Module 04: Enumeration

Introduction

Enumeration is the process of actively gathering detailed information about a target system, network, or application to identify vulnerabilities and potential entry points. It involves probing services, extracting user and group details, identifying running applications, and mapping network resources. This phase is essential in penetration testing and vulnerability assessments, bridging the gap between reconnaissance and exploitation by providing actionable insights into the target environment.

At the conclusion of this module, you will be able to:

- Illustrate concepts related to enumeration
- Explain various methods for enumerating NetBIOS
- Describe different approaches to SNMP enumeration
- Outline several techniques for LDAP and Active Directory (AD) enumeration
- Discuss various methods for NTP enumeration
- Explain different techniques for NFS enumeration
- Detail the approaches for SMTP and DNS enumeration
- Describe additional enumeration techniques such as IPsec, VoIP, RPC, Linux/Unix, and SMB enumeration
- Implement countermeasures for enumeration

Enumeration Concepts

Enumeration is a vital phase in penetration testing where active connections to target systems reveal detailed information about architecture, users, and services. Building on reconnaissance, it transitions from passive observation to active probing to uncover system details like usernames, shared resources, and service banners. Techniques include querying network services, extracting metadata, and using tools like Nmap and Metasploit. Key services and ports, such as NetBIOS/SMB (137, 139, 445), SNMP (161), LDAP (389), SMTP (25), and DNS (53), along with protocols like NTP and Telnet, are examined for insights into network topology and vulnerabilities, essential for effective security assessments.

What is Enumeration?

In the Enumeration phase, an attacker initiates active connections with the target system. Through this active connection, direct queries are generated to gain more information. This information helps to identify the system's attack points. Once an attacker discovers attack points, they can gain unauthorized access by using the collected information to reach the assets.

The information enumerated in this phase is:

- Routing Information
- SNMP Information
- DNS Information
- Machine Name
- User Information
- Group Information
- Application and Banners
- Network Sharing Information
- Network Resources

While enumerating, attackers might encounter a remote Inter-Process Communication (IPC) share, like IPC$ in Windows, which they can investigate further to access an administrative share by attempting to guess admin credentials, allowing them to acquire full details regarding the file-system listing that the share denotes.

The previous modules demonstrated the methods attackers use to collect essential information about a target without any illegal activity. Nevertheless, enumeration practices could be considered illegal, depending on the regulations set by the organization and the applicable laws. A penetration tester or an ethical hacker needs to obtain the necessary permission before conducting enumeration.

Techniques for Enumeration

The following techniques are used to gather information about a target:

Extract User Name using an Email ID

Using an Email ID to extract information can provide useful information such as username, domain name, etc. An email address usually contains the username and domain name, formatted as:

```
username@domainname
```

Extract Information using the Default Password

Another way of enumeration is by using default passwords. Every device and software has default credentials and settings. It is recommended that these default settings and configurations be changed. Certain administrators continue to utilize default passwords and configurations, making it very easy for an attacker to gain unauthorized access by using default credentials. Finding default settings, configurations, and passwords of devices is no longer difficult.

Brute force AD

Active Directory (AD) offers a centralized way to manage and control computers, domain users, and network printers. It restricts access to network resources to defined users and computers. The AD is a big target as it is a good source of sensitive information for an attacker. Brute forcing or generating queries to LDAP services helps to gather information such as username, address, credentials, privileges information, etc.

Microsoft Active Directory is vulnerable to username enumeration during the verification of user-supplied input. This flaw arises from how Microsoft Active Directory is designed. When a user activates the "logon hours" feature, every attempt at service authentication yields different error messages. Malicious actors exploit this to identify valid usernames. If an attacker manages to obtain valid usernames, they can execute a brute-force attack to gain access to the corresponding passwords.

Extract information using DNS Zone Transfer

Enumeration through the DNS zone transfer process includes extracting information such as the DNS server's location, DNS Records, and other valuable network-related information like hostname, IP address, username, etc. A zone transfer is a process of updating DNS servers; a zone file carries valuable information that an attacker can retrieve. UDP port 53 is used for DNS requests. TCP 53 is utilized for DNS zone transfers to ensure that the transfer goes through.

A network administrator can use DNS zone transfer to replicate DNS data across servers or back up DNS files by executing a zone transfer request. If permitted, the name server will convert DNS names and IP addresses to ASCII text. Improperly configured DNS servers can inadvertently expose organizational information, including lists of hosts, sub-zones, and IP addresses. Zone transfers can be performed using nslookup and dig commands.

Extract user groups from Windows

To obtain user groups from Windows, the individual must have a registered account as a user within the Active Directory. The individual can subsequently retrieve information from the groups they belong to by utilizing either the Windows interface or command-line approaches.

Extract usernames using SNMP

Enumeration using SNMP is a process of collecting information through SNMP. The attacker utilizes default community strings or attempts to guess the string in order to retrieve information pertaining to a device. The SNMP protocol was developed to allow administrators to manage devices such as servers, routers, switches, and workstations on an IP network. It enables network administrators to manage the performance of the network, troubleshoot and resolve network problems, as well as design a highly available and scalable plan for network growth. SNMP is an application layer protocol. It provides communication between managers and agents. The SNMP system consists of three elements:

- SNMP Manager
- SNMP Agents (managed node)
- Management Information Base (MIB)

Attackers can effortlessly deduce read-only or read-write community strings by leveraging the SNMP Application Programming Interface (API) to retrieve usernames.

Extract network resources and topology using SNMP

Attackers can systematically probe the SNMP hierarchy to collect comprehensive data regarding network resources and topology.

Services and Ports to Enumerate

The Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) regulate how data is communicated between terminals within a network.

TCP is a connection-oriented protocol that enables the transmission of messages or emails across the Internet. It ensures a reliable communication service for multiple processes in a multi-network setting. The key features and functionalities of TCP include:

- It supports acknowledgments for data received via a sliding window acknowledgment mechanism
- It automatically retransmits lost or acknowledged data
- It facilitates the addressing and multiplexing of data
- It allows for the establishment, management, and termination of connections
- It provides a quality-of-service transmission
- It includes mechanisms for congestion management and flow control

UDP is a connectionless protocol that transmits short messages across a computer network and offers an unreliable service. The use cases for UDP encompass:

- Audio streaming
- Videoconferencing and teleconferencing

The services related to TCP/UDP ports that can be identified are mentioned in Table 4-01.

Services	Ports
DNS Zone Transfer	TCP/UDP 53
DNS Queries	UDP 53
SNMP	UDP 161
SNMP Trap	TCP/UDP 162
Microsoft RPC Endpoint Mapper	TCP/UDP 135
LDAP	TCP/UDP 389
NBNS	UDP 137
Global Catalog Service	TCP/UDP 3268
NetBIOS	TCP 139
SMTP	TCP 25
SMB over TCP	TCP/UDP 445
NFS	TCP 2049
ISAKMP/IKE	UDP 500
SSH/SFTP	TCP 22
SIP	TCP/UDP 5060,5061
FTP	TCP 20/21
Telnet	TCP 23
TFTP	UDP 69

Table 4-01: Services and Ports to Enumerate

TCP/UDP 53: DNS Zone Transfer

The process of DNS resolution facilitates the connection between DNS clients and DNS servers. DNS clients transmit DNS messages to DNS servers that are active on UDP port 53. When the size of a DNS message exceeds the standard limit of UDP (512 octets), the response includes only the information that fits within the UDP size limitation, and the DNS server sets a flag to denote that the response has been truncated. The DNS client may then resend the request using TCP through port 53 to the DNS server. In this method, the default protocol utilized by the DNS server is UDP. For lengthy queries where UDP fails, TCP serves as a backup option. Malicious software like the ADM worm and Bonk Trojan takes advantage of port 53 to exploit vulnerabilities in DNS servers, facilitating attacks by intruders.

TCP/UDP 135: Microsoft RPC Endpoint Mapper

RPC is a protocol that enables a client system to request services from a server. An endpoint refers to the protocol port where the server awaits RPC requests from the client. The RPC Endpoint Mapper allows RPC clients to identify the port number currently assigned to a particular RPC service. There exists a vulnerability in the section of RPC responsible for transmitting messages over TCP/IP. Improper handling of malformed messages leads to failures. This issue impacts the RPC Endpoint Mapper, which operates on TCP/IP port 135. This security flaw could permit an attacker to send RPC messages to the RPC Endpoint Mapper process on a server, potentially resulting in a Denial-of-Service (DoS) attack.

UDP 137: NetBIOS Name Service (NBNS)

NBNS, which is also referred to as the Windows Internet Name Service (WINS), offers a service for resolving names for computers that use NetBIOS. The servers for NetBIOS names keep a record of the NetBIOS names associated with hosts and the IP addresses that those hosts utilize. The purpose of NBNS is to correlate IP addresses with NetBIOS names and their associated queries. Typically, attackers target the name service as their first point of attack. Generally, NBNS employs UDP 137 as its primary transport protocol. For certain operations, it may also utilize TCP 137 as a transport protocol, although this is unlikely to happen in practice.

TCP 139: NetBIOS Session Service (SMB over NetBIOS)

TCP 139 is arguably the most recognized Windows port. This port is utilized for transferring files across a network. Systems rely on this port for both establishing null sessions and enabling file and printer sharing. A system administrator who is contemplating limiting access to ports on a Windows system should prioritize the restriction of TCP 139. A poorly configured TCP 139 port could allow an attacker to gain unauthorized access to essential system files or the entire file system, leading to data theft or other malicious activities.

TCP/UDP 445: SMB over TCP (Direct Host)

Windows enables file and printer sharing through the SMB protocol, which is hosted directly on TCP. In previous operating systems, SMB traffic needed the NetBIOS over TCP (NBT) protocol to function on TCP/IP transport. Directly hosted SMB traffic utilizes port 445 (TCP and UDP) rather than NetBIOS.

UDP 161: Simple Network Management Protocol (SNMP)

SNMP is commonly utilized in network management systems to monitor devices connected to the network, including routers, switches, firewalls, printers, and servers. It is made up of a manager and agents. The agent listens for requests on port 161 from the managers and replies to them on port 162.

TCP/UDP 389: Lightweight Directory Access Protocol (LDAP)

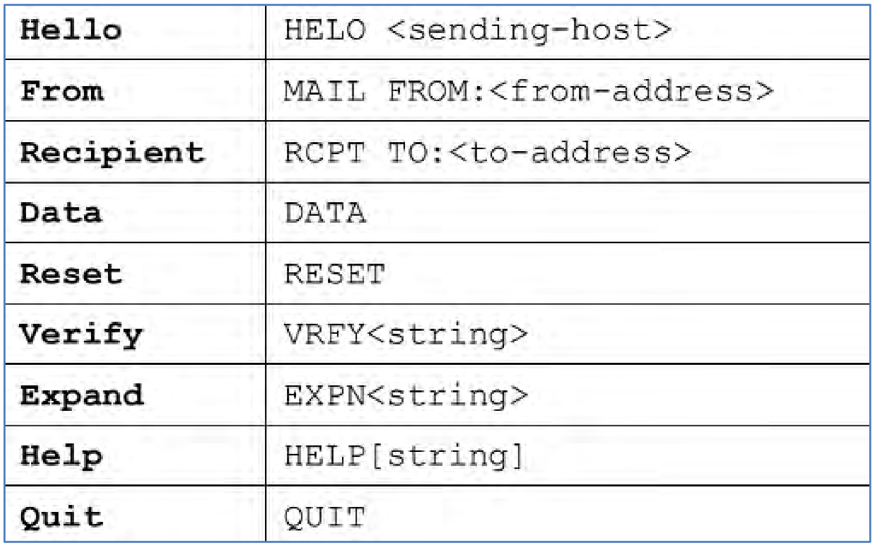
LDAP is a protocol designed for the access and management of distributed directory information services across an IP network. By default, LDAP operates over TCP or UDP as its transport protocol, utilizing port 389.

TCP 2049: Network File System (NFS)

The NFS protocol facilitates the mounting of file systems from a remote host via a network, allowing users to interact with these file systems as if they were mounted locally. NFS servers communicate with their client systems through TCP port 2049. If the NFS services are not configured correctly, attackers might exploit the NFS protocol to gain control over a remote system, execute privilege escalation, or inject backdoors or malware onto a remote host, among other threats.

TCP 25: Simple Mail Transfer Protocol (SMTP)

SMTP is a mail delivery protocol that operates over TCP/IP. It facilitates the transmission of emails over the Internet and local networks. This protocol functions on the connection-oriented services offered by TCP and utilizes the commonly recognized port number 25. The following table outlines several commands utilized by SMTP along with their corresponding syntaxes.



Hello	HELO <sending-host>
From	MAIL FROM:<from-address>
Recipient	RCPT TO:<to-address>
Data	DATA
Reset	RESET
Verify	VRFY<string>
Expand	EXPN<string>
Help	HELP[string]
Quit	QUIT

Table 4-02: SMTP Commands and their Respective Syntaxes

TCP/UDP 162: SNMP Trap

An SNMP trap utilizes TCP/UDP port 162 to transmit notifications, which may include optional variable bindings along with the sysUpTime value from an agent to a manager.

UDP 500: Internet Security Association and Key Management Protocol (ISAKMP)/Internet Key Exchange (IKE)

The Internet Security Association and Key Management Protocol (ISAKMP), along with Internet Key Exchange (IKE), is a protocol employed to create a Security Association (SA) within the IPsec protocol suite. It utilizes UDP port 500 to initiate, negotiate, alter, and remove SAs and cryptographic keys in a Virtual Private Network (VPN) setting.

TCP 22: Secure Shell (SSH) / Secure File Transfer Protocol (SFTP)

Secure Shell (SSH) is a command-line protocol primarily utilized for the secure management of various network devices. Typically, it serves as a secure alternative to the less secure Telnet protocol. SSH operates on a client/server model, with the SSH server, by default, listening for clients on TCP port 22. Attackers can exploit the SSH protocol by attempting to brute-force SSH login credentials.

By default, SFTP also operates on port 22, allowing for the secure transfer of data over a single Internet connection. The use of this designated port for SFTP improves its security and ease of use in comparison to protocols like FTP/S, which need multiple ports to function. The fact that SFTP depends on a single port simplifies its implementation while still ensuring secure data transfers through SSH encryption. Attackers may scan SFTP to obtain information regarding user accounts, file and directory permissions, and the configuration of the server.

TCP/UDP 3268: Global Catalog Service

Microsoft's Global Catalog server serves as a domain controller that holds additional information and operates on port 3268. Its database comprises entries for every object in the entire organization rather than just those in a single domain. The Global Catalog enables users to find objects from any domain without needing to know the domain's name. The LDAP protocol on the Global Catalog server operates over port 3268. This service is active on port 3268 via a TCP connection. Administrators utilize port 3268 to troubleshoot with the Global Catalog by establishing a connection using LDP.

TCP/UDP 5060, 5061: Session Initiation Protocol (SIP)

The Session Initiation Protocol (SIP) is a protocol utilized in Internet telephony for making voice and video calls. It generally employs TCP/UDP port 5060 for non-encrypted signaling traffic or 5061 for encrypted communications using TLS when connecting to servers and other endpoints.

TCP 20/21: File Transfer Protocol

FTP is a protocol that establishes a connection for transferring files across the Internet and private networks. It operates on TCP port 21 for control commands, while data transfer utilizes TCP port 20 or various dynamic port numbers based on the server setup. If attackers discover that the FTP server ports are open, they carry out enumeration on the FTP service to gather details like the software version and any existing vulnerabilities, which can lead to further exploits such as intercepting FTP traffic and executing brute-force attacks against FTP.

TCP 23: Telnet

The Telnet protocol is utilized for remote management of various network devices. It is considered an insecure protocol as it sends login information in cleartext. Consequently, it is primarily employed within private networks. The Telnet server accepts connections from clients on port 23. Malicious actors can exploit the Telnet protocol for activities like banner grabbing on other protocols, including SSH and SMTP, conducting brute-force attacks on login credentials, and executing port forwarding attacks.

UDP 69: Trivial File Transfer Protocol (TFTP)

TFTP is a protocol that operates without a connection and is utilized for file transfers over the Internet. Because TFTP is reliant on connectionless UDP, it does not ensure that files are transmitted correctly to their intended destination. This protocol is primarily employed to update or upgrade the software and firmware of connected devices from a distance. It utilizes UDP port 69 for sending files to a remote server. Cyber attackers might take advantage of TFTP to introduce harmful software or firmware onto remote devices.

TCP 179: Border Gateway Protocol (BGP)

BGP is commonly utilized by Internet Service Providers (ISPs) to manage large routing tables and to efficiently handle Internet traffic. BGP routers initiate sessions using TCP port 179. Improper configuration of BGP can result in different types of attacks, including dictionary attacks, resource-exhaustion attacks, flooding attacks, and hijacking attacks.

NetBIOS Enumeration

This part outlines the process of NetBIOS enumeration, the types of information gathered, and the different tools used for NetBIOS enumeration. NetBIOS is prioritized for enumeration since it reveals a substantial amount of sensitive data regarding the target network, including details about users and shared network resources.

The initial action in gathering information about a Windows system involves utilizing the NetBIOS API. NetBIOS was initially created as an API to enable client applications to connect to resources on a Local Area Network (LAN). Windows employs NetBIOS for the sharing of files and printers.

NetBIOS stands for Network Basic Input/Output System. This program facilitates communication among various applications operating on different systems within a local area network. NetBIOS utilizes a distinct 16-character ASCII string to identify network devices via TCP/IP. The first 15 characters serve to identify the device, while the 16th character designates the service. The NetBIOS service operates on TCP port 139. NetBIOS over TCP (NetBT) employs the following TCP and UDP ports:

- UDP port 137 (name services)
- UDP port 138 (datagram services)
- TCP port 139 (session services)

Attackers commonly target the NetBIOS service because it is easy to exploit and often runs on Windows systems even when not in use. They utilize NetBIOS enumeration to gather the following information:

- List of machines within a domain
- File sharing
- Printer sharing
- Username
- Group information
- Password
- Policies

If an attacker discovers a Windows system with port 139 open, they can investigate which resources on that remote system can be accessed or viewed. For NetBIOS name enumeration to be possible, the remote system must have file and printer sharing enabled. This enumeration could allow the attacker to read from or write to the remote computer system, depending on the available shares. Additionally, it could enable the attacker to launch a Denial-of-Service (DoS) attack.

NetBIOS names are classified into the following types:

- Unique
- Group
- Domain Name
- Internet Group
- Multihomed

Name	NetBIOS Code	Type	Information Obtained
<host name>	<00>	UNIQUE	Hostname
<domain>	<00>	GROUP	Domain name
<host name>	<03>	UNIQUE	Messenger service running for the computer
<username>	<03>	UNIQUE	Messenger service running for the logged-in user
<host name>	<20>	UNIQUE	Server service running
<domain>	<1D>	GROUP	Master browser name for the subnet
<domain>	<1B>	UNIQUE	Domain master browser name (Primary Domain Controller)
<domain>	<1E>	GROUP	Browser service elections

Table 4-03: NetBIOS Name List

EXAM TIP: Microsoft does not provide support for NetBIOS name resolution in IPv6.

Nbtstat Utility

Nbtstat is a utility in Windows designed to assist in diagnosing issues related to NETBIOS name resolution. The nbtstat command is utilized to eliminate and rectify preloaded entries with the help of various case-sensitive options. Attackers may leverage Nbtstat to gather details such as statistics on the NetBIOS over TCP/IP (NetBT) protocol, as well as the NetBIOS name tables for both local and remote systems, along with the NetBIOS name cache.

The command syntax for nbtstat is outlined as follows:

```
nbtstat [-a <remotename>] [-A <IPaddress>] [-c] [-n] [-r] [-R] [-RR] [-s] [-S] [<interval>][-?]
```

Table 4-04 lists various Nbtstat parameters and their respective functions.

Nbtstat Parameter	Function
-a <remotename>	Displays the NetBIOS name table of a remote computer, where <remotename> is the NetBIOS name.
-A <IPaddress>	Displays the NetBIOS name table of a remote computer, specified by the IP address.
-c	Lists the contents of the NetBIOS name cache and resolved IP addresses.
-n	Displays names registered locally by NetBIOS applications.
-r	Displays a count of all names resolved by broadcast or WINS server.
-R	Purges the name cache and reloads all #PRE-tagged entries from the LMHOSTS file.
-RR	Releases and re-registers all names with the name server.
-s	Lists the NetBIOS sessions table, converting destination IP addresses to NetBIOS names.
-S	Lists current NetBIOS sessions and their status with IP addresses.
<interval>	Re-displays selected statistics, pausing for the specified interval (in seconds).
-?	Displays help for the nbtstat command.

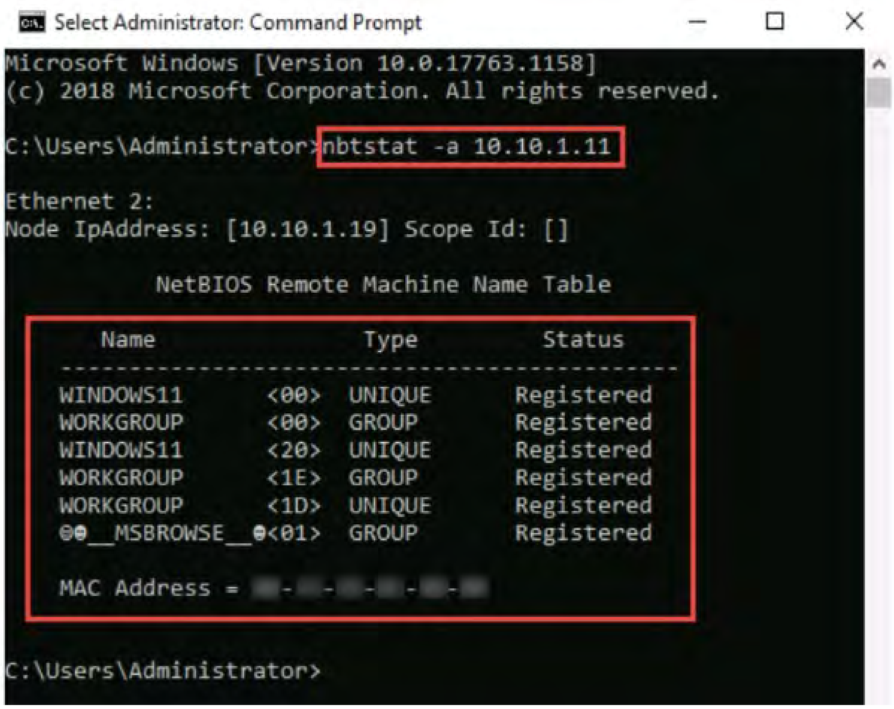
Table 4-04: Nbtstat Parameters and their Respective Functions

The following are some examples of nbtstat commands:

Example 1:

```
nbtstat –a <IP address of the remote machine>
```

This command can be run to retrieve the NetBIOS name table of a remote computer.



Select Administrator: Command Prompt

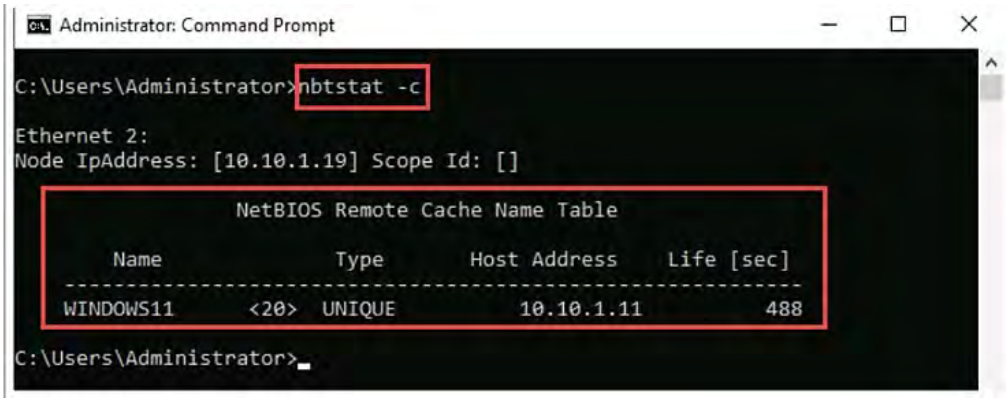
```
Microsoft Windows [Version 10.0.17763.1158]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>nbtstat -a 10.10.1.11

Ethernet 2:
Node IpAddress: [10.10.1.19] Scope Id: []

           NetBIOS Remote Machine Name Table

       Name               Type         Status
    ---------------------------------------------
    WINDOWS11      <00>  UNIQUE      Registered
    WORKGROUP      <00>  GROUP       Registered
    WINDOWS11      <20>  UNIQUE      Registered
    WORKGROUP      <1E>  GROUP       Registered
    WORKGROUP      <1D>  UNIQUE      Registered
    ☺☻__MSBROWSE__☻<01>  GROUP       Registered

    MAC Address = [redacted]

C:\Users\Administrator>
```

Figure 4-01: Nbtstat Command to Obtain the Name Table of a Remote System

Example 2:

```
nbtstat –c
```

This command can be run to retrieve the contents of the NetBIOS name cache, which includes a list of NetBIOS names and their associated resolved IP addresses.



Administrator: Command Prompt

```
C:\Users\Administrator>nbtstat -c

Ethernet 2:
Node IpAddress: [10.10.1.19] Scope Id: []

                  NetBIOS Remote Cache Name Table

        Name              Type       Host Address    Life [sec]
    ------------------------------------------------------------
    WINDOWS11      <20>  UNIQUE          10.10.1.11          488

C:\Users\Administrator>_
```

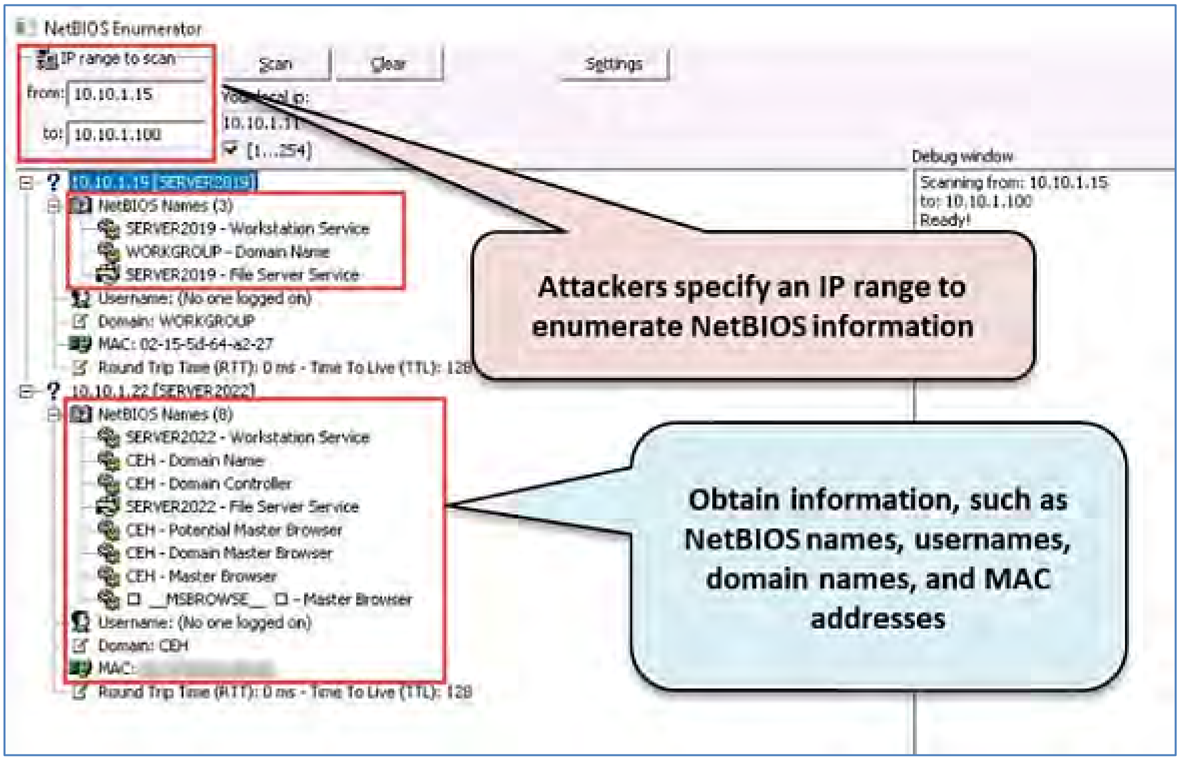
Figure 4-02: Nbtstat Command to obtain the Contents of the NetBIOS Name Table

NetBIOS Enumeration Tools

NetBIOS enumeration tools investigate and scan a network across specific ranges of IP addresses and lists of computers to uncover security vulnerabilities or weaknesses in connected systems. These tools also identify Operating Systems (OSs), users, groups, Security Identifiers (SIDs), password policies, services, service packs and hotfixes, NetBIOS shares, transports, sessions, disks, and security event logs, among other things.

NetBIOS Enumerator

The NetBIOS Enumerator is a tool used for enumeration that demonstrates how to utilize remote network support and manage various web protocols, including SMB. As illustrated in Figure 4-03, attackers utilize the NetBIOS Enumerator to gather information such as NetBIOS names, usernames, domain names, and Media Access Control (MAC) addresses within a specified range of IP addresses.



from: 10.10.1.15
to: 10.10.1.100

10.10.1.11
☑ [1...254]

```
10.10.1.19 [SERVER2019]
  NetBIOS Names (3)
    SERVER2019 - Workstation Service
    WORKGROUP - Domain Name
    SERVER2019 - File Server Service
  Username: (No one logged on)
  Domain: WORKGROUP
  MAC: 02-15-5d-64-a2-27
  Round Trip Time (RTT): 0 ms - Time To Live (TTL): 128
10.10.1.22 [SERVER2022]
  NetBIOS Names (8)
    SERVER2022 - Workstation Service
    CEH - Domain Name
    CEH - Domain Controller
    SERVER2022 - File Server Service
    CEH - Potential Master Browser
    CEH - Domain Master Browser
    CEH - Master Browser
    □ __MSBROWSE__ □ - Master Browser
  Username: (No one logged on)
  Domain: CEH
  MAC: [redacted]
  Round Trip Time (RTT): 0 ms - Time To Live (TTL): 128
```

```
Scanning from: 10.10.1.15
to: 10.10.1.100
Ready!
```

Attackers specify an IP range to enumerate NetBIOS information

Obtain information, such as NetBIOS names, usernames, domain names, and MAC addresses

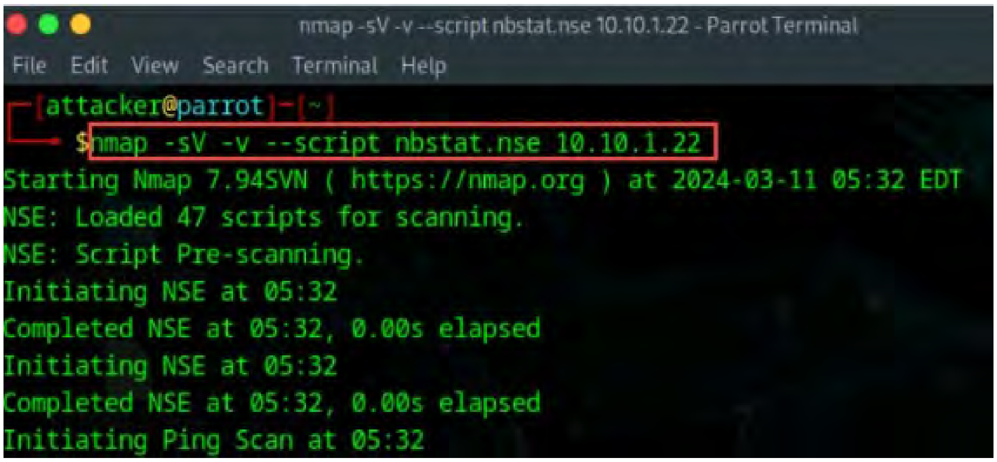
Figure 4-03: Screenshot of NetBIOS Enumerator

Nmap

Attackers utilize the Nmap Scripting Engine (NSE) to identify NetBIOS shares within a network. The NSE nbstat script enables attackers to obtain the NetBIOS names and MAC addresses of the target. By default, this script reveals the name of the machine and the user currently logged in. However, if the verbosity is turned up, it will show all names associated with that system.

As demonstrated in Figure 4-04, an attacker executes the following Nmap command to carry out NetBIOS enumeration on a specified host:

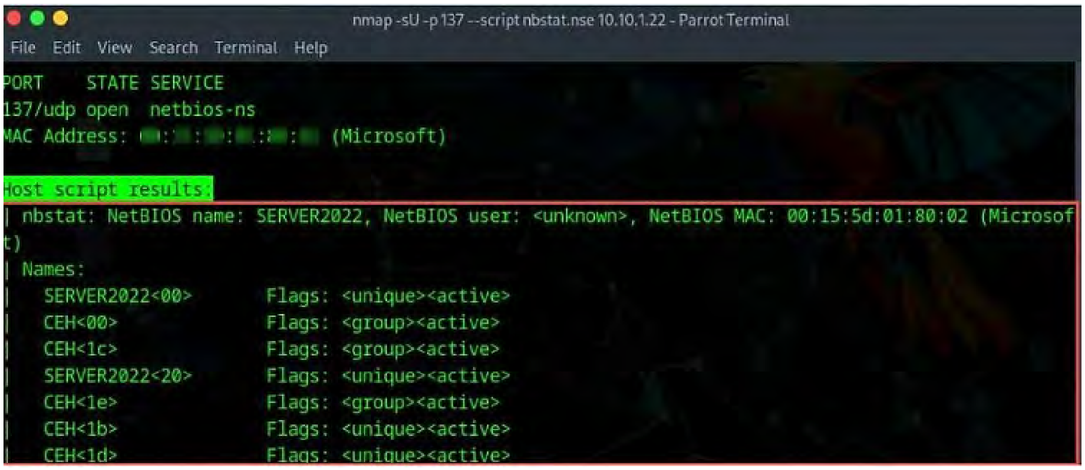
```
nmap -sV -v --script nbstat.nse <target IP address>
```



nmap -sV -v --script nbstat.nse 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $nmap -sV -v --script nbstat.nse 10.10.1.22
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-11 05:32 EDT
NSE: Loaded 47 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 05:32
Completed NSE at 05:32, 0.00s elapsed
Initiating NSE at 05:32
Completed NSE at 05:32, 0.00s elapsed
Initiating Ping Scan at 05:32
```

Figure 4-04: Nmap Command for NetBIOS Enumeration



nmap -sU -p 137 --script nbstat.nse 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
PORT    STATE SERVICE
137/udp open  netbios-ns
MAC Address: [redacted] (Microsoft)

Host script results:
| nbstat: NetBIOS name: SERVER2022, NetBIOS user: <unknown>, NetBIOS MAC: 00:15:5d:01:80:02 (Microsof
t)
| Names:
|   SERVER2022<00>       Flags: <unique><active>
|   CEH<00>              Flags: <group><active>
|   CEH<1c>              Flags: <group><active>
|   SERVER2022<20>       Flags: <unique><active>
|   CEH<1e>              Flags: <group><active>
|   CEH<1b>              Flags: <unique><active>
|   CEH<1d>              Flags: <unique><active>
```

Figure 4-05: Nmap NetBIOS Enumeration Output

Enumerating User Accounts

Using the PsTools suite to list user accounts aids in the management and oversight of remote systems via the command line. Below are some commands for enumerating user accounts.

PsExec

PsExec serves as a compact replacement for Telnet, allowing processes to run on different machines while maintaining full interaction for console applications, all without the need to manually install client software. One of the most effective applications of PsExec is to initiate interactive command prompts on remote machines and to allow tools like ipconfig to access information about those remote systems, which they otherwise could not.

The syntax for using the PsExec command is structured as follows:

```
psexec [\\computer[,computer2[,...] | @file]][-u user [-p psswd][-n  s][-r servicename][-h][-l][-s|-e][-x][-i [session]][-c
executable [-f|-v]][-w directory][-d][-<priority>][-a n,n,...] cmd [arguments]
```

PsFile

PsFile is a command-line tool that displays a list of files on a system that have been accessed remotely, and it can terminate opened files by either their name or file identifier. By default, PsFile lists the files on the local system that are opened by remote devices. Entering a command followed by a "-" will show the syntax information for that command.

The syntax for using the PsFile command is as follows:

```
psfile [\\RemoteComputer [-u Username [-p Password]]] [[Id | path] [-c]]
```

PsGetSid

PsGetSid converts SIDs into their corresponding display names and the other way around. It functions with built-in accounts, domain accounts, and local accounts. Additionally, it shows the SIDs associated with user accounts and can convert an SID into the name that corresponds to it. It is capable of querying SIDs remotely across the network.

The command syntax for PsGetSid is as follows:

```
psgetsid [\\computer[,computer[,...] | @file] [-u username [-p password]]] [account|SID]
```

PsKill

PsKill is a utility designed to terminate processes on both remote systems and the local machine. By specifying a process ID, PsKill will kill the corresponding process on the local computer. If a process name is provided, PsKill will kill all processes of that name. It is not necessary to install a client on the target machine in order to use PsKill to end a remote process.

The command structure for PsKill is as follows:

```
pskill [-] [-t] [\\computer [-u username] [-p password]] <process name | process id>
```

PsInfo

PsInfo is a command-line utility that collects essential details about local or remote legacy Windows systems, such as the type of installation, the kernel version, the registered owner and organization, the number of processors along with their types, the size of physical memory, the system's installation date, and the expiration date for trial versions. By default, PsInfo displays information for the local machine. To retrieve details from a remote computer, you can specify its name.

The syntax for using the PsInfo command is as follows:

```
psinfo [[\\computer[,computer[,..] | @file [-u user [-p psswd]]] [-h] [-s] [-d] [-c [-t delimiter]]
[filter]
```

PsList

PsList is a command-line utility that provides information on Central Processing Unit (CPU) usage and memory statistics, as well as thread data. The tools from the Resource Kits, pstat and pmon present various types of information solely for the processes running on the respective system where the tools are executed.

PsLoggedOn

PsLoggedOn is a utility that shows both users who are logged in locally and those accessing the system through shared resources, whether on the local machine or a remote one. When a username is provided instead of a computer name, PsLoggedOn scans the computers in the network neighborhood to check if that user is currently signed in. PsLoggedOn defines a user logged in locally as one whose profile is loaded into the registry. Consequently, PsLoggedOn identifies logged-in users by examining the keys located under the HKEY_USERS key. For each key with a name or user SID, PsLoggedOn retrieves the matching username and presents it. To find out which users have accessed a computer via resource shares, PsLoggedOn employs the NetSessionEnum API.

The command syntax for PsLoggedOn is as follows:

```
psloggedon [-] [-l] [-x] [\\computername | username]
```

PsLogList

The elogdump utility allows you to extract the contents of an Event Log from either a local or remote system. PsLogList serves as a counterpart to elogdump, but it can connect to remote machines even when the user's security credentials might restrict access to the Event Log, and it gathers message strings from the computer that holds the event log. By default, PsLogList's function is to present the contents of the System Event Log on the local machine with a user-friendly visual format.

The command syntax for PsLogList is as follows:

```
psloglist [-] [\\computer[,computer[,...] | @file [-u username [-p password]]] [-s [-t delimiter]] [-m #|-n #|-h #|-d #|-w][-c][-x][-r][-a mm/dd/yy][-b mm/dd/yy][-f filter] [-i ID[,ID[,...] | -e ID[,ID[,...]]] [-o event source[,event source][,..]]] [-q event source[,event source][,..]]] [-l event log file] <eventlog>
```

PsPasswd

PsPasswd allows for changing an account's password on both local and remote machines, and system administrators can develop batch files to execute PsPasswd on the systems they oversee to execute a mass password change for the administrator. PsPasswd utilizes Windows password reset APIs, ensuring that passwords are not transmitted over the network in plaintext format.

The command syntax for PsPasswd is as follows:

```
pspasswd [[\\computer[,computer[,..] | @file [-u user [-p psswd]]] Username [NewPassword]
```

PsShutdown

PsShutdown is capable of shutting down or restarting both local and remote computers. It does not need any manual installation of client software.

The command syntax for PsShutdown is structured as follows:

```
psshutdown [[\\computer[,computer[,..] | @file [-u user [-p psswd]]] -s|-r|-h|-d|-k|-a|-l|-o [-f] [-c] [-t nn|h:m] [-n s] [-v nn] [-e [u|p]:xx:yy] [-m "message"]
```

Enumerating Shared Resources using Netview

Net View is a command-line tool that shows a list of computers within a designated workgroup or reveals shared resources present on a specific computer. It can be utilized in the following ways:

```
net view \\<computername>
```

In the command mentioned above, **<computername>** refers to the name or IP address of a particular computer whose resources you want to view.

```
net view \\<computername> /ALL
```

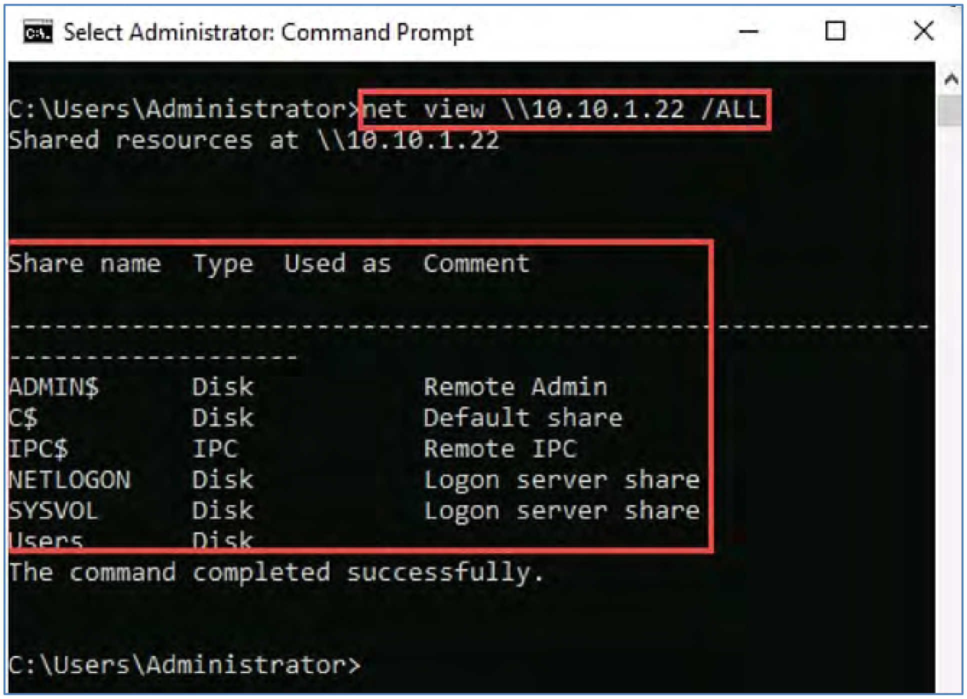
The command mentioned above reveals all the shared resources on the designated remote machine, including any hidden shares.

```
net view /domain
```

The above command displays all the shares in the domain.

```
net view /domain:<domain name>
```

This command displays all the shares on the specified domain.



```
C:\Users\Administrator>net view \\10.10.1.22 /ALL
Shared resources at \\10.10.1.22



Share name   Type   Used as   Comment


------------------------------------------------------------
-------------------
ADMIN$       Disk             Remote Admin
C$           Disk             Default share
IPC$         IPC              Remote IPC
NETLOGON     Disk             Logon server share
SYSVOL       Disk             Logon server share
Users        Disk
The command completed successfully.


C:\Users\Administrator>
```

Figure 4-06: Output of Net View Command

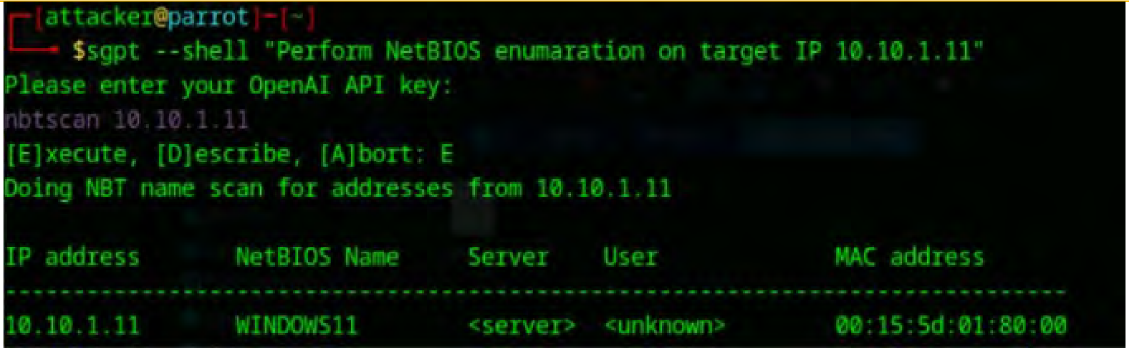
NetBIOS Enumeration using AI

Attackers can utilize AI-driven technologies to improve and automate their network enumeration activities. By harnessing AI, these individuals can easily conduct NetBIOS enumeration to gather NetBIOS details, related names and identify NetBIOS services on specific IP addresses.

