

# EXPERIMENT 1.A

**Name: Shivam Dixit**

**TE COMP B**

# Roll no: 70

**Aim**: Understand the use of network reconnaissance tools like WHOIS, dig, traceroute, ns lookup to gather information about network and domain registrars.

# Theory:

1. **WHOIS:**

[Whois](https://www.domaintools.com/support/key-terms-definitions/#whois) is a widely used Internet record listing that identifies who owns a domain and how to get in contact with them. The Internet Corporation for Assigned Names and Numbers ([ICANN](https://www.domaintools.com/support/key-terms-definitions/#icann)) regulates domain name registration and ownership. Whois records have proven to be extremely useful and have developed into an essential resource for maintaining the integrity of the domain name registration and website ownership process.

# Dig:

Dig is a [network administration](https://en.wikipedia.org/wiki/Network_administration) [command-line](https://en.wikipedia.org/wiki/Command-line) tool for querying the [Domain Name](https://en.wikipedia.org/wiki/Domain_Name_System) [System](https://en.wikipedia.org/wiki/Domain_Name_System) (DNS).dig is useful for network troubleshooting and for educational purposes. It can operate based on command-line options and flag arguments or in batch mode by reading requests from an operating system file. When a specific [name server](https://en.wikipedia.org/wiki/Name_server) is not specified in the command invocation, it uses the operating system's default resolver, usually configured in the file [resolv.conf.](https://en.wikipedia.org/wiki/Resolv.conf) Without any arguments, it queries the [DNS root](https://en.wikipedia.org/wiki/DNS_root_zone) [zone](https://en.wikipedia.org/wiki/DNS_root_zone).

# Traceroute:

The traceroute command is used to determine the path between two connections. Often a connection to another device will have to go through multiple routers. The traceroute command will return the names or IP addresses of all the routers between two devices. This also allows you to see where a packet may be misguided.

# Nslookup:

Nslookup (stands for “Name Server Lookup”) is a useful command for getting information from the DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS-related problems.

# Conclusion:

Thus, we have successfully seen how this network commands function and their information with respect to their working and peculiarities.

For Faculty use

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correction Parameters** | **Formative Assessment [40%]** | **Timely completion of Practical [ 40%]** | **Attendance / Learning Attitude [20%]** |  |
| **Marks Obtained** |  |  |  |



# CSS EXPERIMENT 1.B

**Name: Shivam Dixit**

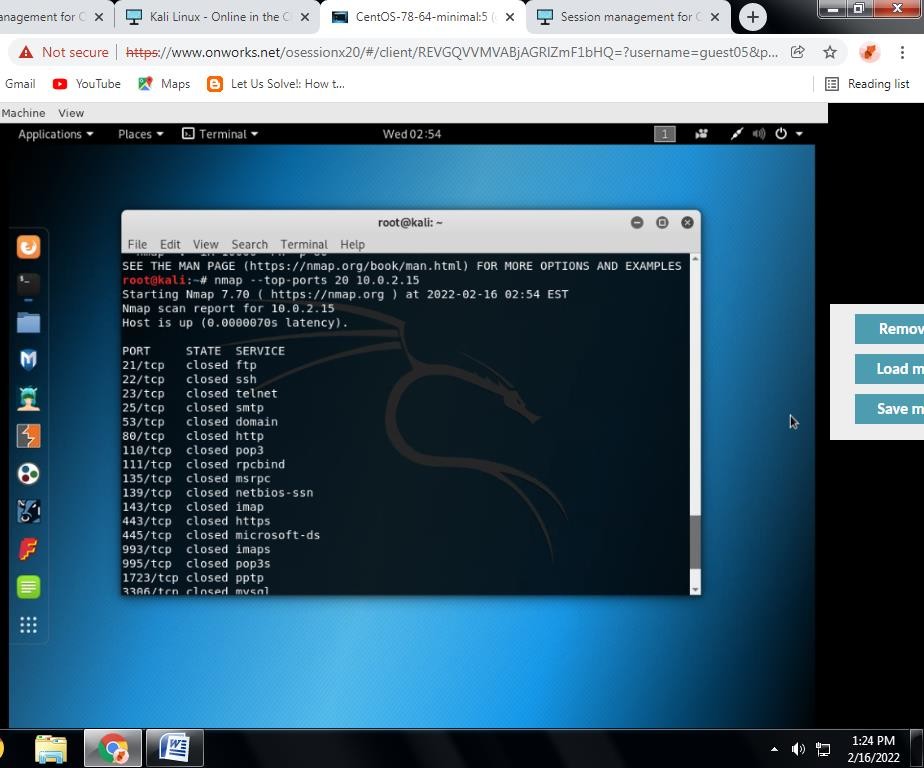
**TE COMP B**

# Roll no: 70

**Aim:** To Analyze the tool nmap and use it with different options to scan open ports, perform OS fingerprinting, do a ping scan, TCP port scan, UDP port scan, Xmas scan, etc.

# Nmap

* Nmap is a network scanner
* Nmap is used to discover hosts and services on a computer network by sending packets and analyzing the responses.
* nmap provides several features for probing computer networks, including host discovery and service and operating system detection.

1. **Port Scanning**

It automatically scans a number of the most ‘popular’ ports for a host. You can run this command using:

nmap –top-ports <ip>

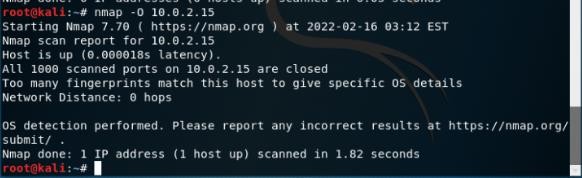
nmap --top-ports 20 <ip>

Replace the “20” with the number of ports to scan, and Nmap quickly scans that many ports. It returns a concise output that details the status of the most common ports, and this lets you quickly see whether you have any unnecessarily open ports.

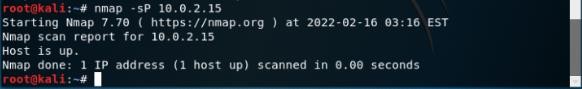
# OS fingerprinting

OS scanning is one of the most powerful features of Nmap. When using this type of scan, Nmap sends TCP and UDP packets to a particular port and then analyzes its response. It compares this response to a database of 2600 operating systems, and return information on the OS (and version) of a host.

To run an OS scan, use the following command: # nmap -O <target IP>



# ping scan



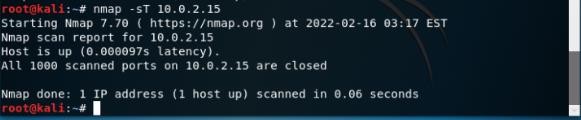
As mentioned above, a ping scan returns information on every active IP on your network. You can execute a ping scan using this command:

# nmap -sP 10.0.2.15

1. **TCP port scan**

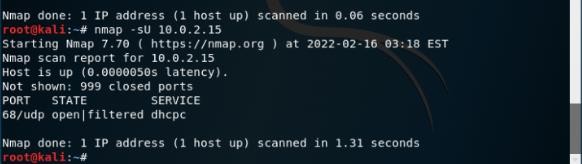
One of the best things is the Nmap command to check open ports, and the second-best thing about Nmap is its power to work with TCP and UDP without any hiccups. Several services are limited to just TCP, but people understand the advantage of scanning UDP-based services. Here are examples of both these services that are allowed by Nmap.

Tcp port scan nmap -sT 10.0.2.15



# UDP port scan

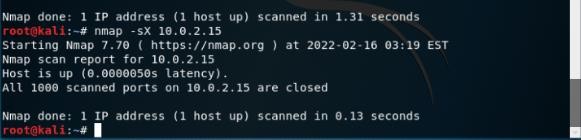
nmap -sU 10.0.2.15



# Xmas scan

It Sets the FIN, PSH, and URG flags, lighting the packet up like a Christmas tree.

*nmap -sX 10.0.2.15*



As we can see in the above picture when we send a packet with FIN/PSH/URG flag to set and send it to the destination if we don’t get any response from Destination we will know that Port is OPEN. if we get RST in return then we know that Port is Closed.

# Conclusion:

Successfully studied and implemented various nmap tool commands in kalli Linux. Also, we have seen how these tools can give us some vital security information about wired/wireless network connections.

For Faculty use

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correction Parameters** | **Formative Assessment [40%]** | **Timely completion of Practical [ 40%]** | **Attendance / Learning Attitude [20%]** |  |
| **Marks Obtained** |  |  |  |





# EXPERIMENT 2

**Name: Shivam Dixit**

**TE COMP B**

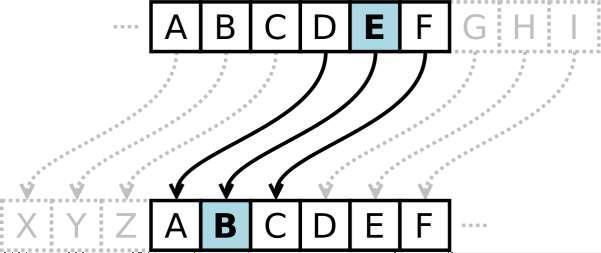
# Roll no: 70

**Aim:** Implement and design the product cipher using Substitution and Transposition ciphers.

# Theory:

A Caesar cipher is a simple method of encoding messages. Caesar ciphers use a substitution method where letters in the alphabet are shifted by some fixed number of spaces to yield an encoding alphabet. A Caesar cipher with a shift of 1 would encode an A as a B, an M as an N, and a Z as an A, and so on.

It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on.



To encrypt a message, enter the message in the Plaintext textbox, specify the shift, and click Encrypt. To decrypt a message, enter the message in the Ciphertext textbox, specify the shift, and click Decrypt.

