UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE CIENCIAS Y SISTEMAS

MAESTRÍA EN GESTIÓN DE LA SEGURIDAD DE LA INFORMACIÓN

CICLO ACADÉMICO 2013-2015

Informe Final de Tesis para optar al Título de

Máster en Gestión de la Seguridad de la Información

ANALISIS COMPARATIVO DE SOLUCIONES

WAF

OPEN SOURCE

Autor:

Ing. Néstor Traña Obando

Tutor: Msc. Reynaldo Castaño

Managua, Nicaragua noviembre 2018

I. AGRADECIMIENTO

A mi Madre: *Por mover el mundo para mí.*

A mi Padre: *Por ser un ejemplo de que la edad no importa.*

A mi Esposa: *Por estar en las buenas y en las malas hasta que la muerte nos separe.*

A mi Hija: *Por ser la luz que ilumina todas mis mañanas y me anima a esforzarme cada día más.*

A mi Hijo: *Por ser el complemento perfecto de nuestra familia.*

II. Índice de contenido

III. Índice de Tablas

IV. Índice de figuras

V. Resumen

Las sucursales en línea de los bancos y micro financieras, las tiendas que ahora ofrecen compras en línea, sitios web del gobierno, entre muchas otras plataformas web que poseen las grandes, medianas y pequeñas empresas en nuestro país, están disponibles para todos sus clientes y también para los atacantes. Ataques ampliamente conocidos como Sql Injection, XSS y otros ataques derivados a partir de la entrada de cadenas de textos especiales, están dirigidos a vulnerabilidades en las propias aplicaciones web y no a nivel de la infraestructura de red, donde un firewall común y un IDS/IPS serían totalmente incapaces de proteger estos sitios contra los diferentes ataques (OWASP, Best Practice, 2008).

Por lo anterior, en esta tesis se logró realizar un análisis comparativo de tres herramientas (Endian, Shadow Daemon, Sophos UTM), las cuales son Open Source de Web Application Firewall (WAF), el mismo logro mostrar un panorama claro y preciso de cuál es la herramienta que mejor se puede adaptar a las necesidades de protección de los sitios o sistemas web.

Dado que para realizar el análisis comparativo de las herramientas WAF no existe actualmente metodologías a seguir, se optó por proponer una metodología para procesar la evaluación de estas herramientas, auxiliado del proyecto Benchmark for Security Automation, que brindó las pruebas para evaluar la velocidad y precisión de las herramientas WAF.

Palabras Claves: Seguridad, Vulnerabilidades, Test de Penetración, waf.

1) Introducción

Los desarrolladores de sitios web, aplicaciones web y las aplicaciones móviles normalmente poseen poca o nada de experiencia en cuanto a seguridad informática a la hora de su diseño y programación, las aplicaciones web son de las más atacadas para intento de conseguir acceso a la red de datos de las instituciones, esto debido a que los sitios web se encuentran expuestos al exterior.

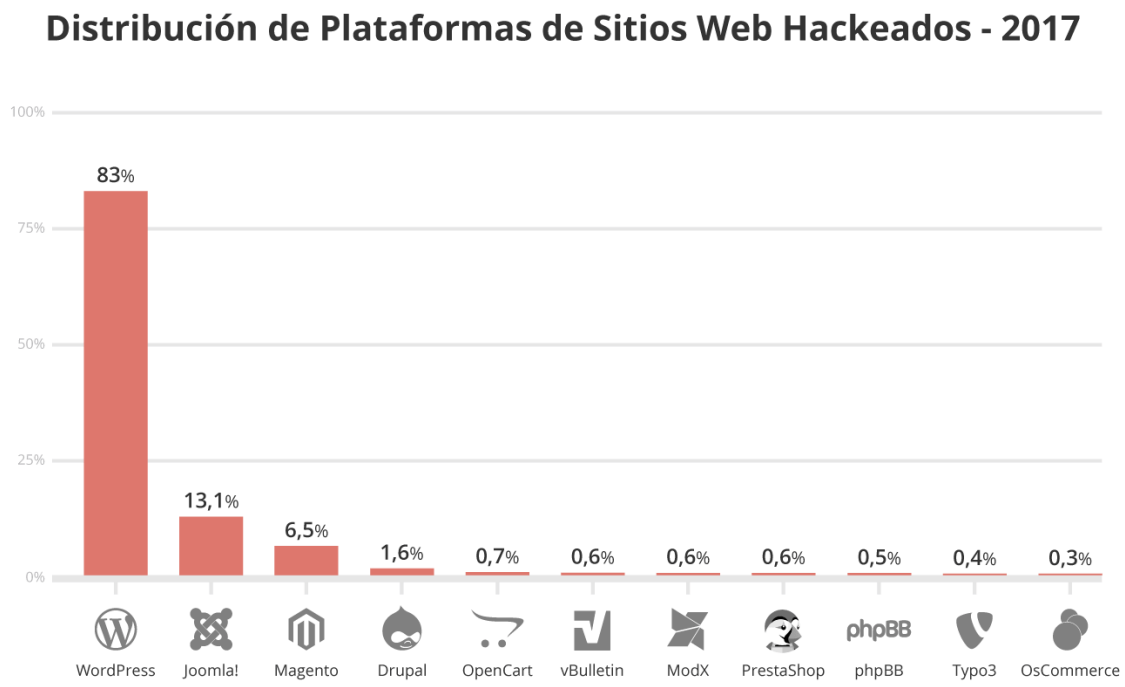


Figura 1 Sitios web hackeados. (2017). Distribución de plataformas de sitios web hackeados 2017. Recuperado de https://sucuri.net

OWASP – Best Practice (2008) define "Web Application Firewall (WAF) como una solución de seguridad en el nivel de aplicación y es uno de los métodos más usados para proteger las aplicaciones web, las cuales se interponen entre la aplicación y el atacante que intenta el ingreso a la misma, explorando el tráfico entrante en la capa de aplicación del modelo de referencia OSI, con harás de

búsqueda de patrones comunes de ataques y denegando el traspaso de tráfico malicioso a la aplicación web" (p.5). Esta solución de software o hardware busca proteger los sitios web de las amenazas o ataques a los que puede estar expuesto.

Todas las empresas u organizaciones deben tomar en cuenta la importancia de la seguridad de sus aplicaciones web para su buen funcionamiento y por ende el correcto flujo de su giro de negocio. Uno de los aspectos sumamente importantes a tener en cuenta es la cantidad de aplicaciones que poseen y que son parte de su área productiva.

El presente estudio propone la realización de un análisis comparativo de soluciones WAF Open Source, en el cual se pretende indagar los pros y contra de este tipo de herramientas, para que se puedan tomar decisiones y así su puesta en marcha e implantación en sistemas web de las distintas organizaciones. A su vez se examinará los principios de detección de estas herramientas, la eficiencia de las mismas y los requerimientos mínimos y óptimos a nivel de hardware como de software.

El trabajo está organizado considerando los siguientes acápites: Definición de la problemática, Justificación, Objetivos, Antecedentes (Marco Teórico) y Metodología.

2) Definición de la Problemática

Los sistemas o aplicaciones web están expuestas a un sinnúmero de amenazas y se debe de buscar la manera de mitigarlas. Los Sistemas de Detección de Intrusos (IDS) o los Sistemas de Prevención de Intrusos (IPS), son una solución externa a la aplicación web que no amerita la modificación del código fuente de la misma para tratar de protegerla. Esos IDS/IPS se implantan en la infraestructura de red y sirven para el monitoreo de eventos y buscando comportamientos anómalos o amenazas en la red que puedan llegar a comprometer la integridad de los sistemas de información.

Los IDS se han aplicado de una forma general, analizando el tráfico de distintos protocolos, tales como TCP, FTP o HTTP. Los WAF se puede decir que son un caso especial de los IDS, ya que se especializan en analizar exclusivamente el tráfico HTTP/HTTPS, con el objetivo de resguardas los sitios o aplicaciones web (OWASP, Best Practice, 2008).

El incremento exponencial de los datos que circulan en la red y la creciente sofisticación de las herramientas de ataques, hacen necesario que las empresas cuenten con un mecanismo de protección que sea efectivo y eficiente. Pero al haber un sinnúmero de herramientas WAF tanto de pago como *open source*[1], se hace complicado seleccionar la que más se adecue a las necesidades de la organización.

[1] Software de código abierto

3) Justificación

La seguridad de la información de las empresas es un tema que cada vez cobra mayor fuerza en el mundo de los negocios, ya que las empresas dependen mayoritariamente del flujo de información, si esta información se distorsiona (por intrusiones no autorizadas) puede afectar significativamente la reputación de la empresa, el giro o resultados del negocio.

Existen personas mal intencionadas (tanto fuera, como dentro de las empresas) que intentan acceder de forma no autorizada a los datos sensibles de las empresas, dicho acceso no autorizado a una red informática, puede ocasionar en su mayoría problemas graves como perdida de información, suplantación de identidad, fraude, falsificaciones, etc., lo cual implica un delito informático[2]. Las empresas sufren incidentes que podrían haberse evitado si los mecanismos de protección hubieran sido reforzados en su momento.

Es común que las organizaciones como parte de su análisis interno indaguen sobre la siguiente interrogante ¿el sitio web de la organización se encuentra preparado para cualquier ataque de hackers? ¡Creo que la respuesta sería “No!”, nadie lo está. Es por esta razón que en este trabajo se hace un análisis de tres herramientas WAF Open Source que ayudan a mitigar los ataques informáticos a los sitios web, pero las organizaciones deben de hacerse otra pregunta: ¿Cuál es la herramienta WAF que debe utilizar la organización para mitigar estos ataques? A su vez el presente trabajo brinda respuesta a esta y otras interrogantes.

Legislación y el Manejo de la Información en la era del conocimiento Managua Nicaragua Noviembre 2005

4) Definición de los Objetivos

4.1) Objetivo General

Determinar la efectividad de los Web Application Firewall Open Source, que faciliten a las organizaciones a tomar decisiones sobre su implementación para proteger sus sistemas web.

4.2) Objetivos Específicos

- ✓ Conocer los mecanismos de ataque que son utilizados a la hora de intentar vulnerar los sitios web para tomar medidas de protección.
- ✓ Comparar las incidencias que tienen las herramientas y métodos de ataques en los sistemas web al no poseer un Web Application Firewall.
- ✓ Demostrar las mitigaciones de los ataques a sitios web con el uso de un Web Application Firewall.
- ✓ Elaborar una comparativa de tres Web Application Firewall Open Source para la toma de decisiones a la hora de implementar el que más se adapte a los sistemas web.

5) Marco Teórico

5.1) ¿Qué es un Web Application Firewall?

En el presente trabajo se abordará WAF como una solución de seguridad en la capa siete del nivel de referencia OSI (Open Systems Interconnection), específicamente en aplicaciones Web. El trabajo se centra en el análisis y evaluación de las funciones de seguridad proporcionados por distintos WAF Open Source. Los aspectos de despliegue dentro de la infraestructura de red de las organizaciones ya existentes, ya sea como un dispositivo de hardware o un complemento de software para un servidor web, no son parte del análisis, pero se abordarán de igual manera.

Cuando se inicia con un proyecto de realización de un sistema web, todo programador y manager del proyecto parte de la primicia que cada aplicación web debe de desarrollarse lo más segura posible, siguiendo alguna metodología o buenas prácticas de programación. Pero las vulnerabilidades de los sitios son detectadas posteriormente en el ciclo de vida y es en este ciclo donde el riesgo de un ataque informático exitoso es mayor. Siendo los WAF una herramienta para diversas medidas de seguridad, nos enfocaremos en la comparativa para protección de los sistemas web.

En la fase de desarrollo, a la hora de realizar el análisis del código fuente, ayudan a detectar y a su vez corregir las vulnerabilidades de los sistemas. A su vez la realización de pruebas de penetración que son llevadas a cabo por expertos que simulan el comportamiento de la aplicación desde un punto externo. Es en todo este contexto que la función principal de un WAF asegura las aplicaciones web contra vulnerabilidades detectadas y con el menor esfuerzo posible, de manera tal que no puedan ser vulneradas por atacantes. Todo esto se convierte en una tarea muy difícil por el alto grado de complejidad de la no tan

típica infraestructura de aplicaciones web: servidores web, servidores de aplicaciones, framework, etc.

El objetivo principal de usar un WAF es asegurar las aplicaciones web existentes, donde los cambios requeridos dentro de la aplicación ya no pueden ser implementados a corto plazo. Esto aplica a las vulnerabilidades encontradas a través de las pruebas de penetración y/o a través del análisis del código fuente.

Los análisis de Test de Penetración sirven también para determinar el nivel de seguridad en: un equipo, en una red de equipos LAN (Local Área Network) o WLAN (Wireless local Área Network), aplicaciones Web entre otros, por medio de ataques informáticos simulados idénticos a los que realizaría un Cracker o Black Hat Hacker pero sin poner en riesgo la información o la disponibilidad de los servicios, esto se hace con el fin de encontrar las posibles amenazas en los sistemas IT antes de que las descubra un atacante (externo o interno).

Los ataques más relevantes a los que pueden estar expuestos los sistemas web son:

- Sql Injection: Aprovecha una vulnerabilidad en la validación de los campos de entradas de datos de un sistema determinado, para realizar operaciones no permitidas sobre la base de dato.
- Inyección de comandos: Aprovecha igualmente una vulnerabilidad en la validación de los campos de entrada del sistema, para ejecutar comandos más allá de los permitidos por la aplicación.
- Cross Site Scripting: Aprovechando la falta de mecanismos de filtrado en las entradas de la aplicación web, esta vulnerabilidad permite el envió de scripts completos, que pueden llevar a la captura de datos del usuario, secuestro de sesiones, etc.
- Denegación de servicio: debido al consumo de ancho de banda o a la sobrecarga de los recursos de computo del sistema atacado, los servicios ofrecidos por éste dejan de ser accesibles a los usuarios y entidades oficiales.

Las soluciones WAF trabajan en la capa de aplicación (capa 7 modelo OSI), interceptando y analizando las peticiones realizadas a los sistemas web, contra un sistema de reglas, con el objeto de impedir que una petición maliciosa llegue al sistema (IpSecSiverSecurit, 2017).

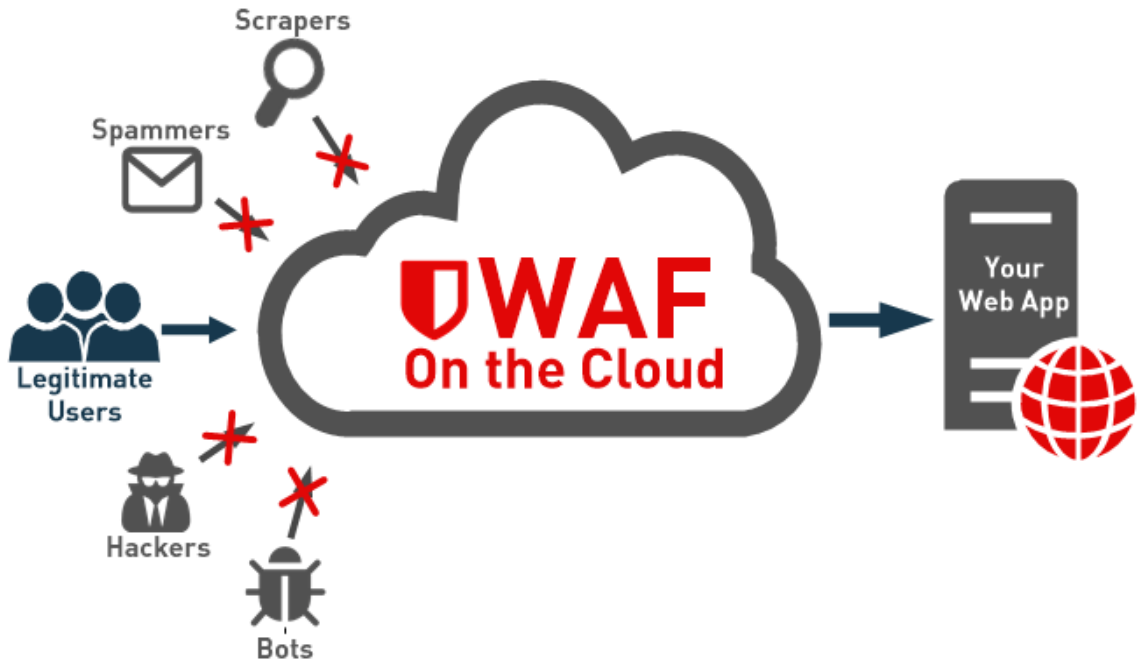


Figura 2 WAF para la nube. (2017). Aplicación para protección de la web. Recuperado de https://www.wizlynxgroup.com

Básicamente, todas las soluciones WAF funcionan de manera similar: actúan como un servicio intermediario entre la aplicación de su sitio web y el visitante que navega por su sitio web, interceptando y eliminando solicitudes maliciosas antes de que puedan causar daños. La diferencia real viene en cómo se despliegan dentro de la arquitectura de la web. Las tecnologías WAF no pretenden reemplazar los controles existentes, sino complementarlos.

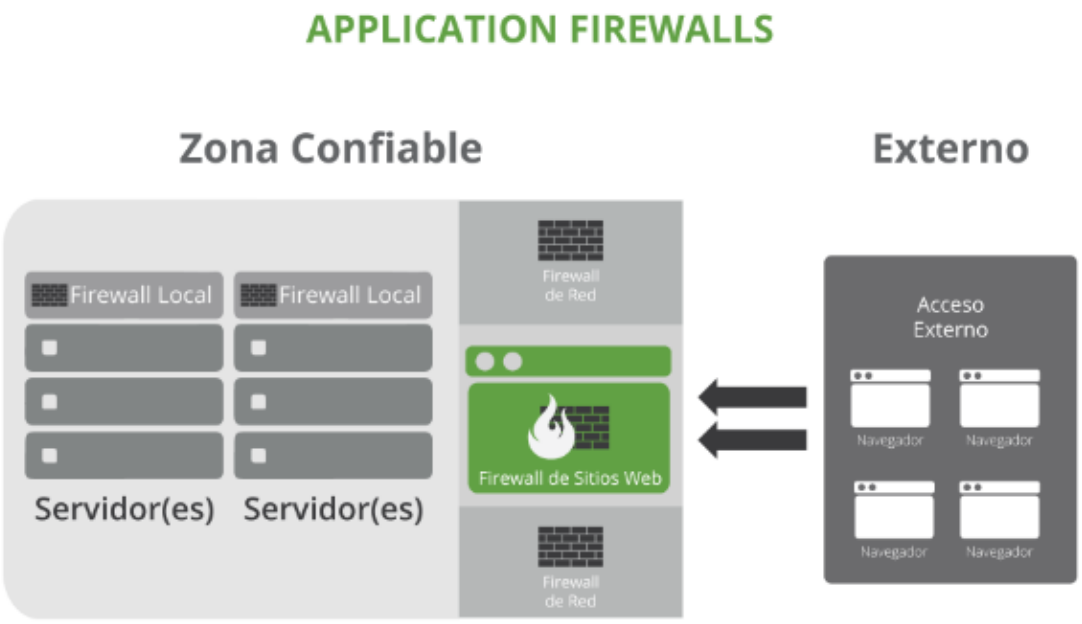


Figura 3 Firewall para Aplicaciones, enero 2018, recuperado de https://blog.sucuri.net

5.2) Principales razones para usar WAF

1. El principal beneficio es la protección posterior al desarrollo de los sistemas web completos y productivos, con una cantidad razonable de esfuerzo y sin tener que cambiar la aplicación en si.

2. WAF ofrece una protección básica contra ataques conocidos a los sistemas web y está basado en estándares de seguridad internacionales.

3. Un WAF es particularmente importante para asegurar aplicaciones web en producción que a su vez derivan otros módulos y que no pueden ser cambiados o modificados rápidamente por la organización, por ejemplo, aquellas aplicaciones poco documentadas o productos de terceros. Un WAF es la única opción para cerrar rápidamente vulnerabilidades externas.

5.3) Principales riesgos al usar WAF

1. Se debe de tener en cuenta que el uso de WAF requiere de cambios en la infraestructura de red.

2. Falsos positivos que pueden llegar a tener un impacto en el negocio.

3. Capacitación sobre el uso y manejo de un WAF al personal encargado

4. Cada versión del sistema web debe de contar con un WAF

Existen tres opciones de despliegue para los WAFs:

- **In-line Appliance Firewalls** – Se despliegan en la red de su organización, tradicionalmente dentro de la DMZ.

