Sushma Shivshankar Nandiyawar
Lab 7

Q1: What is the result of running the command? Use tshark data to explain how the rootkit sets up/ implements the backdoor. Use screenshots to capture the tshark data. In addition, save the Packet Capture Data (pcap) file and submit the screenshots and the pcap file as part of your writeup

Answer:
1. Unusual Communication: Utilize tshark to detect unconventional communication patterns by filtering for traffic involving the suspicious host and relevant ports.

2. Outbound Connections: Focus on outbound connections initiated by the host. Leverage tshark to filter for outbound connections specifically from the identified IP address.

3. Encrypted Traffic: Investigate tshark data for encrypted traffic on non-standard ports, as rootkits might employ encryption to conceal their communication.

4. Payload Analysis: Scrutinize packet payloads for any unusual or suspicious data. Utilize tshark to reveal packet details and analyze payload data.

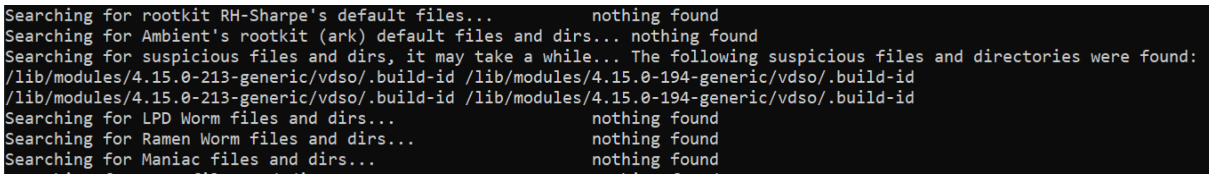
5. Persistence Mechanisms:
Identify persistent communication patterns by observing repeated connections or modifications to system configurations in tshark data.

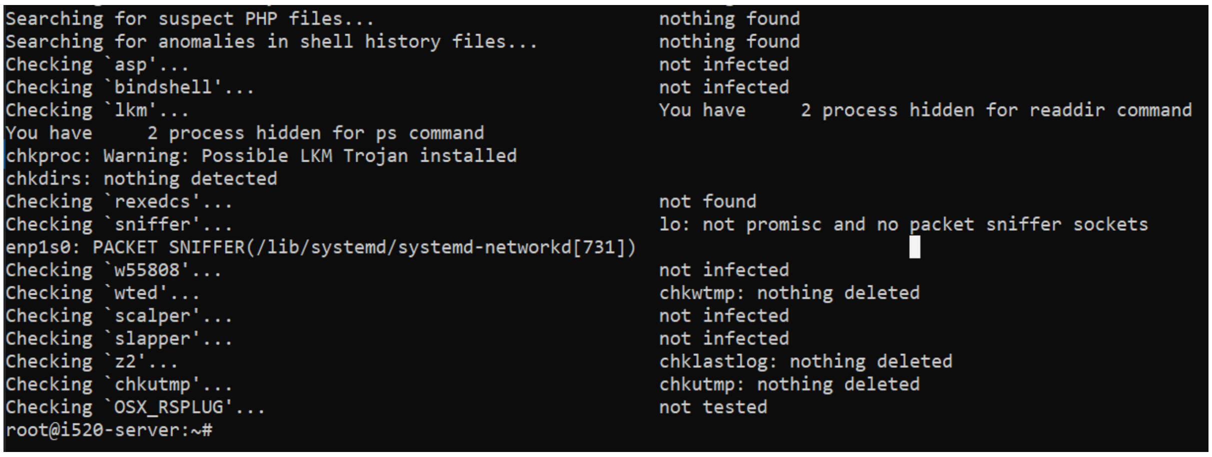
6. Timing and Frequency: Analyze communication timing and frequency using timestamps in tshark. Look for rapid or sporadic communication patterns that may indicate malicious activity.

7. Authentication Attempts: Filter tshark data for packets indicating authentication attempts, with a focus on specific ports associated with authentication (e.g., port 666).

