IBM Security

IBM Security Verify Access

V10.0.4 – Next Generation Web Application Firewall
Technical Preview Configuration Guide

October 2022

Table of Contents

Support Disclaimer

The new Web Application Firewall capability in IBM Security Verify Access v10.0.4.0 is provided as a technical preview only. IBM discourages its use in mission critical/production environments. Administrators should be prepared to disable this functionality if it causes unexpected behaviour.

IBM Support is available for this new capability; however for this technical preview release, it is limited to tickets raised with a Severity of 4 (lowest).

Standard support for this new capability is targeted in the next release*.

It should also be noted that IBM does not claim responsibility for the OWASP ModSecurity Core Rule Set (CRS) files which are supplied with IBM Security Verify Access. These files are provided as-is and any issues which are encountered with the Core Rule Set files should be raised using the recommended support procedure for the CRS.

(*) IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Overview

In the past IBM Security Verify Access has provided Web Application Firewall support using the 'Web Content Protection' (WCP) functionality. The engine which is used by the WCP functionality is provided by IBM X-Force and is also used in the Intrusion Protection Systems offered by IBM. Unfortunately, this engine is no longer being developed by IBM and will go out of support at the end of 2022. Customers will continue to be able to use the WCP functionality, but no further updates to the engine will be available after 2022.

An alternative Web Application Firewall capability, which is based on the ModSecurity rules engine, is being introduced into the Reverse Proxy.

ModSecurity, sometimes called Modsec, is an open-source web application firewall (WAF). Originally designed as a module for the Apache HTTP Server, it has evolved to provide an array of Hypertext Transfer Protocol request and response filtering capabilities along with other security features across a number of different platforms including Apache HTTP Server, Microsoft IIS and Nginx.

The platform provides a rule configuration language known as 'SecRules' for real-time monitoring, logging, and filtering of Hypertext Transfer Protocol communications based on user-defined rules.

Although not its only configuration, ModSecurity is most commonly deployed to provide protection against generic classes of vulnerabilities using the OWASP ModSecurity Core Rule Set (CRS). This is an open-source set of rules written in ModSecurity's SecRules language. The project is part of OWASP, the Open Web Application Security Project.

WebSEAL now incorporates the ModSecurity rules processing engine, which can be enabled on a per request basis, and the IBM Security Verify Access firmware embeds v3.3.2 of the CRS.

Limitations

To help protect the Verify Access environments certain ModSecurity rule constructs have been disabled, namely the ‘exec’ and ‘inspectFile’ actions.

In addition to this, certain ModSecurity configuration elements are fixed and cannot be modified. These configuration elements include:

- SecTmpDir
- SecDataDir
- SecUploadDir
- SecDebugLog
- SecAuditLog
- SecAuditLog2
- SecAuditLogType
- SecAuditLogStorageDir
- SecRuleScript
- SecGeoLookupDb
- SecUnicodeMapFile

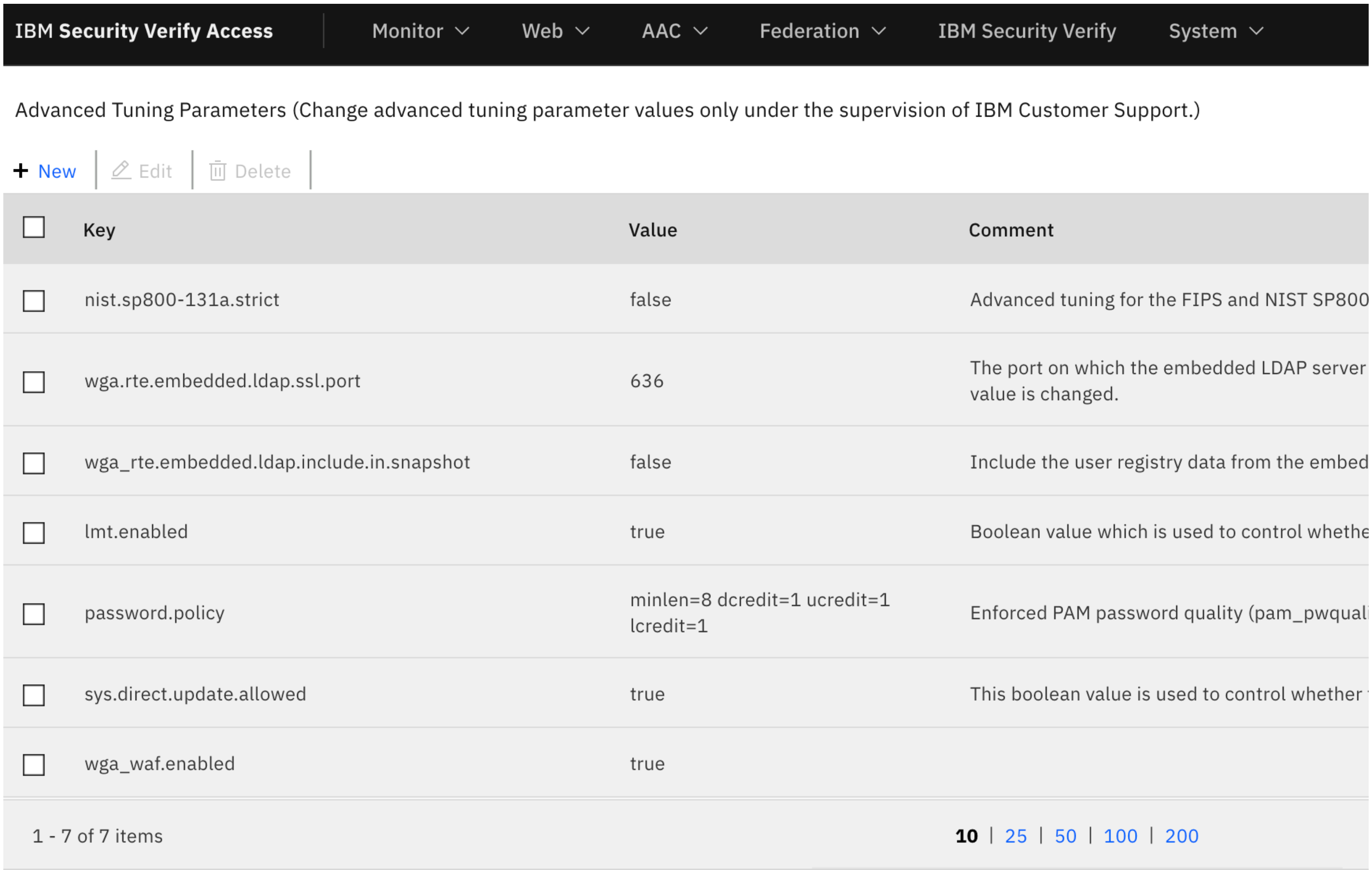
Support for the GeoIP database, used for geolocation rules, is not currently available.

