

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

Lab Overview

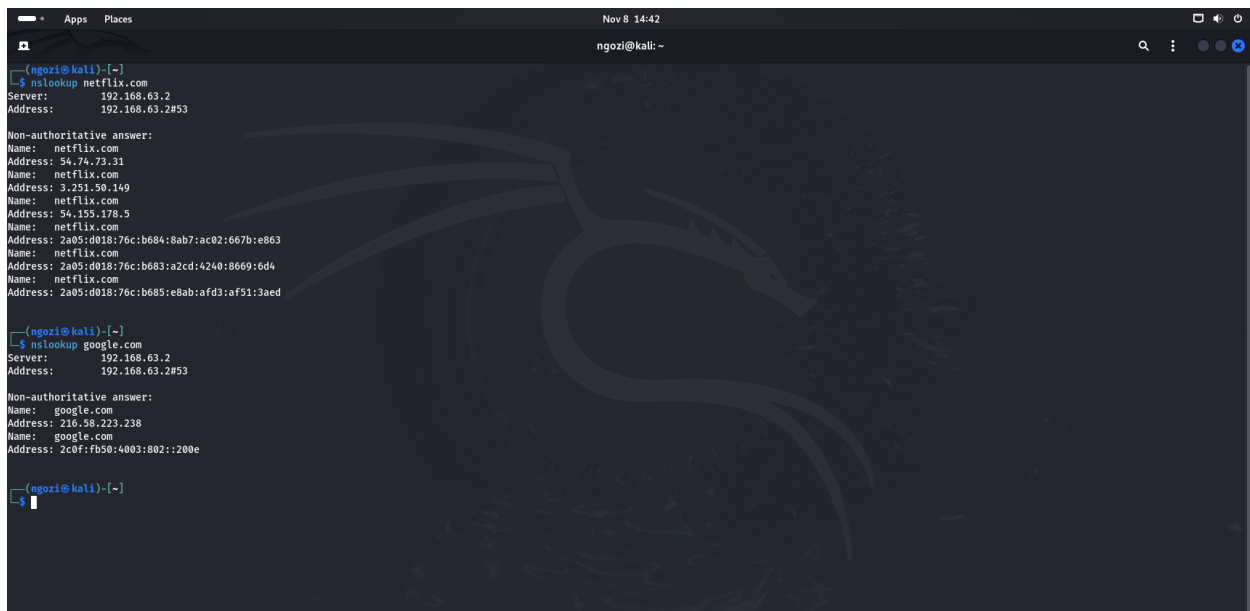
In this lab, participants will explore essential DNS query tools (nslookup, host, and dig) and learn how to enumerate SMB shares and users using enum4linux. This lab will help students understand how to gather information about domains, hosts, and networked services. Lab Objectives By the end of this lab, you will be able to: 1. Perform DNS queries using various tools to gather domain information. 2. Use enum4linux to enumerate SMB shares and users on a target system. 3. Analyze the results to inform penetration testing efforts.

Lab Steps 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. _____

Answer:



```
(ngozi@kali)-[~]
└─$ nslookup netflix.com
Server:      192.168.63.2
Address:     192.168.63.2#53

Non-authoritative answer:
Name:   netflix.com
Address: 54.74.73.31
Name:   netflix.com
Address: 3.251.50.149
Name:   netflix.com
Address: 54.155.278.5
Name:   netflix.com
Address: 2a05:d018:76c:b684:8ab7:ac02:667b:e863
Name:   netflix.com
Address: 2a05:d018:76c:b683:a2cd:4240:8669:6d4
Name:   netflix.com
Address: 2a05:d018:76c:b685:e8ab:afd3:af51:3aed

(ngozi@kali)-[~]
└─$ nslookup google.com
Server:      192.168.63.2
Address:     192.168.63.2#53

Non-authoritative answer:
Name:   google.com
Address: 216.58.223.238
Name:   google.com
Address: 2001:48f0:4003:802::200e

(ngozi@kali)-[~]
└─$
```

For Netflix:

Server: 192.168.63.2

Address: 192.168.63.2#53

Non-authoritative answer:

Name: netflix.com

Address: 54.74.73.31

Name: netflix.com

Address: 3.251.50.149

Name: netflix.com

Address: 54.155.178.5

Name: netflix.com

Address: 2a05:d018:76c:b684:8ab7:ac02:667b:e863

Name: netflix.com

Address: 2a05:d018:76c:b683:a2cd:4240:8669:6d4

Name: netflix.com

Address: 2a05:d018:76c:b685:e8ab:afd3:af51:3aed

For google.com

Server: 192.168.63.2

Address: 192.168.63.2#53

Non-authoritative answer:

Name: google.com

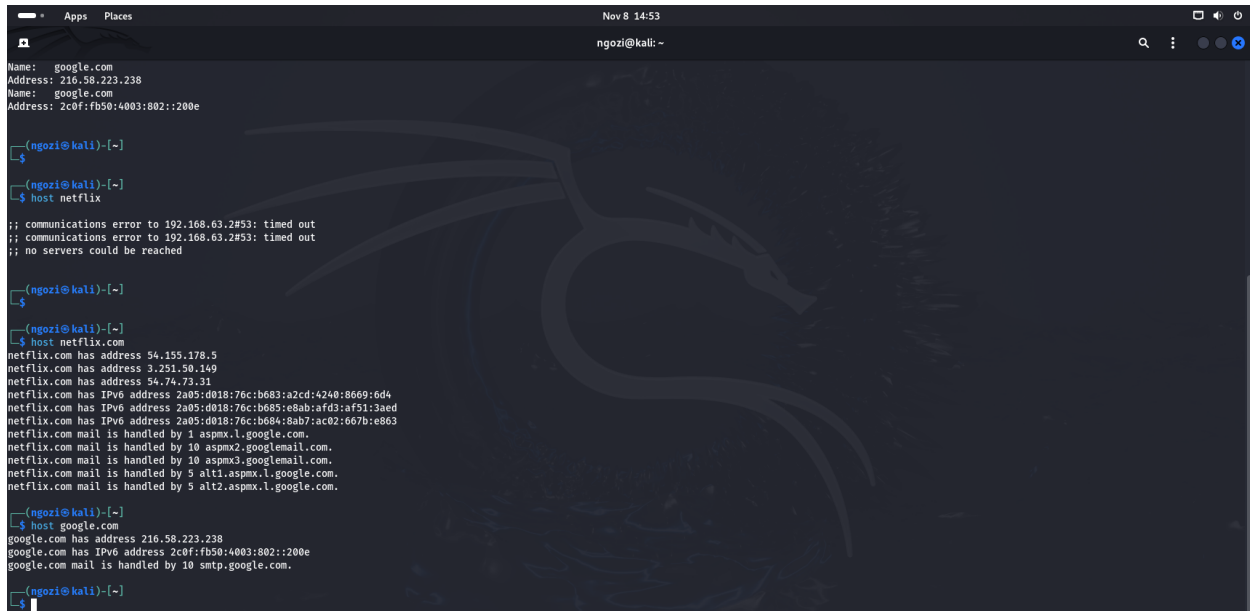
Address: 216.58.223.238

Name: google.com

Address: 2c0f:fb50:4003:802::200e

Exercise 2:

- Compare the output of host with nslookup. What differences did you observe? _____



```
(ngozi@kali)-[~]
$ host google.com
Name: google.com
Address: 216.58.223.238
Name: google.com
Address: 2c0f:fb50:4003:802::200e

(ngozi@kali)-[~]
$ host netflix
;; communications error to 192.168.63.2#53: timed out
;; communications error to 192.168.63.2#53: timed out
;; no servers could be reached

(ngozi@kali)-[~]
$ host netflix.com
netflix.com has address 54.155.178.5
netflix.com has address 3.251.50.149
netflix.com has address 54.74.73.31
netflix.com has IPv6 address 2a05:d018:76c:b683:a2cd:4248:8669:6d4
netflix.com has IPv6 address 2a05:d018:76c:b683:e8ab:afd3:af51:3aed
netflix.com has IPv6 address 2a05:d018:76c:b684:8ab7:ac02:667b:e863
netflix.com mail is handled by 1 aspmx1.google.com.
netflix.com mail is handled by 10 aspmx2.googlemail.com.
netflix.com mail is handled by 10 aspmx3.googlemail.com.
netflix.com mail is handled by 5 alt1.aspmx1.google.com.
netflix.com mail is handled by 5 alt2.aspmx1.google.com.

(ngozi@kali)-[~]
$ host google.com
google.com has address 216.58.223.238
google.com has IPv6 address 2c0f:fb50:4003:802::200e
google.com mail is handled by 10 smtp.google.com.

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

Difference between host and nslookup

1. Host provides a more concise output, while nslookup can give more detailed information
2. Nslookup offers an interactive mode, which host does not
3. Nslookup can provide more detailed information depending on the query type and mode used.

Exercise 3:

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

```
ngozl@kali: ~  
$ dig netflix.com  
  
;<>> Dig 9.28.0-Debian <>> netflix.com  
;; global options: +cmd  
;; Got answer:  
;;->HEADER<- opcode: QUERY, status: NOERROR, id: 9629  
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; MBZ: 0x0005, udp: 1280  
;; QUESTION SECTION:  
;netflix.com.  
; IN A  
  
;; ANSWER SECTION:  
netflix.com. 5 IN A 3.251.50.149  
netflix.com. 5 IN A 54.155.178.5  
netflix.com. 5 IN A 54.74.73.31  
  
;; Query time: 36 msec  
;; SERVER: 192.168.63.2#53(192.168.63.2) (UDP)  
;; WHEN: Fri Nov 08 15:20:08 EST 2024  
;; MSG SIZE rcvd: 88  
  
ngozl@kali: ~
```

The dig command provides a wealth of information compared to nslookup and host.

Below is a detailed section in the dig output:

1. **Header Section:**
 - **Opcode:** Indicates the type of query (which is QUERY).
 - **Status:** Shows the status of the query (which is NOERROR)
 - **Flags:** Various flags indicating the nature of the response (qr, rd, ra).
2. **Question Section:**
 - Displays the query that was made, including the domain name and the type of record requested (e.g., A, MX, NS).
3. **Answer Section:**
 - Provides the actual answer to the query, including IP addresses for A records,
4. **Authority Section:**
 - Lists the authoritative name servers for the domain, which can be useful for understanding the DNS hierarchy and delegation.
5. **Additional Section:**
 - Contains additional information that may be relevant to the query, such as IP addresses of the authoritative name servers.
6. **OPT Pseudo-Section:**
 - Shows extended DNS (EDNS) information, including the version and UDP packet size.
7. **Query Time:**
 - Indicates how long the query took to complete, which can be useful for performance analysis.
8. **Server Information:**
 - Displays the DNS server that provided the response.

9. Message Size:

- Shows the size of the response message, which can be important for understanding the amount of data transferred.

4. Advanced DNS Queries:

Exercise 4:

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

```
Nov 8 17:26
ngozi@kali: ~
(ngozi@kali)-[~]
$ dig netflix.com mx

;<<>> DIG 9.20.0-Debian <<>> netflix.com mx
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 32968
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 4, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0x0005, udp: 1280
;; QUESTION SECTION:
;netflix.com. IN MX
;; ANSWER SECTION:
netflix.com. 5 IN MX 1 aspmx.l.google.com.
netflix.com. 5 IN MX 10 aspmx2.googlemail.com.
netflix.com. 5 IN MX 10 aspmx3.googlemail.com.
netflix.com. 5 IN MX 5 alt1.aspmx.l.google.com.
netflix.com. 5 IN MX 5 alt2.aspmx.l.google.com.

;; AUTHORITY SECTION:
netflix.com. 5 IN NS ns-1372.awsdns-43.org.
netflix.com. 5 IN NS ns-1984.awsdns-56.co.uk.
netflix.com. 5 IN NS ns-659.awsdns-18.net.
netflix.com. 5 IN NS ns-81.awsdns-10.com.

;; Query time: 256 msec
;; SERVER: 192.168.63.2#53(192.168.63.2) (UDP)
;; WHEN: Fri Nov 08 17:26:03 EST 2024
;; MSG SIZE rcvd: 306