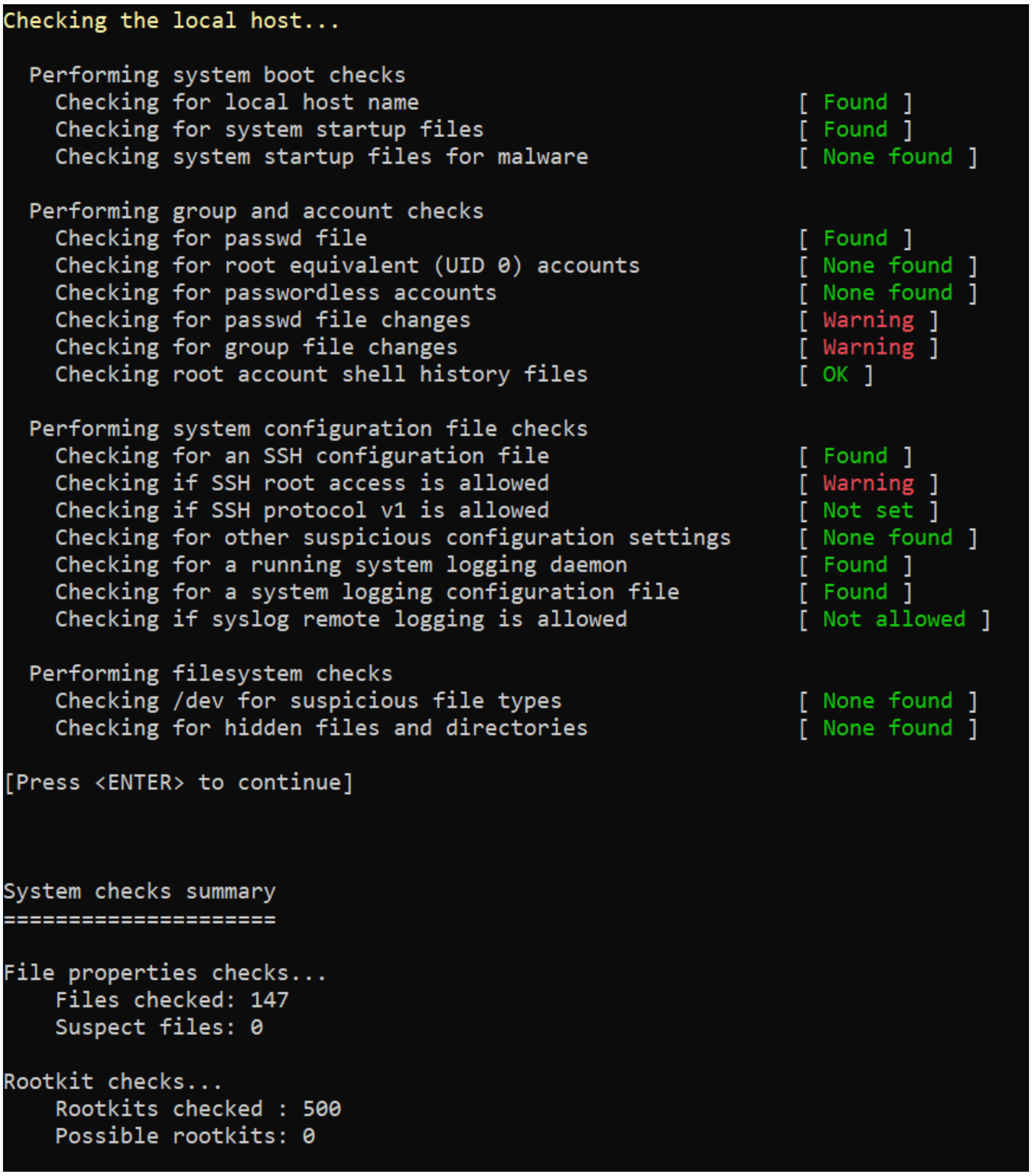
8. Evasion Techniques: Examine tshark data for signs of evasion techniques. Check for fragmentation, packet reordering, or other methods used to evade detection.

The screenshots illustrating the execution of the pcap command and the contents of the corresponding file are attached.

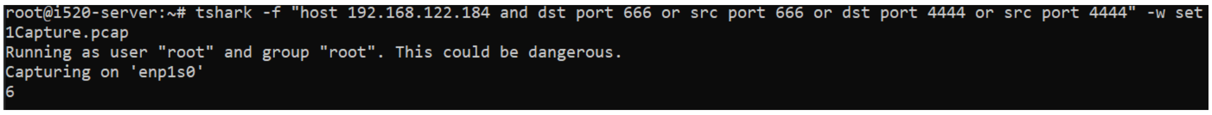

```
Searching for rootkit RH-Sharpe's default files...          nothing found
Searching for Ambient's rootkit (ark) default files and dirs... nothing found
Searching for suspicious files and dirs, it may take a while... The following suspicious files and directories were found:
/lib/modules/4.15.0-213-generic/vdso/.build-id /lib/modules/4.15.0-194-generic/vdso/.build-id
/lib/modules/4.15.0-213-generic/vdso/.build-id /lib/modules/4.15.0-194-generic/vdso/.build-id
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
```