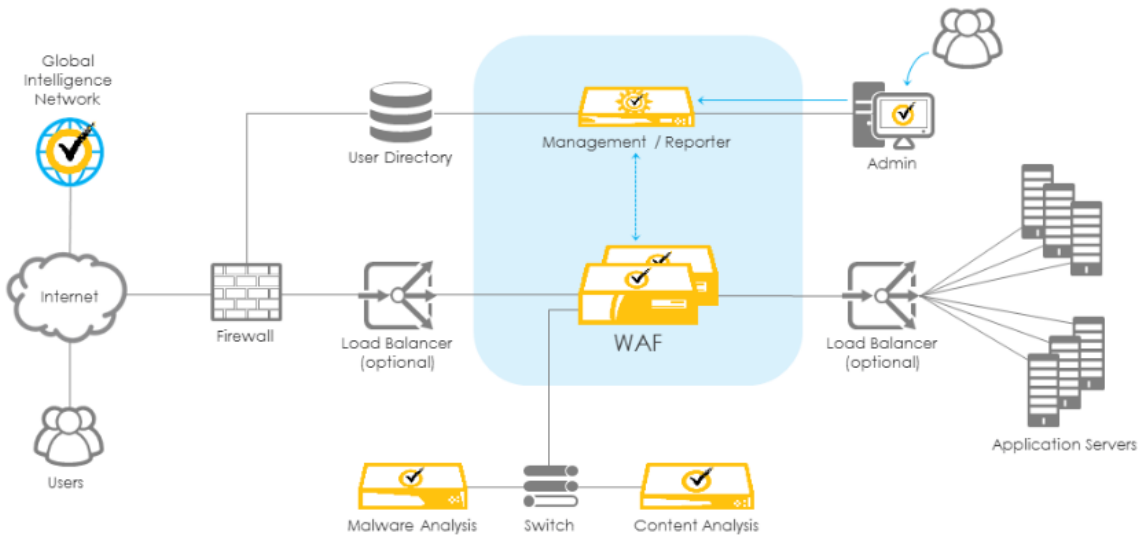


Figura 4 WAF de aplicacion en una DMZ, recuperado de https://origin-symwisedownload.symantec.com

- **End-point Firewalls** – Se despliegan dentro del servidor de alojamiento. Se pueden implementar en diferentes niveles: en el sistema operativo (por ejemplo, IDS/IPS), en el servidor web (es decir, Apache) y en la aplicación (por ejemplo, WordPress, Drupal).

Figura 5 WAF como endpoint, recuperado de https://www.pandasecurity.com

- **Cloud-based Firewalls** – Se despliegan en la nube, fuera de su entorno de alojamiento.

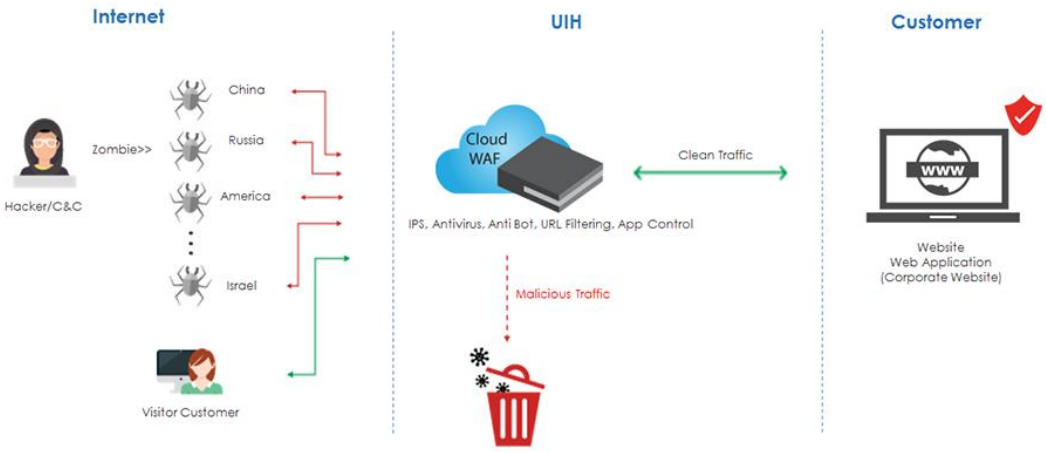


Figura 6 WAF basado en la nube, recuperado de https://www.uih.co.th

5.4) Soluciones WAF Open Source

5.4.1) ModSecurity

ModSecurity es un módulo de firewall de aplicaciones web multiplataforma (WAF) de código abierto. Conocido como el "cuchillo de ejército suizo" de los WAF, permite a los defensores de las aplicaciones web obtener visibilidad del tráfico HTTP(S) y proporciona un lenguaje de reglas de potencia y API para implementar protecciones avanzadas (modsecurity.org, 2017).

Figura 7 ModSecurity, recuperado de https://www.modsecurity.org/

ModSecurity posee a su vez un conjunto de herramientas para la supervisión, el registro y el control de acceso de aplicaciones web en tiempo real. Se puede decir que es un facilitador: ya que depende de la persona administradoras elegir las configuraciones disponibles.

El código abierto de ModSecurity da la libertad de elegir que hacer y es la parte central de su identidad por ser de naturaleza de código abierto. Teniendo pleno acceso al código fuente, brinda la capacidad de personalizar y ampliar la herramienta para que se adapte a las necesidades de cualquier entorno.

Dentro de las grandes funciones que brinda ModSecurity se encuentran:

5.4.1.1) Control de acceso y control de seguridad de aplicaciones en tiempo real.

En esencia, ModSecurity le da acceso a la secuencia de tráfico HTTP, en tiempo real, junto con la capacidad de inspeccionarlo. Esto es suficiente para la supervisión de seguridad en tiempo real. Hay una dimensión adicional de lo que es posible a través del mecanismo de almacenamiento persistente de ModSecurity, que le permite rastrear los elementos del sistema a lo largo del tiempo y realizar la correlación de eventos. Puede bloquear de forma fiable, si lo desea, porque ModSecurity utiliza el búfer completo de solicitud y respuesta.

5.4.1.2) Registro completo de tráfico HTTP

Los servidores web tradicionalmente hacen muy poco cuando se trata de iniciar sesión por razones de seguridad. Registran muy poco de manera predeterminada, e incluso con muchos ajustes no puede obtener todo lo que necesita. ModSecurity brinda la capacidad de registrar todo lo que se necesita, incluidos los datos de transacciones sin procesar, que son esenciales para el análisis forense. Además, puede elegir qué transacciones se registran, qué partes de una transacción se registran y qué partes se desestiman.

5.4.1.3) Evaluación de seguridad pasiva continua

La evaluación de seguridad se ve en gran parte como un evento programado activo, en el cual un equipo independiente se organiza para tratar de realizar un ataque simulado. La evaluación de seguridad pasiva continua es una variación del monitoreo en tiempo real, donde, en lugar de enfocarse en el comportamiento de las partes externas, se enfoca en el comportamiento del sistema mismo. Es un tipo de sistema de alerta temprana que puede detectar rastros de muchas anormalidades y debilidades de seguridad antes de ser explotados.

5.4.1.4) El endurecimiento de aplicaciones web

Uno de los usos más completo de ModSecurity es la reducción de superficie de ataque, en la que se reducen de forma selectiva las características HTTP que está dispuesto a aceptar (por ejemplo, métodos de solicitud, encabezados de solicitud, tipos de contenido, etc.). ModSecurity puede ayudar a imponer muchas restricciones similares, ya sea directamente o mediante la colaboración con otros módulos de Apache. Todos caen bajo el endurecimiento de aplicaciones web. Por ejemplo, es posible corregir muchos problemas de gestión de sesiones, así como vulnerabilidades de falsificación de solicitudes entre sitios.

5.4.1.5) Principios de ModSecurity

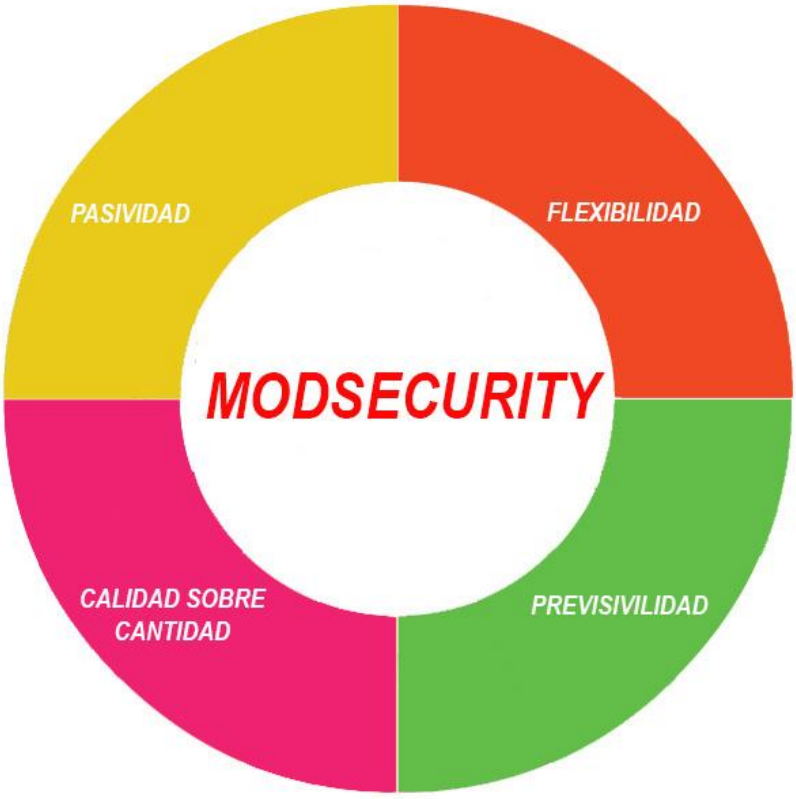


Figura 8 - Principios ModSecurity. Creación Propia

Flexibilidad

ModSecurity logra flexibilidad al brindarle un poderoso lenguaje de reglas, que le permite hacer exactamente lo que necesita, en combinación con la capacidad de aplicar reglas solo donde lo necesite.

Pasividad

ModSecurity tiene cuidado de nunca interactuar con una transacción a menos que se lo indique. ModSecurity le dará mucha información, pero finalmente le dejará las decisiones.

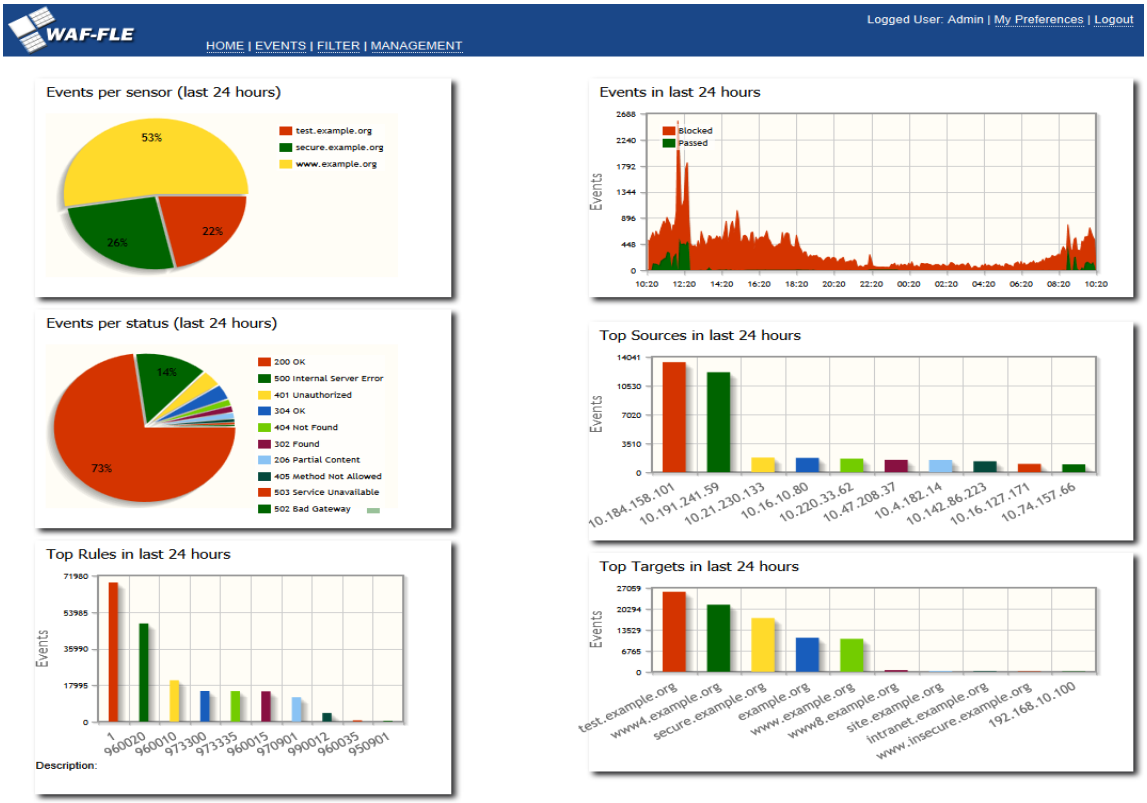
Previsibilidad

No existe una herramienta perfecta, pero una predecible es la siguiente mejor opción. ModSecutity da información de todos los hechos para entender los puntos débiles y tratar de solucionarlos.

Calidad sobre cantidad

ModSecurity entiende que se cuenta con recursos limitados disponibles, por lo tanto, deja fuera implementaciones de seguridad a otras herramientas que puedan realizar ese trabajo

De manera general podemos mencionar como una pequeña desventaja que ModSecurity no posee una interfaz gráfica de usuario que facilite su configuración y seguimientos de los logs del sistema, por lo tanto, este trabajo debe de ser realizado desde el terminal. Pero esta desventaja es minimizada por diferentes soluciones (también open source) que brindan alternativas gráficas para la gestión de ModSecurity, entre estas alternativas se puede mencionar a WAF-LFE Project (waf-fle.org, 2017)



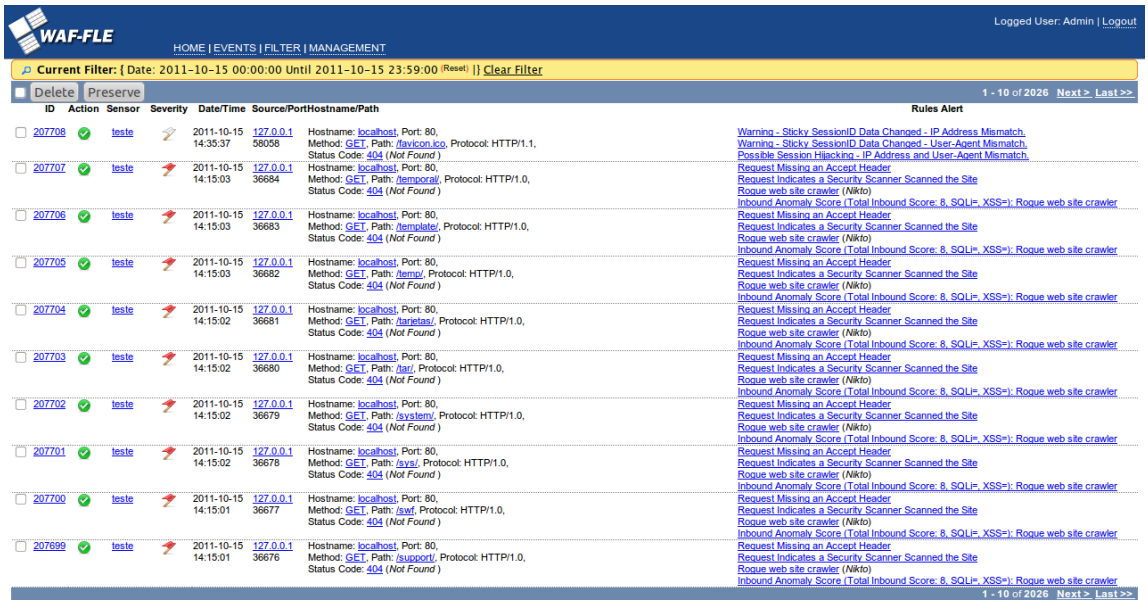


Figura 10 - Gestión de Eventos ModSecurity (2017)

5.4.2) Shadow Daemon

Shadow Daemon es una colección de herramientas para detectar, registrar y prevenir ataques en aplicaciones web. Técnicamente hablando, Shadow Daemon es un firewall de aplicaciones web que intercepta solicitudes y filtra parámetros maliciosos. Es un sistema modular que separa la aplicación web, el análisis y la interfaz para aumentar la seguridad, la flexibilidad y la capacidad de expansión. (shadowd.zecure.org , 2017)

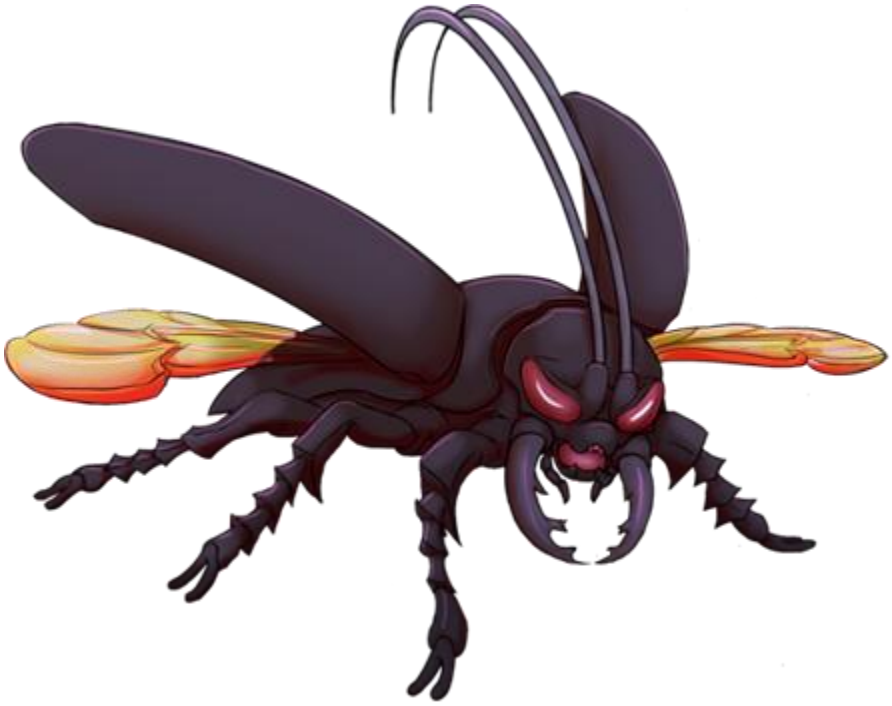


Figura 11 Logo Shadow Daemon, recuperado sitio ofocial https://shadowd.zecure.org

Posee las siguientes características

5.4.2.1) Detección precisa

Shadow Daemon combina lista negra, listas blancas y verificación de integridad para detectar con precisión solicitudes maliciosas. La lista negra hace uso de expresiones regulares sofisticadas para buscar patrones de ataque conocidos en la entrada del usuario. La lista blanca, por otro lado, busca irregularidades en la entrada del usuario basadas en reglas estrictas que definen cómo debería verse la entrada. La comprobación de integridad compara las sumas de comprobación criptográficamente seguras de las secuencias de comandos ejecutadas con los valores predefinidos.

Shadow Daemon es capaz de detectar ataques comunes como:

- ✓ Inyecciones de SQL
- ✓ Inyecciones XML
- ✓ Inyecciones de código

- ✓ Inyecciones de comando
- ✓ Cross-site scripting
- ✓ Inclusiones de archivos locales / remotos
- ✓ Acceso de puerta trasera

5.4.2.2) Protección discreta

A diferencia de muchos otros firewalls de aplicaciones web, Shadow Daemon no bloquea completamente las solicitudes maliciosas si es posible. En cambio, solo filtra las partes peligrosas de una solicitud y permite que continúe después. Esto hace que los ataques sean imposibles, pero no frustra innecesariamente a los visitantes en el caso de falsos positivos.

5.4.2.3) Arquitectura segura

Shadow Daemon está más cerca de la aplicación que la mayoría de los firewalls de otras aplicaciones web. Recibe exactamente la misma entrada que recibe la aplicación web y, por lo tanto, es casi imposible eludir la detección ofuscando el ataque. Sin embargo, las partes más complejas de Shadow Daemon están separadas de la aplicación web para garantizar un cierto nivel de seguridad.

También trabaja teniendo en cuenta los siguientes sistemas de reglas:

Blacklisting: Utiliza expresiones regulares para buscar patrones de ataques conocidos en el trafico de red. La prioridad de una regla se determina por la probabilidad que un ataque tenga éxito o no. Se le brinda puntuación a un alto riesgo de falsos positivos. Si varias reglas se superponen, entonces se penaliza la puntuación resultante, todo esto dependerá cuán grandes sean las probabilidades de detectar el mismo patrón de ataque más de una vez.

Whitelisting; Explora irregularidades en el tráfico de red, basadas en reglas muy estrictas que definen cómo debe de ser la mima. Inicialmente verifica si el

parámetro de entrada tiene una regla asociada, en caso que no la tenga, entonces es considerado una amenaza. Si existe al menos una regla, el algoritmo tiene la función de comprobar si existe una restricción de longitud y se respecta la restricción. Por último, se comprueba en el conjunto de caracteres de la entrada con la ayuda de expresiones regulares.

