic, i.e. HTTPS)
- WAF gateway (always required)
- Layer 7 VIP (always required)

In general, for each service that needs protecting, a chain of the above elements must be created on a Loadbalancer.org appliance. The specifics can vary slightly depending on the deployment scenario, as described throughout the remainder of this document.

6.2.1. Example: HTTP Service

> **Note** In line with Metaswitch guidance, **it is strongly advised that internet facing services are HTTPS based only**.
>
> **Plaintext HTTP services should not be used.**

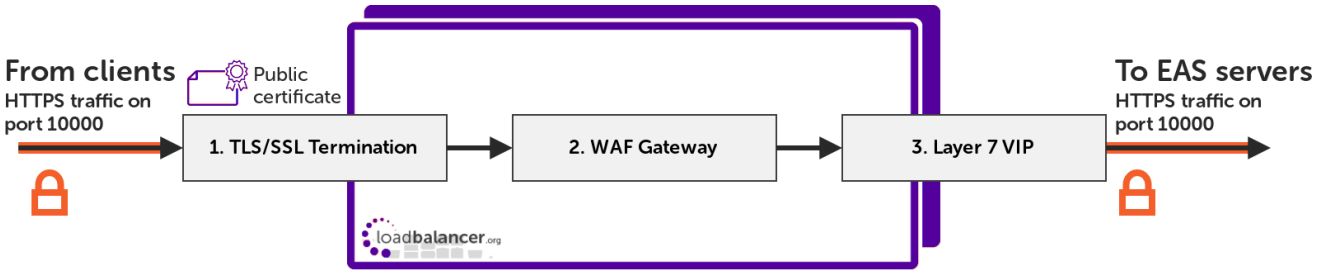
Consider an example chain of services put in front of a plaintext HTTP service listening on port 80:

Elements in the chain

1. **WAF**: scans the incoming plain text HTTP traffic and blocks malicious traffic.
2. **Layer 7 VIP**: passes the scanned, safe traffic on toward the Metaswitch EAS deployment.

6.2.2. Example: HTTPS Service

Consider an example chain of services put in front of an (encrypted) HTTPS service listening on port 10000:



Elements in the chain

1. **TLS/SSL Termination**: decrypts the incoming HTTPS traffic so that the WAF can scan it in the next step. The appropriate certificate for the service is used to perform decryption.
2. **WAF**: scans the plain text HTTP traffic, and blocks malicious traffic.
3. **Layer 7 VIP**: passes the scanned, safe traffic on toward the Metaswitch EAS deployment.

6.3. TLS/SSL Termination

A WAF gateway can only handle plain text HTTP traffic (because HTTPS traffic is opaque, due to being encrypted, and cannot be inspected.) If HTTPS-based Metaswitch services are in use and need a WAF gateway placed in front of them then it is necessary to set up TLS/SSL terminating services on the load balancer to decrypt the HTTPS traffic.

In the context of a Metaswitch EAS deployment, traffic that is decrypted for scanning by a WAF gateway must be re-encrypted before it is sent onward to the Metaswitch EAS servers. This is because, for HTTPS services, the EAS servers are still expecting to receive encrypted traffic.

Full instructions on how to set up TLS/SSL termination and re-encryption are given in the 'Appliance Configuration' sections for each different deployment scenario.

6.3.1. Legacy TLS/SSL Options for Older Equipment

In most deployments, it is necessary to enable legacy TLS/SSL options on the load balancer. This is often required to support older equipment, for example legacy hardware phones that don't support any recent TLS/SSL cryptographic protocols (e.g. TLS 1.2). The 'High Compatibility' mode, which is ***used by default*** in the instructions throughout this document, enables this legacy support.

In the event that an EAS WAF deployment ***definitely does not*** require legacy cryptographic support, or if TLS 1.0 and TLS 1.1 need to be disabled for security reasons, then the SSL operation mode 'High Security' should be used instead of 'High Compatibility'.

If support for SSL 3.0 is required, the SSL/TLS offload services should first be created in 'High Compatibility'

mode, as per the instructions, and then modified and set to 'Custom' mode. This displays the 'Disable SSLv3 Ciphers' checkbox which can then be deselected to enable SSL 3.0 support.

6.3.2. Warning Regarding Older Certificates

Important: Some equipment vendors mandate the use of public certificates that they have signed. For example, a hardphone from the Acme Corporation may only function correctly when communicating with a server that holds a public certificate also signed by the Acme Corporation.

Vendors following this practice have tended to issue customers with long life, multi-year public certificates. As a result, there are certificates in active use on production systems which are many years old. Often, these old certificates may use outdated standards which are now considered to be insecure, for example they may be shorter than 2048 bits in length or they may use an insecure hashing algorithm.

It is imperative to check for the presence and use of old certificates as **they may not function on modern systems**. Such certificates should be checked, uploaded to the load balancer, and tested well in advance of any WAF deployment.

6.4. Load Balancing & HA Requirements

In addition to the WAF gateway related functionality described in this document, Loadbalancer.org appliances can also be used to provide load balancing and high availability to Metaswitch EAS deployments.

For inquiries regarding load balancer solutions, Loadbalancer.org can provide assistance and advice. Please contact us at support@loadbalancer.org.

6.5. NIC Bonding for Link-Level Redundancy

If using physical load balancers, the NICs can be bonded. 'Mode 1' (active-backup) NIC bonding can be used to provide link-level redundancy. This places one of the network interfaces in a backup state, and will only become active if the link is lost to the active interface. This could be used in conjunction with a redundant switch, with the **eth0** interface connected to the primary switch and **eth1** connected to the backup switch.

NIC bonding is not recommended for virtual appliances. If required, bonding should be handled at the hypervisor level.

Please refer to the Administration Manual for full details and instructions on how to configure bonding.

6.6. How the WAF Gateway and SIP Provisioning Protection Work

An entire section is dedicated to this topic. For details on how the WAF gateway and SIP provisioning protection work, see Custom WAF Rule Set Description.

6.7. Clearing the Database Files

The custom Metaswitch WAF rules use an underlying database mechanism to track user activity over time, most importantly: which users are blocked and when they are blocked until.

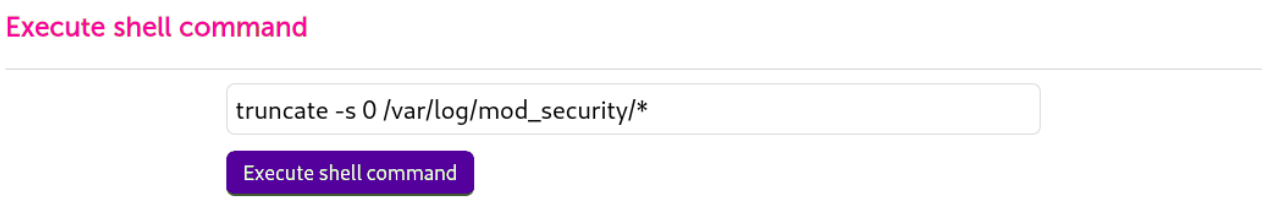
If a user is blocked in error, it is possible to clear the database files to erase an erroneous block and start from fresh. Note that **this procedure will clear all current blocks for all users**.

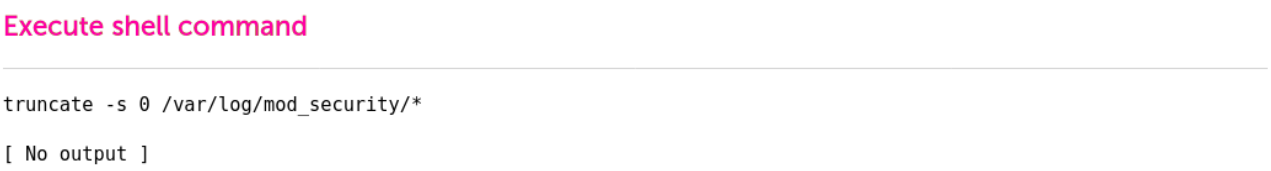
The command below can be executed from the appliance's WebUI, under ***Local Configuration > Execute shell command***. It can also be executed from the appliance console or an SSH session.

> **Note**
>
> The load balancer appliance is "secure by default". This means that the ***Execute shell command*** option, as well as (password based) console and SSH access, is disabled by default.
>
> For information on what this means, how to re-enable this functionality, and the ramifications of doing so, refer to the section ***Appliance Security - Security Mode*** in the Administration Manual.

To truncate the underlying database files to 0 bytes (emptying them), use the following command:

```
truncate -s 0 /var/log/mod_security/*
```





6.8. Log4j/Log4Shell Defence

The custom Metaswitch WAF rule set features a special rule designed to address exploits against the Log4j library described in several CVEs:

- CVE-2021-44228
- CVE-2021-44832
- CVE-2021-45046
- CVE-2021-45105

> **Note**
>
> For detected Log4Shell exploits, the attack payload is not displayed in the alert message since Log4j could potentially be executed on a log viewer.

For further details, see https://coreruleset.org/20211213/crs-and-log4j-log4shell-cve-2021-44228/.

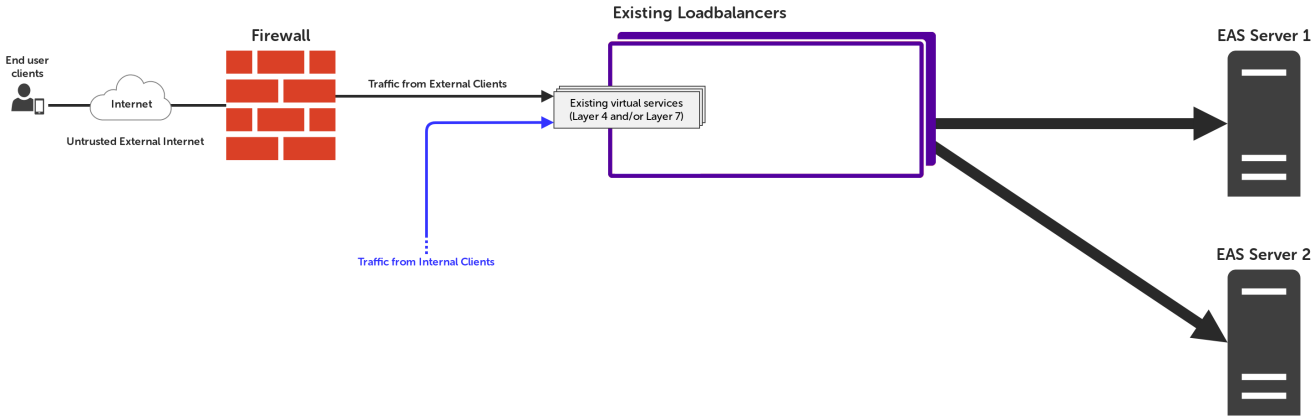
7. Deployment Concept

7.1. Scenario 1 – Adding WAF Gateways to a Deployment Currently Using Hardware Load Balancers

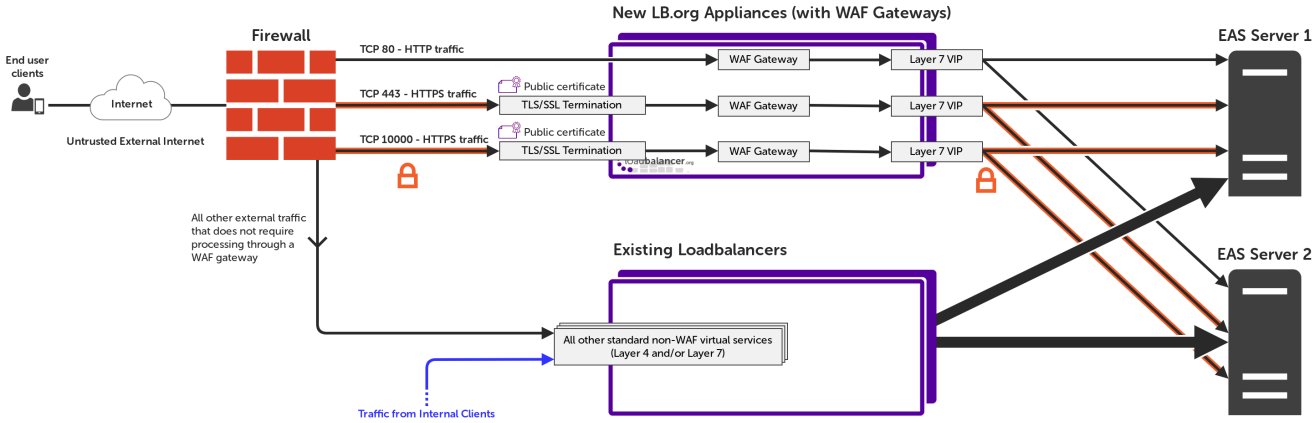
Full instructions for setting up this scenario are given in Appliance Configuration for Hardware Deployments (Scenario 1).

This scenario applies to EAS deployments that are currently being load balanced by a pair of hardware load balancers. Whether it is a long-standing **existing deployment or a new installation** which has just been configured, WAF gateways can be added to the deployment.

The existing EAS deployment should look like this:



After implementing WAF gateways, the deployment should look like this:



This scenario involves deploying a new pair of load balancers to handle the WAF gateway processing. This means that no additional load is put on the existing production hardware load balancers. It also means that no configuration changes need to be made to the existing load balancers.

7.1.1. Virtual Service (VIP) Requirements

To provide WAF gateway protection to a Metaswitch EAS deployment, one chain of virtual services is required for each Metaswitch EAS service that needs to have a WAF gateway put in front of it.

As an example, a Metaswitch EAS deployment that has three services requiring WAF protection, listening on ports 80, 10000, and 10001, would require three chains of VIPs to be configured.

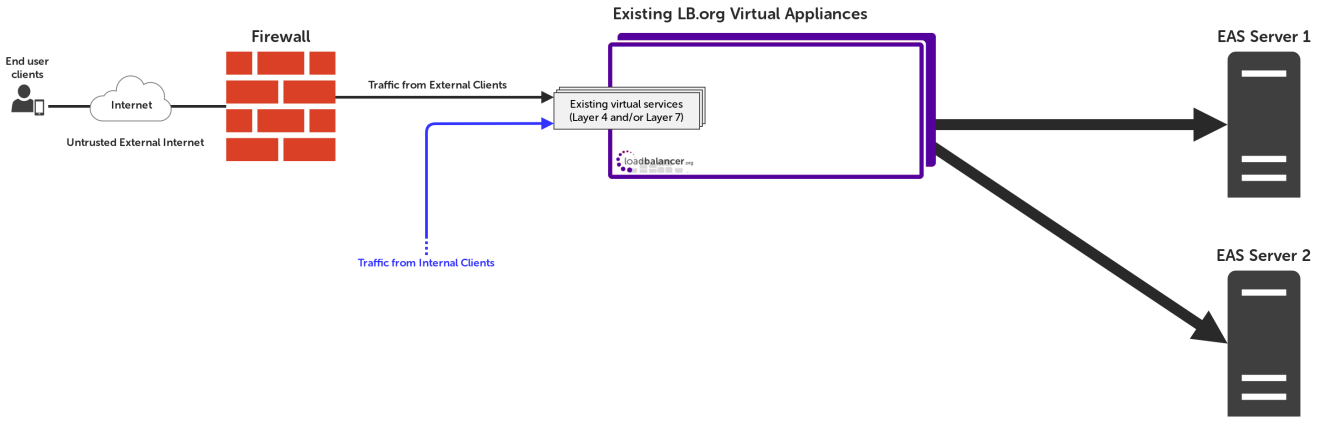
Each chain must contain a layer 7 VIP, a WAF gateway, and optionally a TLS/SSL termination service (only required for HTTPS based services).

7.2. Scenario 2 – Adding WAF Gateways to a Deployment Currently Using Virtual Load Balancers

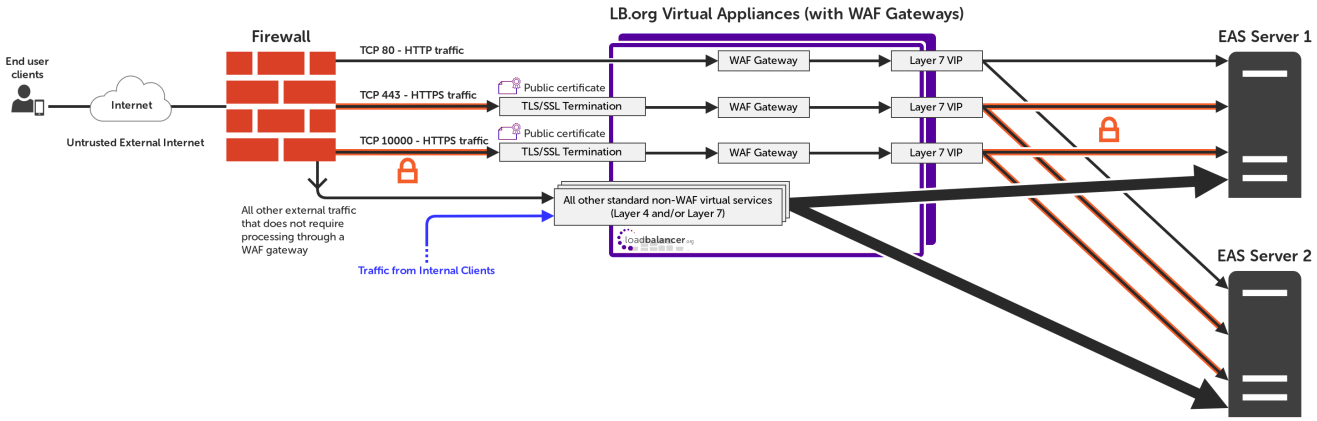
Full instructions for setting up this scenario are given in Appliance Configuration for Virtual Deployments (Scenario 2).

This scenario applies to EAS deployments that are currently being load balanced by a pair of virtual load balancers. Whether it is a long-standing **existing deployment or a new installation** which has just been configured, WAF gateways can be added to the deployment.

The existing EAS deployment should look like this:



After implementing WAF gateways, the deployment should look like this:



This scenario involves recreating some of the virtual services on the existing virtual load balancers and setting up WAF gateway services. This does mean that additional load is put on the existing production virtual load balancers, as they handle the WAF gateway processing.

The underlying virtual machines may need to be assigned additional resources in order to handle the extra load. For minimum specification guidelines, see : Sizing, Capacity, and Performance for a Virtual WAF Gateway Deployment.

7.2.1. Virtual Service (VIP) Requirements

To provide WAF gateway protection to a Metaswitch EAS deployment, one chain of virtual services is required for each Metaswitch EAS service that needs to have a WAF gateway put in front of it.

As an example, a Metaswitch EAS deployment that has three services requiring WAF protection, listening on ports 80, 10000, and 10001, would require three chains of VIPs to be configured.

Each chain must contain a layer 7 VIP, a WAF gateway, and optionally a TLS/SSL termination service (only required for HTTPS based services).

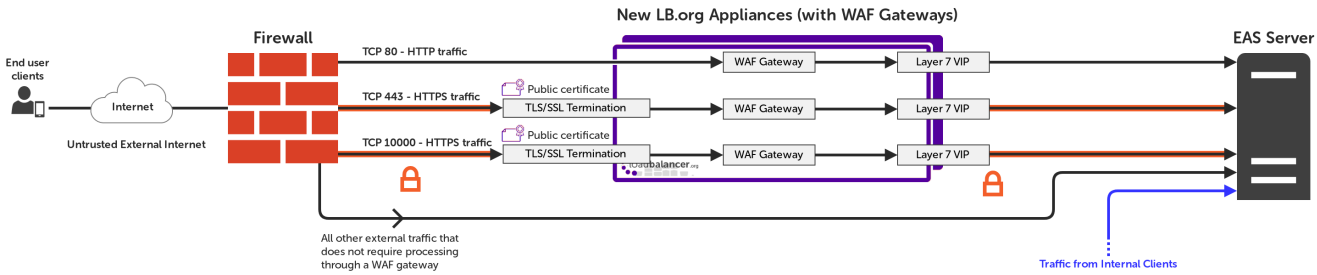
7.3. Scenario 3 – Adding WAF Gateways to a Non-Load Balanced Deployment

Full instructions for setting up this scenario are given in Appliance Configuration for Non-Load Balanced Deployments (Scenario 3).

This scenario applies to EAS deployments that are not currently being load balanced. WAF gateways can be added to the deployment.

The existing EAS deployment should look like this:

After implementing WAF gateways, the deployment should look like this:



This scenario involves deploying a pair of load balancers to handle the WAF gateway processing.

7.3.1. Virtual Service (VIP) Requirements

To provide WAF gateway protection to a Metaswitch EAS deployment, one chain of virtual services is required for each Metaswitch EAS service that needs to have a WAF gateway put in front of it.

As an example, a Metaswitch EAS deployment that has three services requiring WAF protection, listening on ports 80, 10000, and 10001, would require three chains of VIPs to be configured.