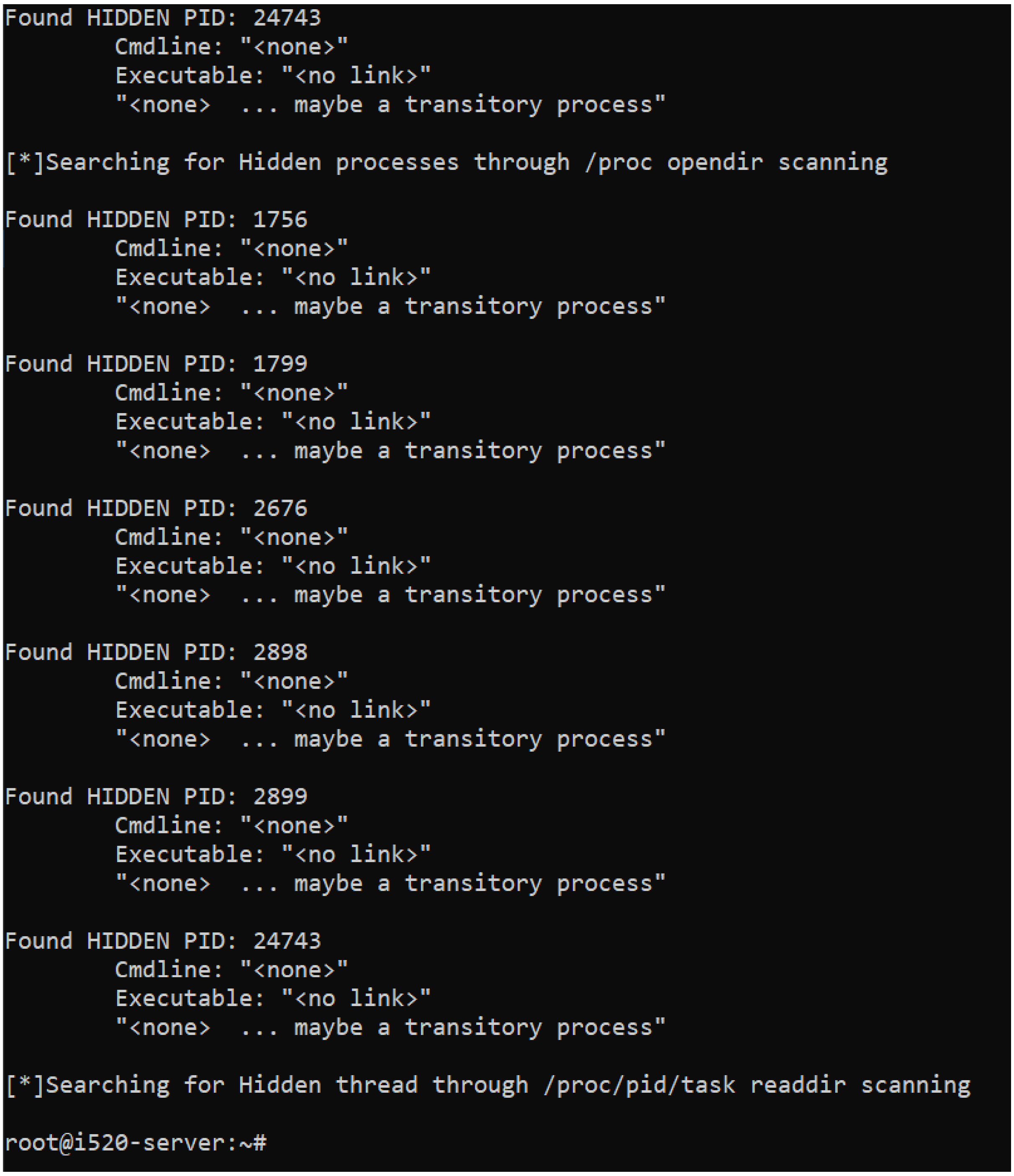
```
Searching for suspect PHP files...                          nothing found
Searching for anomalies in shell history files...           nothing found
Checking `asp'...                                           not infected
Checking `bindshell'...                                     not infected
Checking `lkm'...                                           You have     2 process hidden for readdir command
You have     2 process hidden for ps command
chkproc: Warning: Possible LKM Trojan installed
chkdirs: nothing detected
Checking `rexedcs'...                                       not found
Checking `sniffer'...                                       lo: not promisc and no packet sniffer sockets
enp1s0: PACKET SNIFFER(/lib/systemd/systemd-networkd[731])
Checking `w55808'...                                        not infected
Checking `wted'...                                          chkwtmp: nothing deleted
Checking `scalper'...                                       not infected
Checking `slapper'...                                       not infected
Checking `z2'...                                            chklastlog: nothing deleted
Checking `chkutmp'...                                       chkutmp: nothing deleted
Checking `OSX_RSPLUG'...                                    not tested
root@i520-server:~#
```



