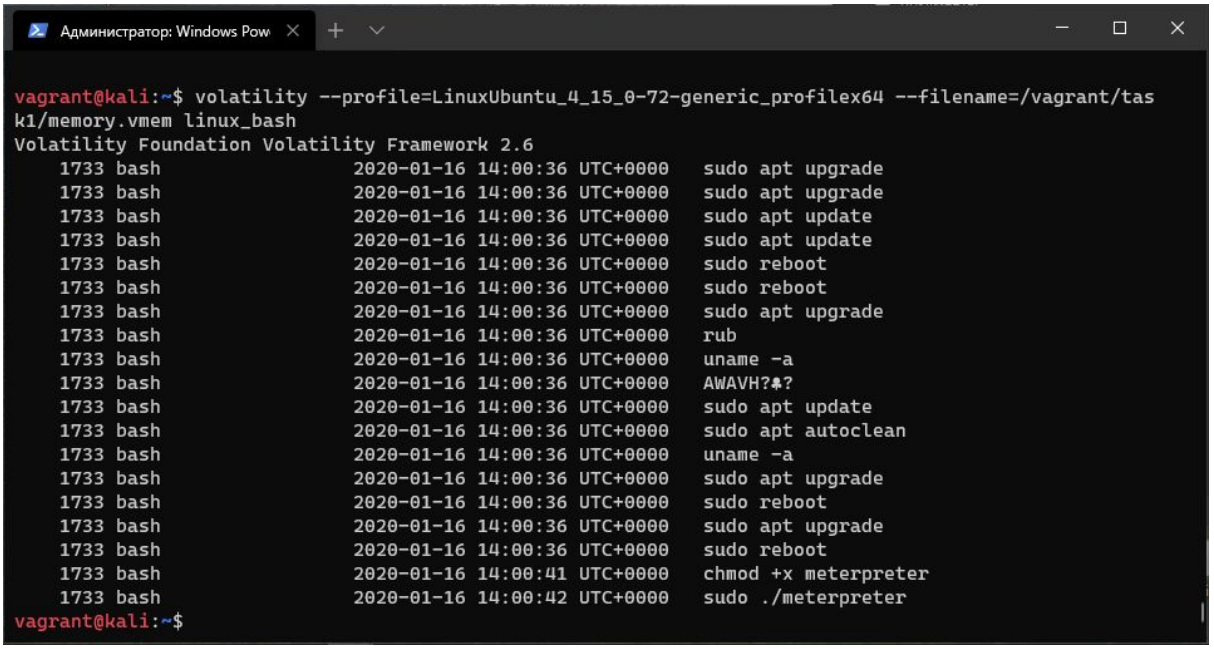
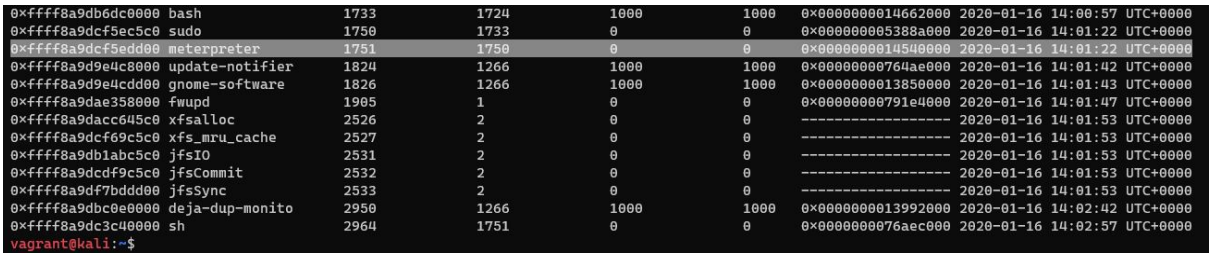
Task 1. Анализ дампа памяти с помощью утилиты Volatility