Configuration

Enabling

By default, during the technical preview stage, the new Web Application Firewall functionality is not visible in the Web management console (aka LMI). In order to enable the new functionality, the following advanced tuning parameter must be set:

Name	Value
wga_waf.enabled	true



Advanced Tuning Parameters (Change advanced tuning parameter values only under the supervision of IBM Customer Support.)

☐	Key	Value	Comment
☐	nist.sp800-131a.strict	false	Advanced tuning for the FIPS and NIST SP800
☐	wga.rte.embedded.ldap.ssl.port	636	The port on which the embedded LDAP server value is changed.
☐	wga_rte.embedded.ldap.include.in.snapshot	false	Include the user registry data from the embed
☐	lmt.enabled	true	Boolean value which is used to control whethe
☐	password.policy	minlen=8 dcredit=1 ucredit=1 lcredit=1	Enforced PAM password quality (pam_pwqual
☐	sys.direct.update.allowed	true	This boolean value is used to control whether
☐	wga_waf.enabled	true	

1 - 7 of 7 items **10** \| 25 \| 50 \| 100 \| 200

WebSEAL

In order to trigger the new WAF rules processing for a specific request the ‘request-match’ configuration entry, within the ‘[waf]’ stanza of the reverse proxy configuration file must be set. When a request is received by WebSEAL the HTTP request line, which includes the method, URI and protocol, is pattern matched against each ‘request-match’ configuration entry. If a match is found the request will be passed to the ModSecurity engine for processing.

An example configuration entry would be:

```
[waf]
request-match = GET /jct/*
```

An optional list of ModSecurity phases can be prepended to the configuration entry to indicate the ModSecurity phases which will be invoked for request processing. If a list is not provided all phases will be invoked.

The phases include:

Phase	Description
1	Request Headers
2	Request Body
3	Response Headers
4	Response Body
5	Logging

Example configuration entries could be:

```
[waf]
request-match = [1,2]GET /jct/*
request-match = [1-2,5]GET /jct_2/*
```

The idea behind allowing specific phases to be set is that the performance cost for invoking the ModSecurity engine, especially for request and response body processing, can be quite high.

An alternative method to triggering the WAF rules processing, if additional flexibility is required, is to use a Lua HTTP Transformation rule. The *HTTPControl.triggerWAF()* function can be used to trigger the WAF rules processing for a particular request. This function takes a single string parameter, which is used to specify the WAF phases for which rules processing should be invoked. An empty string is used to indicate that all phases should be enabled for processing. Please note that this function may only be called in the 'request' processing phase of the transformation rule and should not be used in the 'postazn' or 'response' phases.

An example transformation rule could be:

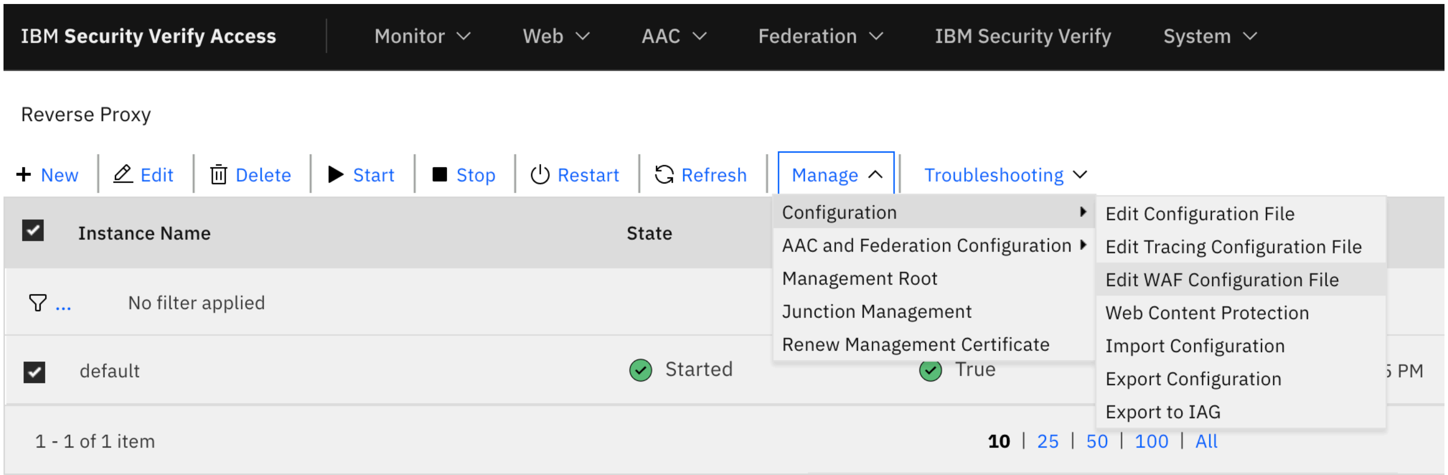
```
-- Only trigger the WAF processing if the 'ignore_waf' string is
-- missing from the request.
if not string.match(HTTPRequest.getURL(), "ignore_waf") then
    Control.triggerWAF("1,2")
end
```

