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.

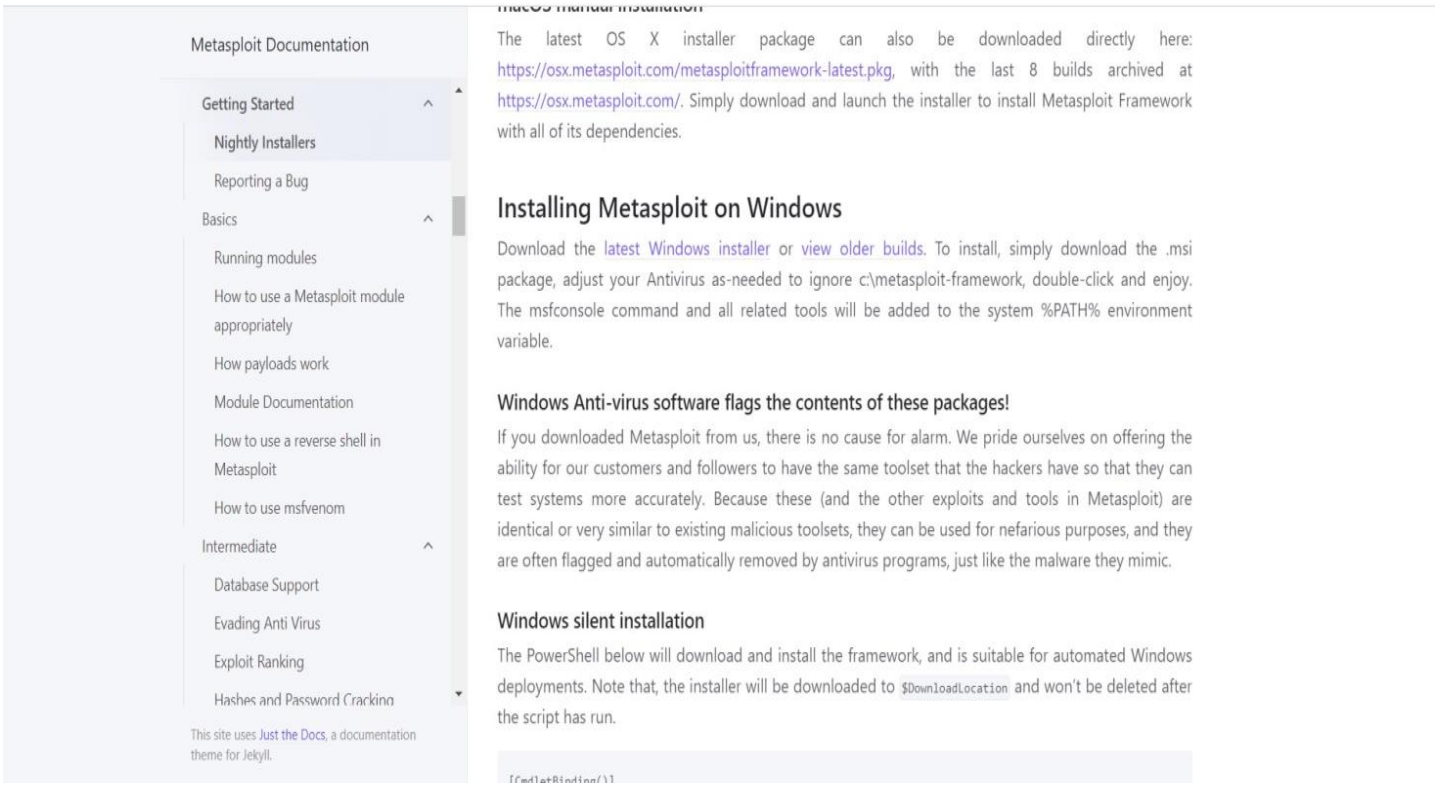
However, understanding server vulnerabilities and taking appropriate steps to secure them is crucial for system administrators and security professionals. Here are some general steps to follow when dealing with server vulnerabilities:

- **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

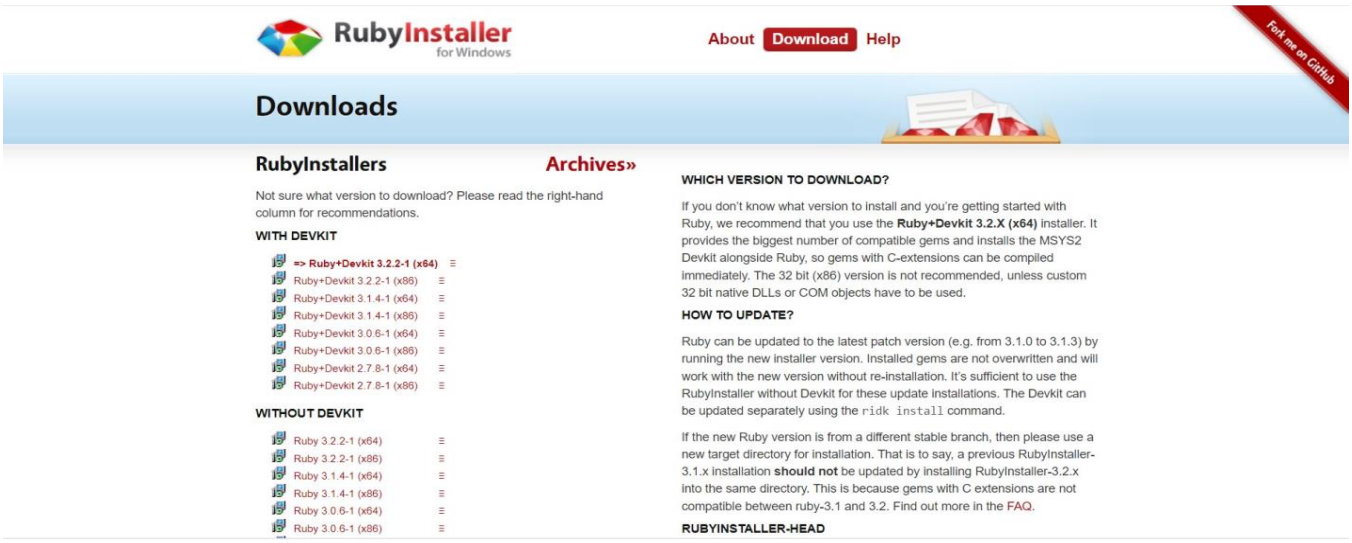
The **msfconsole** is the command-line interface for Metasploit Framework, a powerful open-source penetration testing tool. It provides an extensive collection of exploit modules, payload modules, and auxiliary modules to aid in penetration testing, vulnerability assessment, and security research. We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



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/.



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"

```
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
   -  ----                                      ---------------  ----    -----  -----------
   0  auxiliary/scanner/smtp/smtp_version                        normal  No     SMTP Banner Grabber
   1  auxiliary/scanner/smtp/smtp_ntlm_domain                    normal  No     SMTP NTLM Domain Extraction
   2  auxiliary/scanner/smtp/smtp_relay                          normal  No     SMTP Open Relay Detection
   3  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                        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.
```

```
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.

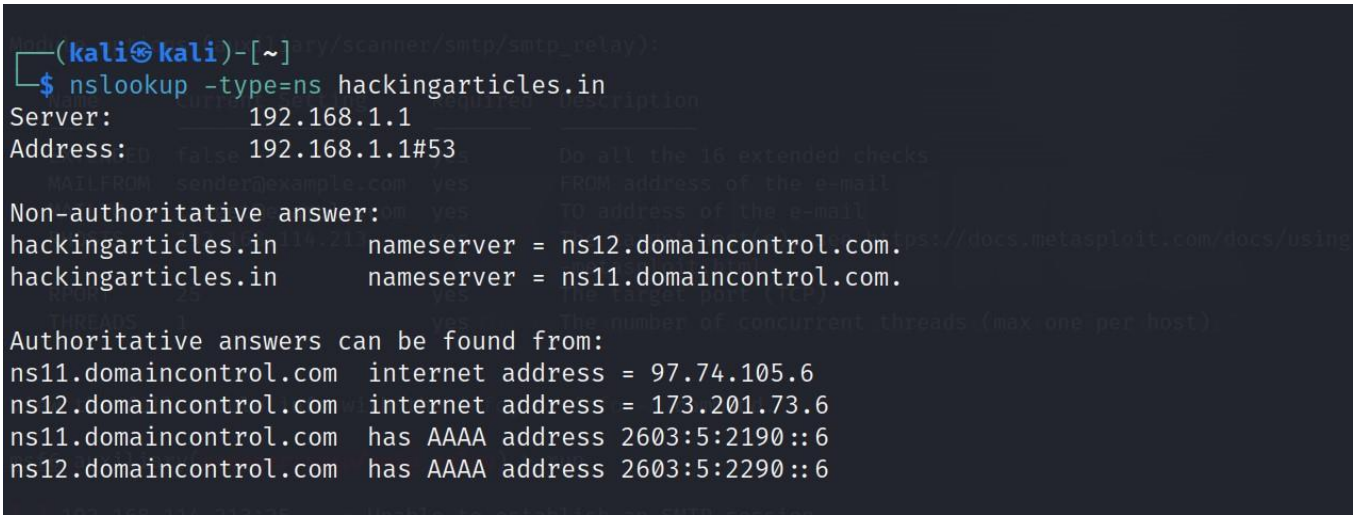
We received the error message "unable to establish an SMTP session" after running the program, it indicates that the module or exploit was not able to establish a connection with the target SMTP server.

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.

