LAB 9: LINUX FORENSICS (VOLATILE DATA COLLECTION)

Lab Requirements

1. Two (2) Linux Virtual Machines with Internet Connections. Each of the virtual machines should have two interfaces; a NAT for internet access and host-only network to allow virtual machines to communicate.

Content

Part I: Linux File Structure and Important Files

STEP 1: Kali Linux file structure conforms to the Filesystem Hierarchy Standard (FHS), which is adopted by many Linux distributions, including Debian from which Kali Linux is derived. Th following table lists the different directories in kali Linux and describes their content.

Directory	Description
`/bin/`	Basic user program (it is a symbolic link to `/usr/bin/` in Kali Linux)
`/boot/`	Boot files and the Kali Linux Kernel
`/dev/`	Device related files
`/etc/`	System and configuration files
`/home/`	User profiles and personal files
`/lib/`	Software libraries (a symbolic link to `/usr/lib/` in Kali Linux)
`/media/`	Mount folder for removable media (USB, etc.)
`/mnt/`	Temporary mount point
`/opt/`	Third party software
`/proc/`	System details

/root/	Root's user home directory
/run/	Volatile runtime data
/sbin/	System level binaries (it is a symbolic link to /usr/sbin/ in Kali Linux)
/srv/	Contains files for server applications
/tmp/	Temporary directory for files
/usr/	User shared files and binaries
/usr/bin	Contains Kali tools
/var/	Logs generated by daemons. A daemon is a utility program that runs in the background to perform and monitor certain tasks and to make sure that the operating system is running properly. (Use pstree or htop to display the tree of processes, including daemons).
/var/www/html/	Apache web server document root

```
1
2
3  # ~ refers to the current user's home directory
4  # / refers to the root directory of the file system
5  # The following is a partial list kali@kali [~] ls -l / total
6  68 lrwxrwxrwx   1 root root     7 Feb 11 01:53 bin -> usr/bin
7  drwxr-xr-x   3 root root  4096 Mar 18 14:13 boot drwxr-xr-x  17
8  root root  3280 Mar 19 15:13 dev drwxr-xr-x 164 root root 12288
9  Mar 19 14:43 etc drwxr-xr-x   3 root root  4096 Feb 11 02:05
10 home lrwxrwxrwx   1 root root     7 Feb 11 01:53 lib -> usr/lib
11 dr-xr-xr-x 258 root root     0 Mar 19 14:43 proc drwx------   4
12 root root  4096 Mar 13 15:34 root lrwxrwxrwx   1 root root
13 8 Feb 11 01:53 sbin -> usr/sbin drwxr-xr-x   3 root root  4096
14 Feb 11 02:00 srv dr-xr-xr-x  13 root root     0 Mar 19 14:43
15 sys drwxr-xr-x  14 root root  4096 Feb 11 01:53 usr drwxr-xr-x
16 12 root root  4096 Feb 11 01:55 var
17
18
```

STEP 2: Kali stores important information in files. The following is a list of important files.

File	Description
/etc/apt/sources.list	Sources that publish Debian packages

`/etc/fstab`	Static file system information
`/etc/group`	Local group information
`/etc/hostname`	Local machine's hostname
`/etc/hosts`	Contains hostname to IP address mapping
`/etc/network/interfaces`	Network configuration file
`/etc/passwd`	Local user account information
`/etc/profile`	Environment parameters
`/etc/resolv.conf`	Name server configuration file
`/etc/shadow`	Local user password hashes
`/etc/ssh/sshd_config`	SSH server configuration
`/etc/timezone`	System's time zone
`/home/kali/.bashrc`	A script that runs in every bash terminal session
`/home/kali/.bash_history`	Bash history file
`/home/kali/.zshrc`	A script that runs in every zsh terminal session
`/home/kali/.zshrc_history`	Zsh history file
`/proc/cpuinfo`	Information about the CPU
`/proc/crypto`	List of ciphers, hashing algorithms, and authentication algorithms supported by the kernel
`/proc/filesystems`	List of file systems supported by the kernel
`/proc/meminfo`	Information about the physical memory
`/proc/modules`	Currently loaded kernel modules
`/proc/partitions`	List of partitions (verify the list by checking the contents of `/proc/devices`)
`/proc/swaps`	Contains information about the system swap space. Swaps are used when the physical memory (RAM) is full. It is similar to the pagefile in Windows OS.
`/proc/version`	The version of the Linux kernel, the version of gcc, and the date the kernel was compiled.