Transposition cipher:

In cryptography, a transposition cipher is a method of encryption by which the positions held by units of plaintext (which are commonly characters or groups of characters) are shifted according to a regular system, so that the ciphertext constitutes a permutation of the plaintext.

**Conclusion:**

Successfully implemented and designed the product cipher using Substitution and Transposition ciphers.

For Faculty Use

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correction** | **Formative** | **Timely** | **Attendance /** |  |
| **Parameter** | **Assessmen** | **completion of** | **Learning** |
| **s** | **t[40%]** | **Practical [ 40%]** | **Attitude** |
|  |  |  | **[20%]** |
| **Marks** |  |  |  |
| **Obtaine** |
| **d** |



# EXPERIMENT 3

**Name: Shivam Dixit**

**TE COMP B**

# Roll no: 70

**Aim**: To Implement RSA Algorithm

**Theory**: RSA algorithm is asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e., **Public Key** and **Private Key.** As the name describes that the Public Key is given to everyone and Private key is kept private.

# An example of asymmetric cryptography:

* 1. A client (for example browser) sends its public key to the server and requests for some data.
  2. The server encrypts the data using client’s public key and sends the encrypted data.
  3. Client receives this data and decrypts it.

Since this is asymmetric, nobody else except browser can decrypt the data even if a third party has public key of browser.

The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So, if somebody can factorize the large number, the private key is compromised. Therefore, encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024-bit keys could be broken in the near future. But till now it seems to be an infeasible task.

**Conclusion:**

It is concluded that, while establishing RSA key pairs, usage keys and general-purposekeys are integrated. In usage RSA keys, two key pairs are used for encryption and signatures. In General-purpose key, one single pair is used for both encryption and signature.

For Faculty Use

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correction Parameters** | **Formative Assessment [40%]** | **Timely completion of Practical [ 40%]** | **Attendance / Learning Attitude [20%]** |  |
| **Marks**  **Obtained** |  |  |  |



# EXPERIMENT 4

**Name: Shivam Dixit**

**TE COMP B**

# Roll no: 70

**Aim**: To Implement Diffie-Hellman Algorithm



# Theory:

Diffie Hellman (DH) key exchange algorithm is a method for securely exchanging cryptographic keys over a public communications channel. Keys are not actually exchanged – they are jointly derived. It is named after their inventors Whitfield Diffie and Martin Hellman.

If Alice and Bob wish to communicate with each other, they first agree between them a large prime number p, and a generator (or base) g (where 0 < g < p).

Alice chooses a secret integer a (her private key) and then calculates g^a mod p (which is her public key). Bob chooses his private key b, and calculates his public key in the same way.

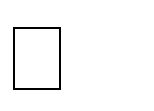
Bob knows b and g^a, so he can calculate (g^a)^b mod p = g^ab mod p. Therefore both Alice and Bob know a shared secret g^ab mod p. An eavesdropper Eve who was listening in on the communication knows p, g, Alice’s public key (g^a mod p) and Bob’s public key (g^b mod p). She is unable to calculate the shared secret from these values.

In static-static mode, both Alice and Bob retain their private/public keys over multiple communications. Therefore the resulting shared secret will be the same every time. In ephemeral-static mode one party will generate a new private/public key every time, thus a new shared secret will be generated.

Advantages:

The sender and receiver don’t need any prior knowledge of each other. Once the keys are exchanged, the communication of data can be done through aninsecure channel.



The sharing of the secret key is safe.

Disadvantages:

The algorithm cannot be sued for any asymmetric key exchange. Similarly, it cannot be used [for signing digital signatures.](https://www.educba.com/digital-signature-cryptography/)



Since it doesn’t authenticate any party in the transmission, the Diffie Hellman key exchange is susceptible to a [man-in-the-middle attack.](https://www.educba.com/what-is-man-in-the-middle-attack/)

# Conclusion:

The Diffie Hellman key Exchange has proved to be a useful key exchange system due to its advantages. While it is really tough for someone snooping the network to decrypt the data and get the keys, it is still possible if the numbers generated are not entirely random.

For Faculty use

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correction Parameters** | **Formative Assessment [40%]** | **Timely completion of Practical [ 40%]** | **Attendance / Learning Attitude [20%]** |  |
| **Marks**  **Obtained** |  |  |  |



# EXPERIMENT 5

**Name: Shivam Dixit**

**TE COMP B**

**Roll no: 70**

**Aim:** Study of packet sniffer tools: Wireshark:

1. Download and install Wireshark and capture icmp, tcp, and http packets in promiscuous mode.
2. Explore how the packets can be traced based on different filters

# Theory:

Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format. Wireshark includes filters, color-coding and other featuresthat let you dig deep into network traffic and inspect individual packets.

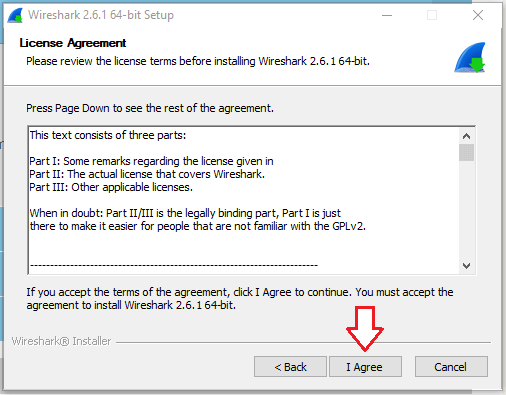
# Features of Wireshark:

* Available for UNIX and Windows.
* Capture live packet data from a network interface.
* Open files containing packet data captured with tcpdump/WinDump, Wireshark, and a
* number of other packet capture programs.
* Import packets from text files containing hex dumps of packet data.
* Display packets with very detailed protocol information.
* Export some or all packets in a number of capture file formats.
* Filter packets on many criteria.
* Search for packets on many criteria.
* Colorize packet display based on filters.
* Create various statistics.

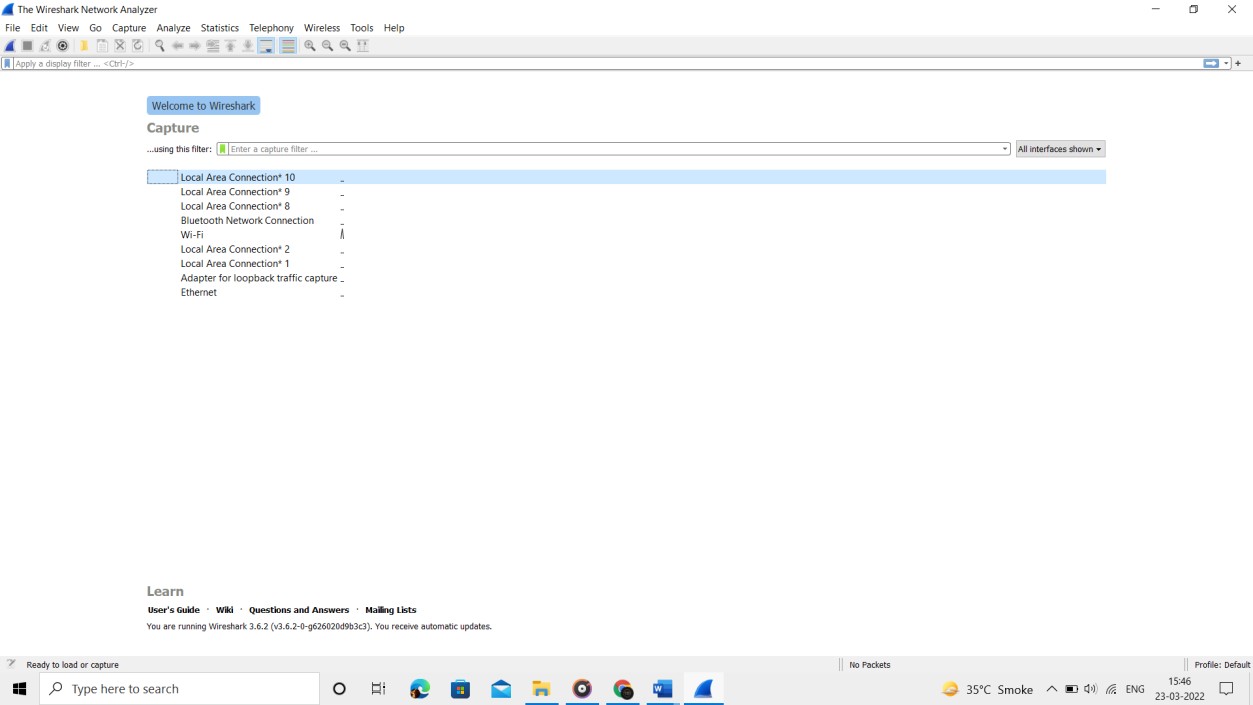
# Capturing Packets

After downloading and installing wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface. For example, if you want to capture traffic on the wireless network, click your wireless interface. You can configure advanced features by clicking Capture Options.

