Limon Sandbox for Analyzing Linux Malwares

Information Security Inc.

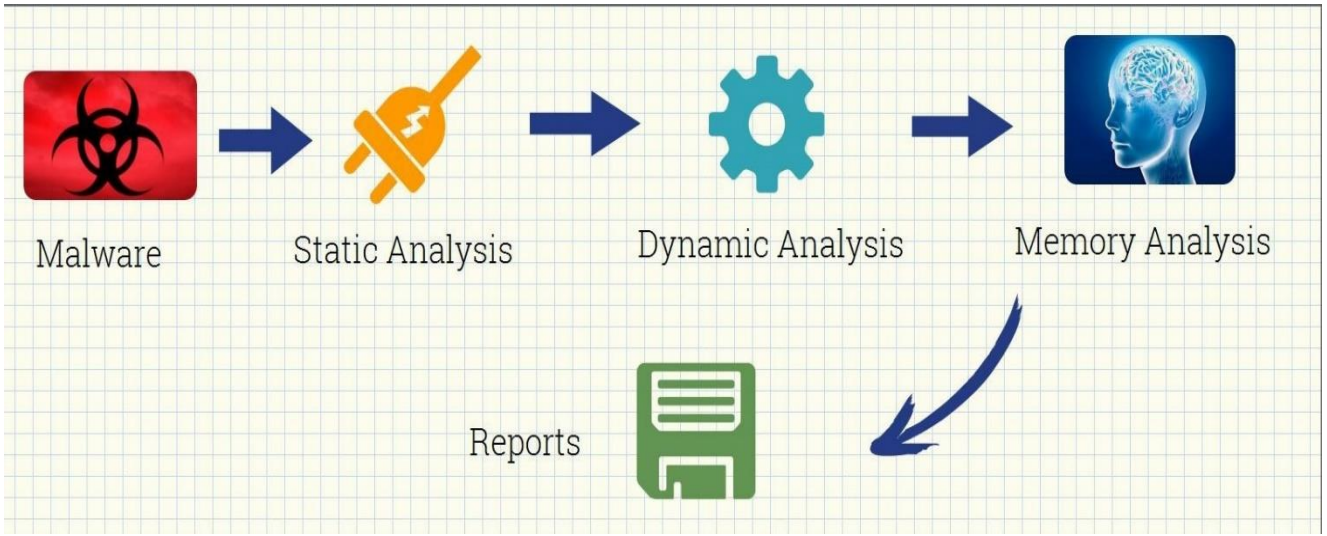
Contents

- About Limon Sandbox
- How Limon works
- Tools used by Limon
- Supported file types
- Testing environment
- Configuring and Installing Tools on the Host
- Configuring and Installing Tools on the Guest
- Virus Total API
- Configuring Limon
- Running Limon
- FAQ
- References

About Limon Sandbox

- Limon is a sandbox which automatically collects, analyzes, and reports on the run time indicators of Linux malware
- Performs static,dynamic and memory analysis
- Use various open source tools

How Limon works



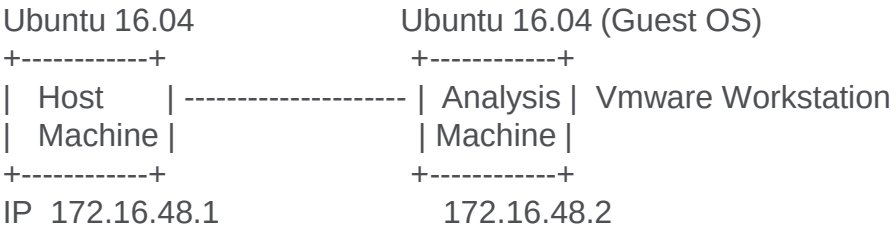
Tools used by Limon

- YARA-python (https://github.com/plusvic/yara/releases)
- VirusTotal Public api (https://www.virustotal.com/en/documentation/public-api/)
- ssdeep (http://ssdeep.sourceforge.net/)
- strings utility (http://linux.die.net/man/1/strings)
- ldd (http://linux.die.net/man/1/ldd)
- readelf (https://sourceware.org/binutils/docs/binutils/readelf.html)
- Inetsim (http://www.inetsim.org/downloads.html)
- Tcpdump (http://www.tcpdump.org/)
- strace (http://linux.die.net/man/1/strace)
- Sysdig (http://www.sysdig.org)
- Volatility memory forensics framework (http://www.volatilityfoundation.org/#!releases/component_7140)

