Instalar Modsecurity como WAF

Modsecurity es una de las aplicaciones que más se usa para proteger los servidores web mediante las funcionalidades WAF que ofrece, aunque se rumorea que va a dejar de existir, todavía tiene validez, ya que las alternativas están poco maduras.

Instalar ModSecurity en Debian 12

Actualizar el sistema

Como siempre comenzaremos por actualizar nuestro sistema.

```
apt update && apt upgrade
```

Instalar ModSecurity

A continuación instalamos el paquete modsecurity

```
apt install libapache2-mod-security2
```

Habilitar ModSecurity

Habilitamos el Modsecurity en el apache, añadiendo el módulo.

```
a2enmod security2
```

Reiniciamos apache

```
systemctl restart apache2
```

Configurar ModSecurity

El archivo de configuración principal de ModSecurity se encuentra en **/etc/apache2/mods-enabled/security2.conf.** Este archivo contiene una gran cantidad de opciones que puede modificar para personalizar el comportamiento de ModSecurity.

- SecRuleEngine: Esta opción define si ModSecurity está en modo de detección (**DetectionOnly**) o si está bloqueando solicitudes (**On**).
- SecDefaultAction: Esta opción define la acción predeterminada que ModSecurity tomará cuando se detecte una violación de las reglas.
- SecRulesFile: Esta opción define la ruta al archivo de reglas de ModSecurity.

Ahora lo preparamos

```
mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf
```

Editamos el fichero y buscamos la línea

```
SecRuleEngine DetectionOnly
```

Y la cambiamos por **SecRuleEngine On**, como hemos comentado antes. Luego buscaa siguiente línea (línea 186), que le indica a ModSecurity qué información debe incluirse en el registro de auditoría.

```
SecAuditLogParts ABDEFHIJZ
```

La cambiamos por la siguiente

```
SecAuditLogParts ABCEFHJKZ
```

Reiniciamos apache

```
systemctl restart apache2
```

Ejemplo de configuración

Vamos a ver como configurar ModSecurity con OWASP CRS v3

Instalar ModSecurity y OWASP CRS v3

```
wget https://github.com/coreruleset/coreruleset/archive/v3.3.0.tar.gz
tar xvf v3.3.0.tar.gz
mkdir /etc/apache2/modsecurity-crs/
mv coreruleset-3.3.0/ /etc/apache2/modsecurity-crs/
```

Vamos a la carpeta

```
cd /etc/apache2/modsecurity-crs/coreruleset-3.3.0/
mv crs-setup.conf.example crs-setup.conf
```

Editar el archivo /etc/apache2/mods-enabled/security2.conf

```
nano /etc/apache2/mods-enabled/security2.conf
```

Buscamos la línea

```
IncludeOptional /usr/share/modsecurity-crs/*.load
```

Y la sustituimos por:

```
IncludeOptional /etc/apache2/modsecurity-crs/coreruleset-3.3.0/crs-setup.conf
IncludeOptional /etc/apache2/modsecurity-crs/coreruleset-3.3.0/rules/*.conf
```

Guardamos y probamos la configuración de apache

```
apache2ctl -t
```

Si todo es correcto. reiniciamos apache

```
systemctl restart apache2
```

Reglas personalizadas

Podemos editar el archivo de configuración de OWASP CRS /etc/apache2/modsecurity-crs/coreruleset-3.3.0/crs-setup.conf para configurar reglas personalizadas.

Reiniciar Apache

```
systemctl restart apache2
```

Probar ModSecurity

Puedes probar ModSecurity visitando la web con un navegador web y ejecutando una herramienta de prueba de vulnerabilidades como OWASP ZAP.

Puedes ver que OWASP CRS se puede ejecutar en dos modos:

self-contained. Este es el modo tradicional utilizado en CRS v2.x. Si una solicitud HTTP coincide con una regla, ModSecurity bloqueará la solicitud HTTP inmediatamente y dejará de evaluar las reglas restantes.
anomaly scoring mode. Este es el modo predeterminado utilizado en CRS v3.x. ModSecurity comparará una solicitud HTTP con todas las reglas y agregará una puntuación a cada regla coincidente. Si se alcanza un umbral, la solicitud HTTP se considera un ataque y se bloqueará. La puntuación predeterminada para las solicitudes entrantes es 5 y para la respuesta saliente es 4.

Cuando se ejecuta en modo de puntuación de anomalías, hay 4 niveles de paranoia.

- Nivel de paranoia 1 (predeterminado)
- Paranoia nivel 2
- Paranoia nivel 3
- Paranoia nivel 4

Con cada aumento del nivel de paranoia, el CRS habilita reglas adicionales que le brindan un mayor nivel de seguridad. Sin embargo, los niveles más altos de paranoia también aumentan la posibilidad de bloquear parte del tráfico legítimo debido a falsas alarmas.

