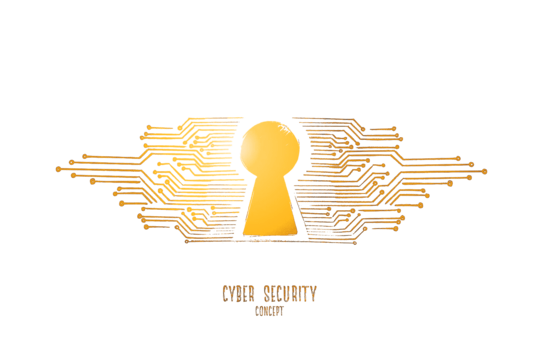
Bastionado de redes 

y sistemas.



# Prueba de modsecurity con web vulnerables

# Introducción

OWASP Mutillidae II es una aplicación web libre, open source, deliberadamente vulnerable, la cual proporciona un objetivo para entusiastas en seguridad web. Mutillidae puede ser instalado sobre Linux y Window utilizando LAMP, WAMP, y XAMMP. Incluye docenas de vulnerabilidades y sugerencias para ayudar al usuario. Es un entorno para hacking web fácil de utilizar, diseñado para laboratorios, entusiastas de seguridad, salones de clase, CTF, y herramientas para la evaluación de vulnerabilidades. Mutillidae ha sido utilizado en cursos de seguridad, cursos de entrenamiento para empresas, y como objetivo para evaluar diverso software de seguridad.

Open Web Application Security Project® (OWASP) es una fundación sin fines de lucro que trabaja para mejorar la seguridad del software. A través de proyectos de software de código abierto liderados por la comunidad, cientos de capítulos locales en todo el mundo, decenas de miles de miembros y conferencias educativas y de capacitación líderes, la Fundación OWASP es la fuente para que los desarrolladores y tecnólogos protejan la web.

Con el objetivo de comparar los efectos conseguidos mediante ataques a la aplicación mutillidae, hay que llevar a cabo las siguientes pruebas en cada una de las dos máquinas por separado (lamp, fwmodsecurity).

Se arranca una máquina, se pasan todas las pruebas, se monitorizan y se recogen resultados; y, posteriormente, se realiza la misma operación con la otra máquina. También se puede utilizar solo la máquina fwmodsecurity habilitando y deshabilitando el firewall mediante el parámetro SecRuleEngine On/Off en /etc/modsecurity/modsecurity.conf

# 

# Objetivo

1. **Familiarización con ModSecurity:** Entender y familiarizarse con ModSecurity, un firewall de aplicaciones web (WAF) de código abierto ampliamente utilizado para proteger aplicaciones web contra ataques maliciosos.
2. **Evaluación de la eficacia de ModSecurity:** Analizar cómo ModSecurity puede detectar y prevenir ataques web, en particular ataques de inyección SQL, en una aplicación web vulnerable como Mutillidae.
3. **Prueba de vulnerabilidades web:** Utilizar Mutillidae como una máquina de pruebas para simular varios tipos de ataques web, principalmente inyecciones SQL, con el fin de evaluar cómo ModSecurity responde a estos ataques y si puede mitigarlos eficazmente.
4. **Comparación de resultados:** Comparar los resultados obtenidos antes y después de aplicar ModSecurity para determinar cómo el WAF puede mejorar la seguridad de la aplicación web al detectar y bloquear ataques web.
5. **Aprendizaje práctico de seguridad en aplicaciones web:** Brindar a los estudiantes una experiencia práctica en la evaluación de la seguridad de aplicaciones web y en el uso de herramientas como ModSecurity para mitigar vulnerabilidades comunes en ellas.

| Herramientas |
| --- |
| * git * curl (opcional, un buscador es la alternativa) |
| * vim (opcional, nano es la alternativa) |

**ÍNDICE**

[**Prueba de modsecurity con web vulnerables 2**](#_xfpg6pduf6p7)

[**Introducción 2**](#_b9oaw39qdr1r)

[**Objetivo 2**](#_3uoprqr5rx0x)

[**Herramientas 2**](#_wzp689uhopv7)

[**Instalación LAMP 4**](#_9r2tp0onntrz)

[mariadb 4](#_ihs58u9pnvo1)

[php 4](#_jawkcsbwpy2s)

[Apache (ya instalado) 4](#_h81rstedg0wl)

[phpmyadmin 5](#_625qtodxtywf)

[OpenLdap 7](#_ep6vvqpo5is8)

[Instalación de LDAP 8](#_j3gukr88ioud)

[Configuración 9](#_kjc3p6510j2f)

[**Mutillidae 10**](#_baxsa2wpwcu9)

[Ataques sin Modsecurity 11](#_vj0b75bksp5y)

