**NAME: Aminu Ibrahim Jalo** **DATE:15/11/24**

**ID: IDEAS/24/7357**

**<u>COURSE: INT 302 Assignment</u>** **<u>LAB 8:</u>**

**Web Application Security Testing with Burp Suite and OWASP ZAP.**

**Lab Steps**

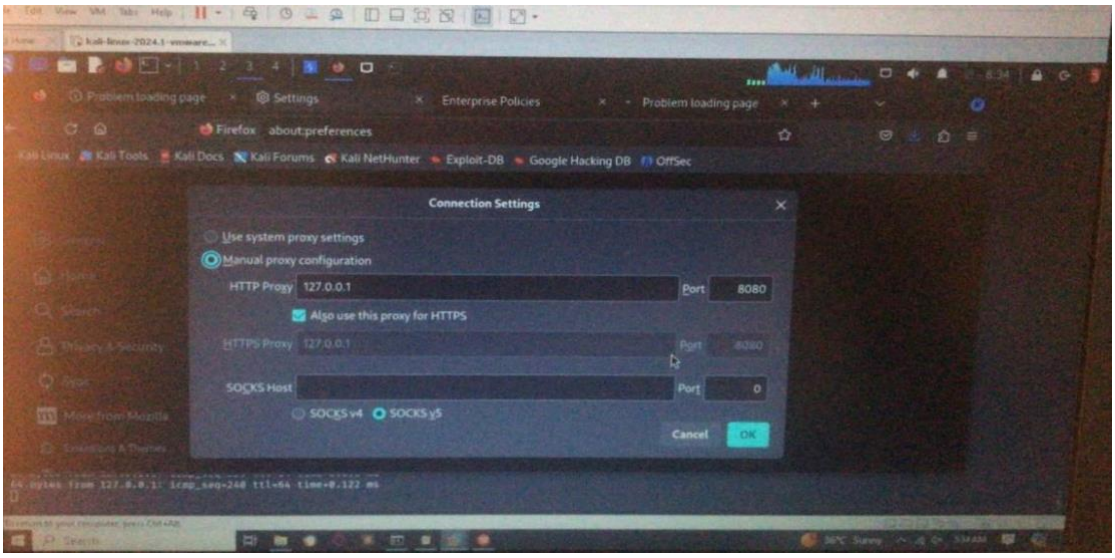
**Step 1: Setting Up Burp Suite**

**1Launch Burp Suite:**

• Start Burp Suite from your Kali Linux environment.
• Choose the Community edition for this lab.

**2.Configure Browser to Use Burp Proxy:**

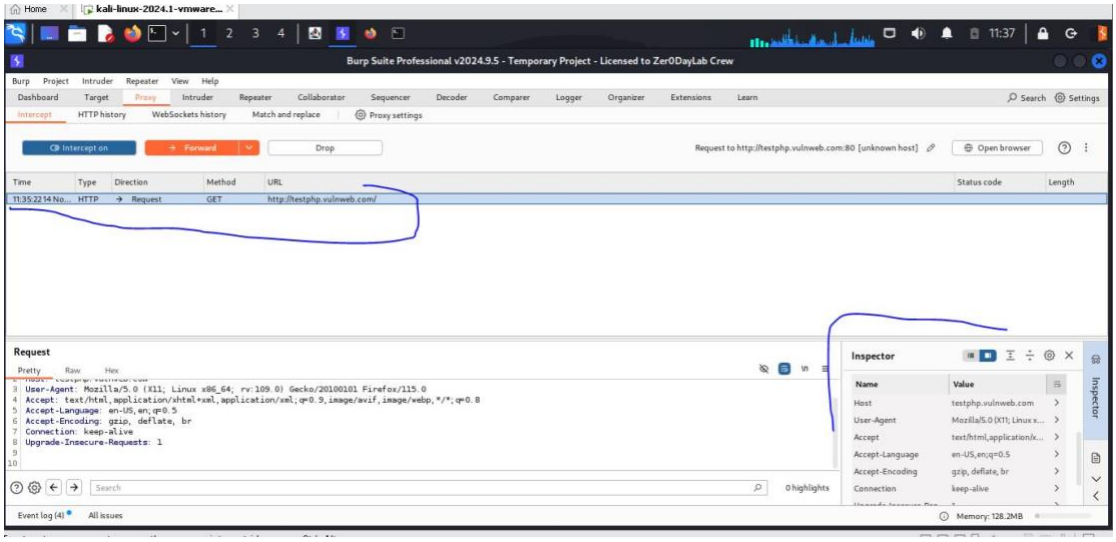
• Set up your browser (Firefox or Chrome) to route traffic through Burp Suite:
• Go to your browser settings.
• Configure the proxy settings to use 127.0.0.1 and port 8080.




