

Stredná priemyselná škola elektrotechnická
Komenského 44, 040 01 Košice

Forenzná analýza operačnej pamäte Linux

Autor: Marek Horňák
Trieda: 4.B
Školský rok: 2023/2024
Odbor: 2561M – Informačné a sieťové technológie

Obsah

0. Úvod.....	3
1. Lab1 – Linux.....	3
2. Lab2 – Linux.....	6
3. Lab3 – Linux.....	6
4. Referencia a syntax prikazov	6

0. Úvod

Tento učebný materiál opisuje analýzu obrazu operačnej pamäte počítača s Linuxom. V prvých dvoch úlohách (laboch) budete podľa návodu analyzovať obrazy operačnej pamäte a hľadať v nich flagy. Na rozdiel od materiálu s Windowsom bude táto analýza vo forme pracovných listov. V jednotlivých úlohách bude popísané, aké údaje máte v pamäti nájsť. Teda už nehľadáme flagy ako také, ale už reálne údaje. V každej úlohe je potrebné nájsť 5 údajov. Tretiu úlohu už budete analyzovať svojpomocne za použitia skúseností nadobudnutých v prvých dvoch úlohách.

1. Lab1 – Linux

Do virtuálneho počítača s Ubuntu si z Github stránky stiahnite súbor **lab1_lin.raw** a premiestnite ho do priečinka s Volatility3. Nebudete tak musieť špecifikovať celú cestu k súboru. Presuňte sa do priečinka s Volatility3, aby bol vašim pracovným priečinkom.

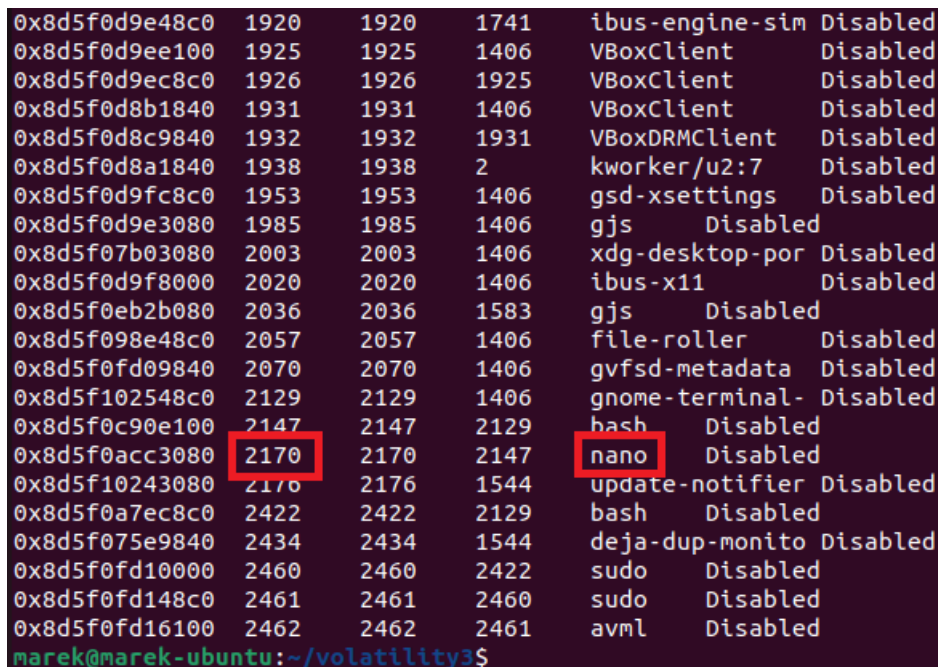
Znenie pracovného listu:

1. napíšte číslo procesu, ktorý patrí programu nano
2. napíšte, aké terminály boli otvorené počas behu počítača
3. napíšte, aký command netypický pre Linux bol spustený v termináli
4. napíšte celé meno súboru aj s absolútnou cestou, ktorý bol otvorený programom file-roller
5. napíšte, koľko sieťových spojení počítača bolo v stave LISTEN

Úloha 1.

