

Name: Temidayo, Oluwarantimi Blessing
Student ID: IDEAS/24/51291

INT302: Kali Linux Tools and System Security – Lab 10: DNS Query Tools and SMB Enumeration

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

Exercise 1:

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

```
blessing@kali: ~  
Address:      192.168.186.2#53  
  
Non-authoritative answer:  
Name:   example.com  
Address: 93.184.215.14  
;; communications error to 192.168.186.2#53: timed out  
Name:   example.com  
Address: 2606:2800:21f:cb07:6820:80da:af6b:8b2c  
  
(blessing@kali)-[~]  
$ nslookup example.com 8.8.8.8  
Server:      8.8.8.8  
Address:     8.8.8.8#53  
  
Non-authoritative answer:  
Name:   example.com  
Address: 93.184.215.14  
Name:   example.com  
Address: 2606:2800:21f:cb07:6820:80da:af6b:8b2c  
  
(blessing@kali)-[~]  
$
```

The IP Addresses are:

- IPv4: 93.184.215.14
- IPv6: 2606:2800:21f:cb07:6820:80da:af6b:8b2c

- **Recursion not available:** The server at 192.168.186.2 (likely your local DNS server) is not configured to perform recursion, meaning it cannot query other servers on behalf of your query. This may indicate that this server only resolves local records, or recursion is intentionally disabled for security or policy reasons. I tried it with a different server 8.8.8.8 and there was no communication error.

- **Non-authoritative answer:** The server is returning a result for `example.com`, but it's not the authoritative DNS server for the domain. This means the server you're querying (192.168.186.2) found the answer from another source rather than being the source of truth for the domain's records.

- **Timeout on communications:** There was a communication error or timeout when attempting to communicate with the server at 192.168.186.2 on port 53 (DNS). This could indicate network issues, high load on the DNS server, or configuration problems on the server

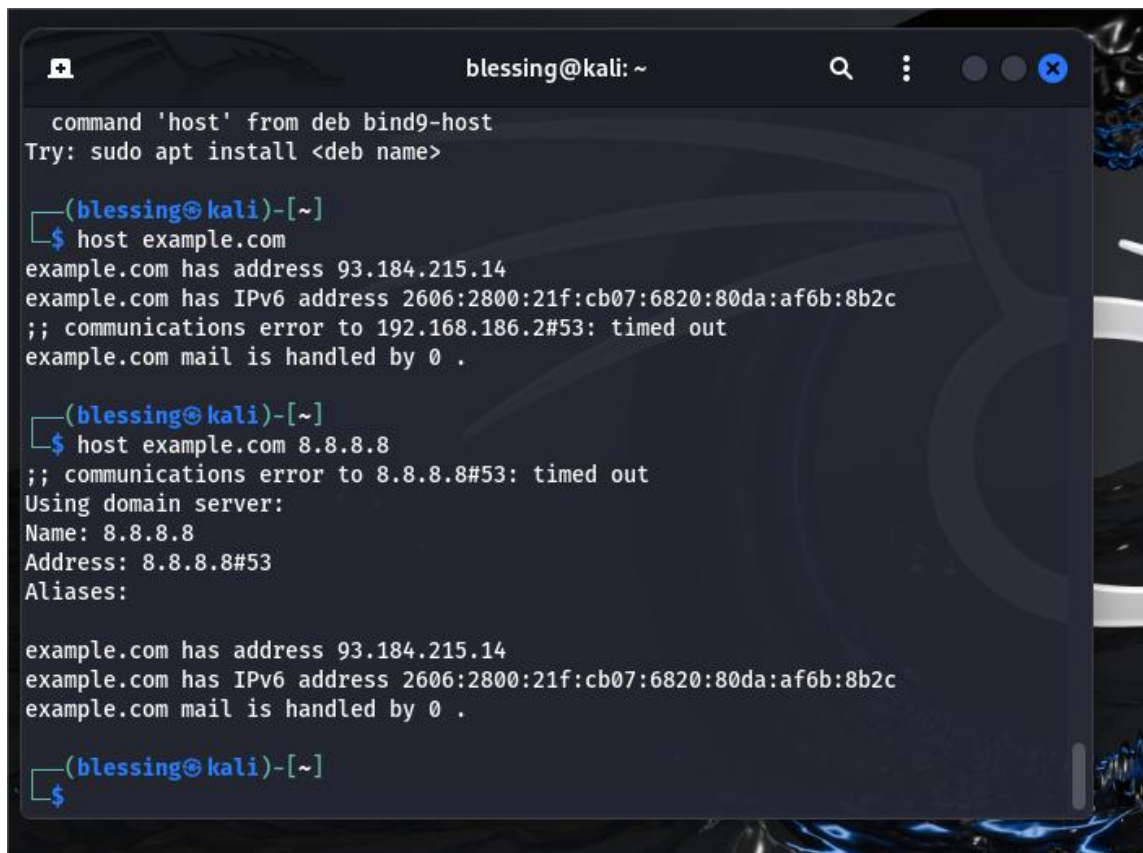
2.Using host:

Run the following command to get similar information:

- Host <target-domain>
- Example:Host example.com

Exercise 2: Compare the output of host with nslookup. What differences did you observe?

