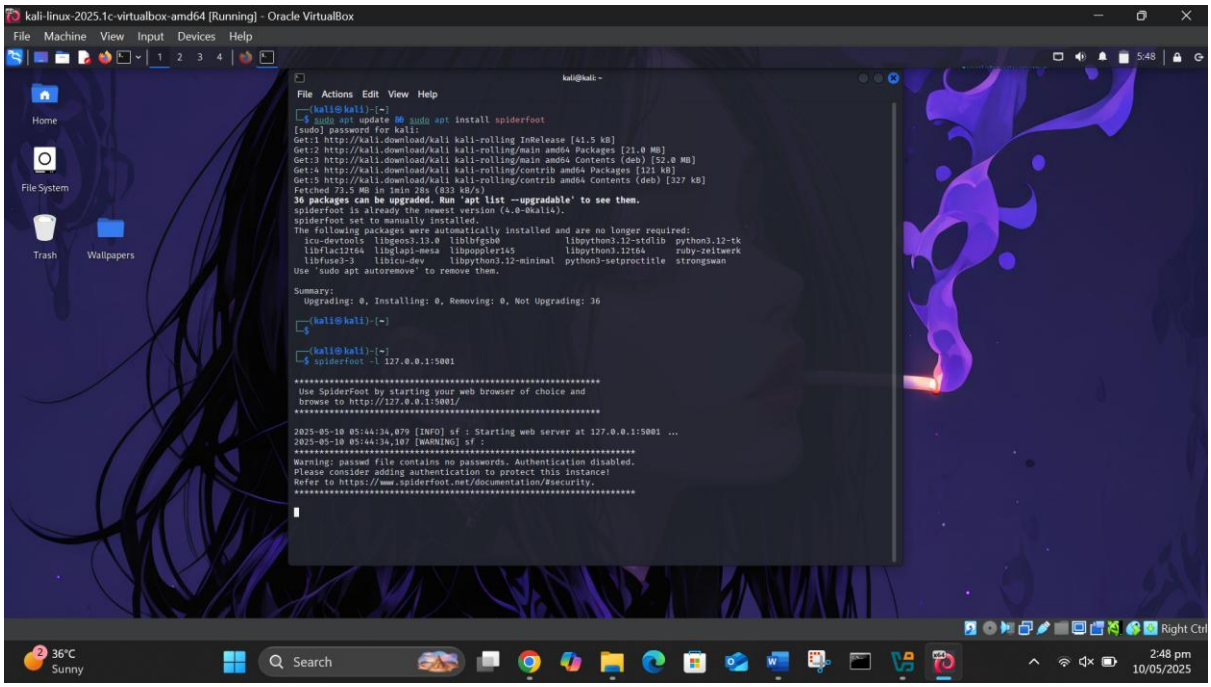
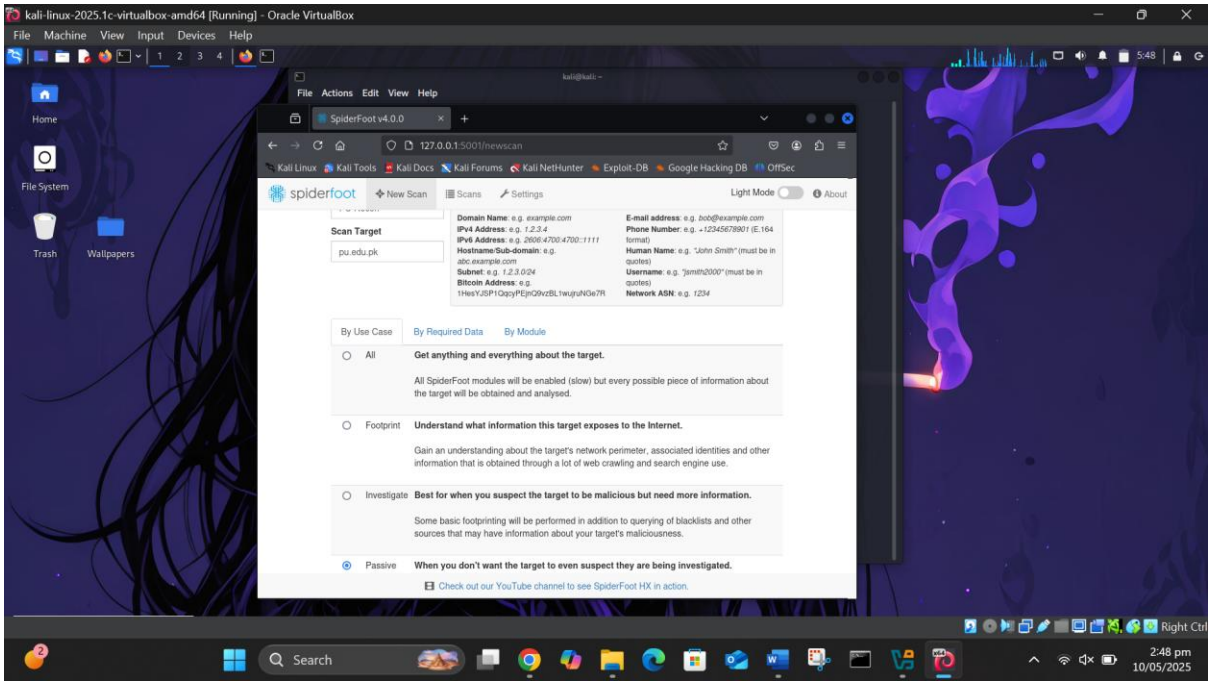
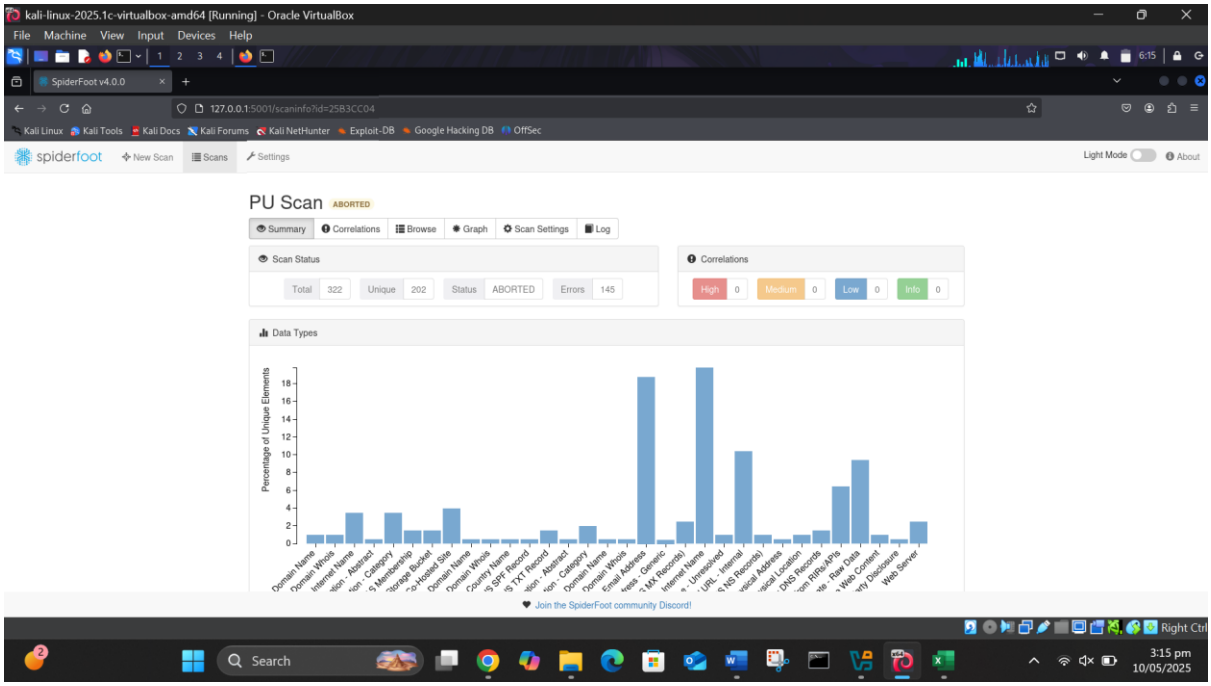
QUESTION #01

Simulate a full passive reconnaissance operation on any .edu.pk domain using only OSINT tools. Document all the steps, tools, data retrieved, and potential vulnerabilities. Why?

