Protege tu servidor web con el módulo ModSecurity

Cardenal Gardoki, 1
48008 BILBAO (Vizcaya)
Teléfono: 902 012 199
www.hostalia.com

HOSTALIA .com

Mantener la seguridad de un servidor web no es tarea sencilla, más aún cuando sobre él corren páginas web que funcionan mediante programación PHP, Python… lenguajes en los que en ocasiones se puede dejar alguna puerta abierta por error que sea utilizada por los hackers. Afortunadamente nos podemos encontrar herramientas que nos ayudan a combatir este tipo de situaciones, siendo uno de lo más utilizado el módulo "ModSecurity", que será el protagonista de nuestro White Paper de este mes.

Qué es ModSecurity

ModSecurity es un módulo que se instala en algunos servidores web que nos ayuda a protegerlo de posibles ataques, realizando para ello filtrados de los distintos métodos HTTP que existen en la actualidad como GET o POST. Esto le permite adquirir el comportamiento de un firewall orientado a web, filtrando ataques potenciales a los sitios que se encuentran hospedados en esa máquina.

Para evitar este tipo de ataques, ModSecurity funciona mediante el uso de expresiones regulares y conjuntos de reglas, que podemos personalizar todo lo que queramos y que pueden ser cargadas o excluidas según el virtualhost o directorio. Entre los ataques que puede llegar a filtrar este módulo están Cross Scripting o XSS, inyecciones SQL, anomalías en protocolos, Robots maliciosos, Troyanos, inclusión de archivos (LFI)…

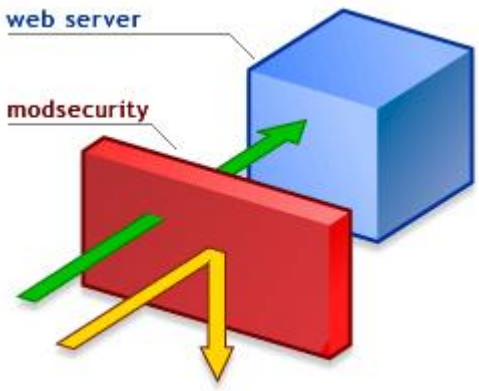
Un ejemplo de regla muy básico sería el siguiente:

```
SecRule REQUEST_HEADERS:User-Agent "xxx.2.6.5" "log,drop"
```

La instrucción SecRule crea una regla que examina las cabeceras, en este caso el valor del "User Agent", valor que comparará con lo que haya entre las comillas dobles que será una expresión regular. Si se cumple la regla, entonces el módulo lo registrará en el archivo log y denegará el acceso (drop).

En un principio ModSecurity sólo estaba disponible para servidores Apache, pero ya es posible hacer uso de él en otros tipos de servidores como pueden ser Microsoft IIS o Nginx.

Ciclo de vida de ModSecurity



Para comprender el funcionamiento de este módulo, pasaremos a explicar su ciclo de vida, explicando cada uno de los pasos involucrados en él.

1.- Request Headers

Lo que hace es interceptar la petición realizada por un cliente cogiendo de ella los headers y el body para enviar todo al motor de reglas de ModSecurity. Todo este proceso se realiza antes de que el servidor web empiece el procesamiento de esa petición.

2.- Request Body

Uno de los más importantes ya que es donde se pone en marcha el análisis de la petición que ha sido interceptada. Llegados a este punto, el motor de reglas de ModSecurity tiene a su disposición toda la información de la petición por lo que procede a determinar si cumple alguna de las reglas establecidas para ver si la petición sigue su curso normal, la interrumpe o si debe llevarse a cabo otra acción.

3.- Response Headers

Este tercer paso se inicia justo después de que el servidor web haya generado la respuesta para la petición recibida, cuando las cabeceras ya han sido analizadas y generadas pero el body aún no ha sido leído. Llegados a este punto, el motor de reglas determina qué campos deben ser inspeccionados en el cuerpo de la respuesta.

4.- Response Body

En esta fase se lleva a cabo el análisis del cuerpo de la respuesta. Continúa después de la fase anterior donde el motor de reglas para la respuesta ya ha sido inicializado. El cuerpo de la respuesta se ha leído completamente y con toda esa información, el motor de reglas toma la decisión más adecuada.

5. Logging

Se trata de la única fase del ciclo de vida que no puede ser bloqueada. Una vez que se ejecuta, la transacción habrá finalizado y el administrador poco podrá hacer más. Las reglas definidas en este paso indican la forma en la que el logging se llevará a cabo.