1. Смотрим историю bash

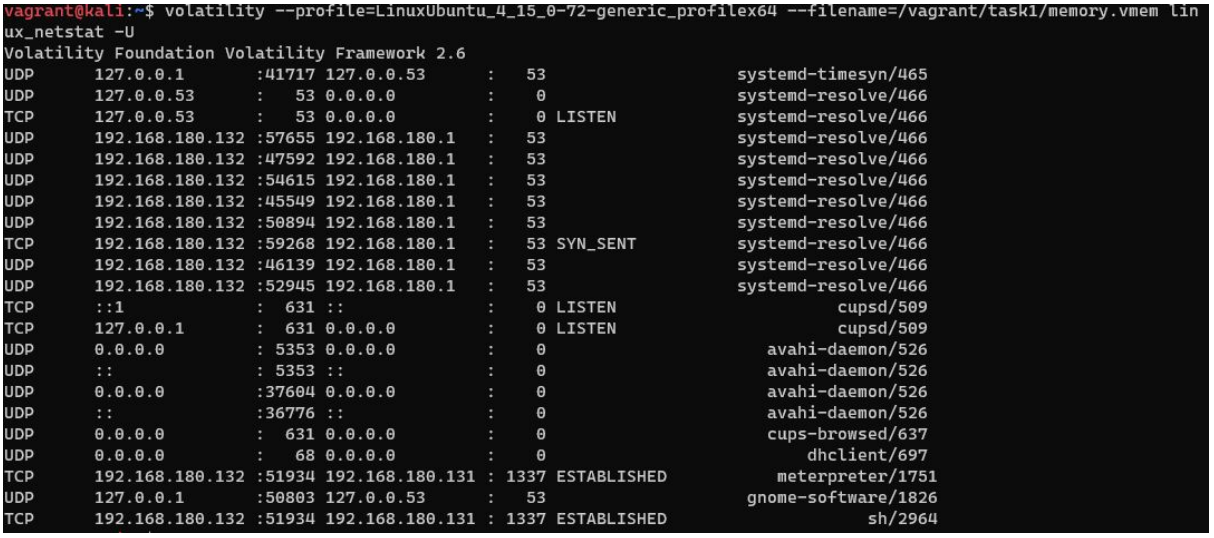


из истории видно, что был запущен meterpreter из-под суперпользователя.

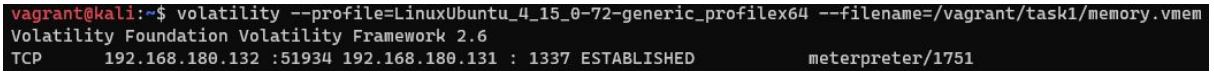
2. Ищем pid и адрес памяти meterpreter с помощью команды linux_pslist



3. Командой linux_netstat просматриваем сетевые соединения.



использовал флаг -U чтобы игнорировать unix сокеты. Отфильтровав соединения по pid выясняем meterpreter установил соединение с другим хостом



4. Получим отчет с помощью команды linux_malfind

```
vagrant@kali:~$ volatility --profile=LinuxUbuntu_4_15_0-72-generic_profilex64 --filename=/vagrant/task1/memory.vmem lin
ux_malfind > /vagrant/task1-report.txt
Volatility Foundation Volatility Framework 2.6
vagrant@kali:~$ cat /vagrant/task1-report.txt | grep Pid | awk '{print $4}' | uniq
573
657
835
1034
1389
1751
```

сохранив отчет в txt вытаскиваем все уникальные pid-ы

5. Запустим linux_yarascan по найденным pid-ам

```
vagrant@kali:~$ wget https://raw.githubusercontent.com/cuckoosandbox/community/master/data/yara/shellcode/metasploit.ya
r
--2020-05-23 22:19:30--  https://raw.githubusercontent.com/cuckoosandbox/community/master/data/yara/shellcode/metasploit
.yar
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.76.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.76.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4286 (4.2K) [text/plain]
Saving to: ‘metasploit.yar’

metasploit.yar               100%[=================================================>]   4.19K  --.-KB/s    in 0s

2020-05-23 22:19:31 (44.1 MB/s) - ‘metasploit.yar’ saved [4286/4286]

vagrant@kali:~$ volatility --profile=LinuxUbuntu_4_15_0-72-generic_profilex64 --filename=/vagrant/task1/memory.vmem lin
ux_yarascan -y metasploit.yar -p 573,657,835,1034,1389,1751
Volatility Foundation Volatility Framework 2.6
```

увы яраскан ничего не нашел.