Attackers may employ ChatGPT for this purpose by using suitable prompts like:

Example 1:

Perform NetBIOS enumeration on target IP 10.10.1.11.



```
┌─[attacker@parrot]─[~]
└──╼ $sgpt --shell "Perform NetBIOS enumaration on target IP 10.10.1.11"
Please enter your OpenAI API key:
nbtscan 10.10.1.11
[E]xecute, [D]escribe, [A]bort: E
Doing NBT name scan for addresses from 10.10.1.11

IP address       NetBIOS Name     Server    User             MAC address
------------------------------------------------------------------------------
10.10.1.11       WINDOWS11        <server>  <unknown>        00:15:5d:01:80:00
```

Figure 4-07: Perform NetBIOS Enumeration on Target IP

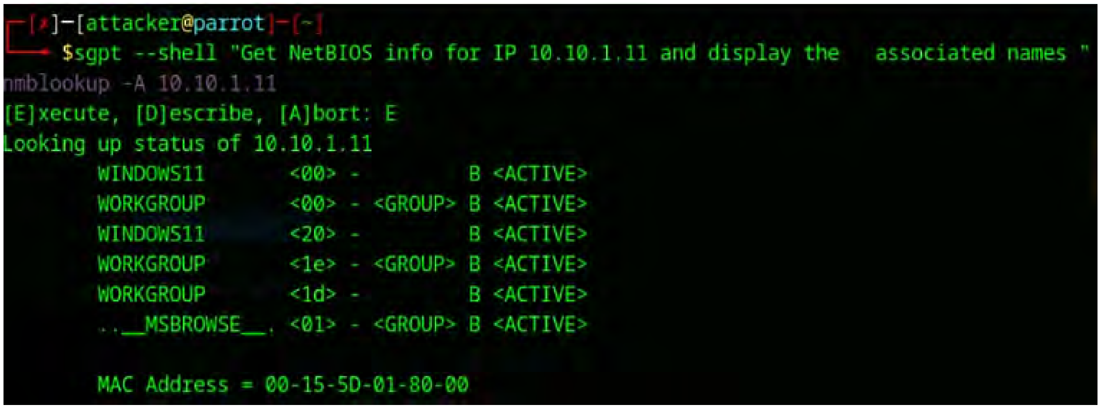
The following prompt is designed to automate NetBIOS enumeration on the specified target IPs:

```
nbtscan 10.10.1.11
```

The **nbtscan** command is utilized to conduct NetBIOS enumeration on the specified IP address 10.10.1.11.

Example 2:

Get NetBIOS info for IP 10.10.1.11 and display the associated names.



```
┌─[✗]─[attacker@parrot]─[~]
└──╼ $sgpt --shell "Get NetBIOS info for IP 10.10.1.11 and display the   associated names "
nmblookup -A 10.10.1.11
[E]xecute, [D]escribe, [A]bort: E
Looking up status of 10.10.1.11
        WINDOWS11       <00> -         B <ACTIVE>
        WORKGROUP       <00> - <GROUP> B <ACTIVE>
        WINDOWS11       <20> -         B <ACTIVE>
        WORKGROUP       <1e> - <GROUP> B <ACTIVE>
        WORKGROUP       <1d> -         B <ACTIVE>
        ..__MSBROWSE__. <01> - <GROUP> B <ACTIVE>

        MAC Address = 00-15-5D-01-80-00
```

Figure 4-08: Perform NetBIOS Enumeration on Target IP and Display the Names

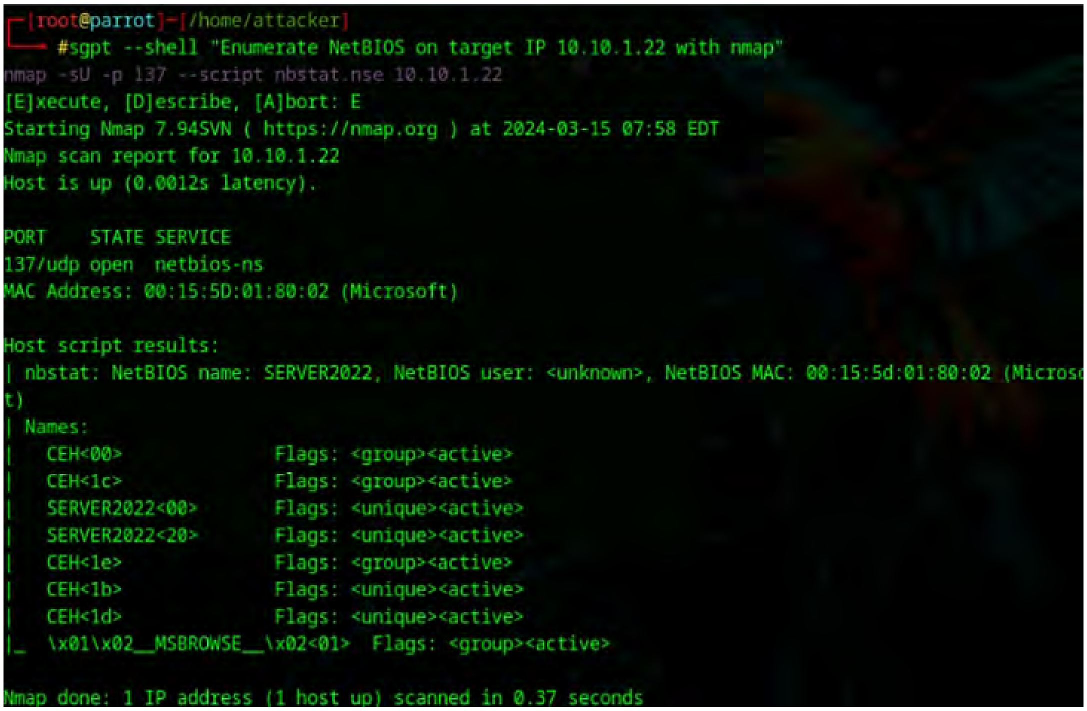
The following prompt is designed to automate NetBIOS enumeration on the specified target IPs:

```
nmblookup -A 10.10.1.11
```

The **nmblookup** command, when used with the **-A** option, retrieves NetBIOS details for the specified IP address 10.10.1.11 and shows the corresponding names.

Example 3:

Enumerate NetBIOS on target IP 10.10.1.22 with nmap.



```
┌─[root@parrot]─[/home/attacker]
└──╼ #sgpt --shell "Enumerate NetBIOS on target IP 10.10.1.22 with nmap"
nmap -sU -p 137 --script nbstat.nse 10.10.1.22
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 07:58 EDT
Nmap scan report for 10.10.1.22
Host is up (0.0012s latency).

PORT    STATE SERVICE
137/udp open  netbios-ns
MAC Address: 00:15:5D:01:80:02 (Microsoft)

Host script results:
| nbstat: NetBIOS name: SERVER2022, NetBIOS user: <unknown>, NetBIOS MAC: 00:15:5d:01:80:02 (Microso
t)
| Names:
|   CEH<00>             Flags: <group><active>
|   CEH<1c>             Flags: <group><active>
|   SERVER2022<00>      Flags: <unique><active>
|   SERVER2022<20>      Flags: <unique><active>
|   CEH<1e>             Flags: <group><active>
|   CEH<1b>             Flags: <unique><active>
|   CEH<1d>             Flags: <unique><active>
|_  \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

Figure 4-09: Perform NetBIOS Enumeration on Target IP with nmap

The following prompt is designed to automate NetBIOS enumeration on the specified target IPs:

```
nmap -sU -p 137 --script nbstat.nse 10.10.1.22
```

Script **nbstat.nse** is utilized to gather information about NetBIOS services on the specified IP 10.10.1.22.

These commands streamline the process of NetBIOS enumeration and deliver comprehensive details regarding NetBIOS services on the given target IPs.

SNMP Enumeration

The Simple Network Management Protocol (SNMP) enables network administrators to manage devices remotely. SNMP enumeration is a method used to gather information about user accounts and devices by targeting the most commonly utilized network management protocol, SNMP. SNMP requires a community string to authenticate the management station.

Nonetheless, SNMP has numerous security weaknesses, including a lack of auditing capabilities. Attackers can exploit these vulnerabilities for account and device enumeration. This section outlines SNMP enumeration, the types of information obtained through this process, and various tools utilized for enumerating user accounts and devices on a target system.

SNMP is an application-layer protocol that operates over UDP and is responsible for maintaining and managing routers, hubs, and switches within an IP network. Agents of SNMP can be found on both Windows and Unix networks within networking devices.

SNMP enumeration refers to the method of compiling a list of user accounts and devices present on a target computer via SNMP. SNMP communicates using two main software components: the SNMP agent and the SNMP management station. The SNMP agent resides on the networking device, while the SNMP management station interacts with the agent.

Nearly all infrastructure devices in a network, like routers and switches, have an SNMP agent for system or device management. The SNMP management station dispatches requests to the agent, which then responds after processing the request. Both the requests and responses involve configuration variables that the agent software can access. The SNMP management stations send requests to modify certain variable values. Traps inform the management station about any abnormal events, such as a reboot or an interface failure, occurring on the agent's side.

SNMP includes two passwords that are used for configuring and accessing the SNMP agent from the management station.

1. **Read Community String**
    - This password allows users to view the configuration of the device or system.
    - These strings are considered public
2. **Read/Write Community String**
    - This password enables users to modify or edit the device configuration
    - These strings are deemed private

When administrators leave the community strings set to their default values, attackers can exploit these default community strings (passwords) to change or access the device or system's configuration. Attackers perform SNMP enumeration to gather information about network resources such as hosts, routers, devices, and shares, along with network information, including ARP tables, routing tables, device-specific details, and traffic statistics.

Common tools for SNMP enumeration include OpUtils (https://www.manageengine.com) and Network Performance Monitor (https://www.solarwinds.com).

Working of SNMP

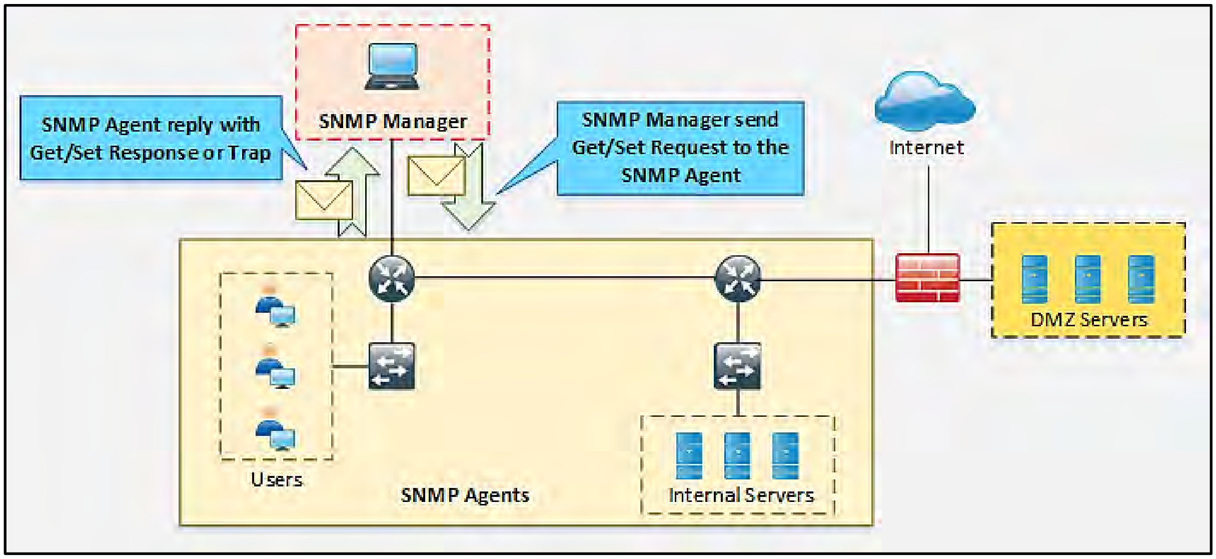

SNMP Agent reply with Get/Set Response or Trap
SNMP Manager
SNMP Manager send Get/Set Request to the SNMP Agent
Internet
DMZ Servers
Users
SNMP Agents
Internal Servers

Figure 4-10: SNMP Working

SNMP operates on a distributed architecture that includes SNMP managers, SNMP agents, and various associated components. The interaction process between an SNMP manager and an SNMP agent unfolds as follows:

Initialization

Start-Up: When a network device powers on, the SNMP agent on that device sets up its configuration and readies itself for communication with the SNMP manager by listening on the specified port (commonly UDP port 161).

Discovery

Manager Discovers Agents: The SNMP manager identifies SNMP-enabled devices within the network by dispatching a request to the broadcast address or particular IP addresses known to host agents.

Information Exchange

The exchange of information between an SNMP manager and an agent entails several types of operations, predominantly utilizing SNMP messages known as Protocol Data Units (PDUs). The main operations consist of:

a. Get Request

The SNMP manager transmits a Get Request to an SNMP agent to obtain the value of a specific variable, such as the operational status of a router interface or the bandwidth consumption on a network link.

b. GetNext Request

This request is employed to retrieve the subsequent variable in the MIB (Management Information Base) tree. It enables the manager to query a series of variables without knowing their precise names.

c. Set Request

The SNMP manager utilizes Set Requests to alter the value of a variable in the agent's MIB, thereby modifying the configuration or functionality of the network device.

d. GetBulk Request

Introduced in SNMPv2, this function facilitates the acquisition of large amounts of data with a single request, enhancing efficiency compared to multiple GetNext Requests.

e. Response

Following the reception of a Get, GetNext, Set, or GetBulk request, the SNMP agent evaluates the request, executes the required actions, and returns a Response PDU containing the requested data or an acknowledgment of the executed action.

f. Inform Request

An SNMP agent employs Inform Requests to convey unsolicited information to the SNMP manager, typically regarding significant occurrences or errors. This mechanism is also utilized for communication between managers.

g. Trap

Traps are unsolicited notifications dispatched from an SNMP agent to the manager to inform it of important events or network changes, such as a device reboot or a link failure. SNMPv3 introduced the idea of Notifications, which encapsulates both Traps and Informs, providing added authentication and encryption.

Monitoring and Management

The SNMP manager leverages the data gathered from SNMP agents to assess network performance, identify and resolve issues, and manage network devices remotely. This continuous process involves regular polling (sending Get Requests) and monitoring for Traps or Inform Requests from agents.

Management Information Base (MIB)

MIB serves as a virtual database that offers a formal outline of all network entities managed by SNMP. It consists of information that is organized hierarchically. This structure supplies a standardized representation of the data and storage of the SNMP agent. The elements within the MIB are identified by Object Identifiers (OIDs). An OID is a numeric designation assigned to an object, starting from the root of the MIB tree. It can distinctly identify the object within the MIB hierarchy.

Objects managed by the MIB include scalar objects, which specify a single instance, and tabular objects, which outline a collection of related instances. OIDs encompass the object's type (such as counter, string, or address), the access level (like read or read/write), size limitations, and range specifications. The SNMP manager translates the OIDs into a format that is easily understood by using the MIB as a reference guide.

Users can view the MIB contents through a web browser by entering either the IP address followed by Lseries.mib or the DNS library name with Lseries.mib. For instance, they can use http://IP.Address/Lseries.mib or http://library_name/Lseries.mib. Microsoft offers a list of MIBs included with the SNMP service in the Windows resource kit. The key MIBs include:

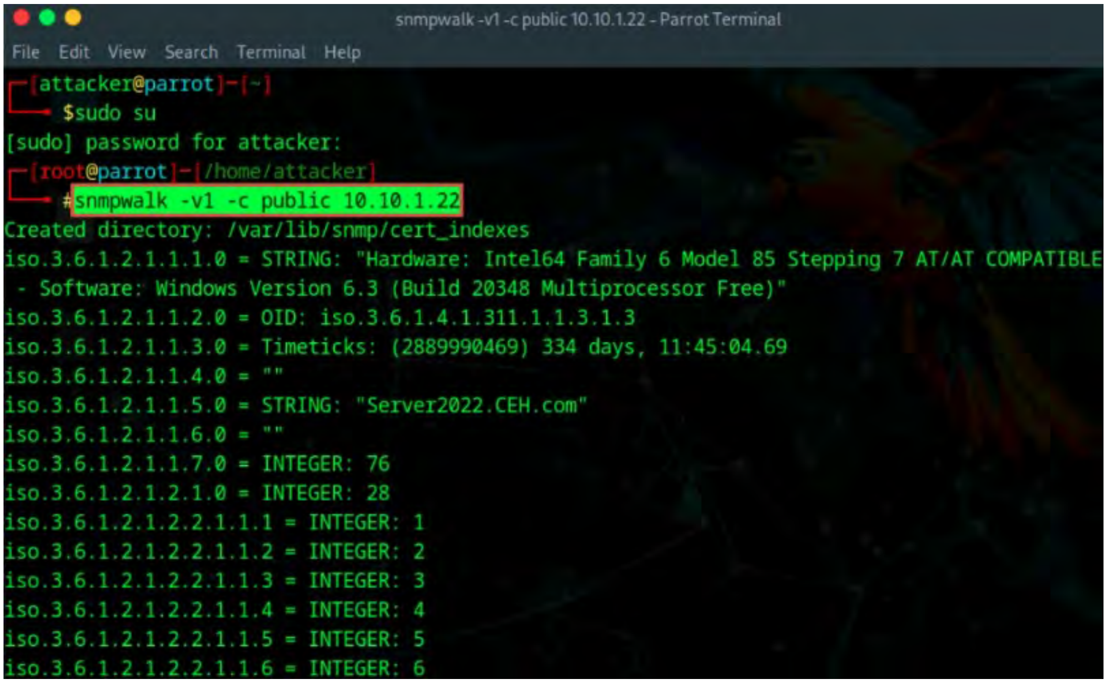
- **DHCP.MIB:** Tracks network activity between DHCP servers and remote clients
- **HOSTMIB.MIB:** Oversees and regulates host resources
- **LNMIB2.MIB:** Contains object types for both workstation and server services
- **MIB_II.MIB:** Administers TCP/IP-based Internet using a simple architecture and system
- **WINS.MIB:** For the Windows Internet Name Service (WINS)

Enumerating SNMP using SnmpWalk

Simple Network Management Protocol (SNMP) nodes immediately detect and delineate a variety of variables accessible within the target network. With this tool, attackers focus on the root node to extract information from all the sub-nodes, including routers and switches. This information can be accessed through an Object Identifier (OID), which is linked to the Management Information Base (MIB) for devices that have SNMP enabled. Attackers use the following command to pull SNMP details from the target device:

```
snmpwalk -v1 -c public <Target IP Address>
```

This command permits attackers to view all OIDs, variables, and other related information. By using this command, attackers can also obtain all data traveling to the SNMP server from the SNMP agent, which includes details such as the server in use, user credentials, and various other parameters.



```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #snmpwalk -v1 -c public 10.10.1.22
Created directory: /var/lib/snmp/cert_indexes
iso.3.6.1.2.1.1.1.0 = STRING: "Hardware: Intel64 Family 6 Model 85 Stepping 7 AT/AT COMPATIBLE
 - Software: Windows Version 6.3 (Build 20348 Multiprocessor Free)"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.311.1.1.3.1.3
iso.3.6.1.2.1.1.3.0 = Timeticks: (2889990469) 334 days, 11:45:04.69
iso.3.6.1.2.1.1.4.0 = ""
iso.3.6.1.2.1.1.5.0 = STRING: "Server2022.CEH.com"
iso.3.6.1.2.1.1.6.0 = ""
iso.3.6.1.2.1.1.7.0 = INTEGER: 76
iso.3.6.1.2.1.2.1.0 = INTEGER: 28
iso.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.1.3 = INTEGER: 3
iso.3.6.1.2.1.2.2.1.1.4 = INTEGER: 4
iso.3.6.1.2.1.2.2.1.1.5 = INTEGER: 5
iso.3.6.1.2.1.2.2.1.1.6 = INTEGER: 6
```

Figure 4-11: Screenshot of SnmpWalk

Other SnmpWalk Commands

To list SNMPv2 using a community string of public, use the following command:

```
snmpwalk -v2c -c public <Target IP Address>
```

To find the installed software, use the command:

```
snmpwalk -v2c -c public <Target IP Address> hrSWInstalledName
```

To check the total RAM on the host, use the command:

```
snmpwalk -v2c -c public <Target IP Address> hrMemorySize
```

To modify an OID to a different value, use the command:

```
snmpwalk -v2c -c public <Target IP Address> <OID> <New Value>
```

To update the sysContact OID, use the following command:

```
snmpwalk -v2c -c public <Target IP Address> sysContact <New Value>
```

Enumerating SNMP using Nmap

Attackers utilize the Nmap Scripting Engine (NSE) script called snmp-processes to gather information about the SNMP services running on a remote server.

```
nmap -sU -p 161 --script=snmp-processes <Target IP Address>
```

When this Nmap command is run, it collects a list of all active SNMP processes along with their corresponding ports on the target machine.

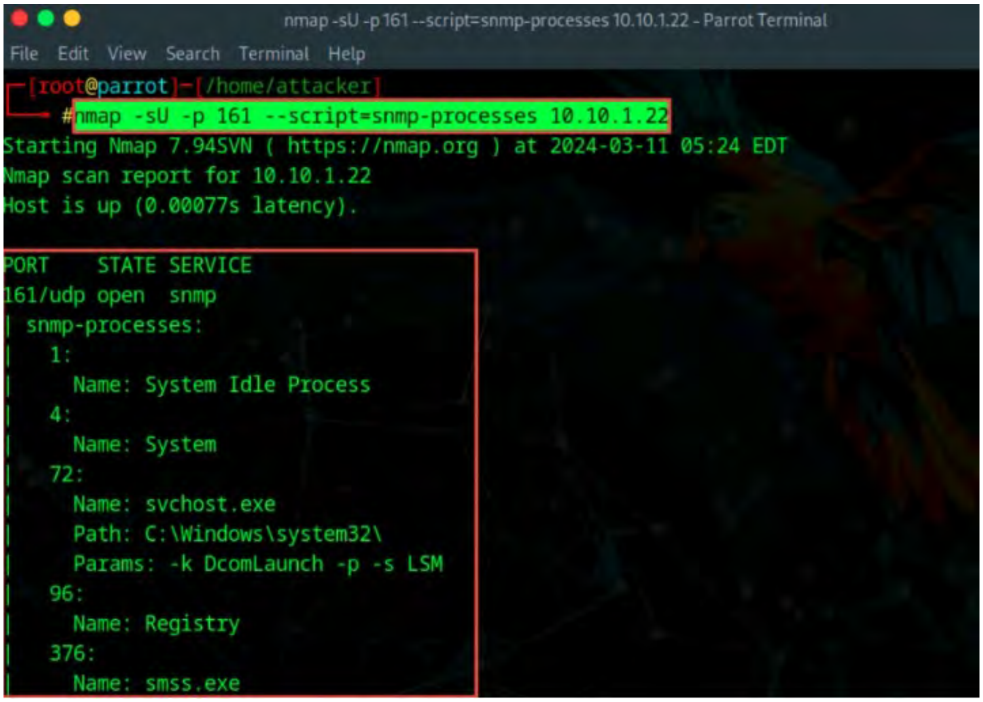
Other Nmap commands for conducting SNMP enumeration include:

```
nmap -sU -p 161 --script=snmp-sysdescr <Target IP Address>
```

This command obtains details about the type of SNMP server and the operating system.

```
nmap -sU -p 161 --script=snmp-win32-software <Target IP Address>
```

This command gathers a list of all applications currently running on the target machine.



nmap -sU -p 161 --script=snmp-processes 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sU -p 161 --script=snmp-processes 10.10.1.22
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-11 05:24 EDT
Nmap scan report for 10.10.1.22
Host is up (0.00077s latency).

PORT    STATE SERVICE
161/udp open  snmp
| snmp-processes:
|   1:
|     Name: System Idle Process
|   4:
|     Name: System
|   72:
|     Name: svchost.exe
|     Path: C:\Windows\system32\
|     Params: -k DcomLaunch -p -s LSM
|   96:
|     Name: Registry
|   376:
|     Name: smss.exe
```

Figure 4-12: Screenshot of Nmap using the snmp-processes NSE Script

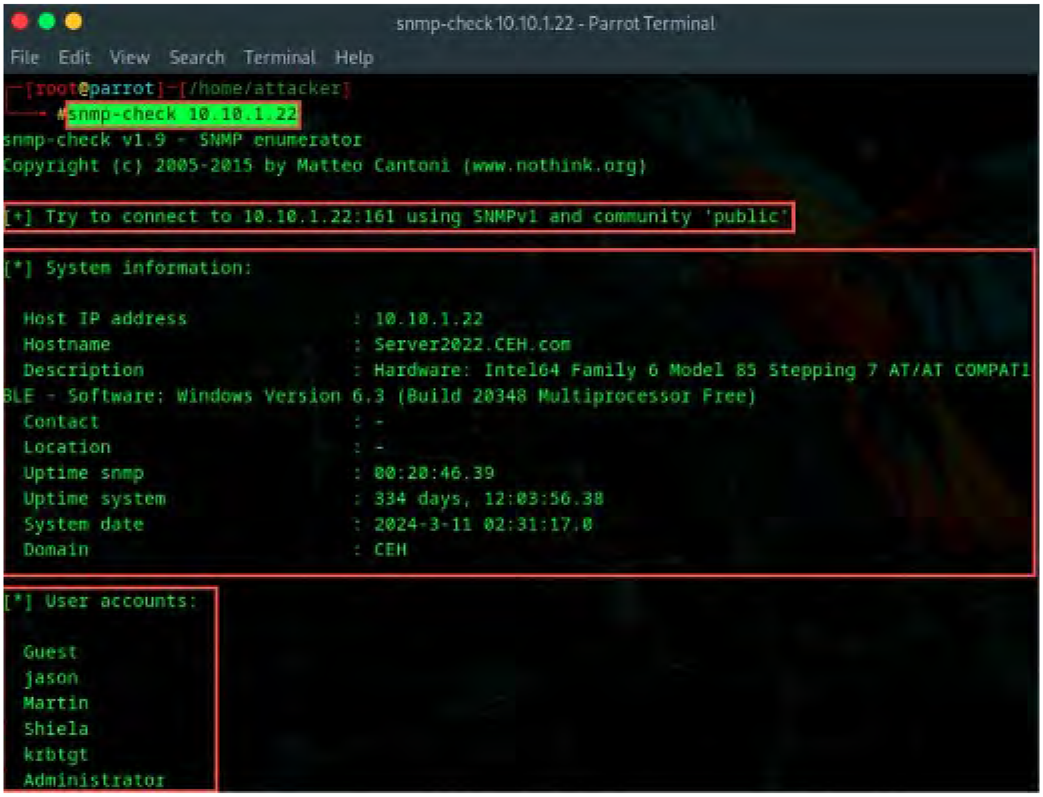
SNMP Enumeration Tools

SNMP enumeration tools scan SNMP-enabled network devices to monitor, diagnose, and troubleshoot security threats on specific IP addresses or ranges of IP addresses.

snmp-check (snmp_enum Module)

snmp-check is a free, open-source tool available under the GNU General Public License (GPL). Its purpose is to automate the collection of information from any devices that support SNMP (including Windows, Unix-like systems, network devices, printers, etc.). snmp-check facilitates the enumeration of SNMP devices and displays the outcomes in an easily understandable format. This tool can be beneficial for penetration testing or monitoring systems.

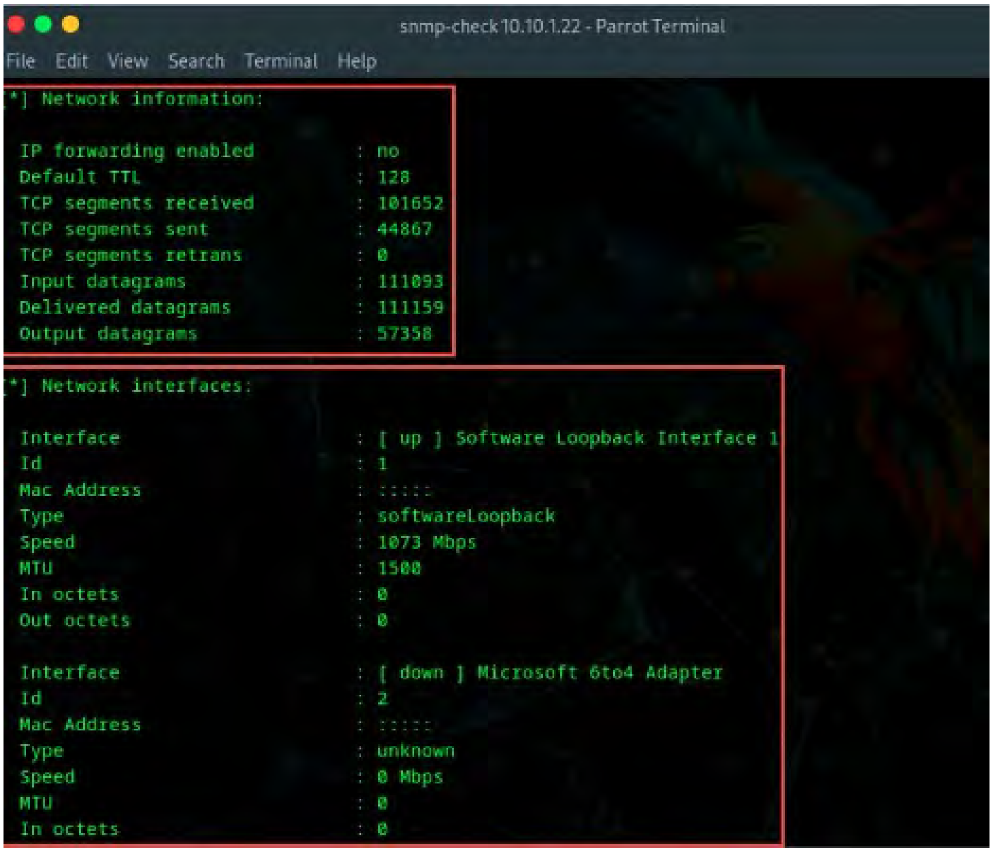
Attackers utilize this tool to collect details about the target, such as contact information, descriptions, write permissions, devices, domains, hardware and storage details, hostnames, Internet Information Services (IIS) metrics, IP forwarding settings, active UDP ports, locations, mountpoints, network interfaces, network services, routing data, software components, system uptime, TCP connections, total memory, uptime, and user accounts.



snmp-check 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #snmp-check 10.10.1.22
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 10.10.1.22:161 using SNMPv1 and community 'public'

[*] System information:

  Host IP address               : 10.10.1.22
  Hostname                      : Server2022.CEH.com
  Description                   : Hardware: Intel64 Family 6 Model 85 Stepping 7 AT/AT COMPATI
BLE - Software: Windows Version 6.3 (Build 20348 Multiprocessor Free)
  Contact                       : -
  Location                      : -
  Uptime snmp                   : 00:20:46.39
  Uptime system                 : 334 days, 12:03:56.38
  System date                   : 2024-3-11 02:31:17.0
  Domain                        : CEH

[*] User accounts:

  Guest
  jason
  Martin
  Shiela
  krbtgt
  Administrator
```

Figure 4-13: Screenshot of snmp-check showing System Information and User Accounts



snmp-check 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
[*] Network information:

  IP forwarding enabled         : no
  Default TTL                   : 128
  TCP segments received         : 101652
  TCP segments sent             : 44867
  TCP segments retrans          : 0
  Input datagrams               : 111093
  Delivered datagrams           : 111159
  Output datagrams              : 57358

[*] Network interfaces:

  Interface                     : [ up ] Software Loopback Interface 1
  Id                            : 1
  Mac Address                   : :::::
  Type                          : softwareLoopback
  Speed                         : 1073 Mbps
  MTU                           : 1500
  In octets                     : 0
  Out octets                    : 0

  Interface                     : [ down ] Microsoft 6to4 Adapter
  Id                            : 2
  Mac Address                   : :::::
  Type                          : unknown
  Speed                         : 0 Mbps
  MTU                           : 0
  In octets                     : 0
```

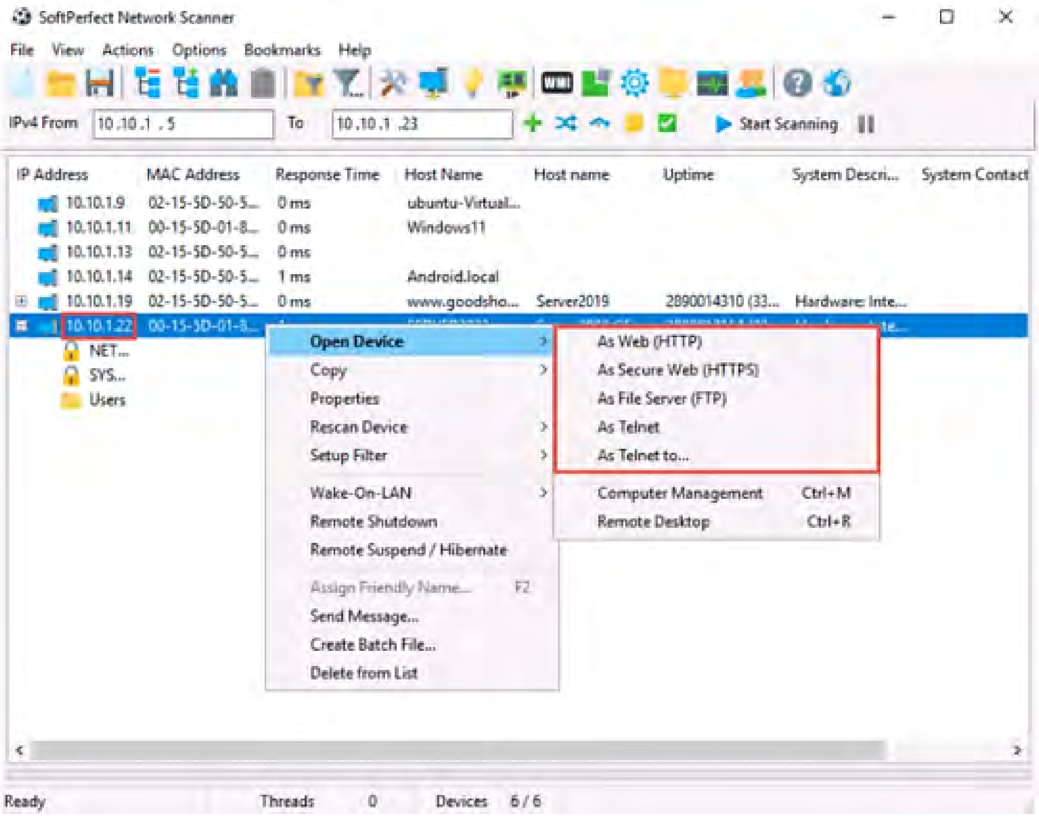
Figure 4-14: Screenshot of snmp-check showing Network Information and Interfaces

SoftPerfect Network Scanner

SoftPerfect Network Scanner allows users to ping computers, check ports, identify shared folders, and gather extensive information about network devices using Windows Management Instrumentation (WMI), SNMP, Hypertext Transfer Protocol (HTTP), SSH, and PowerShell. It is capable of scanning for remote services, registry entries, files, and performance counters; provides versatile filtering and display functionalities; and can export NetScan results in various formats, including Extensible Markup Language (XML) and JavaScript Object Notation (JSON).

Additionally, SoftPerfect Network Scanner can verify the status of a user-specified port and inform whether it is accessible. It also can resolve hostnames and automatically detect both local and external IP ranges. The tool includes support for remote shutdowns and Wake-on-LAN features.

Attackers utilize this software to collect information related to shared folders and network devices.



IPv4 From 10.10.1 . 5 To 10.10.1 .23

IP Address	MAC Address	Response Time	Host Name	Host name	Uptime	System Descri...	System Contact
10.10.1.9	02-15-5D-50-5...	0 ms	ubuntu-Virtual...				
10.10.1.11	00-15-5D-01-8...	0 ms	Windows11				
10.10.1.13	02-15-5D-50-5...	0 ms					
10.10.1.14	02-15-5D-50-5...	1 ms	Android.local				
10.10.1.19	02-15-5D-50-5...	0 ms	www.goodsho...	Server2019	2890014310 (33...	Hardware: Inte...	
10.10.1.22	00-15-5D-01-8...						

Ready Threads 0 Devices 6 / 6

Figure 4-16: Screenshot of SoftPerfect Network Scanner

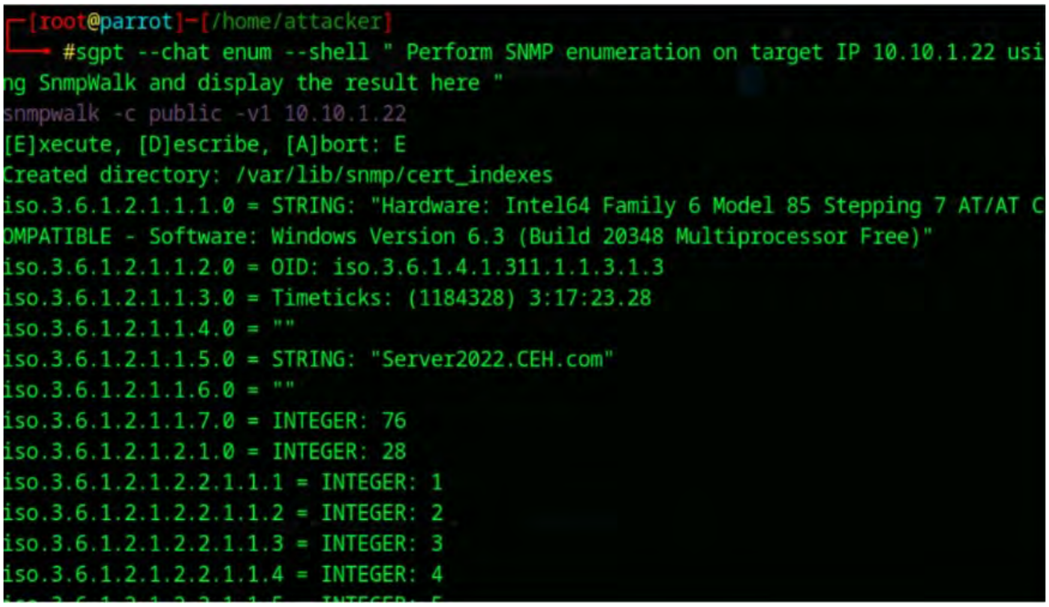
SNMP Enumeration using SnmpWalk and Nmap using AI

Attackers can utilize AI-driven technologies to improve and automate their network enumeration efforts. With AI's assistance, attackers can easily conduct SNMP enumeration and collect SNMP data and processes related to targeted IP addresses.

Attackers can leverage tools like ChatGPT to accomplish this task by using suitable prompts, such as:

Example 1:

```
Perform SNMP enumeration on target IP 10.10.1.22 using SnmpWalk and display the result here.
```



```
┌─[root@parrot]─[/home/attacker]
└──╼ #sgpt --chat enum --shell " Perform SNMP enumeration on target IP 10.10.1.22 usi
ng SnmpWalk and display the result here "
snmpwalk -c public -v1 10.10.1.22
[E]xecute, [D]escribe, [A]bort: E
Created directory: /var/lib/snmp/cert_indexes
iso.3.6.1.2.1.1.1.0 = STRING: "Hardware: Intel64 Family 6 Model 85 Stepping 7 AT/AT C
OMPATIBLE - Software: Windows Version 6.3 (Build 20348 Multiprocessor Free)"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.311.1.1.3.1.3
iso.3.6.1.2.1.1.3.0 = Timeticks: (1184328) 3:17:23.28
iso.3.6.1.2.1.1.4.0 = ""
iso.3.6.1.2.1.1.5.0 = STRING: "Server2022.CEH.com"
iso.3.6.1.2.1.1.6.0 = ""
iso.3.6.1.2.1.1.7.0 = INTEGER: 76
iso.3.6.1.2.1.2.1.0 = INTEGER: 28
iso.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.1.3 = INTEGER: 3
iso.3.6.1.2.1.2.2.1.1.4 = INTEGER: 4
```

Figure 4-17: Perform SNMP Enumeration on Target IP using SnmpWalk

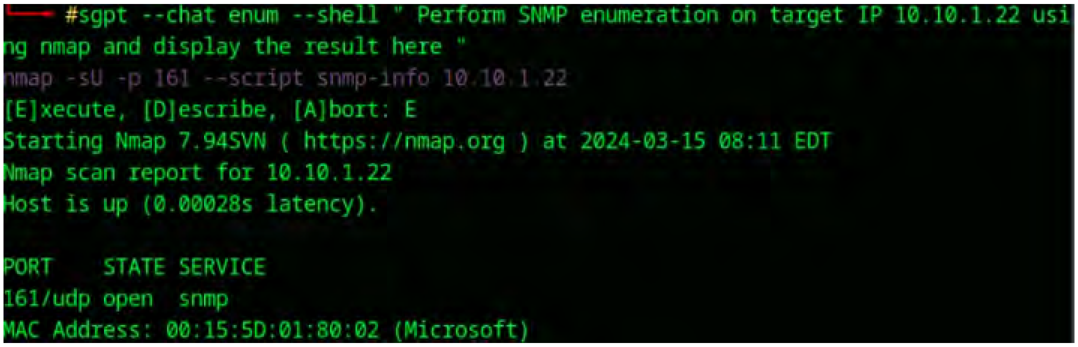
The command outlined below is intended to automate the process of SNMP enumeration on the designated target IP:

```
snmpwalk -c public -v1 10.10.1.22
```

The **snmpwalk** command serves to carry out SNMP enumeration on the specified IP address 10.10.1.22 using SnmpWalk.

Example 2:

```
Perform SNMP enumeration on target IP 10.10.1.22 using nmap and display the result here.
```



```
└──╼ #sgpt --chat enum --shell " Perform SNMP enumeration on target IP 10.10.1.22 usi
ng nmap and display the result here "
nmap -sU -p 161 --script snmp-info 10.10.1.22
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 08:11 EDT
Nmap scan report for 10.10.1.22
Host is up (0.00028s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 00:15:5D:01:80:02 (Microsoft)
```

Figure 4-18: Perform SNMP Enumeration on Target IP using Nmap

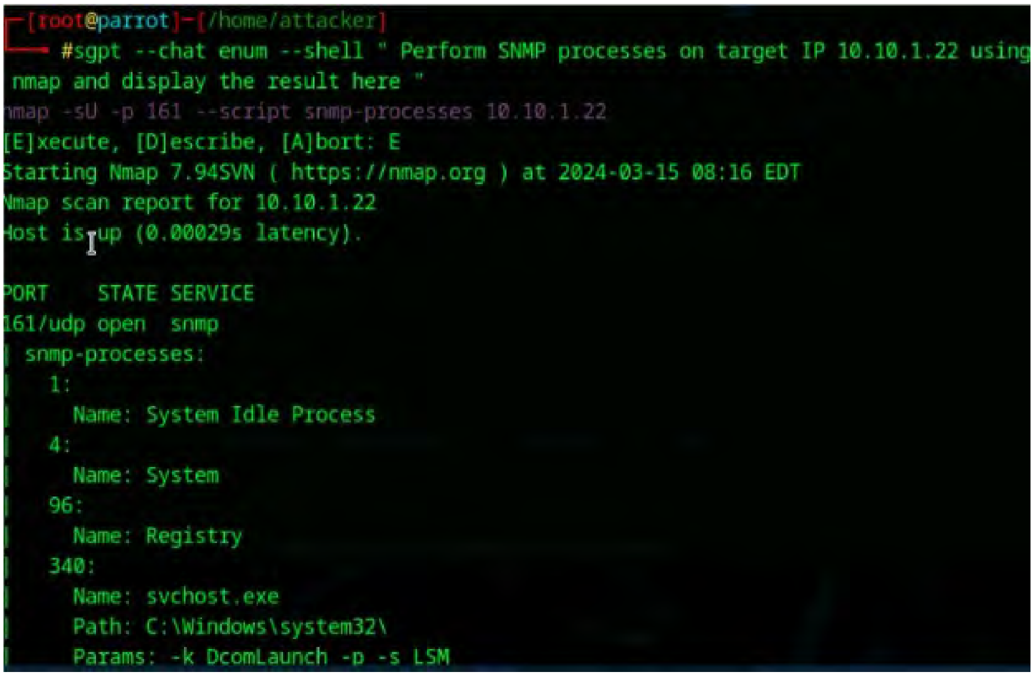
The command below is intended to automate the tasks of enumerating SNMP on the given target IP:

```
nmap -sU -p 161 --script snmp-info 10.10.1.22
```

The nmap command, along with its specified options and the script (**snmp-info**), is utilized to carry out SNMP enumeration on the target IP 10.10.1.22 using nmap.

