Contents

1. Overview................................................................................................................................. 5
   Introduction to the WAF Gateway with Metaswitch EAS......................................................... 5
   Virtualized Deployments........................................................................................................ 5
   Hardware Deployments........................................................................................................... 5
   Non-Load Balanced Deployments.......................................................................................... 5
2. About this Guide..................................................................................................................... 5
3. Loadbalancer.org Appliances Supported............................................................................... 5
4. Loadbalancer.org Software Versions Supported.................................................................... 6
5. Metaswitch EAS Versions Supported..................................................................................... 6
6. Sizing, Capacity, and Performance for a Virtual WAF Gateway Deployment............................ 6
7. Using Loadbalancer.org WAF Gateways with Metaswitch EAS................................................ 7
   Conceptual Overview.............................................................................................................. 7
      Example: HTTP Service......................................................................................................... 7
      Example: HTTPS Service....................................................................................................... 7
   TLS/SSL Termination................................................................................................................ 8
      Legacy TLS/SSL Options for Older Equipment................................................................. 8
   Core WAF and Extended WAF Options................................................................................ 8
      Core WAF (Default)............................................................................................................... 8
   Extended WAF........................................................................................................................ 9
   Load Balancing & HA Requirements..................................................................................... 9
   NIC Bonding for Link-Level Redundancy.............................................................................. 9
8. Deployment Concept.............................................................................................................. 10
   Scenario 1 – Adding WAF Gateways to a Deployment Currently Using Hardware Load Balancers... 10
      Virtual Service (VIP) Requirements.................................................................................. 10
   Scenario 2 – Adding WAF Gateways to a Deployment Currently Using Virtual Load Balancers.... 11
      Virtual Service (VIP) Requirements.................................................................................. 12
   Scenario 3 – Adding WAF Gateways to a Non-Load Balanced Deployment............................ 12
      Virtual Service (VIP) Requirements.................................................................................. 12
9. Loadbalancer.org Appliance – the Basics............................................................................. 14
   Virtual Appliance Download & Deployment....................................................................... 14
   Initial Network Configuration................................................................................................. 14
   Accessing the Web User Interface (WebUI)........................................................................... 14
   HA Clustered Pair Configuration............................................................................................ 16
10. Appliance Configuration for Hardware Deployments (Scenario 1)............................................ 17
    Deploy the New Pair of Load Balancers............................................................................... 17
    Assign IP addresses in the Required Subnets..................................................................... 17
    Recreate the existing virtual services Requiring WAF Protection........................................ 17
    Identify and Note.................................................................................................................... 17
    Recreating the Virtual Service............................................................................................. 18
    Creating the WAF Gateway.................................................................................................. 19
    Setting Up the TLS/SSL Termination................................................................................... 20
    Recreating the Remaining Existing Virtual Services............................................................ 22
    Setting the PCRE Match Limits............................................................................................. 22

© Copyright Loadbalancer.org • www.loadbalancer.org • sales@loadbalancer.org
11. Appliance Configuration for Virtual Deployments (Scenario 2)

Considering the Resources Assigned to the Virtual Load Balancers
Recreate the existing virtual services Requiring WAF Protection
Identify and Note
Recreating the Virtual Service
Creating the WAF Gateway
Setting Up the TLS/SSL Termination
Recreating the Remaining Existing Virtual Services
Setting the PCRE Match Limits
Setting the Recommended Values
Finalizing the Configuration
Putting the New WAF Services Into Production (Detection Only Mode)
Directing Traffic to the WAF Services
Fully Enabling the WAF Rule Engine

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

Deploy the Pair of Load Balancers
Assign IP addresses in the Required Subnets
Identify the services Requiring WAF Protection
Identify and Note
creating the Virtual Service
Creating the WAF Gateway
Setting Up the TLS/SSL Termination
Creating the Remaining Virtual Services
Setting the PCRE Match Limits
Setting the Recommended Values
Finalizing the Configuration
Putting the New WAF Services Into Production (Detection Only Mode)
Directing Traffic to the WAF Services
Fully Enabling the WAF Rule Engine

13. Custom WAF Rule Set Description

Whitelisted rules
Rule 1: DOS Protection for login.html
Rule 2: Detect Failed Login and Block
Rule 3: DOS Protection POST to /login
Rule 4: Detect username abuse and block
Rule 5: Detect password abuse and block