Integrity checking: Compara los checksums criptográficos seguros de los scripts ejecutados, realizando un match contra los valores predefinidos. Funciona con un enfoque Whitelisting ya que comprueba primero si el parámetro posee una regla asociada y en caso contrario se considera una amenaza. Pero si existe una regla asociada, entonces el algoritmo es capaz de comprobar si la petición tiene un hash asociado, de ser positivo, se comprueba si el hash coincide con el hash proporcionado por la regla.

El conector se ejecuta cada vez que un cliente solicita un recurso. Establece una conexión TCP con el servidor shadowd y transmite la IP del cliente, la persona que llama, el recurso, la entrada del usuario y las sumas de comprobación. El servidor procesa y analiza los datos con la lista negra, la lista blanca y la verificación de integridad y devuelve los identificadores de entrada peligrosa. El conector usa los identificadores para desactivar todas las amenazas y se carga el recurso solicitado originalmente.

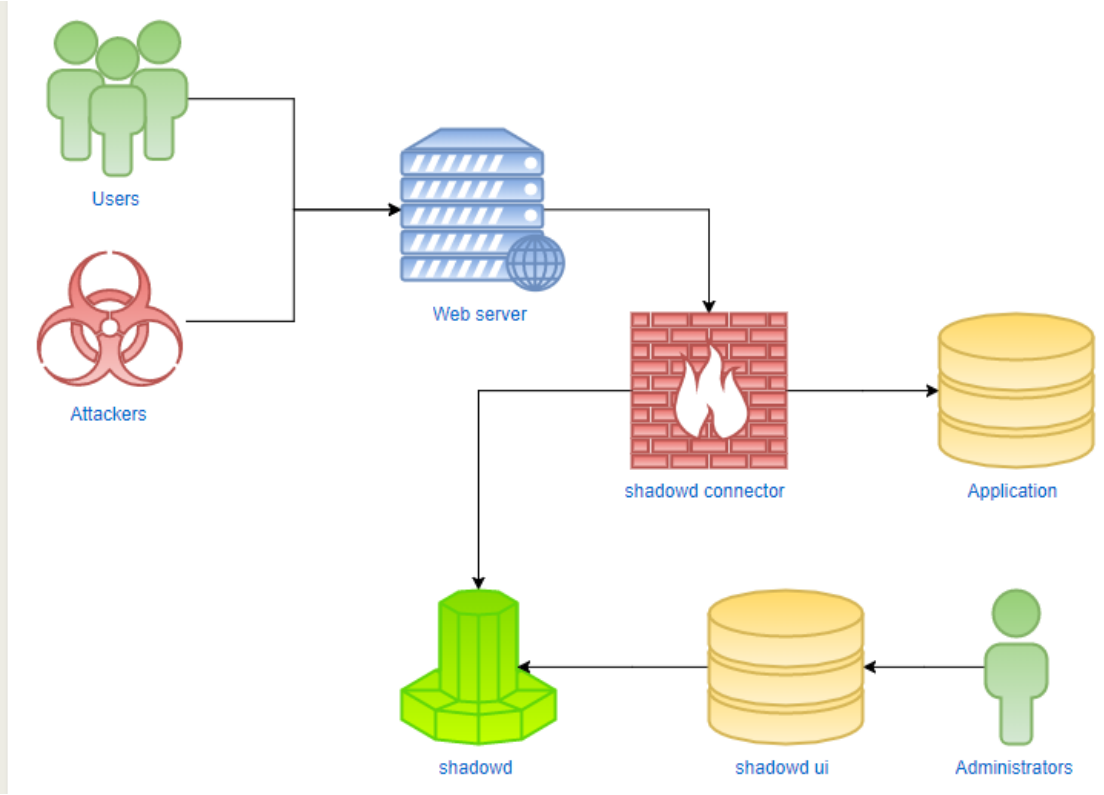
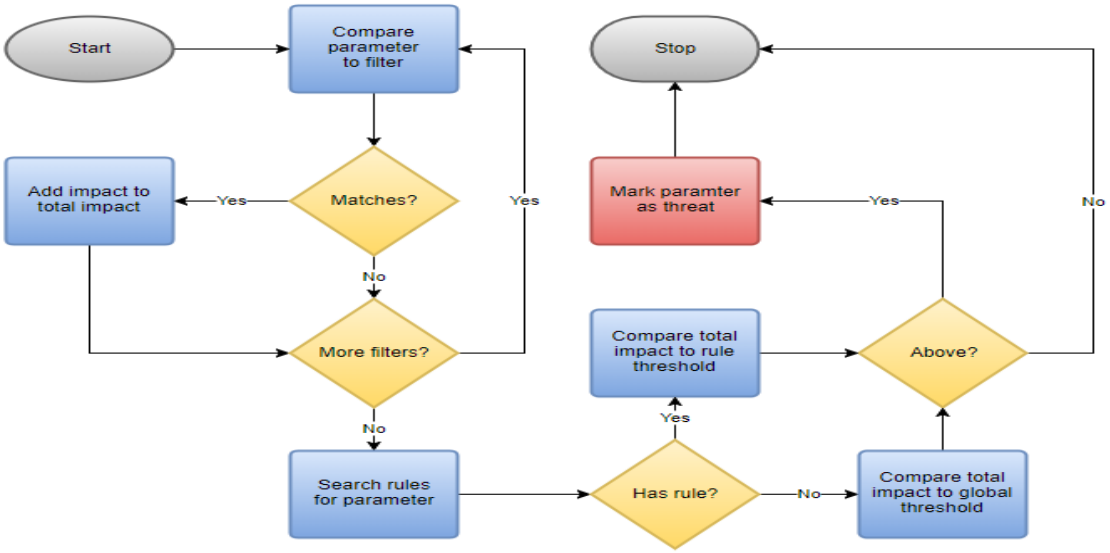


Figura 12 - Arquitectura Shadow Daemon (2017)

El algoritmo de blacklist usa expresiones regulares para identificar patrones de ataque conocidos. Cada filtro tiene un impacto numérico que intenta especificar la peligrosidad y su falta de ambigüedad. Los impactos de todos los filtros coincidentes se agregan y se comparan con un umbral. Si el impacto total es mayor que el umbral, la entrada se clasifica como una amenaza.



El algoritmo de la whitelist es uno de los tres métodos de Shadow Daemon para identificar solicitudes maliciosas. Compara la entrada del usuario a las reglas que especifican cómo debería ser la entrada.

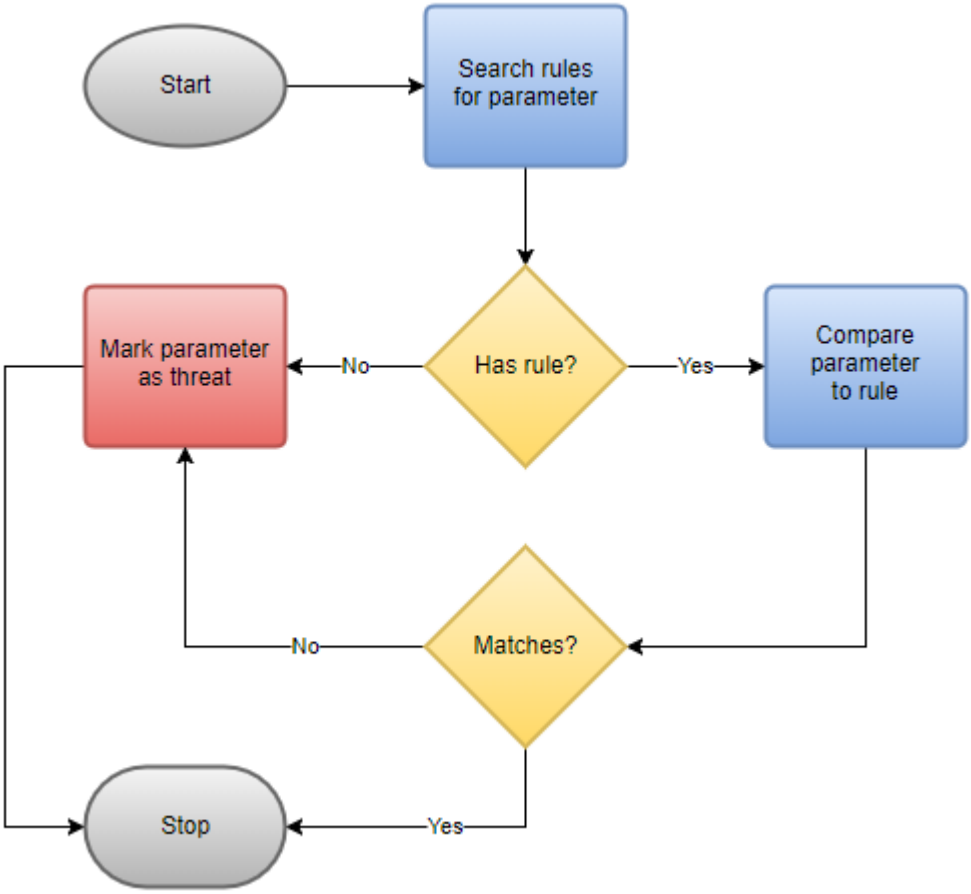


Figura 14 - Algoritmo regla Whitelist Shadow Daemon (2017)

El algoritmo de integridad es uno de los tres métodos de Shadow Daemon para identificar solicitudes maliciosas. Compara las sumas de comprobación criptográficamente seguras del script ejecutado con las reglas que especifican qué deberían ser las sumas de comprobación.

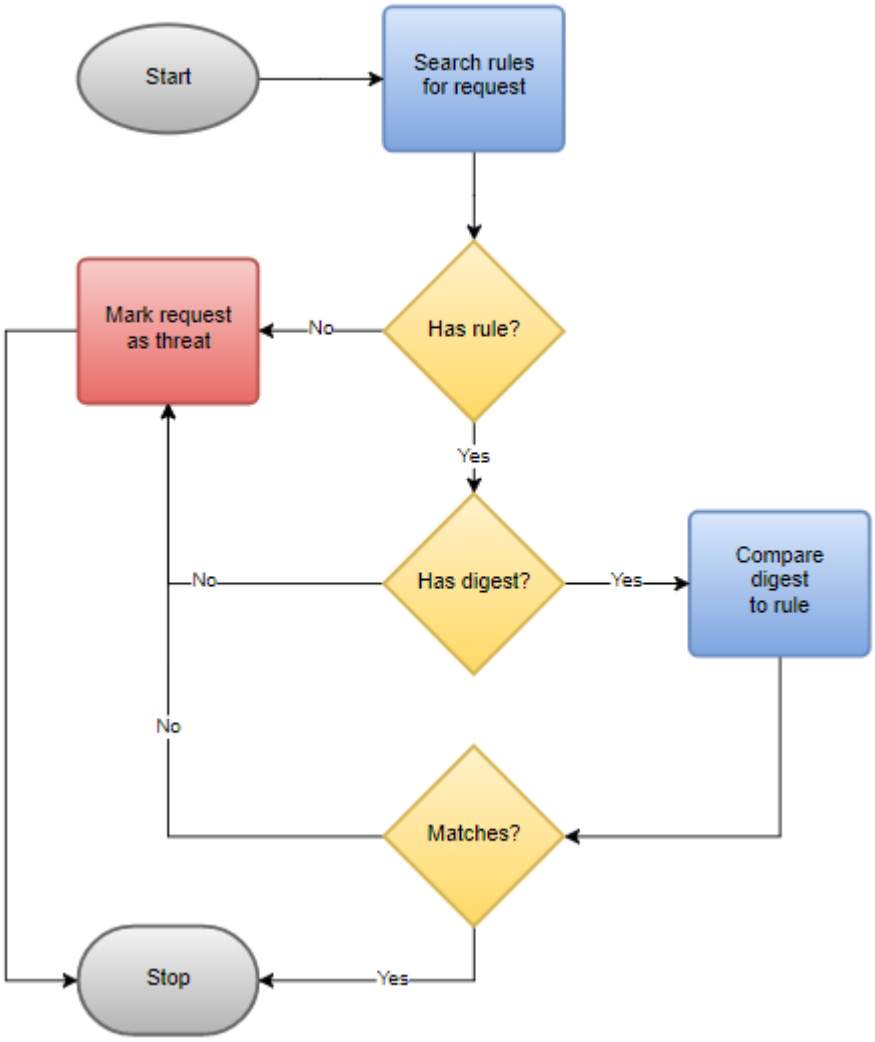


Figura 15 - Algoritmo Integrity Checking Shadow Daemon (2017)

5.4.3) Sophos UTM 9

Es la versión gratuita del Firewall Sophos UTM, con un WAF integrado. La característica Firewall de aplicaciones web (WAF) de Sophos Firewall protege los servidores web implementados en una red y las aplicaciones relacionadas de cualquier vulnerabilidad. Protege las aplicaciones a las que se accede a través de HTTP y HTTPS - Capa de aplicación. Además de los ataques basados en la capa 7, el servidor web está protegido contra la manipulación de cookies, la exploración enérgica y la manipulación oculta en el campo. El WAF también mitiga las vulnerabilidades inducidas por el usuario en las aplicaciones que dejan las aplicaciones web abiertas a los ataques, como las secuencias de comandos entre sitios, el recorrido de directorios y la exploración forzada de URL. (community.sophos.com, 2017)

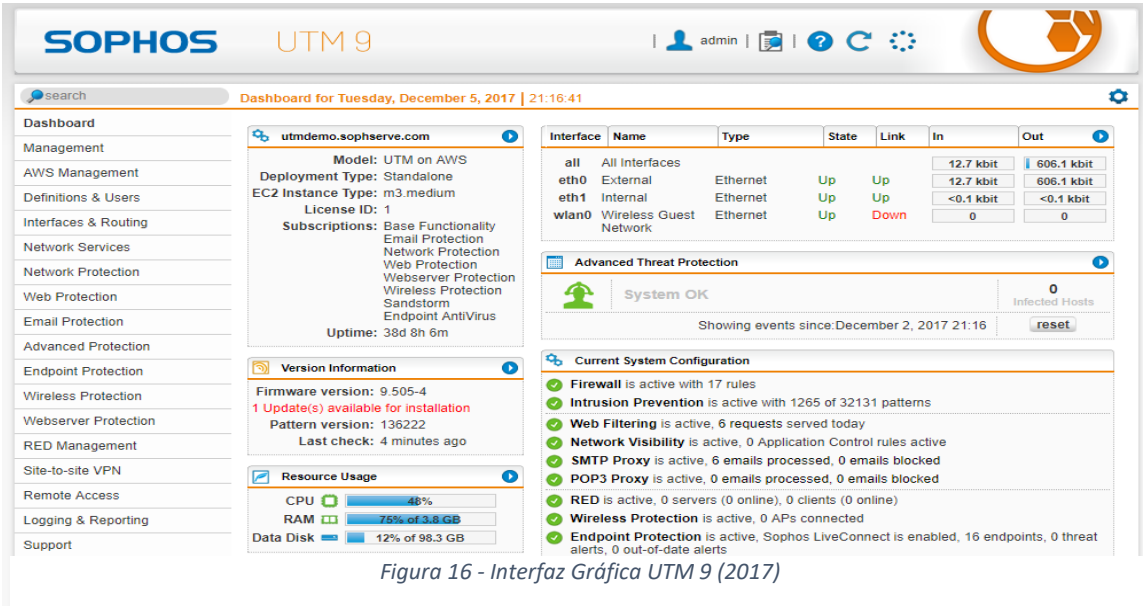


Figura 16 - Interfaz Gráfica UTM 9 (2017)

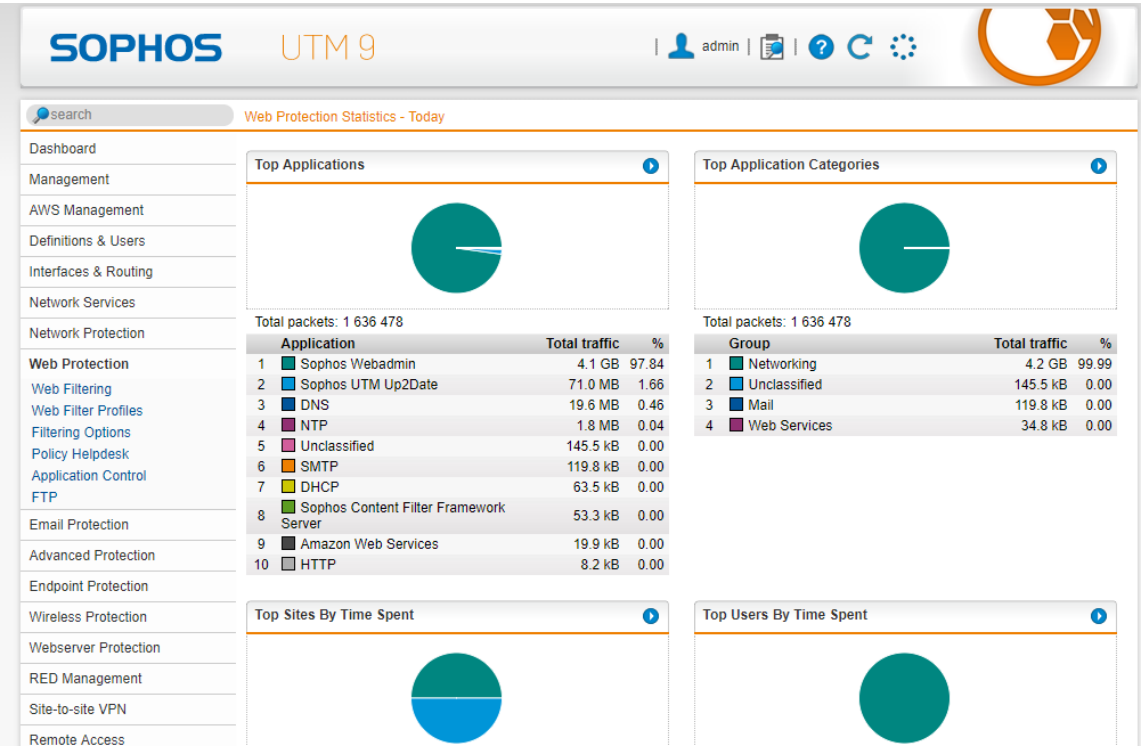


Figura 17 - Interfaz Estadistica Protección Web Sophos UTM9 (2017)

En el proceso de esta investigación, logramos notar que Sophos UTM9 implementa por debajo de su interfaz la solución WAF ModSecurity y aplica todas sus reglas.

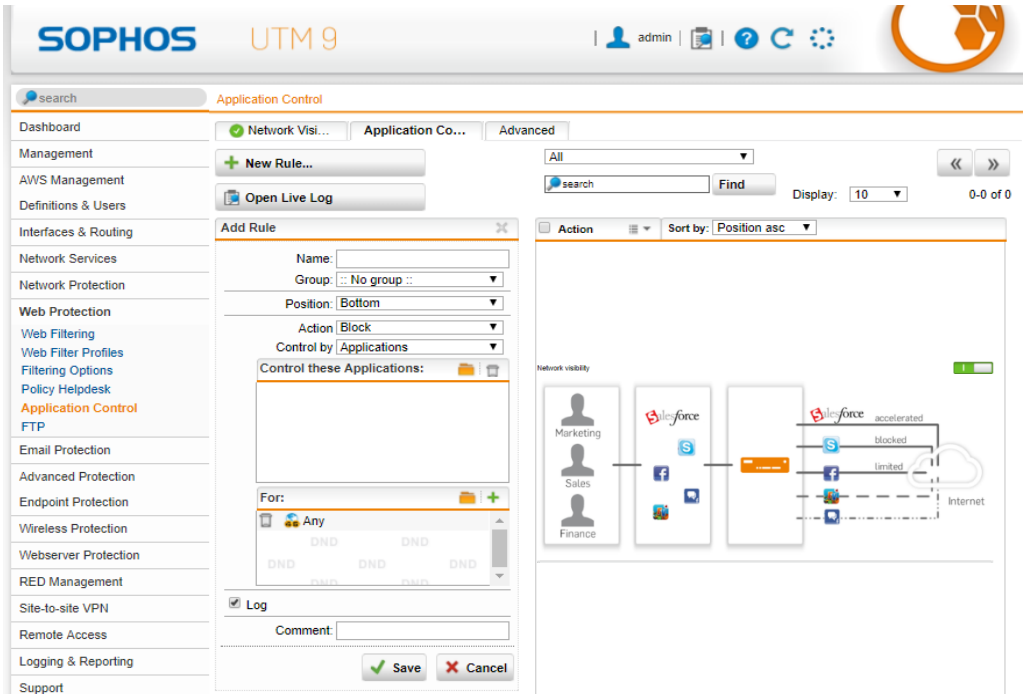


Figura 18 - Aplicación de reglas UTM9 - ModSecurity

5.4.4) Endian UTM

Endian es una distribución OpenSource de Linux, desarrollada para actuar no solamente como Firewall sino como solución integral para proteger una red de amenazas externas, ofreciendo todos los servicios que brinda un UTM (Gestión Unificada de Amenazas) fácil de usar e instalar. (endian.com, 2017)

Además de ser útil para el control de amenazas, también cuenta con características especiales, permitiendo configurar proxys, canales VPN, enrutadores, filtrado de datos, así como herramientas antivirus y anti spam.