Example 3:

Perform SNMP processes on target IP 10.10.1.22 using nmap and display the result here.



```
┌─[root@parrot]─[/home/attacker]
└──╼ #sgpt --chat enum --shell " Perform SNMP processes on target IP 10.10.1.22 using
 nmap and display the result here "
nmap -sU -p 161 --script snmp-processes 10.10.1.22
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 08:16 EDT
Nmap scan report for 10.10.1.22
Host is up (0.00029s latency).

PORT    STATE SERVICE
161/udp open  snmp
| snmp-processes:
|   1:
|     Name: System Idle Process
|   4:
|     Name: System
|   96:
|     Name: Registry
|   340:
|     Name: svchost.exe
|     Path: C:\Windows\system32\
|     Params: -k DcomLaunch -p -s LSM
```

Figure 4-19: Perform SNMP Processes on Target IP

The following command is intended to automate tasks for SNMP enumeration on the designated target IP:

```
nmap -sU -p 161 --script snmp-processes 10.10.1.22
```

Another nmap command with particular options and script (**snmp-processes**) is utilized to conduct SNMP processes on the target IP 10.10.1.22 using nmap.

These commands streamline SNMP enumeration tasks and present the outcomes for each command performed on the specified target IP 10.10.1.22.

LDAP Enumeration

Different protocols facilitate communication and manage data transfer between network resources. Each of these protocols transmits critical information regarding network resources alongside the data. An external user who successfully gathers this information by exploiting the protocols can infiltrate the network and may misuse its resources. The Lightweight Directory Access Protocol (LDAP) is one such protocol that retrieves directory listings. This section will examine LDAP enumeration, the types of information obtained through LDAP enumeration, and tools used for LDAP enumeration.

LDAP is an open standard internet protocol. LDAP is used for accessing and maintaining distributed directory information services in a hierarchical and logical structure. A directory service plays an important role by allowing information such as user, system, network, service information, etc., to be shared throughout the network. LDAP provides a central place to store usernames and passwords. Applications and services connect to the LDAP server to validate users. The client initiates an LDAP session by sending an operation request to the Directory System Agent (DSA) using TCP port 389. The communication between the client and server uses Basic Encoding Rules (BER). Directory services using LDAP include:

- Active Directory
- Open Directory
- Oracle iPlanet
- Novell eDirectory
- OpenLDAP

An attacker can anonymously request sensitive data from the LDAP service, including usernames, addresses, departmental information, and server names, which can then be exploited to carry out further attacks.

Manual and Automated LDAP Enumeration

Attackers may employ either manual or automated methods to perform LDAP enumeration. Below are some commands that can be utilized for LDAP enumeration:

Manual LDAP Enumeration

Attackers can execute manual LDAP enumeration using Python. Adhere to the steps described below in order to perform manual LDAP enumeration using Python:

1. Use Nmap to verify if the target LDAP server is operational on port 389 for LDAP and port 636 for secure LDAP.

2. If the target server is active on the indicated ports, initiate the enumeration by installing LDAP with the following command:

```
pip3 install ldap3
```

3. As demonstrated in the following code, create a server object (**server**) and provide the target IP address or hostname along with the port number. If the target server is using secure LDAP, set:

```
use_ssl = True
```

4. Obtain the Directory System Agent (DSA) specific entry (DSE) naming contexts by specifying:

```
get_info = ldap3.ALL
```

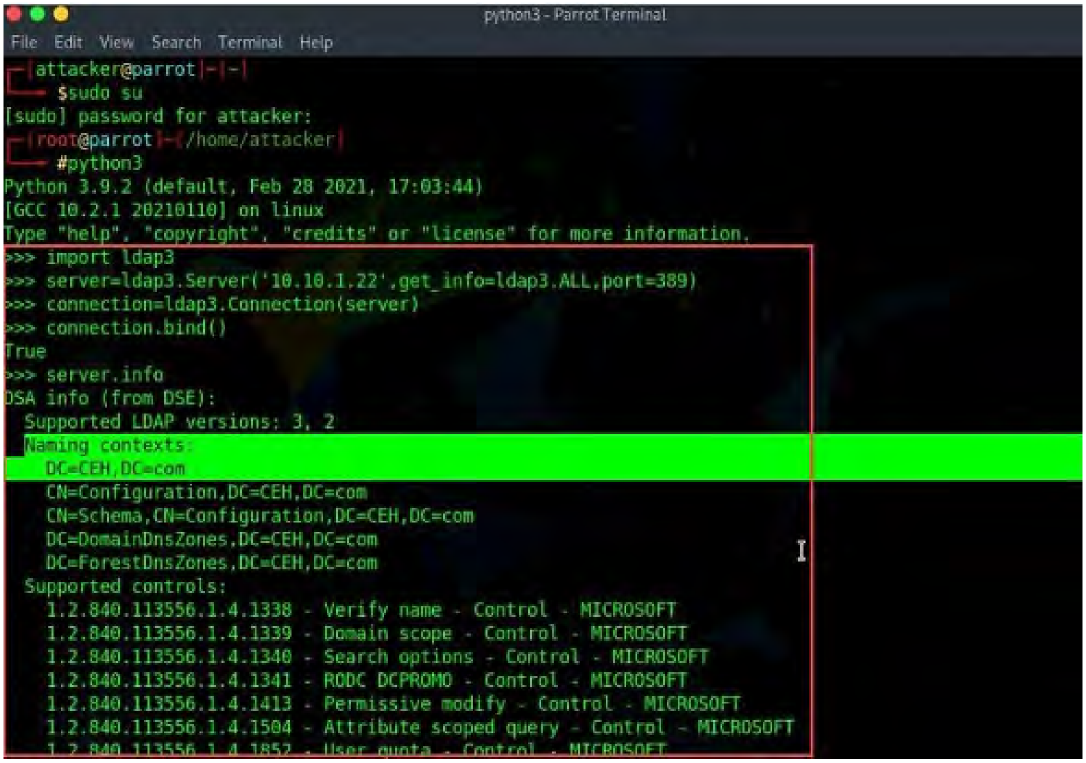
5. Next, create a connection object, named **connection**, and make a call to **bind()**.

6. If the binding is successful, **True** will be displayed on the screen as shown below:

```
>>> import ldap3
>>> server = ldap3.Server('Target IP Address', get_info = ldap3.ALL, port = 389)
>>> connection = ldap3.Connection(server)
>>> connection.bind()
True
```

7. You can now retrieve information such as the domain name and naming context using the script below:

```
>>> server.info
```



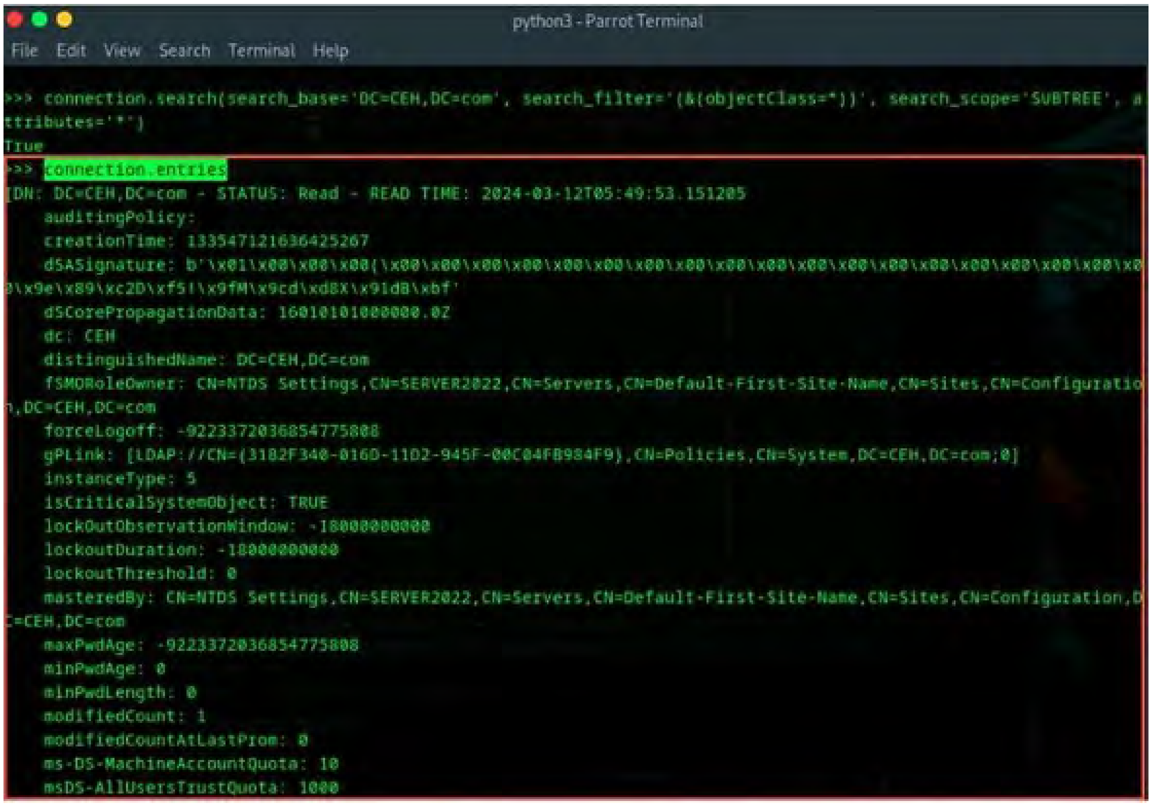
python3 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import ldap3
>>> server=ldap3.Server('10.10.1.22',get_info=ldap3.ALL,port=389)
>>> connection=ldap3.Connection(server)
>>> connection.bind()
True
>>> server.info
DSA info (from DSE):
  Supported LDAP versions: 3, 2
  Naming contexts:
    DC=CEH,DC=com
    CN=Configuration,DC=CEH,DC=com
    CN=Schema,CN=Configuration,DC=CEH,DC=com
    DC=DomainDnsZones,DC=CEH,DC=com
    DC=ForestDnsZones,DC=CEH,DC=com
  Supported controls:
    1.2.840.113556.1.4.1338 - Verify name - Control - MICROSOFT
    1.2.840.113556.1.4.1339 - Domain scope - Control - MICROSOFT
    1.2.840.113556.1.4.1340 - Search options - Control - MICROSOFT
    1.2.840.113556.1.4.1341 - RODC DCPROMO - Control - MICROSOFT
    1.2.840.113556.1.4.1413 - Permissive modify - Control - MICROSOFT
    1.2.840.113556.1.4.1504 - Attribute scoped query - Control - MICROSOFT
    1.2.840.113556.1.4.1852 - User quota - Control - MICROSOFT
```

Figure 4-20: LDAP Enumeration using Python Script

8. Once you have acquired the naming context, use the script provided below to obtain all the directory objects:

```
>>>connection.search(search_base='DC=DOMAIN,DC=DOMAIN',
search_filter='(&(objectClass=*))', search_scope='SUBTREE', attributes='*')
True
>> connection.entries
```


python3 - Parrot Terminal
File Edit View Search Terminal Help

```
>>> connection.search(search_base='DC=CEH,DC=com', search_filter='(&(objectClass=*))', search_scope='SUBTREE', a
ttributes='*')
True
>>> connection.entries
[DN: DC=CEH,DC=com - STATUS: Read - READ TIME: 2024-03-12T05:49:53.151205
    auditingPolicy:
    creationTime: 133547121636425267
    dSASignature: b'\x01\x00\x00\x00(\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
0\x9e\x89\xc2D\xf5!\x9fM\x9cd\xd8X\x91d8\xbf'
    dSCorePropagationData: 16010101000000.0Z
    dc: CEH
    distinguishedName: DC=CEH,DC=com
    fSMORoleOwner: CN=NTDS Settings,CN=SERVER2022,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuratio
n,DC=CEH,DC=com
    forceLogoff: -9223372036854775808
    gPLink: [LDAP://CN={3182F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=CEH,DC=com;0]
    instanceType: 5
    isCriticalSystemObject: TRUE
    lockOutObservationWindow: -18000000000
    lockoutDuration: -18000000000
    lockoutThreshold: 0
    masteredBy: CN=NTDS Settings,CN=SERVER2022,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,D
C=CEH,DC=com
    maxPwdAge: -9223372036854775808
    minPwdAge: 0
    minPwdLength: 0
    modifiedCount: 1
    modifiedCountAtLastProm: 0
    ms-DS-MachineAccountQuota: 10
    msDS-AllUsersTrustQuota: 1000
```

Figure 4-21: Output of LDAP Enumeration

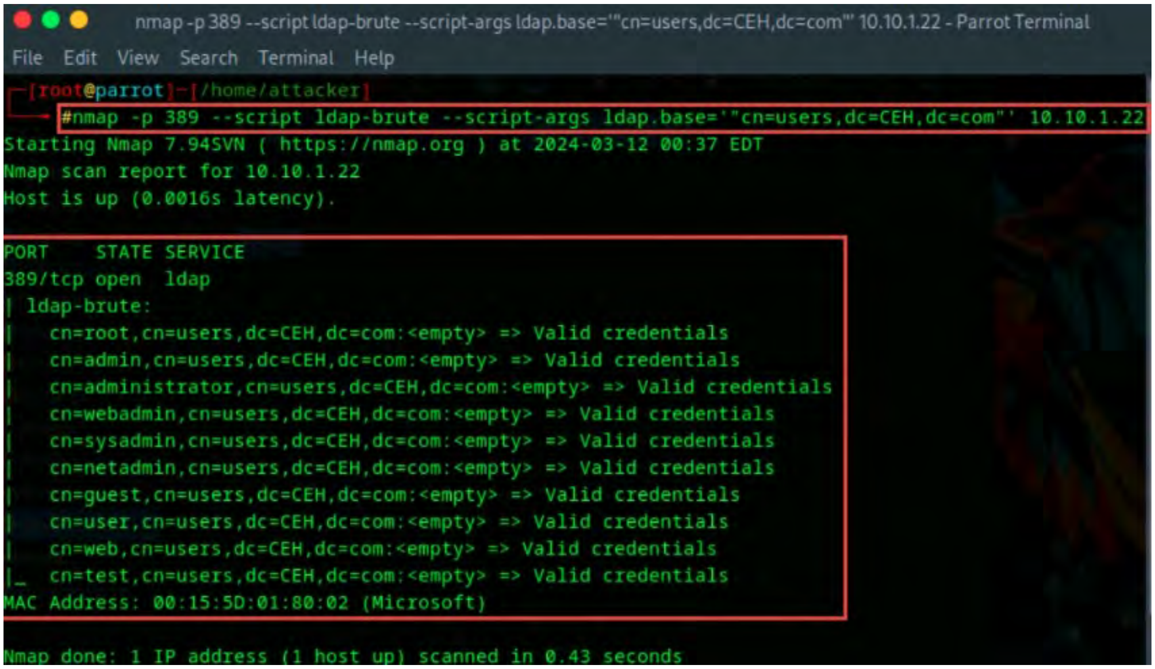
9. Now, use the following script to dump the entire LDAP:

```
>>connection.search(search_base='DC=DOMAIN,DC=DOMAIN',
search_filter='(&(objectClass=person))',search_scope='SUBTREE',      attributes='userPassword')
True
>>> connection.entries
```

Automated LDAP Enumeration

Attackers utilize the **ldap-brute** NSE script for brute-forcing LDAP authentication. By default, it relies on the standard username and password lists.

```
nmap -p 389 –script ldap-brute --script-args ldap.base='"cn=users,dc=CEH,dc=com"' <Target IP Address>
```


nmap -p 389 --script ldap-brute --script-args ldap.base='"cn=users,dc=CEH,dc=com"' 10.10.1.22 - Parrot Terminal
File
Edit
View
Search
Terminal
Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -p 389 --script ldap-brute --script-args ldap.base='"cn=users,dc=CEH,dc=com"' 10.10.1.22
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-12 00:37 EDT
Nmap scan report for 10.10.1.22
Host is up (0.0016s latency).

PORT    STATE SERVICE
389/tcp open  ldap
| ldap-brute:
|   cn=root,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=admin,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=administrator,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=webadmin,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=sysadmin,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=netadmin,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=guest,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=user,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=web,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|_  cn=test,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
MAC Address: 00:15:5D:01:80:02 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
```

Figure 4-22: Screenshot showing Output of the Nmap ldap-brute NSE Script

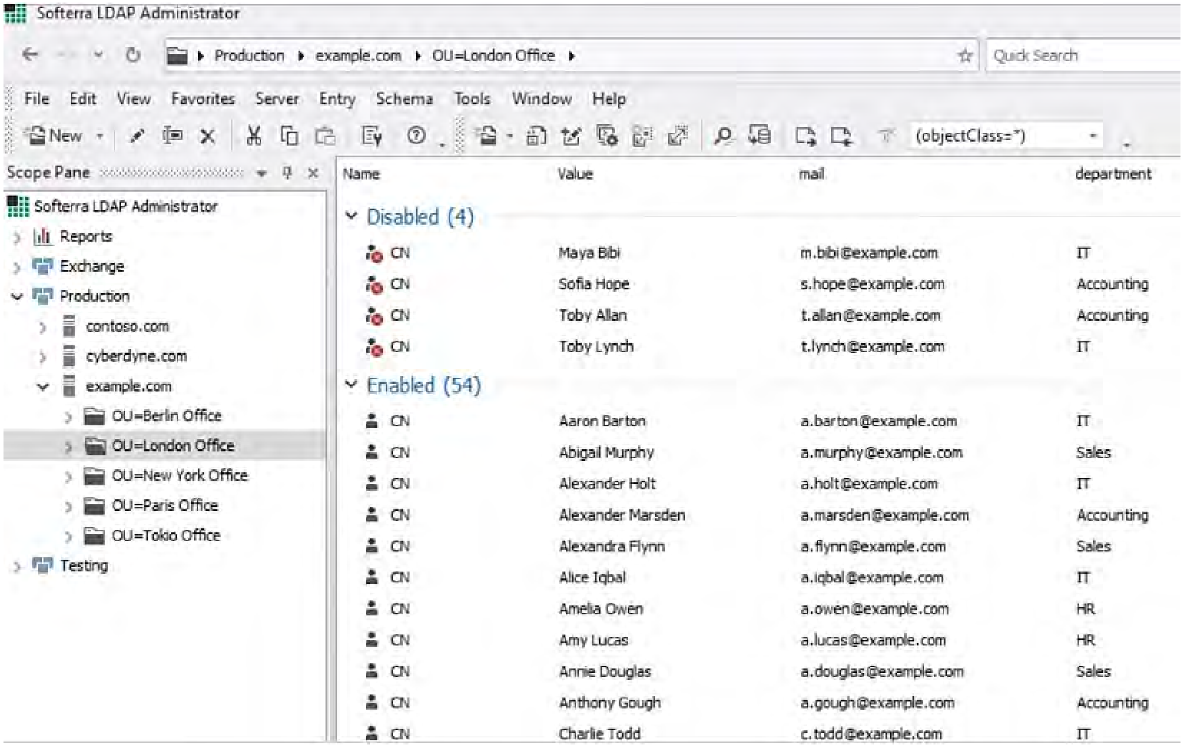
EXAM TIP: Custom lists can be utilized by employing the **userdb** and **passdb** script arguments.

LDAP Enumeration Tools

Numerous LDAP enumeration tools are available that can access directory listings in Active Directory (AD) or other directory services. By employing these tools, attackers can gather information, including valid usernames, addresses, and departmental information from various LDAP servers.

Softerra LDAP Administrator

Softerra LDAP Administrator is a tool used for managing LDAP directories and is compatible with LDAP servers like Active Directory (AD), Novell Directory Services, and Netscape/iPlanet. It allows users to navigate and administer LDAP directories. As illustrated in the screenshot, malicious actors utilize Softerra LDAP Administrator to gather information about users, including their usernames, email addresses, and departments.



Name	Value	mail	department
Disabled (4)			
CN	Maya Bibi	m.bibi@example.com	IT
CN	Sofia Hope	s.hope@example.com	Accounting
CN	Toby Allan	t.allan@example.com	Accounting
CN	Toby Lynch	t.lynch@example.com	IT
Enabled (54)			
CN	Aaron Barton	a.barton@example.com	IT
CN	Abigail Murphy	a.murphy@example.com	Sales
CN	Alexander Holt	a.holt@example.com	IT
CN	Alexander Marsden	a.marsden@example.com	Accounting
CN	Alexandra Flynn	a.flynn@example.com	Sales
CN	Alice Iqbal	a.iqbal@example.com	IT
CN	Amelia Owen	a.owen@example.com	HR
CN	Amy Lucas	a.lucas@example.com	HR
CN	Annie Douglas	a.douglas@example.com	Sales
CN	Anthony Gough	a.gough@example.com	Accounting
CN	Charlie Todd	c.todd@example.com	IT

Figure 4-23: Screenshot of Softerra LDAP Administrator

ldapsearch

ldapsearch provides a command-line interface for the **ldap_search_ext(3)** library function. It establishes a connection to an LDAP server, authenticates, and conducts a search based on the indicated parameters. The filter must adhere to the string format of search filters as outlined in RFC 4515. In the absence of a specified filter, the default filter, **(objectClass**=*), is applied.

When ldapsearch locates one or more entries, it returns the attributes defined by **attrs**. If * is included, all user attributes are retrieved. If + is included, all operational attributes are fetched. When no **attrs** are specified, all user attributes are provided. If only **1.1** is specified, no attributes are returned.

The results of the search are displayed in an extended version of the LDAP Data Interchange Format (LDIF). The option **-L** determines the format of the output.

Attackers utilize ldapsearch to enumerate Active Directory users. This enables them to connect with an LDAP server and execute various searches using specific filters. The following command can be employed to carry out an LDAP search with simple authentication:

```
ldapsearch -h <Target IP Address> -x
```

If the preceding command runs successfully, the next command can be executed to gather more details about the naming contexts:

```
ldapsearch -h <Target IP Address> -x -s base namingcontexts
```

For instance, from the output of this command, if the primary domain component is identified as **DC=htb,DC=local**, the subsequent command can be used to retrieve further information about the primary domain:

```
ldapsearch -h <Target IP Address> -x -b “DC=htb,DC=local”
```

The following commands can be utilized to access information regarding a particular object or all the objects within a directory tree:

```
ldapsearch -h <Target IP Address> -x -b "DC=htb,DC=local" '(objectClass=Employee)'
```

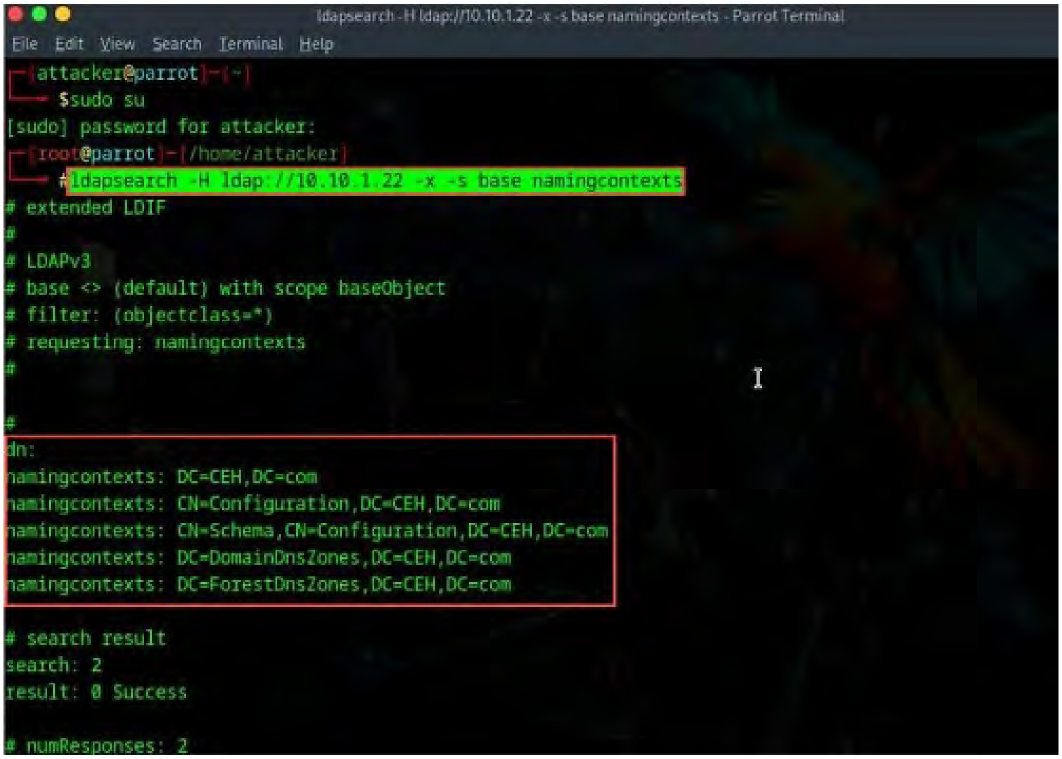
This command retrieves information related to the object class Employee.

```
ldapsearch -x -h <Target IP Address> -b "DC=htb,DC=local" "objectclass=*"
```

This command retrieves information related to all the objects in the directory tree.

The command below fetches a list of users associated with a specific object class:

```
ldapsearch -h <Target IP Address> -x -b "DC=htb,DC=local" '(objectClass= Employee)' sAMAccountName sAMAccountType
```



ldapsearch -H ldap://10.10.1.22 -x -s base namingcontexts - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #ldapsearch -H ldap://10.10.1.22 -x -s base namingcontexts
# extended LDIF
#
# LDAPv3
# base <> (default) with scope baseObject
# filter: (objectclass=*)
# requesting: namingcontexts
#


#
dn:
namingcontexts: DC=CEH,DC=com
namingcontexts: CN=Configuration,DC=CEH,DC=com
namingcontexts: CN=Schema,CN=Configuration,DC=CEH,DC=com
namingcontexts: DC=DomainDnsZones,DC=CEH,DC=com
namingcontexts: DC=ForestDnsZones,DC=CEH,DC=com

# search result
search: 2
result: 0 Success

# numResponses: 2
```

Figure 4-24: Screenshot of ldapsearch

NTP and NFS Enumeration

Network administrators frequently neglect the importance of the Network Time Protocol (NTP) server when assessing security measures. However, if utilized correctly, it can reveal significant network details to an attacker. Therefore, understanding the information an attacker might derive from NTP enumeration is essential. The Network File System (NFS) facilitates the management of access to remote files. NFS enumeration enables attackers to collect data such as a list of clients connected to the NFS server, their IP addresses, and the directories that have been exported. This section outlines NTP enumeration, the types of information obtained through NTP enumeration, different NTP enumeration commands, tools for NTP enumeration, as well as techniques and tools for NFS enumeration.

NTP Enumeration

NTP stands for Network Time Protocol and is used in a network to synchronize the clocks across the hosts and network devices. NTP is an important protocol, as directory services, network devices, and hosts rely on clock settings for login and logging purposes to keep a record of events. NTP helps in correlating events by time system logs are received by Syslog servers. NTP uses UDP port 123, and its whole communication is based on Coordinated Universal Time (UTC).

NTP uses a term known as stratum to describe the distance between the NTP server and the device. It is just like a TTL number that decreases with every hop when a packet passes by. The stratum value, starting from one, increases with every hop. For example, if we see stratum number 10 on a local router, it means that the NTP server is nine hops away. Securing NTP is also an important aspect. The attacker may alter timings to mislead the forensic teams who investigate and correlate the events to find the root cause of the attack.

NTP is intended to synchronize the clocks of computers that are connected to a network. It utilizes UDP port 123 as its main communication method. NTP can maintain time accuracy within an error margin of 10 ms across the public Internet. Additionally, it can reach an accuracy of 200 µs or better in Local Area Networks (LANs) under optimal conditions.

An attacker can gather the following information through queries to an NTP server:

- Information of the host connected to the NTP server
- Client’s IP address, machine’s name, Operating System information
- Network information such as internal IPs or topology maps may be disclosed from NTP packets depending upon the deployment of the NTP server, i.e., if the NTP server is deployed in DMZ

NTP Enumeration Commands

Commands for NTP enumeration, including ntpdate, ntptrace, ntpdc, and ntpq, are utilized to query important information from an NTP server.

ntpdate

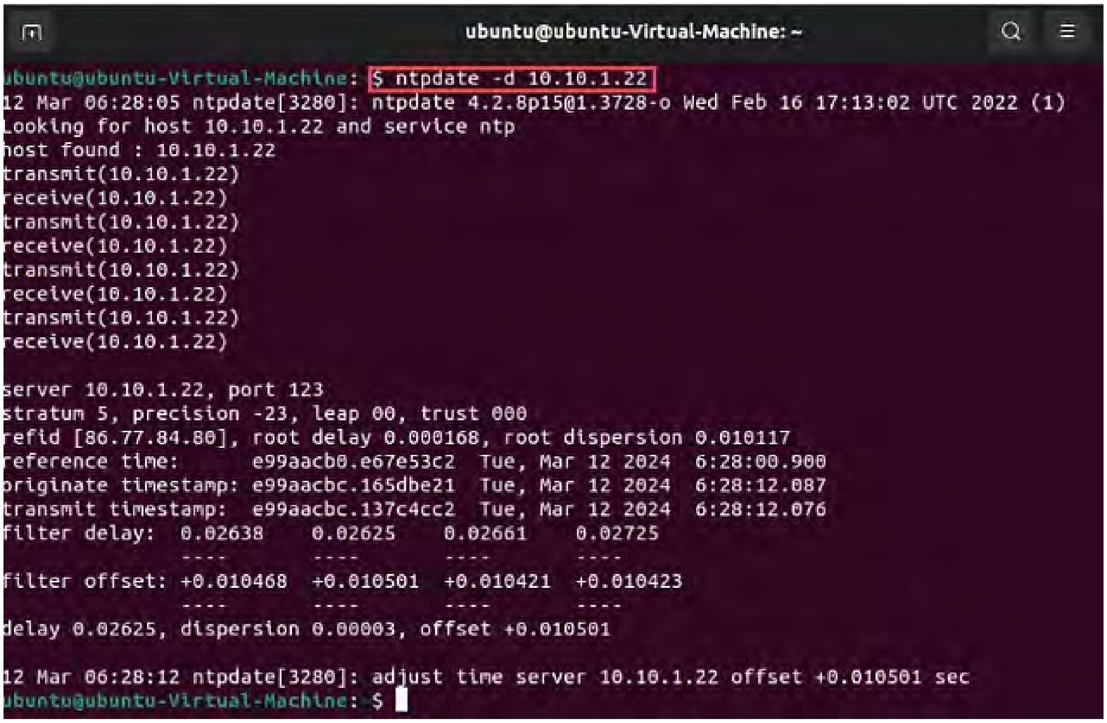
This command gathers the number of time samples from multiple time sources. Its syntax is outlined as follows:

```
ntpdate [-46bBdqsuv] [-a key] [-e authdelay] [-k keyfile] [-o ersion] [-p samples] [-t timeout] [ -
U user_name] server [...]
```

The image outlines various parameters and their corresponding functions for an NTP-related command. Below is a textual representation of the table:

Parameter	Function
-4	Force DNS resolution of given hostnames to the IPv4 namespace.
-6	Force DNS resolution of given hostnames to the IPv6 namespace.
-a key	Enable the authentication function and specify the key identifier for authentication.
-B	Force the time to always be slewed.
-b	Force the time to be stepped.
-d	Enable debugging mode.
-e authdelay	Specify the processing delay to perform an authentication function.
-k keyfile	Specify the path for the authentication key file; the default is /etc/ntp/keys.
-o version	Specify the NTP version for outgoing packets as an integer (1, 2); the default is 4.
-p samples	Specify the number of samples to be acquired from each server, ranging from 1–8; the default is 4.
-q	Query only; do not set the clock.
-s	Divert logging output from standard output (default) to the system syslog facility.
-t timeout	Specify the maximum wait time for a server response; the default is 1 second.
-u	Use an unprivileged port for outgoing packets.
-v	Be verbose; logs ntpdate's version identification string.

Table 4-05: ntpdate Parameters and their Respective Functions



```
ubuntu@ubuntu-Virtual-Machine: $ ntpdate -d 10.10.1.22
12 Mar 06:28:05 ntpdate[3280]: ntpdate 4.2.8p15@1.3728-o Wed Feb 16 17:13:02 UTC 2022 (1)
Looking for host 10.10.1.22 and service ntp
host found : 10.10.1.22
transmit(10.10.1.22)
receive(10.10.1.22)
transmit(10.10.1.22)
receive(10.10.1.22)
transmit(10.10.1.22)
receive(10.10.1.22)
transmit(10.10.1.22)
receive(10.10.1.22)

server 10.10.1.22, port 123
stratum 5, precision -23, leap 00, trust 000
refid [86.77.84.80], root delay 0.000168, root dispersion 0.010117
reference time:      e99aacb0.e67e53c2  Tue, Mar 12 2024  6:28:00.900
originate timestamp: e99aacbc.165dbe21  Tue, Mar 12 2024  6:28:12.087
transmit timestamp:  e99aacbc.137c4cc2  Tue, Mar 12 2024  6:28:12.076
filter delay:  0.02638    0.02625    0.02661    0.02725
               ----       ----       ----       ----
filter offset: +0.010468  +0.010501  +0.010421  +0.010423
               ----       ----       ----       ----
delay 0.02625, dispersion 0.00003, offset +0.010501

12 Mar 06:28:12 ntpdate[3280]: adjust time server 10.10.1.22 offset +0.010501 sec
ubuntu@ubuntu-Virtual-Machine: $
```

Figure 4-25: Screenshot of the ntpdate Command, showing Debugging Information for a Given IP

ntptrace

ntptrace is a Perl script that uses ntpq to follow the chain of NTP servers from a given host back to the primary time source. ntptrace requires implementing the NTP Control and Monitoring Protocol specified in RFC 1305, and NTP Mode 6 packets are enabled to work properly.

This command identifies the source from which the NTP server receives its time and outlines the sequence of NTP servers leading back to its main time source. Malicious actors utilize this command to investigate the series of NTP servers linked to the network. The format is as follows:

```
ntptrace [-n] [-m maxhosts] [servername/IP_address]
```

-n	Do not print host names and show only IP addresses; may be useful if a name server is down
-m maxhosts	Set the maximum number of levels up the chain to be followed

Table 4-06: ntptrace Parameters and their Respective Functions

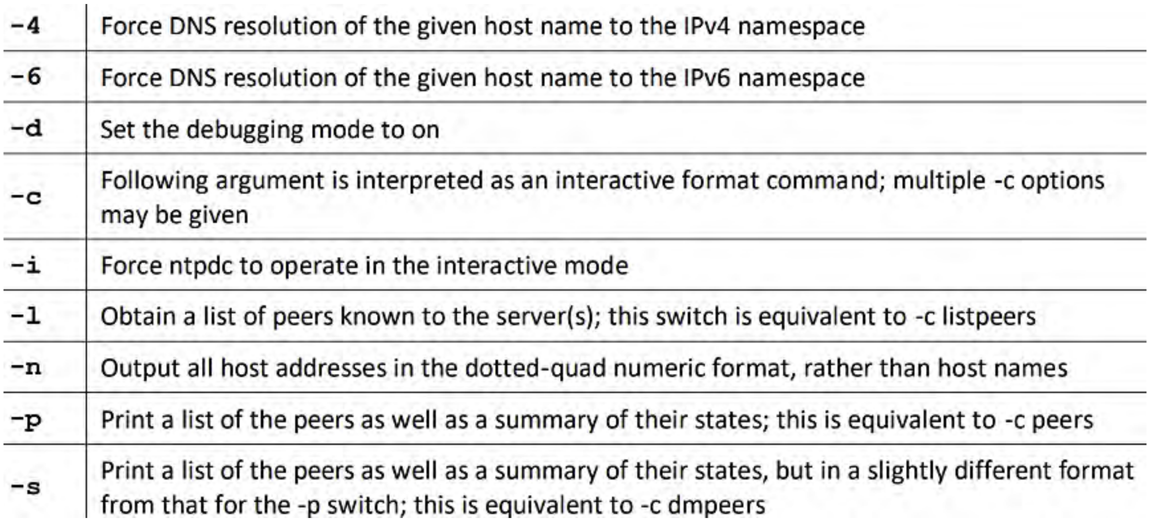
Example:

```
# ntptrace
localhost: stratum 4, offset 0.0019529, synch distance 0.143235
10.10.0.1: stratum 2, offset 0.01142 73, synch distance 0.115554
10.10.1.1: stratum 1, offset 0.0017698, synch distance 0.011193
```

ntpdc

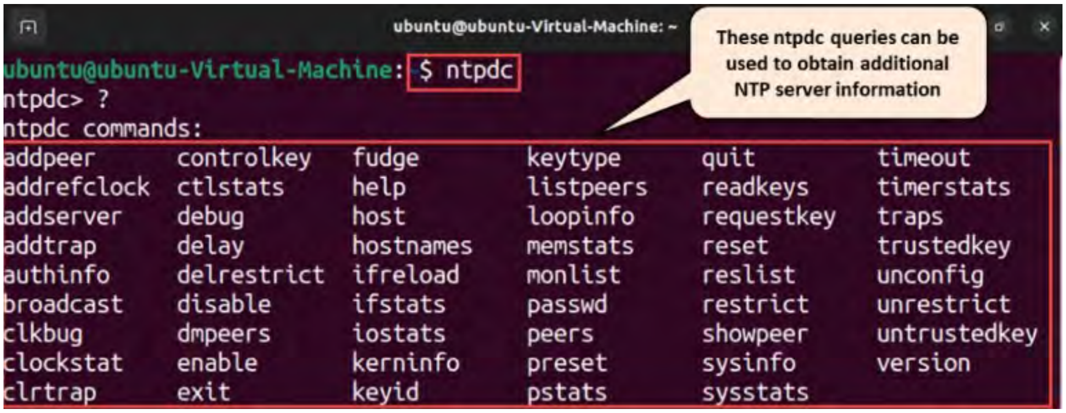
ntpdc is used for questioning the ntpd daemon regarding the current state and requested changes in state. Attackers utilize this command to obtain the status and statistics of every NTP server linked to the targeted network. The syntax is outlined below:

```
ntpdc [ -46dilnps ] [ -c command] [hostname/IP_address]
```



-4	Force DNS resolution of the given host name to the IPv4 namespace
-6	Force DNS resolution of the given host name to the IPv6 namespace
-d	Set the debugging mode to on
-c	Following argument is interpreted as an interactive format command; multiple -c options may be given
-i	Force ntpdc to operate in the interactive mode
-l	Obtain a list of peers known to the server(s); this switch is equivalent to -c listpeers
-n	Output all host addresses in the dotted-quad numeric format, rather than host names
-p	Print a list of the peers as well as a summary of their states; this is equivalent to -c peers
-s	Print a list of the peers as well as a summary of their states, but in a slightly different format from that for the -p switch; this is equivalent to -c dmpeers

Table 4-07: ntpdc Parameters and their Respective Functions



```
ubuntu@ubuntu-Virtual-Machine:~$ ntpdc
ntpdc> ?
ntpdc commands:
addpeer      controlkey   fudge        keytype      quit         timeout
addrefclock  ctlstats     help         listpeers    readkeys     timerstats
addserver    debug        host         loopinfo     requestkey   traps
addtrap      delay        hostnames    memstats     reset        trustedkey
authinfo     delrestrict  ifreload     monlist      reslist      unconfig
broadcast    disable      ifstats      passwd       restrict     unrestrict
clkbug       dmpeers      iostats      peers        showpeer     untrustedkey
clockstat    enable       kerninfo     preset       sysinfo      version
clrtrap      exit         keyid        pstats       sysstats
```

These ntpdc queries can be used to obtain additional NTP server information

Figure 4-26: Screenshot of the ntpdc Command

ntpq

ntpq is a command-line utility that is used for inquiring about the NTP server. The ntpq is used to monitor NTP daemon ntpd operations and determine performance. It uses the standard NTP mode 6 control message formats.

Its syntax is as follows:

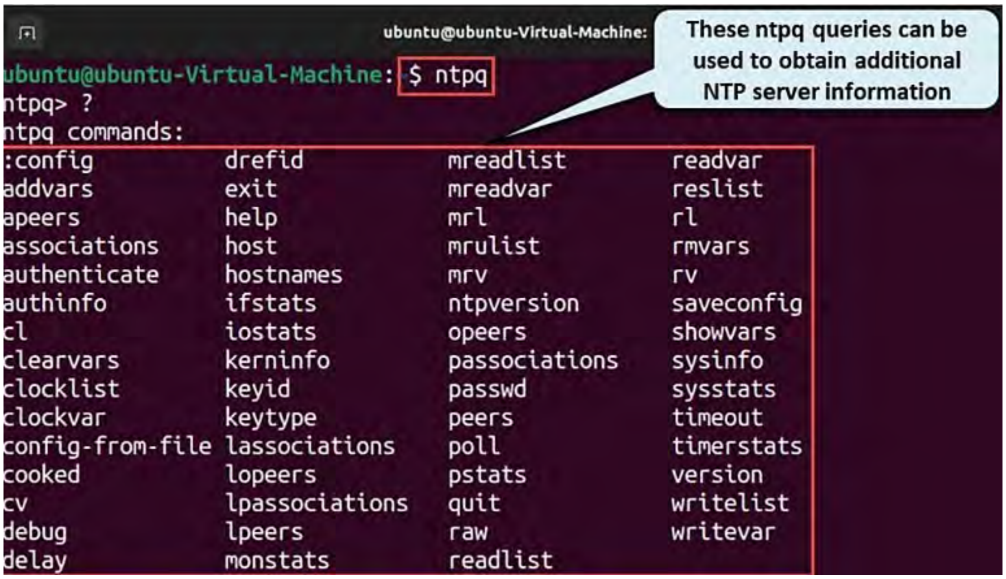
```
ntpq [-46dinp] [-c command] [host/IP_address]
```

-4	Force DNS resolution of the given host name to the IPv4 namespace
-6	Force DNS resolution of the given host name to the IPv6 namespace
-c	Following argument is an interactive format command; multiple -c options may be given
-d	Debugging mode
-i	Force ntpq to operate in the interactive mode
-n	Output all host addresses in the dotted-quad numeric format, rather than host names
-p	Print a list of the peers as well as a summary of their states

Table 4-08: ntpq Parameters and their Respective Functions

Example:

```
ntpq> version
ntpq 4.2.8p15@1.3728-o
ntpq> host
current host is localhost
```