14. Writing Site-Specific Custom Rules
21. Document Revision History

20. Appendix

19. Conclusion

18. Further Documentation

17. Technical Support

16. Testing & Verification

15. WAF OWASP Protection


13. Metaswitch Specific Fail Over / High Availability Test

12. Using System Overview

11. Blocking and Whitelisting by IP Address

10. Whitelisting Trusted Addresses

9. Blocking Bad Addresses

8. Blocking by User-Agent Request Header

7. Searching for Multiple Phrases

6. Blocking by Geographic Location

5. Country and Continent Codes

4. Only Allow Traffic from One Country

3. Only Allow Traffic from a Range of Countries

2. Explicitly Block Traffic from a Specific Country

1. Explicitly Block Traffic from a Range of Countries

0. Only Allow Traffic from One Continent

- Redirecting Blocked Requests to a Custom Webpage

- Blocking Access to the IPPS Directory at the WAF

© Copyright Loadbalancer.org - www.loadbalancer.org - sales@loadbalancer.org
1. Overview

**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. An off-the-shelf WAF is not very useful unless it is configured to protect a specific application. Loadbalancer.org have developed five custom WAF rules specifically to protect a Metaswitch EAS deployment, ensuring total protection against security vulnerabilities. This is described in detail in this document.

**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.

**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.

**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 relevant Administration Manual:

- v7 Administration Manual
- v8 Administration Manual

3. Loadbalancer.org Appliances Supported

All our products can be used for a WAF gateway with Metaswitch EAS deployment. The complete list of models is shown below:
Discontinued Models | Current Models *
--- | ---
Enterprise R16 | Enterprise R20
Enterprise VA R16 | Enterprise MAX
Enterprise VA | Enterprise 10G
Enterprise R320 | Enterprise 40G
Enterprise Ultra | Enterprise VA R20
Enterprise VA MAX | Enterprise AWS **
Enterprise AWS ** | Enterprise AZURE **

* For full specifications of these models please refer to: [http://www.loadbalancer.org/products/hardware](http://www.loadbalancer.org/products/hardware)

** Some features may not be supported, please check with Loadbalancer.org support

4. Loadbalancer.org Software Versions Supported

- V8.3.4 and later

5. Metaswitch EAS Versions Supported

- Metaswitch EAS – all versions

6. 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.
7. 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.

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 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)

The specific details vary depending on the type of EAS deployment in question, for example the location where the chains of services need to be created varies.

In general, for each service that needs protecting, a chain of the above elements must be created on a Loadbalancer.org appliance.

Example: HTTP Service

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

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 the 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.

**TLS/SSL Termination**

A WAF gateway can only have plain text HTTP traffic passed to it for scanning. If HTTPS-based 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 toward the Metaswitch EAS servers. This is because, for HTTPS services, the EAS servers are 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 type.

**Legacy TLS/SSL Options For Older Equipment**

In some deployments, it can be necessary to enable legacy TLS/SSL options on the load balancer. This may be required to support older equipment, for example legacy hardware phones that don’t support any recent TLS/SSL cryptographic protocols.

The legacy options can be found in the load balancer’s WebUI. Navigate to Cluster Configuration > SSL Termination, click on **Modify** next to the termination service in question, and set the SSL Operation Mode to **Custom**. Uncheck the Disable X Ciphers check boxes as necessary, and click **Update** to save all changes. When prompted, click the **Reload STunnel** and **Reload HAProxy** buttons to put the amended configuration into effect.

**Core WAF and Extended WAF Options**

The WAF solution has two modes of operation: ‘core WAF’ and ‘extended WAF’.

**Core WAF (Default)**

The default mode of operation is ‘core WAF’. This setup uses the five custom WAF rules that have been written to protect the CommPortal login from attack.

The core WAF option can be quickly and easily implemented, and is likely to be disruption-free in a production environment.
Extended WAF

The optional ‘extended WAF’ mode uses the same five custom WAF rules from the ‘core’ mode to protect the CommPortal login. The extended mode also uses the industry standard OWASP WAF rule set to scan all traffic going to the EAS servers for malicious looking behaviour, not only traffic destined for the CommPortal login.

A comprehensive amount of whitelisting is required in the WAF to allow genuine traffic through to the EAS servers and to prevent false positives. A significant amount of whitelisting has been carried out while working with Metaswitch and Metaswitch partners operating large EAS deployments. Despite the whitelisting work that has been completed so far, it is likely that some corner cases and unusual setups will still cause false positives to be flagged in the WAF.