Supported file types

- ELF Executable (x86 andx86_64)
- Perl Script
- Python Script
- Shell Script
- PHP Script
- Loadble kernel module(LKM)

Testing environment



Configuring and Installing Tools on the Host

- Installing Vmware Workstation; Installing Guest OS

```
root@admin1-virtual-machine:~# file VMware-Workstation-Full-12.5.5-5234757.x86_64.bundle
VMware-Workstation-Full-12.5.5-5234757.x86_64.bundle: a /usr/bin/env bash script executable (binary data)
root@admin1-virtual-machine:~# ./VMware-Workstation-Full-12.5.5-5234757.x86_64.bundle
Extracting VMware Installer...done.
You must accept the VMware Workstation End User License Agreement to
continue.  Press Enter to proceed.
```

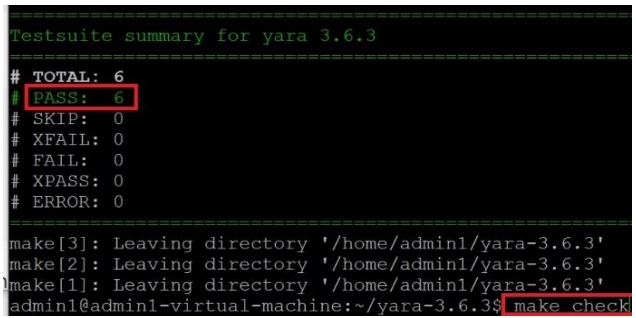
```
Installing VMware Workstation 12.5.5
    Configuring...
[##################################################################################] 100%
Installation was successful.
```

Configuring and Installing Tools on the Host

- Installing YARA

```
wget https://github.com/VirusTotal/yara/archive/v3.6.3.tar.gz
tar -zxf v3.6.3.tar.gz
cd yara-3.6.3/
sudo apt-get install automake libtool make gcc
./bootstrap.sh
```

- Run the test cases to make sure everything is fine



Configuring and Installing Tools on the Host

- Installing YARA-python

```
git clone https://github.com/VirusTotal/yara-python.git
rm -rf yara-python/
git clone  --recursive https://github.com/VirusTotal/yara-python.git
cd yara-python/
python setup.py build
sudo python setup.py install
```

- Installing ssdeep

```
apt-get install ssdeep
```

- Installing psutil

```
apt-get install python-psutil
```

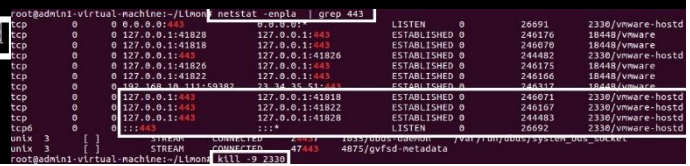
Configuring and Installing Tools on the Host

- Installing Sysdig (https://www.sysdig.org/install/)

```
curl -s https://s3.amazonaws.com/download.draios.com/DRAIOS-GPG-KEY.public | apt-key add -
curl -s -o /etc/apt/sources.list.d/draios.list http://download.draios.com/stable/deb/draios.list
apt-get update
apt-get -y install linux-headers-$(uname -r)
apt-get -y install sysdig
```

- Install inetsim (http://www.inetsim.org/packages.html)

```
echo "deb http://www.inetsim.org/debian/ binary/" > /etc/apt/sources.list.d/inetsim.list
wget -O - http://www.inetsim.org/inetsim-archive-signing-key.asc | apt-key add -
apt update
apt install inetsim
root@admin1-virtual-machine:/var/log/inetsim# pwd
/var/log/inetsim
root@admin1-virtual-machine:/var/log/inetsim# ls
debug.log  main.log  report  service.log
```



Configuring and Installing Tools on the Host

- Install Volatility

◎ Install requirements (distorm3)

```
sudo apt-get install python-distorm3
```

◎ Download volatility and run it

```
admin1@admin1-virtual-machine:~$ git clone https://github.com/volatilityfoundation/volatility
Cloning into 'volatility'...
remote: Counting objects: 26070, done.
remote: Total 26070 (delta 0), reused 0 (delta 0), pack-reused 26070
Receiving objects: 100% (26070/26070), 19.76 MiB | 2.56 MiB/s, done.
Resolving deltas: 100% (18686/18686), done.
Checking connectivity... done.
admin1@admin1-virtual-machine:~$ cd volatility/
admin1@admin1-virtual-machine:~/volatility$ chmod +x vol.py
admin1@admin1-virtual-machine:~/volatility$ ./vol.py
Volatility Foundation Volatility Framework 2.6
```

