SCHOOL OF COMPUTING SCIENCES
CSE (AIML, CYS, IoT)
Academic Year 2024 -25 Even Semester

Course Code/Title: 22SBE10 / ETHICAL HACKING
Year/Sem : III/VI

LABORATORY MANUAL

LIST OF EXPERIEMENTS

S.No	List of Experiments
1	Install Kali or Backtrack Linux / Metasploitable/ Windows XP
2	Practice the basics of reconnaissance.
3	Using FOCA / SearchDiggity tools, extract metadata and expanding the target list.
4	Aggregates information from public databases using online free tools like Paterva's Maltego
5	Information gathering using tools like Robtex.
6	Scan the target using tools like Nessus.
7	View and capture network traffic using Wireshark.

1. Install Kali or Backtrack Linux / Metasploitable/ Windows XP

Aim

The aim is to set up a virtual penetration testing lab environment using VirtualBox (or VMware), which includes:

1. **Kali Linux** - A penetration testing distribution that will be used as the attacking machine.
2. **Metasploitable** - A vulnerable virtual machine intentionally designed for testing security tools and techniques.
3. **Windows XP** - A deprecated version of Windows, still used for legacy systems testing and learning.

Prerequisites

- A computer with at least 8GB of RAM (16GB recommended).
- Sufficient disk space (approximately 50-100 GB for multiple VMs).
- Virtualization software installed (e.g., VirtualBox or VMware).
- ISO images and VM files for Kali Linux, Metasploitable, and Windows XP.
- A stable internet connection to download these images.

Step-by-Step Procedure

1. Install VirtualBox/VMware

- Download and install VirtualBox or VMware Workstation from their official websites.
- Follow the installation wizard to complete the setup.

2. Install Kali Linux

- **Download Kali Linux ISO:** Visit the Kali Linux official website and download the appropriate ISO file.

Create a New Virtual Machine:

- Open VirtualBox/VMware and click on "New" to create a new VM.
- Name the VM "Kali Linux".
- Choose "Linux" as the type and "Debian (64-bit)" as the version.
- Allocate at least 2GB RAM (4GB or more recommended).
- Create a virtual hard disk (20GB or more recommended).

Install Kali Linux:

- Start the newly created VM and select the Kali Linux ISO file as the startup disk.
- Follow the installation process, setting up language, keyboard, username, and password.
- Choose "Guided - use entire disk" for partitioning.
- Complete the installation and reboot.

3. Install Metasploitable

- **Download Metasploitable:** Get the Metasploitable 2 VM from SourceForge.
- **Import Metasploitable VM:**
  1. In VirtualBox/VMware, go to "File" -> "Import Appliance".
  2. Select the Metasploitable2 .ova file and proceed with the import.

- **Network Configuration:**
  1. Set the network adapter of Metasploitable to "Host-only Adapter" or "NAT" to ensure it communicates with the other VMs but remains isolated from the internet.
- **Start the VM:** Login with the default credentials (username: msfadmin, password: msfadmin)

4. Install Windows XP

- **Download Windows XP ISO:** If you don't have the ISO file, search for an available one, ensuring it's legal to use for educational purposes.
- **Create a New Virtual Machine:**
  1. In VirtualBox/VMware, click "New" and name the VM "Windows XP".
  2. Choose "Microsoft Windows" as the type and "Windows XP" as the version.
  3. Allocate at least 1GB RAM.
  4. Create a virtual hard disk (10GB or more recommended).
- **Install Windows XP:**
  1. Start the VM, select the Windows XP ISO file as the startup disk, and proceed
  2. with the installation.
  3. Complete the installation by following the on-screen instructions.
- **Network Configuration:**
- Set the network adapter to "Host-only Adapter" or "NAT" to keep it isolated.
- **Install Guest Additions (optional):** Install VirtualBox Guest Additions/VMware Tools for better integration.

5. Network Configuration

- **Host-Only Network:** Configure all the VMs to use a Host-Only network for internal communication.
- ✓ In VirtualBox, go to File -> Host Network Manager, create a new Host-Only network if one doesn't exist.
- ✓ Set each VM to use this Host-Only Adapter in its network settings.

6. Testing the Environment

- **Start All VMs:** Power on Kali Linux, Metasploitable, and Windows XP VMs.
- **Network Connectivity:** Check the network connectivity by pinging each VM from the other.
- ✓ Open a terminal in Kali Linux and use the ping command to check if Metasploitable and Windows XP are reachable.
- **Run Initial Scans:** Use tools like nmap from Kali Linux to scan the network and discover open ports on Metasploitable and Windows XP.
  nmap -sP 192.168.56.0/24 # Replace with your network range

Output and Expected Results

- **Successful Installation:** All VMs (Kali Linux, Metasploitable, Windows XP) are up and running without errors.
- **Network Connectivity:** VMs should be able to communicate within the Host-Only network.

- **Ping Response:** Each VM should respond to ping requests from other VMs.
- S**ecurity Testing:** Kali Linux should detect open ports and services on Metasploitable and Windows XP.