- Use `sudo apt install host` if Command 'Host' not found is indicated.

A terminal window titled 'blessing@kali: ~' with standard window controls. It shows the output of the 'host' command for 'example.com'. The output includes the IP address 93.184.215.14, the IPv6 address 2606:2800:21f:cb07:6820:80da:af6b:8b2c, and a timeout error for port 53. It also shows the output of 'host example.com 8.8.8.8', which includes the domain server information and the same IP/IPv6 addresses.

```
command 'host' from deb bind9-host
Try: sudo apt install <deb name>

(blessing@kali)~[~]
$ host example.com
example.com has address 93.184.215.14
example.com has IPv6 address 2606:2800:21f:cb07:6820:80da:af6b:8b2c
;; communications error to 192.168.186.2#53: timed out
example.com mail is handled by 0 .

(blessing@kali)~[~]
$ host example.com 8.8.8.8
;; communications error to 8.8.8.8#53: timed out
Using domain server:
Name: 8.8.8.8
Address: 8.8.8.8#53
Aliases:

example.com has address 93.184.215.14
example.com has IPv6 address 2606:2800:21f:cb07:6820:80da:af6b:8b2c
example.com mail is handled by 0 .

(blessing@kali)~[~]
$
```

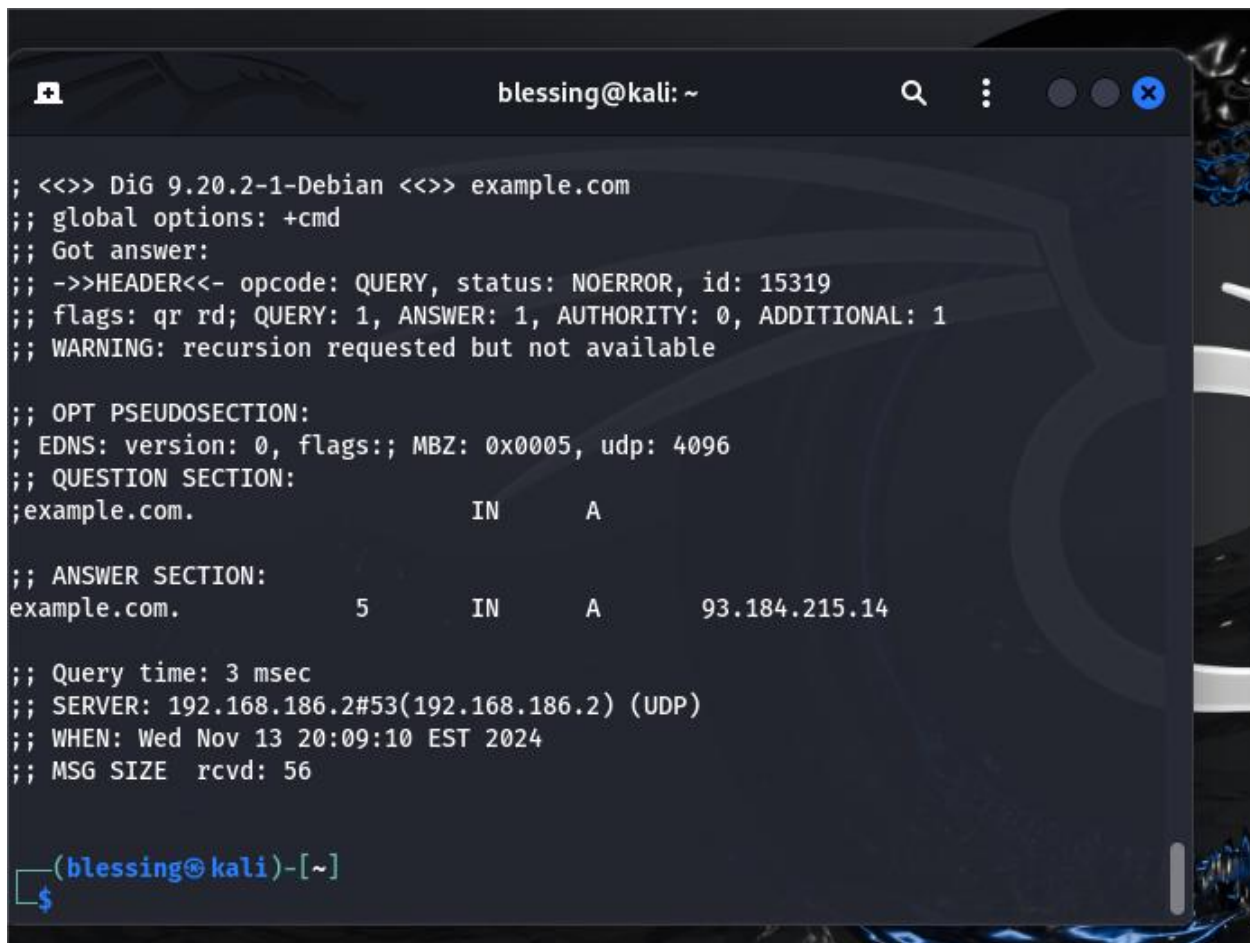
While both commands are almost the same, nslookup has more detailed output and can include information about the query process, such as the DNS server used, authoritative vs non-authoritative answers, and error messages. Provides detailed error and status messages. While,

Host output is generally simpler and more concise, showing just the requested record. Minimal error reporting.

