Locking down your Apache Web Server with mod_security
TODAYS ROADMAP

• Introduction

• Why should I lock down my server?

• How do I lock down my server?

• How can I use mod_security?

• Q&A
Introduction

• **Background information**
  – Who is Hans Kind

• **The 4Cs**
  – Check

• **Disclaimer**
  – Know what you are doing
The 4Cs

- Check
- Double Check
- Re-Check
- Check again
The 4Cs

The 4 C’s, are introduced here to make you realize that you need to be aware of your specific and unique situation.

Never copy or introduce new methods without knowing there ins and outs.

Never created new rules without testing them on a test server or better a staging environment.

Allows ask yourself this Question:

“Is this solution for me?”

If you are unsure on the answer, check the forums, use a search engine, read the manual, hire a professional or leave the solution alone.
Disclaimer

All examples used in this presentation are just that. Be careful to use them on a production server. Use them on your own risk.

Know what you are doing……..

The 4 Cs
Why should I lock down my server?

• Sooner or later you will be attacked

• You are interesting enough to be targeted

• The opportunity is there

• According to http://www.dshield.org/ the average survival time of an insecure system is 16 minutes.
Why should I lock down my server?

Is there a “hackers” profile?
Why should I lock down my server?

Profile of a hacker.

Hackers generally like to think of themselves as an elite group of information seekers who are adept at exploring computer systems and networks. Although hacking into network computer systems is illegal, hackers believe it is ethically acceptable as long as a hacker does not commit theft, vandalism or breach any confidentiality -- the so-called hacker code of ethics. In fact, many hackers believe it is their responsibility to seek out security holes in computer networks so that systems administrators may fix them.
Why should I lock down my server?

Profile of a cracker.

But not all hackers follow a code of ethics. Those who break into computer systems with malicious intent are known in the hacking world as crackers. The word itself was devised by hackers who wanted to differentiate themselves from crackers. Whereas hackers possess a great deal of knowledge of computers and generally write their own hacking programs, crackers tend to be young and unskilled (script kiddies).
Why should I lock down my server?

- (Young) Script kiddies
- (Former) Staff
- Competition
- Organized crime
- .........................
How do I lock down my server?

Reactive security

Much of our time is spent on reacting to security vulnerabilities:

• Patching hosts against new vulnerabilities
• Reactive firewalling
• Incident response
How do I lock down my server?

Reactive security

• Patching hosts against new vulnerabilities
  – We apply a huge number of patches each year to operating systems, applications and networking hardware
  – Even if we are diligent about patching, it’s not good enough, we still have a “window of vulnerability”
How do I lock down my server?

Reactive security

• Lifecycle of a vulnerability
  – Someone finds a bug in a program
  – Someone discovers that the bug is a security vulnerability
  – Someone writes an exploit to compromise machines with this vulnerability
How do I lock down my server?

Reactive security

- **Window of vulnerability**
  - The time frame between the moment a new exploit is published, and the moment a fix is available is *THE window of vulnerability*
How do I lock down my server?

Reactive security

• Patching isn’t enough
  – The “window of vulnerability” provides the time a attacker needs to compromise our systems
  – It’s not enough to hope we are save. We should be working proactively to reduce our probability of being attacked
How do I lock down my server?

Proactive security

• **Hardening**: Configure your hosts and your network to decrease their odds of being successfully hacked

• Intelligently choose policies ahead of time
How do I lock down my server?

Proactive security

• What is hardening:
  – Hardening is the process of configuring a system for increased security
  – It *does* involve deactivating unnecessary programs and auditing the configurations of those that remain
  – It *does* involve auditing the permissions and/or file access control lists and considering whether permissions are appropriate or too lax
How do I lock down my server?

Proactive security

- **Hardening**
  - Install patches as necessary
  - Disable telnet
  - Remove regular user and world access from c compilers and wget
  - Secure host.conf
  - Install chkrootkit and RootKit Hunter, and set up in cron
  - Install logwatch
  - Disable direct root login
  - Only allow certain IP’s access to SSH
  - Install a firewall
  - Recreate the /tmp directory on its own partition
  - Remove unnecessary packages
  - Turn off server signatures
  - Change your passwords on a regular basis
  - Daily routine
How do I lock down my server?