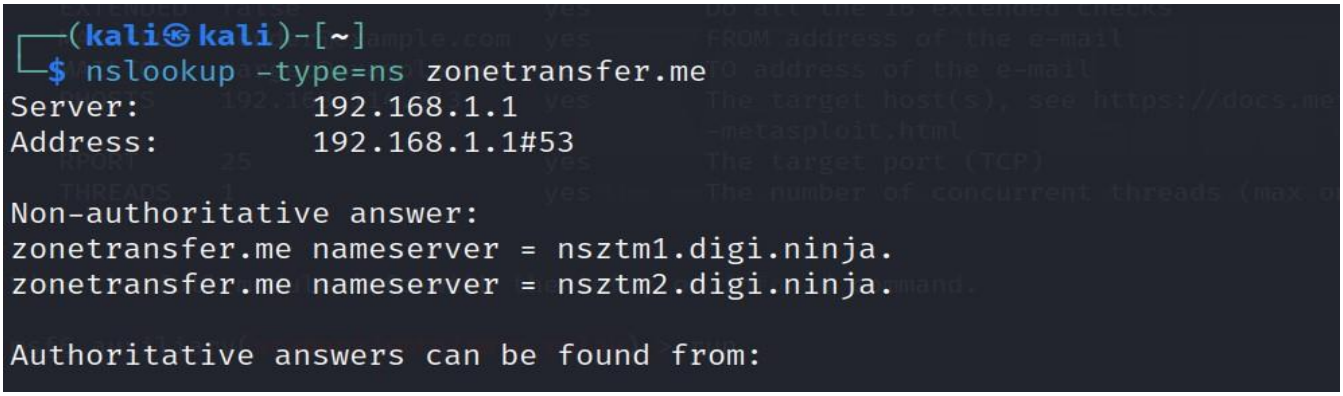


Explanation of the "**nslookup -type=ns hacking articles.in**" command in detail:

- **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.



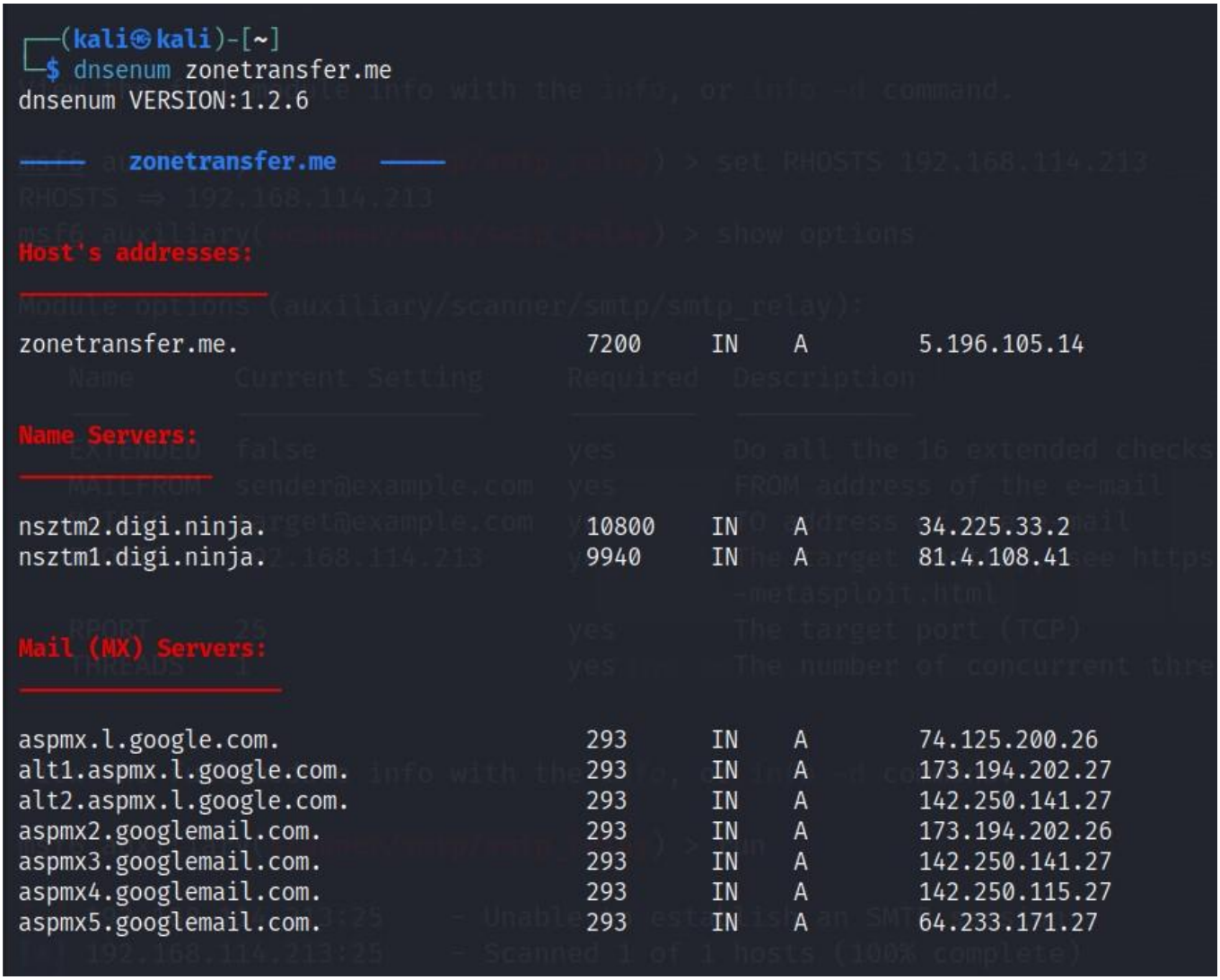
Let's break down the "**nslookup -type=ns zonetranfer.me"** command in detail:

- **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.
- **zonetranfer.me:** This is the domain name for which we want to retrieve the NS records. In this case, it is "zonetranfer.me".

When you execute the **nslookup -type=ns zonetranfer.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 "zonetranfer.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 "zonetranfer.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.



Trying Zone Transfers and getting Bind Versions:

```
Trying Zone Transfer for zonetransfer.me on nsztm2.digi.ninja ...
zonetransfer.me.                        7200    IN    SOA        (
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    MX            20
zonetransfer.me.                        7200    IN    MX            20
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
asfdbbox.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            (
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    MX            20
zonetransfer.me.                        7200    IN    MX            20
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
asfdbbox.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)-[~]
└─$ 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 nsztml.dig.ninja"** is used to perform a DNS zone transfer from the domain "zonetransfer.me" using the DNS server "nsztml.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\a1\f4\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\loca
l-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
[*]     NS nsztm1.digi.ninja 81.4.108.41
[*]     Bind Version for 81.4.108.41 secret"
[*]     NS nsztm2.digi.ninja 34.225.33.2
[*]     Bind Version for 34.225.33.2 you"
[*]     MX aspmx3.googlemail.com 142.250.141.27
[*]     MX aspmx4.googlemail.com 142.250.115.27
[*]     MX aspmx5.googlemail.com 64.233.171.27
[*]     MX aspmx.l.google.com 74.125.200.26
[*]     MX alt1.aspmx.l.google.com 173.194.202.27
[*]     MX alt2.aspmx.l.google.com 142.250.141.27
[*]     MX aspmx2.googlemail.com 173.194.202.26
[*]     MX aspmx3.googlemail.com 2607:f8b0:4023:c0b::1b
[*]     MX aspmx4.googlemail.com 2607:f8b0:4023:1004::1a
[*]     MX aspmx5.googlemail.com 2607:f8b0:4003:c15::1b
[*]     MX aspmx.l.google.com 2404:6800:4003:c1a::1b
[*]     MX alt1.aspmx.l.google.com 2607:f8b0:400e:c00::1b
[*]     MX alt2.aspmx.l.google.com 2607:f8b0:4023:c0b::1a
[*]     MX aspmx2.googlemail.com 2607:f8b0:400e:c00::1b
[*]     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.

```
┌──(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

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:

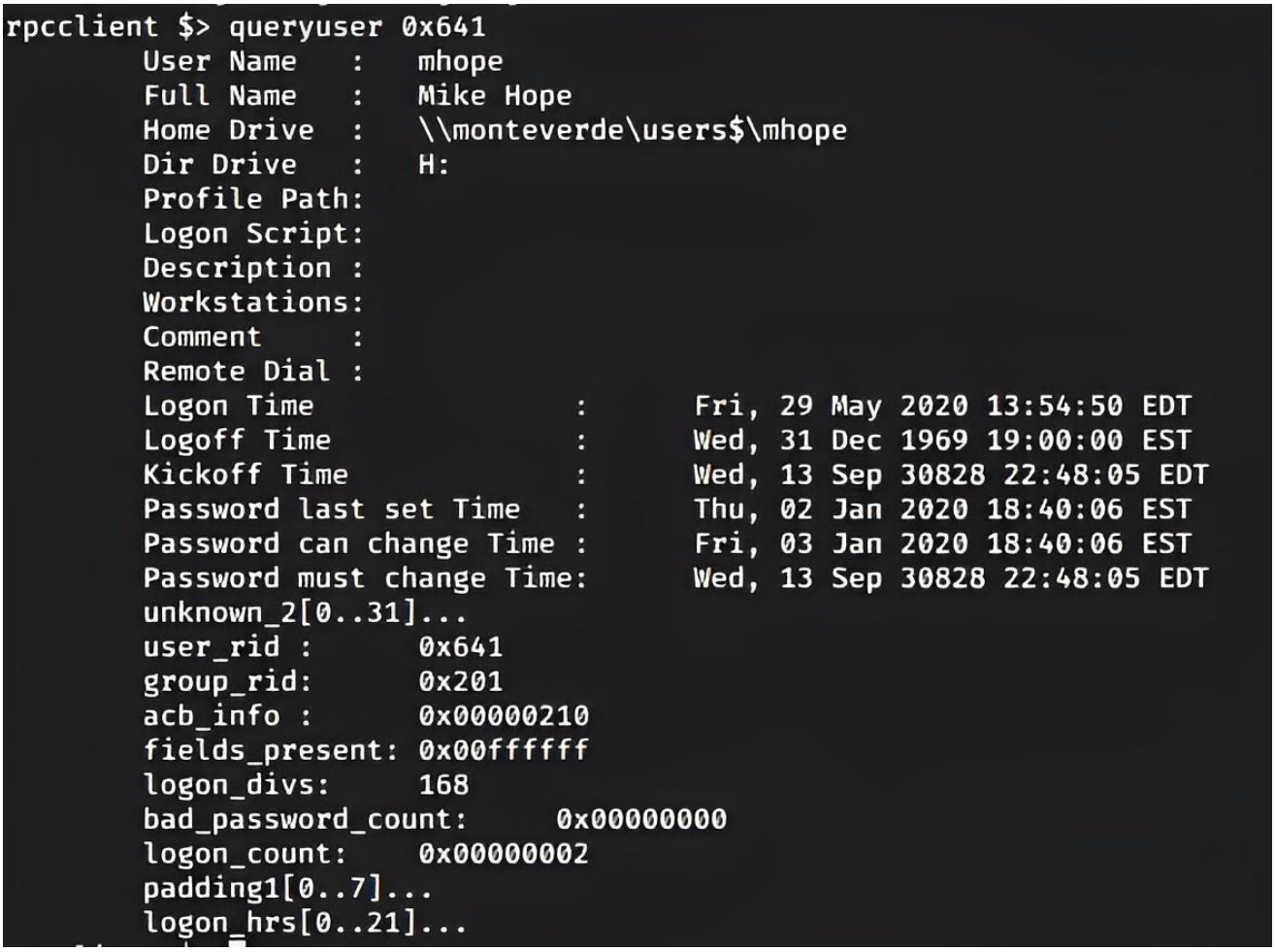
1. rpcclient is the command used to invoke the rpcclient tool.
