Roll No 06

Name- Prasad Sunil Arote

Date- 28/08/2023

Lab Assignment 3

AIM: Block cipher modes of operation using Advanced Encryption Standard (AES).

LO2: Demonstrate key management, distribution and user authentication.

THEORY:

Briefly explain AES algorithm (What type of cipher it is? number of rounds, keysize, block size, operations in each round)

The Advanced Encryption Standard (AES) is a widely used symmetric encryption algorithm that falls under the category of block ciphers. It replaced the older Data Encryption Standard (DES) due to its stronger security features. AES operates on fixed-size blocks of data and is known for its efficiency and resistance against various types of attacks.

Type of Cipher: AES is a symmetric key block cipher, which means the same secret key is used for both encryption and decryption. It transforms plaintext blocks into ciphertext blocks using a series of complex operations.

Number of Rounds: AES operates with different numbers of rounds depending on the key size:

AES-128: 10 rounds

AES-192: 12 rounds

AES-256: 14 rounds

Key Size: AES supports key sizes of 128, 192, or 256 bits. The security and strength of the encryption increase with larger key sizes.

Block Size: AES operates on fixed-size blocks of 128 bits.

Operations in Each Round:

SubBytes: Non-linear substitution of each byte in the block using a predefined substitution table (S-box).

ShiftRows: Byte shifting within each row to provide diffusion in the data.

MixColumns: Mixing operation that transforms columns of data to provide diffusion across columns.

AddRoundKey: Each byte of the block is combined with the corresponding round key derived from the original encryption key.

These operations are applied repeatedly for the specified number of rounds, with each round using a different round key. The complex interaction of these operations ensures that even a small change in the plaintext results in a significantly different ciphertext, a property known as the avalanche effect. This contributes to the security and robustness of AES against various cryptographic attacks.

With diagram explain in brief block cipher modes of operation

ECB mode

CBC mode

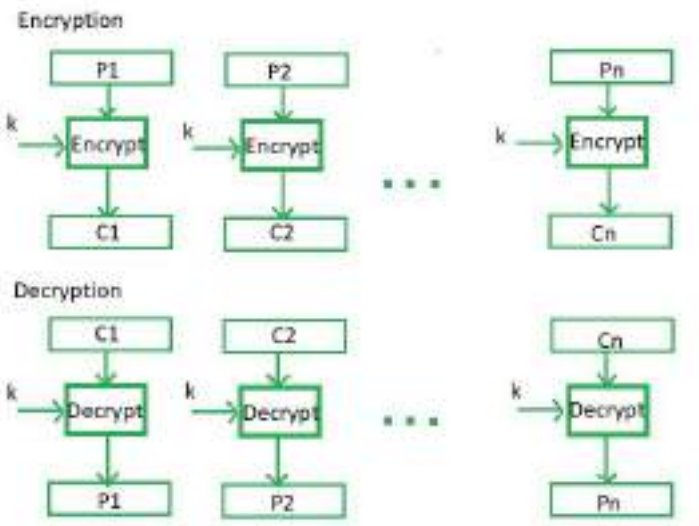
OFB mode

Counter mode

Block cipher modes of operation are techniques used to apply a block cipher, like AES, to encrypt or decrypt data that is larger than the block size of the cipher. These modes define how blocks of plaintext are transformed into ciphertext and vice versa. Here's a brief explanation of some common block cipher modes of operation:

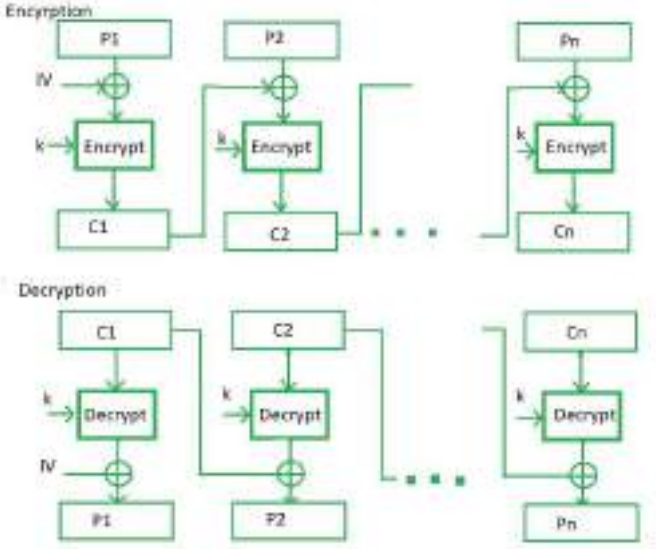
Electronic Codebook (ECB) Mode:

ECB mode is the simplest mode, where each block of plaintext is independently encrypted using the same encryption key. However, this mode has a significant limitation: identical plaintext blocks result in identical ciphertext blocks, making it vulnerable to certain attacks. ECB mode is not suitable for encrypting large amounts of data or data with patterns.



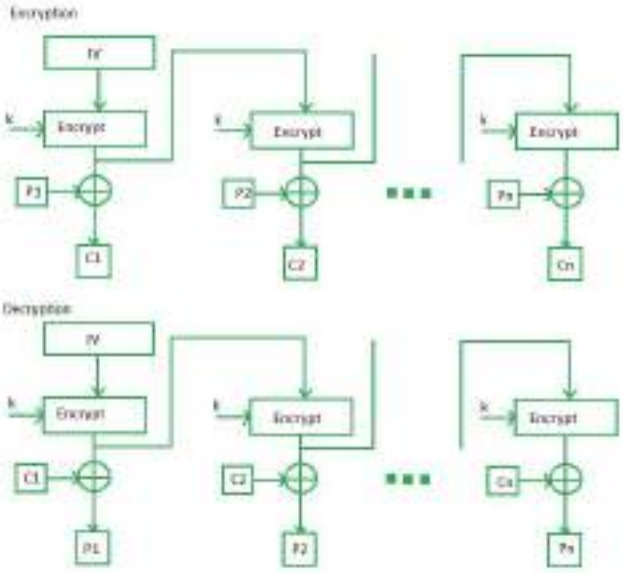
Cipher Block Chaining (CBC) Mode:

In CBC mode, each plaintext block is XORed with the previous ciphertext block before encryption. This introduces a form of feedback, where the ciphertext from the previous block affects the encryption of the current block. Initialization Vector (IV) is used to start the process. CBC mode prevents identical plaintext blocks from producing identical ciphertext blocks and provides a basic level of security. Decryption requires the previous ciphertext block to be available.



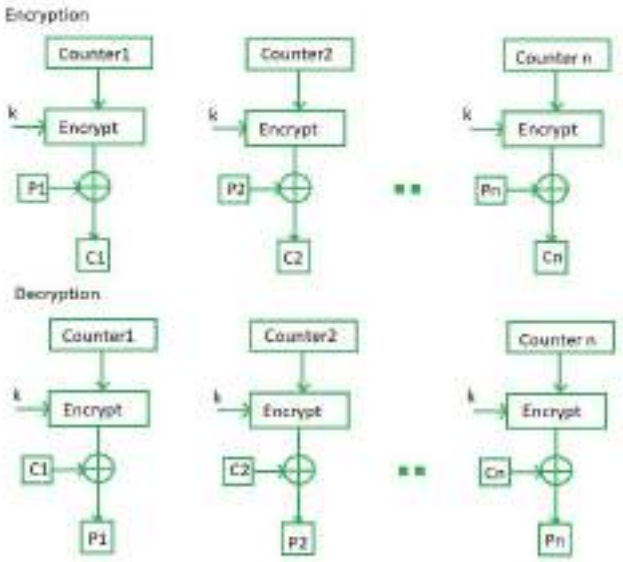
Output Feedback (OFB) Mode:

OFB mode converts a block cipher into a stream cipher. It generates a keystream using the encryption of an IV and successive values (feedback) derived from the encryption of the previous block's ciphertext. The keystream is XORed with the plaintext to produce the ciphertext and vice versa. OFB mode does not require decryption in the encryption process and is suitable for applications where error propagation is a concern.



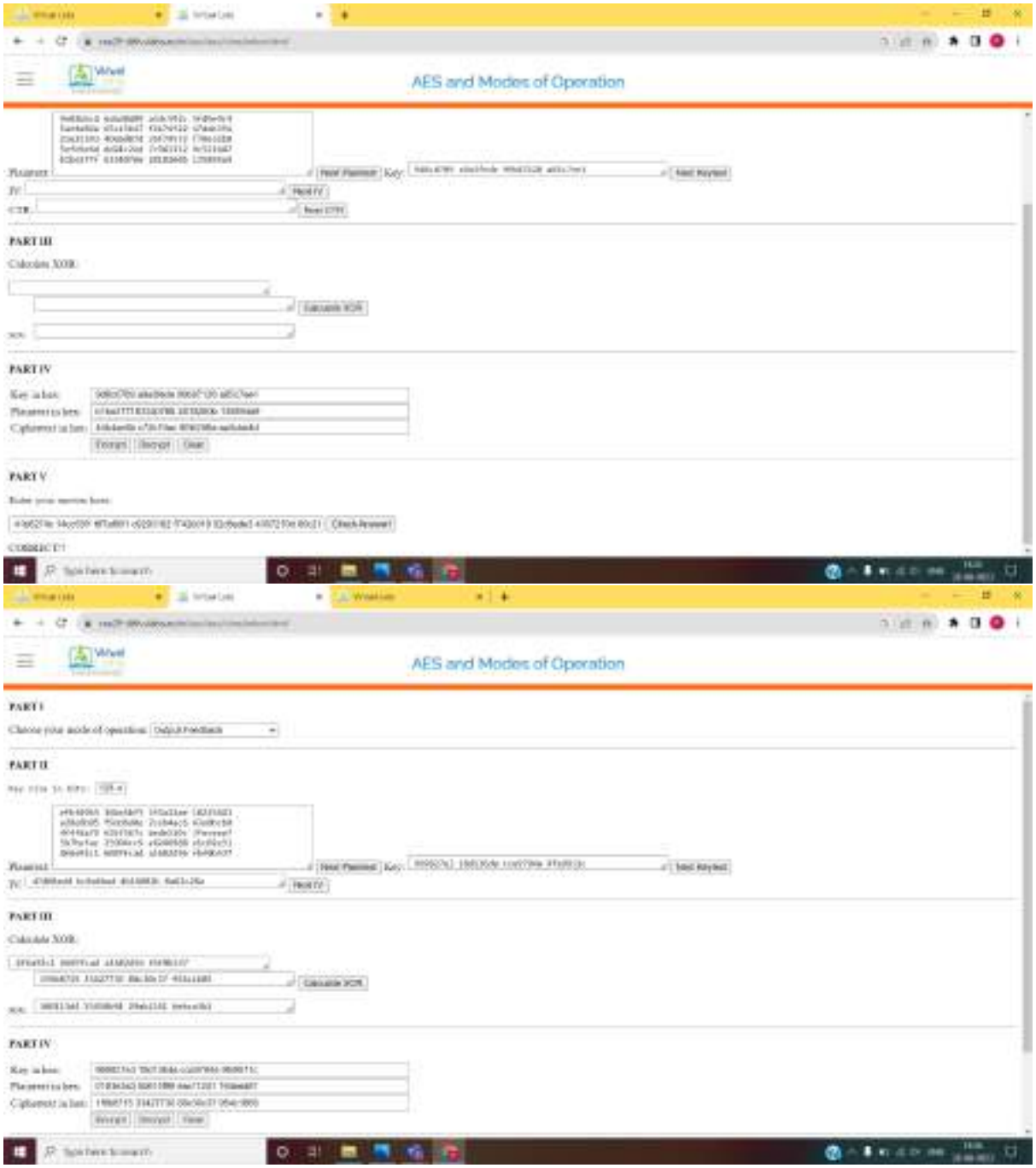
Counter (CTR) Mode:

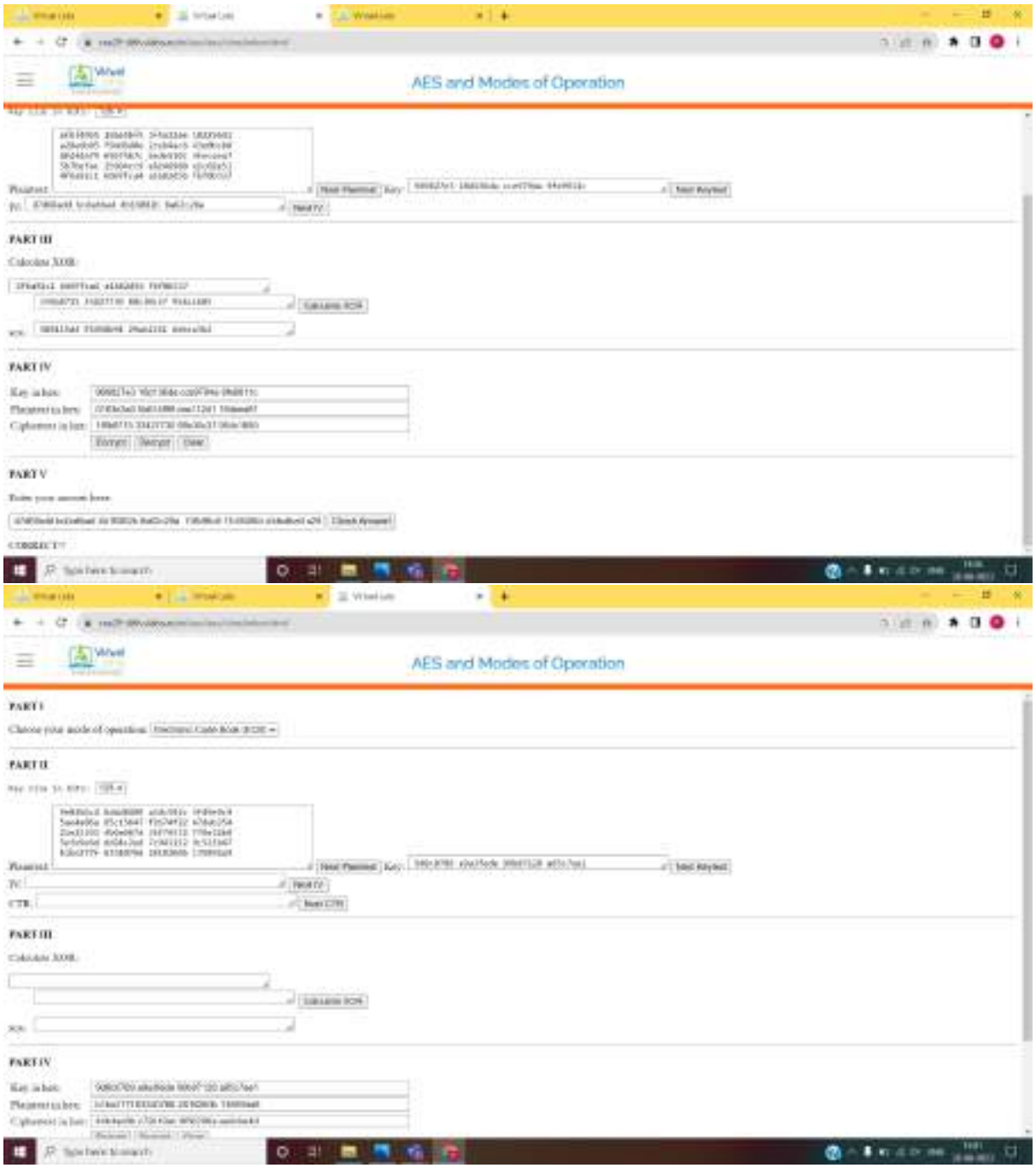
CTR mode also turns a block cipher into a stream cipher. It involves encrypting a counter value using the encryption key, and the resulting output is XORed with the plaintext to produce the ciphertext. The counter value is incremented for each block. CTR mode allows for parallel encryption and decryption, making it efficient for multi-core processors. It also offers excellent error propagation.

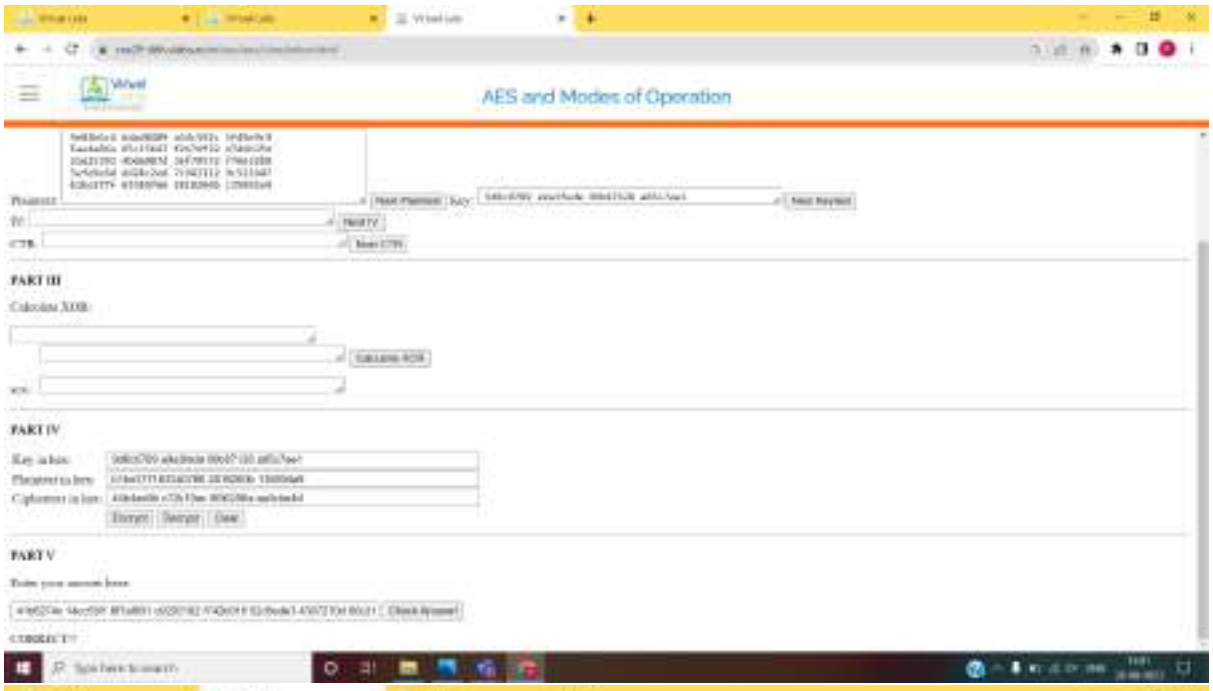


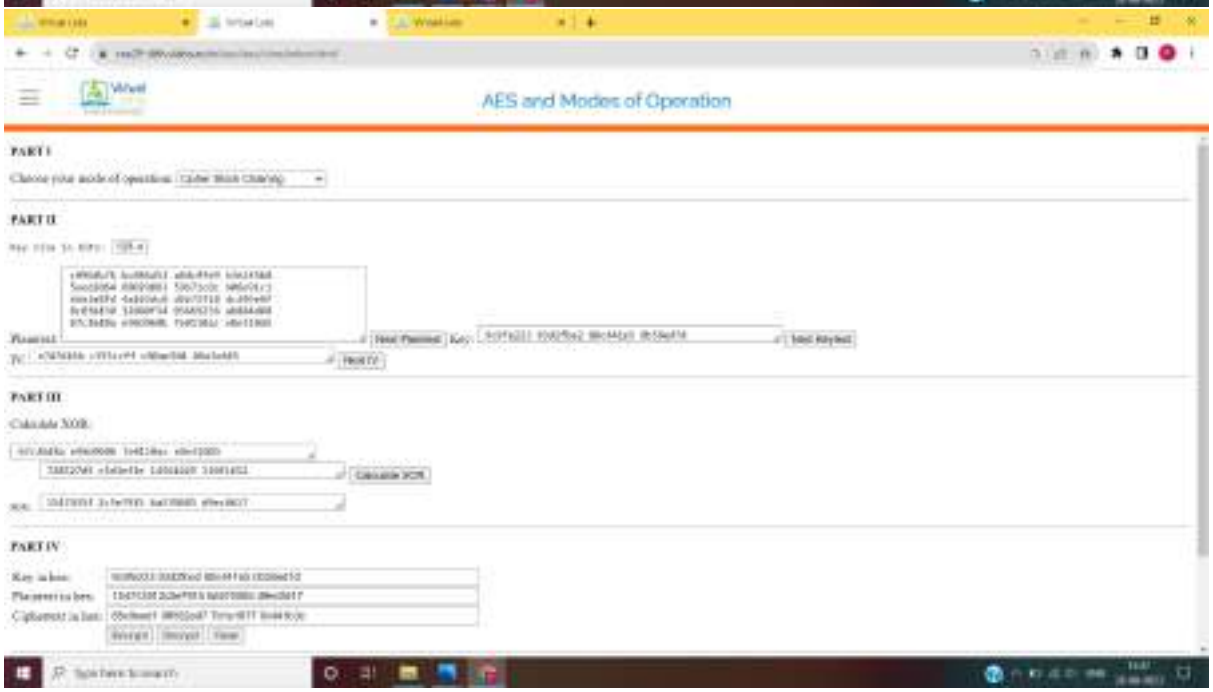
These modes provide different trade-offs between security, performance, and error propagation. It's important to choose the appropriate mode based on the specific requirements of your application. Additionally, some modes, like CBC and CTR, require the use of Initialization Vectors (IVs) to ensure uniqueness and security of the encryption process.

