ModSecurity Rules

Abstract

ModSecurity provides a flexible open source web application firewall (WAF) to the community. The code is actively maintained and supported across many platforms. While the syntax for writing rules can seem daunting, some basic powerful rules can be written with limited exposure.

Where to Acquire

Note: ModSecurity is already installed in the Security511 Linux VM.

ModSecurity - Details for installing ModSecurity on various platforms
OWASP ModSecurity CRS - ModSecurity Core Rule Set (CRS) provides open source rules for ModSecurity
ModSecurity Reference Manual

Examples/Use Case

Rules will need to be added to a .conf file that will be processed with ModSecurity starting up. On the SEC511 VM, this path is **`/etc/modsecurity`**. A new file for custom rules should generally be created rather than overwriting a Core Rule Set (CRS) provided file.

Basic Rule Structure:

```
SecRule VARIABLE "OPERATOR" "ACTION"
```

We will replace VARIABLE, OPERATOR, and ACTION with appropriate options provided by ModSecurity.

Example Rules

The default action we will use simply causes log information to be generated and a user defined message to be supplied. For example:

```
"log,auditlog,msg:'alert message'"
```

Detect an HTTP user agent containing the string ‘sqlmap’

```
SecRule REQUEST_HEADERS:User-Agent "@contains sqlmap" "log,auditlog,msg:'alert message'"
```

Detect an HTTP user agent NOT containing the string ‘sqlmap’

```
SecRule REQUEST_HEADERS:User-Agent "!@contains sqlmap" "log,auditlog,msg:'alert message'"
```

Match any HTTP user agents that begin with the string ‘Mozilla/5’

```
SecRule REQUEST_HEADERS:User-Agent "^Mozilla/5" "log,auditlog,msg:'alert message'"
```

Match an argument named ‘ip’ being set to an IPv4 address or any string of simply numbers and periods

```
SecRule ARGS:ip "^[\d.]+$" "log,auditlog,msg:'alert message'"
```

Detect the Host header not being equal to the string www.sec511.org

```
SecRule REQUEST_HEADERS:Host "!@streq www.sec511.org" "log,auditlog,msg:'alert message'"
```

Match the Host header being set to any IP address within 10.5.11.0-255

```
SecRule REQUEST_HEADERS:Host "@ipMatch 10.5.11.0/24" "log,auditlog,msg:'alert message'"
```

Detect the OPTIONS method being used

```
SecRule REQUEST_HEADERS:Method "@streq OPTIONS" "log,auditlog,msg:'alert message'"
```

Detect HTTP responses that lack a Content-Type header

```
SecRule &RESPONSE_HEADERS:Content-Type "@eq 0" "log,auditlog,msg:'alert message'"
```

Detect HTTP requests without a User-Agent

```
SecRule &REQUEST_HEADERS:User-Agent "@eq 0" "log,auditlog,msg:'alert message'"
```

Detect HTTP requests with more than one parameter named password

```
SecRule &ARGS:password "@gt 1" "log,auditlog,msg:'alert message'"
```

Detect HTTP requests without a Host header.

```
SecRule &REQUEST_HEADERS:Host "@eq 0" "log,auditlog,msg:'alert message'"
```

Detect HTTP requests without a Host header. Add the HTTP User Agent to the information provided in the error.log

```
SecRule &REQUEST_HEADERS:Host "@eq 0" "log,auditlog,msg:'alert message',logdata:%{REQUEST_HEADERS.User-A
```

Reading Logs

The default ModSecurity logging configuration will result in alerts being raised in the error.log file and the modsec_audit.log file. In the Security511 Linux VM both of these files are found in /var/log/apache2/

When a rule triggers the log action, ModSecurity will write one line to the error.log file providing some summary data by default. Additional information can be supplied in the error.log entry by using the 'tag' action and/or the 'logdata' action. See the Actions section above for details on those two items

error.log - example

Error log entry for CRS rule:

```
[Sun Dec 13 20:06:18 2015]
[error]
[client 127.0.0.1]
ModSecurity: Warning. Pattern match "\\\\bor\\\\b ?(?:\\\\d{1,10}|[\\\\'\\"][^=]{1,10}[\\\\'\\"]) ?[=<>]
[file "/etc/modsecurity/modsecurity_crs_41_sql_injection_attacks.conf"]
[line "427"]
[id "959071"]
[rev "2.2.0"]
[msg "SQL Injection Attack"]
[data "or 1="]
[severity "CRITICAL"]
[tag "WEB_ATTACK/SQL_INJECTION"]
[tag "WASCTC/WASC-19"]
[tag "OWASP_TOP_10/A1"]
[tag "OWASP_AppSensor/CIE1"]
[tag "PCI/6.5.2"]
[hostname "localhost"]
[uri "/scanners/pilots.php"]
[unique_id "Vm3S7X8AAQEAAB@2CLQAAAAD"]
```

Note: for easier reading, I put each field form the error.log on its own line, but normally this would be all on one line in the error.log file.

Some fields of note:
file - indicates the file that contains the rule that triggered
line - indicates the line number in that file
id - the rule id, if one is defined in the rule
unique_id - allows searching for correlating this alert with other entries for this connection in the error.log and modsec_audit.log

Much more complete details may also be written to the modsec_audit.log. What and whether anything is written will depend upon the rule and ModSecurity logging configuration. Unfortunately, the modsec_audit.log was not built with ease of parsing in mind.

Additional Info

ModSecurity Reference Manual
For additional details on ModSecurity Rule writing see https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual (http://cyber.gd/511_245).
For a tremendous resource on all things ModSecurity check Ivan Ristic's ModSecurity Handbook, https://www.feistyduck.com/books/modsecurity-handbook/ (http://cyber.gd/511_246).