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

```
ngozl@kali: ~  
$ dig netflix.com txt  
;; Truncated, retrying in TCP mode.  
;<>> DiG 9.20.0-Debian <>> netflix.com txt  
;; global options: +cmd  
;; Got answer:  
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 41211  
;; flags: qr rd ra; QUERY: 1, ANSWER: 26, AUTHORITY: 4, ADDITIONAL: 1  
;<>> OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; udp: 1280  
;; QUESTION SECTION:  
;netflix.com.  
;<>> ANSWER SECTION:  
netflix.com. 298 IN TXT "1password-site-verification=BXCRTZRWNVG4PFLIYFI8WSYHX4"  
netflix.com. 298 IN TXT "8cd4e8d7d5994fcc9d350683a8cb07a1"  
netflix.com. 298 IN TXT "apple-domain-verification=0hlo8qLyb9N4JaIm"  
netflix.com. 298 IN TXT "asv=4853f01bie9226ed9d0031284948059f"  
netflix.com. 298 IN TXT "atlassian-domain-verification=TX0EFjn8bXAu0o9GAHyYowM0mcu40DPHFF10cqaDXFCvU9tRB7R/A9oeQcDmEAD8"  
netflix.com. 298 IN TXT "canva-site-verification=DW0T-0KEapku9QB9ChMocw"  
netflix.com. 298 IN TXT "docker-verification=5f9a055c-22b9-4d40-b07f-5af4171e1e71"  
netflix.com. 298 IN TXT "docuSign=f249396f-8150-48f8-8bd2-705be6e03826"  
netflix.com. 298 IN TXT "docuSign=f3d30bef-ec7d-42e5-9334-626611acb127"  
netflix.com. 298 IN TXT "dropbox-domain-verification=htwo1lxk2yl1"  
netflix.com. 298 IN TXT "facebook-domain-verification=k05vedr09b2tp2q144ho1zewp3xsc6"  
netflix.com. 298 IN TXT "google-site-verification=9DgwSKXMLFzcnW-HuGwef0aVVHWDQCNeHxHTq0P59IA"  
netflix.com. 298 IN TXT "google-site-verification=VQx0V3pv-QYIDfbqa1N4r97x8W07veRTK6JhUavIuc"  
netflix.com. 298 IN TXT "google-site-verification=VX0Af7gFR4vFk1RkUwiYt3pZL2AVUP6aPdBgV1qtwcw"  
netflix.com. 298 IN TXT "google-site-verification=a8Lak2UwVj1mH1xRYU3mJ6nSQ7rJnyf2VKWtH4nKZI"  
netflix.com. 298 IN TXT "google-site-verification=nC1qdLMabPJ0vtQNC05KaPyDfwog9pDr3d8IN767YA"  
netflix.com. 298 IN TXT "h1-domain-verification=AYCqXftcqVZAhLWt58GvYZWrbTfkgEms1jza2jPS2E1qcn1"  
netflix.com. 298 IN TXT "infoblox-domain-mastery=83433630723145c8e700674aa65ad12bf58d7cd22434a5527c383d131ccc354a77"  
netflix.com. 298 IN TXT "klaviyo-site-verification=UM4UEX"  
netflix.com. 298 IN TXT "klaviyo-site-verification=WabqJa"  
netflix.com. 298 IN TXT "login-verification-code=905b1ed4-1c2e-466f-b24c-75ae6ca39eb5"
```

Querying different DNS record types for netflix.com provides valuable insights that can be leveraged in a penetration test. Here's what I learnt and how this information can be useful:

What I Learnt From Different Record Types

- 1. **A Records (Address Records):**
 - **Purpose:** Maps netflix.com to its corresponding IPv4 addresses.
 - **Usefulness:** Identifies the IP addresses of Netflix's servers, which can be targeted for further network scanning and reconnaissance.
- 2. **MX Records (Mail Exchange Records):**
 - **Purpose:** Specifies the mail servers responsible for receiving email for netflix.com
 - **Usefulness:** Reveals the email infrastructure, which can be targeted for phishing attacks or email server vulnerabilities.
- 3. **NS Records (Name Server Records):**
 - **Purpose:** Indicates the authoritative name servers for netflix.com
 - **Usefulness:** Helps understand the DNS hierarchy and can be used to identify potential points of DNS misconfiguration or attacks.
- 4. **TXT Records (Text Records):**
 - **Purpose:** Holds arbitrary text data, often used for SPF (Sender Policy Framework), DKIM (Domain Keys Identified Mail), and domain verification.
 - **Usefulness:** Provides insights into email security configurations and other domain-related information that can be exploited if misconfigured.
- 5. **CNAME Records (Canonical Name Records):**
 - **Purpose:** Maps an alias name to a true or canonical domain name.
 - **Usefulness:** Identifies subdomains and aliases that might be overlooked but could be entry points for attacks.

6. SOA Records (Start of Authority Records):

- **Purpose:** Contains administrative information about the domain, including the primary name server and email of the domain administrator.
- **Usefulness:** Provides details about the domain's administrative setup, which can be useful for social engineering or identifying potential misconfigurations.

How This Information Is useful In A Penetration Test

1. Reconnaissance:

- Gathering detailed information about Netflix's infrastructure, including IP addresses, mail servers, and name servers, helps in mapping the network and identifying potential targets.

2. Identifying Vulnerabilities:

- Understanding the DNS setup and configurations can reveal misconfigurations, outdated software, or other vulnerabilities that can be exploited.

3. Phishing and Social Engineering:

- Information from MX and TXT records can be used to craft targeted phishing attacks or to bypass email security measures.

4. Subdomain Enumeration:

- CNAME records can help identify subdomains that might be vulnerable to attacks or misconfigurations.

5. DNS Attacks:

- Knowledge of the authoritative name servers and SOA records can be used to perform DNS hijacking, cache poisoning, or other DNS-based attacks.