Características:

- ✓ Establecimiento de reglas de firewall de entrada y salida
- ✓ Nat (traducción de direcciones de red)
- ✓ Soporte para DNS dinámicos
- ✓ Soporte para DMZ
- ✓ Interfaz de administración web mediante protocolo https
- ✓ Gráficos detallados de las interfaces de red
- ✓ Detalle de todas las conexiones activas
- ✓ Log detallado de todos los procesos del sistema
- ✓ Servidor dhcp
- ✓ Permite implementar comunicaciones seguras con otras sedes o clientes remotos a través de vpn
- ✓ Antivirus y filtrado de contenido para un acceso a internet más seguro

- ✓ Manejo de proxy
- ✓ Enrutamiento
- ✓ Antivirus y anti spam para el correo electrónico
- ✓ Alta disponibilidad

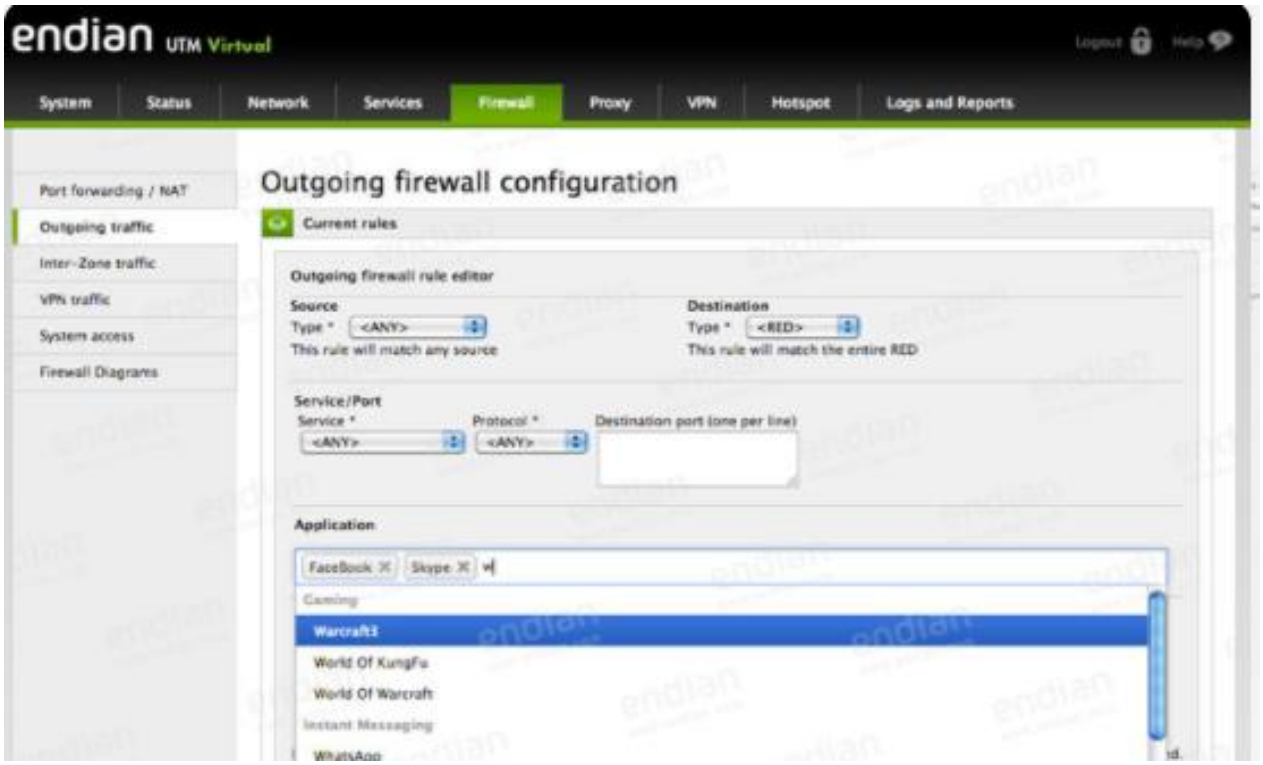


Figura 19 - Interfaz Configuración Firewall Endian (2017)

Figura 20 - Interfaz Log y Reportes Endian (2017)

5.5) Objetivo y metodología de trabajo

El objetivo central del trabajo es realizar un análisis comparativo del nivel de protección que pueden brindar las herramientas WAF frente a distintos tipos de ataques ejecutados contra aplicaciones, sitios o sistemas web.

En la actualidad no existe alguna metodología a seguir pensado específicamente en análisis de soluciones WAF Open Source, por lo tanto, en el presente trabajo se propone una metodología para poder procesar las distintas herramientas a ser analizadas y evaluadas.

1. Se optará por el proyecto OWASP Benchmark para la evaluación del desempeño de las herramientas WAF, específicamente realizando una categoría de vulnerabilidades: Sql Injections, cross-site scripting, DDoS, path traversal.
2. Se seleccionarán al menos tres herramientas WAF Open Source
3. Se montará una infraestructura de red a nivel de prueba que simula una red de datos con sitios web en donde se implantaran los WAF Open Source para su análisis.
4. Se verificará las mejores herramientas para el escaneo de vulnerabilidades y se harán uso de las mismas.
5. Se realizarán análisis de los ataques perpetrados y detectados por las herramientas WAF.
6. Se revisará las alertas generadas por los WAF en indicadores Verdaderos Positivos, Verdaderos Negativos, Falsos Positivos, Falsos Negativos.
7. Análisis de los resultados obtenidos en las pruebas

5.6) Presentación de soluciones a evaluar

Las soluciones WAF propuestas para el análisis son las que se muestran a continuación:

SOLUCIÓN	TECNOLOGIA	LICENCIA
Shadow Daemon	WAF	Open Source
Sophos UTM 9	WAF	Open Source
Endian	WAF	Open Source

Tabla 1 Soluciones WAF a evaluar

5.7) Desarrollo de la metodología

Para el desarrollo de la metodología planteada, se han utilizado tanto maquinas físicas como virtuales para poder recrear un entorno de red en que sea posible realizar las pruebas de una manera controlada.

El OWASP Benchmark for Security Automation (OWASP Benchmark), es un conjunto de pruebas gratuito, diseñado para evaluar la velocidad, cobertura y precisión de las herramientas y servicios de detección de vulnerabilidades de software automatizadas.

Se puede usar OWASP Benchmark con herramientas de prueba de seguridad de aplicaciones estáticas (SAST), herramientas de prueba de seguridad de aplicaciones dinámicas (DAST) como OWASP ZAP y herramientas de prueba interactiva de seguridad de aplicaciones (IAST).

A continuación, se muestran las configuraciones realizadas para la ejecución de las pruebas, detallando la configuración de cada escenario.

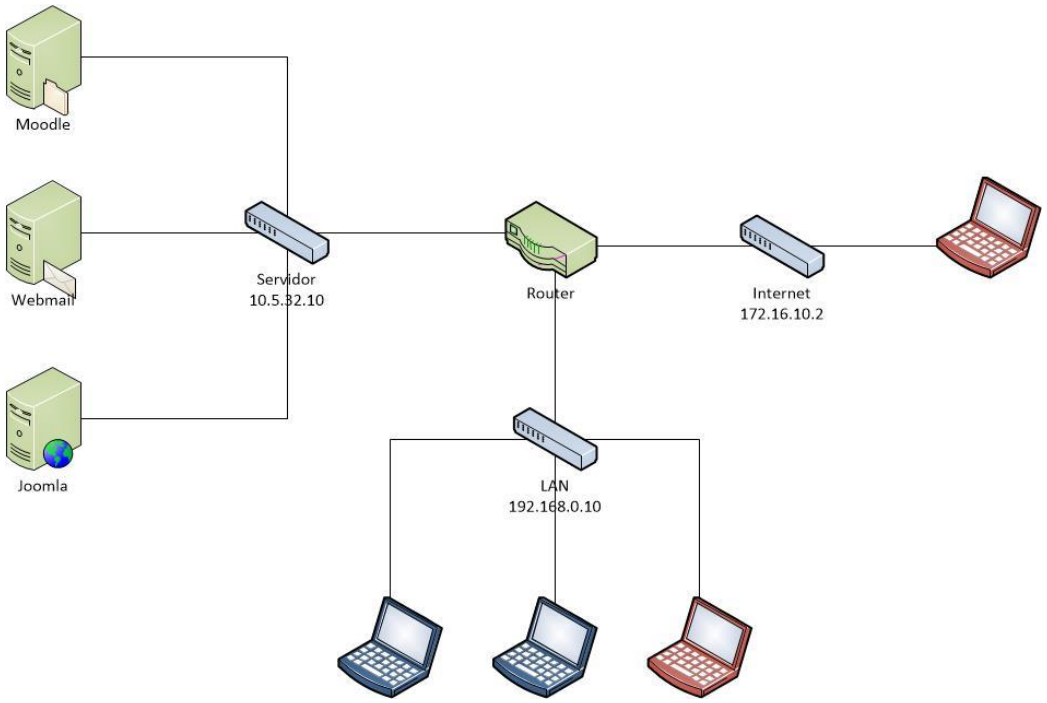


Figura 21 - Configuración entorno de prueba sin protección

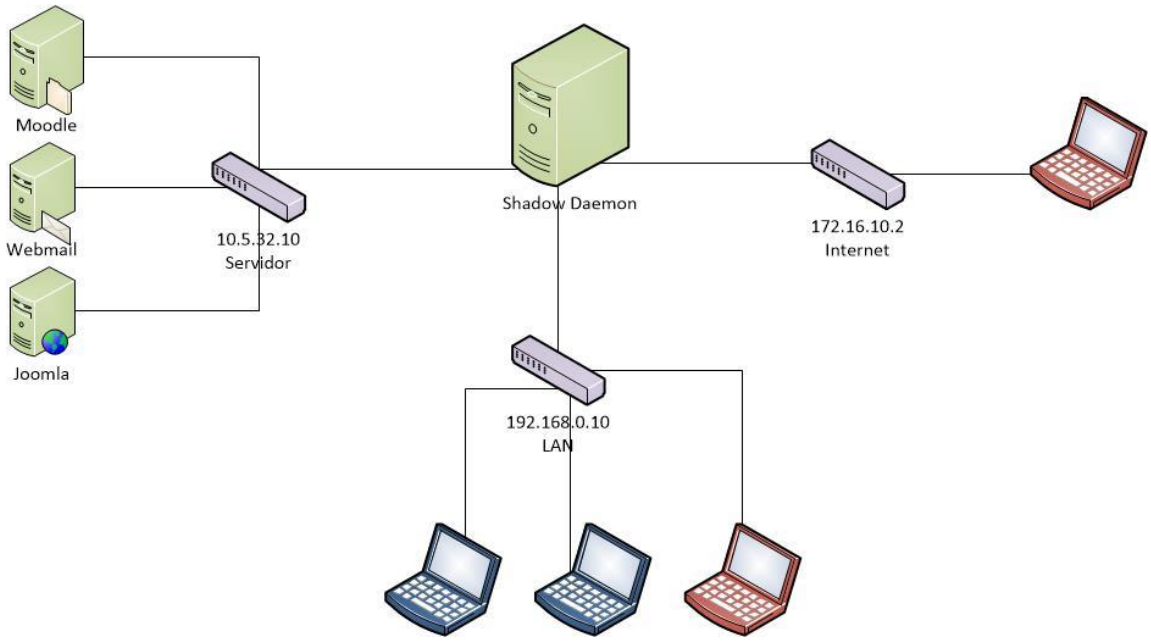


Figura 22 - Configuración entorno de prueba con la protección Shadow Daemon

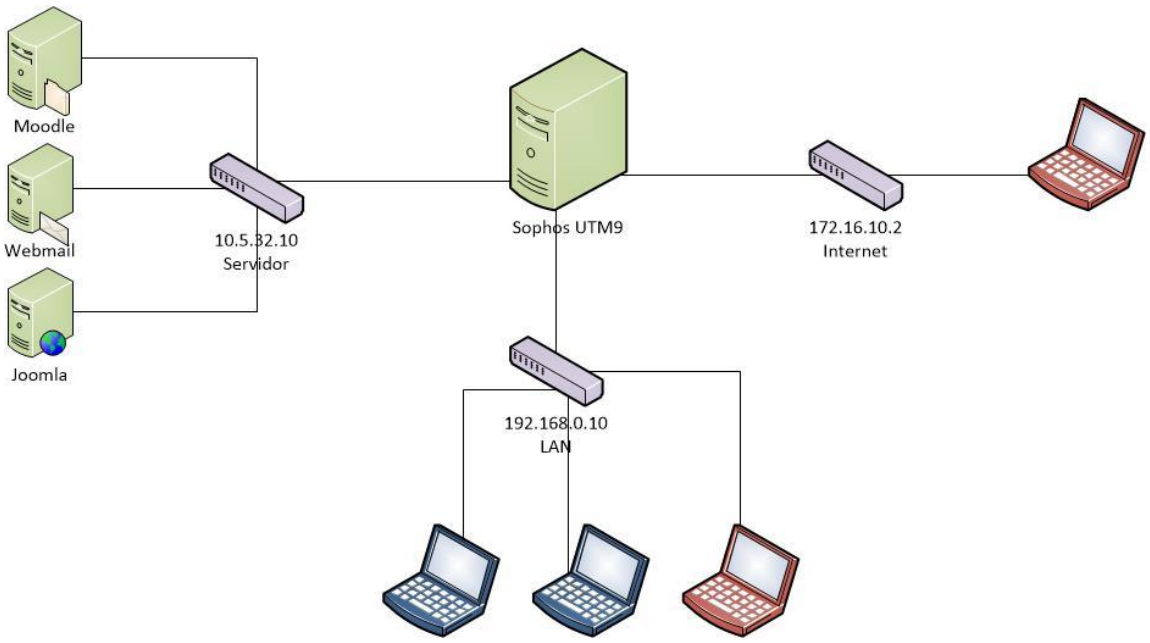


Figura 23 - Configuración entorno de prueba con la protección Sophos UTM9

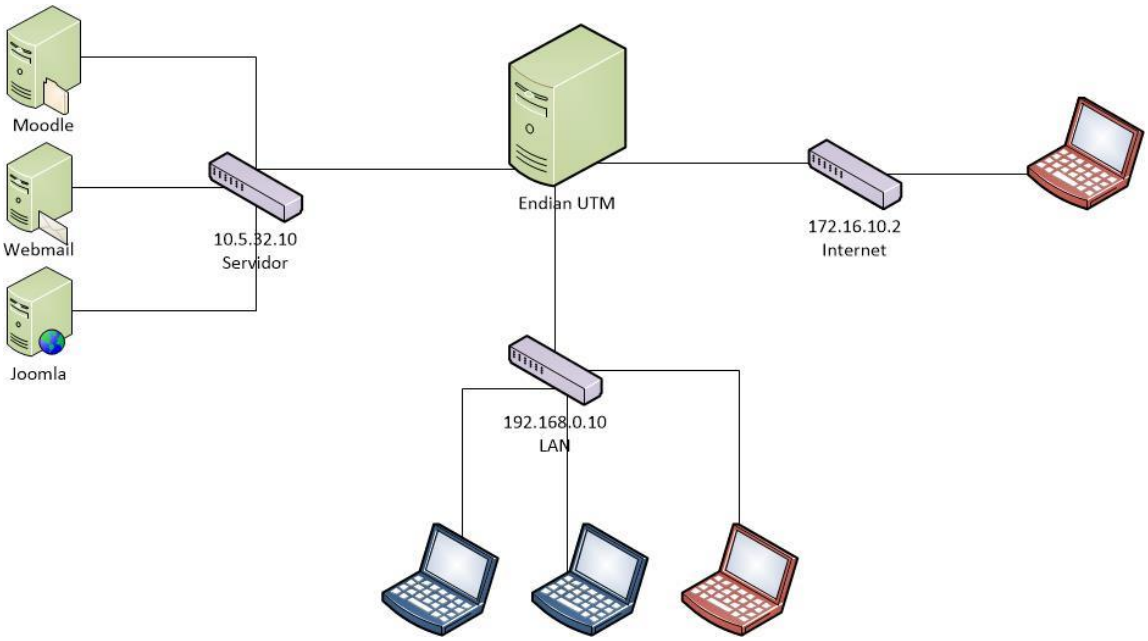


Figura 24 - Configuración entorno de prueba con la protección Endian UTM

5.7.1) Vulnerabilidades analizadas

5.7.1.1) SQL Injection (Inyecciones SQL)

CWE-Mitre [3]la clasifica como CWE-89 (CWE-89: Improper Neutralization of Special Elements used in an SQL Command) (Mitre, 2017)

Un ataque de inyección SQL consiste en la inserción o "inyección" de una consulta SQL a través de los datos de entrada del cliente a la aplicación. Un exploit de inyección SQL exitoso puede leer datos sensibles de la base de datos, modificar datos de base de datos (Insertar / Actualizar / Eliminar), ejecutar operaciones de administración en la base de datos, recuperar el contenido de un archivo dado presente en el archivo DBMS sistema y, en algunos casos, emitir comandos al sistema operativo. (www.owasp.org – SQL Injection, 2017).

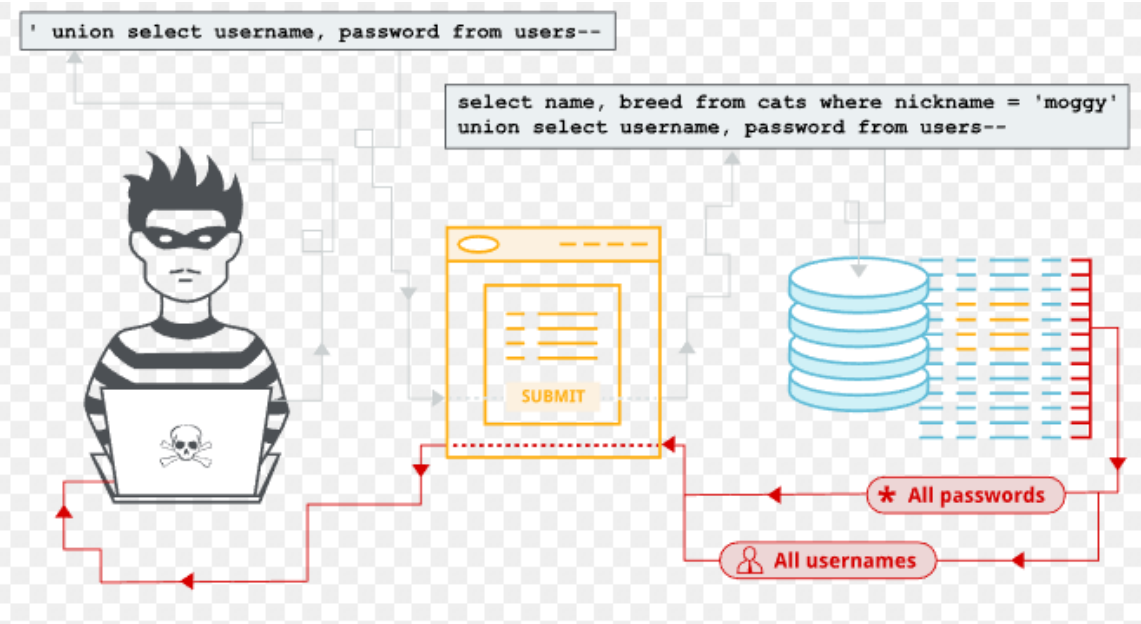


Figura 25 Representación ataque sql injection, recuperado de https://portswigger.net

[3] Common Weakness Enumeration (A community developed List of software weakness type)

Como se observa en la figura 17, la aplicación web al realizar un cambio en la sintaxis de su dirección web (introducción un comodin (‘) de SQL), genera un error de sintaxis y se logra observar que en esa dirección es vulnerable ante ataques de SQL Injection.

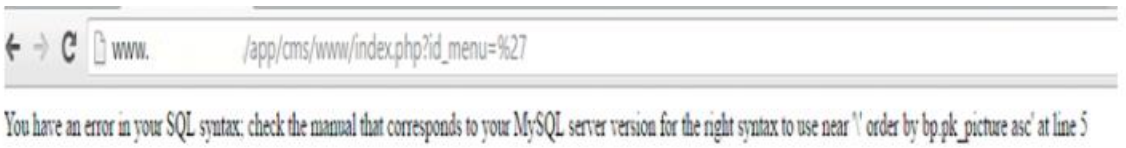


Figura 26 - Sitio web vulnerable a sql injection

5.7.1.2) Path Traversal

CWE-Mitre la clasifica como CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') (Mitre, 2017).

Un ataque Path Traversal (también conocido como recorrido de directorio) tiene como objetivo acceder a los archivos y directorios que se almacenan fuera de la carpeta raíz web. Al manipular variables que hacen referencia a archivos con secuencias "punto-punto-barra (../)" y sus variaciones o mediante el uso de rutas de archivos absolutas, es posible acceder a archivos y directorios arbitrarios almacenados en el sistema de archivos, incluido el código fuente o configuración de la aplicación y archivos críticos del sistema. Cabe señalar que el acceso a los archivos está limitado por el control de acceso operativo del sistema (como en el caso de los archivos bloqueados o en uso en el sistema operativo Microsoft Windows). (www.owasp.org – Path Traversal, 2017).