RESULT

2. Practice the basics of reconnaissance.

Aim

The aim of this task is to practice the basics of reconnaissance, which is the first phase of ethical hacking or penetration testing.

Objective

Reconnaissance involves gathering information about a target system or network to identify potential vulnerabilities that can be exploited. The primary goal is to gather as much information as possible without alerting the target.

Types of Reconnaissance

1. **Passive Reconnaissance:** Gathering information without directly interacting with the target. This might include using publicly available information and tools to find data about the target.
2. **Active Reconnaissance:** Involves directly interacting with the target system, which might be detectable by the target. This includes using tools to scan ports, identify services, etc.

Tools Required

1. **Kali Linux**: The main operating system used for penetration testing.
2. **Nmap**: A network scanning tool used to discover hosts and services on a network.
3. **WHOIS**: A query tool to get domain information.
4. **Nslookup / Dig**: Tools for DNS lookups.
5. **TheHarvester**: A tool for gathering emails, subdomains, hosts, employee names, open ports, and banners from different public sources.

Step-by-Step Procedure

Step 1: Passive Reconnaissance

- **Objective:** Gather information without directly interacting with the target.
- **Tools Used:** WHOIS, Google Dorks, Nslookup, and TheHarvester.
- **WHOIS Lookup:**

o Use the whois command to get domain registration details, including the owner, contact information, and registration dates.

whois example.com

This command will provide the domain's registrant information, creation and expiration dates, and nameservers.

 Google Dorks:

o Use specific search queries on Google to find sensitive information related to the target.

o Example queries:

site:example.com filetype:pdf
site:example.com intitle:"index of"

o This method can help find publicly exposed sensitive files or directories.

- **DNS Information Gathering (Nslookup and Dig):**

o **Nslookup:**

 Perform a DNS lookup to get the IP address of the domain.

nslookup example.com

- **Dig (Domain Information Groper):**

 Use dig for more detailed DNS information, including records like MX (Mail Exchanger), NS (Name Server), and A records.

dig example.com
dig example.com MX
dig example.com NS

TheHarvester:
o Use TheHarvester to gather emails, subdomains, hosts, employee names, and other data using public sources like Google, LinkedIn, etc.

theharvester -d example.com -l 500 -b google

Step 2: Active Reconnaissance
 Objective: Actively interact with the target to gather more detailed information. This could alert the target to your activities.
 Tools Used: Nmap
 Network Scanning with Nmap:
o **Ping Sweep:** To find live hosts on a network.

nmap -sn 192.168.1.0/24

o **Port Scanning:** To find open ports on a specific target IP.
nmap -p- 192.168.1.10 # Scans all 65535 ports
o **Service and Version Detection:** To identify services and their versions running on open ports.
nmap -sV 192.168.1.10

o **Operating System Detection:**

nmap -O 192.168.1.10

Step 3: Analyze and Document Findings
 Create a Reconnaissance Report:
o Document all the findings from the passive and active reconnaissance steps.
o Include information such as domain registration details, discovered IP addresses, DNS records, identified services and versions, open ports, etc.
 Identify Potential Vulnerabilities:
o Based on the information gathered, list possible vulnerabilities that can be exploited, such as outdated services, unpatched software, open ports, exposed sensitive files, etc.

Expected Output and Results
1. **WHOIS Lookup Output:**
o Information about the domain owner, registrar, contact information, registration and expiration dates, and nameservers.
2. **Google Dorks:**
o List of sensitive files, directories, or information found using specific search queries.
3. **DNS Information:**
o List of DNS records, including A, MX, and NS records, and corresponding IP addresses of the target domain.
4. **TheHarvester Results:**
o Emails, subdomains, hosts, employee names, open ports, and other relevant information.

5. **Nmap Scans:**
o List of live hosts within the target network.
o Open ports on specific hosts.
o Services running on these ports along with their versions.
o Operating system detection results.

RESULT

3. Using FOCA / SearchDiggity tools, extract metadata and expanding the target list

Aim

The aim of this task is to use tools like **FOCA** (Fingerprinting Organizations with Collected Archives) and **SearchDiggity** to extract metadata from documents available online.

Objective

Metadata extraction helps gather information about a target organization by analyzing publicly available files such as PDFs, Word documents, and images. This information can be used to expand the target list by identifying additional domains, subdomains, IP addresses, usernames, email addresses, software versions, and other potentially sensitive information.

Tools Required

1. **FOCA**: A Windows-based tool used for metadata extraction and information gathering. It is effective in finding metadata and hidden information in documents.
2. **SearchDiggity**: A suite of tools developed by Stach & Liu that leverages search engines to identify vulnerabilities and extract metadata from documents.

Step-by-Step Procedure

Using FOCA to Extract Metadata

Step 1: Setup FOCA

1. **Download and Install FOCA**:
o FOCA can be downloaded from its official site or repositories.
o It requires a Windows environment to run.
o Install FOCA by following the installation instructions provided on the download site.