/proc/uptime	Returns two values; the first is the total number of seconds the system has been up. The second is the sum of the idle time of all the processors.
/var/log/apache2/access.log	A log file containing access information of the Apache web server
/var/log/auth.log	A log file containing system authentication information
/var/log/boot.log	A log file containing information about the booting process
/var/log/btmp	A log file containing records of failed login attempts
/var/log/daemon.log	A log file containing information logged by background daemons
/var/log/dpkg.log	A log file containing information about packages installed or removed using the dpkg command
/var/log/messages	A log file containing all the global system messages including auth, kern, mail, etc. This is the main log file.
/var/log/syslog	A log file containing all the global system messages
/var/log/user.log	A log file containing user's logging

```
# Use the cat command to display the contents of a file
kali@kali [~] cat /proc/uptime 23499.92 92903.86

kali@kali [~] cat /proc/swaps
    Filename    Type       Size      Used  Priority
    /dev/sda5   partition  998396     0    -2

kali@kali [~] cat /proc/resolv.conf
    # Generated by NetworkManager
    search localdomain nameserver
    172.16.200.2

kali@kali [~] cat /proc/version
    Linux version 5.16.0-kali5-amd64 (devel@kali.org) (gcc-11 (Debian 11.2.016)
    11.2.0, GNU ld (GNU Binutils for Debian) 2.38) #1 SMP PREEMPT Debian
    5.16.14-1kali1 (2022-03-15)

```

Part II: Collecting Basic Volatile Information

STEP 3: Collect hostname, and time information using the following commands.

```
1
2
3  kali@kali [~] hostname kali
4
5  kali@kali [~] cat /proc/resolv.conf
6      Sat Mar 19 09:41:33 PM EDT 2022
7
8  kali@kali [~] cat /etc/timezone US/Eastern
9
10 # up time (25300.66), sum of idle time of all processors (100012.91)
11 kali@kali [~] cat /proc/uptime 25300.66
12     100012.91
13
14 # current time (21:44:25) | up time (days, hours): (0, 7:01) | number of logged
15 # on users (1) | system load average for the past 1, 5, and 15 minutes (0.17,
16 # 0.16, 0.17) kali@kali
17 [~] uptime
18     21:44:25 up  7:01,  1 user,  load average: 0.17, 0.16, 0.17
```