Na vypísanie zoznamu bežiacich procesov slúži plugin *linux.pslist*:

```
python3 vol.py -f lab1_lin.raw linux.pslist
```



Address	PID	PPID	UID	Process Name	Status
0x8d5f0d9e48c0	1920	1920	1741	ibus-engine-sim	Disabled
0x8d5f0d9ee100	1925	1925	1406	VBoxClient	Disabled
0x8d5f0d9ec8c0	1926	1926	1925	VBoxClient	Disabled
0x8d5f0d8b1840	1931	1931	1406	VBoxClient	Disabled
0x8d5f0d8c9840	1932	1932	1931	VBoxDRMClient	Disabled
0x8d5f0d8a1840	1938	1938	2	kworker/u2:7	Disabled
0x8d5f0d9fc8c0	1953	1953	1406	gsd-xsettings	Disabled
0x8d5f0d9e3080	1985	1985	1406	gjs	Disabled
0x8d5f07b03080	2003	2003	1406	xdg-desktop-por	Disabled
0x8d5f0d9f8000	2020	2020	1406	ibus-x11	Disabled
0x8d5f0eb2b080	2036	2036	1583	gjs	Disabled
0x8d5f098e48c0	2057	2057	1406	file-roller	Disabled
0x8d5f0fd09840	2070	2070	1406	gvfsd-metadata	Disabled
0x8d5f102548c0	2129	2129	1406	gnome-terminal-	Disabled
0x8d5f0c90e100	2147	2147	2129	bash	Disabled
0x8d5f0acc3080	2170	2170	2147	nano	Disabled
0x8d5f10243080	2176	2176	1544	update-notifier	Disabled
0x8d5f0a7ec8c0	2422	2422	2129	bash	Disabled
0x8d5f075e9840	2434	2434	1544	deja-dup-monito	Disabled
0x8d5f0fd10000	2460	2460	2422	sudo	Disabled
0x8d5f0fd148c0	2461	2461	2460	sudo	Disabled
0x8d5f0fd16100	2462	2462	2461	avml	Disabled

marek@marek-ubuntu:~/volatility3\$

Obrázok 1: Bežiace procesy a číslo procesu nano

Úloha 2.

Terminály v Linuxe sú označované ako TTY – znamená to teletype, no označujú sa tým terminály. Na vypísanie spustených terminálov vo Volatility3 existuje plugin *linux.tty_check*:

```
python3 vol.py -f lab1_lin.raw linux.tty_check
```

```
marek@marek-ubuntu:~/volatility3$ python3 vol.py -f /media/sf_VMshare/lab1_lin.raw linux.tty_check
Volatility 3 Framework 2.5.2
Progress: 100.00 Stacking attempts finished
Name Address Module Symbol
tty2 0xffff96029d40 __kernel__ n_tty_receive_buf
tty6 0xffff96029d40 __kernel__ n_tty_receive_buf
marek@marek-ubuntu:~/volatility3$
```

Obrázok 2: Spustené terminály

TTY1 je pre login screen, TTY2 je pre Desktop GUI a TTY3 – 7 sú command line terminály.

Úloha 3.

Ak chceme zistiť, čo bolo zadané v termináli Linuxu, treba použiť plugin *linux.bash*. Tento plugin dokáže extrahovať celú históriu bash commandov, pretože bash si svoju históriu ukladá do súboru a Volatility do tohto súboru vie vstúpiť:

```
python3 vol.py -f lab1_lin.raw linux.bash
```

```
2422 bash 2024-02-02 18:56:00.000000 sudo useradd marek vboxsf
2422 bash 2024-02-02 18:56:00.000000 ♦♦H♦♦♦
2422 bash 2024-02-02 18:56:00.000000 sudo usermod vboxsf marek
2422 bash 2024-02-02 18:56:00.000000 sudo avml /media/sf_VMshare/
2422 bash 2024-02-02 18:56:00.000000 avml /media/sf_VMshare/
2422 bash 2024-02-02 18:56:00.000000 ./S0m3_c0mM4nD
2422 bash 2024-02-02 18:56:00.000000 ./S0m3_c0mM4nD
2422 bash 2024-02-02 18:56:00.000000 q♦♦ ♦♦
2422 bash 2024-02-02 18:56:00.000000 ping 8.8.8.8
2422 bash 2024-02-02 18:56:00.000000 ls -l /media/
2422 bash 2024-02-02 18:56:00.000000 sudo usermod marek vboxsf
2422 bash 2024-02-02 18:56:00.000000 sudo usermod -a -G vboxsf marek
2422 bash 2024-02-02 18:56:00.000000 ./ S0m3_c0mM4nD
2422 bash 2024-02-02 18:56:00.000000 nano S0m3_c0mM4nD
2422 bash 2024-02-02 18:56:00.000000 sudo usermod vboxsf marek
2422 bash 2024-02-02 18:56:00.000000 ip a
2422 bash 2024-02-02 18:56:00.000000 avml /media/sf_VMshare/
2422 bash 2024-02-02 18:57:13.000000 ./S0m3_c0mM4nD
2422 bash 2024-02-02 18:57:40.000000 sudo ./avml /media/sf_VMshare/lab1_lin.raw
marek@marek-ubuntu:~/volatility3$
```