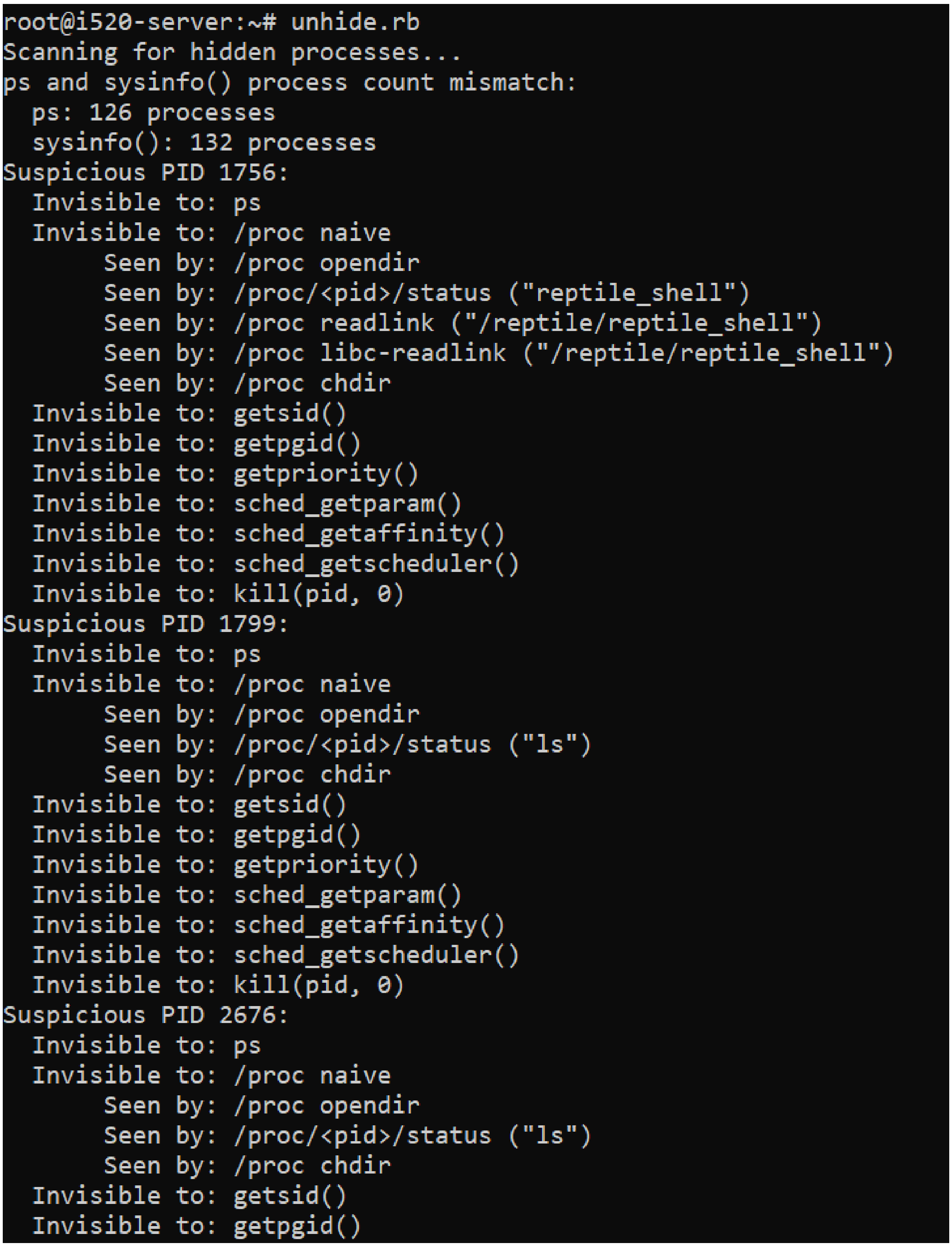
```
Checking the local host...

  Performing system boot checks
    Checking for local host name                             [ Found ]
    Checking for system startup files                        [ Found ]
    Checking system startup files for malware                [ None found ]

  Performing group and account checks
    Checking for passwd file                                 [ Found ]
    Checking for root equivalent (UID 0) accounts            [ None found ]
    Checking for passwordless accounts                       [ None found ]
    Checking for passwd file changes                         [ Warning ]
    Checking for group file changes                          [ Warning ]
    Checking root account shell history files                [ OK ]

  Performing system configuration file checks
    Checking for an SSH configuration file                   [ Found ]
    Checking if SSH root access is allowed                   [ Warning ]
    Checking if SSH protocol v1 is allowed                   [ Not set ]
    Checking for other suspicious configuration settings     [ None found ]
    Checking for a running system logging daemon             [ Found ]
    Checking for a system logging configuration file         [ Found ]
    Checking if syslog remote logging is allowed             [ Not allowed ]

  Performing filesystem checks
    Checking /dev for suspicious file types                  [ None found ]
    Checking for hidden files and directories                [ None found ]

[Press <ENTER> to continue]



System checks summary
=====================

File properties checks...
    Files checked: 147
    Suspect files: 0

Rootkit checks...
    Rootkits checked : 500
    Possible rootkits: 0
```



```
root@i520-server:~# tshark -f "host 192.168.122.184 and dst port 666 or src port 666 or dst port 4444 or src port 4444" -w set
1Capture.pcap
Running as user "root" and group "root". This could be dangerous.
Capturing on 'enp1s0'
6
```