The extended WAF option is likely to require additional work to implement, and potential false positives could block end users in error causing some disruption in a production environment.

Instructions on how to switch to using extended WAF mode are presented in each of the three later sections covering each deployment scenario. The procedure is also detailed in the ‘Core WAF’ / ‘Extended WAF’ setting” section at the beginning of the WAF rule set itself.

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.

NIC Bonding for Link-Level Redundancy

If using physical load balancers, the NICs can be bonded in pairs. ‘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.
8. Deployment Concept

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

Full instructions for setting up this scenario are given in section 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.

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).

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

Full instructions for setting up this scenario are given in section 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:

![Diagram of existing EAS deployment](image)

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

![Diagram of EAS deployment with WAF gateways](image)

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, please see section: Sizing, Capacity, and Performance for a Virtual WAF Gateway Deployment.

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).

Scenario 3 – Adding WAF Gateways to a Non-Load Balanced Deployment
Full instructions for setting up this scenario are given in section 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.

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).
9. Loadbalancer.org Appliance – the Basics

Virtual Appliance Download & Deployment
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 and XEN and has been optimized for each Hypervisor. By default, the VA is allocated 1 CPU, 2GB of RAM and has an 8GB 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 the Administration Manual and the ReadMe.txt text file included in the VA download for more detailed information on deploying the VA using various Hypervisors.

Initial Network Configuration
The IP address, subnet mask, default gateway and DNS settings can be configured in several ways as detailed below:

Method 1 - Using the Network Setup Wizard at the console
After boot up, follow the instructions on the console to configure the IP address, subnet mask, default gateway and DNS settings.

Method 2 - Using the WebUI
Using a browser, connect to the WebUI on the default IP address/port: https://192.168.2.21:9443
To set the IP address & subnet mask, use: Local Configuration > Network Interface Configuration
To set the default gateway, use: Local Configuration > Routing
To configure DNS settings, use: Local Configuration > Hostname & DNS

Accessing the Web User Interface (WebUI)
The WebUI can be accessed via HTTPS at the following URL: https://192.168.2.21:9443/lbadmin
* Note the port number → 9443

(replace 192.168.2.21 with the IP address of your load balancer if it’s been changed from the default)
Login using the following credentials:

**Username:** loadbalancer  
**Password:** loadbalancer

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

Once logged in, the WebUI will be displayed as shown on the following page:
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 slave unit is covered in section 1 of the appendix on page 48.
10. Appliance Configuration for Hardware Deployments (Scenario 1)

The end result should look like the following diagram:

**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](http://loadbalancer.org).

Once deployed, the new load balancers should be paired together to create a highly available clustered pair. This procedure is covered in section 1 of the appendix on page 48.

**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.

**Recreate the existing virtual services Requiring WAF Protection**

**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)

**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 master 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**
   - 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
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

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. Click Update to create the WAF gateway

![WAF - Add A New Gateway]

4. Click Modify next to the newly created WAF service
5. Set WAF Proxy Timeout to 900 (the units are seconds)
6. Click Update
7. Navigate to Cluster Configuration > Layer 7 – Virtual Services
8. Click Modify next to the automatically created WAF VIP, e.g. WAF-SERV_HTTP-80
9. In the Protocol section click Advanced to expand the menu
10. Enable option Accept Invalid HTTP Requests by checking the checkbox
11. In the Other section click Advanced to expand the menu
12. Check the Timeout checkbox
13. Set Client Timeout to 900000 (the units are milliseconds; this value equates to 15 minutes)
14. Set Real Server Timeout to 901000 (the units are milliseconds; this value equates to 15 minutes 1 second)
15. Click Update
17. 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
18. Paste your custom Metaswitch EAS WAF configuration into the text box

Note: Two complete Metaswitch EAS WAF configurations, based on custom rule sets, can be found on our website. Either of these configurations can be copy and pasted into the text box in their entirety. Links to the rule sets can be found in section 13, Custom WAF Rule Set Description, on page 38.

19. Optional: by default, the WAF rule set is set to ‘core WAF’ mode. If desired, to switch to ‘extended WAF’ mode, comment out the following line (by putting a # at the beginning of the line) near the top of the rule set:
   
   `SecRule REQUEST_FILENAME "!(?i)^.*/login"` 

20. Click Update to save the manual WAF configuration

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 on page 22.