Step 2: Configure FOCA Project

1. **Create a New Project**:
o Launch FOCA and create a new project.
o Enter a project name, and specify the target domain (e.g., example.com).
o Choose a location to save the project.
2. **Add Domains**:
o In FOCA, you can specify additional domains or subdomains if known.
o This helps expand the scope of the metadata extraction.

Step 3: Search and Download Public Documents

1. **Search Engines**:
o FOCA can search for documents using Google, Bing, or other search engines.
o Use the "Search All" feature to let FOCA automatically search for documents related to the target domain.
o Alternatively, specify search filters manually (e.g., search for filetype:pdf site:example.com).
2. **Download Documents**:
o FOCA will list documents found on the internet related to the target domain.
o Download these documents automatically through FOCA for metadata extraction.

Step 4: Extract Metadata

1. **Analyze the Documents**:
o FOCA extracts metadata such as author names, usernames, email addresses, software versions used, network paths, printer information, etc.
o FOCA can also analyze document properties like creation date, last modification, and GPS coordinates (if available).

2. **View Metadata Information**:
o FOCA will display extracted metadata in a structured format.
o Identify sensitive information, including potential usernames, email addresses, document creation paths, software versions, etc.
3. **Expand Target List**:
o Use extracted metadata to find additional domains, subdomains, IP addresses, or potential entry points for further reconnaissance.

Using SearchDiggity to Extract Metadata

Step 1: Setup SearchDiggity

1. **Download and Install SearchDiggity**:
o Download SearchDiggity from its official repository or website.
o It requires a Windows environment with .NET framework installed.
o Follow the installation instructions to set up SearchDiggity.

Step 2: Configure SearchDiggity

1. **Choose Search Options**:
o Launch SearchDiggity and select the type of search you want to perform (e.g., Google Hacking Diggity, Bing Hacking Diggity).
o Select the relevant search queries for metadata extraction.
2. **Enter Target Information**:
o Specify the domain or keyword related to the target organization.
o Choose file types to focus on (e.g., PDFs, Word documents, Excel files).

Step 3: Perform Searches

1. **Execute Searches**:
o Use predefined queries or create custom queries to search for documents containing metadata.
o SearchDiggity uses search engines to find files that match the criteria.
2. **Download and Analyze Files**:
o Review the search results.
o Download files manually or automate the process (depending on the tool configuration).
o Analyze files to extract metadata information similar to FOCA.

Step 4: Analyze Metadata and Expand Target List

1. **Extract Metadata**:
o Analyze the downloaded documents using SearchDiggity's tools or external metadata analysis tools.
o Extract useful information such as usernames, email addresses, internal IP addresses, and software versions.
2. **Expand Target List**:
o Utilize the extracted metadata to identify new targets, such as additional domains or potential user accounts.
o Cross-reference this information with other reconnaissance data.

Sample Program to Automate Metadata Extraction (Python Script Using ExifTool)

To complement the use of FOCA and SearchDiggity, a simple Python script using ExifTool can be employed to automate the extraction of metadata from downloaded documents:

```
import os
import subprocess
    def extract_metadata(file_path):
```

```
    try:
    result = subprocess.run(['exiftool', file_path],
    capture_output=True, text=True)
    metadata = result.stdout
    return metadata
    except Exception as e:
    return f"Error extracting metadata: {e}"
    def analyze_directory(directory):
    for root, dirs, files in os.walk(directory):
    for file in files:
    file_path = os.path.join(root, file)
    print(f"Extracting metadata from {file_path}")
    metadata = extract_metadata(file_path)
    print(metadata)
    print("-" * 40)

# Directory containing downloaded documents
    directory = 'path_to_downloaded_documents'
    analyze_directory(directory)
```

Expected Output and Results

1. **FOCA and SearchDiggity Metadata Extraction:**

o **Usernames and Email Addresses**: FOCA and SearchDiggity can extract author names and emails embedded in document metadata.

o **Software Information**: Information about the software used to create or modify the documents, which can be useful for vulnerability identification.

o **Document Paths and Network Shares**: FOCA can reveal paths used to save or print the documents, showing network configurations.

o **Subdomains and IP Addresses**: Analysis can reveal hidden or lesser-known subdomains associated with the target organization.

o **Physical Location (GPS)**: If available, geographic coordinates may be extracted from certain types of media files.

2.Python Script (ExifTool) Output:

The script will output metadata for each file found in the specified directory. It will display extracted details such as document creation date, modification date, author, and other metadata tags.

o *Example output:*

Extracting metadata from
path_to_downloaded_documents/sample.pdf
Author: John Doe
Creation Date: 2024-01-15T10:45:00
Software: Microsoft Word 2016
Document Path: \\companyserver\shared\documents\sample.pdf

RESULT

4. Aggregates information from public databases using online free tools like Paterva's Maltego.

Aim

The aim of this task is to use **Maltego**, a popular open-source intelligence (OSINT) and forensics tool developed by Paterva, to aggregate information from public databases.