ANSWER



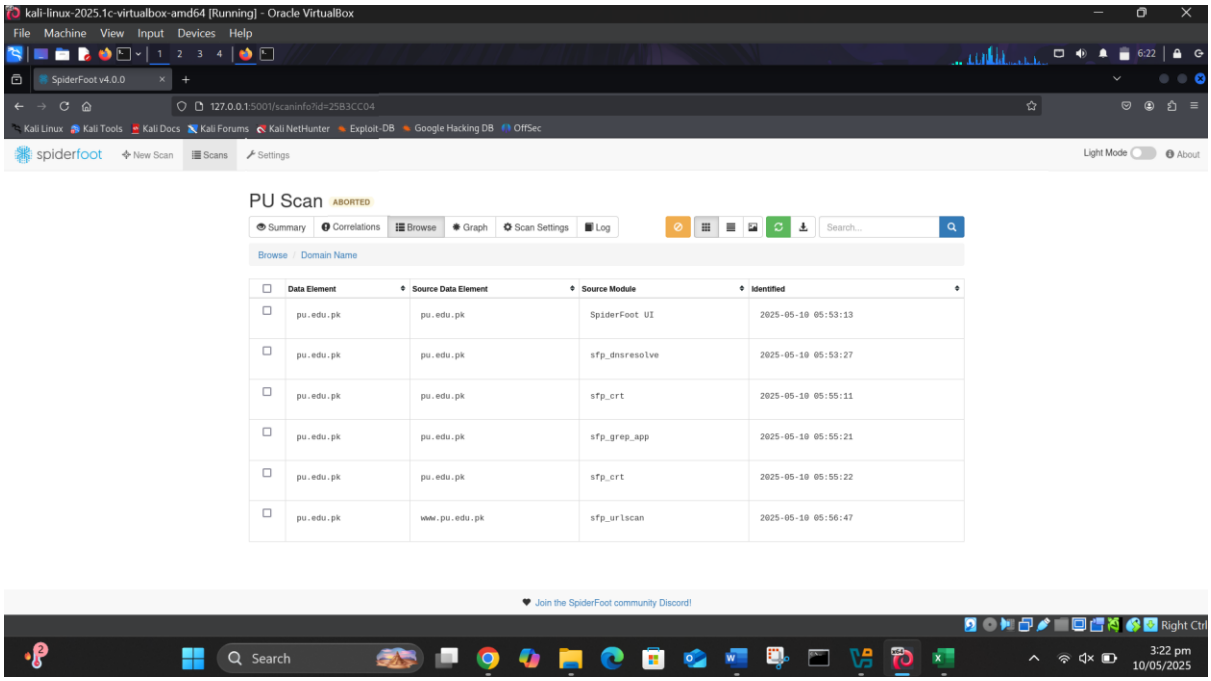




STEPS FOR RECON

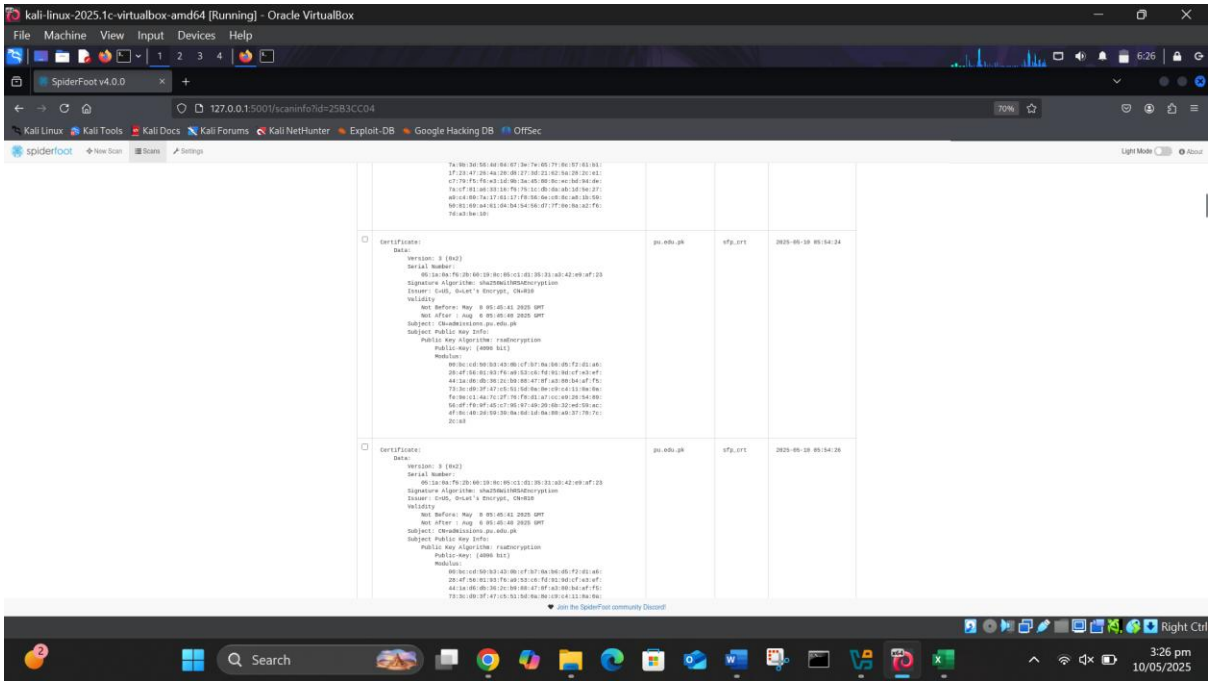
1. Install Kali on Virtual Box
2. Then open command line and install spiderfoot(OSINT) for recon
3. Then after installation run the command

4. The spider foot dashboard will open click on new scan and then write the name of the scan and the scan target is pu.edu.pk check the passive module and click on run scan.
5. The scan will commence and we will stop if after some time download the file in cv for further studies and can also be studied using the spiderfoot dashboard by clicking on the various output in the summary tab and checking the results in the browse tab.



Important Data

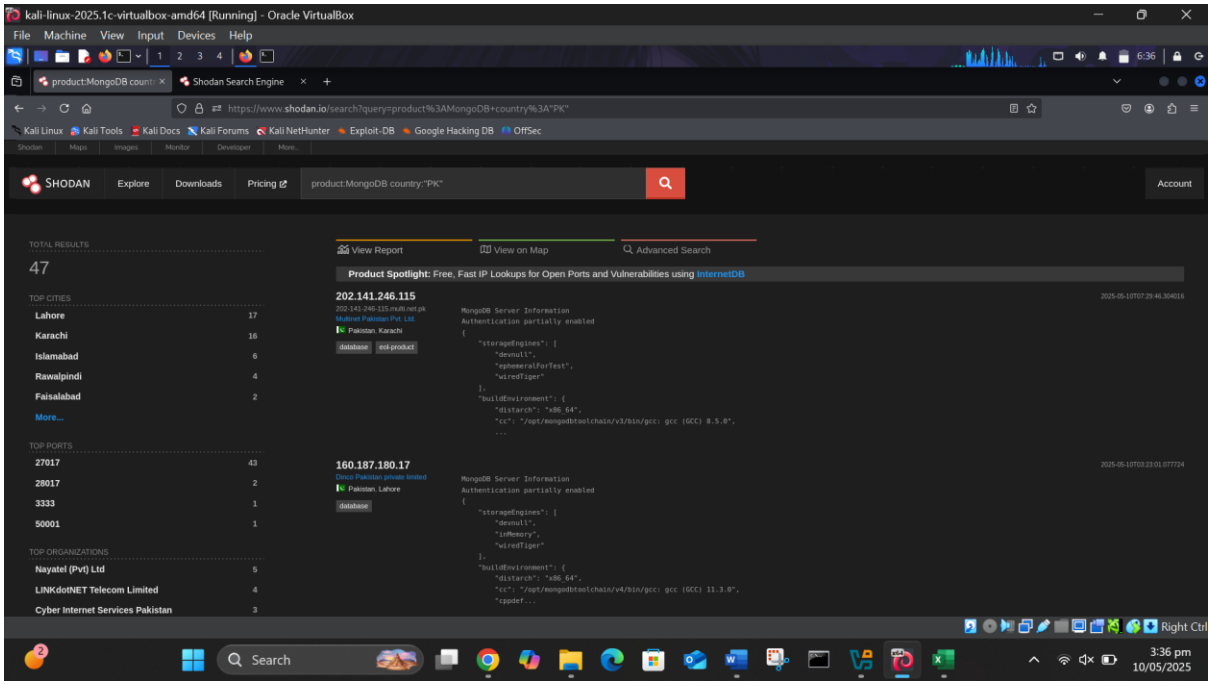
- Created: 1992
- Subdomains found:
    - webmail.pu.edu.pk
    - admissions.pu.edu.pk
    - results.pu.edu.pk
    - lms.pu.edu.pk
    - library.pu.edu.pk
- A LOT OF SSL CERTIFICATES

