Introducción

Wordpress es una moderna plataforma de personal de publicación semántica enfocada en la estética, los estándares web, y la usabilidad. Mientras que desafortunadamente carece de las funciones de seguridad vitales que protegen la aplicación de ataques maliciosos. Una instalación predeterminada de WordPress no es tan segura como a un profesional de seguridad de aplicaciones web le gustaría, de ahí la necesidad de capas de defensa extra para asegurarse de que la aplicación permanece segura en todo momento.

¿Qué es ModSecurity y por qué lo necesito?

ModSecurity funciona como una capa de defensa entre el servidor web y la aplicación y funciona con una serie de reglas que definen como necesita reaccionar a ciertos tipos de peticiones y comportamientos. Con un ratio de ataques dirigido a aplicaciones web siempre en crecimiento, ModSecurity ayuda añadiendo una capa externa de seguridad que incrementa la seguridad, detecta, y previene ataques antes de que alcancen las aplicaciones web.

Instalando ModSecurity

Esta guía no profundizará en los detalles de instalación de ModSecurity ya que ya existen por ahí varias guías excelentes. Si no está ya ejecutando ModSecurity y le gustaría saber más, le sugerimos visitar:

- http://modsecurity.org/documentation/index.html
- http://atomicplayboy.net/blog/2005/01/30/an-introduction-to-mod-security/

Configuración general

Los siguientes ajustes de configuración están incluidos como parte de la instalación estándar, así que asegúrese de que los suyos se parecen a estos de abajo.

```
SecFilterEngine On
SecServerSignature "Largo"
SecFilterCheckURLEncoding On
SecFilterCheckUnicodeEncoding Off
SecFilterForceByteRange 1 255
SecAuditEngine RelevantOnly
SecAuditLog /var/log/httpd/modsec_log
SecFilterScanPOST On
SecFilterDefaultAction "deny,log,status:500"
```

Reglas de seguridad adicionales para aplicaciones web

Aunque no están directamente relacionadas con Wordpress, las siguientes reglas previenen ataques conocidos y actividad maliciosa y por lo tanto incrementan la seguridad general de su blog.

```
# No aceptar peticiones GET o HEAD con la etiqueta  body

SecFilterSelective REQUEST_METHOD "^(GET|HEAD)$" chain
SecFilterSelective HTTP_Content-Length "!^$"
```

```
# Obligar a que se proporcione Content-Length
# con cada petición POST
SecFilterSelective REQUEST_METHOD "^POST$" chain
SecFilterSelective HTTP_Content-Length "^$"

# Prevención genérica de inyección SQL
SecFilter "delete[[:space:]]+from"
SecFilter "insert[[:space:]]+into"
SecFilter "select.+from"
SecFilterSelective ARGS "(or.+1[[:space:]]*=[[:space:]]1|(or 1=1|'.+)--')"
"id:300014,rev:1,severity:2,msg:'Generic SQL injection protection'"
SecFilterSelective ARGS
"((alter|create|drop)[[:space:]]+(column|database|procedure|table)|delete[[:space:]]+from|update.
+set.+=)" "id:300015,rev:1,severity:2,msg:'Generic SQL injection protection'"

# PHP
#Protección para ataques XSS a través de la cookie de sesión PHP
SecFilterSelective ARG_PHPSESSID "!^[0-9a-z]*$"
SecFilterSelective COOKIE_PHPSESSID "!^[0-9a-z]*$"
SecFilterSelective REQUEST_URI
“(&(cmd|command)=(id|uname)\x20|cmd\?(cmd|command)=|(spy|cmd|cmd_out|sh)\.(gif|jpg|pn
g|bmp|txt)\?&(cmd|command)=|\.php\?&(cmd|command)=)"

 # Firmas genéricas de vulnerabilidad PHP
SecFilterSelective POST_PAYLOAD|REQUEST_URI "<\?php
(chr|fwrite|fopen|system|echr|passthru|popen|proc_open|shell_exec|exec|proc_nice|proc_termin
ate|proc_get_status|proc_close|pfsockopen|leak|apache_child_terminate|posix_kill|posix_mkfifo|
posix_setpgid|posix_setsid|posix_setuid|phpinfo)\(.*\)\;" id:330002,rev:1,severity:2,msg:'Generic
PHP exploit pattern denied'"

#Inyección genérica de PHP con archivo remoto
SecFilterSelective REQUEST_URI "!(/do_command)" chain
SecFilterSelective REQUEST_URI "\.php\?.*=(https?|ftp)\:/.*(cmd|command)="

#Protecciones generales de [url] para foros php (phpbb y otros, para protegerse contra ataques de
inyección de scripts en enlaces en urls)
SecFilterSelective THE_REQUEST "\.php\?" chain
SecFilter "\[url=(script|javascript|applet|about|chrome|activex)\:/.*\].*\[/url\]"

#Firmas genéricas de vulnerabilidad PHP
SecFilterSelective THE_REQUEST
"(chr|fwrite|fopen|system|e?chr|passthru|popen|proc_open|shell_exec|exec|proc_nice|proc_ter
minate|proc_get_status|proc_close|pfsockopen|leak|apache_child_terminate|posix_kill|posix_mkfi
fo|posix_setpgid|posix_setsid|posix_setuid|phpinfo)\(.*\)\;"
"id:330001,rev:1,severity:2,msg:'Generic PHP exploit pattern denied'"
```