Objective

Maltego helps visualize relationships between data sets such as domains, IP addresses, email addresses, organizations, and individuals by utilizing various online free tools and services. This process aids in identifying potential threats, vulnerabilities, and connections that might not be apparent otherwise.

Tools Required

1. **Maltego**: A powerful tool for data mining and link analysis. Maltego is available in different versions, including a free community edition (Maltego CE).
2. **Transforms**: Pre-built queries and scripts that allow Maltego to interact with various online databases and APIs. Examples include WHOIS lookups, DNS queries, social media searches, etc.

Step-by-Step Procedure

Step 1: Install Maltego

1. **Download Maltego**:

o Visit the Maltego official website and download the community edition (Maltego CE) or purchase the professional edition if needed.

o Maltego is available for Windows, Linux, and macOS.

2. **Install Maltego**:

o Run the installer and follow the installation steps for your operating system.

o Register for a free Maltego Community Edition account if you do not have one.

Step 2: Set Up Maltego

1. **Launch Maltego**:

o Open the Maltego application and log in with your Maltego CE account credentials.

o Choose the **Maltego CE** (Community Edition) option if you are using the free version.

2. **Set Up API Keys (Optional)**:

o Some transforms may require API keys for enhanced functionality. For basic tasks, Maltego's default transforms (open-source and free) will suffice.

Step 3: Create a New Maltego Graph

1. **Start a New Graph**:

o Click on the "New" button to create a new graph. This graph will be used to visualize relationships between different entities.

2. **Add a Seed Entity**:

o Choose an initial entity to start the investigation. This could be a **domain name**, **email address**, **IP address**, **person's name**, or any other type of entity.

o For example, drag a "Domain" entity from the left-side entity palette onto the graph canvas and set its value to example.com.

Step 4: Running Transforms

1. **Select the Entity**:

o Click on the entity you added to select it (e.g., the domain example.com).

2. **Run Transforms**:

o Right-click on the entity and choose transforms to run. Maltego provides a variety of transforms that query different data sources. Some useful transforms include:

 DNS from domain: Retrieves DNS records (A, MX, NS, etc.).

 WHOIS lookup: Retrieves WHOIS registration details.
 Find email addresses: Finds email addresses associated with the domain.
 Social Network Analysis: Identifies social media profiles related to the entity.
o Select the transform or set of transforms you wish to run. Maltego will query the corresponding data sources and display the results.
3. **Expand Results**:
o Maltego will show the results as new entities linked to the original entity. For example, running a WHOIS lookup may show the registrant's name, contact email, and phone number.
o Click on these new entities and run further transforms to dig deeper into the data.

Step 5: Analyzing and Visualizing Data

1. **Visualize Connections**:
o Maltego displays data in a graph format, making it easy to see relationships and connections between different entities. Use different layout options like **organic layout**, **hierarchical layout**, and **circular layout** to better visualize the connections.
2. **Filter and Focus**:
o Use Maltego's filtering tools to focus on specific types of entities or to hide unnecessary information. This helps to simplify complex graphs and highlight critical data.
3. **Generate Reports**:
o Maltego allows exporting the graph and its data into reports. Use the export function to generate detailed reports for documentation or further analysis.

Step 6: Document Findings

1. **Save Graphs**:
o Save your Maltego graphs regularly to preserve the data and visualizations.
2. **Create a Report**:
o Use the findings from Maltego to create a structured report. Include information such as identified email addresses, associated IP addresses, domain registration details, and any discovered vulnerabilities or threats.

Sample Workflow Using Maltego

1. **Start Maltego and Create a New Graph**.
2. **Add a Domain Entity** (example.com) to the graph.
3. **Run the Following Transforms** on the Domain Entity:
o **To DNS Name – MX, NS, and A Records**: Extract mail servers, name servers, and IP addresses associated with the domain.
o **To WHOIS Information**: Extract domain registration details, registrant information, and contact details.
o **To Email Address**: Find email addresses related to the domain.
o **To Social Network Information**: Identify any social media profiles or accounts linked to the domain.

Expected Output and Results

1. **DNS Records**:
o Maltego extracts MX (mail exchange), NS (name server), and A records (address records) associated with example.com.
o **Example Output**:
 MX Record: mail.example.com
 NS Record: ns1.example.com, ns2.example.com
 A Record: 192.168.1.1
2. **WHOIS Information**:
o Extracts details such as the registrant's name, organization, contact email, phone number, and registration dates.

o **Example Output**:
 Registrant Name: John Doe
 Organization: Example Corp
 Contact Email: johndoe@example.com
3. **Email Addresses**:
o Maltego finds email addresses that are publicly associated with the domain.
o **Example Output**:
 support@example.com
 admin@example.com
4. **Social Network Information**:
o Identification of social media profiles and user accounts linked to the domain or organization.
o **Example Output**:
 Twitter: @examplecorp
 LinkedIn: Example Corp
5. **Relationships and Connections**:
o The graph visualizes how different entities (email addresses, domains, IPs, etc.) are related, providing a clear picture of the target's digital footprint.