Configuring and Installing Tools on the Host

- Create a directory /root/yara_rules to store YARA rules
- Create a directory /root/linux_reports to store the analysis results

```
mkdir /root/yara_rules
mkdir /root/linux_reports
```

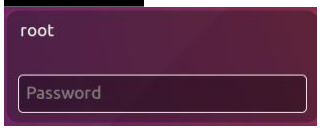
Configuring and Installing Tools on the Guest

- Set root password and enable graphical root login

```
sudo passwd root
nano /etc/lightdm/lightdm.conf
```

```
root@ubuntu:~# cat /etc/lightdm/lightdm.conf
[SeatDefaults]
greeter-session=unity-greeter
user-session=ubuntu
greeter-show-manual-login=true
```

```
reboot
```



Configuring and Installing Tools on the Guest

- Installing Sysdig (https://www.sysdig.org/install/)

```
curl -s https://s3.amazonaws.com/download.draios.com/DRAIOS-GPG-KEY.public | apt-key add -
curl -s -o /etc/apt/sources.list.d/draios.list http://download.draios.com/stable/deb/draios.list
apt-get update
apt-get -y install linux-headers-$(uname -r)
apt-get -y install sysdig
```

- Install strace (http://sourceforge.net/projects/strace/)

```
apt-get remove strace
wget https://sourceforge.net/projects/strace/files/latest/download
tar xvf download
cd strace-4.18/
./configure
make
make install
```

Configuring and Installing Tools on the Guest

- Installing PHP

```
apt-get install      php7.0-cli
```

- Install packages to run 32 bit executable on 64 bit ubuntu

```
dpkg --add-architecture i386
apt-get update
apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386
```

- Create directory to transfer malware sample

```
mkdir /root/malware_analysis
mkdir /root/logdir
nano /etc/environment
root@ubuntu:~# cat /etc/environment
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/root/malware analysis"
```

Configuring and Installing Tools on the Guest

- Create Volatility profile (https://github.com/volatilityfoundation/volatility/wiki/Linux)
- Limon relies on Volatility to perform memory analysis. After the malware is executed in the analysis machine, the analysis machine suspended to captures its memory image and memory analysis is performed
- Install Volatility (same as page 12)
- Install the following tools

```
apt-get install dwarfdump
apt-get install build-essential
```

```
root@ubuntu:~# uname -r
4.10.0-28-generic
root@ubuntu:~# apt-cache search linux-headers | grep 4.10.0-28
linux-headers-4.10.0-28 - Header files related to Linux kernel version 4.10.0
linux-headers-4.10.0-28-generic - Linux kernel headers for version 4.10.0 on 64 bit x86 SMP
linux-headers-4.10.0-28-lowlatency - Linux kernel headers for version 4.10.0 on 64 bit x86 SMP
root@ubuntu:~# apt-get install linux-headers-4.10.0-28-generic
Reading package lists... Done
Building dependency tree
Reading state information... Done
linux-headers-4.10.0-28-generic is already the newest version (4.10.0-28.32~16.04.2).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

Configuring and Installing Tools on the Guest

- Install the following tools

```
apt-get install dwarfdump
apt-get install build-essential
```

```
root@ubuntu:~# uname -r
4.10.0-28-generic
root@ubuntu:~# apt-cache search linux-headers | grep 4.10.0-28
linux-headers-4.10.0-28 - Header files related to Linux kernel version 4.10.0
linux-headers-4.10.0-28-generic - Linux kernel headers for version 4.10.0 on 64 bit x86 SMP
linux-headers-4.10.0-28-lowlatency - Linux kernel headers for version 4.10.0 on 64 bit x86 SMP
root@ubuntu:~# apt-get install linux-headers-4.10.0-28-generic
Reading package lists... Done
Building dependency tree
Reading state information... Done
linux-headers-4.10.0-28-generic is already the newest version (4.10.0-28.32~16.04.2).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

Configuring and Installing Tools on the Guest

- Creating the kernel data structures file using dwarfdump

◎ Vtypes => module.dwarf is created

```
cd volatility/tools/linux/
ls
make
ls -la
root@ubuntu:~/volatility/tools/linux# ls -la
total 2108
drwxr-xr-x 3 root root    4096 Aug 15 21:50 .
drwxr-xr-x 6 root root    4096 Aug 15 21:39 ..
drwxr-xr-x 2 root root    4096 Aug 15 21:39 kcore
-rw-r--r-- 1 root root     384 Aug 15 21:39 Makefile
-rw-r--r-- 1 root root     314 Aug 15 21:39 Makefile.enterprise
-rw-r--r-- 1 root root   17549 Aug 15 21:39 module.c
-rw-r--r-- 1 root root 2114384 Aug 15 21:50 module.dwarf
```