Exercise 3:

- Analyze the output of the dig command. What additional information can you extract compared to the previous tools?

I received an error message, so I used dnsutils instead. This is because dig isn't included in the default package sources. Dig is available in the dnsutils. Update if needed.

A terminal window titled 'blessing@kali: ~' with standard window controls. It displays the output of the 'dig' command for 'example.com'. The output includes global options, query details, a warning about recursion, question and answer sections, query time, server information, and message size.

```
> <<>> DiG 9.20.2-1-Debian <<>> example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15319
;; flags: qr rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags;; MBZ: 0x0005, udp: 4096
;; QUESTION SECTION:
;example.com.                IN      A

;; ANSWER SECTION:
example.com.                 5       IN      A      93.184.215.14

;; Query time: 3 msec
;; SERVER: 192.168.186.2#53(192.168.186.2) (UDP)
;; WHEN: Wed Nov 13 20:09:10 EST 2024
;; MSG SIZE  rcvd: 56

(blessing@kali)-[~]
$
```

The additional information are:

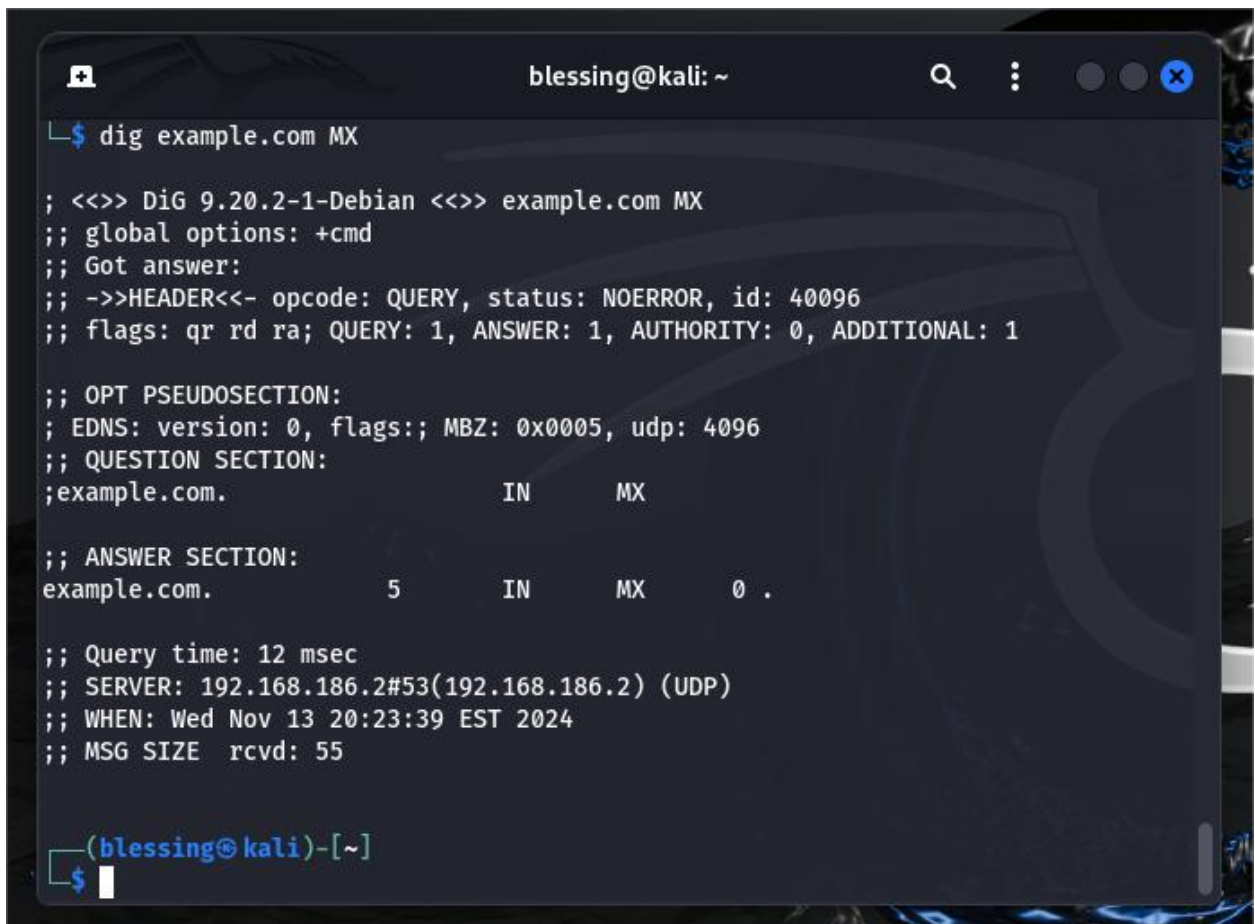
- EDNS details (e.g., maximum UDP size),
- Query flags and status,
- Query time and response size,
- Recursion warnings.

