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.

Tools Used

- **Nslookup**: A command-line tool for querying the Domain Name System (DNS).
- **Host**: A simple utility for performing DNS lookups.
- **Dig**: A more flexible and detailed DNS query tool.
- **Enum4linux**: A tool for gathering information from Windows machines via SMB.

Prerequisites

- Basic understanding of DNS and networking concepts.
- Familiarity with command-line operations in Linux.

Lab Steps

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

1.Using nslookup:

- Open your terminal in Kali Linux.
- Perform a DNS lookup for a live domain:
- Nslookup <target-domain>
- Example:Nslookup example.com

Exercise 1:

- What information did you obtain from the nslookup command? Document the IP addresses and any additional records retrieved. **93.184.215.14**

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**? More detailed with IP version**

3.Using dig:

Perform a more detailed query using dig:

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

Exercise 3:

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

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?

- 

Step 2: SMB Enumeration with enum4linux

1.Installing enum4linux (if not already installed):

- Ensure that enum4linux is installed on your system: Apt install enum4linux

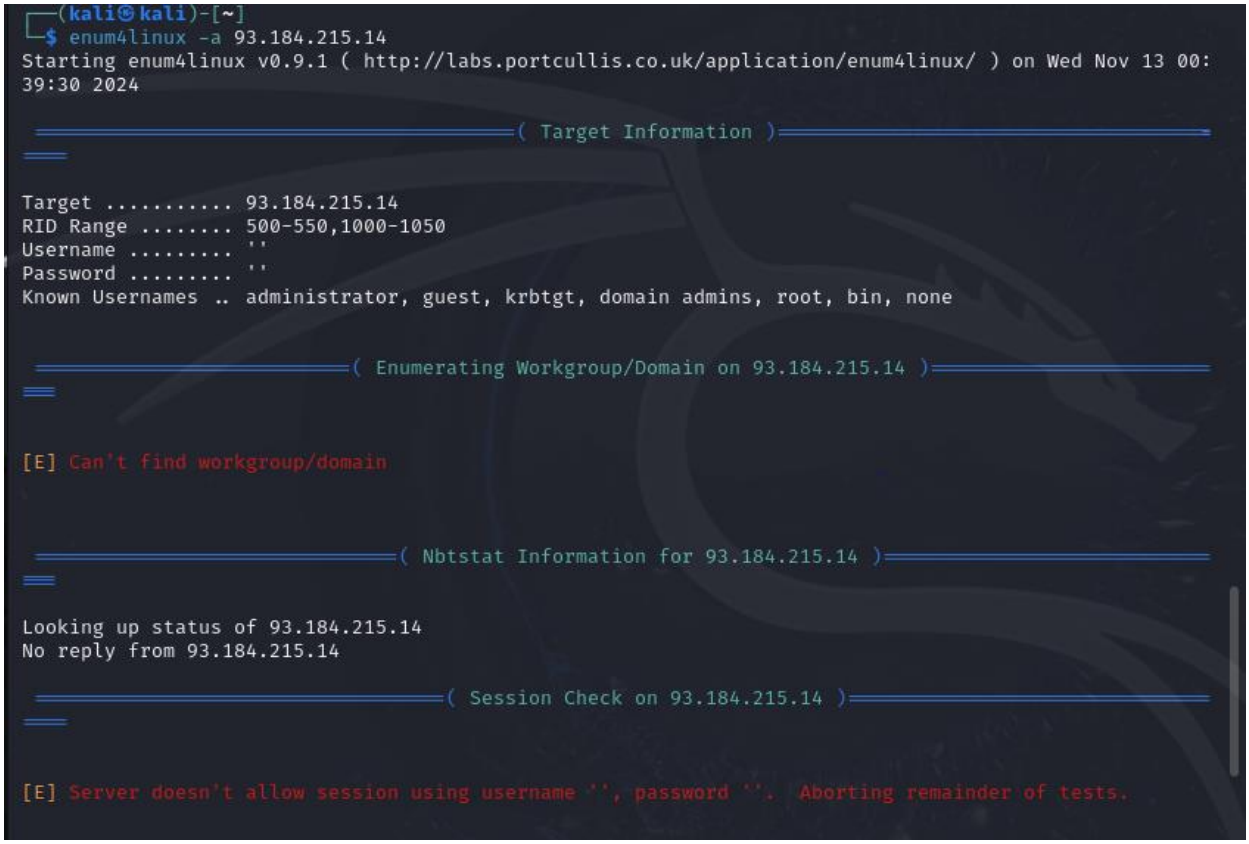
2.Using enum4linux:

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.