**3.Intercepting Traffic:**

• Ensure that the “Intercept” feature is turned on in Burp Suite.
• Visit any web application, like http://testphp.vulnweb.com, to observe how Burp Suite captures
and displays the traffic.

**Exercise 1:**• Document the HTTP request and response headers for the home page of the target application.
What information do you find in these headers?




**Step 2: Using Burp Suite for Vulnerability Scanning**

1. **Spidering the Application:**

• Use the Spider tool in Burp Suite to crawl the application and gather all available URLs.

• Right-click on the target site in the site map and select “Spider this host.”

**Exercise 2:**

• List the URLs discovered during the spidering process. Did you find any hidden or interesting
pages?

**2.Active Scanning**:

• After spidering, select the site and choose “Scan” to start an active scan.

• Review the scan results to identify any vulnerabilities found.

**Exercise 3:**

• What vulnerabilities were detected by Burp Suite? Choose one vulnerability and explain how it could be exploited.

**Step 3: Setting Up OWASP ZAP**

**1.Launch OWASP ZAP:**

• Start OWASP ZAP from your Kali Linux environment.

**2.Configure Proxy Settings:**

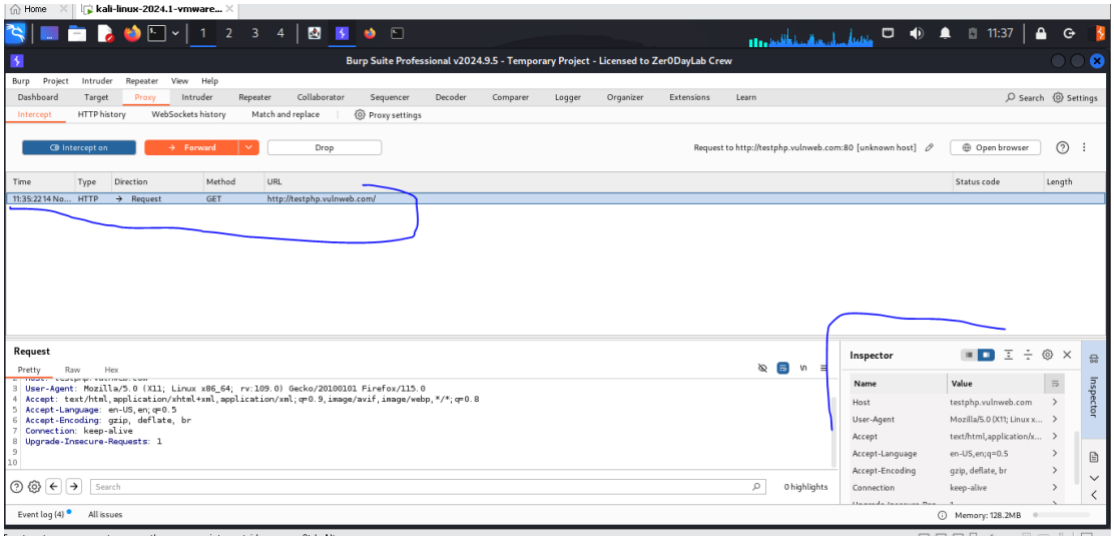
• Similar to Burp Suite, configure your browser to route traffic through OWASP ZAP using
127.0.0.1 and port 8080.