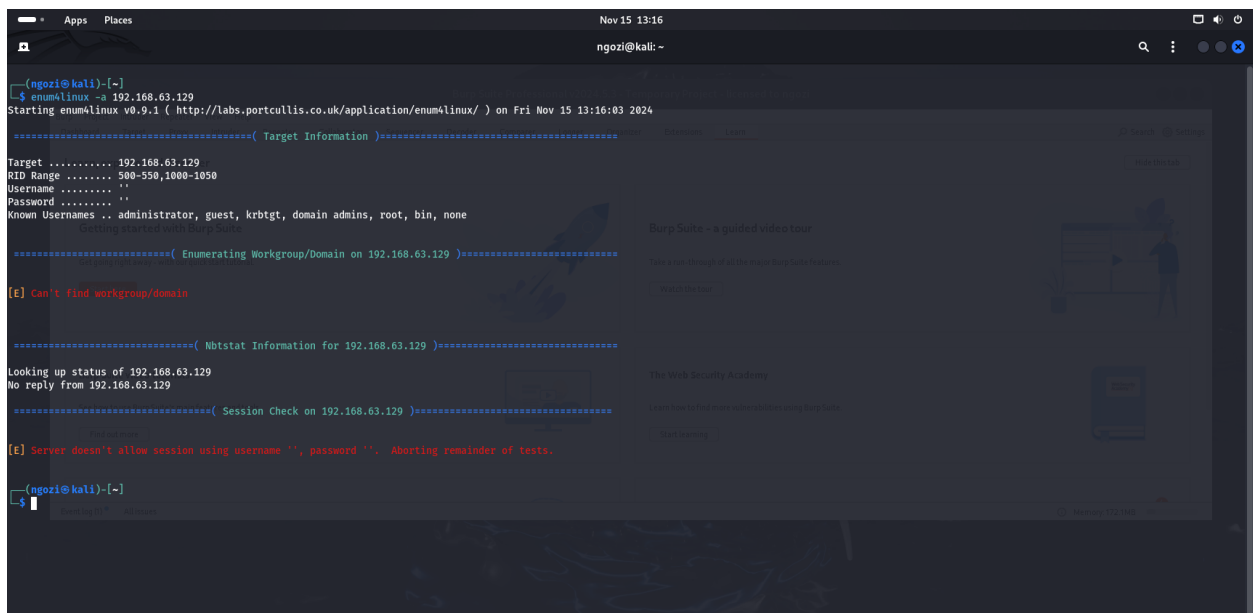
Step 2: SMB Enumeration with enum4linux

```
Nov 13 16:12
ngozi@kali: ~
[+] Found new SID:
S-1-5-32
[+] Found new SID:
S-1-5-32
[+] Enumerating users using SID S-1-5-21-2814459928-1332494333-2211073762 and logon username '', password ''
S-1-5-21-2814459928-1332494333-2211073762-501 OWASPBWA\nobody (Local User)
S-1-5-21-2814459928-1332494333-2211073762-513 OWASPBWA\None (Domain Group)
S-1-5-21-2814459928-1332494333-2211073762-1000 OWASPBWA\User (Local User)
S-1-5-21-2814459928-1332494333-2211073762-1001 OWASPBWA\root (Local User)
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\User (Local User)
[+] Enumerating users using SID S-1-5-32 and logon username '', password ''
S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)
===== ( Getting printer info for 192.168.63.129 )=====
No printers returned.
enum4linux complete on Wed Nov 13 16:12:24 2024
(ngozi@kali)-[~]
```

Exercise 5: • What information did you gather about the target system? Document the shares, users, and any other relevant details found. _____

Answer:

No workgroup found for Ip address 192.168.63.129



```
(ngozi@kali)-[~]
$ enum4linux -a 192.168.63.129
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Fri Nov 15 13:16:03 2024

===== ( Target Information )=====
Target ..... 192.168.63.129
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
Enumerating started with no results

===== ( Enumerating Workgroup/Domain on 192.168.63.129 )=====
[E] Can't find workgroup/domain

===== ( Nbtstat Information for 192.168.63.129 )=====
Looking up status of 192.168.63.129
No reply from 192.168.63.129

===== ( Session Check on 192.168.63.129 )=====
[E] Server doesn't allow session using username '', password ''. Aborting remainder of tests.

(ngozi@kali)-[~]
```

Filtering Results:

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

Enum4linux -S # Lists shares


```
#####( Nbtstat Information for 192.168.63.129 )#####
Looking up status of 192.168.63.129
No reply from 192.168.63.129
#####( Session Check on 192.168.63.129 )#####

[E] Server doesn't allow session using username '', password ''. Aborting remainder of tests.
#####( Enumerating Workgroup/Domain on 192.168.63.129 )#####

[E] Can't find workgroup/domain.

#####( Session Check on 192.168.63.129 )#####

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

(gn0zi@kali)-[~]
$ enum4linux -u 192.168.63.129
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Fri Nov 15 13:24:37 2024
#####( Target Information )#####
Target ..... 192.168.63.129
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

#####( Enumerating Workgroup/Domain on 192.168.63.129 )#####

[E] Can't find workgroup/domain.

#####( Session Check on 192.168.63.129 )#####

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

(gn0zi@kali)-[~]
$
```

Enum4linux -u # Lists users

```
(gn0zi@kali)-[~]
$ enum4linux -u 192.168.63.129
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 consecutive RIDs don't correspond to
  a username. Implies RID range ends at 999999. Useful
  against DCs.
-l Get some (limited) info via LDAP 389/TCP (for DCs only)
-s file brute force guessing for share names
-k user User(s) that exists on remote system (default: administrator,guest,krbtgt,domain admins,root,bin,none)
  Used to get sid with "lookupsid known_username"
-o Use comas to try several users: "-k admin,user1,user2"
-i Get OS information
-w wrkg Specify workgroup manually (usually found automatically)
-n Do an nmblookup (similar to nbtstat)
```

Exercise 6:

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

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.

Answer:

Comparing Results from enum4linux and DNS Queries

1. DNS Queries:

- **A Records:** These provide the IP addresses associated with the domain names.
- **MX Records:** These indicate the mail servers responsible for receiving emails on behalf of the domain.
- **TXT Records:** These can contain various types of information, such as domain ownership verification and email security policies.

2. SMB Enumeration with enum4linux:

- **Shares:** Lists the shared resources available on the target system.
- **Users:** Provides information about the user accounts on the target system.

Insights Gained

By combining the data from DNS queries and SMB enumeration, you can gain a comprehensive understanding of the target network's structure and potential vulnerabilities:

- **Network Structure:** DNS records help map out the network's domain structure and associated IP addresses, while SMB enumeration reveals the shared resources and user accounts on the network.
- **Potential Vulnerabilities:** Identifying exposed SMB shares and user accounts can highlight potential entry points for attackers. Additionally, analyzing DNS records can reveal misconfigurations or outdated information that could be exploited

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. _____

Methodologies Used

1. DNS Queries:

- **Nslookup:** Used to perform DNS lookups and gather information about domain names and IP addresses.
- **Host:** A simple utility for performing DNS lookups and obtaining similar information as nslookup.
- **Dig:** A more flexible and detailed DNS query tool used to extract additional information such as MX and TXT records.

2. SMB Enumeration:

- **Enum4linux:** A tool used to enumerate SMB shares and user information on a target system.

Tools Employed

- **Nslookup:** Command-line tool for querying the Domain Name System (DNS).
- **Host:** Utility for performing DNS lookups.
- **Dig:** Detailed DNS query tool.
- **Enum4linux:** Tool for gathering information from Windows machines via SMB.

Key Insights

1. DNS Records:

- **A Records:** Provide the IP addresses associated with domain names.
- **MX Records:** Indicate the mail servers responsible for receiving emails on behalf of the domain.
- **TXT Records:** Contain various types of information, such as domain ownership verification and email security policies.

2. SMB Shares and User Information:

- **Shares:** Lists the shared resources available on the target system.
- **Users:** Provides information about the user accounts on the target system.

How This Information Could Be Useful in a Penetration Testing Engagement

1. Network Mapping:

DNS records help map out the network's domain structure and associated IP addresses, providing a clear picture of the target network's layout.

