3/15/2021

Vulnerability assessment with Nessus

Contents

Task 1: Using chkrootkit to check your box 2

Task 2: Vulnerability assessment with Nessus 10

Task 3: Independent tools learning 21

REFERENCES: 29

Task 1: Using chkrootkit to check your box

Step 1(3) Linux box updating

```
┌──(kali㉿kali)-[~]
└─$ apt update
Reading package lists ... Done
E: List directory /var/lib/apt/lists/partial is missing. - Acquire (13: Permission denied)
W: Problem unlinking the file /var/cache/apt/pkgcache.bin - RemoveCaches (13: Permission denied)
W: Problem unlinking the file /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Permission denied)

┌──(kali㉿kali)-[~]
└─$ sudo apt update                                                          100 ×

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [30.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [17.7 MB]
Get:3 http://kali.download/kali kali-rolling/contrib amd64 Packages [108 kB]
Get:4 http://kali.download/kali kali-rolling/non-free amd64 Packages [204 kB]
Fetched 18.1 MB in 4s (4,108 kB/s)
Reading package lists ... Done
Building dependency tree
Reading state information ... Done
1255 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Step:1&2 Create a user and change default shell of the user.

```
Kali-Linux-2020.4-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
root@kali: /home/kali                                                  01:04 PM
root@kali: /home/kali
File  Actions  Edit  View  Help

┌──(root💀kali)-[/home/kali]
└─# useradd -m spetsnaz

┌──(root💀kali)-[/home/kali]
└─# passwd 123
passwd: user '123' does not exist

┌──(root💀kali)-[/home/kali]
└─# passwd spetsnaz                                                          1 ×
New password:
Retype new password:
passwd: password updated successfully

┌──(root💀kali)-[/home/kali]
└─# chsh -s /bin/bash spetsnaz

┌──(root💀kali)-[/home/kali]
└─#
```

Step 2(1): We can see the user added to /bin/bash shell

```
┌──(root💀 kali)-[/home/kali]
└─# chsh -s /bin/bash spetsnaz

┌──(root💀 kali)-[/home/kali]
└─# ps -p $$
    PID TTY          TIME CMD
   1358 pts/0    00:00:01 zsh

┌──(root💀 kali)-[/home/kali]
└─# grep spetsnaz /etc/passwd
spetsnaz:x:1001:1001::/home/spetsnaz:/bin/bash

┌──(root💀 kali)-[/home/kali]
└─#
```

Step 3: Adding user with group name ‘Spetsnaz’ to the Sudo group

```
┌──(root💀 kali)-[/home/kali]
└─# usermod -a -G sudo spetsnaz

┌──(root💀 kali)-[/home/kali]
└─#
```

Step 3.1: Viewing user in Sudo list

```
root@kali: /home/kali
File  Actions  Edit  View  Help

┌──(root💀 kali)-[/home/kali]
└─# awk -F"" '{ print $1 }' /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:101:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
mysql:x:104:110:MySQL Server,,,:/nonexistent:/bin/false
tss:x:105:111:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:106:65534::/var/lib/strongswan:/usr/sbin/nologin
ntp:x:107:112::/nonexistent:/usr/sbin/nologin
messagebus:x:108:113::/nonexistent:/usr/sbin/nologin
redsocks:x:109:114::/var/run/redsocks:/usr/sbin/nologin
rwhod:x:110:65534::/var/spool/rwho:/usr/sbin/nologin
iodine:x:111:65534::/run/iodine:/usr/sbin/nologin
miredo:x:112:65534::/var/run/miredo:/usr/sbin/nologin
_rpc:x:113:65534::/run/rpcbind:/usr/sbin/nologin
usbmux:x:114:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
tcpdump:x:115:120::/nonexistent:/usr/sbin/nologin
rtkit:x:116:121:RealtimeKit,,,:/proc:/usr/sbin/nologin
sshd:x:117:65534::/run/sshd:/usr/sbin/nologin
statd:x:118:65534::/var/lib/nfs:/usr/sbin/nologin
postgres:x:119:123:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
avahi:x:120:125:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
stunnel4:x:121:126::/var/run/stunnel4:/usr/sbin/nologin
Debian-snmp:x:122:127::/var/lib/snmp:/bin/false
sslh:x:123:128::/nonexistent:/usr/sbin/nologin
nm-openvpn:x:124:129:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
```

User added can be seen in the last line

```
avahi:x:120:125:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
stunnel4:x:121:126::/var/run/stunnel4:/usr/sbin/nologin
Debian-snmp:x:122:127::/var/lib/snmp:/bin/false
sslh:x:123:128::/nonexistent:/usr/sbin/nologin
nm-openvpn:x:124:129:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:125:130:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin
pulse:x:126:131:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
saned:x:127:134::/var/lib/saned:/usr/sbin/nologin
inetsim:x:128:136::/var/lib/inetsim:/usr/sbin/nologin
colord:x:129:137:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:130:138::/var/lib/geoclue:/usr/sbin/nologin
lightdm:x:131:139:Light Display Manager:/var/lib/lightdm:/bin/false
king-phisher:x:132:140::/var/lib/king-phisher:/usr/sbin/nologin
kali:x:1000:1000:Kali,,,:/home/kali:/usr/bin/zsh
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
spetsnaz:x:1001:1001::/home/spetsnaz:/bin/bash
```

Step 6: Tail command output of last 6 lines

```
spetsnaz@kali: ~
File  Actions  Edit  View  Help
┌──(spetsnaz㉿kali)-[~]
└─$ tail -n 6 /etc/passwd
spetsnaz:x:1001:1001::/home/spetsnaz:/bin/bash
navneet:x:1002:1002::/home/navneet:/bin/bash
ali:x:1003:1003::/home/ali:/bin/bash
gagan:x:1004:1004::/home/gagan:/bin/bash
gurinder:x:1005:1005::/home/gurinder:/bin/bash
tho:x:1006:1006::/home/tho:/bin/bash
┌──(spetsnaz㉿kali)-[~]
└─$
```

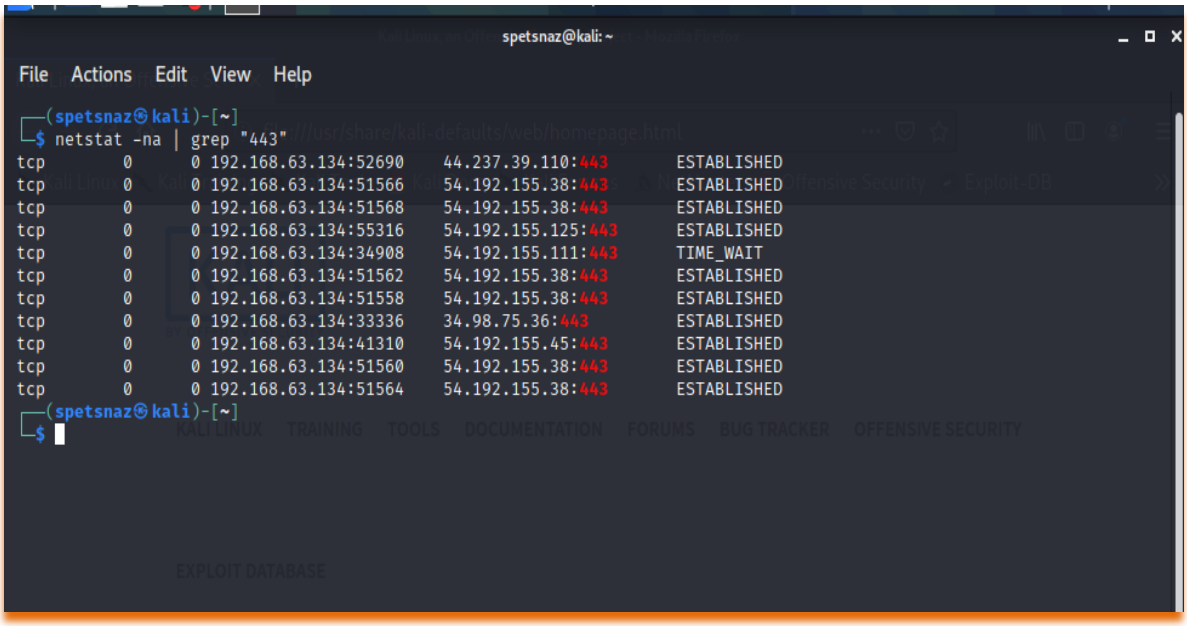
Step 7: Network interfaces of the machine

```
File  Actions  Edit  View  Help

┌──(spetsnaz㉿kali)-[~]
└─$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 00:0c:29:b8:9c:91 brd ff:ff:ff:ff:ff:ff
┌──(spetsnaz㉿kali)-[~]
└─$ netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0      1500    21279      0      0 0         3471      0      0      0 BMRU
lo       65536       20      0      0 0           20      0      0      0 LRU
┌──(spetsnaz㉿kali)-[~]
└─$ /sbin/ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.63.134  netmask 255.255.255.0  broadcast 192.168.63.255
        inet6 fe80::20c:29ff:feb8:9c91  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:b8:9c:91  txqueuelen 1000  (Ethernet)
        RX packets 21282  bytes 29752603 (28.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3473  bytes 359095 (350.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 20  bytes 956 (956.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20  bytes 956 (956.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(spetsnaz㉿kali)-[~]
└─$
```

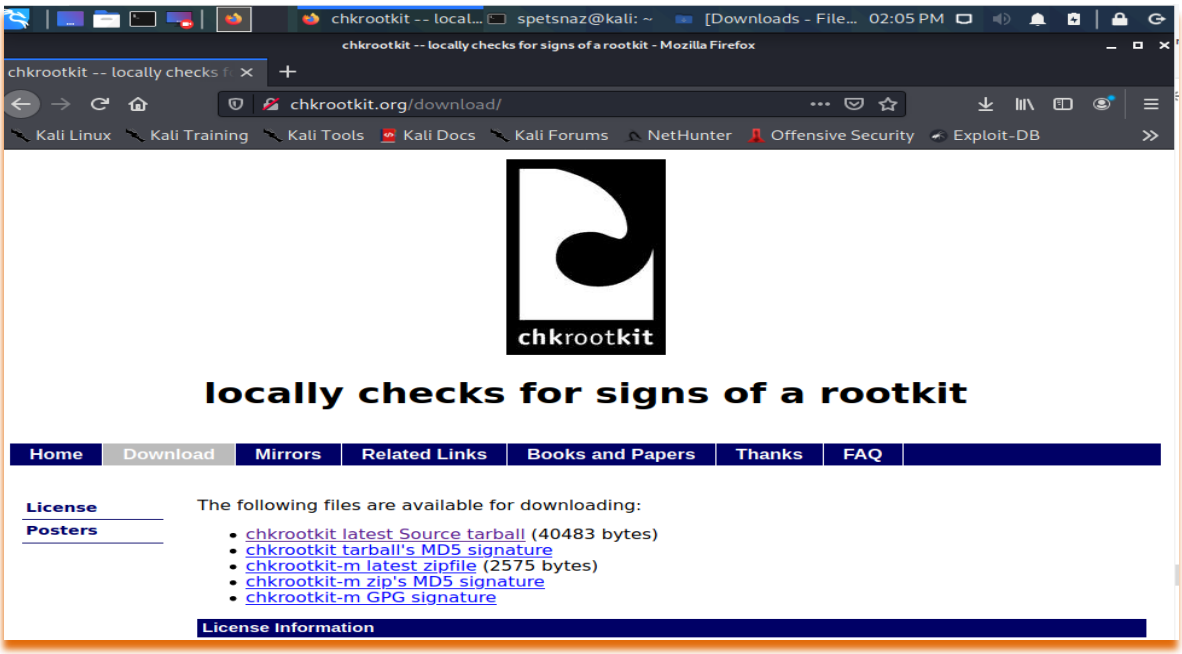
Step 8: Screenshot of the list of listening ports