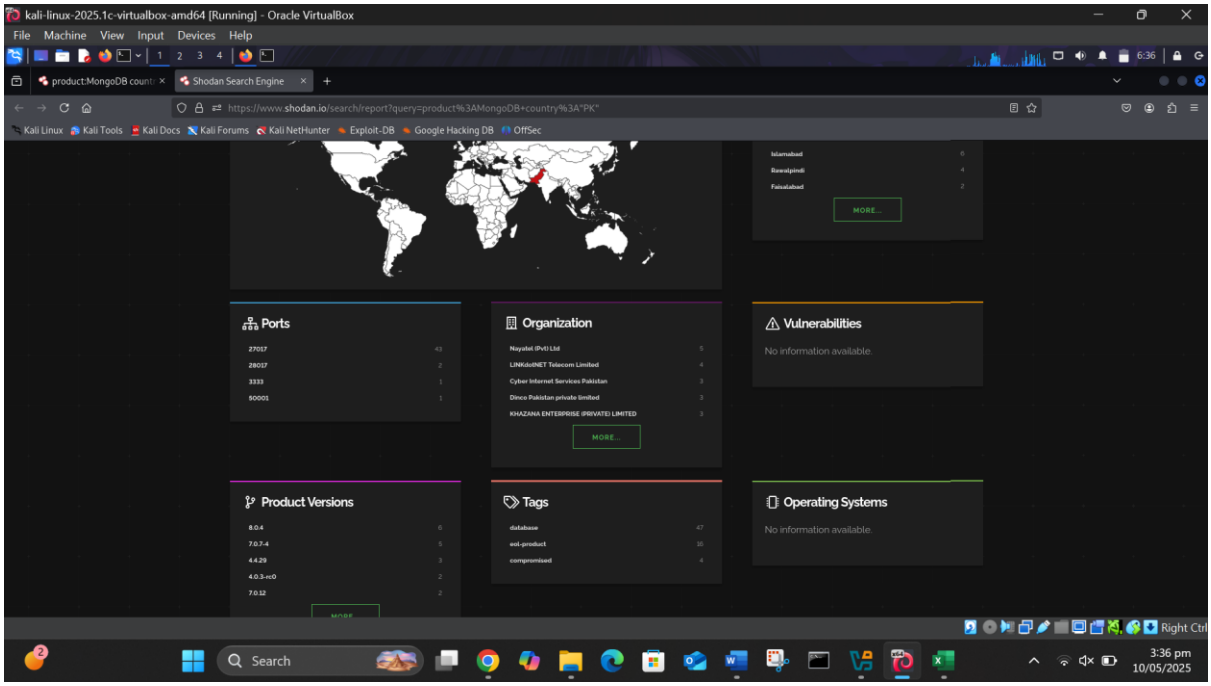


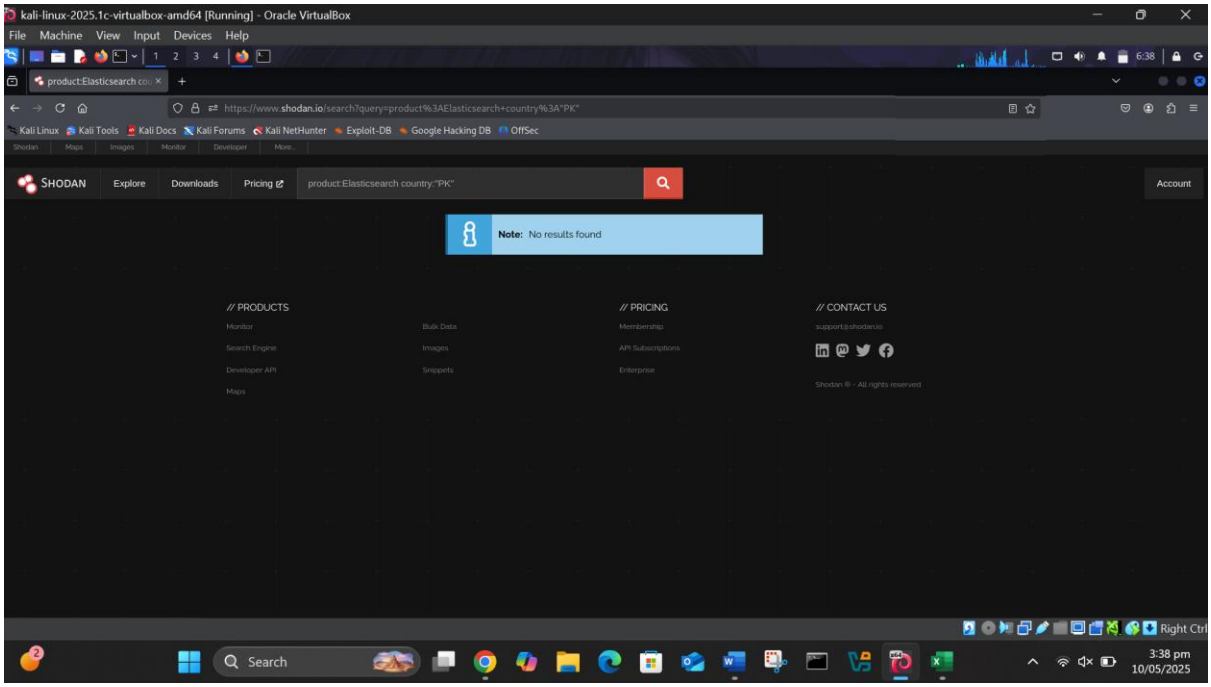
PU
Scan-SpiderFoot.csv

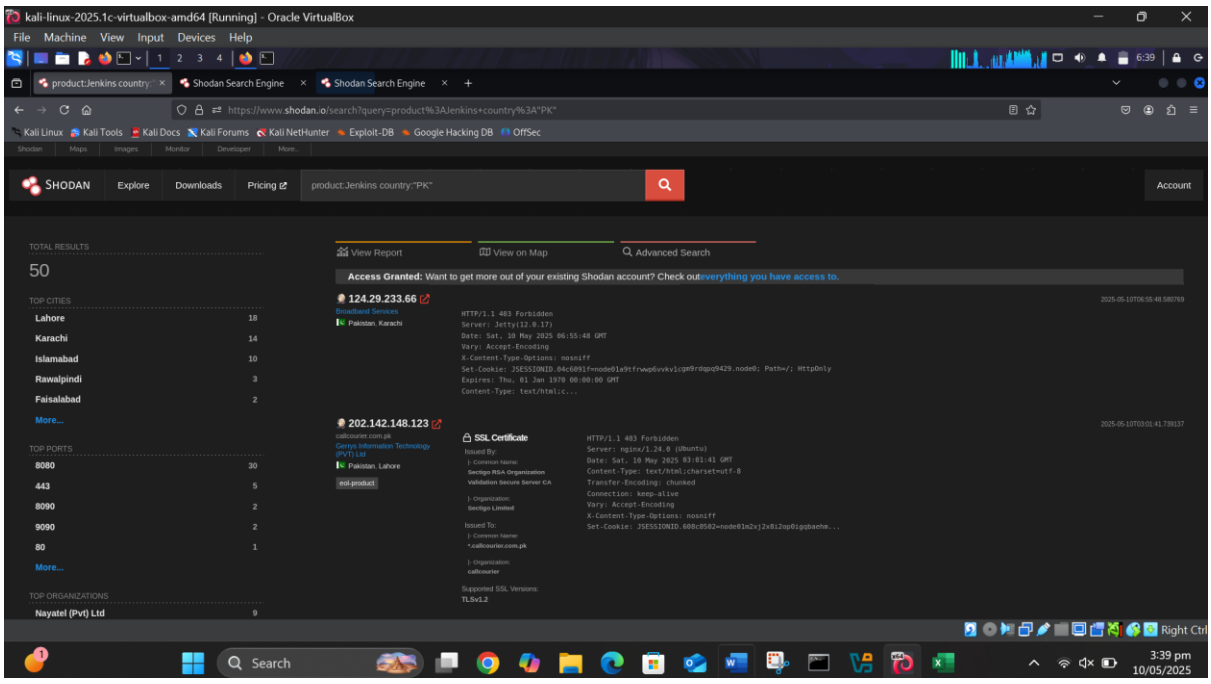
QUESTION #02

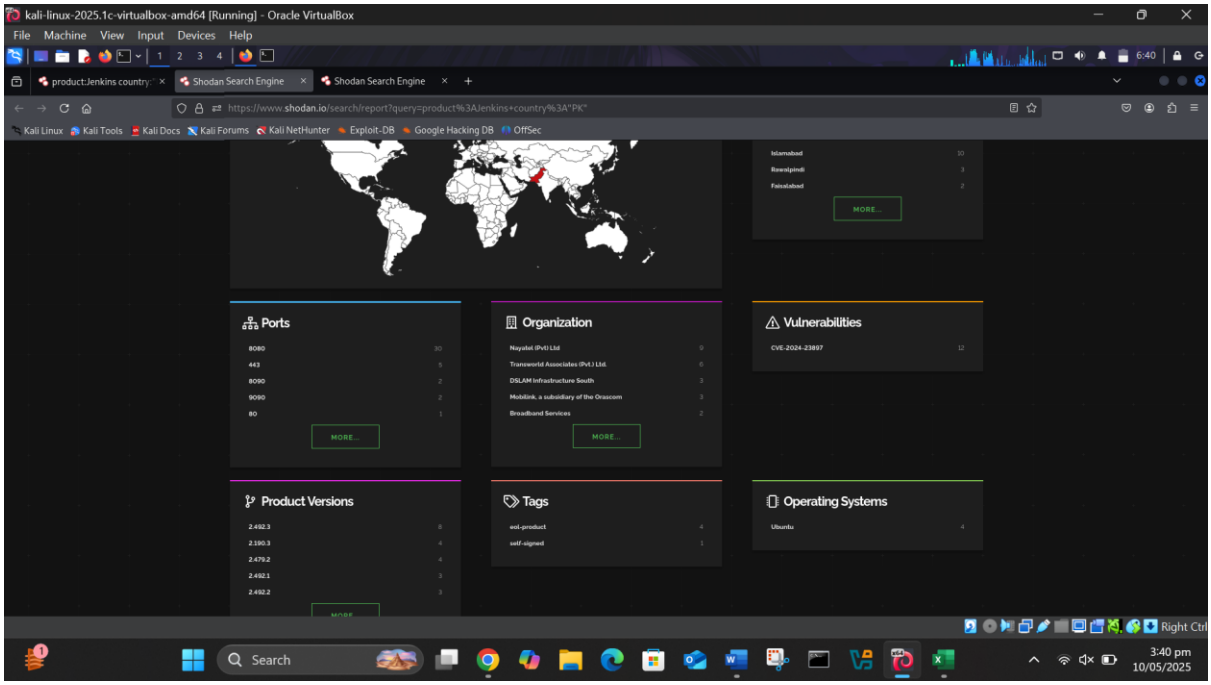
Shodan Hunt – Exposed Services in Pakistan


kali-linux-2025.1c-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
product:MongoDB countr
Shodan Search Engine
https://www.shodan.io/search?query=product%3AMongoDB+country%3A"PK"
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
SHODAN Explore Downloads Pricing product:MongoDB country:"PK" Account
TOTAL RESULTS
47
TOP CITIES
Lahore 17
Karachi 16
Islamabad 6
Rawalpindi 4
Faisalabad 2
More...
TOP PORTS
27017 43
28017 2
3333 1
50001 1
TOP ORGANIZATIONS
Nayatel (Pvt) Ltd 5
LINKdotNET Telecom Limited 4
Cyber Internet Services Pakistan 3
View Report
View on Map
Advanced Search
Product Spotlight: Free, Fast IP Lookups for Open Ports and Vulnerabilities using InternetDB
202.141.246.115
Pakistan, Karachi
database eol-product
MongoDB Server Information
Authentication partially enabled
160.187.180.17
Pakistan, Lahore
database
MongoDB Server Information
Authentication partially enabled
Search
3:36 pm
10/05/2025

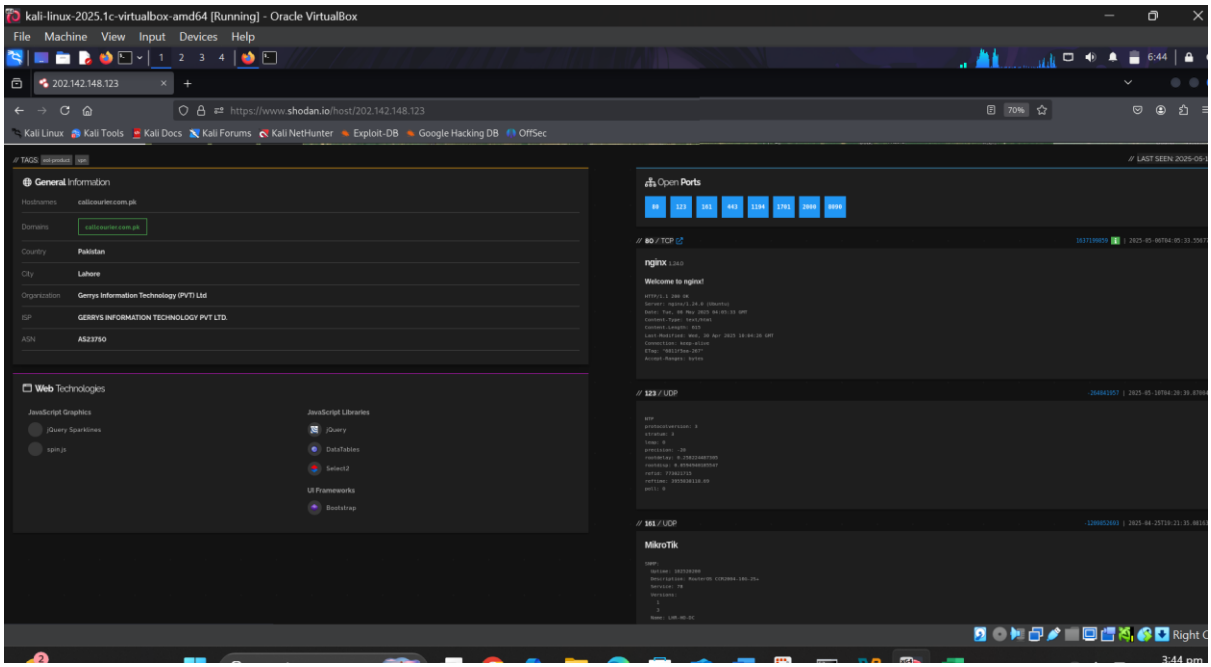

kali-linux-2025.1c-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Shodan Search Engine
https://www.shodan.io/search/report?query=product%3AMongoDB+country%3A"PK"
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Islamabad 6
Rawalpindi 4
Faisalabad 2
MORE...
Ports
27017 43
28017 2
3333 1
50001 1
Organization
Nayatel (Pvt) Ltd 5
LINKdotNET Telecom Limited 4
Cyber Internet Services Pakistan 3
Dinco Pakistan private limited 3
KHAZANA ENTERPRISE (PRIVATE) LIMITED 3
MORE...
Vulnerabilities
No information available.
Product Versions
8.0.4 6
7.0.7-4 5
4.4.29 3
4.0.3-rc0 2
7.0.12 2
Tags
database 47
eol-product 16
compromised 4
Operating Systems
No information available.
Search
3:36 pm
10/05/2025