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

Further information on creating PEM files and converting between certificate formats is presented in our Administration Manual: http://pdfs.loadbalancer.org/loadbalanceradministrationv8.pdf

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 Certificate drop-down list, select the certificate for the service in question, which in this example is northwest-telco.com
5. Click Update to create the TLS/SSL termination service
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 on page 17.

Setting the PCRE Match Limits

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 green PCRE Match Limit Recursion button

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.

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 service, and stunnel must be reloaded as follows:

1. Using the WebUI, navigate to: Maintenance > Restart Services and click Reload HAProxy
2. On the Restart Services page, click Reload WAF
3. On the Restart Services page, click Reload STunnel
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.

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.

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 HAProy and Reload WAF when prompted, to put the change into effect
11. Appliance Configuration for Virtual Deployments (Scenario 2)

The end result should look like the following diagram:

---

**Considering the Resources Assigned to the Virtual Load Balancers**

The WAF gateway service on the load balancer is CPU and 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.

**Recreate the existing virtual services Requiring WAF Protection**

**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](mailto:support@loadbalancer.org) for 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

**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**.

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**
   - 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
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

**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,
3. Click Update to create the WAF gateway

4. Click Modify next to the newly created WAF service
5. Set WAF Proxy Timeout to 900 (the units are seconds)
6. Click Update
7. Navigate to Cluster Configuration > Layer 7 – Virtual Services
8. Click Modify next to the automatically created WAF VIP, e.g. WAF-NEW_SERV_HTTP-80
9. In the Protocol section click Advanced to expand the menu
10. Enable option Accept Invalid HTTP Requests by checking the checkbox
11. In the Other section click Advanced to expand the menu
12. Check the Timeout checkbox
13. Set Client Timeout to 900000 (the units are milliseconds; this value equates to 15 minutes)
14. Set Real Server Timeout to 901000 (the units are milliseconds; this value equates to 15 minutes 1 second)
15. Click Update
17. 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
18. Paste your custom Metaswitch EAS WAF configuration into the text box

Note: Two complete Metaswitch EAS WAF configurations, based on custom rule sets, can be found on our website. Either of these configurations can be copy and pasted into the text box in their entirety. Links to the rule sets can be found in section 13, Custom WAF Rule Set Description, on page 38.

19. Optional: by default, the WAF rule set is set to ‘core WAF’ mode. If desired, to switch to ‘extended WAF’ mode, comment out the following line (by putting a # at the beginning of the line) near the top of the rule set:

   `SecRule REQUEST_FILENAME "!(?i)^.*/login"`

20. Click Update to save the manual WAF configuration

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
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.


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 Certificate drop-down list, select the certificate for the service in question, which in this example is northwest-telco.com.
5. Click Update to create the TLS/SSL termination service.
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 on page 17.

Setting the PCRE Match Limits

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 green PCRE Match Limit Recursion button

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.

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 service, and stunnel must be reloaded as follows:

1. Using the WebUI, navigate to: Maintenance > Restart Services and click Reload HAProxy
2. On the Restart Services page, click Reload WAF
3. On the Restart Services page, click Reload STunnel
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.

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.

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 HAProy and Reload WAF when prompted, to put the change into effect
12. Appliance Configuration for Non-Load Balanced Deployments (Scenario 3)

The end result should look like the following diagram:

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 section 1 of the appendix on page 48.

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.

Identify the services Requiring WAF Protection

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.

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 master 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. 172.315.200

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
   - 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
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**

**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. Click **Update** to create the WAF gateway

![WAF - Add A New Gateway](image)

4. Click **Modify** next to the newly created WAF service
5. Set **WAF Proxy Timeout** to 900 (the units are seconds)
6. Click **Update**
7. Navigate to **Cluster Configuration > Layer 7 – Virtual Services**
8. Click **Modify** next to the automatically created WAF VIP, e.g. WAF-HTTP-80
9. In the **Protocol** section click **Advanced** to expand the menu
10. Enable option **Accept Invalid HTTP Requests** by checking the checkbox
11. In the **Other** section click **Advanced** to expand the menu
12. Check the **Timeout** checkbox
13. Set **Client Timeout** to 900000 (the units are milliseconds; this value equates to 15 minutes)
14. Set **Real Server Timeout** to 901000 (the units are milliseconds; this value equates to 15 minutes 1 second)
15. Click **Update**
16. Navigate to **Cluster Configuration > WAF – Manual Configuration**
17. 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
18. Paste your custom Metaswitch EAS WAF configuration into the text box

Note: Two complete Metaswitch EAS WAF configurations, based on custom rule sets, can be found on our website. Either of these configurations can be copy and pasted into the text box in their entirety. Links to the rule sets can be found in section 13, [Custom WAF Rule Set Description](#), on page 38.