2. -U " " is the option used to specify the username for authentication. In this case, the username is left blank, which means you are not providing a specific username for authentication.
3. 192.168.111.130 is the IP address of the target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

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.



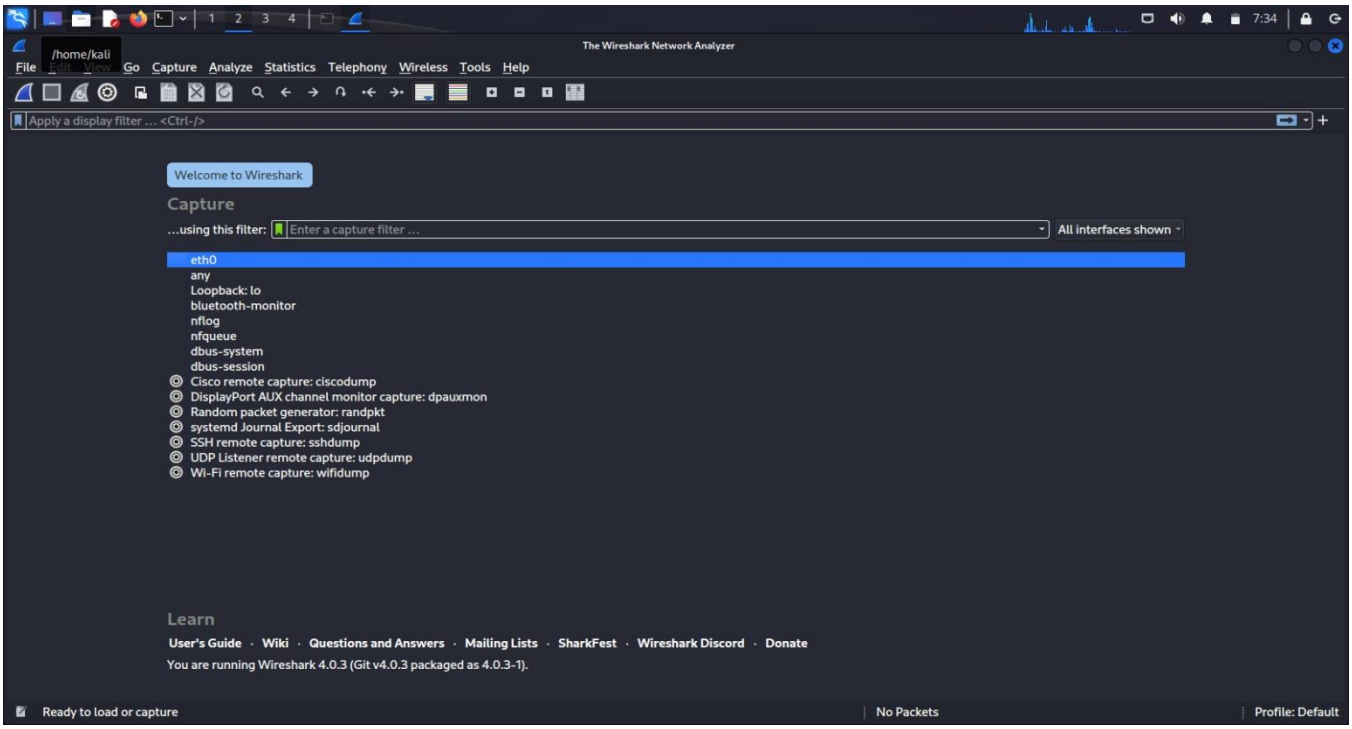
The command **rpcclient $> queryuser 0x641** seems to indicate that you are using the rpcclient tool to query user information using a user ID of "0x641".
If we intend to query user information using rpcclient, you would typically establish an RPC session with the target system and use the appropriate 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.

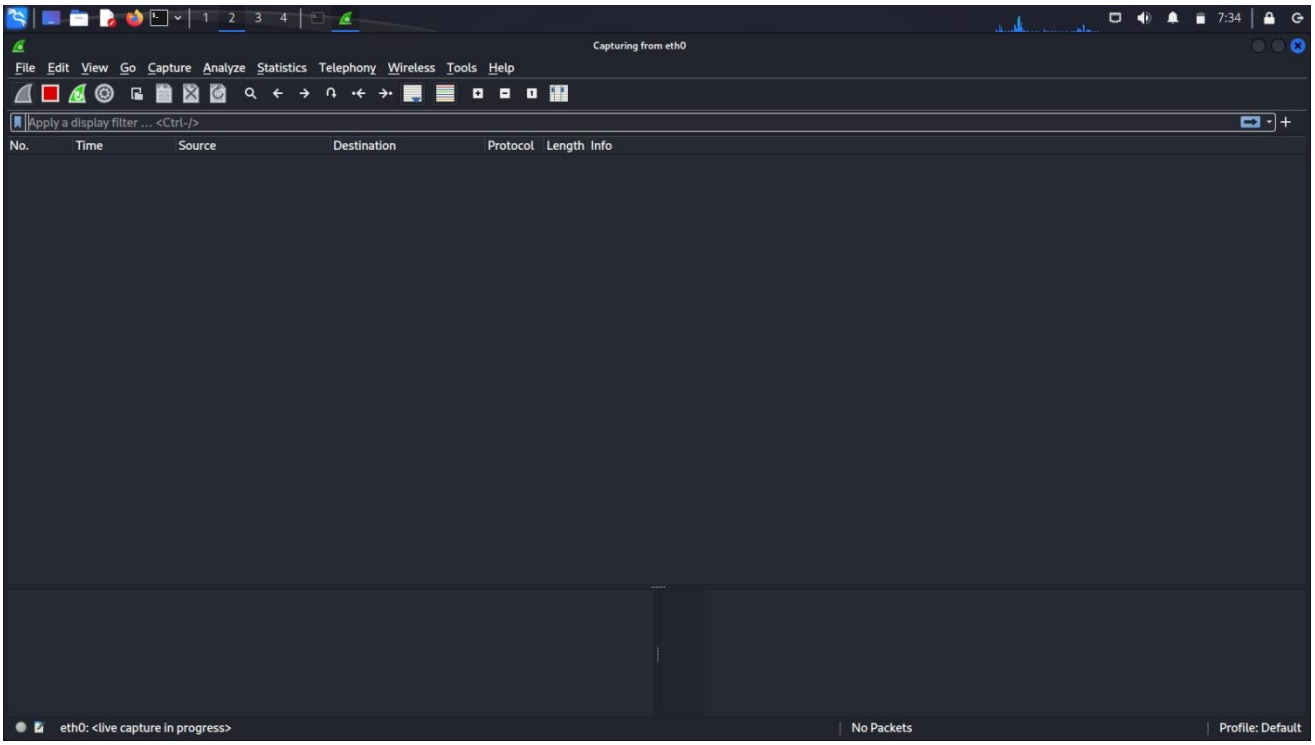
There are different techniques and tools used for network sniffing, including:

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.