# Installation of Wireshark:

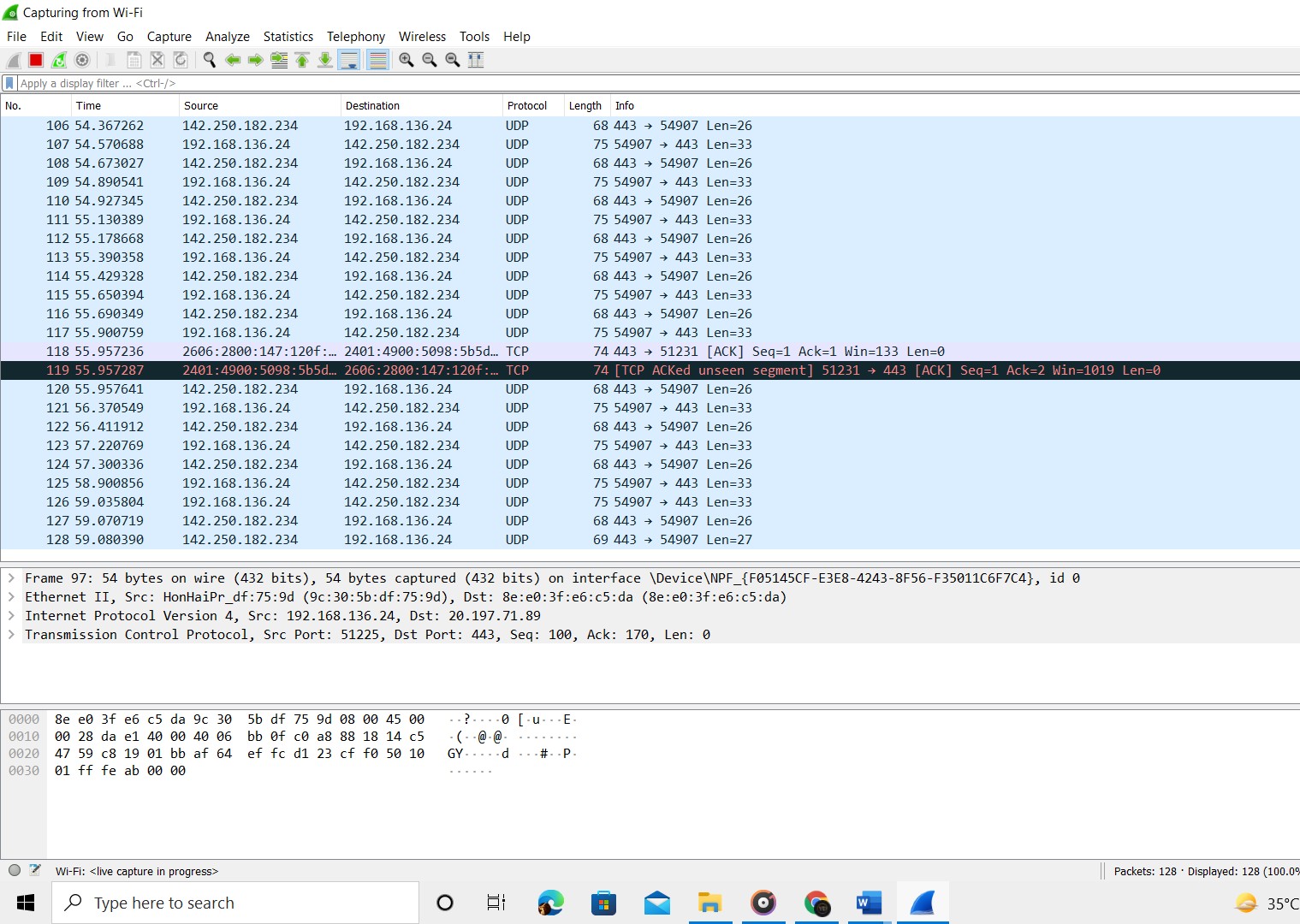


After downloading and installing wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface. For example, if you want to capture traffic on the wireless network, click your wireless interface. You can configure advanced features by clicking Capture Options.

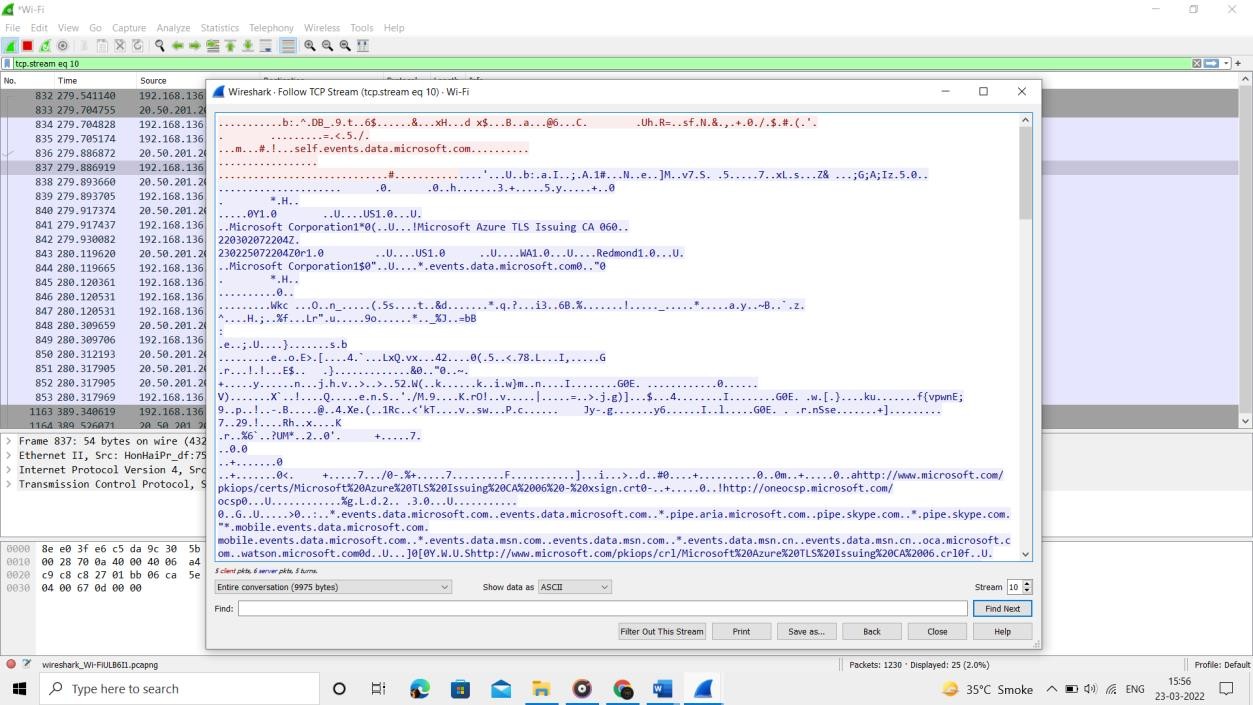
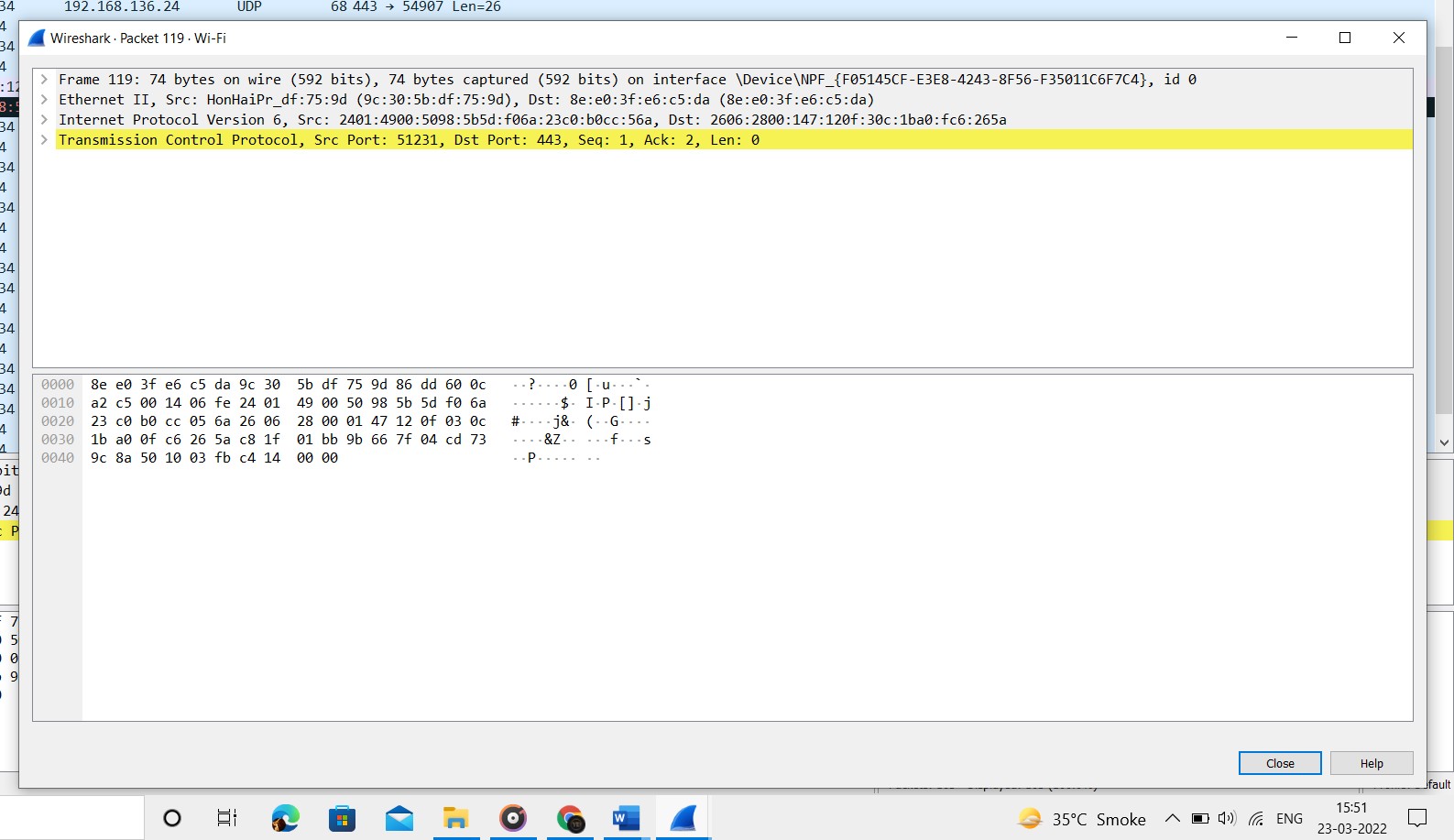