19. **Optional**: by default, the WAF rule set is set to ‘core WAF’ mode. If desired, to switch to ‘extended WAF’ mode, comment out the following line (by putting a # at the beginning of the line) near the top of the rule set:

   ```
   SecRule REQUEST_FILENAME "!(?i)^.*/login" 
   ```

20. Click **Update** to save the manual WAF configuration

### 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](#) on page 36.
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**

Further information on creating PEM files and converting between certificate formats is presented in our *Administration Manual*: [http://pdfs.loadbalancer.org/loadbalanceradministrationv8.pdf](http://pdfs.loadbalancer.org/loadbalanceradministrationv8.pdf)

### 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 Certificate drop-down list,** select the certificate for the service in question, which in this example is **northwest-telco.com**
5. **Click Update to create the TLS/SSL termination service**
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 on page 17.

Setting the PCRE Match Limits

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 green PCRE Match Limit Recursion button

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.

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 service, and stunnel must be reloaded as follows:

1. Using the WebUI, navigate to: Maintenance > Restart Services and click Reload HAProxy
2. On the Restart Services page, click Reload WAF
3. On the Restart Services page, click **Reload STunnel**

**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.

**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.

**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
13. Custom WAF Rule Set Description

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

Two complete WAF rule sets can be found on our website:


- A rule set for use where a significant proportion of users share the same IP address: the NAT Deployment Rule Set located at [https://downloads.loadbalancer.org/metaswitch/metaswitch-waf_nat-deployment-ruleset.txt](https://downloads.loadbalancer.org/metaswitch/metaswitch-waf_nat-deployment-ruleset.txt).

Whitelisted rules

To enable a Metaswitch EAS deployment to work correctly with WAF gateways, it is necessary to whitelist some rules from the standard WAF OWASP ModSecurity core rule set (further details about this can be found in the WAF OWASP Protection section on page 45). The whitelisted rules prevent genuine, safe Metaswitch EAS traffic from being blocked.

The set of whitelisted rules can be found at the top of each Metaswitch specific rule set.

**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.

The IP DOS for login forms. It records all access to login.html pages. Regardless of whether access is via a GET or a POST request method, any access to this URL will increment a counter.

Increment counter "ip.dos_counter" per page hit

If counter = 200+ set "ip.dos_block" = "1" which restricts access

Both counters reset after 60s

```
SecRule SCRIPT_FILENAME "@rx ^.*login.html$" "phase:5,chain,t:none,nolog,pass, id:5002010,severity:'INFO',tag:'DoS', 
 setvar:ip.dos_counter=+1,expirevar:ip.dos_counter=60"
SecRule IP:dos_counter "@gt 200" "t:none, setvar:ip.dos_block, 
 expirevar:ip.dos_block=60, setvar:ip.dos_counter=0"
```

**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 to block actual failed logins. It detects the presence of either "error/authenticationFailed" or "error=retryLimitExceeded" in the response headers and increments a counter.

Increment counter "ip.authFail-retryLimit_counter" per detected failed login

```
SecRule HTTP:REQUEST_HEADERS "error.*authenticationFailed:1" "t:deny,log,pass, id:5002011,severity:'WARNING',tag:'Detect Failed Login',
 setvar:ip.authFail-retryLimit_counter=+0,expirevar:ip.authFail-retryLimit_counter=60"
```