Proactive security

• Hardening

Check out:

**Increasing Security**

• FreeBSD System Hardening Documents
• Microsoft Windows 200x System Hardening Documents
• RedHat Linux System Hardening Documents
How do I lock down my server?

What’s available

• **Portsentry**
• **The Port Scan Attack Detector (PSAD)**
• **chkrootkit & Rootkit Hunter**
• **Tripwire**
• **Logcheck**
• **Logwatch**
• **Bastille Linux**
• **Snort**
• **Mod_security**
How do I lock down my server?

Portsentry

The Sentry tools provide host-level security services for the Unix platform. PortSentry protect against portscans, automate log file auditing, and detect suspicious login activity on a continuous basis.

The reason why we (still) use portsentry:

1. Ease of install
2. Works with ipchains/iptables
3. Works with the hosts deny file
4. Definable port numbers (TCP & UDP)
5. Incidents viewable in LogWatch
How do I lock down my server?

Portsentry

Logwatch entry

--------------------- Kernel Begin ------------------
Denied 471 packets on interface eth0
From 65.54.190.7 - 212 packets to 21 tcp ports
From 194.70.179.162 - 6 packets to tcp(41827)
From 200.96.99.194 - 84 packets to tcp(80)
From 200.96.255.194 - 168 packets to udp(52255,53656)
From 82.53.91.6 - 1 packet to tcp(23)
--------------------- Kernel End ------------------
--------------------- PortSentry Begin ---------------------
Warning: Blocked route from/to 65.54.190.7
Warning: Blocked route from/to host6-91.pool8253.interbusiness.it (82.53.91.6)
--------------------- PortSentry End ---------------------
How do I lock down my server?

PSAD

PSAD is a collection of three lightweight system daemons (two main daemons and one helper daemon) that run on Linux machines and analyze iptables log messages to detect port scans and other suspicious traffic.
How do I lock down my server?

PSAD

PSAD incorporates many signatures from the snort intrusion detection system to detect probes for various backdoor programs (e.g. EvilFTP, GirlFriend, SubSeven), DDoS tools (mstream, shaft), and advanced port scans (fin, null, Xmas) which are easily leveraged against a machine via nmap. When combined with fwsnort, psad is capable of detecting approximately 75% of all snort rules, including those that inspect the application portion of ip packets. In addition, psad makes use of packet ttl, tos, ip id, and tcp window sizes to passively fingerprint remote operating systems from which scans originate.
How do I lock down my server?

Chkrootkit & Rootkit Hunter

Chkrootkit:
Is a shell script that checks system binaries for rootkit modification.

Rootkit Hunter:
Is a scanning tool to ensure you for about 99.9% you're clean of nasty tools. This tool scans for rootkits, backdoors and local exploits
How do I lock down my server?

Tripwire

Tripwire is a tool that checks to see what has changed on your system. The program monitors key attributes of files that should not change, including binary signature, size, expected change of size, etc. The hard part is doing it the right way, balancing security, maintenance, and functionality.
How do I lock down my server?

Logcheck

Logcheck is software package that is designed to automatically run and check system log files for security violations and unusual activity. Logcheck utilizes a program called logtail that remembers the last position it read from in a log file and uses this position on subsequent runs to process new information. All source code is available for review and the implementation was kept simple to avoid problems. This package is a clone of the frequentcheck.sh script from the Trusted Information Systems Gauntlet(tm) firewall package.
How do I lock down my server?

Logwatch

LogWatch is a customizable, pluggable log-monitoring system. It will go through your logs for a given period of time and make a report in the areas that you wish with the detail that you wish. Easy to use - works right out of the package on almost all systems.
How do I lock down my server?