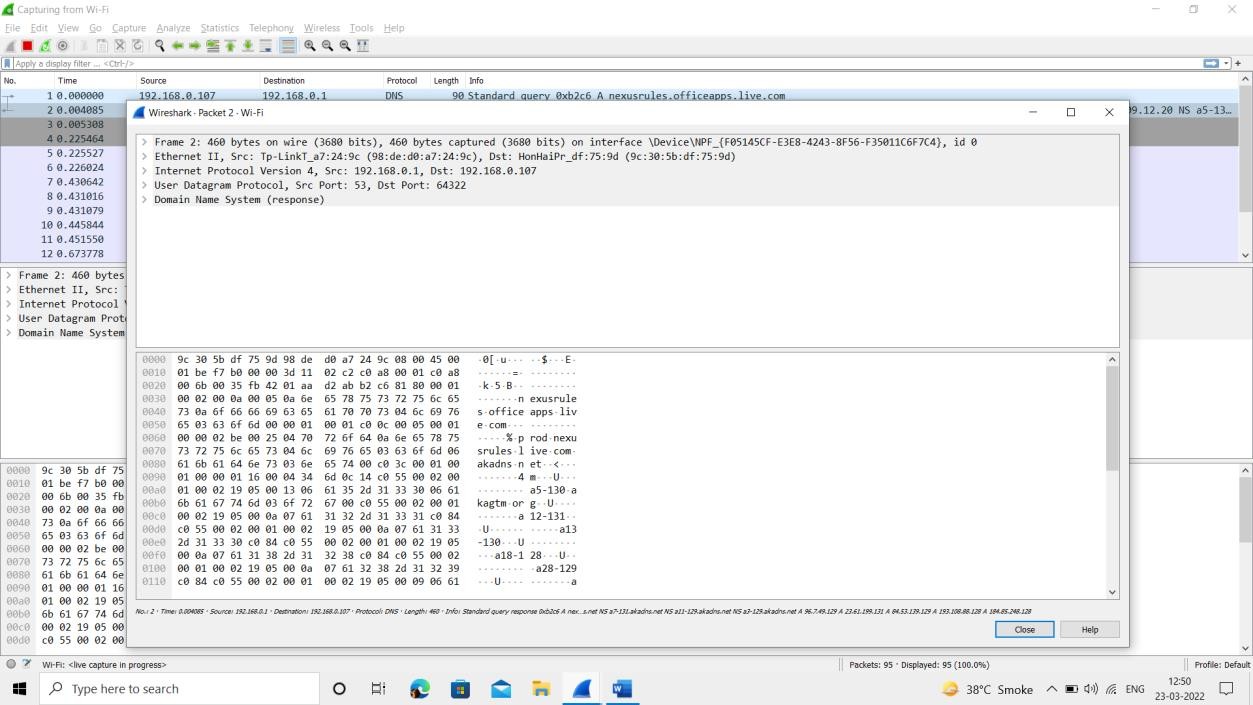
As soon as you click the interface‘s name, you‘ll see the packets start to appear in real time.

Wireshark captures each packet sent to or from your system. If you‘re capturing on a wireless interface and have promiscuous mode enabled in your capture options, you‘ll also see other the other packets on the network

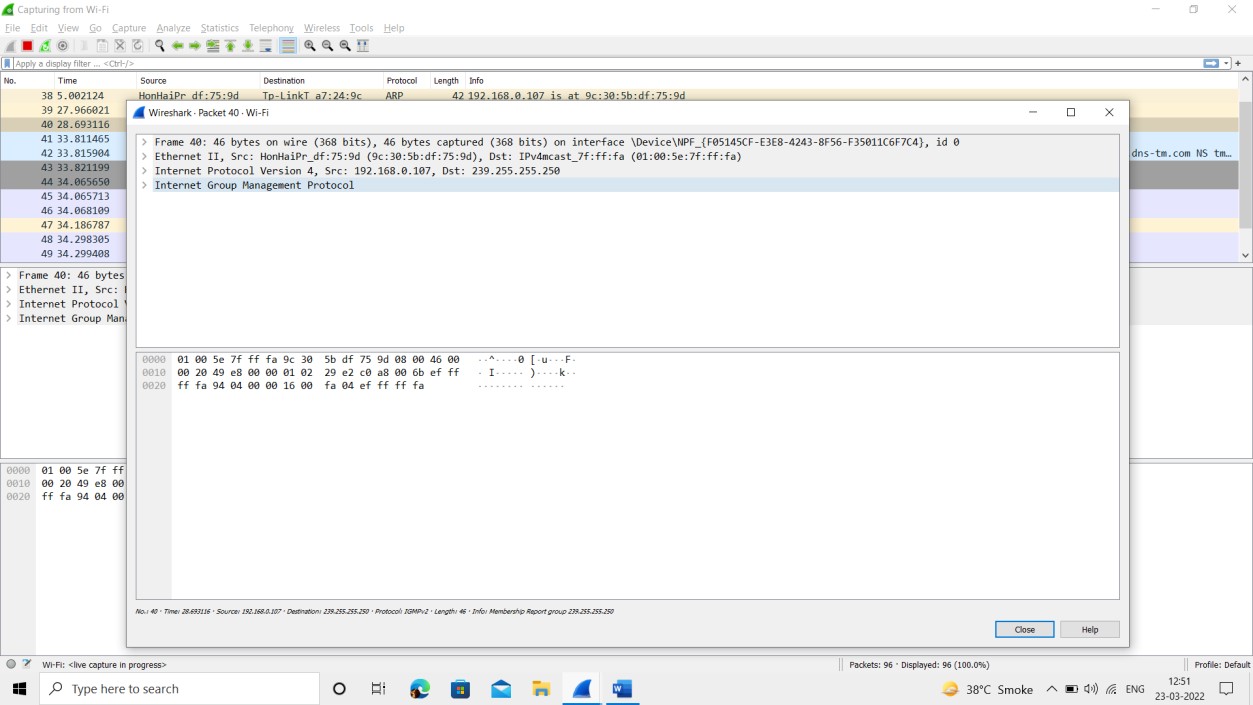


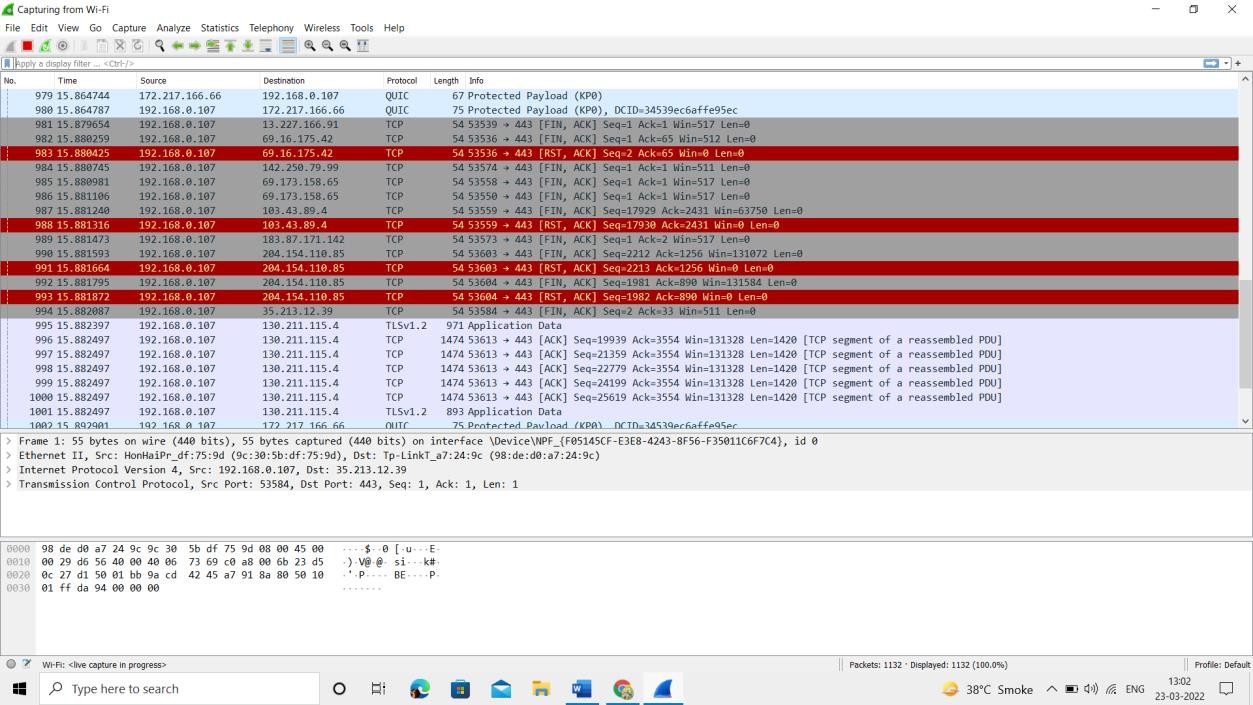
# Capturing TCP





**ICMP**





**Conclusion:**

Wireshark installation and network traffic analysis using packet sniffing is done. Detailed information about packets are explored by applying filters**.**

For Faculty use

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correction Parameters** | **Formative Assessment [40%]** | **Timely completion of Practical [ 40%]** | **Attendance / Learning Attitude [20%]** |  |
| **Marks**  **Obtained** |  |  |  |



# EXPERIMENT 6

**Name: Shivam Dixit**

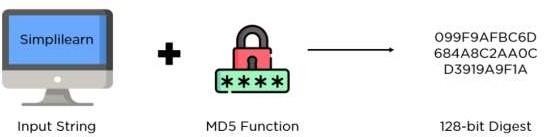
**TE COMP B**

**Roll no: 70**

**Aim:** Analyze and implement MD5 and SHA1 cryptographic algorithm.

# Theory:

Hashing consists of converting a general string of information into an intricate piece of data. This is done to scramble the data so that it completely transforms the original value, making the hashed value utterly different from the original. Hashing uses a hash function to convert standard data into an unrecognizable format. These hash functions are a set of mathematical calculations that transform the original information into their hashed values, known as the hash digest or digest in general. The digest size is always the same for a particular hash function like MD5 or SHA1, irrespective of input size.