[Ver configuración de php (phpinfo) 11](#_vj0b75bksp5y)

[SQL Injection 11](#_yw5kzdb6g9yg)

[Insert injection 12](#_9ebhdnps0qiy)

[Cross site Script (XSS) Injection 13](#_qk1zdxdjor4n)

[Ataques con Modsecurity 15](#_vxfnipec94i4)

[Ver configuración de php (phpinfo) 15](#_huwhk6d31sn9)

[SQL Injection 15](#_sk47a1rnfg0w)

[Insert injection 16](#_got91f7ylu6a)

[Cross site Script (XSS) Injection 17](#_o1zw18vnuvtj)

[Modsecurity 18](#_yi76hcb14op6)

[Instalación 18](#_yi76hcb14op6)

[Configuración 19](#_7vn9dmiww4s2)

[Comprobar que se puede añadir configuraciones 19](#_8nqctiejo1rg)

[Definición de reglas OWASP (CRS) 19](#_9iklplpg98x0)

[Entendiendo OWASP CRS (Core Rule Set) 20](#_sl1czfbdztio)

[Registros de ModSecurity 21](#_dquursris8wz)

[Manejo de falsos positivos 22](#_2n1z2r8i5h7x)

[**Incidentes 24**](#_lo4w8ndn6bdc)

[Mutillidae no conecta con el servicio de mysql 24](#_5l9wari5zmvi)

[Solución 26](#_1f31hjhpjick)

[**Bibliografía 27**](#_klanqwnbyw90)

# Instalación LAMP

sudo apt-get update;

| mariadb |
| --- |
| **sudo apt-get install mariadb-client mariadb-server -y** |
| **sudo mysql -u root -p** |
| **“CREATE DATABASE mutillidae;”** |
| **CREATE USER 'm\_user'@'%' IDENTIFIED BY ‘m\_pass’';** |
| **GRANT ALL PRIVILEGES ON mutillidae.\* to m\_user@’%’;** |
| **FLUSH PRIVILEGES; quit** |
| php |
| **sudo apt-get install php; php -v** |
|  |
| **sudo apt-get install php8.1-mysql** |
| Apache |
| **sudo apt-get install apache2;** |
| 1 |
| **sudo apt-get install libapache2-mod-php8.1** |
|  |
| Prueba del servidor |
| **sudo touch /var/www/html/miprueba.php** |
| **sudo vim /var/www/html/miprueba.php** |
| Escribimos algo de código php “<?php phpinfo();?> |
| En el buscador **localhost/miprueba.php** |
|  |
| phpmyadmin |
| Sudo apt-get install phpmyadmin |
|  |
| \*\*Espacio + ok |
| Usamos el configurador automático de phpmyadmin |
|  |
| \*\*pepe1234º |
|  |
| <https://www.elegantthemes.com/blog/resources/a-quick-guide-to-phpmyadmin-and-how-you-can-use-it> |
| Para la práctica es necesario que el siguiente enlace simbólico se encuentre funcionando |
| **ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-enabled/phpmyadmin.conf** |
|  |
| OpenLdap |
| LDAP (Lightweight Directory Access Protocol) es un protocolo de aplicación utilizado para acceder y mantener servicios de directorio. Un servicio de directorio es un repositorio de información organizado de manera jerárquica que almacena y recupera datos de manera eficiente. LDAP facilita la búsqueda, consulta y actualización de información en estos servicios de directorio.  Conceptos clave relacionados con LDAP:   1. **Directorio:** Es un conjunto organizado de información, similar a una base de datos, pero diseñado para una búsqueda y recuperación eficientes de datos. Los servicios de directorio se utilizan comúnmente para almacenar información de usuarios, como nombres, contraseñas, direcciones de correo electrónico, etc. 2. **Entrada (Entry):** Es un elemento individual en un directorio LDAP, que contiene información sobre un objeto específico (por ejemplo, un usuario o un recurso). 