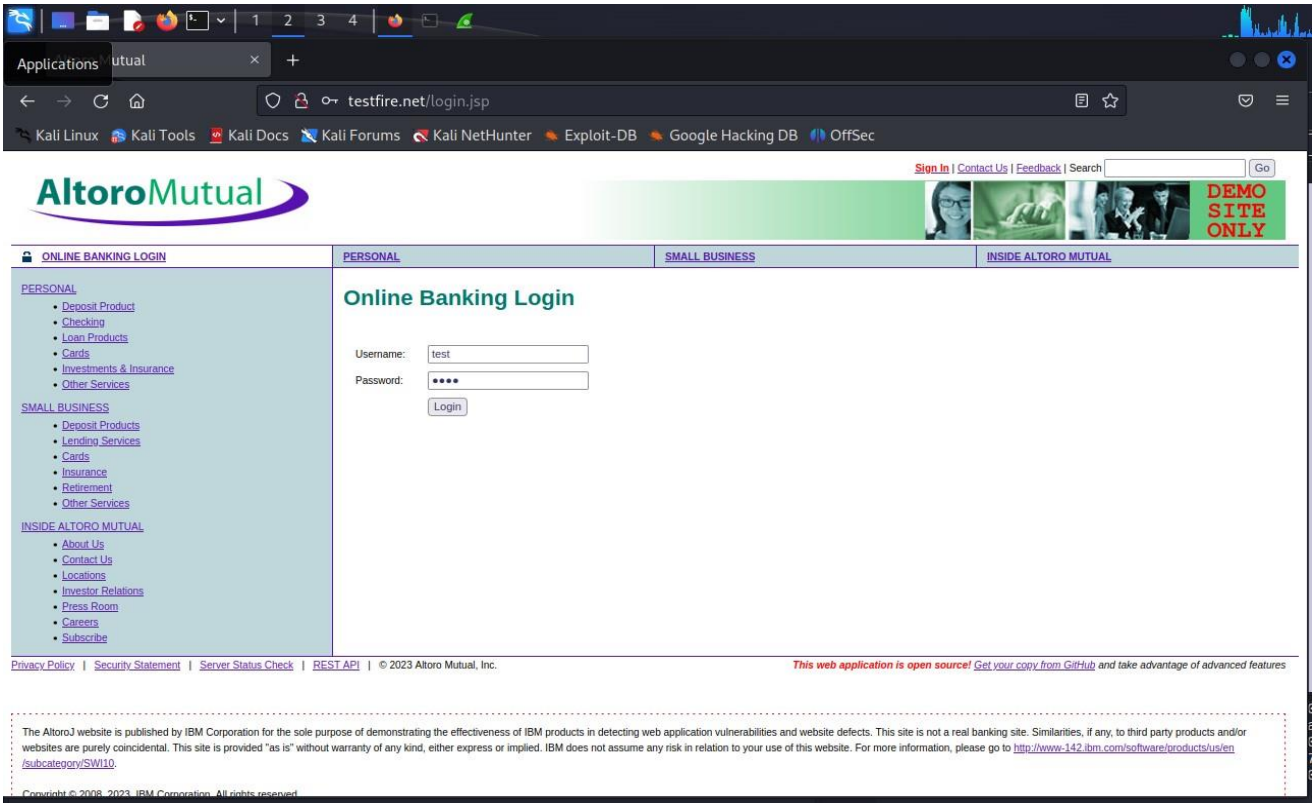
Here we are using Wireshark network sniffing tools like Wireshark can be valuable for legitimate purposes such as network troubleshooting, performance analysis, and protocol analysis. However, using such tools to capture sensitive information like login credentials and passwords without proper authorization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



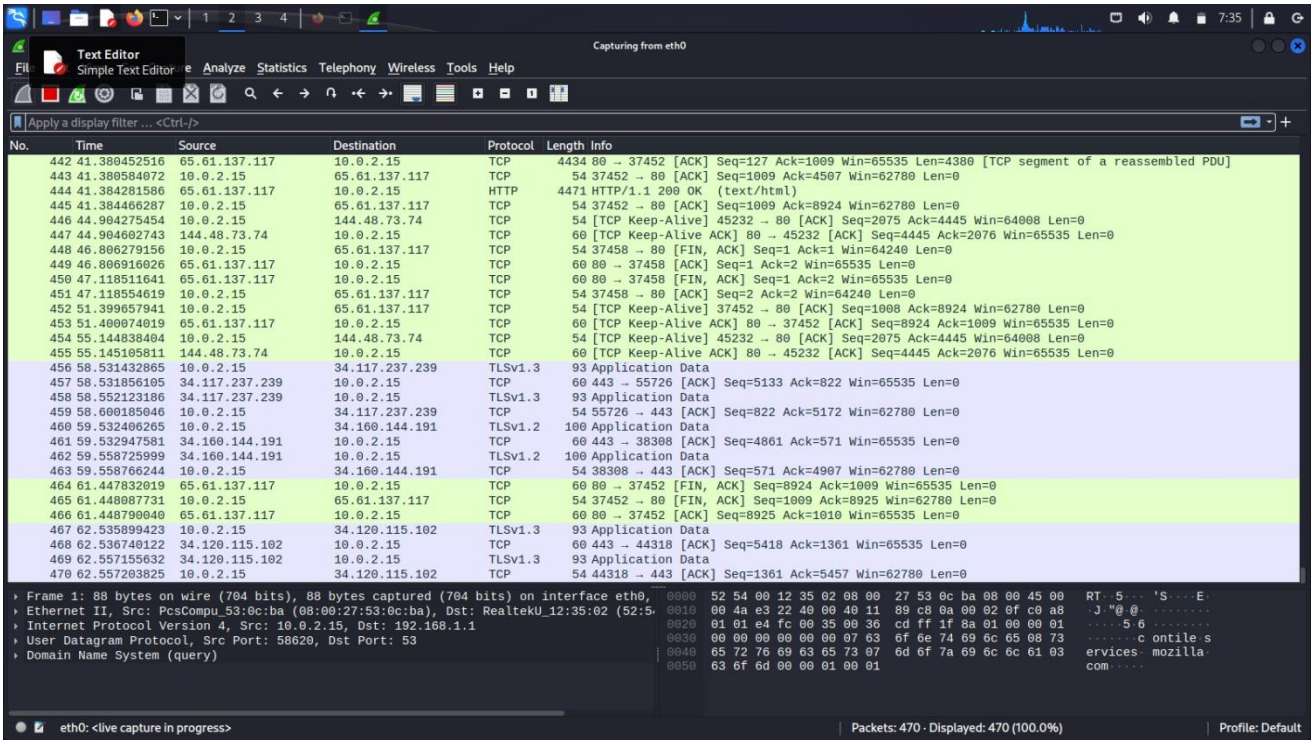
To capture the network traffic we are firstly selecting eth0 interface and Beginning to capture network traffic by clicking the "Start" or "Capture" button in Wireshark.

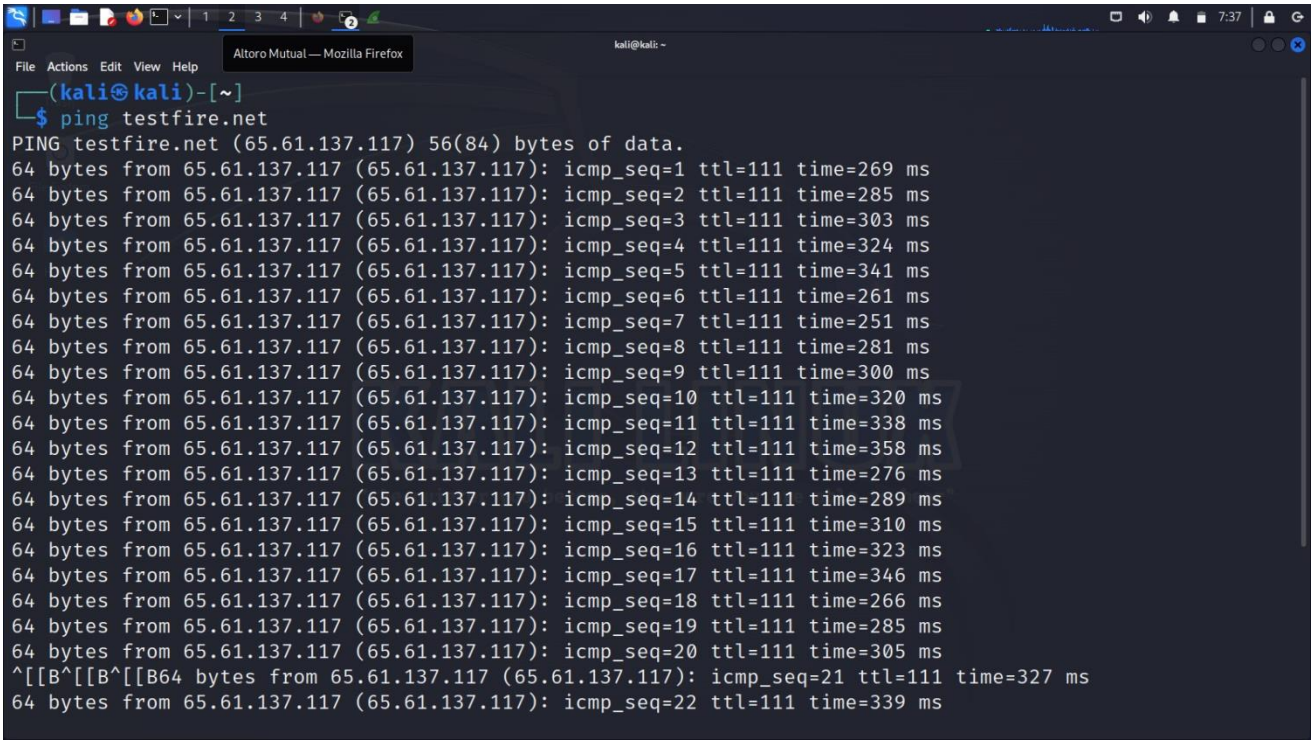


After Launching Wireshark as we did not select the appropriate network interface that you want to sniff there is no traffic to capture.