Each chain must contain a layer 7 VIP, a WAF gateway, and optionally a TLS/SSL termination service (only required for HTTPS based services).

8. Loadbalancer.org Appliance – the Basics

8.1. Virtual Appliance

A fully featured, fully supported 30 day trial is available if you are conducting a PoC (Proof of Concept) deployment. The VA is currently available for VMware, Virtual Box, Hyper-V, KVM, XEN and Nutanix AHV and has been optimized for each Hypervisor. By default, the VA is allocated 2 vCPUs, 4GB of RAM and has a 20GB virtual disk. The Virtual Appliance can be downloaded here.

Note	The same download is used for the licensed product, the only difference is that a license key file (supplied by our sales team when the product is purchased) must be applied using the appliance's WebUI.

Note	Please refer to Virtual Appliance Installation and the ReadMe.txt text file included in the VA download for additional information on deploying the VA using the various Hypervisors.

Note	The VA has 4 network adapters. For VMware only the first adapter (**eth0**) is connected by default. For HyperV, KVM, XEN and Nutanix AHV all adapters are disconnected by default. Use the network configuration screen within the Hypervisor to connect the required adapters.

8.2. Initial Network Configuration

After boot up, follow the instructions on the appliance console to configure the management IP address, subnet mask, default gateway, DNS servers and other network and administrative settings.

Important	Be sure to set a secure password for the load balancer, when prompted during the setup routine.

8.3. Accessing the Appliance WebUI

The WebUI is accessed using a web browser. By default, users are authenticated using Apache authentication. Users can also be authenticated against LDAP, LDAPS, Active Directory or Radius - for more information, please refer to External Authentication.

Note	There are certain differences when accessing the WebUI for the cloud appliances. For details, please refer to the relevant Quick Start / Configuration Guide.

1. Using a browser, navigate to the following URL:

   https://<IP-address-configured-during-the-network-setup-wizard>:9443/lbadmin/

Note	You'll receive a warning about the WebUI's SSL certificate. This is due to the default self signed certificate that is used. If preferred, you can upload your own certificate - for more

information, please refer to Appliance Security Features.

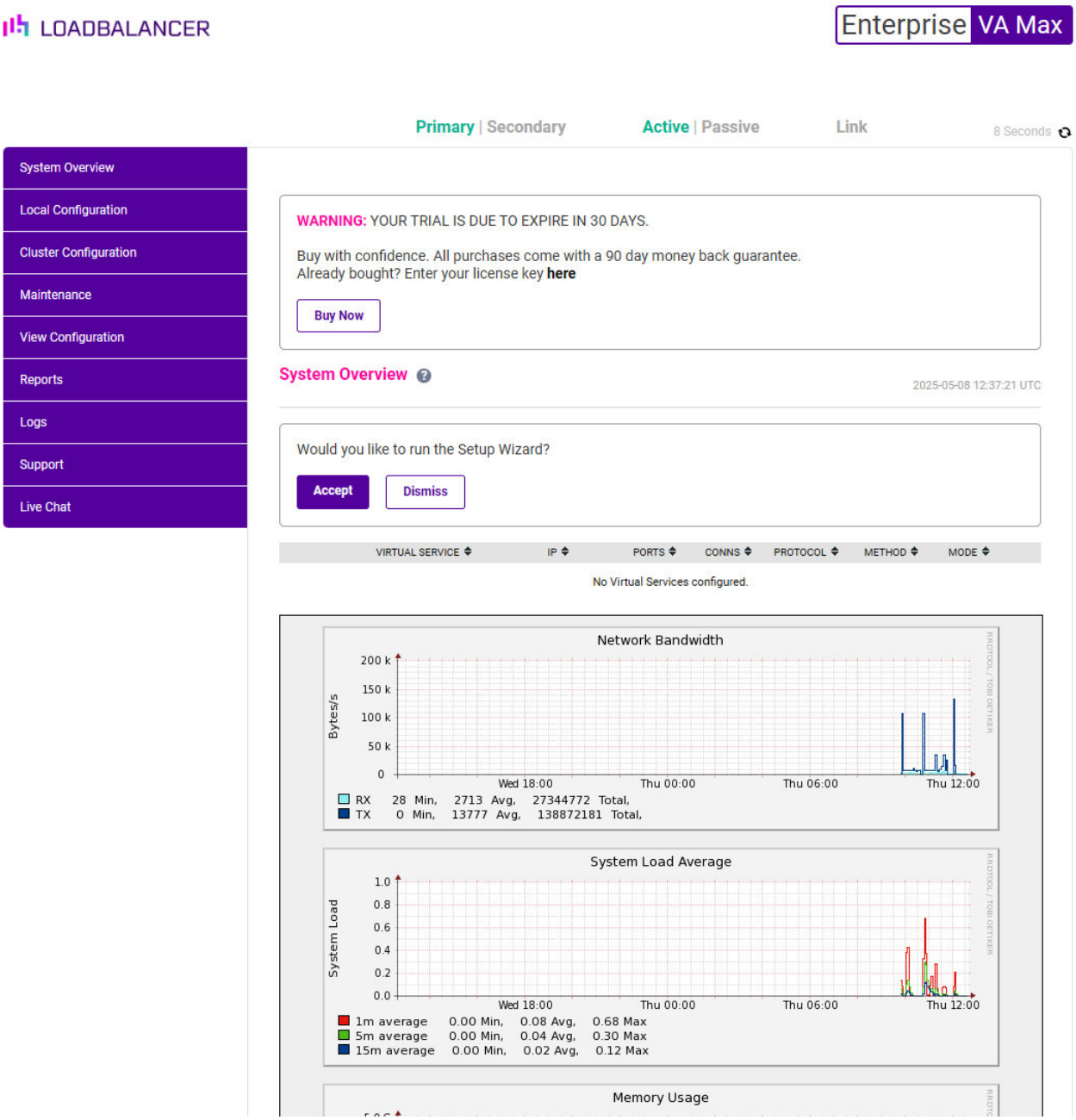
> **Note** If you need to change the port, IP address or protocol that the WebUI listens on, please refer to Service Socket Addresses.

2. Log in to the WebUI using the following credentials:

Username: loadbalancer
Password: <configured-during-network-setup-wizard>

> **Note** To change the password, use the WebUI menu option: *Maintenance > Passwords*.

Once logged in, the WebUI will be displayed as shown below:



3. You'll be asked if you want to run the Setup Wizard. Click **Dismiss** if you're following a guide or want to configure the appliance manually. Click **Accept** to start the Setup Wizard.

> **Note** The Setup Wizard can only be used to configure Layer 7 services.

8.3.1. Main Menu Options

System Overview - Displays a graphical summary of all VIPs, RIPs and key appliance statistics
Local Configuration - Configure local host settings such as IP address, DNS, system time etc.
Cluster Configuration - Configure load balanced services such as VIPs & RIPs
Maintenance - Perform maintenance tasks such as service restarts and creating backups
View Configuration - Display the saved appliance configuration settings
Reports - View various appliance reports & graphs
Logs - View various appliance logs
Support - Create a support download, contact the support team & access useful links
Live Chat - Start a live chat session with one of our Support Engineers

8.4. Appliance Software Update

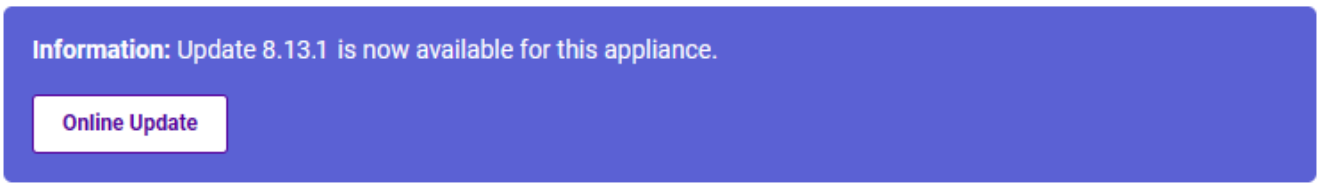
We recommend that the appliance is kept up to date to ensure that you benefit from the latest bug fixes, security updates and feature improvements. Both online and offline update are supported.

> **Note** For full details, please refer to Appliance Software Update in the Administration Manual.

> **Note** Services may need to be restarted/reloaded after the update process completes or in some cases a full appliance restart may be required. We therefore recommend performing the update during a maintenance window.

8.4.1. Online Update

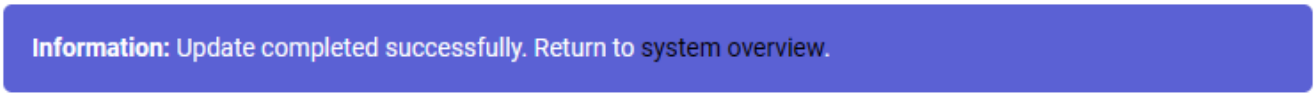
The appliance periodically contacts the Loadbalancer.org update server (**update.loadbalancer.org**) and checks for updates. This is the default behavior and can be disabled if preferred. If an update is found, a notification similar to the example below will be displayed at the top of the WebUI:



Click **Online Update**. A summary of all new features, improvements, bug fixes and security updates included in the update will be displayed. Click **Update** at the bottom of the page to start the update process.

> **Important** Do not navigate away whilst the update is ongoing, this may cause the update to fail.

The update can take several minutes depending on download speed and upgrade version. Once complete, the following message will be displayed:



If services need to be reloaded/restarted or the appliance needs a full restart, you'll be prompted accordingly.

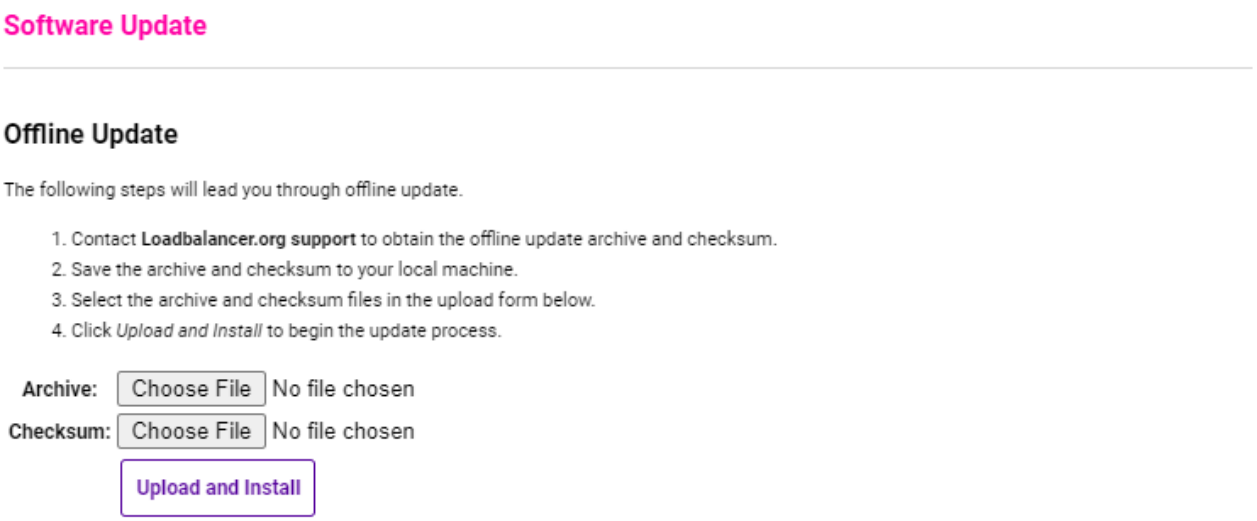
8.4.2. Offline Update

If the appliance does not have access to the Internet, offline update can be used.

To check for the latest version, please refer to our product roadmap page available here. To obtain the latest offline update files contact support@loadbalancer.org.

To perform an offline update:

1. Using the WebUI, navigate to: *Maintenance > Software Update*.
2. Select **Offline Update**.
3. The following screen will be displayed:



4. Select the *Archive* and *Checksum* files.
5. Click **Upload and Install**.
6. If services need to be reloaded/restarted or the appliance needs a full restart, you'll be prompted accordingly.

8.5. Ports Used by the Appliance

By default, the appliance uses the following TCP & UDP ports:

Protocol	Port	Purpose
TCP	22 *	SSH
TCP & UDP	53 *	DNS / GSLB
TCP & UDP	123	NTP

Protocol	Port	Purpose
TCP & UDP	161 *	SNMP
UDP	6694	Heartbeat between Primary & Secondary appliances in HA mode
TCP	7778	HAProxy persistence table replication
TCP	9000 *	Gateway service (Centralized/Portal Management)
TCP	9080 *	WebUI - HTTP (disabled by default)
TCP	9081 *	Nginx fallback page
TCP	9443 *	WebUI - HTTPS
TCP	25565 *	Shuttle service (Centralized/Portal Management)

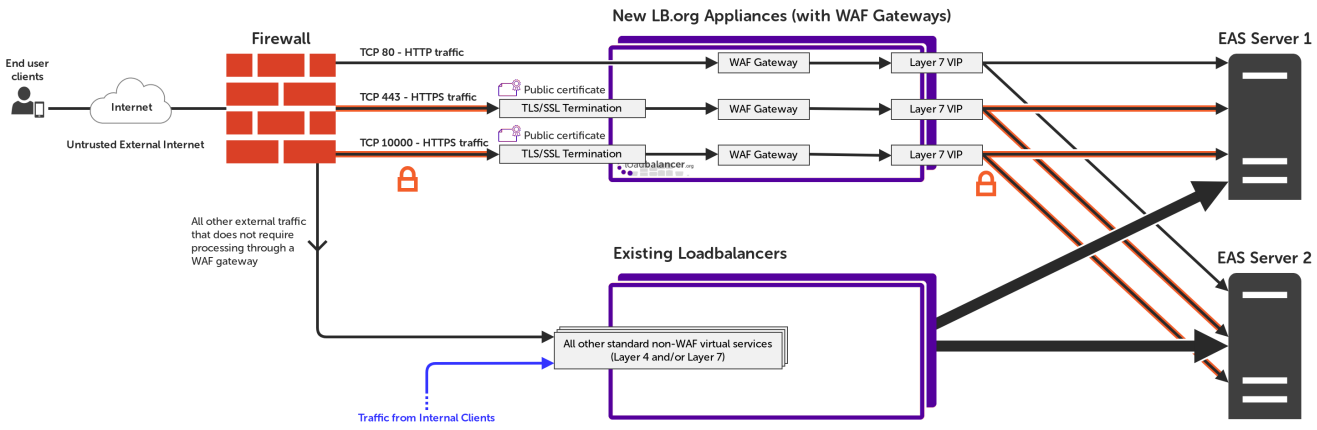
> **Note** The ports used for SSH, GSLB, SNMP, the WebUI, the fallback page, the gateway service and the shuttle service can be changed if required. For more information, please refer to Service Socket Addresses.

8.6. HA Clustered Pair Configuration

Loadbalancer.org recommend that load balancer appliances are deployed in pairs for high availability. In this guide a single unit is deployed first, adding a secondary unit is covered in Configuring HA - Adding a Secondary Appliance.

9. Appliance Configuration for Hardware Deployments (Scenario 1)

The end result should look like the following diagram:



9.1. Deploy the New Pair of Load Balancers

This scenario involves deploying a new pair of load balancers to handle the WAF gateway processing.

Each new load balancer should be deployed separately. Guidance for initial setup and deployment can be found in the section Loadbalancer.org Appliance – the Basics.

Once deployed, the new load balancers should be paired together to create a highly available clustered pair. This procedure is covered in Configuring HA - Adding a Secondary Appliance.

9.2. Assign IP addresses in the Required Subnets

The new load balancers must be assigned at least one IP address in each subnet in which they will be required to operate.

A typical EAS deployment has a dedicated EAS service network, also referred to as the 'untrusted internal network'. This network handles the traffic from external clients. The existing load balanced virtual services that handle external traffic, which are the virtual services that need WAF protection, reside in this network. As such, it may be the case that the new load balancers only need to be assigned IP addresses in the service network.

To assign IP addresses from the WebUI:

1. Navigate to *Local Configuration > Network Interface Configuration*.
2. Under *IP Address Assignment*, define the required IP addresses next to the appropriate interfaces.
3. Press the **Configure Interfaces** button to apply the configuration.

Additional IP addresses in additional networks and subnets can be assigned as required, for example in the trusted internal management network, if one exists.

9.3. Recreate the existing virtual services Requiring WAF Protection

9.3.1. Identify and Note

Identify the first virtual service that requires protection from a WAF gateway. This service will be recreated in a subsequent step, so it is important to note down the following properties of the service:

- The port that the existing virtual service is listening on (if unsure, examine the firewall NAT rule associated to the existing service and see which port it forwards traffic to).
- The IP address and port of each real server associated to the existing virtual service (it is assumed that these are known. If unsure, refer to the pools.txt file associated to the EAS deployment if one exists, or alternatively contact your Metaswitch representative).

9.3.2. Recreating the Virtual Service

The virtual service must now be recreated on the new load balancers along with an associated WAF service. Access the WebUI of the Primary appliance of the new pair of load balancers, as this is where the new configuration will be made.

Configuring the Virtual Service (VIP)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click on **Add a new Virtual Service**.
2. Define the *Label* for the virtual service as required, e.g. **SERV_HTTP-80**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **172.31.5.200**.

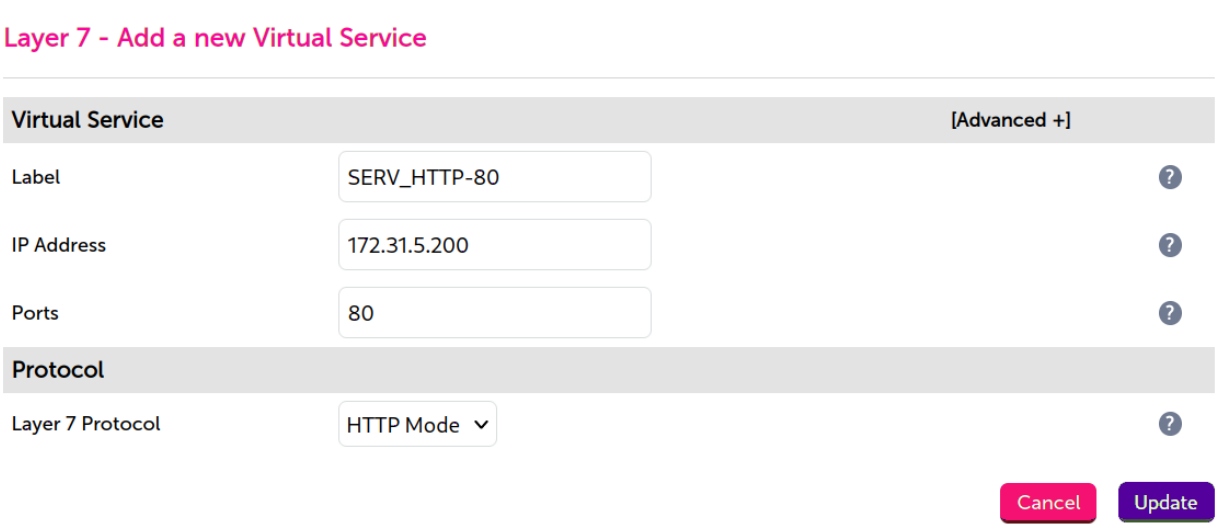
> **Note**
>
> Pick an IP address that is not currently in use on the network to be the virtual IP address.
>
> If recreating a series of virtual services that currently share the same IP address then it may make sense for the recreated services to also share a (new) IP address.
>
> Do not use the same IP address as the existing virtual service that is being replaced. Doing so would cause an IP conflict resulting in disruption to the existing production service.

4. Set the *Ports* field based on the type of traffic that the existing virtual service is processing:
   - For HTTP traffic, reuse the port that the existing virtual service is listening on, e.g. **80**.