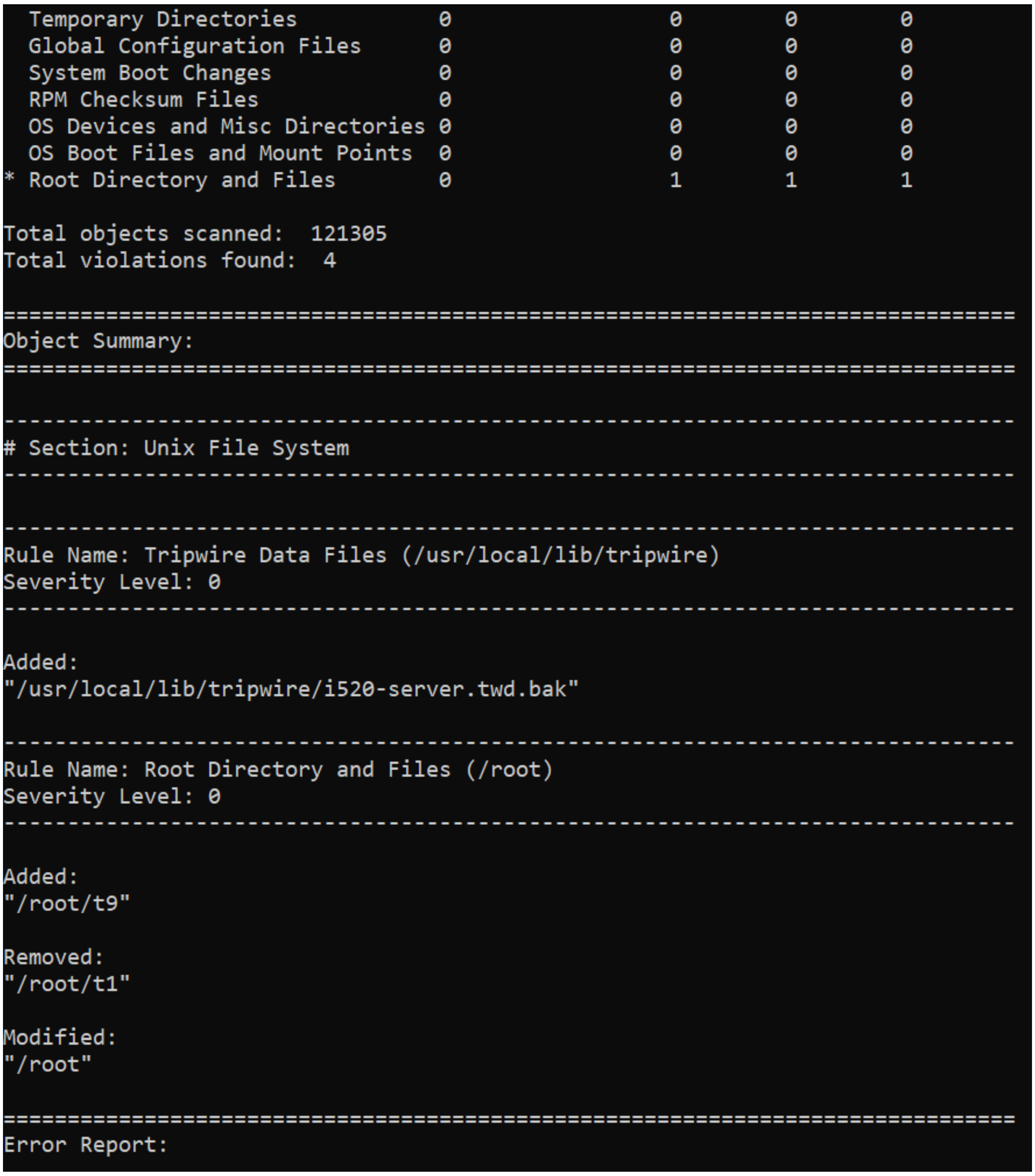


```
Found HIDDEN PID: 24743
        Cmdline: "<none>"
        Executable: "<no link>"
        "<none>  ... maybe a transitory process"

[*]Searching for Hidden processes through /proc opendir scanning

Found HIDDEN PID: 1756
        Cmdline: "<none>"
        Executable: "<no link>"
        "<none>  ... maybe a transitory process"

Found HIDDEN PID: 1799
        Cmdline: "<none>"
        Executable: "<no link>"
        "<none>  ... maybe a transitory process"

Found HIDDEN PID: 2676
        Cmdline: "<none>"
        Executable: "<no link>"
        "<none>  ... maybe a transitory process"

Found HIDDEN PID: 2898
        Cmdline: "<none>"
        Executable: "<no link>"
        "<none>  ... maybe a transitory process"

Found HIDDEN PID: 2899
        Cmdline: "<none>"
        Executable: "<no link>"
        "<none>  ... maybe a transitory process"

Found HIDDEN PID: 24743
        Cmdline: "<none>"
        Executable: "<no link>"
        "<none>  ... maybe a transitory process"

[*]Searching for Hidden thread through /proc/pid/task readdir scanning

root@i520-server:~#
```



```
root@i520-server:~# unhide.rb
Scanning for hidden processes...