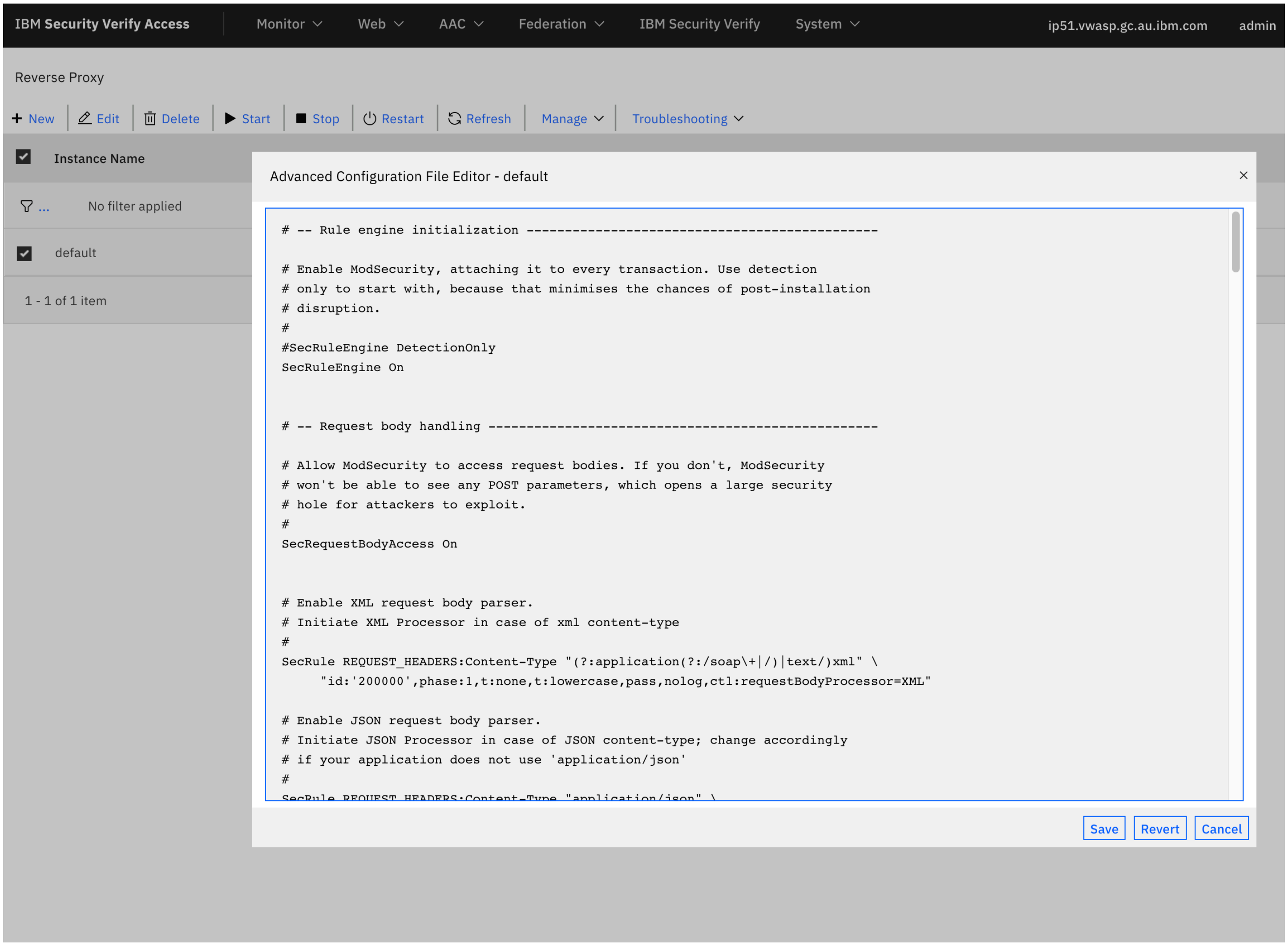
ModSecurity

A separate ModSecurity configuration is available for each reverse proxy instance. This file allows you to tune the ModSecurity engine for optimal performance (e.g. setting the maximum amount of data which will be parsed from a response body).

The configuration file itself contains comments which provide a description of each configuration entry, but full documentation for the configuration entries is available in the ModSecurity Reference Manual. Refer to the Limitations section of this document for configuration entries which are fixed and cannot be changed.

In order to modify the ModSecurity configuration for a specific instance, access the ‘Manage ⇒ Configuration ⇒ Edit WAF Configuration File’ menu item within the Reverse Proxy configuration screen of the LMI.





Advanced Configuration File Editor - default

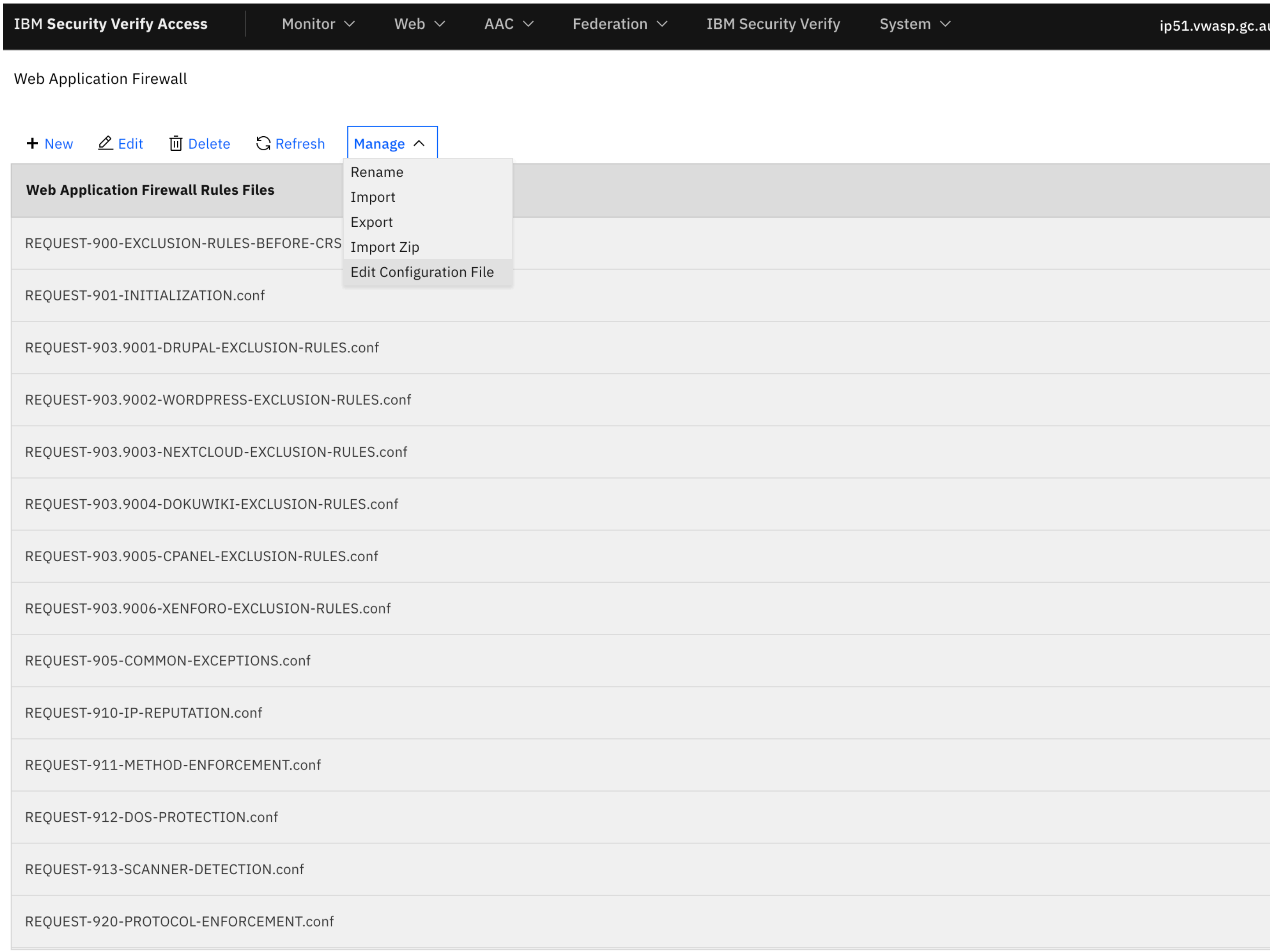
```
# -- Rule engine initialization ----------------------------------------------

# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
#SecRuleEngine DetectionOnly
SecRuleEngine On


# -- Request body handling ---------------------------------------------------

# Allow ModSecurity to access request bodies. If you don't, ModSecurity
# won't be able to see any POST parameters, which opens a large security
# hole for attackers to exploit.
#
SecRequestBodyAccess On


# Enable XML request body parser.
# Initiate XML Processor in case of xml content-type
#
SecRule REQUEST_HEADERS:Content-Type "(?:application(?:/soap\+|/)|text/)xml" \
     "id:'200000',phase:1,t:none,t:lowercase,pass,nolog,ctl:requestBodyProcessor=XML"

# Enable JSON request body parser.
# Initiate JSON Processor in case of JSON content-type; change accordingly
# if your application does not use 'application/json'
#
SecRule REQUEST_HEADERS:Content-Type "application/json" \
```

Note: By default, the ModSecurity engine is started in 'DetectionOnly' mode which means that it will log any issues which are encountered but will not take any action. In order to fully enable the ModSecurity engine the 'SecRuleEngine' configuration entry must be set to 'On'.