Obrázok 3: História bash commandov

Úloha 4.

Ak chceme zistiť, aký súbor bol otvorený pomocou programu file-roller, musíme si najprv zistiť číslo jeho procesu, pod ktorým bol spustený. Opäť použijeme plugin *linux.pslist*:

```
python3 vol.py -f lab1_lin.raw linux.pslist
```

```

0x8d5f0d8c9840 1932 1932 1931 VBoxDRMClient Disabled
0x8d5f0d8a1840 1938 1938 2 kworker/u2:7 Disabled
0x8d5f0d9fc8c0 1953 1953 1406 gsd-xsettings Disabled
0x8d5f0d9e3080 1985 1985 1406 gjs Disabled
0x8d5f07b03080 2003 2003 1406 xdg-desktop-por Disabled
0x8d5f0d9f8000 2020 2020 1406 ibus-x11 Disabled
0x8d5f0eb2b080 2036 2036 1583 qis Disabled
0x8d5f098e48c0 2057 2057 1406 file-roller Disabled
0x8d5f0fd09840 2070 2070 1406 gvfsd-metadata Disabled
0x8d5f102548c0 2129 2129 1406 gnome-terminal- Disabled
0x8d5f0c90e100 2147 2147 2129 bash Disabled
0x8d5f0acc3080 2170 2170 2147 nano Disabled
0x8d5f10243080 2176 2176 1544 update-notifier Disabled
0x8d5f0a7ec8c0 2422 2422 2129 bash Disabled
0x8d5f075e9840 2434 2434 1544 deja-dup-monito Disabled
0x8d5f0fd10000 2460 2460 2422 sudo Disabled
0x8d5f0fd148c0 2461 2461 2460 sudo Disabled
0x8d5f0fd16100 2462 2462 2461 avml Disabled
marek@marek-ubuntu:~/volatility3$

```

Obrázok 4: Bežiace procesy a číslo procesu file-roller

Na základe čísla procesu programu file-roller si môžeme vyfiltrovať zo všetkých otvorených súborov v pamäti a zobrazit' si len tie, ktoré daný proces používal. Použijeme plugin *linux.lsof* s filtrovaním podľa procesu:

```
python3 vol.py -f lab1_lin.raw linux.lsof --pid 2057
```

```

marek@marek-ubuntu:~/volatility3$ python3 vol.py -f /media/sf_VMshare/lab1_lin.raw linux.lsof --pid 2057
Volatility 3 Framework 2.5.2
Progress: 100.00 Stacking attempts finished
PID Process FD Path
2057 file-roller 0 /dev/null
2057 file-roller 1 socket:[25987]
2057 file-roller 2 socket:[25988]
2057 file-roller 3 anon_inode:[66]
2057 file-roller 4 anon_inode:[66]
2057 file-roller 5 anon_inode:[66]
2057 file-roller 6 socket:[25994]
2057 file-roller 7 socket:[25996]
2057 file-roller 8 anon_inode:[66]
2057 file-roller 9 socket:[25999]
2057 file-roller 10 /:[87]
2057 file-roller 11 socket:[26001]
2057 file-roller 12 anon_inode:[66]
2057 file-roller 13 anon_inode:[66]
2057 file-roller 14 /home/marek/Topsecret.rar
marek@marek-ubuntu:~/volatility3$

```

Obrázok 5: Súbory otvorené procesom file-roller

Úloha 5.