MD5 (Message Digest Method 5) is a cryptographic hash algorithm used to generate a 128 - bit digest from a string of any length. It represents the digests as 32 digit hexadecimal numbers. The digest size is always 128 bits, and thanks to hashing function guidelines, a minor change in the input string generate a drastically different digest. This is essential to prevent similar hash generation as much as possible, also known as a hash collision.

# Functions associated:

encode (): Converts the string into bytes to be acceptable by hash function. digest()

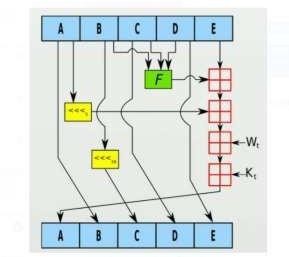
: Returns the encoded data in byte format.

hexdigest () : Returns the encoded data in hexadecimal format.

SHA-1 produces a 160-bit hash value or message digests from the inputted data (data that requires encryption), which resembles the hash value of the MD5 algorithm. It uses 80 rounds of cryptographic operations to encrypt and secure a data object. Some of the protocols that use SHA-1 include:

1. Transport Layer Security (TLS)
2. Secure Sockets Layer (SSL)
3. Pretty Good Privacy (PGP)
4. Secure Shell (SSH)
5. Secure/Multipurpose Internet Mail Extensions (S/MIME)
6. Internet Protocol Security (IPSec)

SHA-1 is commonly used in cryptographic applications and environments where the need for data integrity is high. It is also used to index hash functions and identify data corruption and checksum errors.



# Advantages of MD5:

Easy to Compare: Unlike the latest hash algorithm families, a 32 digit digest is relatively easier to compare when verifying the digests.

Storing Passwords: Passwords need not be stored in plaintext format, making them accessible for hackers and malicious actors. When using digests, the database also gets a boost since the size of all hash values will be the same.

Low Resource: A relatively low memory footprint is necessary to integrate multiple services into the same framework without a CPU overhead.

Integrity Check: You can monitor file corruption by comparing hash values before and after transit. Once the hashes match, file integrity checks are valid, and it avoids data corruption.

# Disadvantages of MD5:

When compared to other algorithms like the SHA algorithm, MD5 is comparatively slow. It is possible to construct the same hash function for two distinct inputs using MD5.



MD5 is less secure when compared to the SHA algorithm since MD5 is more vulnerable

to collision attacks.

# Application:

Password Verification

It is common to store user credentials of websites in a hashed format to prevent third parties from reading the passwords. Since hash functions always provide the same output for the same input, comparing password hashes is much more private.

The entire process is as follows:

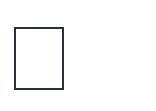
* 1. User signs up to the website with a new password
  2. It passes the password through a hash function and stores the digest on the server
  3. When a user tries to log in, they enter the password again
  4. It passes the entered password through the hash function again to generate a digest
  5. If the newly developed digest matches the one on the server, the login is verified.

Integrity Verification

Some files can be checked for data corruption using hash functions. Like the above scenario, hash functions will always give the same output for similar input, irrespective of iteration parameters. The entire process follows this order:

1. A user uploads a file on the internet
2. It also uploads the hash digest along with the file
3. When a user downloads the file, they recalculate the hash digest
4. If the digest matches the original hash value, file integrity is maintained

**Algorithm:**

MD5

A 512-bit string is divided into 16 words of 32 bits each using the MD5 message- digest hashing method. MD5 generates a 128-bit message digest as a result of the operation.

There are four steps involved in producing a message digest:

Appending padding bits to the original message.

1. Appending length bits.
2. Initializing MD or Message digest buffer.
3. Processing of messages in 16-word blocks to produce the final output or result.

# Conclusion:

Successfully understood and implemented menu driven MD5 and Sha-1 hash algorithm using python library hashlib and found equivalent hash, hexadecimal and Byte value for a string input.

For faculty use

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correction Parameters** | **Formative**  **Assessment [40%]** | **Timely completion of Practical [ 40%]** | **Attendance /**  **Learning Attitude [20%]** |  |
| **Marks Obtained** |  |  |  |





# Title:

**EXPERIMENT 9 CASE STUDY**

# Name: Shivam Dixit Class: TE COMP B

**Roll no: 70**

Setting up firewall using iptables in Kali Linux.

# Introduction:

Kali Linux is funded and maintained by Offensive Security, an information training company. It is a Debian-based Linux distribution built with the aim of advanced penetration testing and security auditing. You may ask what the Debian standard is; it is an entire command-line system without an x11 or GUI environment. It is just a primary server. If you don’t use it with a landline connection, you can only use it for learning the command line. It is equipped with a tonne of tools installed with it that makes an ethical hacker can go on a war with these tools in his arsenal.

Kali Linux is packed with essential tools for information security tasks, penetration testing, computer forensics, reverse engineering, and much more. As Kali Linux is just an operating software, it is not illegal by itself. But when someone uses it for hacking, it is considered illegal if someone uses it to learn, teach, or understand the intricacies as it is licensed for download.

Kali Linux contains several hundred tools targeted towards various information security tasks, such as Penetration Testing, Security Research, Computer Forensics and Reverse Engineering. Kali Linux is a multi-platform solution, accessible and freely available to information security professionals and hobbyists.

Advantages:

* It is free: It is free for download.
* Multi-lingual support: Although most applications and tools are primarily written in English, Kali Linux provides multi-lingual support. That opens opportunities for people to use these resources in their local language and use it for their customizable purpose.
* Customizable: The developers at Kali Linux have been very liberal while developing this and have left opportunities for customization for fellow developers to come on board and modify it as per their likings.
* Open source: As this belongs to the Linux family, it is available on an open-source platform. The entire development tree, along with codes, is known for viewing and modifying on Git.
* A plethora of tools available: Kali Linux comes equipped with more than 600 different devices.

4 Reasons Why Hackers Use Kali Linux

* Requires Minimal System Resources
* Kali Linux Is Good for Beginners
* Kali Linux Is Legal
* Kali Linux Ships With Hundreds of Testing Tools

A Firewall is a network security device that monitors and filters incoming and outgoing network traffic based on an organization’s previously established security policies. At its most basic, a firewall is essentially the barrier that sits between a private internal network and the public Internet. A firewall’s main purpose is to allow non-threatening traffic in and to keep dangerous traffic out.

Firewalls have existed since the late 1980’s and started out as packet filters, which were networks set up to examine packets, or bytes, transferred between computers. Though packet filtering firewalls are still in use today, firewalls have come a long way as technology has developed throughout the decades.

# Background:

Iptables is an extremely flexible firewall utility built for Linux operating systems. Whether you’re a novice Linux geek or a system administrator, there’s probably some way that iptables can be a great use to you. Read on as we show you how to configure the most versatile Linux firewall. iptables is a command-line firewall utility that uses policy chains to allow or block traffic. When a connection tries to establish itself on your system, iptables looks for a rule in its list to match it to. If it doesn’t find one, it resorts to the default action.

* Types of chains: input, forward and output.
* Input – This chain is used to control the behavior for incoming connections. For example, if a user attempts to SSH into your PC/server, iptables will attempt to match the IP address and port to a rule in the input chain.
* Forward – This chain is used for incoming connections that aren’t actually being delivered locally. Think of a router – data is always being sent to it but rarely actually destined for the router itself; the data is just forwarded to its target. Unless you’re doing some kind of routing, NATing, or something else on your system that requires forwarding, you won’t even use this chain.
* Output – This chain is used for outgoing connections. For example, if you try to ping howtogeek.com, iptables will check its output chain to see what the rules are regarding ping and howtogeek.com before making a decision to allow or deny the connection attempt.

Connection-specific Responses - With your default chain policies configured, you can start adding rules to iptables so it knows what to do when it encounters a connection from or to a

particular IP address or port. In this guide, we’re going to go over the three most basic and commonly used “responses”. Accept – Allow the connection.

Drop – Drop the connection, act like it never happened. This is best if you don’t want the source to realize your system exists.

Reject – Don’t allow the connection, but send back an error. This is best if you don’t want a

particular source to connect to your system, but you want them to know that your firewall blocked them.

# Methodology:

**Adding rules**

By default, iptables does not have any rules defined. You can verify this yourself on a new server by typing the following command: **iptables -L**

iptables -A INPUT -i lo -j ACCEPT iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT iptables -A INPUT -p tcp -m tcp --dport 7822

-j ACCEPT iptables -A INPUT -j DROP

In all of these commands, the **-A** option instructs iptables to append the rule to the end of the specified chain (in this case, the **INPUT** chain). Let's step through each command:

* The first command permits all packets for the local loopback interface. Many programs use the loopback interface, so it is a good idea to accept packets on it.
* The second command uses the **-m** option to load the state module. This module determines and monitors a packet's state, which can be **NEW**, **ESTABLISHED**, or **RELATED**. In this rule, we accept incoming packets that belong to a connection that has already been established.
* The third command accepts incoming TCP connections on port 7822 (SSH).
* The last command drops (rejects) incoming packets that do not match any of the preceding rules.

**Inserting rules**

The set of rules we defined above is pretty limited. If SSH is the only incoming connection you want to allow, then you're all set. Most likely, though, you will need to add access to services as you configure your server.