```
ubuntu@ubuntu-Virtual-Machine: $ ntpq
ntpq> ?
ntpq commands:
:config          drefid           mreadlist        readvar
addvars          exit             mreadvar         reslist
apeers           help             mrl              rl
associations     host             mrulist          rmvars
authenticate     hostnames        mrv              rv
authinfo         ifstats          ntpversion       saveconfig
cl               iostats          opeers           showvars
clearvars        kerninfo         passociations    sysinfo
clocklist        keyid            passwd           sysstats
clockvar         keytype          peers            timeout
config-from-file lassociations    poll             timerstats
cooked           lopeers          pstats           version
cv               lpassociations   quit             writelist
debug            lpeers           raw              writevar
delay            monstats         readlist
```

These ntpq queries can be used to obtain additional NTP server information

Figure 4-27: Screenshot of the ntpq Command

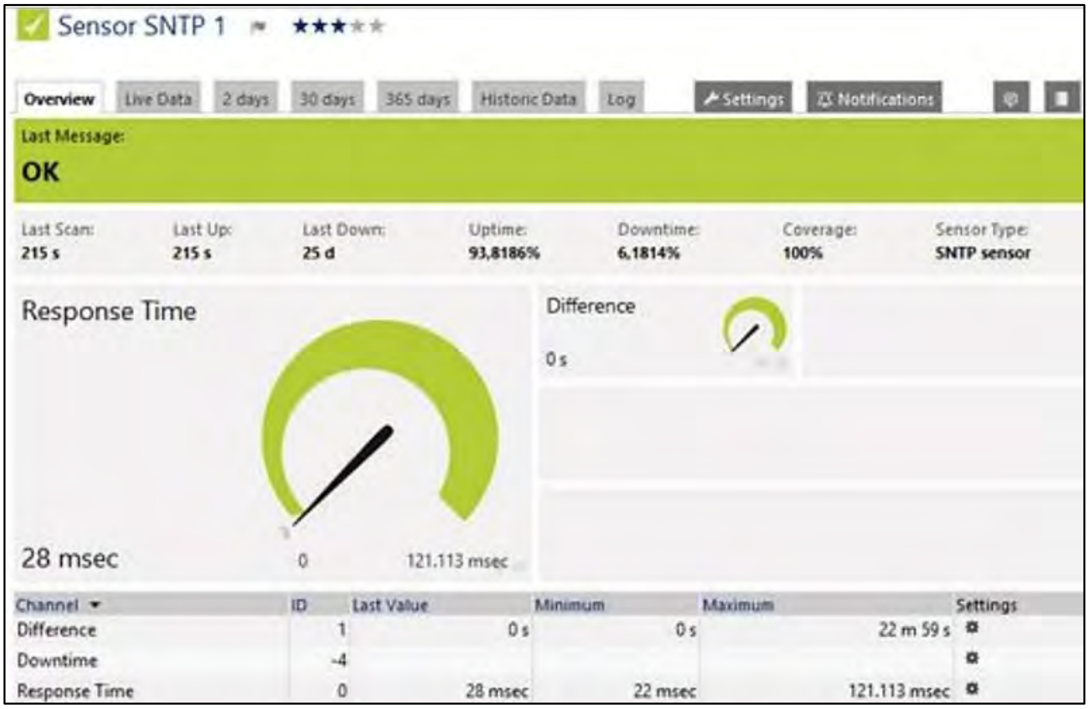
EXAM TIP: In several Linux distributions, the NTP daemon ntpd has been integrated with Chrony, known as chronyd. Both daemons ensure that the local system's time is synchronized with a remote time server.

NTP Enumeration Tools

NTP enumeration tools are utilized to oversee the operation of NTP and SNTP servers within the network and assist in setting up and confirming the connectivity between the time client and the NTP servers.

PRTG Network Monitor

PRTG monitors every component, device, traffic, and application within the IT infrastructure by employing different technologies like SNMP, WMI, and SSH. Attackers exploit the PRTG Network Monitor to gather information about SNTP server specifics, including the server's response time, the active sensors associated with the server, and the synchronization time.



Last Message:
OK

Last Scan:	Last Up:	Last Down:	Uptime:	Downtime:	Coverage:	Sensor Type:
215 s	215 s	25 d	93,8186%	6,1814%	100%	SNTP sensor

Response Time
28 msec

Difference
0 s

Channel ▾	ID	Last Value	Minimum	Maximum	Settings
Difference	1	0 s	0 s	22 m 59 s	⚙
Downtime	-4				⚙
Response Time	0	28 msec	22 msec	121.113 msec	⚙

Figure 4-28: Screenshot of PRTG Network Monitor

NFS Enumeration

Network File System (NFS) is a kind of file system that allows users to access, view, store, and modify files on a remote server. Clients can access this remote data in the same manner as they do on their local systems. Based on the privileges granted to the clients, they can be restricted to read-only access or allowed to read and write data.

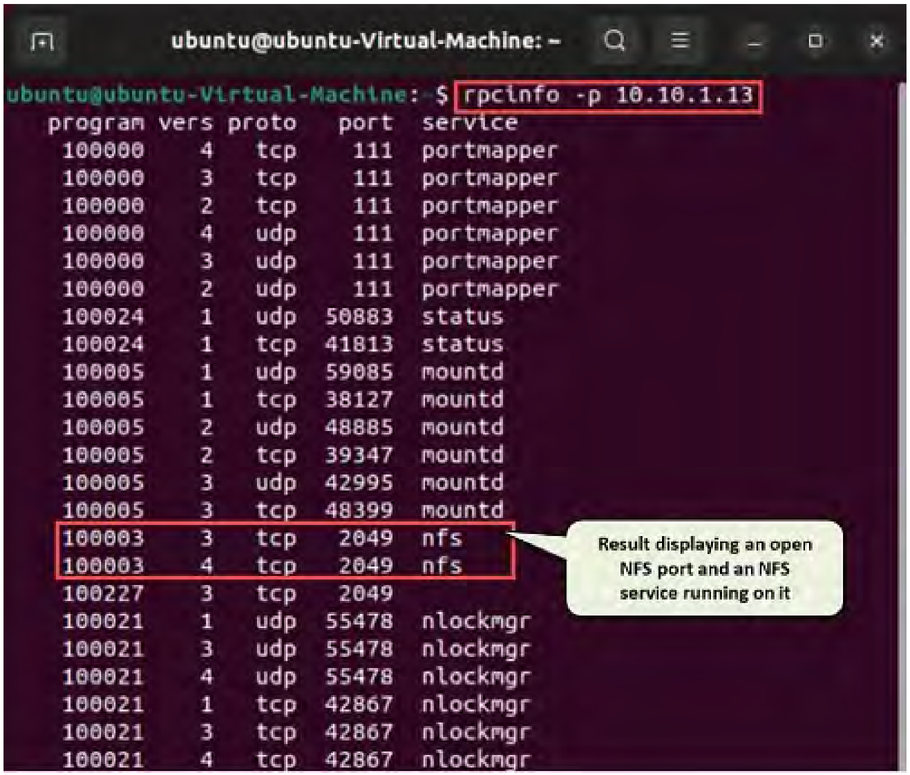
An NFS system is typically set up in a computer network where centralization of data is necessary for essential resources. A Remote Procedure Call (RPC) facilitates the routing and processing of requests between clients and servers. To share files and directories across the network, the process of "exporting" is employed. Initially, the client tries to enable file sharing via the "mounting" process. The /etc/exports directory on the NFS server maintains a record of clients permitted to share files on the server. In this setup, the only credential required for server access is the client's IP address. NFS versions earlier than version 4 operate under the same security specifications.

NFS allows hosts running different operating systems, such as Windows, Linux, or Unix, to mount file systems over a network. Mounting a file system helps in accessing those mounted files as they are mounted locally. This enables system administrators to consolidate resources onto centralized servers on the network. In Windows Server, the NFS protocol includes NFS Server and Client features.

Enumerating NFS services allows attackers to discover the exported directories, see which clients are connected to the NFS server along with their IP addresses, and access the shared data linked to those IP addresses. Once this information is collected, attackers can impersonate the IP addresses to gain complete access to the files shared on the server.

As illustrated in Figure 4-29, an attacker executes the rpcinfo command to probe the target IP address for an open NFS port (port 2049) and the NFS services operating on it.

```
rpcinfo -p <Target IP Address>
```



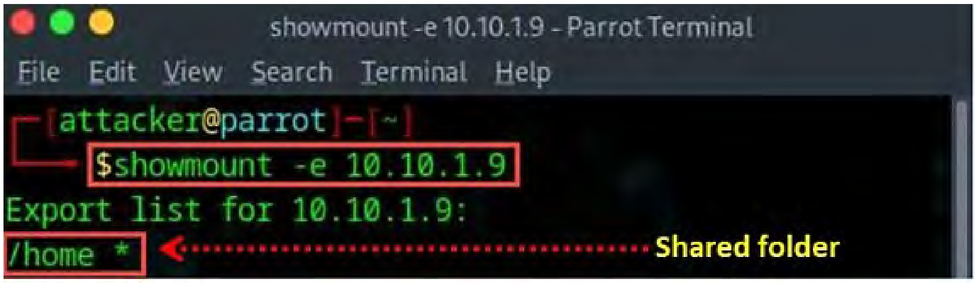
```
ubuntu@ubuntu-Virtual-Machine:~$ rpcinfo -p 10.10.1.13
   program vers proto   port  service
    100000    4   tcp    111  portmapper
    100000    3   tcp    111  portmapper
    100000    2   tcp    111  portmapper
    100000    4   udp    111  portmapper
    100000    3   udp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  50883  status
    100024    1   tcp  41813  status
    100005    1   udp  59085  mountd
    100005    1   tcp  38127  mountd
    100005    2   udp  48885  mountd
    100005    2   tcp  39347  mountd
    100005    3   udp  42995  mountd
    100005    3   tcp  48399  mountd
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100227    3   tcp   2049
    100021    1   udp  55478  nlockmgr
    100021    3   udp  55478  nlockmgr
    100021    4   udp  55478  nlockmgr
    100021    1   tcp  42867  nlockmgr
    100021    3   tcp  42867  nlockmgr
    100021    4   tcp  42867  nlockmgr
```

Result displaying an open NFS port and an NFS service running on it

Figure 4-29: Screenshot of rpcinfo Command Displaying Open NFS Port and Services

As displayed in Figure 4-30, an attacker executes the subsequent command to check the list of shared files and folders:

```
showmount -e <Target IP Address>
```



```
[attacker@parrot]-[~]
$showmount -e 10.10.1.9
Export list for 10.10.1.9:
/home *
```

showmount -e 10.10.1.9 - Parrot Terminal
File Edit View Search Terminal Help
Shared folder

Figure 4-30: Screenshot of the showmount Command displaying a Shared Directory

Additionally, an attacker can employ a range of commands and utilities to infiltrate the NFS server and transfer harmful files onto the server to initiate additional attacks.

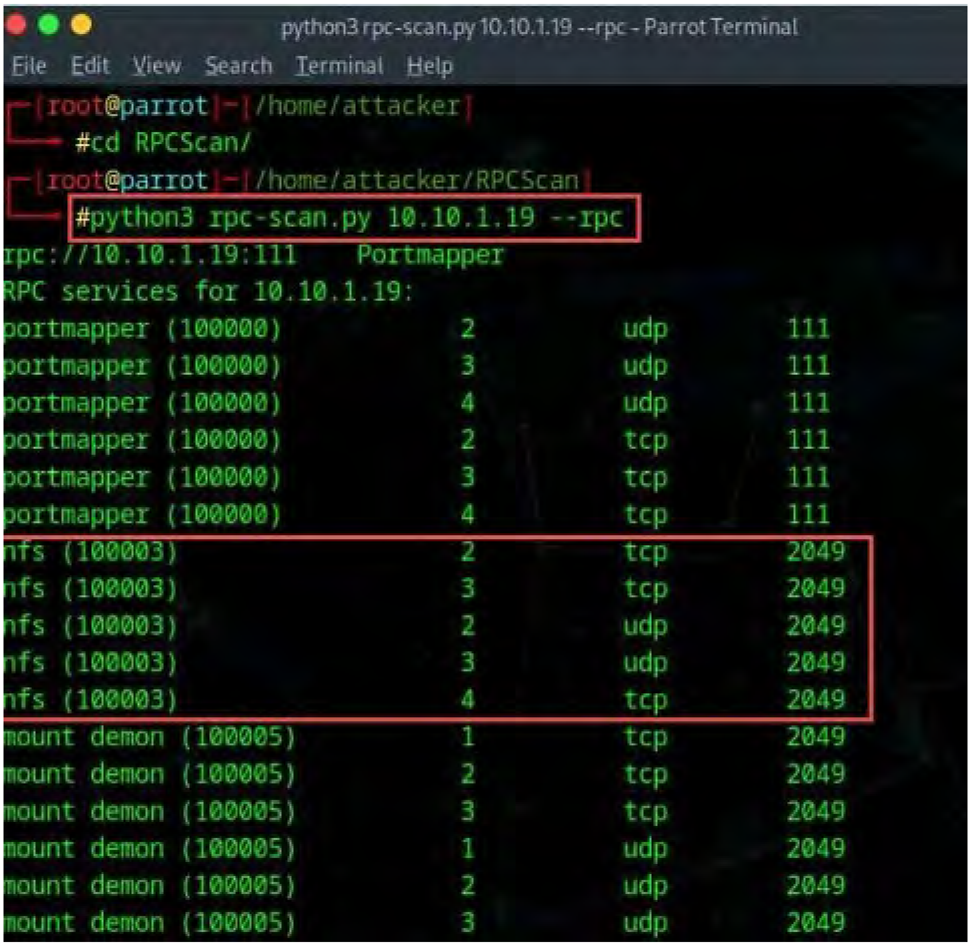
NFS Enumeration Tools

NFS enumeration tools probe a network within a specified range of IP addresses or a single IP address to detect the NFS services operating on it. These tools also help in acquiring a list of RPC services via portmap, a catalog of NFS shares, and a list of directories that can be accessed through NFS; additionally, they enable the retrieval of files shared via the NFS server. Attackers utilize tools like RPCScan and SuperEnum to conduct NFS enumeration.

RPCScan

RPCScan interacts with RPC services and verifies the configuration settings of NFS shares. As illustrated in Figure 4-31, an attacker executes the following command to identify active NFS services on a specific target IP address:

```
python3 rpc-scan.py <Target IP Address> --rp
```



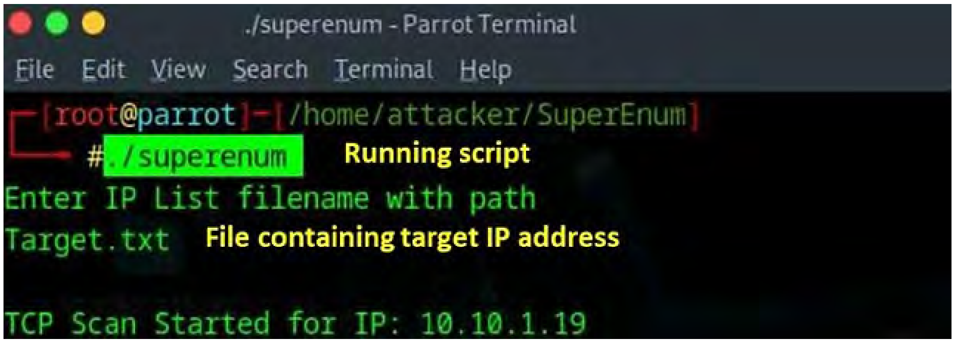
python3 rpc-scan.py 10.10.1.19 --rpc - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #cd RPCScan/
┌─[root@parrot]─[/home/attacker/RPCScan]
└──╼ #python3 rpc-scan.py 10.10.1.19 --rpc
rpc://10.10.1.19:111    Portmapper
RPC services for 10.10.1.19:
portmapper (100000)          2        udp        111
portmapper (100000)          3        udp        111
portmapper (100000)          4        udp        111
portmapper (100000)          2        tcp        111
portmapper (100000)          3        tcp        111
portmapper (100000)          4        tcp        111
nfs (100003)                 2        tcp        2049
nfs (100003)                 3        tcp        2049
nfs (100003)                 2        udp        2049
nfs (100003)                 3        udp        2049
nfs (100003)                 4        tcp        2049
mount demon (100005)         1        tcp        2049
mount demon (100005)         2        tcp        2049
mount demon (100005)         3        tcp        2049
mount demon (100005)         1        udp        2049
mount demon (100005)         2        udp        2049
mount demon (100005)         3        udp        2049
```

Figure 4-31: Screenshot of RPCScan displaying Open NFS Ports and Services

SuperEnum

SuperEnum comes with a script designed to carry out fundamental enumeration on any open port. In the example shown in Figure 4-32, a malicious user executes the **./superenum** script and inputs a text file named Target.txt, which contains either a specific IP address or a list of IP addresses for the enumeration process.

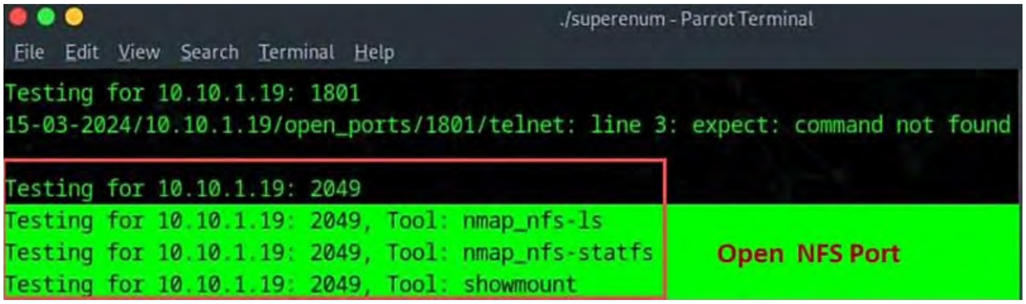


./superenum - Parrot Terminal
File Edit View Search Terminal Help
Running script
File containing target IP address

```
┌─[root@parrot]─[/home/attacker/SuperEnum]
└──╼ #./superenum
Enter IP List filename with path
Target.txt

TCP Scan Started for IP: 10.10.1.19
```

Figure 4-32: Screenshot of SuperEnum Running a Script

Upon scanning the target IP address, the script reveals all active ports, as illustrated in Figure 4-33 below. The NFS service is operating on port 2049.



./superenum - Parrot Terminal
File Edit View Search Terminal Help
Open NFS Port

```
Testing for 10.10.1.19: 1801
15-03-2024/10.10.1.19/open_ports/1801/telnet: line 3: expect: command not found

Testing for 10.10.1.19: 2049
Testing for 10.10.1.19: 2049, Tool: nmap_nfs-ls
Testing for 10.10.1.19: 2049, Tool: nmap_nfs-statfs
Testing for 10.10.1.19: 2049, Tool: showmount
```

Figure 4-33: Screenshot of SuperEnum displaying Open NFS Ports

SMTP and DNS Enumeration

This section outlines various enumeration methods to gather information about network resources. It also addresses techniques for DNS enumeration that provide insights into the DNS servers and the network infrastructure of the target organization. The section elaborates on both SMTP and DNS enumeration methodologies, detailing SMTP enumeration, which involves acquiring a list of legitimate users on an SMTP server, along with tools for SMTP enumeration, DNS zone transfer enumeration, DNS cache snooping, and DNS zone walking.

SMTP Enumeration

SMTP Enumeration is another way to extract information about the target by using a Simple Mail Transfer Protocol (SMTP). SMTP Protocol ensures the mail communication between email servers and recipients over internet port 25. SMTP is one of the most popular TCP/IP protocols widely used by most email servers, now defined in RFC 821.

Email systems typically utilize SMTP along with POP3 and IMAP, allowing users to store messages in a server mailbox and retrieve them from the server when needed. SMTP relies on mail exchange (MX) servers to route email through DNS. It operates over TCP ports 25, 2525, or 587.

SMTP provides the following three built-in commands:

VRFY

Validates users.

```
$ telnet 192.168.168.1 25
Trying 192.168.168.1...
Connected to 192.168.168.1.
Escape character is '^]'.
220 NYmailserver ESMTP Sendmail 8.9.3
HELO
501 HELO requires domain address
HELO x
250 NYmailserver Hello [10.0.0.86], pleased to meet you
VRFY Jonathan
250 Super-User <Jonathan@NYmailserver>
VRFY Smith
```

```
550 Smith... User unknown
```

EXPN

Shows the real delivery addresses associated with aliases and mailing lists.

```
$ telnet 192.168.168.1 25
Trying 192.168.168.1...
Connected to 192.168.168.1.
Escape character is '^]'.
220 NYmailserver ESMTP Sendmail 8.9.3
HELO
501 HELO requires domain address
HELO x
250 NYmailserver Hello [10.0.0.86], pleased to meet you
EXPN Jonathan
250 Super-User <Jonathan@NYmailserver>
EXPN Smith
550 Smith... User unknown
```

RCPT TO

Defines the recipients of the message.

```
$ telnetl 192.168.168.1 25
Trying 192.168.168.1 ...
Connected to 192.168.168.1.
Escape character is '^]'.
220 NYmailserver ESMTP Sendmail 8.9.3
HELO
HELO x
250 NYmailserver Hello [10.0.0.86], pleased to meet you
MAIL FROM:Jonathan
250 Jonathan... Sender ok
RCPT TO:Ryder
250 Ryder... Recipient ok
RCPT TO: Smith
550 Smith... User unknown
501 HELO requires domain address
```

SMTP servers exhibit varying behaviors when responding to VRFY, EXPN, and RCPT TO commands for both valid and invalid users, thus making it possible to identify legitimate users on the SMTP server. Attackers can engage with SMTP through the Telnet command line and compile a roster of valid users on the SMTP server.

Administrators and penetration testers can execute SMTP enumeration with command-line tools such as Telnet and netcat or by utilizing software like Metasploit, Nmap, NetScanTools Pro, and smtp-user-enum to gather information on valid users, delivery addresses, message recipients, and more.

SMTP Enumeration using Nmap

Attackers utilize Nmap to gather details from the intended SMTP server. They extract information from the target SMTP server by employing different SMTP commands found in the Nmap Scripting Engine (NSE) scripts.

The command listed below, when run, displays all the SMTP commands found in the Nmap directory:

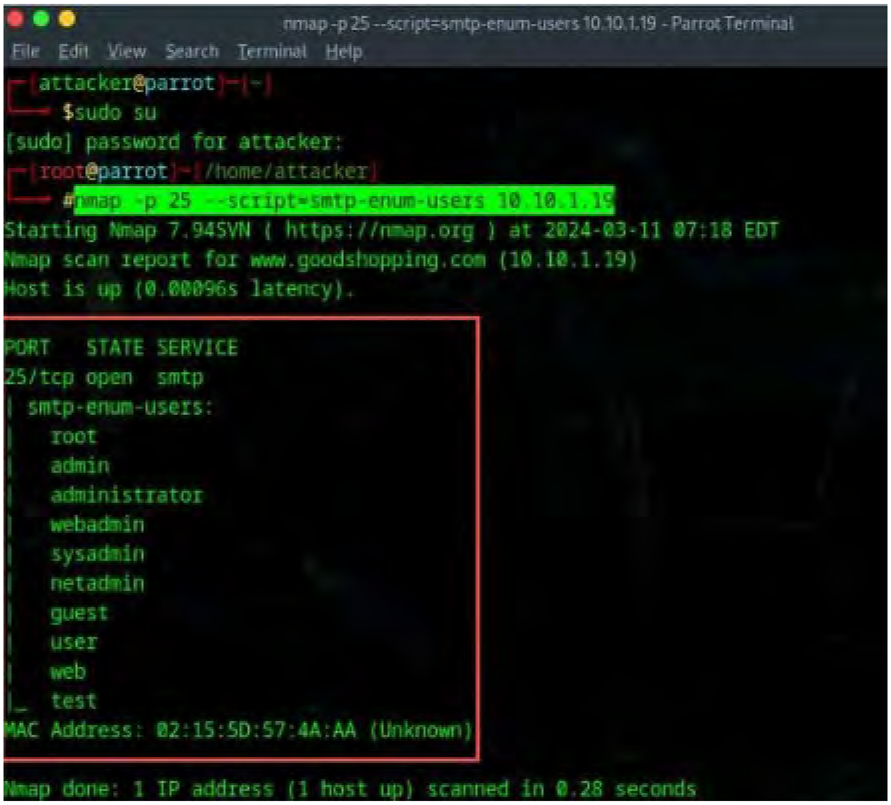
```
nmap -p 25, 365, 587 -script=smtp-commands <Target IP Address>
```

Execute the following command to detect open SMTP relays:

```
nmap -p 25 -script=smtp-open-relay <Target IP Address>
```

Use this command to list all mail users on the SMTP server:

```
nmap -p 25 --script=smtp-enum-users <Target IP Address>
```


nmap -p 25 --script=smtp-enum-users 10.10.1.19 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -p 25 --script=smtp-enum-users 10.10.1.19
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-11 07:18 EDT
Nmap scan report for www.goodshopping.com (10.10.1.19)
Host is up (0.00096s latency).

PORT   STATE SERVICE
25/tcp open  smtp
| smtp-enum-users:
|   root
|   admin
|   administrator
|   webadmin
|   sysadmin
|   netadmin
|   guest
|   user
|   web
|_  test
MAC Address: 02:15:5D:57:4A:AA (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds
```

Figure 4-34: Screenshot showing Output of the smtp-enum-users NSE Script

SMTP Enumeration using Metasploit

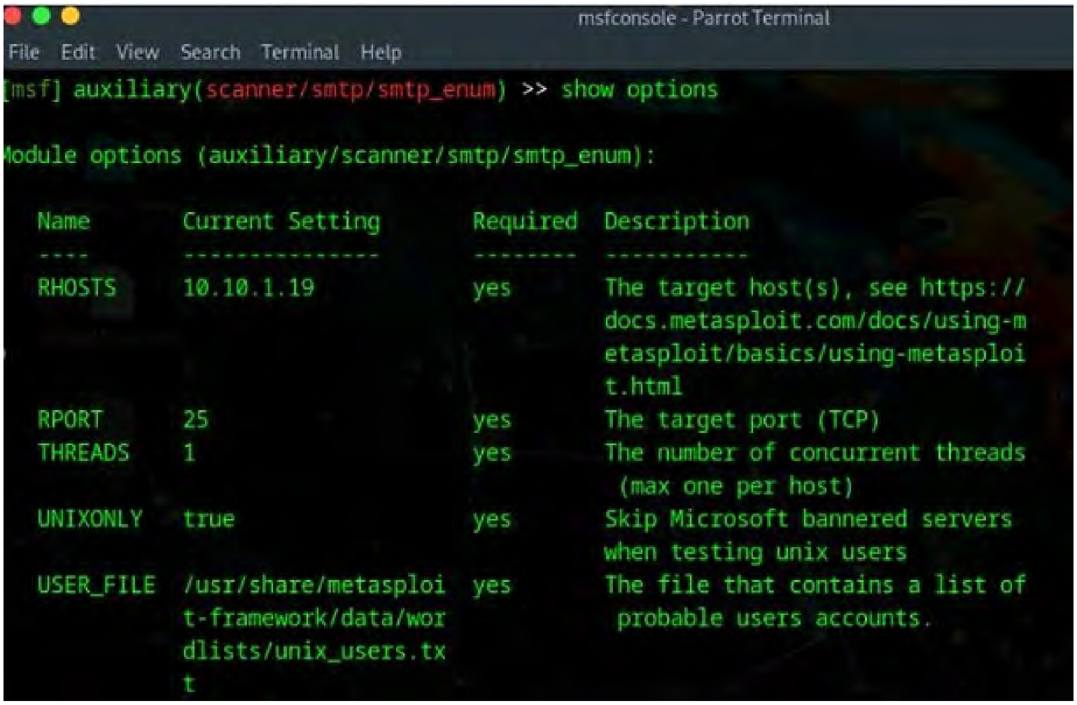
Attackers utilize the Metasploit framework to list SMTP users. This framework includes a module for SMTP enumeration that enables attackers to connect to the specified SMTP server and retrieve usernames from predefined wordlists. The SMTP server employs its built-in VRFY method to confirm the usernames in the wordlist against the users available on the server and presents the list of matched users.

Steps to Enumerate SMTP Users Using Metasploit

Step 1: Start the Metasploit msfconsole and change to the appropriate auxiliary scanner to begin the process: auxiliary/scanner/smtp/smtp_enum.

```
msf > use auxiliary/scanner/smtp/smtp_enum
msf auxiliary(smtp_enum) >
```

Step 2: Execute the command show options to see the complete list of choices needed to carry out this task. Alternatively, you can use the command show evasion to discover options for bypassing security measures.



msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
[msf] auxiliary(scanner/smtp/smtp_enum) >> show options

Module options (auxiliary/scanner/smtp/smtp_enum):

   Name       Current Setting      Required  Description
   ----       ---------------      --------  -----------
   RHOSTS     10.10.1.19           yes       The target host(s), see https://
                                             docs.metasploit.com/docs/using-m
                                             etasploit/basics/using-metasploi
                                             t.html
   RPORT      25                   yes       The target port (TCP)
   THREADS    1                    yes       The number of concurrent threads
                                              (max one per host)
   UNIXONLY   true                 yes       Skip Microsoft bannered servers
                                             when testing unix users
   USER_FILE  /usr/share/metasploi yes       The file that contains a list of
              t-framework/data/wor            probable users accounts.
              dlists/unix_users.tx
              t
```

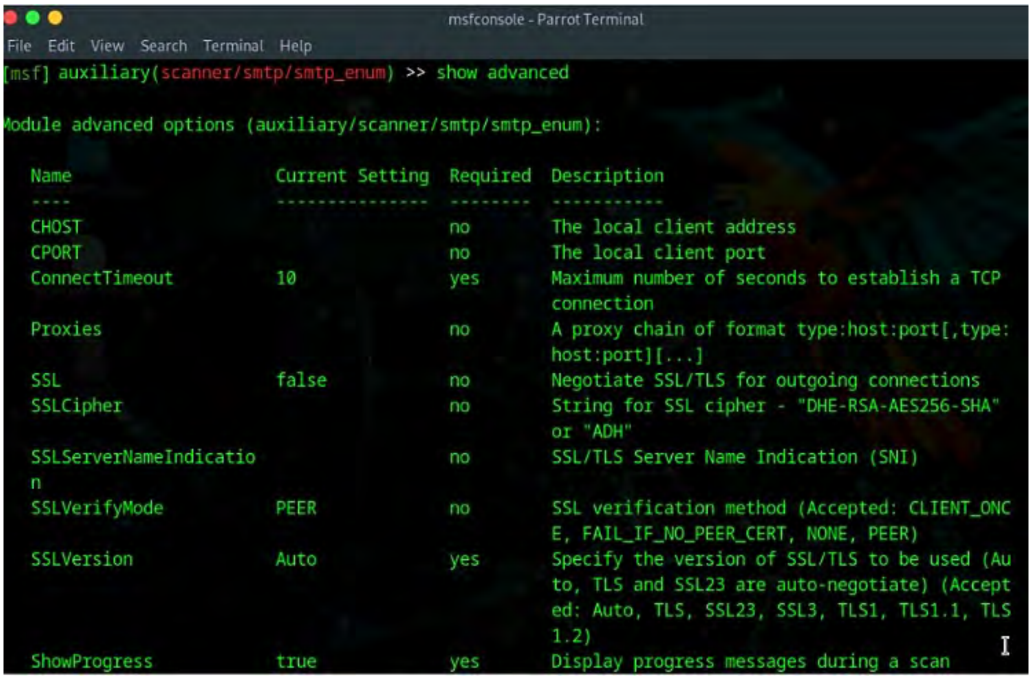
Figure 4-35: Screenshot of Metasploit showing smtp_enum Options

Step 3: Utilize the option **set RHOST** to specify the IP address of the target SMTP server or to indicate a range of IP addresses.

Step 4: By default, the Metasploit framework employs standard wordlists found at **/usr/share/metasploit-framework/data/wordlists/unix_users.txt** to identify SMTP users. The USER_FILE option can be configured to employ custom wordlists.

```
msf auxiliary(smtp_enum) > set USER_FILE <path to the wordlists file>
```

Step 5: Enter the command show advanced to see the full list of available options in the SMTP user enumeration module.

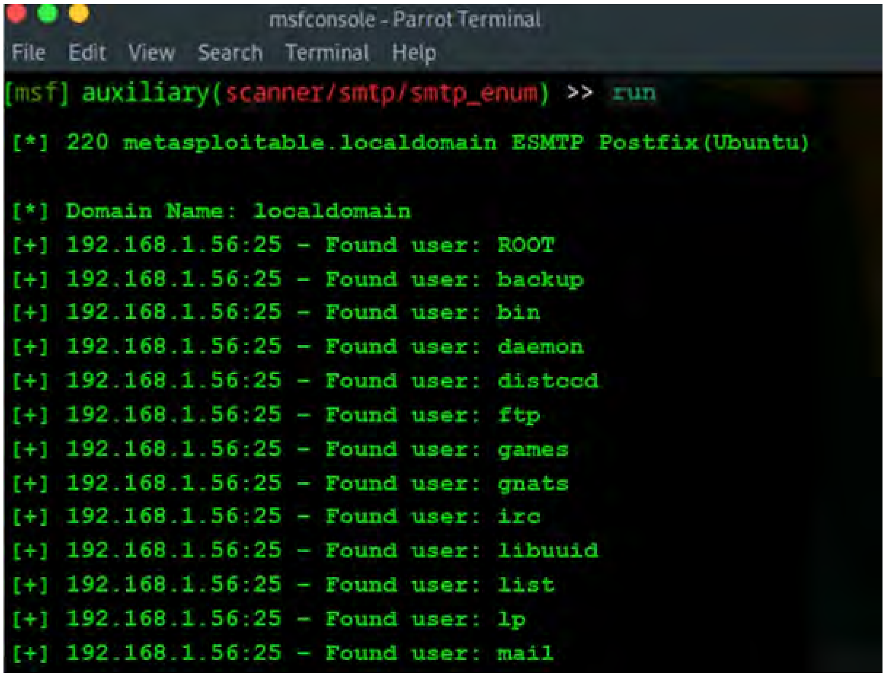


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
[msf] auxiliary(scanner/smtp/smtp_enum) >> show advanced

Module advanced options (auxiliary/scanner/smtp/smtp_enum):

   Name                   Current Setting  Required  Description
   ----                   ---------------  --------  -----------
   CHOST                                   no        The local client address
   CPORT                                   no        The local client port
   ConnectTimeout         10               yes       Maximum number of seconds to establish a TCP
                                                     connection
   Proxies                                 no        A proxy chain of format type:host:port[,type:
                                                     host:port][...]
   SSL                    false            no        Negotiate SSL/TLS for outgoing connections
   SSLCipher                               no        String for SSL cipher - "DHE-RSA-AES256-SHA"
                                                     or "ADH"
   SSLServerNameIndicatio                  no        SSL/TLS Server Name Indication (SNI)
   n
   SSLVerifyMode          PEER             no        SSL verification method (Accepted: CLIENT_ONC
                                                     E, FAIL_IF_NO_PEER_CERT, NONE, PEER)
   SSLVersion             Auto             yes       Specify the version of SSL/TLS to be used (Au
                                                     to, TLS and SSL23 are auto-negotiate) (Accept
                                                     ed: Auto, TLS, SSL23, SSL3, TLS1, TLS1.1, TLS
                                                     1.2)
   ShowProgress           true             yes       Display progress messages during a scan
```

Figure 4-36: Screenshot of Metasploit showing smtp_enum advanced Options

Step 6: Execute the **run** command to initiate the enumeration procedure. It examines the specified wordlists against the usernames of the SMTP server and displays all the matching usernames.



msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
[msf] auxiliary(scanner/smtp/smtp_enum) >> run

[*] 220 metasploitable.localdomain ESMTP Postfix(Ubuntu)

[*] Domain Name: localdomain
[+] 192.168.1.56:25 - Found user: ROOT
[+] 192.168.1.56:25 - Found user: backup
[+] 192.168.1.56:25 - Found user: bin
[+] 192.168.1.56:25 - Found user: daemon
[+] 192.168.1.56:25 - Found user: distccd
[+] 192.168.1.56:25 - Found user: ftp
[+] 192.168.1.56:25 - Found user: games
[+] 192.168.1.56:25 - Found user: gnats
[+] 192.168.1.56:25 - Found user: irc
[+] 192.168.1.56:25 - Found user: libuuid
[+] 192.168.1.56:25 - Found user: list
[+] 192.168.1.56:25 - Found user: lp
[+] 192.168.1.56:25 - Found user: mail
```

Figure 4-37: Screenshot of Metasploit retrieving SMTP Users

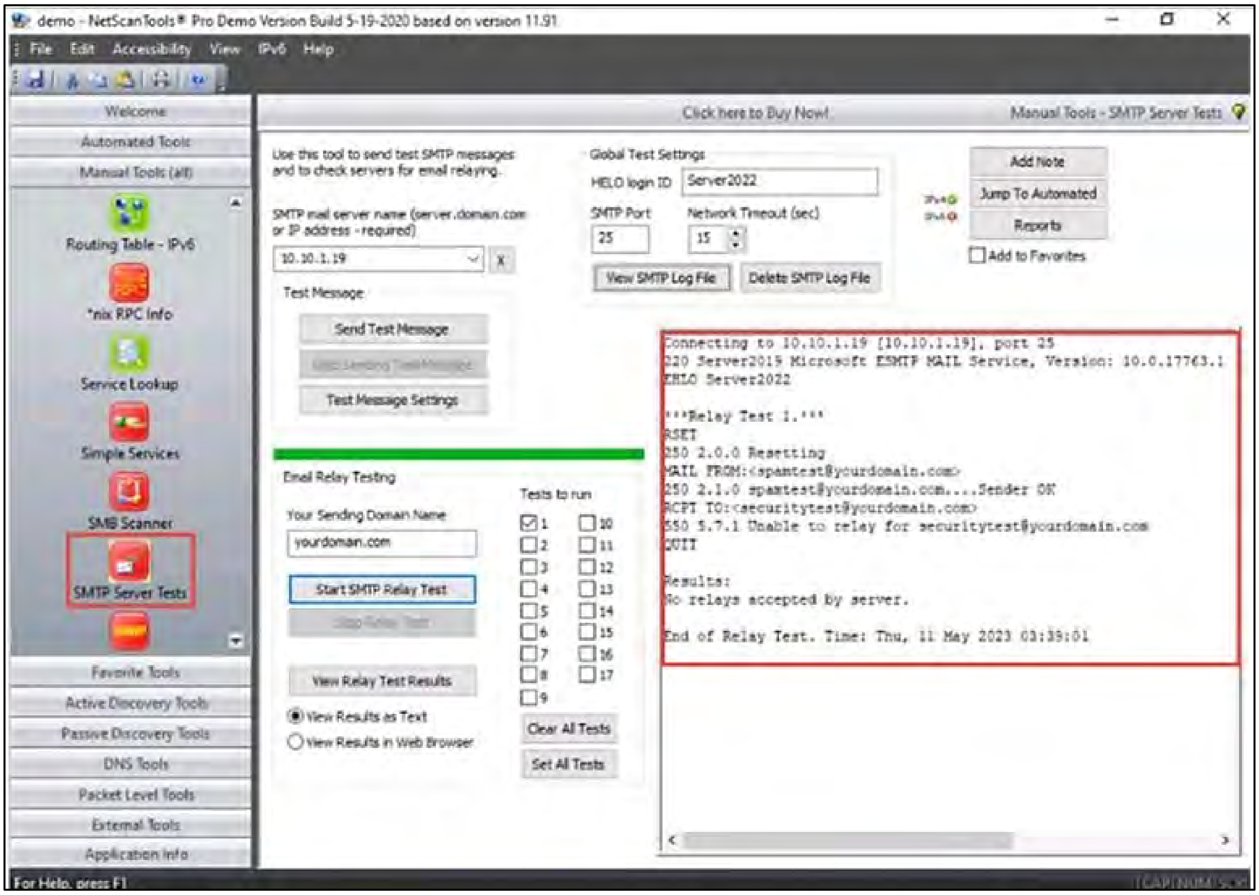
As illustrated in Figure 4-37, attackers acquire a list of legitimate SMTP users from the targeted SMTP server, which they can utilize to launch focused attacks.

SMTP Enumeration Tools

SMTP enumeration tools are utilized to carry out username enumeration. Attackers can leverage the usernames gathered from this process to initiate additional attacks on other systems within the network.

NetScanTools Pro

NetScanTools Pro's SMTP Email Generator utility assesses the functionality of dispatching an email via an SMTP server. Malicious actors utilize NetScanTools Pro for SMTP enumeration to gather all email header details, which encompass confirm/urgent flags. Additionally, attackers can document the email session in a log file and subsequently examine the interactions between NetScanTools Pro and the SMTP server within that log file.



Use this tool to send test SMTP messages and to check servers for email relaying.

SMTP mail server name (server.domain.com or IP address - required): 10.10.1.19

Global Test Settings

HELO login ID: Server2022

SMTP Port: 25

Network Timeout (sec): 15

□ Add to Favorites

Test Message

Email Relay Testing

Your Sending Domain Name: yourdomain.com

Tests to run: ☑1 □2 □3 □4 □5 □6 □7 □8 □9 □10 □11 □12 □13 □14 □15 □16 □17

● View Results as Text
○ View Results in Web Browser

```
Connecting to 10.10.1.19 [10.10.1.19], port 25
220 Server2019 Microsoft ESMTP MAIL Service, Version: 10.0.17763.1
EHLO Server2022

***Relay Test 1.***
RSET
250 2.0.0 Resetting
MAIL FROM:<spamtest@yourdomain.com>
250 2.1.0 spamtest@yourdomain.com....Sender OK
RCPT TO:<securitytest@yourdomain.com>
550 5.7.1 Unable to relay for securitytest@yourdomain.com
QUIT

Results:
No relays accepted by server.

End of Relay Test. Time: Thu, 11 May 2023 03:39:01
```

Figure 4-38: Screenshot of NetScanTools Pro

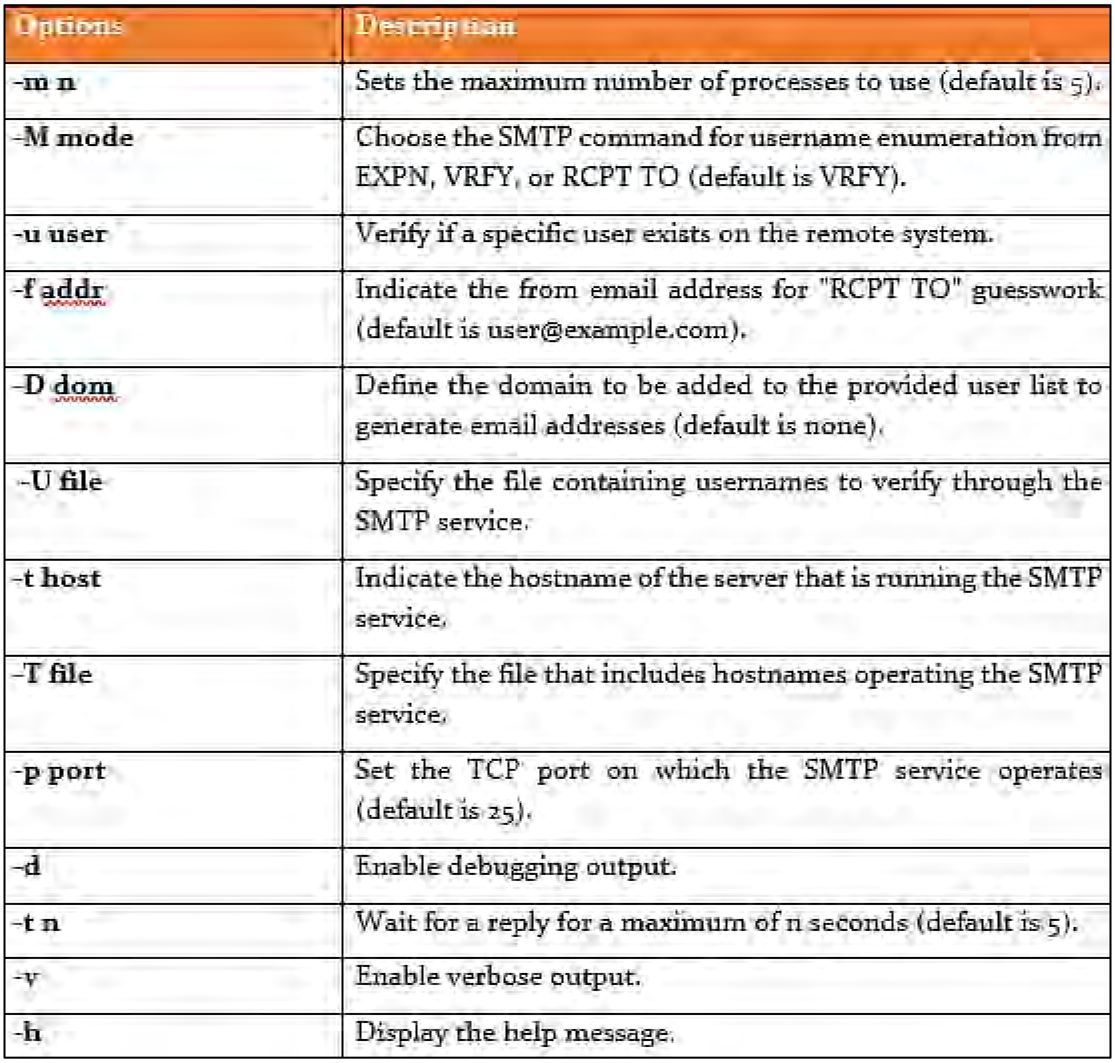
smtp-user-enum

smtp-user-enum is a utility designed to identify operating system-level user accounts on Solaris through the SMTP service (sendmail). The enumeration process involves analyzing the replies to the VRFY, EXPN, and RCPT TO commands. As depicted in the screenshot, smtp-user-enum requires a list of users to be provided along with at least one target that operates an SMTP service.