3. **DN (Distinguished Name):** Es una cadena única que identifica de manera única una entrada en un directorio LDAP. La DN refleja la posición de la entrada en la jerarquía del directorio. 4. **Atributo:** Es una propiedad o característica de una entrada en un directorio. Por ejemplo, un usuario puede tener atributos como nombre, dirección, número de teléfono, etc. 5. **Operaciones LDAP:**    * **Bind:** Autenticación en el servidor LDAP.    * **Search:** Búsqueda de información en el directorio.    * **Add:** Agregar una nueva entrada al directorio.    * **Modify:** Modificar los atributos de una entrada existente.    * **Delete:** Eliminar una entrada del directorio. |
| Instalación de LDAP |
| **sudo apt-get install slapd ldap-utils -y** |
|  |
| \*\*pepe1234 |
| Configuración |
| **sudo dpkg-reconfigure slapd** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| Mutillidae |
| **cd /var/www/html**  **git clone** [**https://github.com/webpwnized/mutillidae.git**](https://github.com/webpwnized/mutillidae.git) |
| \*\*Already installed aptitudes after the php installation step |
| **sudo apt-get install php8.1-curl php8.1-mbstring php8.1-xml** |
| \*\* [incidente](#_lo4w8ndn6bdc) |
|  |
| Ataques sin Modsecurity |
| Ver configuración de php (phpinfo) |
| **OWASP 2017 → A9 - Using Components with known Vulnerabilities -> php info page** |
|  |
| SQL Injection |
| Buscamos un formulario y probamos añadiendo una comilla (‘). Esto rompe las sentencias SQL asociadas a este input. |
|  |
|  |
|  |
| Añadamos en name (‘**or 1=1–- ),**  incluyendo el espacio al final.  Buscar un nombre de usuario que sea igual a nada O donde 1 sea igual a 1. Por lo tanto, creamos una condición que siempre es verdadera (OR 1=1). La cadena "-- " es un comentario en SQL. |
| Al comentar el resto de la consulta SQL (AND password=''), lo que elimina la autenticación de contraseña. |
|  |
| Se toma el primer usuario de la tabla sql si la inyección se deja vacía, en este caso, con **admin**. |
| Insert injection |
|  |
| Editando la url podemos acceder a los archivos del sistema, e interpretando los mensajes de error o la ausencia de ellos, enumerar los directorios y archivos. |
|  |
|  |
| Cross site Script (XSS) Injection |
| Simulamos un inicio de sesión. Usamos credenciales de mutillidae |
| Nombre: **samurai** |
| Contraseña: **samurai** |
|  |
|  |
|  |
| Probamos a lanzar un script de javascript |
| **<script> alert(“ivan\_ra”) </script>** |
|  |
| Ataques con Modsecurity |
| Ver configuración de php (phpinfo) |
| **OWASP 2017 → A9 - Using Components with known Vulnerabilities -> php info page** |
|  |
| SQL Injection |
| Buscamos un formulario y probamos añadiendo una comilla (‘). Esto rompe las sentencias SQL asociadas a este input. |
|  |
|  |
|  |
| Insert injection |
|  |
|  |
| Cross site Script (XSS) Injection |
| Simulamos un inicio de sesión |
| Usamos credenciales de mutillidae |
| Nombre: samurai |
| Contraseña: samurai |
|  |
|  |
|  |
|  |