ps and sysinfo() process count mismatch:
  ps: 126 processes
  sysinfo(): 132 processes
Suspicious PID 1756:
  Invisible to: ps
  Invisible to: /proc naive
       Seen by: /proc opendir
       Seen by: /proc/<pid>/status ("reptile_shell")
       Seen by: /proc readlink ("/reptile/reptile_shell")
       Seen by: /proc libc-readlink ("/reptile/reptile_shell")
       Seen by: /proc chdir
  Invisible to: getsid()
  Invisible to: getpgid()
  Invisible to: getpriority()
  Invisible to: sched_getparam()
  Invisible to: sched_getaffinity()
  Invisible to: sched_getscheduler()
  Invisible to: kill(pid, 0)
Suspicious PID 1799:
  Invisible to: ps
  Invisible to: /proc naive
       Seen by: /proc opendir
       Seen by: /proc/<pid>/status ("ls")
       Seen by: /proc chdir
  Invisible to: getsid()
  Invisible to: getpgid()
  Invisible to: getpriority()
  Invisible to: sched_getparam()
  Invisible to: sched_getaffinity()
  Invisible to: sched_getscheduler()
  Invisible to: kill(pid, 0)
Suspicious PID 2676:
  Invisible to: ps
  Invisible to: /proc naive
       Seen by: /proc opendir
       Seen by: /proc/<pid>/status ("ls")
       Seen by: /proc chdir
  Invisible to: getsid()
  Invisible to: getpgid()
```

