ICS-1-02P: Introduction to Cyber Security Lab

Total Marks: 50
External Marks: 35
Internal Marks: 15
Credits: 2
Pass Percentage: 40%

1. Identify Open Ports, Services, and Potential Vulnerabilities on Target Systems

- **Tool:** `Nmap`, `Nessus`, `OpenVAS`
- **Basic Nmap Commands:**

```
nmap -sS 192.168.1.1          # TCP SYN scan
nmap -sV 192.168.1.1          # Service version detection
nmap -O 192.168.1.1           # OS detection
nmap -A 192.168.1.1           # Aggressive scan: OS, version, script
```

- **Vulnerability Detection (Nmap NSE):**

```
nmap --script vuln 192.168.1.1
```

2. Scan and Enumerate Devices on a Network Using Nmap

- **Network Discovery:**

```
nmap -sn 192.168.1.0/24       # Ping scan
```

- **Enumeration Examples:**

```
nmap -sV -T4 -O -F 192.168.1.0/24    # Fast scan with version and OS
detection
```

- **Save Output:**

```
nmap -oN output.txt 192.168.1.0/24
```

3. Analyze Malware in a Controlled Environment

- **Environment:** Use a virtual machine (VirtualBox/VMware), snapshot enabled.
- **Tools:** Cuckoo Sandbox, Wireshark, PEStudio, IDA Free
- **Process:**
    1. Disable internet access (simulate with host-only).
    2. Run malware in controlled VM.
    3. Use tools like `ProcMon`, `Wireshark`, `Autoruns` to monitor behavior.

⚠ **Warning**: Always isolate the environment and never run malware on your host machine.

4. Conduct a Phishing Simulation

- **Tool:** `GoPhish`, `SocialFish`
- **Steps:**
    1. Create a phishing email template.
    2. Host a fake login page (e.g., cloned Facebook/Gmail using SocialFish).
    3. Send simulated phishing emails.
    4. Monitor who clicks (for awareness/training).

Ensure compliance with **ethical/legal guidelines** – only test within your organization or lab.

5. Configure a Firewall to Control Network Traffic

- **Tool:** `iptables`, `ufw`, or Windows Defender Firewall
- **Basic UFW Commands (Linux):**

```
sudo ufw enable
sudo ufw allow 80/tcp
sudo ufw deny 23/tcp
```

- **Windows:**
    - Use `wf.msc` → Create Inbound/Outbound rules.

6. Design and Implement Firewall Rules

- **Example: Block all, allow only HTTP and SSH**

```
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow 22/tcp
sudo ufw allow 80/tcp
```

- **Iptables Example:**

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -j DROP
```

7. Secure Communication Using OpenSSL or GPG

- **GPG:**

```
gpg --gen-key                       # Generate key
gpg -e -r receiver@example.com file.txt   # Encrypt
```

```
gpg -d file.txt.gpg                # Decrypt
```

- **OpenSSL:**

```
openssl genrsa -out private.pem 2048
openssl rsa -in private.pem -pubout -out public.pem
openssl rsautl -encrypt -inkey public.pem -pubin -in msg.txt -out msg.enc
openssl rsautl -decrypt -inkey private.pem -in msg.enc
```

8. Simulate Password Cracking Techniques

- **Tools:** `John the Ripper`, `Hashcat`, `Hydra`
- **Crack with John:**

```
john --wordlist=rockyou.txt --format=raw-md5 hashes.txt
```

- **Hydra (Brute Force SSH):**

```
hydra -l admin -P passwords.txt ssh://192.168.1.5
```

Always test on systems you own or have permission to attack.

9. Computer Forensics and Tools

- **Tools:**
  - **Autopsy:** GUI forensic suite
  - **FTK Imager:** Disk imaging
  - **Sleuth Kit (TSK):** CLI forensic tools
  - **Volatility:** RAM forensics
- **Example: Analyze USB device history with Autopsy**
  - Add disk image → Analyze metadata, deleted files, USB history.

10. Encrypt and Decrypt Messages with Analysis

- **Example with Caesar Cipher (Python):**

```
def encrypt(text, shift): return ''.join([chr((ord(c)-65+shift)%26+65) if
c.isalpha() else c for c in text.upper()])
def decrypt(text, shift): return encrypt(text, -shift)
```

- **Security Analysis:**
  - Caesar cipher is **not secure** (brute-force in 25 tries).
  - Use AES/GPG for modern encryption (see #7).