**3.Intercepting Traffic:**

• Visit the same web application used in Burp Suite while OWASP ZAP is running.

**Exercise 4:**

• Capture and analyze the traffic with OWASP ZAP. What differences do you notice compared to Burp Suite?




**Step 4: Using OWASP ZAP for Vulnerability Scanning**

**1.Automated Scanner:**

• Utilize the "Quick Start" feature to run an automated scan on the target web application.

• Monitor the alerts generated by ZAP during the scan.

**Exercise 5:**• Review the vulnerabilities identified by OWASP ZAP. Which tools detected the same
vulnerabilities? What are the potential impacts of these vulnerabilities?

**2.Active Scan:**

• Perform an active scan by selecting the target site and initiating the scan.

• Review the detailed reports generated.

**Exercise 6:**

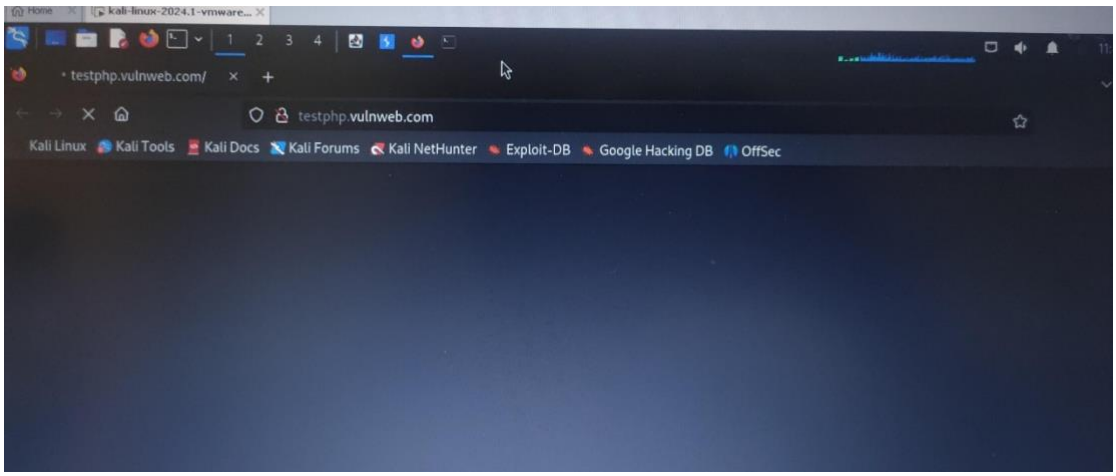
• Compare the findings of OWASP ZAP with Burp Suite. Which tool provided more detailed information? Which tool do you prefer for vulnerability scanning? Why?

**Step 5: Manual Testing Techniques**

**1.Fuzzing:**

• Use both tools to perform fuzzing against input fields in the web application (e.g., login forms, search fields).

• Attempt to inject various payloads to test for common vulnerabilities like SQL injection or XSS.