Bastille Linux

The Bastille Hardening System attempts to "harden" or "tighten" Unix operating systems. It currently supports the Red Hat, Debian, Mandrake, SuSE and TurboLinux Linux distributions along with HP-UX and Mac OS X. We attempt to provide the most secure, yet usable, system possible. The project is run by Jon Lasser, Lead Coordinator and Jay Beale, Lead Developer, and involves a number of developers, beta-testers and concept-creators. Bastille Linux was developed with several major goals:
How do I lock down my server?

Bastille Linux

Comprehensiveness

Bastille Linux draws from every available major reputable source on Linux Security. The initial development integrated Jay Beale’s existing O/S hardening experience for Solaris and Linux with most major points from the SANS' Securing Linux Step by Step, Kurt Seifried's Linux Administrator's Security Guide, and countless other sources.
How do I lock down my server?

Bastille Linux

Instructiveness

Bastille Linux has been designed to educate the installing administrator about the security issues involved in each of the script's tasks, thereby securing both the box and the administrator. Each step is optional and contains a description of the security issues involved.
How do I lock down my server?

Bastille Linux

Community

Once the initial development was near complete, we brought the effort to the developers of the Bastille Discussion mailing list. Further, we began soliciting outside suggestions and testing. The script was GPL'd promptly and the Specification shared.
How do I lock down my server?
How do I lock down my server?

Snort

Snort is an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more.
How do I lock down my server?

Snort

Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plugin architecture. Snort has a real-time alerting capability as well, incorporating alerting mechanisms for syslog, a user specified file, a UNIX socket, or WinPopup messages to Windows clients using Samba's smbclient.
How do I lock down my server?

Snort

Snort has three primary uses.

1. It can be used as a straight packet sniffer like tcpdump
2. a packet logger (useful for network traffic debugging, etc)
3. as a full blown network intrusion detection system.
How do I lock down my server?
How can I use mod_security?

ModSecurity is an open source intrusion detection and prevention engine for web applications. Operating as an Apache Web server module, the purpose of ModSecurity is to increase web application security, protecting web applications from known and unknown attacks.
How can I use mod_security?

Introducing mod_security

Running public web applications may seem like playing Russian roulette. Although achieving robust security on the Web is possible in theory, there's always a weak link in real life. It only takes one slip of the code to allow attackers unrestricted access to your data. If you have a public web application of modest complexity running, chances are good that is has some kind of security problem.
How can I use mod_security?

Introducing mod_security

Let’s look at the following example URL:

http://www.webapp.com/login.php?username=admin';DROP%20TABLE%20users--

If your application is vulnerable to SQL injection, invoking the URL above may very well delete all user data from your application.

Did you make regular database backups?
How can I use mod_security?

Introducing mod_security

Fortunately, the mod_security Apache module can protect you from this and other forms of web attacks.

To prevent the "drop table" SQL injection attack with mod_security, add the following to your Apache configuration:

SecFilter "drop[:space:]table"
How can I use mod_security?

Installing mod_security

http://www.modsecurity.org/download/

When installing from source you have two choices: to install the module into the web server itself, or to compile mod_security into a dynamic shared object (DSO).
How can I use mod_security?

Installing mod_security

Installing as DSO is easier, and the procedure is the same for both Apache branches:

1. Unpack mod_security source code
2. Compile the module into the web server:
   - /apachehome/bin/apxs -cia mod_security.c
3. Stop and start the server
   - /apachehome/bin/apachectl stop
   - /apachehome/bin/apachectl start
How can I use mod_security?

Installing mod_security

Static installation with Apache 1.x

When a module is compiled statically, it gets embedded into the body of the web server. This method results in a slightly faster executable but the compilation method is a bit more complicated.
How can I use mod_security?

Installing mod_security

Static installation with Apache 1.x

1. Copy the file mod_security.c to /src/modules/extra
2. Configure Apache distribution with two additional configuration options
   --activate-module=src/modules/extra/mod_security
   --enable-module=security
3. Compile and install as usual
How can I use mod_security?

Configuring mod_security

Assuming you used the DSO method to install mod_security, the procedure adds the following 2 lines to your httpd.conf file:

- LoadModule security_module libexec/mod_security.so

- AddModule mod_security.c

It does not activate mod_security by default.
How can I use mod_security?