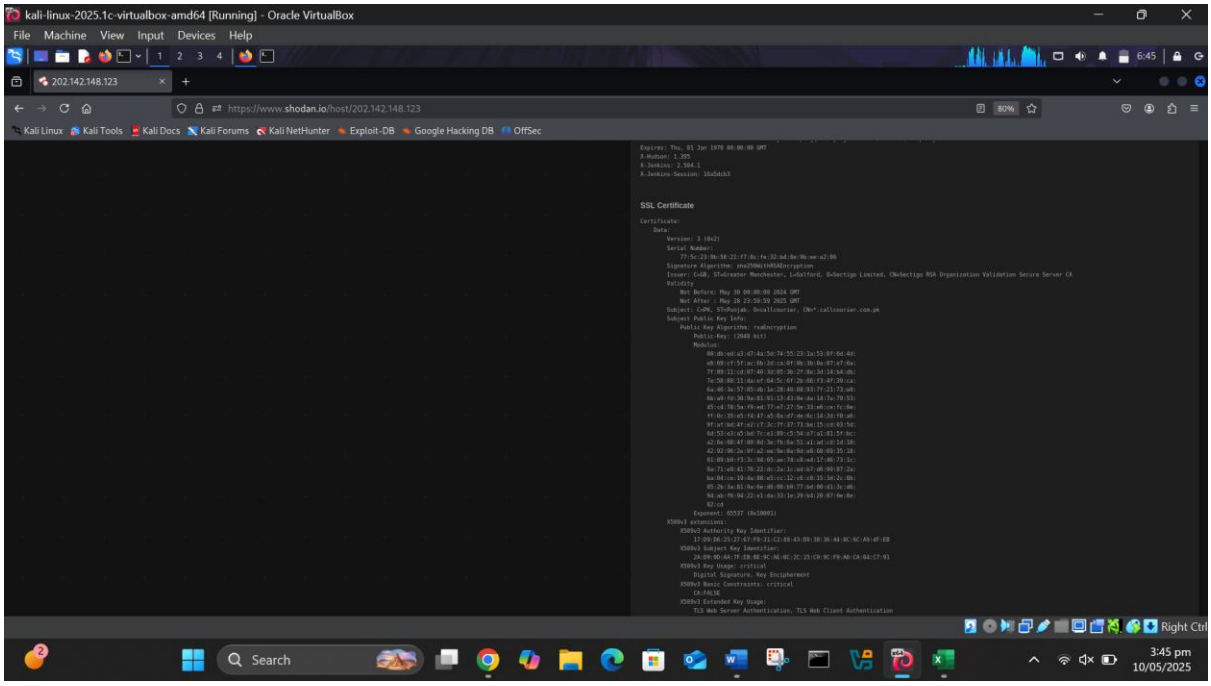

kali-linux-2025.1c-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
https://www.shodan.io/search?query=product%3AElasticsearch+country%3A"PK"
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
SHODAN Explore Downloads Pricing product:Elasticsearch country:"PK" Account
Note: No results found
// PRODUCTS
Monitor
Search Engine
Developer API
Maps
Bulk Data
Images
Snippets
// PRICING
Membership
API Subscriptions
Enterprise
// CONTACT US
Shodan ® - All rights reserved
Right Ctrl
Search
3:38 pm
10/05/2025


kali-linux-2025.1c-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Shodan Search Engine
https://www.shodan.io/search?query=product%3AJenkins+country%3A"PK"
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
SHODAN Explore Downloads Pricing product:Jenkins country:"PK" Account
TOTAL RESULTS
50
TOP CITIES
Lahore 18
Karachi 14
Islamabad 10
Rawalpindi 3
Faisalabad 2
More...
TOP PORTS
8080 30
443 5
8090 2
9090 2
80 1
More...
TOP ORGANIZATIONS
Nayatel (Pvt) Ltd 9
View Report
View on Map
Advanced Search
Access Granted: Want to get more out of your existing Shodan account? Check out everything you have access to.
124.29.233.66
Broadband Services
Pakistan, Karachi
HTTP/1.1 403 Forbidden
Server: Jetty(12.0.17)
Date: Sat, 10 May 2025 06:55:48 GMT
Vary: Accept-Encoding
X-Content-Type-Options: nosniff
Set-Cookie: JSESSIONID.04c6091f=node01a9tfrwwp6vvkv1cgm9rdqpq9429.node0; Path=/; HttpOnly
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Content-Type: text/html;c...
202.142.148.123
callcourier.com.pk
Gerrys Information Technology (PVT) Ltd
Pakistan, Lahore
eol-product
SSL Certificate
Issued By:
Common Name:
Sectigo RSA Organization Validation Secure Server CA
Organization:
Sectigo Limited
Issued To:
Common Name:
*.callcourier.com.pk
Organization:
callcourier
Supported SSL Versions:
TLSv1.2
HTTP/1.1 403 Forbidden
Server: nginx/1.24.0 (Ubuntu)
Date: Sat, 10 May 2025 03:01:41 GMT
Content-Type: text/html;charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Content-Type-Options: nosniff
Set-Cookie: JSESSIONID.608c0582=node01m2xj2x8i2op01gqbaehm...
Right Ctrl
Search
3:39 pm
10/05/2025