2. Identifying Potential Vulnerabilities:

Exposed SMB shares and user accounts can highlight potential entry points for attackers. Misconfigurations or outdated DNS records can also be exploited.

3. Targeted Attacks:

Information about user accounts and shared resources can be used to craft targeted attacks, such as phishing or brute-force attacks on user credentials.

4. Comprehensive Analysis:

Combining data from DNS queries and SMB enumeration allows for a comprehensive analysis of the target network's structure and vulnerabilities, enabling more effective penetration testing strategies.

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? _____

Answer:

Below is the output of the Tor status. Yes, it is running because it's showing "active"

```
Nov 13 16:53
ngozi@kali: ~
$ sudo nano /etc/resolv.conf
[sudo] password for ngozi:
(ngozi@kali)-[~]
$ systemctl start tor

(ngozi@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 Wed 2024-11-13 16:52:31 EST; 31s ago
  Invocation: 1ad86705e49e4d76bf080ad1047f951a
     Process: 9503 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 9503 (code=exited, status=0/SUCCESS)

Nov 13 16:52:31 kali systemd[1]: Starting tor.service - Anonymizing overlay network for TCP (multi->
Nov 13 16:52:31 kali systemd[1]: Finished tor.service - Anonymizing overlay network for TCP (multi->

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

Exercise 2:

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

Proxychains offers three different proxy modes, each with its own unique way of handling proxy servers:

1. Dynamic Chain (dynamic_chain):

- In this mode, Proxychains will dynamically skip any non-working proxies in the chain. If a proxy server is unavailable, the traffic is automatically forwarded to the next available proxy. This ensures that your connection remains uninterrupted even if some proxies fail.

2. Random Chain (random_chain):

- This mode randomly selects a proxy server from the list for each connection. It adds an element of unpredictability to your proxy usage, which can enhance privacy by making it harder to trace your activities.

