

ModSecurity 2 Deployment

Installation

- ModSecurity can be deployed in **embedded mode**, when it is added directly into web server.
- Or it can function as a **network gateway**, combined with Apache (use 2.2.2 or better) configured to work as reverse proxy.

ModSecurity does not actually care about the mode of operation. It is only the Apache configuration that will be different. In fact, you can have a **hybrid installation too.**

Apache Reverse Proxy Installation (1)

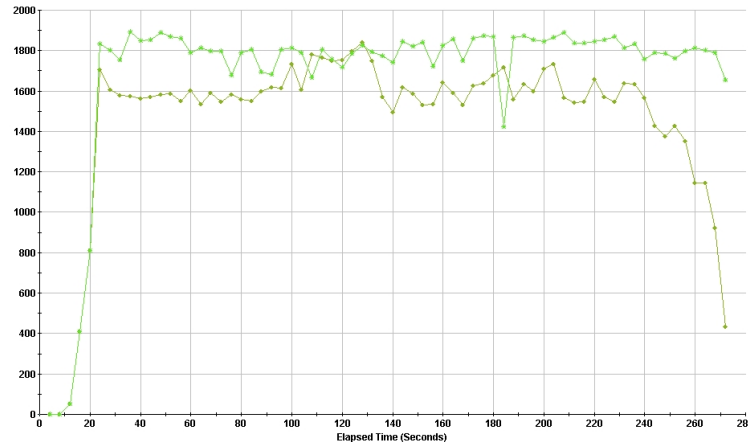
- **Enable the modules you need.**
- **Worker MPM** is slightly faster (on Linux) than Prefork.
- Core modules needed:

- ▶ mod_so
- ▶ mod_unique_id
- ▶ mod_proxy
- ▶ mod_proxy_http
- ▶ mod_proxy_balancer

- Third-party modules

- ▶ mod_proxy_html

http://apache.webthing.com/mod_proxy_html/



Apache Reverse Proxy Installation (2)

■ Useful modules:

- ▶ mod_rewrite
- ▶ mod_headers
- ▶ mod_setenvif
- ▶ mod_logio
- ▶ mod_expires

■ Enablers:

- ▶ mod_ssl
- ▶ mod_deflate
- ▶ mod_cache, mod_file_cache

Apache Reverse Proxy Installation (3)

- **Remove unused modules installed by default.**
- For example:
 - ▶ mod_include (although this one might be useful)
 - ▶ mod_autoindex
 - ▶ mod_asis
 - ▶ mod_cgi(d)
 - ▶ mod_negotiation
 - ▶ mod_userdir

Apache Reverse Proxy Installation (4)

■ Finally:

```
# ./configure
--prefix=/opt/apache
--with-mpm=worker --enable-so
--enable-unique-id
--enable-proxy --enable-proxy-http --enable-proxy-balancer
--enable-rewrite --enable-headers --enable-setenvif
--enable-logio
--enable-expires
--enable-ssl
--enable-deflate --enable-cache --enable-file-cache
--disable-autoindex --disable-asis --disable-cgi --disable-cgid
--disable-negotiation --disable-userdir
```

```
# make && make install
```

Apache Hardening (1)

- In the nutshell (refer to **Apache Security** for detailed coverage):
 1. Use own system account (e.g. `httpd`).
 2. Configure account limits.
 3. Configure process generation limits (necessary as processes are spawned by root):
 4. Put on a separate file system.
 5. Change server signature (`SecServerSignature ABC`).
 6. Disable TRACE (`TraceEnable Off`).
 7. Put in `jail` or restrict using `grsecurity` or `SELinux`.

Apache Hardening (2)

- Jailing Apache is very easy on Debian with help from [makejail](#).
- In some cases (e.g. reverse proxy) it is even easier to do with ModSecurity (but `libgcc_s.so.1` and `libxml2.so` must go into jail to, otherwise restart won't work):

```
# cd /opt
# mkdir -r ./jail/opt
# mv /opt/apache ./jail/opt
# ln -s /opt/jail/opt/apache
```

- Then add:

```
LoadFile /lib/libgcc_s.so.1
SecChrootDir /opt/jail
```

- Check with `lsuf`:

```
# lsuf | grep httpd | grep DIR
httpd  4440  root cwd  DIR  8,1  1024  319542 /opt/jail
httpd  4440  root rtd  DIR  8,1  1024  319542 /opt/jail
```


Apache Hardening (3)

■ Memory consumption limits:

| | |
|-----------------------|-------|
| LimitRequestBody | 64000 |
| LimitRequestFields | 32 |
| LimitRequestFieldSize | 8000 |
| LimitRequestLine | 4000 |
| LimitXMLRequestBody | 32000 |