Como con cualquier guía de restricciones, las reglas anteriores pueden evitar que otras aplicaciones funcionen en el servidor. Nosotros no nos hacemos responsables de nada, no importa lo que pase.

Reglas específicas de ModSecurity para Wordpress

Las reglas comentadas en este artículo son en relación a la versión 1.9.x de ModSecurity. Tan pronto como el tiempo no los permita, actualizaremos el documento para contemplar el lanzamiento de ModSecurity 2.x. Las reglas más abajo no son una lista definitiva de reglas, pero sí una propuesta como una guía para bloquear vectores de ataque conocidos y métodos que tiene su objetivo únicamente en instalaciones de WordPress. Como con cualquier otro programa, a menudo se descubren nuevas vulnerabilidades, así que asegúrese de consultar la web para cualquier nueva regla.

```
# Añade control de acceso a la página de inicio de sesión (¡recuerde cambiar la dirección  IP!)
SecFilterSelective REMOTE_ADDR “!192\.168\.0\.69. chain
SecFilterSelective REQUEST_URI “/wp-login.php” \
log,deny,redirect:http://www.yoursite.com/nologin.html

# Prevención de vulnerabilidad de inyección SQL en WordPress
SecFilterSelective REQUEST_URI "/index\.php" chain
SecFilterSelective ARG_poll|ARG_category|ARG_ctg
"((select|grant|delete|insert|drop|alter|replace|truncate|update|create|rename|describe)[[:space
:]]+[A-Z|a-z|0-9|\*| |\,]+[[:space:]]+(from|into|table|database|index|view)[[:space:]]+[A-Z|a-z|0-
9|\*| |\,]|\'|UNION.*SELECT.*INTO.*FROM)"
SecFilterSelective REQUEST_URI "/wp-
trackback\.php\?tb_id=*(select|grant|delete|insert|drop|do|alter|replace|truncate|update|create
|rename|describe)[[:space:]]+[A-Z|a-z|0-9|\*| ]+[[:space:]](from|into|table|database|index|view)"
SecFilterSelective REQUEST_URI "/wp-trackback\.php" chain
SecFilterSelective ARG_tb_id
"(select|grant|delete|insert|drop|do|alter|replace|truncate|update|create|rename|describe)[[:sp
ace:]]+[A-Z|a-z|0-9|\*| ]+[[:space:]](from|into|table|database|index|view)"
SecFilterSelective REQUEST_URI
"/index\.php\?cat=.*(select|grant|delete|insert|drop|do|alter|replace|truncate|update|create|ren
ame|describe)[[:space:]]+[A-Z|a-z|0-9|\*| |,]+[[:space:]](from|into|table|database|index|view)"
SecFilterSelective REQUEST_URI "wp-pass\.php\?\_\wp\_\http\_\referer\="
SecFilterSelective HTTP_USER_AGENT "WordPress Hash Grabber"

# Prevención de vulnerabilidad de categoría de WordPress
SecFilterSelective REQUEST_URI "/WordPress/" chain
SecFilterSelective ARG_cat "!^[0-9]*$"

# Vulnerabilidad de inyección por consola en WordPress
SecFilterSelective REQUEST_URI "/cache/user.*/.*\.php\?cmd="