The syntax for employing smtp-user-enum is outlined as follows:

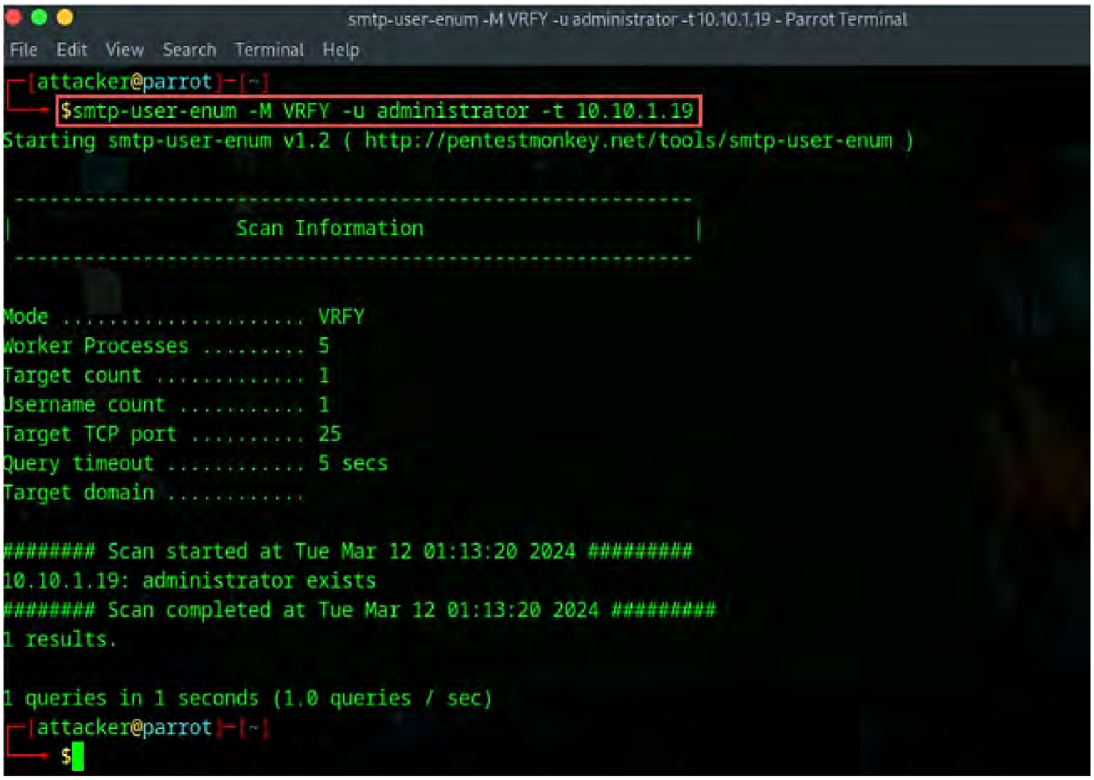
```
smtp-user-enum.pl [options] (-u username|-U file-of-usernames) (-t host|-T file-of-targets)
```

smtp-user-enum offers the options mentioned in Table 4-09.



Options	Description
-m n	Sets the maximum number of processes to use (default is 5).
-M mode	Choose the SMTP command for username enumeration from EXPN, VRFY, or RCPT TO (default is VRFY).
-u user	Verify if a specific user exists on the remote system.
-f addr	Indicate the from email address for "RCPT TO" guesswork (default is user@example.com).
-D dom	Define the domain to be added to the provided user list to generate email addresses (default is none).
-U file	Specify the file containing usernames to verify through the SMTP service.
-t host	Indicate the hostname of the server that is running the SMTP service.
-T file	Specify the file that includes hostnames operating the SMTP service.
-p port	Set the TCP port on which the SMTP service operates (default is 25).
-d	Enable debugging output.
-t n	Wait for a reply for a maximum of n seconds (default is 5).
-v	Enable verbose output.
-h	Display the help message.

Table 4-09: smtp-user-enum Options with Description



smtp-user-enum -M VRFY -u administrator -t 10.10.1.19 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $smtp-user-enum -M VRFY -u administrator -t 10.10.1.19
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Mode ..................... VRFY
Worker Processes ......... 5
Target count ............. 1
Username count ........... 1
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............

######## Scan started at Tue Mar 12 01:13:20 2024 #########
10.10.1.19: administrator exists
######## Scan completed at Tue Mar 12 01:13:20 2024 #########
1 results.

1 queries in 1 seconds (1.0 queries / sec)
┌─[attacker@parrot]─[~]
└──╼ $
```

Figure 4-39: Screenshot of smtp-user-enum

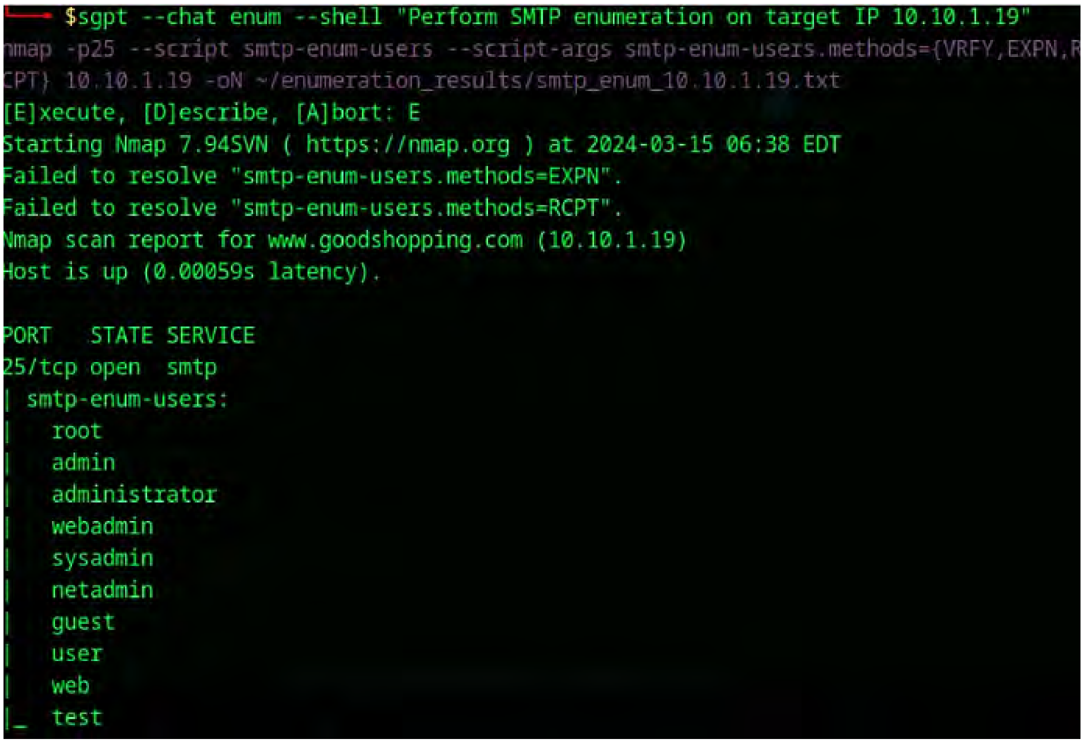
SMTP Enumeration using AI

Attackers can utilize AI-driven technologies to improve and automate their processes for network enumeration. With the support of AI, these attackers can easily carry out SMTP enumeration and collect user data from SMTP on specific IP addresses.

Attackers can employ ChatGPT for this purpose by crafting suitable prompts such as:

Example 1:

```
Perform SMTP enumeration on target IP 10.10.1.19.
```



```
└──╼ $sgpt --chat enum --shell "Perform SMTP enumeration on target IP 10.10.1.19"
nmap -p25 --script smtp-enum-users --script-args smtp-enum-users.methods={VRFY,EXPN,R
CPT} 10.10.1.19 -oN ~/enumeration_results/smtp_enum_10.10.1.19.txt
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 06:38 EDT
Failed to resolve "smtp-enum-users.methods=EXPN".
Failed to resolve "smtp-enum-users.methods=RCPT".
Nmap scan report for www.goodshopping.com (10.10.1.19)
Host is up (0.00059s latency).

PORT   STATE SERVICE
25/tcp open  smtp
| smtp-enum-users:
|   root
|   admin
|   administrator
|   webadmin
|   sysadmin
|   netadmin
|   guest
|   user
|   web
|_  test
```

Figure 4-40: Perform SMTP Enumeration on Target IP with nmap

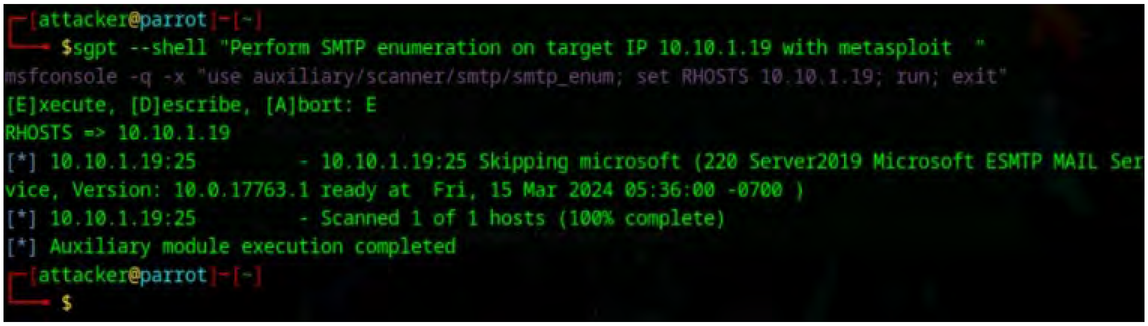
The command below is intended to streamline the process of SMTP enumeration for the given target IP:

```
nmap -p25 --script smtp-enum-users --script-args smtp-enum-users.methods={VRFY, EXPN, RCPT} 10.10.1.19 -oN ~/enumeration_results/smtp_enum_10.10.1.19.txt
```

This command utilizes nmap with particular options and the smtp-enum-users script to conduct SMTP enumeration on port 25 of the target IP 10.10.1.19. The output is stored in the file smtp_enum_10.10.1.19.txt located in the ~/enumeration_results/ directory.

Example 2:

Perform SMTP enumeration on target IP 10.10.1.19 with Metasploit.



```
┌─[attacker@parrot]─[~]
└──╼ $sgpt --shell "Perform SMTP enumeration on target IP 10.10.1.19 with metasploit  "
msfconsole -q -x "use auxiliary/scanner/smtp/smtp_enum; set RHOSTS 10.10.1.19; run; exit"
[E]xecute, [D]escribe, [A]bort: E
RHOSTS => 10.10.1.19
[*] 10.10.1.19:25         - 10.10.1.19:25 Skipping microsoft (220 Server2019 Microsoft ESMTP MAIL Ser
vice, Version: 10.0.17763.1 ready at  Fri, 15 Mar 2024 05:36:00 -0700 )
[*] 10.10.1.19:25         - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
┌─[attacker@parrot]─[~]
└──╼ $
```

Figure 4-41: Perform SMTP Enumeration on Target IP with Metasploit

The following command is designed to automate SMTP enumeration tasks on the specified target IP:

```
msfconsole -q -x "use auxiliary/scanner/smtp/smtp_enum; set RHOSTS 10.10.1.19; run; exit"
```

This command runs Metasploit in quiet mode by using msfconsole with the -q option. The -x option specifies the commands to be executed within Metasploit. It designates the target IP as 10.10.1.19 and runs the smtp_enum auxiliary module, which conducts SMTP enumeration.

These commands streamline the SMTP enumeration process and present the results for every command executed on the specified target IP 10.10.1.19.

DNS Enumeration using Zone Transfer

DNS zone transfer is the method of transferring a copy of the DNS zone file from the primary DNS server to a secondary DNS server. Typically, the primary DNS server has a backup or secondary server for redundancy, which contains all the information from the primary server. The DNS server utilizes zone transfers to convey any modifications made on the main server to the secondary server(s). An attacker conducts DNS zone transfer enumeration to identify the DNS server and access the records of the target organization. If the target organization's DNS server permits zone transfers, attackers can execute a DNS zone transfer to gather DNS server names, hostnames, machine names, usernames, IP addresses, aliases, and so on, allocated within a target domain.

In the context of DNS enumeration through zone transfer, an attacker seeks to obtain a full copy of the zone file for a domain from the DNS server. Attackers can use tools like nslookup, the dig command, and DNSRecon to carry out DNS zone transfers. If the DNS transfer configuration is enabled on the targeted name server, it will share the DNS information; otherwise, it will return an error indicating that the zone transfer has failed or been denied.

To initiate a DNS zone transfer, the attacker sends a zone-transfer request to the DNS server, masquerading as a client; the DNS server then responds by sending a portion of its database as a zone to the attacker. This zone can contain a considerable amount of information related to the DNS zone network.

dig Command

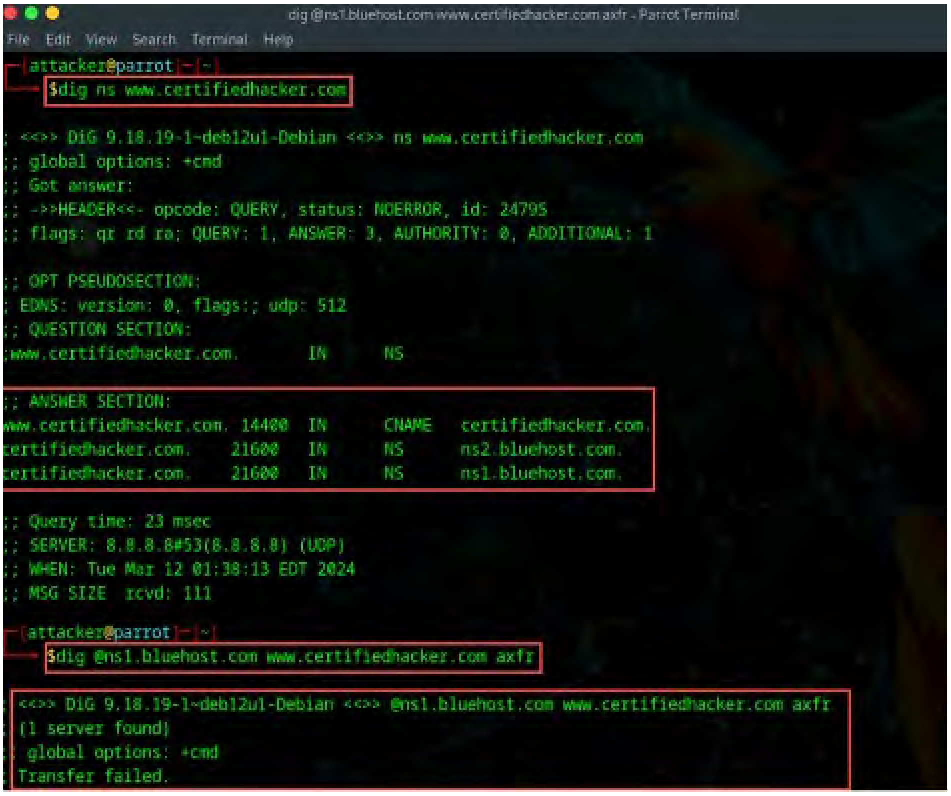
Attackers employ the **dig** command on Linux systems to interrogate DNS name servers and obtain details regarding target host addresses, name servers, mail exchange servers, and more. As demonstrated in the screenshot, attackers execute the subsequent command to carry out a DNS zone transfer:

```
dig ns <target domain>
```

The command mentioned above retrieves all the DNS name servers associated with the specified domain.

Following that, attackers utilize one of the name servers obtained from the previous command to check if the target DNS permits zone transfers. For this purpose, they employ the following command:

```
dig @<domain of name server> <target domain> axfr
```



dig @ns1.bluehost.com www.certifiedhacker.com axfr - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $dig ns www.certifiedhacker.com

; <<>> DiG 9.18.19-1-deb12u1-Debian <<>> ns www.certifiedhacker.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24795
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.certifiedhacker.com.       IN      NS

;; ANSWER SECTION:
www.certifiedhacker.com. 14400  IN      CNAME   certifiedhacker.com.
certifiedhacker.com.    21600   IN      NS      ns2.bluehost.com.
certifiedhacker.com.    21600   IN      NS      ns1.bluehost.com.

;; Query time: 23 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (UDP)
;; WHEN: Tue Mar 12 01:38:13 EDT 2024
;; MSG SIZE  rcvd: 111

┌─[attacker@parrot]─[~]
└──╼ $dig @ns1.bluehost.com www.certifiedhacker.com axfr

; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> @ns1.bluehost.com www.certifiedhacker.com axfr
; (1 server found)
;; global options: +cmd
; Transfer failed.
```

Figure 4-42: Screenshot of Linux DNS Zone Transfer using dig Command

nslookup Command

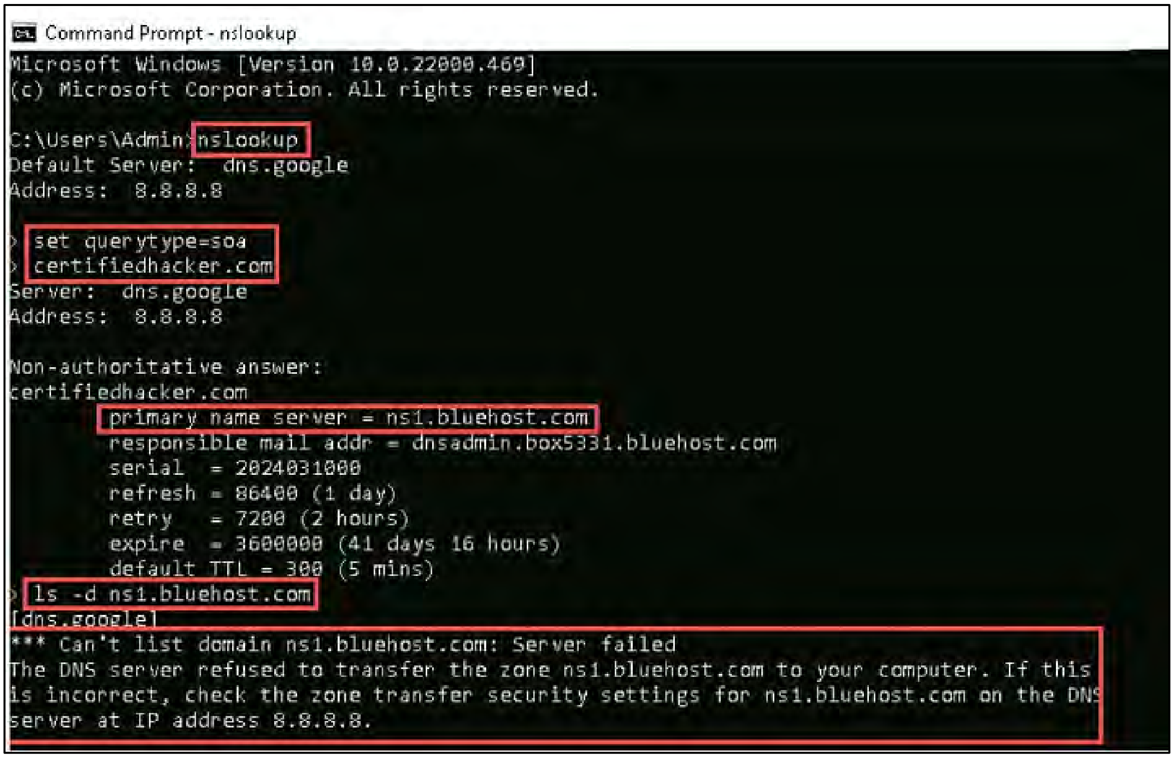
Attackers utilize the nslookup command on Windows systems to request information from DNS name servers, such as the host addresses, name servers, and mail exchanges associated with the target. As illustrated in Figure 4-43, attackers execute the command below to carry out a DNS zone transfer:

```
nslookup
set querytype=soa
<target domain>
```

The command mentioned above configures the query type to the Start of Authority (SOA) record in order to obtain administrative details about the DNS zone for the target domain certifiedhacker.com.

The following command is utilized to try to transfer the zone of the designated name server:

```
/ls -d <domain of name server>
```



```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>nslookup
Default Server:  dns.google
Address:  8.8.8.8

> set querytype=soa
> certifiedhacker.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
certifiedhacker.com
        primary name server = ns1.bluehost.com
        responsible mail addr = dnsadmin.box5331.bluehost.com
        serial  = 2024031000
        refresh = 86400 (1 day)
        retry   = 7200 (2 hours)
        expire  = 3600000 (41 days 16 hours)
        default TTL = 300 (5 mins)
> ls -d ns1.bluehost.com
[dns.google]
*** Can't list domain ns1.bluehost.com: Server failed
The DNS server refused to transfer the zone ns1.bluehost.com to your computer. If this
is incorrect, check the zone transfer security settings for ns1.bluehost.com on the DNS
server at IP address 8.8.8.8.
```

Figure 4-43: Screenshot of Windows DNS Zone Transfer using the nslookup Command

DNSRecon

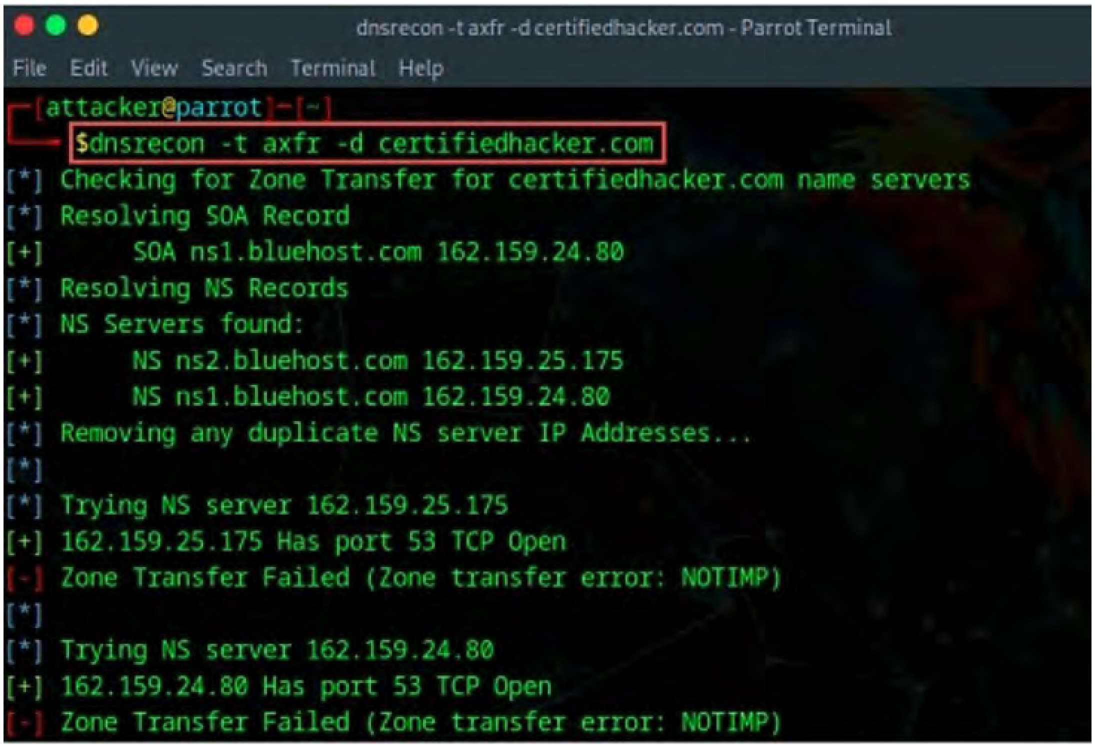
Attacker utilizes DNSRecon to verify all Name Server (NS) records associated with the target domain for potential zone transfers. As demonstrated in the screenshot, these attackers execute the subsequent command for performing a DNS zone transfer:

```
dnsrecon -t axfr -d <target domain>
```

Table 4-10 describes each option used in the command.

Command Breakdown	Description
-t	iIdicates the type of enumeration to execute.
axfr	Refers to the enumeration type that assesses all NS servers for a zone transfer.
-d	Designates the target domain.

Table 4-10: dnsrecon Command Description



dnsrecon -t axfr -d certifiedhacker.com - Parrot Terminal

```
┌─[attacker@parrot]─[~]
└──╼ $dnsrecon -t axfr -d certifiedhacker.com
[*] Checking for Zone Transfer for certifiedhacker.com name servers
[*] Resolving SOA Record
[+]      SOA ns1.bluehost.com 162.159.24.80
[*] Resolving NS Records
[*] NS Servers found:
[+]      NS ns2.bluehost.com 162.159.25.175
[+]      NS ns1.bluehost.com 162.159.24.80
[*] Removing any duplicate NS server IP Addresses...
[*]
[*] Trying NS server 162.159.25.175
[+] 162.159.25.175 Has port 53 TCP Open
[-] Zone Transfer Failed (Zone transfer error: NOTIMP)
[*]
[*] Trying NS server 162.159.24.80
[+] 162.159.24.80 Has port 53 TCP Open
[-] Zone Transfer Failed (Zone transfer error: NOTIMP)
```

Figure 4-44: DNS Zone Transfer using DNSRecon

DNS Cache Snooping

DNS cache snooping is a method of DNS enumeration where an attacker queries a DNS server for a particular cached DNS record. By accessing this cached information, the attacker can identify the websites that the user has recently visited. This data can further uncover critical information, including the name of the DNS server owner, the service provider, the vendor's name, and banking details. With this information, the attacker can conduct a social engineering attack on the targeted user. Attackers employ various tools, such as the dig command and DNSRecon, to carry out DNS cache snooping.

Attackers use the following two techniques to snoop on a target domain:

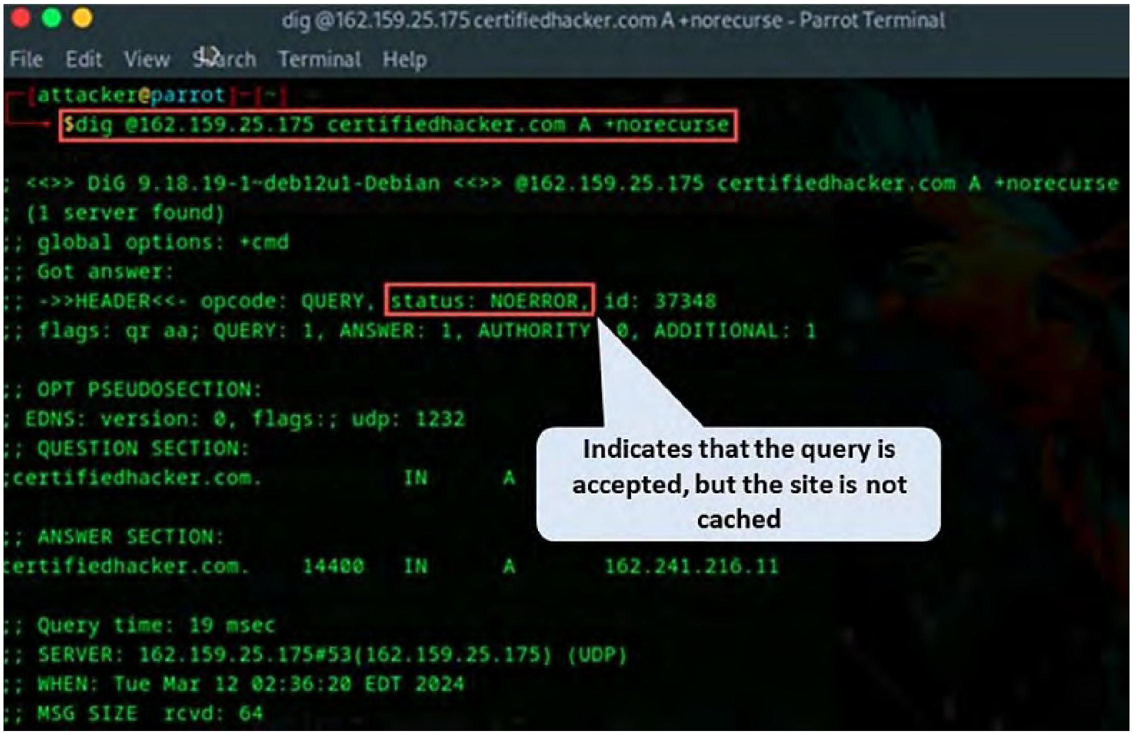
Non-recursive Method

In this approach, attackers gather information about a DNS server by issuing a non-recursive query, which involves setting the Recursion Desired (RD) bit in the query header to zero. They request specific DNS records such as A, CNAME, PTR, CERT, SRV, and MX from the DNS cache. If the requested record exists in the DNS cache, the DNS server will provide a response indicating that a user on the system has accessed a particular domain. If not, the DNS server will offer information about another DNS server that can potentially provide an answer to the query, or it may return the root.hints file that contains details about all root DNS servers.

Attackers employ the **dig** command along with the DNS server's name or IP address, the domain name, and the type of DNS record. The +**norecurse** option is utilized to configure the query as non-recursive.

```
dig @<IP of DNS server> <Target domain> A +norecurse
```

Figure 4-45 demonstrates that the **NOERROR** status suggests the query was received, but no response was provided, indicating that no user from the system accessed the requested site.



dig @162.159.25.175 certifiedhacker.com A +norecurse - Parrot Terminal
File Edit View Search Terminal Help

```
┌[attacker@parrot]─[~]
└──╼ $dig @162.159.25.175 certifiedhacker.com A +norecurse

; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> @162.159.25.175 certifiedhacker.com A +norecurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37348
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;certifiedhacker.com.           IN      A

;; ANSWER SECTION:
certifiedhacker.com.    14400   IN      A       162.241.216.11

;; Query time: 19 msec
;; SERVER: 162.159.25.175#53(162.159.25.175) (UDP)
;; WHEN: Tue Mar 12 02:36:20 EDT 2024
;; MSG SIZE  rcvd: 64
```

Indicates that the query is accepted, but the site is not cached

Figure 4-45: Screenshot of a dig query for a Site that is not Cached

Recursive Method

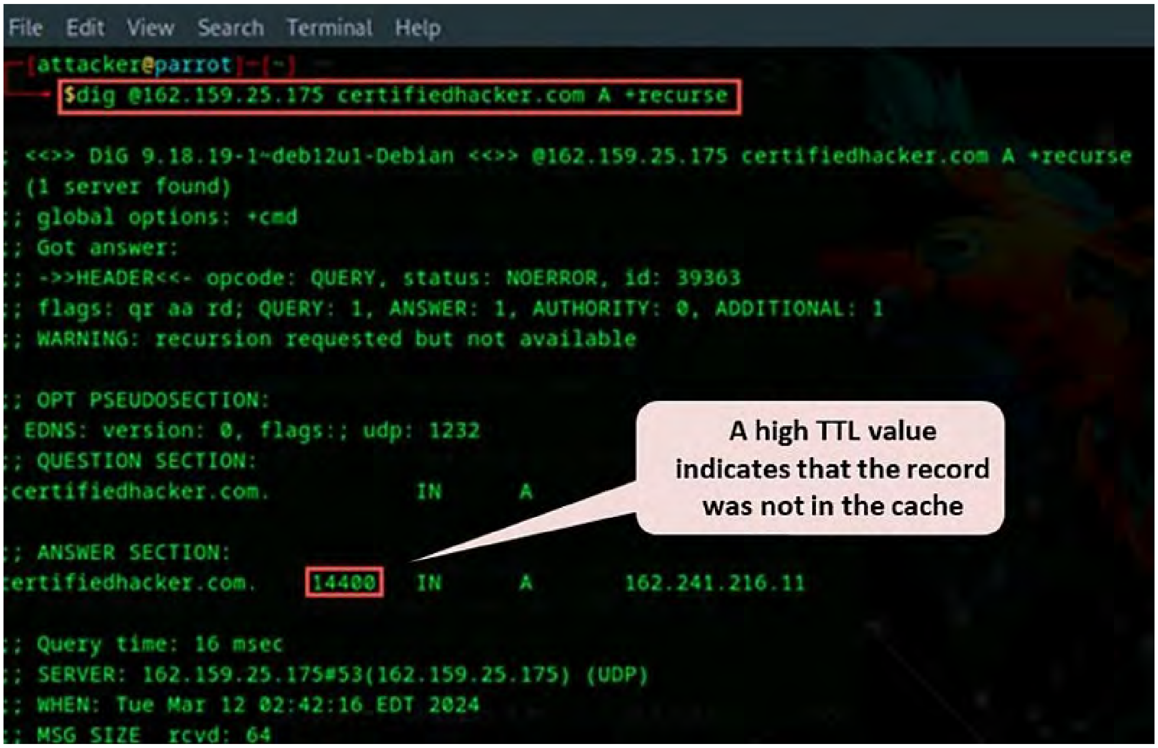
In this technique, attackers aim to observe the DNS server by issuing a recursive query while using the **+recurse** option in place of the **+norecurse** option. Like the non-recursive approach, the attackers seek specific DNS records such as A, CNAME, PTR, CERT, SRV, and MX from the DNS cache.

In this technique, the Time-To-Live (TTL) value is analyzed to ascertain how long the DNS record will stay in the cache. The TTL value resulting from the query is then compared to the TTL that was originally configured in the TTL field. If the TTL value in the result is lower than the initial TTL value, this indicates that the record is cached, suggesting that someone on the system has accessed that site. Conversely, if the requested record is not found in the cache, it will be stored in the cache following the initial query.

Attackers utilize the same **dig** command as in the non-recursive approach but substitute the **+recurse** option for the **+norecurse** option:

```
dig @<IP of DNS server> <Target domain> A +recurse
```

The TTL value for the domain certifiedhacker.com, as displayed in Figure 4-46, is quite high, indicating that the domain record was likely not present in the cache at the time the query was made.



```
-[attacker@parrot]-[~]
   $dig @162.159.25.175 certifiedhacker.com A +recurse

 <<>> DiG 9.18.19-1~deb12u1-Debian <<>> @162.159.25.175 certifiedhacker.com A +recurse
 (1 server found)
; global options: +cmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39363
; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
; WARNING: recursion requested but not available

; OPT PSEUDOSECTION:
 EDNS: version: 0, flags:; udp: 1232
; QUESTION SECTION:
certifiedhacker.com.          IN      A

; ANSWER SECTION:
ertifiedhacker.com.     14400   IN      A       162.241.216.11

; Query time: 16 msec
; SERVER: 162.159.25.175#53(162.159.25.175) (UDP)
; WHEN: Tue Mar 12 02:42:16 EDT 2024
; MSG SIZE  rcvd: 64
```

A high TTL value indicates that the record was not in the cache

Figure 4-46: Screenshot of a dig query for a Cached Site

DNSSEC Zone Walking

Domain Name System Security Extensions (DNSSEC) zone walking is a method used by attackers to gather internal records when the DNS zone is poorly configured. The gathered zone data can help the attacker create a map of the host network.

Organizations implement DNSSEC to enhance the security features of DNS data and defend against established DNS threats. This security mechanism utilizes digital signatures derived from public-key cryptography to improve DNS authentication. These digital signatures are kept in the DNS name servers alongside standard records such as MX, A, AAAA, and CNAME.

Although DNSSEC enhances Internet security, it remains vulnerable to a threat known as zone enumeration or zone walkingBy exploiting this vulnerability, attackers are able to obtain network details from a targeted domain, which they could then utilize to execute Internet-based attacks.

To mitigate the issue of zone enumeration, a newer iteration of DNSSEC that employs Next Secure version 3 (NSEC3) has been developed. The NSEC3 record serves the same purpose as NSEC records but offers cryptographically hashed record names to inhibit the listing of record names within the zone.

Attackers can deploy various DNSSEC zone enumeration tools, such as LDNS, DNSRecon, nsec3map, nsec3walker, and DNSwalk, to execute zone enumeration.

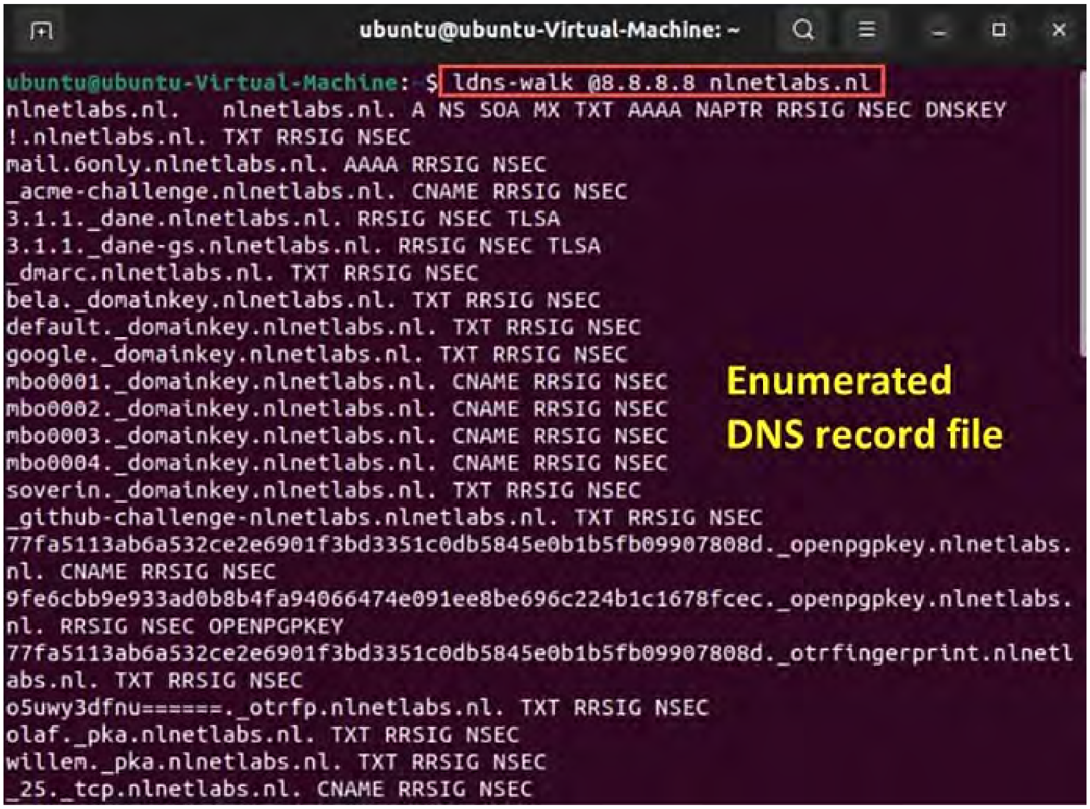
DNSSEC Zone Walking Tools

DNSSEC zone walking tools are utilized to gather information about the DNS record files of the target domain. These tools are also capable of conducting zone enumeration on NSEC and NSEC3 record files. They can leverage the acquired data to execute attacks like Denial-of-Service (DoS) attacks and phishing attacks.

LDNS

LDNS-walk performs an enumeration of the DNSSEC zone and retrieves results related to the DNS record files. As illustrated in Figure 4-47, attackers employ the following query to enumerate the target domain iana.org by utilizing the DNS server 8.8.8.8 in order to acquire DNS record files:

```
ldns-walk @<IP of DNS Server> <Target domain>
```



```
ubuntu@ubuntu-Virtual-Machine: $ ldns-walk @8.8.8.8 nlnetlabs.nl
nlnetlabs.nl.   nlnetlabs.nl. A NS SOA MX TXT AAAA NAPTR RRSIG NSEC DNSKEY
!.nlnetlabs.nl. TXT RRSIG NSEC
mail.6only.nlnetlabs.nl. AAAA RRSIG NSEC
_acme-challenge.nlnetlabs.nl. CNAME RRSIG NSEC
3.1.1._dane.nlnetlabs.nl. RRSIG NSEC TLSA
3.1.1._dane-gs.nlnetlabs.nl. RRSIG NSEC TLSA
_dmarc.nlnetlabs.nl. TXT RRSIG NSEC
bela._domainkey.nlnetlabs.nl. TXT RRSIG NSEC
default._domainkey.nlnetlabs.nl. TXT RRSIG NSEC
google._domainkey.nlnetlabs.nl. TXT RRSIG NSEC
mbo0001._domainkey.nlnetlabs.nl. CNAME RRSIG NSEC
mbo0002._domainkey.nlnetlabs.nl. CNAME RRSIG NSEC
mbo0003._domainkey.nlnetlabs.nl. CNAME RRSIG NSEC
mbo0004._domainkey.nlnetlabs.nl. CNAME RRSIG NSEC
soverin._domainkey.nlnetlabs.nl. TXT RRSIG NSEC
_github-challenge-nlnetlabs.nlnetlabs.nl. TXT RRSIG NSEC
77fa5113ab6a532ce2e6901f3bd3351c0db5845e0b1b5fb09907808d._openpgpkey.nlnetlabs.
nl. CNAME RRSIG NSEC
9fe6cbb9e933ad0b8b4fa94066474e091ee8be696c224b1c1678fcec._openpgpkey.nlnetlabs.
nl. RRSIG NSEC OPENPGPKEY
77fa5113ab6a532ce2e6901f3bd3351c0db5845e0b1b5fb09907808d._otrfingerprint.nlnetl
abs.nl. TXT RRSIG NSEC
o5uwy3dfnu======._otrfp.nlnetlabs.nl. TXT RRSIG NSEC
olaf._pka.nlnetlabs.nl. TXT RRSIG NSEC
willem._pka.nlnetlabs.nl. TXT RRSIG NSEC
_25._tcp.nlnetlabs.nl. CNAME RRSIG NSEC
```

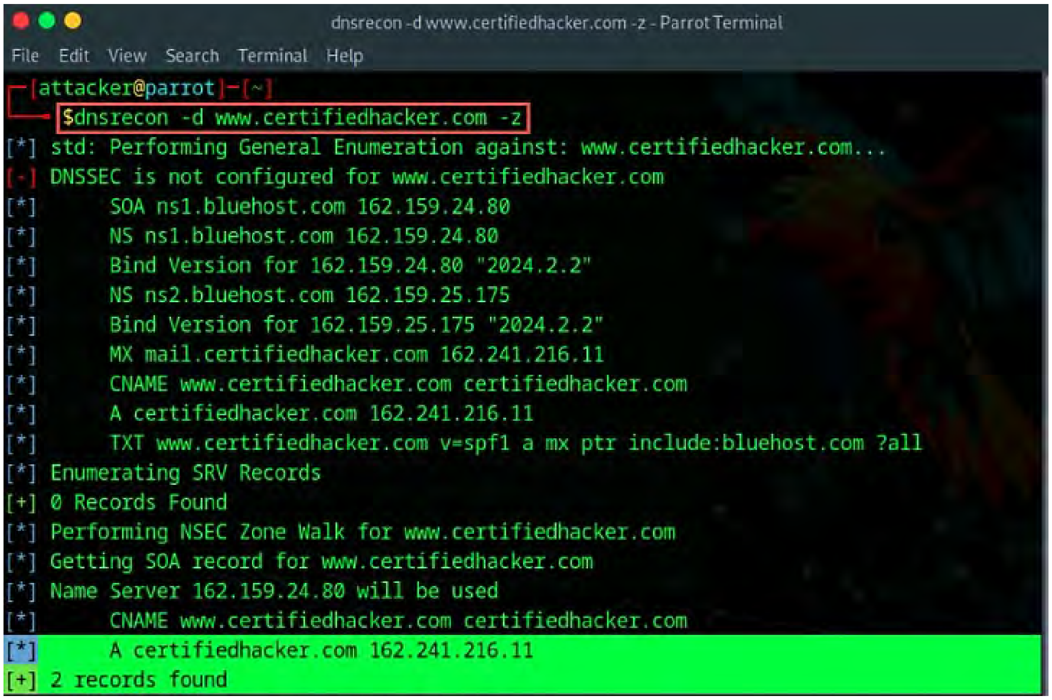
Figure 4-47: Screenshot of LDNS displaying Results on the Target Domain

DNSRecon

DNSRecon is a tool designed for zone enumeration that helps users gather DNS records like A, AAAA, and CNAME. It also conducts NSEC zone enumeration to retrieve DNS record files associated with a specific domain.