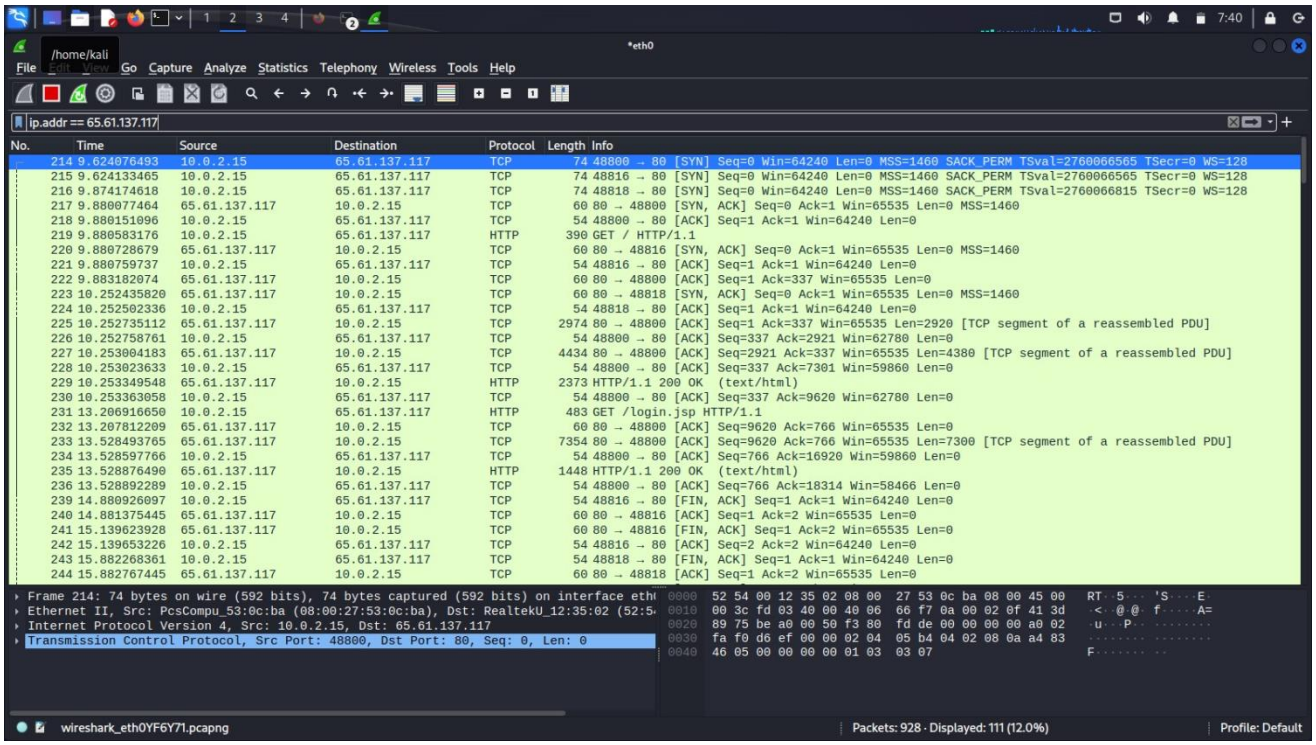


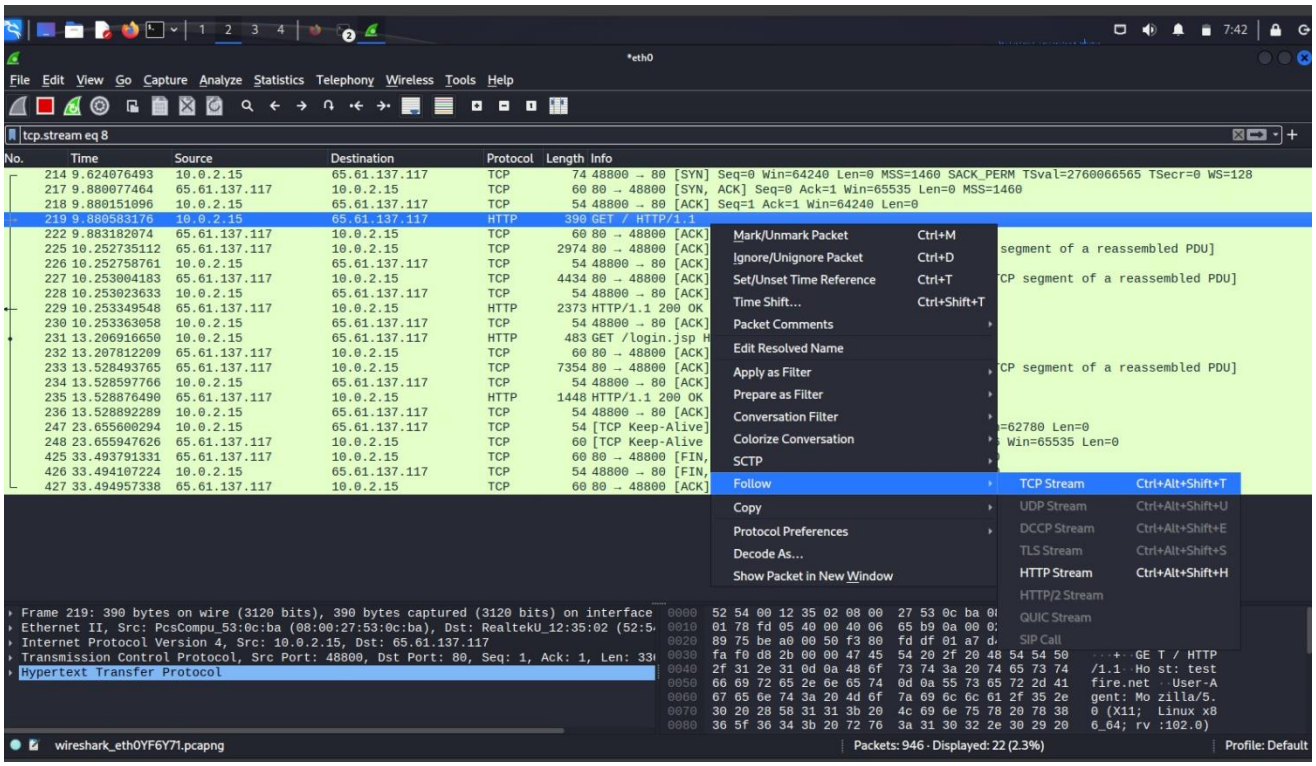
Now we are visiting a testing website to create traffic



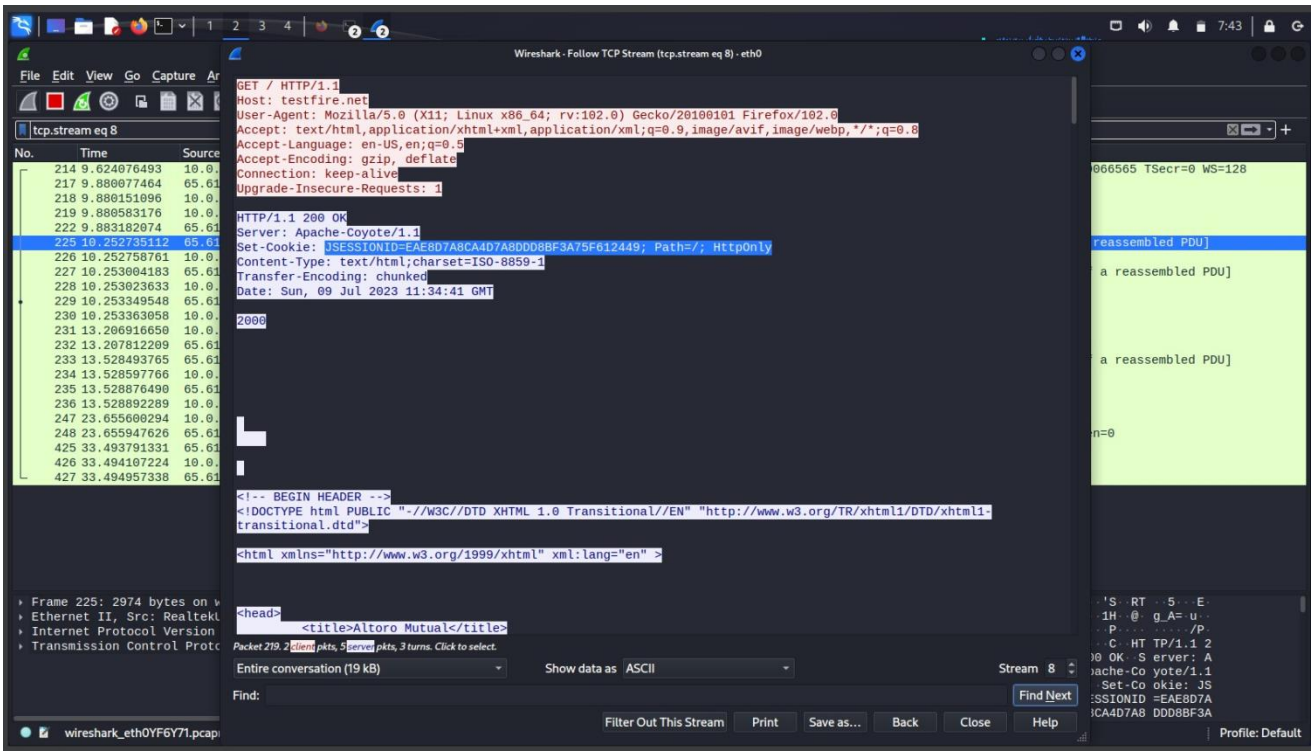


Now we are finding the ip address of the test website so that we can locate it in the traffic by running **ip.addr == 65.61.137.117** in the search bar to filter them out from other traffic.

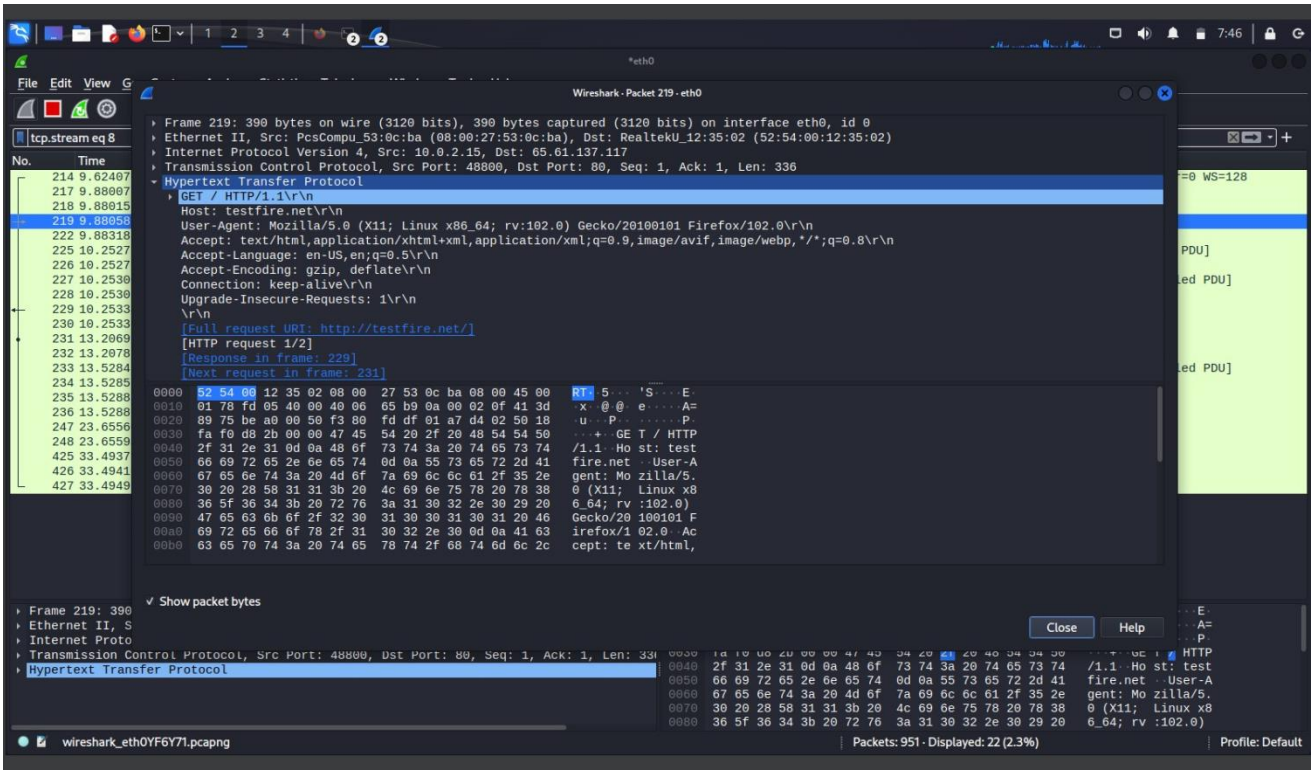




We should follow TCP stream in HTTP protocol. We have to do this for every signal in the traffic which has HTTP protocol to find sensitive information.