Tripwire Screenshot:



```
  Temporary Directories        0                 0        0        0
  Global Configuration Files   0                 0        0        0
  System Boot Changes          0                 0        0        0
  RPM Checksum Files           0                 0        0        0
  OS Devices and Misc Directories 0              0        0        0
  OS Boot Files and Mount Points  0              0        0        0
* Root Directory and Files     0                 1        1        1

Total objects scanned:  121305
Total violations found:  4

===============================================================================
Object Summary:
===============================================================================

-------------------------------------------------------------------------------
# Section: Unix File System
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
Rule Name: Tripwire Data Files (/usr/local/lib/tripwire)
Severity Level: 0
-------------------------------------------------------------------------------

Added:
"/usr/local/lib/tripwire/i520-server.twd.bak"

-------------------------------------------------------------------------------
Rule Name: Root Directory and Files (/root)
Severity Level: 0
-------------------------------------------------------------------------------

Added:
"/root/t9"

Removed:
"/root/t1"

Modified:
"/root"

===============================================================================
Error Report:
```

Q2: As stated in 3) execute either command b) or c) and repeat step 4. Compare and contrast the way by which the backdoor is implemented by command a) and either b) or c).

Answer:

ICMP/UDP-Based Backdoor:

ICMP: ICMP functions as a covert channel due to its inconspicuous profile and potential firewall acceptance. Backdoors employing ICMP can discreetly communicate within seemingly innocuous ping traffic, offering a stealthy means of interaction. However, the limited payload size of ICMP messages restricts their suitability for handling small amounts of data.

UDP: UDP enables connectionless communication, making it well-suited for swift, lightweight interactions between the backdoor and a remote server. The absence of handshakes or acknowledgments minimizes communication overhead. Moreover, backdoors may exploit non-standard UDP ports for port redirection, enhancing evasion by blending with legitimate traffic.

ICMP/TCP-Based Backdoor:

ICMP: ICMP echo requests and replies (ping) can function as a covert channel, utilizing their widespread allowance through firewalls. ICMP-based backdoors communicate with reduced visibility through ping traffic, maintaining a low profile. However, akin to ICMP/UDP, ICMP messages within this context are constrained by a limited payload size.

TCP: TCP provides reliable, ordered communication, making it suitable for backdoors requiring assured delivery and ordered data transmission. The connection-oriented nature of TCP, involving a three-way handshake, can render TCP-based backdoors more easily detectable. To enhance stealth, these backdoors may utilize non-standard TCP ports for port redirection, masking traffic as legitimate.

These distinctions underscore the trade-offs between stealth and reliability, payload size considerations, and strategies for evading detection in ICMP/UDP and ICMP/TCP-based backdoors.

The distinctions observed across various ports and protocols are outlined as follows:

1. ICMP :
   Communication ports :
      From : 54058
      To : 4444
2. UDP :
   Communication ports :
      From : 53356
      To : 4445
3. TCP :
   Communication ports :
      From : 49334
      To : 4446

Section 4

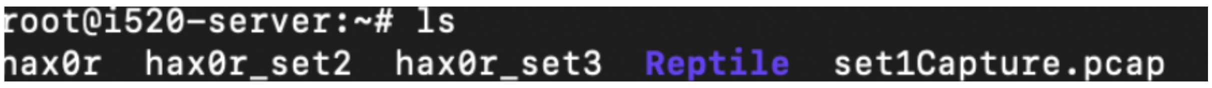
Q1: How did you determine that the rootkit is installed? Did tripwire fnd the rootkit? If so, what did tripwire tell you about it? If not, what other tools helped you to fnd the rootkit? What is the output (screenshot) of the detection?
Answer:

1. Confirming the presence of a rootkit necessitates a multifaceted approach as these malicious programs are adept at circumventing conventional security measures.
2. In this instance, although Tripwire did not explicitly identify the Reptile rootkit, alternative tools offered compelling evidence of its existence.

3. Chkrootkit flagged numerous suspicious files and registry entries associated with the Reptile rootkit, bringing attention to hidden processes and network connections that eluded standard tools.
4. Rkhunter, in its analysis, uncovered anomalies indicative of a rootkit, such as questionable file modifications, concealed directories, and unfamiliar network connections. Unhide exposed hidden files and directories generated by the Reptile rootkit, revealing malicious code and configuration data. Complementarily, Unhide.rb provided additional insights into the rootkit's operations, unveiling concealed processes, network connections, and registry alterations.
5. While Tripwire did not directly pinpoint the Reptile rootkit, it did signal alterations to the system consistent with rootkit installation, including the introduction of new files, modifications to registry entries, and the presence of concealed processes.
6. Collectively, the amalgamation of these detection tools presented compelling evidence strongly suggesting the presence of the Reptile rootkit on the system.