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? _____

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. _____

Conclusion

In this lab, you gained hands-on experience using nslookup, host, dig, and enum4linux for gathering information about domains and networked systems. You learned how to analyze and document your findings effectively. **Next Steps**

In the next lab, we will focus on advanced enumeration techniques and exploit development.

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

Lab Overview

In this lab, participants will explore the use of Tor for anonymous browsing and Proxychains for routing traffic through multiple proxies. This lab will provide students with practical skills to protect their identity and enhance their security while conducting penetration testing and other security-related activities.

Lab Objectives

By the end of this lab, you will be able to:

1. Understand how Tor operates and its role in anonymity.
2. Configure and use Proxychains to route network traffic through Tor.
3. Conduct secure browsing and maintain anonymity using Tor and Proxychains.

Tools Used

- **Tor**: A free software for enabling anonymous communication over the internet.
- **Proxychains**: A tool that forces any TCP connection made by any given application to follow through proxy (Tor, in this case).

Prerequisites

- Basic understanding of networking concepts.
- Familiarity with command-line operations in Linux.

Lab Steps

Step 1: Installing Tor and Proxychains

1.Installing Tor:

- Open your terminal in Kali Linux and install Tor:

Sudo apt update

Sudo apt install tor

2.Installing Proxychains:

- Ensure Proxychains is installed:

Sudo apt install proxychains

Step 2: Configuring Tor

1.Starting the Tor Service:

- Start the Tor service to enable anonymous browsing:

Sudo service tor start

2.Verifying Tor is Running:

- Check if Tor is running correctly: Systemctl status tor

Exercise 1:

- What output do you see when checking the Tor status? Is it running? No, it was not active.

- Then I used this command (torbrowser-launcher) and it became active and running.

Step 3: Configuring Proxychains

1.Editing Proxychains Configuration:

- Open the Proxychains configuration file for editing:

Sudo nano /etc/proxychains.conf

- Uncomment the following line to enable Tor:

Dynamic_chain

Proxy_dns

[ProxyList]

add proxy here …

meanwile

defaults set to tor Socks5

127.0.0.1 9050

Exercise 2:

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

_____

-

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? **85.220.101.61**

2.Browsing with Firefox:

- Open Firefox with Proxychains to browse the web anonymously: Proxychains firefox

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? **No**

Step 5: Advanced Usage of Proxychains

1.Using Proxychains with Other Tools:

- Try using Proxychains with other command-line tools like nmap: Proxychains nmap -sT -Pn <target-ip>

Exercise 5:

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

```
┌──(kali㉿kali)-[~]
└─$ proxychains nmap -sT -Pn 192.168.23
[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] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-13 06:15 EST
Nmap scan report for 192.168.23 (192.168.0.23)
Host is up.
All 1000 scanned ports on 192.168.23 (192.168.0.23) are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 202.07 seconds
```

2.Combining Proxychains with Other Proxies:

- Add additional proxies to the Proxychains configuration file to test different routes.

Exercise 6:

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

```
┌──(kali㉿kali)-[~]
└─$ proxychains nmap -sT -Pn https//www.google.com
[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] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-14 18:24 EST
Unable to split netmask from target expression: "https//www.google.com"
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.21 seconds
```

Step 6: Analyzing Results

1.Understanding Limitations and Risks:

- Discuss the limitations of using Tor and Proxychains for anonymity.
- Consider the potential for traffic analysis and exit node vulnerabilities.