Každý počítač s prístupom na internet po spustení predvolene otvára sieťové spojenia potrebné pre jeho správne fungovanie. Tých procesov je pomerne veľa a dobrý forenzný analytik by sa v nich mal vedieť orientovať, pretože niekedy zohrávajú kľúčovú rolu v odhalení kyberzločinu. Na vyriešenie tejto úlohy použijeme plugin *linux.sockstat* spolu s programom *grep* a *wc*. Tento skript extrahuje z pamäte všetky sieťové spojenia, vyfiltrovať všetky v stave LISTEN a spočíta ich. Vo výstupe nenájdeme ani jedno sieťové spojenie, pretože skript vypíše len ich počet:

```
Python3 vol.py -f lab1_lin.raw linux.sockstat | grep LISTEN | wc -l
```

```
marek@marek-ubuntu:~/volatility3$ python3 vol.py -f 54ogress: 100.00 Stacking attempts f
marek@marek-ubuntu:~/volatility3$
```

Obrázok 6: Počet sieťových spojení v stave LISTEN

Výstup nie je úplne najlepší, no vieme z toho vyčítať, že sieťových spojení v stave LISTEN bolo 54.

Správne odpovede:

1. 2170
2. tty2 a tty6
3. S0m3_c0mM4nD
4. /home/marek/Topsecret.rar
5. 54

2. Lab2 – Linux

Do virtuálneho počítača s Ubuntu si z Github stránky stiahnite súbor **lab2_lin.raw** a premiestnite ho do priečinka s Volatility3. Nebudete tak musieť špecifikovať celú cestu k súboru. Presuňte sa do priečinka s Volatility3, aby bol vašim pracovným priečinkom.

Znenie pracovného listu:

1. napíšte, aký program bol nainštalovaný počas behu počítača
2. napíšte, koľko elf súborov patrí procesu s názvom cron
3. napíšte celé meno súboru aj s absolútnou cestou, ktorý bol otvorený procesom číslo 14375
4. napíšte najnižšie a najvyššie číslo procesu, ktorému patrí sieťové spojenie v stave ESTABLISHED
5. napíšte major:minor index mountovacieho pointu s mount ID 123

Úloha 1.

Inštalácia dodatočných programov a balíčkov sa v Linuxe väčšinou realizuje pomocou nejakého programu (správcu balíčkov – package manager). V Ubuntu je to program apt. Inštalácia sa spúšťa v príkazovom riadku príkazom *sudo apt install package*. Dôležitou informáciou pre nás je, že sa spúšťa v príkazovom riadku, takže môžeme použiť plugin *linux.bash* na vypísanie histórie bashu:

```
python3 vol.py -f lab2_lin.raw linux.bash
```

```

14573 bash 2024-02-06 17:21:06.000000 ls
14573 bash 2024-02-06 17:21:06.000000 mkdir /home/marek/Desktop/new_dir
14573 bash 2024-02-06 17:21:06.000000 sudo ./avml /media/sf_VMshare/
14573 bash 2024-02-06 17:21:06.000000 ./avml
14573 bash 2024-02-06 17:21:06.000000 cat /etc/group
14573 bash 2024-02-06 17:21:06.000000 sudo useradd marek vboxsf
14573 bash 2024-02-06 17:21:06.000000 sudo apt install net-tools
14573 bash 2024-02-06 17:21:06.000000 sudo ./avml /media/sf_VMshare/lab2_lin.raw
14573 bash 2024-02-06 17:21:06.000000 ♦♦S♦♦♦
14573 bash 2024-02-06 17:21:06.000000 sudo apt update
14573 bash 2024-02-06 17:21:06.000000 sudo avml /media/sf_VMshare/
14573 bash 2024-02-06 17:21:06.000000 ♦♦H♦H♦|
14573 bash 2024-02-06 17:21:06.000000 avml /media/sf_VMshare/
14573 bash 2024-02-06 17:21:06.000000 ♦삼8V
14573 bash 2024-02-06 17:21:06.000000 sudo ./avml /media/sf_VMshare/lab2_lin.raw
14573 bash 2024-02-06 17:21:06.000000 ls -l /media/
14573 bash 2024-02-06 17:21:06.000000 sudo usermod marek vboxsf
14573 bash 2024-02-06 17:21:06.000000 sudo usermod -a -G vboxsf marek
14573 bash 2024-02-06 17:21:06.000000 sudo chmod u+x avml
14573 bash 2024-02-06 17:21:06.000000 sudo usermod -a -G marek vboxadd
14573 bash 2024-02-06 17:21:06.000000 sudo usermod vboxsf marek
14573 bash 2024-02-06 17:21:06.000000 avml /media/sf_VMshare/
14573 bash 2024-02-06 17:21:21.000000 cd Desktop/
14573 bash 2024-02-06 17:21:26.000000 cd
14573 bash 2024-02-06 17:22:03.000000 sudo ./avml /media/sf_VMshare/lab2_lin.raw
marek@marek-ubuntu:~/volatility3$