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

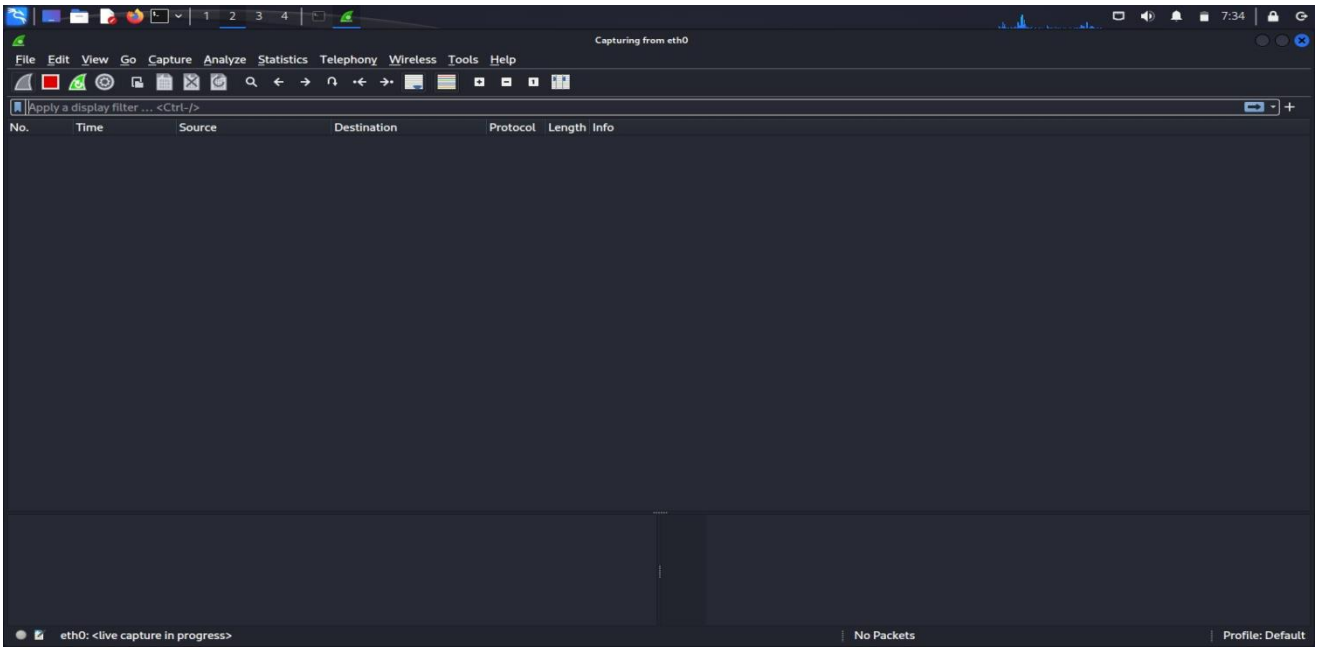
.

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.



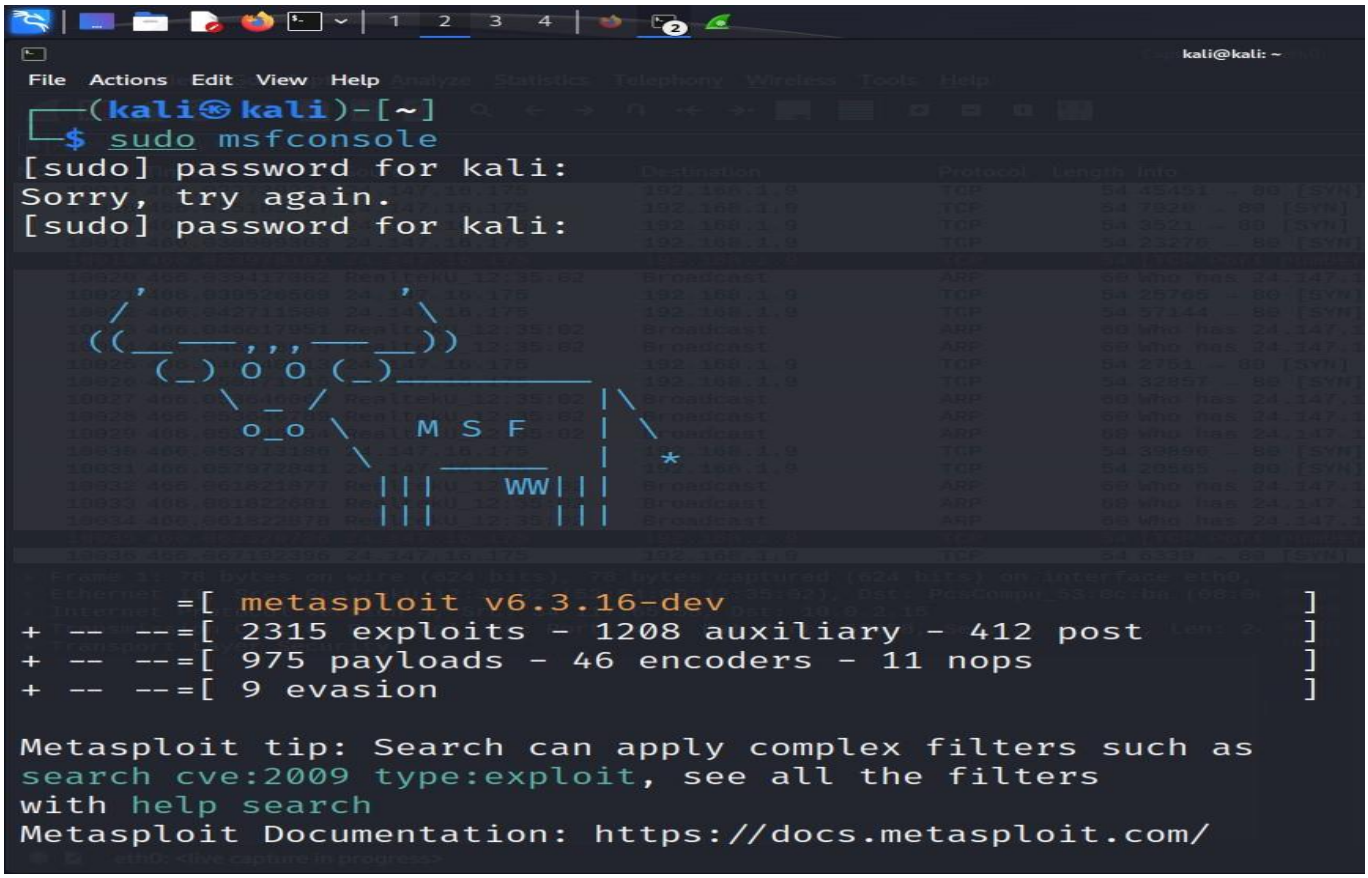
First when we open wireshark we can see that there is no traffic at all

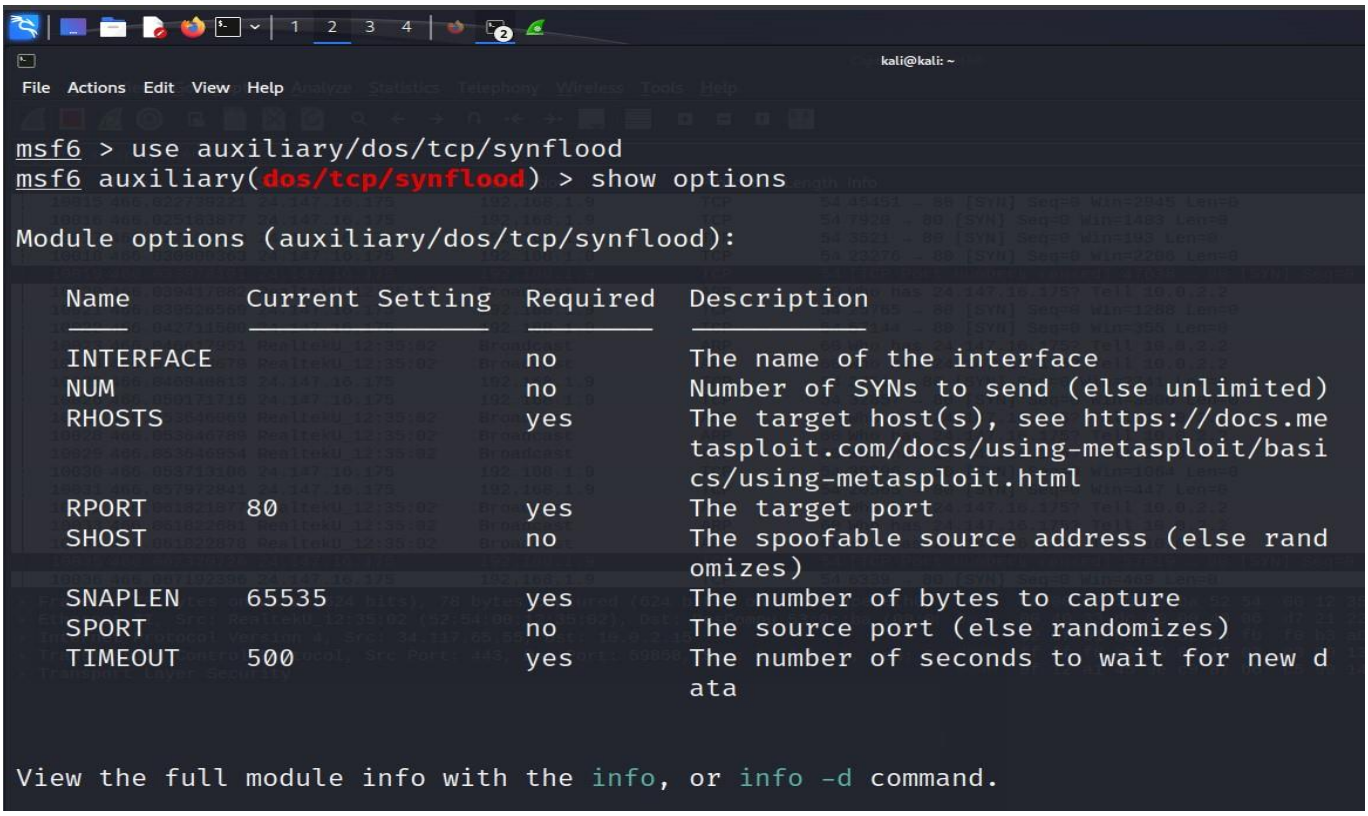
As conducting DoS attacks is illegal and unethical. Engaging in such activities can lead to severe legal consequences. Understanding the principles and techniques behind DoS attacks can help organizations implement effective countermeasures to protect their networks and services. I'm conducting on my 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:d361%20
   IPv4 Address. . . . . . . . . . . : 192.168.1.9
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::1%20
                                       192.168.1.1

C:\Users\iragh>
```

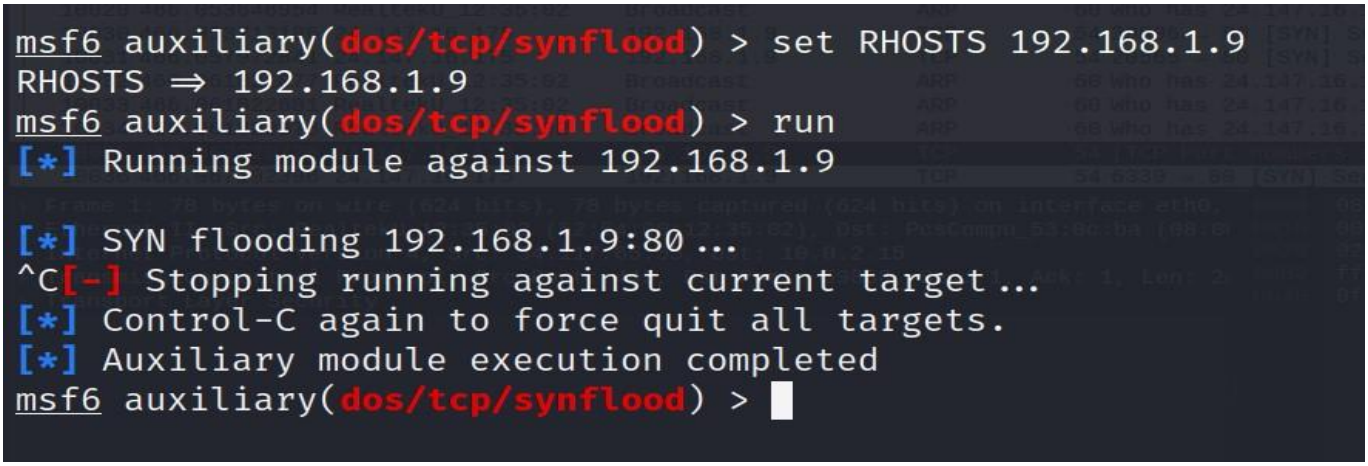
For finding the configurations of my computer on which we are practicing the DOS attack.