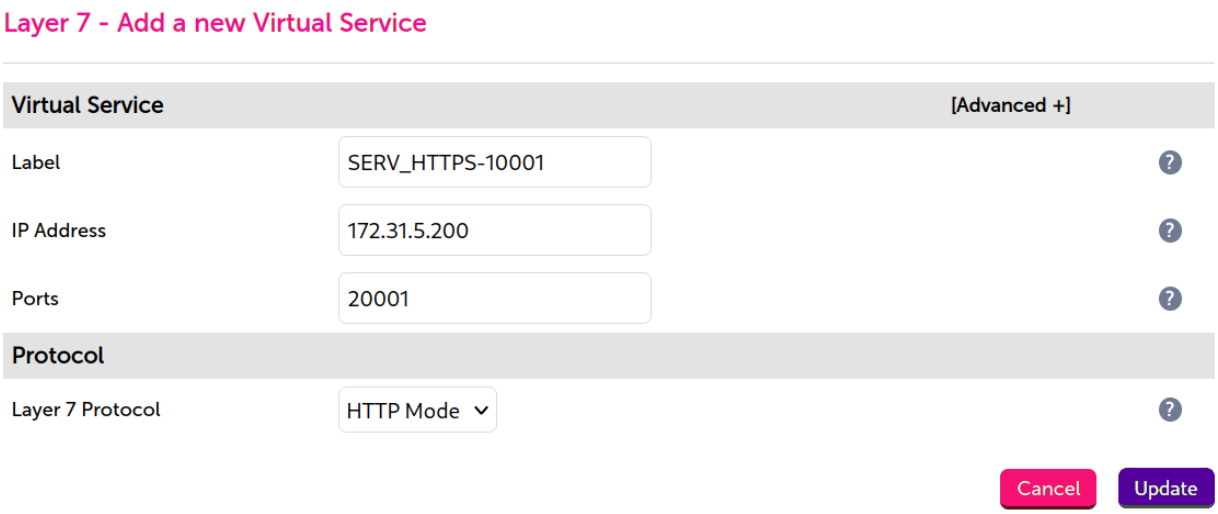
     > **Note**
     >
     > In line with Metaswitch guidance, it is strongly advised that internet facing services are HTTPS based ***only***. Plaintext HTTP services should ***not*** be used.

   - For HTTPS traffic, choose a new and an unused port to use. For example, if the existing virtual service is listening on port 10001 then perhaps use port 20001. The original port cannot be reused here as it is needed later for the TLS/SSL decryption service, which will be the front end for the chain of services.
5. Set the *Layer 7 Protocol* to **HTTP Mode**.
6. Click **Update** to create the virtual service.

Example of an HTTP service:



Example of an HTTPS service:



7. Click **Modify** next to the newly created VIP.
8. In the *Protocol* section click **Advanced** to expand the menu.
9. Enable option **Accept Invalid HTTP Requests** by checking the checkbox.
10. In the *Persistence* section click **Advanced** to expand the menu.
11. Set the *Persistence Mode* to **X-Forwarded-For and Source IP** and set the *Persistence Timeout* to **2100**.
12. If the existing virtual service is processing HTTPS traffic then the *Enable Backend Encryption* checkbox must be checked, as the back end EAS servers are expecting to receive encrypted traffic.
13. In the *Other* section click **Advanced** to expand the menu.
14. Check the **Timeout** checkbox.
15. Set *Client Timeout* to **900000** (the units are milliseconds; this value equates to 15 minutes).
16. Set *Real Server Timeout* to **901000** (the units are milliseconds; this value equates to 15 minutes 1 second).
17. Click **Update**.

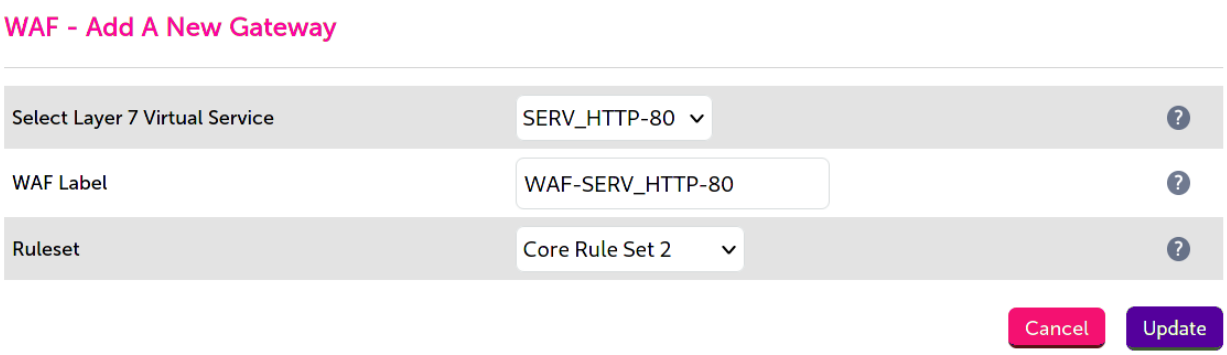
Defining the Real Servers (RIPs)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Real Servers* and click on **Add a new Real Server** next to the newly created VIP.
2. Define the *Label* for the real server as required, e.g. **EAS-SSS-1**.
3. Set the *Real Server IP Address* field to the IP address of the EAS server, e.g. **172.60.5.101**.
4. Set the *Real Server Port* field to the required port, e.g. **80**.
5. Click **Update**.
6. Repeat these steps to add each EAS server.

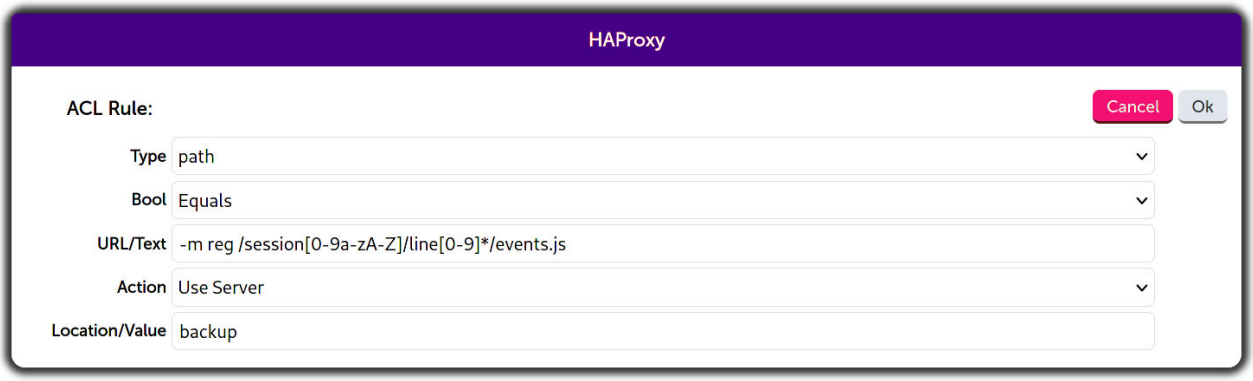
9.3.3. Creating the WAF Gateway

1. Using the web user interface, navigate to *Cluster Configuration > WAF – Gateway* and click on **Add a new WAF gateway**.

2. From the *Select Layer 7 Virtual Service* drop-down list select the associated layer 7 service that was just created, which in this example is **SERV_HTTP-80**.
3. Set *Ruleset* to **Core Rule Set 2**.
4. Click **Update** to create the WAF gateway.



5. Click **Modify** next to the newly created WAF service.
6. Set *Inbound Anomaly Score* to **12**.
7. Set *WAF Proxy Timeout* to **900** (the units are seconds).
8. Click **Update**.
9. Navigate to *Cluster Configuration > Layer 7 – Virtual Services*.
10. Click **Modify** next to the automatically created WAF VIP, e.g. WAF-SERV_HTTP-80.
11. In the *Protocol* section click **Advanced** to expand the menu.
12. Enable option **Accept Invalid HTTP Requests** by checking the checkbox.
13. Under *ACL Rules* click **Add Rule**.
14. Set *Type* to **path**.
15. Set *Bool* to **Equals**.
16. Set *URL/Text* to **-m reg /session[0-9a-zA-Z]+/line[0-9]*/events.js**
17. Set *Action* to **Use Server**.
18. Set *Location/Value* to **backup**.
19. Click **Ok**.



20. In the *Other* section click **Advanced** to expand the menu.
21. Check the **Timeout** checkbox.
22. Set *Client Timeout* to **900000** (the units are milliseconds; this value equates to 15 minutes).
23. Set *Real Server Timeout* to **901000** (the units are milliseconds; this value equates to 15 minutes 1 second).
24. Click **Update**.
25. Navigate to *Cluster Configuration > WAF – Manual Configuration*.
26. Using the drop-down list, select the WAF gateway that was previously created, e.g. **WAF-SERV_HTTP-80**. The existing default WAF configuration will appear in the text box.
27. Paste your custom Metaswitch EAS WAF configuration into the text box.
28. Click **Update** to save the manual WAF configuration.

9.3.4. Setting Up the TLS/SSL Termination

If the existing virtual service that is being recreated handles **plaintext HTTP** traffic (not encrypted) then this step can be ignored, and the next applicable step will be Recreating the Remaining Existing Virtual Services.

If the existing virtual service that is being recreated handles **HTTPS** traffic (encrypted) then a TLS/SSL termination service must be created. This is required so that the traffic can be decrypted on the load balancer and fed into the WAF gateway service as plaintext HTTP traffic for scanning.

Uploading the Certificate

The appropriate certificate for the service in question must be uploaded to the load balancer for TLS/SSL termination to work. The process for doing this is as follows:

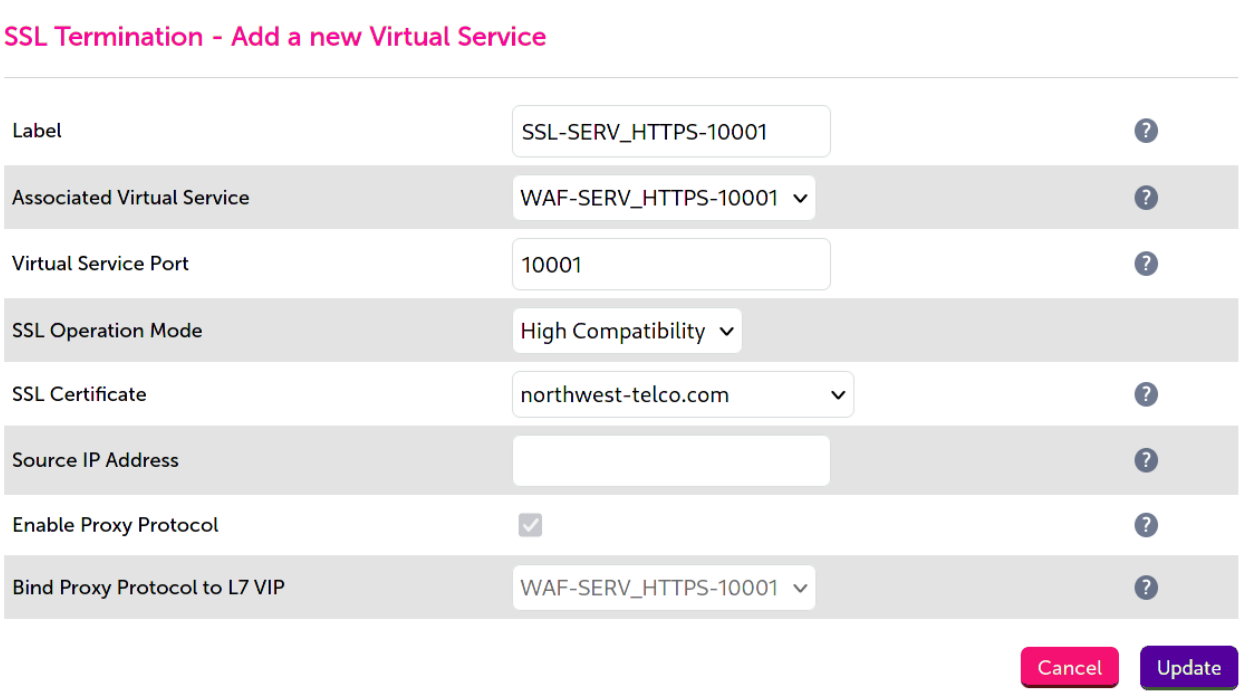
1. Using the web user interface, navigate to *Cluster Configuration > SSL Certificate* and click on **Add a new SSL Certificate**.
2. Press the *Upload prepared PEM/PFX file* radio button.
3. Define the *Label* for the certificate as required. It may make sense to use the domain that the certificate is associated to, e.g. **northwest-telco.com**.
4. Click on **Browse** and select the appropriate PEM or PFX style certificate.
5. If uploading a PFX certificate, enter the certificate's password in the *PFX File Password* field.

6. Click **Upload certificate**.

For more information on creating PEM certificate files and converting between certificate formats please refer to Creating a PEM File.

Creating the TLS/SSL Termination

1. Using the web user interface, navigate to *Cluster Configuration > SSL Termination* and click on **Add a new Virtual Service**.
2. From the *Associated Virtual Service* drop-down list, select the associated WAF gateway that was created previously, e.g. **WAF-SERV_HTTPS-10001**.
3. Set the *Virtual Service Port* field to the port that the existing service that is being replicated is listening on, e.g. **10001**.
4. From the *SSL Operation Mode* drop-down list, select **High Compatibility**.
5. From the *SSL Certificate* drop-down list, select the certificate for the service in question, which in this example is **northwest-telco.com**.
6. Click **Update** to create the TLS/SSL termination service.



9.3.5. Recreating the Remaining Existing Virtual Services

Repeat the steps listed above for every additional virtual service that needs to be protected with a WAF gateway. The instructions in this section begin here.

9.4. Setting the PCRE Match Limits

9.4.1. Setting the Recommended Values

1. Using the WebUI, navigate to: *Cluster Configuration > WAF – Advanced Configuration*.

2. Set *PCRE Match Limit* to **500000**.
3. Set *PCRE Match Limit Recursion* to **500000**.
4. Click the **PCRE Match Limit Recursion** button.

9.4.2. Explanation

PCRE is an acronym for Perl Compatible Regular Expressions, which is a pattern matching library. It implements a regular expression engine, which the WAF uses to inspect HTTP traffic for malicious looking behaviour.

There are two PCRE match limits in the WAF implementation. These limits define the maximum number of calls that can be made to the underlying match function when the WAF evaluates a regular expression. Having such limits prevents PCRE from consuming huge amounts of system resources. It also protects against specially crafted regular expression attacks designed to overpower a WAF by forcing it to continuously perform a substantial number of pattern matches.

PCRE match limit values of 500000 are proven and sensible values to use in production.

9.5. Finalizing the Configuration

Once all of the virtual services that need WAF protection have been recreated on the new load balancer pair, the new configuration needs to be finalized and put into use by reloading the appropriate services.

To apply the new settings, HAProxy, the WAF and STunnel must be reloaded. This can be done using the buttons in the "Commit changes" box at the top of the screen or by using the *Restart Services* menu option:

1. Using the WebUI, navigate to: *Maintenance > Restart Services*.
2. Click **Reload HAProxy**.
3. Click **Reload WAF**.
4. Click **Reload STunnel**.

9.6. Putting the New WAF Services Into Production (Detection Only Mode)

Once they have been fully configured, the new WAF protected services can be put into production use.

By default, the newly created WAF services will operate in detection only mode. This means that the WAF services will scan traffic and perform logging, but will ***not*** perform any blocking actions. This reduces the risk of disrupting traffic to zero for practical purposes.

The WAF solution can be left in detection only mode for a length of time prior to fully enabling the WAF rule engine. Once fully enabled, the WAF services will perform blocking actions to actively block traffic that appears to be malicious. Operating for a period in detection only mode allows for observing how the WAF ***would*** function in full production ***before*** fully enabling it and allowing it to block traffic.

9.6.1. Directing Traffic to the WAF Services

It is likely the case that traffic from external users passes through an external-facing firewall before being sent on toward the EAS cluster. The firewall would have NAT rules configured to pass the external production traffic on toward the EAS cluster. These NAT rules would need to be modified on the firewall to send the external traffic to

the new WAF protected services.

The exact procedure for changing firewall NAT rules varies between firewall vendors and is outside the scope of this document.

It is possible that instead of making firewall changes, it may be necessary to change the DNS record for a service's FQDN to point to the IP address of the new WAF services.

For deployments that do not use firewall NAT rules or DNS records to direct production traffic, or for other related inquiries, please contact support@loadbalancer.org for further advice.

9.7. Fully Enabling the WAF Rule Engine

The WAF rule engine needs to be enabled for a given WAF service to allow it to actively block traffic that appears to be malicious.

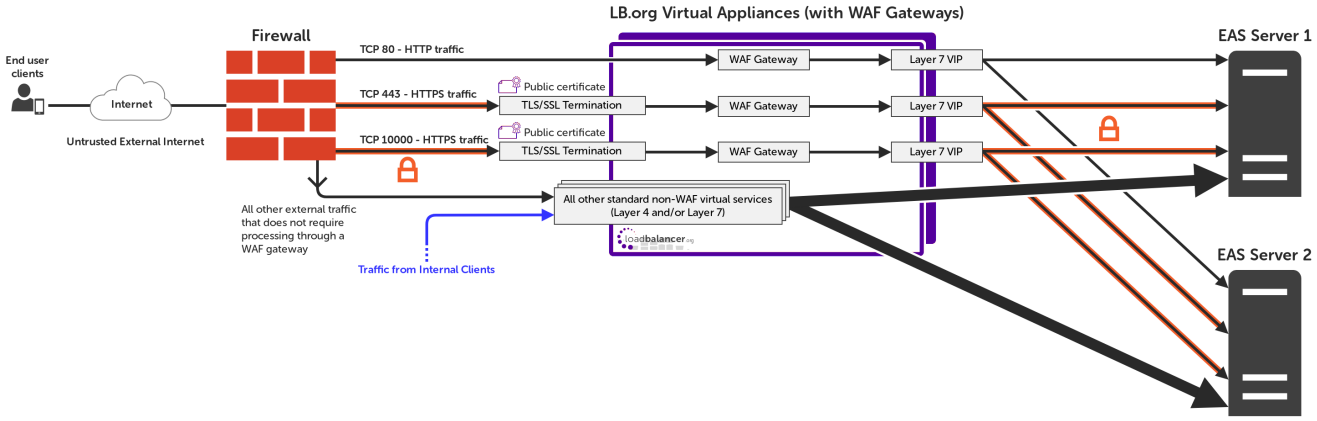
To enable the WAF rule engine for a given WAF service:

1. Using the WebUI, navigate to: *Cluster Configuration > WAF – Gateway*.
2. Click on **Modify** next to the WAF service in question.
3. Check the **Rule Engine Traffic Blocking** checkbox.

4. Click **Update**.
5. Click **Reload HAProxy** and **Reload WAF** when prompted, to put the change into effect.

10. Appliance Configuration for Virtual Deployments (Scenario 2)

The end result should look like the following diagram:



10.1. Considering the Resources Assigned to the Virtual Load Balancers

The WAF gateway functionality on the load balancer is CPU and (particularly) memory intensive. The underlying virtual machines may need to be assigned additional resources in order to handle the extra load. For minimum specification guidelines, please see section: Sizing, Capacity, and Performance for a Virtual WAF Gateway Deployment.

If a virtual load balancer's assigned resources do not meet the minimum specifications, the load balancer should be assigned additional resources before attempting to implement WAF functionality. This needs to be done at the hypervisor level.

10.2. Recreate the existing virtual services Requiring WAF Protection

10.2.1. Identify and Note

For a deployment using virtual load balancers, the new WAF-related services are added to the existing virtual appliances. Access to the WebUI is assumed for this process. Contact support@loadbalancer.org for any issues regarding accessing the virtual load balancers.

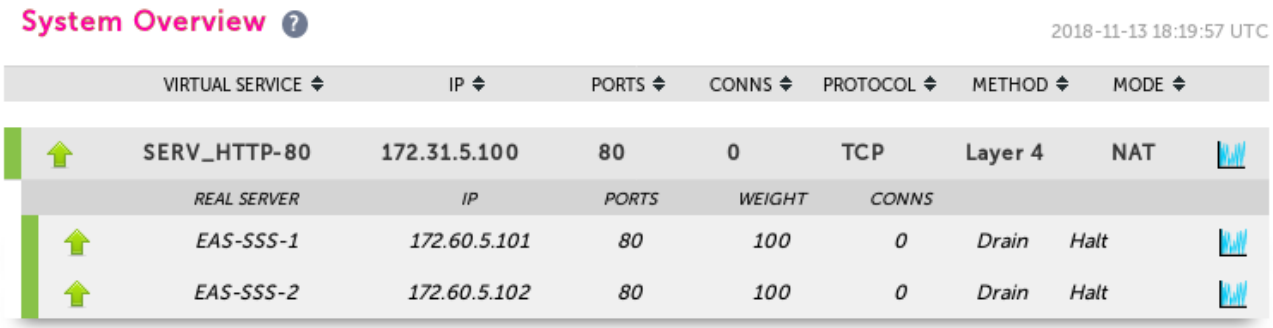
Identify the first virtual service that requires protection from a WAF gateway. This service will be recreated in a subsequent step, so it is important to note down the following properties of the service:

- The name of the virtual service.
- The port that the virtual service is listening on.
- The IP address and port of each real server associated to the virtual service.

(The end result is to leave the existing, functioning virtual services in place, while creating WAF enabled alternative services. It should be possible to fall back to the original, non-WAF services in the future, should the WAF functionality be retired or no longer required for any reason.)

Example

Taking an external-facing HTTP service as an example:



It can be seen that the service name is **SERV_HTTP-80**, it is listening on port **80**, and it has two real servers defined which are at **172.60.5.101** and **172.60.5.102**, with both servers listening on port **80**.