3. Strict Chain (strict_chain):

- In strict chain mode, all proxy servers in the list must be passed in the exact order they are listed. If any proxy in the chain is unavailable, the connection will fail. This mode ensures that your traffic passes through all specified proxies, providing a consistent and predictable path.

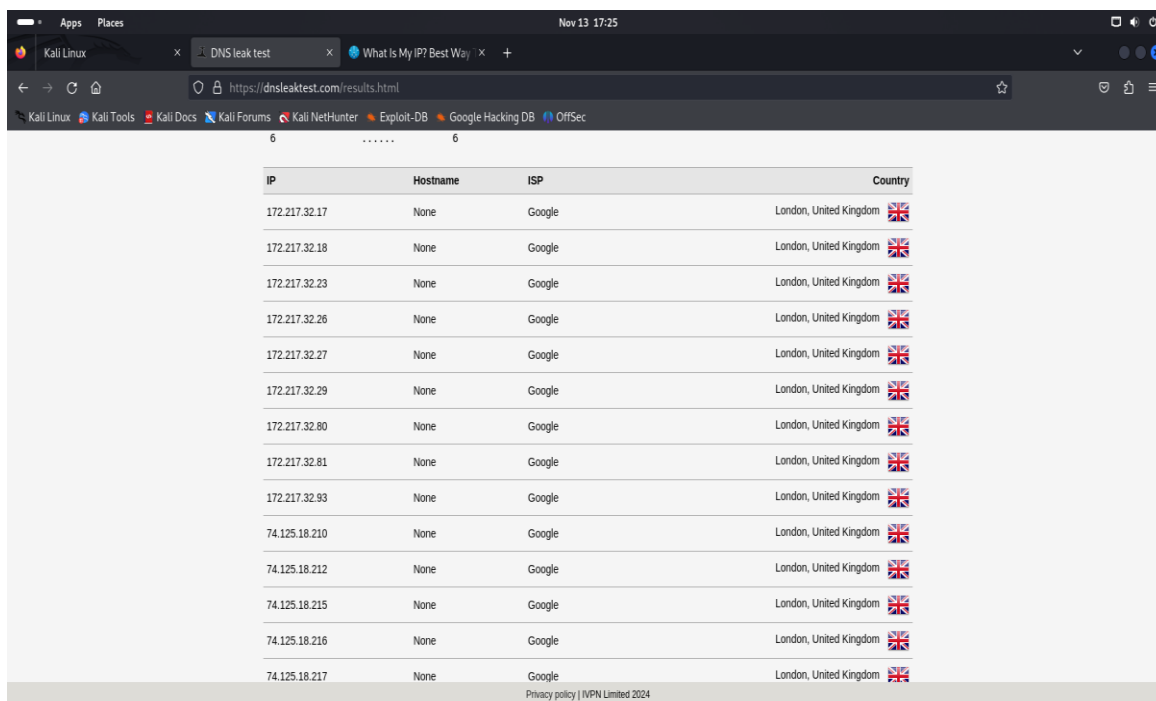
These modes allow you to tailor your proxy usage based on your specific needs and the reliability of your proxy servers.















Exercise 3:

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

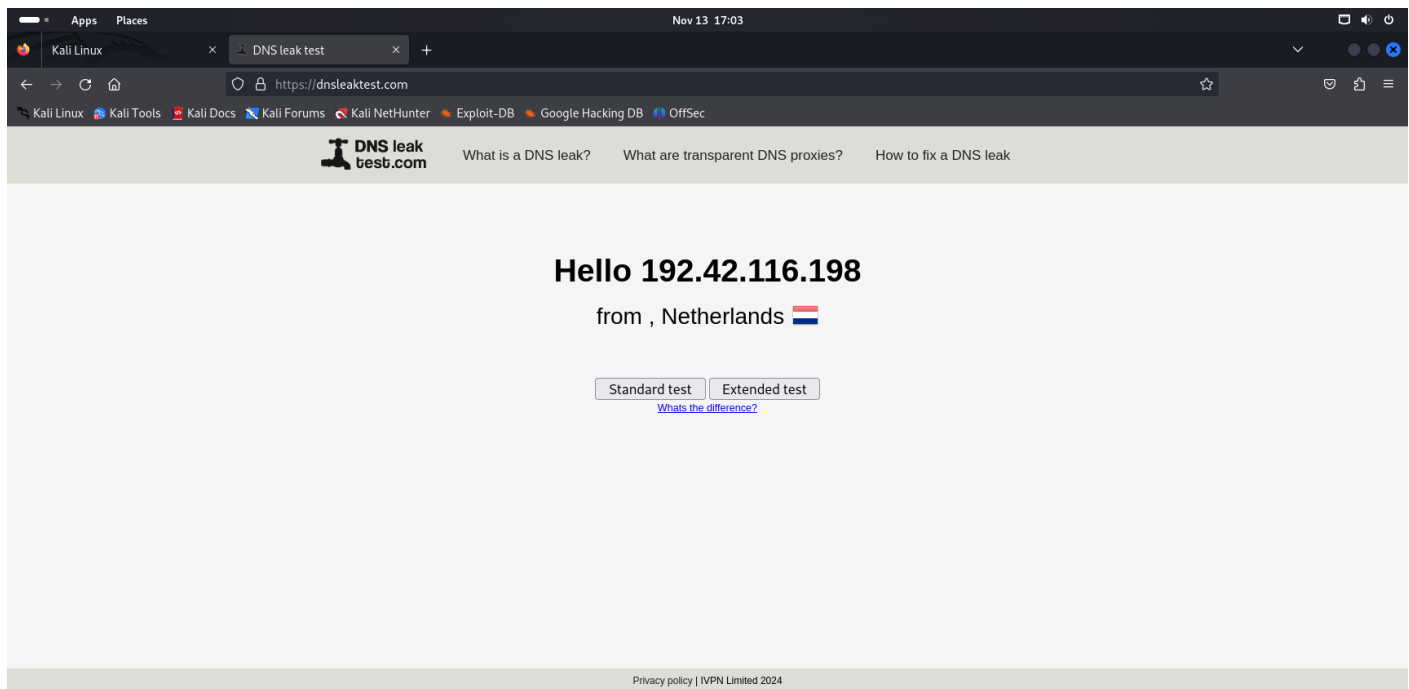
Answer:

These are the IP address I saw in the output while my actual IP address is 192. 168.63.129



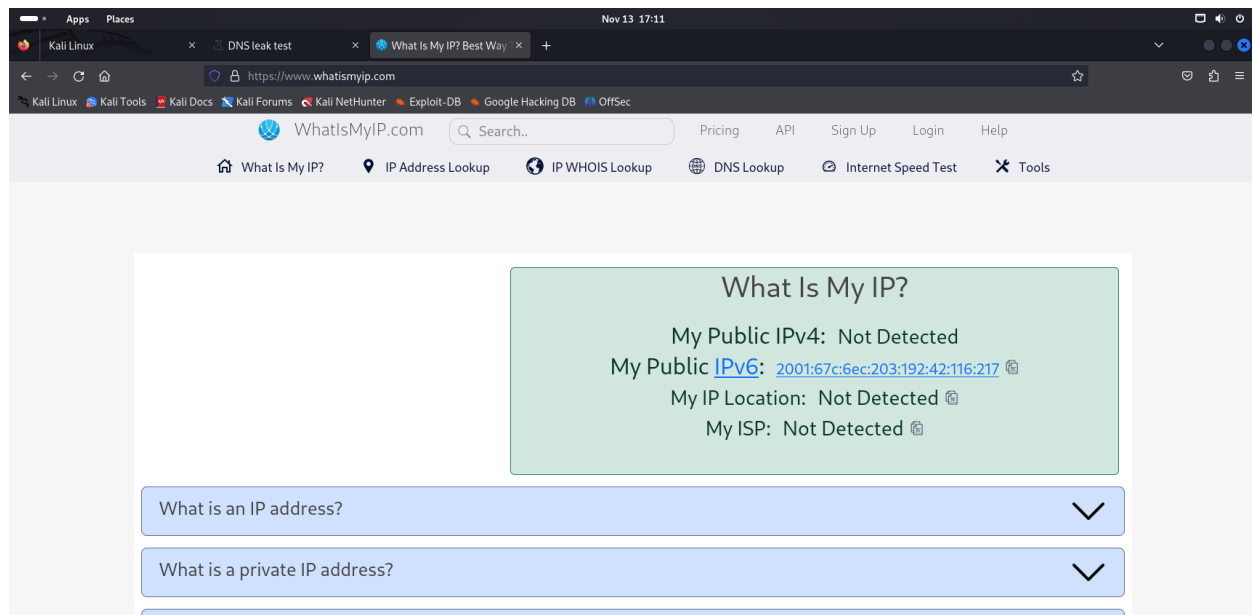
IP	Hostname	ISP	Country
172.217.32.17	None	Google	London, United Kingdom 
172.217.32.18	None	Google	London, United Kingdom 
172.217.32.23	None	Google	London, United Kingdom 
172.217.32.26	None	Google	London, United Kingdom 
172.217.32.27	None	Google	London, United Kingdom 
172.217.32.29	None	Google	London, United Kingdom 
172.217.32.80	None	Google	London, United Kingdom 
172.217.32.81	None	Google	London, United Kingdom 
172.217.32.93	None	Google	London, United Kingdom 
74.125.18.210	None	Google	London, United Kingdom 
74.125.18.212	None	Google	London, United Kingdom 
74.125.18.215	None	Google	London, United Kingdom 
74.125.18.216	None	Google	London, United Kingdom 
74.125.18.217	None	Google	London, United Kingdom 

[Privacy policy | IVPN Limited 2024](#)



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? _____



Exercise 5

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

Answer:

Reliability: The reliability of your scans can be affected by the stability of the Tor network and the exit nodes you are using. If an exit node is unstable or slow, it can disrupt your scanning process.

Anonymity: While Tor provides anonymity, it also means that your scans are more likely to be flagged or blocked by target systems, as traffic from Tor exit nodes is often associated with malicious activity