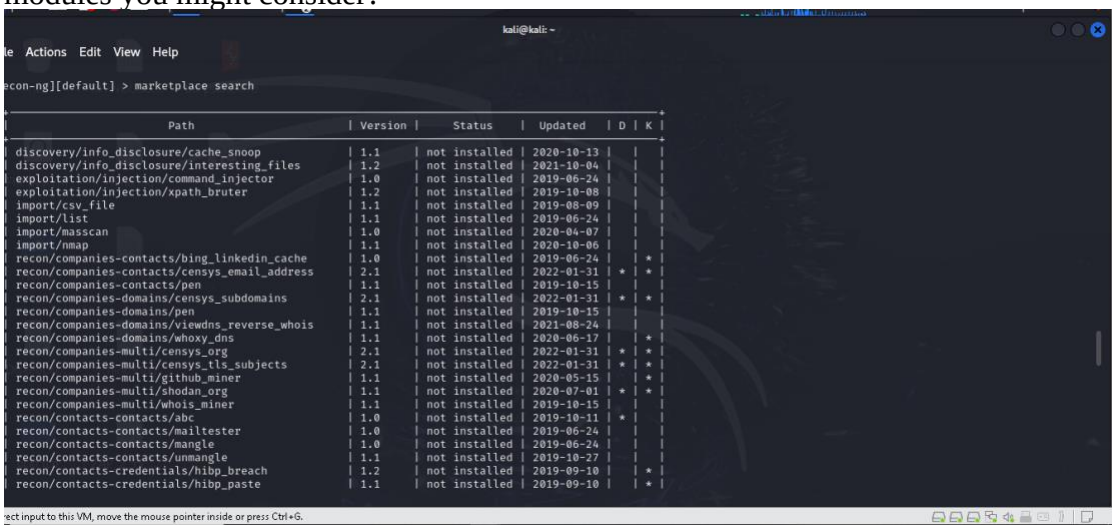


**Exercise 7:**
• Document any successful injections or errors encountered during fuzzing. What techniques were effective?

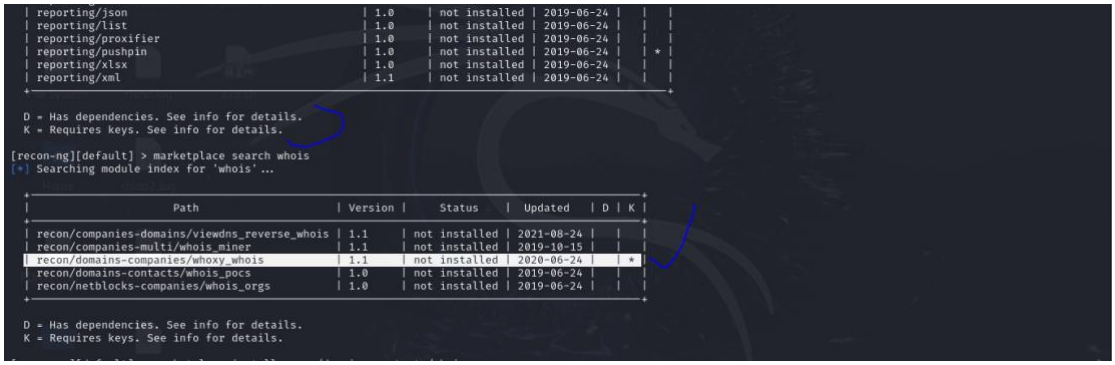
**Exercise 8:**
• Prepare a report detailing the vulnerabilities discovered, your methodology, and recommendations for securing the application.
**Conclusion**:

## Lab 9:

**Information Gathering with Recon-ng and Shodan**
**Step 1: Setting Up Recon-ng**
**Exercise 1:**
**List available modules in Recon-ng:**
• List the modules that can be used for domain reconnaissance. What are some key modules you might consider?




**Step 2: Using Recon-ng for Information Gathering1.Adding a Domain:**
**2.Running Modules:**
Use the whois module to gather registration information:
• Use recon/domains-hosts/whoisRun
• Explore other modules for gathering information such as social_media, contacts, etc.
**Exercise 2:**
• Document the registration details obtained from the whois module. What information did you find useful?