STEP 4: Collecting basic network information.

```
# Show IP addresses of all interfaces and related information kali@kali
[~] ip addr
    1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
    default qlen 1000     link/loopback 00:00:00:00:00:00 brd
    00:00:00:00:00:00     inet 127.0.0.1/8 scope host lo        valid_lft
    forever preferred_lft forever     inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
    2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    group default qlen 1000     link/ether 00:0c:29:7f:05:7d brd
    ff:ff:ff:ff:ff:ff
    3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    group default qlen 1000     link/ether 00:50:56:3d:f6:70 brd
    ff:ff:ff:ff:ff:ff     inet 172.16.200.134/24 brd 172.16.200.255 scope
    global dynamic noprefixroute eth1        valid_lft 1583sec preferred_lft
    1583sec
        inet6 fe80::250:56ff:fe3d:f670/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

# Show information for all interfaces kali@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 fq_codel state UP
    mode DEFAULT group default qlen 1000     link/ether 00:0c:29:7f:05:7d brd
    ff:ff:ff:ff:ff:ff
    3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    mode DEFAULT group default qlen 1000
    link/ether 00:50:56:3d:f6:70 brd ff:ff:ff:ff:ff:ff

# Show information for a given interface (eth1) kali@kali
[~] ip link show dev eth1
```

```
    3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    mode DEFAULT group default qlen 1000
    link/ether 00:50:56:3d:f6:70 brd ff:ff:ff:ff:ff:ff

# Display interface statistics (eth1) kali@kali
[~] ip -s link show dev eth1
    3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    mode DEFAULT group default qlen 1000     link/ether 00:50:56:3d:f6:70 brd
    ff:ff:ff:ff:ff:ff
        RX:  bytes packets errors dropped  missed   mcast
    690034    3350      0       0       0       0
        TX:  bytes packets errors dropped carrier collsns
             33632     315      0       0       0       0

# Show the routing table kali@kali [~] ip route default
via 172.16.200.2 dev eth1 proto dhcp metric 100
    172.16.200.0/24 dev eth1 proto kernel scope link src 172.16.200.134 metric
    100

# Display the ARP table (neigh is short for neighbours) kali@kali
[~] ip neigh
    172.16.200.2 dev eth1 lladdr 00:50:56:e2:d0:8a STALE
    172.16.200.254 dev eth1 lladdr 00:50:56:ea:f7:ae STALE
    172.16.200.132 dev eth1 lladdr 00:50:56:37:2e:d6 STALE

# Display socket statistics: -a (show all sockets), -e (detailed socket
# information), -o (timer information), -n (don't resolve addresses), -p (show
# process using the socket) kali@kali
[~] ss -a | head
Netid State  Recv-Q Send-Q      Local Address:Port   Peer Address:Port  Process
nl    UNCONN 0      0           rtnl:kernel                     *
nl    UNCONN 0      0           rtnl:NetworkManager/630         *
nl    UNCONN 0      0           rtnl:NetworkManager/630         *
nl    UNCONN 768    0           tcpdiag:kernel                  *
nl    UNCONN 4352   0           tcpdiag:ss/134012               *
nl    UNCONN 0      0           selinux:kernel                  *
nl    UNCONN 0      0           audit:-2029709523               *
nl    UNCONN 0      0           audit:kernel                    *
nl    UNCONN 0      0           audit:systemd/1                 *
```

NOTE 4-1: The commands `ip` and `ss` replace the obsolete command `netstat`.

STEP 5: Is any of the networks set to the promiscuous mode? If an interface is in the promiscuous mode, it accepts all received packets. This might be a malicious/benign packet sniffing technique. The ifconfig command can be used to check the mode of operation of an interface and change it if needed.

```
                    <UP,LOOPBACK,RUNNING>
kali@kali [~] ifconfig lo
    lo: flags=73m
    inet 127.0.0.1  netmask 255.0.0.0 inet6 ::1
    prefixlen  128    scopeid  0x10<host>  loop
    txqueuelen 1000  (Local Loopback)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
# Set interface lo to promiscuous mode kali@kali
[~] sudo ifconfig lo promisc

# Display the inte                  kali
                    <UP,LOOPBACK,RUNNING,PROMISC>
[~] ifconfig lo
    lo: flags=329m
    inet 127.0.0.1  netmask 255.0.0.0 inet6 ::1
    prefixlen  128    scopeid  0x10<host>  loop
    txqueuelen 1000  (Local Loopback)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
# Disable promiscuous mode on lo kali@kali
[~] sudo ifconfig lo -promisc

# Verify results
                    <UP,LOOPBACK,RUNNING>
[~] ifconfig lo
    lo: flags=73m
    inet 127.0.0.1  netmask 255.0.0.0 inet6 ::1
    prefixlen  128    scopeid  0x10<host>  loop
    txqueuelen 1000  (Local Loopback)
    RX packets 0  bytes 0 (0.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 0  bytes 0 (0.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