```
┌──(spetsnaz㉿kali)-[~]
└─$ sudo netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
raw6       0      0 [::]:ipv6-icmp          [::]:*                  7
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     28628    @/tmp/.ICE-unix/1794
unix  2      [ ACC ]     STREAM     LISTENING     15924    /run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     27240    @/tmp/.X11-unix/X0
unix  2      [ ACC ]     STREAM     LISTENING     28507    /run/user/1001/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     21601    /run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     28512    /run/user/1001/bus
unix  2      [ ACC ]     STREAM     LISTENING     21606    /run/user/1000/bus
unix  2      [ ACC ]     STREAM     LISTENING     11329    /run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     28513    /run/user/1001/gnupg/S.dirmngr
unix  2      [ ACC ]     STREAM     LISTENING     21607    /run/user/1000/gnupg/S.dirmngr
unix  2      [ ACC ]     STREAM     LISTENING     28514    /run/user/1001/gnupg/S.gpg-agent.browser
unix  2      [ ACC ]     STREAM     LISTENING     21608    /run/user/1000/gnupg/S.gpg-agent.browser
unix  2      [ ACC ]     STREAM     LISTENING     11331    /run/systemd/userdb/io.systemd.DynamicUser
unix  2      [ ACC ]     STREAM     LISTENING     28515    /run/user/1001/gnupg/S.gpg-agent.extra
unix  2      [ ACC ]     STREAM     LISTENING     21609    /run/user/1000/gnupg/S.gpg-agent.extra
unix  2      [ ACC ]     STREAM     LISTENING     28516    /run/user/1001/gnupg/S.gpg-agent.ssh
unix  2      [ ACC ]     STREAM     LISTENING     21610    /run/user/1000/gnupg/S.gpg-agent.ssh
unix  2      [ ACC ]     STREAM     LISTENING     28517    /run/user/1001/gnupg/S.gpg-agent
unix  2      [ ACC ]     STREAM     LISTENING     21611    /run/user/1000/gnupg/S.gpg-agent
unix  2      [ ACC ]     STREAM     LISTENING     15370    /run/systemd/fsck.progress
unix  2      [ ACC ]     STREAM     LISTENING     28518    /run/user/1001/pulse/native
unix  2      [ ACC ]     STREAM     LISTENING     21612    /run/user/1000/pulse/native
unix  2      [ ACC ]     STREAM     LISTENING     15376    /run/systemd/journal/stdout
unix  2      [ ACC ]     SEQPACKET  LISTENING     15380    /run/udev/control
unix  2      [ ACC ]     STREAM     LISTENING     18428    @/tmp/dbus-r7QyRtivj7
unix  2      [ ACC ]     STREAM     LISTENING     28593    /tmp/ssh-zYQE9CQ30Jfz/agent.1794
unix  2      [ ACC ]     STREAM     LISTENING     27241    /tmp/.X11-unix/X0
unix  2      [ ACC ]     STREAM     LISTENING     28629    /tmp/.ICE-unix/1794
unix  2      [ ACC ]     STREAM     LISTENING     16881    /run/systemd/journal/io.systemd.journal
unix  2      [ ACC ]     STREAM     LISTENING     30798    @/tmp/dbus-G7wdlgi0RC
┌──(spetsnaz㉿kali)-[~]
└─$
```

Step 9: Output of netstat command after using grep for 443 port

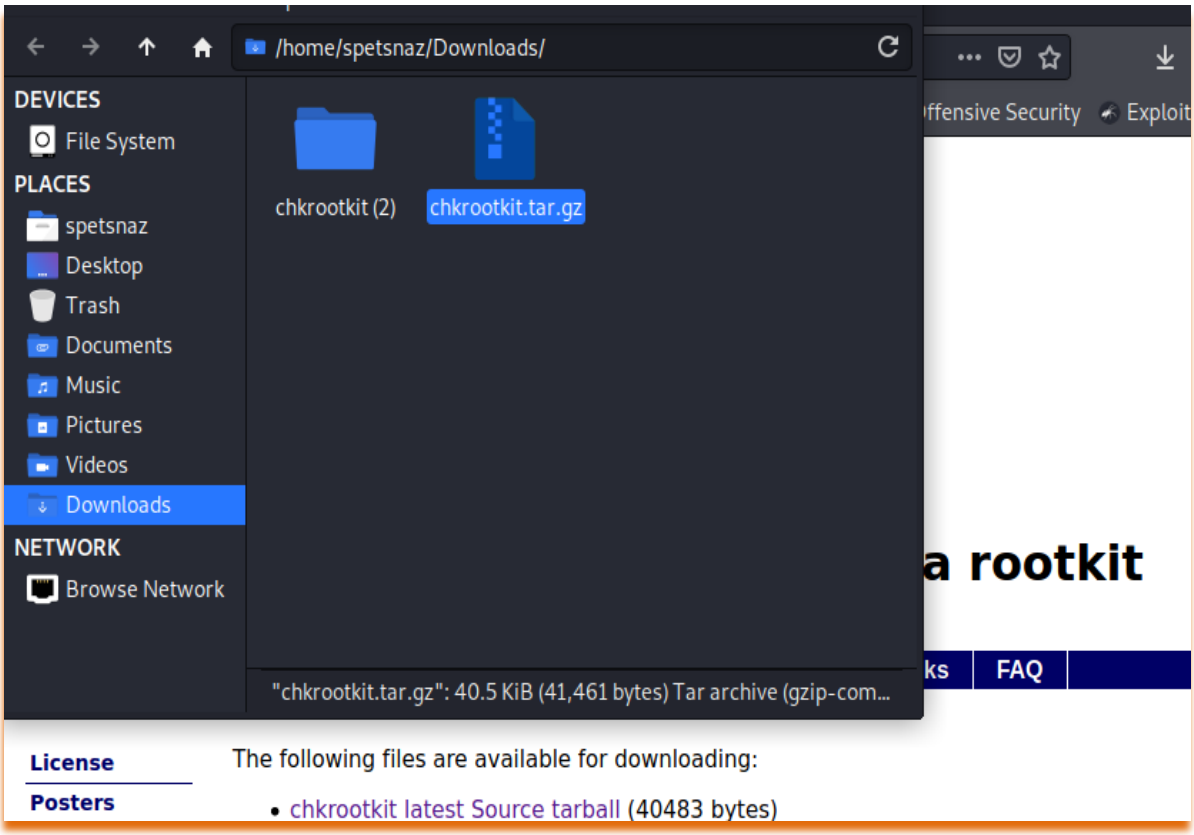


Step 10:



Step 11: Untar the chkrootkit.tar.gz file

```
┌──(spetsnaz㉿kali)-[/home]
└─$ cd /home/spetsnaz/Downloads
┌──(spetsnaz㉿kali)-[~/Downloads]
└─$ tar -xvzf chkrootkit.tar.gz
chkrootkit-0.54/ACKNOWLEDGMENTS
chkrootkit-0.54/check_wtmpx.c
chkrootkit-0.54/chk54.tgz
chkrootkit-0.54/chkdirs.c
chkrootkit-0.54/chklastlog.c
chkrootkit-0.54/chkproc.c
chkrootkit-0.54/chkrootkit
chkrootkit-0.54/chkrootkit.lsm
chkrootkit-0.54/chkutmp.c
chkrootkit-0.54/chkwtmp.c
chkrootkit-0.54/COPYRIGHT
chkrootkit-0.54/ifpromisc.c
chkrootkit-0.54/Makefile
chkrootkit-0.54/README
chkrootkit-0.54/README.chklastlog
chkrootkit-0.54/README.chkwtmp
chkrootkit-0.54/strings.c
┌──(spetsnaz㉿kali)-[~/Downloads]
└─$
```



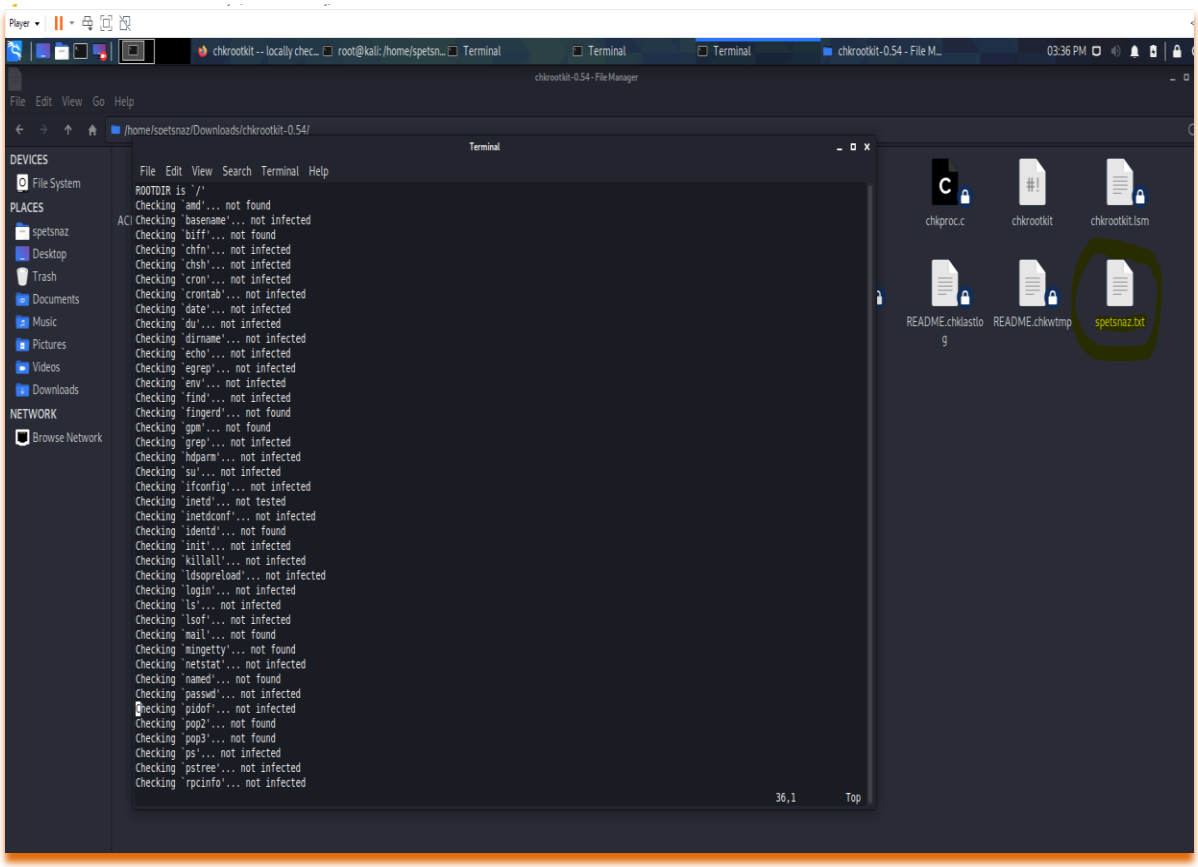
Step 12: Run the CHKROOTKIT script after making the file