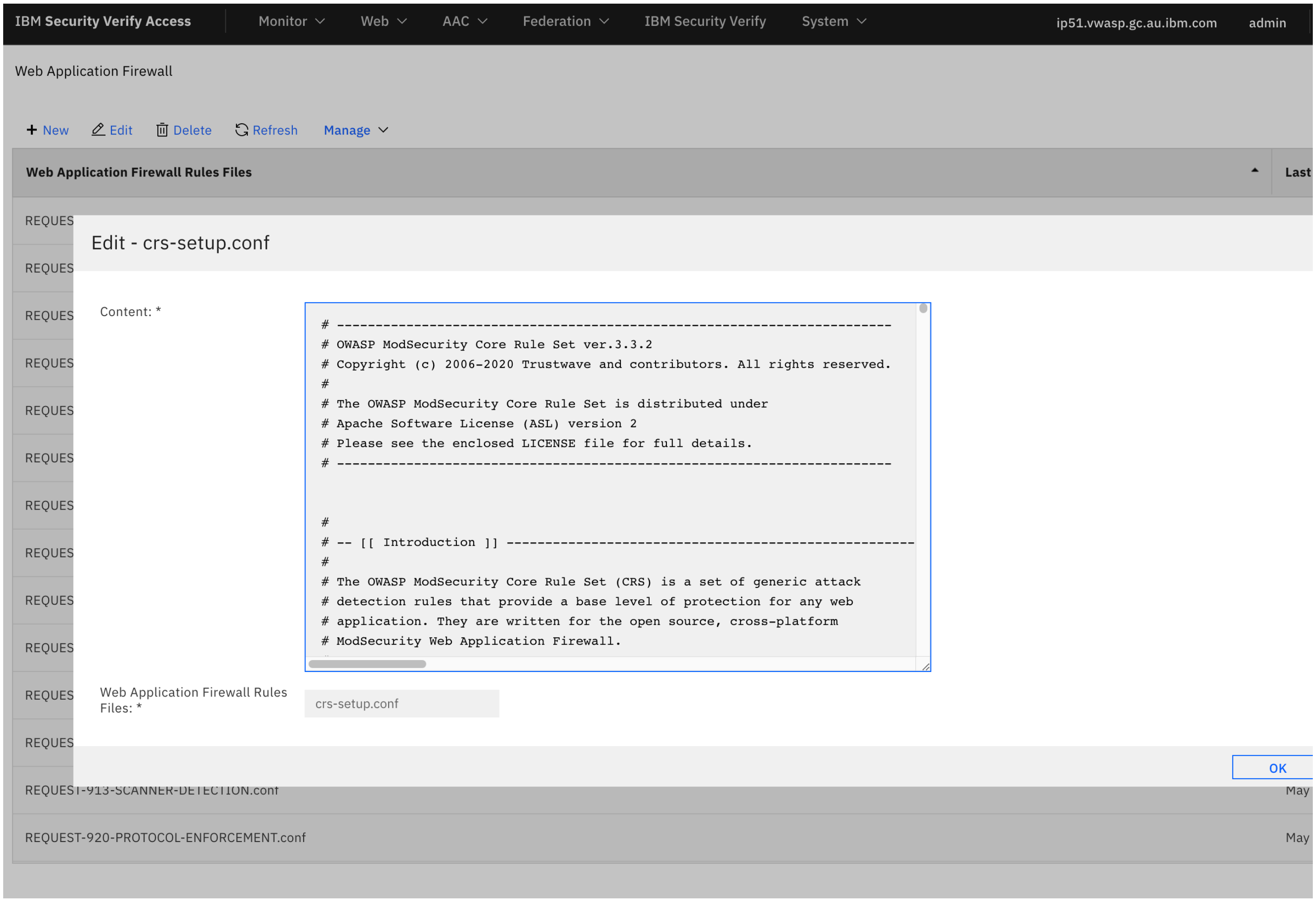
Core Rule Set

The OWASP ModSecurity Core Rule Set (CRS) is installed into the environment by default. A configuration file for the environment (i.e., this configuration is not ISVA specific) is available which allows you to fine tune the CRS processing. The file itself contains comments which provide a description of each configuration entry.

The configuration for the CRS is accessible via the 'Manage ⇒ Edit Configuration File' menu item within the Web Application Firewall configuration screen of the LMI.



Web Application Firewall Rules Files
REQUEST-900-EXCLUSION-RULES-BEFORE-CRS
REQUEST-901-INITIALIZATION.conf
REQUEST-903.9001-DRUPAL-EXCLUSION-RULES.conf
REQUEST-903.9002-WORDPRESS-EXCLUSION-RULES.conf
REQUEST-903.9003-NEXTCLOUD-EXCLUSION-RULES.conf
REQUEST-903.9004-DOKUWIKI-EXCLUSION-RULES.conf
REQUEST-903.9005-CPANEL-EXCLUSION-RULES.conf
REQUEST-903.9006-XENFORO-EXCLUSION-RULES.conf
REQUEST-905-COMMON-EXCEPTIONS.conf
REQUEST-910-IP-REPUTATION.conf
REQUEST-911-METHOD-ENFORCEMENT.conf
REQUEST-912-DOS-PROTECTION.conf
REQUEST-913-SCANNER-DETECTION.conf
REQUEST-920-PROTOCOL-ENFORCEMENT.conf



Edit - crs-setup.conf

Content: *

```
# ------------------------------------------------------------------------
# OWASP ModSecurity Core Rule Set ver.3.3.2
# Copyright (c) 2006-2020 Trustwave and contributors. All rights reserved.
#
# The OWASP ModSecurity Core Rule Set is distributed under
# Apache Software License (ASL) version 2
# Please see the enclosed LICENSE file for full details.
# ------------------------------------------------------------------------


#
# -- [[ Introduction ]] --------------------------------------------------------
#
# The OWASP ModSecurity Core Rule Set (CRS) is a set of generic attack
# detection rules that provide a base level of protection for any web
# application. They are written for the open source, cross-platform
# ModSecurity Web Application Firewall.
```

Web Application Firewall Rules Files: * crs-setup.conf

REQUEST-913-SCANNER-DETECTION.conf	May
REQUEST-920-PROTOCOL-ENFORCEMENT.conf	May

Rules

Rule Syntax

The ModSecurity Reference Manual contains a full description of the syntax of rules. However, the basic syntax of a rule is:

```
SecRule VARIABLES OPERATOR [ACTIONS]
```

Element	Description	Reference Documentation
VARIABLES	The variables which are used in the rule, for example the request URI.	https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual-%28v3.x%29#Variables
OPERATOR	This specifies a regular expression, pattern, or keyword to be checked in the variable(s). Operators begin with the '@' character.	https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual-%28v3.x%29#Operators
ACTIONS	This specifies what to do if the rule matches.	https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual-%28v3.x%29#Actions

An example rule would be:

```
SecRule REQUEST_LINE|REQUEST_HEADERS|REQUEST_HEADERS_NAMES \
  "@contains () {" \
  "id:420008,\
  phase:2,\
  t:none,\
  t:lowercase,\
  deny,\
  status:500,\
  log,\
  msg: Bash ENV Variable Injection Attack'"
```

Note: Refer to the Limitations section of this document for ModSecurity actions and operators which have been disabled.