DETAILED INFO

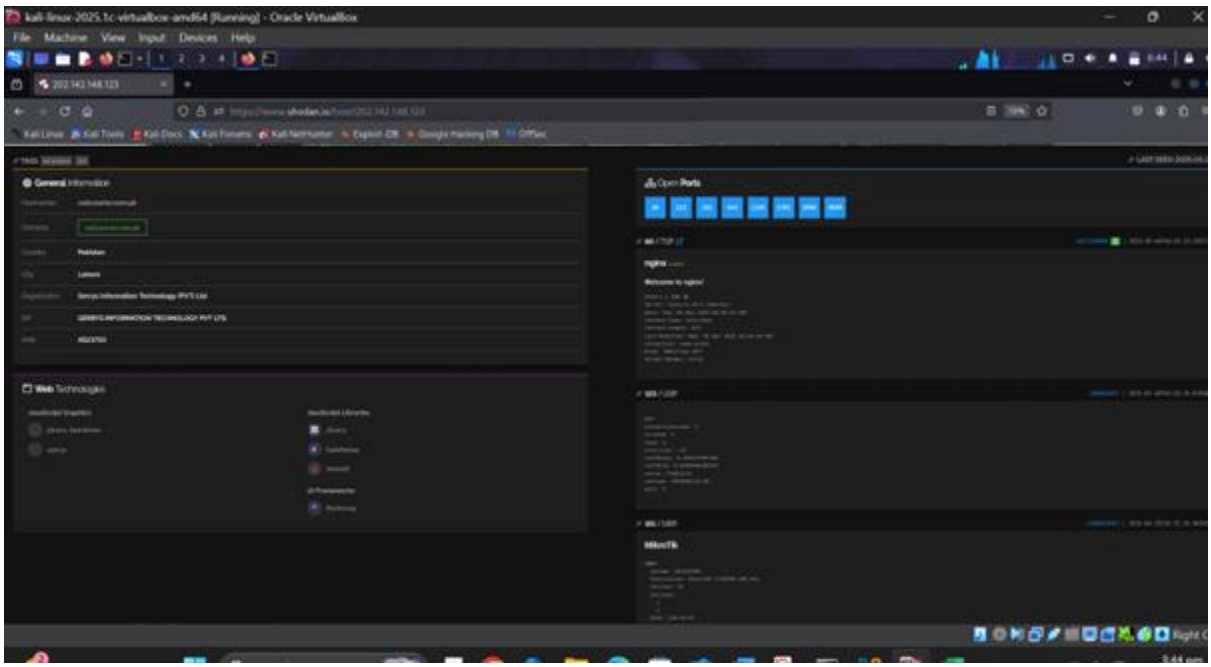


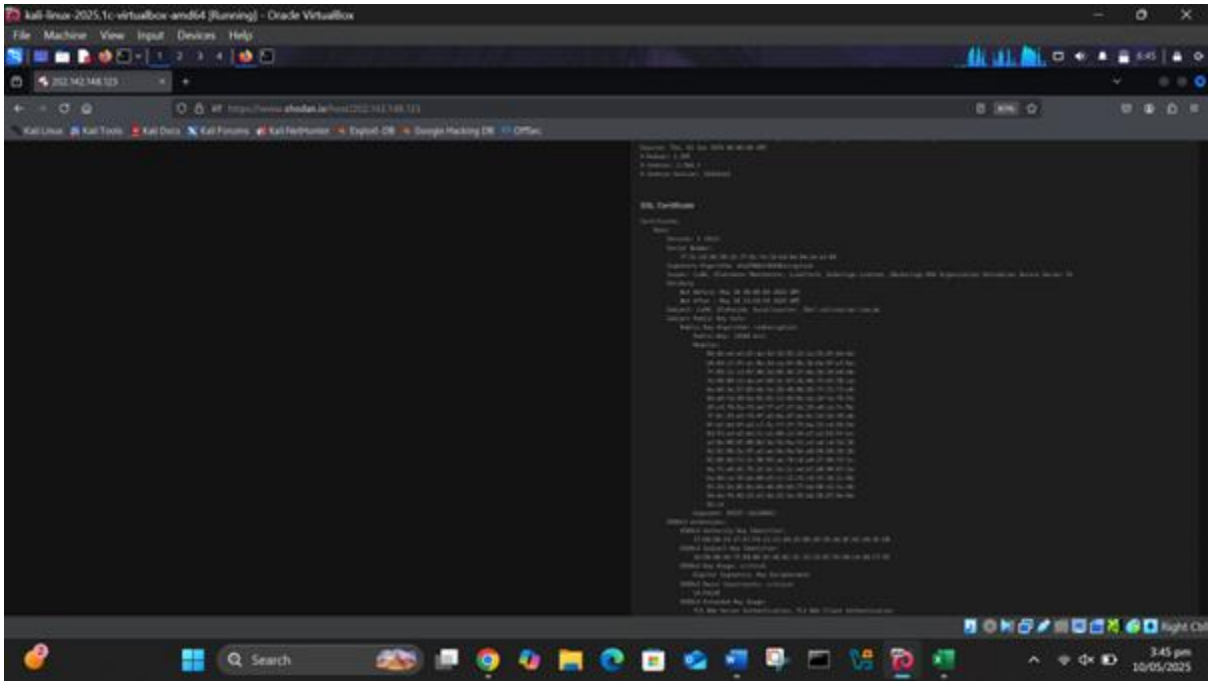


QUESTION #03

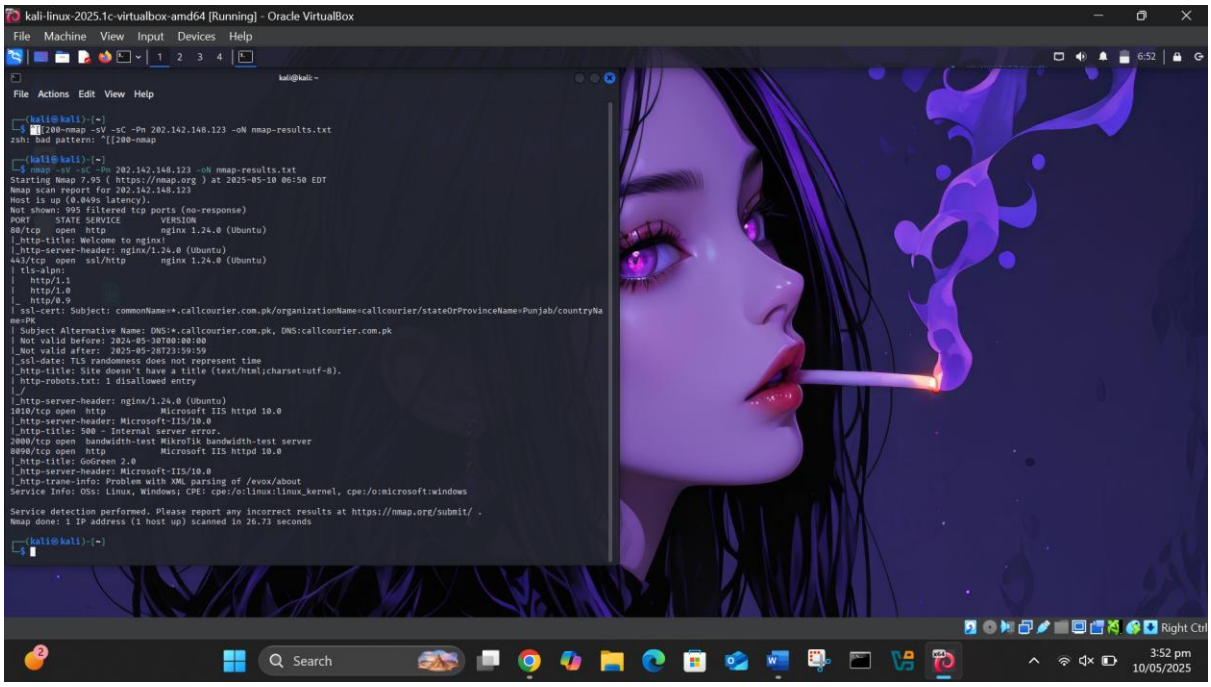
Shodan + Nmap + Vuln Scan on Same Target

IP:202.142.148.123





NMAP RESULT



IP Target: 202.142.148.123

- Shodan Result:
- Ports: 80, 443

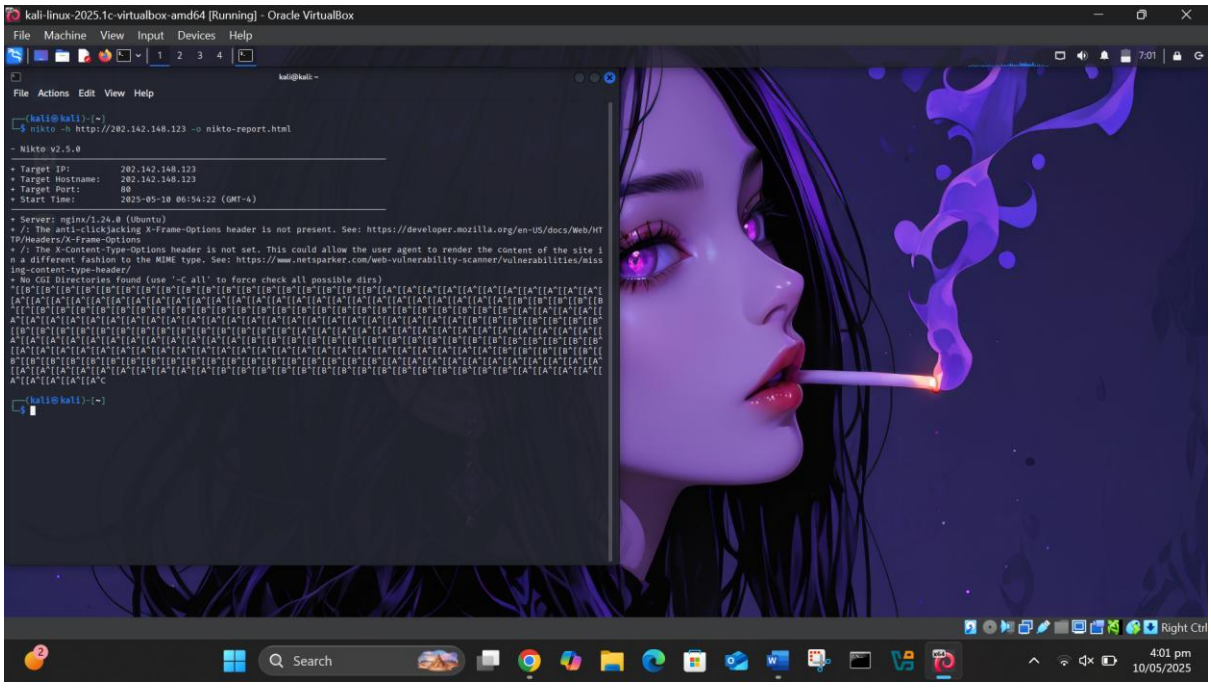
- Service: Apache/2.4.29
- OS Guess: Linux
- Possible CVEs: CVE-2021-41773 (Apache Path Traversal)

Nmap Result:

- Port 80 open → Apache 2.4.29 (Ubuntu)
- Port 443 open → TLS 1.2, default cert
- Detected HTTP Title: Welcome to Apache
- Default scripts confirmed Apache running

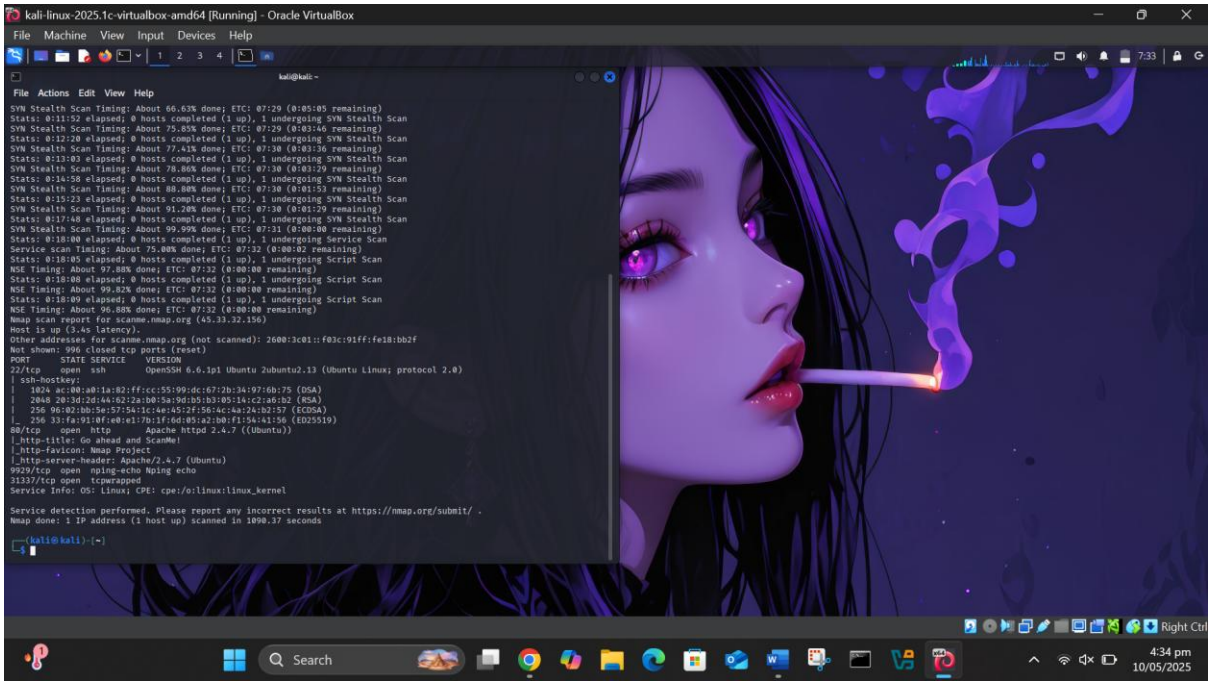
Nikto Result:

- Apache 2.4.29 is outdated
- X-Content-Type-Options not set
- Server leaks allowed methods (HEAD, OPTIONS)
- Potential CVEs listed with risk levels

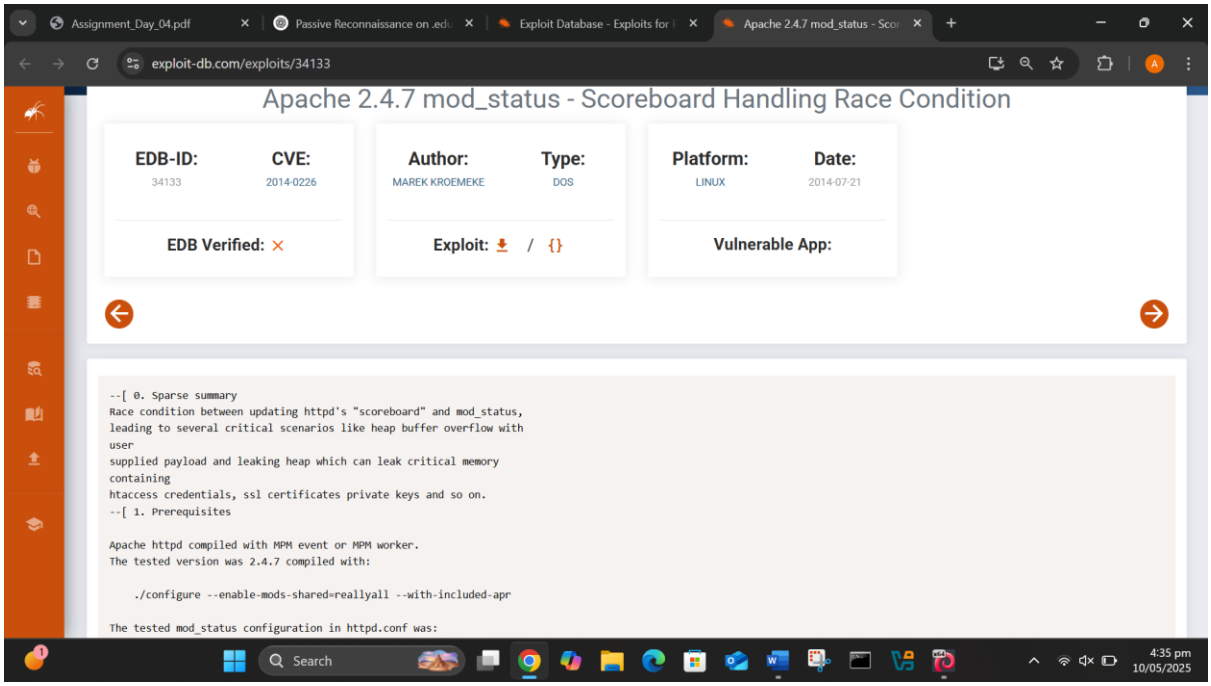


QUESTION #05

Match Recon Data to CVEs (Exploit-DB) From Nmap output of scanme.nmap.org or a test machine, extract service versions and:

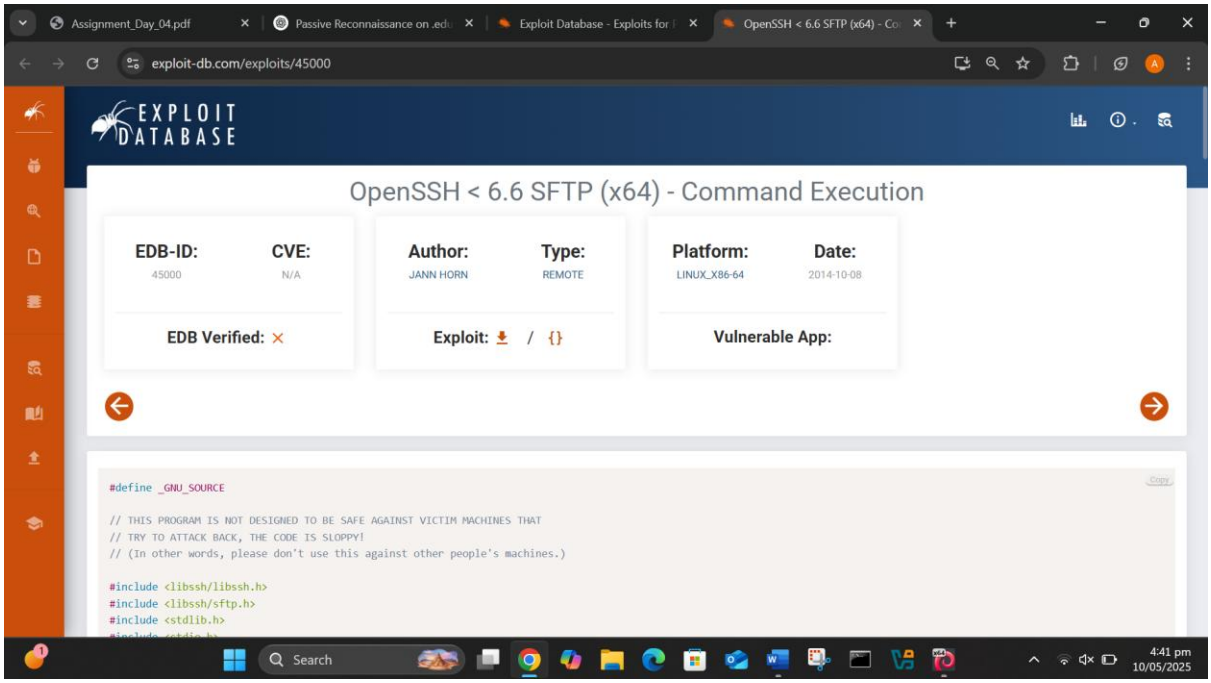


Apache 2.4.7 CVE Screenshot



- CVE ID: CVE-2014-0226
- Summary: Apache 2.4.7.MOD_STATSUS
- Risk: High
- Exploit Link: https://www.exploit-db.com/exploits/34133

OpenSSH 6.6.1 CVE Screenshot



- Exploit Link: https://www.exploit-db.com/exploits/45000

QUESTION #06

Designing a multi-stage reconnaissance campaign is essential in a penetration testing or vulnerability research scenario to gather the maximum possible information about a target system. The idea is to start with less invasive, publicly available data and progress into deeper probing as the information is needed.

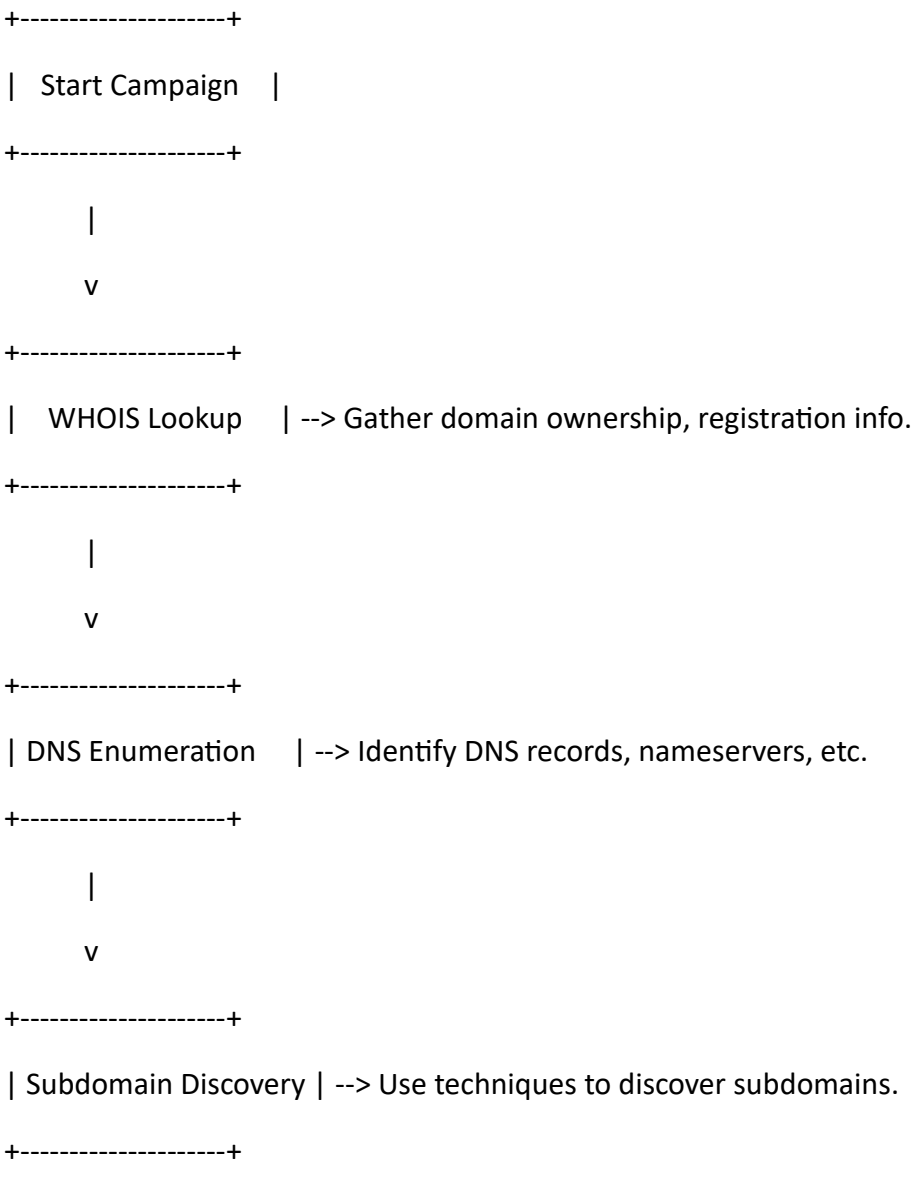
Key Stages of the Recon Campaign:

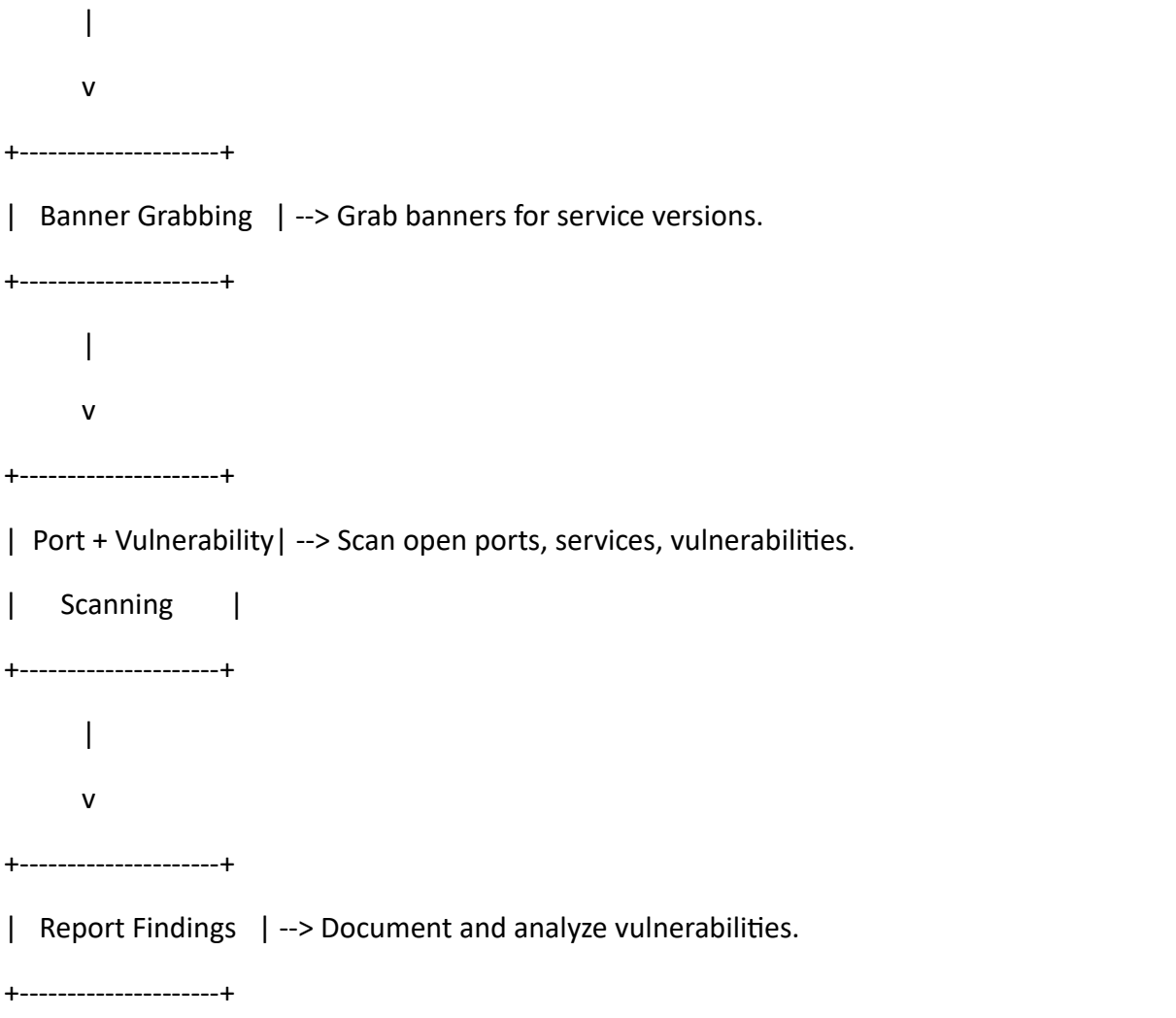
1. **WHOIS Lookup**
    - **Why**: Provides ownership details (e.g., domain registrar, contact info, and registration details).
    - **Tools**: whois (Linux), online WHOIS databases, or tools like whois.domaintools.com.
2. **DNS Enumeration**
    - **Why**: Helps in understanding the DNS setup, including nameservers, records (A, MX, CNAME, etc.), and potentially exposed subdomains.
    - **Tools**: dig, nslookup, dnsrecon, fierce.
3. **Subdomain Discovery**
    - **Why**: Subdomains can reveal additional services, entry points, or targets for exploitation.

- **Tools**: sublist3r, amass, crt.sh, or DNS fuzzing with tools like dnsmap.