Ejemplos de reglas de OWASP CSR v3

```
SecRule REQUEST_METHOD "^(TRACE|TRACK)$" "phase:1,deny,log,status:405"
SecRule REQUEST_HEADERS:Content-Type "application/x-www-form-urlencoded"
"phase:2,deny,log,status:403"
SecRule REQUEST_URI "@rx (..|~) " "phase:2,deny,log,status:403"
```

A continuación por ejemplo tenemos un ejemplo de un conjunto de reglas de hardening de OWASP CSR v3 para Wordpress. Lo puedes encontrar en este enlace de GitHub

Aquí ponemos 3 ejemplos del código anterior para bloquear accesos no autorizados a la carpeta wp-includes para subir archivos del tipo php, el acceso a la ruta wp-admin y el acceso al XML-RPC

```
SecRule REQUEST_FILENAME "^/wp\-includes(/.*\.php(|[\/].*)|(|\/))$" "phase:1,id:22200001,\
  t:lowercase,t:normalizePath,t:trim,\
  block,\
  log,\
  rev:'1',\
  severity:'6',\
  maturity:'9',\
  accuracy:'9',\
  ver:'%{tx.wprs_version}',\
  tag:'wordpress',\
  tag:'includes',\
  logdata:'Request Filename %{REQUEST_FILENAME}',\
  msg:'WordPress: /wp-includes access attempt'"
SecRule REQUEST_FILENAME "^/wp-admin/(?:install|includes)" "phase:1,id:22200003,\
  t:lowercase,t:normalizePath,t:trim,\
  block,\
  log,\
  rev:'1',\
  severity:'6',\
  maturity:'9',\
```

```
    accuracy:'9',\
    ver:'%{tx.wprs_version}',\
    tag:'wordpress',\
    tag:'includes',\
    logdata:'Request Filename %{REQUEST_FILENAME}',\
    msg:'WordPress: File /wp-admin access attempt'"
  SecRule tx:wprs_allow_xmlrpc "@eq 1" \
    "phase:1,\
    id:22200013,\
    pass,\
    nolog,\
    skipAfter:END_WPRS_XMLRPC"

  SecMarker BEGIN_WPRS_XMLRPC

  SecRule REQUEST_FILENAME "^/xmlrpc\.php" "phase:1,id:22200015,\
    t:lowercase,t:normalizePath,t:trim,\
    block,\
    log,\
    rev:'1',\
    severity:'6',\
    maturity:'9',\
    accuracy:'9',\
    ver:'%{tx.wprs_version}',\
    tag:'wordpress',\
    tag:'xmlrpc',\
    logdata:'Request Filename %{REQUEST_FILENAME}',\
    msg:'WordPress: /xmlrpc.php access attempt'"

  SecMarker END_WPRS_XMLRPC
```

Interpretar los registros de ModSecurity

Es importante analizar los registros de ModSecurity para saber qué tipo de ataques se dirigen a sus aplicaciones web y tomar mejores medidas para defenderse de los actores de amenazas. Existen principalmente dos tipos de registros en ModSecurity:

registro de depuración: deshabilitado de forma predeterminada.
registro de auditoría: /var/log/apache2/modsec_audit.log
Para comprender los registros de auditoría de ModSecurity, necesita conocer las 5 fases de procesamiento en ModSecurity, que son:

- Fase 1: inspeccionar los encabezados de las solicitudes
- Fase 2: inspeccionar el cuerpo de la solicitud
- Fase 3: inspeccionar los encabezados de respuesta
- Fase 4: inspeccionar el cuerpo de respuesta
- Fase 5: Acción (registro/bloqueo de solicitudes maliciosas)

También hay dos tipos de archivos de registro.

Serial / Serie : un archivo para todos los registros. Este es el valor predeterminado.
Concurrent /Simultáneo: múltiples archivos de log. Esto puede proporcionar un mejor rendimiento de escritura. Si nota que sus páginas web se ralentizan después de habilitar ModSecurity, puede optar por utilizar el tipo de registro simultáneo.
Los eventos del registro se dividen en varias secciones.

- sección A: encabezado del registro de auditoría
- sección B: encabezado de solicitud
- sección C: cuerpo de la solicitud
- sección D: reservada
- sección E: intermediary response body
- sección F: encabezados de respuesta final
- sección G: reservada
- sección H: audit log trailer
- sección I: compact request body alternative, which excludes files
- sección J: información sobre archivos cargados
- sección K: cada regla que coincide con un evento, en orden de coincidencia
- sección Z: límite final

Si ejecuta un sitio web con mucho tráfico, el registro de auditoría de ModSecurity puede volverse demasiado grande muy rápidamente, por lo que debemos configurar la rotación de registros para el registro de auditoría de ModSecurity. Cree un archivo de configuración logrotate para ModSecurity.

Revision #10
Created 6 February 2024 05:59:16 by etaboada
Updated 9 February 2024 18:45:59 by etaboada