```
┌──(spetsnaz㉿kali)-[~]
└─$ sudo su
[sudo] password for spetsnaz:
┌──(root💀kali)-[/home/spetsnaz]
└─# cd /home/spetsnaz/Downloads/chkrootkit-0.54/

┌──(root💀kali)-[/home/spetsnaz/Downloads/chkrootkit-0.54]
└─# make sense
cc -DHAVE_LASTLOG_H -o chklastlog chklastlog.c
cc -DHAVE_LASTLOG_H -o chkwtmp chkwtmp.c
cc -DHAVE_LASTLOG_H   -D_FILE_OFFSET_BITS=64 -o ifpromisc ifpromisc.c
cc  -o chkproc chkproc.c
cc  -o chkdirs chkdirs.c
cc  -o check_wtmpx check_wtmpx.c
cc -static  -o strings-static strings.c
cc  -o chkutmp chkutmp.c

┌──(root💀kali)-[/home/spetsnaz/Downloads/chkrootkit-0.54]
└─# ./chkrootkit
ROOTDIR is `/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not infected
Checking `chsh'... not infected
Checking `cron'... not infected
Checking `crontab'... not infected
Checking `date'... not infected
Checking `du'... not infected
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not found
Checking `gpm'... not found
Checking `grep'... not infected
Checking `hdparm'... not infected
Checking `su'... not infected
Checking `ifconfig'... not infected
Checking `inetd'... not tested
Checking `inetdconf'... not infected
Checking `identd'... not found
Checking `init'... not infected
Checking `killall'... not infected
Checking `ldsopreload'... not infected
Checking `login'... not infected
Checking `ls'... not infected
Checking `lsof'... not infected
Checking `mail'... not found
Checking `mingetty'... not found
Checking `netstat'... not infected
```

Step 13: Sending output to a text file

```
! root         5737 pts/0  ./chkutmp
! root         5738 pts/0  sh -c ps ax -o "tty,pid,ruser,args"
! root         5739 pts/0  ps ax -o tty,pid,ruser,args
chkutmp: nothing deleted
Checking `OSX_RSPLUG'... not tested

┌──(root💀kali)-[/home/spetsnaz/Downloads/chkrootkit-0.54]
└─# ./chkrootkit > spetsnaz.txt

┌──(root💀kali)-[/home/spetsnaz/Downloads/chkrootkit-0.54]
└─#
```

Step 13(2): Text file data

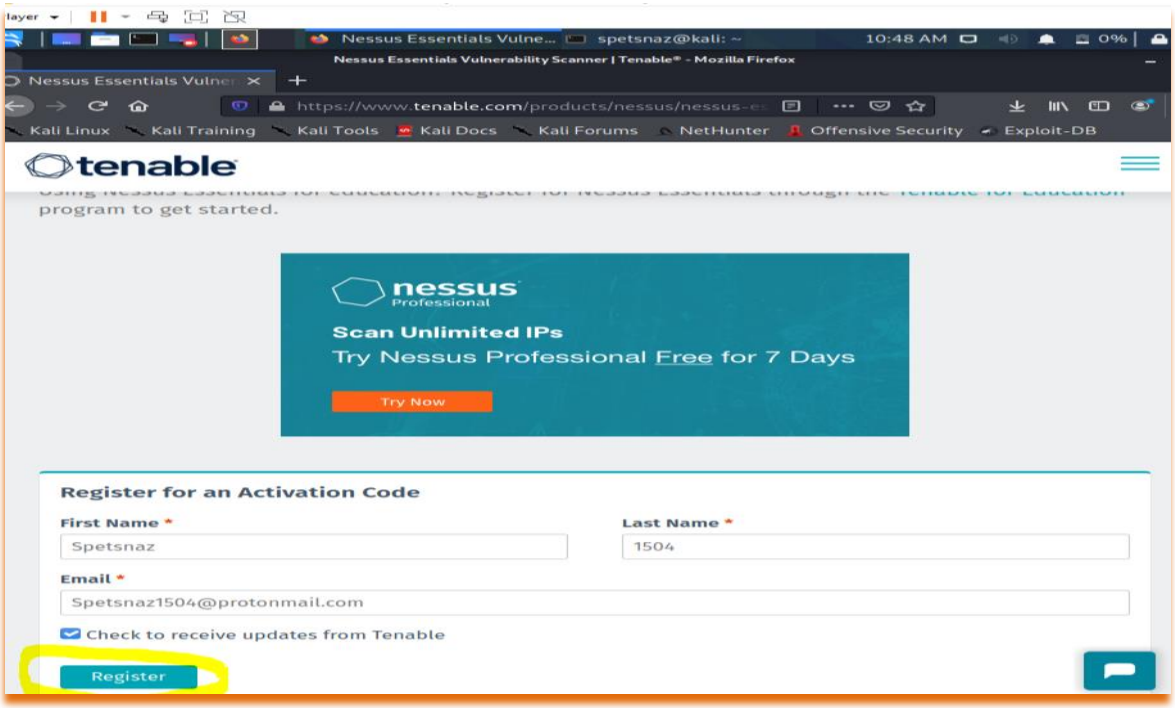

Terminal
File Edit View Search Terminal Help
ROOTDIR is `/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not infected
Checking `chsh'... not infected
Checking `cron'... not infected
Checking `crontab'... not infected
Checking `date'... not infected
Checking `du'... not infected
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not found
Checking `gpm'... not found
Checking `grep'... not infected
Checking `hdparm'... not infected
Checking `su'... not infected
Checking `ifconfig'... not infected
Checking `inetd'... not tested
Checking `inetdconf'... not infected
Checking `identd'... not found
Checking `init'... not infected
Checking `killall'... not infected
Checking `ldsopreload'... not infected
Checking `login'... not infected
Checking `ls'... not infected
Checking `lsof'... not infected
Checking `mail'... not found
Checking `mingetty'... not found
Checking `netstat'... not infected
Checking `named'... not found
Checking `passwd'... not infected
Checking `pidof'... not infected
Checking `pop2'... not found
Checking `pop3'... not found
Checking `ps'... not infected
Checking `pstree'... not infected
Checking `rpcinfo'... not infected
36,1 Top
chkproc.c
chkrootkit
chkrootkit.lsm
README.chklastlog
README.chkwtmp
spetsnaz.txt

Task 2: Vulnerability assessment with Nessus

Updating the Linux box

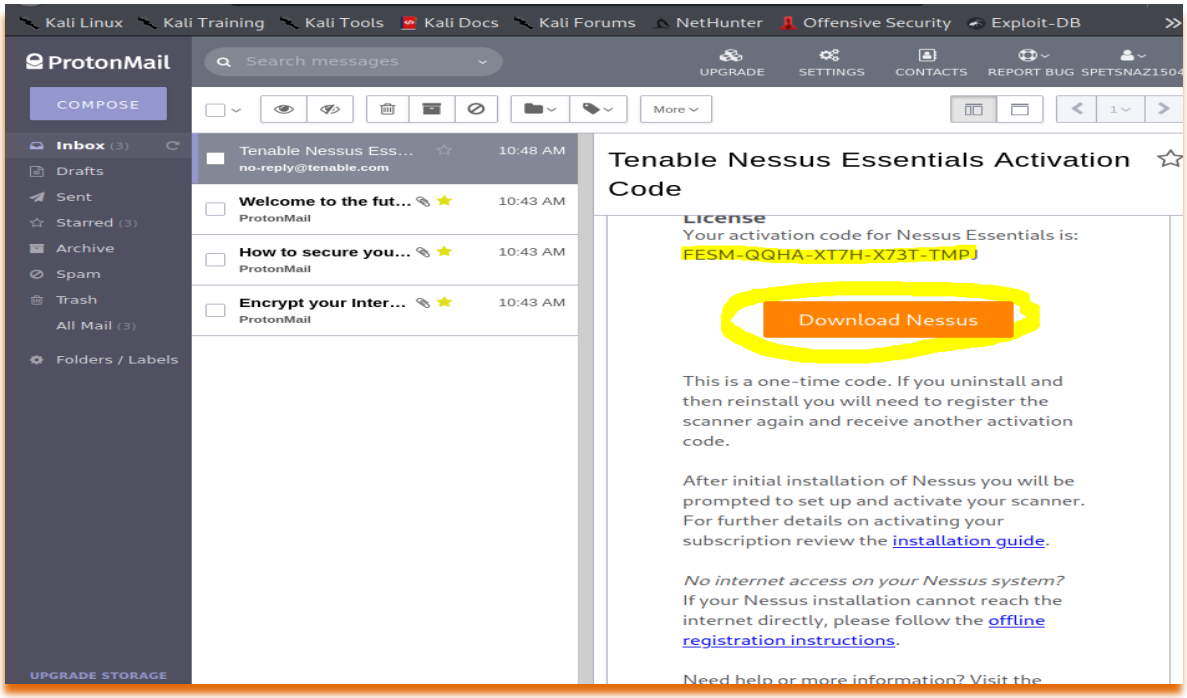
```
└─$ sudo apt update
[sudo] password for spetsnaz:
Get:1 http://ftp.harukasan.org/kali kali-rolling InRelease [30.5 kB]
Get:2 http://ftp.harukasan.org/kali kali-rolling/main amd64 Packages [17.7 MB]
Get:3 http://ftp.harukasan.org/kali kali-rolling/contrib amd64 Packages [108 kB]
Get:4 http://ftp.harukasan.org/kali kali-rolling/non-free amd64 Packages [199 kB]
Fetched 18.1 MB in 1min 7s (270 kB/s)
Reading package lists ... Done
Building dependency tree
Reading state information ... Done
1304 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Step 14:



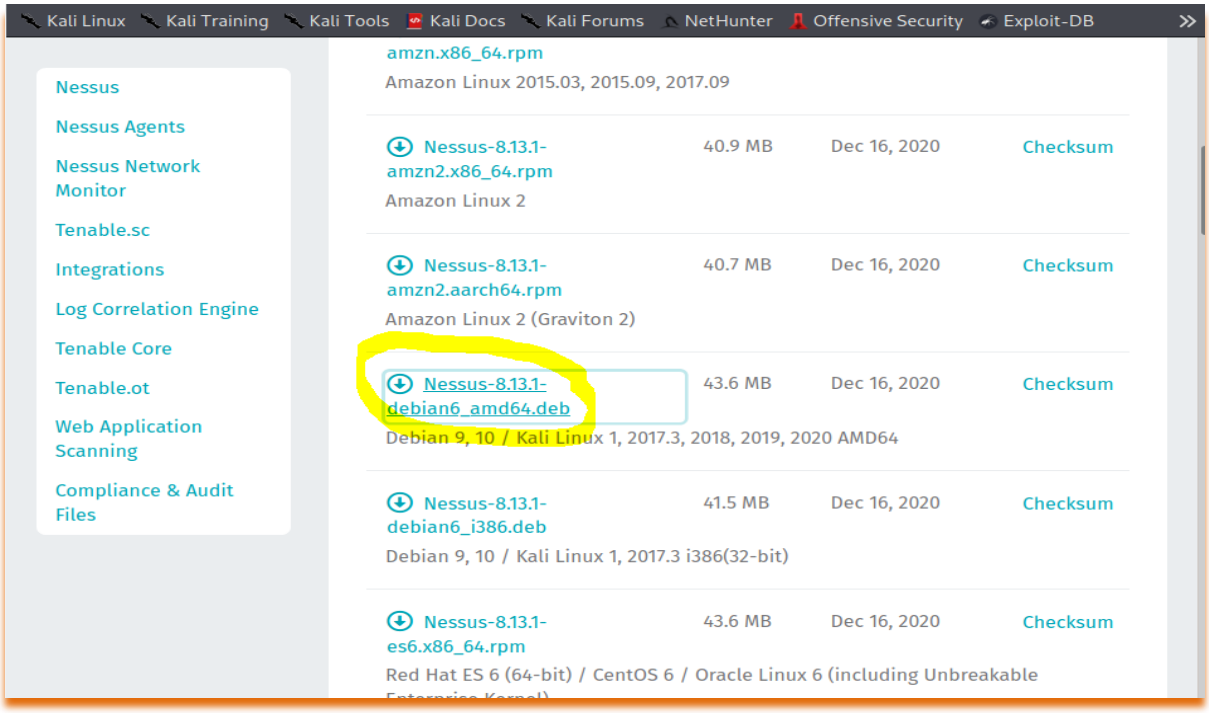
+-

Step 14(1):