RESULT

5. Information gathering using tools like Robtex.

Aim

The aim of this task is to use **Robtex**, a popular online tool for network and domain information gathering, to extract valuable information about a target.

Objective

Robtex aggregates data from multiple public sources to provide insights into domains, IP addresses, DNS records, subdomains, related domains, routing information, and other network-related details. Using Robtex effectively can help cybersecurity professionals understand the digital footprint of a target and identify potential vulnerabilities or connections that might not be immediately visible.

Tools Required

1. **Robtex**: An online database and tool that provides comprehensive information on domains, IP addresses, and network infrastructure. It can be accessed via its web interface at robtex.com.

Step-by-Step Procedure

Step 1: Access Robtex

1. **Open a Web Browser**:

o Go to the Robtex website: https://www.robtex.com.

2. **Understanding the Interface**:

o Familiarize yourself with the search bar and the different sections of the website, which display information about domains, IP addresses, AS (Autonomous Systems), and routes.

Step 2: Gather Domain Information

1. **Enter a Target Domain**:

o In the search bar, enter a target domain (e.g., example.com) and press Enter.

2. **Review Domain Overview**:

o Robtex provides an overview that includes the domain's DNS records, associated IP addresses, WHOIS information, and related domains.

3. **Analyze DNS Information**:

o **DNS Records**: Look at the A (address), MX (mail exchange), NS (name server), and TXT records to understand the domain's infrastructure.

o **Example Output**:

 A Record: 93.184.216.34 (IP address associated with the domain)

 MX Record: mail.example.com (Mail server for the domain)

 NS Record: ns1.example.com, ns2.example.com (Name servers)

4. **Find Related Domains**:

o Robtex displays domains that share the same IP addresses, name servers, or other infrastructure. This can help identify additional targets or related domains.

o **Example Output**:

 Related Domains: example.net, example.org (Domains using the same name server or IP address)

5. **WHOIS Information**:

o Review the WHOIS data provided by Robtex for registration details, including registrant name, organization, email address, and registration dates.

o **Example Output**:

- *Registrant Name: John Doe*
- *Organization: Example Corp*
- *Email*: *johndoe@example.com*

Step 3: IP Address Information

1. **Enter a Target IP Address**:

o Search for a specific IP address (e.g., 93.184.216.34) using the search bar.

2. **Review IP Overview**:
   o Robtex provides information about the IP, including which domains are hosted on it, geographical location, and associated AS (Autonomous System) information.
   o **Example Output**:
   IP Location: United States
   Hosted Domains: example.com, example.net
   Autonomous System: AS13335 Cloudflare, Inc.
3. **Routing Information**:
   o Robtex shows routing paths and BGP (Border Gateway Protocol) information for the IP, which can be useful for understanding how traffic is routed to and from the IP address.

Step 4: Autonomous System (AS) Information

1. **Enter an AS Number**:
o If you have an AS number *(e.g., AS13335)*, enter it in the search bar to get details about the autonomous system.
2. **Review AS Overview**:
Robtex provides details about the AS, including the IP ranges it covers, the organization managing it, and its peering relationships.
o **Example Output**:
AS Name: Cloudflare, Inc.
IP Ranges: 93.184.216.0 - 93.184.216.255
Peers: Information about other AS numbers that have routing connections with the AS.

Step 5: Visualize and Analyze Data

1. **Graphical View**:
o Robtex offers graphical visualizations that show how domains, IPs, AS numbers, and other entities are connected. Use these visualizations to see relationships and identify potential points of interest.
2. **Document Findings**:
o Take screenshots or notes of key findings. Document relationships between domains, IP addresses, and AS numbers, as well as any anomalies or unexpected connections.

Step 6: Additional Analysis

1. **Cross-reference Data**:
o Use the information from Robtex in conjunction with other OSINT tools and databases (e.g., WHOIS lookups, Shodan, VirusTotal) to gain a comprehensive view of the target.
2. **Expand Investigation**:
o Use related domains, IP addresses, or AS numbers found in Robtex to expand your investigation and uncover more information about the target's infrastructure and digital footprint.

Sample Python Script for Automating IP and Domain Lookup with RobtexAPI

To automate some of the lookup tasks, you can use the Robtex API with Python. Here is a simple example of how you could use Python to perform domain and IP lookups:

python
Copy code

```
import requests
def get_domain_info(domain):
url = f"https://freeapi.robtex.com/domain/{domain}"
response = requests.get(url)
```

```
return response.json()
def get_ip_info(ip):
url = f"https://freeapi.robtex.com/ipquery/{ip}"
response = requests.get(url)
return response.json()
# Example usage
domain = "example.com"
ip = "93.184.216.34"
domain_info = get_domain_info(domain)
ip_info = get_ip_info(ip)
print("Domain Information:")
print(domain_info)
print("\nIP Information:")
print(ip_info)
```