Exercise 7:

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

Conclusion

In this lab, you gained hands-on experience using Tor for anonymous browsing and Proxychains for routing network traffic. You learned how to configure both tools and analyze their effectiveness in maintaining anonymity online.

Next Steps

In the next lab, we will focus on advanced techniques for web application testing, including the use of tools like Burp Suite and OWASP ZAP.

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

Lab Overview

In this lab, you will gain hands-on experience with John the Ripper, a powerful password-cracking tool. This lab will cover the practical use of John the Ripper for cracking password hashes in different formats. By simulating real-world password attacks, you will understand how to audit passwords and strengthen your security knowledge.

Lab Objectives

By the end of this lab, students will:

1. Understand how to use John the Ripper to crack password hashes.
2. Learn about various password hash formats and modes supported by John the Ripper.
3. Perform password cracking exercises using different attack techniques, including dictionary and brute force attacks.
4. Analyze the success of these techniques and understand password complexities.

Tools Used

- John the Ripper: A powerful password-cracking tool.
- Wordlists: Common password files for dictionary attacks.

- Hashcat: Can be introduced for comparison with John the Ripper, though focus will remain on John the Ripper.

Prerequisites

- Basic knowledge of Linux commands.
- Familiarity with cryptography and password hashing.

Lab Steps

Step 1: Installing John the Ripper

1.Ensure John the Ripper is Installed:

- In Kali Linux, John the Ripper comes pre-installed, but if not:

Sudo apt install john

2.Check the Version:

- Verify the installation by checking the version: John –version

Exercise 1:

- What version of John the Ripper are you using? **1.9.0-Jumbo-1+git20211102-0kali9**
-

Step 2: Understanding Password Hashes

1.Identify Hash Types:

- Before cracking passwords, identify the hash type. For example, you can check hash examples here:

Cat /etc/shadow

- Observe the hashed passwords in the /etc/shadow file.

2.Popular Hash Formats:

- MD5, SHA-1, and DES are common formats that John can handle.

Exercise 2:

ss

- 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>

Step 3: Cracking Passwords with Wordlists

1.Using Default Wordlist:

- Use the default wordlist (rockyou.txt) to attempt cracking a simple MD5 hash: John –wordlist=/usr/share/wordlists/rockyou.txt hash.txt

Exercise 3:

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

2.Adding Custom Wordlists:

- Create a custom wordlist file with possible passwords: Echo -e "password1\nletmein\nadmin123" > custom_wordlist.txt

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.

Exercise 5:

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

Step 5: Cracking Windows Password Hashes

1.Dumping Windows Hashes:

- Extract password hashes from a Windows machine using tools like pwdump or hashdump.
- Example hash (LM or NTLM):

John –format=nt hashfile

Exercise 6:

- Attempt cracking NTLM hashes using the rockyou.txt wordlist. Were you successful? How complex was the password? _____ **Step 6: Advanced Cracking with Rules**

1.Adding Rules for More Power:

- John the Ripper allows you to create rules that modify the wordlist: John –wordlist=rockyou.txt –rules –format=raw-md5 hash.txt

Exercise 7:

- Use rules with a wordlist attack to crack a complex password. What was the result? _____

Analysis and Conclusion

- Discuss the success rates of different cracking techniques and the importance of strong passwords.
- Highlight why password length, complexity, and use of salts are essential to improving security. **Additional Exercises**

1.Create Custom Hashes:

- Create a custom hash (MD5 or SHA-256) using Python:

```
Import hashlib

Print(hashlib.md5(b'password').hexdigest())
```

- Try cracking it using John.

2.Benchmark Performance:

- Run a benchmark test to see how fast John can run on your machine:

```
John –test
```

3.Compare with Hashcat:

- (Optional) Install Hashcat and compare its performance with John the Ripper for similar tasks.

Conclusion

In this lab, you have explored the power of John the Ripper for cracking passwords and learned various techniques to improve your penetration testing skills. Continue to experiment with different hash types and explore password-cracking techniques.