```

STEP 6: The `dmesg` command is used to retrieve the Kernel messages to help investigators track actions performed on the investigated machine. The command displays all messages since the kernel is started. Without any parameter, the time stamp is the period in seconds since the kernel was started. To display a human readable timestamp, use the `-T` option.

```
# Display the first 5 lines of the output kali@kali
[~] sudo dmesg | head -n 5


    [    0.000000] Linux version 5.16.0-kali5-amd64 (devel@kali.org) (gcc-11
    (Debian 11.2.0-16) 11.2.0, GNU ld (GNU Binutils for Debian) 2.38) #1 SMP
    PREEMPT Debian 5.16.14-1kali1 (2022-03-15)
    [    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.16.0-kali5-amd64
    root=UUID=c846c5cd-8447-4d17-a782-8e5bf4be60ae ro quiet splash
    [    0.000000] Disabled fast string operations
    [    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point
    registers'
    [    0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
# Display the first two lines of the dmesg output. The timestamp is in human
# readable format. kali@kali [~] sudo
dmesg -T | head -n 2
    [Sun Mar 20 05:12:32 2022] Linux version 5.16.0-kali5-amd64
    (devel@kali.org) (gcc-11 (Debian 11.2.0-16) 11.2.0, GNU ld (GNU Binutils
for Debian) 2.38) #1 SMP PREEMPT Debian 5.16.14-1kali1 (2022-03-15) [Sun Mar
20 05:12:32 2022] Command line: BOOT_IMAGE=/boot/vmlinuz-5.16.0kali5-amd64
root=UUID=c846c5cd-8447-4d17-a782-8e5bf4be60ae ro quiet splash
# Display lines of output that contain “device lo”. As I have already enables
# and disabled promiscuous mode on the lo interface, the following messages
# were logged in the kernel log. kali@kali [~]
sudo dmesg -T | grep "device lo"
    [Sun Mar 20 14:21:45 2022] device lo entered promiscuous mode
    [Sun Mar 20 14:24:18 2022] device lo left promiscuous mode
    [Sun Mar 20 14:29:18 2022] device lo entered promiscuous mode
    [Sun Mar 20 14:31:19 2022] device lo left promiscuous mode
```

NOTE 6-1: Check the manual of the `dmesg` command to get more information on the usage of this important command.

STEP 7: To list the open files, use the `lsof` command.

```
1
2  # Display the first open files (8 is the device sda, 1 is the partition sda1).
3  kali@kali [~] sudo lsof | head -n 4
4      COMMAND      PID    TID TASKCMD     USER   FD      TYPE             DEVICE
5      SIZE/OFF        NODE NAME systemd         1                       root  cwd
       DIR               8,1
6      36864          2 / systemd        1                    root  rtd       DIR
7      8,1
8      36864          2 / systemd        1                    root  txt       REG
9      8,1   1845808    3805338 /usr/lib/systemd/systemd
10
11 # Display the files opened by network connections kali@kali
12 [~] sudo lsof -i
13
```

```
    COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
    NetworkMa 630 root   25u  IPv4  20949      0t0  UDP 172.16.200.134:bootpc-
    >172.16.200.254:bootps

# Display the files opened by a given user – first count the number ot returned
# lines
kali@kali [~] sudo lsof -u kali | wc -l 4692

# Display the first 5 lines from the returned results
kali@kali [~] sudo lsof -u kali | head -n 5
    COMMAND      PID USER   FD      TYPE       DEVICE  SIZE/OFF       NODE NAME
    systemd      894 kali  cwd       DIR       8,1     36864          2 / systemd
    894 kali  rtd       DIR       8,1     36864          2 / systemd      894
    kali  txt       REG       8,1   1845808    3805338
    /usr/lib/systemd/systemd systemd      894 kali  mem       REG
    8,1    157768    3802031 /usr/lib/x86_64-linux-gnu/libgpg-
    error.so.0.32.1

# Display all opened files by a particular process – I use ssh in the following
# example. [DIR: directory, REG: regular file, CHR: character special file]
kali@kali [~] sudo lsof -c ssh
    COMMAND   PID USER   FD   TYPE             DEVICE SIZE/OFF    NODE NAME
    ssh-agent 969 kali  cwd    DIR                8,1    36864       2 /
    ssh-agent 969 kali  rtd    DIR                8,1    36864       2 /
    ssh-agent 969 kali  txt    REG               8,1   457088 3803932
    /usr/bin/ssh-agent ssh-agent 969 kali  mem    REG
    8,1   143768 3814507
    /usr/lib/x86_64-linux-gnu/libpthread-2.33.so ssh-agent 969 kali
    mem    REG               8,1    22864 3814497  ...
    ssh-agent 969 kali    0u   CHR               1,3      0t0       4
    /dev/null ssh-agent 969 kali    1u   CHR               1,3
    0t0       4
    /dev/null ssh-agent 969 kali    2u   CHR               1,3
    0t0       4
    /dev/null ssh-agent 969 kali    3u  unix 0x0000000069015f71
    0t0   21270
    /tmp/ssh-XXXXXXi2C87u/agent.919 type=STREAM
```