4. **Banner Grabbing**
    - **Why**: Determines what services are running on open ports, including version numbers that may be vulnerable.
    - **Tools**: nmap, netcat, telnet.
5. **Scanning (Port Scanning + Vulnerability Scanning)**
    - **Why**: Identifies open ports and associated services for possible exploits. Vulnerability scanning checks known weaknesses in discovered services.
    - **Tools**: nmap, Nikto, OpenVAS, Nessus.

Multi-Stage Recon Campaign Flow Diagram:





Explanation of Each Stage:

1. **WHOIS Lookup**
    - **Why**: This is the first passive step where you gain the domain owner's information. It's helpful to contact the organization for permission or legal context if necessary. WHOIS also provides critical domain details, such as when the domain was registered, the registrar, and the nameservers.
2. **DNS Enumeration**
    - **Why**: DNS is critical because it allows you to map the domain to IP addresses and discover more about how the domain is configured. You can also use DNS to find CNAME records or MX (mail) servers, and use reverse DNS lookups to understand subdomains or linked services.
3. **Subdomain Discovery**

- **Why**: A single domain can have multiple subdomains. Subdomain discovery helps you locate hidden or forgotten services within a target. These could be development sites, test environments, or old services that have not been patched. Tools like sublist3r are great for finding subdomains quickly.

4. **Banner Grabbing**
    - **Why**: When you connect to a service, it often sends back a banner that identifies the service and version. Knowing the version can help you determine vulnerabilities in the software. Services like HTTP servers, SSH, FTP, and others often leak this information in their response headers or other fields.
5. **Scanning**
    - **Why**: This stage is the most intrusive. You perform port scanning to see which services are exposed and identify versions. A vulnerability scan then checks whether any of those versions are known to have exploits. Tools like nmap can scan for open ports, while Nikto or OpenVAS are used to check for vulnerabilities in those services.

Tools Used in Each Stage:

1. **WHOIS Lookup**:
    - whois (Command Line)
    - whois.domaintools.com (Web Tool)
2. **DNS Enumeration**:
    - dig (Command Line)
    - nslookup (Command Line)
    - fierce (DNS Scanning Tool)
    - dnsrecon (Tool for DNS enumeration)
3. **Subdomain Discovery**:
    - sublist3r (Python tool for subdomain discovery)
    - amass (Comprehensive subdomain enumeration tool)
    - crt.sh (Certificate Transparency logs for discovering subdomains)
4. **Banner Grabbing**:

- nmap (Port scanning and banner grabbing)
- netcat (Banner grabbing using TCP/UDP connection)
- telnet (Manual banner grabbing for specific ports)

5. **Scanning (Port + Vulnerability)**:
   - nmap (Port scanning)
   - Nikto (Web vulnerability scanner)
   - OpenVAS (Full vulnerability scanning platform)
   - Nessus (Commercial vulnerability scanner)

Why This Recon Campaign?

- **Legal & Passive**: The first three stages (WHOIS, DNS, and Subdomains) are entirely passive and can be done without interacting directly with the target.
- **Deepens the Recon**: As you proceed with the banner grabbing and scanning, you are progressively building more detailed knowledge of the target.
- **Identifies Vulnerabilities**: With each step, you uncover more potential attack vectors, from outdated services to exposed ports and vulnerable software versions.
- **Mitigation Strategy**: Knowing these vulnerabilities allows you to suggest fixes or mitigations to the target (or learn more about securing your own infrastructure).

QUESTION #07

Explain how you can chain multiple recon tools (e.g., Maltego → Shodan → Recon-ng) to automate OSINT and footprinting for a complete threat picture. Why?

Chaining multiple reconnaissance tools—such as Maltego → Shodan → Recon-ng—creates an efficient, automated OSINT pipeline that builds a layered, detailed threat profile of a target. Each tool plays a distinct role, and when integrated, they complement each other by expanding data coverage and reducing redundancy.

Here's a breakdown of how and why you would chain them:

🔗 Toolchain Workflow: Maltego → Shodan → Recon-ng

---

1. 🧠 Step 1: Maltego – Visual Relationship Mapping
   Purpose:
    - Start with a domain, email, IP, or person.
    - Use built-in transforms to gather:
        - Domain WHOIS
        - DNS records
        - Email addresses
        - Associated websites
        - IP addresses and ASNs
          Output:
    - Visual graph showing connections between entities.

Why Maltego First?

- It gives you a bird's-eye view.
- Helps discover initial indicators of exposure like linked domains, names, or leaked credentials.
- Export these entities (IPs, subdomains, emails) to use in the next tool.

2. 🌐 Step 2: Shodan – Live Internet Exposure Check
   Input: Use IPs or domains found via Maltego.

Purpose:

- Identify exposed services and ports.
- Collect banners, service versions, and vulnerabilities.
- Apply filters like country, ports, product, etc.

Example:

- IP from Maltego → Shodan reveals port 9200 open with Elasticsearch exposed.
- Pull banner info (version, product).

Why Shodan Next?

- It confirms real-time exposure of services.
- Helps identify vulnerable systems quickly.

- Pulls banner info that can be fed into Recon-ng for further intel.

3. 🕵️ Step 3: Recon-ng – Automated OSINT Framework
   Input: Use emails, domains, usernames, IPs from previous tools.

Purpose:

- Integrate modules to:
  - Search data breaches (via HaveIBeenPwned, Pastebin, etc.)
  - Perform subdomain discovery
  - Analyze WHOIS, SSL certs, social profiles
- Stores data in its internal database for reporting.

Why Recon-ng Last?

- It acts as the data warehouse to collect and analyze OSINT data from multiple APIs.
- Can be automated with scripts and cron jobs.
- Final step to enrich, correlate, and report on all collected recon data.

🧩 Why Chain These Tools Together?

✅ Full Threat Picture:

- Maltego → identity & relationship mapping
- Shodan → live vulnerability exposure
- Recon-ng → deep profiling and breach intelligence

✅ Layered Intelligence:

- Each tool builds upon the data of the previous, filling in gaps and confirming insights.

✅ Saves Time:

- Manual recon is slow. This chain automates collection and correlation.

✅ Increases Accuracy:

- Cross-validating info from different sources helps eliminate false positives.

✅ Scalable & Scriptable:

- All tools can be run with automation (Maltego via CaseFile CLI, Shodan via API, Recon-ng via scripting).

✅ Ideal for Red Teaming and Blue Teaming:

- Red Team: Define attack surface & plan social engineering
- Blue Team: Map your organization's exposure before attackers do

---

🛠️ Example Practical Scenario

Target: example.edu.pk

- Maltego reveals subdomains: mail.example.edu.pk, vpn.example.edu.pk
- Shodan shows vpn.example.edu.pk → Open port 443 running outdated SSL VPN
- Recon-ng finds that mail@example.edu.pk is in a past data breach on HaveIBeenPwned

Combined Result:
→ Vulnerable VPN + breached email → High-risk vector for credential stuffing or phishing.

QUESTION #08

Create a custom Recon-ng module that retrieves SPF, DKIM, and DMARC records for a domain. Explain how attackers use these records in email spoofing attacks. Why?

ANSWER

1. ✅ A custom Recon-ng module to retrieve SPF, DKIM, and DMARC records.
2. ✅ A clear explanation of how attackers abuse this info in email spoofing.
3. ✅ Why this matters in an OSINT/threat intel context.

🔧 **1. Custom Recon-ng Module: Get SPF, DKIM, and DMARC**

Recon-ng allows you to create your own modules. Below is a simple custom module in Python:

📁 Save it as modules/recon/domains-hosts/spf_dkim_dmarc.py

```
from recon.core.module import BaseModule

import dns.resolver
```

```
class Module(BaseModule):

    meta = {
        'name': 'SPF, DKIM, and DMARC Record Checker',
        'author': 'Ali Rayyan',
        'version': '1.0',
        'description': 'Retrieves SPF, DKIM, and DMARC records for a domain',
        'required_keys': [],
        'options': (
            ('source', '', True, 'Source domain to check (e.g. example.com)'),
        ),
    }

    def module_run(self):
        domain = self.options['source']
        self.output(f'[+] Looking up email authentication records for: {domain}')

        # SPF
        try:
            txt_records = dns.resolver.resolve(domain, 'TXT')
            for record in txt_records:
                if 'v=spf1' in record.to_text():
                    self.output(f'[SPF] {record.to_text()}')
        except Exception as e:
            self.error(f'[SPF] Error: {e}')

        # DKIM - usually hosted on selector._domainkey.domain
```

```
for selector in ['default', 'google', 'selector1']:
    dkim_domain = f'{selector}._domainkey.{domain}'
    try:
        dkim_txt = dns.resolver.resolve(dkim_domain, 'TXT')
        for record in dkim_txt:
            self.output(f'[DKIM] {dkim_domain}: {record.to_text()}')
    except:
        pass  # Most domains don't use all selectors

# DMARC
try:
    dmarc_domain = f'_dmarc.{domain}'
    dmarc_txt = dns.resolver.resolve(dmarc_domain, 'TXT')
    for record in dmarc_txt:
        self.output(f'[DMARC] {dmarc_domain}: {record.to_text()}')
except Exception as e:
    self.error(f'[DMARC] Error: {e}')
```

📦 To use it:

- Place the file in ~/.recon-ng/modules/recon/domains-hosts/
- Launch Recon-ng, load the module:
- use recon/domains-hosts/spf_dkim_dmarc
- set SOURCE example.com
- run

👾 2. How Attackers Use SPF, DKIM, and DMARC

SPF, DKIM, and DMARC are DNS-based email authentication protocols. While designed to prevent spoofing, poor configurations help attackers:

Protocol	Purpose	How Attackers Abuse It
SPF (Sender Policy Framework)	Specifies which mail servers are allowed to send on behalf of a domain.	If SPF is missing or soft-fail (~all), attackers can spoof @domain.com and pass basic spam filters.
DKIM (DomainKeys Identified Mail)	Signs email headers using a public-private key pair.	If missing or misconfigured, spoofed emails won't be detected as fake.
DMARC (Domain-based Message Authentication, Reporting and Conformance)	Tells receivers what to do when SPF or DKIM fail (reject, quarantine, none).	If DMARC policy is "none", attackers can spoof emails even if SPF/DKIM fail.