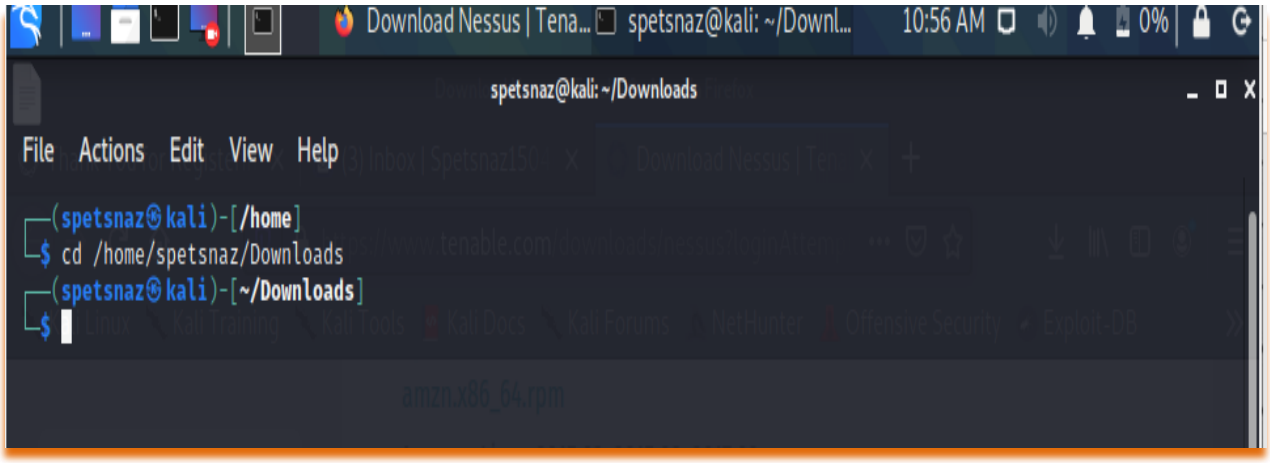


Step 15: checking OS version

Step 15(2):



Step 16: Screenshot after navigating to downloads folder.



Step 17: Installing Nessus in the system

```
┌──(spetsnaz㉿kali)-[~/Downloads]
└─$ sudo dpkg -i Nessus-8.13.1-debian6_amd64.deb
Selecting previously unselected package nessus.
(Reading database ... 261779 files and directories currently installed.)
Preparing to unpack Nessus-8.13.1-debian6_amd64.deb ...
Unpacking nessus (8.13.1) ...
Setting up nessus (8.13.1) ...
Unpacking Nessus Scanner Core Components ...

 - You can start Nessus Scanner by typing /bin/systemctl start nessusd.service
 - Then go to https://kali:8834/ to configure your scanner

┌──(spetsnaz㉿kali)-[~/Downloads]
└─$
```

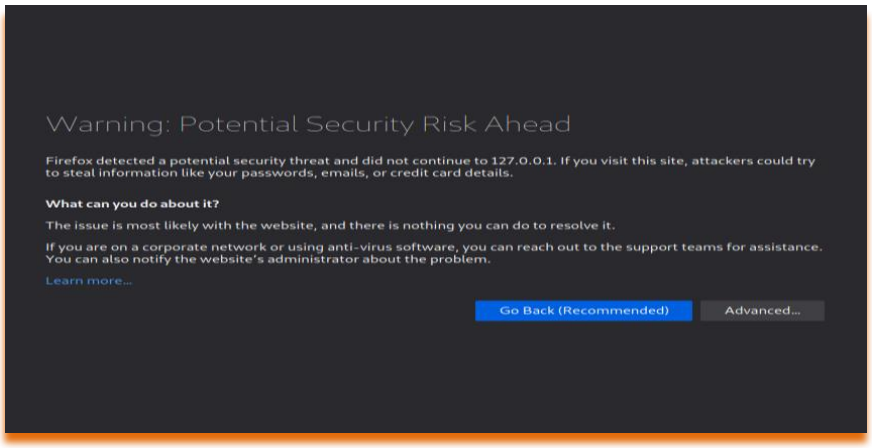
Step 18: Status of Nessus Package

```
┌──(spetsnaz㉿kali)-[~]
└─$ systemctl status nessusd
● nessusd.service - The Nessus Vulnerability Scanner
     Loaded: loaded (/lib/systemd/system/nessusd.service; disabled; vendor preset: disabled)
     Active: inactive (dead)
┌──(spetsnaz㉿kali)-[~]
└─$
```

Step 19: Starting Nessus Daemon and Status of Nessus after starting

```
┌──(spetsnaz㉿kali)-[~]
└─$ systemctl enable --now nessusd
Created symlink /etc/systemd/system/nessusd.service → /lib/systemd/system/nessusd.service.
Created symlink /etc/systemd/system/multi-user.target.wants/nessusd.service → /lib/systemd/system/nessusd.service.
┌──(spetsnaz㉿kali)-[~]
└─$ systemctl status nessusd
● nessusd.service - The Nessus Vulnerability Scanner
     Loaded: loaded (/lib/systemd/system/nessusd.service; enabled; vendor preset: disabled)
     Active: active (running) since Thu 2021-03-04 11:21:06 EST; 13s ago
   Main PID: 9077 (nessus-service)
      Tasks: 12 (limit: 2300)
     Memory: 98.5M
     CGroup: /system.slice/nessusd.service
             ├─9077 /opt/nessus/sbin/nessus-service -q
             └─9078 nessusd -q
┌──(spetsnaz㉿kali)-[~]
└─$
```

Step 20: Login to Nessus web interface



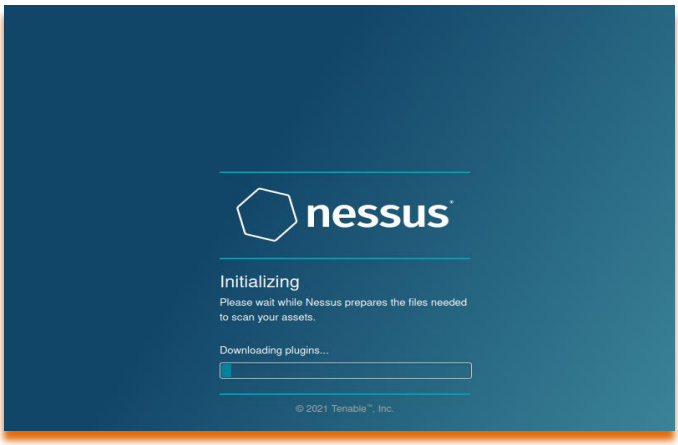
Step 21:

Step 21(2):

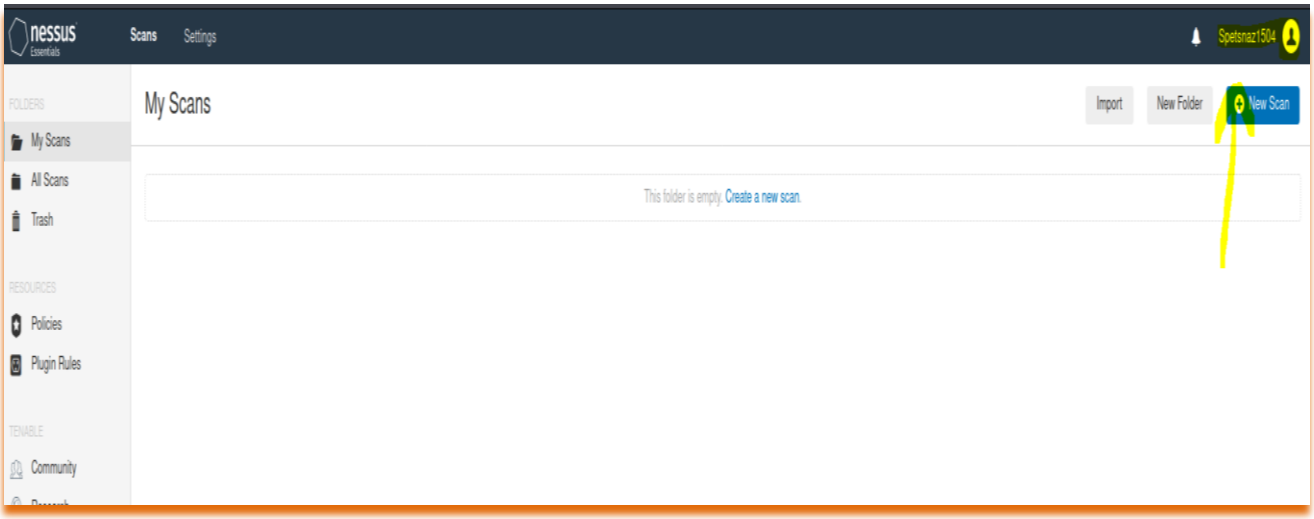
Step 21(3):


nessus
Essentials
Create a user account
Create a Nessus administrator user account. Use this username and password to log in to Nessus.
Username *
Spetsnaz1504
Password *
Back
Submit
© 2021 Tenable™, Inc.