Interventions

An 'intervention' is a ModSecurity term for a disruptive action which needs to be taken for a particular request. The possible interventions include:

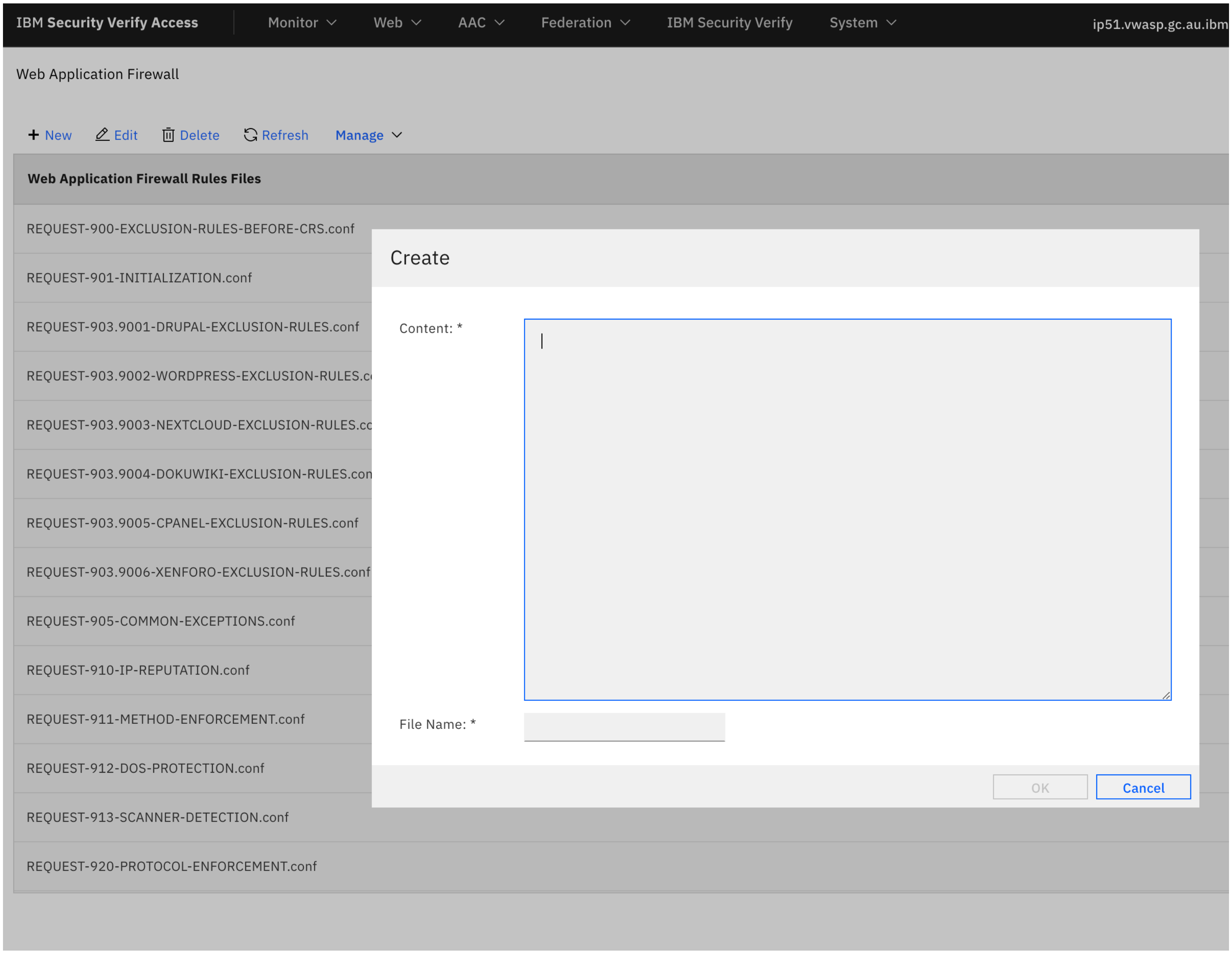
- Log a message.
- Redirect the client to a URL specified in the rule (aka HTTP 302).

- Send a non-200 status code to the client (e.g., '403 Forbidden'). When a non-200 status code is specified by the rule a corresponding WebSEAL generated page for the status code will be returned to the client.

Note: According to the ModSecurity documentation an action of 'drop' should cause the client connection to be closed immediately. However, this action is instead handled in the same fashion as the 'deny' action, where the request processing is interrupted, and an error page is returned to the client.

Creating a New Rule

In order to create a new rules file, the 'New' button should be selected in the 'Web Application Firewall' screen of the LMI. Add the rule to the 'Content' section and specify a unique name for the file (without path information) in the 'File Name' field. The file name should end with the '.conf' suffix otherwise it will not be recognised as a rules file.