10.2.2. Recreating the Virtual Service

The virtual service must now be recreated along with an associated WAF service. The new configuration will be

made alongside the existing production configuration, while leaving the working production virtual service untouched.

Configuring the Virtual Service (VIP)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click on **Add a new Virtual Service**.
2. Define the *Label* for the virtual service as required, e.g. **NEW_SERV_HTTP-80**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **172.31.5.200**.

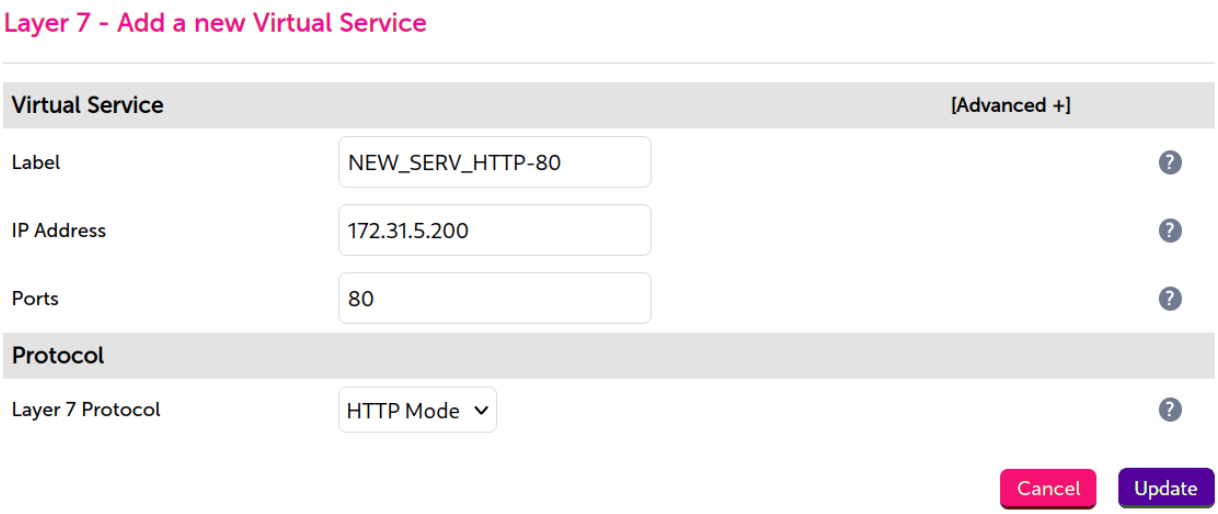
> **Note**
>
> Pick an IP address that is not currently in use on the network to be the virtual IP address.
>
> If recreating a series of virtual services that currently share the same IP address then it may make sense for the recreated services to also share a (new) IP address.
>
> It is not possible to use the same IP address as the existing virtual service that is being replaced. Doing so would cause an IP conflict resulting in disruption to the existing service, and so the WebUI does not allow this.

4. Set the *Ports* field based on the type of traffic that the existing virtual service is processing:
   - For HTTP traffic, reuse the port that the existing virtual service is listening on, e.g. **80**.

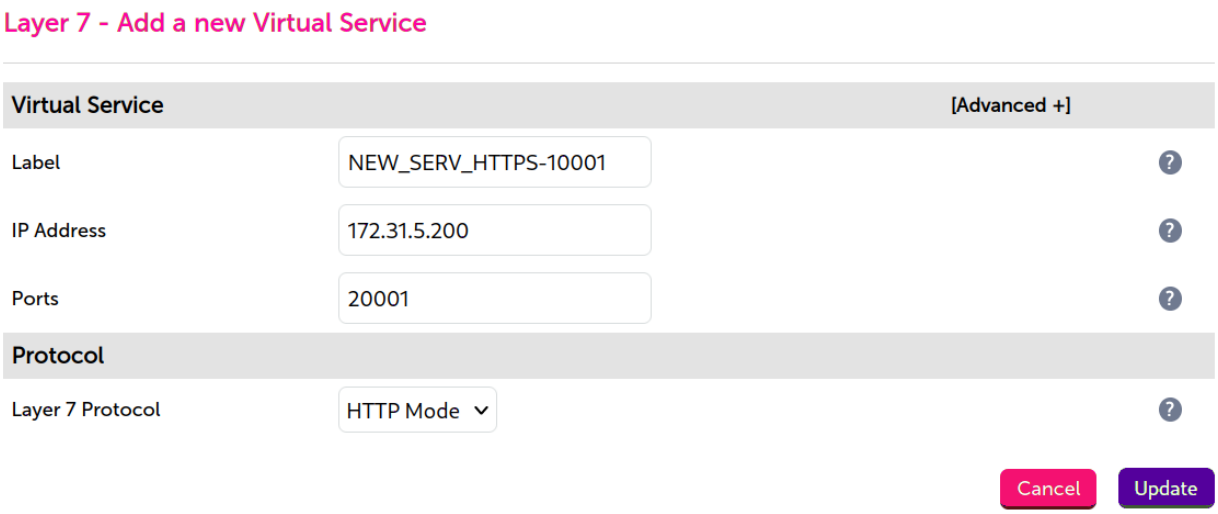
     > **Note**
     >
     > In line with Metaswitch guidance, it is strongly advised that internet facing services are HTTPS based ***only***. Plaintext HTTP services should ***not*** be used.

   - For HTTPS traffic, choose a new and an unused port to use. For example, if the existing virtual service is listening on port 10001 then perhaps use port 20001. The original port cannot be reused here as it is needed later for the TLS/SSL decryption service, which will be the front end for the chain of services.
5. Set the *Layer 7 Protocol* to **HTTP Mode**.
6. Click **Update** to create the virtual service.

Example of an HTTP service:



Example of an HTTPS service:



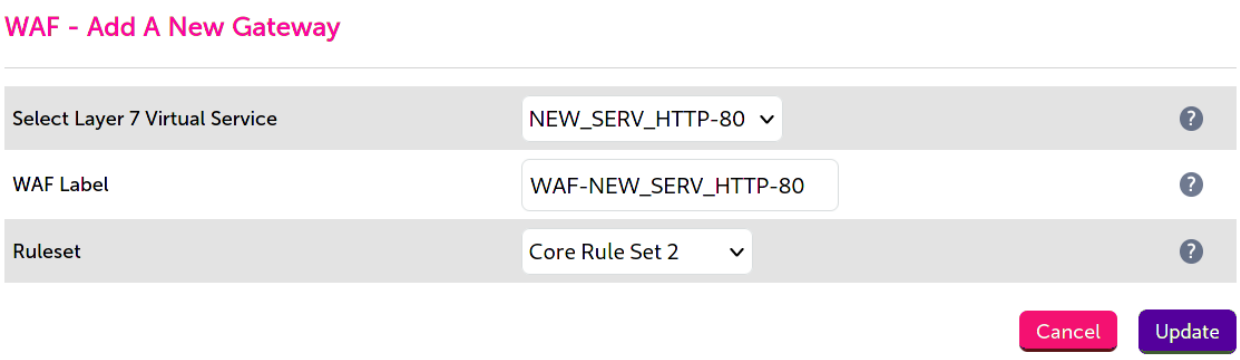
7. Click **Modify** next to the newly created VIP.
8. In the *Protocol* section click **Advanced** to expand the menu.
9. Enable option **Accept Invalid HTTP Requests** by checking the checkbox.
10. In the *Persistence* section click **Advanced** to expand the menu.
11. Set the *Persistence Mode* to **X-Forwarded-For and Source IP** and set the *Persistence Timeout* to **2100**.
12. If the existing virtual service is processing HTTPS traffic then the ***Enable Backend Encryption*** checkbox must be checked, as the back end EAS servers are expecting to receive encrypted traffic.
13. In the *Other* section click **Advanced** to expand the menu.
14. Check the **Timeout** checkbox.
15. Set *Client Timeout* to **900000** (the units are milliseconds; this value equates to 15 minutes).
16. Set *Real Server Timeout* to **901000** (the units are milliseconds; this value equates to 15 minutes 1 second).
17. Click **Update**.

Defining the Real Servers (RIPs)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Real Servers* and click on **Add a new Real Server** next to the newly created VIP.
2. Define the *Label* for the real server as required, e.g. **EAS-SSS-1**.
3. Set the *Real Server IP Address* field to the IP address of the EAS server, e.g. **172.60.5.101**.
4. Set the *Real Server Port* field to the required port, e.g. **80**.
5. Click **Update**.
6. Repeat these steps to add each EAS server.

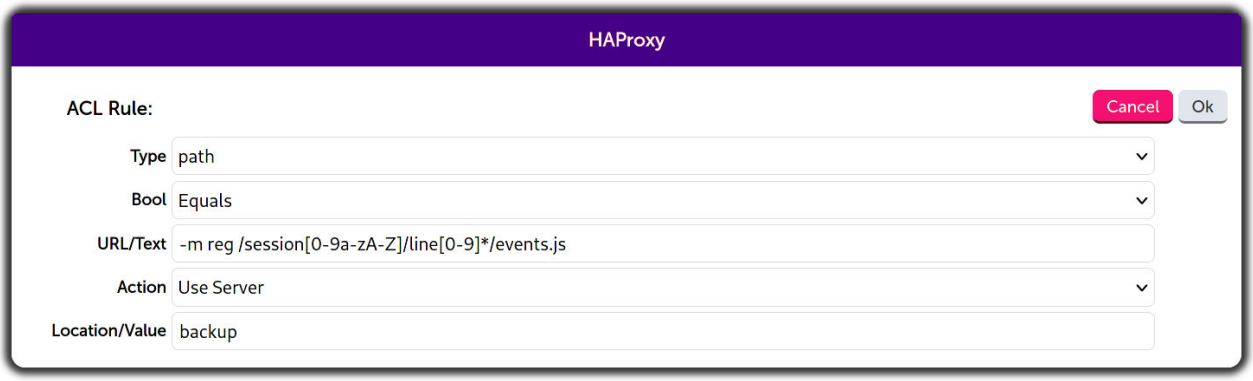
10.2.3. Creating the WAF Gateway

1. Using the web user interface, navigate to *Cluster Configuration > WAF – Gateway* and click on **Add a new WAF gateway**.
2. From the *Select Layer 7 Virtual Service* drop-down list select the associated layer 7 service that was just created, which in this example is **NEW_SERV_HTTP-80**.
3. Set *Ruleset* to **Core Rule Set 2**.
4. Click **Update** to create the WAF gateway.



5. Click **Modify** next to the newly created WAF service.
6. Set *Inbound Anomaly Score* to **12**.
7. Set *WAF Proxy Timeout* to **900** (the units are seconds).
8. Click **Update**.
9. Navigate to *Cluster Configuration > Layer 7 – Virtual Services*.
10. Click **Modify** next to the automatically created WAF VIP, e.g. WAF-NEW_SERV_HTTP-80.
11. In the *Protocol* section click **Advanced** to expand the menu.
12. Enable option **Accept Invalid HTTP Requests** by checking the checkbox.
13. Under *ACL Rules* click **Add Rule**.
14. Set *Type* to **path**.
15. Set *Bool* to **Equals**.

16. Set *URL/Text* to **-m reg /session[0-9a-zA-Z]+/line[0-9]*/events.js**
17. Set *Action* to **Use Server**.
18. Set *Location/Value* to **backup**.
19. Click **Ok**.



20. In the *Other* section click **Advanced** to expand the menu.
21. Check the **Timeout** checkbox.
22. Set *Client Timeout* to **900000** (the units are milliseconds; this value equates to 15 minutes).
23. Set *Real Server Timeout* to **901000** (the units are milliseconds; this value equates to 15 minutes 1 second).
24. Click **Update**.
25. Navigate to *Cluster Configuration > WAF – Manual Configuration*.
26. Using the drop-down list, select the WAF gateway that was just created, e.g. **WAF-NEW_SERV_HTTP-80**. The existing default WAF configuration will appear in the text box.
27. Paste your custom Metaswitch EAS WAF configuration into the text box.
28. Click **Update** to save the manual WAF configuration.

10.2.4. Setting Up the TLS/SSL Termination

If the existing virtual service that is being recreated handles **plaintext HTTP** traffic (not encrypted) then this step can be ignored, and the next applicable step will be Recreating the Remaining Existing Virtual Services.

If the existing virtual service that is being recreated handles **HTTPS** traffic (encrypted) then a TLS/SSL termination service must be created. This is required so that the traffic can be decrypted on the load balancer and fed into the WAF gateway service as plaintext HTTP traffic for scanning.

Uploading the Certificate

The appropriate certificate for the service in question must be uploaded to the load balancer for TLS/SSL termination to work. The process for doing this is as follows:

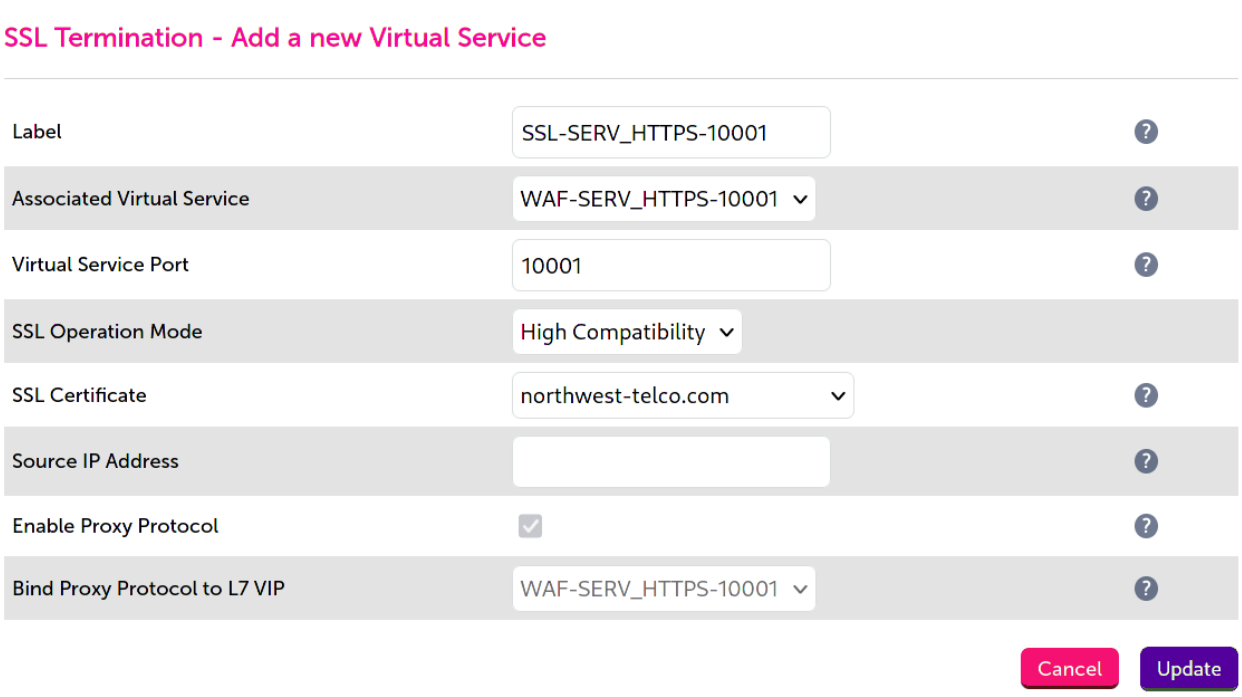
1. Using the web user interface, navigate to *Cluster Configuration > SSL Certificate* and click on **Add a new SSL Certificate**.

2. Press the *Upload prepared PEM/PFX file* radio button.
3. Define the *Label* for the certificate as required. It may make sense to use the domain that the certificate is associated to, e.g. **northwest-telco.com**.
4. Click on **Browse** and select the appropriate PEM or PFX style certificate.
5. If uploading a PFX certificate, enter the certificate's password in the *PFX File Password* field.
6. Click **Upload certificate**.

For more information on creating PEM certificate files and converting between certificate formats please refer to Creating a PEM File.

Creating the TLS/SSL Termination

1. Using the web user interface, navigate to *Cluster Configuration > SSL Termination* and click on **Add a new Virtual Service**.
2. From the *Associated Virtual Service* drop-down list, select the associated WAF gateway that was created previously, e.g. **WAF-NEW_SERV_HTTPS-10001**.
3. Set the *Virtual Service Port* field to the port that the existing service that is being replicated is listening on, e.g. **10001**.
4. From the *SSL Operation Mode* drop-down list, select **High Compatibility**.
5. From the *SSL Certificate* drop-down list, select the certificate for the service in question, which in this example is **northwest-telco.com**.
6. Click **Update** to create the TLS/SSL termination service.



10.2.5. Recreating the Remaining Existing Virtual Services

Repeat the steps listed above for every additional virtual service that needs to be protected with a WAF gateway. The instructions in this section begin here.

10.3. Setting the PCRE Match Limits

10.3.1. Setting the Recommended Values

1. Using the WebUI, navigate to: *Cluster Configuration > WAF – Advanced Configuration*.
2. Set *PCRE Match Limit* to **500000**.
3. Set *PCRE Match Limit Recursion* to **500000**.
4. Click the **PCRE Match Limit Recursion** button.

10.3.2. Explanation

PCRE is an acronym for Perl Compatible Regular Expressions, which is a pattern matching library. It implements a regular expression engine, which the WAF uses to inspect HTTP traffic for malicious looking behaviour.

There are two PCRE match limits in the WAF implementation. These limits define the maximum number of calls that can be made to the underlying match function when the WAF evaluates a regular expression. Having such limits prevents PCRE from consuming huge amounts of system resources. It also protects against specially crafted regular expression attacks designed to overpower a WAF by forcing it to continuously perform a substantial number of pattern matches.

PCRE match limit values of 500000 are proven and sensible values to use in production.

10.4. Finalizing the Configuration

Once all of the virtual services that need WAF protection have been recreated on the new load balancer pair, the new configuration needs to be finalized and put into use by reloading the appropriate services.

To apply the new settings, HAProxy, the WAF and STunnel must be reloaded. This can be done using the buttons in the "Commit changes" box at the top of the screen or by using the *Restart Services* menu option:

1. Using the WebUI, navigate to: *Maintenance > Restart Services*.
2. Click **Reload HAProxy**.
3. Click **Reload WAF**.
4. Click **Reload STunnel**.

10.5. Putting the New WAF Services Into Production (Detection Only Mode)

Once they have been fully configured, the new WAF protected services can be put into production use.

By default, the newly created WAF services will operate in detection only mode. This means that the WAF services will scan traffic and perform logging, but will ***not*** perform any blocking actions. This reduces the risk of disrupting traffic to zero for practical purposes.

The WAF solution can be left in detection only mode for a length of time prior to fully enabling the WAF rule engine. Once fully enabled, the WAF services will perform blocking actions to actively block traffic that appears to be malicious. Operating for a period in detection only mode allows for observing how the WAF ***would*** function in full production ***before*** fully enabling it and allowing it to block traffic.

10.5.1. Directing Traffic to the WAF Services

It is likely the case that traffic from external users passes through an external-facing firewall before being sent on toward the EAS cluster. The firewall would have NAT rules configured to pass the external production traffic on toward the EAS cluster. These NAT rules would need to be modified on the firewall to send the external traffic to the new WAF protected services.

The exact procedure for changing firewall NAT rules varies between firewall vendors and is outside the scope of this document.

It is possible that instead of making firewall changes, it may be necessary to change the DNS record for a service's FQDN to point to the IP address of the new WAF services.

For deployments that do not use firewall NAT rules or DNS records to direct production traffic, or for other related inquiries, please contact support@loadbalancer.org for further advice.

10.6. Fully Enabling the WAF Rule Engine

The WAF rule engine needs to be enabled for a given WAF service to allow it to actively block traffic that appears to be malicious.

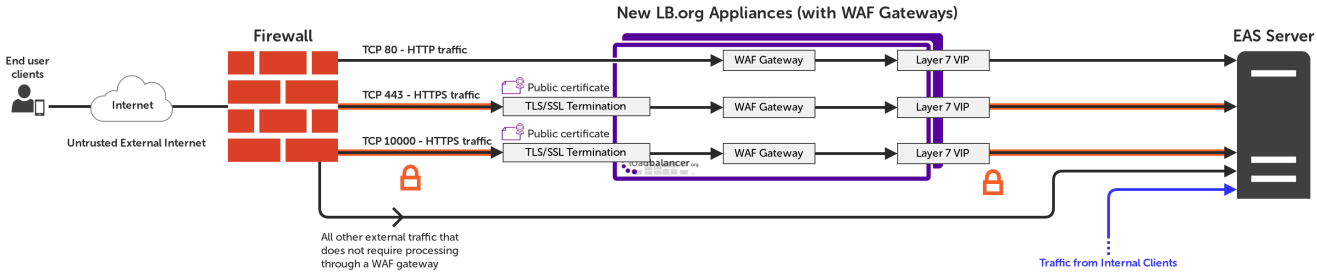
To enable the WAF rule engine for a given WAF service:

1. Using the WebUI, navigate to: *Cluster Configuration > WAF – Gateway*.
2. Click on **Modify** next to the WAF service in question.
3. Check the **Rule Engine Traffic Blocking** checkbox.