**3.Automating Data Gathering:**

Use additional modules for automated data collection, such as:

• Use recon/hosts-hosts/resolve

• Run

**Exercise 3:**

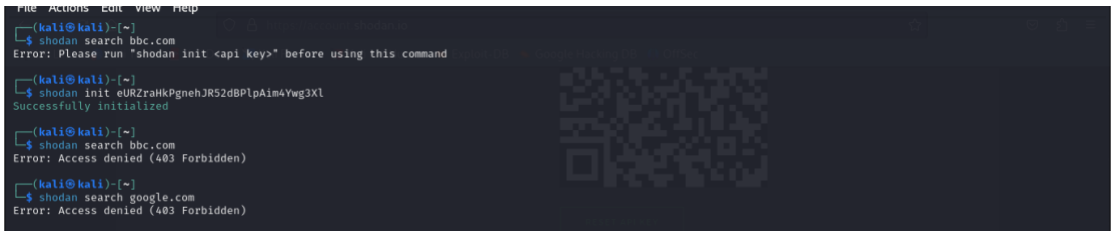
• What new information was discovered about the target domain? List the subdomains or IP addresses obtained.


**Step 3: Setting Up Shodan**

**Exercise 4:**

• Verify that your API key is working by running:

• Shodan info

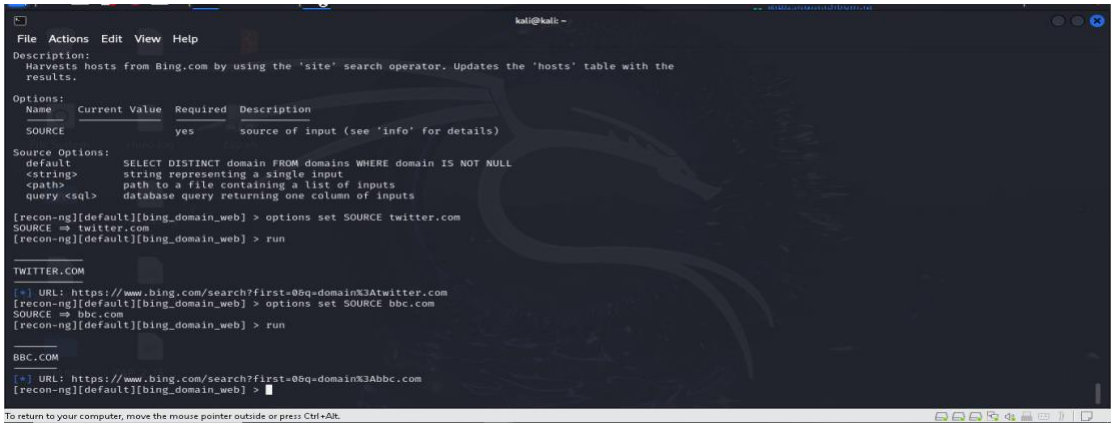



**Step 4: Using Shodan for Device Discovery1.Searching for Devices:**

**Exercise 5:**

**Shodan search example.com**

• What devices were discovered related to the target domain? Provide a brief description of the findings.




**2.Advanced Searches**:

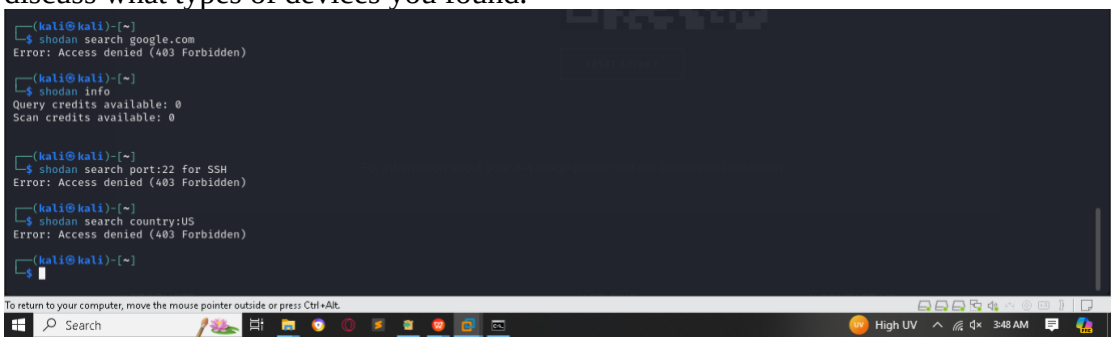
Utilize advanced search filters, such as:

• Port: Find devices on specific ports (e.g., port:22 for SSH).