Configuring and Installing Tools on the Guest

- Creating the kernel data structures file using dwarfdump
  - ◎ Getting symbols

```
root@ubuntu:~# cd /boot
root@ubuntu:/boot# ls -la
total 53880
drwxr-xr-x  3 root root     4096 Aug 15 13:10 .
drwxr-xr-x 24 root root     4096 Aug 15 13:06 ..
-rw-r--r--  1 root root  1443598 Jul 20 07:11 abi-4.10.0-28-generic
-rw-r--r--  1 root root   204970 Jul 20 07:11 config-4.10.0-28-generic
drwxr-xr-x  5 root root     4096 Aug 15 13:07 grub
-rw-r--r--  1 root root 41822687 Aug 15 13:10 initrd.img-4.10.0-28-generic
-rw-r--r--  1 root root   182704 Jan 28  2016 memtest86+.bin
-rw-r--r--  1 root root   184380 Jan 28  2016 memtest86+.elf
-rw-r--r--  1 root root   184840 Jan 28  2016 memtest86+_multiboot.bin
-rw-------  1 root root  3718582 Jul 20 07:11 System.map-4.10.0-28-generic
-rw-r--r--  1 root root  7398656 Aug 15 13:06 vmlinuz-4.10.0-28-generic
root@ubuntu:/boot# uname -r
4.10.0-28-generic
root@ubuntu:/boot# file System.map-4.10.0-28-generic
System.map-4.10.0-28-generic: ASCII text
```

Configuring and Installing Tools on the Guest

- Creating the kernel data structures file using dwarfdump
  - ◎ Making the profile

```
root@ubuntu:~# zip volatility/volatility/plugins/overlays/linux/Ubuntu1604.zip volatility/tools/linux/module.dwarf /boot/System.map-4.10.0-28-generic
  adding: volatility/tools/linux/module.dwarf (deflated 89%)
  adding: boot/System.map-4.10.0-28-generic (deflated 79%)
```

- Clear bash history

```
history -c
history -w
```

- Take a clean snapshot
  - ◎ power off the analysis machine, power it on and then take a snapshot

Virus Total API

- Obtain Virus Total API Public key

◎ Login into an existing Virus Total account or create a new one (https://virustotal.com/#/join-us)

◎ Settings > API Key

Configuring Limon

- Download Limon and configure /// config.py ///

```
git clone https://github.com/monnappa22/Limon.git

cd Limon/

cp conf.py conf.pyorig
nano conf.py
```

Configuring Limon

- config.py

Configuring Limon

◎ config.py

###############[general variables]###################################

- py_path = r'/usr/bin/python'
- report_dir = r'/root/linux_reports'
- dash_lines = "-" * 40
- is_elf_file = False
- virustotal_key = "99e4bba81eb86c217be0bc0581e9bea96badfff03fe5b5bdb875aecb11d66a34"

###############[vm variables]###

- host_analysis_vmpath = r'/root/vmware/Ubuntu 64-bit/Ubuntu 64-bit.vmx'
- host_vmrunpath = r'/usr/bin/vmrun'
- host_vmtype = r'ws'
- analysis_username = "root"
- analysis_password = "CONstantin82"
- analysis_clean_snapname = "LimonSnap"
- analysis_mal_dir = r"/root/malware_analysis"
- analysis_py_path = r'/usr/bin/python'
- analysis_perl_path = r'/usr/bin/perl'
- analysis_bash_path = r'/bin/bash'
- analysis_sh_path = r'/bin/sh'
- analysis_insmod_path = r'/sbin/insmod'
- analysis_php_path = r'/usr/bin/php'