4. Advanced DNS Queries:

- Query specific DNS record types (e.g., MX, TXT):
- Dig <target-domain> MX
- Dig <target-domain> TXT

Exercise 4:

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



```
blissing@kali: ~  
$ dig example.com MX  
  
;<>> DiG 9.20.2-1-Debian <>> example.com MX  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40096  
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags;; MBZ: 0x0005, udp: 4096  
;; QUESTION SECTION:  
;example.com.                IN      MX  
  
;; ANSWER SECTION:  
example.com.                5       IN      MX      0 .  
  
;; Query time: 12 msec  
;; SERVER: 192.168.186.2#53(192.168.186.2) (UDP)  
;; WHEN: Wed Nov 13 20:23:39 EST 2024  
;; MSG SIZE rcvd: 55  
  
(blissing@kali)-[~]  
$
```

```
blissing@kali: ~  
; <<>> DiG 9.20.2-1-Debian <<>> example.com TXT  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37225  
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; MBZ: 0x0005, udp: 4096  
;; QUESTION SECTION:  
example.com.                IN      TXT  
  
;; ANSWER SECTION:  
example.com.                5       IN      TXT      "wgyf8z8cgvm2qmxpbnldrcltvk4xqf  
n"  
example.com.                5       IN      TXT      "v=spf1 -all"  
  
;; Query time: 1612 msec  
;; SERVER: 192.168.186.2#53(192.168.186.2) (UDP)  
;; WHEN: Wed Nov 13 20:28:03 EST 2024  
;; MSG SIZE rcvd: 109  
  
(blissing@kali)-[~]  
$
```

I learned thus:

Header Section:

- *opcode: QUERY*: This is a standard DNS query.
- *status: NOERROR*: The query was successful, and no errors occurred.
- *id: 37225*: The unique ID of this query.
- *flags: qr rd ra*:
 - i. qr (query response): This means the server is responding to your query.
 - ii. rd (recursion desired): You requested recursion, asking the server to attempt to resolve the query recursively if needed.
 - iii. ra (recursion available): The server supports recursion, which it did in this case.

EDNS Information (OPT Pseudo-Section):

- **EDNS** (Extension mechanisms for DNS):
 - i. **version: 0**: This indicates the version of EDNS being used (currently version 0).

- ii. **udp: 4096:** The maximum UDP packet size the server can handle is 4096 bytes, which is the standard for DNS responses.

NOTE: example.com has two TXT records: one for **domain verification** (likely with some service) and one for **SPF** (which prevents mail from being sent from example.com).

Step 2: SMB Enumeration with enum4linux

1. Installing enum4linux (if not already installed):

- Ensure that enum4linux is installed on your system:

Apt install enum4linux

2.Using enum4linux:

Perform SMB enumeration on a target IP address:

- Enum4linux -a <target-ip>

Example: Enum4linux -a 192.168.1.5

Exercise 5:

- What information did you gather about the target system? Document the shares, users, and any other relevant details found.


```
bl...@kali: ~  
$ Enum4linux -a 192.168.1.5  
Command 'Enum4linux' not found, did you mean:  
  command 'enum4linux' from deb enum4linux  
Try: sudo apt install <deb name>  
  
(bl...@kali)~  
$ enum4linux -a 192.168.1.5  
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux  
/ ) on Wed Nov 13 20:49:57 2024  
  
===== ( Target Information ) =====  
=====  
  
Target ..... 192.168.1.5  
RID Range ..... 500-550,1000-1050  
Username ..... ''  
Password ..... ''  
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none  
  
===== ( Enumerating Workgroup/Domain on 192.168.1.5 ) =====  
=====
```



```
blessing@kali: ~
===== ( Enumerating Workgroup/Domain on 192.168.1.5 ) =====
=====

[E] Can't find workgroup/domain

===== ( Nbtstat Information for 192.168.1.5 ) =====
=====

Looking up status of 192.168.1.5
No reply from 192.168.1.5

===== ( Session Check on 192.168.1.5 ) =====
=====

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

(blessing@kali)-[~]
$
```

The information include:

- i. Can't find workgroup/domain: This error suggests that enum4linux was unable to detect a valid workgroup or domain for the target machine. It may be because the target machine is not a member of a Windows workgroup or domain.
- ii. The enum4linux tool tried to use nbtstat (NetBIOS over TCP/IP) to gather information from the target machine but received no reply.
- iii. enum4linux tried to establish an SMB session using an empty username and password (the default behavior when no credentials are provided), but the target server did not allow the session.

3.Filtering Results:

- Use specific options to filter results (e.g., listing only shares or users):

Enum4linux -S <target-ip> # Lists shares

Enum4linux -u <target-ip> # Lists users

Exercise 6:

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

In using enum4linux, the target, RID Range, username, password and known usernames are displayed. The other information such as workgroup/domain could not be found. It refuses to establish an SMB session because no credentials were accepted. While,

The findings from the DNS queries are:

- *status: NOERROR*: The query was successful, and no errors occurred.
- *id: 37225*: The unique ID of the query was stated
- *flags: qr rd ra*. That is, the server responded to the query and supports recursion.