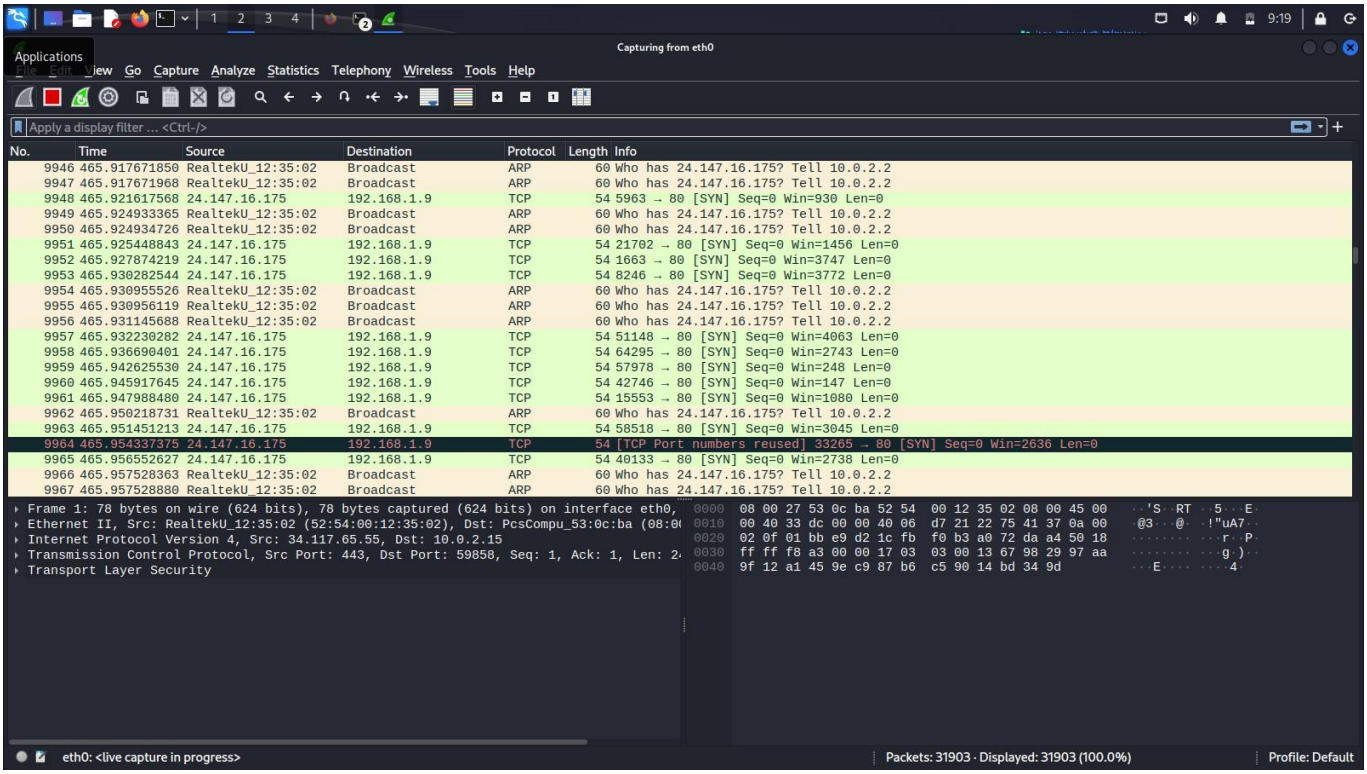
The module auxiliary/dos/tcp/syncflood is a component in the Metasploit Framework, which is a popular open-source penetration testing framework. This module is designed to launch a TCP SYN flood attack against a target system.

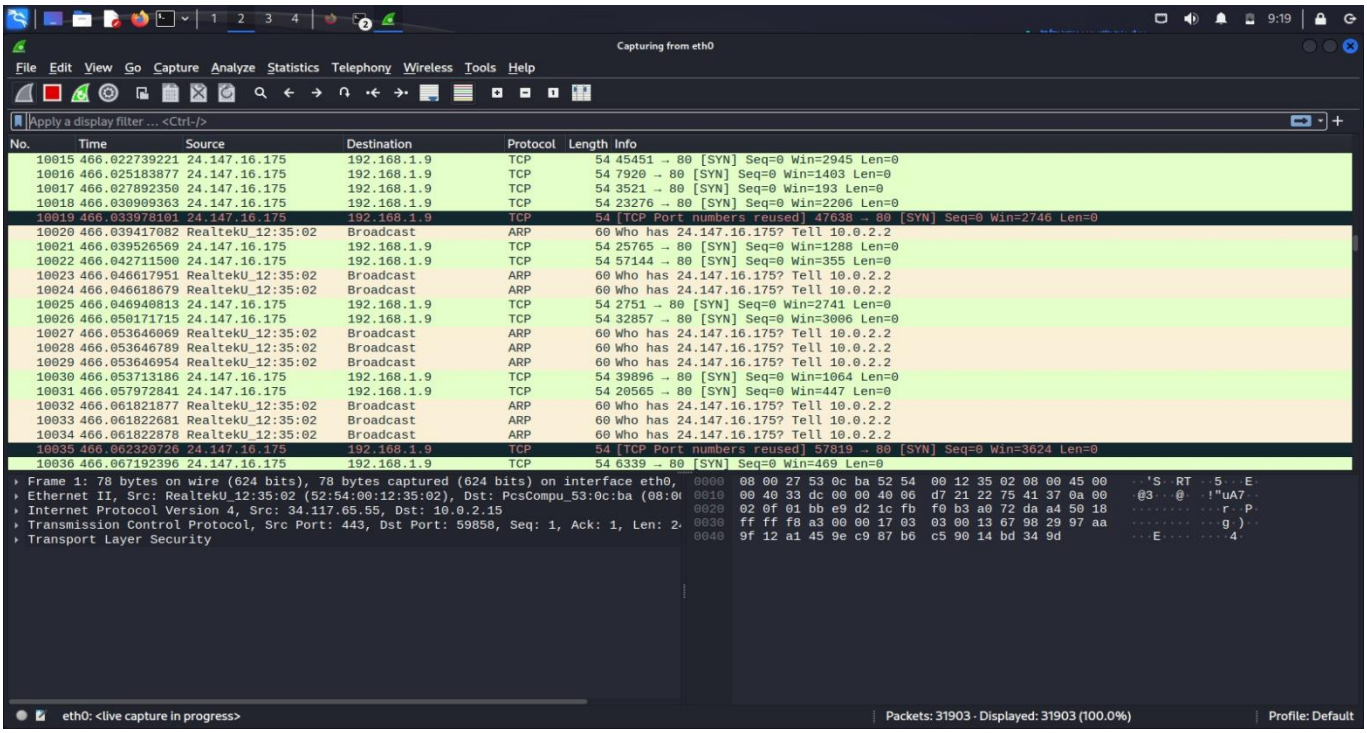
TCP SYN flood is a type of Denial-of-Service (DoS) attack that targets the TCP three-way handshake process. In this attack, the attacker sends a flood of TCP SYN packets to the target system, consuming its resources and overwhelming its ability to respond to legitimate connection requests.



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.