Example Language: **Perl**

```
my $dataPath = "/users/cwe/profiles";
my $username = param("user");
my $profilePath = $dataPath . "/" . $username;

open(my $fh, "<$profilePath") || ExitError("profile read error: $profilePath");
print "<ul>\n";
while (<$fh>) {
  print "<li>$_</li>\n";

}
print "</ul>\n";
```

Figura 27 - Código vulnerable a Path Traversal

El programador tiene la intención de acceder a archivos como "/ users / cwe / profiles / maria" o "/ users / cwe / profiles / pedro", no hay verificación del parámetro de usuario entrante. Un atacante podría proporcionar una cadena como:

```
../../../etc/passwd
```

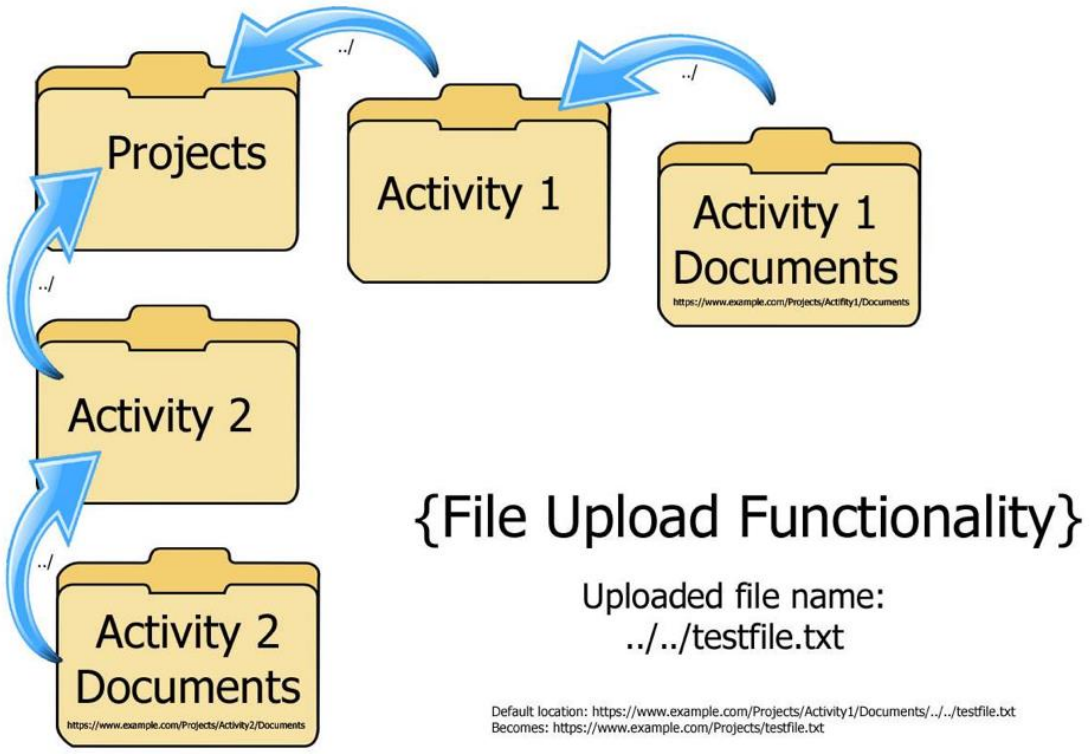
Y el programa gener de perfil como este:

```
/users/cwe/profiles/../../../etc/passwd
```

Cuando se abre el archivo, el sistema operativo resuelve el "../" y realmente accede a este archivo:

```
/etc/passwd
```

Como resultado, el atacante podría leer todo el texto del archivo de contraseña.



5.7.1.3) Command Injection

La inyección de comandos es un ataque en el cual el objetivo es la ejecución de comandos arbitrarios en el sistema operativo host a través de una aplicación vulnerable (OWASP.ORG, 2017). Los ataques de inyección de comandos son posibles cuando una aplicación pasa datos inseguros proporcionados por el usuario (formularios, cookies, encabezados HTTP, etc.) a un shell del sistema. En este ataque, los comandos del sistema operativo suministrados por el atacante generalmente se ejecutan con los privilegios de la aplicación vulnerable. Los ataques de inyección de comandos son posibles en gran parte debido a la validación de entrada insuficiente.

Este ataque difiere de Code Injection, en que la inyección de código permite al atacante agregar su propio código que luego es ejecutado por la aplicación. En Code Injection, el atacante extiende la funcionalidad predeterminada de la aplicación sin la necesidad de ejecutar comandos del sistema.

El siguiente programa simple acepta un nombre de archivo como un argumento de línea de comando y muestra el contenido del archivo al usuario. El programa está instalado en setuid root porque está diseñado para ser utilizado como una herramienta de aprendizaje que permita a los administradores del sistema en entrenamiento inspeccionar archivos del sistema privilegiados sin darles la capacidad de modificarlos o dañar el sistema.

```
int main(char* argc, char** argv) {
        char cmd[CMD_MAX] = "/usr/bin/cat ";
        strcat(cmd, argv[1]);
        system(cmd);
}
```

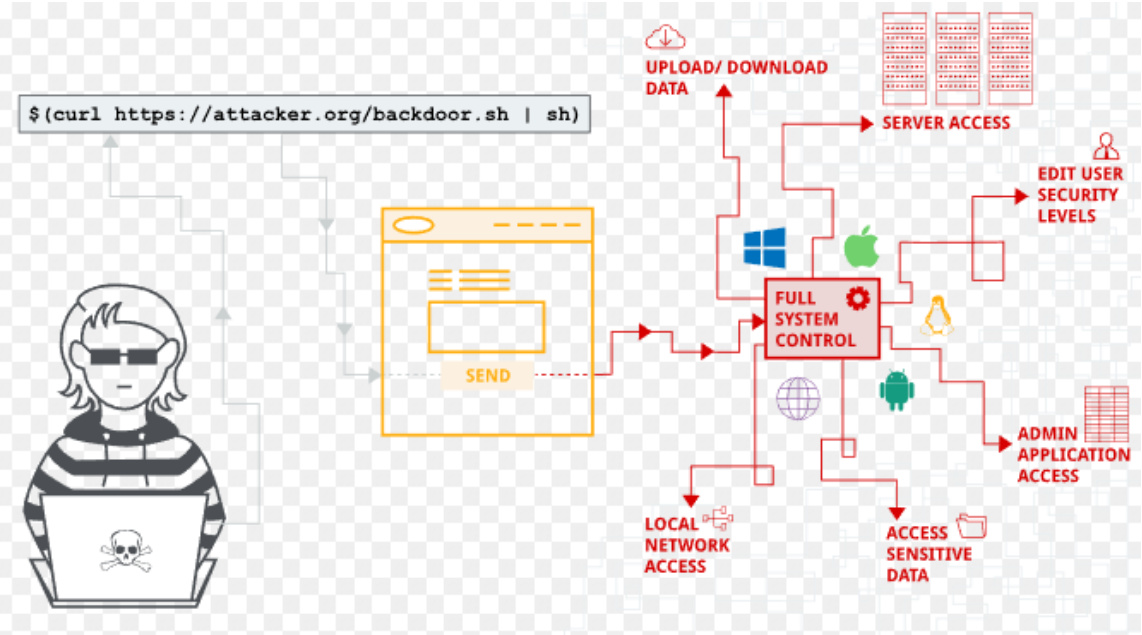


Figura 28 Representacion command injection, recuperado de https://portswigger.net

5.7.1.4) Cross-Site-Scripting (xss)

Los Cross Site Scripting (XSS) son un tipo de inyección, en la cual los scripts maliciosos se inyectan en sitios web benignos y confiables. Los ataques XSS ocurren cuando un atacante usa una aplicación web para enviar código malicioso, generalmente en la forma de un script del lado del navegador, a un usuario final diferente. Los fallos que permiten que estos ataques tengan éxito son bastante comunes y ocurren en cualquier lugar en el que una aplicación web utiliza la información de entrada de un usuario dentro de la salida que genera sin validarla ni codificarla. (OWASP.ORG, 2017).

Un atacante puede usar XSS para enviar un script malicioso a un usuario desprevenido. El navegador del usuario final no tiene manera de saber que el script no debe ser confiable y ejecutará el script. Debido a que piensa que el script proviene de una fuente confiable, el script malicioso puede acceder a cualquier cookie, token de sesión u otra información sensible retenida por el navegador y utilizada con ese sitio. Estos scripts incluso pueden reescribir el contenido de la

página HTML.

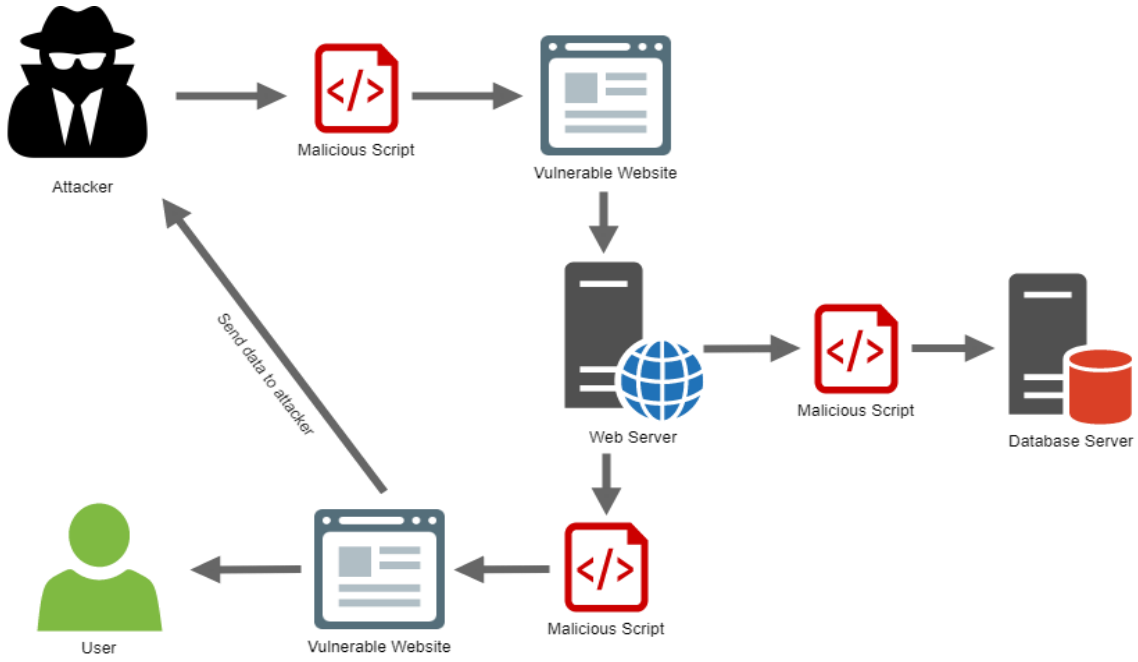


Figura 29 Representacion ataque cross site, recuperado de https://dejanstojanovic.net

5.7.2) Herramienta de ataque OWASP ZAP

Se optó por la herramienta OWASP ZAP como escáner de vulnerabilidades, exactamente la versión 2.7.0 para Windows. Con esta herramienta se realizaron los ataques automatizados para las categorías de vulnerabilidades seleccionadas. Cabe mencionar que estos ataques se realizaron en dos momentos: ataques contra los sitios web directamente sin ningún WAF de por medio y luego interponiendo las herramientas WAF (por separado) entre la herramienta de escaneo y los sitios web.

Los ataques se organizaron por categoría de la siguiente manera:

- ✓ XSS
- ✓ Path Traversal
- ✓ Sql Injection
- ✓ Command Injection

Cada una de estas categorías que presenta ZAP se agrupan por los tipos de ataques, los cuales se activan y se desactivan y configuran por intensidad y umbral. Por defecto ZAP tiene todos los ataques activados y su intensidad y umbral establecidos por defecto. A continuación, se detallan las políticas generadas.

Política	Categoría	Ataque
Command Injection	Inyección	Inyección remota de comandos OS
	Seguridad del Servidor	Remote code execution shell
SQL Injection	Inyección	Inyección CRLF
		Fallo de ejecución SQL
Path Traversal	Recopilación de Información	Exploración de directorios
	Seguridad del Servidor	Directory Traversal
		Inclusión remota de archivo
XSS	Inyección	Cross Site Scripting (persistente)
		Cross Site Scripting (principal)
		Cross Site Scripting (spider)
		Cross Site Scripting (reflejada)

Tabla 2 Politicas de Scaneo OWASP ZAP

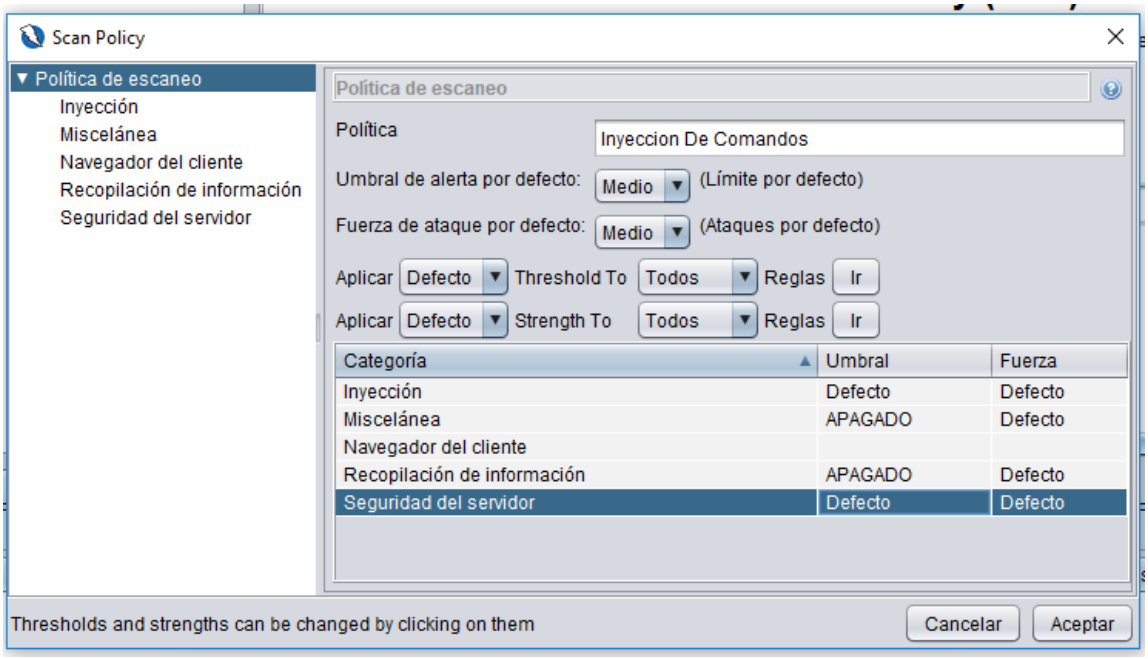


Figura 30 Configuración de categoría en política de scaneo command injection

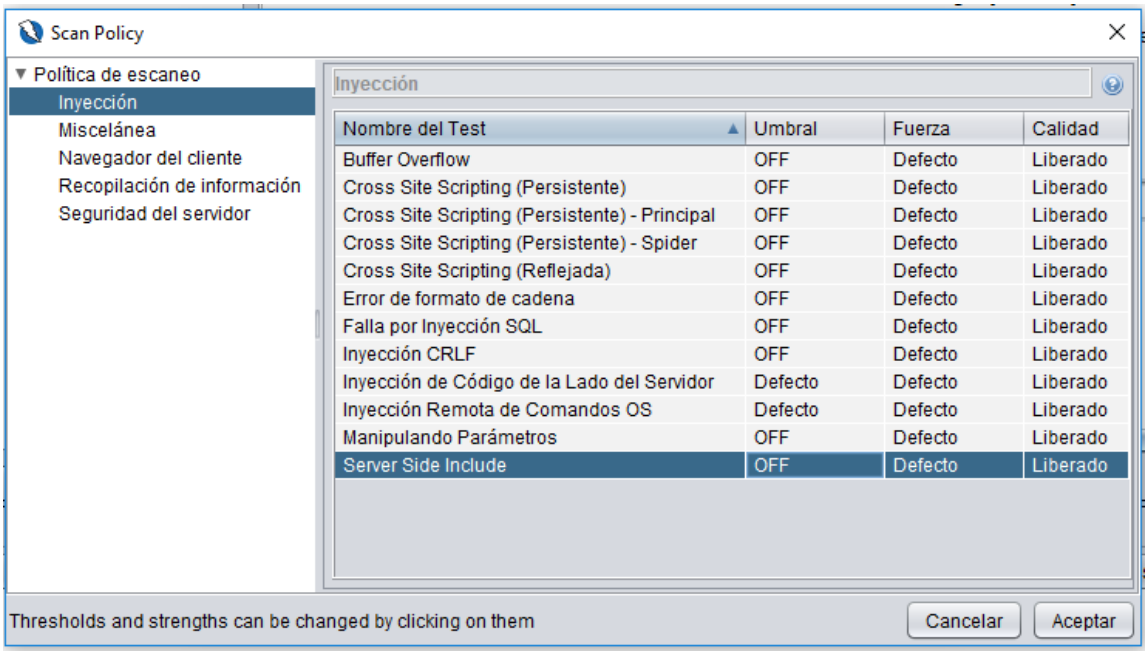


Figura 31 Configuración de los ataques de inyección en la política command injection