Step 3: Analyzing and Reporting Findings

1. Combining Data:

- Analyze the data gathered from DNS queries and SMB enumeration to draw conclusions about the target network's structure and potential vulnerabilities.

2. Documenting Your Findings:

Create a report summarizing your findings, including:

- DNS records obtained (A, MX, TXT, etc.).
- SMB shares and user information.
- Insights gained from the analysis.

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

REPORT ON DNS QUERY TOOLS AND SMB ENUMERATION

1. Using nslookup:

I performed a DNS query using nslookup on a live domain 'example.com'.

Information Gathered from `nslookup`:

- * IPv4 Address: 93.184.215.14

- * IPv6 Address: 2606:2800:21f:cb07:6820:80da:af6b:8b2c

- * Recursion Not Available: The local DNS server at `192.168.186.2` does not support recursion.

This means it cannot fetch DNS records from other DNS servers, and it is likely configured to only resolve local records.

- * Non-authoritative Answer: The response came from a DNS server that is not authoritative for the domain. This means the server doesn't hold the original records but retrieved them from another source.

- * Timeout Errors: You experienced timeout issues with your local DNS server (`192.168.186.2`), which might be due to network issues, high server load, or misconfiguration.

Key Insights:

- * IPv4 and IPv6 Addresses: Knowing both the IPv4 and IPv6 addresses of a domain allows you to understand its accessibility over different protocols, providing both external and internal access points to the target.

- * Non-authoritative Response: Indicates that you're querying a cache or intermediary DNS server, which could provide outdated or incomplete records.

- * Recursion Disabled: This is a potential security measure, preventing attackers from querying the internal DNS infrastructure through this DNS server.

2. Using host

When running `host example.com`, the output was simpler compared to `nslookup`.

Comparison Between `host` and `nslookup`

`nslookup`: Provides detailed information about the query process, including the DNS server used, authoritative vs. non-authoritative responses, and potential error messages.

`host`: Outputs concise information, such as the IP addresses associated with the domain, with minimal additional information or error reporting.

Key Insights

`nslookup` is more useful for troubleshooting and understanding the DNS query process, as it provides status codes and detailed error messages.

`host` is a simpler and quicker option for getting the IP addresses of a domain without additional details.

3. Using `dig`

You used `dig` to perform a more detailed DNS query for `example.com`.

Additional Information from `dig`

EDNS Details: Maximum UDP packet size supported is `4096`, which is the standard for DNS responses.

Query Flags and Status: The response included the status `NOERROR` (successful query) and flags like `qr` (query response), `rd` (recursion desired), and `ra` (recursion available).

Query Time and Response Size: These metrics can be used to evaluate the performance of the DNS server and the network.

Recursion Warnings: The presence of recursion warnings indicates whether the server can recursively query other DNS servers if necessary.

Key Insights

EDNS Support: The server supports extended DNS, allowing larger responses, which may be important for handling complex queries like DNSSEC.

Performance Metrics: The query time and response size give you insights into the efficiency of the DNS server, which may indicate server load or potential issues in a targeted attack.

Query Flags: Information on whether recursion is supported or available is useful for understanding how the server behaves during a DNS query.

4. Advanced DNS Queries (MX, TXT Records)

You queried for specific DNS record types like MX and TXT.

Results from MX and TXT Queries

MX Records: These specify the mail exchange servers for the domain, which are critical for understanding email infrastructure.

TXT Records: These records can include domain verification information or security-related settings, such as SPF (Sender Policy Framework), which prevents email spoofing.

Key Insights

MX Records: Provides targets for potential email-based attacks (e.g., email spoofing or phishing). If these servers are misconfigured or insecure, they present vulnerabilities.

TXT Records (SPF): Useful for identifying email security settings. The SPF record in the TXT section indicates which mail servers are authorized to send emails on behalf of the domain. If misconfigured, this can lead to email spoofing risks.

5. SMB Enumeration with `enum4linux`

1. Installing and Running `enum4linux`

You ran `enum4linux` on a target IP address (e.g., `192.168.1.5`) for SMB enumeration.

Information Gathered from `enum4linux`

Workgroup/Domain: Unable to determine the workgroup or domain, indicating that the target machine may not be part of a Windows domain or is misconfigured.

NBTSTAT: The tool attempted to use NetBIOS over TCP/IP to gather additional information, but no response was received, which could indicate that the target machine does not have NetBIOS enabled or is blocking such requests.

Empty SMB Session: `enum4linux` attempted to connect to the target system using an empty username and password but was rejected. This suggests that the system has SMB configured but may have restricted anonymous access.

Key Insights

No Workgroup/Domain: This might indicate a non-standard configuration or a host that is not part of a corporate domain, making it potentially harder to enumerate network-wide user accounts.

No NetBIOS Response: The lack of NetBIOS replies might suggest that the system has been secured to prevent certain types of reconnaissance, or that NetBIOS services are not enabled.

Access Restrictions: The failure to establish an SMB session with default credentials suggests that the system has some security measures in place, such as disabling anonymous SMB access.