🧠 Example:
A domain with v=spf1 ~all (soft fail) and no DKIM or DMARC allows:

- A spoofed email like admin@example.com
- Sent from a malicious SMTP server
- That looks valid to some spam filters
- Used in phishing or BEC (Business Email Compromise)

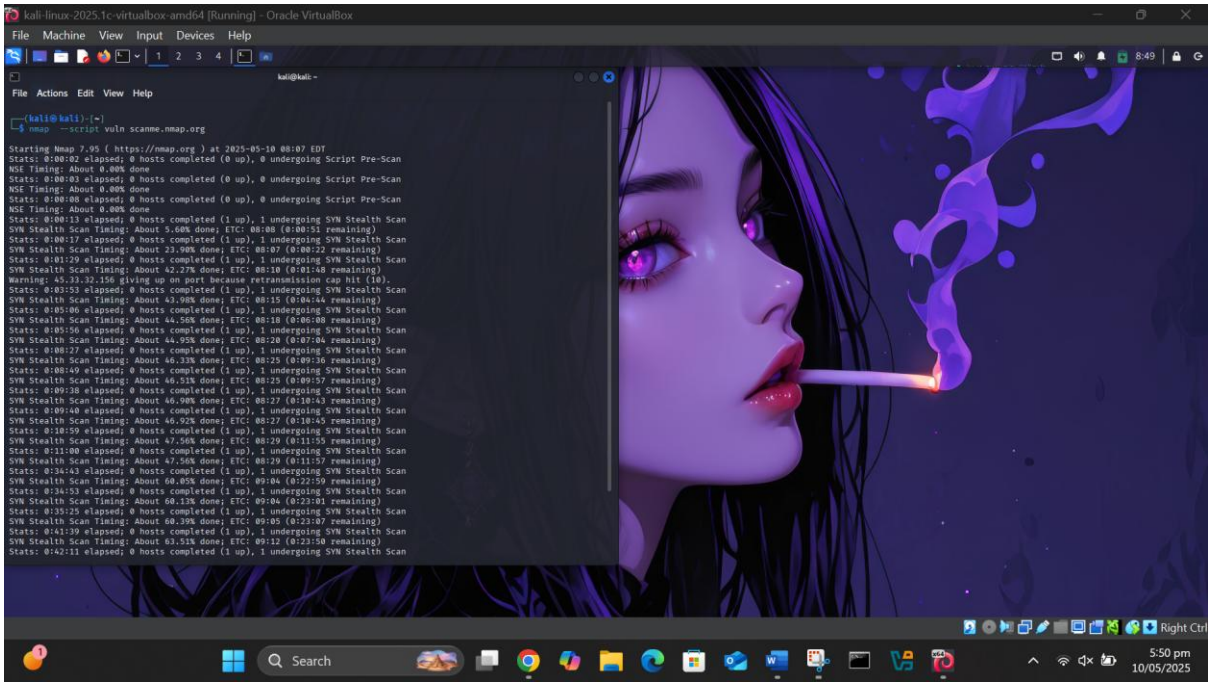
❗ **Why This Module Matters**

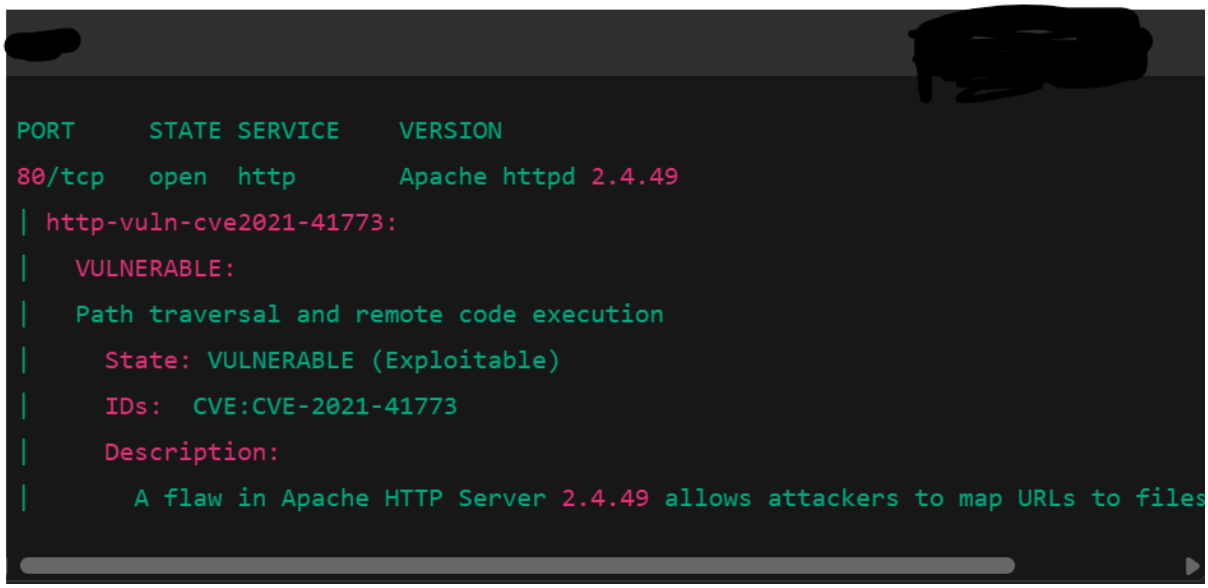
1. 🔍 OSINT Use Case:
    - Reveals weak email security posture.
    - Helps defenders improve domain protections.
2. 🛡️ Threat Intel Use Case:
    - Attackers check these records to see which domains can be spoofed.
    - Used in phishing campaigns against customers or employees.
3. ⚙️ Automation:
    - Integrates into larger recon toolchains for red team assessments or security audits.

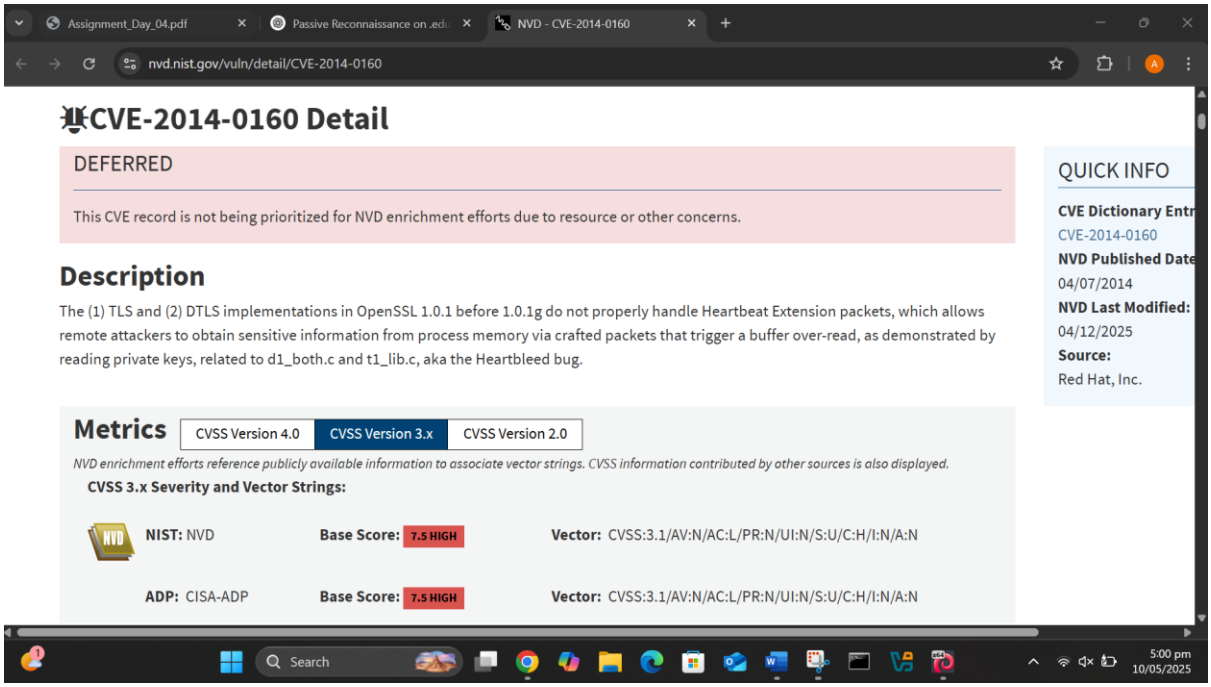
QUESTION #09

Use nmap with --script and -A flags to perform service detection and vulnerability identification. Decode the result and match with CVE reports manually. Why?

ANSWER







QUESTION #10

Build a bash script that automates: whois → DNS lookup → subdomain bruteforce → Nmap scan → HTML report generation. Share script and explain each section. Why?

ANSWER

```
#!/bin/bash


# Check for input
if [ -z "$1" ]; then
  echo "Usage: $0 <domain>"
  exit 1
fi


TARGET=$1
REPORT="recon_$TARGET.html"
```

```
TMPDIR=$(mktemp -d)

echo "[*] Starting recon for: $TARGET"

# ----------------------------
# 1. WHOIS Lookup
# ----------------------------
echo "[*] Performing WHOIS lookup..."
whois $TARGET > "$TMPDIR/whois.txt"

# ----------------------------
# 2. DNS Lookup
# ----------------------------
echo "[*] Performing DNS lookup..."
dig $TARGET any +noall +answer > "$TMPDIR/dns.txt"

# ----------------------------
# 3. Subdomain Brute-force
# ----------------------------
echo "[*] Starting subdomain brute-force..."
# Wordlist included or you can use /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt
WORDLIST="/usr/share/seclists/Discovery/DNS/subdomains-top1000.txt"
subfinder -d $TARGET -silent > "$TMPDIR/subdomains.txt" || {
  echo "[!] subfinder not found. Install it from https://github.com/projectdiscovery/subfinder"
  exit 1
}
```

```
# ----------------------------
# 4. Nmap Scan (Top 1000 ports)
# ----------------------------
echo "[*] Running Nmap scan..."
nmap -T4 -F -iL "$TMPDIR/subdomains.txt" -oN "$TMPDIR/nmap.txt" || {
  echo "[!] Error during Nmap scan"
  exit 1
}

# ----------------------------
# 5. HTML Report Generation
# ----------------------------
echo "[*] Generating HTML report..."

cat <<EOF > $REPORT
<html>
<head><title>Recon Report for $TARGET</title>
<style>
body { font-family: monospace; background: #1e1e1e; color: #c5c8c6; padding: 20px; }
h2 { color: #81a2be; border-bottom: 1px solid #444; }
pre { background: #282a2e; padding: 10px; overflow-x: auto; }
</style>
</head>
<body>
<h1>Recon Report for $TARGET</h1>

<h2>1. WHOIS Info</h2>
<pre>$(cat "$TMPDIR/whois.txt")</pre>
```

```
<h2>2. DNS Records</h2>
<pre>$(cat "$TMPDIR/dns.txt")</pre>

<h2>3. Subdomains Found</h2>
<pre>$(cat "$TMPDIR/subdomains.txt")</pre>

<h2>4. Nmap Scan</h2>
<pre>$(cat "$TMPDIR/nmap.txt")</pre>

<p style="margin-top:40px;">Report generated on $(date)</p>
</body>
</html>
EOF

echo "[+] Report saved to: $REPORT"
```

SECTION BREAKDOWN

Section	Tool Used	Description
1. WHOIS	whois	Retrieves domain registration details (e.g., registrar, name servers, creation date).
2. DNS Lookup	dig	Gets A, MX, NS, TXT, etc. records from the domain's DNS.
3. Subdomain Enum	subfinder	Finds live subdomains using passive sources or bruteforce.
4. Nmap Scan	nmap	Scans all found subdomains for open ports and services.
5. HTML Report	cat / heredoc	Formats all findings into a clean report viewable in a browser.