Web Application Firewall Rules Files

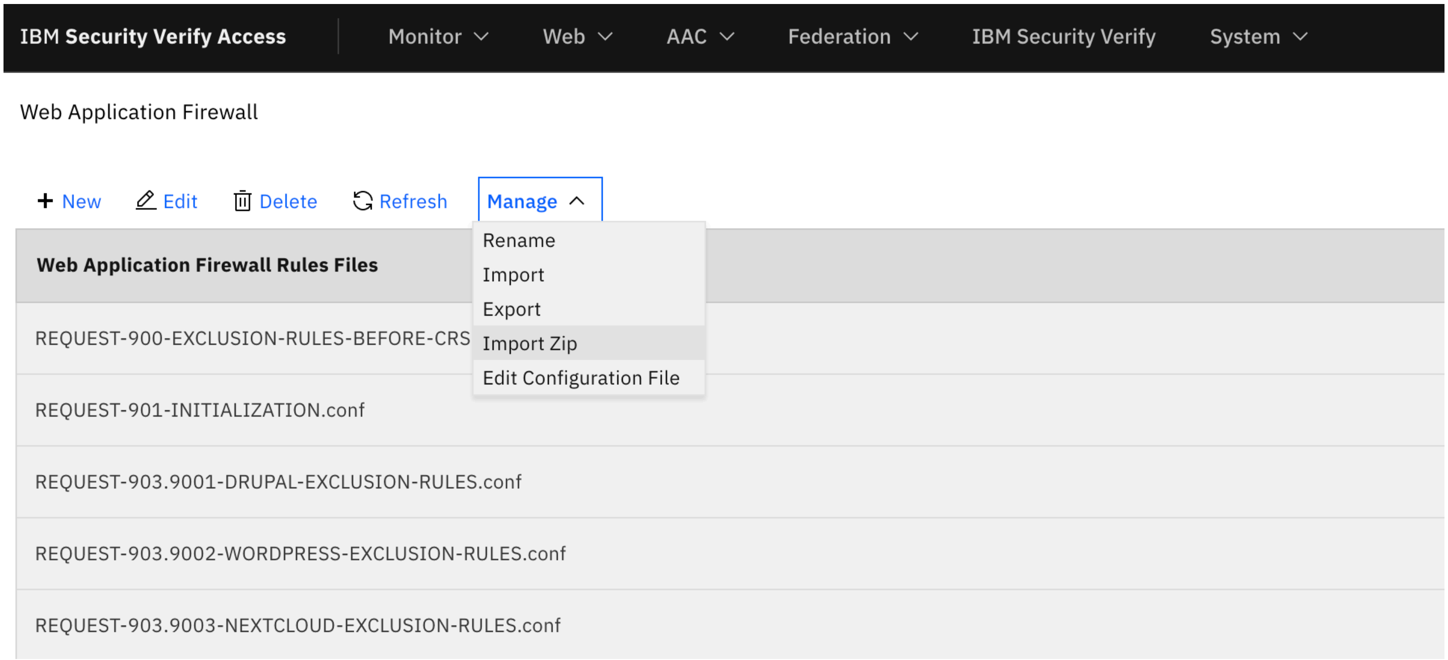
REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf
REQUEST-901-INITIALIZATION.conf
REQUEST-903.9001-DRUPAL-EXCLUSION-RULES.conf
REQUEST-903.9002-WORDPRESS-EXCLUSION-RULES.c
REQUEST-903.9003-NEXTCLOUD-EXCLUSION-RULES.cc
REQUEST-903.9004-DOKUWIKI-EXCLUSION-RULES.con
REQUEST-903.9005-CPANEL-EXCLUSION-RULES.conf
REQUEST-903.9006-XENFORO-EXCLUSION-RULES.conf
REQUEST-905-COMMON-EXCEPTIONS.conf
REQUEST-910-IP-REPUTATION.conf
REQUEST-911-METHOD-ENFORCEMENT.conf
REQUEST-912-DOS-PROTECTION.conf
REQUEST-913-SCANNER-DETECTION.conf
REQUEST-920-PROTOCOL-ENFORCEMENT.conf

Note: A single rules file can host multiple rules. It is not however recommended to directly modify any of the CRS rule files as this will make it more difficult to import future updates of the CRS distribution.

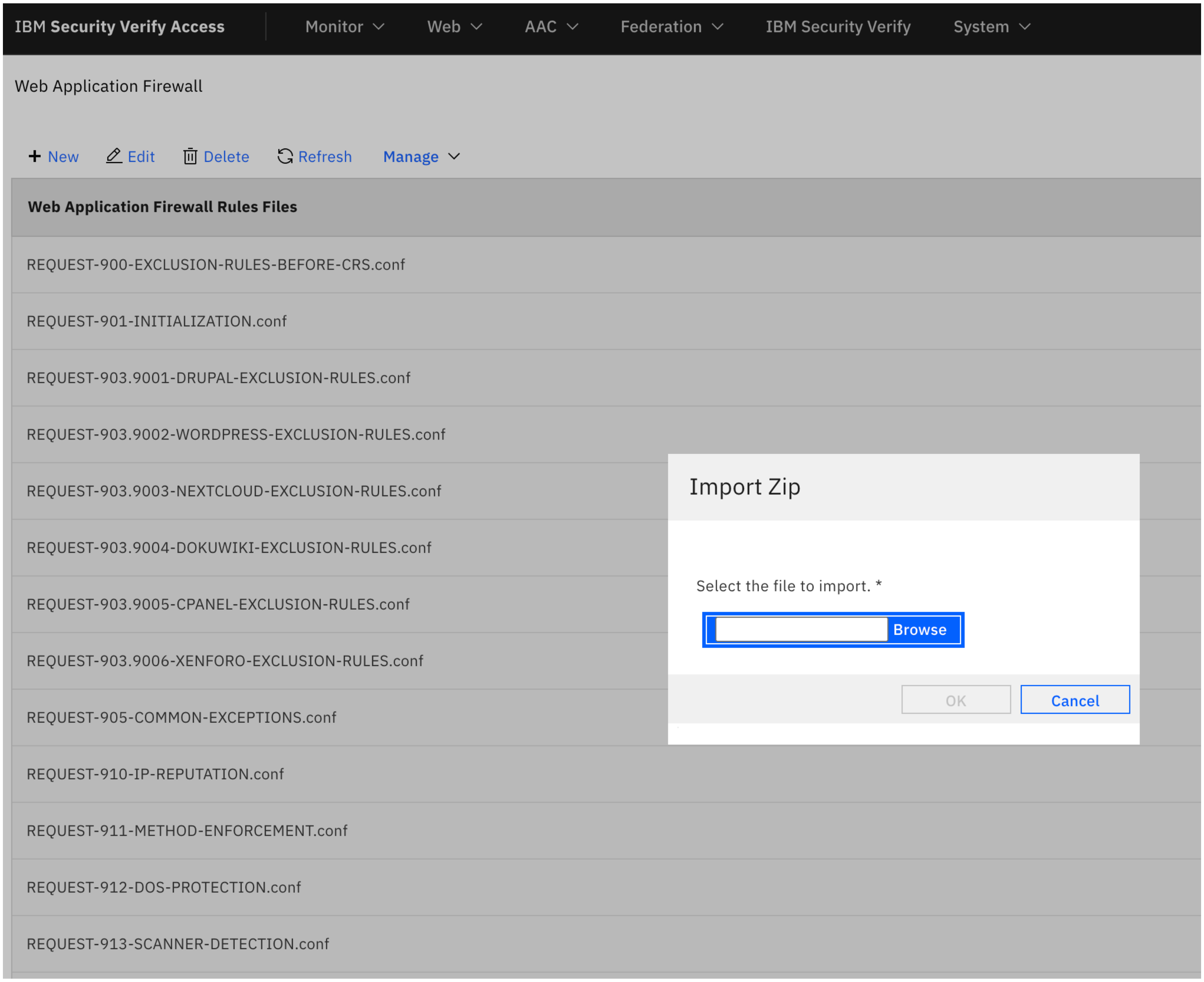
A single set of rule files is used for the entire environment and will apply to all WebSEAL instances (i.e., the rule files are not WebSEAL specific).