However, if we just add a rule using the **-A** option shown above, it will be the last rule in the chain, right after our **DROP** rule. Because iptables works through rules in sequence, this means that it will never get to the new rule, because the packet will have already been dropped. Therefore, we need a way to insert new rules into the chain.

The **-I** option enables us to insert a new rule anywhere in the chain. Let's insert a rule that allows incoming TCP connections on port 80 (HTTP). We want the rule to come just before the **DROP** rule, which is currently the fourth rule in the chain:

This inserts our HTTP rule in the fourth line, and pushes the **DROP** rule down to the fifth line.

iptables -I INPUT 4 -p tcp -m tcp --dport 80 -j ACCEPT

**Deleting rules**

To delete a rule, use the **-D** option. You need to know the number of the rule you want to delete (just as you must know the number when you insert a rule). The following command demonstrates how to delete the fifth rule from the **INPUT** chain:

iptables -D INPUT 5

If you want to delete **all** of the rules at once, type the following command:

iptables -F

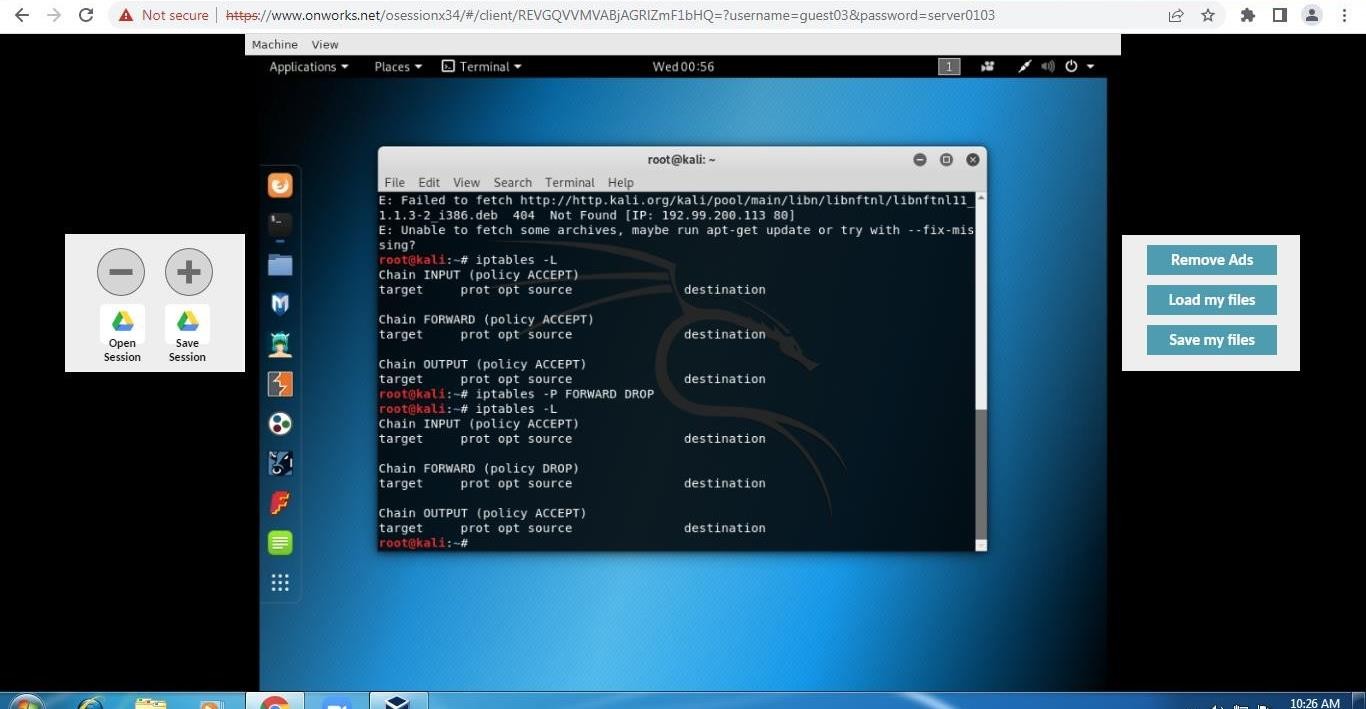
**Saving rules**

If you reboot the server now, all of the rules you defined will be erased. To maintain rules across system restarts, you must save them. The steps to do this depend on the Linux distribution you are running.

# Results:

1. **Displaying iptables:**

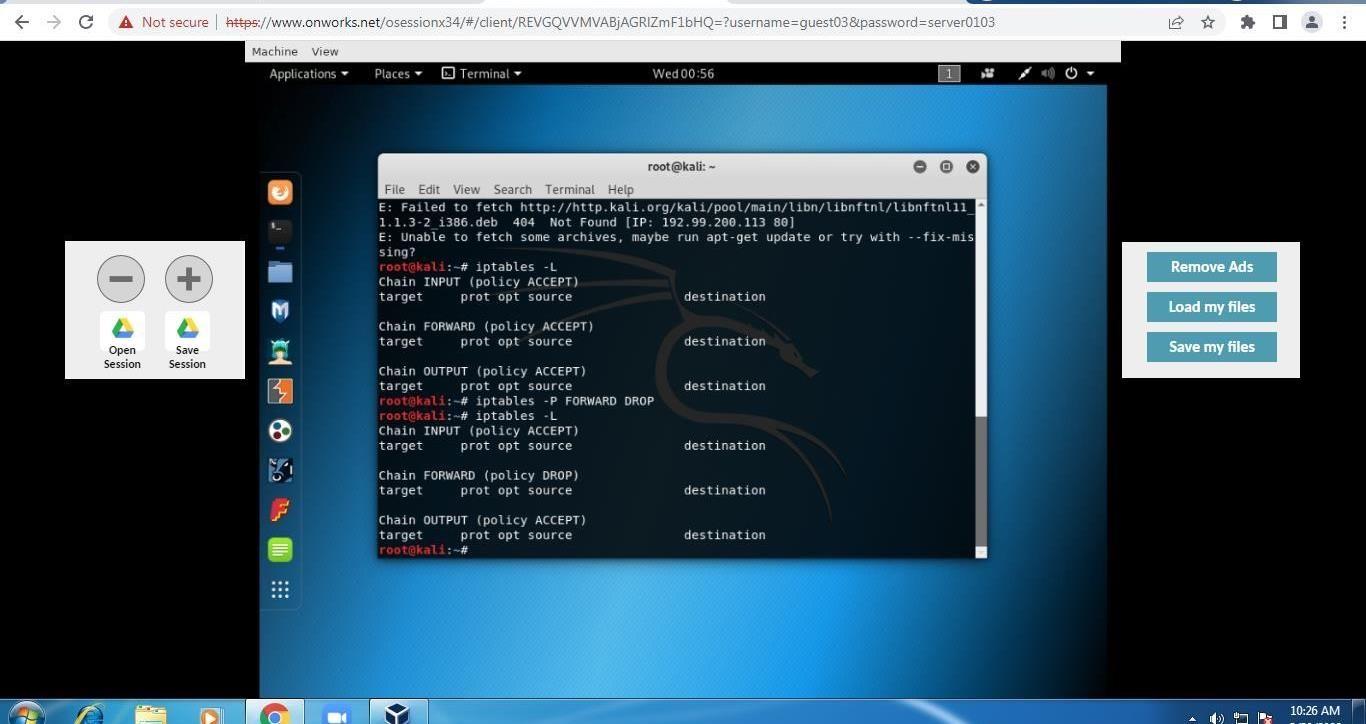
*Command - iptables - L*



# To drop Forward

Command - *iptables -P FORWARD DROP*

*iptables -L*



# To drop given IP address

Let’s assume we want to block the traffic coming from an IP address 192.168.0.23 Command -

*Iptables -A INPUT -S 192.168.0.23 -j DROP iptables*

*-L*

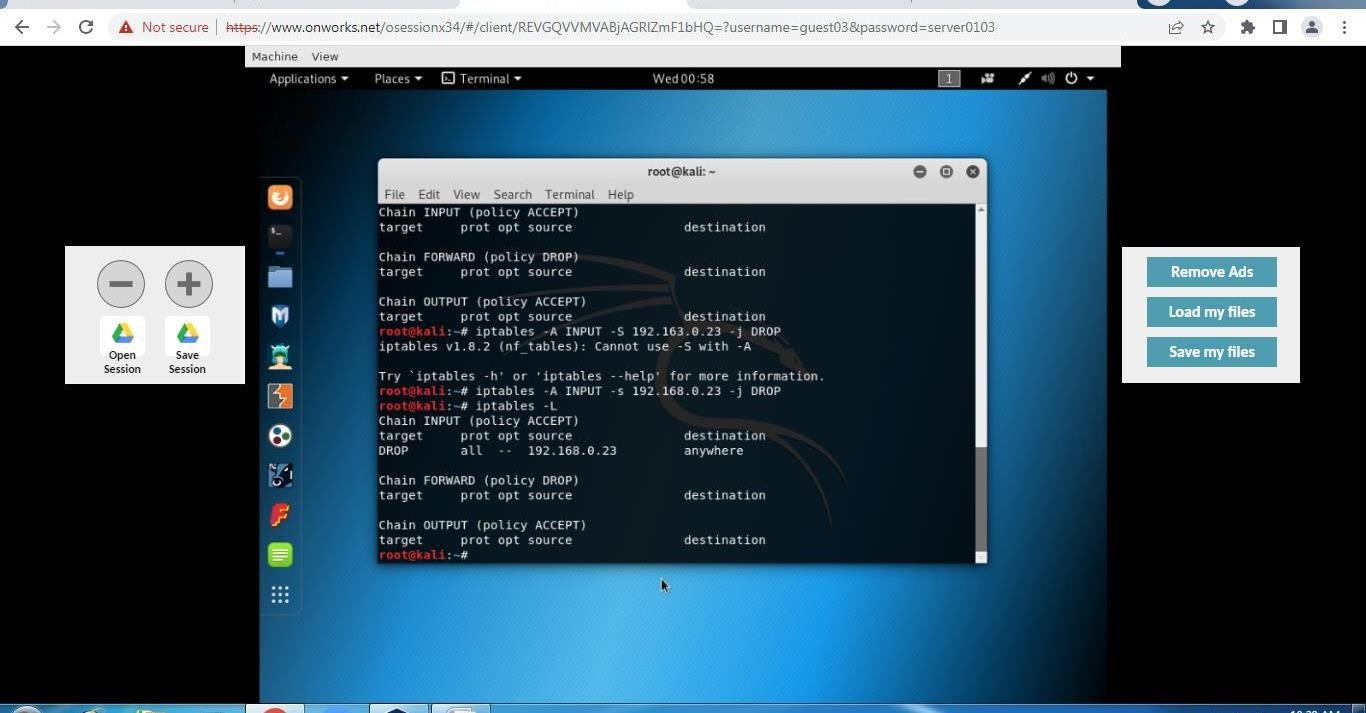
**A INPUT :-** The flag -A is used to append a rule to the end of a chain. This part of the command tells the iptable that we want to add a rule to the end of the INPUT chain.