Q2: Where is the rootkit installed in the server VM?

Answer:



```
root@i520-server:~# ls
hax0r   hax0r_set2   hax0r_set3   Reptile   set1Capture.pcap
```

The rootkit can be found as seen in the screenshot.

Q3: If you use none of the above tools (tripwire, chkrootkit, rkhunter, unhide, and unhide.rb), how can you fnd the rootkit and/or detect its existence?

Answer:

If you're not utilizing specialized tools such as Tripwire, Chkrootkit, Rkhunter, Unhide, or Unhide.rb, you can still employ a range of techniques to manually identify the presence of a rootkit. Here are some general strategies:

1. **Examine System Logs**:Regularly scrutinize system logs for any unusual or suspicious activities. Keep an eye out for unexpected network connections, login attempts, or modifications to critical system files. Repeated errors or unexplained entries in logs could indicate malicious activity.

2. Monitor Network Traffic: Use network monitoring tools to assess network traffic for irregular patterns. Pay attention to unexpected outbound connections, particularly to unfamiliar or suspicious IP addresses. Abnormalities in network behavior may signal the existence of a rootkit.

3. Verify File Integrity: Periodically validate the integrity of critical system files by comparing them against known-good copies. Alterations to system binaries or configuration files might suggest a rootkit. Tools like `md5sum` or `sha256sum` can be employed to generate checksums.

4. Inspect Running Processes: Examine active processes on your system using tools like `ps` or `top`. Look for processes with dubious names, unusually high resource usage, or those running from unexpected locations. Investigate processes that cannot be attributed to legitimate system functions.

5. Manually Examine the File System: Inspect the file system manually for unauthorized files or directories. Be vigilant for hidden files or directories that could have been created by the rootkit. Also, be cautious of files with unconventional permissions.

6. Review System Startup Processes: Scrutinize the system's startup processes and configurations. Rootkits often aim for persistence across reboots. Look for any unauthorized entries in startup scripts or configuration files.

7. Analyze Network Ports and Services: Utilize tools like `netstat` to identify open network ports and associated services. The presence of unexpected or unknown services running on non-standard ports may indicate a rootkit.

8. Identify Behavioral Anomalies: Watch for unexpected system behavior, including performance degradation, unexplained crashes, or sudden increases in network activity. Behavioral anomalies could be indicative of a rootkit.

9. Perform Memory Analysis: Conduct memory analysis to detect anomalies. Some rootkits may reside in the system's memory and could be identified through memory analysis tools.

10. Regular System Audits: Conduct routine security audits of your system to uncover any changes or discrepancies. Regularly reviewing the overall system health and configuration can help detect subtle signs of compromise.

While these manual methods can assist in rootkit detection, it's essential to acknowledge that dedicated rootkit detection tools are specifically designed for this purpose and may offer more comprehensive and efficient results. Additionally, maintaining up-to-date system patches, implementing least privilege principles, and adhering to good cybersecurity practices contribute to overall system security.

Q4: Can you disable the backdoor of the rootkit without uninstalling the rootkit or removing it?

Answer:

1. Trying to deactivate a backdoor within a rootkit without uninstalling the entire rootkit is a complex and risky undertaking.
2. Rootkits, being sophisticated malware, are intricately designed for deep integration into systems. Disabling just one component, like a backdoor, might not eradicate the overall threat.

3. The process carries a high risk of unintended consequences, incomplete removal, and the potential for the rootkit to resist attempts at disabling.
4. Additionally, handling rootkits often necessitates specialized knowledge in malware analysis and reverse engineering.
5. The recommended approach is generally to prioritize the complete removal of the rootkit. This involves isolating the system, conducting a thorough assessment, using reputable antivirus tools, manually eliminating identified components, and, in some cases, rebuilding the system entirely.
6. Seeking assistance from cybersecurity professionals or incident response teams is advisable for a more effective and secure resolution.

Q5: How can you completely disable the rootkit?

Answer:

To disable rootkit we can use the following commands:

1. /reptile/reptile_cmd show
2. rmmod reptile_module
3. rm -rf /reptile /lib/udev/rules.d/57-reptile.rules /lib/udev/reptile