"id:390064,rev:1,severity:2,msg:'JITP: WordPress shell injection Vulnerability'"

#Prevención de inserción de código PHP "cache_lastpostdate" en WordPress
SecFilterSelective ARG_cache_lastpostdate "<\?php"

#Prevención de inyección SQL en WordPress y vulnerabilidad de revelación de ruta del feed
SecFilterSelective REQUEST_URI "/\?feed\=rss2\&p=\-1"
SecFilterSelective REQUEST_URI "/wp\/WordPress\/\?feed\=rss2\&p=\-1"
```

```
# Firmas de ataque experimental de WordPress con XML-RPC
SecFilter "\'\,\'\'\)\)\;"
SecFilter "\<param\>\<name\>.*\'\)\;"
SecFilter "(\<xml|\<.*xml)" chain
SecFilter "(echo(
|\(|\').*\;|chr|fwrite|fopen|system|echr|passthru|popen|proc_open|shell_exec|exec|proc_nice|proc_terminate|proc_get_status|proc_close|pfsockopen|leak|apache_child_terminate|posix_kill|posix_mkfifo|posix_setpgid|posix_setsid|posix_setuid|phpinfo)\(.*\)\;" chain
SecFilter "methodCall\>"

# Ataques XML-RPC específicos de WordPress sobre  xmlrpc.php
SecFilterSelective THE_REQUEST "(/xmlrpc|.*xmlrpc_services)\.php" chain
SecFilter "(\<xml|\<.*xml)" chain
SecFilter "(echo(
|\(|\').*\;|chr|fwrite|fopen|system|echr|passthru|popen|proc_open|shell_exec|exec|proc_nice|proc_terminate|proc_get_status|proc_close|pfsockopen|leak|apache_child_terminate|posix_kill|posix_mkfifo|posix_setpgid|posix_setsid|posix_setuid|phpinfo)\(.*\)\;"

# Firma genérica de inyección XML-RPC SQL en WordPress
SecFilterSelective THE_REQUEST "(/xmlrpc|.*xmlrpc_services)\.php" chain

SecFilter
"<methodName>.*</methodName>.*<value><string>.*(select|grant|delete|insert|drop|do|alter|replace|truncate|update|create|rename|describe)[[:space:]]+[A-Z|a-z|0-9|\*|
|,]+[[:space:]](from|into|table|database|index|view).*methodName\>"

# Previene ataque XSS basado en themes de  WP: Búsqueda de WordPress
SecFilterSelective THE_REQUEST "\?s=" chain
SecFilter "<[[:space:]]*(script|javascript|applet|about|chrome|activex|object|iframe|img)"

 # Firmas de ataques genéricos de WordPress XML-RPC
SecFilterSelective POST_PAYLOAD "^Content-Type\: application/xml" chain
```

Conclusión

Instalar y usar ModSecurity añadirá la capa extra de seguridad necesaria para asegurarse de que el blog permanece seguro. Planeamos desarrollar este documento de referencia sobre la marcha. Las opiniones y comentarios serán bienvenidos. Agradecimiento especial a Daniel Cuthbert quien fue el autor principal de este documento de referencia y por su gran investigación y contribuciones al proyecto BlogSecurity.

Créditos

Autor: Daniel Cuthbert http://danielcuthbert.com/
Coautor: David Kierznowski http://blogsecurity.net
Traductor: Samuel Aguilera http://agamum.net/blog/