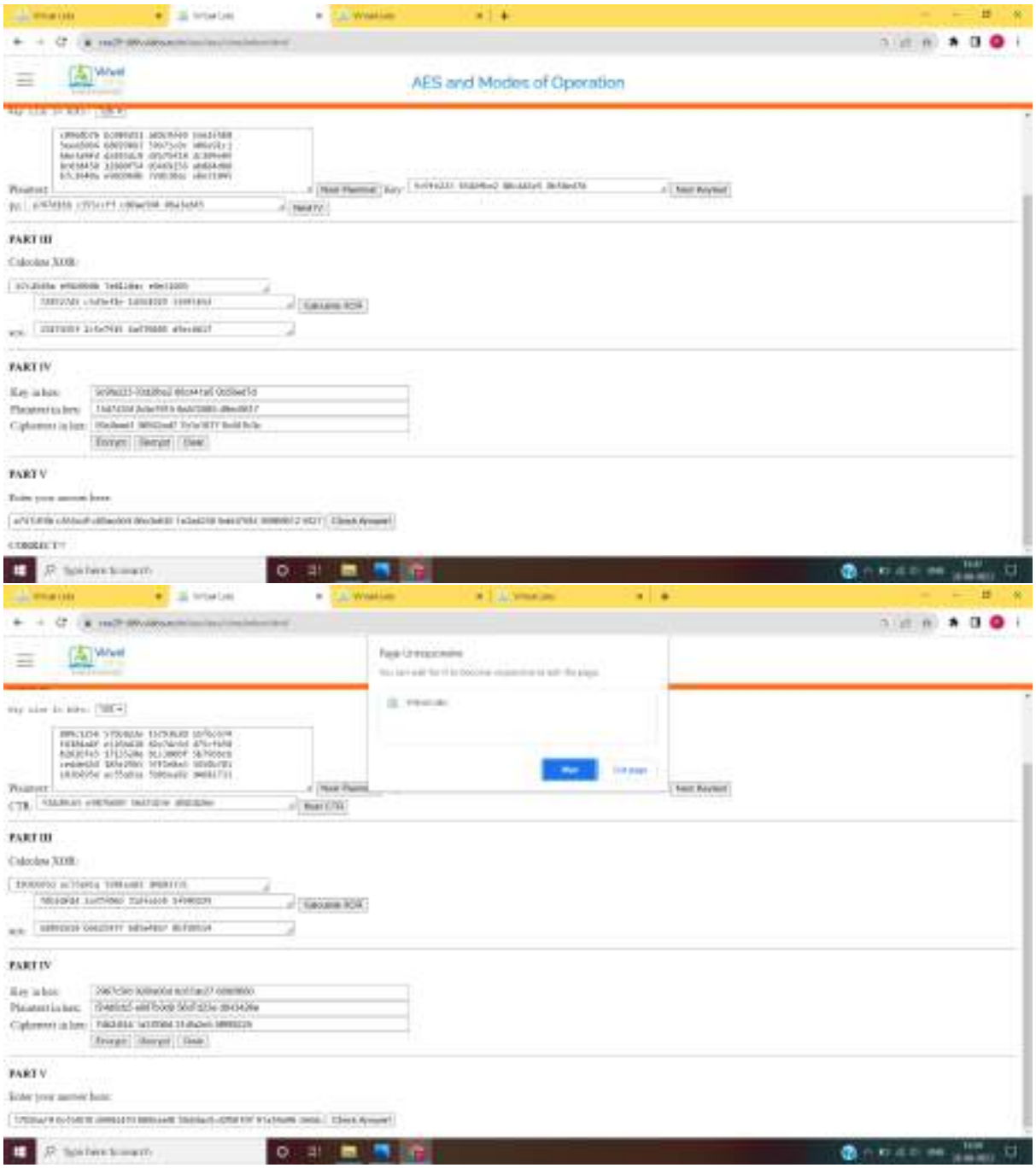
OUTPUT

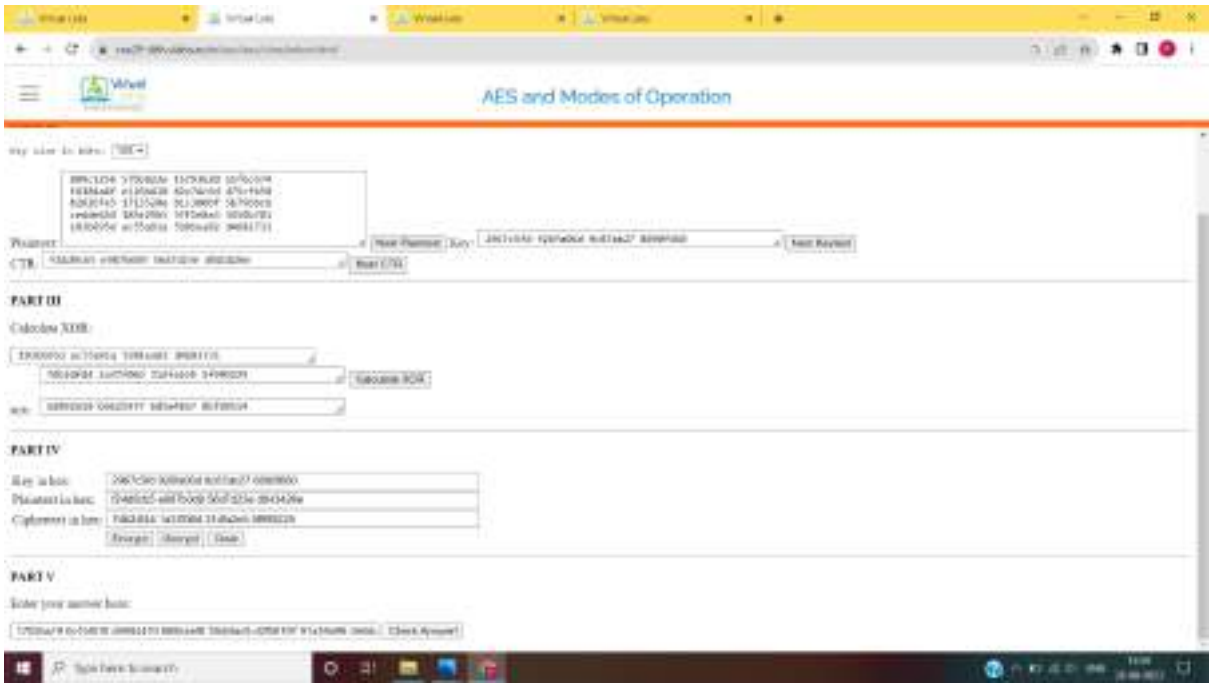

AES and Modes of Operation
PART III
PART IV
PART V
AES and Modes of Operation
PART I
PART II
PART III
PART IV


AES and Modes of Operation
PART III
PART IV
PART V
AES and Modes of Operation
PART I
PART II
PART III
PART IV


AES and Modes of Operation
PART III
Calculate XOR
PART IV
PART V


AES and Modes of Operation
PART I
PART II
PART III
Calculate XOR
PART IV


AES and Modes of Operation
PART III
Calculate XOR
PART IV
PART V
PART III
Calculate XOR
PART IV
PART V



CONCLUSION:

The AES experiment offered a practical glimpse into the world of symmetric key cryptography. We explored AES's encryption processes, recognizing its efficiency and adaptability for secure data handling. By employing different modes of operation, such as ECB, CBC, OFB, and Counter, we comprehended the distinct trade-offs between security, performance, and error propagation.

Roll No :- 06

Name :- Prasad Sunil Arote

Date :- 8/08/2023

Assignment 04

Aim: Implementation and analysis of RSA cryptosystem and Digital Signature scheme using RSA.

LO2: To analyze and implement public key encryption algorithms and digital signature algorithms.

Theory:

Steps of RSA key generation

RSA (Rivest-Shamir-Adleman) is a widely used asymmetric cryptographic algorithm that involves the generation of a key pair: a public key and a private key. The public key is used for encryption, while the private key is used for decryption. Here are the steps of RSA key generation:

1. **Choose Two Large Prime Numbers:** Select two distinct prime numbers, usually denoted as "p" and "q." These primes will form the basis for generating the keys. The security of RSA relies on the difficulty of factoring the product of these primes, so they need to be sufficiently large.
2. **Calculate the Modulus**: Compute the modulus "n" by multiplying the two prime numbers: n = p * q. The modulus is a part of both the public and private keys and is used in the encryption and decryption processes.
3. **Calculate Euler's Totient Function:** Calculate Euler's totient function ϕ(n) for the modulus "n." For two distinct primes p and q, ϕ(n) = (p - 1) * (q - 1). This function is important for generating the private key.
4. **Choose Public Exponent:** Select a small odd integer "e" (usually a prime) that is greater than 1 and less than ϕ(n). This value will be part of the public key and is used for encryption.
5. **Calculate Private Exponent:** Calculate the private exponent "d" using the modular inverse of the public exponent "e" modulo ϕ(n). In other words, find "d" such that (e * d) % ϕ(n) = 1. This private exponent is part of the private key and is used for decryption.
6. **Key Pair Generation:** Now you have a public key (n, e) and a private key (n, d). The public key consists of the modulus "n" and the public exponent "e," and the private key consists of the modulus "n" and the private exponent "d."
7. **Optional:** Additional Parameters: Depending on the specific RSA implementation and security requirements, additional parameters such as padding schemes might be applied to enhance the security and reliability of the RSA algorithm.

Steps of Digital signature generation and verification process

Digital signatures are cryptographic mechanisms used to ensure the authenticity, integrity, and non-repudiation of digital documents or messages. They involve the use of asymmetric encryption and hash functions to create and verify signatures. Here are the steps for digital signature generation and verification:

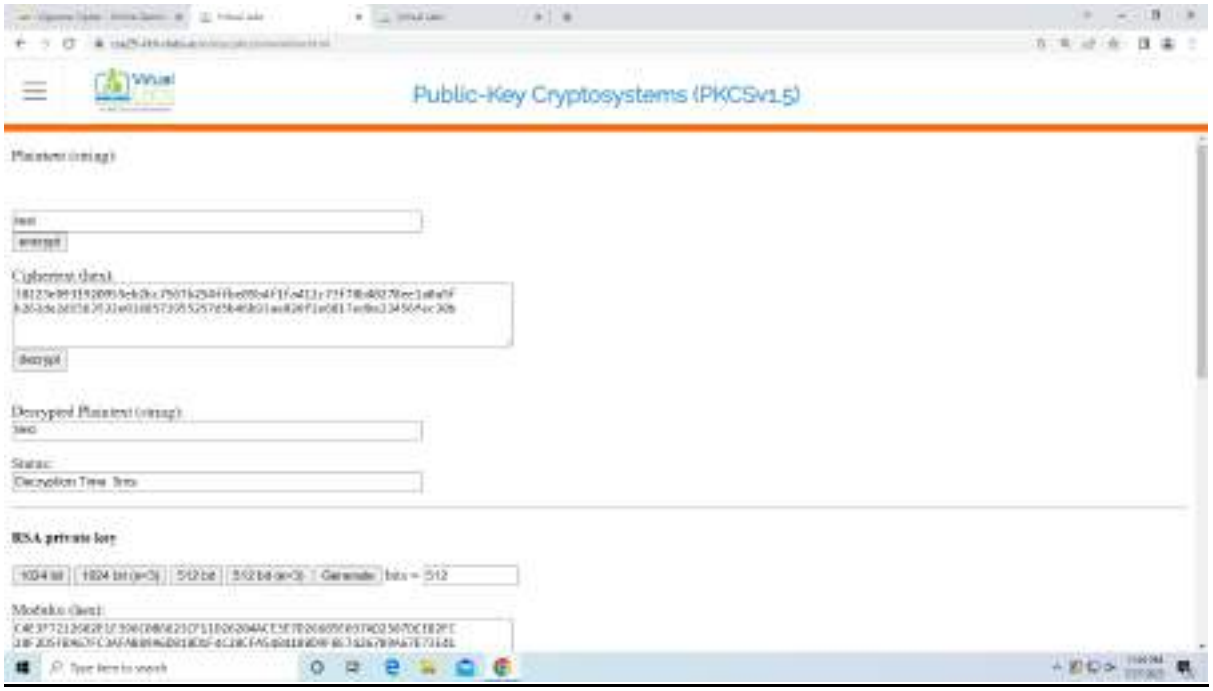
Digital Signature Generation:

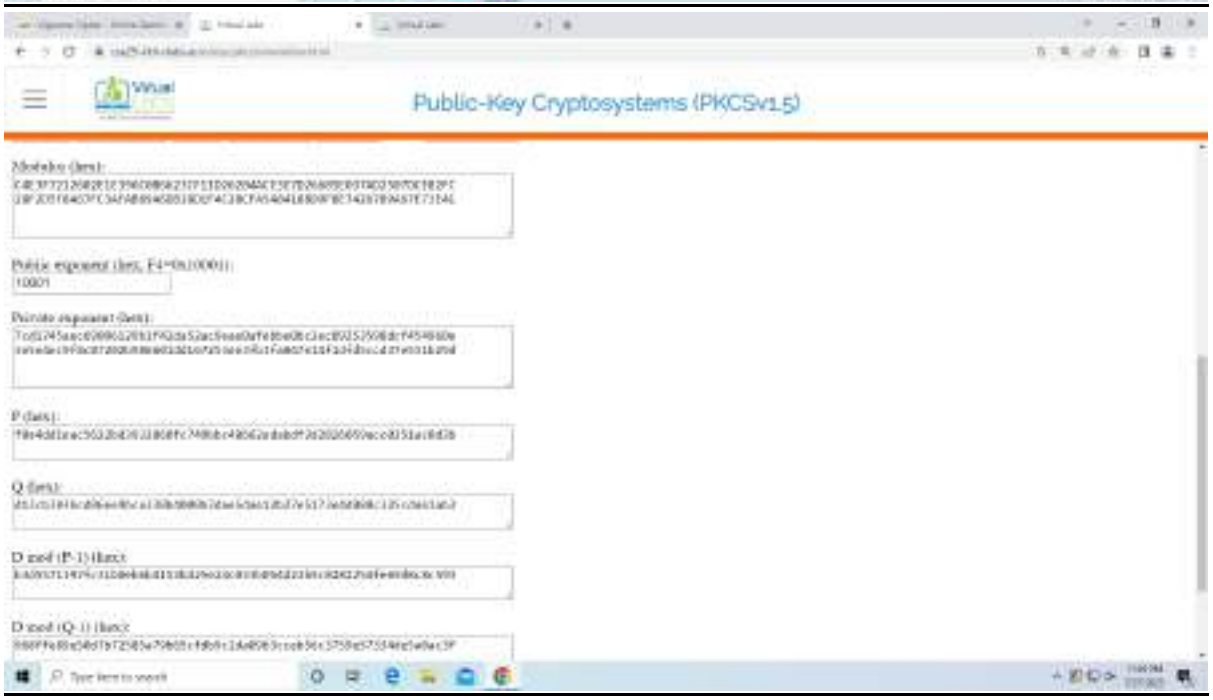
1. Hashing: The first step is to create a hash of the document or message that needs to be signed. A hash function converts the variable-length input (document) into a fixed-length string of characters, which represents the unique fingerprint of the document. Common hash functions used are SHA-256 or SHA-512.
2. Private Key Signing: The hash value is then encrypted using the sender's private key. This creates the digital signature. The private key is known only to the sender and is used to ensure that only the sender could have created this particular signature.
3. Attachment: The digital signature is attached to the original document, creating a signed message. This message includes both the original document and the digital signature.

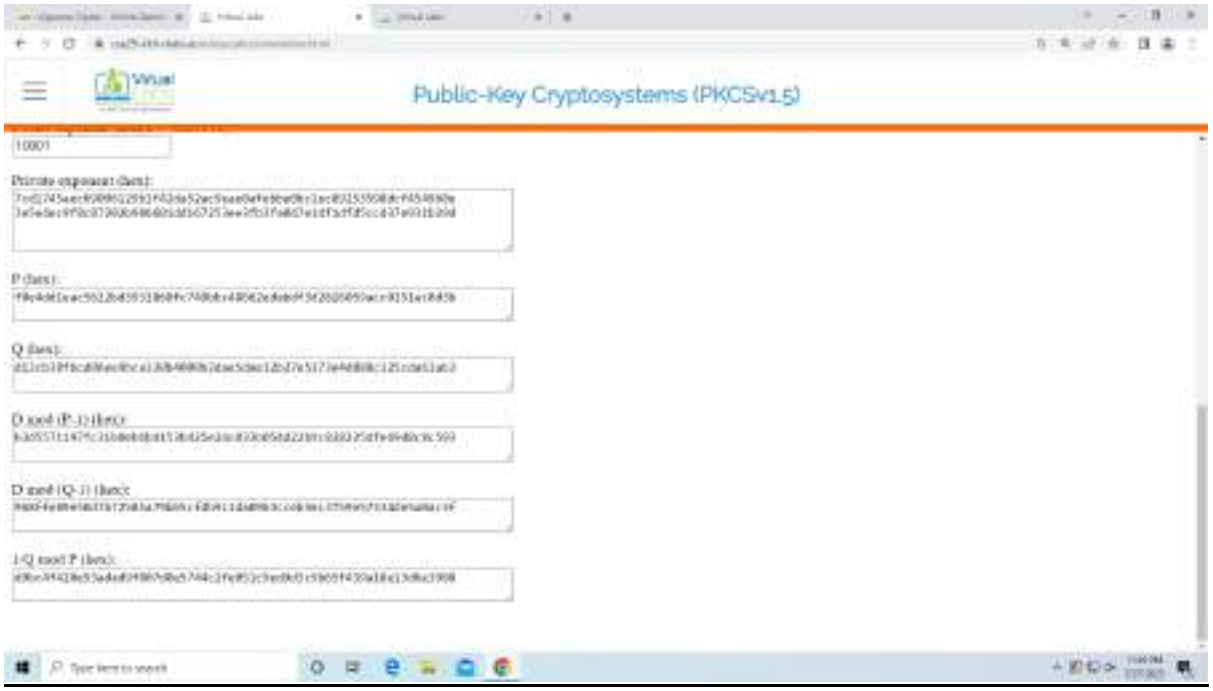
Digital Signature Verification:

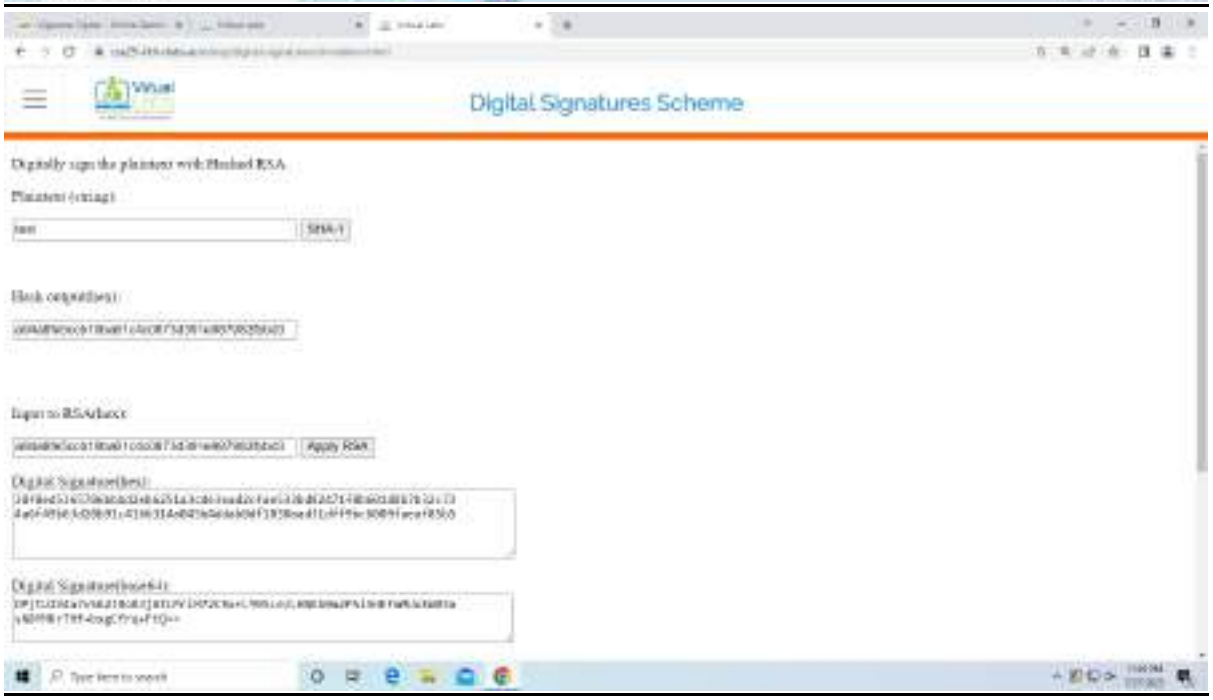
1. Hashing: The receiver starts by creating a hash of the received document using the same hash function used by the sender.
2. Public Key Decryption: The digital signature is decrypted using the sender's public key. Remember that in asymmetric encryption, only the private key can decrypt what the public key encrypts, ensuring that the signature could only have been created by the holder of the private key.
3. Compare Hashes: The decrypted signature should match the hash of the received document. If they match, it means the document hasn't been tampered with during transmission, as any modification would have resulted in a different hash value. This step ensures the integrity of the document.
4. Authentication: The verification process confirms the authenticity of the sender because only the sender's private key could have produced a matching signature. This step ensures that the document was indeed signed by the claimed sender.
5. Non-Repudiation: The verification process establishes non-repudiation, meaning the sender cannot later deny having signed the document since their private key was used to create the digital signature.

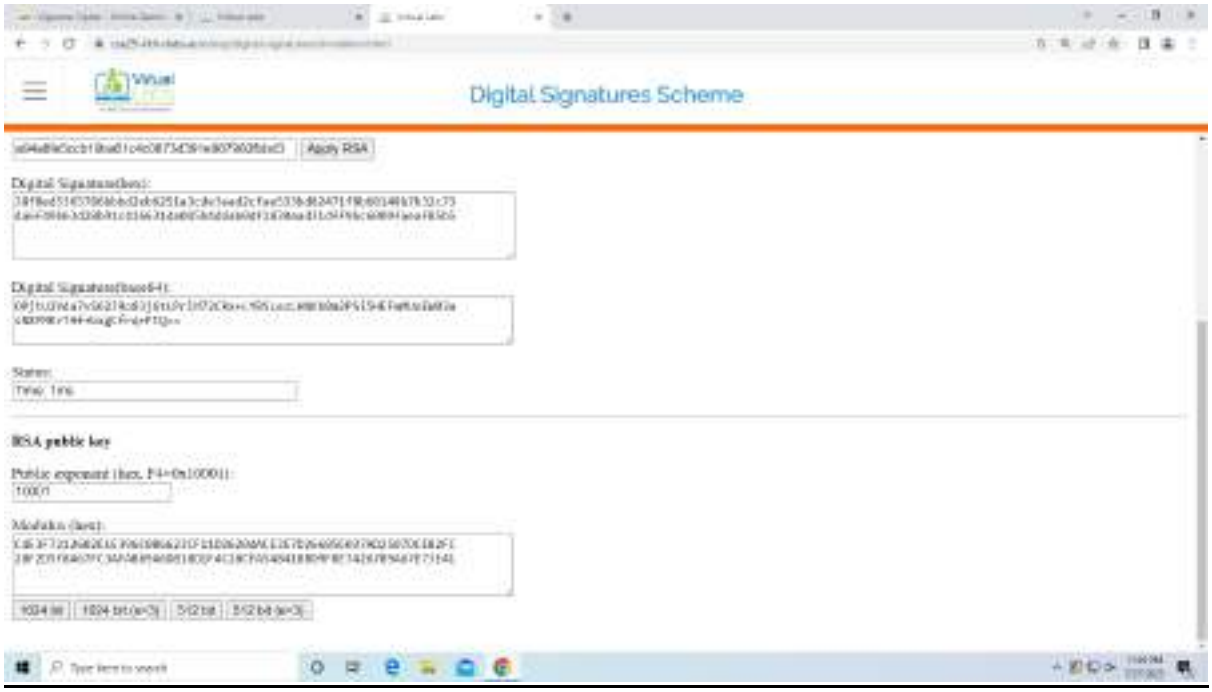
OUTPUT:


Public-Key Cryptosystems (PKCSv1.5)
Plaintext (string):
test
encrypt
Ciphertext (hex):
decrypt
Decrypted Plaintext (string):
test
Status:
Decryption Time: 5ms
RSA private key
1024 bit
1024 bit (e=3)
512 bit
512 bit (e=3)
Generate
bits = 512
Modulo (hex):


Public-Key Cryptosystems (PKCSv1.5)
Modulo (hex):
Public exponent (hex, F4=0x10001):
10001
Private exponent (hex):
P (hex):
Q (hex):
D mod (P-1) (hex):
D mod (Q-1) (hex):


Public-Key Cryptosystems (PKCSv1.5)
10001
Private exponent (hex):
P (hex):
Q (hex):
D mod (P-1) (hex):
D mod (Q-1) (hex):
1/Q mod P (hex):


Digital Signatures Scheme
Digitally sign the plaintext with Hashed RSA
Plaintext (string):
test
SHA-1
Hash output(hex):
Input to RSA(hex):
Apply RSA
Digital Signature(hex):
Digital Signature(base64):



CONCLUSION:

The steps of RSA key generation and the digital signature process were understood, gained insight into the foundations of modern cybersecurity practices.

Roll No 06

Name- Prasad Sunil Arote

Date- 17-08-2023

Lab Assignment 5

Aim: To explore hashdeep tool in kali linux for generating, matching and auditing hash of files.

LO2: Demonstrate key management, distribution and user authentication.

Theory:

What is the need of hashing? List different hashing algorithms.

Need for Hashing:

1. Hashing is a process of converting input data (such as text, files, passwords) into a fixed-size value (hash value or hash code) using a hashing algorithm. Hashing serves several purposes in computer science and cybersecurity:
2. Data Integrity: Hashing can verify whether data has been altered or corrupted. If the hash value of the original data matches the hash value calculated from the received data, it's likely that the data hasn't been tampered with.
3. Password Security: Hashing is used to securely store passwords. Instead of storing plain-text passwords, systems store the hash values of passwords. When a user logs in, the input password's hash is compared with the stored hash.
4. Digital Signatures: Hashing plays a role in creating and verifying digital signatures. A hash value of a message is signed with a private key to create a digital signature, providing authenticity and non-repudiation.
5. Data Structures: Hashing is used in hash tables, which provide efficient data retrieval. Hash functions map keys to indices in an array, allowing quick access to data.

Different Hashing Algorithms:

There are various hashing algorithms, each with its characteristics and use cases. Some commonly used ones are:

1. MD5 (Message Digest Algorithm 5)
2. SHA-1 (Secure Hash Algorithm 1)
3. SHA-256, SHA-384, SHA-512 (Secure Hash Algorithms 256, 384, 512)
4. bcrypt (Adaptive Hashing Algorithm)
5. scrypt (Memory-Hard Function)
6. Argon2 (Winner of Password Hashing Competition)

Write the commands used for generating hash values, matching them with stored hash values and auditing using hashdeep tool.

1.To check the version of Hashdeep - hashdeep -V

2. To display help about hashdeep - hashdeep -h or hashdeep -hh

3. To display the manual page of hashdeep- man hashdeep

4. To display the manual page of any specific hash algorithm supported

by hashdeep- man md5deep

By default, hashdeep generates MD5 n SHA256 hash values.