```

Obrázok 7: Nainštalovaný balíček net-tools

Úloha 2.

Executable and Linking Format (ELF) je spoločný štandardný súborový formát pre spúšťaťelné súbory, objektový kód, zdieľané knižnice a core dumpy. Cron je program, ktorý slúži na plánovanie pravidelne spúšťaných úloh. Používa sa, keď potrebujeme, aby systém pravidelne vykonával nejaké úlohy. Napríklad, aby sa pravidelne aktualizoval. Dokážeme si taktiež vybrať čas, kedy sa má daná úloha spustiť. Zoznam elf súborov používaných programom cron si vieme zobrazit' pomocou pluginu *linux.elfs*:

```
python3 vol.py -f lab2_lin.raw linux.elfs
```

```

marek@marek-ubuntu:~/volatility3$ python3 vol.py -f /media/sf_VMshare/lab2_lin.raw linux.elfs | grep cron
514grecron 0x563013a11000 0x563013a14000 /usr/sbin/cron Disabled
514 cron 0x7f1daff69000 0x7f1daff6b000 /usr/lib/x86_64-linux-gnu/libpcre2-8.so.0.10.4 Disabled
514 cron 0x7f1db0000000 0x7f1db0028000 /usr/lib/x86_64-linux-gnu/libc.so.6 Disabled
514 cron 0x7f1db02a1000 0x7f1db02a7000 /usr/lib/x86_64-linux-gnu/libselinux.so.1 Disabled
514 cron 0x7f1db02f1000 0x7f1db02f3000 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2 Disabled
514 cron 0x7f1db031d000 0x7f1db0328000 /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2 Disabled
514 cron 0x7fffa23d3000 0x7fffa23d5000 [vdso] Disabled

```

Obrázok 7: Elf súbory programu cron

Úloha 3.

Ak potrebujeme zistiť, aký súbor bol otvorený daným číslom procesu, stačí použiť plugin *linux.lsof* a vyfiltrovať výstup podľa daného procesu. My si však najprv vypíšeme, aký proces to vlastne bol, aby sme vedeli, aký typ súborov treba očakávať (môžu to byť obrázky, archívy, textové súbory). Použijeme plugin *linux.pslist* a vyfiltroujeme si výstup len na ten konkrétny proces:

```
python3 vol.py -f lab2_lin.raw linux.pslist --pid 14375
```



```
marek@marek-ubuntu:~/volatility3$ python3 vol.py -f /media/sf_VMshare/lab2_lin.raw linux.pslist --pid 14375
Volatility 3 Framework 2.5.2
Progress: 100.00 Stacking attempts finished
OFFSET (V) PID TID PPID COMM File output
0x9e6a099a9840 14375 14375 1411 file-roller Disabled
marek@marek-ubuntu:~/volatility3$
```

Obrázok 8: Zoznam procesov filtrovaný podľa procesu 14375

Teraz už vieme, že to bol program file-roller, takže to bude asi nejaký archív. Teraz použijeme plugin *linux.lsof* a opäť si výstup obmedzíme len na program file-roller.