Configuring Limon

◎ config.py

```
################[static analysisvariables]#########################
• yara_packer_rules = r'/root/yara_rules/packer.yara'
• yara_rules = r'/root/yara_rules/capabilities.yara'

• #################[network variables]#################################
• analysis_ip = "172.16.48.2"
• host_iface_to_sniff = "ens33"
• host_tcpdumppath = "/usr/sbin/tcpdump"

########################[memory analysis variables]###################

• vol_path = r'/root/volatility/vol.py'
• mem_image_profile = '--profile=LinuxUbuntu1604x64'

######################[inetsim variables]#########################
• inetsim_path = r"/usr/bin/inetsim"
• inetsim_log_dir = r"/var/log/inetsim"
• inetsim_report_dir = r"/var/log/inetsim/report"

#####################[monitoring variables]#########################

• analysis_sysdig_path = r'/usr/bin/sysdig'
• host_sysdig_path = r'/usr/bin/sysdig'
• analysis_capture_out_file = r'/root/logdir/capture.scap'

• cap_format = "%proc.name (%thread.tid) %evt.dir %evt.type %evt.args"
• cap_filter = r"""evt.type=clone or evt.type=execve or evt.type=chdir or evt.type=open or
• evt.type=creat or evt.type=close or evt.type=socket or evt.type=bind or evt.type=connect or
• evt.type=accept or evt.is_io=true or evt.type=unlink or evt.type=rename or evt.type=brk or
• evt.type=mmap or evt.type=munmap or evt.type=kill or evt.type=pipe"""

• analysis_strace_path = r'/usr/local/bin/strace'
• strace_filter = r"-etrace=fork,clone,execve,chdir,open,creat,close,socket,connect,accept,bind,read,write,unlink,rename,kill,pipe,dup,dup2"
• analysis_strace_out_file = r'/root/logdir/trace.txt'

• analysis_log_outpath = r'/root/logdir'
```

Configuring Limon

◎ Limon Options

```
root@admin1-virtual-machine:~/Limon# python limon.py --help
Usage: limon.py [Options] <file> [args]

Options:
  -h, --help            show this help message and exit
  -t TIMEOUT, --timeout=TIMEOUT
                        timeout in seconds, default is 60 seconds
  -i, --internet        connects to internet
  -p, --perl            perl script (.pl)
  -P, --python          python script (.py)
  -z, --php             php script
  -s, --shell           shell script
  -b, --bash            BASH script
  -k, --lkm             load kernel module
  -C, --ufctrace        unfiltered call trace(full trace)
  -e, --femonitor       filtered system event monitoring
  -E, --ufemonitor      unfiltered system event monitoring
  -m, --memfor          memory forensics
  -M, --vmemfor         verbose memory forensics(slow)
  -x, --printhexdump    print hex dump in call trace (both filtered and
                        unfiltered call trace)
```