Updating the Core Rule Set

The Core Rule Set, available for download from GitHub, is updated periodically. IBM Security Verify Access currently embeds v3.3.2 of the CRS. If a new version of the CRS is released on GitHub this can be applied to the environment by downloading the release zip file from GitHub and then importing this zip file by selecting the ‘Manage ⇒ Import Zip’ menu item within the Web Application Firewall configuration screen of the LMI.



Web Application Firewall Rules Files
REQUEST-900-EXCLUSION-RULES-BEFORE-CRS
REQUEST-901-INITIALIZATION.conf
REQUEST-903.9001-DRUPAL-EXCLUSION-RULES.conf
REQUEST-903.9002-WORDPRESS-EXCLUSION-RULES.conf
REQUEST-903.9003-NEXTCLOUD-EXCLUSION-RULES.conf



Web Application Firewall Rules Files

REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf

REQUEST-901-INITIALIZATION.conf

REQUEST-903.9001-DRUPAL-EXCLUSION-RULES.conf

REQUEST-903.9002-WORDPRESS-EXCLUSION-RULES.conf

REQUEST-903.9003-NEXTCLOUD-EXCLUSION-RULES.conf

REQUEST-903.9004-DOKUWIKI-EXCLUSION-RULES.conf

REQUEST-903.9005-CPANEL-EXCLUSION-RULES.conf

REQUEST-903.9006-XENFORO-EXCLUSION-RULES.conf

REQUEST-905-COMMON-EXCEPTIONS.conf

REQUEST-910-IP-REPUTATION.conf

REQUEST-911-METHOD-ENFORCEMENT.conf

REQUEST-912-DOS-PROTECTION.conf

REQUEST-913-SCANNER-DETECTION.conf

Select the file to import. *

Logging

Events

Verify Access will log ModSecurity events and messages using the standard logging mechanism. This logging mechanism is configured using the ‘log-cfg’ configuration entry within the ‘[waf]’ configuration stanza of the reverse proxy configuration file.

The syntax of the ‘log-cfg’ configuration entry is:

```
agent [parameter=value],[parameter=value],…
```

The supported agents include:

- stdout
- stderr
- file
- rsyslog

A list of supported logging parameters can be found in the official documentation for IBM Security Verify Access: https://www.ibm.com/docs/en/sva/10.0.3?topic=logging-defining-logcfg-entries

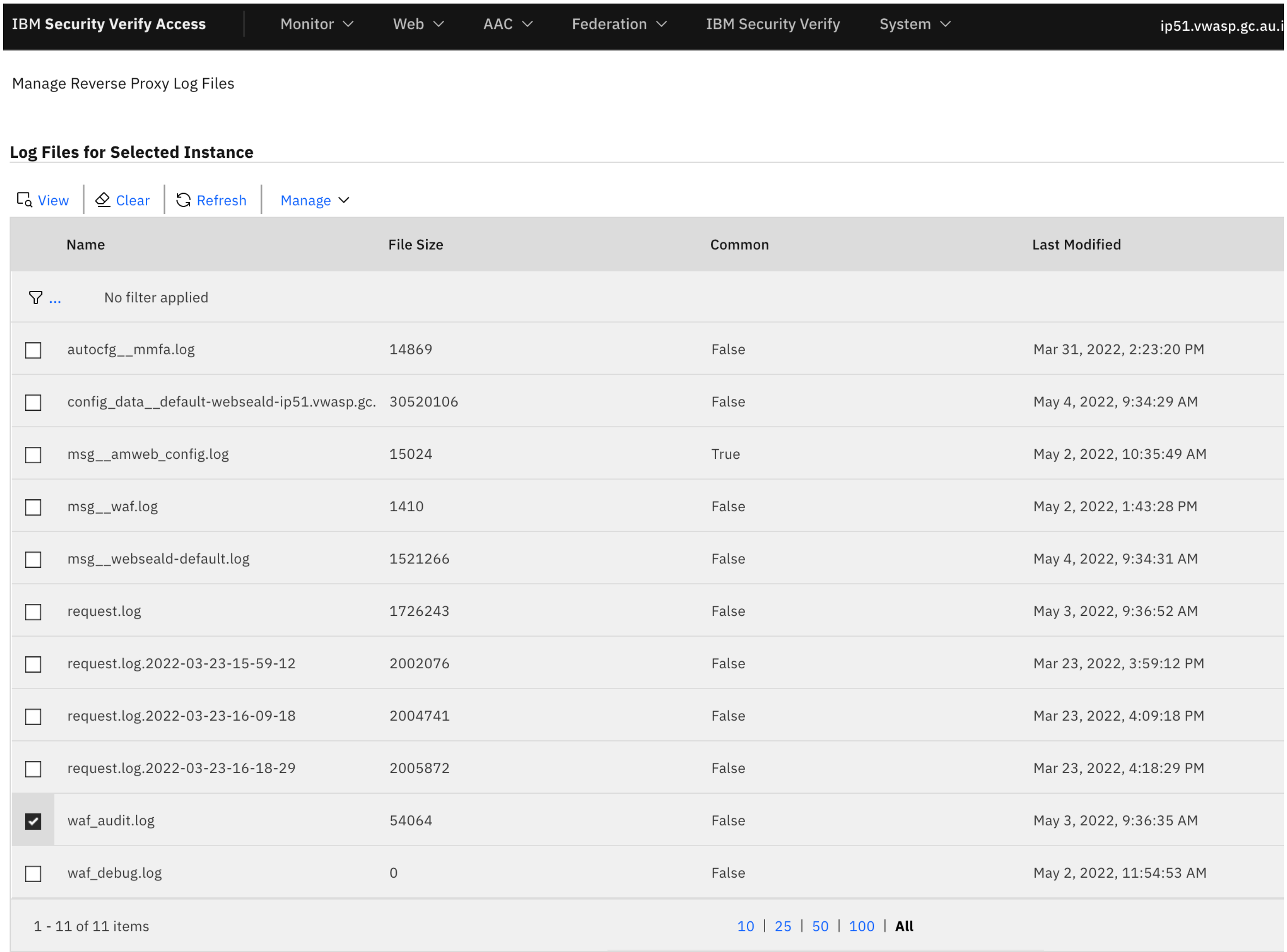
An example configuration entry, to send the events to a log file in the WebSEAL log directory, would be:

```
[waf]
log-cfg = file path=waf_events.log
```

Auditing

The ModSecurity engine has the ability to generate auditing records and save these auditing records to a file. The configuration of the auditing component is managed by the SecAuditXXX (e.g. SecAuditEngine) configuration entries within the ModSecurity configuration file (see the reference manual for details).

The name of the file which receives the auditing records is fixed to ‘waf_audit.log’, and the log file is generated in the reverse proxy log directory.