• Country: Limit searches to specific countries (e.g., country:US).

**Exercise 6:**

• Perform an advanced search using two different filters. Document the results and discuss what types of devices you found.




**Step 5: Analyzing and Reporting Findings**

**1.Combining Data:**

• Compare the information gathered from Recon-ng and Shodan. Identify overlaps and unique findings.

**Exercise 7:**

In your report, outline the methodologies used, tools employed, and key insights. Discuss how this information could be useful in a penetration testing engagement.

## <u>Lab 10:</u>

**DNS Query Tools and SMB Enumeration**

**Step 1: DNS Queries with nslookup, host, and dig**

**Exercise 1:**

**nslookup google.com**

• What information did you obtain from the nslookup command? Document the IP addresses and any additional records retrieved.

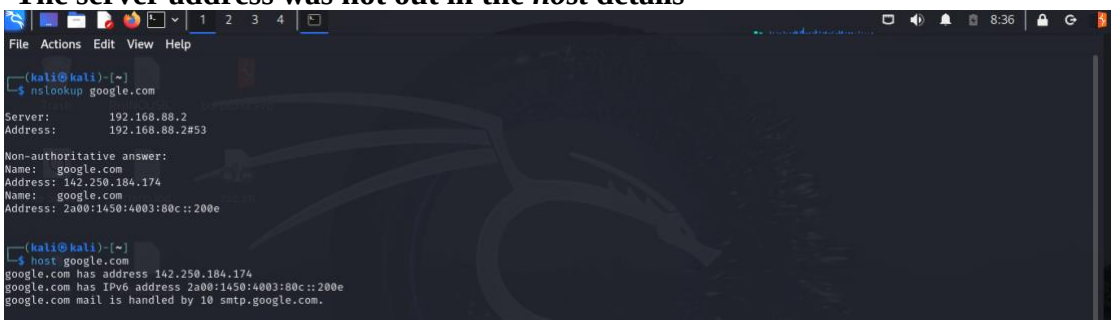
**IPv4 Address: 142.250.184.174**

**IPv6 Address: 2a00:1450:4003:80c::200e**

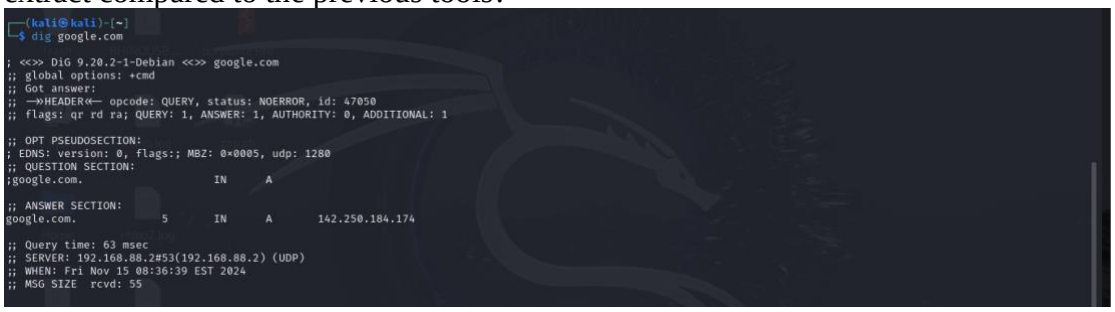
**Name: google.com**

**Server address: 192.168.88.2#53**

**Exercise 2:•**
**host  google.com**
**google.com has address 142.250.184.174**
**google.com has IPv6 address 2a00:1450:4003:80c::200e**
**google.com mail is handled by 10 smtp.google.com.**
What differences did you observe?
**-The server address was not out in the *host* details**