Funcionalidades

El módulo cuenta con diversas funcionalidades que pasamos a mostraros a continuación.

Filtrado de peticiones

Las peticiones HTTP entrantes son analizadas por el módulo ModSecurity antes de que sean interpretadas por el servidor web en busca de algún tipo de vulnerabilidad. Para ello se le aplican un conjunto de reglas las cuales son definidas por el administrador del servidor.

Técnicas antievasión

Se lleva a cabo una normalización de las rutas y los parámetros antes del análisis para evitar técnicas de evasión. Entre ellas están la de eliminar múltiples barras (//), elimina directorios referenciados por sí mismo (./), decodificación de url o el reemplazo de valores nulos por espacios (%00).

Compresión del protocolo HTTP

Al entender las peticiones realizadas mediante el protocolo HTTP, ModSecurity puede realizar filtrados específicos.

Análisis Post PayLoad

Es capaz de interceptar y analizar el contenido transmitido mediante el método POST.

Filtrado HTTPS

ModSecurity tiene acceso a los datos de las peticiones una vez que estos hayan sido descifrados.

Monitorización en tiempo real

Este módulo es capaz de monitorizar el tráfico en tiempo real para detectar ataques, es decir, puede funcionar como una herramienta que es capaz de detectar intrusos.

Principales directivas de ModSecurity

A continuación haremos un repaso por algunas de las directivas más interesantes que nos ofrece este módulo y que pueden ser aplicadas en los archivos de configuración del módulo.

SecRequestBodyAccess

Se trata de una directiva que puede tomar los valores de “on” y “off”, y dependiendo de su valor indica al módulo cuándo debe acceder al cuerpo de las peticiones, tal y como vimos en el apartado del ciclo de vida. En el caso de configurarlo a “off”, estaremos indicando que no se acceda a los valores que viajan en el cuerpo de la petición, por ejemplo los parámetros que viajan por POST, una situación que no siempre es deseado que ocurra ya que puede ser una puerta de entrada a posibles ataques. Si toma el valor “on” podrá acceder a esa información pero estaremos aumentando el uso de la memoria RAM de la máquina.

SecRequestBodyInMemoryLimit

Mediante esta directiva se indica el tamaño en bytes que será reservado en la memoria RAM para que el módulo almacene ahí los valores que vienen en los cuerpos de las peticiones. Para que esta directiva tenga algún efecto, debe estar activada la que hemos visto anteriormente.

Como hemos dicho, el tamaño se expresa en bytes, por lo que si queremos asignar un tamaño de 512 KB, lo deberemos indicar de la siguiente forma:

```
SecRequestBodyInMemoryLimit 524288
```

SecRequestBodyLimit

Indica el tamaño máximo de bytes permitidos para los buffers correspondientes a los cuerpos de las peticiones. Si una petición supera este valor, será rechazada con un error 413 (Request Entity Too Large). Si en el servidor hay alguna aplicación que permite la subida de archivos, el tamaño que se le debe asignar a esta directiva debe ser el mismo que el tamaño máximo de los ficheros que se podrán subir.

SecRequestBodyNoFilesLimit

Se trata de una directiva similar a la anterior pero con la diferencia de que en este caso no se tiene en cuenta si existen aplicaciones que realicen subida de archivos al servidor. En el caso de existir, la anterior

directiva tendría preferencia sobre ésta. De forma habitual, este valor debe ser menor que el establecido en el caso anterior.

SecResponseBodyAccess

Mediante esta directiva se activa o desactiva el procesamiento de ModSecurity para las respuestas generadas por el servidor web, es decir, se indica si queremos que las respuestas sean analizadas en busca de posibles amenazas. Puede tomar el valor de “off” u “on”.

SecResponseBodyLimit

Se trata de una directiva que permite establecer el límite en bytes de las respuestas generadas por el servidor. Si el valor asignado es muy pequeños corremos el riesgo de que no se muestre la página, y de un “Internal Server Error” en caso de que la respuesta supere el tamaño asignado aquí.

SecResponseMimeType

Mediante el uso de esta directiva se le indica al módulo los tipos MIME que pueden ir en las respuestas dadas por el servidor, lo que será útil para indicar qué está y qué no está permitido. Un ejemplo de uso de esta directiva es el siguiente:

```
SecResponseMimeType text/plain text/html text/xml
```

Por todo lo que hemos comentado, ModSecurity se ha convertido desde su aparición en una herramienta imprescindible para asegurar una mejor defensa de los servidores, una herramienta que todo administrador de sistemas debería tener en cuenta para defenderse de posibles ataques.