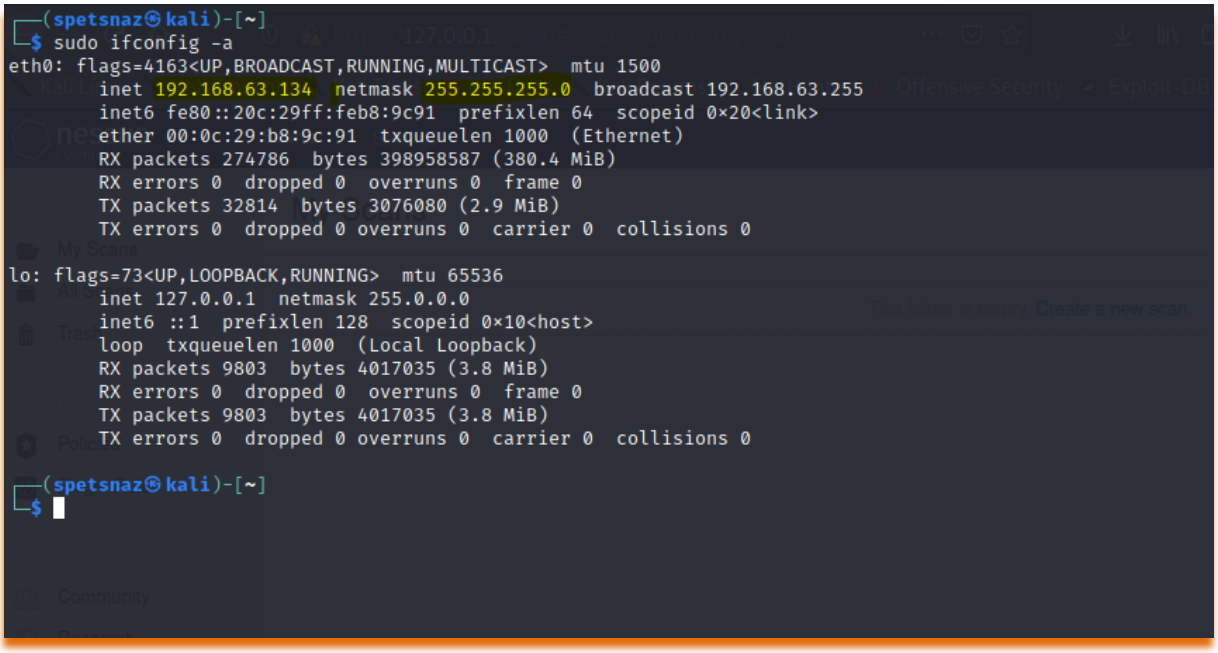
Step 21(4):


nessus
Initializing
Please wait while Nessus prepares the files needed to scan your assets.
Downloading plugins...
© 2021 Tenable™, Inc.

Step 22: Login to Nessus web portal with username “Spetsnaz1504” and password

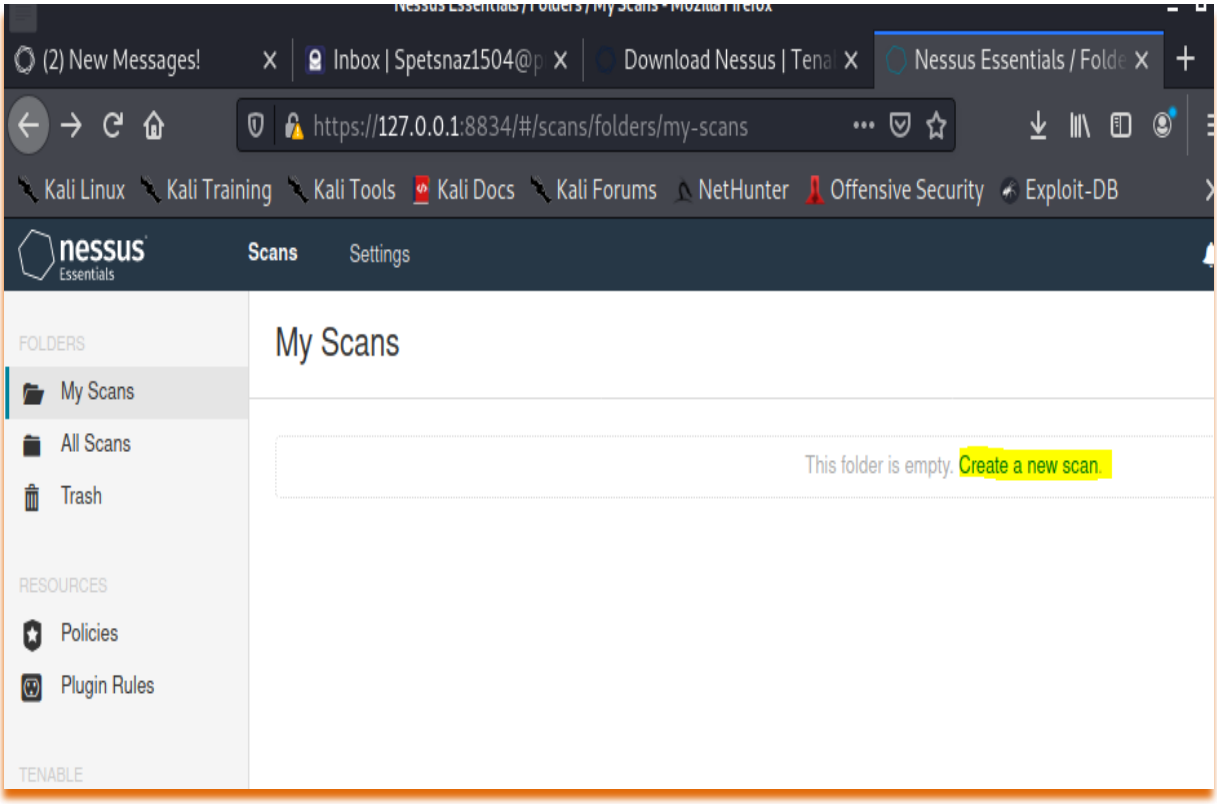


Step 23: Noting the IP address using “sudo ifconfig -a” command

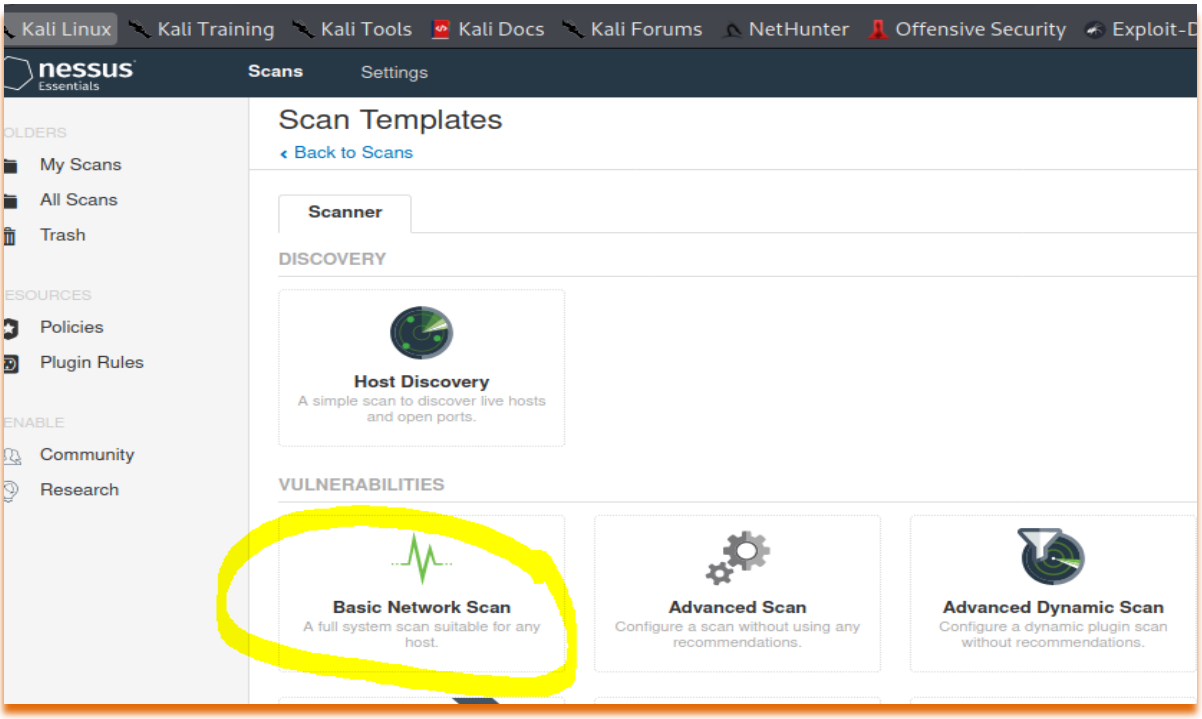


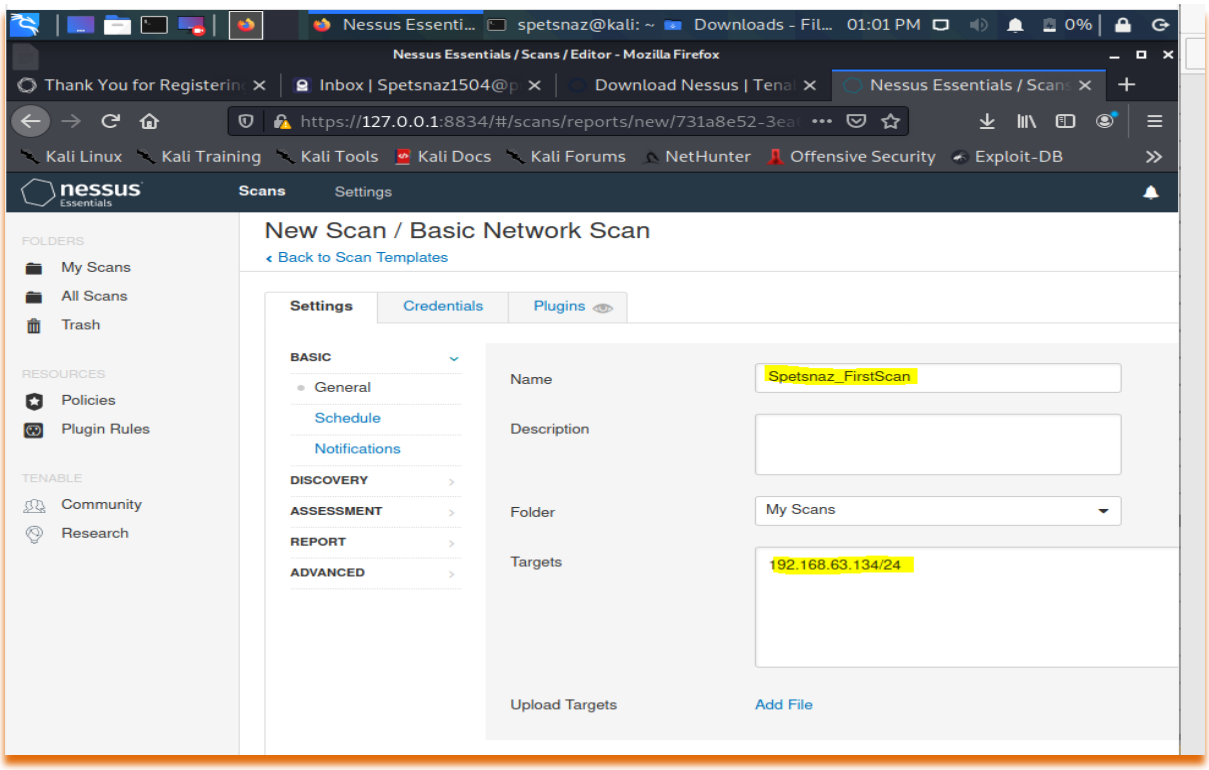
```
┌──(spetsnaz㉿kali)-[~]
└─$ sudo ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.63.134  netmask 255.255.255.0  broadcast 192.168.63.255
        inet6 fe80::20c:29ff:feb8:9c91  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:b8:9c:91  txqueuelen 1000  (Ethernet)
        RX packets 274786  bytes 398958587 (380.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 32814  bytes 3076080 (2.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 9803  bytes 4017035 (3.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9803  bytes 4017035 (3.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(spetsnaz㉿kali)-[~]
└─$
```