As shown in Figure 4-48, attackers utilize the following query to carry out zone enumeration on the target domain certifiedhacker.com:

```
dnsrecon -d <target domain> -z
```



dnsrecon -d www.certifiedhacker.com -z - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $dnsrecon -d www.certifiedhacker.com -z
[*] std: Performing General Enumeration against: www.certifiedhacker.com...
[-] DNSSEC is not configured for www.certifiedhacker.com
[*]      SOA ns1.bluehost.com 162.159.24.80
[*]      NS ns1.bluehost.com 162.159.24.80
[*]      Bind Version for 162.159.24.80 "2024.2.2"
[*]      NS ns2.bluehost.com 162.159.25.175
[*]      Bind Version for 162.159.25.175 "2024.2.2"
[*]      MX mail.certifiedhacker.com 162.241.216.11
[*]      CNAME www.certifiedhacker.com certifiedhacker.com
[*]      A certifiedhacker.com 162.241.216.11
[*]      TXT www.certifiedhacker.com v=spf1 a mx ptr include:bluehost.com ?all
[*] Enumerating SRV Records
[+] 0 Records Found
[*] Performing NSEC Zone Walk for www.certifiedhacker.com
[*] Getting SOA record for www.certifiedhacker.com
[*] Name Server 162.159.24.80 will be used
[*]      CNAME www.certifiedhacker.com certifiedhacker.com
[*]      A certifiedhacker.com 162.241.216.11
[+] 2 records found
```

Figure 4-48: Screenshot of DNSRecon displaying Results on the Target Domain

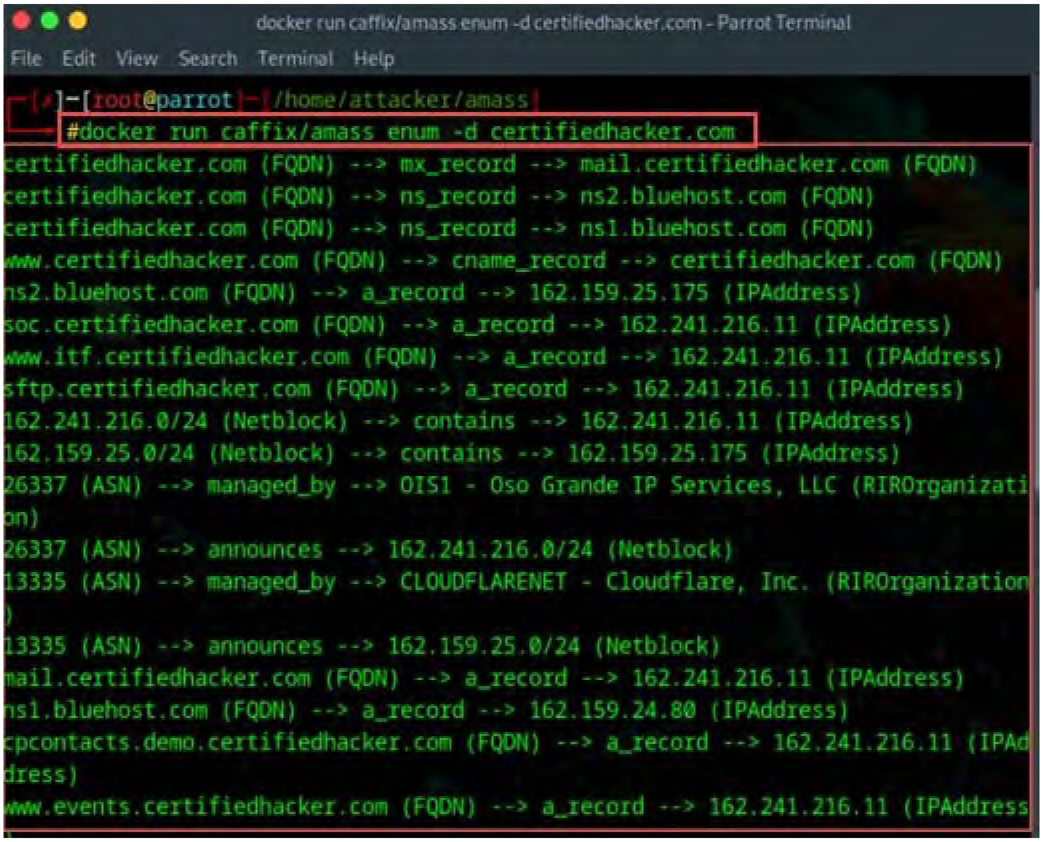
DNS Enumeration using OWASP Amass

OWASP Amass is a tool for DNS enumeration that enables attackers to outline the target network and identify possible attack surfaces. Utilizing a mix of active and passive reconnaissance methods, attackers collect information through DNS. This tool allows attackers to gather crucial data without activating any security alerts in the DNS infrastructure of the targeted network.

Attackers can run the following command to gather DNS information from the target network:

```
amass enum -d <Target Domain>
```

The command mentioned above enables attackers to collect comprehensive information about DNS, along with its subdomains. With this command, they can also enumerate details regarding IP addresses, SSL/TLS configurations, HTTP, APIs, certificates, web archives, and data scraping related to the target domain.


docker run caffix/amass enum -d certifiedhacker.com - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[✗]─[root@parrot]─[/home/attacker/amass]
└──╼ #docker run caffix/amass enum -d certifiedhacker.com
ertifiedhacker.com (FQDN) --> mx_record --> mail.certifiedhacker.com (FQDN)
ertifiedhacker.com (FQDN) --> ns_record --> ns2.bluehost.com (FQDN)
ertifiedhacker.com (FQDN) --> ns_record --> ns1.bluehost.com (FQDN)
ww.certifiedhacker.com (FQDN) --> cname_record --> certifiedhacker.com (FQDN)
s2.bluehost.com (FQDN) --> a_record --> 162.159.25.175 (IPAddress)
oc.certifiedhacker.com (FQDN) --> a_record --> 162.241.216.11 (IPAddress)
ww.itf.certifiedhacker.com (FQDN) --> a_record --> 162.241.216.11 (IPAddress)
ftp.certifiedhacker.com (FQDN) --> a_record --> 162.241.216.11 (IPAddress)
162.241.216.0/24 (Netblock) --> contains --> 162.241.216.11 (IPAddress)
162.159.25.0/24 (Netblock) --> contains --> 162.159.25.175 (IPAddress)
26337 (ASN) --> managed_by --> OIS1 - Oso Grande IP Services, LLC (RIROrganizati
on)
26337 (ASN) --> announces --> 162.241.216.0/24 (Netblock)
13335 (ASN) --> managed_by --> CLOUDFLARENET - Cloudflare, Inc. (RIROrganization
)
13335 (ASN) --> announces --> 162.159.25.0/24 (Netblock)
ail.certifiedhacker.com (FQDN) --> a_record --> 162.241.216.11 (IPAddress)
s1.bluehost.com (FQDN) --> a_record --> 162.159.24.80 (IPAddress)
pcontacts.demo.certifiedhacker.com (FQDN) --> a_record --> 162.241.216.11 (IPAd
dress)
ww.events.certifiedhacker.com (FQDN) --> a_record --> 162.241.216.11 (IPAddress
```

Figure 4-49: Screenshot of OWASP Amass

Other OWASP Amass commands for DNS Enumeration:

Execute the following command to carry out passive enumeration:

```
amass enum -passive -d <Target Domain> -src
```

Execute the following command to conduct active enumeration through brute-forcing with a designated wordlist:

```
amass enum -active -d <Target Domain> /usr/share/wordlists/amass/all.txt
```

Execute the following command to monitor or compare the last two enumeration scans conducted on the target domain:

```
amass track -config /root/amass/config.ini -dir amass4owasp -d <Target Domain> -last 2
```

Execute the following command to view the enumeration results saved in the amass database (amass4owasp):

```
amass db -dir amass4owasp -list
```

Execute the following command to generate a d3-force HTML visual graph:

```
amass viz -d3 -dir amass4owasp
```

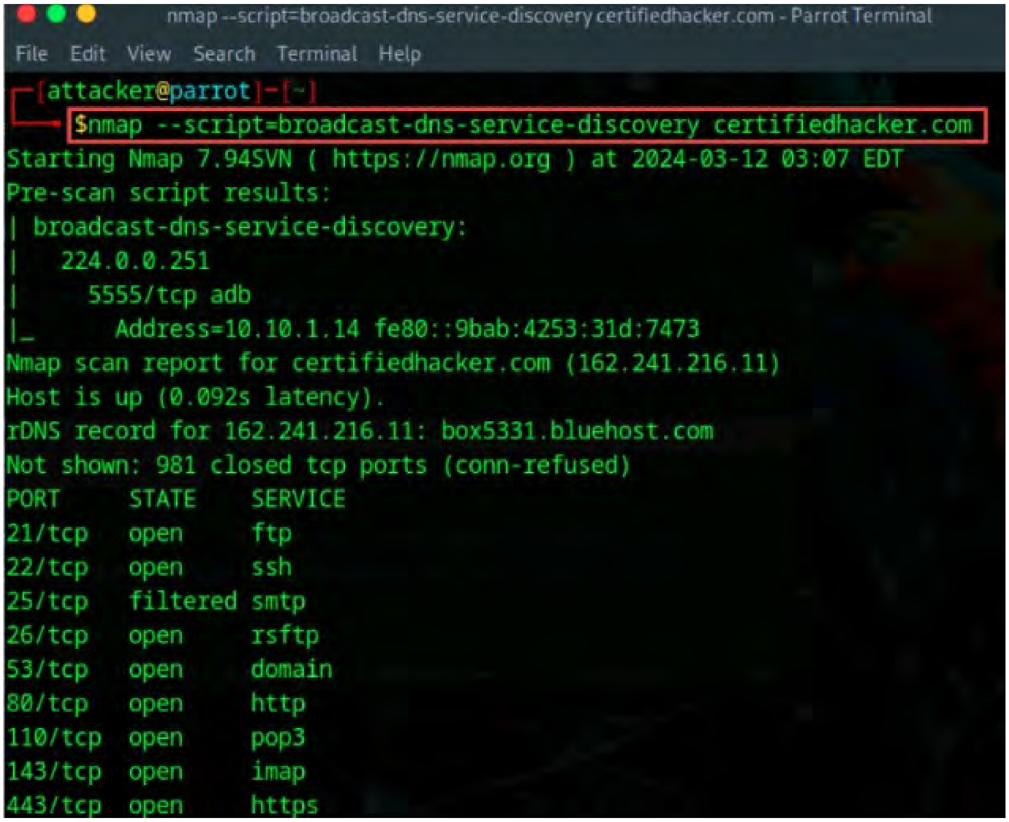
DNS and DNSSEC Enumeration using Nmap

DNS Enumeration Using Nmap

Attackers utilize Nmap to explore domains and gather a comprehensive list of subdomains, records, IP addresses, and other useful data from the targeted host.

1. Execute the command below to display all the services present on the target host:

```
nmap --script=broadcast-dns-service-discovery <Target Domain>
```



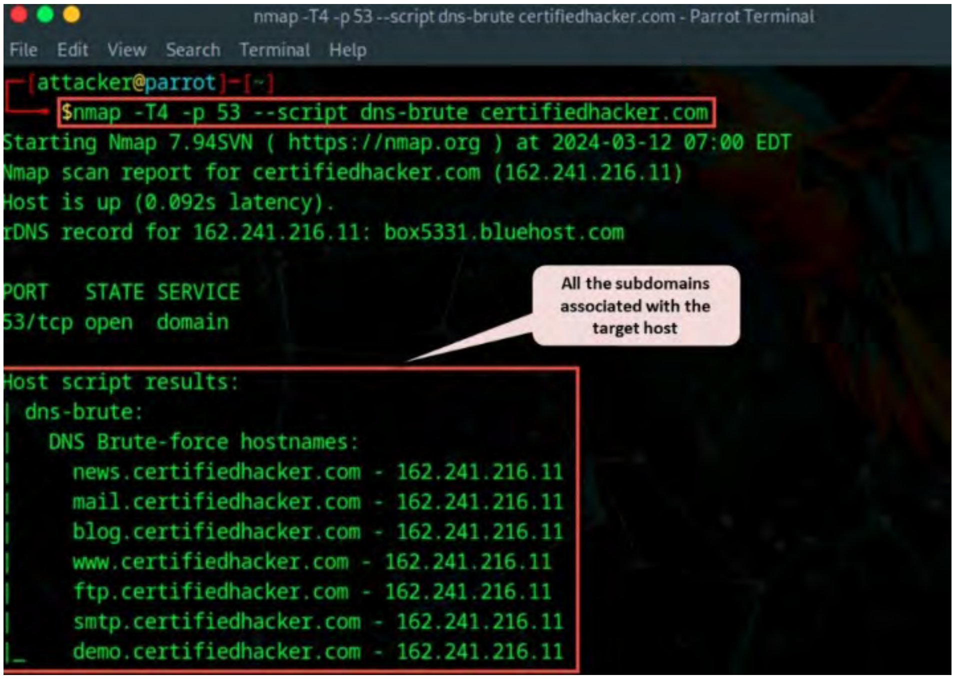
nmap --script=broadcast-dns-service-discovery certifiedhacker.com - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $nmap --script=broadcast-dns-service-discovery certifiedhacker.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-12 03:07 EDT
Pre-scan script results:
| broadcast-dns-service-discovery:
|   224.0.0.251
|     5555/tcp adb
|_      Address=10.10.1.14 fe80::9bab:4253:31d:7473
Nmap scan report for certifiedhacker.com (162.241.216.11)
Host is up (0.092s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com
Not shown: 981 closed tcp ports (conn-refused)
PORT     STATE    SERVICE
21/tcp   open     ftp
22/tcp   open     ssh
25/tcp   filtered smtp
26/tcp   open     rsftp
53/tcp   open     domain
80/tcp   open     http
110/tcp  open     pop3
143/tcp  open     imap
443/tcp  open     https
```

Figure 4-50: Screenshot of Nmap DNS Service Discovery

2. Execute the following command to obtain all subdomains linked to the target host:

```
nmap -T4 -p 53 --script dns-brute <Target Domain>
```


nmap -T4 -p 53 --script dns-brute certifiedhacker.com - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $nmap -T4 -p 53 --script dns-brute certifiedhacker.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-12 07:00 EDT
Nmap scan report for certifiedhacker.com (162.241.216.11)
Host is up (0.092s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com

PORT   STATE SERVICE
53/tcp open  domain

Host script results:
| dns-brute:
|   DNS Brute-force hostnames:
|     news.certifiedhacker.com - 162.241.216.11
|     mail.certifiedhacker.com - 162.241.216.11
|     blog.certifiedhacker.com - 162.241.216.11
|     www.certifiedhacker.com - 162.241.216.11
|     ftp.certifiedhacker.com - 162.241.216.11
|     smtp.certifiedhacker.com - 162.241.216.11
|_    demo.certifiedhacker.com - 162.241.216.11
```

All the subdomains associated with the target host

Figure 4-51: Screenshot of the dns-brute NSE script

The command above returns a list of subdomains paired with their corresponding IP addresses. Wildcard entries, if present, will appear as ***A*** for IPv4 addresses and ***AAAA*** for IPv6 addresses.

3. To verify if DNS recursion is active on the target server, execute the following command:

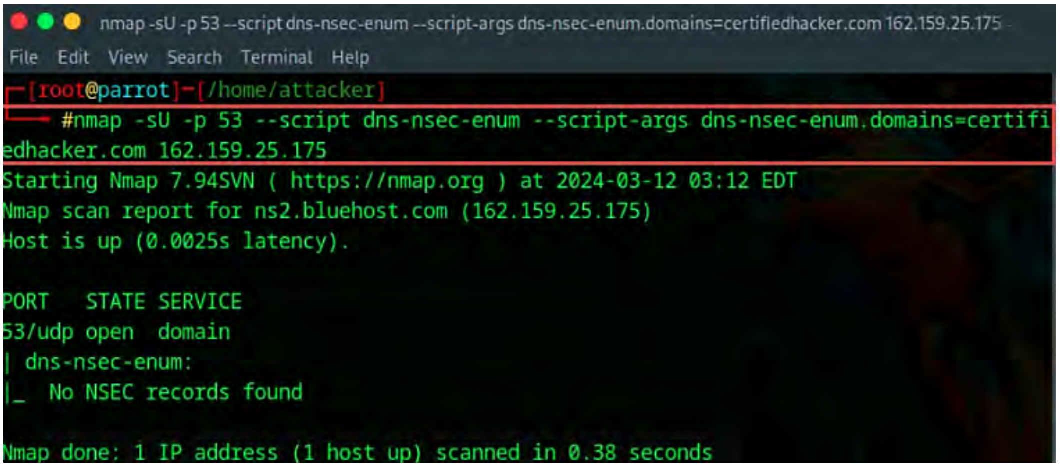
```
nmap -Pn -sU -p 53 --script=dns-recursion 192.168.1.150
```

DNS Security Extensions (DNSSEC) Enumeration using Nmap

DNSSEC offers protection for DNS queries and responses. Attackers can use the dns-nsec-enum.nse or dns-nsec3-enum.nse NSE scripts to gather details about domains and their subdomains.

1. To obtain a list of subdomains linked to the target domain, execute the following command:

```
nmap -sU -p 53 --script dns-nsec-enum --script-args dns-nsec-enum.domains=eccouncil.org <target>
```



```
nmap -sU -p 53 --script dns-nsec-enum --script-args dns-nsec-enum.domains=certifiedhacker.com 162.159.25.175
File  Edit  View  Search  Terminal  Help
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sU -p 53 --script dns-nsec-enum --script-args dns-nsec-enum.domains=certifi
edhacker.com 162.159.25.175
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-12 03:12 EDT
Nmap scan report for ns2.bluehost.com (162.159.25.175)
Host is up (0.0025s latency).

PORT   STATE SERVICE
53/udp open  domain
| dns-nsec-enum:
|_  No NSEC records found

Nmap done: 1 IP address (1 host up) scanned in 0.38 seconds
```

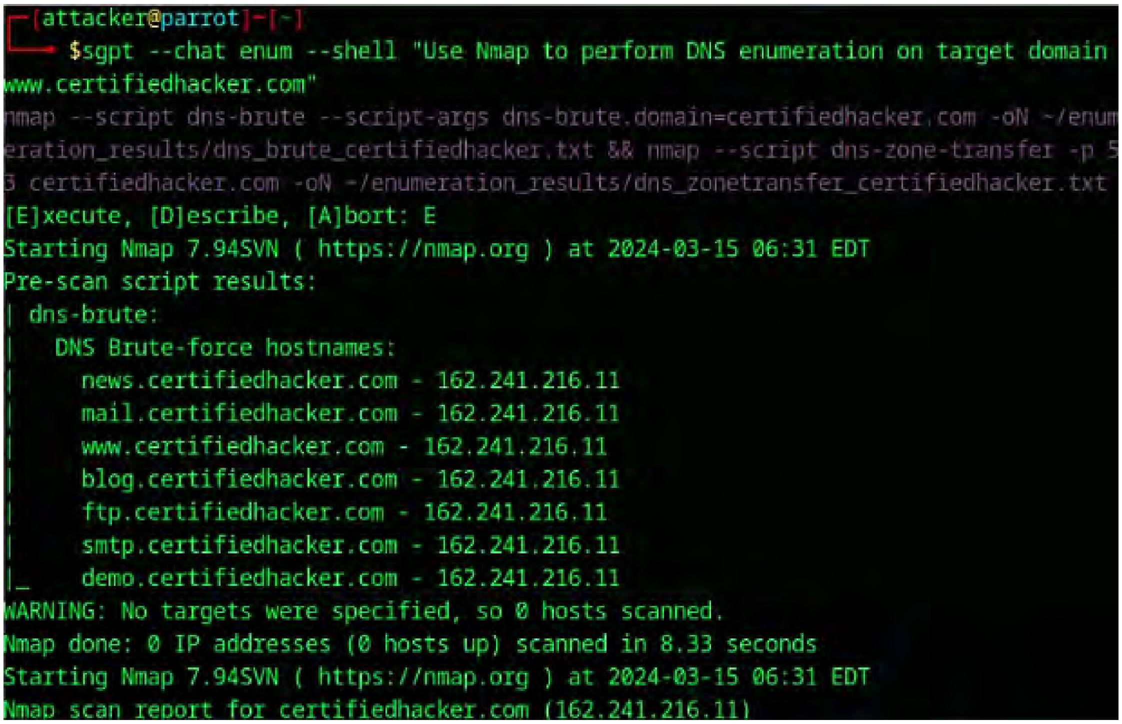
Figure 4-52: Screenshot of Nmap dns-nsec-enum NSE Script

DNS Enumeration with Nmap using AI

Attackers can utilize AI-driven tools to improve and automate their tasks related to network enumeration. By employing AI, these attackers can easily conduct DNS enumeration and collect DNS data on targeted domains.

They can utilize ChatGPT for this purpose by using suitable prompts such as:

> Use Nmap to perform DNS Enumeration on target domain www.certifiedhacker.com.

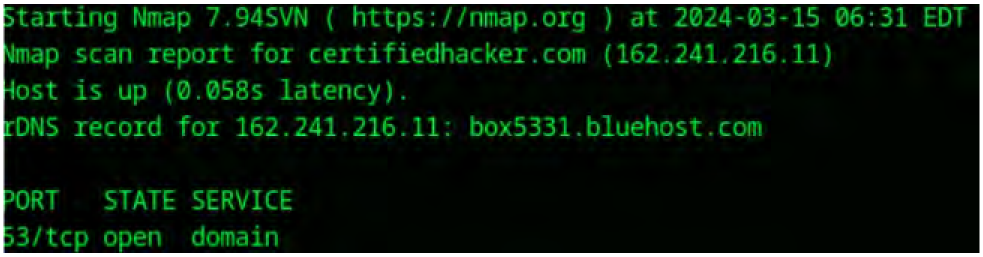


```
┌─[attacker@parrot]─[~]
└──╼ $sgpt --chat enum --shell "Use Nmap to perform DNS enumeration on target domain
www.certifiedhacker.com"
nmap --script dns-brute --script-args dns-brute.domain=certifiedhacker.com -oN ~/enum
eration_results/dns_brute_certifiedhacker.txt && nmap --script dns-zone-transfer -p 5
3 certifiedhacker.com -oN ~/enumeration_results/dns_zonetransfer_certifiedhacker.txt
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 06:31 EDT
Pre-scan script results:
| dns-brute:
|   DNS Brute-force hostnames:
|     news.certifiedhacker.com - 162.241.216.11
|     mail.certifiedhacker.com - 162.241.216.11
|     www.certifiedhacker.com - 162.241.216.11
|     blog.certifiedhacker.com - 162.241.216.11
|     ftp.certifiedhacker.com - 162.241.216.11
|     smtp.certifiedhacker.com - 162.241.216.11
|_    demo.certifiedhacker.com - 162.241.216.11
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 8.33 seconds
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 06:31 EDT
Nmap scan report for certifiedhacker.com (162.241.216.11)
```

Figure 4-53: Perform DNS Enumeration on Target Domain with Nmap

The following command is designed to automate DNS enumeration tasks on the specified target domain:

```
nmap --script dns-brute --script-args dns-brute.domain=certifiedhacker.com -oN ~/enumeration_results/dns_brute_certifiedhacker.txt && nmap --script dns-zone-transfer -p 53 certifiedhacker.com –oN ~/enumeration_results/dns_zonetransfer_certifiedhacker.txt
```



```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 06:31 EDT
Nmap scan report for certifiedhacker.com (162.241.216.11)
Host is up (0.058s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com

PORT   STATE SERVICE
53/tcp open  domain
```

Figure 4-54: Perform DNS Enumeration on Target Domain with nmap

The first command executes nmap with the dns-brute script to carry out DNS brute-force enumeration on the domain certifiedhacker.com. The findings are stored in the file dns_brute_certifiedhacker.txt located in the ~/enumeration_results/ directory.

The second command utilizes nmap with the dns-zone-transfer script to conduct DNS zone transfer enumeration on port 53 of the domain certifiedhacker.com. The results are preserved in the file dns_zonetransfer_certifiedhacker.txt within the ~/enumeration_results/ directory.

This command automates tasks related to DNS enumeration and records the outcomes for every command run on the designated target domain www.certifiedhacker.com.

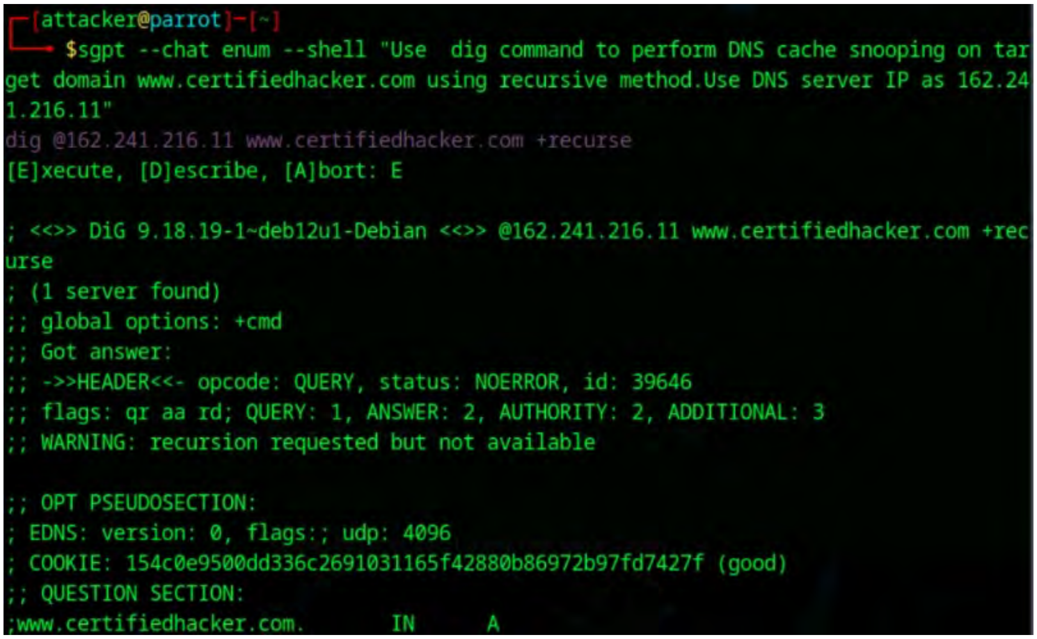
DNS Cache Snooping using AI

Attackers can utilize AI-driven technologies to improve and automate their tasks related to network enumeration. By using AI, attackers can easily carry out DNS cache snooping on the targeted domain.

Example 1:

Attackers can utilize ChatGPT to perform this task by using appropriate prompts such as:

```
Use dig command to perform DNS cache snooping on target domain www.certifiedhacker.com using recursive method. Use DNS server IP as 162.241.216.11.
```



```
┌─[attacker@parrot]─[~]
└──╼ $sgpt --chat enum --shell "Use  dig command to perform DNS cache snooping on tar
get domain www.certifiedhacker.com using recursive method.Use DNS server IP as 162.24
1.216.11"
dig @162.241.216.11 www.certifiedhacker.com +recurse
[E]xecute, [D]escribe, [A]bort: E

; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> @162.241.216.11 www.certifiedhacker.com +rec
urse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39646
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 154c0e9500dd336c2691031165f42880b86972b97fd7427f (good)
;; QUESTION SECTION:
;www.certifiedhacker.com.       IN      A
```

Figure 4-55: Perform DNS Cache Snooping on Target Domain with dig

The command below is intended to automate the process of DNS cache snooping for the specified target domain by utilizing the dig command:

```
dig @162.241.216.11 www.certifiedhacker.com +recurse
```

Table 4-11 describes each option used in the command.

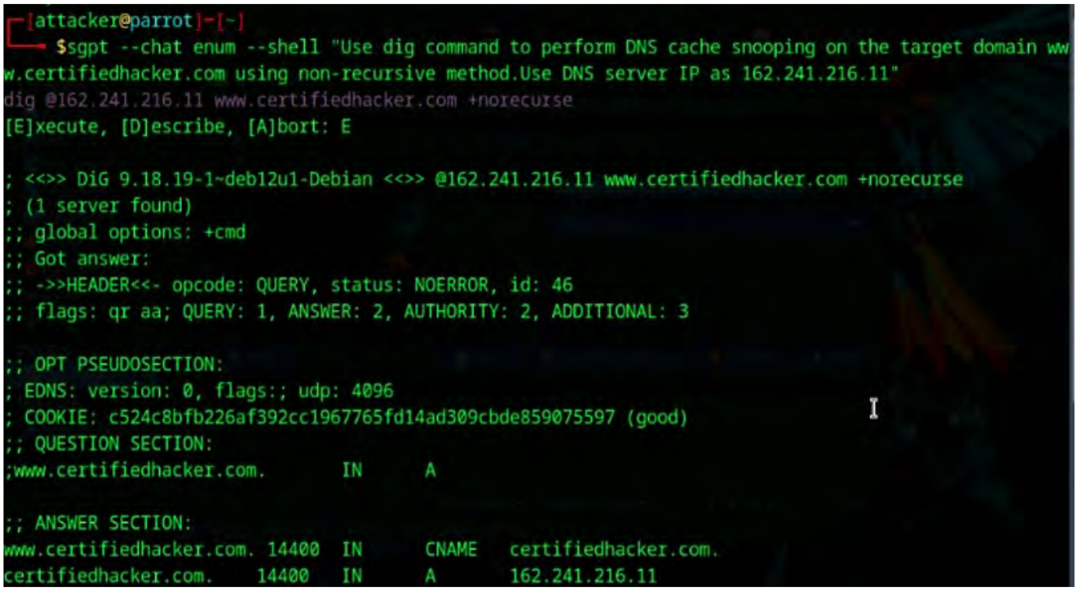
Command Breakdown	Description
dig	Conduct DNS cache snooping on the specified domain (www.certifiedhacker.com).
@162.241.216.11	Indicates the IP address of the DNS server to be queried.
www.certifiedhacker.com	Designates the domain being targeted for the query.
+recurse	Signifies that recursive DNS resolution will be utilized for the query.

Table 4-11: DNS Cache Snooping (Recursive Method) Command Description

This command streamlines the process of DNS cache snooping and presents the findings for the designated target domain www.certifiedhacker.com using the recursive approach.

Example 2:

Similarly, attackers can use the non-recursive approach to automate DNS cache snooping on the designated target domain.



```
┌─[attacker@parrot]─[~]
└──╼ $sgpt --chat enum --shell "Use dig command to perform DNS cache snooping on the target domain ww
w.certifiedhacker.com using non-recursive method.Use DNS server IP as 162.241.216.11"
dig @162.241.216.11 www.certifiedhacker.com +norecurse
[E]xecute, [D]escribe, [A]bort: E

; <<>> DiG 9.18.19-1~deb12u1-Debian <<>> @162.241.216.11 www.certifiedhacker.com +norecurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: c524c8bfb226af392cc1967765fd14ad309cbde859075597 (good)
;; QUESTION SECTION:
;www.certifiedhacker.com.       IN      A

;; ANSWER SECTION:
www.certifiedhacker.com. 14400  IN      CNAME   certifiedhacker.com.
certifiedhacker.com.    14400   IN      A       162.241.216.11
```

Figure 4-56: Perform DNS Cache Snooping on Target Domain with dig – using Non-Recursive Method

Other Enumeration Techniques

This section covers IPsec, VoIP, RPC, Unix/Linux users, Telnet, SSH users, and SMB enumeration.

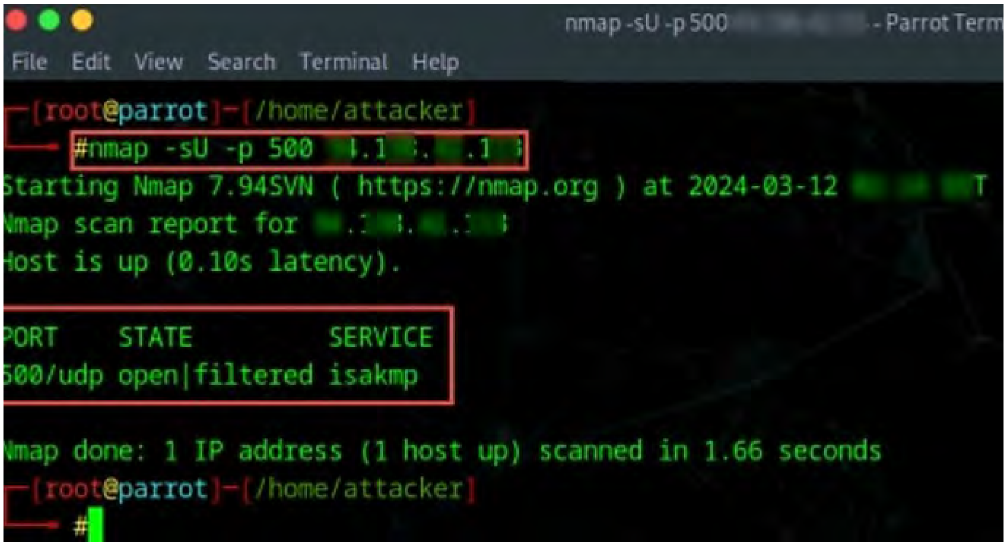
IPsec Enumeration

IPsec is the most widely used technology for both gateway-to-gateway (LAN-to-LAN) and host-to-gateway (remote access) enterprise VPN implementations. IPsec ensures data protection by utilizing various components like Encapsulating Security Payload (ESP), Authentication Header (AH), and Internet Key Exchange (IKE) to secure communications between VPN endpoints.

The majority of IPsec-based VPNs rely on the Internet Security Association Key Management Protocol (ISAKMP), which is part of IKE, to create, negotiate, modify, and remove Security Associations (SA) and cryptographic keys within a VPN setting. Attackers can conduct straightforward direct scanning for ISAKMP at UDP port 500 using tools like Nmap to obtain details regarding the existence of a VPN gateway.

Execute the following command to perform an Nmap scan to check the status of ISAKMP over port 500:

```
# nmap –sU –p 500 <target IP address>
```



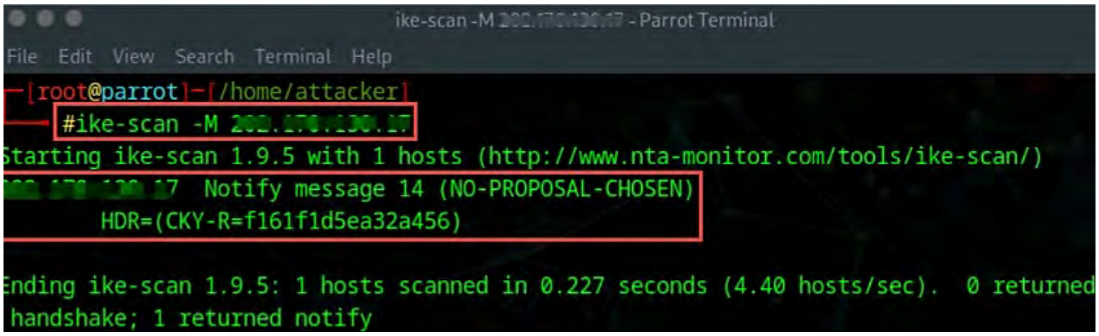
```
nmap -sU -p 500 [redacted] - Parrot Term
File  Edit  View  Search  Terminal  Help
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sU -p 500 [redacted]
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-12 [redacted]
Nmap scan report for [redacted]
Host is up (0.10s latency).

PORT     STATE          SERVICE
500/udp open|filtered isakmp

Nmap done: 1 IP address (1 host up) scanned in 1.66 seconds
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

Figure 4-57: Screenshot displaying an Nmap Scan Over Port 500 for ISAKMP

Attackers can use fingerprinting tools like ike-scan to gather sensitive details, which includes the encryption and hashing algorithm, authentication type, key distribution algorithm, and SA LifeDuration. During this scanning process, specifically designed IKE packets containing an ISAKMP header are sent to the target gateway, and the responses are recorded.

The command used for the initial discovery of IPsec VPN with the ike-scan tool is:

```
# ike-scan –M <target gateway IP address>
```



```
ike-scan -M [redacted] - Parrot Terminal
File  Edit  View  Search  Terminal  Help
┌─[root@parrot]─[/home/attacker]
└──╼ #ike-scan -M [redacted]
Starting ike-scan 1.9.5 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
[redacted]  Notify message 14 (NO-PROPOSAL-CHOSEN)
        HDR=(CKY-R=f161f1d5ea32a456)

Ending ike-scan 1.9.5: 1 hosts scanned in 0.227 seconds (4.40 hosts/sec).  0 returned
 handshake; 1 returned notify
```

Figure 4-58: Screenshot displaying ike-scan Enumeration

ike-scan

ike-scan identifies IKE hosts and can analyze them based on their retransmission backoff patterns. It can execute several functions, including:

- **Discovery:** It can reveal which hosts in a specific IP range are running IKE by showing the hosts that reply to the IKE requests sent by ike-scan.
- **Fingerprinting:** It can identify the IKE implementation utilized by the hosts and, in certain instances, the version of the software they are operating. This is achieved through two methods:

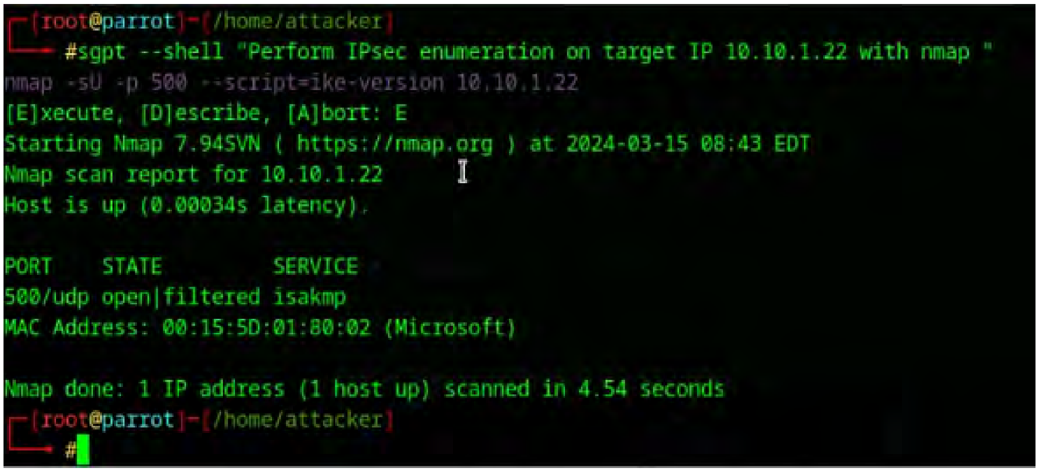
    - UDP backoff fingerprinting, where the response packet arrival times from the target hosts are logged, and the observed retransmission backoff pattern is matched against known patterns.
    - Vendor ID fingerprinting, which compares the Vendor ID payloads from the VPN servers to established Vendor ID patterns.
- **Transform Enumeration:** It can ascertain the transform attributes that the VPN server supports for IKE phase 1 (such as encryption and hash algorithms).
- **User Enumeration:** For some VPN systems, it can reveal valid VPN usernames.
- **Pre-Shared Key Cracking:** It is capable of performing offline dictionary or brute-force password cracking for IKE Aggressive Mode with pre-shared key authentication. This process utilizes ike-scan to gather the hash and other parameters alongside the psk-crack, which is included in the ike-scan package, to carry out the cracking.

IPsec Enumeration with AI

Attackers can utilize AI-driven technologies to improve and automate their network enumeration activities. With the help of AI, they can easily carry out IPsec enumeration on the targeted domain.

For instance, an attacker might employ ChatGPT to complete this task by utilizing a suitable prompt, such as:

```
Perform IPsec enumeration on target IP 10.10.1.22 with nmap.
```



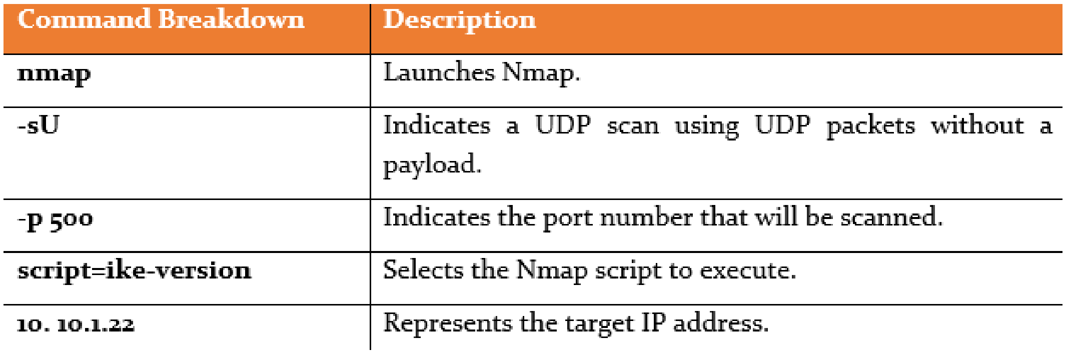
```
┌─[root@parrot]─[/home/attacker]
└──╼ #sgpt --shell "Perform IPsec enumeration on target IP 10.10.1.22 with nmap "
nmap -sU -p 500 --script=ike-version 10.10.1.22
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 08:43 EDT
Nmap scan report for 10.10.1.22
Host is up (0.00034s latency).

PORT    STATE         SERVICE
500/udp open|filtered isakmp
MAC Address: 00:15:5D:01:80:02 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 4.54 seconds
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

Figure 4-59: Perform IPsec Enumeration on Target IP

To carry out an Internet Protocol Security (IPsec) enumeration on the specified IP address 10.10.1.22 using Nmap, you can execute the following command:

```
nmap -sU -p 500 –script=ike-version 10.10.1.22
```

Table 4-12 describes each option used in the command.



Command Breakdown	Description
nmap	Launches Nmap.
-sU	Indicates a UDP scan using UDP packets without a payload.
-p 500	Indicates the port number that will be scanned.
script=ike-version	Selects the Nmap script to execute.
10. 10.1.22	Represents the target IP address.

Table 4-12: IPSec Enumeration Command Description

This command instructs Nmap to carry out a UDP scan on port 500 and to execute the ike-version script against the specified IP address 10.10.1.22 to identify the IKE version in use, which can offer valuable insights into the IPsec configuration.

VoIP Enumeration

VoIP represents a modern technology that has supplanted the traditional Public Switched Telephone Network (PSTN) in both business and residential settings. VoIP utilizes Internet infrastructure to facilitate connections for voice communications, while also transmitting data over the same network. Nevertheless, VoIP is susceptible to various TCP/IP attack methods. The Session Initiation Protocol (SIP) is one of the protocols employed by VoIP for enabling voice and video calls over an IP network. Typically, this SIP service operates on UDP/TCP ports 2000, 2001, 5060, and 5061.

Attackers utilize tools such as Svmap and Metasploit to carry out VoIP enumeration. By performing VoIP enumeration, attackers can collect sensitive details like VoIP gateways/servers, IP-Private Branch Exchange (PBX) systems, as well as User-Agent IP addresses and user extensions corresponding to client software (softphones) or VoIP phones. This data can be exploited to execute various VoIP attacks, including Denial of Service (DoS) attacks, session hijacking, caller ID spoofing, eavesdropping, spam over Internet telephony (SPIT), and VoIP phishing (Vishing).

