# Bloque 1: Instalación de un proxy inverso

## Ejercicio L10B1\_REVPROXY: Apache 2 instalado como un proxy inverso

Para configurar Apache2 para que sirva como proxy inverso, debes editar el archivo /etc/apache2/sitesenabled/000-default.conf

Por ejemplo, puedes redirigir las solicitudes a "/epi" y "/eii" a las IPs indicadas en la figura de ambos servidores web. Para garantizar un proxy inverso transparente de las máquinas necesitamos utilizar las directivas ProxyPass y ProxyPassReverse (https://httpd.apache.org/docs/2.4/mod/mod\_proxy.html) sobre las IPs de la red interna. Ambos deben apuntar a la misma IP del servidor correspondiente. Por lo tanto, debemos crear dos entradas Location en ese archivo con esta estructura, ambas dentro del bloque VirtualHost existente:



IMPORTANTE: copiar el archivo /etc/apache2/sites-enabled/000-default.conf a /shared para conservarlo y restaurar los cambios en caso de que el contenedor proxy se cerrase accidentalmente.

# Bloque 2. Web Application Firewalls (WAFs)

## Ejercicio L10B2\_WAF: Instalación de un WAF

Añadir reglas adecuadas al WAF

### Instalar un OWASP ModSecurity Core Rule Set (CRS) actualizado

Primero debemos mover y cambiar el nombre del siguiente archivo de ModSecurity para tener una configuración recomendada predeterminada inicial sudo mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf

Descargamos la última versión de OWASP ModSecurity CRS desde GitHub: git clone <https://github.com/SpiderLabs/owasp-modsecurity-crs.git>

Copiar la carpeta descargada a /volume\_data/rules de la infraestructura del lab10 hará que dicha carpeta aparezca en el sistema de archivos del contenedor proxy (/rules). Solo entonces podemos continuar.

Una vez dentro del proxy, vete al directorio donde están las reglas descargadas y copia y cambia el nombre crs-setup.conf.example a crs-setup.conf. Luego copia el directorio rules/ también junto con su contenido.

cd owasp-modsecurity-crs

cp crs-setup.conf.example /etc/modsecurity/crs-setup.conf

cp -r rules/ /etc/modsecurity/

Ahora debemos modificar el fichero /etc/apache2/mods-available/security2.conf para que coincida con la ruta de los archivos descargados. Esto significa modificar el valor del primer IncludeOptional para que coincida con la ruta del archivo crs-setup.conf (si no tiene ya un valor correcto) y añadir otra directiva Include que apunte al conjunto de reglas. Debe quedar así:

Texto

Descripción generada automáticamente

### Comprueba que el WAF está en funcionamiento

• Vete hasta la configuración predeterminada de Apache2 y agrega dos directivas adicionales en el archivo de configuración del sitio web predeterminado (/etc/apache2/sites-enabled/000- default.conf).

o La primera directiva habilita ModSecurity en modo de intercepción (On detecta amenazas entrantes y las bloquea) en lugar del modo de detección (DetectionOnly detecta amenazas entrantes pero solo las registra en un log, lo cual es útil cuando se prueba que el WAF no está bloqueando solicitudes legítimas).

Texto

Descripción generada automáticamenteo La segunda directiva agrega una nueva regla de prueba a ModSecurity que se incorporará al conjunto de reglas CRS. Esta regla es muy simple, y se activa cuando alguien usa el parámetro testarg en una URL con un valor que contiene la cadena ssi. La respuesta del WAF es denegar (deny) la solicitud que sirve una página de error con el estado HTTP 403 (status:403), registrando el incidente con el mensaje "regla de prueba SSI disparada!". Este es solo un ejemplo simple de cómo se crean las reglas WAF, que también verifica que el motor de reglas funciona correctamente.

Reinicia Apache 2 de nuevo y, a continuación, accede a la página. Si se carga correctamente, activa intencionadamente la regla para terminar esta parte del ejercicio.

• La segunda prueba que vamos a hacer es comprobar que las reglas CRS se están leyendo. Para ello, activamos a posta una advertencia de ataque de ejecución remota de comandos (RCE), creando una petición que coincida con las reglas CRS que evitan este tipo de ataques, como pasar un parámetro cualquiera con valor /bin/bash

### WAF contra "el mal"

Ejecuta el escáner de servidores web Nikto instalado en el contenedor de ataque Kali de la infraestructura del Lab 10 contra el proxy para ver qué sucede.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Ejercicio L10B2\_WAFPROT: WAF contra ataques web

Comprobar si el WAF está realmente bloqueando peticiones web maliciosas (XSS y SQL Injection)

Seguiremos un procedimiento muy similar al anterior para ver si el WAF se activa cuando se realiza un intento de ataque XSS. De esta forma, en lugar de pasar el ataque usando GET pondremos el payload del ataque como parámetro POST gracias a la opción -d del comando curl. Para activar el WAF pasaremos una cadena de prueba XSS típica:

<script>alert(‘test’)</script> con

curl <proxy IP>/ -d “<script>alert(‘test’)</script>”

Inyección SQL Al igual que las pruebas XSS, realizamos una prueba de detección de ataques de inyección SQL pasando como parámetro POST a los usuarios de la cadena DROP DATABASE;.

# Bloque 3: Herramientas AST para el desarrollo de aplicaciones

## Ejercicio L10B3\_SASTSCA: Herramienta de detección automatizada de vulnerabilidades en aplicaciones SonarQube (SAST, con módulo SCA opcional)

Elegir un proyecto en un lenguaje soportado para analizar (preferiblemente en Java o C#, usando Maven o Gradle)

Instalar Maven en la máquina virtual de Ubuntu (sudo apt install maven)

Ejecuta un contenedor SonarQube ejecutando el script run\_sonarqube.sh

Inicia sesión en http://localhost:9000 con las credenciales de administrador del sistema (login=admin, password=admin). Cambiar la contraseña en el primer inicio de sesión es obligatorio.