Step 24: Starting the New Scan

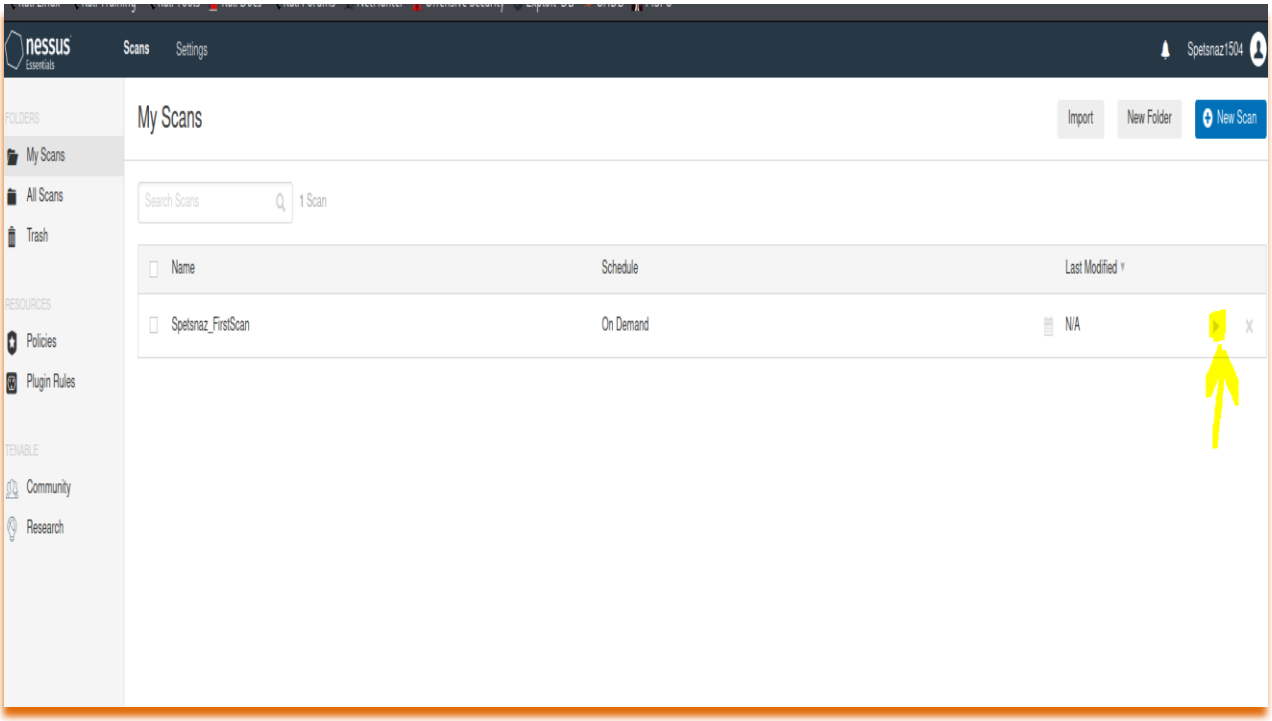

(2) New Messages!
Inbox | Spetsnaz1504@p
Download Nessus | Tena
Nessus Essentials / Folde
https://127.0.0.1:8834/#/scans/folders/my-scans
Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-DB
nessus
Essentials
Scans
Settings
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
TENABLE
My Scans
This folder is empty. Create a new scan.

Step 24(2):

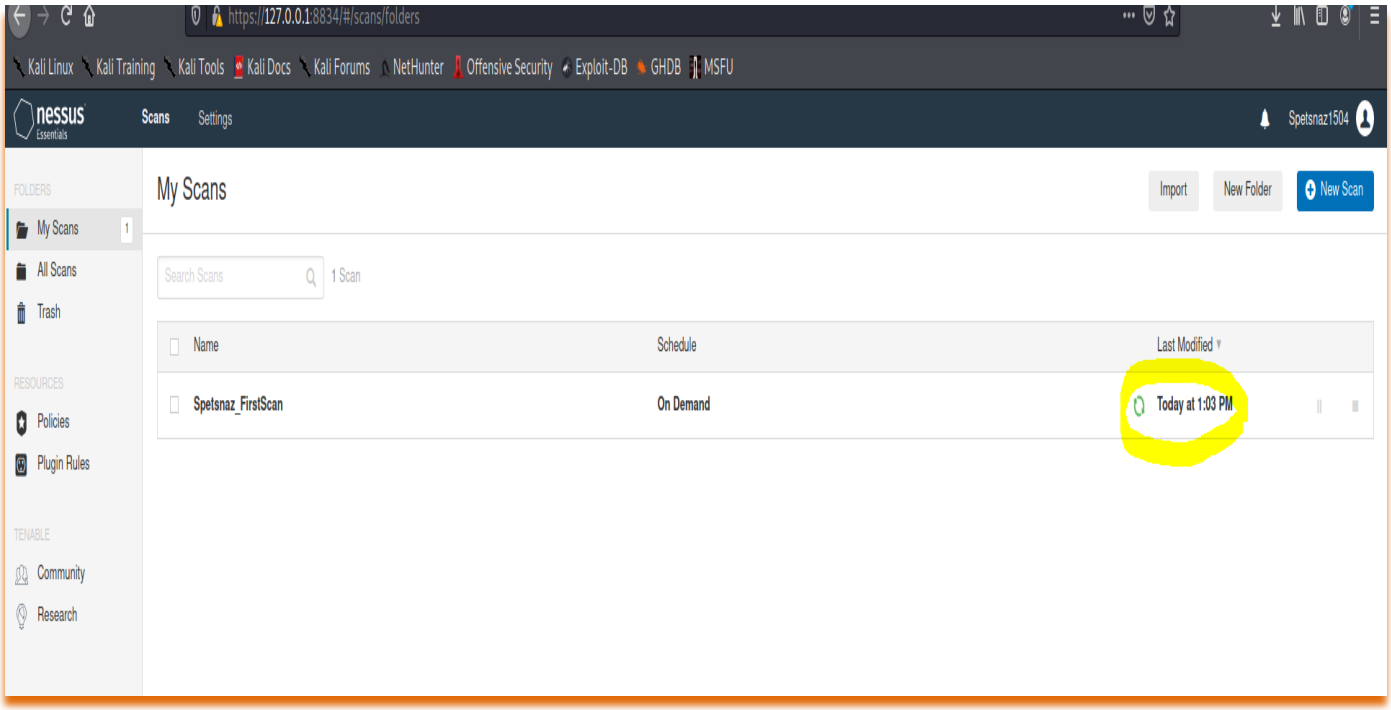

Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-D
nessus
Essentials
Scans
Settings
My Scans
All Scans
Trash
Policies
Plugin Rules
Community
Research
Scan Templates
Back to Scans
Scanner
DISCOVERY
Host Discovery
A simple scan to discover live hosts and open ports.
VULNERABILITIES
Basic Network Scan
A full system scan suitable for any host.
Advanced Scan
Configure a scan without using any recommendations.
Advanced Dynamic Scan
Configure a dynamic plugin scan without recommendations.



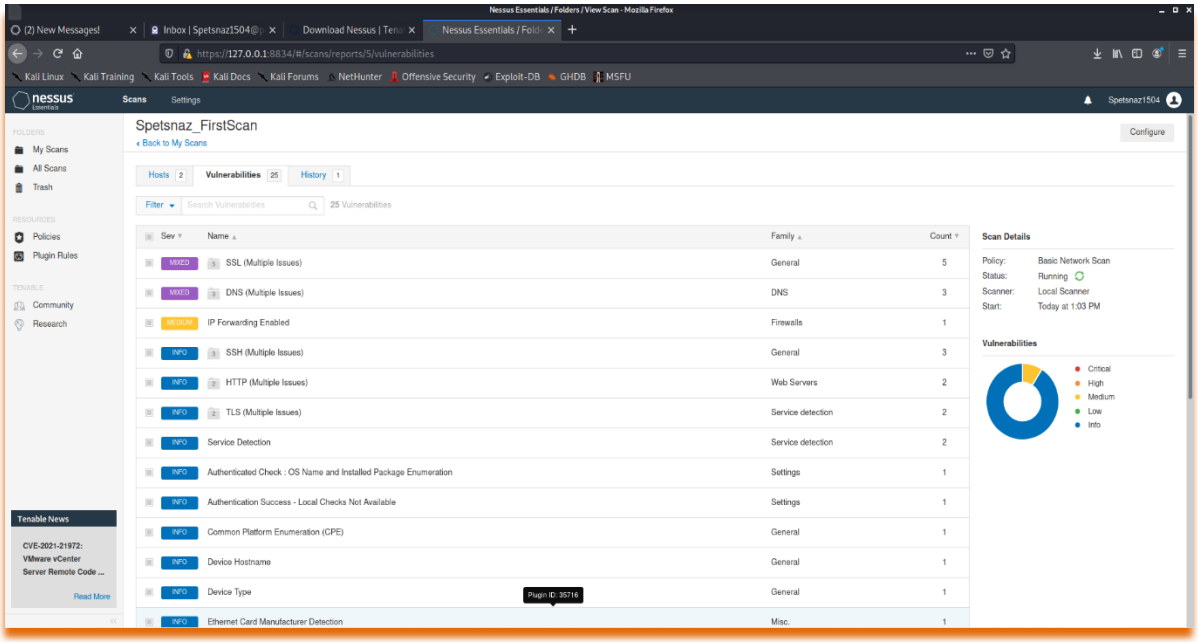
Step 27: Screenshot of clicking the play button to start the scan.



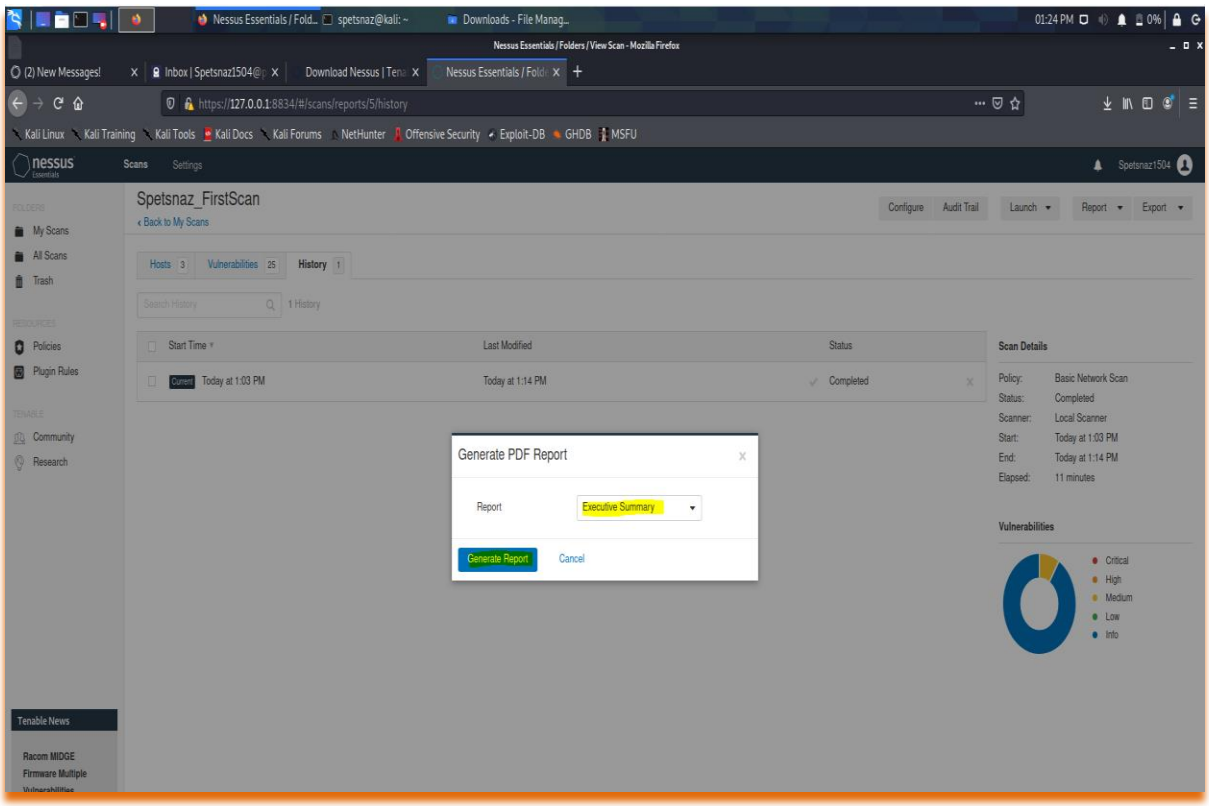
Scan still running…



Step 28: Screenshot of vulnerabilities tab.