Expected Output and Results

1. **Domain Information Output**:

o The output from the script will display information about the domain, including its DNS records, related domains, and possibly WHOIS details.

o **Example Output**:

```
json
Copy code
{
"domain": "example.com",
"a_records": ["93.184.216.34"],
"mx_records": ["mail.example.com"],
"ns_records": ["ns1.example.com", "ns2.example.com"],
"related_domains": ["example.net", "example.org"]
}
```

2. **IP Information Output**:

o The script will display details about the IP address, including its location, associated domains, and AS information.

o **Example Output**:

```
json
Copy code
{
"ip": "93.184.216.34",
"location": "United States",
"domains": ["example.com", "example.net"],
"as": "AS13335 Cloudflare, Inc."
}
```

RESULT

6. Scan the target using tools like Nessus.

Aim

The aim of this task is to perform a vulnerability scan on a target system or network using **Nessus**, a widely-used vulnerability scanner.

Objective

Nessus helps identify potential vulnerabilities, misconfigurations, and security weaknesses in operating systems, applications, and network devices. By using Nessus, cybersecurity professionals can proactively detect and address security issues before they can be exploited by attackers.

Tools Required

1. **Nessus**: A popular vulnerability assessment tool developed by Tenable Inc. It is available in several editions, including a free version (Nessus Essentials) and commercial versions (Nessus Professional, Tenable.io).
2. **Target System or Network**: The IP address or hostname of the system or network to be scanned. Ensure that you have authorization to scan the target to avoid legal and ethical issues.

Step-by-Step Procedure

Step 1: Install Nessus

1. **Download Nessus**:

o Visit the official Nessus website: Tenable Nessus and download the appropriate version for your operating system (Windows, Linux, or macOS).

2. **Install Nessus**:

o Follow the installation instructions provided by Tenable. Installation typically involves running an installer package and starting the Nessus service.

3. **Register Nessus**:

o For Nessus Essentials (free version), register on the Tenable website to receive an activation code. Enter the activation code in the Nessus interface to activate the product.

4. **Access Nessus**:

o Once installed, Nessus can be accessed through a web browser by navigating to https://localhost:8834 (or replacing localhost with the server's IP address if installed remotely).

Step 2: Configure Nessus

1. **Login to Nessus**:

o Use your credentials to log in to the Nessus web interface.

2. **Update Plugins**:

o Before starting a scan, ensure Nessus has the latest vulnerability definitions by updating its plugins. This can be done through the **Settings** menu in the Nessus interface.

Step 3: Create a New Scan

1. **Start a New Scan**:

o Click on the **"Scans"** tab, and then click the **"New Scan"** button.

2. **Choose a Scan Template**:

o Select an appropriate scan template based on the objective:

 Basic Network Scan: General purpose scan for detecting vulnerabilities in network services.

 Advanced Scan: Customize scan settings for in-depth analysis.

 Web Application Tests: Scan web applications for vulnerabilities.

 Credentialed Scans: Perform authenticated scans using login credentials for more accurate results.

o For this example, choose **"Basic Network Scan"**.

3. **Configure Scan Settings**:
o **Name**: Provide a name for the scan (e.g., "Target Network Scan").
o **Targets**: Enter the IP address or hostname of the target system(s) to scan (e.g., 192.168.1.100 or example.com).
o **Schedule**: (Optional) Configure scan scheduling options if you want the scan to run at specific times.
o **Advanced Settings**: (Optional) Configure additional settings like port ranges, scan timeouts, or performance tuning if needed.
4. **Save and Launch the Scan**:
o Click on **"Save"** to create the scan.
o Click on the **"Launch"** button to start the scan immediately

Step 4: Monitor the Scan Progress
1. **View Scan Progress**:
o Once the scan starts, Nessus will display the progress in real time. You can monitor the number of hosts being scanned, vulnerabilities detected, and other details.
2. **Wait for Scan Completion**:
o Depending on the size of the network and the number of targets, the scan may take some time to complete. Nessus will notify you when the scan is finished.
Step 5: Review Scan Results
1. **Access Scan Results**:
o Once the scan is complete, click on the scan name to view the results.
2. **Analyze Vulnerabilities**:
o Nessus will display a list of detected vulnerabilities, categorized by severity (e.g., Critical, High, Medium, Low, Info).
o Each vulnerability entry provides details such as the vulnerability name, affected hosts, description, CVE identifiers, and recommended remediation steps.

3. **Example Output**:

Vulnerability Name	Severity	Affected Host	CVE ID	Description
OpenSSL Heartbleed Vulnerability	High	192.168.1.100	CVE-2014-0160	This vulnerability allows attackers to steal information Protected by SSL/TLS.
SMBv1 Protocol Detection	Medium	192.168.1.101	CVE-2017-0144	This protocol is outdated and susceptible to ransomware attacks
Outdated Apache Server	Medium	192.168.1.102	CVE-2021-40438	The server is running a version with known vulnerabilities.

4. **View Detailed Reports**:
Nessus provides detailed reports that include information about each detected vulnerability, its impact, and how to remediate it. Reports can be exported in various formats such as PDF, CSV, or HTML.