4. Click **Update**.
5. Click **Reload HAProxy** and **Reload WAF** when prompted, to put the change into effect.

11. Appliance Configuration for Non-Load Balanced Deployments (Scenario 3)

The end result should look like the following diagram:



11.1. Deploy the Pair of Load Balancers

This scenario involves deploying a pair of load balancers to handle the WAF gateway processing.

Each load balancer should be deployed separately. Guidance for initial setup and deployment can be found in the section *Loadbalancer.org Appliance – the Basics*.

Once deployed, the load balancers should be paired together to create a highly available clustered pair. This procedure is covered in Configuring HA - Adding a Secondary Appliance.

11.2. Assign IP addresses in the Required Subnets

The load balancers must be assigned at least one IP address in each subnet in which they will be required to operate. For example, if the EAS deployment is split into different networks (such as 'trusted traffic' and 'untrusted traffic' networks) and there are services requiring WAF protection in those different networks then the load balancers will need IP addresses in those networks. If the EAS deployment is situated in a single network then the load balancer will only need a single base IP address, which would sit in that network.

To assign IP addresses from the WebUI:

1. Navigate to *Local Configuration > Network Interface Configuration*.
2. Under *IP Address Assignment*, define the required IP addresses next to the appropriate interfaces.
3. Press the **Configure Interfaces** button to apply the configuration.

Additional IP addresses in additional networks and subnets can be assigned as required, for example in the trusted internal management network, if one exists.

11.3. Identify the services Requiring WAF Protection

11.3.1. Identify and Note

The EAS server provides a series of services, which may be listening on different IP addresses and ports.

Identify the first service that requires protection from a WAF gateway. It is important to note down the following properties of the service:

- The IP address that the service is listening on
- The port that the service is listening on

For example, an EAS server may be providing an HTTP service on IP address **172.60.5.101** on port 80.

11.3.2. Creating the Virtual Service

A virtual service must now be created on the load balancers along with an associated WAF service. Access the WebUI of the Primary appliance of the pair of load balancers, as this is where the configuration will be made.

Configuring the Virtual Service (VIP)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Virtual Services* and click on **Add a new Virtual Service**.

2. Define the *Label* for the virtual service as required, e.g. **HTTP-80**.
3. Set the *Virtual Service IP Address* field to the required IP address, e.g. **172.31.5.200**.

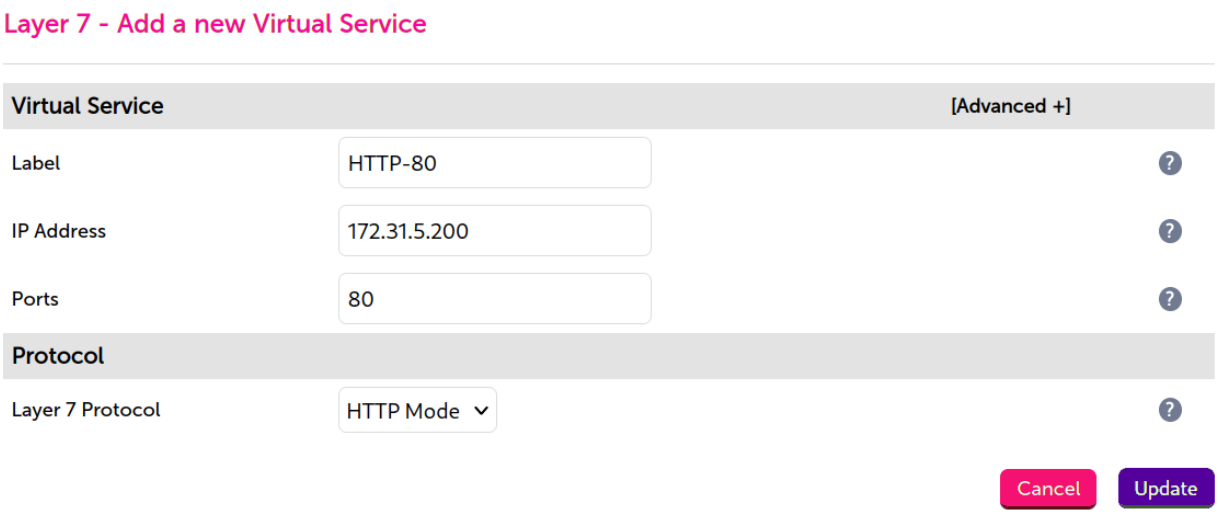
> **Note**
>
> Pick an IP address that is not currently in use on the network to be the virtual IP address.
>
> If creating a series of virtual services to sit in front of a series of services on the EAS server that currently all use the same IP address then it may make sense for the virtual services to also share a (new) IP address. For example, if you are putting WAF gateways in front of three EAS services that all listen on the IP address 10.10.10.100 then consider using the same IP address for all three of the associated virtual services that need to be created on the load balancers, such as the address 10.10.10.200.
>
> Do not use the same IP address as the existing service on the EAS server. Doing so would cause an IP conflict resulting in disruption to the existing production service.

4. Set the *Ports* field based on the type of traffic that the service processes:
   - For HTTP traffic, reuse the port that the existing service is listening on, e.g. **80**.

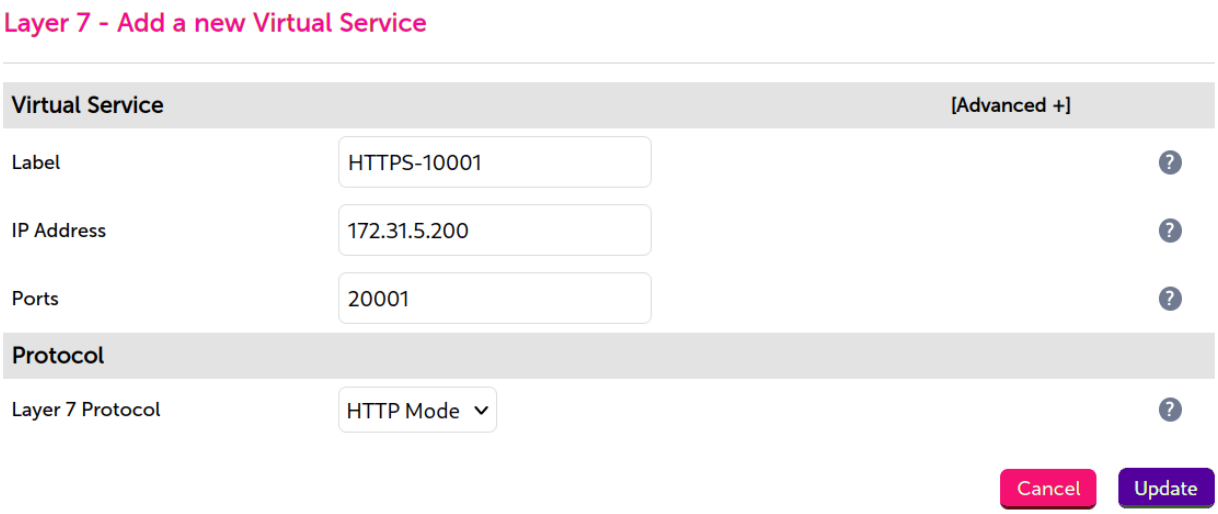
     > **Note**
     >
     > In line with Metaswitch guidance, it is strongly advised that internet facing services are HTTPS based ***only***. Plaintext HTTP services should ***not*** be used.

   - For HTTPS traffic, choose a new and an unused port to use. For example, if the existing service is listening on port 10001 then perhaps use port 20001. The original port cannot be reused here as it is needed later for the TLS/SSL decryption service, which will be the front end for the chain of services.
5. Set the *Layer 7 Protocol* to **HTTP Mode**.
6. Click **Update** to create the virtual service.

Example of an HTTP service:



Example of an HTTPS service:



7. Click **Modify** next to the newly created VIP.
8. In the *Protocol* section click **Advanced** to expand the menu.
9. Enable option **Accept Invalid HTTP Requests** by checking the checkbox.
10. In the *Persistence* section click **Advanced** to expand the menu.
11. Set the *Persistence Mode* to **X-Forwarded-For and Source IP** and set the *Persistence Timeout* to **2100**.
12. If the service on the EAS server in question processes HTTPS traffic then click **Modify** next to the newly created VIP and ensure that the *Enable Backend Encryption* checkbox is checked. This must be done because the EAS server is expecting to receive encrypted traffic.
13. In the *Other* section click **Advanced** to expand the menu.
14. Check the **Timeout** checkbox.
15. Set *Client Timeout* to **900000** (the units are milliseconds; this value equates to 15 minutes).
16. Set *Real Server Timeout* to **901000** (the units are milliseconds; this value equates to 15 minutes 1 second).
17. Click **Update**.

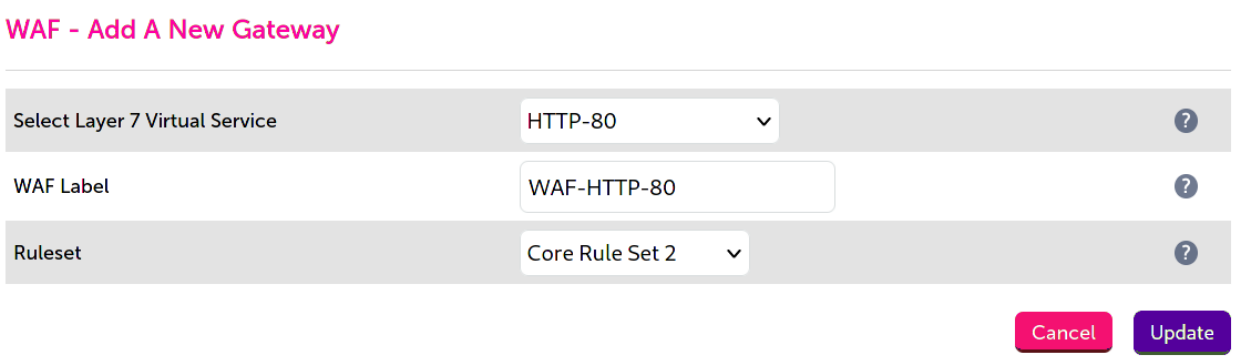
Defining the Real Server (RIP)

1. Using the web user interface, navigate to *Cluster Configuration > Layer 7 – Real Servers* and click on **Add a new Real Server** next to the newly created VIP.
2. Define the *Label* for the real server as required, e.g. **EAS-Server**.
3. Set the *Real Server IP Address* field to the IP address of the EAS server, e.g. **172.60.5.101**.
4. Set the *Real Server Port* field to the required port, e.g. **80**.
5. Click **Update**.

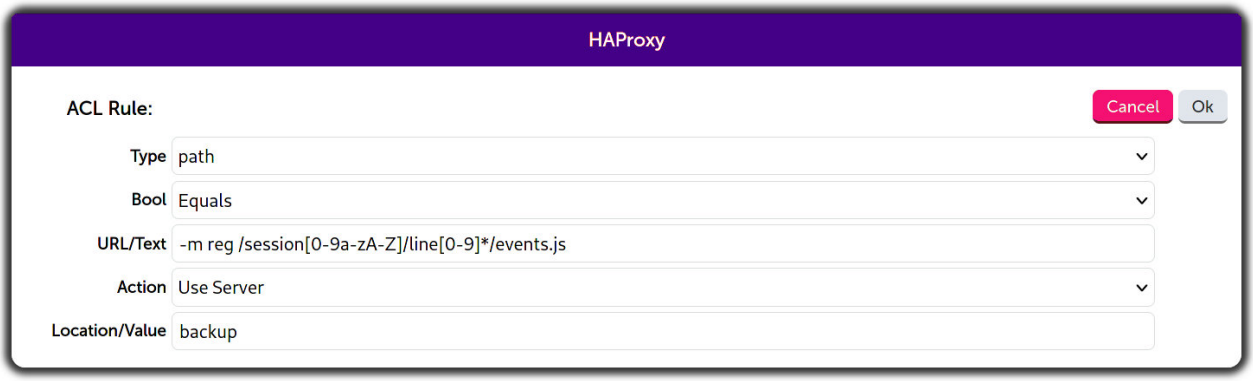
11.3.3. Creating the WAF Gateway

1. Using the web user interface, navigate to *Cluster Configuration > WAF – Gateway* and click on **Add a new WAF gateway**.

2. From the *Select Layer 7 Virtual Service* drop-down list select the associated layer 7 service that was just created, which in this example is **HTTP-80**.
3. Set *Ruleset* to **Core Rule Set 2**.
4. Click **Update** to create the WAF gateway.



5. Click **Modify** next to the newly created WAF service.
6. Set *Inbound Anomaly Score* to **12**.
7. Set *WAF Proxy Timeout* to **900** (the units are seconds).
8. Click **Update**.
9. Navigate to *Cluster Configuration > Layer 7 – Virtual Services*.
10. Click **Modify** next to the automatically created WAF VIP, e.g. WAF-HTTP-80.
11. In the *Protocol* section click **Advanced** to expand the menu.
12. Enable option **Accept Invalid HTTP Requests** by checking the checkbox.
13. Under *ACL Rules* click **Add Rule**.
14. Set *Type* to **path**.
15. Set *Bool* to **Equals**.
16. Set *URL/Text* to **-m reg /session[0-9a-zA-Z]+/line[0-9]*/events.js**
17. Set *Action* to **Use Server**.
18. Set *Location/Value* to **backup**.
19. Click **Ok**.



20. In the *Other* section click **Advanced** to expand the menu.
21. Check the **Timeout** checkbox.
22. Set *Client Timeout* to **900000** (the units are milliseconds; this value equates to 15 minutes).
23. Set *Real Server Timeout* to **901000** (the units are milliseconds; this value equates to 15 minutes 1 second).
24. Click **Update**.
25. Navigate to *Cluster Configuration > WAF – Manual Configuration*.
26. Using the drop-down list, select the WAF gateway that was just created, e.g. **WAF-HTTP-80**. The existing default WAF configuration will appear in the text box.
27. Paste your custom Metaswitch EAS WAF configuration into the text box.
28. Click **Update** to save the manual WAF configuration.

11.3.4. Setting Up the TLS/SSL Termination

If the service on the EAS server that is being protected with a WAF gateway handles **plaintext HTTP** traffic (not encrypted) then this step can be ignored, and the next applicable step will be Creating the Remaining Virtual Services.

If the service on the EAS server that is being protected with a WAF gateway handles **HTTPS** traffic (encrypted) then a TLS/SSL termination service must be created. This is required so that the traffic can be decrypted on the load balancer and fed into the WAF gateway service as plaintext HTTP traffic for scanning.

Uploading the Certificate

The appropriate certificate for the service in question must be uploaded to the load balancer for TLS/SSL termination to work. The process for doing this is as follows:

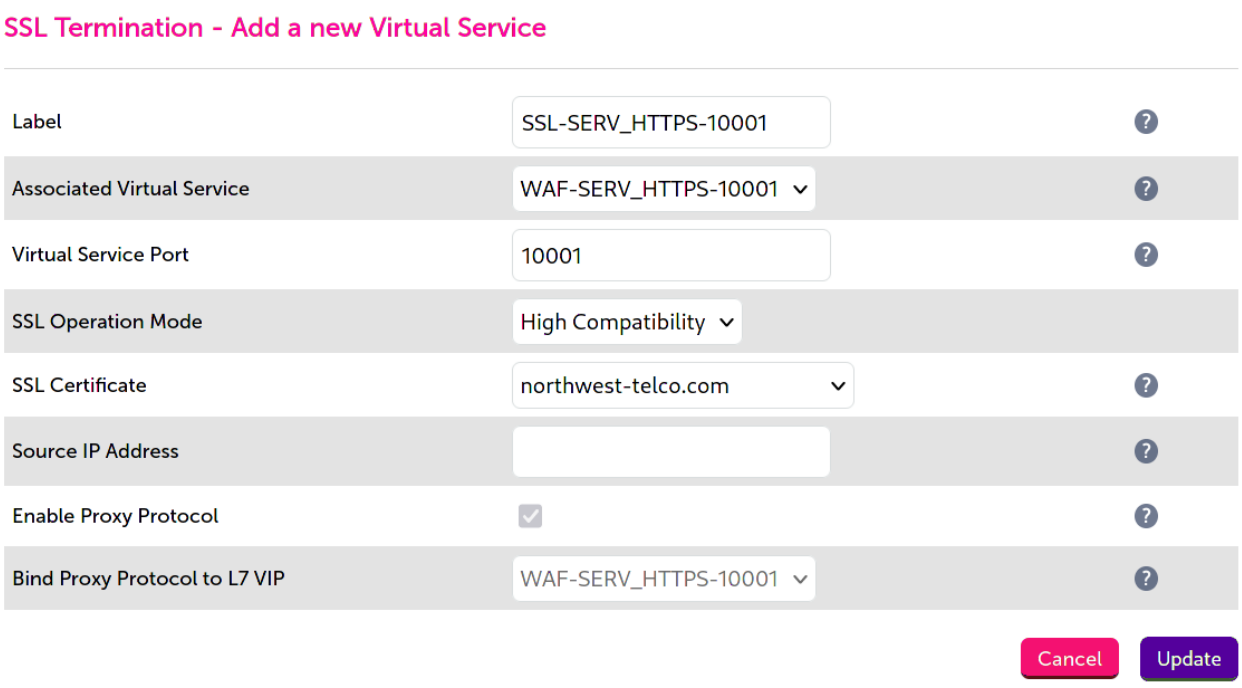
1. Using the web user interface, navigate to *Cluster Configuration > SSL Certificate* and click on **Add a new SSL Certificate**.
2. Press the *Upload prepared PEM/PFX file* radio button.
3. Define the *Label* for the certificate as required. It may make sense to use the domain that the certificate is associated to, e.g. **northwest-telco.com**.
4. Click on **Browse** and select the appropriate PEM or PFX style certificate.

5. If uploading a PFX certificate, enter the certificate's password in the *PFX File Password* field.
6. Click **Upload certificate**.

For more information on creating PEM certificate files and converting between certificate formats please refer to Creating a PEM File.

Creating the TLS/SSL Termination

1. Using the web user interface, navigate to *Cluster Configuration > SSL Termination* and click on **Add a new Virtual Service**.
2. From the *Associated Virtual Service* drop-down list, select the associated WAF gateway that was created previously, e.g. **WAF-HTTPS-10001**.
3. Set the *Virtual Service Port* field to the port that the existing service that is being replicated is listening on, e.g. **10001**.
4. From the *SSL Operation Mode* drop-down list, select **High Compatibility**.
5. From the *SSL Certificate* drop-down list, select the certificate for the service in question, which in this example is **northwest-telco.com**.
6. Click **Update** to create the TLS/SSL termination service.



11.3.5. Creating the Remaining Virtual Services

Repeat the steps listed above for every additional service provided by the EAS server that needs to be protected with a WAF gateway. The instructions in this section begin here.

11.4. Setting the PCRE Match Limits

11.4.1. Setting the Recommended Values

1. Using the WebUI, navigate to: *Cluster Configuration > WAF – Advanced Configuration*.
2. Set *PCRE Match Limit* to **500000**.
3. Set *PCRE Match Limit Recursion* to **500000**.
4. Click the **PCRE Match Limit Recursion** button.

11.4.2. Explanation

PCRE is an acronym for Perl Compatible Regular Expressions, which is a pattern matching library. It implements a regular expression engine, which the WAF uses to inspect HTTP traffic for malicious looking behaviour.

There are two PCRE match limits in the WAF implementation. These limits define the maximum number of calls that can be made to the underlying match function when the WAF evaluates a regular expression. Having such limits prevents PCRE from consuming huge amounts of system resources. It also protects against specially crafted regular expression attacks designed to overpower a WAF by forcing it to continuously perform a substantial number of pattern matches.

PCRE match limit values of 500000 are proven and sensible values to use in production.

11.5. Finalizing the Configuration

Once virtual services, WAF gateways, and TLS/SSL terminations have been created as required for all of the services that need WAF protection, the new configuration needs to be finalized and put into use by reloading the appropriate services.

To apply the new settings, HAProxy, the WAF and STunnel must be reloaded. This can be done using the buttons in the "Commit changes" box at the top of the screen or by using the *Restart Services* menu option:

1. Using the WebUI, navigate to: *Maintenance > Restart Services*.
2. Click **Reload HAProxy**.
3. Click **Reload WAF**.
4. Click **Reload STunnel**.

11.6. Putting the New WAF Services Into Production (Detection Only Mode)

Once they have been fully configured, the new WAF protected services can be put into production use.

By default, the newly created WAF services will operate in detection only mode. This means that the WAF services will scan traffic and perform logging, but will ***not*** perform any blocking actions. This reduces the risk of disrupting traffic to zero for practical purposes.