Python3 vol.py -f lab2_lin.raw linux.lsof --pid 14375

```
marek@marek-ubuntu:~/volatility3$ python3 vol.py -f /media/sf_VMshare/lab2_lin.raw linux.lsof --pid 14375
Volatility 3 Framework 2.5.2
Progress: 100.00 Stacking attempts finished
PID Process FD Path
14375 file-roller 0 /dev/null
14375 file-roller 1 socket:[365557]
14375 file-roller 2 socket:[365558]
14375 file-roller 3 anon_inode:[66]
14375 file-roller 4 anon_inode:[66]
14375 file-roller 5 anon_inode:[66]
14375 file-roller 6 socket:[403316]
14375 file-roller 7 socket:[403320]
14375 file-roller 8 anon_inode:[66]
14375 file-roller 9 socket:[403324]
14375 file-roller 10 /:[900]
14375 file-roller 11 socket:[403326]
14375 file-roller 12 anon_inode:[66]
14375 file-roller 13 /home/marek/Documents/archives/monday/Suspicious.rar
14375 file-roller 14 anon_inode:[66]
marek@marek-ubuntu:~/volatility3$
```

Obrázok 9: Súbory otvorené programom file-roller

Úloha 4.

Na vypísanie sieťových spojení slúži plugin *linux.sockstat*. Výstup si vyfiltrujeme len na spojenia v stave ESTABLISHED:

python3 vol.py -f lab2_lin.raw linux.sockstat | grep ESTABLISHED

```
marek@marek-ubuntu:~/volatility3$ python3 vol.py -f /media/sf_VMshare/lab2_lin.raw linux.sockstat | grep ESTABLISHED
4026531840 100.01 17 0x9e6a08c4e200ptAF_UNIXhSTREAM - - 360222 /run/dbus/sy
4026531840 1 40 0x9e6a0ed16a80 AF_UNIX STREAM - /run/systemd/journal/stdout
4026531840 1 46 0x9e6a08c4c440 AF_UNIX STREAM - /run/systemd/io.system.Manag
4026531840 1 114 0x9e6a09f16ec0 AF_UNIX STREAM - /run/systemd/journal/stdout
4026531840 1 115 0x9e6a09f17b80 AF_UNIX STREAM - /run/systemd/journal/stdout
4026531840 1 117 0x9e6a08c4d980 AF_UNIX STREAM - /run/systemd/journal/stdout
4026531840 1 118 0x9e6a08c4f300 AF_UNIX STREAM - /run/systemd/journal/stdout
4026531840 1 119 0x9e6a07db2200 AF_UNIX STREAM - /run/systemd/journal/stdout
4026531840 1 120 0x9e6a06f6a640 AF_UNIX STREAM - /run/systemd/journal/stdout
4026531840 1 121 0x9e6a08c4ccc0 AF_UNIX STREAM - /run/systemd/journal/stdout
4026531840 1 122 0x9e6a07db2640 AF_UNIX STREAM - /run/systemd/journal/stdout
4026531840 1 123 0x9e6a07dba200 AF_UNIX STREAM - /run/systemd/journal/stdout
```

Obrázok 10: Sieťové spojenia - najnižšie číslo procesu


```

4026531840 14375 2 0x9e6a09f15100 AF_UNIX STREAM - - 365558
4026531840 14375 6 0x9e6a0b445100 AF_UNIX STREAM - - 403316
4026531840 14375 7 0x9e6a0b446640 AF_UNIX STREAM - - 403320
4026531840 14375 9 0x9e6a0b457740 AF_UNIX STREAM - - 403324
4026531840 14375 11 0x9e6a0b456640 AF_UNIX STREAM - - 403326
4026531840 14544 1 0x9e6a0ed14000 AF_UNIX STREAM - - 408112
4026531840 14544 2 0x9e6a0ed14000 AF_UNIX STREAM - - 408112
4026531840 14544 3 0x9e6a0c134000 AF_UNIX STREAM - - 408122
4026531840 14544 7 0x9e6a0c136200 AF_UNIX STREAM - - 408123
4026531840 14544 9 0x9e6a0fcba200 AF_UNIX STREAM - - 408129
4026531840 14544 11 0x9e6a0fcbbb80 AF_UNIX STREAM - - 408131
4026531840 16462 11 0x9e6a07dh9dc0 AF_UNIX STREAM - - 481164
4026531840 16613 12 0x9e6a07c29dc0 AF_UNIX STREAM - - 481165
marek@marek-ubuntu:~/volatility3$

```

Obrázok 11: Sieťové spojenia - najvyššie číslo procesu

Úloha 5.