Step 6: Remediation and Follow-Up

1. **Remediation**:

o Based on the scan results, take appropriate actions to fix the identified vulnerabilities. This might involve patching software, reconfiguring systems, or disabling vulnerable services.

2. **Rescan**:

o After remediation, perform a rescan to ensure that the vulnerabilities have been effectively mitigated.

3. **Documentation**:

o Document the findings, remediation steps taken, and any other relevant information for future reference and compliance.

Sample Script to Automate Nessus Scan via API

If you have access to the Nessus API, you can automate scans using Python. Here's an example script to perform a basic scan using the Nessus API:

```
import requests
import json

# Nessus server details
nessus_url = "https://localhost:8834"
username = "your_username"
password = "your_password"

# Disable SSL warnings (Not recommended for production)
requests.packages.urllib3.disable_warnings()

# Function to authenticate and get a session token
def get_nessus_token():
    url = f"{nessus_url}/session"
    payload = {"username": username, "password": password}
    headers = {"Content-Type": "application/json"}

    response = requests.post(url, data=json.dumps(payload),
headers=headers, verify=False)
    response.raise_for_status()
    return response.json()["token"]
# Function to create a scan
def create_scan(token, scan_name, target):
    url = f"{nessus_url}/scans"
    headers = {"X-Cookie": f"token={token}", "Content-Type":
"application/json"}
    payload = {
        "uuid": "7310e82a-e216-4eb1-94b7-3ba8e9df0453", # Basic Network
Scan template UUID
    "settings": {
        "name": scan_name,
        "text_targets": target,
        "description": "Automated scan using API",
        "enabled": True
    }
}
    response = requests.post(url, data=json.dumps(payload),
```

```
headers=headers, verify=False)
        response.raise_for_status()
        return response.json()
# Example usage
try:
        # Get the session token
        token = get_nessus_token()

        # Create a new scan
        scan_name = "Automated Network Scan"
        target = "192.168.1.100"
        scan_response = create_scan(token, scan_name, target)
        print(f"Scan '{scan_name}' created successfully: {scan_response}")

except requests.exceptions.HTTPError as err:
        print(f"HTTP error occurred: {err}")
except Exception as e:
        print(f"An error occurred: {e}")
```

Expected Output and Results

1. **Scan Status**: The script will create a new scan in Nessus using the API and return the scan details upon successful creation.

Example Output:

```
json
Copy code
{
        "scan": {
        "id": 1234,
        "name": "Automated Network Scan",
        "status": "pending",
        "targets": "192.168.1.100",
        "uuid": "7310e82a-e216-4eb1-94b7-3ba8e9df0453"
        }
}
```

2. **Scan Results**:

Once the scan completes, the Nessus interface will display the list of vulnerabilities detected, categorized by severity. You can view detailed information about each vulnerability, including its impact and remediation steps.

RESULT

6. View and capture network traffic using Wireshark.

Aim

The aim of this task is to use **Wireshark**, a widely-used network protocol analyzer, to capture and analyze network traffic.

Objective

Wireshark helps cybersecurity professionals, network administrators, and IT specialists monitor, troubleshoot, and secure networks by providing visibility into data being transmitted over a network. By capturing and analyzing packets, you can identify potential security issues, performance bottlenecks, or unauthorized activities.

Tools Required

1. **Wireshark**: A powerful open-source network protocol analyzer available for Windows, macOS, and Linux. Wireshark allows you to capture and interactively browse the traffic running on a computer network.
2. **Target Network**: A network interface on your computer or a specific network that you want to monitor. Make sure you have the necessary permissions to capture traffic on the target network.

Step-by-Step Procedure

Step 1: Install Wireshark

1. **Download Wireshark**:
   - o Visit the official Wireshark website: Wireshark Download and download the installer for your operating system (Windows, macOS, or Linux).
2. **Install Wireshark**:
   - o Follow the installation instructions provided by Wireshark. On Windows, you may need to install additional components like WinPcap or Npcap for packet capturing.
3. **Run Wireshark**:
   - o Launch Wireshark after installation. It will automatically detect available network interfaces on your system.

Step 2: Capture Network Traffic

1. **Select a Network Interface**:
   - o When you start Wireshark, you'll see a list of network interfaces. Choose the interface you want to monitor (e.g., Ethernet, Wi-Fi, or Loopback).
   - o Double-click on the interface to start capturing traffic.
2. **Start Capturing Traffic**:
   - o Once you select an interface, Wireshark will start capturing packets in realtime. You will see packets scrolling down in the main window as they are captured.
3. **Filter Traffic (Optional)**:
   - o Use Wireshark's powerful filtering capabilities to narrow down the traffic you are interested in. For example:
     - To capture only HTTP traffic: http
     - To capture traffic to a specific IP address: *ip.addr == 192.168.1.10*
     - To capture traffic on a specific port: *tcp.port == 80*
     - To capture only DNS queries: *dns*
4. **Stop Capture**:
   - o Click the red **Stop** button in the toolbar to stop capturing traffic when you have collected enough data.