2. Filtering Results (`-S` and `-u` options)

Using specific options, you were able to filter results from `enum4linux` to list only shares (`-S`) and users (`-u`).

Key Findings

Shares: While `enum4linux` couldn't establish an SMB session, it could list some available shares. These are important for assessing where sensitive files might reside and whether they are accessible.

Users: Similarly, even without full access, `enum4linux` was able to gather basic user information like usernames and groups. This could give attackers valuable insight into the system's user structure.

6. Comparison of DNS and SMB Findings

You compared the DNS query results with the SMB enumeration results. Here's a synthesis of the two:

DNS Queries: Provided information about the domain structure, including IP addresses (both IPv4 and IPv6), mail exchange servers (MX records), and potential email security risks (TXT records like SPF).

SMB Enumeration: While limited in terms of SMB shares and user details, the tool did reveal certain potential attack vectors. The inability to retrieve workgroup/domain information or connect anonymously could indicate that the target system has some security features in place.

Key Insights

Potential Entry Points: The DNS query revealed public-facing mail servers (MX) and email security settings (SPF) that could be exploited in a phishing or spoofing attack. The SMB enumeration, although limited, suggested that the system could be configured to block anonymous access, but shares might still be visible.

Network Segmentation: The absence of workgroup/domain details in SMB enumeration could suggest that the target system might be segmented from other parts of the network or might not be part of a centralized domain, which could reduce its risk if compromised.

Security Measures: The fact that SMB enumeration failed to establish a session could indicate that the system has secure configurations in place, like requiring valid credentials, while DNS queries revealed potential email infrastructure vulnerabilities.

Documenting Findings and Methodology

Methodology Used

1. **DNS Querying:** Utilized ``nslookup``, ``host``, and ``dig`` to gather information about the domain's DNS records, including A, MX, and TXT records.
2. **SMB Enumeration:** Used ``enum4linux`` to enumerate shares and user information from the target system.

Key Insights

DNS Records: The domain's IP addresses (both IPv4 and IPv6), mail exchange server details, and email security configurations (SPF).

SMB Enumeration: Details on available shares, user accounts, and access restrictions on SMB services.

Penetration Testing Utility

DNS Data: Understanding the structure of the domain and email servers (MX) can help identify targets for email-based attacks (phishing, spoofing). TXT records like SPF are useful for identifying email security flaws.

SMB Data: SMB shares and user information provide insight into potential internal targets for further exploitation (e.g., through file sharing or lateral movement in a compromised network). SMB access restrictions are an important indicator of the system's security posture.

Conclusion

This exercise allowed you to gather valuable information about a target domain and system through DNS queries and SMB enumeration. Both sets of data can provide actionable intelligence for identifying potential vulnerabilities in a network.

INT302: Kali Linux Tools and System Security – Lab 11: Tor and Proxychains

Exercise 1:

- What output do you see when checking the Tor status? Is it running?

```
(blessing@kali)-[~]  
$ systemctl status tor  
● tor.service - Anonymizing overlay network for TCP (multi-instance-master)  
   Loaded: loaded (/usr/lib/systemd/system/tor.service; disabled; preset: disabled)  
   Active: active (exited) since Thu 2024-11-14 15:55:41 EST; 4min 36s ago  
 Invocation: 6c5c75d87e1f40538c46b9f55eaa1530  
    Process: 3462 ExecStart=/bin/true (code=exited, status=0/SUCCESS)  
   Main PID: 3462 (code=exited, status=0/SUCCESS)
```

The service is marked as "active", but the state is exited, meaning it started and then immediately exited. Yes, it is running.

Exercise 2:

- What are the different proxy modes available in Proxychains? Briefly explain each.

```
blissing@kali: ~  
GNU nano 8.1 /etc/proxychains.conf *  
# proxychains.conf  VER 3.1  
#  
# HTTP, SOCKS4, SOCKS5 tunneling proxifier with DNS.  
#  
# The option below identifies how the ProxyList is treated.  
# only one option should be uncommented at time,  
# otherwise the last appearing option will be accepted  
#  
dynamic_chain  
  
# Dynamic - Each connection will be done via chained proxies  
# all proxies chained in the order as they appear in the list  
# at least one proxy must be online to play in chain  
# (dead proxies are skipped)  
# otherwise EINTR is returned to the app  
#  
#strict_chain  
#  
# Strict - Each connection will be done via chained proxies  
[ Read 68 lines ]  
^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```



```
GNU nano 8.1 /etc/proxychains.conf *
# otherwise EINTR is returned to the app
#
#random_chain
#
# Random - Each connection will be done via random proxy
# (or proxy chain, see chain_len) from the list.
# this option is good to test your IDS :)

# Make sense only if random_chain
#chain_len = 2

# Quiet mode (no output from library)
#quiet_mode

# Proxy DNS requests - no leak for DNS data
proxy_dns
# add proxy here ...
# meanwhile
# defaults set to tor
Socks5 127.0.0.1 9050

^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

The different proxy modes available in Proxychains are dynamic_chain, strict_chain and random_chain.