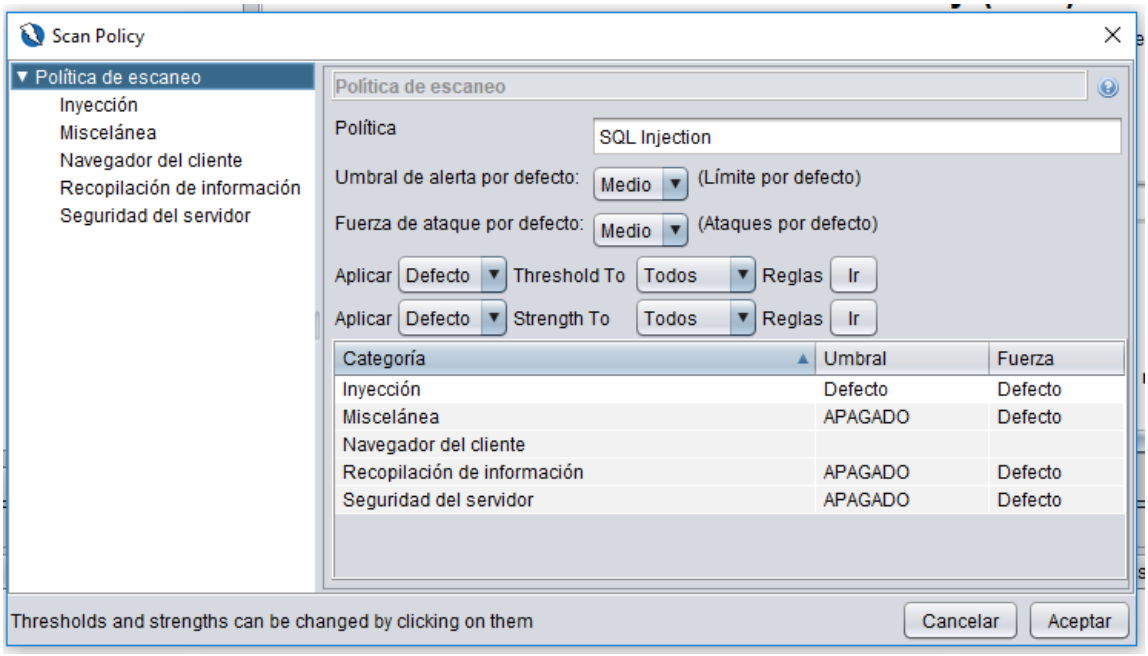


Figura 32 Configuración de categoría en política de scaneo sql injection

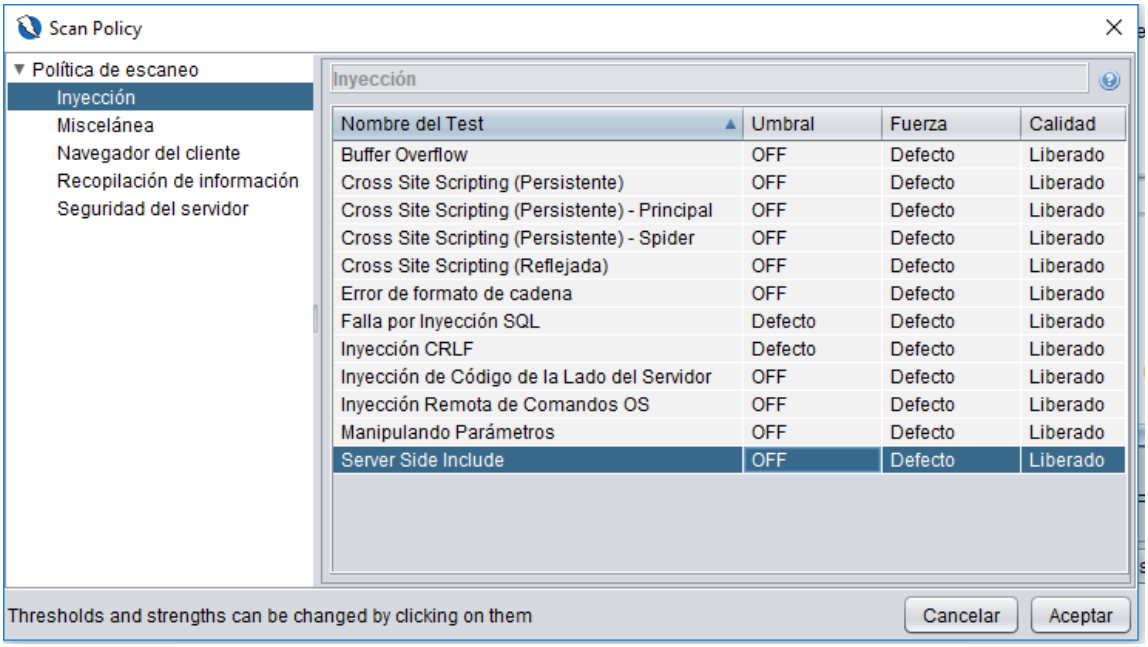


Figura 33 Configuración de los ataques de inyección en la política sql injection

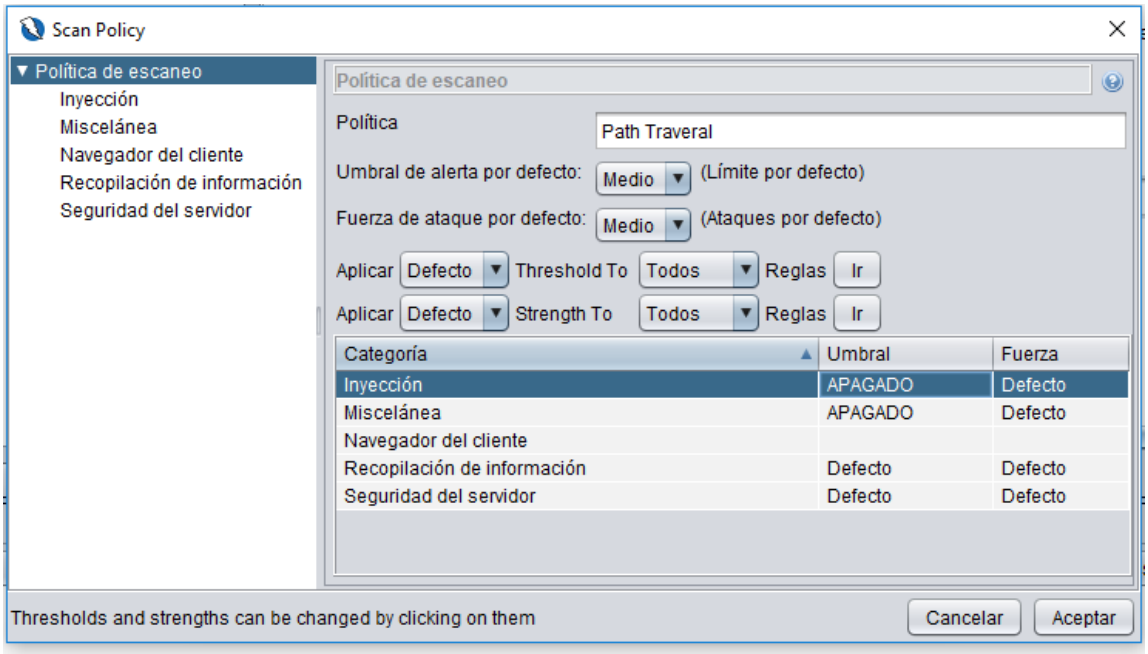


Figura 34 Configuración de categoría en política de scaneo path traversal

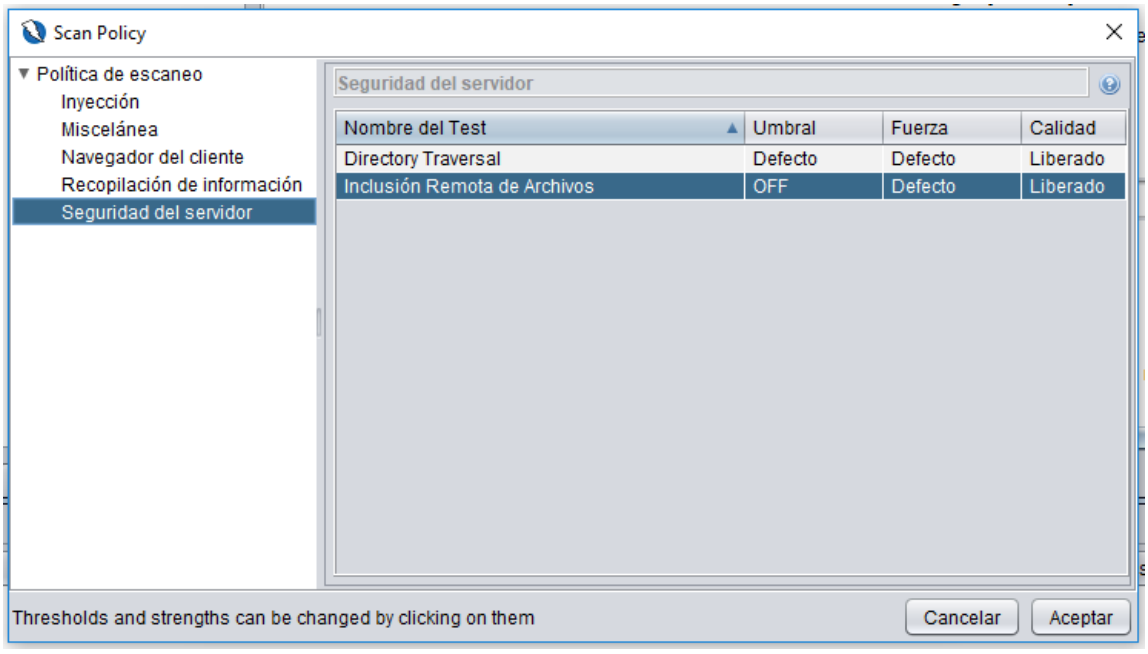


Figura 35 Configuración de los ataques de inyección en la política path traversal

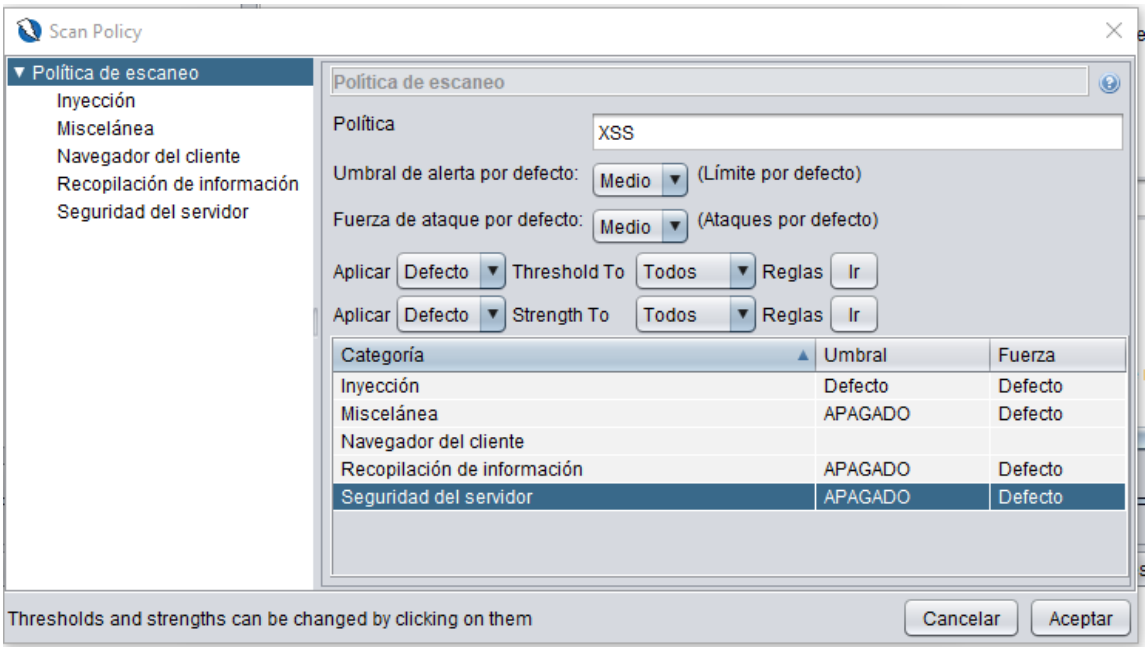


Figura 36 Configuración de categoría en política de scaneo xss

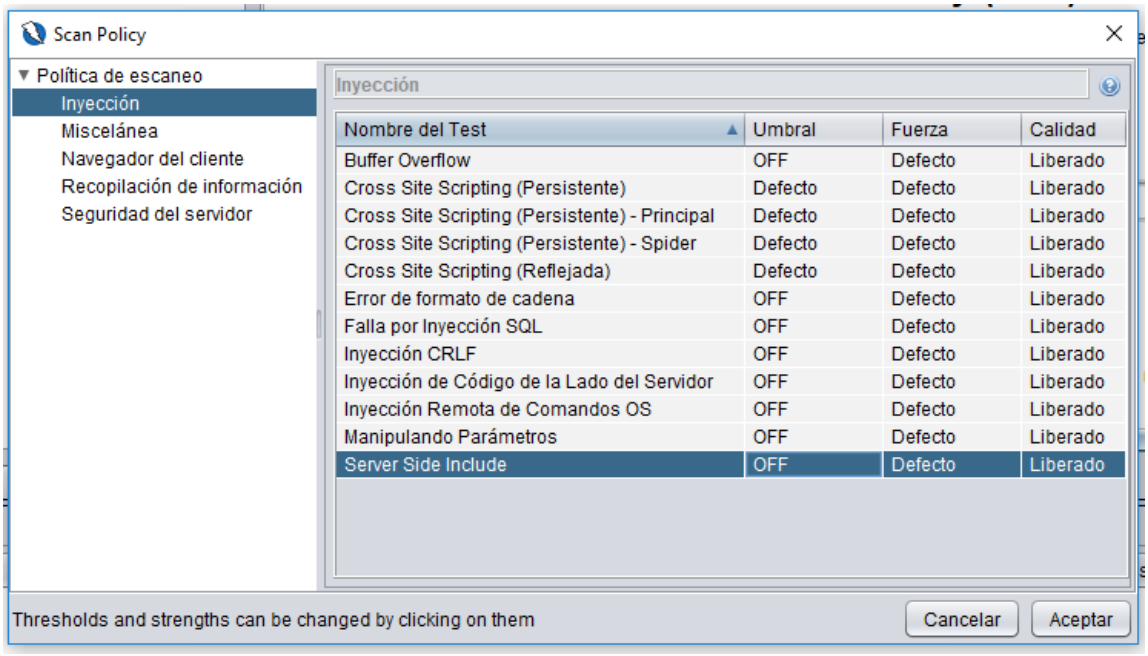


Figura 37 Configuración de los ataques de inyección en la política xss

5.7.3) Formas de ataques y registros

Luego de que se configuraron las políticas de escaneo en OWASP ZAP. El siguiente paso es lanzar los distintos ataques contra los tres sitios web seleccionados para la ejecución de dichas políticas.

Para detallar las formas de ataques y registros, es preciso definir qué se entiende por casos vulnerables y casos no vulnerables:

Casos vulnerables: Se trata de aquellos casos en donde al realizar el ataque hacia los sitios web en busca de una explotación concreta agrupados por categorías, produzca efectos no previstos por el desarrollador de la aplicación.

Casos no vulnerables: Se trata de aquellos casos en donde al realizar el ataque hacia los sitios web en busca de una explotación concreta agrupados por categorías, no produzca ningún efecto que no se haya previsto en el desarrollo de la aplicación. Por lo tanto, no se logrará explotar los sitios.

Como se ha comentado a lo largo del documento, fue necesario lanzar escaneos contra los sitios web sin ninguna herramienta WAF de por medio con el fin de poder seleccionar aquellos ataques que hayan generado alertas. Estas alertas se clasificaron por su nivel de riesgo en: Altas, medias, bajas e informativas.

Los sitios web a los cuales se realizaron los ataques de vulnerabilidades fueron:

- ✓ Sitio Web (Joomla)
- ✓ Aula Virtual (Moodle)
- ✓ Correo Electrónico (Hmail)

A los sitios descriptos anteriormente se ejecutaron los ataques de escaneo en base a la clasificación: command injection, sql injection, path traversal y xss.

Sitio Web (Joomla)

Nombre /Nivel de Riesgo / número alertas	ALTAS	MEDIAS	BAJAS	INFORMATIVAS
Command injection	0	2	4	0
Sql injection	1	2	4	0
Path traversal	0	3	4	0
XSS	1	2	4	0

Tabla 3 Tabla de scaneo a sitio web joomla controlado

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
Low	4
Informational	0

Alert Detail

Medium (Medium)	X-Frame-Options Header Not Set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.
URL	http://10.5.32.10/phpmyadmin/doc/html/import_export.html
Method	GET
Parameter	X-Frame-Options
URL	http://10.5.32.10/joomla/index.php/6-your-template
Method	GET
Parameter	X-Frame-Options
URL	http://10.5.32.10/joomla/index.php/log-out?view=remind
Method	GET
Parameter	X-Frame-Options

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	3
Low	4
Informational	0

Alert Detail

Medium (Medium)	X-Frame-Options Header Not Set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.
URL	http://10.5.32.10/phpmyadmin/doc/html/import_export.html
Method	GET
Parameter	X-Frame-Options
URL	http://10.5.32.10/joomla/index.php/6-your-template
Method	GET
Parameter	X-Frame-Options
URL	http://10.5.32.10/joomla/index.php/log-out?view=remind
Method	GET
Parameter	X-Frame-Options

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	1
Medium	2
Low	4
Informational	0

Alert Detail

High (Medium)	Falla por Inyección SQL
Description	Inyección SQL puede ser posible.
URL	http://10.5.32.10/joomla/index.php?query=query+AND+1%3D1+--+
Method	POST
Parameter	query
Attack	query AND 1=1 --
URL	http://10.5.32.10/joomla/index.php
Method	POST
Parameter	mailto
Attack	ZAP OR 1=1

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	1
Medium	2
Low	4
Informational	0

Alert Detail

High (Medium)	Cross Site Scripting (Reflejada)
Description	Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser ins client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an e HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology. When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) o the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could h redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Cross-sit between a user and the web site. Applications utilizing browser object instances which load content from the file system system compromise. There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based. Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious which when posted to the vulnerable site, will mount the attack. Using a malicious form will oftentimes take place when such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon o the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to embedded client, such as Adobe Flash.

Aula Virtual (Moodle)

Nombre /Nivel de Riesgo / número alertas	ALTAS	MEDIAS	BAJAS	INFORMATIVAS
Command injection	0	0	3	0
Sql injection	1	0	3	0
Path traversal	0	0	3	0
XSS	1	0	3	0

Tabla 4 Tabla de scaneo aula virtual Moodle controlado

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	0
Low	3
Informational	0

Alert Detail

Low (Medium)	X-Content-Type-Options Header Missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of I body, potentially causing the response body to be interpreted and displayed as a content type other than the d will use the declared content type (if one is set), rather than performing MIME-sniffing.
URL	http://10.5.32.10/moodle/mod/forum/view.php?id=2&lang=es
Method	GET
Parameter	X-Content-Type-Options
URL	http://10.5.32.10/moodle/mod/forum/index.php?id=1&lang=en
Method	GET
Parameter	X-Content-Type-Options
URL	http://10.5.32.10/moodle/calendar/view.php?lang=es&time=1516645669
Method	GET
Parameter	X-Content-Type-Options

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	1
Medium	0
Low	3
Informational	0

Alert Detail

High (Medium)	Falla por Inyección SQL
Description	Inyección SQL puede ser posible.
URL	http://10.5.32.10/moodle/login/index.php
Method	POST
Parameter	rememberusername
Attack	1 AND 1=1 --
URL	http://10.5.32.10/moodle/login/index.php
Method	POST
Parameter	anchor
Attack	AND 1=1 --

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	0
Low	3
Informational	0

Alert Detail

Low (Medium)	X-Content-Type-Options Header Missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versior body, potentially causing the response body to be interpreted and displayed as a content type other than will use the declared content type (if one is set), rather than performing MIME-sniffing.
URL	http://10.5.32.10/moodle/mod/forum/view.php?id=2&lang=es
Method	GET
Parameter	X-Content-Type-Options
URL	http://10.5.32.10/moodle/calendar/export.php?course=1&lang=en&time=1516732292
Method	GET
Parameter	X-Content-Type-Options
URL	http://10.5.32.10/moodle/mod/forum/index.php?id=1&lang=en

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	1
Medium	0
Low	3
Informational	0

Alert Detail

High (Medium)	Cross Site Scripting (Reflejada)
Description	Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS re HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported t When an attacker gets a user's browser to execute his/her code, the code will run within the security conte the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted redirected to another location, or possibly shown fraudulent content delivered by the web site they are visit between a user and the web site. Applications utilizing browser object instances which load content from th system compromise. There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based. Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced w which when posted to the vulnerable site, will mount the attack. Using a malicious form will oftentimes take such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaSc the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another t embedded client, such as Adobe Flash. Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period o board posts, web mail messages, and web chat software. The unsuspecting user is not required to interact email), just simply view the web page containing the code.

Correo Electrónico (HMail)

Nombre /Nivel de Riesgo / número alertas	ALTAS	MEDIAS	BAJAS	INFORMATIVAS
Command injection	0	2	3	0
Sql injection	0	2	3	0
Path traversal	0	3	3	0
XSS	0	2	3	0

Tabla 5 Tabla de scaneo correo electronico Hmail controlado

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
Low	3
Informational	0

Alert Detail

Medium (Medium)	X-Frame-Options Header Not Set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.
URL	http://10.5.32.10/webmail/
Method	GET
Parameter	X-Frame-Options
URL	http://10.5.32.10/webmail/?email=foo-bar%40example.com&password=ZAP
Method	GET
Parameter	X-Frame-Options
Instances	2

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
Low	3
Informational	0

Alert Detail

Medium (Medium)	X-Frame-Options Header Not Set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.
URL	http://10.5.32.10/webmail/
Method	GET
Parameter	X-Frame-Options
URL	http://10.5.32.10/webmail/?email=foo-bar%40example.com&password=ZAP
Method	GET
Parameter	X-Frame-Options
Instances	2

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	3
Low	3
Informational	0

Alert Detail

Medium (Medium)	Exploración de Directorios
Description	Es posible ver el listado de directorios. La lista de directorios puede revelar scripts ocultos, incluyen archivos, acceder para leer información sensible.
URL	http://10.5.32.10/webmail/skins/Default/
Method	GET
Attack	Parent Directory
URL	http://10.5.32.10/webmail/static/
Method	GET
Attack	Parent Directory
URL	http://10.5.32.10/webmail/static/js/

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
Low	3
Informational	0

Alert Detail

Medium (Medium)	X-Frame-Options Header Not Set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.
URL	http://10.5.32.10/webmail/
Method	GET
Parameter	X-Frame-Options
URL	http://10.5.32.10/webmail/?email=foo-bar%40example.com&password=ZAP
Method	GET
Parameter	X-Frame-Options
Instances	2

5.7.4) Métricas Utilizadas

Fue necesario clasificar la eficacia de las herramientas WAF evaluadas, mediante el uso de métricas objetivas; el indicador más adecuado para la correcta clasificación de las herramientas es “F-Score”, se emplea en la determinación de un valor único ponderado de precesión y exhaustividad (Zhang & Zhang. 2009).

$$F = 2.\frac{P.R}{P+R}$$

Las medidas que intervienen en el cálculo del indicador:

Precisión (P): La relación del número de payloads maliciosos correctamente detectados entre el número del total de payloads maliciosos detectados. Esta medida se conoce también como “Positive Predictive Value (PPV)”.

$$P = \frac{TP}{TP + FP}$$

Recall (R): La relación del número de payload maliciosos correctamente detectados entre el número de paylods maliciosos totales conocidos. También se denomina como “True Positive Rate (TPR)”.

$$R = \frac{TP}{TP + FN}$$

Dado que:

TP – Verdaderos Positivos: El número de payloads maliciosos correctamente detectados. Aquellos detectados que son capaces de explotar una vulnerabilidad.

FP – Falsos Positivos: El número de payloads detectados como maliciosos pero que en realidad no son capaces de explotar una vulnerabilidad determinada.