Manage Reverse Proxy Log Files

Log Files for Selected Instance

	Name	File Size	Common	Last Modified
□	autocfg__mmfa.log	14869	False	Mar 31, 2022, 2:23:20 PM
□	config_data__default-webseald-ip51.vwasp.gc.	30520106	False	May 4, 2022, 9:34:29 AM
□	msg__amweb_config.log	15024	True	May 2, 2022, 10:35:49 AM
□	msg__waf.log	1410	False	May 2, 2022, 1:43:28 PM
□	msg__webseald-default.log	1521266	False	May 4, 2022, 9:34:31 AM
□	request.log	1726243	False	May 3, 2022, 9:36:52 AM
□	request.log.2022-03-23-15-59-12	2002076	False	Mar 23, 2022, 3:59:12 PM
□	request.log.2022-03-23-16-09-18	2004741	False	Mar 23, 2022, 4:09:18 PM
□	request.log.2022-03-23-16-18-29	2005872	False	Mar 23, 2022, 4:18:29 PM
☑	waf_audit.log	54064	False	May 3, 2022, 9:36:35 AM
□	waf_debug.log	0	False	May 2, 2022, 11:54:53 AM

1 - 11 of 11 items

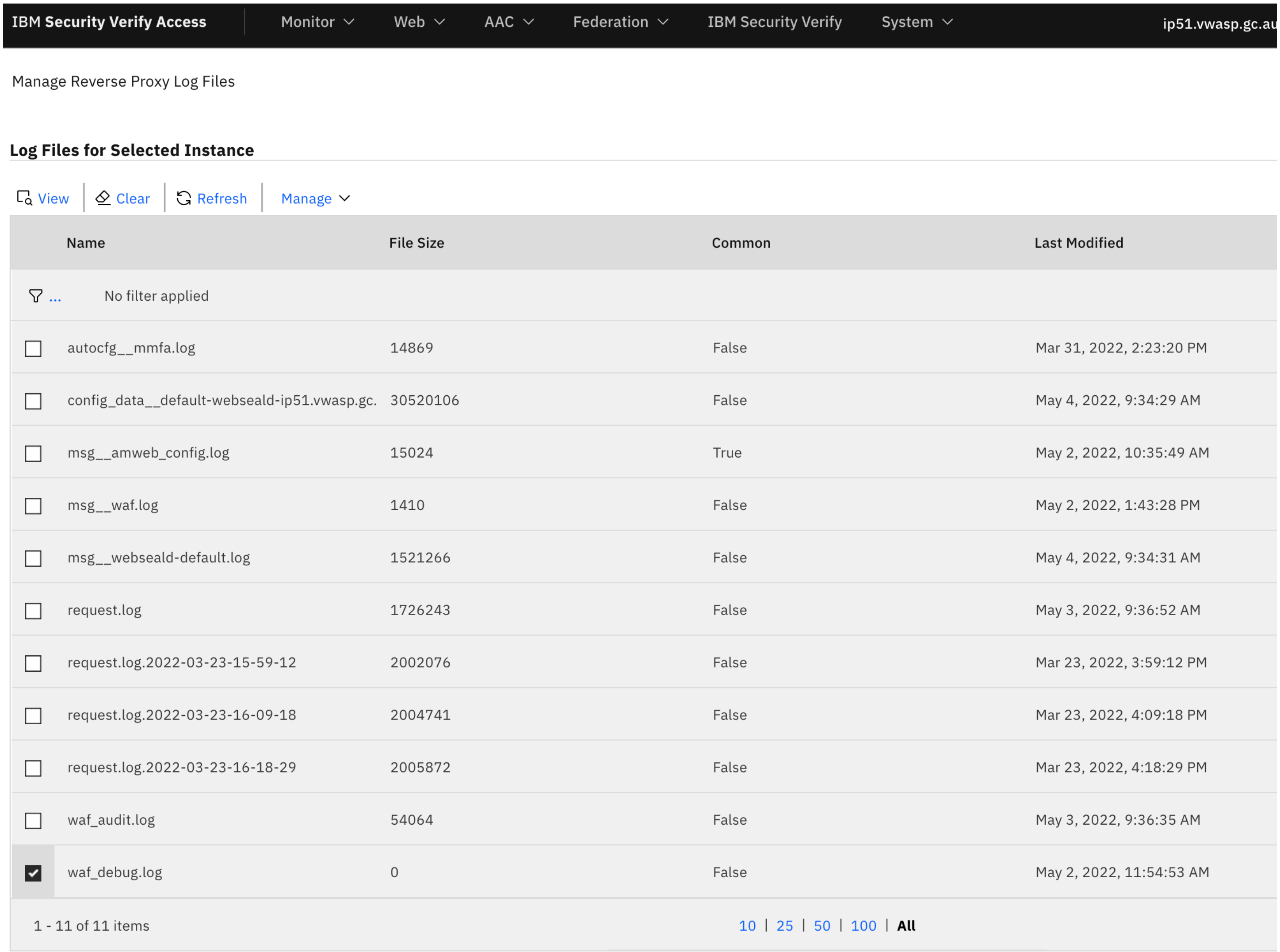
Docker Variables

In a containerized environment, when using the 'verify-access-wrp' image, the following environment variables will influence the WAF logging and auditing:

Environment Variable	Description
LOGGING_CONSOLE_FORMAT	If set to 'json' the audit and event logs will be output in JSON format.
LOG_TO_CONSOLE	This environment variable contains a space separated list of components which should be output to the console. The following component names are used to send the Web Application Firewall audit and event logs to the console: - waf.audit - waf.log

Trouble Shooting

The SecDebugLogLevel configuration entry within the ModSecurity configuration file can be used to enable debug logging within the ModSecurity engine. All debug log entries will be sent to the 'waf_debug.log' file within the WebSEAL logging directory.



Manage Reverse Proxy Log Files

Log Files for Selected Instance

	Name	File Size	Common	Last Modified
	No filter applied			
□	autocfg__mmfa.log	14869	False	Mar 31, 2022, 2:23:20 PM
□	config_data__default-webseald-ip51.vwasp.gc.	30520106	False	May 4, 2022, 9:34:29 AM
□	msg__amweb_config.log	15024	True	May 2, 2022, 10:35:49 AM
□	msg__waf.log	1410	False	May 2, 2022, 1:43:28 PM
□	msg__webseald-default.log	1521266	False	May 4, 2022, 9:34:31 AM
□	request.log	1726243	False	May 3, 2022, 9:36:52 AM
□	request.log.2022-03-23-15-59-12	2002076	False	Mar 23, 2022, 3:59:12 PM
□	request.log.2022-03-23-16-09-18	2004741	False	Mar 23, 2022, 4:09:18 PM
□	request.log.2022-03-23-16-18-29	2005872	False	Mar 23, 2022, 4:18:29 PM
□	waf_audit.log	54064	False	May 3, 2022, 9:36:35 AM
☑	waf_debug.log	0	False	May 2, 2022, 11:54:53 AM

1 - 11 of 11 items

IBM Security
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
February 2021
All Rights Reserved