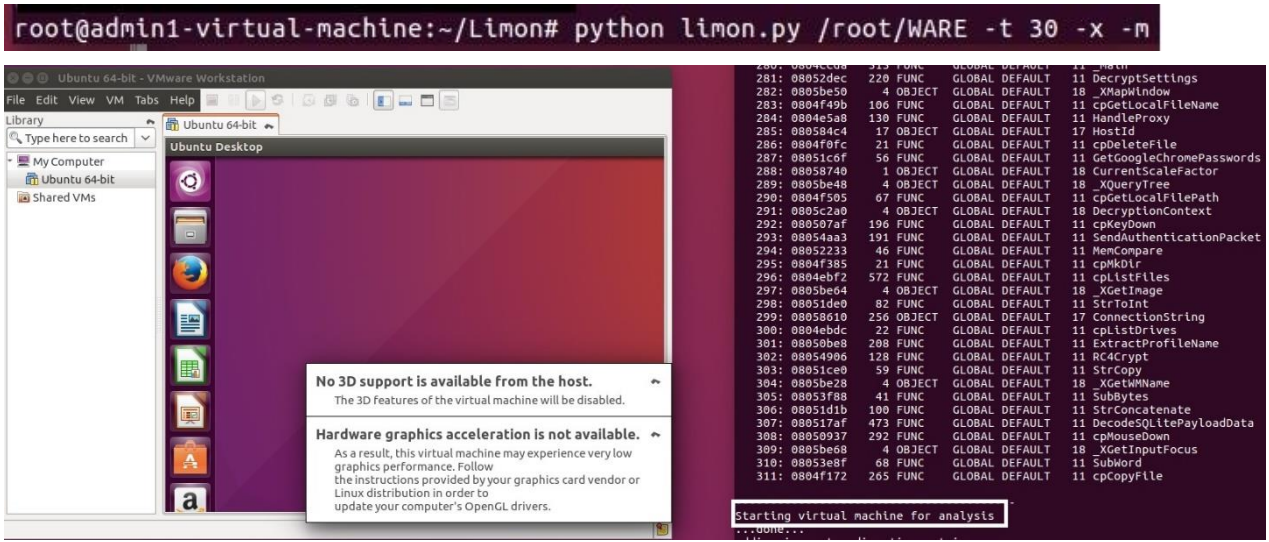
Running Limon

- Running ELF malware

```
root@admin1-virtual-machine:~/Limon# pwd
/root/Limon
root@admin1-virtual-machine:~/Limon# python limon.py /root/ELF_Malware -t 30 -x -m
Filetype: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=86e
a6af3ba586d28611e87948e4ac72665048c1, stripped
File Size: 367.38 KB (376192 bytes)
md5sum: 9361894a5998c6fd56d5cc4c615c79ee
ssdeep: 6144:S+ywNCe5u9iqf2vovYei759qkC1uWrKiZg//aZXXUPV:6wNCe5dZGYeSQfzG
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF64
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Advanced Micro Devices X86-64
  Version:                           0x1
  Entry point address:               0x4070d0
  Start of program headers:          64 (bytes into file)
  Start of section headers:          374336 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         9
  Size of section headers:           64 (bytes)
  Number of section headers:         29
  Section header string table index: 28
```

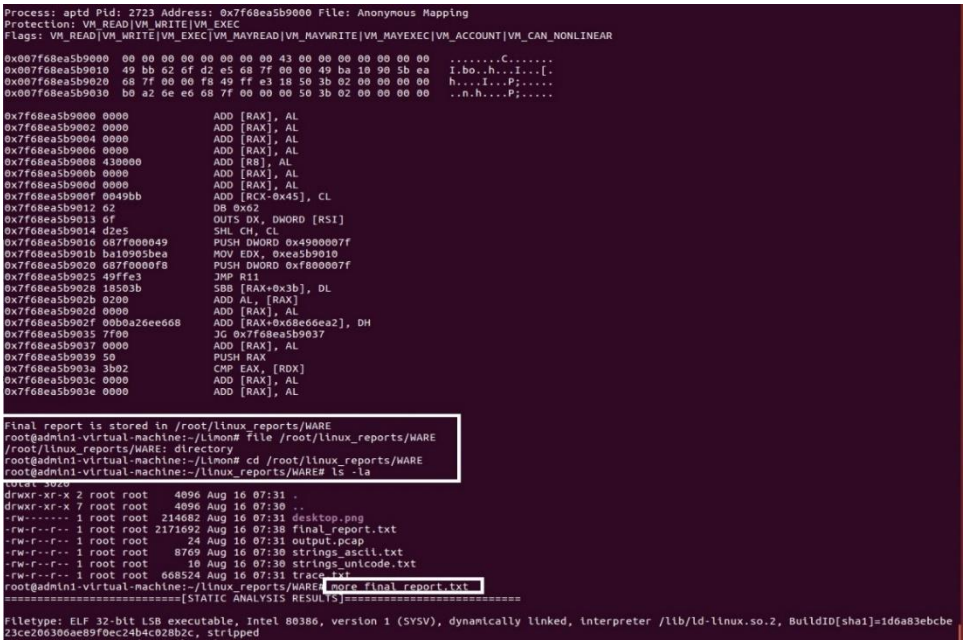
Running Limon

◎ Running ELF malware (Linux Wirenet https://github.com/ytisf/theZoo/tree/master/malwares/Binaries/Linux.Wirenet)



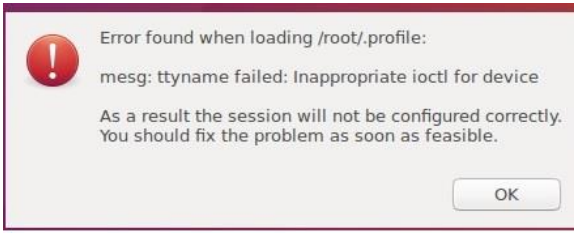
Running Limon

◎ Running ELF malware (Linux Wirenet https://github.com/ytisf/theZoo/tree/master/malwares/Binaries/Linux.Wirenet)



FAQ

◎ If getting the following error when login as root into GUI



◎ Ways to fix the issue (https://github.com/mitchellh/vagrant/issues/1673)

▲ change mesg n to tty -s && mesg n in /root/.profile
▲ remove the mesg n line from /root/.profile completely
▲ put a script named mesg in root's $PATH which only executes the real mesg if stdin is a tty

I chose

```
root@ubuntu:~# cat .profile
# ~/.profile: executed by Bourne-compatible login shells.

if [ "$BASH" ]; then
  if [ -f ~/.bashrc ]; then
    . ~/.bashrc
  fi
fi

tty -s && mesg n
```

References

- Github

https://github.com/monnappa22/Limon