FN – Falsos Negativos: El número de payloads maliciosos no detectados.

El valor F-Score es un compromiso entre Recall y Precision. Este asume valores entre el intervalo (0,1). Dado el contexto, asumimos 0 sin ningún payload malicioso ha sido detectado, y 1 si todos los payload detectados son maliciosos y todos los payloads maliciosos han sido detectados.

A su vez se proponen los siguientes indicadores:

FPR – Falso Positivo Rate: La relación de payloads incorrectamente detectados como capaces de explotar una vulnerabilidad concreta asociada a un caso de prueba no vulnerable, entre el número de payloads incapaces de explotar casos de prueba no vulnerables, es decir, la probabilidad que un payloads no maliciosos sea detectado incorrectamente como malicioso (conocidos como falsas alarmas).

$$FPR = \frac{FP}{FP + TN}$$

NPV – Valores Predictivos Negativos: La relación de payloads detectados correctamente como incapaces de explotar una vulnerabilidad concreta asociada a un caso de prueba no vulnerable, entre la suma del número de payloads incapaces de explotar casos de pruebas no vulnerables y el número de payloads. Es decir, se trata de la confianza que se le puede asignar a la ausencia de alertas ente la generación de un ataque en concreto.

$$NPV = \frac{TN}{TN + FN}$$

Exactitud: La relación de payloads identificados correctamente entre el total de payloads generados:

$$\text{Exactitud: } = \frac{TP + TN}{TP + TN + FP + FN}$$

Dado que:

TN – Verdaderos Negativos: El número de payloads no maliciosos no detectados.

5.8) Resultados

A continuación, se detallan los resultados obtenidos por cada una de las soluciones evaluadas.

5.8.1) Endian

Vulnerabilidad	Payloads	TP	FP	TN	FN	PRECISION	RECALL	FALSOS POSITIVOS RATE	VALORES PREDICTIVOS NEGATIVOS	EXACTITUD	F1-SCORE
Command injection	146	12	0	73	61	1	0.16438356	0	0.54477612	0.58219178	0.28235294
Sql injection	160	74	0	80	6	1	0.925	0	0	0.9625	0.96103896
Path traversal	161	77	0	81	3	1	0.9625	0	0.96428571	0.98136646	0.98089172
XSS	150	51	0	75	24	1	0.68	0	0	0.84	0.80952381
TOTAL	617	214	0	309	94	1	0.68297089	0	0.37726546	0.84151456	0.75845186
						PROMEDIO					

Tabla 6 Tabla resultado solucion Endian Evaluada

Se observa la falta de Falsos Positivos, esto provoca que el índice de precisión se posicione con un valor de 1 y que el valor de falsos positivos rate también sea de 0. El relativo bajo número de Falsos Negativos, en conjunto con el alto número de Verdaderos Positivos (TP), generan que el valor Recall se promedie en 0.68 y que el F-Score este en 0.758.

El valor promedio de los predictivos negativos es de 0.37, esto debido a la correcta identificación de los Verdaderos Negativos y al reducido número de Falsos Negativos.

5.8.2) Shadow Daemon

Vulnerabilidad	Payloads	TP	FP	TN	FN	PRECISION	RECALL	FALSOS POSITIVOS RATE	VALORES PREDICTIVOS NEGATIVOS	EXACTITUD	F1-SCORE
Command injection	146	48	48	25	25	0.5	0.65753425	0.65753425	0.5	0.5	0.56804734
Sql injection	160	80	80	0	0	0.5	1	1	0	0.5	0.66666667
Path traversal	161	47	47	33	34	0.5	0.58024691	0.5875	0.49253731	0.49689441	0.53714286
XSS	150	75	75	0	0	0.5	1	1	0	0.5	0.66666667
TOTAL	617	250	250	58	59	0.5	0.80944529	0.81125856	0.24813433	0.4992236	0.60963088
						PROMEDIO					

Tabla 7 Tabla resultado solucion Shadow Daemon Evaluada

Es notorio el alto número de Falsos Positivos en todas las categorías, en las categorías de sql injection y xss, esto se puede deber a que shadow Daemon realiza un bloqueo de las peticiones, lo cual conlleva a la obtención del mismo número de TP y FP. Lo anterior provoca que el valor de FPR sea de 1.

Al lograr obtener un alto índice de Falsos Negativos y un bajo índice de Falsos Positivos, el valor de valores predictivos negativos se convierte en el valor mas bajo de todos los WAF evaluados (0.24).

También es observable que el índice de precisión es de 0.5, lo cual es consecuencia del alto número de FP. El índice Recall también se logra ver afectado por el número de FN obtenidos en las distintas categorías, esto afecta directamente el F-Score, obteniendo un índice promedio de 0.6

5.8.3) Sophos UTM

Vulnerabilidad	Payloads	TP	FP	TN	FN	PRECISION	RECALL	FALSOS POSITIVOS RATE	VALORES PREDICTIVOS NEGATIVOS	EXACTITUD	F1-SCORE
Command injection	146	0	0	73	73	0	0	0	0.5	0.5	0
Sql injection	160	80	0	80	0	1	1	0	1	1	1
Path traversal	161	0	0	81	80	0	0	0	0	0.50310559	0
XSS	150	72	48	27	3	0.6	0.96	0.64	0	0.66	0.73846154
TOTAL	617	152	48	261	156	0.4	0.49	0.16	0.375	0.6657764	0.43461538
					PROMEDIO						

Tabla 8 Tabla resultado solucion Sophos Evaluada

Se logra observar claramente la nula detección de ataques en las categorías de command injection y path traversal; a pesar de esto el nivel de puntuación obtenido en las otras categorías consiguen que el índice de precisión promedio se establezca en 0.4.

También logramos observar que en la categoría de Sql Injection las detecciones de TP y TN llegan a un 100%, esto hace que se logre alcanzar el valor de 1 en Precisión, Recall, Valores Predictivos negativos, exactitud y F1-score. Los TN obtuvieron un alto índice al igual que los FN, esto penaliza la puntuación obtenida en los valores predictivos negativos.

5.9) Resultados y comparación de soluciones WAF

A raíz de los datos que se obtuvieron de las herramientas evaluadas, se realizaron tablas y graficas que ayudan en el análisis de valorar el desempeño de cada una de las herramientas.

La siguiente tabla muestra las herramientas evaluadas ordenadas descendentemente según el valor obtenido en el índice F-score. Se eligió este orden debido a que el índice relaciona precisión y Recall, de tal manera que se logra observar la relación directa entre TP, FP y TN.

Las probabilidades de las falsas alarmas están dadas por el valor de FPR, lo cual indica que cuanto menor sea el valor → mejor desempeño posee el WAF evaluado.

Los índices restantes presentan una relación directamente proporcional al desempeño del WAF, por lo tanto, y para una mayor comprensión, se opto por usar valores complementarios en función del FPR, dado que el intervalo de cada uno de los índices es de [0-1]. Si el valor real del FPR es de 0.63, el valor a presentarse es 1-0.63= 0.37.

WAF	F-SCORE	PRECISION	RECALL	FALSOS POSITIVOS RATE	VALORES PREDICTIVOS NEGATIVOS	EXACTITUD
ENDIAN	0.758451858	1	0.68297089	1	0.377265458	0.84151456
SHADOWN	0.609630882	0.5	0.80944529	0.188741438	0.248134328	0.499223602
SOPHOS	0.434615385	0.4	0.49	0.84	0.375	0.665776398

Tabla 9 Resumen F-score Soluciones Evaluadas

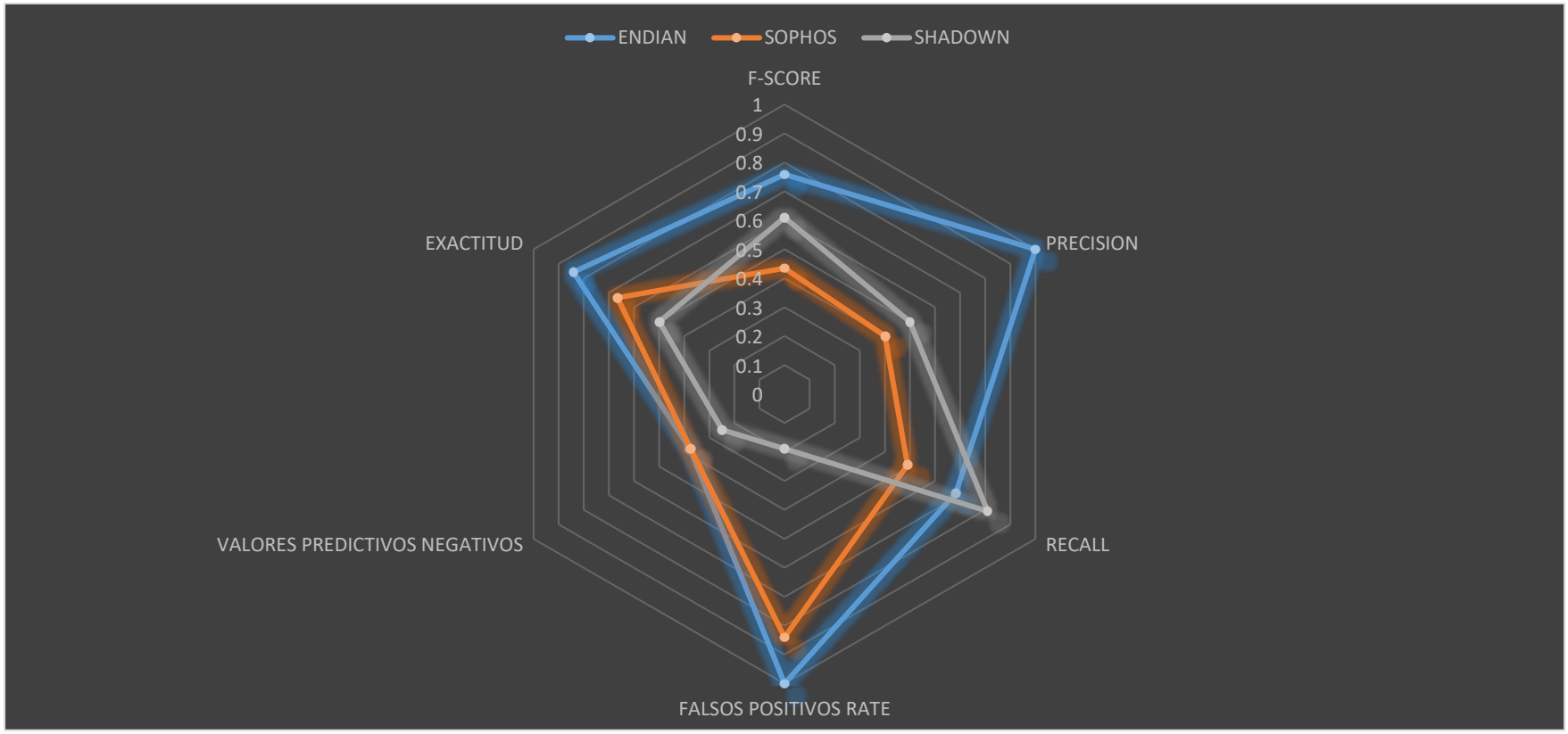


Figura 38 Gráfico Radias de Índices obtenidos por las herramientas WAF

Se logra observar que la herramienta Endian es la que tiene un mayor índice de precisión y también reconoce los falsos positivos de una mejor forma que las otras dos soluciones WAF, pero de forma general, las tres soluciones son muy similares respecto al F-score en algunas categorías evaluadas.

F-SCORE				
WAF	**Command injection**	**Sql injection**	**Path traversal**	**XSS**
ENDIAN	0.282352941	0.961038961	0.98089172	0.80952381
SOPHOS	0	1	0	0.738461538
SHADOWN	0.568047337	0.666666667	0.537142857	0.666666667

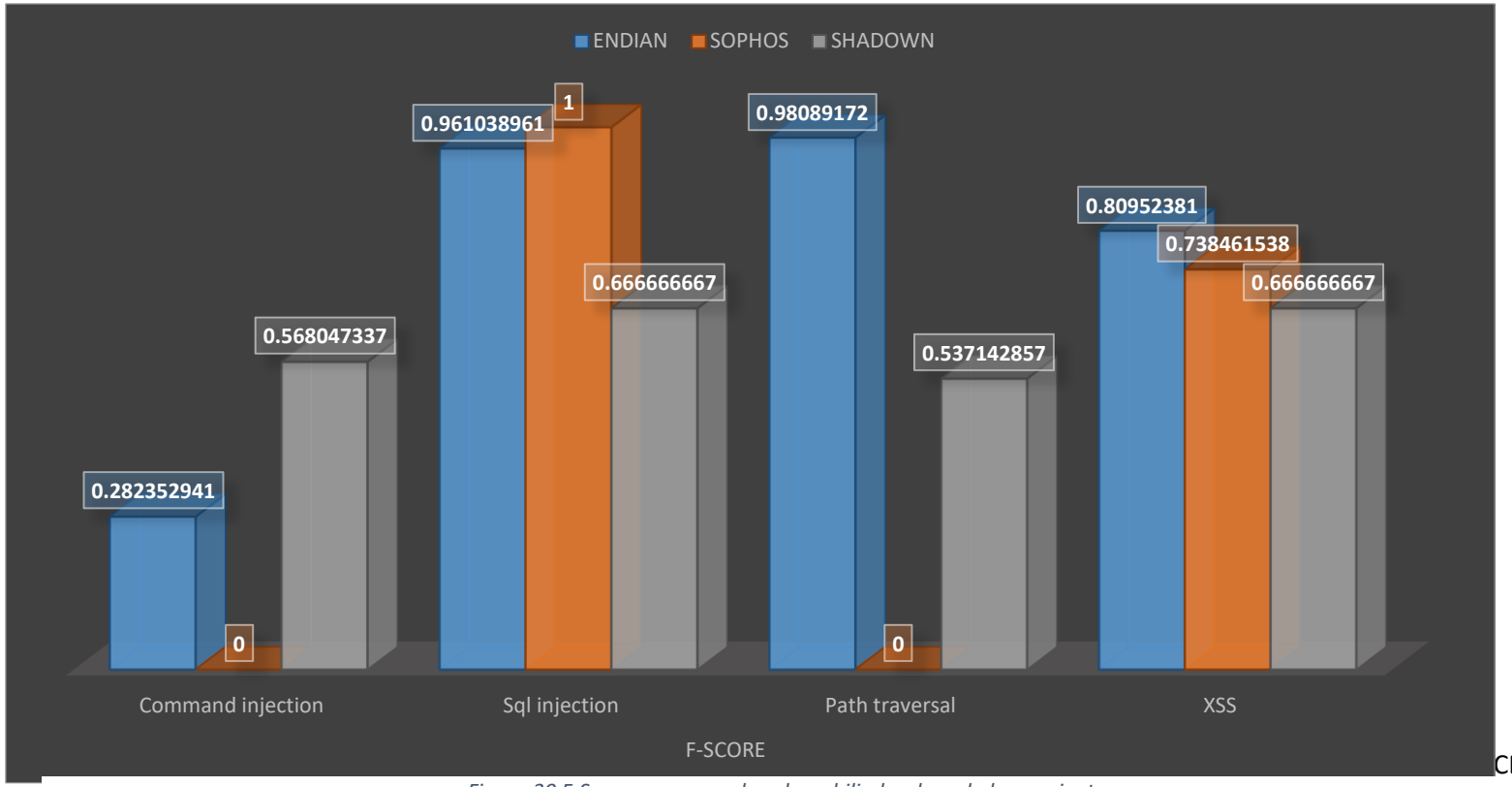


Figura 39 F-Score por grupo de vulnerabiliades de cada herramienta

6) Conclusiones y Recomendaciones

En este trabajo se realizaron diferentes ataques para escanear vulnerabilidades de algunos servicios web y lograr realizar una comparativa de tres soluciones WAF Open Source puestas en el medio de estos servicios, logrando realizar un análisis de los resultados obtenidos.

Luego de haber analizados los resultados obtenidos se logra concluir lo siguiente:

En el proceso investigativo se logró indagar sobre distintos mecanismos de ataque, los cuales fueron de mucha utilidad para poder poner a prueba las soluciones evaluadas. Estas mismas indagaciones conllevaron a hacer uso de herramientas automatizadas de ataques, las cuales ayudaron a obtener sugerencias sobre las medidas de protección a utilizar para mitigar los ataques.

El hecho de no contar con un WAF que proteja los sistemas, estos últimos pueden llegar a ser fácilmente escaneados y vulnerados por herramientas automatizadas, no obstante, el anteponer un WAF con las reglas básicas activadas permite que las vulnerabilidades sean ampliamente mitigadas.

Dos de las tres soluciones WAF evaluadas: Shadown Daemon y Sophos muestran valores muy similares en varios de sus índices analizados (Precisión, Exactitud, Valores Predictivos Negativos), esto puede ser el resultado del bloqueo de la petición antes de llegar a los servicios y a las reglas establecidas en su configuración estándar, lo cual pudo haber aumentado el alto número de Falsos Positivos, lo que coarta su F-score. Los resultados obtenidos por estas dos soluciones WAF evidentemente mejoraran si se realiza la modificación de sus reglas estándar y se logra realizar reglas especificas en función del entorno de trabajo y el tipo de aplicación que se va a proteger.

Endian logró obtener una Precisión de 1, lo que lo destaca de las otras dos soluciones WAF y a su vez obteniendo el mayor puntaje en F-Score de 0.7584.

Queda en evidencia que las tres herramientas evaluadas, con sus configuraciones de reglas estándar establecidas, realizan un buen trabajo y que el resultado varía en virtud del tipo de ataque que intentan mitigar.

Sophos se perfila como una buena opción para la protección de vulnerabilidades como Sql Injectios y XSS.

Las soluciones WAF evaluadas presentan facilidad de uso, configuración y también algunos servicios complementarios (como balanceo de carga), por tal motivo son una excelente opción para las empresas que obtén por proteger sus aplicaciones web con herramientas Open Source.

7) Citas y Referencias Bibliográficas

1- Ministerio de Transporte e infraestructura, Organigrama Institucional, 5 mayo 2015, http://www.mti.gob.ni/index.php/ministerio/organigrama

2- Arévalo Juan Jose, 15 septiembre 2011, Hacker bloquean varias páginas de Internet del gobierno de Nicaragua, La Jornada. 5 de mayo 2015, http://www.lajornadanet.com/diario/archivo/2011/septiembre/15/1.php

3- Nuxbone, 5 mayo 2015, Que es test de penetración, http://www.nyxbone.com/pentest.html

4- Chavarría, Ana Rosa, noviembre 2005, legislación y el Manejo de la Información en la era del conocimiento

5- Herzog Peter, 2010, The Open Source Security Testing Methodology Manual (OSSTMM), ISECOM

6- Jaime Blasco, 2007, Ataques DoS en aplicaciones Web

7- Oracle, agosto 2011, Guía de administración del sistema: servicios IP

8- Tony Pérez, febrero 2017, WAF un enfoque practico para la seguridad de sitios web, https://blog.sucuri.net/espanol/2017/02/website-application-firewalls-waf-un-enfoque-practico-para-la-seguridad-de-sitios-web.html

9- Symantec, agosto 2017, políticas WAF para proteger servidores contra ataques, https://origin-symwisedownload.symantec.com/resources/webguides/managementcenter/1.9.1.1/Content/ConfigurationManagementGuide/6_Policy/WAF/WAF_solution.htm

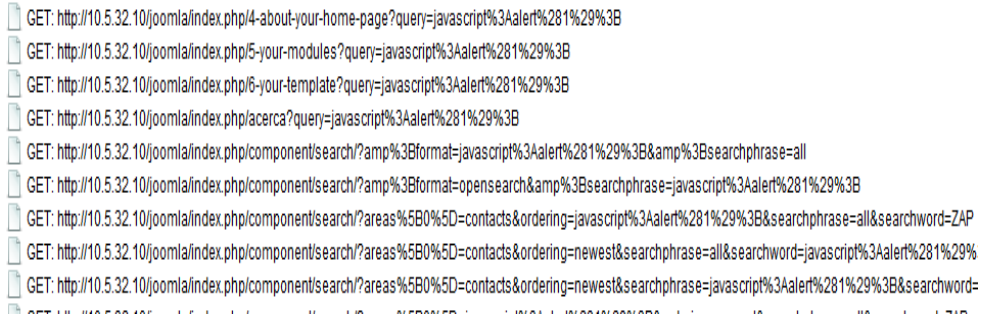
10- PandaSecurity, agosto 2017, configurar firewall endpoint protection, https://www.pandasecurity.com/usa-es/support/card?id=50010

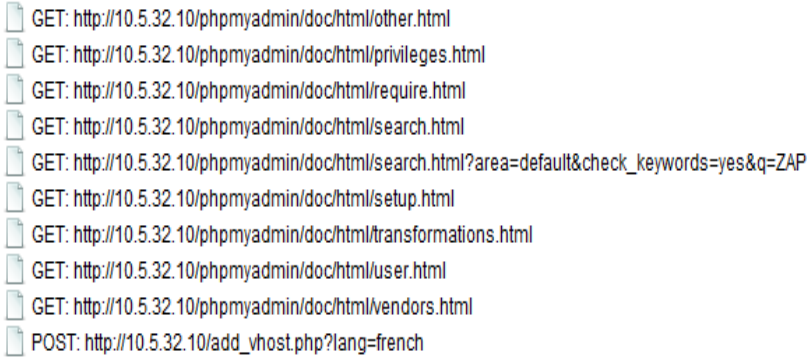
11- Whitehatsecurity,enero 2017, Directory Traversal Attack, https://www.whitehatsec.com/blog/directory-traversal-attack/

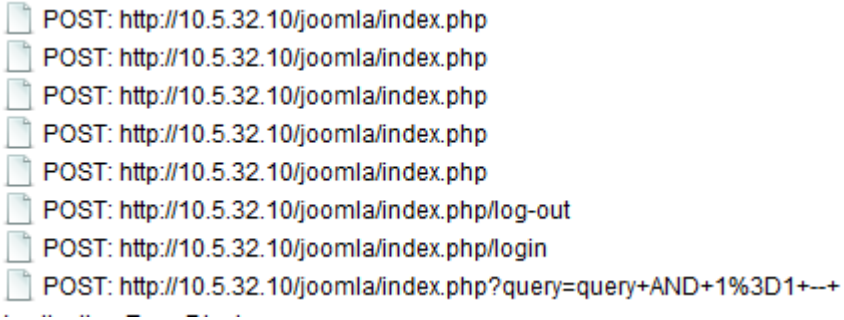
8) Anexos

8.1) Payload Generados con Endian en el medio

Luego de poner en el medio el WAF Endian en el medio del atacante y los servicios a proteger, se realizaron los ataques contra estos últimos. En este anexo se muestran los ataques, pero debido a la extensión de la información generada por la herramienta, solamente se observan parte de los ataques generados.