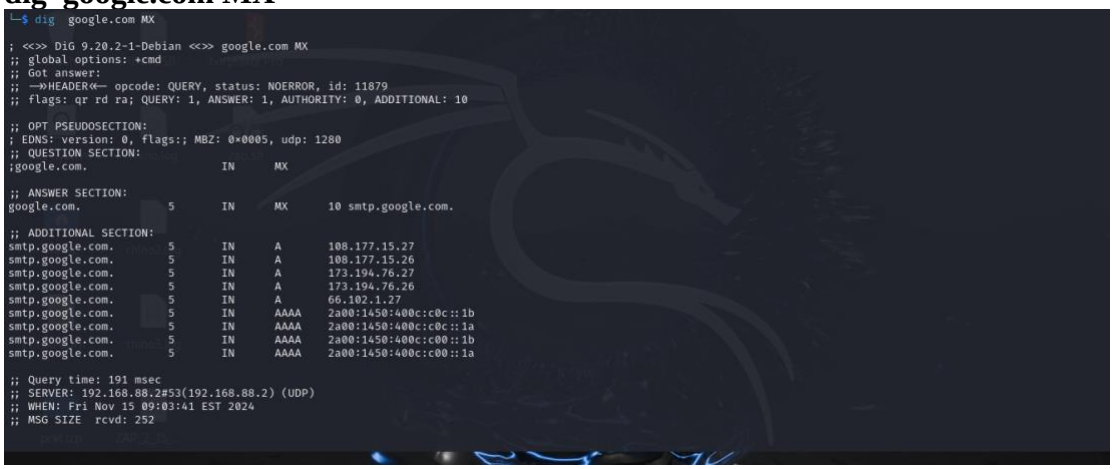



**Exercise 3:**
**dig google.com**
• Analyze the output of the dig command. What additional information can you extract compared to the previous tools?




**There are some additional options like:**
**HEADER: opcode, status, id,flags,  QUERY, ANSWER, AUTHORITY, ADDITIONAL.**
**OPT PSEUDOSECTION:EDNS, version, flags, MBZ, udp:**
**QUESTION SECTION,**
**ANSWER SECTION:Query time, SERVER, (UDP), WHEN, MSG SIZE rcvd.**

**Exercise 4:**
**dig  google.com MX**



**dig  google.com TXT**

```
┌──(kali㉿kali)-[~]
└─$ dig google.com TXT

; <<>> DiG 9.20.2-1-Debian <<>> google.com TXT
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36288
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.                    IN      TXT

;; ANSWER SECTION:
google.com.             5       IN      TXT     "globalsign-smime-dv=CDYX+XFHUw2wml6/Gb8+59BsH31KzUr6c1l2BPvqKX8="
google.com.             5       IN      TXT     "docusign=05958488-4752-4ef2-95eb-aa7ba8a3bd0e"
google.com.             5       IN      TXT     "docusign=1b0a6754-49b1-4db5-8540-d2c12664b289"
google.com.             5       IN      TXT     "apple-domain-verification=30afIBcvSuDV2PLX"
google.com.             5       IN      TXT     "google-site-verification=TV9-DBe4R80X4v0M4U_bd_J9cpOJM0nikft0jAgjmsQ"
google.com.             5       IN      TXT     "facebook-domain-verification=22rm551cu4k0ab0bxsw536tlds4h95"
google.com.             5       IN      TXT     "MS=E4A68B9AB2BB9670BCE15412F62916164C0B20BB"

;; Query time: 331 msec
;; SERVER: 192.168.88.2#53(192.168.88.2) (UDP)
;; WHEN: Fri Nov 15 09:05:58 EST 2024
;; MSG SIZE  rcvd: 485
```

• What did you learn from querying different record types? How can this information be useful in a penetration test?