Mountovací bod možno jednoducho opísať ako adresár na prístup k údajom uloženým na pevných diskoch. Presnejšie povedané, mountovací bod je (zvyčajne prázdny) adresár v aktuálne dostupnom súborovom systéme, ku ktorému je pripojený (pripojený) ďalší súborový systém. Minor a major index sú čísla, pomocou ktorých identifikujeme daný adresár (mount point). Na vypísanie týchto informácií z pamäte slúži plugin *linux.mountinfo*. Volatility neponúka možnosť filtrovať výstup podľa mount ID, takže musíme použiť *grep*:

```
python3 vol.py -f lab2_lin.raw linux.mountinfo | grep 123
```

```

marek@marek-ubuntu:~/volatility3$ python3 vol.py -f /media/sf_VMshare/lab2_lin.raw linux.mountinfo | grep 123
4026531841 100.0123 27 7:12 / /snap/snapd/20290
4026532299 1239 418 0:23 /snapd/ns /run/snapd/ns
4026532300 1238 548 0:23 /snapd/ns /run/snapd/ns
4026532300 1237 512 0:23 /snapd/ns /run/snapd/ns
4026532358 1236 786 0:23 /snapd/ns /run/snapd/ns
4026532358 1235 750 0:23 /snapd/ns /run/snapd/ns
4026532356 1234 628 0:23 /snapd/ns /run/snapd/ns
4026532356 1233 592 0:23 /snapd/ns /run/snapd/ns
4026532357 1232 707 0:23 /snapd/ns /run/snapd/ns
4026532357 1231 671 0:23 /snapd/ns /run/snapd/ns
4026532303 1123 927 0:49 / /media/sf_VMshare
4026532303 1230 969 0:23 /snapd/ns /run/snapd/ns
4026532490 2123 2084 7:13 / /snap/snapd-desktop-int
4026532205 317 181 0:23 /systemd/inaccessible/reg
marek@marek-ubuntu:~/volatility3$

```

Obrázok 12: Mountinfo mount ID 123

Príkaz našiel aj mount pointy, kde sa nachádzalo číslo 123, no z výstupu môžeme jasne vidieť, že mount point s mount ID 123 má Major:Minor index 7:12.

Správne odpovede:

1. net-tools
2. 7
3. /home/marek/Documents/archives/monday/Suspicious.rar
4. najnižšie: 1, najvyššie: 16613
5. 7:12

3. Lab3 – Linux

Do virtuálneho počítača s Ubuntu si z Github stránky stiahnite súbor **lab2_lin.raw** a premiestnite ho do priečinka s Volatility3. Nebudete tak musieť špecifikovať celú cestu k súboru. Presuňte sa do priečinka s Volatility3, aby bol vašim pracovným priečinkom.

Znenie pracovného listu:

1. napíšte, akému pamäťovému offsetu prislúcha kernel modul soundcore
2. napíšte celé meno súboru aj s absolútnou cestou, ktorý bol otvorený programom file-roller
3. napíšte index idt modulu asm_exc_spurious_interrupt_bug
4. napíšte čísla všetkých procesov, v ktorých sa potenciálne nachádza malware
5. napíšte meno používateľa Linuxu, z ktorého je tento dump (extrakt)

4. Referencia a syntax príkazov

Tu nájdete krátky popis a syntax každého pluginu, ktorý budete potrebovať pri analýze jednotlivých obrazov operačnej pamäte:

- `python3 vol.py -f memory_dump.raw linux.pslist`
 - vypíše zoznam bežiacich procesov a ich PID
- `python3 vol.py -f memory_dump.raw linux.bash`
 - vypíše históriu bash commandov
- `python3 vol.py -f memory_dump.raw linux.lsof`
 - vypíše zoznam otvorených súborov aj procesov, ktoré ich používali
- `python3 vol.py -f memory_dump.raw linux.tty_check`
 - vypíše zoznam spustených terminálov
- `python3 vol.py -f memory_dump.raw linux.sockstat`
 - vypíše zoznam aktívnych sieťových spojení
- `python3 vol.py -f memory_dump.raw linux.elfs`
 - vypíše zoznam elf súborov a procesov, ktoré ich používali
- `python3 vol.py -f memory_dump.raw linux.mountinfo`
 - vypíše zoznam mountovacích bodov, ich mount ID a Major:Minor index