```
Nov 13 17:36
ngozi@kali: ~
Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds

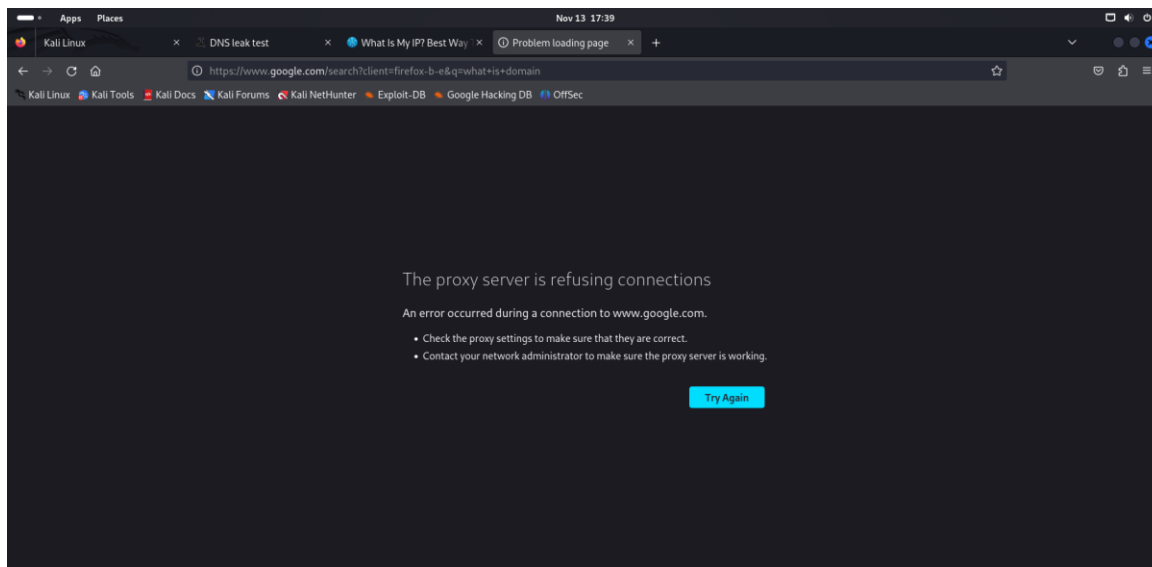
(ngozi@kali)-[~]
$ proxychains nmap -sT -Pn 192.168.63.129
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-13 17:31 EST
[proxychains] Strict chain ... 127.0.0.1:9050 ... 192.168.63.129:3306 <--denied
[proxychains] Strict chain ... 127.0.0.1:9050 ... 192.168.63.129:113 <--denied
[proxychains] Strict chain ... 127.0.0.1:9050 ... 192.168.63.129:1720 <--denied
[proxychains] Strict chain ... 127.0.0.1:9050 ... 192.168.63.129:3389 <--denied
[proxychains] Strict chain ... 127.0.0.1:9050 ... 192.168.63.129:143 <--denied
[proxychains] Strict chain ... 127.0.0.1:9050 ... 192.168.63.129:443 <--denied
[proxychains] Strict chain ... 127.0.0.1:9050 ... 192.168.63.129:135 <--denied
[proxychains] Strict chain ... 127.0.0.1:9050 ... 192.168.63.129:995 <--denied
[proxychains] Strict chain ... 127.0.0.1:9050 ... 192.168.63.129:111 <--denied
[proxychains] Strict chain ... 127.0.0.1:9050 ... 192.168.63.129:1723 <--denied
[proxychains] Strict chain ... 127.0.0.1:9050 ... 192.168.63.129:554 <--denied
[proxychains] Strict chain ... 127.0.0.1:9050 ... 192.168.63.129:5900 <--denied
[proxychains] Strict chain ... 127.0.0.1:9050 ... 192.168.63.129:80 <--denied
```

Exercise 6:

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

1. **IP Address:** The IP address seen by the target server will change to that of the new proxy server. This can be verified by checking the IP address on a service like [WhatIsMyIP.com](https://whatismyip.com).
2. **Latency:** The response time might increase or decrease depending on the speed and reliability of the new proxy server. If the new proxy is slower, you will notice a longer response time.
3. **Content:** In some cases, the content returned by the server might change if the new proxy server has any caching or filtering mechanisms in place. For example, some proxies might block certain types of content or modify the response.
4. **Headers:** The HTTP headers in the response might include additional information related to the new proxy server, such as Via or X-Forwarded-For headers, which indicate the presence of a proxy.
5. **Connection Stability:** The stability of the connection might be affected. If the new proxy server is unreliable, you might experience connection drops or errors.

Exercise 7: • What are some risks associated with using Tor? What precautions can you take while using it?



Using Tor comes with several risks, but there are also precautions you can take to mitigate them:

Risks:

1. Exit Node Monitoring: Tor exit nodes can be monitored by malicious actors, potentially exposing your traffic to eavesdropping.
2. Performance: Tor can significantly slow down your internet connection due to the multiple layers of encryption and routing.
3. Blocking: Some websites and services block traffic from known Tor exit nodes, limiting your access.
4. Legal Issues: In some countries, using Tor might attract unwanted attention from authorities, as it is often associated with illegal activities.
5. Malware: Downloading files through Tor can expose you to malware, as the network is sometimes used to distribute malicious software.

Precautions:

1. Use HTTPS: Always use HTTPS to encrypt your traffic end-to-end, even if the exit node is compromised.
2. Avoid Personal Information: Refrain from logging into personal accounts or sharing sensitive information while using Tor.
3. Regular Updates: Keep your Tor Browser and other software up to date to protect against vulnerabilities.
4. Use Bridges: If Tor is blocked in your region, use Tor bridges to bypass censorship.
5. Combine with VPN: For added security, consider using a VPN in conjunction with Tor to further anonymize your traffic.

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

Exercise 1: • What version of John the Ripper are you using? _____

Answer: The version of John the Ripper I'm using is **1.9.0-jumbo-1+bleeding-aec1328d6c**. This version was released on **2021-11-02**.



```
Nov 14 15:43
ngozi@kali: ~
-> john --help
John the Ripper 1.9.0-jumbo-1+bleeding-aec1328d6c 2021-11-02 10:45:52 +0100 OMP [linux-gnu 64-bit x86_64 AVX2 AC]
Copyright (c) 1996-2021 by Solar Designer and others
Homepage: https://www.openwall.com/john/

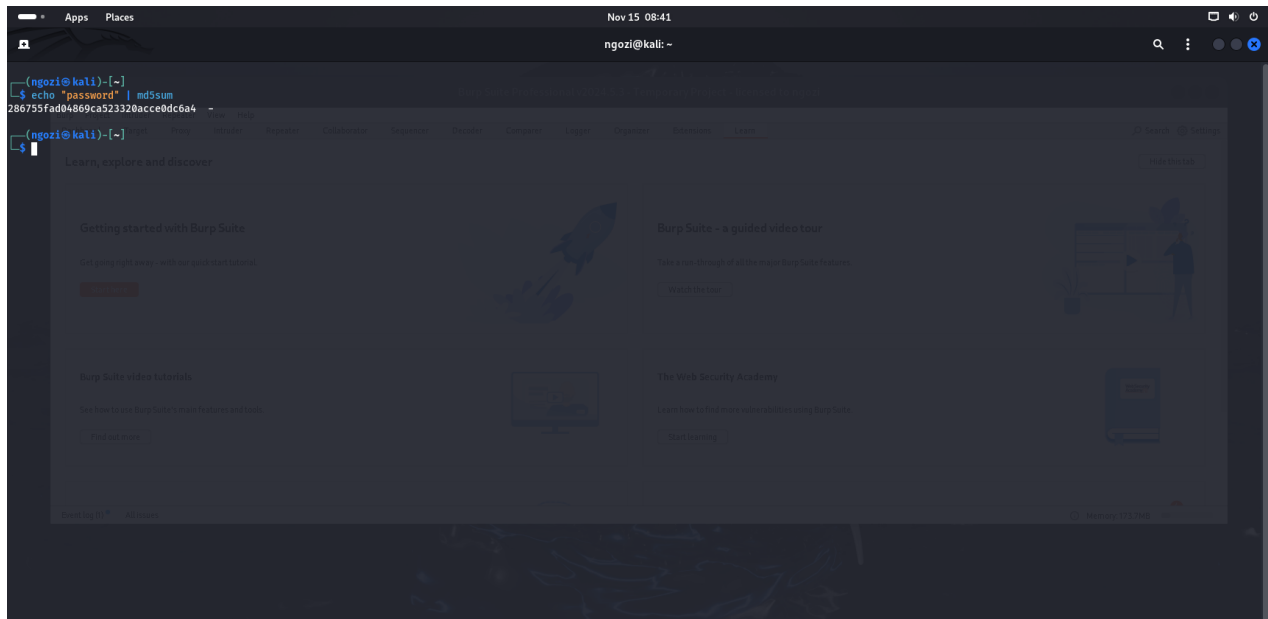
Usage: john [OPTIONS] [PASSWORD-FILES]