**dig  google.com MX** command reveals mail exchange records, identifying mail servers and potential entry points
**dig  google.com TXT** command reveals text records containing sensitive information such as ; SDP, DKIM, and DMARC.
***Both the two helps in enumerate domain infrastructure, identify vulnerability and information gathering.***

**Step 2: SMB Enumeration with enum4linux**
**sudo apt install enum4linux**

**Exercise 5:**

```
┌──(kali㉿kali)-[~]
└─$ enum4linux -a 142.250.184.174
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Fri Nov 15 09:18:49 2024

 ========================== ( Target Information ) ==========================

Target ........... 142.250.184.174
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ===================== ( Enumerating Workgroup/Domain on 142.250.184.174 ) =====================


[E] Can't find workgroup/domain


 ============================ ( Nbtstat Information for 142.250.184.174 ) ============================

Looking up status of 142.250.184.174
No reply from 142.250.184.174

 ========================== ( Session Check on 142.250.184.174 ) ==========================


[E] Server doesn't allow session using username '', password '',  Aborting remainder of tests.
```

**enum4linux -a 142.250.184.174**
- What information did you gather about the target system?
**Target ..... 142.250.184.174**
**RID Range ..... 500-550,1000-1050**
**Username ..... ''**
**Password ..... ''**
**Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none**
-Document the shares, users, and any other relevant details found.

**(Enumerating Workgroup/Domain on 142.250.184.174)**
[E] Can't find workgroup/domain

**(Nbtstat Information for 142.250.184.174)**

Looking up status of 142.250.184.174
No reply from 142.250.184.174

**( Session Check on 142.250.184.174 )**
[E] Server doesn't allow session using username ", password ".  Aborting remainder of tests.

**Exercise 6:**
**enum4linux -s 142.250.184.174 # Lists shares**

```
┌──(kali㉿kali)-[~]
└─$ enum4linux -s 142.250.184.174
enum4linux v0.9.1 (http://labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar
functionality to enum.exe (formerly from www.bindview.com).  Some additional
features such as RID cycling have also been added for convenience.

Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
    -U        get userlist
    -M        get machine list*
    -S        get sharelist
    -P        get password policy information
    -G        get group and member list
    -d        be detailed, applies to -U and -S
    -u user   specify username to use (default "")
    -p pass   specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f

Additional options:
    -a        Do all simple enumeration (-U -S -G -P -r -o -n -i).
              This option is enabled if you don't provide any other options.
    -h        Display this help message and exit
    -r        enumerate users via RID cycling
    -R range  RID ranges to enumerate (default: 500-550,1000-1050, implies -r)
    -K n      Keep searching RIDs until n consective RIDs don't correspond to
```

**enum4linux -u 142.250.184.174 # Lists users**

```
┌──(kali㉿kali)-[~]
└─$ enum4linux -u 142.250.184.174
enum4linux v0.9.1 (http://labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar
functionality to enum.exe (formerly from www.bindview.com).  Some additional
features such as RID cycling have also been added for convenience.

Usage: ./enum4linux.pl [options] ip

Options are (like "enum"):
    -U        get userlist
    -M        get machine list*
    -S        get sharelist
    -P        get password policy information
    -G        get group and member list
    -d        be detailed, applies to -U and -S
    -u user   specify username to use (default "")
    -p pass   specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f

Additional options:
    -a        Do all simple enumeration (-U -S -G -P -r -o -n -i).
              This option is enabled if you don't provide any other options.
    -h        Display this help message and exit
    -r        enumerate users via RID cycling
    -R range  RID ranges to enumerate (default: 500-550,1000-1050, implies -r)
    -K n      Keep searching RIDs until n consective RIDs don't correspond to
              a username.  Impies RID range ends at 999999. Useful
```

• Compare the results obtained from enum4linux with your findings from DNS queries. What insights can you gain about the target network?

**-Observed that -a enumerate all the options at once, while the -s & -u reveals the specific data type, also have the same output.**
**-a, -s -u reveal insights into share permissions, user accounts, password policies, OS versions, and  network architecture**.