■ Process creation limits (worker MPM):

| | |
|---------------------|------|
| StartServers | 2 |
| MaxClients | 150 |
| MinSpareThreads | 25 |
| MaxSpareThreads | 75 |
| ThreadsPerChild | 25 |
| MaxRequestsPerChild | 1000 |

Configuring Proxy

■ Example configuration:

```
ServerName www.example.com
DocumentRoot /opt/apache/htdocs/
ProxyRequests Off
ProxyPass /_error_documents_/ !
ProxyPass / http://192.168.2.101/
ProxyPassReverse / http://192.168.2.101/
ProxyPassReverseCookieDomain www.example.com 192.168.2.101
ErrorDocument 403 /_error_documents_/403.shtml
ErrorDocument 502 /_error_documents_/502.shtml
ErrorDocument 503 /_error_documents_/503.shtml
ErrorDocument 504 /_error_documents_/504.shtml
```

Unique Transaction References

- It is often useful to give a unique transaction reference to the user.
- Makes it easy to nail down false positives.

```
<Location /_error_documents_>  
  Options +IncludesNoExec  
</Location>
```

- Then use (in error document):

```
<!--#echo var="UNIQUE_ID" -->
```

ModSecurity Installation (1)

- Assuming you already have Apache (and libxml2) installed:
 1. Edit `Makefile` to let it know where Apache lives
 2. Do `make && make install`
 3. Stop web server
 4. Edit `httpd.conf` to load libxml2 (`LoadFile`) and `mod_security2.so` (`LoadModule`)
 5. Add minimal configuration to test
 - `SecRuleEngine On`
 - `SecRule REQUEST_URI attack`

ModSecurity Installation (2)

- Verify ModSecurity is operational:
 1. Submit request with `attack` in the URI.
 2. You should get `403 Forbidden` in response.
 3. Observe the error log for the message.

```
[Sun Jun 04 10:44:34 2006] [error] [client 192.168.2.11]  
ModSecurity: Access denied with code 403 (phase 2). Pattern  
match "attack" at REQUEST_URI.  
[hostname "192.168.2.111"] [uri "/attack"]  
[unique_id "3YvyJ38AAAEACL@CiMAAAAA"]
```

4. Removing the offending word should get you a normal page in response.

ModSecurity Configuration

- Complete ModSecurity configuration consists of four groups of configuration directives:
 - 1. General configuration options**
 - 2. Debug logging options**
 - 3. Audit logging options**
 - 4. Rules**

Default ModSecurity Configuration (1)

Basic configuration options

SecRuleEngine On

SecRequestBodyAccess On

SecResponseBodyAccess On

Handling of uploaded files

SecUploadDir /opt/apache-frontent/tmp/

SecUploadKeepFiles Off

Default ModSecurity Configuration (2)

Debug log

SecDebugLog logs/modsec_debug.log

SecDebugLogLevel 0

Serial audit log

SecAuditEngine RelevantOnly

SecAuditLogRelevantStatus ^5

SecAuditLogParts ABIFHZ

SecAuditLogType Serial

SecAuditLog logs/modsec_audit.log

ModSecurity Limits

Maximum request body size we will accept for buffering

SecRequestBodyLimit 131072

Store up to 128 KB in memory

SecRequestBodyInMemoryLimit 131072

Buffer response bodies of up to 512 KB in length

SecResponseBodyLimit 524288

Miscellaneous

- You won't normally use these but they are needed in certain rare circumstances.
- Choosing the Cookie format (**v0** used by default):
`SecCookieFormat 0`
- Choosing argument separator (**&** used by default):
`SecArgumentSeparator ;`

Impact of ModSecurity (1)

- There are three aspects to consider:

1. Request and response buffering

- The only way to insure security.
- Can break some applications (e.g. large file uploads).

2. Memory consumption

- Request and response bodies are sometimes stored in memory.
- Transaction parts need to be normalised.
- Not a problem on a reverse proxy.
- But can be an issue on a server that is already overloaded.