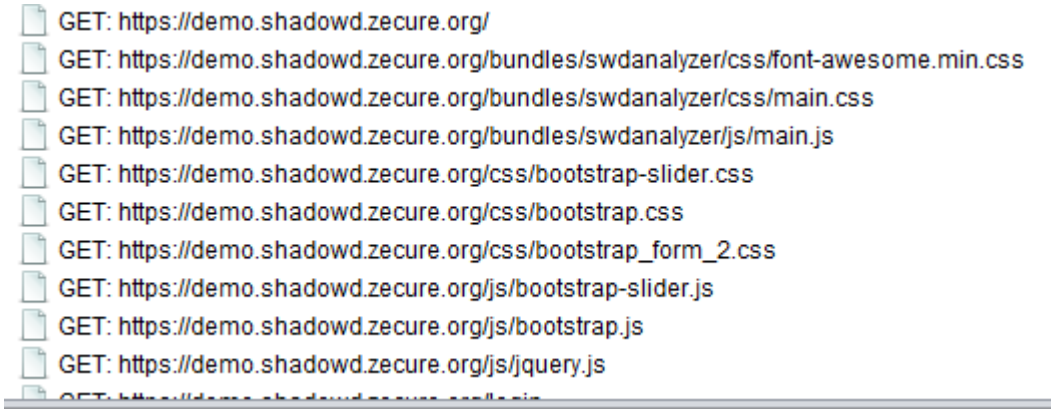






8.2) Payload Generados con Shadow Daemon en el medio

Luego de poner en el medio el Whadow Daemon en el medio del atacante y los servicios a proteger, se realizaron los ataques contra estos últimos. En este anexo se muestran los ataques, pero debido a la extensión de la información generada por la herramienta, solamente se observan parte de los ataques generados.



8.3) Payload Generados con Sophos UTM en el medio

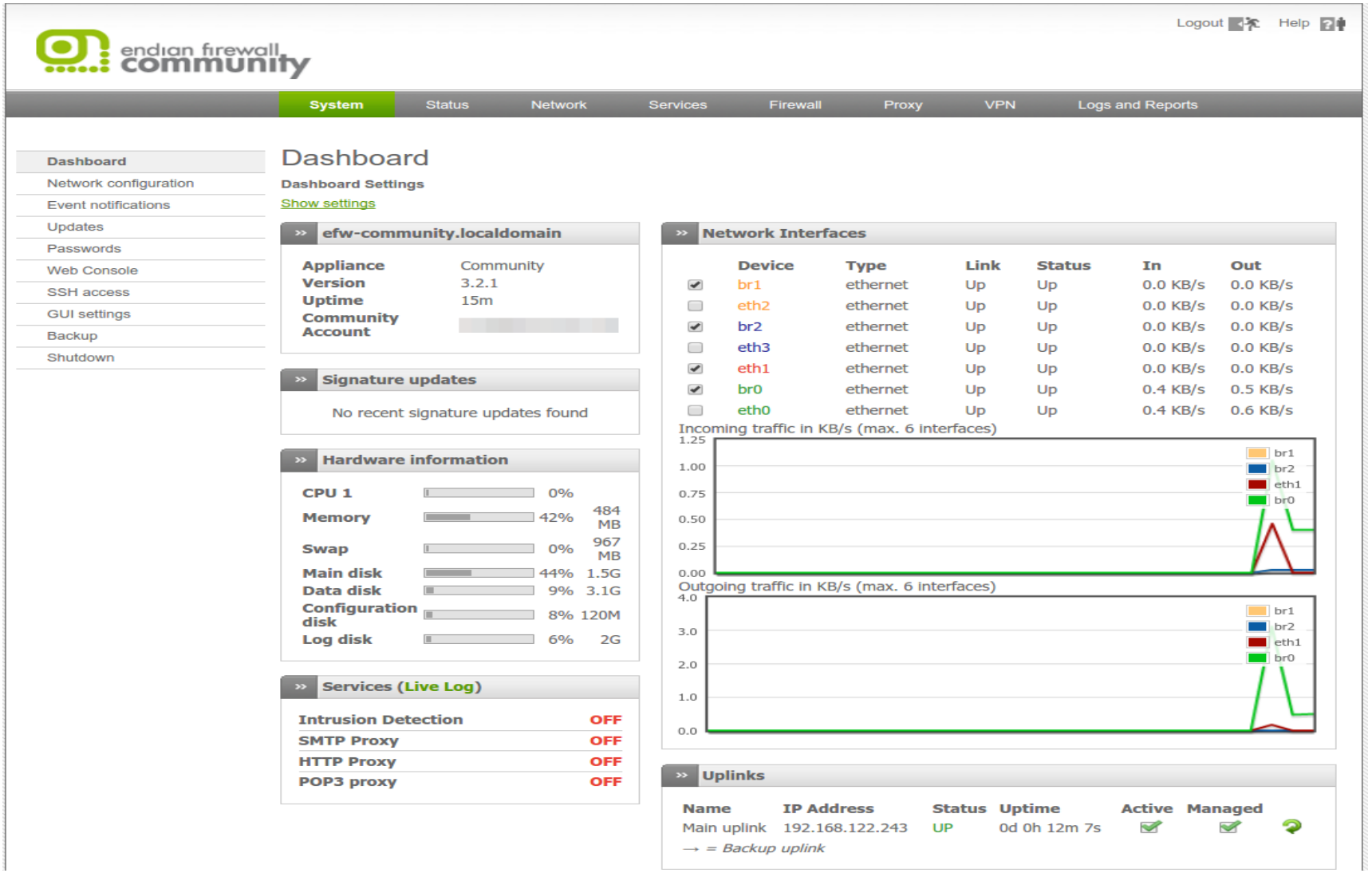
Luego de poner en el medio el Shadow Daemon en el medio del atacante y los servicios a proteger, se realizaron los ataques contra estos últimos. En este anexo se muestran los ataques, pero debido a la extensión de la información generada por la herramienta, solamente se observan parte de los ataques generados.

Processed	Método	URI	Flags
	GET	https://utm.trysophos.com:4444/core/js/	
	GET	https://utm.trysophos.com:4444/wfe/	
	GET	https://utm.trysophos.com:4444/wfe/asg/	
	GET	https://utm.trysophos.com:4444/wfe/asg/js/	
	GET	http://raphaeljs.com/	Out of Scope
	GET	http://www.w3.org/TR/SVG11/feature	Out of Scope
	GET	http://www.w3.org/1999/xlink	Out of Scope
	GET	http://www.w3.org/2000/svg	Out of Scope

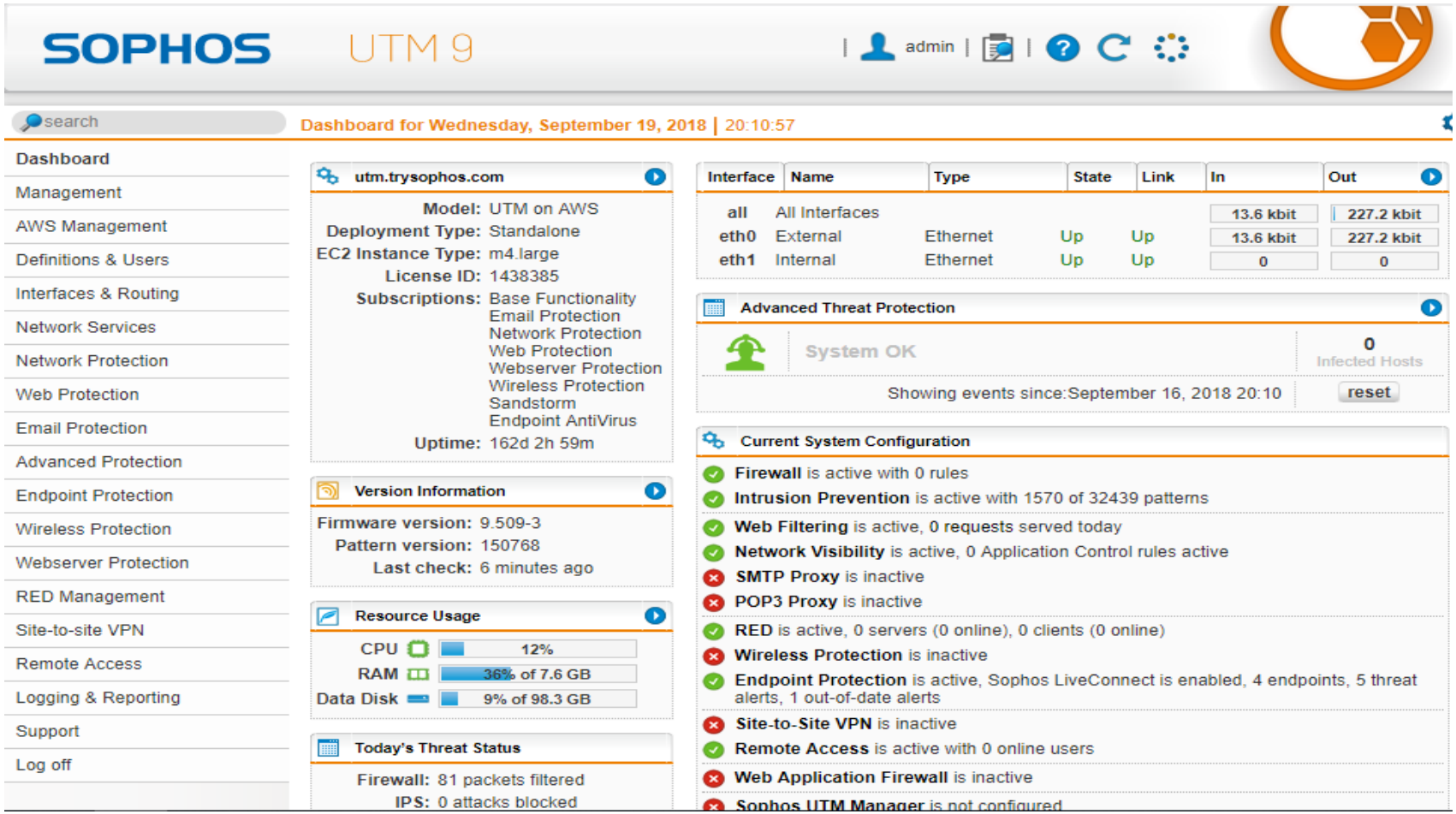
ID	Req. Timestamp	Resp. Timestamp	Método	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
212	19/09/18 14:14:59	19/09/18 14:14:59	GET	https://utm.trysophos.com/?query=..%2F..%2F..%2...	200	OK	614 ms	901 bytes	1.093.153 bytes
213	19/09/18 14:14:59	19/09/18 14:15:00	GET	https://utm.trysophos.com/?query=c%3A%5CWind...	200	OK	704 ms	901 bytes	1.093.153 bytes
214	19/09/18 14:15:00	19/09/18 14:15:01	GET	https://utm.trysophos.com/?query=..%5C..%5C..%5...	200	OK	703 ms	901 bytes	1.093.153 bytes
215	19/09/18 14:15:01	19/09/18 14:15:01	GET	https://utm.trysophos.com/?query=%2Fetc%2Fpas...	200	OK	698 ms	901 bytes	1.093.153 bytes
216	19/09/18 14:15:02	19/09/18 14:15:03	GET	https://utm.trysophos.com/?query=..%2F..%2F..%2...	200	OK	691 ms	901 bytes	1.093.153 bytes
217	19/09/18 14:15:03	19/09/18 14:15:04	GET	https://utm.trysophos.com/?query=WEB-INF%2Fw...	200	OK	689 ms	901 bytes	1.093.153 bytes
218	19/09/18 14:15:04	19/09/18 14:15:05	GET	https://utm.trysophos.com/?query=WEB-INF%5Cw...	200	OK	624 ms	901 bytes	1.093.153 bytes
219	19/09/18 14:15:05	19/09/18 14:15:06	GET	https://utm.trysophos.com/?query=%2FWEB-INF%...	200	OK	624 ms	901 bytes	1.093.153 bytes
220	19/09/18 14:15:06	19/09/18 14:15:07	GET	https://utm.trysophos.com/?query=%5CWEB-INF%...	200	OK	615 ms	901 bytes	1.093.153 bytes
221	19/09/18 14:15:07	19/09/18 14:15:07	GET	https://utm.trysophos.com/?query=thisbouldnotexis...	200	OK	613 ms	901 bytes	1.093.153 bytes

18:18:58	Default DROP	DNS	10.44.0.2:53 → 10.44.1.67:46951	len=118 ttl=255 tos=0x00	srcmac=06:
18:22:01	Default DROP	DNS	10.44.0.2:53 → 10.44.1.67:53104	len=118 ttl=255 tos=0x00	srcmac=06:
18:24:03	Default DROP	DNS	10.44.0.2:53 → 10.44.1.67:57926	len=126 ttl=255 tos=0x00	srcmac=06:
18:39:12	Default DROP	DNS	10.44.0.2:53 → 10.44.1.67:4004	len=126 ttl=255 tos=0x00	srcmac=06:
18:52:18	Default DROP	DNS	10.44.0.2:53 → 10.44.1.67:40039	len=126 ttl=255 tos=0x00	srcmac=06:
18:53:19	Default DROP	DNS	10.44.0.2:53 → 10.44.1.67:51860	len=118 ttl=255 tos=0x00	srcmac=06:
19:25:35	Default DROP	DNS	10.44.0.2:53 → 10.44.1.67:12152	len=462 ttl=255 tos=0x00	srcmac=06:
19:29:38	Default DROP	DNS	10.44.0.2:53 → 10.44.1.67:3236	len=462 ttl=255 tos=0x00	srcmac=06:
19:49:48	Default DROP	DNS	10.44.0.2:53 → 10.44.1.67:10492	len=118 ttl=255 tos=0x00	srcmac=06:
19:50:50	Default DROP	DNS	10.44.0.2:53 → 10.44.1.67:13697	len=126 ttl=255 tos=0x00	srcmac=06:

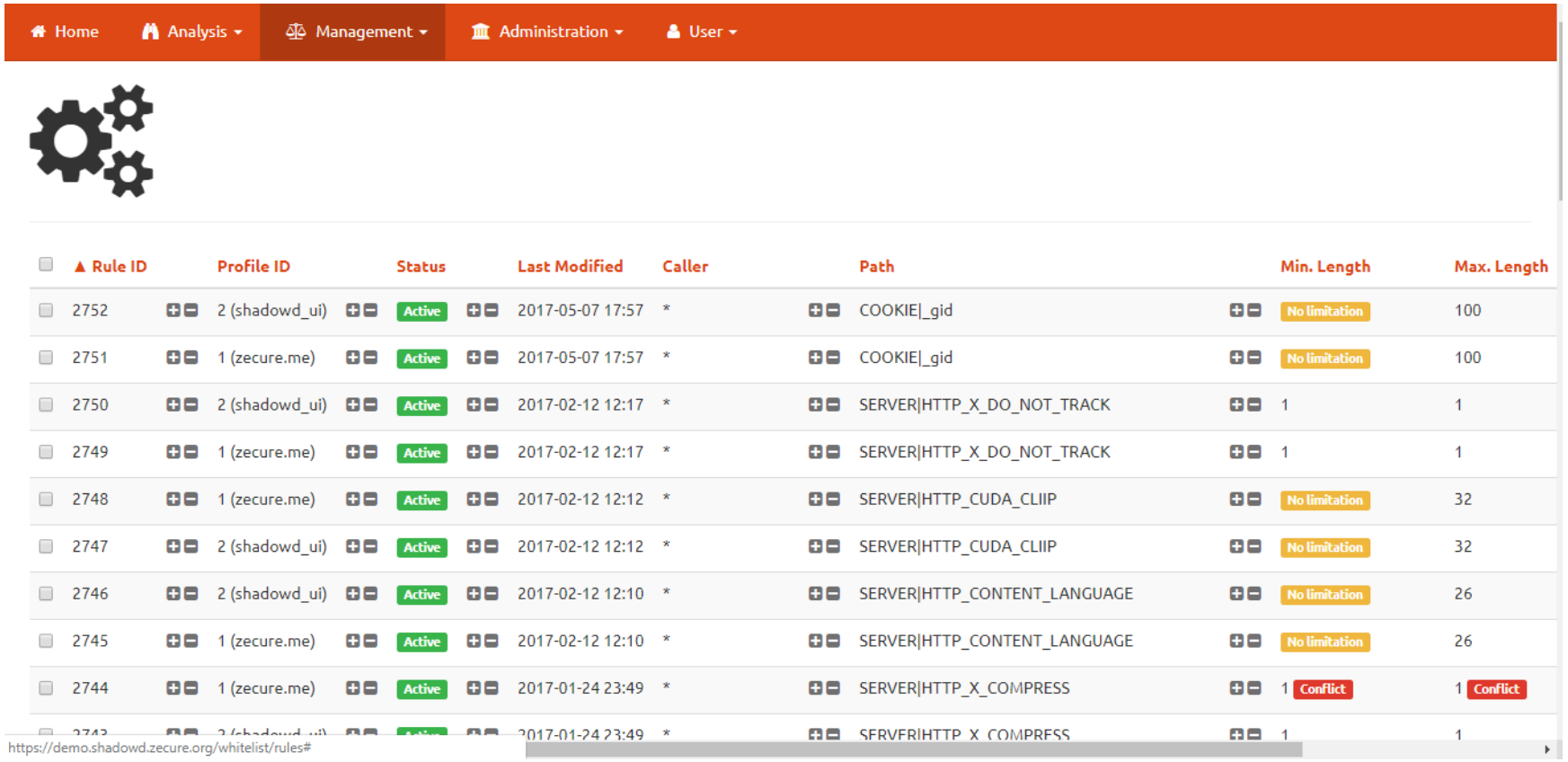
8.4) Interfaz Endian



8.5) Interfaz Sophos UTM



8.6) Interfaz Shadow Daemon



▲ Rule ID	Profile ID	Status	Last Modified	Caller	Path	Min. Length	Max. Length
2752	2 (shadowd_ui)	Active	2017-05-07 17:57	*	COOKIE|_gid	No limitation	100
2751	1 (zecure.me)	Active	2017-05-07 17:57	*	COOKIE|_gid	No limitation	100
2750	2 (shadowd_ui)	Active	2017-02-12 12:17	*	SERVER|HTTP_X_DO_NOT_TRACK	1	1
2749	1 (zecure.me)	Active	2017-02-12 12:17	*	SERVER|HTTP_X_DO_NOT_TRACK	1	1
2748	1 (zecure.me)	Active	2017-02-12 12:12	*	SERVER|HTTP_CUDA_CLIIP	No limitation	32
2747	2 (shadowd_ui)	Active	2017-02-12 12:12	*	SERVER|HTTP_CUDA_CLIIP	No limitation	32
2746	2 (shadowd_ui)	Active	2017-02-12 12:10	*	SERVER|HTTP_CONTENT_LANGUAGE	No limitation	26
2745	1 (zecure.me)	Active	2017-02-12 12:10	*	SERVER|HTTP_CONTENT_LANGUAGE	No limitation	26
2744	1 (zecure.me)	Active	2017-01-24 23:49	*	SERVER|HTTP_X_COMPRESS	1 Conflict	1 Conflict