The WAF solution can be left in detection only mode for a length of time prior to fully enabling the WAF rule engine. Once fully enabled, the WAF services will perform blocking actions to actively block traffic that appears to be malicious. Operating for a period in detection only mode allows for observing how the WAF ***would*** function in full production ***before*** fully enabling it and allowing it to block traffic.

11.6.1. Directing Traffic to the WAF Services

It is likely the case that traffic from external users passes through an external-facing firewall before being sent on toward the EAS server. The firewall would have NAT rules configured to pass the external production traffic on toward the EAS server. These NAT rules would need to be modified on the firewall to send the external traffic to the new WAF protected services.

The exact procedure for changing firewall NAT rules varies between firewall vendors and is outside the scope of this document.

It is possible that instead of making firewall changes, it may be necessary to change the DNS record for a service's FQDN to point to the IP address of the new WAF services.

For deployments that do not use firewall NAT rules or DNS records to direct production traffic, or for other related inquiries, please contact support@loadbalancer.org for further advice.

11.7. Fully Enabling the WAF Rule Engine

The WAF rule engine needs to be enabled for a given WAF service to allow it to actively block traffic that appears to be malicious.

To enable the WAF rule engine for a given WAF service:

1. Using the WebUI, navigate to: *Cluster Configuration > WAF – Gateway*.
2. Click on **Modify** next to the WAF service in question.
3. Check the **Rule Engine Traffic Blocking** checkbox.

4. Click **Update**.
5. Click **Reload HAProxy** and **Reload WAF** when prompted, to put the change into effect.

12. Custom WAF Rule Set Description

The top of the custom WAF rule set contains a series of parameters, along with descriptions of what they do, their default values, and any restrictions on value ranges. The values of these parameters are user definable, although *in practice the default values are almost always acceptable*. The rule set parameters are described throughout the remainder of this section.

12.1. Toggling Source IP Address Based Blocking

Some users may have **unique** source IP addresses. Some users may **share** a single source IP address, for example a satellite office where many users sit behind a single public IP address (with NAT taking place at a firewall or router).

Some of the protective rules in the rule set block by IP address if an address looks like an attacker. If one thousand users in an office are sharing one IP address then all of their combined login attempts and requests, coming from the same source, may look like an attack: this could cause the IP address to be blocked, blocking

out the entire office.

If all users have unique source IP addresses, *or* exemption rules have been added to cover those IP addresses/subnets where NAT is taking place, then leave the "unique source addresses" toggle set to its default value:

```
unique_source_addresses=1
```

If a significant proportion of users share IP addresses (so that writing exemption rules to cover all the shared addresses/subnets is not worthwhile or is practically impossible) then the "unique source addresses" toggle should be set to zero to avoid the false positive blocking scenario mentioned above:

```
unique_source_addresses=0
```

12.2. CommPortal Specific Rules

Five custom WAF rules have been developed to protect the CommPortal login page of a Metaswitch EAS deployment. These rules are presented and described individually in the sections below.

12.2.1. Rule Exclusions

To enable a Metaswitch EAS deployment to work correctly with WAF gateways, it is necessary to exclude some rules from the WAF functionality's standard OWASP ModSecurity Core Rule Set (further details about this can be found in WAF OWASP Protection). The excluded rules prevent genuine, safe Metaswitch EAS traffic from being blocked.

The set of excluded rules can be found near the middle of each Metaswitch specific rule set.

12.2.2. Rule 1: DOS Protection for login.html

This rule is not recommended for use in deployments where a significant proportion of users share the same IP address, such as where NAT is taking place at a firewall or router.

This rule provides DOS protection for the CommPortal login page. It counts all requests for the page `login.html`. Regardless of the request method used (e.g. GET or POST), any access to this URL will increment a counter against the IP address making the request.

If the rule 1 counter for a given IP address exceeds the threshold defined for rule 1 (200 by default, i.e. 200 HTTP requests to `login.html` are allowed) then the IP address is blocked (for 60 seconds by default).

If an IP address makes *no* requests to `login.html` for a period of time (5 seconds by default) then the rule 1 counter for that address is reset to zero.

Parameters

The rule set contains the following parameters for rule 1:

- Number of requests an IP address can make to `/login.html` before being blocked. Default: **200**.

- Time, in seconds, that an IP address is blocked for if it is blocked by rule 1. Default: **60** (max: 600).
- Period of inactivity (no requests made to `login.html`), in seconds, after which an IP address' rule 1 request counter is removed ('reset to zero'). Default: **5**.

12.2.3. Rule 2: Detect Failed Login and Block

This rule is not recommended for use in deployments where a significant proportion of users share the same IP address, such as where NAT is taking place at a firewall or router.

This is an aggressive rule which detects and counts confirmed failed CommPortal login attempts. ***Response*** headers sent by the EAS back end to the client are inspected. If the response headers signal a failed login attempt then a counter is incremented against the IP address that made the failed login attempt.

If the rule 2 counter for a given IP address exceeds the threshold defined for rule 2 (20 by default, i.e. 20 failed login attempts are allowed) then the IP address is blocked (for 10 minutes by default).

The rule 2 counter 'cools off' over time to avoid blocking genuine clients (by default, the counter is decremented by 1 every 30 seconds).

Parameters

The rule set contains the following parameters for rule 2:

- Number of failed CommPortal logins an IP address can make before being blocked. Default: **20**
- Time, in seconds, that an IP address is blocked for if it is blocked by rule 2. Default: **600** (max: 600)
- Time, in seconds, between each decrement action applied to the failed login counter to 'cool it off' over time. Default: **30**
- Value the failed login counter should be decremented by on each decrement action. Default: **1**

12.2.4. Rule 3: DOS Protection for POST Requests to /login

This rule is not recommended for use in deployments where a significant proportion of users share the same IP address, such as where NAT is taking place at a firewall or router.

This rule provides DOS protection, specifically blocking floods of POST requests to the location `/login`. When a client submits a POST request to the `/login` location, which typically indicates a CommPortal login attempt, a counter is incremented against the client's IP address.

If the rule 3 counter for a given IP address exceeds the threshold defined for rule 3 (500 by default, i.e. 500 POST requests to `/login` are allowed) then the IP address is blocked (for 10 minutes by default).

The rule 3 counter 'cools off' over time to avoid blocking genuine clients (by default, the counter is decremented by 1 every 5 seconds).

Parameters

The rule set contains the following parameters for rule 3:

- Number of POST requests an IP address can make to `/login` before being blocked. Default: **500**
- Time, in seconds, that an IP address is blocked for if it is blocked by rule 3. Default: **600**, (max: 600)
- Time, in seconds, between each decrement action applied to the POST request counter to 'cool it off' over time. Default: **5**
- Value the POST request counter should be decremented by on each decrement action. Default: **1**

12.2.5. Rule 4: Detect username abuse and block

> **Note** "Username" is synonymous with "directory number" in Metaswitch parlance.

> **Note** Some CommPortal clients use a "persistent authentication token" (PAT) as a login method. This means that a directory number and password are **not** submitted and therefore this information is not visible to the WAF service. It cannot be guaranteed that a directory number and password will be present with all log in requests.

This rule detects username (directory number) abuse. When a CommPortal login attempt contains a directory number (e.g. a browser based login) **and** the client IP address trying to log in has more than two confirmed failed login attempts recorded against it (as per the rule 2 counter – to avoid false positives) then a counter is incremented against the directory number used in the login attempt.

If the rule 4 counter for a given directory number exceeds the threshold defined for rule 4 (20 by default, i.e. ***at least*** 20 failed login attempts using the same directory number are allowed) then all logins using the **directory number** are blocked (for 10 minutes by default).

The rule 4 counter 'cools off' over time to avoid blocking genuine clients (by default, the counter is decremented by 1 every 30 seconds).

> **Note** The requirement that a client IP address must have more than two confirmed failed login attempts recorded against it (as per the rule 2 counter) is removed if source IP address based blocking has been disabled. (This is because rule 2 is disabled in this mode, and so its counter will never increment.)

Parameters

The rule set contains the following parameters for rule 4:

- How many times the same username can be submitted to the CommPortal login page before being flagged as 'under brute force attack' and further login attempts with that username are blocked. Default: **20**
- Time, in seconds, that a username is blocked for if it is blocked by rule 4. Default: **600** (max: 600)
- Time, in seconds, between each decrement action applied to the username login counter to 'cool it off' over time. Default: **30**
- Value the username login counter should be decremented by on each decrement action. Default: **1**

12.2.6. Rule 5: Detect password abuse and block

> **Note**
>
> Some CommPortal clients use a "persistent authentication token" (PAT) as a login method. This means that a directory number and password are **not** submitted and therefore this information is not visible to the WAF service. It cannot be guaranteed that a directory number and password will be present with all log in requests.

This rule detects password abuse. When a CommPortal login attempt contains a password (e.g. a browser based login) **and** the client IP address trying to log in has more than two confirmed failed login attempts recorded against it (as per the rule 2 counter– to avoid false positives) then a counter is incremented against the password used in the login attempt.

If the rule 5 counter for a given password exceeds the threshold defined for rule 5 (20 by default, i.e. ***at least*** 20 failed login attempts using the same password are allowed) then all logins using the **password** are blocked (for 10 minutes by default).

The rule 5 counter 'cools off' over time to avoid blocking genuine clients (by default, the counter is decremented by 1 every 30 seconds).

> **Note**
>
> The requirement that a client IP address must have more than two confirmed failed login attempts recorded against it (as per the rule 2 counter) is removed if source IP address based blocking has been disabled. (This is because rule 2 is disabled in this mode, and so its counter will never increment.)

Parameters

The rule set contains the following parameters for rule 5:

- How many times the same password can be submitted to the CommPortal login page before being flagged as 'under brute force attack' (e.g. an attacker trying the same common password against many different usernames) and further login attempts with that password are blocked. Default: **20**
- Time, in seconds, that a password is blocked for if it is blocked by rule 5. Default: **600** (max: 600)
- Time, in seconds, between each decrement action applied to the password login counter to 'cool it off' over time. Default: **30**
- Value the password login counter should be decremented by on each decrement action. Default: **1**

12.3. SIP Provisioning Protection

12.3.1. Overview

An endpoint (phone) attempting to retrieve its full configuration from an EAS SIP provisioning service can be a noisy affair. A given endpoint may attempt to download over a dozen different files in the process, including dictionaries, licences, and wallpapers. Many of the requested files may not exist or may relate to functionality not in use by the EAS deployment in question.

Thanks to a list provided by Metaswitch, the WAF solution is able to identify the relevant (SIP) configuration file from each provisioning request. This configuration file can contain sensitive information: it is of value to bad actors and needs protecting. The name of the configuration file can vary by endpoint vendor. Examples include:

- `589cfc866f5f.cfg` (i.e. "MAC_address.cfg")
- `config.cfg`
- `aastra.cfg`
- `KX-TGP550T04.cfg`

> **Important** The SIP provisioning protection rules protect access to SIP provisioning configuration files. Requests to ***other services*** and requests that ***don't match a valid, known SIP configuration file pattern*** are **not affected** by the SIP provisioning protection rules.

The WAF solution maintains a count of how many configuration file retrieval requests a given IP address has made. Two possible scenarios are as follows:

- Power cycling a 'hardphone', e.g. a desk phone, would cause that endpoint to retrieve its full configuration from the provisioning service. The WAF solution would correctly count this entire event as a single configuration retrieval request, even though the endpoint may request a dozen or more individual files. This is because only one of the requested files would be matched as a configuration file, which is what is counted.
- A bad actor could use a script to carry out a brute-force attack against the provisioning service. These attacks commonly cycle through a range of plausible MAC addresses as configuration filenames. The attacker would hope to eventually stumble upon a valid filename by chance and be able to download the configuration file. For example, they may attempt to download

```
589cfc000000.cfg
589cfc000001.cfg
589cfc000002.cfg
...
```

  in sequence. The WAF solution would count every one of these 'guessed' configuration retrieval requests. Without the protection of a WAF, one real world attack alone can easily result in over a million such 'guesses' per day on a production provisioning service.

12.3.2. Identifying and Blocking an Attack

Once per minute, the WAF service checks to see how many configuration retrieval requests a given IP address has made during the past minute. If that number is ***greater than*** the set threshold then the IP address in question is **flagged**.

> **Note** A **default threshold** of 20 is set. This means that an IP address can make a maximum of 20 configuration retrieval requests per minute without arousing suspicion.

If an IP address is flagged three times then it is **blocked** from accessing the provisioning service for one hour (the status code ***429 Too Many Requests*** is returned to the client.)

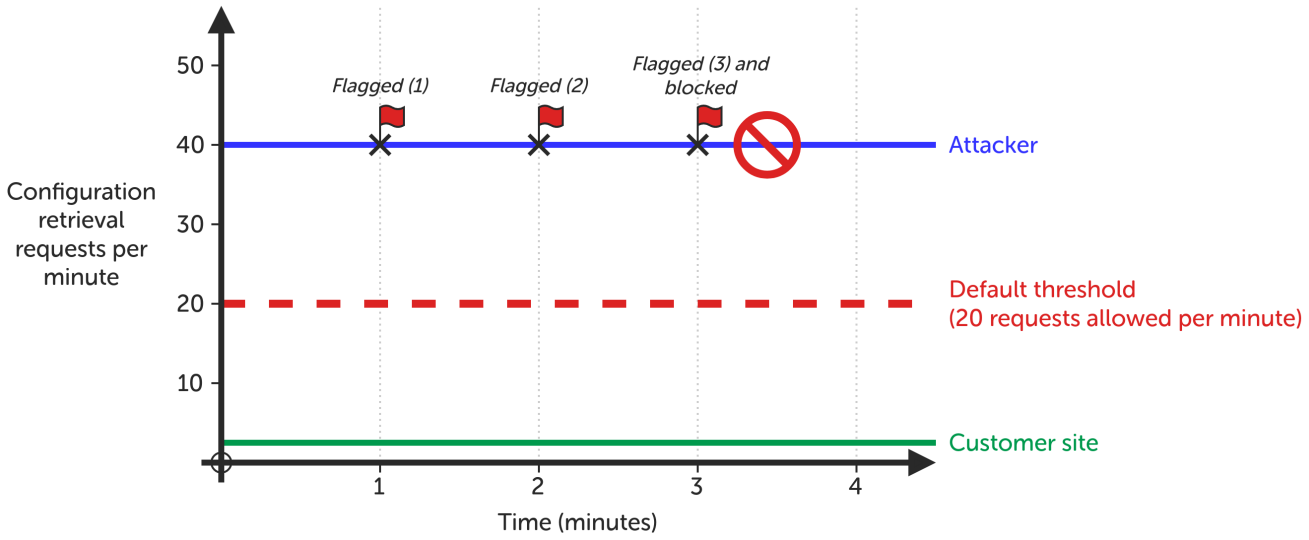
If an IP address is ***flagged but short of being blocked*** then all of its flags will be cleared after two minutes, assuming that the IP address doesn't accumulate a third flag and become blocked during that time. This ensures

that flags are not permanent, reducing the number of false positives from genuine, "bursty" clients.

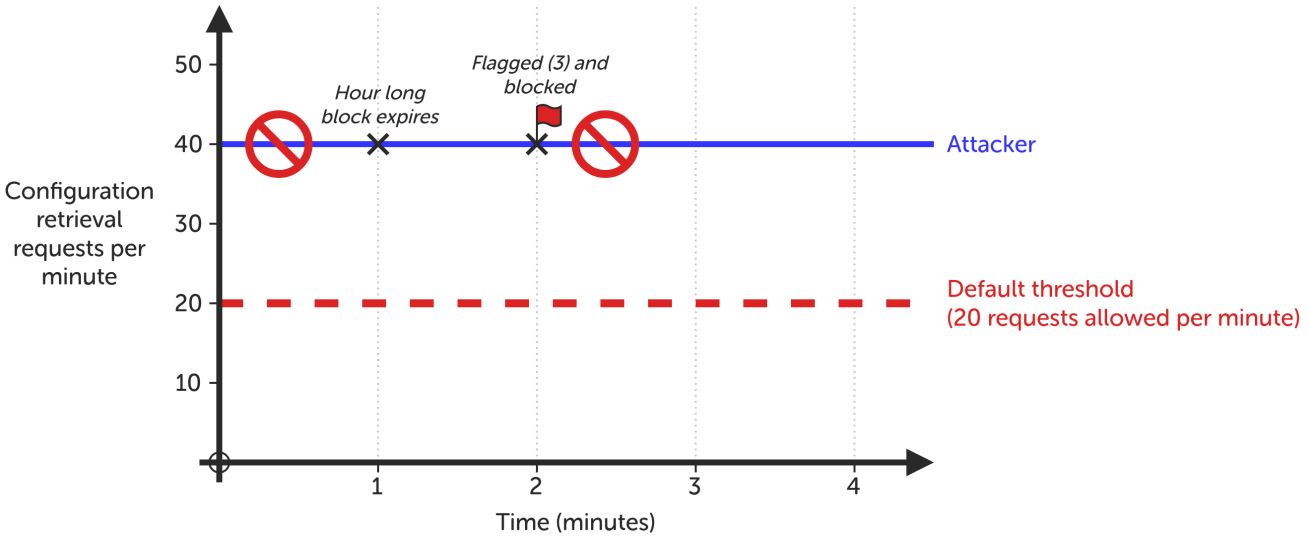
Example 1: A Scripted Attack is Blocked

In this example, an attacker runs a script which makes a consistent number of configuration retrieval requests per minute. (Note that, in reality, the 'requests per minute' figure for the attacker would be more than an order of magnitude greater than shown here.) The attacker's IP address is flagged three times in a row and blocked for the next hour. Any further provisioning requests from the attacker's address will receive a 429 Too Many Requests response from the WAF gateway and will ***not*** be passed through to the EAS back end.

An example customer site is also illustrated on the graph. Perhaps with a few dozen endpoints located behind a single IP address (with NAT taking place), the customer site's address makes only a handful of requests per minute, staying well below the threshold. A real customer site may even go for long periods of time without making any requests at all.



An IP address that is blocked retains all of its flags ***minus one***. When a block expires, the previously blocked address is on the cusp of being blocked again (as the address only requires one more flag to trigger a block). This means that attackers are quickly re-blocked if their behaviour continues after their hour long block expires. Re-blocking takes only 60 seconds.



Example 2: Power Outage at a Customer Site

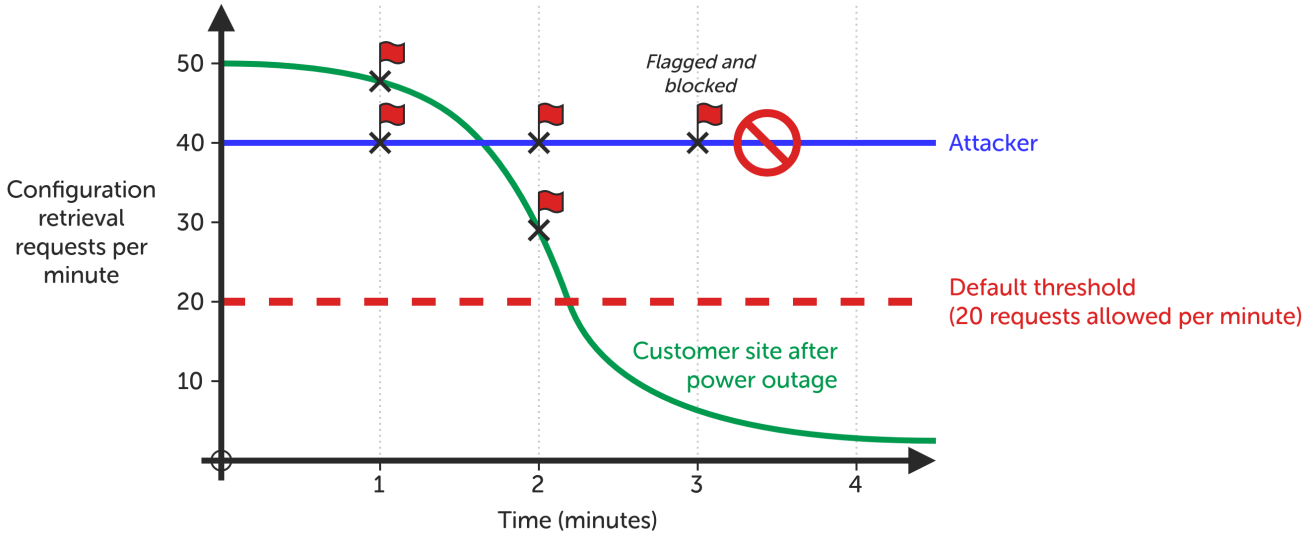
In this example, the IP address of a scripted attack is flagged three times in a row and blocked for the next hour. This is the same as in the first example.