NOTE 7-1: Use the command `cat /proc/devices` to list the IDs of the recognized devices.

STEP 8: The command `mount` lists the mounted file systems and the corresponding mounting directories.

```
kali@kali [~] mount -l
```

```
    sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
    proc on /proc type proc (rw,nosuid,nodev,noexec,relatime) udev
    on /dev type devtmpfs
    (rw,nosuid,relatime,size=953036k,nr_inodes=238259,mode=755,inode64) devpts
    on /dev/pts type devpts
    (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000) tmpfs
    on /run type tmpfs
    (rw,nosuid,nodev,noexec,relatime,size=199348k,mode=755,inode64)
    /dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro) ...


```

STEP 9: The command `df` displays the amounts of free and used space on mounted file systems, and the corresponding mounting directories.

```
kali@kali [~] df
    Filesystem     1K-blocks     Used Available Use% Mounted on
    udev              953036        0    953036   0% /dev tmpfs
    199348     1180    198168   1% /run /dev/sda1       81000912
    16362196  60478104  22% / tmpfs             996724        0
    996724   0% /dev/shm tmpfs               5120        0
    5120   0% /run/lock tmpfs             199344       80    199264
    1% /run/user/1000
```

STEP 10: To display the kernel loaded modules, use the `lsmod` command.

```

# Display the first 6 lines of the output. The complete list is long.
kali@kali [~] lsmod | head -n 6
    Module                  Size  Used by
    xt_recent              24576  0 snd_seq_midi
    20480  0 snd_seq_midi_event     16384  1
    snd_seq_midi snd_seq_dummy          16384  0
    snd_hrtimer            16384  1

```

STEP 11: Use the command `modinfo` to display the information of a particular module. The `vsock` module facilitates the communication between the host machine and the virtual machine.

```
kali@kali [~] modinfo vsock       filename:
     /lib/modules/5.16.0-
     kali5amd64/kernel/net/vmw_vsock/vsock.ko
     license:        GPL v2 version:
     1.0.2.0-k description:    VMware Virtual
     Socket Family author:         VMware, Inc.
     srcversion:     B5B3B334C2D99AA6BCF2F9A depends:         retpoline:
     Y intree:         Y name:           vsock vermagic:       5.16.0-
     kali5-amd64 SMP preempt mod_unload modversions
```

STEP 12: The commands `insmod` and `rmmode` are used to insert modules into and remove modules from the kernel, respectively.