5. To hash a file - hashdeep filename

6. If you don’t want to display the full path of file in output hash

record- hashdeep -b filename

7. To supress any error messages- hashdeep -s filename

8. To apply multiple hash algorithms than defaulthashdeep -c md5,sha1,sha256,tiger filename

9. To hash multiple files (say all text files) using md5

hashdeep -c md5 *.txt

10. To hash multiple files (say all text files) using md5 and sha1

hashdeep -c md5,sha1 *.txt

11. Hashing block of files- hashdeep -c md5 -p 100 example.txt

12. To recursively calculate hash (all files and subdirectories in a

specified directory)

hashdeep c md5 -r /home/lab006/myfiles

13. To redirect the output of md5 hash of files to another file

md5deep *.txt>hashset.txt

hashdeep *.txt>hashtext1.txt

Check the content of output filecat hashset.txt

cat hashset1.txt

14. To display output in matching mode

md5deep -m hashset.txt *

hashdeep -m -k hashset1.txt *

15. To supress unwanted system msgs/error

md5deep -m hashset.txt *

hashdeep -s -m hashset1.txt *

No output is displayed if there is no matching hashed file is found.

16. To display all files which are negatively matching use -x option

Md5deep -s -x hashset.txt *

hashdeep -s -x hashset1.txt *

Forensic auditing can be done using hashdeep tool which means a check to determine if any files in the system are changed due to malware or any normal system operation like update patching.

17. To audit, first create a hashset file and then audit it against the files to be checked if they are modified.

```
hashdeep -c md5,sha1,sha256 -r /home/lab006/myfiles>hashset1.txt
hashdeep -a -r -k hashset1.txt /home/lab006/myfiles
```

18. Add new file to the directory and audit. It fails.

```
touch /home/lab006/myfiles/newfile.txt
hashdeep -a -r -k hashset1.txt /home/lab006/myfiles
```

19. To get where it failed use the command with -v option

```
hashdeep -v -a -r -k hashset1.txt /home/lab006/myfiles
```

20. Move one of the files to another directory and audit n see output

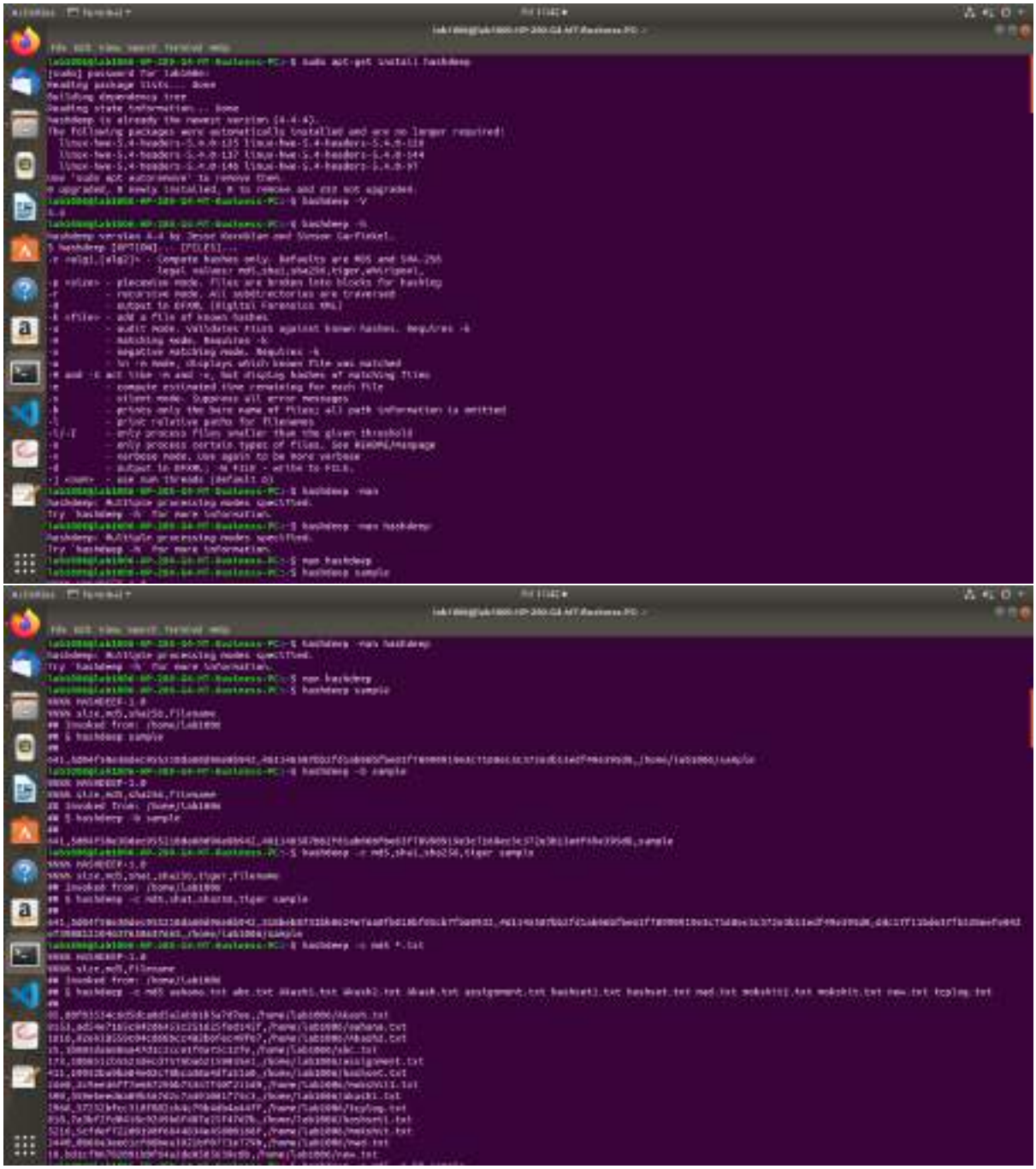
```
mv /home/lab006/myfiles/example.txt /tmp
hashdeep -v -a -r -k hashset1.txt /home/lab006/myfiles
```

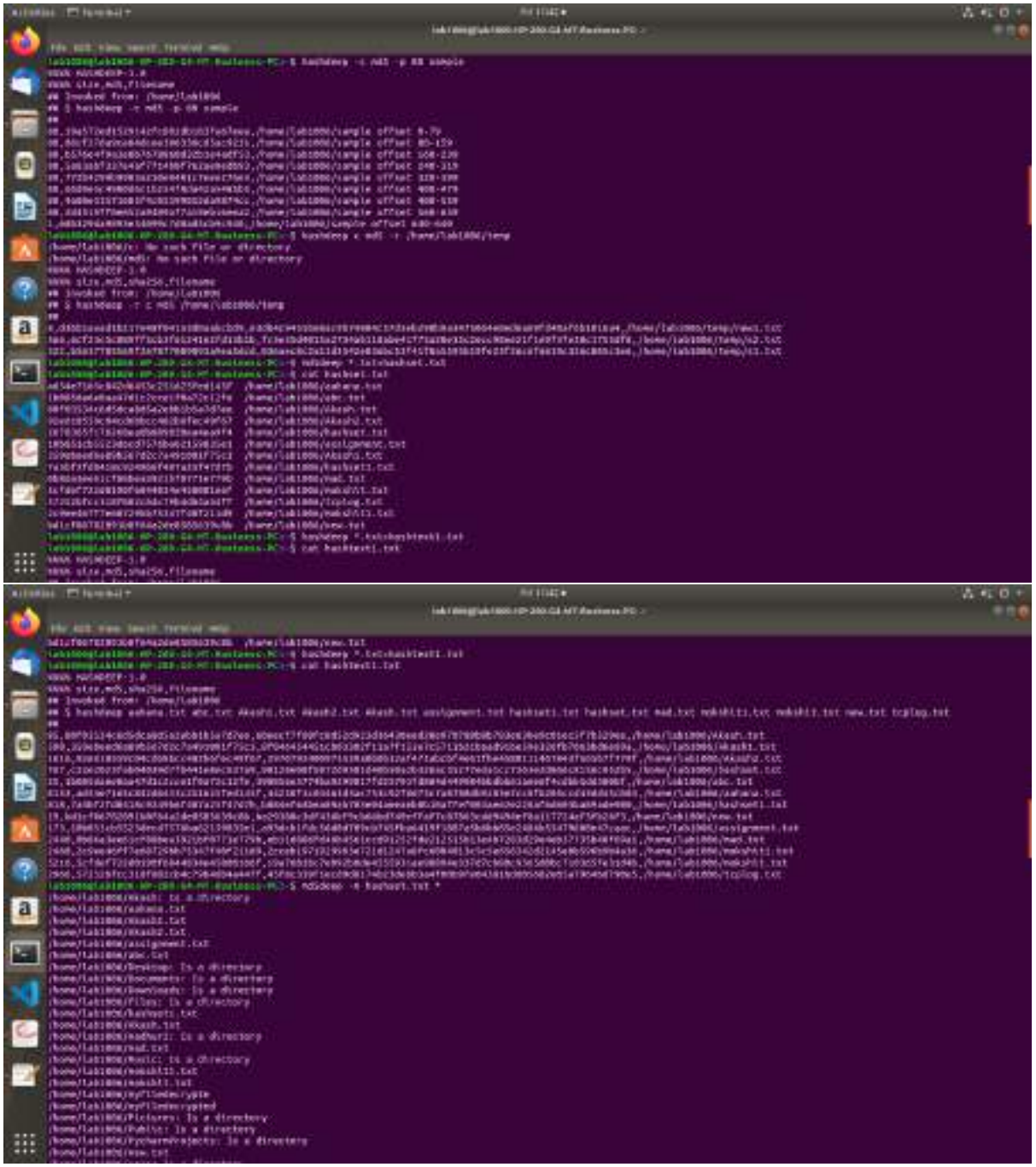
21. Rename one of the files and audit n see the output

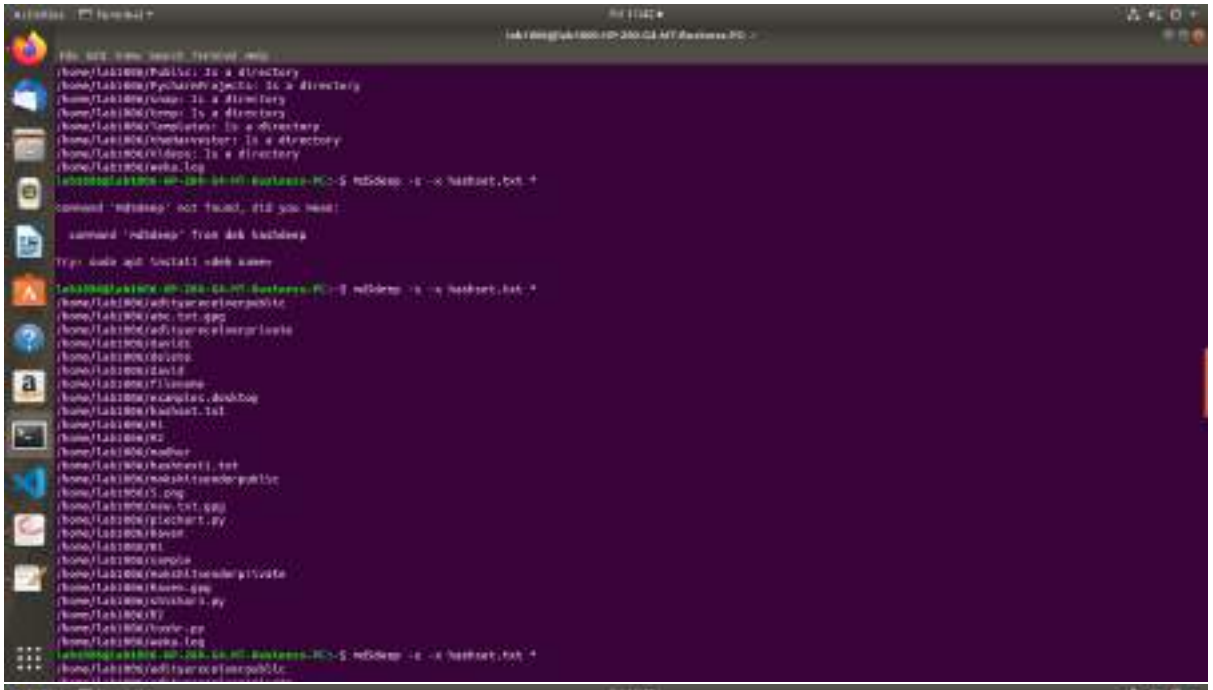
```
mv /home/lab006/myfiles/lab006.txt /home/lab006/myfiles/lab006.bak
hashdeep -v -a -r -k hashset1.txt /home/lab006/myfiles
```

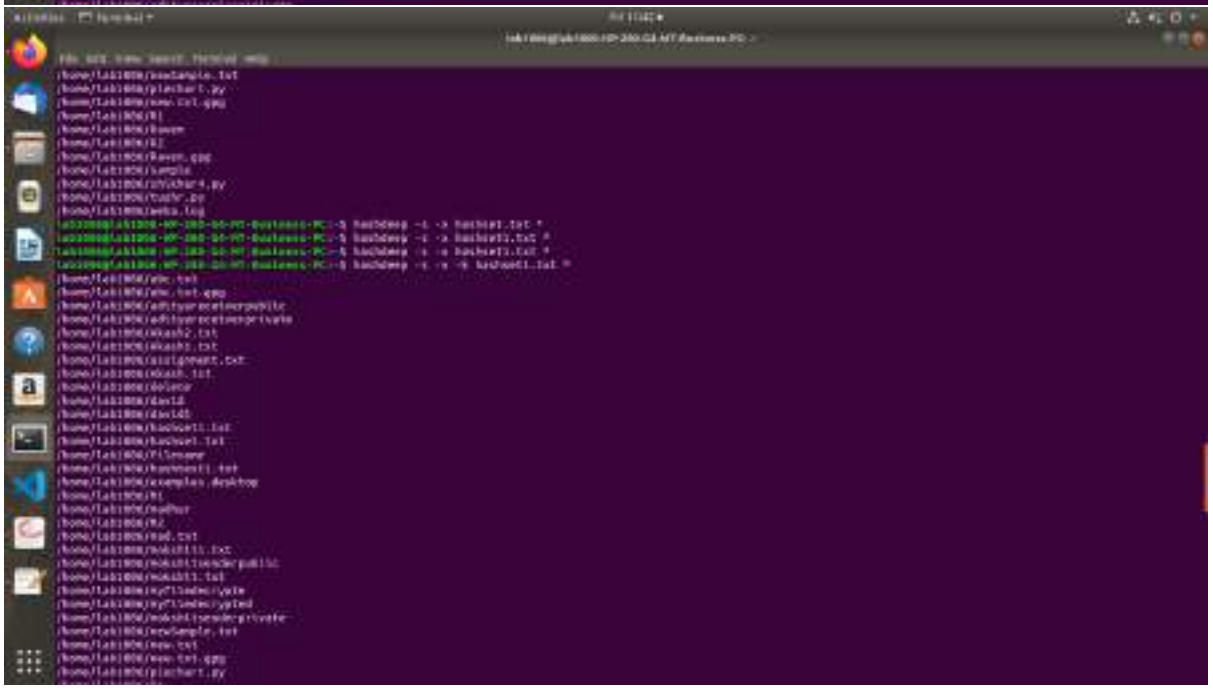
22. For verbose output of audit

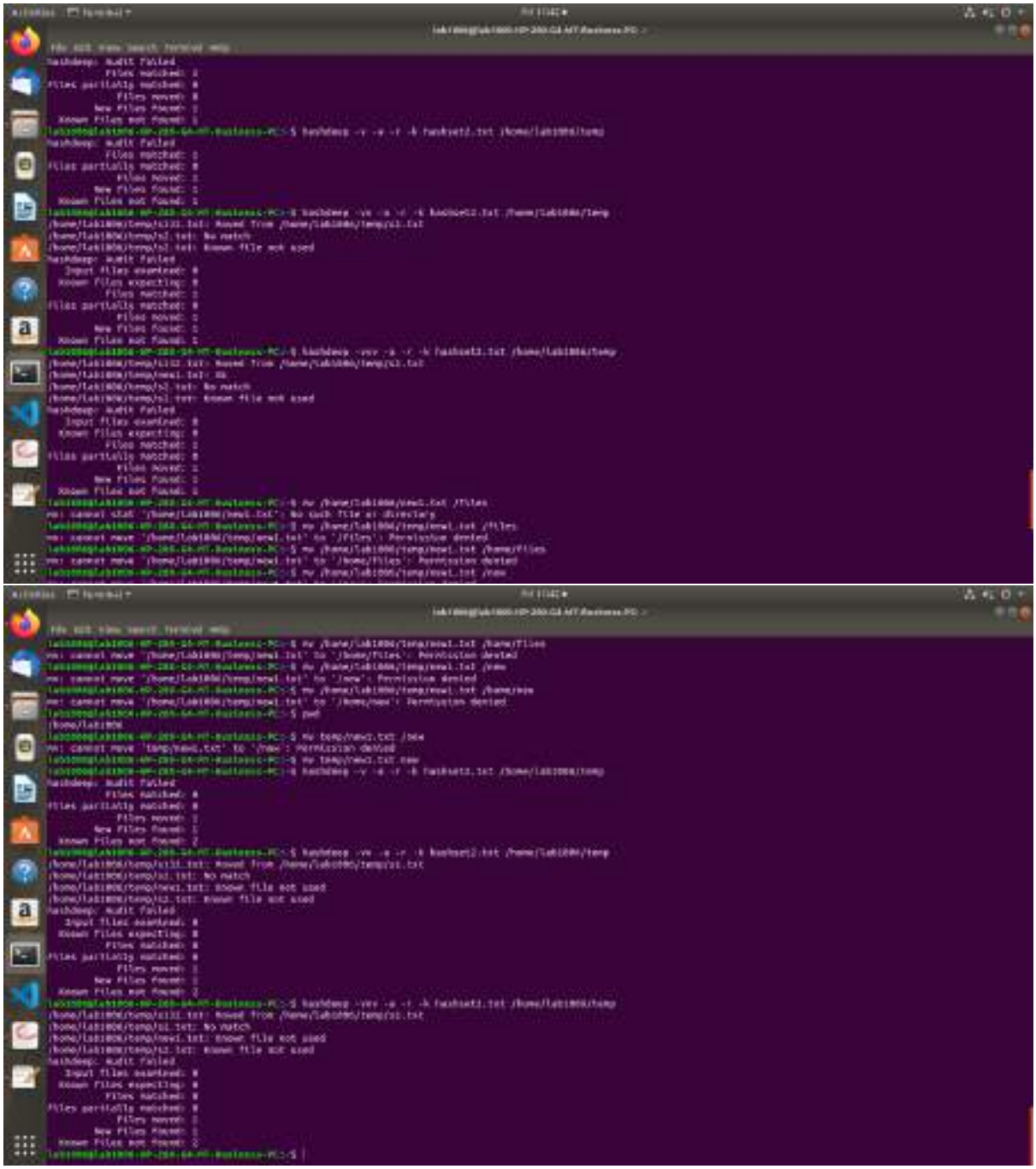
```
hashdeep -vv -a -r -k hashset1.txt /home/lab006/myfiles
hashdeep -vvv -a -r -k hashset1.txt /home/lab006/myfiles
```











CONCLUSION:

In this experiment we studied the need for hashing and its significance in data verification, security, and data structure optimization. Also different hasdeep commands for generating, matching and auditing hash of files.

Roll Number : 06

Name : Prasad Sunil Arote

Date : 08/08/2023

Lab Assignment No.6

Aim: Study the use of network reconnaissance tools like whois , dig , traceroute , nslookup, nikto, Dmitry to gather information about networks and domain registrars.

Lab Outcome Attained: LO3 : Explore the different network reconnaissance tools to gather information about networks .

Theory:

Q NO. 1 What is the important information that attackers look for using whois commands and what attacks can be performed using this information?

The WHOIS command is used to query a database that contains registration information about domain names, IP addresses, and autonomous system numbers. This information is publicly available and can be useful for legitimate purposes like contacting domain owners or network administrators. However, attackers can also use this information to gather intelligence for malicious activities. Here's what attackers might look for using the WHOIS command and potential attacks that can be performed using this information:

1. Domain Ownership and Administrative Contacts: Attackers can use WHOIS to identify the owner of a domain, along with administrative, technical, and billing contacts. This information might provide insights into potential targets for phishing attacks, social engineering, or domain hijacking.

2. Domain Expiration Dates: Attackers can identify domains that are about to expire. They might attempt to register expired domains to launch phishing campaigns, distribute malware, or engage in other malicious activities under a previously reputable domain name.

3. Nameservers: Knowing the nameservers associated with a domain can help attackers understand the infrastructure of a target. This information can be used to identify potential vulnerabilities in the DNS setup and launch attacks like DNS hijacking or DNS amplification attacks.

4. IP Address Ranges: Attackers can use WHOIS to identify IP address ranges owned by organizations. This information can help them target specific networks for attacks such as reconnaissance, network scanning, or even DDoS attacks.

5. Contact Email Addresses: The email addresses listed in the WHOIS records can be used for spear-phishing campaigns or social engineering attempts, as attackers can craft convincing messages that appear to be from legitimate sources.

6. Organizational Information: WHOIS records often contain details about the organization associated with a domain or IP address. Attackers can use this information for targeted attacks, such as crafting tailored social engineering messages.

7. Registrar and Hosting Provider: Information about the domain registrar and hosting provider can provide attackers with insights into the technical infrastructure. They might target vulnerabilities specific to those providers or impersonate support personnel to gain unauthorized access.

8. Geographical Location: By determining the geographical location of an organization's servers or infrastructure, attackers can plan attacks that are geographically targeted, such as launching attacks from regions with less stringent cybercrime laws.

Potential attacks that can be performed using this information include:

- Phishing Attacks: Armed with accurate contact information, attackers can craft convincing phishing emails or messages that appear to be from legitimate sources.

- Domain Hijacking: Attackers can use information about domain ownership and administrative contacts to impersonate domain owners and request unauthorized changes to domain settings, leading to domain hijacking.

- Social Engineering: Attackers can use the collected information to impersonate legitimate personnel, such as IT administrators or support personnel, and manipulate individuals into disclosing sensitive information or granting unauthorized access.

- Targeted Malware Attacks: Attackers can use the gathered information to tailor malware attacks to specific organizations or individuals.

- DNS Attacks: Information about nameservers and IP addresses can aid attackers in launching DNS-related attacks, such as DNS cache poisoning, DNS hijacking, and DDoS attacks.

To mitigate these risks, organizations should be cautious about the information they publicly disclose in WHOIS records and regularly monitor their online presence for signs of malicious activity. Additionally, implementing strong cybersecurity measures and employee training can help defend against the various types of attacks that can be launched using WHOIS information.

Q no. 2 How traceroute command works in order to trace the route of given host ?

The `traceroute` command is a network diagnostic tool used to trace the route that packets take from your computer to a destination host on a network, such as a website or server. It helps you identify the path that network traffic follows and the IP addresses of the intermediate routers or switches it passes through. This information can be crucial for troubleshooting network connectivity issues or understanding the network topology.

Here's how the `traceroute` command works:

1. Sending ICMP or UDP Packets: The `traceroute` command works by sending a series of packets to the destination host. It uses either ICMP (Internet Control Message Protocol) or UDP (User Datagram Protocol) packets with gradually increasing Time to Live (TTL) values. The TTL value represents the maximum number of hops (routers or switches) a packet can pass through before being discarded. Each intermediate device decrements the TTL value by 1 before forwarding the packet.

2. Recording Responses: As the packets travel through the network, each intermediate device decrements the TTL value. If the TTL reaches zero, the device discards the packet and sends an ICMP "Time Exceeded" message back to the source (the `traceroute` tool). The source can then determine the IP address of the device that discarded the packet.

3. Hop-by-Hop Discovery: By sending multiple packets with increasing TTL values, `traceroute` can discover the sequence of devices the packets pass through. The first packet will likely reach the first

router, the second packet might reach the second router, and so on. This process continues until the packets reach the destination host.

4. Displaying Results: The `traceroute` command displays the results of each packet's journey in terms of IP addresses, hostnames (if available), and round-trip times. The round-trip time is the time it takes for a packet to travel from your computer to an intermediate device and back. This information helps you analyze the latency introduced by each hop.

5. Completion: Once a packet successfully reaches the destination host, the `traceroute` command stops sending packets and displays the complete route from your computer to the destination. It typically displays the IP addresses or hostnames of all the intermediate devices the packets traversed.

It's important to note that some network devices or firewalls might be configured to not respond to ICMP or UDP packets, which can result in incomplete or inaccurate `traceroute` results. In such cases, you might see asterisks (*) or timeouts for certain hops.

In summary, the `traceroute` command is a valuable tool for diagnosing network issues and understanding the network path packets take to reach a destination. It provides insights into the topology of the network and helps identify potential bottlenecks or points of failure.

Q No. 3 Explain dig command with various options.