Step 3: Analyze Captured Traffic

1. **Viewing Packets**:

o After stopping the capture, Wireshark will display the captured packets. The interface is divided into three main panes:

- **Packet List Pane**: Displays a summary of each packet captured.
- **Packet Details Pane**: Shows detailed protocol-level information about the selected packet.
- **Packet Bytes Pane**: Displays the raw data of the selected packet in hexadecimal and ASCII format.

2. **Follow a Stream**:

o To view the entire conversation between two endpoints, right-click on a packet and select **"Follow TCP Stream"** or **"Follow UDP Stream"**. This displays the complete session in a readable format, useful for analyzing the content of communications.

3. **Applying Filters**:

o Use the filter bar to apply various filters to isolate specific types of traffic or analyze particular protocols. Examples of useful filters:

- *tcp: Displays only TCP traffic.*
- *udp: Displays only UDP traffic.*
- *http.request: Displays HTTP request packets.*
- *ip.src == 192.168.1.10: Displays traffic from a specific source IP Address*

4. **Analyzing Traffic**:

o Identify abnormal patterns, such as unusual IP addresses, high traffic volumes, or frequent connection attempts, which might indicate scanning activities or attacks.

o Check for unencrypted sensitive data transmitted over the network, such as passwords or personal information.

Step 4: Save and Export Data

1. **Save Capture File**:

o To save captured traffic for later analysis, go to **File > Save As** and choose a filename and location. Wireshark saves the file with a .pcap extension, which can be opened in Wireshark or other packet analysis tools.

2. **Export Specific Packets**:

o You can export specific packets by selecting them and going to **File > Export Specified Packets**. This allows you to share only the relevant parts of the capture.

Step 5: Document Findings

1. **Take Notes**:

o Document your observations, such as notable IP addresses, types of traffic, and any anomalies or suspicious activities identified.

2. **Generate Reports**:

o Wireshark provides options to generate statistics and reports that summarize the captured data. Go to **Statistics** in the menu to access different types of analysis (e.g., protocol hierarchy, endpoint statistics, etc.).

Example Output and Results

1. **Basic Capture Output**:

o After capturing traffic, Wireshark displays packets in real-time. An example output in the Packet List Pane might look like this:

No	Time	Source	Destination	Protocol	Length	Info
1.	0.00001	192.168.1.5	192.168.1.1	ARP	42	Who has 192.168.1.1? Tell 192.168.1.5
2.	0.00253	192.168.1.1	192.168.1.5	ARP	60	192.168.1.1 is at 00:1a:2b:3c:4d:5e
3.	0.01562	192.168.1.5	93.184.216.34	TCP	74	49546 → 80 [SYN] Seq=0 Win=65535 Len=0
4.	0.01876	93.184.216.34	192.168.1.5	TCP	66	80 → 49546 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0
5.	0.01888	192.168.1.5	93.184.216.34	TCP	60	49546 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
6.	0.02032	192.168.1.5	93.184.216.34	HTTP	234	GET / HTTP/1.1

2. **Analyzing HTTP Traffic**:
o You can filter for HTTP requests by entering http in the filter bar. This will show only HTTP traffic, useful for analyzing web interactions.
3. **Following a TCP Stream**:
o By selecting a packet and choosing **"Follow TCP Stream"**, you can see the complete conversation between the client and server. For instance, the following HTTP request might be shown:

GET / HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,/*;q=0.8*

4. **Analyzing DNS Queries**:
o Use the filter *dns* to isolate DNS traffic. This will show all DNS queries and responses, which is useful for identifying domain resolution and potential suspicious activity.

Sample Python Script to Automate Traffic Capture Using Tshark
While Wireshark provides a GUI for capturing and analyzing traffic, you can also use **Tshark**, the command-line version of Wireshark, to automate traffic capture. Below is a sample Python script that uses subprocess to run a Tshark command and capture network traffic:

```
import subprocess
# Define the interface and output file
interface = "eth0"
output_file = "network_capture.pcap"

# Tshark command to capture packets
tshark_command = ["tshark", "-i", interface, "-w", output_file]
```

```
try:
        # Start the Tshark process
        print(f"Starting capture on interface {interface}...")
        process = subprocess.Popen(tshark_command)

        # Capture traffic for a specific duration (e.g., 60 seconds)
        capture_duration = 60
        process.wait(timeout=capture_duration)

except subprocess.TimeoutExpired:
        # Stop the capture after the specified duration
        process.terminate()
        print(f"Capture stopped. Data saved to {output_file}")

except Exception as e:
        print(f"An error occurred: {e}")

# Analyze the captured file using Wireshark or Tshark
```

Expected Output

1. **Captured Packets File**:

o The script will create a .pcap file named network_capture.pcap containing the captured network traffic. This file can be opened with Wireshark for detailed analysis.

2. Analyzing the Capture

RESULT