The difference in this scenario is with the example customer site, illustrated on the graph, which has suffered a power outage. The site's power is restored at time 0, causing all endpoints at the site boot up and retrieve their full configuration from the provisioning service.

Assuming that all endpoints at the customer site are located behind a single IP address (with NAT taking place), the site's address suddenly looks very busy. This sudden flurry of activity pushes the site's IP address above the allowed request threshold and causes the address to be flagged. At this point, the burst of activity looks like a potential attack.

By the two minute mark, the customer site's address is less busy as many phones have finished booting and retrieving their configuration files. Slower phones, however, are still pulling down provisioning information. This causes the site's IP address to remain above the allowed threshold and to be flagged for a second time.

By the end of the third minute, almost all endpoints at the site have booted up and made the necessary provisioning requests. Traffic levels from the site approach normal levels, below the threshold for being flagged. Activity from the site no longer looks suspicious, the site is not flagged again, and the site's IP address is not blocked. The site's activity, while both sudden and busy, is not flagged as an attack thanks to examining the traffic pattern over time.



12.3.3. Parameters

The rule set contains the following parameters for SIP provisioning protection:

- Provisioning service protection master switch: 0 = off, 1 = on. Default: **0**
- 'Detection only' option: limits the provisioning service protection rules to merely logging when a block should be applied or enforced without actually enforcing any blocks. 0 = off, 1 = on. Default: **0**
- Burst (1 minute) request limiting status: 0 = off, 1 = on. Default: **1** ***(should never require changing)***
- Number of configuration retrieval requests an IP address can make per minute before it is flagged (note

"flagged", not immediately blocked). Default: **20**

- Number of times an IP address has to be flagged to be blocked. Default: **3**
- Time, in seconds, that an IP address is blocked for if it exceeds the 1 minute request threshold. Default: **3600** (max: 3600)
- Time, in seconds, between each decrement action applied to the 1 minute request counter to 'cool it off' over time. Default: **60**
- Value the 1 minute request counter should be decremented by on each decrement action: this needs to be large enough to zero out the counter. Default: **10000** ***(should never require changing)***

12.3.4. Preventing Customers being Blocked

The SIP provisioning rules are written to try and avoid blocking genuine users erroneously. It is still possible, however, that a customer could repeatedly breach the "allowed configuration retrieval requests per minute" threshold and cause their IP address to be blocked. For example, a misconfigured or faulty endpoint could continually try to retrieve its configuration file, perhaps dozens or hundreds of times per minute. Such behaviour would cause the customer site's IP address to be blocked.

The risk of accidentally blocking a real customer can be completely mitigated: custom rules can be written to explicitly allow all traffic from a customer site's IP address or subnet. This is recommended for large sites, for example where hundreds of endpoints are located behind a single, or handful of, IP addresses. This requires that a customer site's IP address be static and known in advance, which is often the case.

For more information and instructions on writing explicit "allow" rules, refer to the section Blocking and Allowing by IP Address.

12.3.5. Custom PPS Location

If the old style PPS provisioning service is in use and requires WAF protection, if the PPS service uses a custom path (i.e. not /pps) then the WAF rule set must be customised to account for this. All rules containing instances of *pps* should be replaced with the custom path in question. Contact support@loadbalancer.org if further advice is required.

13. Logging/Alerting

13.1. Logging Overview

Outside of a controlled test environment, WAF logging is typically very noisy. This is especially true for services that are accessible from the public internet due to the chaotic nature of public internet traffic. As discussed later, in the section Tag Data, Metaswitch specific tags are applied to assist in finding relevant alerts in log files.

13.1.1. WAF Audit Mode

Note that the "audit mode" of the WAF functionality should **never** be enabled on a production system. This option is of little use in the context of a Metaswitch EAS WAF deployment and is likely to fill the entire logging partition of the load balancer in a short period of time.

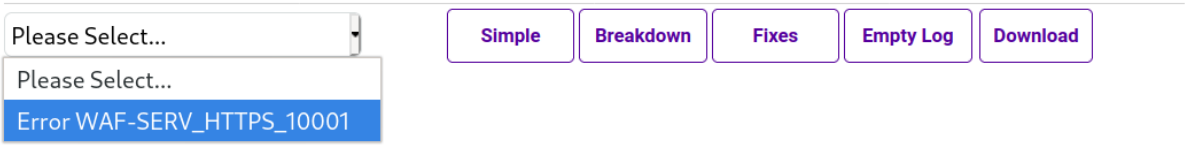
13.1.2. Layer 7 Logging

Note that layer 7 logging should be left set to **Off**, the default value. While enabling layer 7 logging can be very useful during a troubleshooting exercise, these log files can become very large and cause problems on a production system.

13.2. Viewing WAF Logs/Alerts Through the WebUI

Logs/alerts for a given WAF gateway can be viewed through the WebUI as follows:

1. Navigate to *Logs > WAF Error*.
2. From the drop-down menu, select the WAF gateway in question.



3. Click the **Simple** button to display the log data in a more human readable form.

Error WAF-SERV_HTTPS_10001 | Simple | Breakdown | Fixes | Empty Log | Download

---

```
[Wed May 19 12:44:52.295492 2021]
        [:error]
        [pid 8029:tid 140657637164800]
        [client 192.168.85.1:16376]
        [client 192.168.85.1]
      • ModSecurity: Warning. Operator GT matched 20 at IP:rule_2_counter.
        [file "/opt/httpd-waf/modsecurity.d/usr_WAF-SERV_HTTPS_10001_rules.conf"]
        [line "1092"]
      • [id "5002020"]
        [msg "New block applied: IP address blocked from CommPortal login for 600 seconds due to
exceeding the allowed number of 20 failed CommPortal login attempts."]
        [data "Directory Number: "]
        [tag "METASWITCH/COMMPORTAL_LOGIN/NEW_BLOCK"]
        [hostname "this.is.a.test"]
        [uri "/login"]
        [unique_id "YKUIRH8AAAEAAB9dac4AAABM"]
```

```
[Wed May 19 12:44:52.962147 2021]
        [:error]
        [pid 8029:tid 140657620379392]
        [client 192.168.85.1:16382]
        [client 192.168.85.1]
      • ModSecurity: Warning. Operator EQ matched 0 at IP.
        [file "/opt/httpd-waf/modsecurity.d/usr_WAF-SERV_HTTPS_10001_rules.conf"]
        [line "933"]
      • [id "5001025"]
        [msg "Block enforced: CommPortal login access denied to temporarily blocked IP address (600
seconds remain on rule 2 block - exceeded the allowed number of 20 failed CommPortal login attempts).
(1 blocked requests from this IP address since last alert)"]
        [data "Directory Number: "]
        [tag "METASWITCH/COMMPORTAL_LOGIN/ENFORCED_BLOCK"]
        [hostname "this.is.a.test"]
        [uri "/login"]
        [unique_id "YKUIRH8AAAEAAB9dac8AAABO"]
```

13.3. Breakdown of a Single Alert

`[Wed May 19 12:44:52.295492 2021]` ← Date and time of the HTTP request in question

`[:error]`

`[pid 8029:tid 140657637164800]`

`[client 192.168.85.1:16376]`

`[client 192.168.85.1]` ← IP address of the client

- `ModSecurity: Warning. Operator GT matched 20 at IP:rule_2_counter.`

`[file "/opt/httpd-waf/modsecurity.d/usr_WAF-SERV_HTTPS_10001_rules.conf"]`

`[line "1092"]`

- `[id "5002020"]`

`[msg "New block applied: IP address blocked from CommPortal login for 600 seconds due to exceeding the allowed number of 20 failed CommPortal login attempts."]` ← Verbose message explaining the alert

`[data "Directory Number: 1234567890"]` ← Directory number in question, if available

`[tag "METASWITCH/COMMPORTAL_LOGIN/NEW_BLOCK"]` ← Tag categorising the alert

`[hostname "this.is.a.test"]` ← Hostname of the request, e.g. portal.telco.com

`[uri "/login"]` ← URI of the request

`[unique_id "YKUIRH8AAAEAAB9dac4AAABM"]` ← Unique ID linking all alerts for this HTTP request

13.4. Tag Data

Four Metaswitch EAS WAF specific alert tags are in use. These can be very helpful as they allow alerts to be filtered. The tags are as follows:

- `METASWITCH/COMMPORTAL_LOGIN/NEW_BLOCK`
- `METASWITCH/COMMPORTAL_LOGIN/ENFORCED_BLOCK`
- `METASWITCH/PROVISIONING_SERVICES/NEW_BLOCK`
- `METASWITCH/PROVISIONING_SERVICES/ENFORCED_BLOCK`

To find all alerts related to Metaswitch EAS, search for `METASWITCH`.

To find all alerts specifically related to CommPortal, search for `COMMPORTAL_LOGIN`.

To find all alerts specifically related to provisioning service protection, search for `PROVISIONING_SERVICES`.

To find all alerts where a ***new*** block was put in place against a client, search for `NEW_BLOCK`.

13.5. Summary Reporting

If a block is in place against a client and the client continues to make requests then a summary alert for that client will be written to the log file every 60 seconds. This drastically reduces log file sizes as each individual request that is denied and blocked is not written to disk as an individual alert.

An example of such a 'summary' alert is presented below.

```
[Wed May 19 14:20:55.083280 2021]
        [:error]
        [pid 8030:tid 140657570023168]
        [client 192.168.85.1:21384]
        [client 192.168.85.1]
      • ModSecurity: Warning. Operator EQ matched 0 at IP.
        [file "/opt/httpd-waf/modsecurity.d/usr_WAF-SERV_HTTPS_10001_rules.conf"]
        [line "933"]
      • [id "5001025"]
        [msg "Block enforced: CommPortal login access denied to temporarily blocked IP address (360
seconds remain on rule 2 block - exceeded the allowed number of 20 failed CommPortal login attempts).
(223 blocked requests from this IP address since last alert)"]
        [data "Directory Number: 24682468"]
        [tag "METASWITCH/COMMPORTAL_LOGIN/ENFORCED_BLOCK"]
        [hostname "this.is.a.test"]
        [uri "/login"]
        [unique_id "YKUex38AAAEAAB9eQfsAAACU"]
```

Note the message (`msg`) line, which states that 223 individual requests have been blocked from the IP address in question since the previous alert. Also note the `tag` line, which makes use of the `ENFORCED_BLOCK` tag. This identifies the alert as pertaining to the enforcement of an ***existing*** block, as opposed to a ***new*** block being put in place (which is typically more interesting).

13.6. Further Log File Analysis

Logging data, including historic data ("rotated" log files from previous days), can also be viewed, filtered, and analysed using your tool of choice, e.g. vim. To do so, create and download a technical support archive from the load balancer. This creates a compressed archive file containing the logging data in question.

To do this from the WebUI:

1. Navigate to ***Support > Technical Support Download***.
2. Ensure that the option ***Do not include GZ files*** is **unchecked**.

3. Click **Generate Archive** and wait to be presented with a download link.

Once the archive file has been downloaded and uncompressed, the WAF log files can be found under `logs/httpd`.

14. Writing Site-Specific Custom Rules

It is possible to write site-specific custom rules for the WAF implementation. Such rules can provide additional security and address site-specific needs and problems by leveraging functionality of the highly flexible WAF engine, ModSecurity.

The WAF rule set is thoroughly commented, so as to be self contained, and describes how to write additional rules, as well as providing examples. Much of that content is mirrored here for completeness, as well as being more verbose and providing additional advice.

14.1. Blocking and Allowing by IP Address

Rules must have unique IDs. To avoid ID collisions, use the following range for this type of rule:

- Rule IDs: 5,003,000-5,003,999

14.1.1. Explicitly Allowing Traffic from Trusted Addresses

Known trusted IP addresses and subnets can be explicitly "allowed" to ensure they are not scanned by the WAF. This allows traffic from known good sources to bypass the WAF, which removes the possibility of encountering false positives and being blocked.

Here is an example of an "allow" rule, which explicitly disables the WAF rule engine for a known good IP address, for example the address of a help desk system:

```
SecRule REMOTE_ADDR "@IPMatch 10.0.12.32" "id:5003000,phase:1,pass,nolog,ctl:ruleEngine=Off"
```

Here is another example allow rule, which explicitly disables the WAF rule engine for a known good subnet, for example a corporate network:

```
SecRule REMOTE_ADDR "@IPMatch 10.0.0.0/8" "id:5003001,phase:1,pass,nolog,ctl:ruleEngine=Off"
```

A single rule can match against multiple IP addresses or subnets by using a comma separated list, for example: `192.168.85.1,192.168.64.4,192.168.28.0/24`.

14.1.2. Blocking Bad Addresses

Known bad IP addresses and subnets can be blocked outright. Connections originating from specified IP addresses or subnets will be served a 4xx HTTP status code of choice, for example ***403 Forbidden***. This is useful for blocking IP addresses and subnets that are known to be associated with attackers.

Here is an example blocking rule, which denies all connections from a specified IP address by returning a 403

Forbidden status code:

```
SecRule REMOTE_ADDR "@IPMatch 52.51.5.169" "id:5003002,phase:1,deny,status:403"
```

In the same way as when "allowing" trusted addresses, a single rule can match against multiple IP addresses or subnets by using a comma separated list.

14.2. Blocking by User-Agent Request Header

Rules must have unique IDs. To avoid ID collisions, use the following range for this type of rule:

- Rule IDs: 5,004,000-5,004,999

Requests can be blocked based on a phrase found in the User-Agent HTTP request header. For example, a real world scripted attack on an EAS deployment was thwarted after noticing that the word 'python' was present in the User-Agent header. The attack was instantly stopped by enforcing a block on all traffic with a User-Agent header containing the word 'python'.

Here is an example blocking rule, which performs a case insensitive regular expression match against the User-Agent request header, looking for a single phrase: the word 'python':

```
SecRule REQUEST_HEADERS:User-Agent "@contains python" "id:5004000,deny,log,\
t:lowercase,msg:'Keyword Python detected in the User-Agent request header.'"
```

The above example also writes the specified message ('msg') string to the WAF log file for added context and clarity when examining the logs at a later time. The message can be customised as needed.

14.2.1. Searching for Multiple Phrases

It is also possible to search the User-Agent header for multiple phrases simultaneously. A specialised operator is used for this, the parallel match ('pm') operator, which is very efficient when searching for multiple keywords.

Here is an example blocking rule, which performs a case insensitive parallel match for multiple keywords against the User-Agent request header:

```
SecRule REQUEST_HEADERS:User-Agent "@pm apple pear orange" \
    "id:5004001,deny,log,msg:'Suspicious User-Agent detected.'"
```

14.3. Blocking by Geographic Location

Rules must have unique IDs. To avoid ID collisions, use the following range for this type of rule:

- Rule IDs: 5,005,000-5,005,999

Connections can be blocked by geographic location, such as by country or continent. This is achieved by performing a lookup of the source IP address of each incoming connection against a database that maps public IP address ranges to countries. A compatible database of IP addresses can be downloaded from our website.

14.3.1. Database Download: One Time

To download a compatible database on to a load balancer ***once only***, execute the following command, either from an SSH session, the console, or through the WebUI under ***Local Configuration > Execute shell command***:

```
mkdir -p /usr/local/geo/data; curl https://downloads.loadbalancer.org/geo/GeoIP.dat -o
/usr/local/geo/data/GeoIP.dat
```

14.3.2. Database Download: Weekly `cron` Job

To download or update a compatible database on to a load balancer and set up a weekly `cron` job to automatically update it, execute the following command, either from an SSH session, the console, or through the WebUI under ***Local Configuration > Execute shell command***:

```
curl https://downloads.loadbalancer.org/geo/geolocationDBUpdate.sh -o
/etc/cron.weekly/geolocationDBUpdate.sh; chmod +x /etc/cron.weekly/geolocationDBUpdate.sh;
/etc/cron.weekly/geolocationDBUpdate.sh
```

14.3.3. Important Caveats

If using an HA pair of load balancers, the command must be run on **both** the primary appliance and the secondary appliance.

The load balancer appliance is "secure by default". This means that the ***Execute shell command*** option is disabled by default. For information on what this means, how to re-enable this functionality, and the ramifications of doing so, refer to the section ***Appliance Security - Security Mode*** in the Administration Manual.

14.3.4. Loading the Database

To start using the database for performing geographic lookups, the following directive must be added **once** to the WAF rule set:

```
SecGeoLookupDb /usr/local/geo/data/GeoIP.dat
```

14.3.5. Country and Continent Codes

The WAF uses ISO 3166 two-letter country and continent codes. For example, CA is used for Canada, DE for Germany, GB for the United Kingdom of Great Britain and Northern Ireland, and US for the United States of America.

The seven continent codes used are:

- AF: Africa
- AN: Antarctica
- AS: Asia
- EU: Europe

- NA: North America
- OC: Oceania
- SA: South America

14.3.6. Only Allow Traffic from One Country

If an EAS deployment will only be serving genuine traffic to users in a single country, it is possible to instruct the WAF to block **all** connections that don't appear to originate from the appropriate country.

Here is an example chain of rules that blocks all non-Canadian traffic, returning a 403 Forbidden status code:

```
SecRule REMOTE_ADDR "@geoLookup" "chain,id:5005000,phase:1,deny,status:403, \
    msg:'Non-CA IP address: address is in %{GEO.COUNTRY_NAME} (%{GEO.COUNTRY_CODE}).'"
SecRule GEO:COUNTRY_CODE "!@streq CA"
```

The above example also writes to the WAF log where it thinks a blocked connection originated from, giving both the country name and its two-letter code for reference.

14.3.7. Only Allow Traffic from a Range of Countries

Here is an example chain of rules that blocks all traffic that isn't either British, French, or American, returning a 403 Forbidden status code:

```
SecRule REMOTE_ADDR "@geoLookup" "chain,id:5005001,phase:1,deny,status:403, \
    msg:'Non-GB, FR, or US IP address: address is in %{GEO.COUNTRY_NAME} (%{GEO.COUNTRY_CODE}).'"
SecRule GEO:COUNTRY_CODE "!@rx (?:GB|FR|US)"
```

14.3.8. Explicitly Block Traffic from a Specific Country

It is possible to explicitly block traffic that appears to originate from a specific country. This can be useful if attacks regularly appear to originate from a location where it is very unlikely that a genuine user would ever be.

Here is an example chain of rules that explicitly blocks all traffic from Paraguay, returning a 403 Forbidden status code:

```
SecRule REMOTE_ADDR "@geoLookup" "chain,id:5005002,phase:1,deny,status:403, \
    msg:'Paraguayan IP address detected.'"
SecRule GEO:COUNTRY_CODE "@streq PY"
```

14.3.9. Explicitly Block Traffic from a Range of Countries

Here is an example chain of rules that explicitly blocks all traffic from Jordan, Cyprus, or Singapore, returning a 403 Forbidden status code:

```
SecRule REMOTE_ADDR "@geoLookup" "chain,id:5005003,phase:1,deny,status:403, \
    msg:'Jordanian, Cypriot, or Singaporean IP address detected (%{GEO.COUNTRY_NAME}).'"
SecRule GEO:COUNTRY_CODE "@rx (?:JO|CY|SG)"
```

14.3.10. Only Allow Traffic from One Continent

Here is an example chain of rules that blocks all non-North American traffic, returning a 403 Forbidden status code:

```
SecRule REMOTE_ADDR "@geoLookup" "chain,id:5005004,phase:1,deny,status:403, \
    msg:'Non-North American IP address: address is in %{GEO.COUNTRY_NAME} (%{GEO.COUNTRY_CODE}).'"
SecRule GEO:COUNTRY_CONTINENT "!@streq NA"
```

The above example could be changed to block all traffic from the specified continent by removing the '!' in the last line.

14.4. Redirecting Blocked Requests to a Custom Webpage

By default, when a request is blocked at the WAF a ***403 Forbidden*** status code is returned to the client. This behaviour can be changed to instead return a redirect in the form of a ***302 Found*** status code. In this way, blocked requests are redirected to a specified URL. For example, all blocked requests could be redirected to a custom webpage, for example a branded and publicly accessible error page instructing users to contact IT support for further assistance.

Here is an example directive which redirects all blocked requests to the English language front page of Wikipedia:

```
SecRuleUpdateActionById 981176 "chain,redirect:https://en.wikipedia.org"
```

When copying the above example into a Metaswitch WAF rule set, modify the URL specified. This can be changed to whatever URL blocked clients should be redirected to.

If using the above directive, it is recommended to place it in the rule set under the ***Miscellaneous custom rules*** section.

14.5. Blocking Access to the /PPS Directory at the WAF

It is possible to block all access to the `/pps` directory at the WAF. This can be used to protect the older style provisioning service, where enabled.

Here is an example rule which blocks all connections attempting to access the `/pps` directory:

```
SecRule REQUEST_FILENAME "(?i)/pps/" \
    "id:5900100,phase:1,deny,nolog,status:403, \
    t:none,t:htmlEntityDecode,t:lowercase,t:removeNulls,t:removeWhitespace"
```

Such a rule could be temporarily commented out if and when it was necessary to allow genuine access to the `/pps` directory to provision phones.