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) >
```

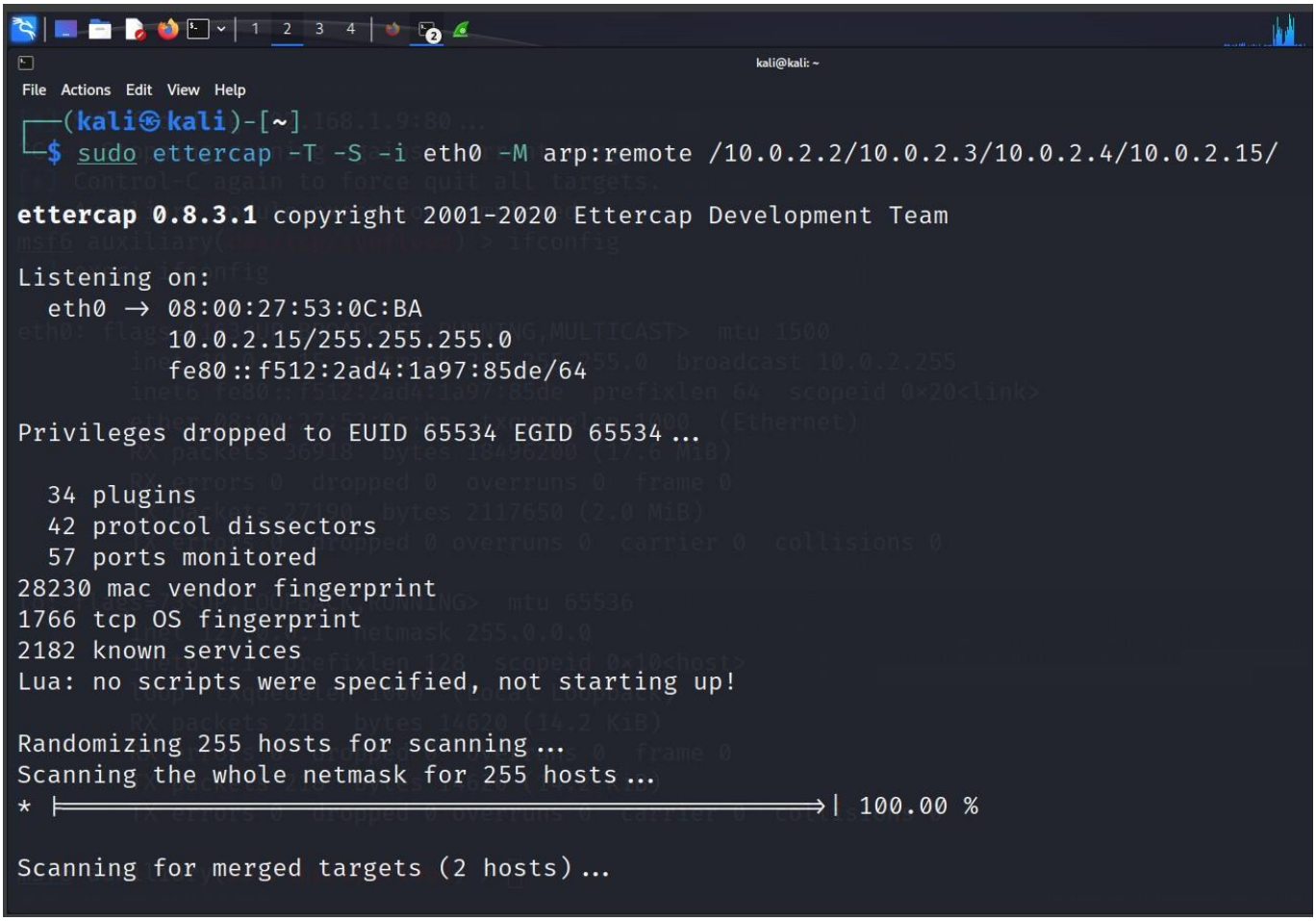
To find the configuration of our linux system simply use the following command:
ifconfig

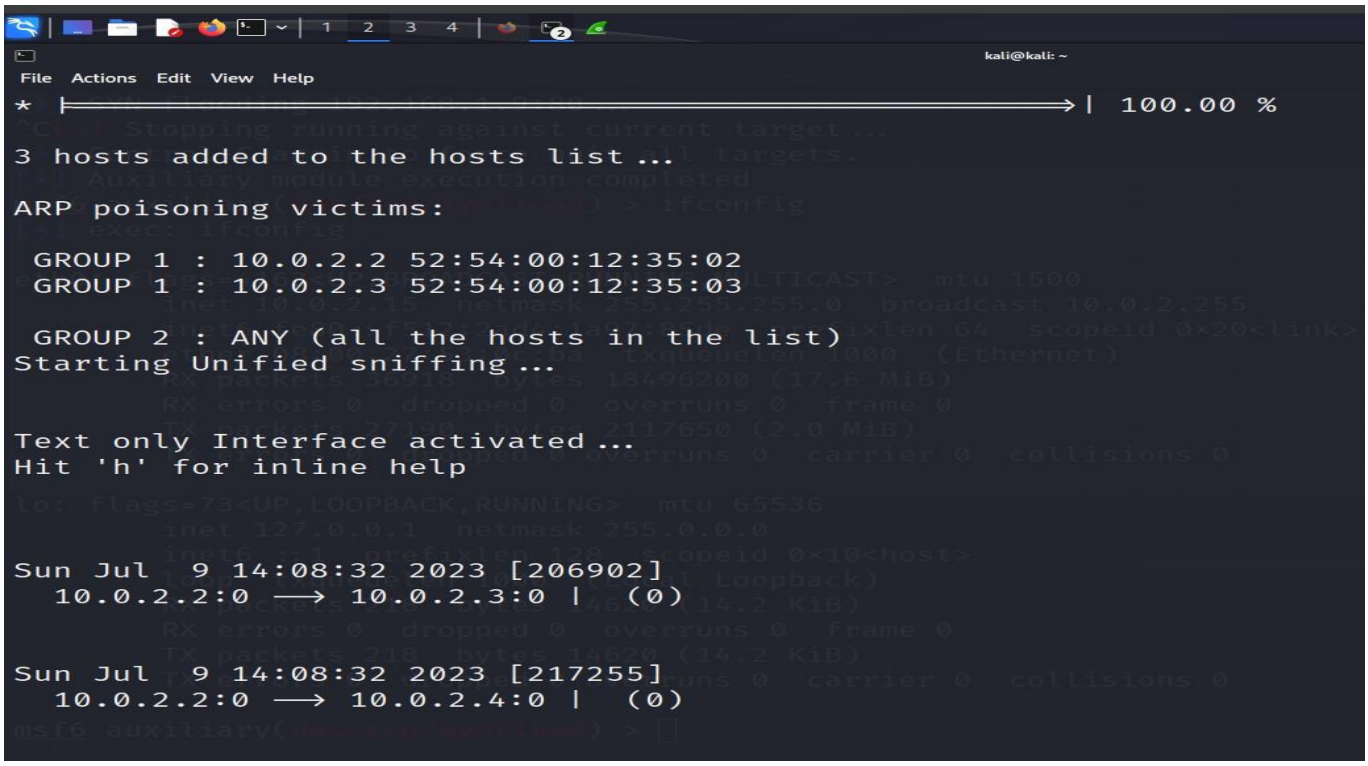
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sn 10.0.2.15/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-09 13:56 EDT
Nmap scan report for 10.0.2.2
Host is up (0.00059s latency).
MAC Address: 52:54:00:12:35:02 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.00058s latency).
MAC Address: 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
```

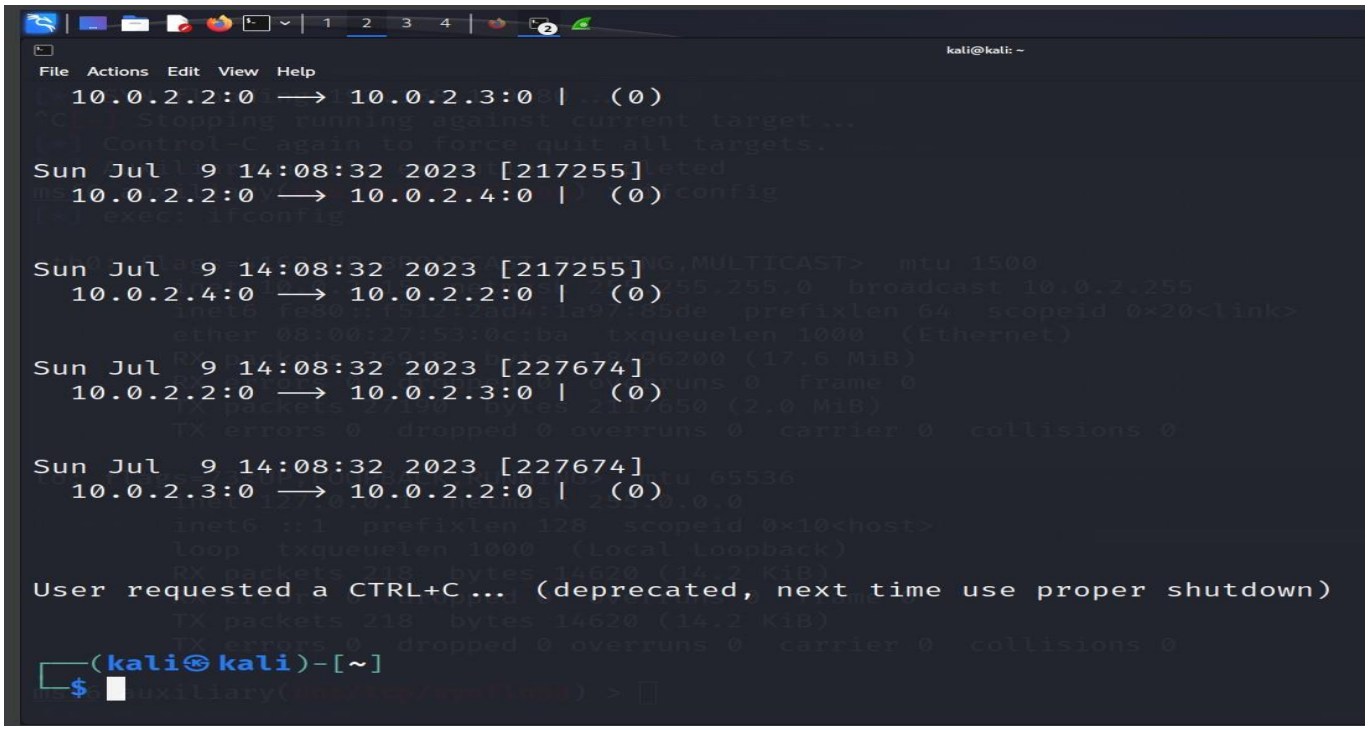
The command sudo nmap -sn 10.0.2.15/24 is used to perform a ping scan (also known as a "host discovery" scan) on a range of IP addresses within the 10.0.2.0/24 subnet.

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.







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.