© Copyright Loadbalancer.org - www.loadbalancer.org - sales@loadbalancer.org
If counter = 20+ set "ip.authFail-retryLimit_block" = "1" which restricts access

ip.authFail-retryLimit_counter decrements 1 every 300s

ip.authFail-retryLimit_block counter reset after 600s

---

```plaintext
SecRule RESPONSE_HEADERS:Location "@contains error=authenticationFailed" 
  "phase:5, id:5002020, chain, t:none, nolog, pass, \ 
  setvar: ip.authFail-retryLimit_counter=+1, deprecatevar: ip.authFail-retryLimit_counter=1/300"
SecRule ip:authFail-retryLimit_counter "@gt 20" "t:none, setvar: ip.authFail-retryLimit_block=1, \ 
  expirevar: ip.authFail-retryLimit_block=600, setvar: ip.authFail-retryLimit_counter=0"

SecRule RESPONSE_HEADERS:Location "@contains error=retryLimitExceeded" 
  "phase:5, id:5002021, chain, t:none, nolog, pass, \ 
  setvar: ip.authFail-retryLimit_counter=+1, deprecatevar: ip.authFail-retryLimit_counter=1/300"
SecRule ip:authFail-retryLimit_counter "@gt 20" "t:none, setvar: ip.authFail-retryLimit_block=1, \ 
  expirevar: ip.authFail-retryLimit_block=600, setvar: ip.authFail-retryLimit_counter=0"
```
Rule 5: Detect password abuse and block

This detects password based abuse. It detects when the same password is used in multiple POST requests and increment a counter. This rule is written so that it will only apply to IP addresses that have more than 2 login failures recorded against them from Rule 2: Detect Failed Login and Block.

If counter "authFail-retryLimit_counter" > 2 (i.e. the IP address already has more than 2 failed logins against it)
Increment counter "global.password_counter" per attempt using the same password
If counter = 20+ set "global.password_block" = "1" which restricts access
global.password_counter decrements 1 every 180s
global.password_block counter reset after 600s

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 the highly extensible nature of the WAF functionality.

The downloadable WAF rule sets are thoroughly commented, so as to be self contained, and describe how to write additional rules as well as providing examples. That content is mirrored here for completeness, as well as being more verbose and providing additional advice.

Blocking and Whitelisting 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

Whitelisting Trusted Addresses

Known trusted IP addresses and subnets can be whitelisted to explicitly ensure that they are not scanned by the WAF. This is useful if the 'extended WAF' mode is in use, as it 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 whitelisting 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,nolog,ctl:ruleEngine=Off"
```

Here is another example whitelisting 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,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
```

**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,deny,status:403"
```

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

**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 REMOTE_ADDR "@IPMatch 52.51.5.169" "id:5003002,deny,status:403"
```
SecRule REQUEST_HEADERS:User-Agent "(?i)python" "log,deny,id:5004000, \
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.

**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 significantly more efficient at searching for multiple keywords than a regular expression.

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" \
"log,deny,id:5004001,msg:'Suspicious User-Agent detected.'"

**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. The easiest way to download this database onto a load balancer is to 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
```

If using an HA pair of load balancers, the command must be run on both the master appliance and the slave appliance.

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

**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

**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, by returning a 403 Forbidden status code:

```plaintext
SecRule REMOTE_ADDR "@geoLookup" "chain,id:5005000,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.

**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, by returning a 403 Forbidden status code:

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

**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, by returning a 403 Forbidden status code:

```plaintext
SecRule REMOTE_ADDR "@geoLookup" "chain,id:5005002,deny,status:403, \ msg:'Paraguayan IP address detected.'"
SecRule GEO:COUNTRY_CODE "!*streq PY"
```
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, by returning a 403 Forbidden status code:

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

Only Allow Traffic From One Continent

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

```
SecRule REMOTE_ADDR @geoLookup "chain,id:5005004,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.

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,t:none, \ redirect:https://en.wikipedia.org, \ log:msg:'Inbound Anomaly Score Exceeded (Total Score: %{TX.ANOMALY_SCORE}, \ SQLi=%{TX.SQL_INJECTION_SCORE}, \ XSS=%{TX.XSS_SCORE}): Last Matched Message: %{tx.msg}', \ logdata:'Last Matched Data: %{matched_var}', \ setvar:tx.inbound_tx_msg=%{tx.msg},setvar:tx.inbound_anomaly_score=%{tx.anomaly_score}"
```

The only thing that should be modified when copying the above example into a Metaswitch WAF rule set is the URL specified in the second line. 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.