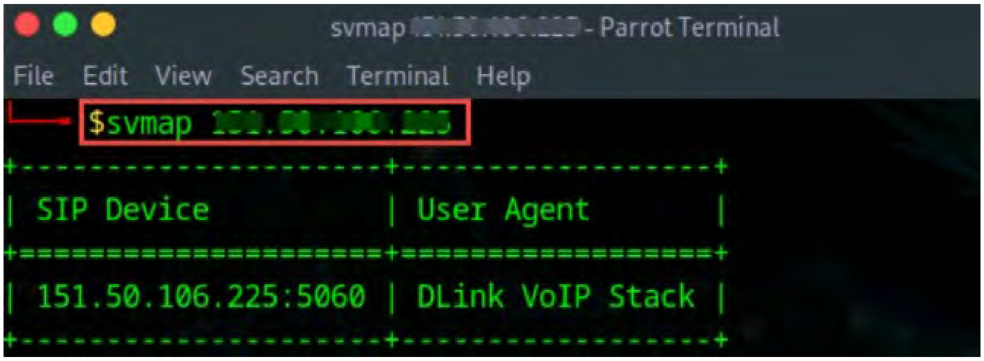
Svmap

Svmap is a publicly available scanner that locates SIP devices and PBX servers within a specified network. It serves as a useful tool for system administrators conducting network inventories. Malicious users exploit Svmap for several purposes:

- Detect SIP devices and PBX servers on both default and alternative ports.
- Scan extensive network ranges.
- Examine a single host across various ports for SIP services or assess multiple hosts on numerous ports.
- Ring all phones on a network at once utilizing the INVITE method.

Figure 4-60 illustrates an example of extracting SIP device information with the Svmap tool by executing the command:

```
# svmap <target network range/IP Address>
```

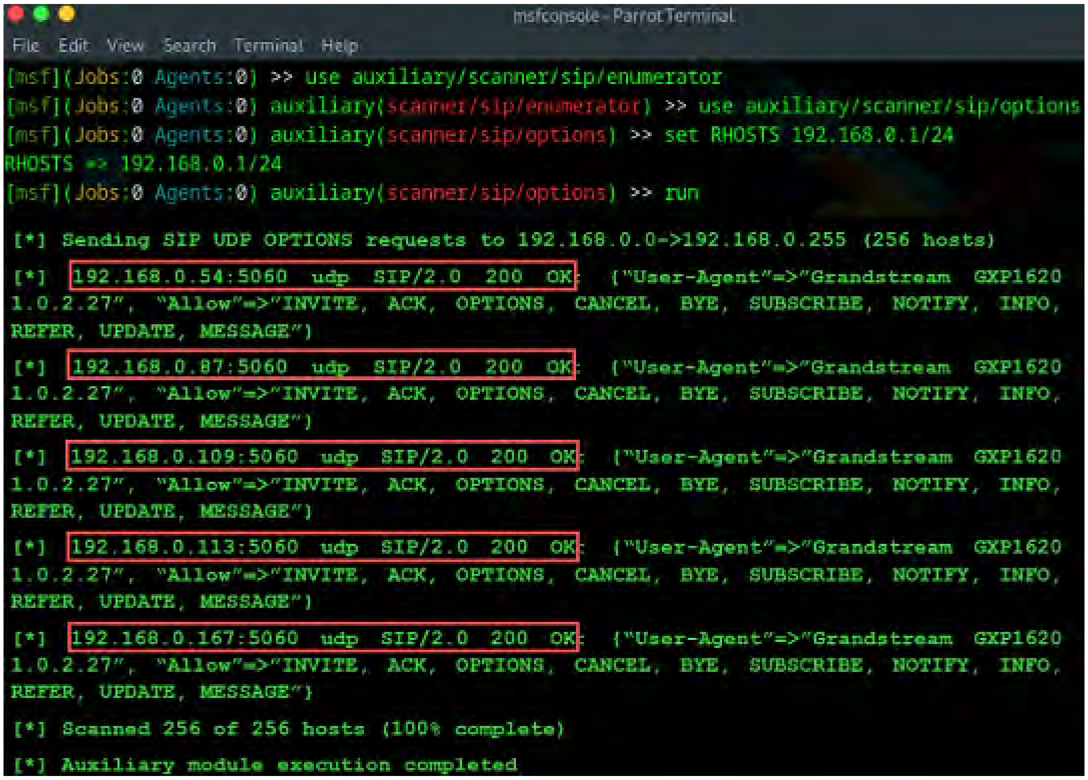


```
svmap [redacted] - Parrot Terminal
$svmap [redacted]
+----------------------+------------------+
| SIP Device           | User Agent       |
+======================+==================+
| 151.50.106.225:5060 | DLink VoIP Stack |
+----------------------+------------------+
```

Figure 4-60: Screenshot displaying Svmap Scan for Enumerating SIP Details

Metasploit

Attackers utilize Metasploit's SIP Username Enumerator to probe numeric usernames/extensions of VoIP phones. Figure 4-61 illustrates an example of how to enumerate SIP using Metasploit.



```
msfconsole - Parrot Terminal
[msf](Jobs:0 Agents:0) >> use auxiliary/scanner/sip/enumerator
[msf](Jobs:0 Agents:0) auxiliary(scanner/sip/enumerator) >> use auxiliary/scanner/sip/options
[msf](Jobs:0 Agents:0) auxiliary(scanner/sip/options) >> set RHOSTS 192.168.0.1/24
RHOSTS => 192.168.0.1/24
[msf](Jobs:0 Agents:0) auxiliary(scanner/sip/options) >> run

[*] Sending SIP UDP OPTIONS requests to 192.168.0.0->192.168.0.255 (256 hosts)
[*] 192.168.0.54:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620
1.0.2.27", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO,
REFER, UPDATE, MESSAGE"}
[*] 192.168.0.87:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620
1.0.2.27", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO,
REFER, UPDATE, MESSAGE"}
[*] 192.168.0.109:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620
1.0.2.27", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO,
REFER, UPDATE, MESSAGE"}
[*] 192.168.0.113:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620
1.0.2.27", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO,
REFER, UPDATE, MESSAGE"}
[*] 192.168.0.167:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620
1.0.2.27", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO,
REFER, UPDATE, MESSAGE"}
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 4-61: Screenshot displaying Metasploit exploit for SIP Enumeration

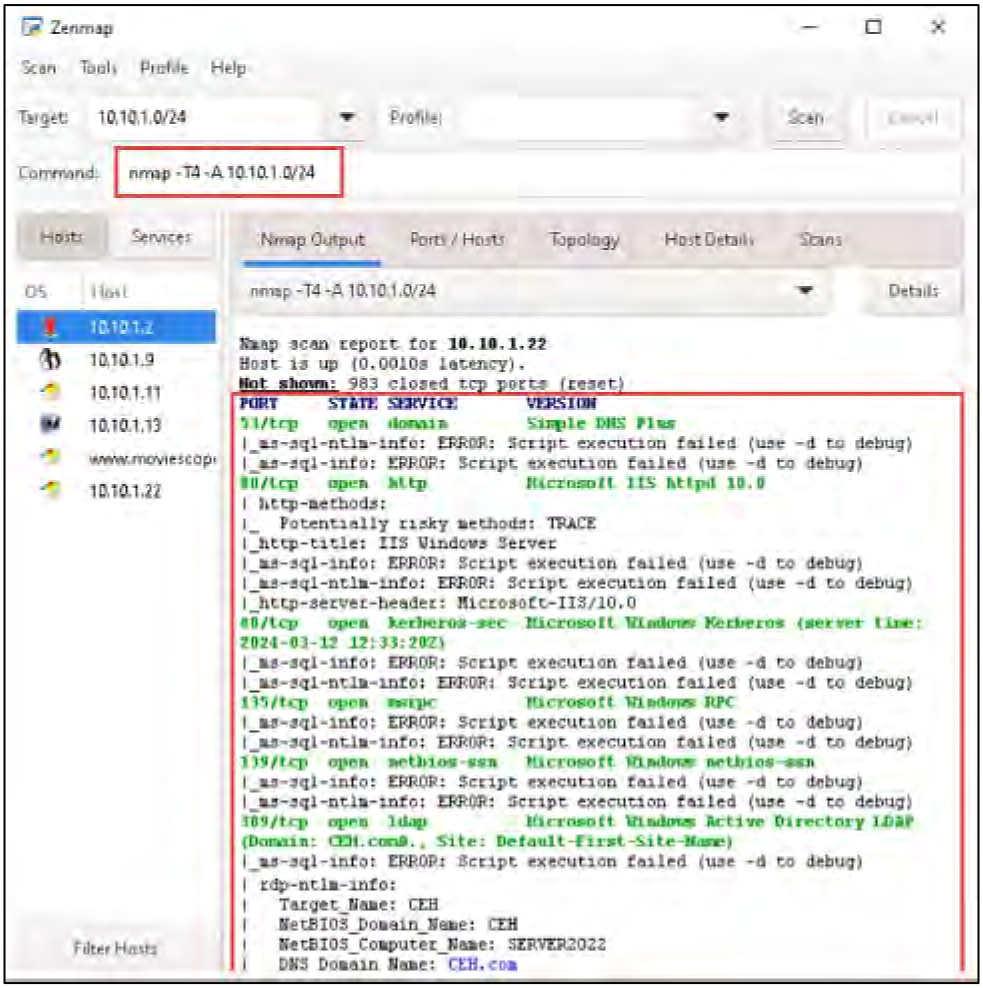
RPC Enumeration

Remote Procedure Call (RPC) is a technology utilized for developing distributed client/server applications. RPC facilitates communication between clients and servers in distributed client/server setups. It serves as an inter-process communication mechanism that allows for data transfer between various processes. Generally, RPC includes components such as a client, a server, an endpoint, an endpoint mapper, a client stub, and a server stub, along with several dependencies.

The portmapper service operates on TCP and UDP port 111 to detect endpoints and provide clients with information about the active RPC services. Identifying RPC endpoints can help attackers locate any vulnerable services running on these ports. In environments secured by firewalls and other protective measures, this portmapper service is frequently filtered. As a result, attackers often scan large ranges of ports to find accessible RPC services that may be susceptible to direct exploitation.

Attackers employ the following Nmap scanning commands to detect the RPC services operating within the network:

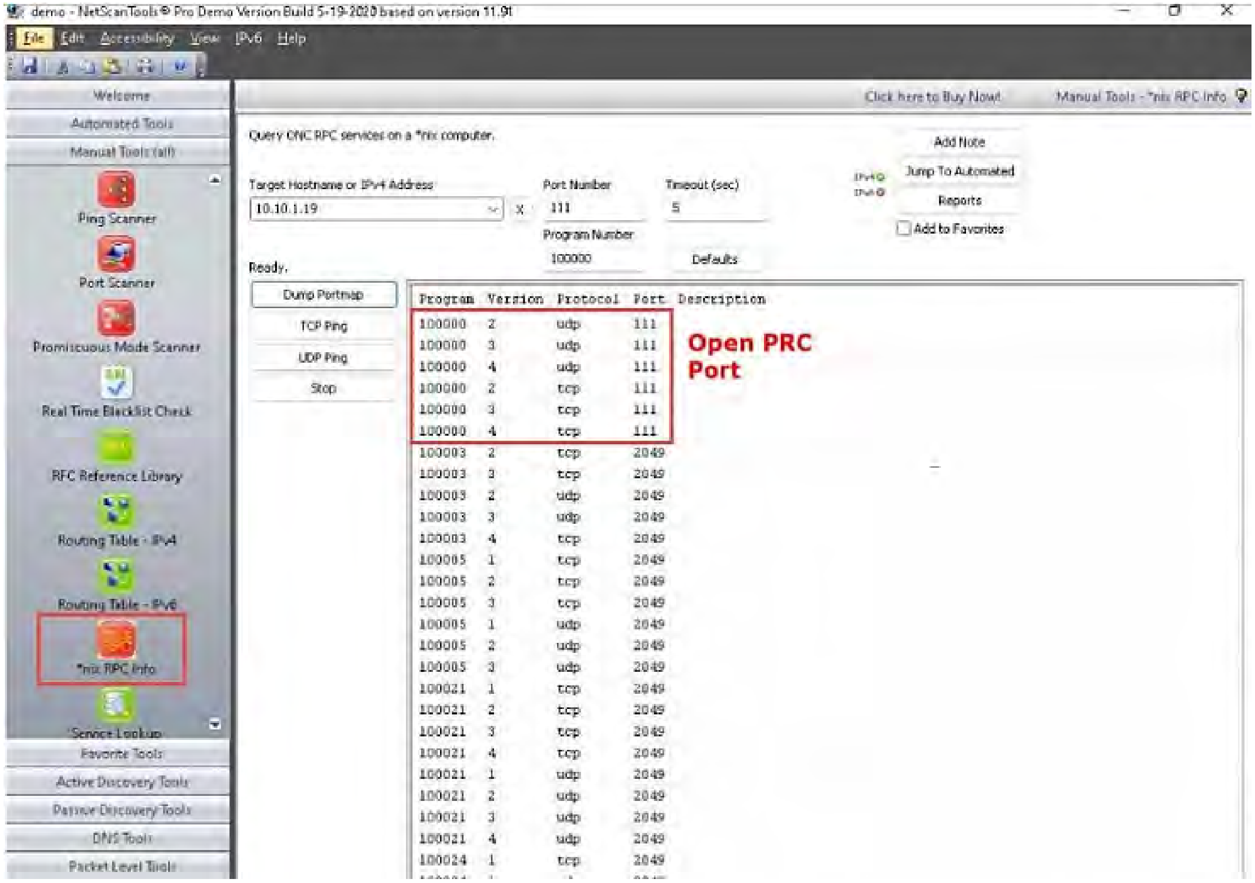
```
# nmap -sR <target IP/network>
# nmap -T4 –A <target IP/network>
```



```
Nmap scan report for 10.10.1.22
Host is up (0.0010s latency).
Not shown: 983 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
|_ms-sql-ntlm-info: ERROR: Script execution failed (use -d to debug)
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
80/tcp   open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: IIS Windows Server
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
|_ms-sql-ntlm-info: ERROR: Script execution failed (use -d to debug)
|_http-server-header: Microsoft-IIS/10.0
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time:
2024-03-12 12:33:20Z)
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
|_ms-sql-ntlm-info: ERROR: Script execution failed (use -d to debug)
135/tcp  open  msrpc         Microsoft Windows RPC
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
|_ms-sql-ntlm-info: ERROR: Script execution failed (use -d to debug)
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
|_ms-sql-ntlm-info: ERROR: Script execution failed (use -d to debug)
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP
(Domain: CEH.com0., Site: Default-First-Site-Name)
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
| rdp-ntlm-info:
|   Target_Name: CEH
|   NetBIOS_Domain_Name: CEH
|   NetBIOS_Computer_Name: SERVER2022
|   DNS_Domain_Name: CEH.com
```

Figure 4-62: Screenshot displaying an Nmap Scan Result for RPC Enumeration

Furthermore, attackers employ utilities like NetScanTools Pro to gather the RPC details of the targeted network. The RPC Info feature of NetScanTools Pro assists attackers in identifying and reaching the portmapper daemon/service commonly operating on port 111 of Unix or Linux systems.



Query ONC RPC services on a *nix computer.

Target Hostname or IPv4 Address: 10.10.1.19

Port Number: 111

Timeout (sec): 5

Program Number: 100000

Ready.

```
Program  Version  Protocol  Port  Description
100000   2        udp       111
100000   3        udp       111
100000   4        udp       111
100000   2        tcp       111
100000   3        tcp       111
100000   4        tcp       111
100003   2        tcp       2049
100003   3        tcp       2049
100003   2        udp       2049
100003   3        udp       2049
100003   4        tcp       2049
100005   1        tcp       2049
100005   2        tcp       2049
100005   3        tcp       2049
100005   1        udp       2049
100005   2        udp       2049
100005   3        udp       2049
100021   1        tcp       2049
100021   2        tcp       2049
100021   3        tcp       2049
100021   4        tcp       2049
100021   1        udp       2049
100021   2        udp       2049
100021   3        udp       2049
100021   4        udp       2049
100024   1        tcp       2049
```

Open PRC Port

Figure 4-63: Screenshot displaying NetScanTools Pro Tool for RPC Enumeration

Unix/Linux User Enumeration

A crucial step in enumeration is conducting user enumeration on Unix/Linux systems. This process yields a list of users, including information such as usernames, hostnames, and the date and time when each session began.

The command-line tools listed below can facilitate user enumeration on Unix/Linux.

rusers

The command rusers provides a list of users currently logged into remote computers or those within the local network. Its output resembles that of the who command but pertains to machines on the local network.

Its syntax is as follows:

```
/usr/bin/rusers [-a] [-l] [-u| -h| -i] [Host ...]
```

Table 4-13 describes each option used in the command.

Command Breakdown	Description
-a	Offers a report for a machine even if there are no logged-in users.
-h	Arranges the output alphabetically by hostname.
-l	Displays an extended listing similar to the who command.
-u	Sorts the list by the number of users.
-i	Sorts based on idle time.

Table 4-13: rusers Command Description

rwho

rwho provides a listing of users currently logged into computers on the local network. Its output resembles that of the who command and includes details about the username, hostname, and the date and time when each session started for all machines on the local network that have the rwho daemon running.

The syntax for using it is as follows:

```
rwho [ -a]
```

Here, **-a** incorporates all users; if this flag is omitted, users whose sessions have been idle for an hour or longer will not be reflected in the report.

finger

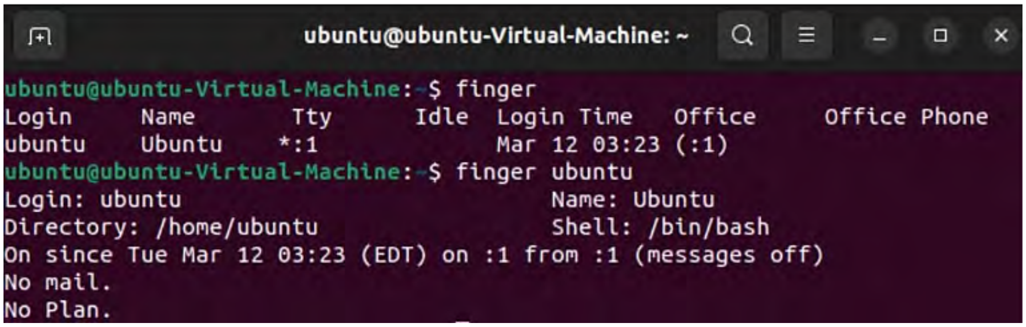
The finger command provides information about system users, including the user's login name, real name, terminal name, idle time, login time, office location, and office phone numbers. Its syntax is as follows:

```
finger [-l] [-m] [-p] [-s] [user ...] [user@host ...]
```

Table 4-14 describes each option used in the command.

Command Breakdown	Description
-s	Shows the user's login name, real name, terminal name, idle time, login time, office location, and office phone number.
-l	Generates a multi-line output that displays all the details provided for the **-s** option.
-p	Stops the **-l** option of finger from showing the contents of the files.
-m	Stops the matching of usernames.

Table 4-14: finger Command Description



```
ubuntu@ubuntu-Virtual-Machine:~$ finger
Login     Name       Tty      Idle  Login Time   Office     Office Phone
ubuntu    Ubuntu    *:1             Mar 12 03:23 (:1)
ubuntu@ubuntu-Virtual-Machine:~$ finger ubuntu
Login: ubuntu                           Name: Ubuntu
Directory: /home/ubuntu                 Shell: /bin/bash
On since Tue Mar 12 03:23 (EDT) on :1 from :1 (messages off)
No mail.
No Plan.
```

Figure 4-64: Screenshot displaying the Execution of the finger Command for User Enumeration

SMB Enumeration

The Server Message Block (SMB) is a protocol used for transporting data, primarily utilized by Windows systems to allow shared access to files, printers, serial ports, and remote access to Windows services. Typically, SMB operates on TCP port 445 or through the NetBIOS API using UDP ports 137 and 138, as well as TCP ports 137 and 139. By leveraging the SMB service, users can access and manage files and data located on a remote server. Additionally, the SMB service enables application users to read, write, and alter files stored on the remote server. Networks utilizing this service are particularly susceptible to SMB enumeration, which can reveal significant information about the target system.

In SMB enumeration, attackers often engage in banner grabbing to gather details such as operating system specifics and the versions of services in operation. With this knowledge, they can execute various types of attacks, including SMB relay attacks and brute-force attacks. Tools for SMB enumeration, like Nmap, SMBMap, enum4linux, nullinux, SMBeagle, and NetScanTool Pro, allow attackers to carry out targeted scans on the SMB service that is active on port 445.

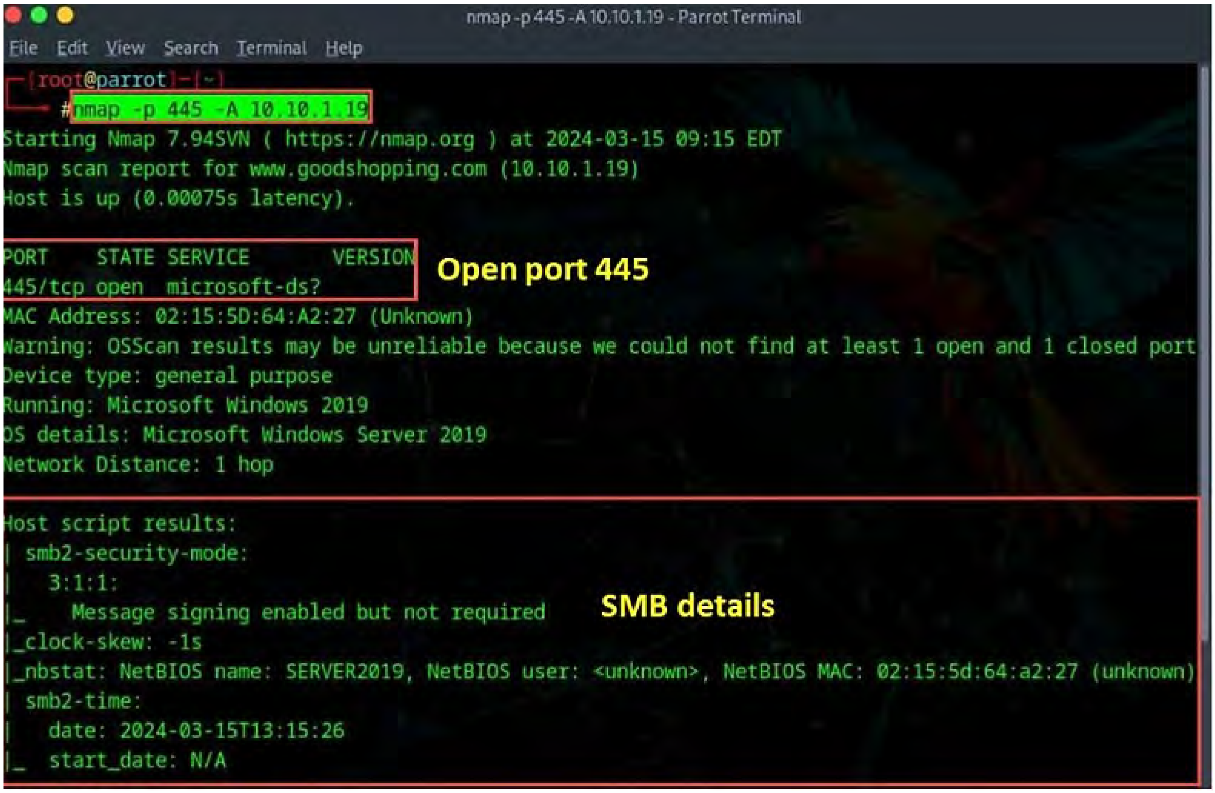
For example, as illustrated in Figure 4-65, attackers employ the following Nmap command to enumerate the SMB service on the specified target IP address:

```
nmap -p 445 -A <target IP>
```

Table 4-15 describes each option used in the command.

Command Breakdown	Description
-p	Indicates a specific port to be scanned (445 in this example).
-A	Detect the operating system, identify the version, scan scripts, and gather traceroute details.

Table 4-15: SMB Enumeration Command Description



nmap -p 445 -A 10.10.1.19 - Parrot Terminal
File Edit View Search Terminal Help
Open port 445
SMB details

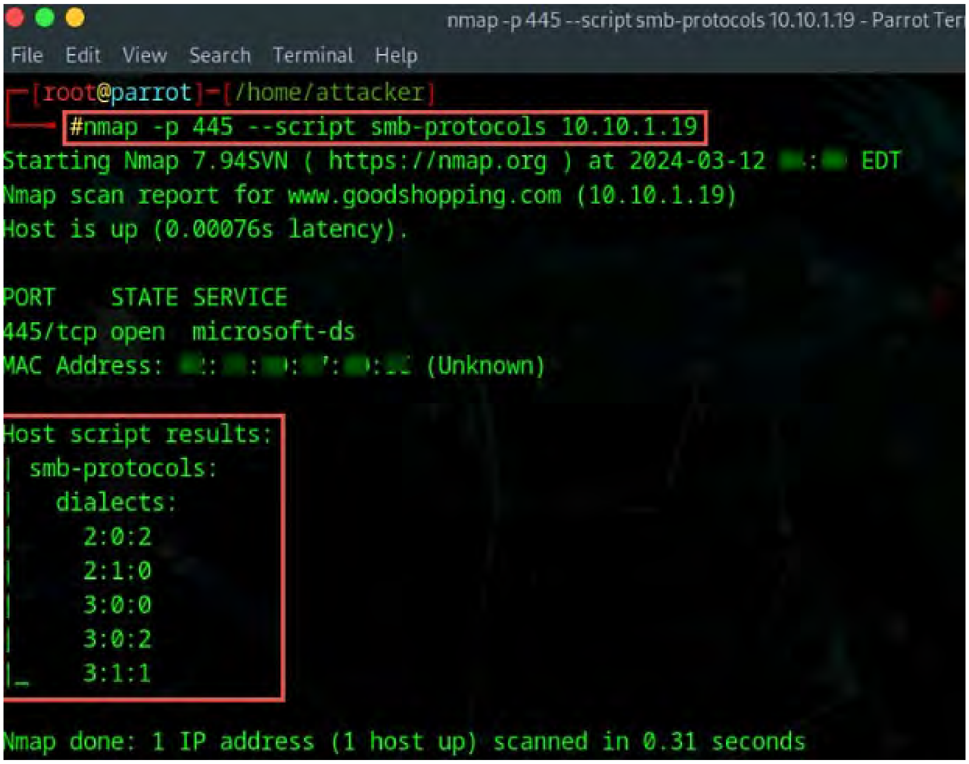
```
┌─[root@parrot]─[~]
└──╼ #nmap -p 445 -A 10.10.1.19
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 09:15 EDT
Nmap scan report for www.goodshopping.com (10.10.1.19)
Host is up (0.00075s latency).

PORT    STATE SERVICE       VERSION
445/tcp open  microsoft-ds?
MAC Address: 02:15:5D:64:A2:27 (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Microsoft Windows 2019
OS details: Microsoft Windows Server 2019
Network Distance: 1 hop

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_clock-skew: -1s
|_nbstat: NetBIOS name: SERVER2019, NetBIOS user: <unknown>, NetBIOS MAC: 02:15:5d:64:a2:27 (unknown)
| smb2-time:
|   date: 2024-03-15T13:15:26
|_  start_date: N/A
```

Figure 4-65: Screenshot of Nmap performing SMB Enumeration

The **STATE** of **PORT 445/tcp** is **OPEN**, indicating that the port is accessible and the SMB service is operational. With this command, attackers can also extract information regarding the operating system and perform a traceroute to the specified target.

As illustrated in Figure 4-66, attackers utilize the following Nmap commands to identify the supported protocols and versions of the target SMB server:

```
# nmap -p 445 –-script smb-protocols <Target IP>
# nmap -p 139 –-script smb-protocols <Target IP>
```


nmap -p 445 --script smb-protocols 10.10.1.19 - Parrot Ter
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -p 445 --script smb-protocols 10.10.1.19
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-12 [redacted]:[redacted] EDT
Nmap scan report for www.goodshopping.com (10.10.1.19)
Host is up (0.00076s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: [redacted] (Unknown)

Host script results:
| smb-protocols:
|   dialects:
|     2:0:2
|     2:1:0
|     3:0:0
|     3:0:2
|_    3:1:1

Nmap done: 1 IP address (1 host up) scanned in 0.31 seconds
```

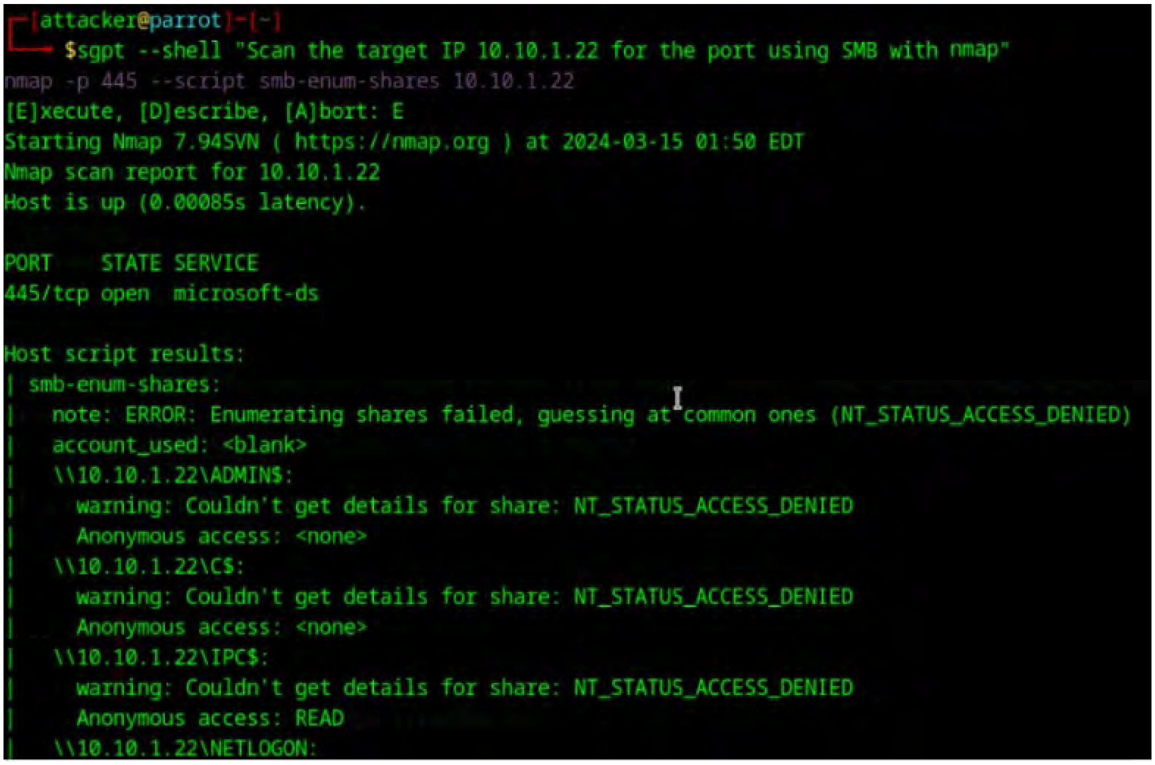
Figure 4-66: Screenshot of Nmap performing SMB Enumeration

SMB Enumeration with AI

Attackers can utilize AI-driven technologies to improve and streamline their network enumeration activities. By employing AI, these individuals can easily conduct SMB enumeration on specific domains.

For instance, an attacker might use ChatGPT to carry out this operation by employing a suitable prompt like:

```
Scan the target IP 10.10.1.22 for the port using SMB with nmap.
```



```
┌─[attacker@parrot]─[~]
└──╼ $sgpt --shell "Scan the target IP 10.10.1.22 for the port using SMB with nmap"
nmap -p 445 --script smb-enum-shares 10.10.1.22
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 01:50 EDT
Nmap scan report for 10.10.1.22
Host is up (0.00085s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-enum-shares:
|   note: ERROR: Enumerating shares failed, guessing at common ones (NT_STATUS_ACCESS_DENIED)
|   account_used: <blank>
|   \\10.10.1.22\ADMIN$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: <none>
|   \\10.10.1.22\C$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: <none>
|   \\10.10.1.22\IPC$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: READ
|   \\10.10.1.22\NETLOGON:
```

Figure 4-67: Scan the Target IP for the Port using SMB with nmap

To check a target IP for accessible SMB ports using Nmap, you can utilize the following command:

```
nmap -p 445 --script smb-enum-shares 10.10.1.22
```

Table 4-16 describes each option used in the command.

Command Breakdown	Description
nmap	Initiate Nmap.
-p 445	Indicates the specific port number to examine (Port 445 is typically linked with the Server Message Block (SMB) protocol).
--script smb-enum-shares	Indicates the Nmap script to execute (smb-enum-shares script is designed to list shares available on SMB servers).
10.10.1.22	Target IP Address.

Table 4-16: SMB Enumeration Command Description

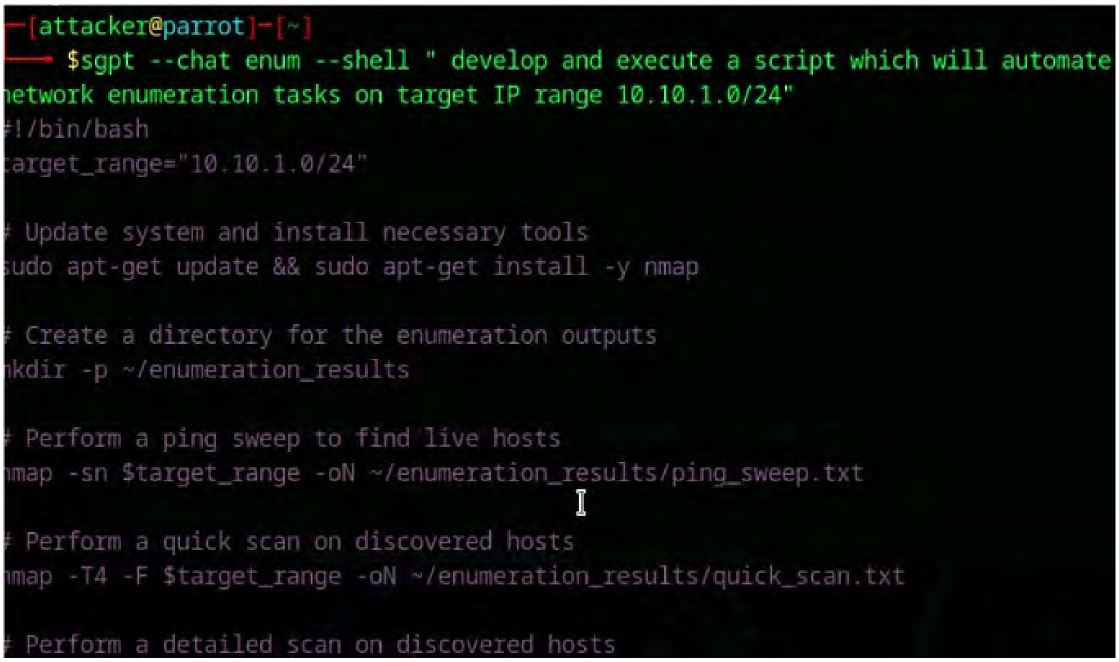
This command directs Nmap to probe the designated target IP for port 445 (SMB) and to apply the smb-enum-shares script to list the accessible shares on SMB servers.

Create and Run Custom Script to Automate Network Enumeration Tasks with AI

Attackers can utilize AI-driven technologies to improve and automate their network enumeration processes. With the support of AI, attackers can easily design and execute tailored network enumeration scripts to gain crucial insights about their objectives. By creating these personalized scripts, attackers can effectively carry out a sequence of network enumeration commands to collect information on a targeted domain.

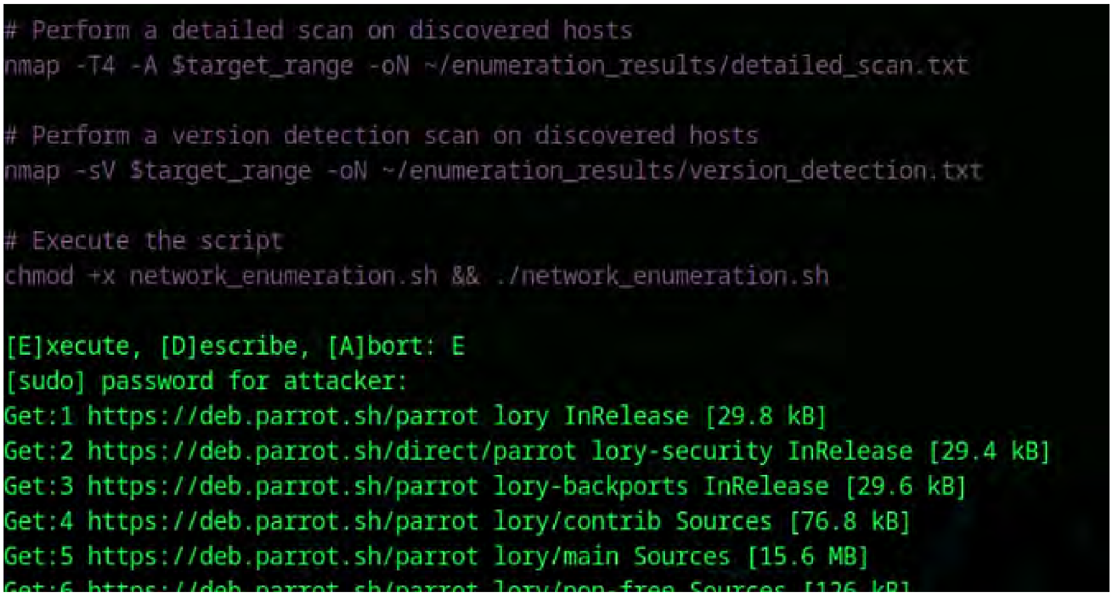
For instance, an attacker might leverage ChatGPT to accomplish this task by employing a suitable prompt such as:

Develop and execute a script that will automate network enumeration tasks on target IP range 10.10.1.0/24.



```
─[attacker@parrot]─[~]
──╼ $sgpt --chat enum --shell " develop and execute a script which will automate
etwork enumeration tasks on target IP range 10.10.1.0/24"
#!/bin/bash
arget_range="10.10.1.0/24"

 Update system and install necessary tools
udo apt-get update && sudo apt-get install -y nmap

 Create a directory for the enumeration outputs
nkdir -p ~/enumeration_results

 Perform a ping sweep to find live hosts
nmap -sn $target_range -oN ~/enumeration_results/ping_sweep.txt

 Perform a quick scan on discovered hosts
nmap -T4 -F $target_range -oN ~/enumeration_results/quick_scan.txt

 Perform a detailed scan on discovered hosts
```

Figure 4-68: Script to Automate Network Enumeration Tasks on Target IP Range-1



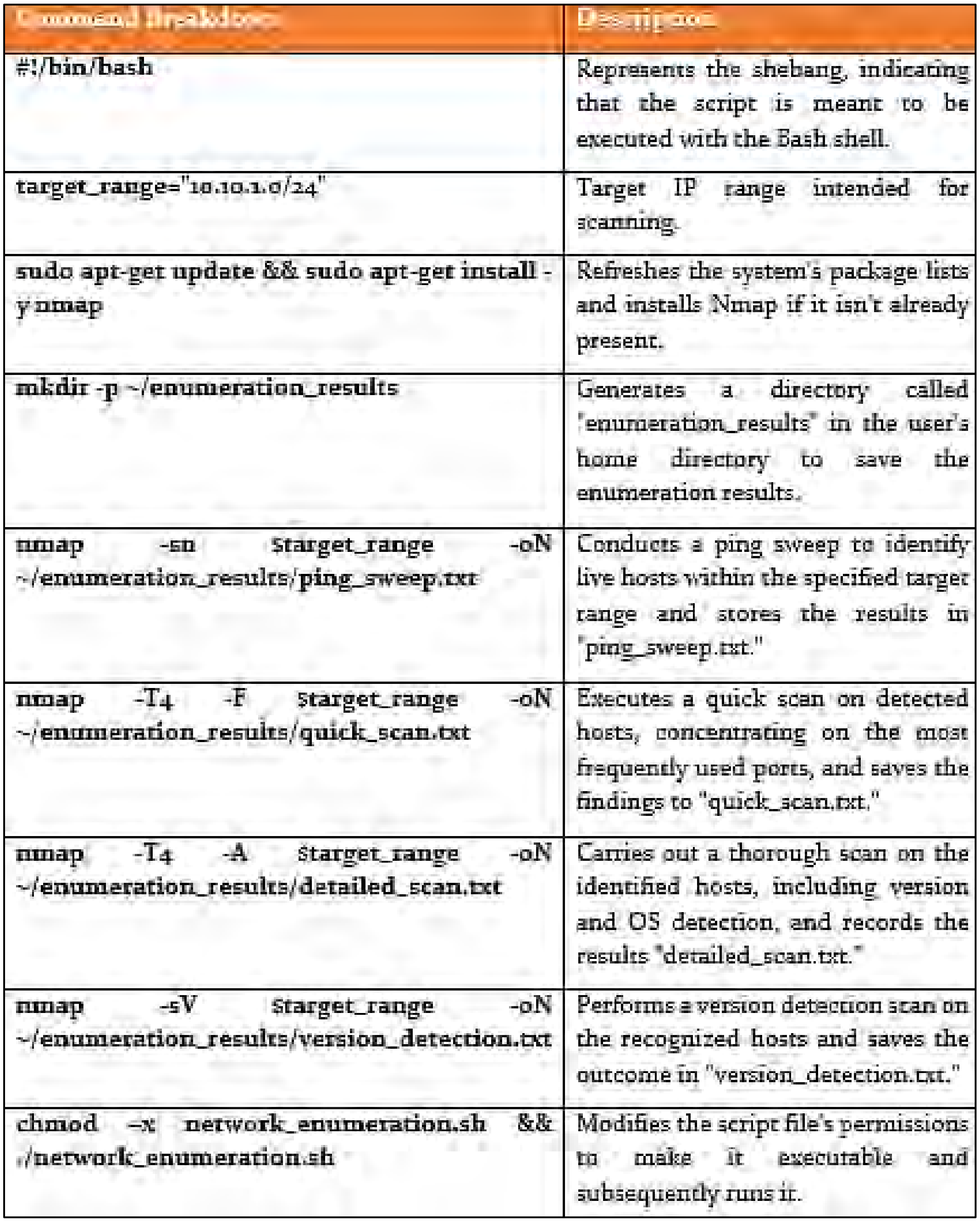
```
# Perform a detailed scan on discovered hosts
nmap -T4 -A $target_range -oN ~/enumeration_results/detailed_scan.txt

# Perform a version detection scan on discovered hosts
nmap -sV $target_range -oN ~/enumeration_results/version_detection.txt

# Execute the script
chmod +x network_enumeration.sh && ./network_enumeration.sh

[E]xecute, [D]escribe, [A]bort: E
[sudo] password for attacker:
Get:1 https://deb.parrot.sh/parrot lory InRelease [29.8 kB]
Get:2 https://deb.parrot.sh/direct/parrot lory-security InRelease [29.4 kB]
Get:3 https://deb.parrot.sh/parrot lory-backports InRelease [29.6 kB]
Get:4 https://deb.parrot.sh/parrot lory/contrib Sources [76.8 kB]
Get:5 https://deb.parrot.sh/parrot lory/main Sources [15.6 MB]
Get:6 https://deb.parrot.sh/parrot lory/non-free Sources [126 kB]
```

Figure 4-69: Script to Automate Network Enumeration Tasks on Target IP Range-2

```
#!/bin/bash
target_range="10.10.1.0/24"
#Update system and install necessary tools
sudo apt-get update && sudo apt-get install -y nmap
#Create a directory for the enumeration outputs
mkdir -p ~/enumeration_results #Perform a ping sweep to find live hosts nmap -sn $target_range
-oN ~/enumeration_results/ping_sweep.txt
#Perform a quick scan on discovered hosts
nmap -T4 -F $target_range -oN ~/enumeration_results/quick_scan.txt
#Perform a detailed scan on discovered hosts
nmap -T4 -A $target_range -oN ~/enumeration_results/detailed_scan.txt
#Perform a version detection scan on discovered hosts
nmap -sV $target_range -oN ~/enumeration_results/version_detection.txt
#Execute the script

chmod +x network_enumeration.sh && ./network_enumeration.sh
```

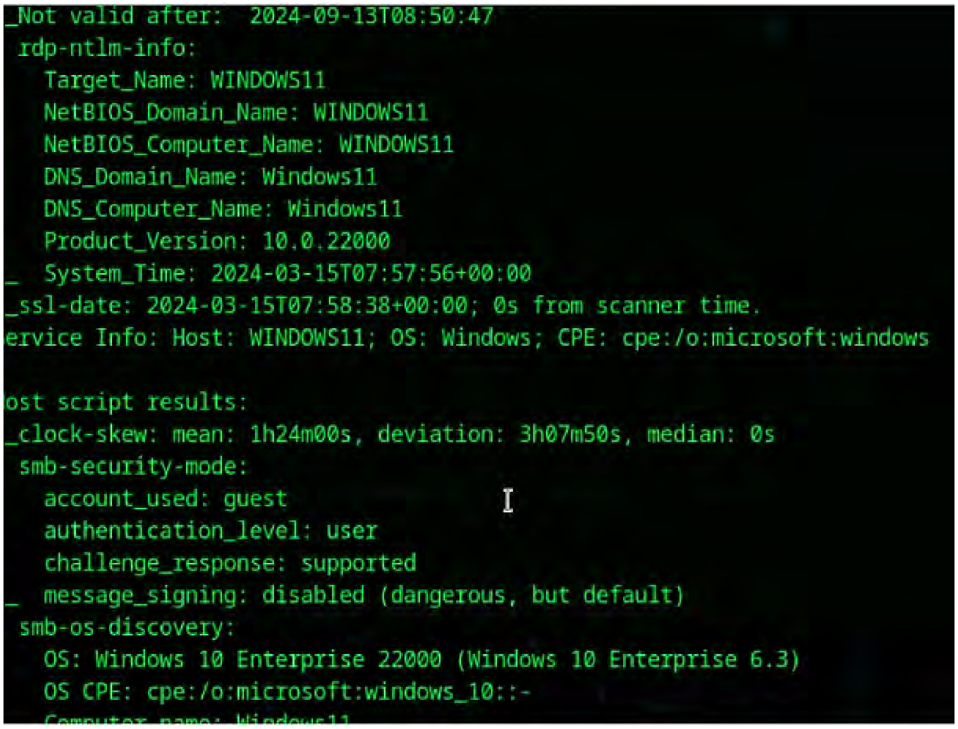
The shell script is created to conduct network enumeration activities utilizing Nmap on a designated target range.