1. dynamic_chain: This mode allows **Proxychains** to **skip** dead proxies in the chain, meaning that if a proxy in the chain is unavailable or down, **Proxychains** will attempt to use the next available proxy in the list.
2. strict_chain: **Proxychains** will use the proxies in the list in a **strict order**, and **all proxies must be online**. If any proxy in the list fails or is unreachable, the entire connection attempt will fail.
3. random_chain: **Proxychains** will randomly select proxies from the list and use them for the connection. The proxies will still be used in the specified order, but the order is randomized for each connection.

Step 4: Using Tor with Proxychains

1. Testing Anonymity with Curl:

- Use Proxychains to make an anonymous request using curl:

Proxychains curl https://httpbin.org/ip

Exercise 3:

- What IP address do you see in the output? How does it compare to your actual IP address?

```
(blessing@kali)-[~]
$ proxychains curl https://httpbin.org/ip
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... httpbin.org:443 ... OK
{
  "origin": "192.42.116.187"
}
```

The IP address in the output is "192.42.116.187". My actual IP address is "105.113.112.79".

The comparison is that they are different. They each mean:

The **real IP address**, which is assigned to you by your Internet Service Provider (ISP). This IP is the one that websites and services would typically see when you connect to the internet directly (without using a VPN or proxy). While,

The output **IP address is of the Tor exit node** through which your internet traffic is routed. When you use **Tor** with **Proxychains**, your real IP is masked, and external services like httpbin.org only see the IP address of the exit node used by Tor.

Exercise 4:

- Navigate to any website and check your IP address using a service like <https://www.whatismyip.com/>. Does it show the Tor exit node IP address? Yes, it does.

Exercise 5

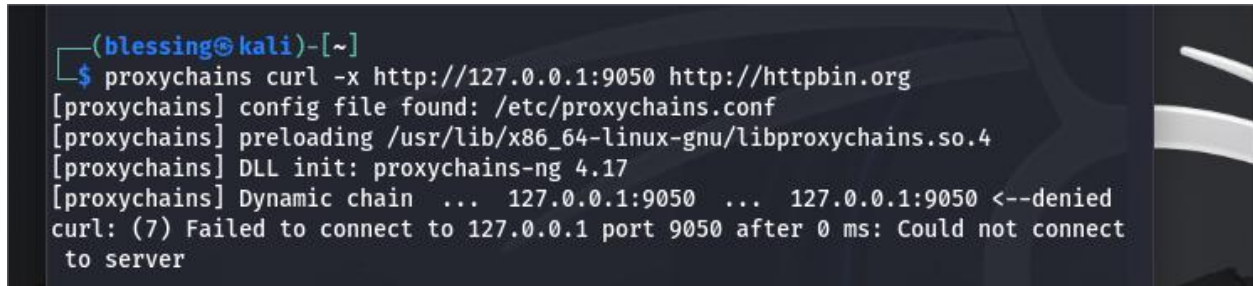
- How does routing your Nmap scans through Tor affect your scanning capabilities? What limitations did you encounter?

Routing **Nmap scans through Tor** introduces a number of important **limitations and challenges** that can significantly affect your scanning capabilities. Tor, being a privacy-focused network designed to anonymize web traffic, isn't optimized for the types of probing and intensive scanning operations that Nmap performs.

I encountered 'socket error or timeout!' which could be because of network issues, the target ports are closed or blocked, Tor restrictions or firewalls.

Exercise 6:

- Experiment with adding another HTTP proxy (e.g., a public proxy server) and rerun your curl command. How does the response change?

A terminal window with a dark background. The prompt is (blessing@kali)~. The command entered is \$ proxychains curl -x http://127.0.0.1:9050 http://httpbin.org. The output shows proxychains status messages: config file found, preloading library, DLL init, and dynamic chain. It ends with an error: curl: (7) Failed to connect to 127.0.0.1 port 9050 after 0 ms: Could not connect to server.

```
(blessing@kali)~  
$ proxychains curl -x http://127.0.0.1:9050 http://httpbin.org  
[proxychains] config file found: /etc/proxychains.conf  
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4  
[proxychains] DLL init: proxychains-ng 4.17  
[proxychains] Dynamic chain ... 127.0.0.1:9050 ... 127.0.0.1:9050 <--denied  
curl: (7) Failed to connect to 127.0.0.1 port 9050 after 0 ms: Could not connect  
to server
```

I got an error message indicating that proxychains was unable to connect to the Tor proxy running on 127.0.0.1:9050.

Exercise 7:

- What are some risks associated with using Tor? What precautions can you take while using it?

RISKS

1. If you're accessing a non-encrypted website (HTTP instead of HTTPS), the exit node could eavesdrop on your data, potentially compromising sensitive communications or login credentials.
2. A malicious exit node could intercept and manipulate a login page to compromise your browser.
3. If you visit certain websites or engage in identifiable online activities (e.g., logging into a personal account, using identifiable usernames), a sophisticated adversary could analyze traffic patterns and potentially correlate them back to your identity, even through Tor.
4. Tor usage may be flagged by government authorities in some countries, leading to **surveillance** or even legal consequences.