3. CPU consumption

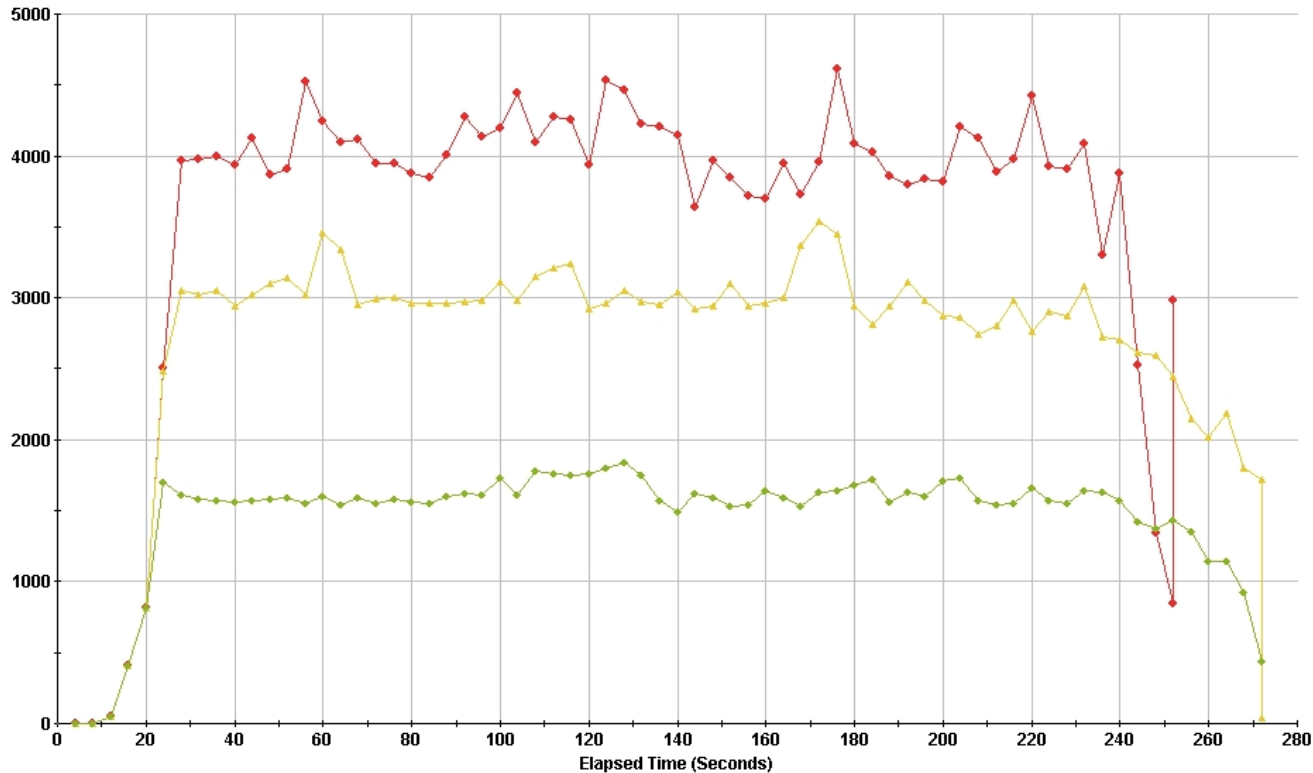
- Regular expressions are CPU-bound.
- Not a problem on a reverse proxy.
- Can be an issue on a server that is already overloaded.

Impact of ModSecurity (2)

- In embedded mode the impact is normally very small because web server needs far more time to process each request.
 - ▶ Processing time of **under 1 millisecond per request**.
- In reverse proxy mode it typically **halves the maximum number of requests per second** the server can push through.
 - ▶ It is not as bad as it sounds.
 - ▶ Still you need to perform basic benchmarking to determine the limits of the hardware on which the proxy is running.
 - ▶ Modest hardware can push **1500 requests per second** with **under 1 millisecond latency**.

Impact of ModSecurity (3)

- **Red** - without ModSecurity; **Yellow** - with 25 rules; **Green** - with 150 rules (Certified ModSecurity Rules).



Impact of ModSecurity (4)

- Tested with **Sun v20Z**:
 - ▶ Single AMD Opteron 244 (1.8 GHz)
 - ▶ 1 GB RAM
 - ▶ Gigabit Network
 - ▶ RedHat Enterprise Linux 4 (2.6.9-11 EL)



ModSecurity Console (1)

- Log & alert centralisation solution, can capture alerts or entire traffic streams.
- Daemon with a GUI (web application).
- Single package (comes with its own web server and database).
- Runs on all platforms that support Java 1.4 or better.
- Will evolve into **Management Console.**




ModSecurity Console (2)

- Alert management.
- Sensor statistics.
- HTML and PDF reports, on demand or scheduled.
- Automatic report distribution via email.
- Can implement a data retention policy.

HTTP Transaction: 2906 (2006-06-04 20:34.39)

Alert Messages

| ID/Rev | Severity | Message |
|--------|-----------------------------------------------------------------------------------------------|---------------------------------------------|
| 1 - |  EMERG (0) | Relevant response status Warning. Pattern m |

Request Details

```
GET /tsn/downloads/favicon.ico HTTP/1.1
Host: www.thinkingstone.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.1) G
ebian-1.8.0.1-11) Galeon/2.0.1 (Debian package 2.0.1-3)
Accept: text/xml,application/xml,application/xhtml+xml,text/html;
;q=0.8,image/png,*/*;q=0.5
Accept-Language: de,de-de;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-15,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Cookie: TSN_SESSIONID=3acd14cc53b45f5c6bc2384c243ac050
```


THE END!

Questions?