STEP 13: The command `ps` is used to get details on processes.

```
# Show the processes for the current terminal kali@kali
[~] ps
    PID TTY          TIME CMD
     84312 pts/1    00:00:01 zsh
    111112 pts/1    00:00:00 ps

# Show the processes for the current terminal
# SPID: Server PID
# ppid: Parent Process ID
# pts: pseudo terminal value
# pty: pseudo terminal device
# tty: terminal type the user is logged on into kali@kali
[~] ps -T
    PID    SPID TTY          TIME     CMD
     84312   84312 pts/1    00:00:01 zsh
    111746  111746 pts/1    00:00:00 ps
# List all running processes, equivalent to ps -A kali@kali
[~] ps -e | head
    PID TTY         TIME CMD
1 ?        00:00:03 systemd
2 ?        00:00:00 kthreadd
3 ?        00:00:00 rcu_gp
4 ?        00:00:00 rcu_par_gp
     6 ?        00:00:00 kworker/0:0H-events_highpri
9  ?        00:00:00 mm_percpu_wq
10 ?        00:00:00 rcu_tasks_kthre
11 ?        00:00:00 rcu_tasks_rude_
12 ?        00:00:00 rcu_tasks_trace

# List processes associated with the current user kali@kali
[~] $ ps -x | head -n 5
    PID TTY      STAT   TIME COMMAND
    934 ?        Ss     0:00 /lib/systemd/systemd --user
```

```
    935 ?        S      0:00 (sd-pam)
949    ?        S<sl   0:00 /usr/bin/pipewire
950    ?        Ssl    0:03 /usr/bin/pipewire-media-session
# List processes by process id kali@kali
[~] $ ps -p 934 935 1019
    PID TTY      STAT   TIME COMMAND
934 ?        Ss     0:00 /lib/systemd/systemd --user
935 ?        S      0:00 (sd-pam)
   1019 ?        Ssl    0:00 /usr/libexec/at-spi-bus-launcher
# List processes: user name, pid, ppid, session id, and arguments (command)
kali@kali [~] $ ps -A -o user,pid,ppid,sess,args | head USER          PID
PPID    SESS COMMAND root          1       0       1 /sbin/init splash root
2       0       0 [kthreadd] root          3       2       0 [rcu_gp] root
4       2       0 [rcu_par_gp] root          6       2       0 [kworker/0:0H-
events_highpri] root             9         2         0 [mm_percpu_wq] root
10        2        0 [rcu_tasks_kthre] root           11        2        0
[rcu_tasks_rude_] root          12       2       0 [rcu_tasks_trace]

# List processes attributed to a particular session
kali@kali [~] $ ps -s 959 | head
    PID TTY          TIME CMD
    959 ?        00:00:00 xfce4-session
   1047 ?        00:02:21 xfwm4
   1077 ?        00:00:00 xfsettingsd
   1085 ?        00:00:02 xfce4-panel
   1089 ?        00:00:00 Thunar
1094 ?        00:00:04 xfdesktop
1095 ?        00:00:01 panel-1-whisker
   1098 ?        00:02:47 panel-13-cpugra
   1101 ?        00:00:00 panel-14-systra
```

STEP 14: The command `pmap` is used to report on the memory map of a particular process.

```
# The .so (shared library) are files are similar to the dll files in Windows.
# Note that 1000x equivalent to 4096 (4k block size) kali@kali
[~] $ pmap -p 84312 | head
     84312:   /usr/bin/zsh
     00005602a46c4000     92K r---- /usr/bin/zsh
     00005602a46db000    596K r-x-- /usr/bin/zsh
     00005602a4770000    136K r---- /usr/bin/zsh
     00005602a4793000      8K r---- /usr/bin/zsh
     00005602a4795000     24K rw--- /usr/bin/zsh
     00005602a479b000     80K rw---   [ anon ]
     00005602a5c3f000   1760K rw---   [ anon ]
     00007fcfbf8f1000      4K r---- /usr/lib/x86_64-
     linuxgnu/zsh/5.8.1/zsh/regex.so
     00007fcfbf8f2000      4K r-x-- /usr/lib/x86_64-
     linuxgnu/zsh/5.8.1/zsh/regex.so
```

STEP 15: The command `strace` is used to trace the system calls and signals issued by a particular process. First, let us find the PID of the zsh (Z shell) and trace its system calls from another terminal.

```
# Find the PD of the zsh: Terminal 1 kali@kali
[~] $ ps -A | grep zsh
     65957 pts/0    00:00:04 zsh
     84312 pts/1    00:00:07 zsh

# Terminal 2 kali@kali [~] $
strace -p 8312

# Now, write any command in Terminal 1 (for example ls). The system calls will
# appear in Terminal 2. s
```

Part III: Linux Firewall, SSH Service, and Port Scanning using nmap

Laboratory settings:

- A Linux VM (Name: VM1; IP: 172.16.200.135) with ufw and ssh services installed (Kali Linux in the following).
- A Linux VM (Name: VM2; IP: 172.16.200.132) – Parrot Security in the following.