The `dig` command is a versatile network diagnostic tool used to perform DNS (Domain Name System) queries and retrieve information about domain names, IP addresses, and DNS records. It is available on most Unix-like operating systems and provides various options to customize and control the type of DNS query and the information displayed in the output. Here's an explanation of the `dig` command with some of its common options:

Basic Usage:

dig [domain] [query-type]

Common `dig` Options:

1. -t (Type): This option specifies the type of DNS record to query. Common types include A (IPv4 address), AAAA (IPv6 address), MX (mail exchange), NS (name server), CNAME (canonical name), TXT (text), and SOA (start of authority).

   Example: `dig example.com -t MX`

2. -c (Class): Specifies the DNS class to query. The default is IN (Internet), which is the most commonly used class.

   Example: `dig example.com -c CH`

3. -q (Query): Specifies the query type without requiring the query type option `-t`.

   Example: `dig example.com A`

4. -4 and -6: These options force the use of IPv4 or IPv6, respectively, for name resolution.

   Example: `dig example.com -4`

5. +short: Displays a concise output by showing only the essential information, such as IP addresses or domain names. Useful for scripting.

Example: `dig example.com +short`

6. +trace: Provides a trace of the DNS delegation path from the root name servers down to the authoritative name servers for the queried domain.

Example: `dig example.com +trace`

7. +all: Displays all information available for the queried domain, including additional records like NS, MX, and SOA records.

Example: `dig example.com +all`

8. +noall: Displays only the answer section of the DNS response, omitting the authority and additional sections.

Example: `dig example.com +noall`

9. +stats: Provides statistics about the query, including the time taken for the query, the number of packets sent and received, and more

Example: `dig example.com +stats`

10. @server: Specifies the DNS server to use for the query. Useful for querying specific DNS servers.

Example: `dig example.com @8.8.8.8`

11. -x: Performs a reverse DNS lookup, where you provide an IP address, and `dig` tries to find the associated domain name.

Example: `dig -x 8.8.8.8`

These are just a few of the many options available with the `dig` command. `dig` is a powerful tool for querying and troubleshooting DNS-related issues. By using different options, you can customize the type of information you retrieve and gain insights into DNS records, server configuration, and network connectivity.

Q No. 4 Explain any two vulnerabilities detected for the website that you have scanned using nikto . Which attacks are possible if these vulnerabilities are exploited?

Some common examples of vulnerabilities that Nikto or similar web vulnerability scanners might detect on a website, along with potential attacks that could be possible if these vulnerabilities are exploited.

1. Cross-Site Scripting (XSS) Vulnerability:

Nikto might detect an XSS vulnerability on a website. XSS occurs when a web application allows users to inject malicious scripts into web pages that are viewed by other users. This can happen when user inputs are not properly sanitized and are directly included in the HTML output.

Potential Attacks:

If exploited, an attacker could inject malicious scripts into web pages, and when other users view those pages, their browsers will execute the malicious scripts within the context of the site. This can lead to various attacks, including:

- Session Hijacking: The attacker could steal session cookies or tokens, allowing them to impersonate the victim user.

- Defacement: The attacker might deface the website, altering its appearance or content.

- Phishing: Malicious scripts could create convincing fake login forms to steal user credentials.

- Malware Distribution: Attackers could deliver malware to site visitors' computers.

2. SQL Injection Vulnerability:

Nikto could detect an SQL injection vulnerability on the website. SQL injection occurs when an attacker is able to manipulate an application's SQL query by injecting malicious SQL code into user inputs that are improperly sanitized.

Potential Attacks:

Exploiting an SQL injection vulnerability can lead to the following attacks:

- Data Leakage: Attackers can extract sensitive data from the database, including usernames, passwords, and personal information.

- Data Modification: Attackers could alter, delete, or insert unauthorized data into the database.

- Authentication Bypass: Attackers could bypass login forms by injecting valid SQL statements, gaining unauthorized access.

- Database Takeover: In some cases, attackers can gain control over the entire database server.

It's important to note that the impact of these vulnerabilities depends on various factors, including the context of the application, the level of user privileges, and the sensitivity of the data being handled. To mitigate these vulnerabilities, developers should follow secure coding practices, validate and sanitize user inputs, and implement proper access controls. Regular security assessments, like those performed by tools like Nikto, can help identify and remediate such vulnerabilities before they are exploited by malicious actors.

Q No.5 Write commands for email harvesting and subdown harvesting?

Commands using two popular tools for email harvesting and subdomain harvesting:

1. Email Harvesting using theHarvester:

theHarvester is a tool designed to gather email addresses, subdomains, hosts, employee names,

open ports, and banners from different public sources.

To install theHarvester, you can use pip:

-> pip install theHarvester

Once installed, you can use the following command to perform email harvesting for a specific

domain (e.g., example.com) using Google as the data source:

-> theHarvester -d example.com -b google

You can replace "example.com" with the target domain of your choice. The -b google option tells theHarvester to use Google as the data source for email harvesting.

2. Subdomain Harvesting using subfinder:

subfinder is a powerful subdomain enumeration tool that uses various public sources to find subdomains associated with a domain.

To install subfinder , you can use the following command:

-> GO111MODULE=on go get -v github.com/projectdiscovery/subfinder/v2/cmd/subfinder

Once installed, you can use the following command to perform subdomain harvesting for a specific domain e.g., example.com:

-> subfinder -d example.com

This command will search for subdomains associated with "example.com" and display the results on the terminal

Q No. 6 What are different functionalities provided by Dmitry . Write Dmitry commands for whois lookup , retrieve Netcraft info, search for subdomains , search for email addresses , do a TCP port scan , and save the output to example.txt for the domain example.com

Dmitry is a command-line tool for gathering information about a target domain. It provides functionalities such as WHOIS lookup, Netcraft information retrieval, subdomain enumeration, email address harvesting, TCP port scanning, and output file saving. Here are the commands to perform each of these tasks using Dmitry for the domain `example.com`:

1. WHOIS Lookup:

   Command: `dmitry -w example.com`

   Explanation: This command performs a WHOIS lookup for the domain `example.com`.

2. Retrieve Netcraft Info:

   Command: `dmitry -n example.com`

   Explanation: This command retrieves Netcraft information for the domain `example.com`.

3. Search for Subdomains:

   Command: `dmitry -s example.com`

  Explanation: This command searches for subdomains of the domain `example.com`.

4. Search for Email Addresses:

   Command: `dmitry -e example.com`

   Explanation: This command searches for email addresses associated with the domain `example.com`.

5. Do a TCP Port Scan:

Command: `dmitry -p example.com`

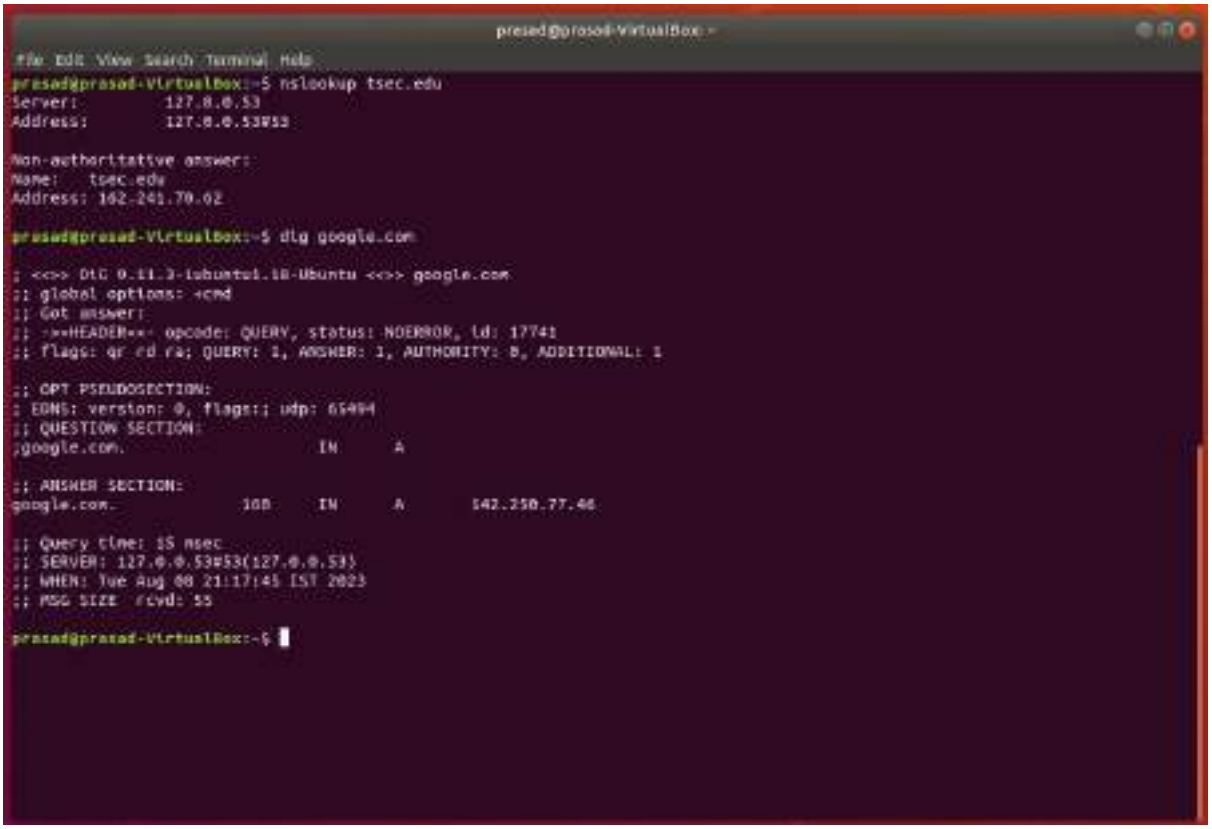
Explanation: This command performs a TCP port scan on the domain `example.com`.

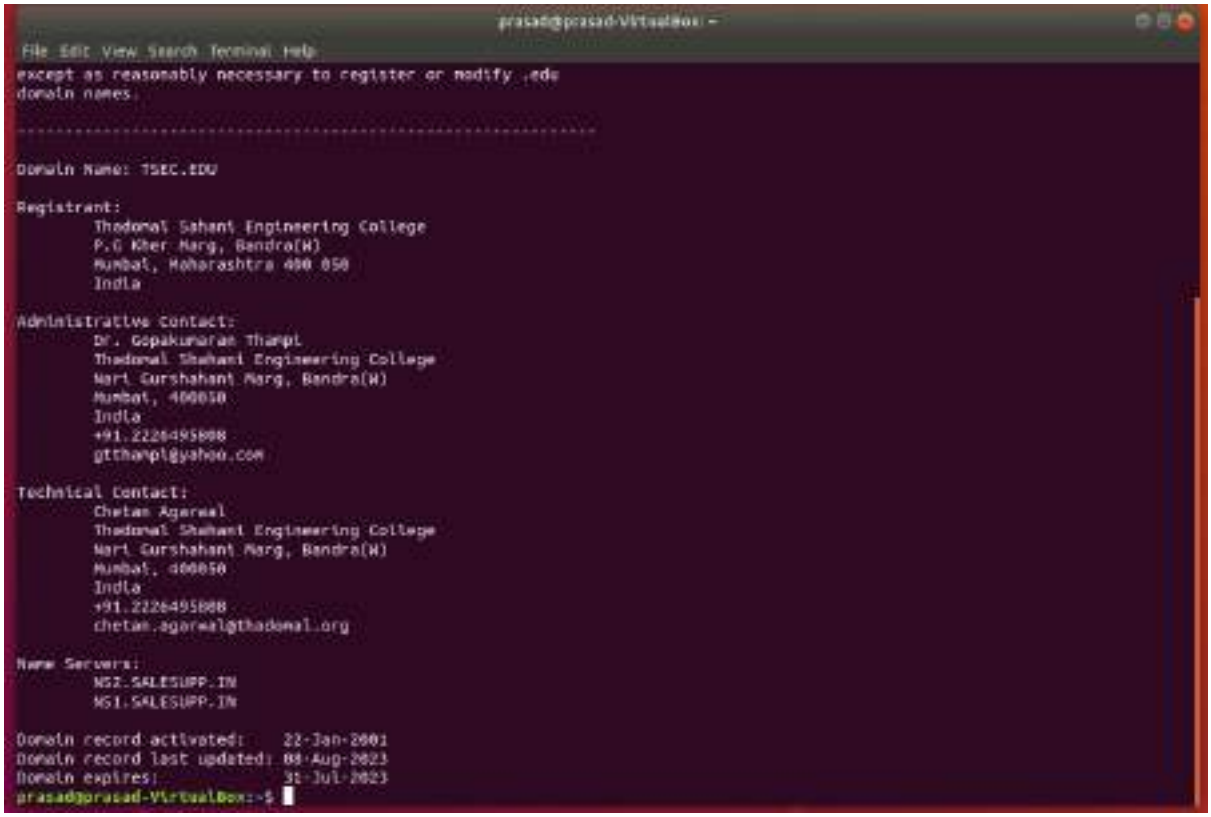
6. Save Output to a File:

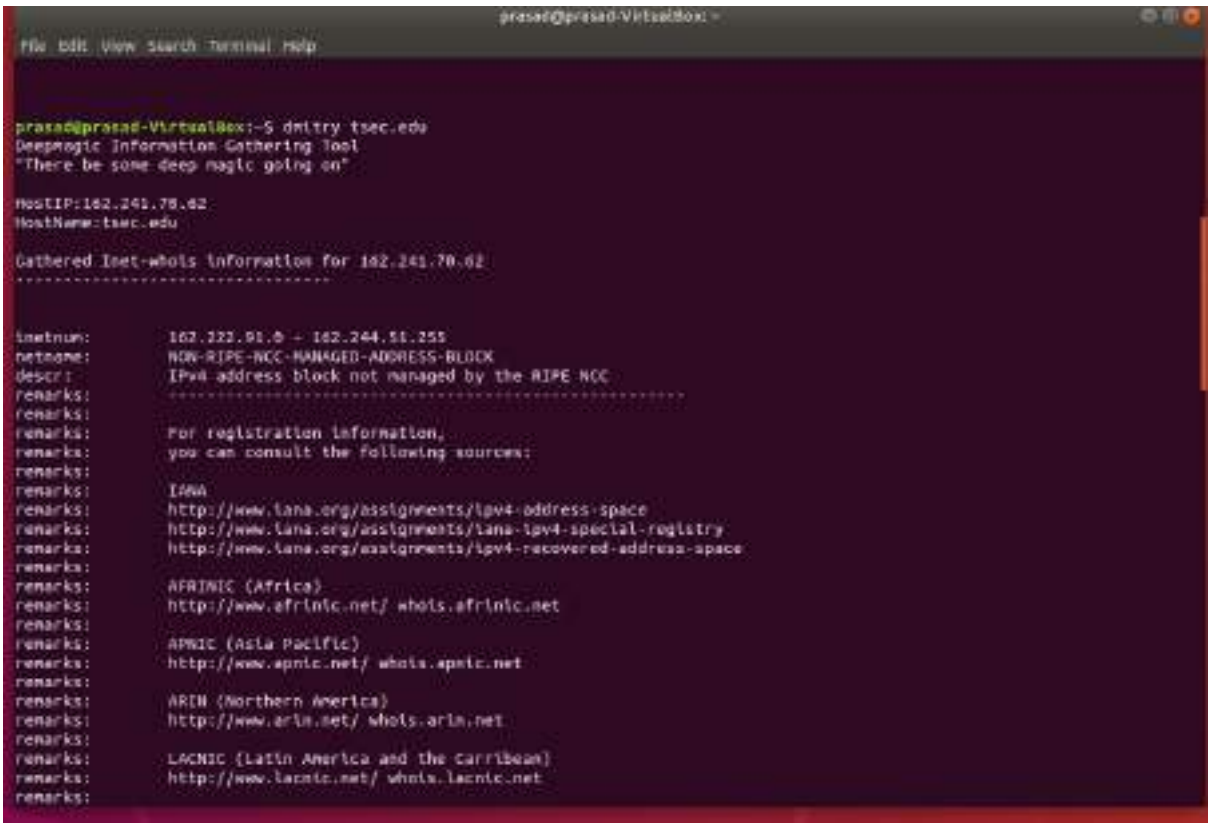
Command: `dmitry -o example.txt example.com`

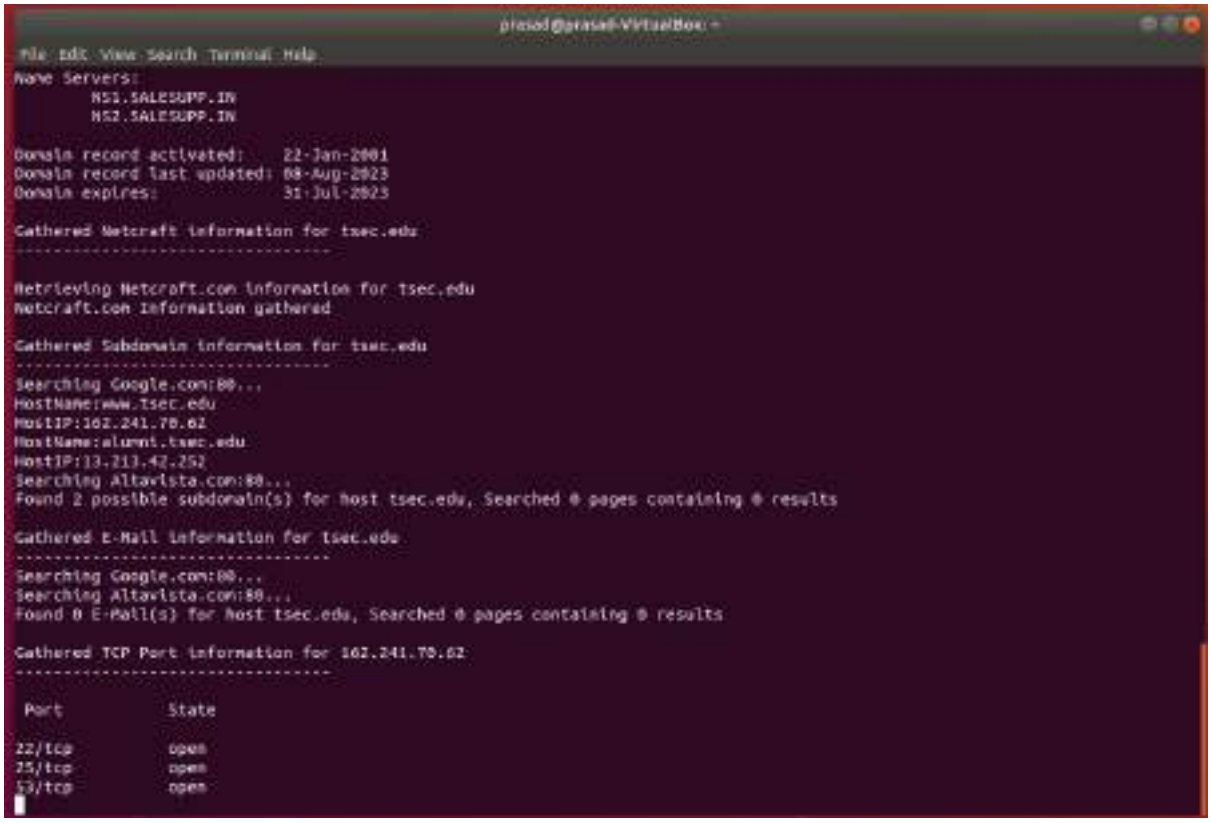
Explanation: This command redirects the output of Dmitry to the file `example.txt` for the domain `example.com`.

Output Screenshots:




prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
except as reasonably necessary to register or modify .edu
domain names.
Domain Name: TSEC.EDU
Registrant:
Thadomal Sahani Engineering College
P.G Kher Marg, Bandra(W)
Mumbai, Maharashtra 400 050
India
Administrative Contact:
Dr. Gopakumaran Thampi
Thadomal Shahani Engineering College
Nort Gurshahani Marg, Bandra(W)
Mumbai, 400050
India
+91.2226495000
gtthampi@yahoo.com
Technical Contact:
Chetan Agarwal
Thadomal Shahani Engineering College
Nort Gurshahani Marg, Bandra(W)
Mumbai, 400050
India
+91.2226495000
chetan.agarwal@thadomal.org
Name Servers:
NS2.SALESUPP.IN
NS1.SALESUPP.IN
Domain record activated: 22-Jan-2001
Domain record last updated: 08-Aug-2023
Domain expires: 31-Jul-2023
prasad@prasad-VirtualBox:~$


prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
prasad@prasad-VirtualBox:~$ dmitry tsec.edu
Deepmagic Information Gathering Tool
"There be some deep magic going on"
HostIP:162.241.70.62
HostName:tsec.edu
Gathered Inet-whois information for 162.241.70.62
inetnum: 162.122.91.0 - 162.244.51.255
netname: NON-RIPE-NCC-MANAGED-ADDRESS-BLOCK
descr: IPv4 address block not managed by the RIPE NCC
remarks:
remarks: For registration information,
remarks: you can consult the following sources:
remarks:
remarks: IANA
remarks: http://www.iana.org/assignments/ipv4-address-space
remarks: http://www.iana.org/assignments/iana-ipv4-special-registry
remarks: http://www.iana.org/assignments/ipv4-recovered-address-space
remarks:
remarks: AFRINIC (Africa)
remarks: http://www.afrinic.net/ whois.afrinic.net
remarks:
remarks: APNIC (Asia Pacific)
remarks: http://www.apnic.net/ whois.apnic.net
remarks:
remarks: ARIN (Northern America)
remarks: http://www.arin.net/ whois.arin.net
remarks:
remarks: LACNIC (Latin America and the Carribean)
remarks: http://www.lacnic.net/ whois.lacnic.net
remarks:



Conclusion : In summary, our exploration of network and web security tools has highlighted their critical roles. Traceroute, Dig, Nikto, and Dmitry offer insights into network paths, DNS information and vulnerability assessment. Ethical and responsible use is essential to leverage these tools for safeguarding systems and data against potential threats.

Roll No 06

Name- Prasad Sunil Arote

Date- 28-08-2023

Lab Assignment 7

AIM: Study of packet sniffer tools TCPDUMP.

LO3: Explore the different network reconnaissance tools to gather information about networks.

THEORY:

What is TCPDUMP and how to install it?

Tcpdump is a command-line packet analyzer that allows you to capture and analyze network traffic in real-time. It's commonly used for troubleshooting network issues, analyzing network behavior, and diagnosing problems related to network communication. tcpdump captures packets as they travel through a network interface and provides detailed information about each packet, including source and destination addresses, protocol information, payload data, and more.

Linux (Debian/Ubuntu):

Open a terminal and run the following command to install tcpdump:

sudo apt-get update

sudo apt-get install tcpdump

Explain various commands in tcpdump to capture different types of packets.

tcpdump provides a wide range of commands and options to capture and analyze different types of packets. Here are some common tcpdump commands and filters to capture specific types of packets:

1. Capture All Traffic on a Specific Interface:

sudo tcpdump -i eth0

This captures all traffic on the "eth0" network interface.

2. Capture Traffic to or from a Specific IP Address:

sudo tcpdump host 192.168.1.100

This captures all traffic to or from the IP address "192.168.1.100".

3. Capture Traffic on a Specific Port:

sudo tcpdump port 80

This captures all traffic on port 80.

4. Capture Traffic Using a Specific Protocol:

sudo tcpdump icmp

This captures ICMP (ping) traffic.

5. Capture Traffic from a Specific Source IP:

sudo tcpdump src 192.168.1.200

This captures traffic originating from IP address "192.168.1.200".

6. Capture Traffic to a Specific Destination IP:

sudo tcpdump dst 192.168.1.100

This captures traffic directed to IP address "192.168.1.100".

7. Capture Traffic on a Specific Port Using a Protocol:

sudo tcpdump udp port 53

This captures UDP traffic on port 53 (DNS).

8. Capture Traffic Using a Combination of Filters:

sudo tcpdump src 192.168.1.100 and port 22

This captures traffic originating from IP address "192.168.1.100" and using port 22 (SSH).

9. Capture Traffic with Specific Packet Size:

sudo tcpdump greater 1000

This captures packets larger than 1000 bytes.

10. Capture Specific Number of Packets:

sudo tcpdump -c 10

This captures 10 packets and then exits.