• Para analizar un proyecto:

o Haz clic en Create new project manually.

o Asigna al proyecto una Project key y un Display name y haz clic en el Configure (ej. SAST\_SSI).

o Haga clic en Analyze your repository locally.

o Mira el token generado y haz clic en continuar. Luego, elije Maven. Esto se debe a que nuestro proyecto de prueba se construye con Maven

o Ahora te mostrará un comando para analizar la aplicación asociada con el proyecto SonarQube que acabas de crear. Ten en cuenta que el token y la clave del proyecto son los que se generaron al configurar el análisis.

o Vete a la carpeta del proyecto que descomprimiste en tu máquina virtual y ejecuta el comando que te mostró la interfaz de SonarQube dentro de ella para analizar el proyecto. El comando será algo como esto.

Texto

Descripción generada automáticamente

• Una vez finalizado el análisis, la interfaz de usuario de SonarQube se actualizará automáticamente con los resultados de este -> importante Security Hotspots

Nota: para lo siguiente es recomendable aumentar la ram a 4gb, apagando la máquina primero.

Para incorporar una herramienta SCA completa, abrimos el pom.xml del proyecto y añadimos las líneas que salen en rojo.

Texto

Descripción generada automáticamente

• Repite el mismo proceso del análisis SAST anterior para iniciar el análisis SonarQube. Ahora, como parte del análisis automatizado de SonarQube, también se hará un análisis SCA de las dependencias del proyecto de prueba.

se generará un informe en

<carpeta donde descomprimimos la aplicación de prueba>/dvjamaster/target/dependency-check-report.html

## Ejercicio L10B3\_HARDPROXY: Combinación del proxy con técnicas anteriores

## Ejercicio L10B3\_DAST: Uso de un DAST contra una aplicación

Necesitaremos una aplicación “víctima”

Si no fuera el caso, una buena forma de probarla es instalando y configurando la Damn Vulnerable Web App (DVWA)

Necesitaremos Docker para ello: docker run --rm -it -p 8080:80 vulnerables/web-dvwa (aunque es mejor poner otro puerto, porque el 8080 choca con ZAP).

Ejecutamos la herramienta con owasp-zap y dejamos que se actualicen todos los add-ons que vienen de serie para lograr el mejor funcionamiento posible (Update All).

### Escaneo activo

Quick Start-Automated Scan

Le damos simplemente la URL de la web a analizar y tras un tiempo nos dará una lista con los problemas de seguridad que ha encontrado en la misma, que podremos analizar clicando en cada uno de ellos para saber sus detalles, gravedad, etc. Veremos también que ha hecho spidering de la aplicación, es decir, que nos mostrará un árbol de páginas que componen el sitio examinado.

### Escaneo manual

Problema del escaneo automático: si tienes una pantalla de login para una zona privada, la herramienta no podrá pasar a examinar esa zona privada porque no sabe las credenciales.

Nosotros navegamos por la aplicación en un navegador que ZAP nos muestra, nos autenticamos normalmente (estamos en un escenario de test de seguridad, se supone que tenemos credenciales) y el proxy va a haciendo pruebas de seguridad (y spidering) a medida que navegamos por la aplicación normalmente.

Mientras navegamos por la web ahora vemos controles adicionales que nos indican el estado de los hallazgos de la herramienta en la misma. Para entender bien lo que está pasando mientras navegamos por la web de pruebas, tenemos que considerar lo siguiente:

• Lo que va encontrando se clasifica por gravedad en la botonera superpuesta del lado izquierdo de la web.

• Al lado derecho de la web podemos activar el escaneo activo anterior en cualquier momento, señal de que podemos lanzarlo una vez superada la pantalla de login

• En función de lo que hayamos hecho, la herramienta dará este aviso, que simplemente aceptamos y pulsamos en “Start” para hacer el escaneo, cuyo progreso podremos ver sobre los propios iconos del HUD.

• Las vulnerabilidades encontradas del escaneo manual o híbrido aparecerán de nuevo en la interfaz de ZAP.

• Este modo de trabajo permite usar una opción especial, el “Attack Mode”, representado por una mirilla de un arma y activado haciendo clic en la misma. El Attack Mode funciona a medida que vamos navegando por las páginas y se puede hacer un active scan en paralelo. Esto es una forma especial de trabajo de ZAP donde hará pruebas intrusivas a todas las páginas que pueda llegar a partir de las que estemos navegando, aumentando por tanto las opciones de encontrar más problemas, pero también el riesgo de “romper” algo en la aplicación por meter datos corruptos, hacer fuzzing, etc. Por ese motivo, antes de usar este “Attack Mode” se recomienda encarecidamente hacer dos cosas:

1. Desconectar la red de la máquina de pruebas por si acaso, si es posible. Es muy frecuente que muchas páginas usen servicios de terceros para algo (por ejemplo, publicidad, trackers...) y se podrían llevar un ataque por accidente si no tenemos cuidado. Por supuesto, solo se podría hacer si la página es capaz de funcionar de esta forma. En cualquier caso, nunca navegar a la web de un servicio de terceros en este modo, porque podrías cometer un delito.

2. Hacer las pruebas sobre un clon de la web para evitar interacciones no deseadas y corrupción de datos por las acciones de ZAP.

• Finalmente, no olvidar que ZAP tiene la opción de obtener informes (Reports) donde se resumen todas las vulnerabilidades encontrar, su gravedad y otra información, ideal para pasar al equipo de desarrollo para que procedan a resolverlas.