If a custom path is used for the PPS service then this can be specified in the rule by replacing `pps` with the appropriate location.

15. WAF OWASP Protection

The WAF service included on Loadbalancer.org appliances is based on the ModSecurity open source project.

The default vulnerability rule set is based on the 'OWASP Top 10' (Open Worldwide Application Security Project Top 10). This defines ten areas of vulnerability that can affect web applications. These are summarized in the following section.

15.1. OWASP Top 10 Application Security Risks: 2017

Category	Description
A1—Injection	Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
A2—Broken Authentication	Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
A3—Sensitive Data Exposure	Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.
A4—XML External Entities (XXE)	Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
A5—Broken Access Control	Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

Category	Description
A6—Security Misconfiguration	Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.
A7—Cross-Site Scripting (XSS)	XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A8—Insecure Deserialization	Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.
A9—Using Components with Known Vulnerabilities	Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.
A10—Insufficient Logging and Monitoring	Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

16. Testing & Verification

> **Note** For additional guidance on diagnosing and resolving any issues you may have, please also refer to Diagnostics & Troubleshooting.

16.1. SIP Provisioning Protection Test

> **Note**
>
> The load balancer appliance is "secure by default". This means that (password based) console and SSH access are both disabled by default.
>
> For information on what this means, how to enable password based access, and the ramifications of doing so, refer to the section *Appliance Security - Security Mode* in the Administration Manual.

The test command presented below can be executed on the active load balancer itself, on the passive load balancer node, or on a Unix/Linux client that has network connectivity to the load balancers.

The command makes 30 configuration retrieval requests per minute, which is sufficient to exceed the default threshold of 20 requests allowed per minute. This means the test client will be flagged and eventually blocked.

The IP address in the command should be changed from `10.0.0.100` to the IP address of the virtual service to be tested.

Using the default, recommended rule set parameters, the test client's IP address will be blocked after three minutes, at which point the response from the WAF service will change to:

```
429 Too Many Requests
```

This indicates that a block is in place and the test has been successful.

```
start_time=$(date +%s); while true; do timer=$(($(date +%s) - start_time)); printf "\nTimer:
%02d:%02d\n" $((timer / 60)) $((timer % 60)); curl -v 10.0.0.100/sip-ps/$(openssl rand -hex 6).cfg
-H "Host: testing.script.com" 2>&1 > /dev/null | grep -E "< HTTP|GET"; sleep 2; done
```

After a successful test, if desired, the blocked test IP address can be unblocked. This can be accomplished by truncating the underlying database files to 0 bytes (emptying them) using the following command:

```
truncate -s 0 /var/log/mod_security/*
```

16.2. Metaswitch Specific Fail Over / High Availability Test

This test is disruptive to end users and should not be run on a live production system.

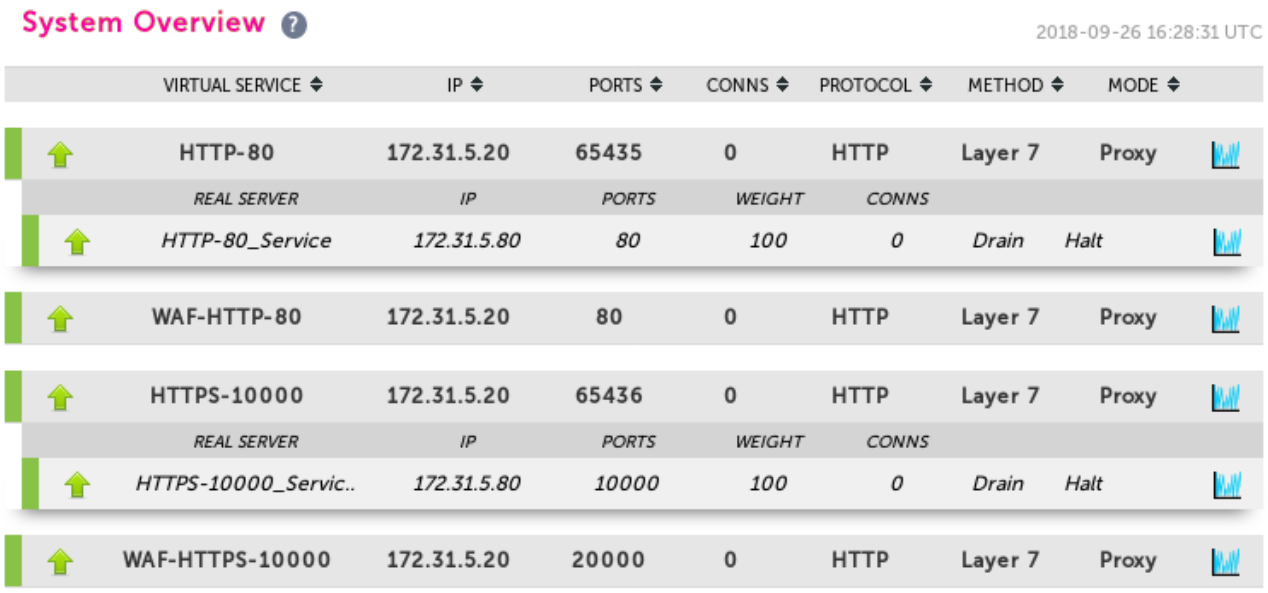
If using a highly available pair of load balancers, the fail over functionality between them can be tested.

1. Log in to the CommPortal web interface through the WAF gateway protected IP address.
2. Trigger a fail over from your active load balancer to your passive load balancer. You could force this by powering off the active load balancer.
3. Press Ctrl+F5 in browser to force refresh the CommPortal page.
4. Once a successful fail over has taken place, the passive load balancer will become active and will start serving traffic. The browser should show the CommPortal again. Note that a new log in may need to be

performed following a fail over.

16.3. Using System Overview

The System Overview can be viewed in the WebUI. It shows a graphical view of all VIPs & RIPs (i.e. the EAS servers and WAF gateways) and shows the state/health of each server as well as the state of the each cluster as a whole. The example below shows that all EAS servers and WAF gateways are healthy and available to accept connections.



17. Technical Support

For more details about configuring the appliance and assistance with designing your deployment please don't hesitate to contact the support team using the following email address: support@loadbalancer.org.

18. Further Documentation

For additional information, please refer to the Administration Manual.

19. Appendix

19.1. Configuring HA - Adding a Secondary Appliance

Our recommended configuration is to use a clustered HA pair of load balancers to provide a highly available and resilient load balancing solution. We recommend that the Primary appliance is fully configured first, then the Secondary appliance can be added to create an HA pair. Once the HA pair is configured, load balanced services must be configured and modified on the Primary appliance. The Secondary appliance will be automatically kept in sync.

> **Note** For Enterprise Azure, the HA pair should be configured first. For more information, please refer to the Azure Quick Start/Configuration Guide available in the documentation library

The clustered HA pair uses Heartbeat to determine the state of the other appliance. Should the active device (normally the Primary) suffer a failure, the passive device (normally the Secondary) will take over.

19.1.1. Non-Replicated Settings

A number of settings are not replicated as part of the Primary/Secondary pairing process and therefore must be manually configured on the Secondary appliance. These are listed by WebUI menu option in the table below:

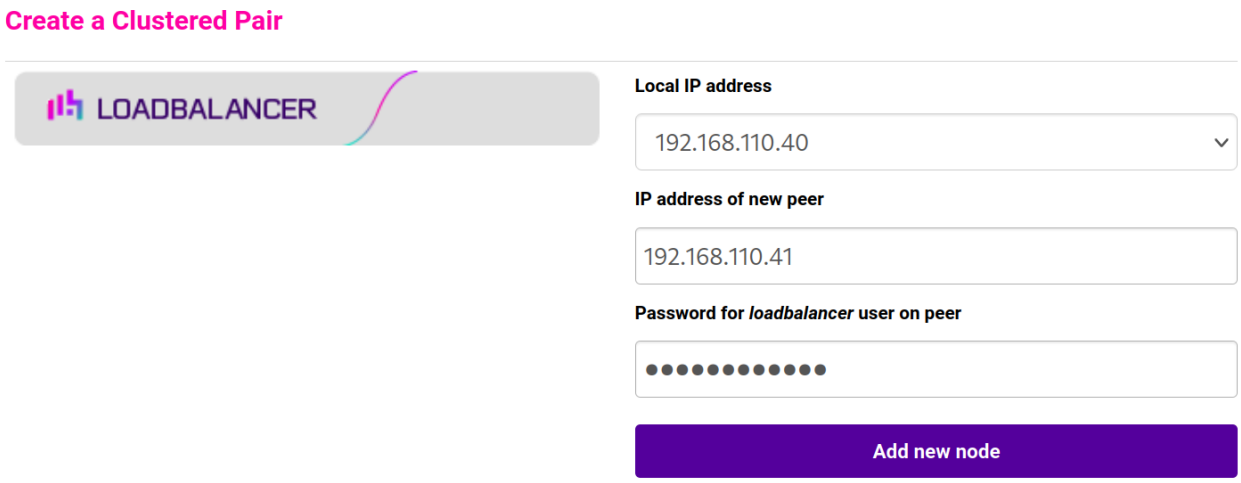
WebUI Main Menu Option	Sub Menu Option	Description
Local Configuration	Hostname & DNS	Hostname and DNS settings
Local Configuration	Network Interface Configuration	Interface IP addresses, bonding configuration and VLANs
Local Configuration	Routing	Default gateways and static routes
Local Configuration	System Date & time	Time and date related settings
Local Configuration	Physical – Advanced Configuration	Various appliance settings
Local Configuration	Portal Management	Portal management settings
Local Configuration	Security	Security settings
Local Configuration	SNMP Configuration	SNMP settings
Local Configuration	Graphing	Graphing settings
Local Configuration	License Key	Appliance licensing
Maintenance	Backup & Restore	Local XML backups
Maintenance	Software Updates	Appliance software updates
Maintenance	Fallback Page	Fallback page configuration
Maintenance	Firewall Script	Firewall (iptables) configuration
Maintenance	Firewall Lockdown Wizard	Appliance management lockdown settings

> **Important** Make sure that where any of the above have been configured on the Primary appliance, they're also configured on the Secondary.

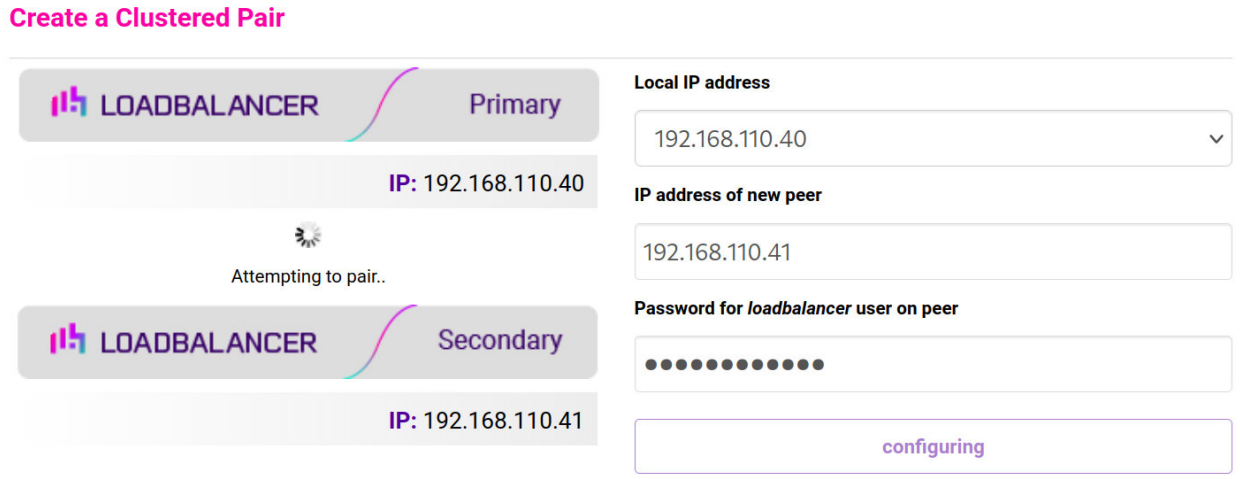
19.1.2. Configuring the HA Clustered Pair

> **Note** If you have already run the firewall lockdown wizard on either appliance, you'll need to ensure that it is temporarily disabled on both appliances whilst performing the pairing process.

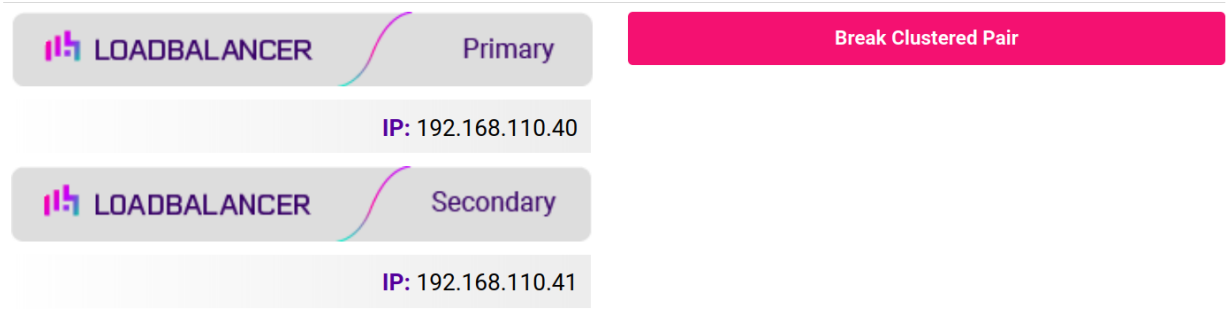
1. Deploy a second appliance that will be the Secondary and configure initial network settings.
2. Using the WebUI on the Primary appliance, navigate to: *Cluster Configuration > High-Availability Configuration*.



3. Specify the IP address and the *loadbalancer* user's password for the Secondary (peer) appliance as shown in the example above.
4. Click **Add new node**.
5. The pairing process now commences as shown below:



6. Once complete, the following will be displayed on the Primary appliance:



7. To finalize the configuration, restart heartbeat and any other services as prompted in the "Commit changes" message box at the top of the screen.

> **Note** Clicking the **Restart Heartbeat** button on the Primary appliance will also automatically restart heartbeat on the Secondary appliance.

> **Note** For more details on configuring HA with 2 appliances, please refer to Appliance Clustering for HA.

> **Note** For details on testing and verifying HA, please refer to Clustered Pair Diagnostics.

20. Document Revision History

Version	Date	Change	Reason for Change	Changed By
1.0.0	27 September 2018	Initial version		AH
1.0.1	3 October 2018	Added new section at the beginning, "Overview", outlining the business case Expanded and rewrote parts of "Custom WAF Rule Set Description" Rewrote WAF rule sets to be more human readable, and added username logging	Required updates	AH
1.0.2	15 October 2018	Changed the wording of section 1, "Overview", so that for hardware deployments (as well as virtual deployments) of WAF appliances end customers are directed to Loadbalancer.org instead of Metaswitch Removed the wording in section 3, "Loadbalancer.org Appliances Supported", that differentiates between hardware and virtual deployments. This changes brings this section in line with our other deployment guides	Required updates	AH
1.1.0	20 November 2018	Changed the document title from "Loadbalancer.org WAF Gateway with Metaswitch EAS DSS/SSS" to "Loadbalancer.org WAF Gateway with Metaswitch EAS" Major overhaul of the document, based on feedback from Metaswitch regarding how they envision WAF services being deployed Broke up the implementation into three different scenarios, based on three existing EAS deployment types (hardware, virtual, and no existing load balancers) Implemented the specific suggestions from Metaswitch regarding how to reference their EAS product, and miscellaneous changes and corrections advised by Metaswitch	Required updates	AH

Version	Date	Change	Reason for Change	Changed By
1.1.1	21 January 2019	Updated the WAF rule sets Added the new "Company Contact Information" page	Required updates	AH
1.2.0	3 May 2019	Changed "protect against the application" wording in the overview section Added labels showing the traffic source and destination to the 'Conceptual Overview' diagrams for clarity Added a section describing the 'Core WAF and Extended WAF' modes Changed the persistence mode for all virtual services to 'X-Forwarded-For and Source IP' Added sections describing detection only mode and how to fully enable the WAF rule engine in the WebUI Added a paragraph describing deployments where changing DNS records is necessary to redirect production traffic Removed the complete rule sets from the appendix and added additional hyperlinks to the rule sets on our website Added sections describing how to set the recommended PCRE match limits and explaining their function Added section "Legacy TLS/SSL Options for Older Hardware" describing enabling TLS 1.0 and SSL 3.0 Added new section 14, "Writing Site-Specific Custom Rules" Added section "NIC Bonding for Link-Level Redundancy"	Required updates	AH

Version	Date	Change	Reason for Change	Changed By
1.3.0	29 August 2019	Styling and layout Added new paragraph to the "Custom Rules" section: 'Redirecting Blocked Requests to a Custom Webpage' Added instructions to enable the 'Accept Invalid HTTP Requests' option for all layer 7 services by default, to explicitly accept error-causing HTTP requests that contain unusual characters Added instructions to increase all the client and server timeouts for EAS/WAF related layer 7 services to 15 minutes to account for Accession's 890000 ms timeouts Added instructions to increase all WAF proxy timeouts to 900 seconds to account for Accession's 890000 ms timeouts	General styling updates Required updates	AH
1.3.1	22 May 2020	New title page Added instructions on defining an ACL to each front end WAF service	Branding update Technical update to account for long-held HTTP requests	AH
1.3.2	22 September 2020	Amended introductory text in section 1, "Overview" Updated Canadian contact details	Bringing document in line with updated marketing copy Change to Canadian contact details	AH

Version	Date	Change	Reason for Change	Changed By
2.0.0	20 May 2021	Amended the list of supported appliances Added clarifying paragraph about the EAS services in question Changed the default SSL offload service mode to 'High Compatibility' and rewrote the 'Legacy TLS/SSL Options' section to reflect the change Added coverage of the new SIP provisioning protection rules Added coverage of logging/alerting Removed the concept of 'Extended' and 'Core' WAF modes Added new instruction to set the inbound anomaly score to 12 Changed to use inclusive language throughout the document Added guidance for deployments with custom PPS paths Added notes explaining the "secure by default" problem and where to find further information Added Metaswitch guidance for using HTTPS exclusively, rather than HTTP, for internet facing services Added notes regarding the PAT log in mechanism Updated all code examples to bring them in line with the latest version of the WAF rule set	Technical change reflecting the requirements of the new rule set Useful clarification Technical simplification due to the prevalence of older phones lacking TLS 1.2 support Required updates	AH
2.0.1	5 November 2021	Added new guidance section, "Warning Regarding Older Certificates"	Useful explicit explanation of problem observed with multiple deployments	AH

Version	Date	Change	Reason for Change	Changed By
2.1.0	1 January 2022	Converted the document to AsciiDoc	Move to new documentation system	AH, RJC, ZAC
2.2.0	27 January 2022	Update minimum supported Loadbalancer.org version to 8.6.1 Update screenshots and instructions for 8.6.1 Add description of new Log4j/Log4Shell rule Add additional screenshots showing example HTTPS services, to add to the existing HTTP-only ones, for clarity Fix AsciiDoc formatting errors Synchronise geolocation database instructions between deployment guide and actual rule set	Required changes for the 8.6 product release Description of Log4Shell rule set addition requested and tested by Metaswitch Fixes from the AsciiDoc conversion process Clarifications based on feedback	AH
2.2.1	28 February 2022	Added note block to SIP provisioning protection explanation	Requested clarification on what requests are affected by the provisioning protection rules	AH
2.2.2	28 September 2022	Updated layer 7 VIP and RIP creation screenshots	Reflect changes in the web user interface	AH
2.2.3	5 January 2023	Combined software version information into one section Added one level of section numbering Added software update instructions Added table of ports used by the appliance Reworded 'Further Documentation' section Removed references to the colour of certain UI elements	Housekeeping across all documentation	AH

Version	Date	Change	Reason for Change	Changed By
2.2.4	2 February 2023	Updated screenshots	Branding update	AH
2.2.5	7 March 2023	Removed conclusion section	Updates across all documentation	AH
2.3.0	24 March 2023	New document theme Modified diagram colours	Branding update	AH

LOADBALANCER

Visit us: www.loadbalancer.org

Phone us: +44 (0)330 380 1064

Phone us: +1 833 274 2566

Email us: info@loadbalancer.org

Follow us: @loadbalancer.org

About Loadbalancer.org

Loadbalancer.org's mission is to ensure that its clients' businesses are never interrupted. The load balancer experts ask the right questions to get to the heart of what matters, bringing a depth of understanding to each deployment. Experience enables Loadbalancer.org engineers to design less complex, unbreakable solutions - and to provide exceptional personalized support.