11. Capture Packets Using Hexadecimal Filter:

sudo tcpdump -X 'tcp[13] & 2 != 0'

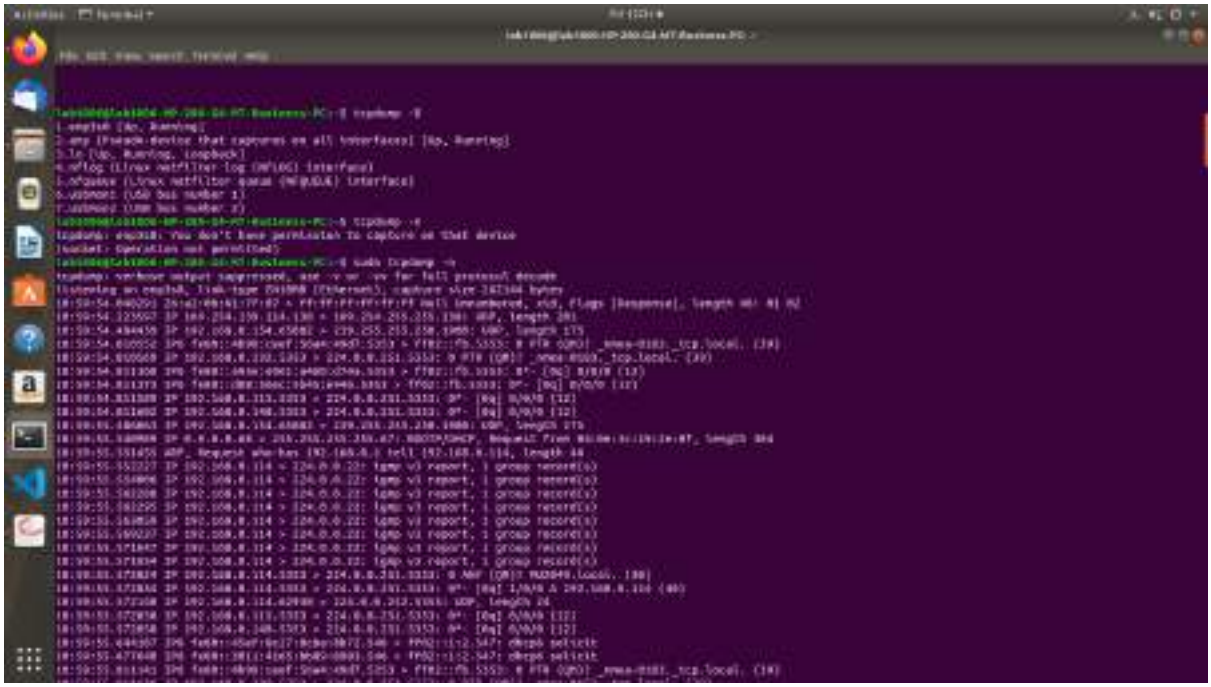
This captures only SYN packets (TCP packets with the SYN flag set).

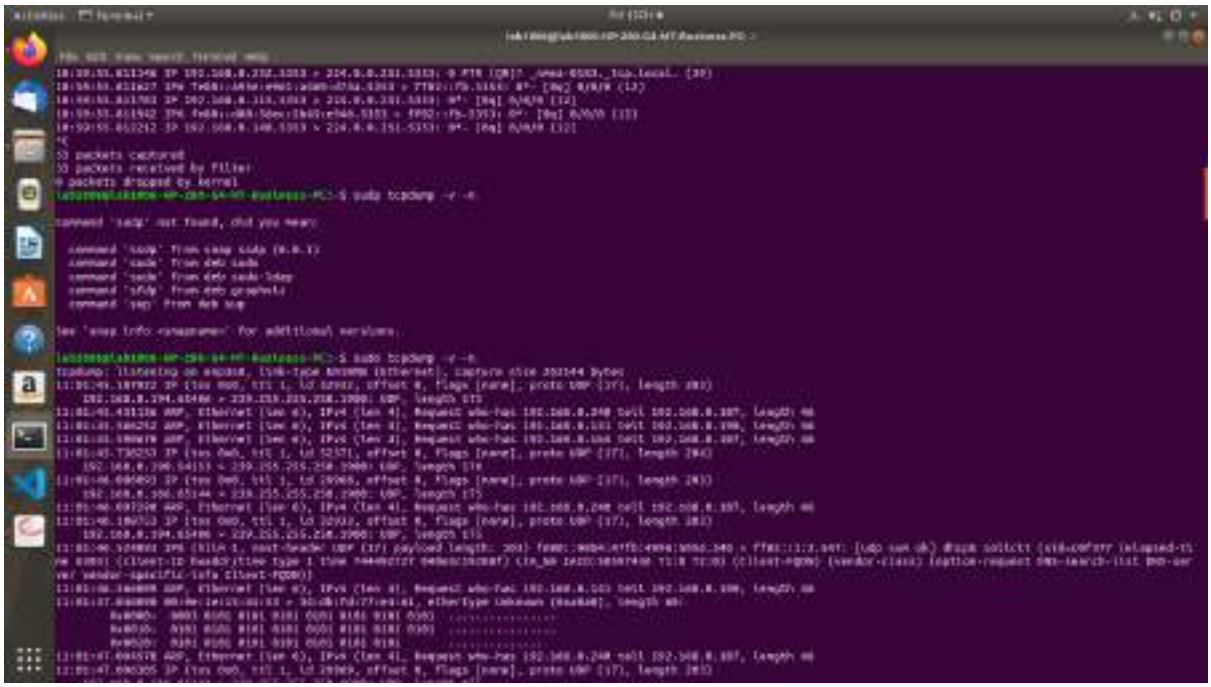
12. Capture and Save Output to a File:

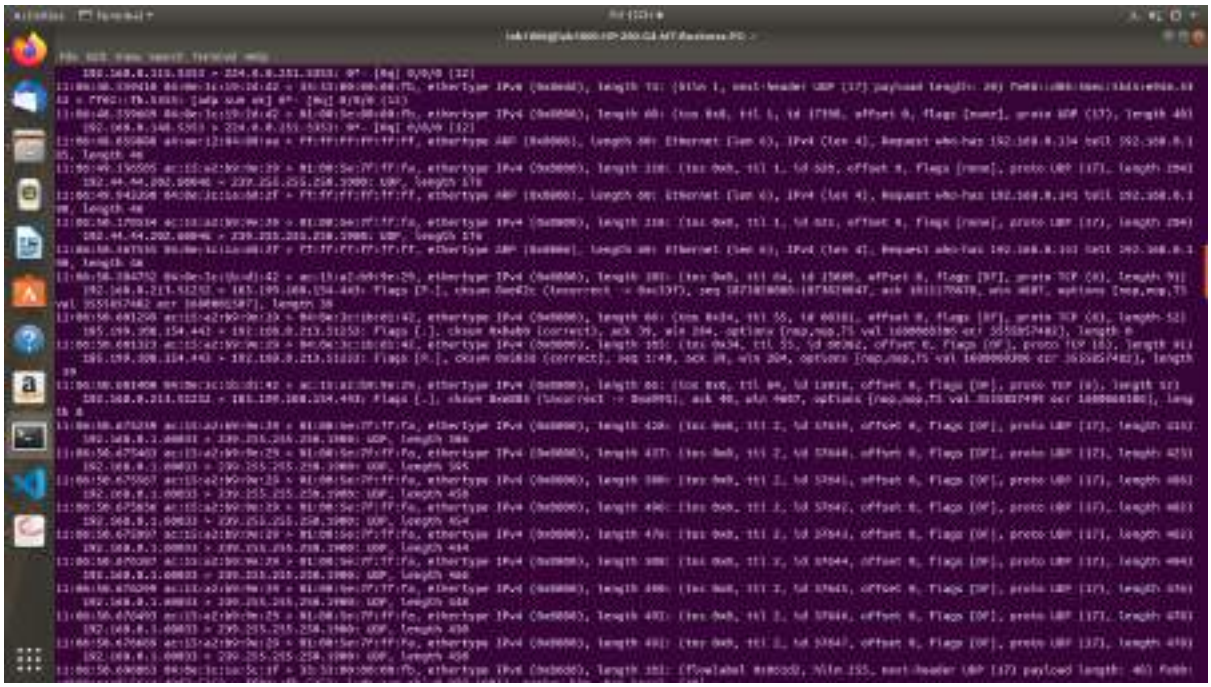
sudo tcpdump -i eth0 -w output.pcap

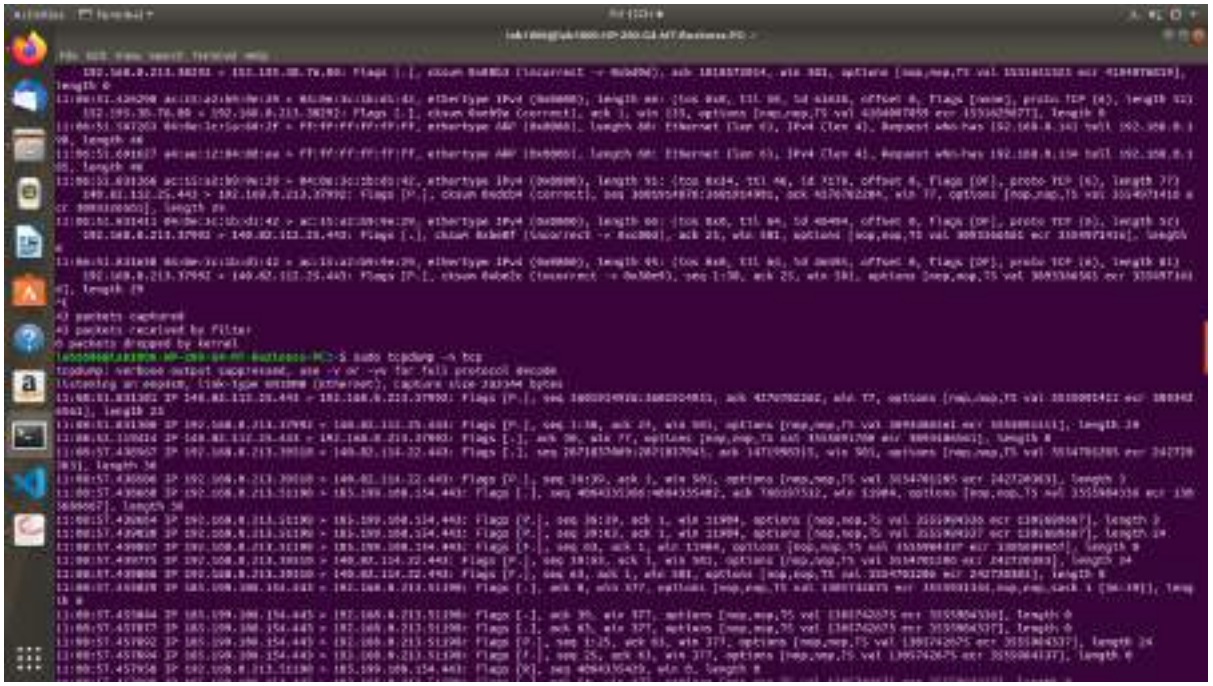
This captures traffic on the "eth0" interface and saves it to the "output.pcap" file.

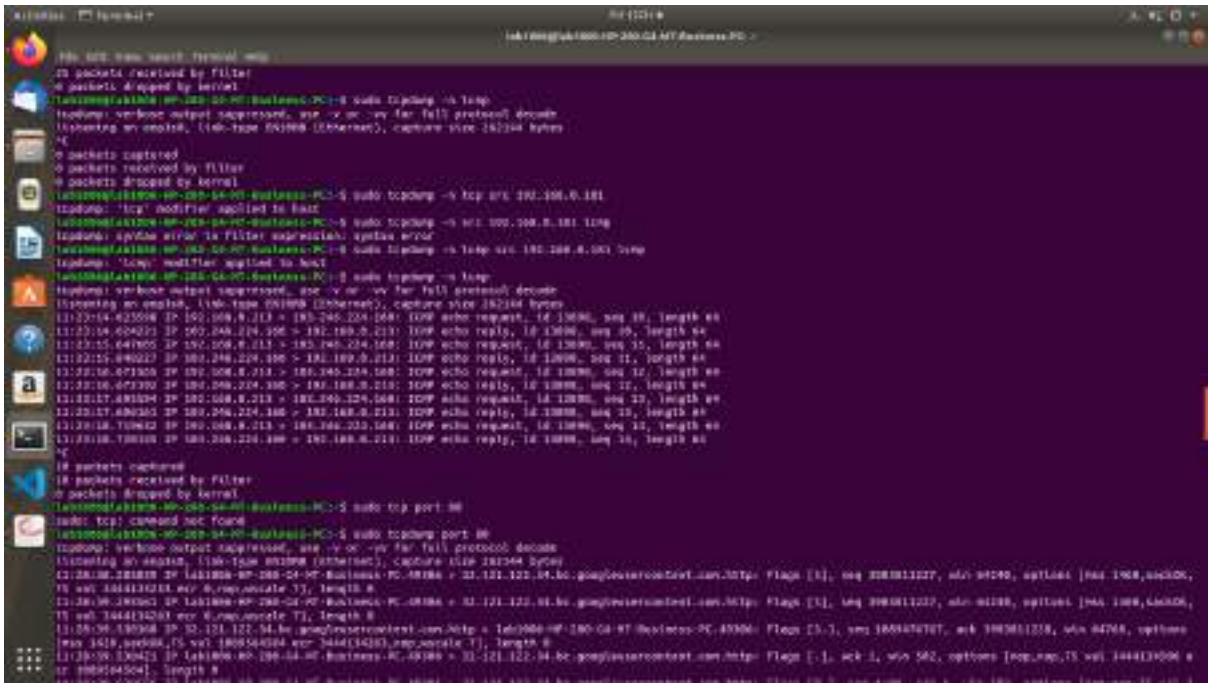
OUTPUT

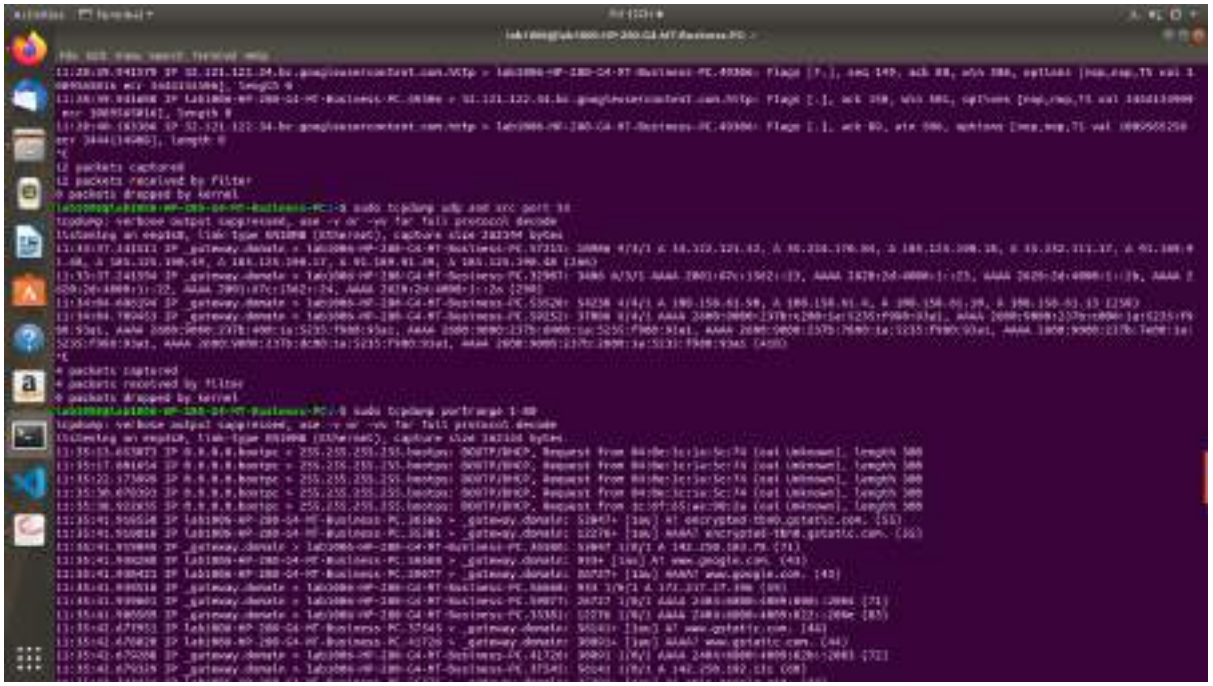


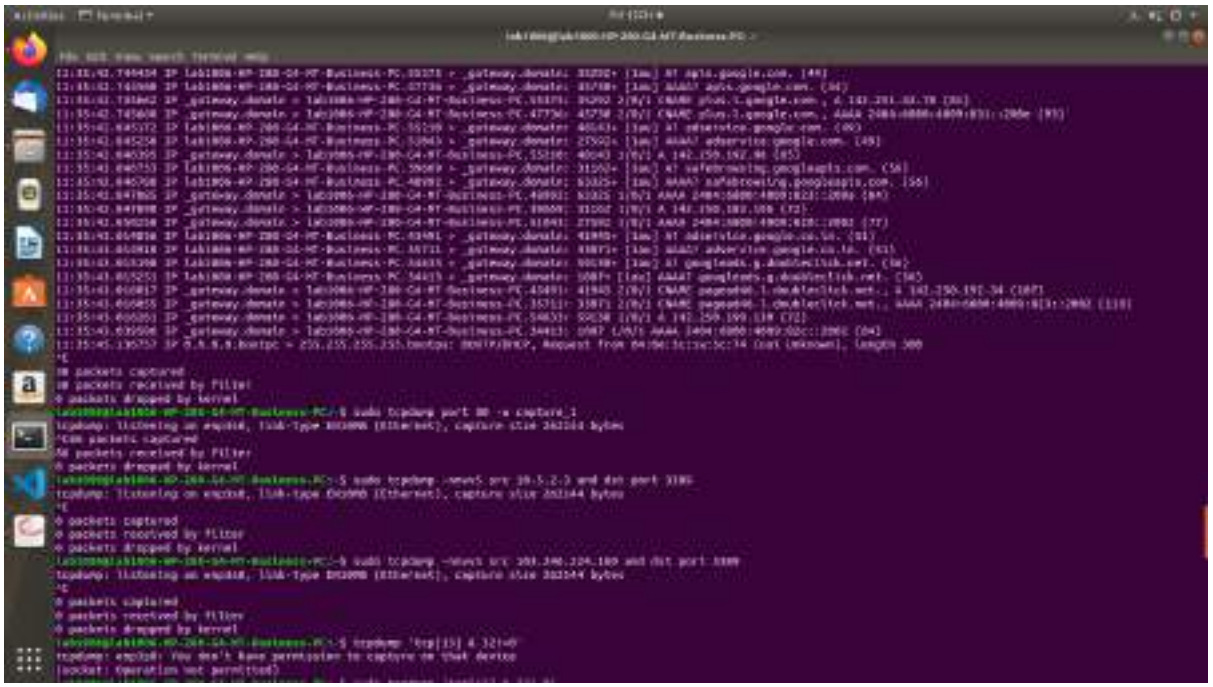


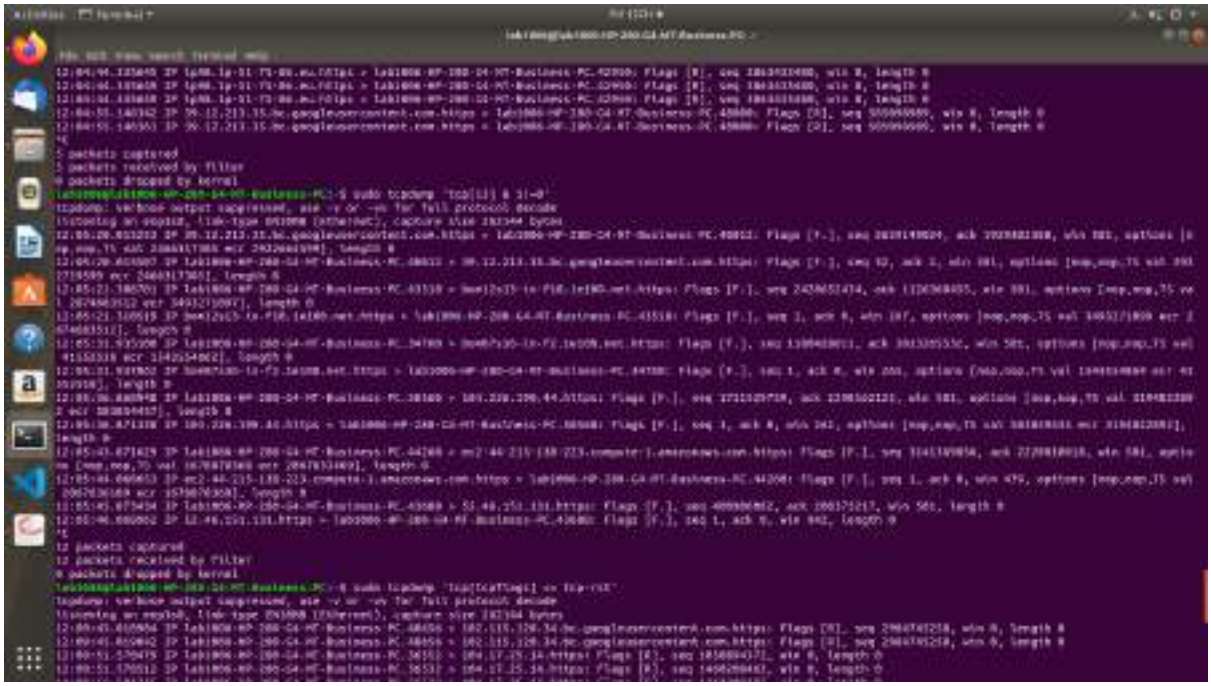


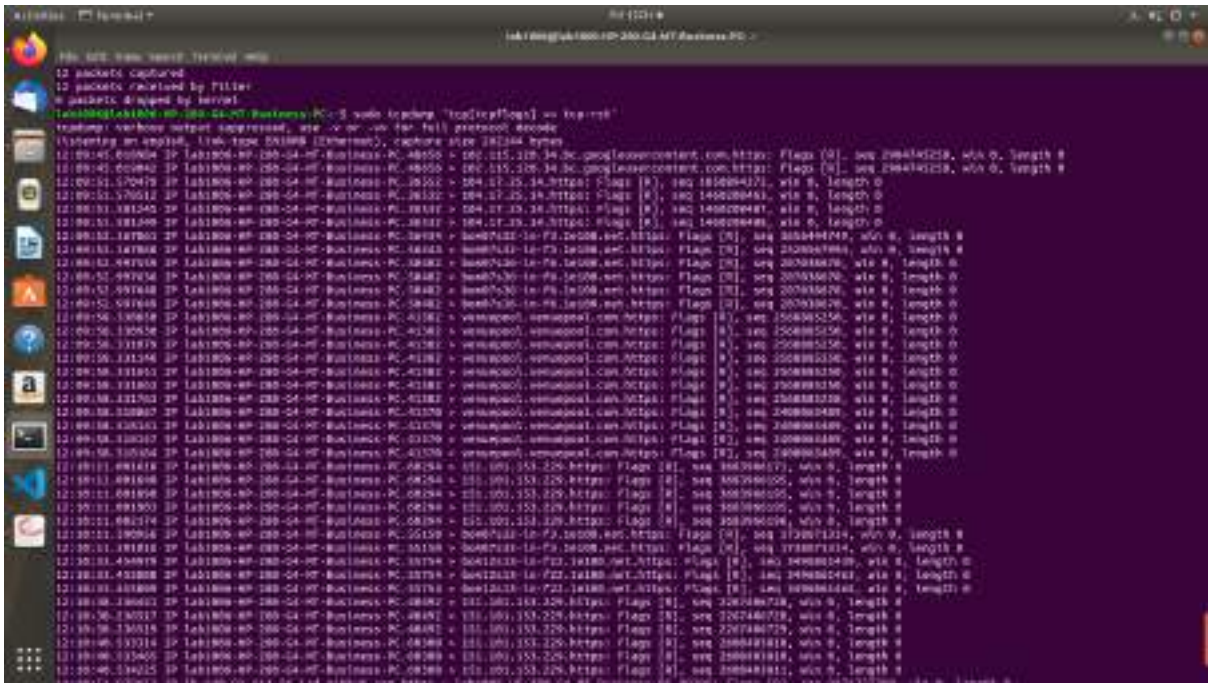












CONCLUSION:

We gained a practical understanding of how TCPDump can be employed to capture, dissect, and interpret network packets in real-time, offering valuable insights into network behavior, troubleshooting, and security assessment. By applying various filters and commands, we were able to capture specific types of traffic based on source and destination addresses, protocols, ports, and packet sizes.

Roll No 06

Name – Prasad Sunil Arote

Date - 07/09/2023

Lab Assignment 8

AIM: Installation of NMAP and using it with different options to scan open ports, perform OS fingerprinting, ping scan, TCP port scan, UDP port scan, etc.

LO4: Use tools like sniffers, port scanners and other related tools for analyzing packets in a network.

THEORY:

Port Scanning:

Port scanning is a network reconnaissance technique used to discover open ports on a target system. It involves sending requests to various ports on a target computer to determine which ports are open, closed, or filtered. This information is valuable for both legitimate network administrators and malicious hackers as it helps identify services running on a system and potential vulnerabilities.