Step 30 and 31: Generating PDF report.



Below is the PDF Scan Report document.

Spetsnaz_FirstScan_
m29imp.pdf

Task 3: Independent tools learning - Volatility

Answer the following questions in presentation:

a. What is the name of the tool and its nature?

The tool we are working on now is Volatility. This can be termed as a non-profit organization, which generally is known as open-source.
Volatility is developed by code written over Python. It can be used or run over Linux, Microsoft Windows, and Mac OS X.
We all can agree on the fact that cybercrime has rapidly increased with the rise in the usage of the Internet. It is an important gateway for providing the information one needs. Cybercrime, in general, can be considered as criminal activity involving a computer system, its associated systems or its implementations.
The one who deals with these crimes and solves them can be considered as a digital forensics' investigator. For a digital forensics' investigator, every cybercrime today poses new challenges.

The method of discovering and analyzing electronic data is known as digital forensics. An investigation aims to protect the evidence gathered during the investigation. This type of evidence is known as digital evidence, and it must be preserved to recreate past events. The study of volatile memory is extremely important in the automated investigation process.
Passwords, event logs, cryptographic keys, process information, and other important details related to the number of processes operating in a system may be included in the data. A traditional technique known as Live Response can be used to gather volatile data from a victimized computer device under investigation.

In this method, the investigator first creates a trustworthy command shell to obtain data for the investigation. The use of a Live Response method for volatile memory analysis aids in the collection of all relevant evidence from a device.

These proofs may be used to show that any event that could have compromised a system and resulted in a cybercrime occurred. Memory image analysis is another way of analyzing volatile memory. The study of volatile memory is done by taking a memory dump, which is a snapshot of the RAM.

b. What does the tool mainly used for?

Volatility is primarily used in the forensics community for uncertainty and memory analysis, to secure the project's intellectual property (trademarks, permits, etc.) and long-term viability, and to help advance creative memory analysis study.
The focus of previous digital investigations has been on locating prohibited items inside hard drive files. Volatility introduced people to the power of using volatile storage data to analyse a system's runtime state (RAM). It also provided a cross-platform, scalable, and configurable platform to facilitate additional research in this exciting field.

The Volatility tool can be used for different forensics techniques. These techniques can be both open-sourced or private.
Based on the type of analysis these techniques/tools are divided into many categories.

The Volatility tool is mainly used by investigators as it encourages transparent knowledge exchange creatively. Since Volatility has been this collaborative, memory processing has grown to be the most important topic and is considered to be very sensitive for the growth of digital investigations.

The above mentioned is also the reason for which this tool is the most used platform by the investigators.

The majorly supported memory formats are as follows:

• 32- and 64-bit Windows Crash Dump
• 32- and 64-bit Windows Hibernation
• 32- and 64-bit MachO files
•
Virtualbox Core Dumps

• VMware Saved State (.vmss) and Snapshot (.vmsn)
• HPAK Format (FastDump)
• LiME (Linux Memory Extractor)
• QEMU VM memory dumps
•
Raw/Padded Physical Memory
• Firewire (IEEE 1394)
• Expert Witness (EWF)

Volatility also offers a one-of-a-kind platform that allows cutting-edge analysis to be quickly transferred to digital investigators. As a result, analysis based on Volatility has been presented at some of the most prestigious academic gatherings, and Volatility has been used in some of the most important studies of the last decade.
It's become a must-have digital investigative tool for law enforcement, military, academic, and industrial investigators all over the world.

c. Are there any dependencies of the tool? Any Java/Python library needed etc?

The Libraries for windows platform are already included in the exe but For Linux we need to install libraries and certain packages as prerequisites for the packages because if not you may see a warning message popup to raise your awareness.
So, for comprehensive plugin support we need to install certain libraries:

1. Distrom3- lightweight, easy to use, fast decomposer library and dissembles instructions in 16-, 32- and 64-bit modes. Use linux_volshell dissemble command.
2. Yara- Help identify and classify malware. To use un the command: `echo "/usr/local/lib" >> /etc/ld.so.conf && ldconfig`
3. Pytz Library: It allows accurate and cross platform time zone calculations. Command used to install: python **setup.py install**
4. Ipython Library: Provides a rich toolkit and a powerful interactive python shell. Provides Comprehensive object introspection, session logging and reloading. Command used to install: **pip install ipython**
5. PyCrypto: collection of secure hash functions and various encryption algorithms. Helps in writing secure administration tools as well as writing daemons and servers. Also, we can encrypt private data for added security. Command to install is **pip install pycryptodome.**

d. List 5 main functionalities of the tool and explain them.

Volatility can be used with varied functions:

- **First: used to get the list of running processes in memory**

  The command: $ python vol.py -f [image] --profile=[profile] **pstree**

```
Shell No.1
File   Actions   Edit   View   Help
root@Spetsnaz:~/Desktop/volatility# ./vol.py -f ../cridex.vmem --profile=WinXPSP2x86 pstr
ee
Volatility Foundation Volatility Framework 2.6.1
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x823c89c8:System                                      4      0     53    240 1970-01-01
 00:00:00 UTC+0000
. 0x822f1020:smss.exe                                 368      4      3     19 2012-07-22
 02:42:31 UTC+0000
.. 0x82298700:winlogon.exe                            608    368     23    519 2012-07-22
 02:42:32 UTC+0000
... 0x81e2ab28:services.exe                           652    608     16    243 2012-07-22
 02:42:32 UTC+0000
.... 0x821dfda0:svchost.exe                          1056    652      5     60 2012-07-22
 02:42:33 UTC+0000
.... 0x81eb17b8:spoolsv.exe                          1512    652     14    113 2012-07-22
 02:42:36 UTC+0000
.... 0x81e29ab8:svchost.exe                           908    652      9    226 2012-07-22
 02:42:33 UTC+0000
.... 0x823001d0:svchost.exe                          1004    652     64   1118 2012-07-22
 02:42:33 UTC+0000
..... 0x8205bda0:wuauclt.exe                         1588   1004      5    132 2012-07-22
 02:44:01 UTC+0000
..... 0x821fcda0:wuauclt.exe                         1136   1004      8    173 2012-07-22
 02:43:46 UTC+0000
.... 0x82311360:svchost.exe                           824    652     20    194 2012-07-22
 02:42:33 UTC+0000
.... 0x820e8da0:alg.exe                               788    652      7    104 2012-07-22
 02:43:01 UTC+0000
.... 0x82295650:svchost.exe                          1220    652     15    197 2012-07-22
 02:42:35 UTC+0000
... 0x81e2a3b8:lsass.exe                              664    608     24    330 2012-07-22
 02:42:32 UTC+0000
.. 0x822a0598:csrss.exe                               584    368      9    326 2012-07-22
 02:42:32 UTC+0000
 0x821dea70:explorer.exe                             1484   1464     17    415 2012-07-22
 02:42:36 UTC+0000
. 0x81e7bda0:reader_sl.exe                           1640   1484      5     39 2012-07-22
 02:42:36 UTC+0000
root@Spetsnaz:~/Desktop/volatility# ./vol.py -f ../cridex.vmem --profile=WinXPSP2x86 dlli
st -p 1484
Volatility Foundation Volatility Framework 2.6.1
Usage: Volatility - A memory forensics analysis platform.
```

  This statement helps us to extract the list of running applications in a tree form at the time we dumped memory. The list will contain important information such as application names, their pid and when they were active.

- **Second: It can also be used for malware analysis, using the Psscan plugin**

The command: $ python vol.py -f [image] --profile=[profile] **psscan**

```
root@Spetsnaz:~/Desktop/volatility# ./vol.py -f ../cridex.vmem --profile=WinXPSP2x86 psscan
Volatility Foundation Volatility Framework 2.6.1
Offset(P)          Name                PID   PPID PDB        Time created                   Time exi
ted
------------------ ---------------- ------ ------ ---------- ------------------------------ --------
----------------------
0x0000000002029ab8 svchost.exe         908    652 0x079400e0 2012-07-22 02:42:33 UTC+0000

0x000000000202a3b8 lsass.exe           664    608 0x079400a0 2012-07-22 02:42:32 UTC+0000

0x000000000202ab28 services.exe        652    608 0x07940080 2012-07-22 02:42:32 UTC+0000

0x000000000207bda0 reader_sl.exe      1640   1484 0x079401e0 2012-07-22 02:42:36 UTC+0000

0x00000000020b17b8 spoolsv.exe        1512    652 0x079401c0 2012-07-22 02:42:36 UTC+0000

0x000000000225bda0 wuauclt.exe        1588   1004 0x07940200 2012-07-22 02:44:01 UTC+0000

0x00000000022e8da0 alg.exe             788    652 0x07940140 2012-07-22 02:43:01 UTC+0000

0x00000000023dea70 explorer.exe       1484   1464 0x079401a0 2012-07-22 02:42:36 UTC+0000

0x00000000023dfda0 svchost.exe        1056    652 0x07940120 2012-07-22 02:42:33 UTC+0000

0x00000000023fcda0 wuauclt.exe        1136   1004 0x07940180 2012-07-22 02:43:46 UTC+0000

0x0000000002495650 svchost.exe        1220    652 0x07940160 2012-07-22 02:42:35 UTC+0000

0x0000000002498700 winlogon.exe        608    368 0x07940060 2012-07-22 02:42:32 UTC+0000

0x00000000024a0598 csrss.exe           584    368 0x07940040 2012-07-22 02:42:32 UTC+0000

0x00000000024f1020 smss.exe            368      4 0x07940020 2012-07-22 02:42:31 UTC+0000

0x00000000025001d0 svchost.exe        1004    652 0x07940100 2012-07-22 02:42:33 UTC+0000

0x0000000002511360 svchost.exe         824    652 0x079400c0 2012-07-22 02:42:33 UTC+0000

0x00000000025c89c8 System                4      0 0x002fe000
```

It scans for inactive, hidden processes that a pstree cannot fully cover. Like Pstree, the list will contain information such as their name, their pid, and the time they were active.