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 REMOTE_ADDR @geoLookup "chain,id:5005003,deny,status:403, msg:'Jordanian, Cypriot, or Singaporean IP address detected (%{GEO.COUNTRY_NAME}).'"
SecRule GEO:COUNTRY_CODE "JO|CY|SG"
```
SecRule REQUEST_FILENAME "(?i)/pps/" \ "id:5900100,nolog,deny,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.

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 Web Application Security Project Top 10). This defines ten areas of vulnerability that can affect web applications. These are summarized in the following section.

OWASP Top 10 Application Security Risks: 2017

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1—Injection</td>
<td>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.</td>
</tr>
<tr>
<td>A2—Broken Authentication</td>
<td>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.</td>
</tr>
<tr>
<td>A3—Sensitive Data Exposure</td>
<td>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.</td>
</tr>
<tr>
<td>A4—XML External Entities (XXE)</td>
<td>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.</td>
</tr>
<tr>
<td>A5—Broken Access Control</td>
<td>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.</td>
</tr>
<tr>
<td>A6—Security Misconfiguration</td>
<td>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.</td>
</tr>
<tr>
<td>A7—Cross-Site</td>
<td>XSS flaws occur whenever an application includes untrusted data in a new web page without</td>
</tr>
</tbody>
</table>
Scripting (XSS) | 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

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 into 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

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

19. Conclusion
Loadbalancer.org appliances provide a very cost effective solution for putting a highly available WAF gateway solution in front of Metaswitch EAS environments.
20. Appendix

1 – Clustered Pair Configuration – Adding a Slave Unit

If you initially configured just the master unit and now need to add a slave - our recommended procedure, please refer to the relevant section below for more details:

Note: A number of settings are not replicated as part of the master/slave pairing process and therefore must be manually configured on the slave appliance. These are listed below:

- Hostname & DNS settings
- Network settings including IP addresses, bonding configuration and VLANs
- Routing configuration including default gateways and static routes
- Date & time settings
- Physical – Advanced Configuration settings including Internet Proxy IP address & port, Firewall table size, SMTP relay and Syslog server
- SNMP settings
- Graphing settings
- Firewall Script & Firewall Lockdown Script settings
- Software updates

Version 7:

Please refer to Chapter 8 – Appliance Clustering for HA in the v7 Administration Manual.

Version 8:

To add a slave node – i.e. create a highly available clustered pair:

- Deploy a second appliance that will be the slave and configure initial network settings
- Using the WebUI, navigate to: Cluster Configuration > High-Availability Configuration
• Specify the IP address and the loadbalancer users password (the default is 'loadbalancer') for the slave (peer) appliance as shown above

• Click Add new node

• The pairing process now commences as shown below:

Once complete, the following will be displayed:

To finalize the configuration, restart heartbeat and any other services as prompted in the blue message box at the
top of the screen

Note: Clicking the Restart Heartbeat button on the master appliance will also automatically restart heartbeat on the slave appliance.

Note: Please refer to chapter 9 - Appliance Clustering for HA in the Administration Manual for more detailed information on configuring HA with 2 appliances.
## 21. Document Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Change</th>
<th>Reason for Change</th>
<th>Changed By</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0</td>
<td>27 September 2018</td>
<td>Initial version</td>
<td></td>
<td>AH</td>
</tr>
</tbody>
</table>
| 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/ISSS" 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 | Required updates           | AH         |
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Changes</th>
<th>Updates</th>
<th>Author</th>
</tr>
</thead>
</table>
| 1.1.1   | 21 January 2019 | Updated the WAF rule sets  
Added the new “Company Contact Information” page | Required updates          | AH     |
| 12.0    | 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     |
| 13.0    | 29 August 2019 | Styling and layout  
Added new paragraph to the | General styling updates   | AH     |
"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.

Required updates
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.

United Kingdom
Loadbalancer.org Ltd.
Compass House, North Harbour Business Park, Portsmouth, PO6 4PS
UK: +44 (0) 330 380 1064
sales@loadbalancer.org
support@loadbalancer.org

Canada
Loadbalancer.org Appliances Ltd.
300-422 Richards Street, Vancouver,
BCV6B 2Z4, Canada
TEL: +1 302.213.0122
sales@loadbalancer.org
support@loadbalancer.org

United States
Loadbalancer.org, Inc.
4550 Linden Hill Road, Suite 201
Wilmington, DE 19808, USA
TEL: +1 833.274.2566
sales@loadbalancer.org
support@loadbalancer.org

Germany
Loadbalancer.org GmbH
Tengstraße 2780798,
München, Germany
TEL: +49 (0)89 2000 2179
sales@loadbalancer.org
support@loadbalancer.org