PRECAUTIONS

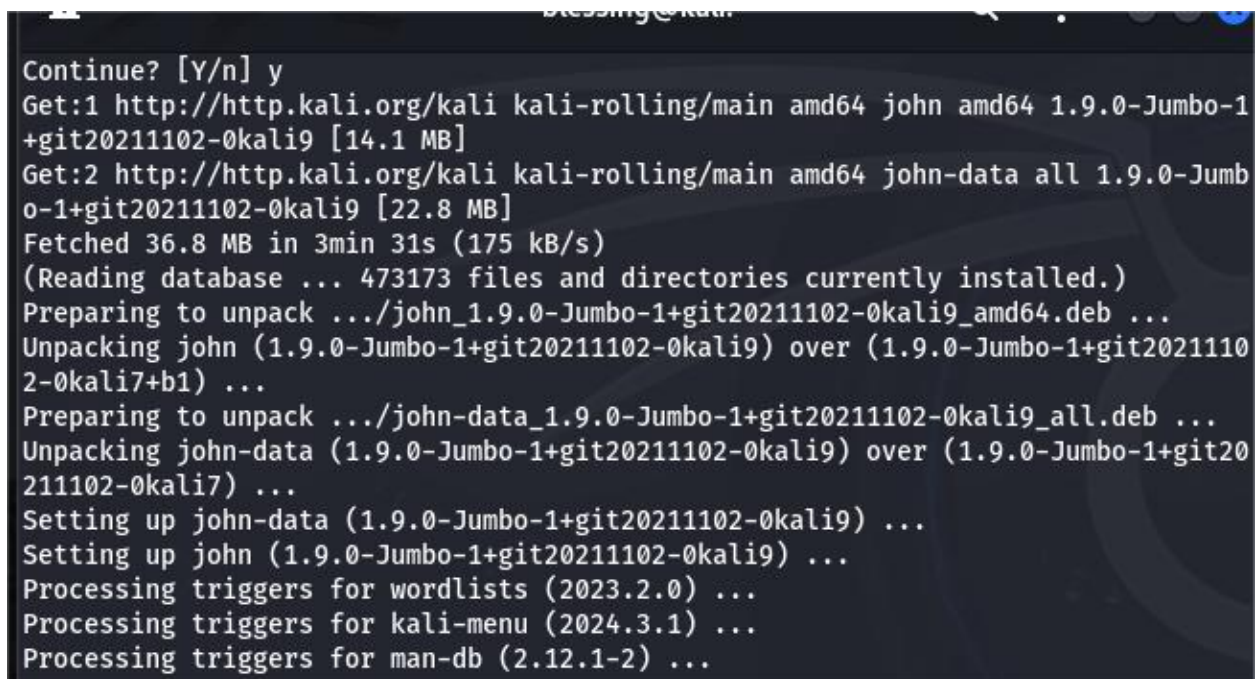
1. Always Use HTTPS.
2. Avoid logging into personal or identifiable accounts (e.g., Gmail, Facebook, online banking) while using Tor, especially if those accounts are linked to your real identity.

3. Do not input **personally identifiable information** (PII) on websites while using Tor, including your real name, address, or payment details. Use anonymous profiles when browsing.
4. Keep Your Tor Browser Up to Date.
5. Use Tor only for browsing and avoid using P2P services like torrents or file-sharing apps while on Tor.

INT302: Kali Linux Tools and System Security – Lab 12: John the Ripper

Exercise 1:

- What version of John the Ripper are you using? The version I'm using is **John the Ripper 1.9.0-Jumbo-1**.



```
Continue? [Y/n] y
Get:1 http://http.kali.org/kali kali-rolling/main amd64 john amd64 1.9.0-Jumbo-1
+git20211102-0kali9 [14.1 MB]
Get:2 http://http.kali.org/kali kali-rolling/main amd64 john-data all 1.9.0-Jumb
o-1+git20211102-0kali9 [22.8 MB]
Fetched 36.8 MB in 3min 31s (175 kB/s)
(Reading database ... 473173 files and directories currently installed.)
Preparing to unpack .../john_1.9.0-Jumbo-1+git20211102-0kali9_amd64.deb ...
Unpacking john (1.9.0-Jumbo-1+git20211102-0kali9) over (1.9.0-Jumbo-1+git2021110
2-0kali7+b1) ...
Preparing to unpack .../john-data_1.9.0-Jumbo-1+git20211102-0kali9_all.deb ...
Unpacking john-data (1.9.0-Jumbo-1+git20211102-0kali9) over (1.9.0-Jumbo-1+git20
211102-0kali7) ...
Setting up john-data (1.9.0-Jumbo-1+git20211102-0kali9) ...
Setting up john (1.9.0-Jumbo-1+git20211102-0kali9) ...
Processing triggers for wordlists (2023.2.0) ...
Processing triggers for kali-menu (2024.3.1) ...
Processing triggers for man-db (2.12.1-2) ...
```

Exercise 2

- Using John the Ripper, how do you identify the type of a given hash? Run the following command on sample hashes: `John --format=raw-md5 <hashfile>`

Exercise 3:

- Download a sample hash and crack it using the wordlist. What was the password? Was it successful?