**-I INPUT:-** In this flag the rules are added to the top of the chain.

**-s 192.168.0.23:-** The flag -s is used to specify the source of the packet. This tells the iptable to look for the packets coming from the source 192.168.1.3

**-j DROP** - This specifies what the iptable should do with the packet.

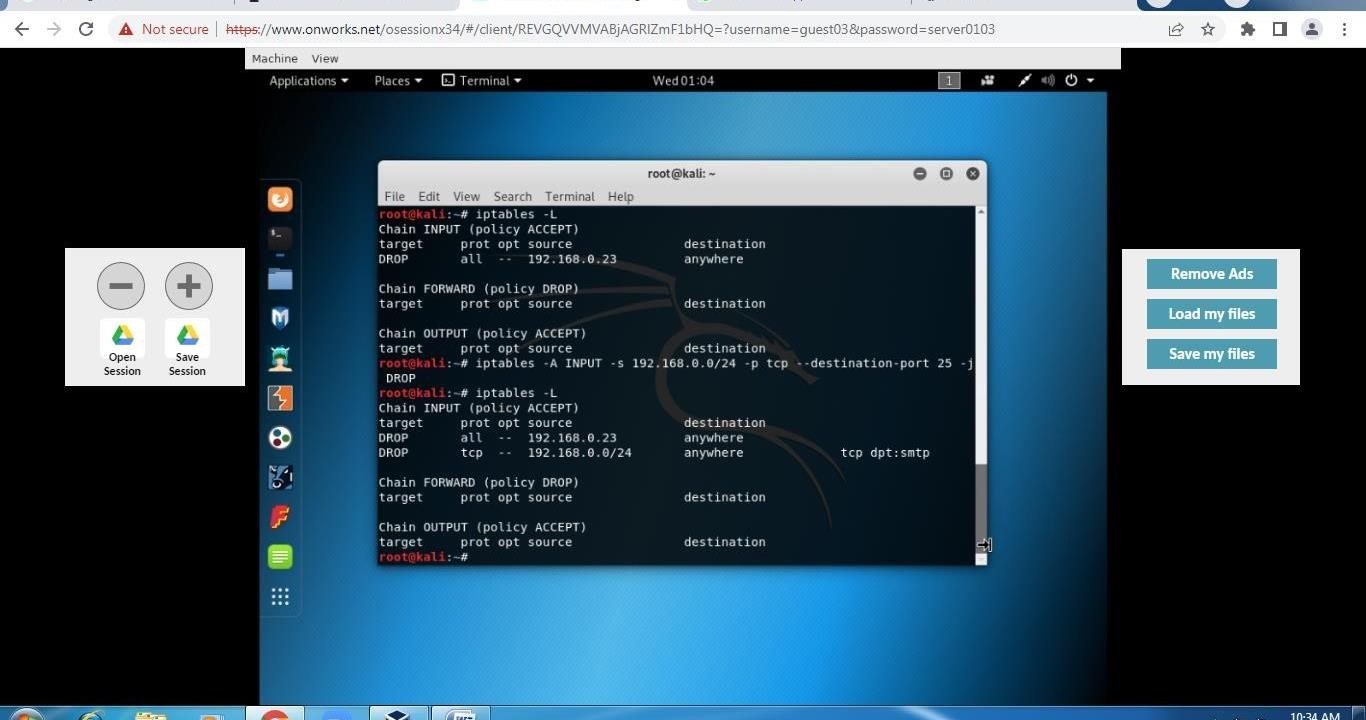
In short, the above command adds a rule to the INPUT chain which says, if any packet arrives whose source address is 192.168.0.23 then drop that packet, that means do not allow the packet reach the computer.



# To drop emails coming from a specific email address as well as blocking all the traffic using tcp and smtp

Command -

*iptables -A INPUT -s 192.168.0.0/24 -p tcp - - destination-port 25 -j DROP iptables -L*



# To accept rule:

If you want to add rules to specific ports of your network,then the following commands can be used.

***Syntax:-*** sudo iptables -A/-I chain\_name -s source\_ip -p protocol\_name --dport port\_number -j Action\_to\_take

## -p protocol\_name:-

This option is used to match the packets that follow the protocol protocol\_name.

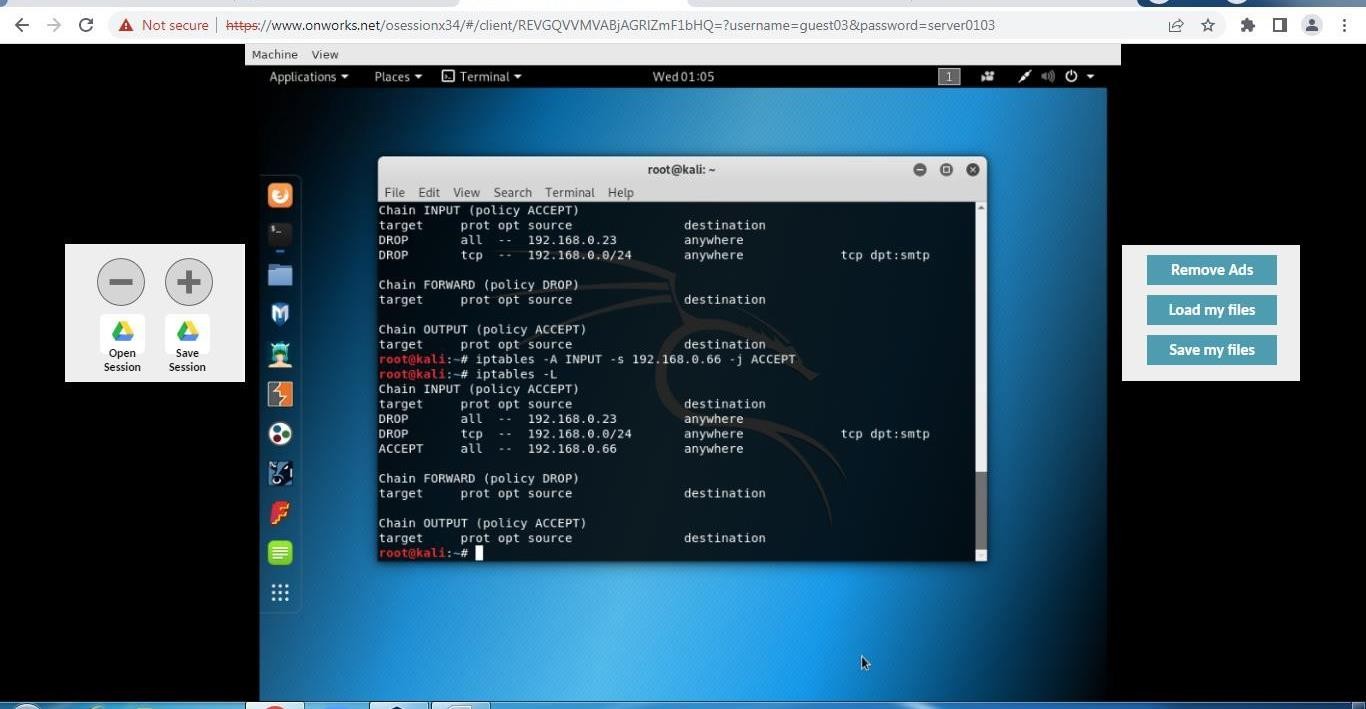
## -dport port\_number:

This is option is available only if you give the -p protocol\_name option. It specifies to look for the packets that are going to the port “port\_number”.