STEP 16: Beware that open port scanning without receiving permission to do so might be suspicious and might be illegal as it is used by attackers in the reconnaissance phase of initiating attacks. Refer to the following page for more details: https://nmap.org/book/legal-issues.html

STEP 17: Use `nmap` (network mapper) command for port scanning. The server whose IP address 45.33.32.156 is scanned. The option `-s` means scan and `-T` means TCP.

```
kali@kali [~] nmap -sT scanme.nmap.org
    Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 08:53 EDT Nmap
    scan report for scanme.nmap.org (45.33.32.156)
    Host is up (0.071s latency).
    Other addresses for scanme.nmap.org (not scanned):
    2600:3c01::f03c:91ff:fe18:bb2f
    Not shown: 992 closed tcp ports (conn-refused)
    PORT      STATE    SERVICE
    22/tcp    open     ssh
    25/tcp    filtered smtp
    80/tcp    open     http
    135/tcp   filtered msrpc
    139/tcp   filtered netbios-ssn
    445/tcp   filtered microsoft-ds
    9929/tcp  open     nping-echo
    31337/tcp open     Elite
```

NOTE 11-1: An application is listening on **open** ports. A port is **filtered** if the packet is dopped by a firewall, filter or a midway device, and `nmap` can't decide whether the port is open or closed. A port is **closed** if it is accessible by `nmap` but there is no application listening on that port. Refer to the following webpage for more information: https://wiki.onap.org/display/DW/Nmap

STEP 18 [VM1 & VM2]: The `ufw` firewall is pre-installed on many Linux distributions. To install and start `ufw` on VM1 (`ufw` stands for uncomplicated firewall), use the following commands:

```
# Upgrade system packages and install ufw firewall
kali@kali [~] sudo apt-get upgrade kali@kali [~]
sudo apt-get install ufw
# Display the status of the ufw service using the systemctl command kali@kali
[~] sudo systemctl status ufw
     ○ ufw.service - Uncomplicated firewall
          Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor
     preset: enabled)
          Active: inactive (dead) since Mon 2022-03-21 19:40:38 EDT; 8s ago
            Docs: man:ufw(8)
         Process: 15818 ExecStop=/lib/ufw/ufw-init stop (code=exited,
     status=0/SUCCESS)
        Main PID: 377 (code=exited, status=0/SUCCESS)
     CPU: 248ms

# Start the ufw service using the systemctl command kali@kali
[~] sudo systemctl start ufw

kali@kali [~] sudo systemctl status ufw
     ● ufw.service - Uncomplicated firewall
          Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor
     preset: enabled)
```

```
          Active: active (exited) since Mon 2022-03-21 22:33:26 EDT; 2s ago
     Docs: man:ufw(8)
         Process: 39026 ExecStart=/lib/ufw/ufw-init start quiet (code=exited,
     status=0/SUCCESS)
        Main PID: 39026 (code=exited, status=0/SUCCESS)
             CPU: 285ms

# Display the status of firewall
kali@kali [~] sudo ufw status Status:
active

# Display more details of the firewall status kali@kali
[~] sudo ufw status verbose
     Status: active
     Logging: on (low)
     Default: deny (incoming), allow (outgoing), disabled (routed)
     New profiles: skip

# Add a new rule: allow tcp protocol on port 22 (SSH service uses TCP on port
# 22
kali@kali [~] sudo ufw allow 22/tcp
     Rule added
     Rule added (v6)

# Display more details of the firewall status
kali@kali [~] sudo ufw status verbose
     Status: active
     Logging: on (low)
     Default: deny (incoming), allow (outgoing), disabled (routed)
     New profiles: skip

     To                         Action      From
     --                         ------      ----
     22/tcp                     ALLOW IN    Anywhere
     22/tcp (v6)                ALLOW IN    Anywhere (v6)

# Use nmap to list the open ports on the local host
# I assume that even if the firewall is allowing traffic in on port 22, the SSH
# service is not started (no application is listening on that port) #
NOTE: nmap does not check the firewall when scanning the localhost.
kali@kali [~] sudo nmap -sT localhost
     Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 23:09 EDT Nmap
     scan report for localhost (127.0.0.1)
     Host is up (0.00013s latency).
     Other addresses for localhost (not scanned): ::1
     All 1000 scanned ports on localhost (127.0.0.1) are in ignored states.
     Not shown: 1000 closed tcp ports (conn-refused)
     Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```