NMAP:

Nmap (Network Mapper) is a widely used open-source tool for network discovery and security auditing. It provides a variety of scanning techniques and options to probe networks and identify open ports, services, and operating systems.

Different States of Ports:

1. Open: The target system actively accepts connections on the specified port. This indicates that a service is running and listening on that port.
2. Closed: The target system actively rejects connections on the specified port. This means there's no service listening on that port.
3. Filtered: The target system actively drops incoming packets, making it difficult to determine whether the port is open or closed. Firewalls or security measures often cause this state.
4. Unfiltered: Nmap cannot determine whether the port is open or closed due to the lack of response from the target system. This state indicates a less common configuration.
5. Open | Filtered: Nmap cannot reliably determine whether the port is open or filtered. This state often occurs when firewalls are in place.
6. Closed | Filtered: Nmap cannot reliably determine whether the port is closed or filtered. This state is also often the result of firewalls.

Port Scanning Techniques using NMAP:

TCP Connect Scan:

Command: nmap -sT target

Explanation: This scan establishes a full TCP connection to each specified port. It actively opens a connection to each target port to check if it's open. This method is reliable but not as stealthy as other scans because it leaves a clear trace in the target's logs.

TCP SYN Scan:

Command: nmap -sS target

Explanation: The SYN scan, also known as a half-open scan, sends SYN packets to target ports. If a port is open, it responds with a SYN-ACK packet, allowing Nmap to determine that the port is open. If the port is closed, it responds with a RST packet. This scan is stealthier than a connect scan.

FIN Scan:

Command: nmap -sF target

Explanation: In a FIN scan, Nmap sends FIN packets to target ports. If a port is closed, it responds with a RST packet, indicating that the port is closed. However, if the port is open, it ignores the packet. This scan is used to identify open ports without triggering alarms.

Null Scan:

Command: nmap -sN target

Explanation: A null scan involves sending TCP packets with no flags set (i.e., all flags set to zero) to target ports. Similar to the FIN scan, if a port is closed, it responds with a RST packet, but if the port is open, it ignores the packet. This scan can help identify open ports while evading detection.

XMAS Scan:

Command: nmap -sX target

Explanation: An XMAS scan sends packets with the FIN, URG, and PSH flags set to target ports. Like the FIN and Null scans, if a port is closed, it responds with a RST packet. If open, it usually doesn't respond. This scan can help identify open ports in stealthy scenarios.

ACK Scan:

Command: nmap -sA target

Explanation: The ACK scan sends ACK packets to target ports. If a port is unfiltered and open, it will respond with an RST packet. However, if the port is filtered or closed, it typically won't respond. This scan is primarily used to identify firewall rules.

Ping Sweep:

Command: nmap -sn target

Explanation: A ping sweep is used to discover live hosts in a network by sending ICMP echo requests (ping) to multiple IP addresses within a specified range. It helps identify which hosts are online and reachable.

Service and Version Detection:

Command: nmap -sV target

Explanation: This scan detects the services running on open ports and attempts to determine their versions by analyzing the responses from those services. It helps in identifying specific software and their versions.

Port and Port Range Scanning:

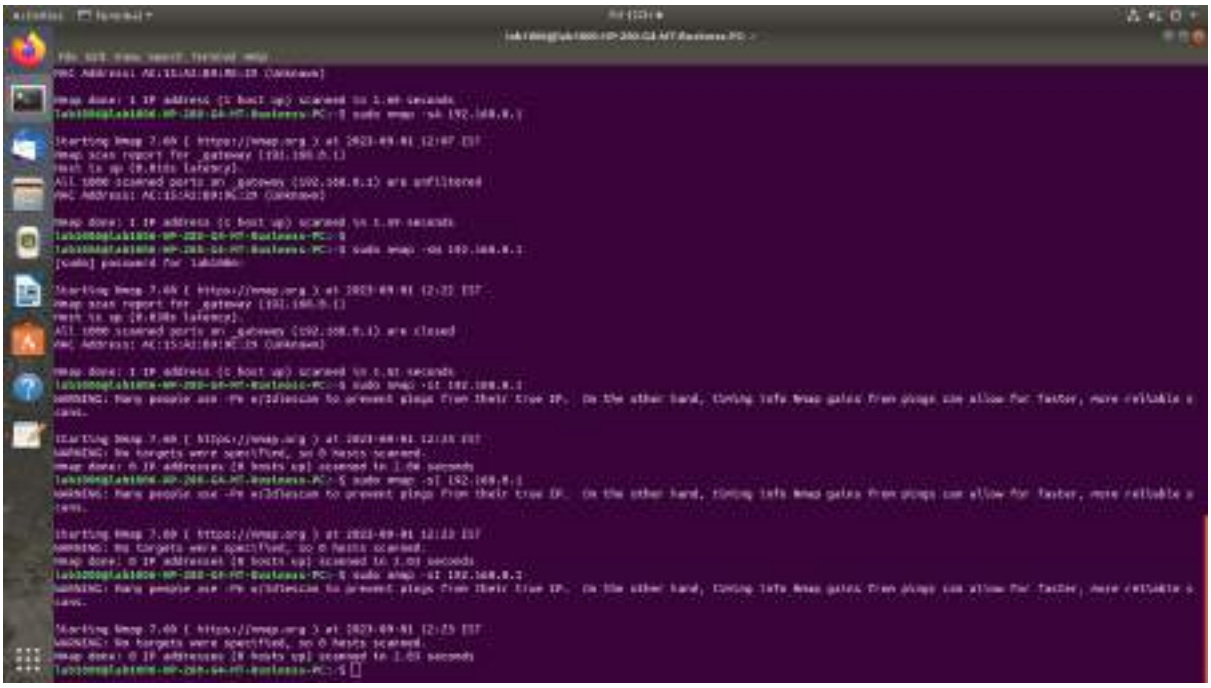
Command: nmap -p port(s) target

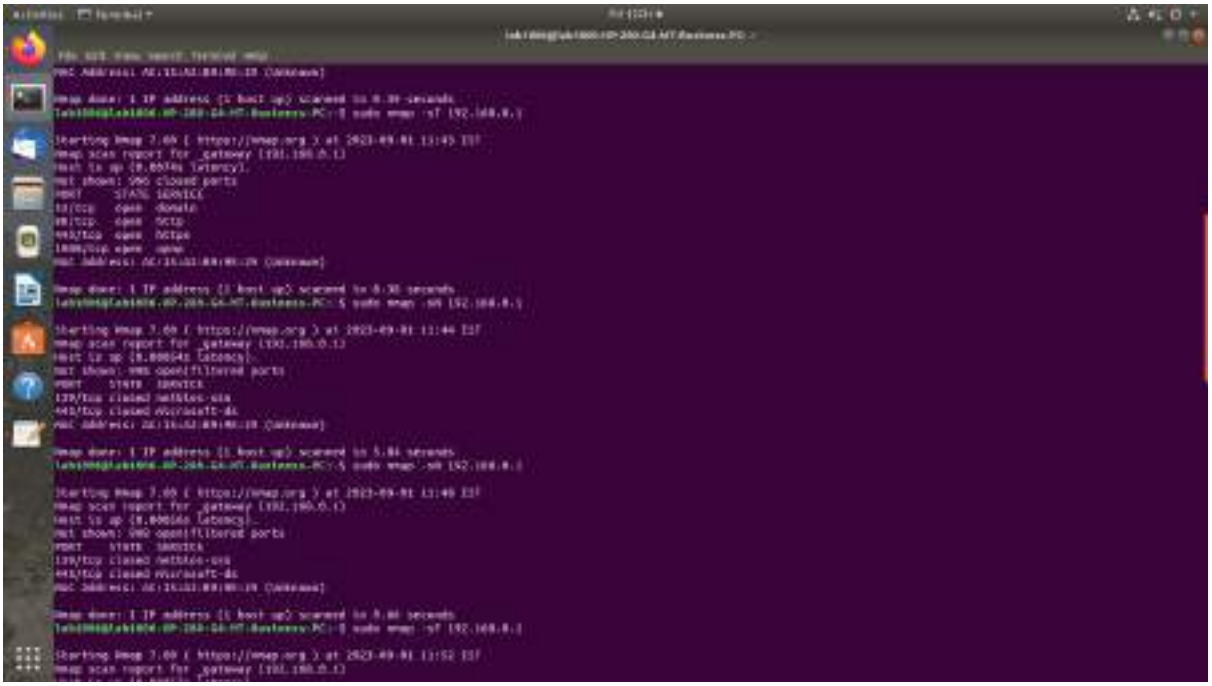
Explanation: You can use this command to specify specific ports or a range of ports to scan. For example, nmap -p 80,443 target scans only ports 80 and 443.

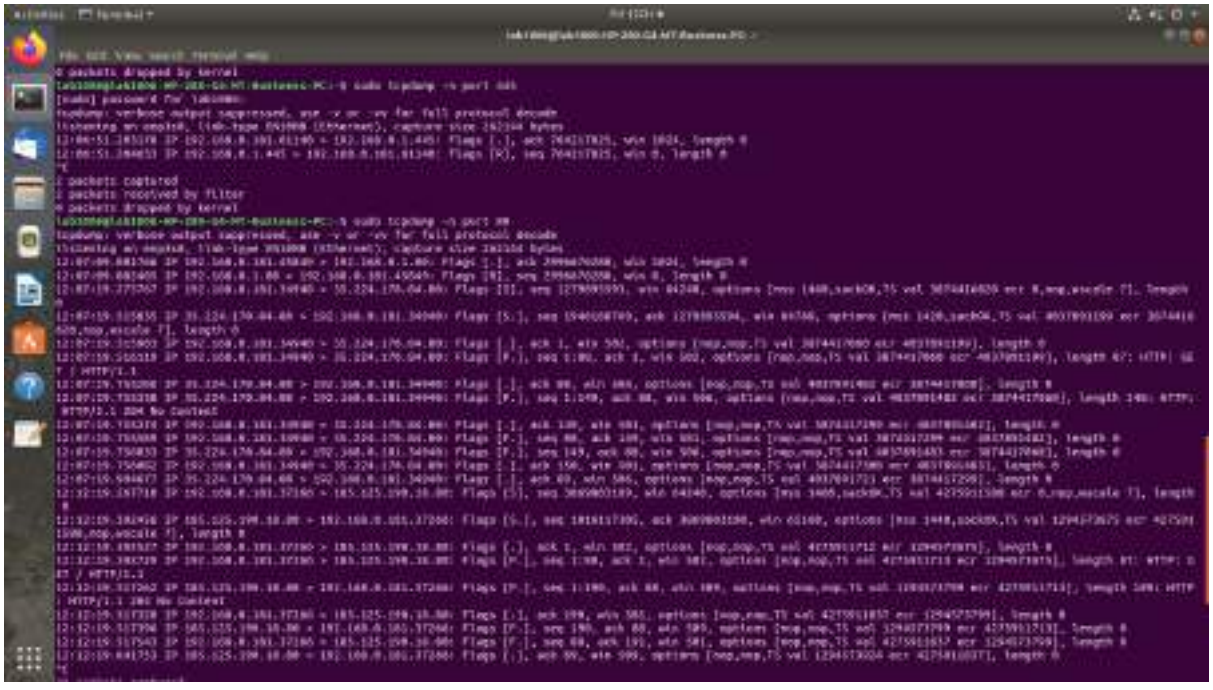
OS Fingerprinting:

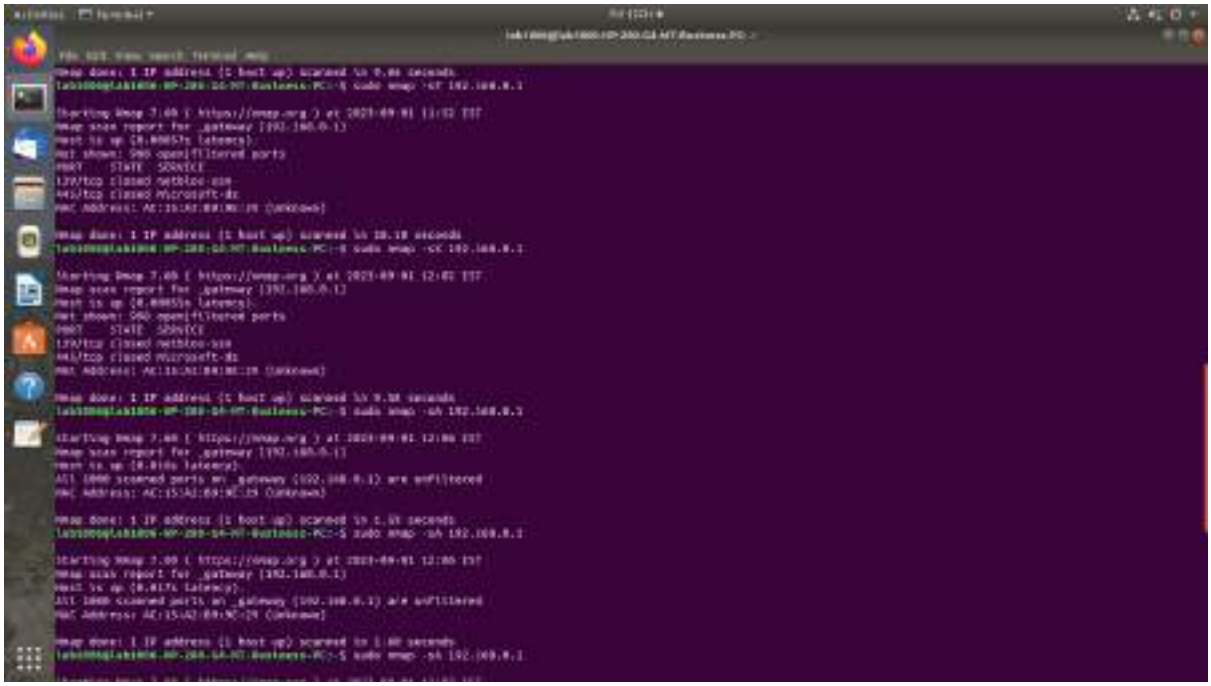
Command: nmap -O target

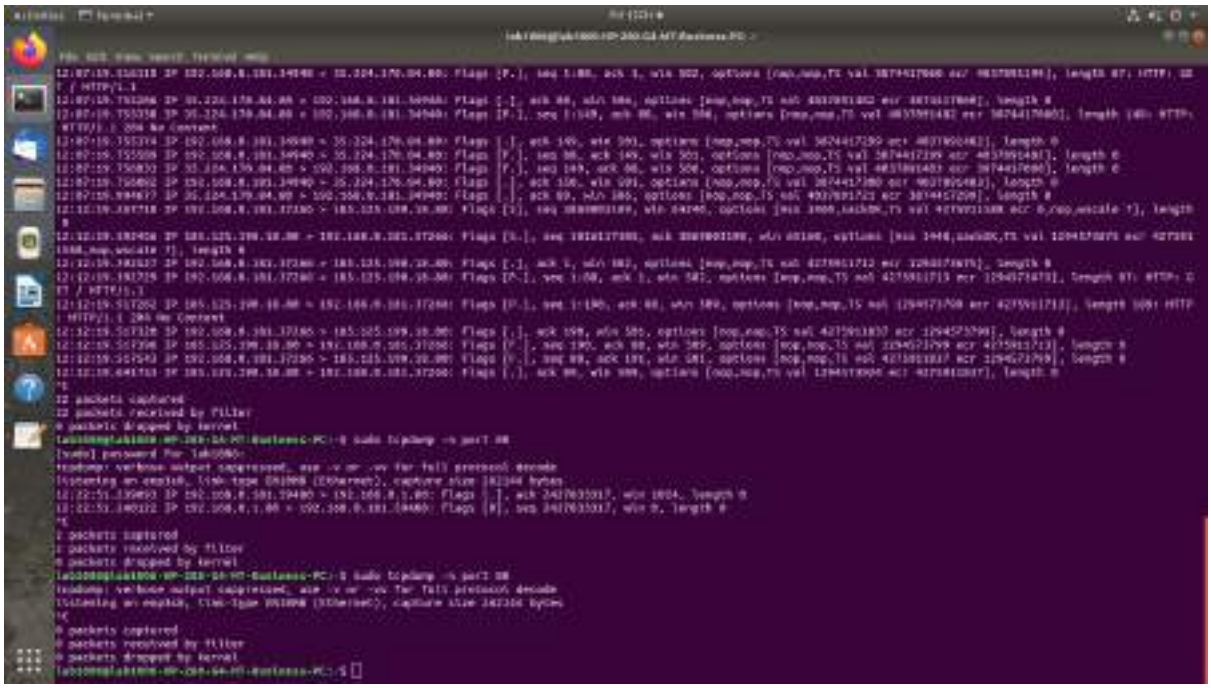
Explanation: This scan attempts to identify the operating system running on the target by analyzing various network responses and characteristics. Nmap compares these patterns to its database to make an educated guess about the OS.











Conclusion:

Port scanning is a crucial technique for network reconnaissance, helping administrators identify security weaknesses and ensuring proper configuration. Nmap provides a comprehensive set of scanning options for various scenarios, from identifying open ports to determining service versions and even fingerprinting the target's operating system. However, it's important to use these tools and techniques responsibly and with proper authorization, as unauthorized scanning can be considered malicious and illegal.

Roll No 06

Name: Prasad Sunil Arote

Date: 10/9/2023

Lab Assignment 9

AIM: Simulate DOS attack using HPING3.

LO5: Use open source tools to scan the networks for vulnerabilities and simulate attacks.

THEORY:

A Denial of Service (DoS) attack is a malicious attempt to disrupt the normal functioning of a computer system, network, or online service by overwhelming it with a flood of illegitimate requests or traffic. The primary goal of a DoS attack is to make a resource, such as a website or server, unavailable to its intended users. It does so by consuming the target's resources, such as bandwidth, processing power, or memory, to the point where it cannot handle legitimate requests.

Here are explanations of three common types of DoS attacks:

SYN Flood Attack:

A SYN flood attack is a type of network-based DoS attack that targets the three-way handshake process in the Transmission Control Protocol (TCP), which is used for establishing connections between devices on the internet.

In a TCP connection, the client sends a SYN (synchronize) packet to initiate a connection with a server. The server is expected to respond with a SYN-ACK (synchronize-acknowledgment) packet, and then the client responds with an ACK (acknowledgment) packet to complete the handshake and establish the connection.

In a SYN flood attack, the attacker sends a high volume of SYN packets to the target server, but they do not complete the handshake by sending the expected ACK packets. This leaves the server waiting for the final ACKs, tying up its resources and preventing it from accepting legitimate connections.

SYN flood attacks can quickly overwhelm a server's ability to handle incoming connections, leading to service disruption.

ICMP Flood Attack:

An ICMP (Internet Control Message Protocol) flood attack, also known as a "ping flood" attack, targets the ICMP protocol, which is used for network diagnostics, particularly the "ping" command.

In this type of attack, the attacker sends a high volume of ICMP echo requests (ping requests) to the target system. Each request typically generates a response from the target, creating a flood of traffic.

ICMP flood attacks can consume the target's network bandwidth and processing resources, making it difficult for legitimate network traffic to pass through. This results in network congestion and service degradation or unavailability.

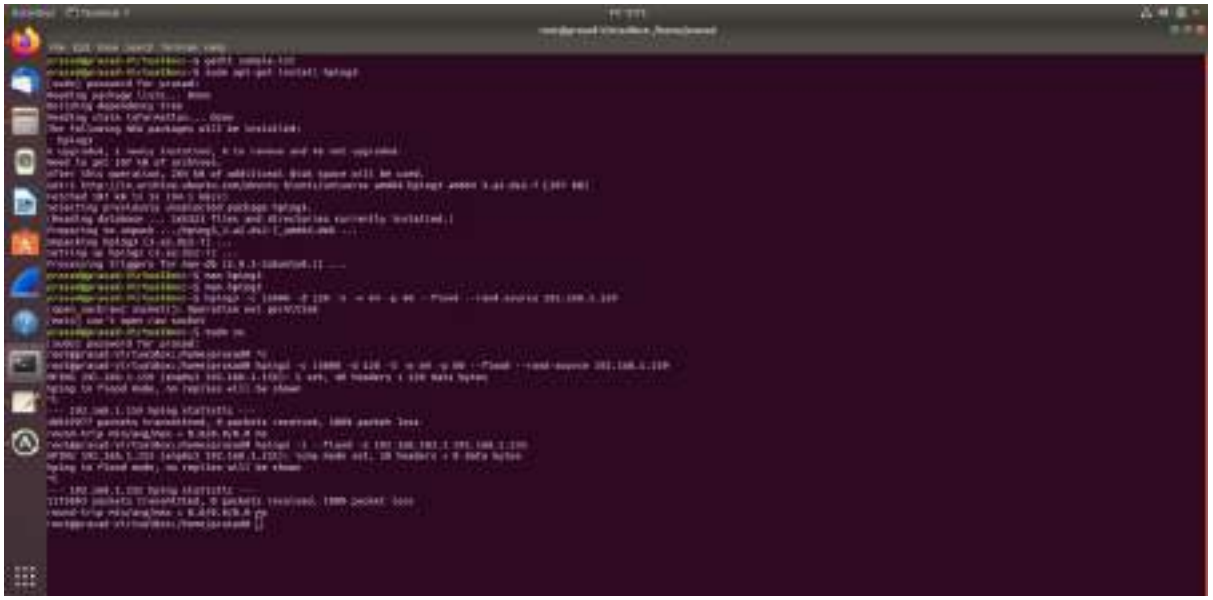
SMURF Attack:

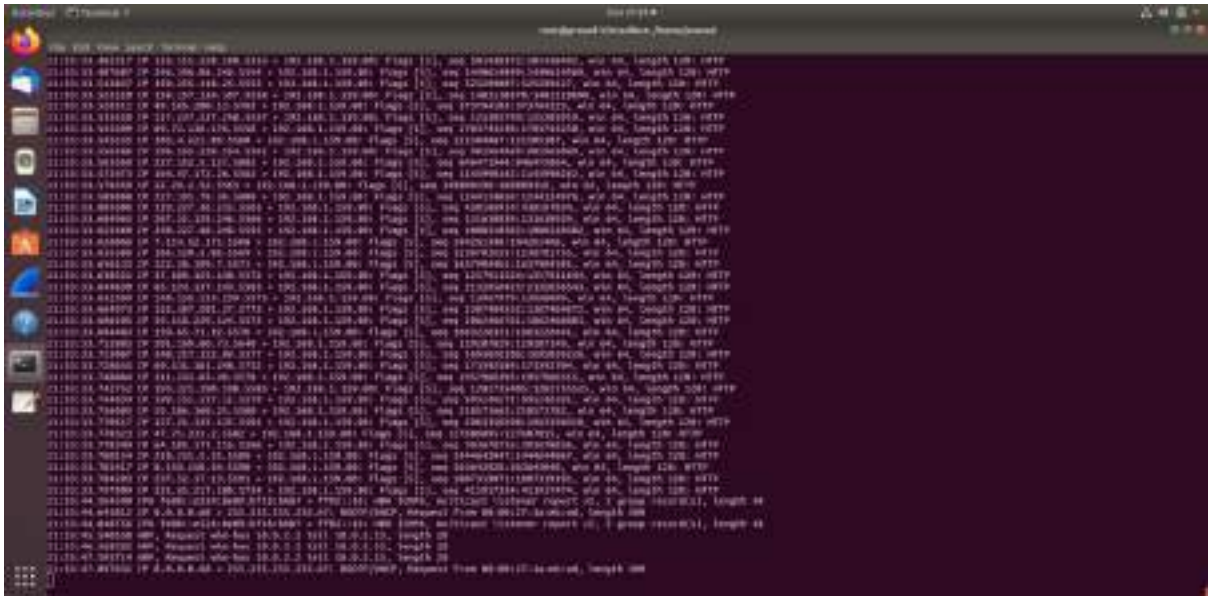
A SMURF attack is a network-based DoS attack that takes advantage of ICMP and IP addressing.