Configuring mod_security

Mod_security configuration directives need to be added to your configuration file (httpd.conf) manually.

<IfModule mod_security.c>

    # Comment
    Directive [option] [action]

</IfModule>
How can I use mod_security?

Configuring mod_security

Mod_security comes with a sample httpd.conf file. We will follow this example, and show you the issues we ran into when we first started to use mod_security.

Yes, we did test mod_security on a test server, but once we completed the testing, and moved mod_security to our production server(s), we still ran into some unexpected issues.
How can I use mod_security?

Configuring mod_security

Turning filtering on and off

Filtering engine is turned off by default. To use it, you need to turn it on:

```<IfModule mod_security.c>
   # Turn the filtering engine On or Off
   SecFilterEngine On
</IfModule>```
How can I use mod_security?

Configuring mod_security

Turning filtering on and off

Supported parameter values for this parameter are:

– On – analyze every request
– Off – do nothing
– DynamicOnly
How can I use mod_security?

Configuring mod_security

Turning filtering on and off

DynamicOnly – analyze only requests generated dynamically at runtime. Using this option will prevent your web server from using precious CPU cycles on checking access to static files.
How can I use mod_security?

Configuring mod_security

Getting mod_security to log dynamic requests could require a little bit of work. In theory, a response to a request is generated by a handler, and if there is a handler attached to a request it can be considered to be of a dynamic nature. In practice, however, Apache can be configured to serve dynamic pages without a handler (it will choose the module based on the mime type).
How can I use mod_security?

Configuring mod_security

This will happen, for example, if you configure PHP as instructed in the main distribution:

- AddType application/x-httpd-php .php

While this works, it won't serve our purpose here. However, if you replace the above line with the following:

- AddHandler application/x-httpd-php .php

PHP will work just as well and audit logger will be able to do its job.
How can I use mod_security?

Logging mod_security

Standard Apache logging will not help much if you need to trace back steps of a particular user or an attacker. The problem is that only a very small subset of each request is written to a log file. This problem can be remedied with the audit logging feature of mod_security.
How can I use mod_security?

Logging mod_security

Use the following 2 directives:

SecAuditEngine On
SecAuditLog /var/log/modsecure/audit_log
How can I use mod_security?

Example entry for SecFilterSelective "ARG_p1" 666

========================================
Request: 192.168.0.2 - - [[18/May/2003:11:20:43 +0100]] "GET /cgi-bin/printenv?p1=666 HTTP/1.0" 406 822
Handler: cgi-script
========================================
GET /cgi-bin/printenv?p1=666 HTTP/1.0
Host: wkx.dyndns.org:8080
User-Agent: mod_security regression test utility
Connection: Close
mod_security-message: Access denied with code 406. Pattern match "666" at ARG_SELECTIVE.
mod_security-action: 406
HTTP/1.0 406 Not Acceptable

15 November 2004
How can I use mod_security?

Logging mod_security

The SecAuditEngine parameter accepts one of four values:

1. On – log all requests
2. Off – do not log requests at all
3. RelevantOnly – only log relevant requests. Relevant requests are those requests that caused a filter match.
4. DynamicOrRelevant – log dynamically generated or relevant requests. A request is considered dynamic if its handler is not null.
How can I use mod_security?

URL Encoding validation

Special characters need to be encoded before they can be transmitted in the URL. Any character can be replaced using the three character combination %XY, where XY is an hexadecimal character code. Hexadecimal numbers only allow letters A to F, but attackers sometimes use other letters in order to trick the decoding algorithm. Mod_security checks all supplied encodings in order to verify they are valid.
How can I use mod_security?

URL Encoding validation

Turn URL encoding validation on with the following line:

SecFilterCheckURLEncoding On
How can I use mod_security?

Unicode encoding validation

Unicode encoding validation is disabled by default. You should turn it on if your application or the underlying operating system accept/understand Unicode.

You can turn Unicode encoding on with the following line:

```
SecFilterCheckUnicodeEncoding On
```
How can I use mod_security?

Unicode encoding validation

This feature will assume UTF-8 encoding and check for three types of errors:

1. Not enough bytes. UTF-8 supports two, three, four, five, and six byte encodings. ModSecurity will locate cases when a byte or more is missing.

2. Invalid encoding. The two most significant bits in most characters are supposed to be fixed to 0x80. Attackers can use this to subvert Unicode decoders.
How can I use mod_security?

Unicode encoding validation

This feature will assume UTF-8 encoding and check for three types of errors:

3. Overlong characters. ASCII characters are mapped directly into the Unicode space and are thus represented with a single byte. However, most ASCII characters can also be encoded with two, three, four, five, and six characters thus tricking the decoder into thinking that the character is something else (and, presumably, avoiding the security check).
How can I use mod_security?

Byte range check

You can force requests to consist only of bytes from a certain byte range. This can be useful to avoid stack overflow attacks (since they usually contain "random" binary content).

```
SecFilterForceByteRange 32 126
```

This however is causing problems, since it only allows characters from ascii code 32 to 162 (obviously).
How can I use mod_security?

Byte range check

/fo/curator.php?achternaam=&check%5Bmaatschapsnaam%5D=on&maatschapsnaam
=Dani%EBIs+Dijkman+%26+Huisman+Advocaten&plaats=&submit=zoeken
HTTP/1.1" 407 492

Dani%EBIs = Daniëls

character ë (ascii 137) is used in the request, and translated to hex EB, ascii 235.

Result, false positive.
How can I use mod_security?

Byte range check

On the mod_security TODO list, there is a multiple-range aware SecFilterForceByteRange directive planned.

Default value for SecFilterForceByteRange is 0 – 255, so if you need to accept ascii codes above 126, either do not use this directive, or use it like for now:

SecFilterForceByteRange 0 255
How can I use mod_security?

Default action

Whenever a filter is matched against a request, an action (or a series of actions) is taken. Individual filters can each have their own actions but in practice you will want to define a set of actions for all filters. You can do this with the configuration directive SecFilterDefaultAction.

SecFilterDefaultAction "deny,log,status:500"
How can I use mod_security?

Allowing others to see mod_security

Normally, attackers won't be able to tell whether your web server is running mod_security or not. You can give yourself away by sending specific messages, or by using unusual HTTP codes (e.g. 406 - Not Acceptable “encoding”). If you want to stay hidden your best bet is to use HTTP 500, which stands for "Internal Server Error". Attackers that encounter such a response might think that your application has crashed.
How can I use mod_security?

Allowing others to see mod_security

One technique that often helps slow down and confuse attackers is the web server identity switch. Web servers typically send their identity with every HTTP response in the Server: header. Apache is particularly helpful here, not only sending its name and full version by default, but it also allows server modules to append their versions too.
How can I use mod_security?

Allowing others to see mod_security

Apache does have directives that will hide the Apache version you are running. It’s advisable to set the following 2 directives in the httpd.conf file.

ServerSignature On
Server Tokens Prod

Full Server: Apache/1.3.0 (Unix) PHP/3.0 MyMod/1.2
OS Server: Apache/1.3.0 (Unix)
Min Server: Apache/1.3.0
Prod Server: Apache
How can I use mod_security?

Allowing others to see mod_security

Mod_security offers a directive that will mask the identity of you Apache web server:

```
SecServerSignature "Microsoft-IIS/5.0"
```

You will need to set Server Tokens to Full in the httpd.conf file, for this mod_security directive to work.
How can I use mod_security?

Allowing others to see mod_security

SecServerSignature "FlyingServers OS 10.1-3 on a Warpath powered by Atari-2600 featuring PeopleHatePerl/4.3.5 FrontPain/5.0.2.2510 mod_vollthrostten/3.1.2"

Apache Server Status for flyingservers.nl

Server Version: FlyingServers OS 10.1-3 on a Warpath powered by Atari-2600 featuring PeopleHatePerl/4.3.5 FrontPain/5.0.2.2510 mod_vollthrostten/3.1.2"
How can I use mod_security?

Filter inheritance

Filters defined in parent folders are usually inherited by nested Apache configurations. This behavior is acceptable (and required) in most cases, but not all. Sometimes you need to relax checks in some part of the site. By using the following directive:

SecFilterInheritance Off
How can I use mod_security?

Filter inheritance

One way to use SecFilterInheritance is in controlling file uploads to your server.

# Reject requests with header "Content-Type" set to "multipart/form-data"
SecFilterSelective "HTTP_CONTENT_TYPE" multipart/form-data

# Only for the script that performs upload
<Location "/upload.php">
# Do not inherit filters from the parent folder
SecFilterInheritance Off
</Location>
How can I use mod_security?

Directory traversal

If your scripts are dealing with the file system then you need to pay attention to certain meta characters and constructs. For example, a character combination "./" in a path is a request to go up one directory level. In normal operation there is no need for this character combination to occur in requests and you can forbid them with the following filter:

    SecFilter ".\./"
How can I use mod_security?

Directory traversal

Audit log entry for the SecFilter "\./" rule

Handler: (null)
----------------------------------------
GET /scripts/%255c%255c../winnt/system32/cmd.exe?/c+dir
mod_security-message: Access denied with code 405. Pattern match "\./" at THE_REQUEST.
mod_security-action: 405
How can I use mod_security?

Cross site scripting attacks

Cross site scripting attacks (XSS) occur when an attacker injects HTML and/or Javascript code into your Web pages and then that code gets executed by other users. This is usually done by adding HTML to places where you would not expect them. A successful XSS attack can result in the attacker obtaining the cookie of your session and gaining full access to the application!

```
SecFilter "<[:space:]]*script"
SecFilter "<.+>"
```
How can I use mod_security?

Cross site scripting attacks

SecFilter "<[[:space:]]*script“

The above filter will protect only against Javascript injection with the "<script>" tag.

SecFilter "<.+>“

This second filter is more general, and disallows any HTML code in parameters.
How can I use mod_security?

Cross site scripting attacks

You need to be careful when applying filters like this since many application want HTML in parameters (e.g. CMS applications, forums, etc). You can do this with selective filtering. For example, you can have the second filter from above

```
SecFilter "<.+>"
```

as a general site wide rule, but later relax rules for a particular script with the following code:
How can I use mod_security?

Cross site scripting attacks

<Location /cms/article-update.php>
  SecFilterInheritance Off
  # other filters here ...
  SecFilterSelective "ARGS|!ARG_body" "<(.|\n)+>"
</Location>
How can I use mod_security?

Cross site scripting attacks

GET /phpinfo.php?SERVER_ADDR="&gt;&lt;script&gt;alert('test');&lt;/script&gt; HTTP/1.1

Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*

Pattern match "<( \n)*script" at THE_REQUEST
How can I use mod_security?

SQL/database attacks

Most Web applications nowadays rely heavily on databases for data manipulation. Unless great care is taken to perform database access safely, an attacker can inject arbitrary SQL commands directly into the database. This can result in the attacker reading sensitive data, changing it, or even deleting it from the database altogether.
How can I use mod_security?

SQL/database attacks

Filters like:

SecFilter "delete[:space:]++from"
SecFilter "insert[:space:]++into"
SecFilter "select++from"

can protect you from most SQL-related attacks. These are only examples, you need to craft your filters carefully depending on the actual database engine you use.
How can I use mod_security?

SQL/database attacks

Let's examine SecFilter "select.+from"

From the audit log file:

GET /myadmin/sql.php?lang=en
   en-iso-8859-1&server=1&db=snort_db&table=acid_ag&sql_query=SELECT+%2A+FROM+%60acid_ag%60&pos=0&goto=tbl_properties_structure.php HTTP/1.1
How can I use mod_security?

SQL/database attacks

Let's examine SecFilter "select.+from"

mod_security-message: Access denied with code 405. Pattern match "select.+from" at THE_REQUEST.

This might not be what you want, after all you should be able to browse a database, delete and insert data.

Plan ahead, when you are about to use these SQL filters.
How can I use mod_security?

Chain

Rule chaining allows you to chain several rules into a bigger test. Only the last rule in the chain will affect the request but in order to reach it, all rules before it must be matched too. Here is an example of how you might use this feature.

SecFilterSelective ARG_username admin chain
SecFilterSelective REMOTE_ADDR "!^YOUR_IP_ADDRESS_HERE$"
How can I use mod_security?

Chain

The following rule shows how you can chain rules. It will send an email when the boss forgets his password again. We have two rules here. The first will trigger only when one specific file is requested (the one showing the "Login failed" message). The second rule will then check to see if the username used was boss. If it was, it will then execute an external script.

SecFilterSelective REQUEST_URI "login_failed\.php" chain
SecFilterSelective ARG_username "^boss$" log,exec:/home/apache/bin/notagain.pl
How can I use mod_security?

Redirect

The following rule sends Google back home by redirecting Googlebot somewhere else, based on the User-Agent header. It does not log rule matches.

SecFilter HTTP_USER_AGENT "Google" nolog,redirect:http://www.google.com
How can I use mod_security?

Exec

Execute a binary on filter match. Full path to the binary is required:

SecFilter KEYWORD "log,exec:/home/ivanr/report-attack.pl"

This directive does not effect the primary action (allow or deny). This action will always call script with no parameters, but providing all information in the environment. All the usual CGI environment variables will be there.
How can I use mod_security?

Operating system command execution

Web applications are sometimes written to execute operating system commands to perform operations. A persistent attacker may find a hole in the concept, allowing him to execute arbitrary commands on the system.

A filter like this:

```
SecFilterSelective THE_REQUEST "bin/"
```

will detect attempts to execute binaries residing in various folders on a Unix-related operating system.
How can I use mod_security?

Operating system command execution


mod_security-message: Access denied with code 405. Pattern match "bin/" at THE_REQUEST.
How can I use mod_security?

Supporting Snort rules

Snort classifies rules into web attacks and web activities. Web attack rules are converted to reject incoming requests, while web activity rules only log the activity into the error log.

Be warned that some rules won't make sense after being converted; Snort and mod_security are, after all, different tools. I suggest that you scan through the rules and delete the ones that do not make sense or the ones that do not apply to your circumstances:
How can I use mod_security?

Supporting Snort rules

Ivan wrote a Perl script to convert Snort rules to mod_security rules in bulk. Here you can download the script to apply it to a freshly downloaded set of Snort rules, but you can also download the end result (based on the rules I downloaded at the beginning of October).

The perl script
Converted Snort rules
Almost done

Daily routine

– Review logwatch output to see if anyone is trying to FTP or SSH into the system who should not have access. Block them out in your firewall.

– Review chkrootkit and rootkit hunter output to see if your system has been compromised with a rootkit; if so review the CERT reporting guidelines and CERT recovery steps.

– Review firewall logs and port scan logs to see if anyone is attacking your servers. Block them out in your firewall.
Almost done

Daily routine

– Change your root password or any root equivalent password on a regular basis.

– Review your system log files on a regular basis looking for errors and suspicious activity.

– Regularly review new technologies that are available for helping with security.
Conclusion

Mod_security is a powerful tool, but can be overwhelming at first. Start with some simple rules, and plan your rules ahead.

In a shared hosting environment it’s difficult to apply system wide rules, since you have no control over the way your users and customers program.

In a closed environment, like a intranet, all users will need to adhere to rules and policies.
Almost done

Acknowledgement

Mod_security is created and maintained by Ivan Ristic from England. He is currently working on a book called “Apache Security” that’s due to be published in February 2005 by O'Reilly.

A preview of his book can be found at:

Apache Security Book

with some interesting free tools.
Almost done

Acknowledgement

Mod_security home page is located at:

http://www.modsecurity.org/

Some papers by the hand of Ivan:

Introducing mod_security
Web Security Appliance with Apache and mod_security
Done

A blog with the latest information on this presentation can be found at:

http://www.flyingtalks.com

For question, remarks etc, contact met at

hans@flyingtalks.com
Locking down your Apache Web Server with mod_security
Locking down your Apache Web Server with mod_security

Q&A