```
72
73  # Run the same command on VM2 and replace localhost with the IP address of  #
74  VM1's host-only network interface. Commands eexecuted on VM2 are in italic
75  # font in the following shell commands and results.
76  # NOTE: Nmap on VM2 can check the status of the ports set by the firewall as it
77  # is outside the network perimeter firewall. user@parrot
78  [~] sudo nmap -sT 172.16.200.135
79       Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 03:15 GMT Nmap
80       scan report for localhost 172.16.200.135
81       Host is up (0.0012s latency).
82       Not shown: 999 filtered tcp ports (no-response)
83       PORT   STATE   SERVICE
84       22/tcp closed  ssh
85       MAC Address: 00:60:56:2E:51:7A (Network Tools)
86       Nmap done: 1 IP address (1 host up) scanned in 4.70 seconds
87
88
```

STEP 19 [VM1 & VM2]: SSH (Secure Shell) is a protocol used by Linux-based network administrators for management of remote systems. OpenSSH is an open-source secure shell tool. To install, enable, and handle OpenSSH, use the following commands:

```
kali@kali [~] $ sudo apt-get upgrade

# Install the openssh tool kali@kali [~] $ sudo
apt-get install openssh-server

# Display the status of the ssh service
kali@kali [~] sudo systemctl status ssh | grep -i active
     Active: inactive (dead) since Mon 2022-03-21 23:06:13 EDT; 23min ago

# Start the ssh service kali@kali [~]
sudo systemctl start ssh

# Display the status of the ssh service kali@kali [~]
sudo systemctl status ssh | grep -i active
     Active: active (running) since Mon 2022-03-21 23:33:19 EDT; 1s ago
# Run the same command on VM2
user@parrot [~] sudo nmap -sT 172.16.200.135
     Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 03:35 GMT Nmap
     scan report for localhost 172.16.200.135
     Host is up (0.0012s latency).
     Not shown: 999 filtered tcp ports (no-response)
     PORT   STATE   SERVICE
     22/tcp open    ssh
     MAC Address: 00:60:56:2E:51:7A (Network Tools)



     Nmap done: 1 IP address (1 host up) scanned in 4.94 seconds
```

STEP 20 [VM2]: Use the OpenSSH tool to establish a secure connection between VM2 and VM1.

```
# Establish a connection from 172.16.200.132 to 172.16.200.135
user@parrot [~] ssh kali@172.16.200.135
    The authenticity of host '172.16.200.135 (172.16.200.135)' can't be
    established.
    ECDSA key fingerprint is
    SHA256:EIwdqMw+h/QRTW4AXXeaA8GOq3NdKFBZelcz3IbltEs.
    Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
    Warning: Permanently added '172.16.200.135' (ECDSA) to the list of known
    hosts.
    kali@172.16.200.135's password:
    Linux kali 5.16.0-kali5-amd64 #1 SMP PREEMPT Debian 5.16.14-1kali1 (2022-
    03-15) x86_64

    The programs included with the Kali GNU/Linux system are free software; the
    exact distribution terms for each program are described in the individual
    files in /usr/share/doc/*/copyright.

    Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted
    by applicable law.

# Connected to 172.16.200.135 as kali kali@kali [~] echo Test
from the Parrot Security host > test.txt
# Open a terminal on VM1 and list the files on the home directory of the kali
# user. Verify that the file was test.txt was created and display its contents.
kali@kali [~] ls
    Desktop    Downloads  Music     Public    Templates  Videos
    Documents  error.txt  Pictures  sherlock  test.txt

```

Part IV: Linux Auditing System (Next Week)

STEP 21: We will go over `auditd` and `go-audit` Linux auditing systems in the next class.