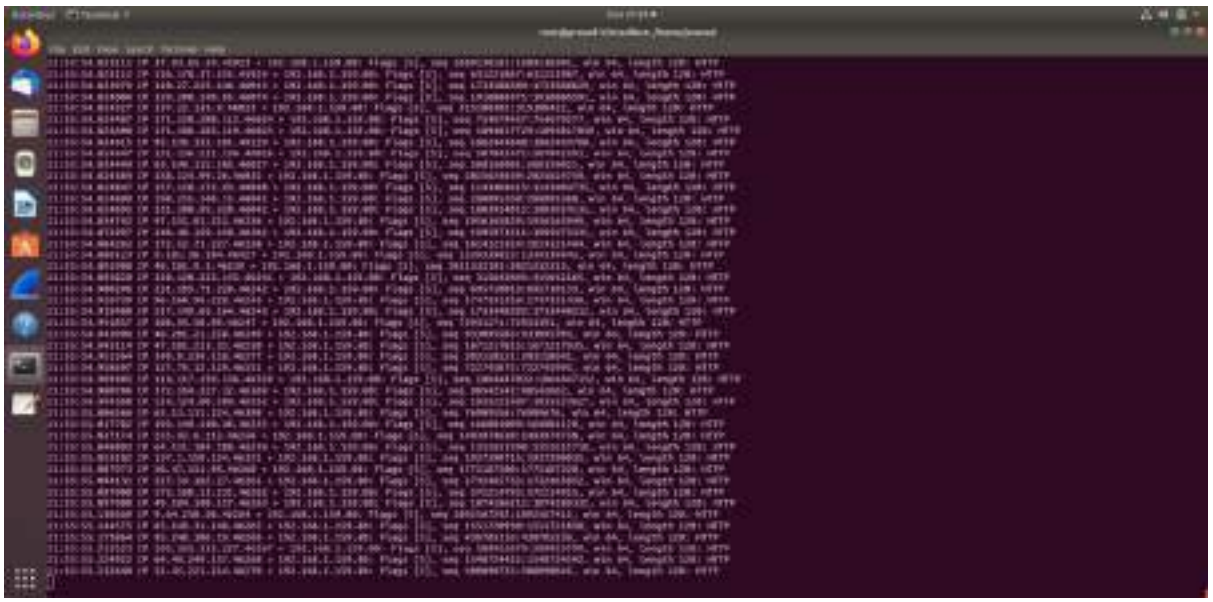
In a SMURF attack, the attacker sends a large number of ICMP echo request (ping) packets to an IP broadcast address, typically spoofing the source IP address to make it appear as if the requests are coming from the victim's IP address.

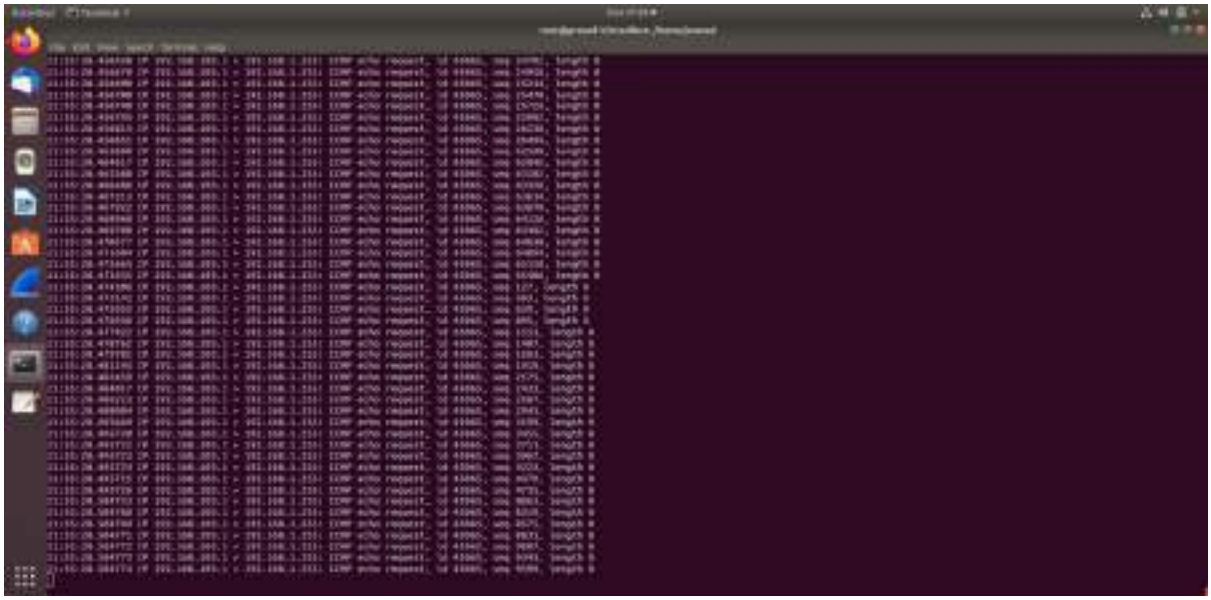
When these requests are sent to the broadcast address, all devices on the target network respond with ICMP echo replies. With a high enough volume of requests, this can flood the victim's network, overwhelming its resources and causing a DoS.

To mitigate DoS attacks, organizations use various security measures, including firewalls, intrusion detection systems (IDS), intrusion prevention systems (IPS), and content delivery networks (CDNs). These tools help identify and filter out malicious traffic, allowing legitimate traffic to reach its destination. Additionally, proper network design and configuration can help minimize the impact of DoS attacks.









CONCLUSION:

Hence, we gained knowledge about the network analysis and security assessment tools. Explore various network probing and testing techniques, which are valuable skills in the field of network administration and cybersecurity. We used various hping3 commands.

Roll No – 06

Name – Prasad Sunil Arote

Date – 13/10/2023

Lab Assignment 10

AIM: To study and configure Firewalls using IP tables.

LO6: Demonstrate network security system using open source tools.

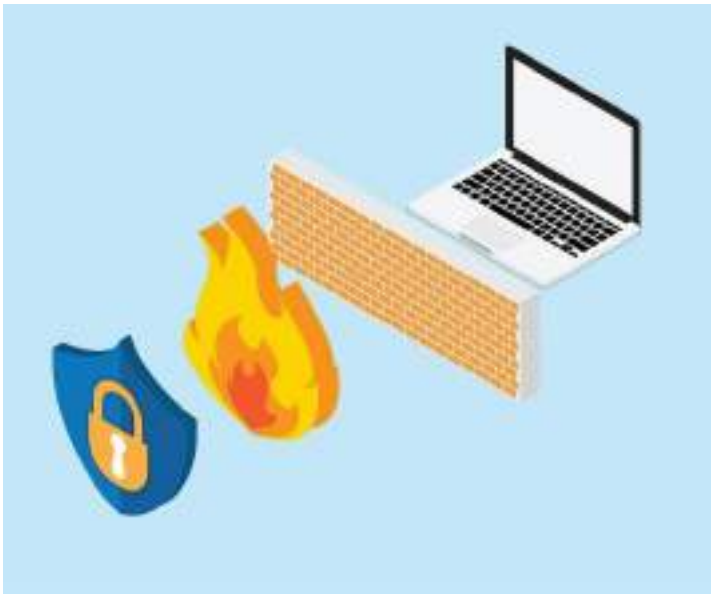
THEORY:

What is a Firewall?

A firewall is a type of cybersecurity tool used to filter traffic on a network. Firewalls can separate network nodes from external traffic sources, internal traffic sources, or even specific applications. Firewalls can be software, hardware, or cloud-based, with each type of firewall having unique pros and cons.

Different types of Firewall

Type 1: Packet-Filtering Firewalls



Packet filtering firewalls operate at the network layer (Layer 3) of the OSI model. They examine network packets and make filtering decisions based on criteria such as source and destination IP addresses, port numbers, and protocols. These firewalls can allow or block traffic based on predefined rules.

Type 2: Stateful Inspection Firewall:

Stateful firewalls operate at the network layer (Layer 3) and transport layer (Layer 4). They keep track of the state of active connections and make decisions based on the state of the connection. This allows stateful firewalls to better understand and control complex traffic flows and prevent unauthorized access.

Proxy Firewall:

Proxy firewalls, also known as application-level gateways (ALGs), operate at the application layer (Layer 7). They act as intermediaries between internal and external systems, forwarding requests and responses on behalf of clients. This can enhance security by not exposing the internal network's IP addresses.

Application Layer Firewall: Application layer firewalls, also known as deep packet inspection firewalls, are highly advanced and operate at the application layer (Layer 7). They can understand and filter traffic based on specific application protocols. This allows them to provide granular control over application-specific traffic, making them effective at detecting and blocking application-layer threats.

Write the different options that can be used in configuring firewall?

When configuring a firewall, you can use various options and parameters to define rules and policies that control the traffic entering and exiting your network. The specific options available can vary depending on the firewall software or device you're using, but here are some common options and parameters:

Source Address (-s): You can specify the source IP address or range of IP addresses from which traffic is allowed or denied.

Destination Address (-d): This option allows you to define the destination IP address or IP address range for which the rule applies.

Protocol (-p): You can specify the network protocol, such as TCP, UDP, or ICMP, to which the rule applies.

Source Port (--sport): Define the source port or port range from which the traffic originates.

Destination Port (--dport): Specify the destination port or port range to which the traffic is headed.

Action (-j): Determine what action to take if a packet matches the rule. Common actions include ACCEPT, DROP, REJECT, and LOG. For example, -j ACCEPT allows the packet, while -j DROP discards it.

Interface In (-i): Define the incoming network interface where the traffic should be filtered.

Interface Out (-o): Specify the outgoing network interface for filtering outbound traffic.

Stateful Inspection (-m state): This option is used in stateful inspection firewalls to track the state of established connections. It is typically used with -p to define the protocol.

Logging (-j LOG): You can log information about packets matching a rule for analysis and auditing. Logging rules are often used with the -j LOG target.

Match Extensions (-m): Some firewalls support extensions or modules that allow you to match packets based on specific criteria. These extensions provide additional filtering capabilities, such as -m tcp, -m udp, or -m multiport.

Default Policy (-P): Set the default action for packets that do not match any of the configured rules. Common policies include ACCEPT and DROP.

Connection Tracking (-m conntrack): In stateful firewalls, this option allows you to match packets based on their connection state (e.g., NEW, ESTABLISHED, RELATED).

Time-Based Rules: Some firewalls support time-based rules that allow you to control traffic based on the time of day or specific schedules.

User and Group-Based Rules: In more advanced firewalls, you can configure rules based on user or group identities, providing fine-grained control over access.

Write the commands used for configuring firewall using IPTABLES?

iptables is a popular command-line tool for configuring a firewall on Linux systems. It allows you to set up rules to control incoming and outgoing network traffic. Below are some common iptables commands for configuring a firewall. Please note that to use these commands, you typically need superuser or root privileges (e.g., using sudo).

Flush Existing Rules: Before setting up your firewall rules, it's a good practice to flush existing rules to start with a clean slate. Use the following command to do this:

sudo iptables -F

Set Default Policies:

To set the default policy for incoming traffic (e.g., deny all incoming traffic by default):

sudo iptables -P INPUT DROP

To set the default policy for outgoing traffic (e.g., allow all outgoing traffic by default):

sudo iptables -P OUTPUT ACCEPT

Allow SSH (Port 22): To allow incoming SSH traffic, which is essential for remote server access:

sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT

Allow HTTP (Port 80) and HTTPS (Port 443): To permit web traffic:

sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT sudo

iptables -A INPUT -p tcp --dport 443 -j ACCEPT

Allow Established and Related Connections: To allow incoming traffic related to established connections, which is crucial for established connections to work:

sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT Save Rules: After configuring your firewall rules, save them to ensure they persist after a reboot. This command depends on your Linux distribution. For example, on Ubuntu, you can use:

sudo netfilter-persistent save

On CentOS/RHEL:

sudo service iptables save

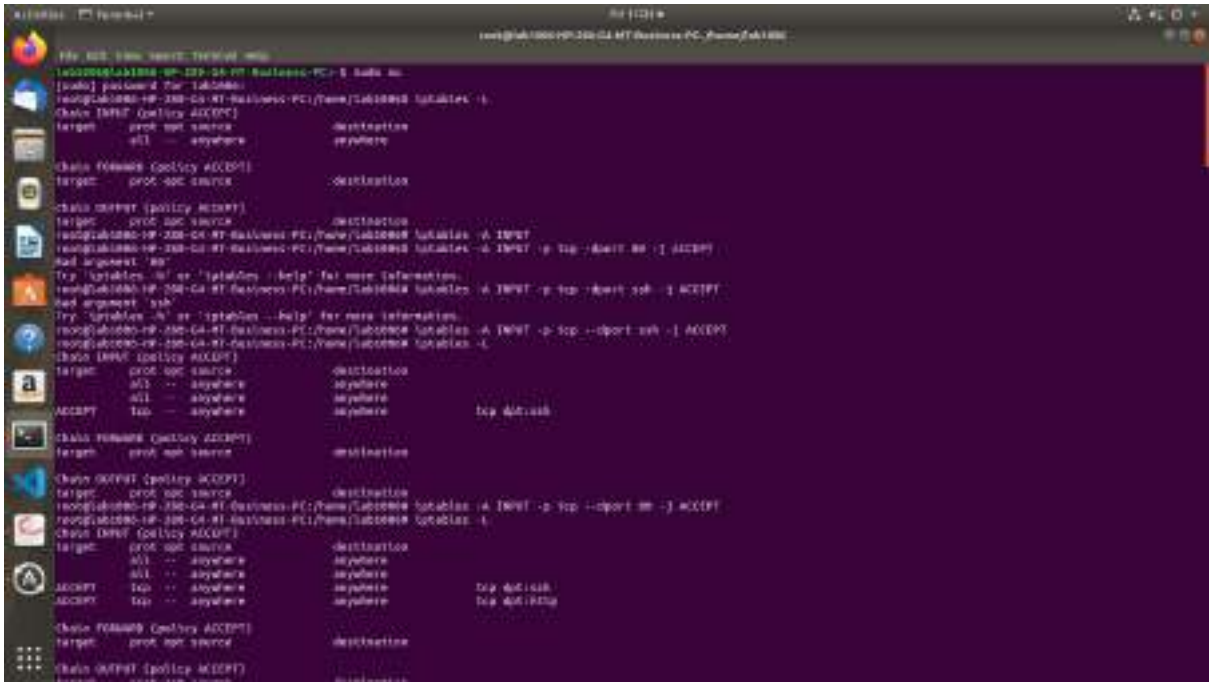
On some distributions, you might need to install iptables-persistent for rule persistence.

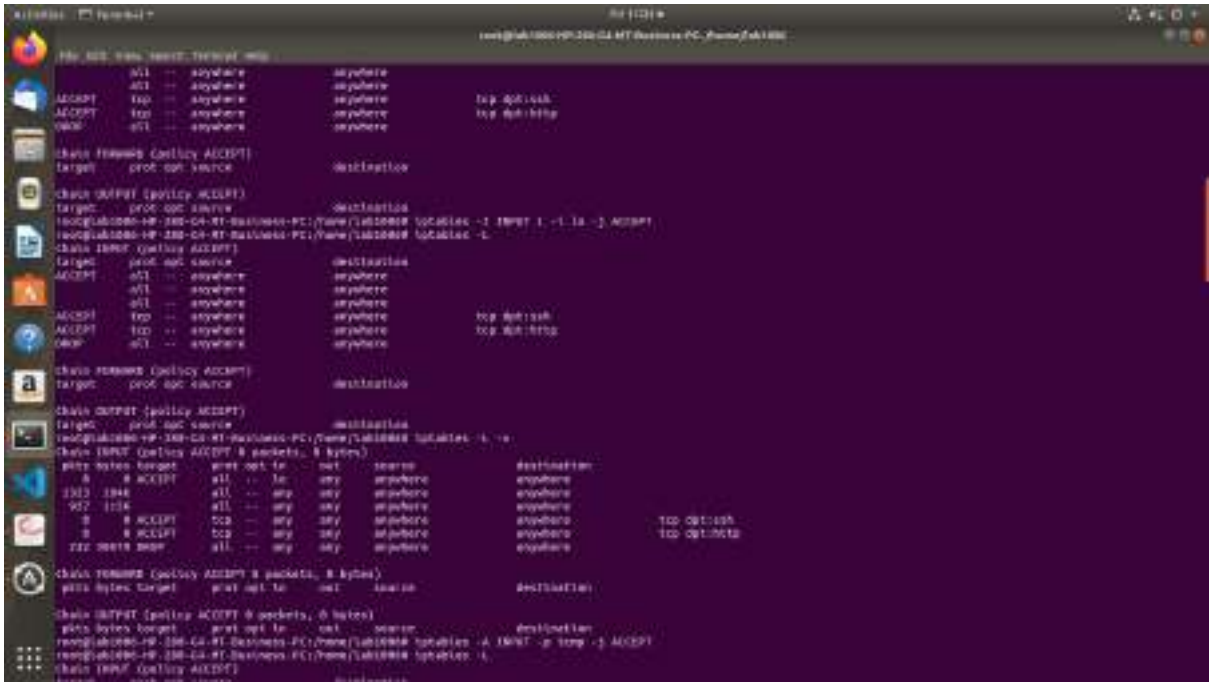
List Rules: To view the configured rules, use the following command:

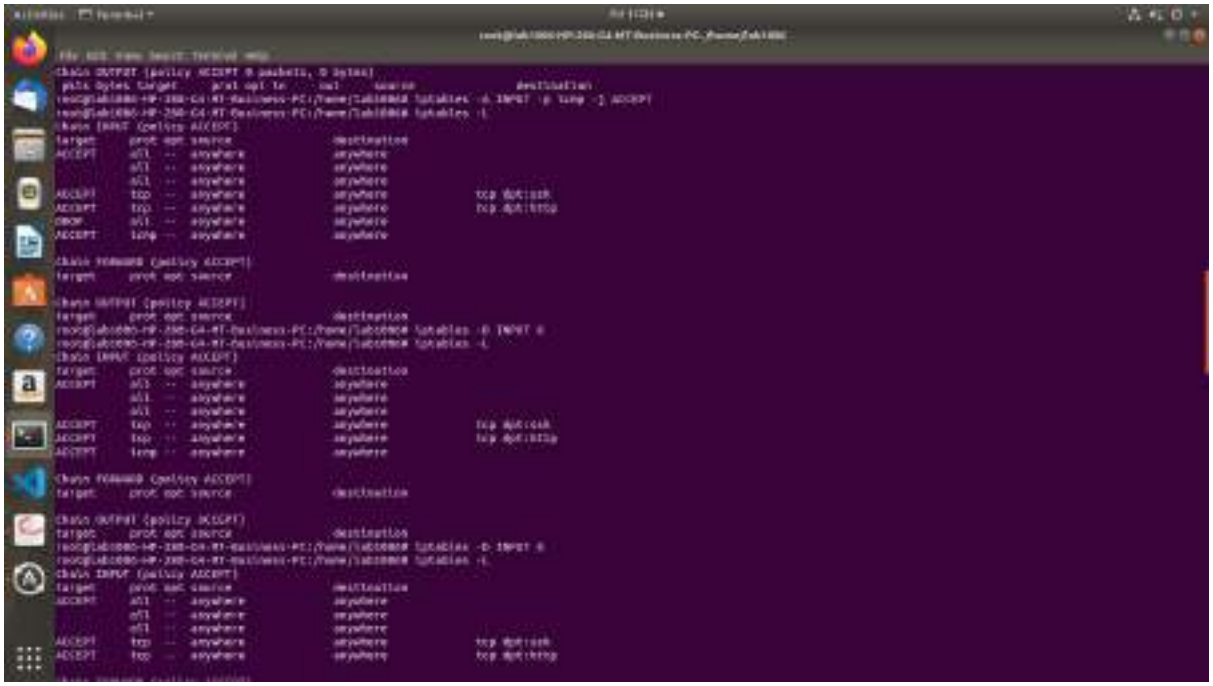
sudo iptables -L

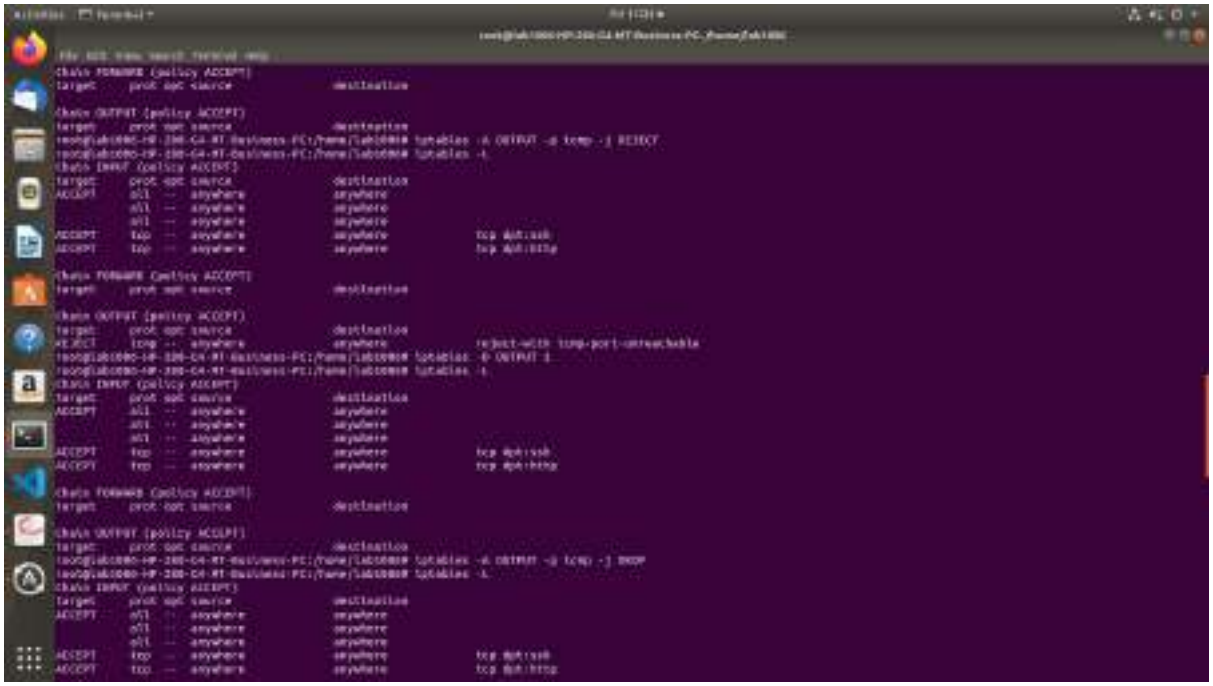
Delete a Rule: To delete a specific rule, identify its number from the list generated by iptables -L and use the -D option. Replace N with the rule number:

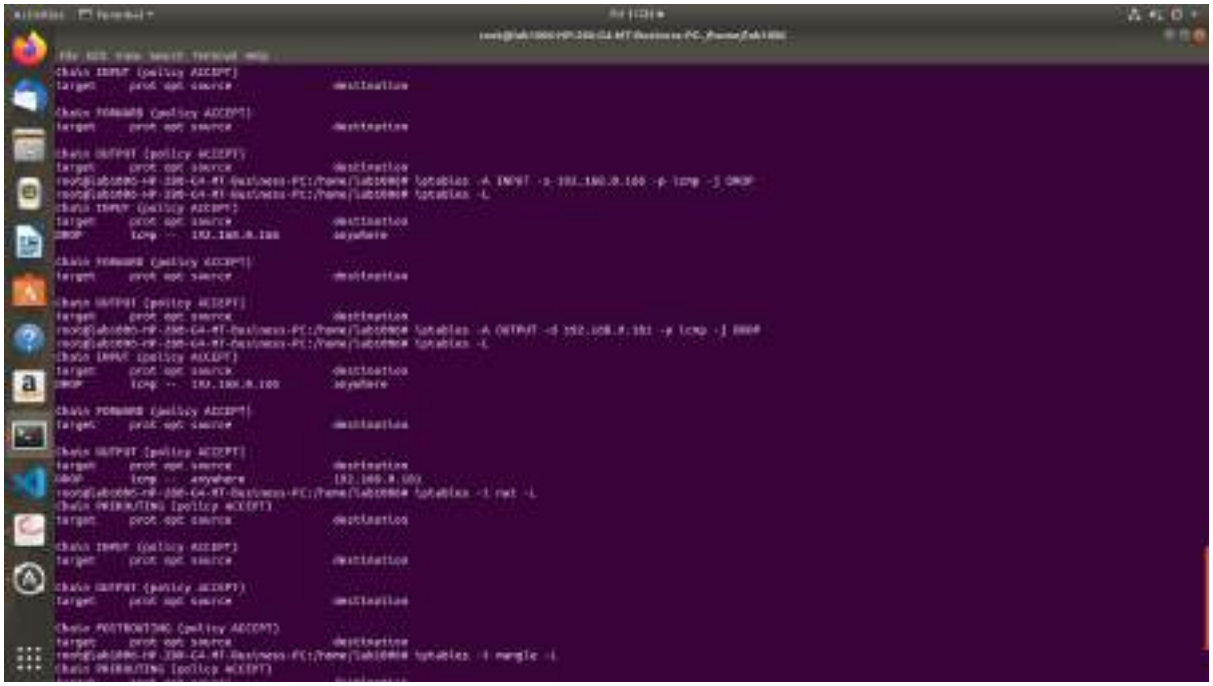
sudo iptables -D INPUT N

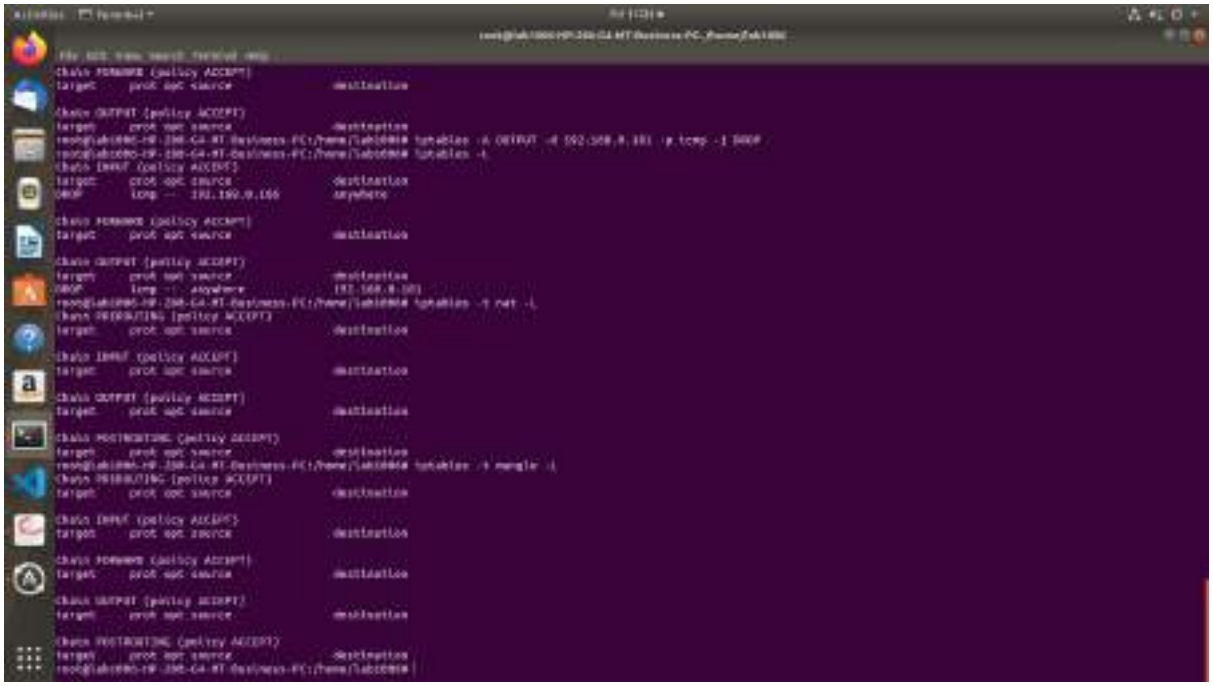












CONCLUSION:

Firewalls are essential components in network security, serving as a crucial defense against cyber threats. They come in various types, each tailored to specific security needs. The choice of firewall and its configuration options depend on the specific requirements of the network or system to be protected. Configuring firewalls should be done with careful consideration of security policies and best practices, as well as an understanding of the firewall software or device in use.