Table 4-17 describes each option used in the script.



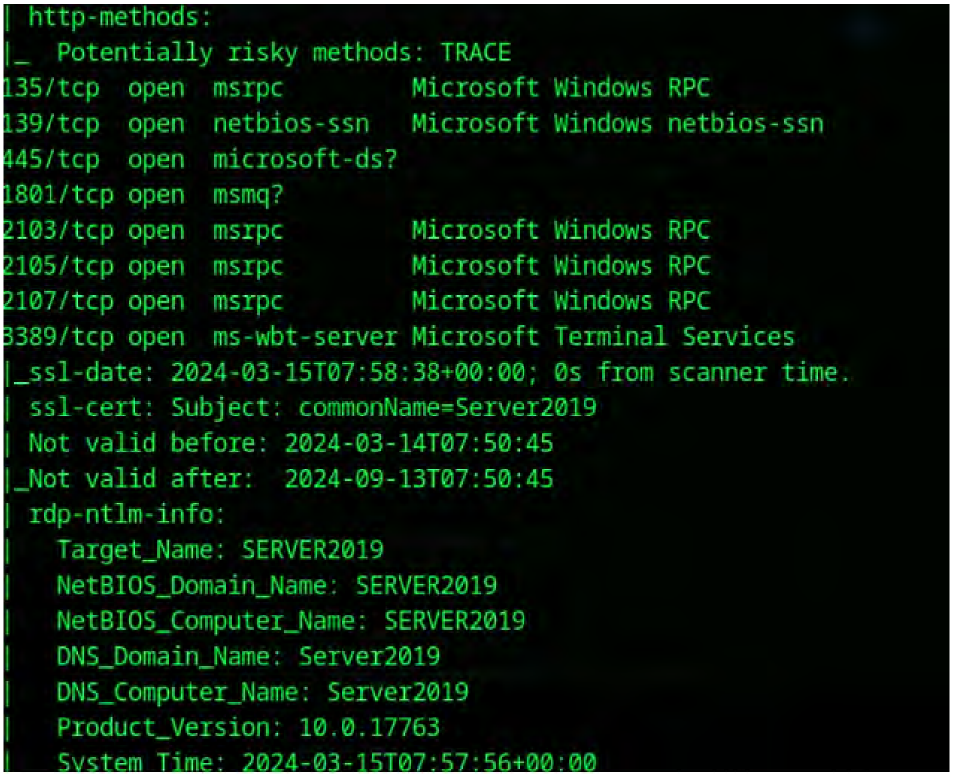
Command Breakdown	Description
#!/bin/bash	Represents the shebang, indicating that the script is meant to be executed with the Bash shell.
target_range="10.10.1.0/24"	Target IP range intended for scanning.
sudo apt-get update && sudo apt-get install -y nmap	Refreshes the system's package lists and installs Nmap if it isn't already present.
mkdir -p ~/enumeration_results	Generates a directory called "enumeration_results" in the user's home directory to save the enumeration results.
nmap -sn $target_range -oN ~/enumeration_results/ping_sweep.txt	Conducts a ping sweep to identify live hosts within the specified target range and stores the results in "ping_sweep.txt."
nmap -T4 -F $target_range -oN ~/enumeration_results/quick_scan.txt	Executes a quick scan on detected hosts, concentrating on the most frequently used ports, and saves the findings to "quick_scan.txt."
nmap -T4 -A $target_range -oN ~/enumeration_results/detailed_scan.txt	Carries out a thorough scan on the identified hosts, including version and OS detection, and records the results "detailed_scan.txt."
nmap -sV $target_range -oN ~/enumeration_results/version_detection.txt	Performs a version detection scan on the recognized hosts and saves the outcome in "version_detection.txt."
chmod –x network_enumeration.sh && ./network_enumeration.sh	Modifies the script file's permissions to make it executable and subsequently runs it.

Table 4-17: Script Commands with Description



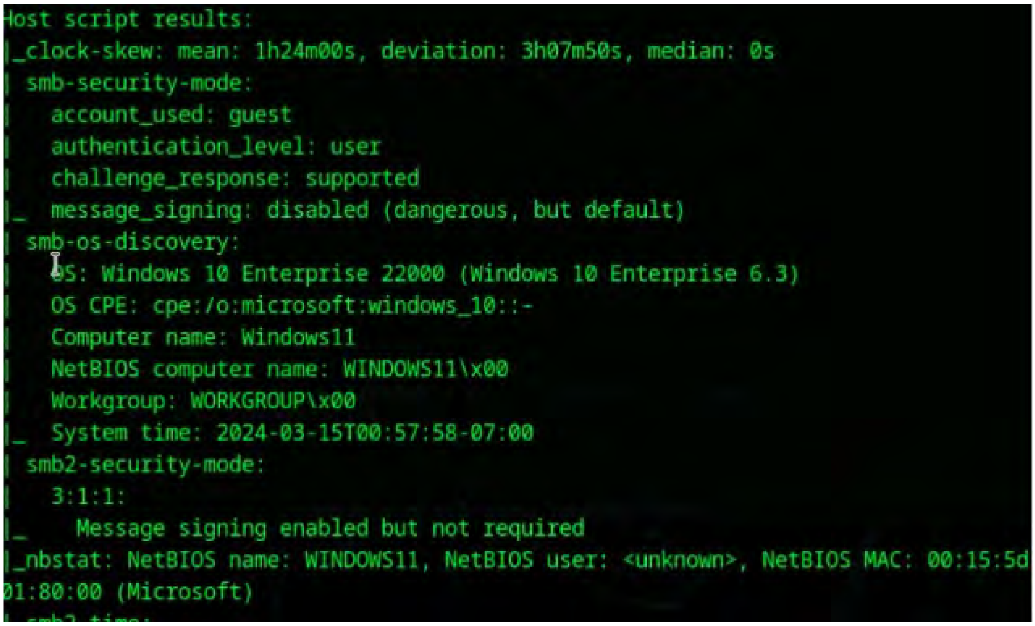
```
_Not valid after:  2024-09-13T08:50:47
 rdp-ntlm-info:
   Target_Name: WINDOWS11
   NetBIOS_Domain_Name: WINDOWS11
   NetBIOS_Computer_Name: WINDOWS11
   DNS_Domain_Name: Windows11
   DNS_Computer_Name: Windows11
   Product_Version: 10.0.22000
_  System_Time: 2024-03-15T07:57:56+00:00
_ssl-date: 2024-03-15T07:58:38+00:00; 0s from scanner time.
ervice Info: Host: WINDOWS11; OS: Windows; CPE: cpe:/o:microsoft:windows

ost script results:
_clock-skew: mean: 1h24m00s, deviation: 3h07m50s, median: 0s
 smb-security-mode:
   account_used: guest
   authentication_level: user
   challenge_response: supported
_  message_signing: disabled (dangerous, but default)
 smb-os-discovery:
   OS: Windows 10 Enterprise 22000 (Windows 10 Enterprise 6.3)
   OS CPE: cpe:/o:microsoft:windows_10::-
   Computer name: Windows11
```

Figure 4-70: Output-1



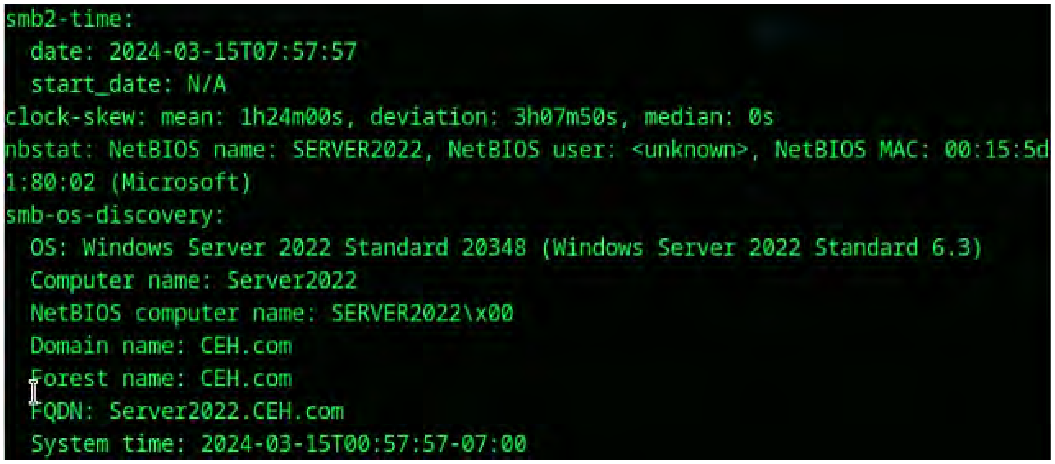
```
| http-methods:
|_  Potentially risky methods: TRACE
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
1801/tcp open  msmq?
2103/tcp open  msrpc         Microsoft Windows RPC
2105/tcp open  msrpc         Microsoft Windows RPC
2107/tcp open  msrpc         Microsoft Windows RPC
3389/tcp open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2024-03-15T07:58:38+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=Server2019
| Not valid before: 2024-03-14T07:50:45
|_Not valid after:  2024-09-13T07:50:45
| rdp-ntlm-info:
|   Target_Name: SERVER2019
|   NetBIOS_Domain_Name: SERVER2019
|   NetBIOS_Computer_Name: SERVER2019
|   DNS_Domain_Name: Server2019
|   DNS_Computer_Name: Server2019
|   Product_Version: 10.0.17763
|   System_Time: 2024-03-15T07:57:56+00:00
```

Figure 4-71: Output-2



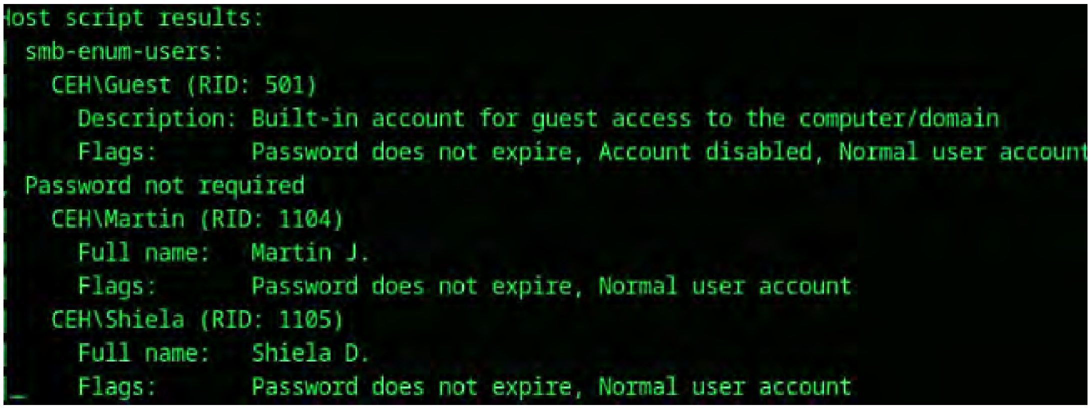
```
Host script results:
|_clock-skew: mean: 1h24m00s, deviation: 3h07m50s, median: 0s
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb-os-discovery:
|   OS: Windows 10 Enterprise 22000 (Windows 10 Enterprise 6.3)
|   OS CPE: cpe:/o:microsoft:windows_10::-
|   Computer name: Windows11
|   NetBIOS computer name: WINDOWS11\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2024-03-15T00:57:58-07:00
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_nbstat: NetBIOS name: WINDOWS11, NetBIOS user: <unknown>, NetBIOS MAC: 00:15:5d
01:80:00 (Microsoft)
```

Figure 4-72: Output-3



```
smb2-time:
  date: 2024-03-15T07:57:57
  start_date: N/A
clock-skew: mean: 1h24m00s, deviation: 3h07m50s, median: 0s
nbstat: NetBIOS name: SERVER2022, NetBIOS user: <unknown>, NetBIOS MAC: 00:15:5d
1:80:02 (Microsoft)
smb-os-discovery:
  OS: Windows Server 2022 Standard 20348 (Windows Server 2022 Standard 6.3)
  Computer name: Server2022
  NetBIOS computer name: SERVER2022\x00
  Domain name: CEH.com
  Forest name: CEH.com
  FQDN: Server2022.CEH.com
  System time: 2024-03-15T00:57:57-07:00
```

Figure 4-73: Output-4



```
Host script results:
| smb-enum-users:
|   CEH\Guest (RID: 501)
|     Description: Built-in account for guest access to the computer/domain
|     Flags:       Password does not expire, Account disabled, Normal user account
| Password not required
|   CEH\Martin (RID: 1104)
|     Full name:   Martin J.
|     Flags:       Password does not expire, Normal user account
|   CEH\Shiela (RID: 1105)
|     Full name:   Shiela D.
|_    Flags:       Password does not expire, Normal user account
```

Figure 4-74: Output-5

This script automates the network enumeration process by executing different Nmap scans on the designated target IP range and storing the results in individual files for further analysis.

Enumeration Countermeasures

Enumeration countermeasures are as follows:

- Eliminate the SNMP agent or disable the SNMP service.
- If disabling SNMP is not feasible, modify the default community string names.
- Upgrade to SNMPv3, which provides encryption for passwords and messages.
- Implement the Group Policy security setting known as "Additional restrictions for anonymous connections."
- Ensure that access to null session pipes, null session shares, and IPsec filtering is limited.
- Restrict access to TCP/UDP port 161.
- Avoid installing the management and monitoring Windows components unless necessary.
- Utilize encryption or authentication through IPsec.
- Ensure the SNMP service is not misconfigured with read-write permissions.
- Set up Access-Control Lists (ACLs) for all SNMP connections to permit access only to authorized users.
- Restrict SNMP access to only those IP addresses or networks that genuinely need it for management tasks. This can be done using ACLs on the devices or through network firewalls.
- Regularly inspect the network traffic.
- Use "AuthNoPriv" mode to encrypt credentials, which utilizes MD5 and SHA for added security.
- Adjust the registry to permit only restricted or allowed access to the SNMP community name.
- Change the default password and update the current password periodically.
- Identify all SNMP devices with read/write permissions and assign read-only permissions to any devices that do not require read/write capabilities.
- Steer clear of the "NoAuthNoPriv" mode since it doesn't provide encryption for communications.
- Apply Role-Based Access Control (RBAC) policies to SNMP communities or users.
- Set up SNMPv3 users in the cluster to bolster security through encryption and authentication.
- For devices still operating on SNMPv1 or SNMPv2c, change the default community strings from the standard "public" and "private" to complex and unique values. Additionally, minimize write access as much as possible.
- Keep management traffic, including SNMP, on a distinct, secure VLAN or network segment. This reduces the risk of SNMP being exposed to potential eavesdroppers or attackers on the primary network.

- If SNMP is unnecessary for network management activities, consider fully disabling it on devices. This removes the protocol as a possible source of data for attackers.
- Manufacturers frequently issue updates that fix security flaws in SNMP implementations. Regularly applying these updates can safeguard against exploitation.
- Deploy monitoring and anomaly detection tools to alert on unusual SNMP traffic patterns that may indicate enumeration or other malicious actions.
- Ensure that SNMP access is logged, and routinely review these logs for unauthorized access attempts or suspicious behavior.

LDAP Enumeration Countermeasures

- LDAP traffic is typically transmitted without security, so it is important to implement a Secure Sockets Layer (SSL) or STARTTLS to encrypt the communication.
- Choose a username that differs from the email address and activate account lockout features.
- Use software such as Citrix to limit access to Active Directory (AD).
- Implement NT LAN Manager (NTLM), Kerberos, or other basic authentication methods to ensure only legitimate users can gain access.
- Record access to Active Directory (AD) services.
- Prevent users from accessing specific AD entities by modifying the permissions on those objects or attributes.
- Establish canary accounts that mimic real accounts to divert attackers.
- Form decoy groups that include the term “Admin” in their names to mislead attackers, as they often look for LDAP admin accounts.
- Activate Multi-Factor Authentication (MFA) for LDAP directory access, adding a layer of security to prevent unauthorized access through compromised credentials.
- Turn off anonymous binds to the LDAP directory unless they are essential for the organization’s functions, ensuring that only authenticated users can query the LDAP server.
- Set up ACLs to define what authenticated users can access and do, restricting visibility of sensitive data based on user credentials and necessity.
- Make sure that all LDAP queries and modifications are recorded. Consistently review these logs for any unusual or unauthorized access patterns that may indicate attempts at enumeration or other malicious activities.
- Utilize monitoring tools that can detect abnormal patterns in LDAP queries, which can notify administrators of potential enumeration or attack attempts in real time.
- Position LDAP servers within a secure network segment that is only accessible to the systems and users that need access reducing the attack surface and minimizing the chances of unauthorized entry.
- Configure firewalls to limit LDAP traffic to and from authorized systems exclusively, which includes blocking unnecessary external access to LDAP services.
- Apply strong password policies for accounts with LDAP access to reduce the risk of brute-force or credential-stuffing attacks.

NFS Enumeration Countermeasures

- Ensure that permissions for reading and writing are restricted to designated users in exported file systems.
- Establish firewall rules to prohibit access to NFS port 2049.
- Confirm that configuration files such as /etc/smb.conf, /etc/exports, and /etc/hosts.allow are properly set up to safeguard the data on the server.
- Examine and revise the /etc/exports file to guarantee that only permitted hosts can access shared directories.
- Utilize /etc/hosts.allow and /etc/hosts.deny files to specify which hosts or networks are authorized or denied access to NFS services.
- Record requests for access to the system files on the NFS server.
- Keep the root_squash option activated in the /etc/exports file to ensure that requests from the client made as root are not trusted.
- Set up NFS tunneling via SSH to encrypt NFS traffic over the network.
- Apply the least privilege principle to reduce threats such as data modification, data addition, and changes to configuration files by standard users.
- Ensure users are not executing suid and sgid on the exported file system.
- Verify that the NIS netgroup has a fully defined hostname to prevent providing higher access to other hosts.
- Configure a Deep Packet Inspection (DPI) firewall to oversee all NFS traffic, regardless of port number.
- Implement Kerberos authentication for NFS to guarantee secure authentication between the client and server, helping to avert unauthorized access.
- NFSv4 offers enhanced security features compared to earlier versions, including the capacity to use Kerberos for both encryption and authentication. Transitioning to NFSv4 can greatly improve security.
- Keep NFS servers and clients in a secluded, segmented area of the network to restrict access from unauthorized network segments.
- Set up firewalls to limit NFS traffic to and from only authorized systems. Blocking unnecessary external access to NFS services can help prevent unauthorized discovery and access.
- Regularly review NFS server access logs for any unusual patterns or attempts at access from unauthorized hosts. This can facilitate early detection of enumeration or attack attempts.
- Utilize file system auditing tools to monitor and log access to NFS shares. Auditing can assist in identifying unauthorized access or changes to sensitive files.
- Periodically update and patch the NFS server software and client systems to defend against known vulnerabilities that might be exploited during enumeration or attacks.

SMTP Enumeration Countermeasures

SMTP servers should be set up as follows:

- Ignore email messages sent to unknown recipients.

- Remove sensitive information from mail responses on mail servers and local hosts.
- Turn off the open relay feature.
- Limit accepted connections from any source to thwart brute-force attacks.
- Disable the EXPN, VRFY, and RCPT TO commands or restrict them to authenticated users.
- Dismiss emails aimed at unknown recipients by configuring SMTP servers accordingly.
- Utilize Machine Learning (ML) solutions to detect spammers.
- Refrain from disclosing internal IP/host details or mail relay system specifics.
- Implement Sender Policy Framework (SPF), Domain Keys Identified Mail (DKIM), and Domain-based Message Authentication And Reporting & Conformance (DMARC).
- Set up the SMTP server to give limited information in error messages. Detailed responses could provide attackers with insights into server configuration or valid user accounts.
- Use Access Control Lists (ACLs) to limit certain SMTP commands to authorized users or IP addresses, preventing anonymous or unauthorized users from trying to gather user information.
- Ensure the SMTP server necessitates authentication before granting access to any information or allowing emails to be sent, which aids in preventing anonymous enumeration attempts.
- Employ Transport Layer Security (TLS) to encrypt communications with the SMTP server, ensuring that all data exchanged, including authentication credentials, is secure.
- Make sure the SMTP server records access attempts and commands used, and regularly review these logs to spot any suspicious activity or attempted enumeration.
- Utilize security tools that can analyze log files and detect unusual behavioral patterns, such as a significant number of failed login attempts, which might suggest an enumeration attempt.
- Implement firewalls to manage access to the SMTP server, permitting only trusted IP addresses or networks to connect.
- Enforce rate limiting to control the number of requests an IP address can make to the SMTP server within a specific timeframe, helping to minimize brute-force attacks.

SMB Enumeration Countermeasures

Common file sharing services or other services that are not utilized can serve as entry points for attackers looking to bypass network security measures. A network utilizing the SMB protocol is particularly vulnerable to enumeration attacks. Since web and DNS servers do not need this protocol, it is recommended to disable it on those servers. SMB can be turned off by disabling the features Client for Microsoft Networks and File and Printer Sharing for Microsoft Networks in Network and Dial-up Connections. For servers that are exposed to the Internet, often called bastion hosts, SMB can also be disabled by turning off these same two features in the TCP/IP properties dialog box. An alternative way to disable the SMB protocol on bastion hosts without specifically turning it off, is to block the ports associated with the SMB service. These ports are TCP ports 139 and 445.

Since turning off SMB services is not always practical, additional measures to counter SMB enumeration may be necessary. The Windows Registry can be adjusted to limit anonymous access from the Internet to a select group of files. These files and folders are listed in the settings for Network access: Named pipes that can be accessed anonymously and Network access: Shares that can be accessed anonymously. Implementing this configuration requires adding the RestrictNullSessAccess parameter to the registry key located at:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters

The RestrictNullSessAccess parameter accepts binary values, where 1 indicates that it is enabled and 0 indicates that it is disabled. Setting this parameter to 1 or enabling it restricts anonymous users' access to the specified files listed in the Network access settings.

The following are additional countermeasures for defending against SMB enumeration:

- Make certain that the Windows Firewall or comparable endpoint protection tools are active on the device.
- Apply the most recent security updates for Windows and any third-party applications.
- Establish a proper authentication method along with a robust password policy.
- Enforce strong permissions to safeguard stored data.
- Conduct regular reviews of system logs.
- Engage in proactive system monitoring to detect any malicious activities.
- Utilize secure VPNs to protect organizational data during remote access.
- Implement file behavioral analysis systems, such as Next-Generation Firewalls (NGFWs), to analyze traffic patterns and generate timely reports on SMB resources.
- Use highly secure monitoring systems like global threat sensors for extremely sensitive and classified information.
- Employ digitally signed data transmission and communications for accessing SMB resources.
- Block or disable TCP ports 88, 139, and 445 and UDP ports 88, 137, and 138 to avert SMB-related attacks.
- Activate public profile settings in the firewall configuration.
- Block or disable the SMB protocol on servers that face the Internet.
- Ensure that SMB convention web-facing and DNS servers are turned off.
- Confirm that all systems utilize SMBv3 or later, which includes improved security features like encryption. Avoid employing SMBv1, as it is obsolete and poses security risks.
- Set up ACLs to limit access to SMB shares solely to necessary users and routinely review and tighten these permissions.
- Apply the least privilege principle to ensure that users and services function with the minimum required permissions, thereby reducing the potential fallout from compromised accounts.
- Configure SMB servers to log access attempts and modifications to shared resources and regularly examine logs for any unusual activity.

DNS Enumeration Countermeasures

Outlined below are various strategies to prevent DNS enumeration:

- **Limit Resolver Access:** Ensure that resolvers are accessible solely by internal hosts within the network to prevent cache poisoning from external sources.
- **Randomize Source Ports:** Ensure that outgoing request packets utilize random ports instead of UDP port 53. Additionally, randomize query IDs and modify the case of domain names to guard against cache poisoning.
- **Review DNS Zones:** Conduct audits of DNS zones to discover vulnerabilities in domains and subdomains and resolve any DNS-related issues.
- **Fix Known Vulnerabilities:** Update and patch nameservers with the latest software versions, such as BIND and Microsoft DNS.
- **Observe Nameservers:** Continuously monitor nameservers to detect any malicious actions or unusual behaviors promptly.
- **Limit DNS Zone Transfers:** Restrict DNS zone transfers to designated slave nameserver IP addresses since such transfers may contain a master copy of the primary server's database. Disable transfers to untrusted hosts.
- **Separate Servers for Authoritative and Resolving Roles:** Splitting the resolver and authoritative nameserver functions can alleviate overload and protect against Denial of Service (DoS) attacks on domains.
- **Utilize Dedicated DNS Servers**: Refrain from hosting application servers together with DNS servers. Use a separate and dedicated server for DNS services to lower the risk of web application attacks.
- **Turn Off DNS Recursion:** Disable DNS recursion in the configuration of the DNS server to restrict queries from other or third-party domains and reduce the risk of DNS amplification and poisoning attacks.
- **Secure the Operating System**: Enhance the security of the OS by closing unused ports and disabling unnecessary services.
- **Employ a VPN:** Use a VPN for secure communications. Additionally, change default passwords.
- **Enforce Two-Factor Authentication:** Implement two-factor authentication to ensure secure access when a third party manages a DNS server.
- **Utilize DNS Change Lock:** Implement a DNS change lock or client lock to prevent changes to DNS settings without proper authorization.
- **Implement DNSSEC:** Use DNSSEC as an extra security layer for the DNS server, allowing only digitally signed DNS requests to mitigate DNS hijacking.
- **Opt for Premium DNS Registration:** Choose premium DNS registration services that conceal sensitive information, such as host details (HINFO), from public visibility.
- **Encrypt DNS Queries:** Consider employing DNS-over-HTTPS (DoH) or DNS-over-TLS (DoT) to secure DNS queries and responses, thus mitigating eavesdropping and man-in-the-middle attack risks that could enable DNS enumeration.

- **Activate DNS Logging and Monitoring:** Enable logging on DNS servers to capture queries and responses. Regularly monitoring and analyzing these logs can help identify suspicious activity that may indicate enumeration attempts.
- **Utilize Anomaly Detection:** Implement anomaly detection systems to automatically highlight unusual DNS query volumes or patterns, which may signify enumeration or other DNS attacks.
- **Apply Rate Limiting:** Set up DNS servers to restrict the rate of accepted queries from individual IP addresses to diminish the impact of brute-force enumeration techniques.
- **Design a Split DNS Architecture:** Create a split DNS architecture that manages internal DNS queries on a separate DNS server from those originating externally, limiting the exposure of internal network configurations.
- **Share Minimal DNS Information:** Exercise caution regarding the amount of information disclosed through DNS records. For example, avoid using detailed subdomain names that could expose internal network information or server functions.

Other measures to protect against DNS enumeration include:

- Ensure that private hosts and their associated IP addresses are not included in the public DNS server's zone files.
- Use standard network administrator contacts for DNS registrations to prevent social engineering threats.
- Simplify DNS zone files to prevent the disclosure of unnecessary information.
- Maintain distinct internal and external DNS servers.
- Regularly remove old or unused DNS records.
- Restrict version.bind request queries using Access Control Lists (ACLs), and run BIND with minimal privileges.
- Utilize the /etc/hosts file for developing or staging subdomains instead of relying on DNS records.
- Implement DNS Firewalls to block harmful queries and protect against DNS-based threats by leveraging threat intelligence to identify and prevent communication with known malicious domains.
- Consistently review and audit DNS configurations to confirm their security and that only essential DNS information is made public.

Summary

Enumeration is the process of actively gathering detailed information about a target system or network, typically used by attackers to identify vulnerabilities for exploitation. It involves probing systems to collect data such as user accounts, network shares, and configurations, which can then assist in planning further attacks.

The chapter covers key enumeration techniques, starting with an introduction to the concept. Unlike passive reconnaissance, enumeration actively interacts with services like NetBIOS, SNMP, and LDAP to reveal valuable system details. Techniques like NetBIOS enumeration are used to identify network shares, users, and services, while SNMP enumeration targets devices to extract configurations that may reveal sensitive data. LDAP enumeration focuses on extracting directory information about users and resources, especially in Windows Active Directory environments.

Other methods include NTP and NFS enumeration, which reveal server details and network file shares. SMTP and DNS enumeration help attackers collect email addresses, mail routing information, and DNS records that aid in phishing or domain spoofing attacks. Additional enumeration techniques include SMB, RPC, and HTTP, each targeting specific services for further exploitation. To defend against these techniques, countermeasures like disabling unnecessary services, implementing strong access controls, securing protocols (e.g., using SNMPv3), and monitoring network traffic are recommended to prevent unauthorized enumeration attempts.

Mind Map

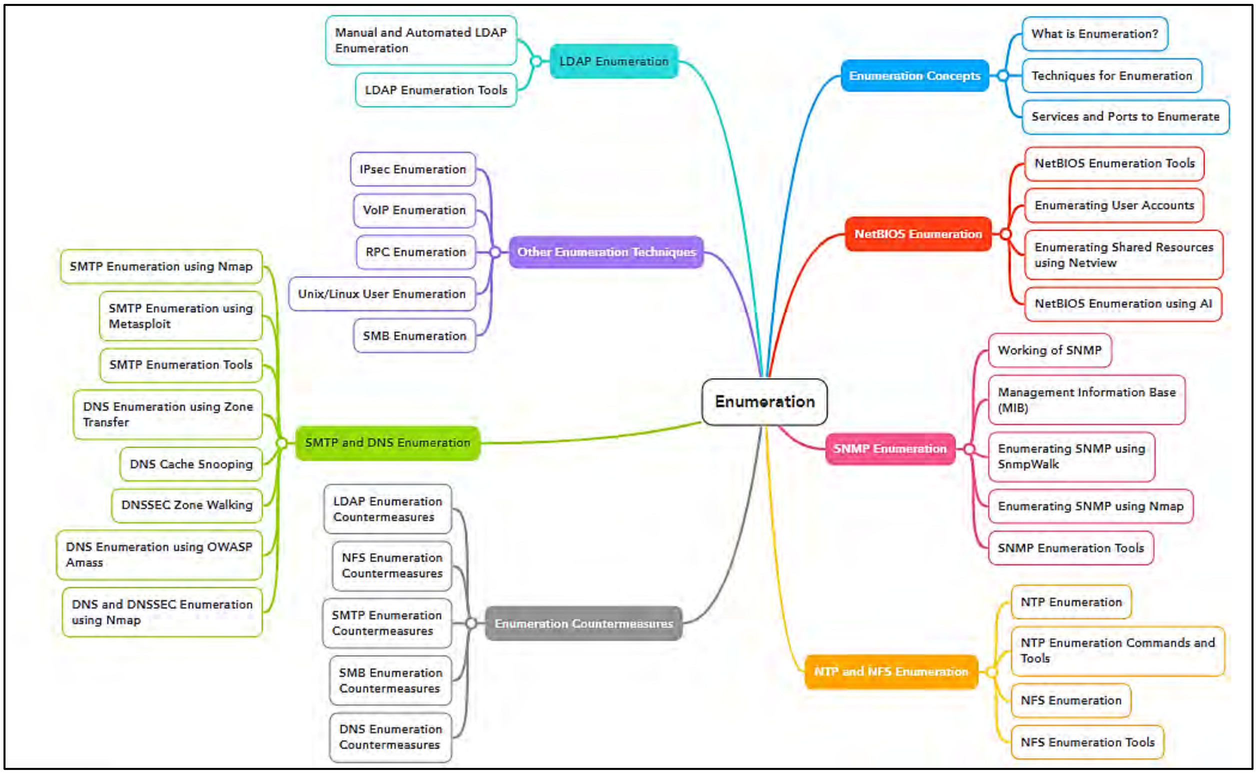

Enumeration
LDAP Enumeration
Manual and Automated LDAP Enumeration
LDAP Enumeration Tools
Other Enumeration Techniques
IPsec Enumeration
VoIP Enumeration
RPC Enumeration
Unix/Linux User Enumeration
SMB Enumeration
SMTP and DNS Enumeration
SMTP Enumeration using Nmap
SMTP Enumeration using Metasploit
SMTP Enumeration Tools
DNS Enumeration using Zone Transfer
DNS Cache Snooping
DNSSEC Zone Walking
DNS Enumeration using OWASP Amass
DNS and DNSSEC Enumeration using Nmap
Enumeration Countermeasures
LDAP Enumeration Countermeasures
NFS Enumeration Countermeasures
SMTP Enumeration Countermeasures
SMB Enumeration Countermeasures
DNS Enumeration Countermeasures
Enumeration Concepts
What is Enumeration?
Techniques for Enumeration
Services and Ports to Enumerate
NetBIOS Enumeration
NetBIOS Enumeration Tools
Enumerating User Accounts
Enumerating Shared Resources using Netview
NetBIOS Enumeration using AI
SNMP Enumeration
Working of SNMP
Management Information Base (MIB)
Enumerating SNMP using SnmpWalk
Enumerating SNMP using Nmap
SNMP Enumeration Tools
NTP and NFS Enumeration
NTP Enumeration
NTP Enumeration Commands and Tools
NFS Enumeration
NFS Enumeration Tools

Figure 4-75: Mind Map

Practice Questions

1. Which port is commonly used for SNMP enumeration?
A. 23
B. 53
C. 161
D. 443

2. What does LDAP primarily provide?
A. File sharing services
B. Directory services
C. Network monitoring
D. Time synchronization

3. In NetBIOS enumeration, which tool is commonly used to query NetBIOS name tables?
A. Nessus
B. nbtscan
C. Metasploit
D. Wireshark

4. Which command can retrieve a list of time servers during NTP enumeration?
A. ntpq -p
B. snmpget
C. ldapsearch
D. dig -x

5. During SNMP enumeration, the most commonly used community strings are __________ and _________.
A. public, private
B. admin, root
C. manager, user
D. guest, test

6. Which tool is commonly used for LDAP enumeration?
A. nmap
B. snmpwalk
C. enum4linux
D. ldapsearch

7. NetBIOS enumeration allows attackers to identify shared folders and printers on a network.
A. True
B. False

8. Which protocol uses port 389 for communication?
A. SNMP
B. LDAP
C. NTP
D. SMTP

9. NetBIOS operates primarily over ports __________, __________, and _________.
A. 21, 23, 25
B. 22, 80, 443
C. 135, 137, 139
D. 53, 161, 162

10. DNS enumeration using the dig command is primarily performed to discover:
A. Subdomains and mail servers
B. Active user accounts
C. SNMP community strings
D. Shared network drives

11. During NFS enumeration, the **showmount** command is used to __________.
A. display exported file systems
B. query running processes
C. check time synchronization
D. verify DNS zones

12. The Simple Network Management Protocol (SNMP) uses encryption to secure transmitted data by default.
A. True
B. False

13. The **AXFR** request in DNS enumeration is used to __________.
A. perform a zone transfer
B. test connectivity
C. query LDAP directories
D. enumerate shared drives

14. Using countermeasures like disabling unused protocols can reduce the risk of enumeration.
A. True
B. False

15. The NFS export list provides information about:
A. Network routing tables
B. Shared file systems accessible to clients
C. Open ports on the server
D. Encrypted user credentials

16. DNS zone transfers can provide attackers with a list of all DNS records in a domain.
A. True
B. False

17. Which command is used to list shared directories on a target system during NetBIOS enumeration?
A. net use
B. nbtstat -A <IP>

C. showmount
D. ldapsearch -x

18. What is the purpose of the **EXPN** command in SMTP enumeration?
A. Validate recipient addresses
B. Expand mailing lists
C. Enumerate LDAP entries
D. Query DNS zones

19. The command _________ can be used to query DNS servers for Name Server (NS) records.
A. nslookup
B. rpcclient
C. ldapsearch
D. snmpget

20. The **snmpwalk** tool can be used to automate bulk requests in SNMP enumeration.
A. True
B. False

21. What is the primary protocol used for VoIP signaling during enumeration?
A. HTTP
B. RTP
C. SIP
D. FTP

22. Which enumeration technique involves querying port 25 for user information?
A. SNMP enumeration
B. NetBIOS enumeration
C. SMTP enumeration
D. LDAP enumeration

23. Enumeration countermeasures involve disabling unnecessary services such as __________ and ____________.
A. NetBIOS and NFS
B. HTTPS and SSH
C. DNS and SNMP
D. SMTP and TLS

24. SNMP enumeration relies on ___________, which act as simple passwords for accessing device information.
A. community strings
B. certificates
C. encryption keys
D. access tokens

25. Which of the following tools can be used to identify open RPC services on a target system?
A. Nessus
B. Nmap

C. Nikto
D. Burp Suite

Answers

1. **Answer:** C
Explanation: SNMP (Simple Network Management Protocol) typically operates over UDP port 161 to query and manage network devices like routers, switches, and servers. It is used for monitoring and collecting information about network health and performance, making it a key target for enumeration efforts.

2. **Answer:** B
Explanation: Lightweight Directory Access Protocol (LDAP) is designed to access and manage directory services over a network. These services offer a centralized location to store and retrieve information about users, groups, devices, and other network resources.

3. **Answer:** B
Explanation: nbtscan is a tool specifically designed to perform NetBIOS enumeration. It queries NetBIOS name tables to gather information about hostnames, workgroups, and shared resources on Windows-based systems, leveraging the NetBIOS protocol.

4. **Answer:** A
Explanation: The ntpq -p command queries an NTP server and displays a list of its peers (time servers) along with their synchronization status. This information is useful during NTP enumeration to identify time synchronization vulnerabilities or other related configuration details.

5. **Answer:** A
Explanation: In SNMP, community strings act as passwords for accessing and managing network devices. The default strings "public" (read-only access) and "private" (read-write access) are widely used but often remain unchanged, making them a common target during enumeration.

6. **Answer:** D
Explanation: ldapsearch is a command-line utility used for querying and interacting with LDAP directories. It allows the enumeration of directory contents such as users, groups, and organizational units, making it a widely used tool for LDAP enumeration.

7. **Answer:** A
Explanation: NetBIOS enumeration can reveal shared resources like folders and printers in Windows networks by querying NetBIOS name services. Attackers can exploit this information to map the network and potentially gain unauthorized access to shared resources.

8. **Answer:** B
Explanation: LDAP typically uses port 389 for communication. It is used for querying and managing directory services, such as accessing information about users, groups, and devices in a networked environment.

9. **Answer:** C
Explanation: NetBIOS operates over ports 135, 137, and 139, each with distinct roles. Port 135 is used for NetBIOS Name Service (NBNS) and RPC, enabling network communication. Port 137

resolves NetBIOS names to IP addresses, helping devices locate each other. Port 139 is for NetBIOS Session Service, supporting file and printer sharing over TCP/IP networks in Windows environments.

10. **Answer:** A
Explanation: The dig command is commonly used for DNS enumeration, allowing the discovery of various DNS records, such as subdomains and mail servers (MX records). This information is helpful for attackers to map a target's domain structure and identify potential points of exploitation.

11. **Answer:** A
Explanation: During NFS enumeration, the showmount command is used to display the exported file systems on an NFS server. It reveals which directories are shared over the network and may be accessible to clients, which is valuable information during penetration testing and security assessments.

12. **Answer:** B
Explanation: By default, SNMP does not use encryption to secure transmitted data. SNMPv1 and SNMPv2c send data in clear text, making it vulnerable to interception. However, SNMPv3 supports encryption and provides additional security features like authentication and data integrity.

13. **Answer:** A
Explanation: The AXFR request in DNS enumeration is used to perform a zone transfer, which allows an attacker to request a full copy of a domain's DNS records from a DNS server. This can reveal valuable information about the domain, such as subdomains, mail servers, and other associated resources.

14. **Answer:** A
Explanation: Disabling unused protocols is an effective countermeasure to reduce the risk of enumeration. By turning off unnecessary services and protocols, such as NetBIOS, SNMP, or LDAP, organizations can limit the amount of information available to attackers, making it more difficult for them to gather details about the network or system.

15. **Answer:** B
Explanation: The NFS export list provides information about the shared file systems that are accessible to clients. It reveals which directories on an NFS server are being exported for remote access and can be queried using tools like showmount.

16. **Answer:** A
Explanation: DNS zone transfers (using the AXFR request) can provide attackers with a complete list of all DNS records in a domain, including subdomains, mail servers, and other resource records.

17. **Answer:** B
Explanation: The nbtstat -A <IP> command is used to query the NetBIOS name table of a remote computer specified by its IP address. This will display information about shared resources, including shared directories and printers.

18. **Answer:** B
Explanation: The EXPN command in SMTP is used to expand mailing lists, revealing the members of a distribution list. This can be valuable for attackers during enumeration, as it exposes additional email addresses within an organization.

19. **Answer:** A
Explanation: The nslookup command is used to query DNS servers for several types of DNS records, including Name Server (NS) records. These records indicate which servers are authoritative for a domain, helping identify the infrastructure responsible for handling DNS queries for that domain.

20. **Answer:** A
Explanation: The snmpwalk tool is used to automate bulk requests during SNMP enumeration. It queries an SNMP-enabled device for a large set of data by walking through the SNMP Management Information Base (MIB), retrieving all available information for specified Object Identifiers (OIDs).

21. **Answer:** C
Explanation: The primary protocol used for VoIP signaling during enumeration is Session Initiation Protocol (SIP). SIP is responsible for setting up, managing, and dismissing voice and video calls over IP networks.

22. **Answer:** C
Explanation: SMTP enumeration involves querying port 25 (the default SMTP port) for user information. Tools and techniques like the VRFY and EXPN commands can be used to verify and expand email addresses, respectively, allowing attackers to gather valid email addresses from the target system.

23. **Answer:** A
Explanation: Enumeration countermeasures involve disabling unnecessary services like NetBIOS and NFS to reduce the attack surface. These services, if left enabled, can expose sensitive information about the network, shared resources, and system configurations, making them targets for attackers.

24. **Answer:** A
Explanation: SNMP enumeration relies on community strings, which act as simple passwords for accessing device information. These strings, typically default values like "public" (read-only) and "private" (read-write), allow SNMP clients to query network devices and retrieve data, such as device configurations and statistics.

25. **Answer:** B
Explanation: Nmap is a widely used network scanning tool that can identify open ports and services, including RPC services, on a target system. By scanning port 135 (the default RPC port), Nmap can help detect active RPC services and their versions, providing valuable information during enumeration.