- **Third: It can also be used to view commands used in cmd.exe**

  The command: $ python vol.py -f [image] --profile=[profile] **cmdscan**

```
Shell No.1
File  Actions  Edit  View  Help
root@Spetsnaz:~/Desktop/volatility# ./vol.py -f ../mem.bin --profile=Win2003SP0x86 cmdscan
Volatility Foundation Volatility Framework 2.6.1
**************************************************
CommandProcess: csrss.exe Pid: 420
CommandHistory: 0x4e5098 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 6 LastAdded: 5 LastDisplayed: 5
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x714
Cmd #0 @ 0x4e2e90: c:\
Cmd #1 @ 0x4e2e80: cd\
Cmd #2 @ 0x1203770: cd ITShare
Cmd #3 @ 0x1203f10: ftp 172.16.150.8
Cmd #4 @ 0x11f7ae0: dir
Cmd #5 @ 0x1203e00: mdd.exe -o dc-memdump.bin
Cmd #6 @ 0x1203418: ??
Cmd #7 @ 0x1203ac8: ??
Cmd #8 @ 0x1203b90: N32tm /config /manualpeerlist:"ntp0.mcs.anl.gov" /reliable:yes /update
Cmd #9 @ 0x1203ce0: ?32tm /stripchart /computer: ntp0.mcs.anl.gov /samples: 2 /dataonly???32tm /stripchart /computer:ntp0.mcs.anl.gov /samples:2 /dataonly
Cmd #10 @ 0x1203d70: ?32tm /stripchart /computer:ntp0.mcs.anl.gov /samples:2 /dataonly
Cmd #11 @ 0x1203e00: mdd.exe -o dc-memdump.bin
Cmd #12 @ 0x1203e90: ?et stop w32time
Cmd #13 @ 0x1203f10: ftp 172.16.150.8
Cmd #14 @ 0x1203f40: hash
Cmd #16 @ 0x1203f58: ?32tm /config /manualpeerlist:"0.us.pool.ntp.org 1.us.pool.ntp.org" /syncfromflags:MANUAL
Cmd #17 @ 0x1203b00: dfir
Cmd #18 @ 0x1204018: ??N?
**************************************************
CommandProcess: csrss.exe Pid: 420
CommandHistory: 0x1203678 Application: mdd.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x370
root@Spetsnaz:~/Desktop/volatility#
```

This plugin can help us find the commands that the attacker entered through cmd.exe

- **Fourth: The tool also helps us identify the correct profile of the memory dump**

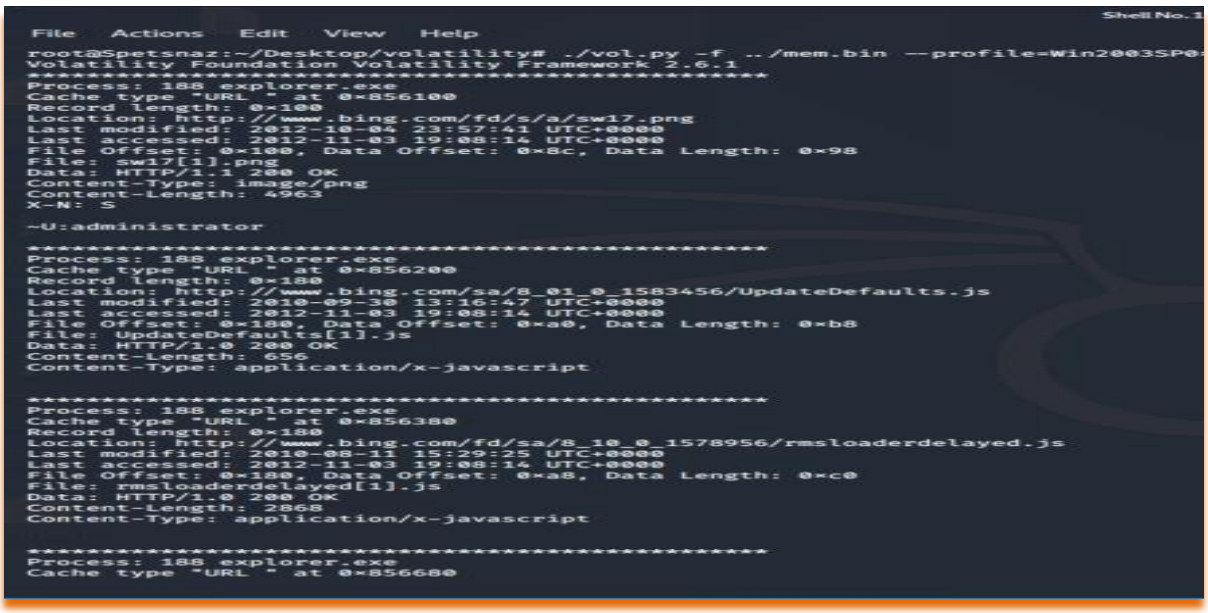
The command: $ python vol.py -f [image] --profile=[profile] **kdbgscan**

```
Shell No.1
File  Actions  Edit  View  Help
root@Spetsnaz:~/Desktop/volatility# ./vol.py -f ../cridex.vmem --profile=WinXPSP2x86 kdbgscan
Volatility Foundation Volatility Framework 2.6.1
**************************************************
Instantiating KDBG using: Kernel AS WinXPSP2x86 (5.1.0 32bit)
Offset (V)                    : 0x80545ae0
Offset (P)                    : 0x545ae0
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): WinXPSP3x86
Version64                     : 0x80545ab8 (Major: 15, Minor: 2600)
Service Pack (CmNtCSDVersion) : 3
Build string (NtBuildLab)     : 2600.xpsp.080413-2111
PsActiveProcessHead           : 0x8055a158 (17 processes)
PsLoadedModuleList            : 0x80553fc0 (109 modules)
KernelBase                    : 0x804d7000 (Matches MZ: True)
Major (OptionalHeader)        : 5
Minor (OptionalHeader)        : 1
KPCR                          : 0xffdff000 (CPU 0)

**************************************************
Instantiating KDBG using: Kernel AS WinXPSP2x86 (5.1.0 32bit)
Offset (V)                    : 0x80545ae0
Offset (P)                    : 0x545ae0
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): WinXPSP2x86
Version64                     : 0x80545ab8 (Major: 15, Minor: 2600)
Service Pack (CmNtCSDVersion) : 3
Build string (NtBuildLab)     : 2600.xpsp.080413-2111
PsActiveProcessHead           : 0x8055a158 (17 processes)
PsLoadedModuleList            : 0x80553fc0 (109 modules)
KernelBase                    : 0x804d7000 (Matches MZ: True)
Major (OptionalHeader)        : 5
Minor (OptionalHeader)        : 1
KPCR                          : 0xffdff000 (CPU 0)
```

Wanting to use this software requires us to use the imageinfo plugin, but this plugin cannot determine the exact memory profile we dumped so I suggest using kdbgscan, which is designed to positively define correct records and correct KDBG addresses (if there are many).

- **Fifth: It can help restores segments of IE browser's index.dat cache file**

The command: $ python vol.py -f [image] --profile=[profile] **iehistory**



```
File   Actions   Edit   View   Help
root@Spetsnaz:~/Desktop/volatility# ./vol.py -f ../mem.bin --profile=Win2003SP0
Volatility Foundation Volatility Framework 2.6.1
**************************************************
Process: 188 explorer.exe
Cache type "URL " at 0x856100
Record length: 0x100
Location: http://www.bing.com/fd/s/a/sw17.png
Last modified: 2012-10-04 23:57:41 UTC+0000
Last accessed: 2012-11-03 19:08:14 UTC+0000
File Offset: 0x100, Data Offset: 0x8c, Data Length: 0x98
File: sw17[1].png
Data: HTTP/1.1 200 OK
Content-Type: image/png
Content-Length: 4963
X-N: S

~U:administrator

**************************************************
Process: 188 explorer.exe
Cache type "URL " at 0x856200
Record length: 0x180
Location: http://www.bing.com/sa/8_01_0_1583456/UpdateDefaults.js
Last modified: 2010-09-30 13:16:47 UTC+0000
Last accessed: 2012-11-03 19:08:14 UTC+0000
File Offset: 0x180, Data Offset: 0xa0, Data Length: 0xb8
File: UpdateDefaults[1].js
Data: HTTP/1.0 200 OK
Content-Length: 656
Content-Type: application/x-javascript

**************************************************
Process: 188 explorer.exe
Cache type "URL " at 0x856380
Record length: 0x180
Location: http://www.bing.com/fd/sa/8_10_0_1578956/rmsloaderdelayed.js
Last modified: 2010-08-11 15:29:25 UTC+0000
Last accessed: 2012-11-03 19:08:14 UTC+0000
File Offset: 0x180, Data Offset: 0xa8, Data Length: 0xc0
File: rmsloaderdelayed[1].js
Data: HTTP/1.0 200 OK
Content-Length: 2868
Content-Type: application/x-javascript

**************************************************
Process: 188 explorer.exe
Cache type "URL " at 0x856680
```

Web history

```
**************************************************
Process: 188 explorer.exe
Cache type "URL " at 0x f85600
Record length: 0x100
Location: Cookie:administrator@c.atdmt.com/
Last modified: 2012-11-03 19:08:14 UTC+0000
Last accessed: 2012-11-03 19:08:15 UTC+0000
File Offset: 0x100, Data Offset: 0x8c, Data Length: 0x0
File: administrator@c.atdmt[2].txt
**************************************************
Process: 188 explorer.exe
Cache type "URL " at 0x f85700
Record length: 0x100
Location: Cookie:administrator@google.com/
Last modified: 2012-11-03 19:08:39 UTC+0000
Last accessed: 2012-11-03 19:08:39 UTC+0000
File Offset: 0x100, Data Offset: 0x8c, Data Length: 0x0
File: administrator@google[1].txt
**************************************************
Process: 188 explorer.exe
Cache type "URL " at 0x f85800
Record length: 0x100
Location: Cookie:administrator@www.googleadservices.com/pagead/conversion/1031414818/
Last modified: 2012-11-03 19:08:42 UTC+0000
Last accessed: 2012-11-03 19:08:42 UTC+0000
File Offset: 0x100, Data Offset: 0xb4, Data Length: 0x0
File: administrator@1031414818[1].txt
**************************************************
Process: 188 explorer.exe
Cache type "URL " at 0x f85900
Record length: 0x100
Location: Cookie:administrator@atdmt.com/
Last modified: 2012-11-03 19:08:42 UTC+0000
Last accessed: 2012-11-03 19:08:42 UTC+0000
File Offset: 0x100, Data Offset: 0x88, Data Length: 0x0
File: administrator@atdmt[2].txt
**************************************************
Process: 188 explorer.exe
Cache type "URL " at 0x f85a00
Record length: 0x100
Location: Cookie:administrator@support.microsoft.com/
Last modified: 2012-11-03 21:34:38 UTC+0000
Last accessed: 2012-11-03 21:34:38 UTC+0000
File Offset: 0x100, Data Offset: 0x94, Data Length: 0x0
File: administrator@support.microsoft[1].txt
**************************************************
Process: 188 explorer.exe
Cache type "URL " at 0x f85b00
Record length: 0x100
```

Window Explorer history

This plugin finds basic accessed links (via FTP or HTTP), redirected links (--REDR), and deleted items (--LEAK).

Any process that uses the wininet.dll library, will be scanned and recorded, meaning not only IE but also Windows Explorer and malware samples.

REFERENCES:

- https://github.com/volatilityfoundation/volatility/wiki/Installation
- https://pypi.org/project/pytz/
- https://github.com/volatilityfoundation/volatility/wiki/Linux-Command-Reference
- https://tools.kali.org/forensics/volatility
- https://pycryptodome.readthedocs.io/en/latest/src/installation.html
- https://www.volatilityfoundation.org/faq