| Modsecurity |
| --- |
| Instalación |
| **sudo apt install libapache2-mod-security2** |
| **sudo a2enmod security2** |
| **sudo systemctl restart apache2** |
| Configuración |
| Comprobar que se puede añadir configuraciones |
| sudo vim /etc/apache2/mods-enabled/security2.conf |
|  |
| Esto significa que Apache incluirá todos los \*.conf archivos en /etc/modsecurity/ directorio. Necesitamos cambiar el nombre del modsecurity.conf-recommended archivo para que funcione. |
| 1. Cambiar el nombre del archivo para que lo reconozca |
| **sudo mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf** |
| 1. Editar **/etc/modsecurity/modsecurity.conf** |
| 1. SecRuleEngine DetectionOnly → SecRuleEngine On 2. SecAuditLogParts ABDEFHIJZ → SecAuditLogParts ABCEFHJKZ |
| **systemctl restart apache2** |
| Definición de reglas OWASP (CRS) |
| Para asegurar la protección de sus aplicaciones web con ModSecurity, es fundamental establecer reglas que detecten y bloqueen a posibles actores maliciosos. Para aquellos que se están iniciando, resulta recomendable instalar conjuntos de reglas preexistentes. Esto permite comenzar rápidamente y, posteriormente, profundizar en el tema. Existen varias opciones gratuitas de conjuntos de reglas para ModSecurity, entre las cuales destaca el Conjunto de Reglas Básicas de OWASP (CRS), que es ampliamente reconocido y utilizado como el estándar en la configuración de ModSecurity. |
| **wget https://github.com/coreruleset/coreruleset/archive/v3.3.0.tar.gz** |
| **tar xvf v3.3.0.tar.gz** |
| **sudo mkdir /etc/apache2/modsecurity-crs/** |
| **sudo mv coreruleset-3.3.0/ /etc/apache2/modsecurity-crs/** |
| **cd /etc/apache2/modsecurity-crs/coreruleset-3.3.0/** |
| **sudo mv crs-setup.conf.example crs-setup.conf** |
| **vim /etc/apache2/mods-enabled/security2.conf** |
| Cambiar la siguiente línea para que se utilice la última versión de OWASP CRS. |
| IncludeOptional /usr/share/modsecurity-crs/\*.load →  IncludeOptional **/etc/apache2/modsecurity-crs/coreruleset-3.3.0/crs-setup.conf**  IncludeOptional **/etc/apache2/modsecurity-crs/coreruleset-3.3.0/rules/\*.conf** |
| **sudo apache2ctl -t** |
| **sudo systemctl restart apache2** |
| A partir de este punto, ya Mutillidae es mucho más restrictivo. Encontraremos controladas gran parte de las vulnerabilidades anteriormente expuestas. |
| Entendiendo OWASP CRS (Core Rule Set) |
| OWASP CRS puede funcionar de dos modos, |
| **Modo autónomo.** 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.  **Modo de puntuación de anomalías**. 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. |
| Los bloqueos se plasman en el log: |
| **/var/log/apache2/modsec\_audit.log** |
| Registros de 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 solicitud * 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) |
| Tipos:   * De **serie**: un archivo para todos los registros. Este es el predeterminado. * **Concurrente**: varios archivos para el registro. 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: reservado * sección E: organismo de respuesta intermedio * sección F: encabezados de respuesta final * sección G: reservado * sección H: tráiler del registro de auditoría * sección I: alternativa de cuerpo de solicitud compacto, que excluye archivos * sección J: información sobre archivos cargados * sección K: cada regla que coincide con un evento, en orden de coincidencia * sección Z: límite final |
| Manejo de falsos positivos |
| ModSecurity es un firewall de aplicaciones web genérico y no está diseñado para una aplicación web específica. El conjunto de reglas básicas de OWASP también es un conjunto de reglas genéricas sin una aplicación en particular en mente, por lo que es probable que vea falsos positivos después de habilitar ModSecurity y OWASP CRS. Si aumenta el nivel de paranoia en el CRS, habrá más falsos positivos. |
| Exclusiones de reglas específicas de la aplicación  Hay algunas exclusiones predefinidas específicas de la aplicación que se envían con OWASP CRS. Edite el crs-setup.conf archivo.  sudo nano /etc/apache2/modsecurity-crs/coreruleset-3.3.0/crs-setup.conf  Ve a la Exclusiones de reglas específicas de la aplicación sección y busque las siguientes líneas.  #SecAction  # "id:900130,  # phase:1,  # nolog,  # pass,  # t:none,  # setvar:tx.crs\_exclusions\_cpanel=1,  # setvar:tx.crs\_exclusions\_drupal=1,  # setvar:tx.crs\_exclusions\_dokuwiki=1,  # setvar:tx.crs\_exclusions\_nextcloud=1,  # setvar:tx.crs\_exclusions\_wordpress=1,  # setvar:tx.crs\_exclusions\_xenforo=1" |
| Por ejemplo, si quiero habilitar las exclusiones de WordPress, las líneas anteriores deben cambiarse a lo siguiente. Tenga cuidado con la sintaxis. No debe haber comentarios entre t:none, y setvar:tx.crs\_exclusions\_wordpress=1". (La barra invertida El carácter al final indica que la siguiente línea es una continuación de la línea actual.) |
| SecAction  "id:900130,  phase:1,  nolog,  pass,  t:none,  setvar:tx.crs\_exclusions\_wordpress=1"  # setvar:tx.crs\_exclusions\_cpanel=1,  # setvar:tx.crs\_exclusions\_drupal=1,  # setvar:tx.crs\_exclusions\_dokuwiki=1,  # setvar:tx.crs\_exclusions\_nextcloud=1,  # setvar:tx.crs\_exclusions\_xenforo=1" |
| **sudo apache2ctl -t** |
| **sudo systemctl restart apache2** |

# 

| Incidentes |
| --- |
| Mutillidae no conecta con el servicio de mysql |
|  |
|  |
|  |
| Solución |
| 1. Verificar las credenciales |
|  |
| Credenciales por defecto |
|  |
| Credenciales actualizadas |
|  |
| Estado: solucionado. |

# Bibliografía

* [Guia de instalación de mutillidae](https://linuxconfig.org/how-to-install-a-lamp-server-on-debian-9-stretch-linux) II
* [Guia de instalación de mutillidae I](https://docs.google.com/document/d/1cnl3j_6GvNEes_bH6z-LTDj8UusP9si6vZLzyydcxxo/edit)
* [Guía de configuración phpmyadmin](https://www.elegantthemes.com/blog/resources/a-quick-guide-to-phpmyadmin-and-how-you-can-use-it)
* [Repositorio Mutillidae](https://github.com/webpwnized/mutillidae.git)
* [Guía de vulnerabilidades de Mutillidae](https://www.computersecuritystudent.com/SECURITY_TOOLS/MUTILLIDAE/MUTILLIDAE_2511/lesson13/index.html)