--help                Print usage summary
--single[=SECTION[...]] "Single crack" mode, using default or named rules
--single=rule[...]     Same, using "immediate" rule(s)
--single-seed=WORD[WORD] Add static seed word(s) for all salts in single mode
--single-wordlist=FILE  *Short* wordlist with static seed words/morphemes
--single-user-seed=FILE wordlist with seeds per username (user:password[s]
                        format)
--single-pair-max=N     Override max. number of word pairs generated (6)
--no-single-pair        Disable single word pair generation
--[no-]single-retest-guess Override config for SingleRetestGuess
--wordlist[=FILE] --stdin Wordlist mode, read words from FILE or stdin
--pipe                  Like --stdin, but bulk reads, and allows rules
--rules[=SECTION[...]] Enable word mangling rules (for wordlist or PRINCE
                        modes), using default or named rules
--rules=rule[...]       Same, using "immediate" rule(s)
--rules-stack=SECTION[...] Stacked rules, applied after regular rules or to
                        modes that otherwise don't support rules
--rules-skip=rule[...]  Same, using "immediate" rule(s)
--rules-skip-nop         Skip any NOP '!' rules (you already ran w/o rules)
--loopback[=FILE]       Like --wordlist, but extract words from a .pot file
--mem-file-size=SIZE     Size threshold for wordlist preload (default 2048 MB)
--dupe-suppression       Suppress all dupes in wordlist (and force preload)
--incremental[=MODE]     "Incremental" mode [using section MODE]
--incremental-charcount=N Override CharCount for incremental mode
--external=MODE          External mode or word filter
--mask[=MASK]            Mask mode using MASK (or default from john.conf)
--markov[=OPTIONS]       "Markov" mode (see doc/MARKOV)
--mkv-stats=FILE         "Markov" stats file
--prince[=FILE]          PRINCE mode, read words from FILE
--prince-loopback[=FILE] Fetch words from a .pot file
--prince-elen-cnt-min=N  Minimum number of elements per chain (1)
--prince-elen-cnt-max[=-N] Maximum number of elements per chain (negative N is
                        relative to word length) (8)
--prince-skip=N          Initial skip
--prince-limit=N         Limit number of candidates generated
--potgen=FILE            Calculate length distribution from wordlist
```

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.

Answer:

m



Exercise 3: • Download a sample hash and crack it using the wordlist. What was the password? Was it successful? _____

Answer: Mine wasn't successful, below is an image showing that my password cracking wasn't successful.

```
ngozikali: ~  
$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt ngozi.txt ~  
Using default input encoding: UTF-8  
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])  
Warning: no OpenMP support for this hash type, consider --fork=2  
Press 'q' or Ctrl-C to abort, almost any other key for status  
0g 0:00:00:01 DONE (2024-11-15 09:34) 0g/s 11567Kp/s 11567Kc/s 11567Kc/s fuckyooh21..*7;Vamos!  
Session completed.  
ngozikali: ~  
$ john --show ngozi.txt  
0 password hashes cracked, 2 left to Suite
```

Exercise 4:

- Run John with your custom wordlist on a given hash. Was your list successful in cracking the hash? _____

Step 4: Brute Force Mode

1. Understanding Brute Force:

- If the wordlist attack fails, you can switch to brute force mode: John --incremental hash.txt

2. Time and Complexity:

- Brute force takes longer but tries every possible combination.

```
ngozikali: ~  
$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt ngozi.txt ~  
Using default input encoding: UTF-8  
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])  
Warning: no OpenMP support for this hash type, consider --fork=2  
Press 'q' or Ctrl-C to abort, almost any other key for status  
0g 0:00:00:00 DONE (2024-11-15 14:46) 0g/s 15761Kp/s 15761Kc/s 15761Kc/s fuckyooh21..*7;Vamos!  
Session completed.  
ngozikali: ~  
$
```

Exercise 5:

- Perform a brute force attack on a hash. How long did the attack take, and was it successful?

Answer:

Yes, it was successful

```
ngozl@kali: ~  
$ john --incremental --format=raw-md5 ngozi.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])  
Warning: no OpenMP support for this hash type, consider --fork=2  
Press 'q' or Ctrl-C to abort, almost any other key for status  
0g 0:00:00.05 0g/s 10140Kp/s 10140Kc/s 10140Kc/s mybmbd..myk032  
Session aborted
```

Exercise 6: • Attempt cracking NTLM hashes using the rockyou.txt wordlist. Were you successful? How complex was the password? _____

Answer:

Yes, it was successful.

```
ngozl@kali: ~  
$ john --format=ntlm --wordlist=/usr/share/wordlists/rockyou.txt ngozi.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (NT [MD4 256/256 AVX2 8x3])  
Warning: no OpenMP support for this hash type, consider --fork=2  
Press 'q' or Ctrl-C to abort, almost any other key for status  
0g 0:00:00.04 DONE (2024-11-15 14:37) 0g/s 3304Kp/s 3304Kc/s 3304Kc/s _ 09..*7jVamos!  
Session completed.
```

Exercise 7: • Use rules with a wordlist attack to crack a complex password. What was the result? _____

```
ngozl@kali: ~  
$ john --wordlist=rockyou.txt --rules --format=raw-md5 ngozi.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])  
Warning: no OpenMP support for this hash type, consider --fork=2  
fopen: rockyou.txt: No such file or directory
```