Command -

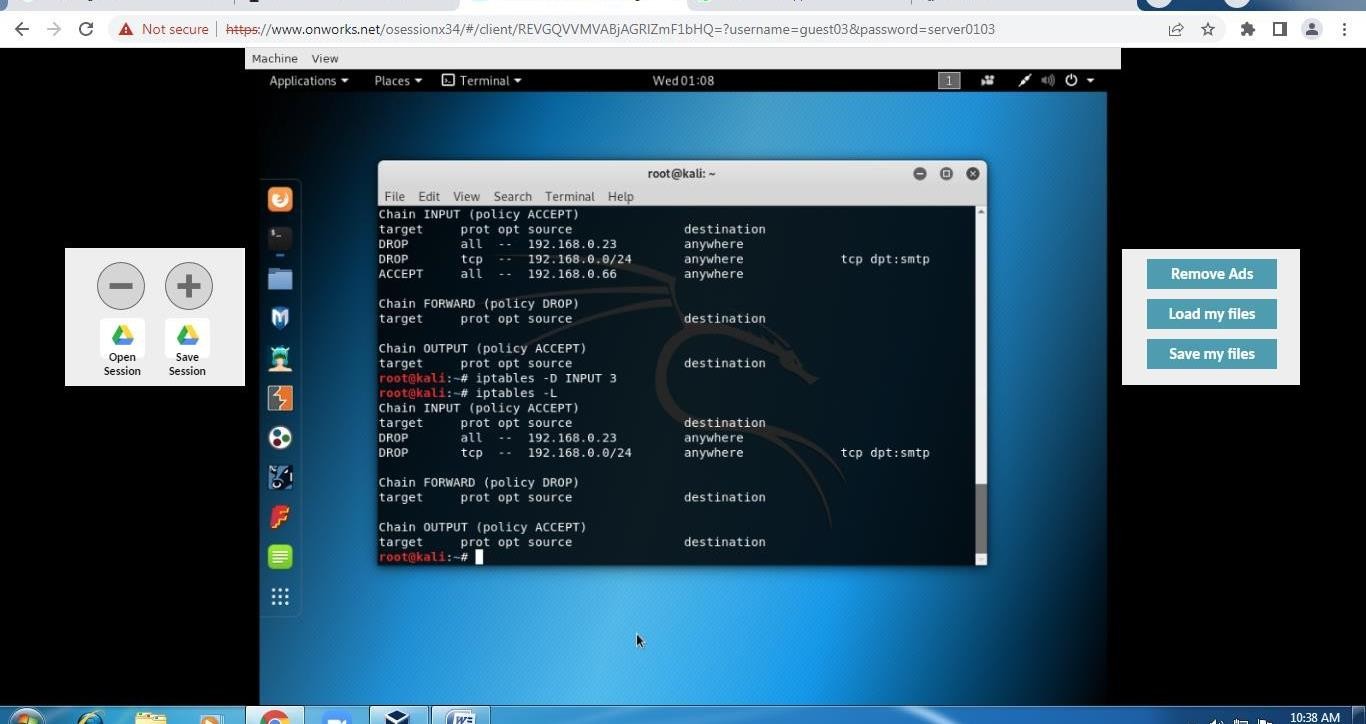
*iptables -A INPUT -s 192.168.0.66 -j ACCEPT*

*iptables -L*



# Deleting rule from iptable

Command - *iptables -D INPUT 3 iptables -L*

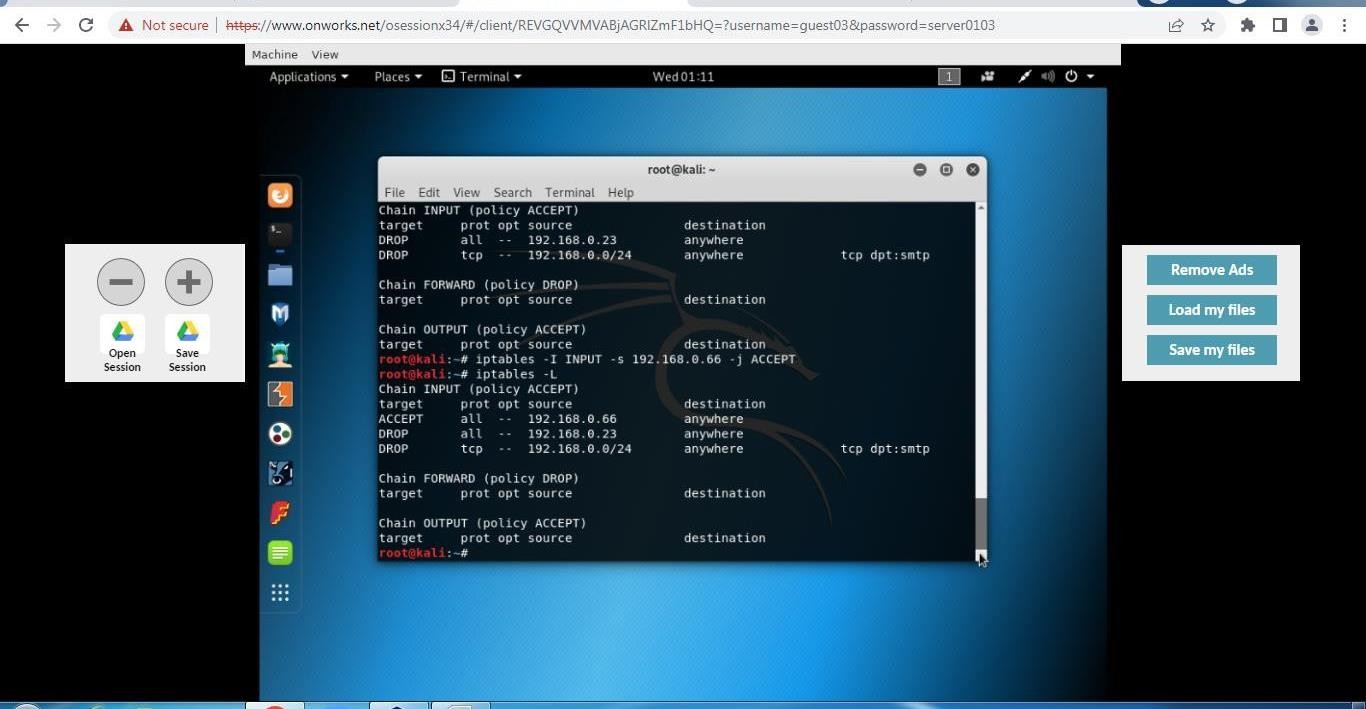


# Accepting rule from iptables in order

Command -

*iptables -I INPUT -s 192.168.0.66 -j ACCEPT*

*iptables -L*

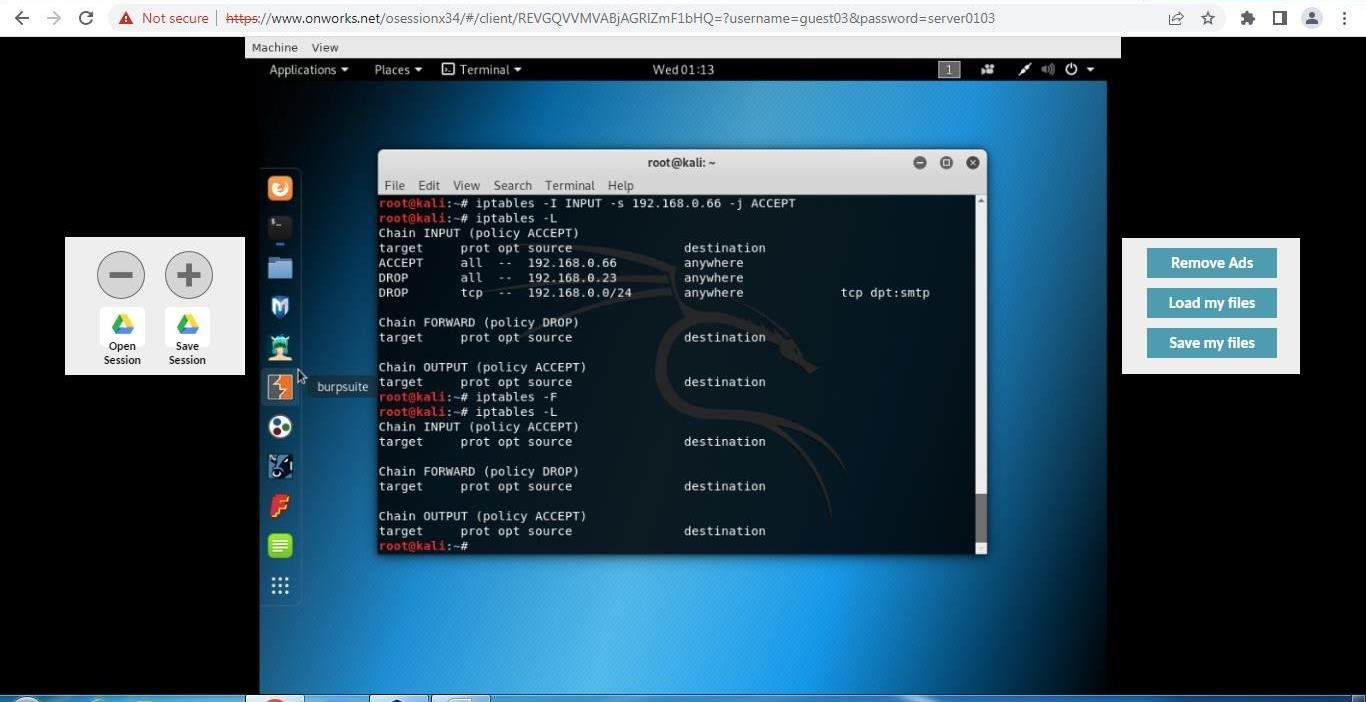


# Flushing all rules from iptable

Command

- *iptables - F iptables*

*-L*



# Analysis:

Using the iptables program, you can explicitly grant and deny access to selected services running on your server, as well as to selected IP addresses. Iptables is a powerful firewall program that you can use to secure your Linux server or VPS. What’s great is that you can define various rules based on your preferences. In this iptables tutorial, you have learned how to install and use the tool. Now, we hope you can manage your sets of rules to filter incoming and outgoing packets.

# Conclusion:

Successfully understood and implemented firewall using iptables and implemented various commands on kali Linux.

For Faculty use

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correction Parameters** | **Formative Assessment**  **[40%]** | **Timely completion of Practical [ 40%]** | **Attendance / Learning**  **Attitude [20%]** |  |
| **Marks**  **Obtained** |  |  |  |