Roll No 06

Name- Prasad Sunil Arote

Date- 25-08-2023

Lab Assignment 11

Aim: Installing snort, setting in Intrusion Detection Mode and writing rules for Intrusion Detection.

LO6: Demonstrate the network security system using open source tools.

Theory:

What is Intrusion Detection System?

1. A system called an intrusion detection system (IDS) observes network traffic for malicious transactions and sends immediate alerts when it is observed. It is software that checks a network or system for malicious activities or policy violations.
2. Each illegal activity or violation is often recorded either centrally using a SIEM system or notified to an administration.
3. IDS monitors a network or system for malicious activity and protects a computer network from unauthorized access from users, including perhaps insiders.
4. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between 'bad connections' (intrusion/attacks) and 'good (normal) connections'.

What are different modes in which Snort works?

Snort operates in three primary modes:

1. Sniffer Mode:

   In this mode, Snort acts as a packet sniffer, analyzing network traffic and displaying the captured packets on the console. It doesn't perform any active intrusion detection or prevention; instead, it's used for network analysis and troubleshooting purposes.

2. Packet Logger Mode:

   In packet logger mode, Snort captures and logs network traffic that matches defined rules to log files. This mode is useful for creating a record of network activity for later analysis.

3. Network Intrusion Detection System (NIDS) Mode:

   This is the main mode of Snort, where it functions as a network intrusion detection system (NIDS). Snort examines network traffic against a set of predefined rules to identify and alert on potential intrusion attempts, malicious activities, or suspicious patterns. When a rule matches, Snort generates alerts that can be sent to various destinations, such as log files, syslog servers, or email.

In addition to these primary modes, Snort also offers inline capabilities through its IPS (Intrusion Prevention System) mode:

4. Network Intrusion Prevention System (IPS) Mode:

   When operating as an IPS, Snort not only detects suspicious activity but can also take active measures to prevent or block potential threats. In this mode, Snort can drop or modify packets that match specific rules, effectively preventing malicious traffic from reaching its intended target.

Write the commands used for installing snort, editing its configuration file and configuring it in Intrusion Detection Mode?

1.Check the name of the interface using command ifconfig.

Installing Snort

2. Install snort in ubuntu machine using command sudo apt-get install snort

3. While installing the snort, name of the interface will be asked on which snort is supposed to listen. Enter the interface name observed in step 1

Editing Configuration File (snort.conf):

4. Run the command sudo gedit /etc/snort/snort.conf. This opens snort configuration file.

5. Make following changes to configuration file.a.ipvar HOME_NET 192.168.44.0/24 (in section 1)

6. Open new terminal. Open ftp.rulefile in it by typing the command sudo gedit /etc/snort/rules/ftp.rules(optional)

7. Open new terminal and type the command sudo snort -T -c /etc/snort/snort.conf -i ens33 to validate that all rules are there.

8. Type the command sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -iens33

9. In ubuntu machine, type the following command to create a file called local.rules: sudo gedit /etc/snort/rules/local.rules

10. Write the following rule in it: alert icmp any any -> $HOME_NET any (msg:"ICMP test detected"; GID:1; sid:10000001; rev:001; classtype:icmp-event;)

11. Add the local.rules file in section7 of configuration file of snort by writing: include $RULE_PATH local.rules

12. Validate the changes made in snort.conf file by writing the command in terminal: sudo snort -T -c /etc/snort/snort.conf -iens33

13. Set the snort in Intrusion Detection Mode by typing the command: sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -iens33

CONCLUSION

The installation and configuration of Snort as an Intrusion Detection System (IDS) play a vital role in enhancing network security. Snort offers multiple modes of operation, including Sniffer Mode for packet analysis, Packet Logger Mode for capturing and logging traffic, and its primary Network Intrusion Detection System (NIDS) Mode for actively monitoring and alerting on potential intrusion attempts.

Configuring Snort involves installing the software, editing the snort.conf configuration file to tailor its behavior to specific network requirements, and then starting the Snort service. While Snort is set to operate in intrusion detection mode by default, its flexibility allows for customization to suit various security needs.

Roll No 06

Name: Prasad Sunil Arote

Date: 10/09/2023

Lab Assignment 12

AIM: Explore the GPG tool of Linux to implement Email Security.

LO6: Demonstrate Network Security system using Open Source tools.

THEORY:

A "private keyring" and a "public keyring" are concepts related to cryptographic key management in applications like GPG (GNU Privacy Guard), which is used for secure communication, digital signatures, and encryption. These terms refer to collections of cryptographic keys.

Private Keyring: A private keyring is a file or database that stores private cryptographic keys. Private keys are used for operations like signing messages or decrypting data. These keys should be kept confidential because anyone with access to a private key can use it to impersonate the owner or access encrypted information.

Public Keyring: A public keyring is a file or database that stores public cryptographic keys. Public keys are shared with others and are used for operations like verifying digital signatures or encrypting data that can only be decrypted by the corresponding private key. Public keys are meant to be distributed openly.

Commands for key generation, export, and import of keys, as well as for signing and encrypting a message in GPG.

Key Generation:

To generate a new GPG key pair (public and private keys), use the following command: gpg --

gen-key

This command will prompt you to enter details such as your name, email address, and passphrase for the private key. It will generate a key pair and add it to your keyring.

Exporting and Importing Keys:

To export your public key to a file (e.g., my_public_key.asc), use:

gpg --export -a "Your Name" > my_public_key.asc To

import a public key from a file, use:

gpg --import < my_public_key.asc Signing a

Message:

To sign a message using your private key, use:

gpg --detach-sign -a my_message.txt

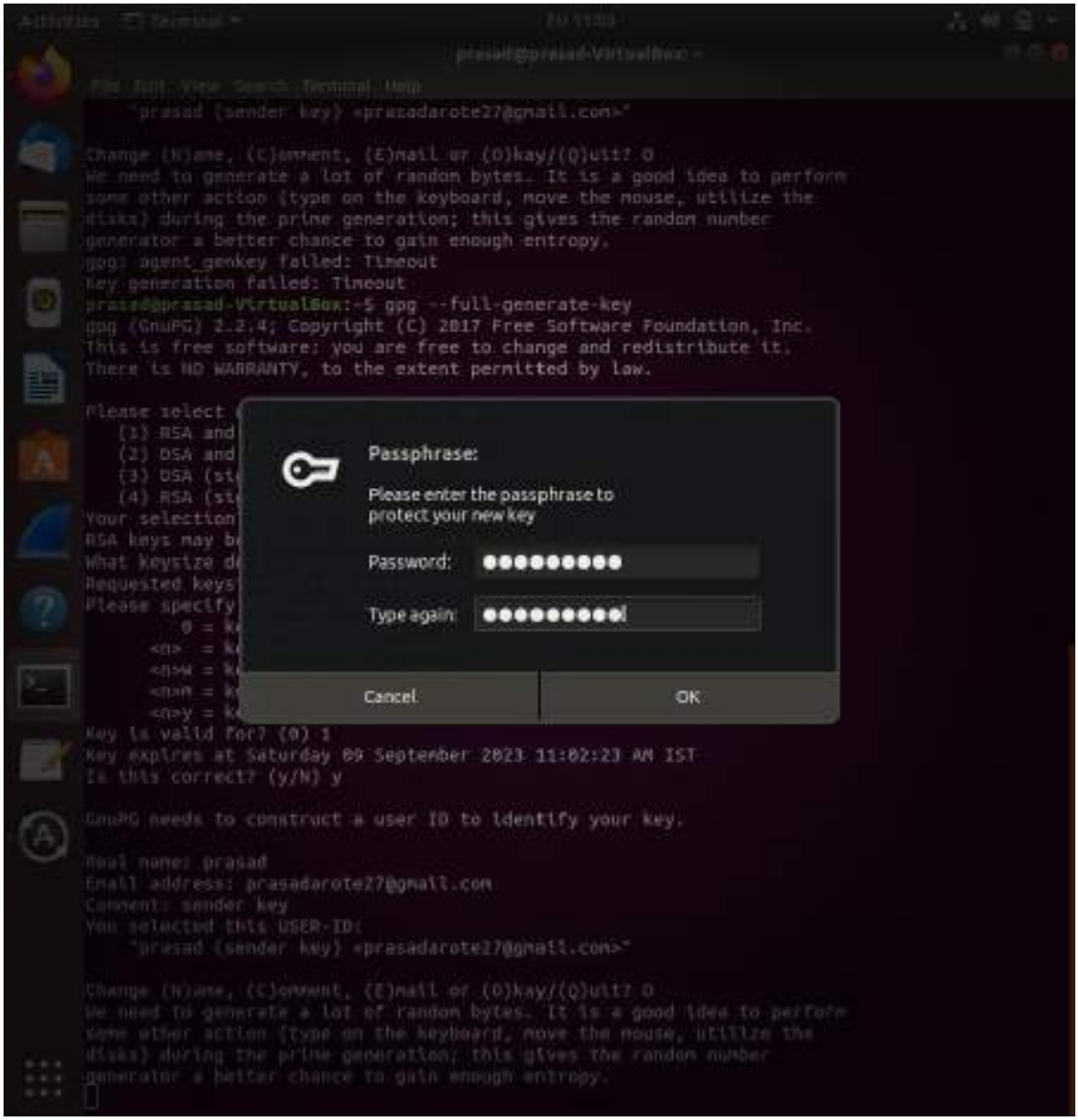
This will create a detached signature file (e.g., my_message.txt.asc) for your message.

Encrypting a Message:

To encrypt a message for someone else using their public key, use:

gpg --encrypt -a -r "Recipient's Name" my_message.txt

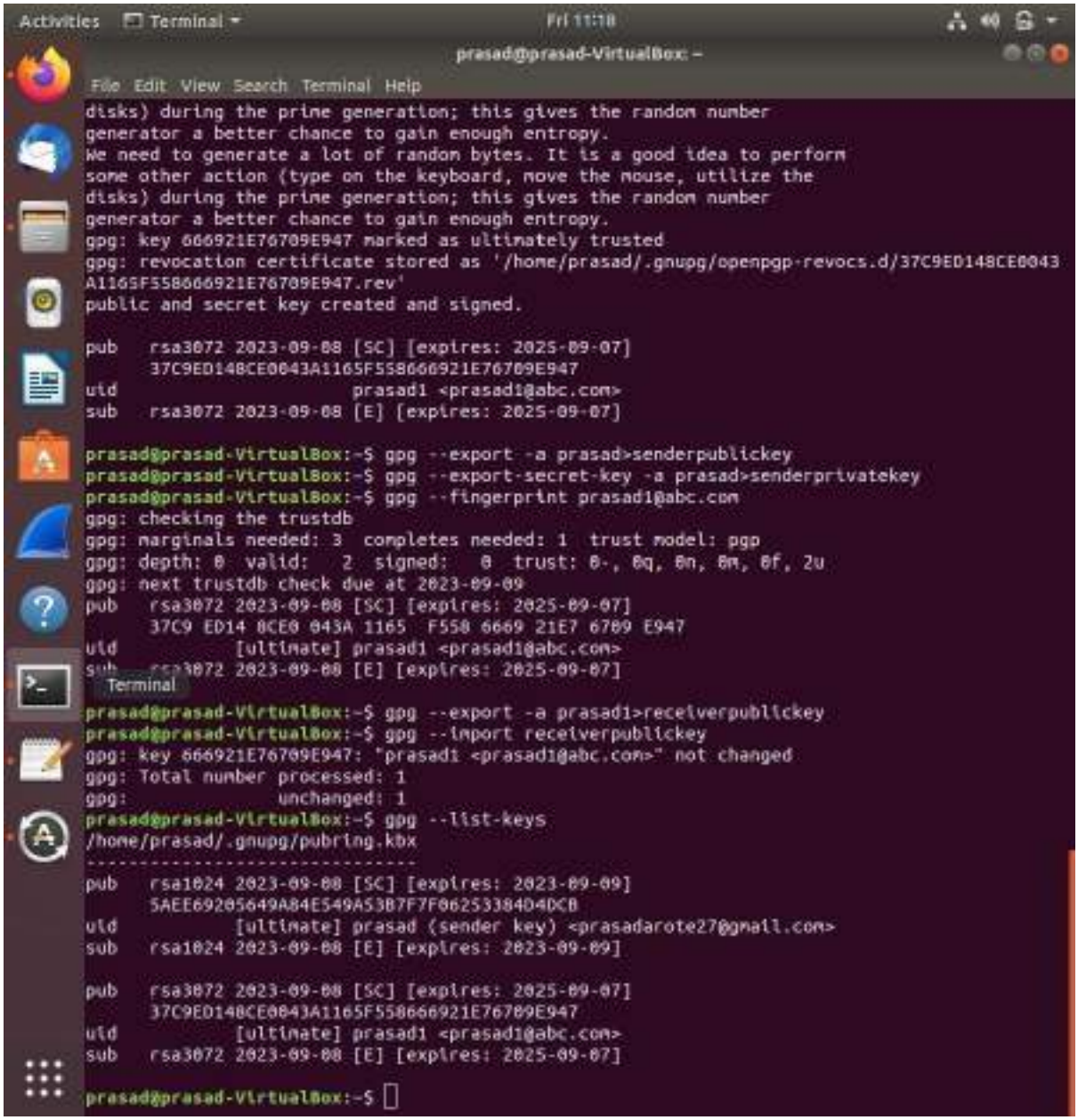
This will create an encrypted file (e.g., my_message.txt.asc) that can only be decrypted by the recipient's private key.

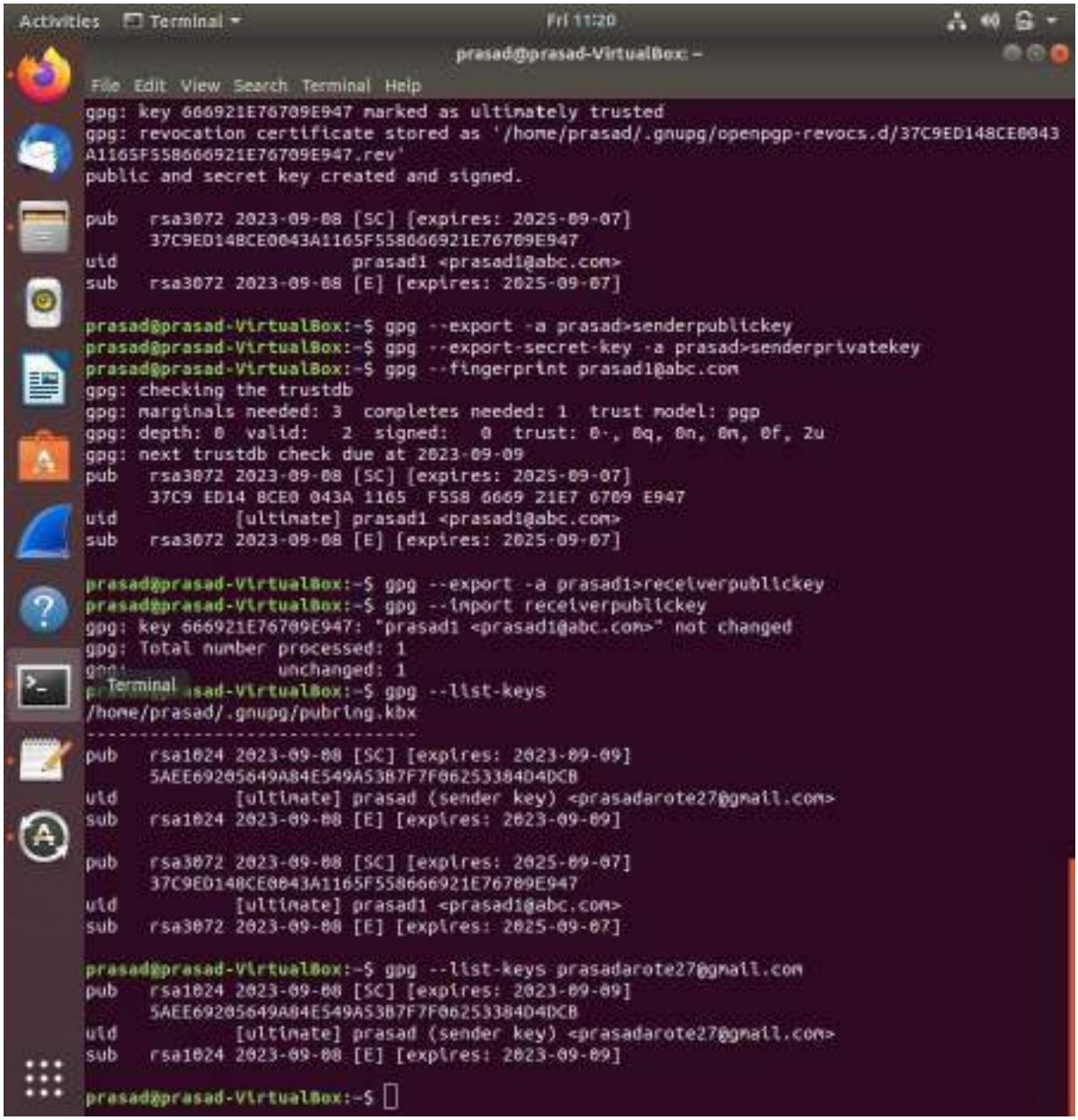

Passphrase:
Please enter the passphrase to protect your new key
Password:
Type again:
Cancel
OK

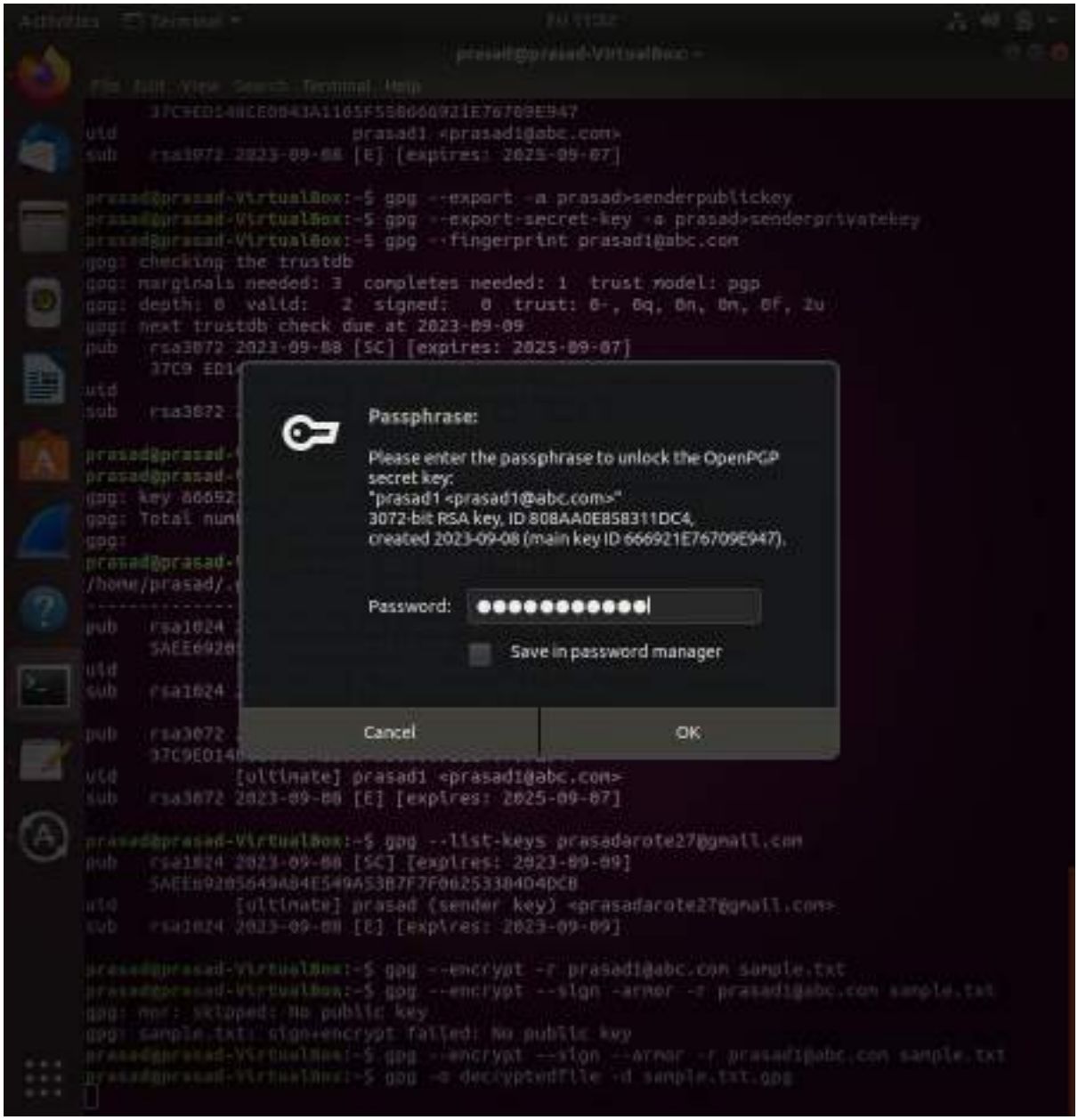

Activities
Terminal
Fri 11:03
prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
(1) RSA and RSA (default)
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 1024
Requested keysize is 1024 bits
Please specify how long the key should be valid.
0 = key does not expire
<n> = key expires in n days
<n>w = key expires in n weeks
<n>m = key expires in n months
<n>y = key expires in n years
Key is valid for? (0) 1
Key expires at Saturday 09 September 2023 11:02:23 AM IST
Is this correct? (y/N) y
GnuPG needs to construct a user ID to identify your key.
Real name: prasad
Email address: prasadarote27@gmail.com
Comment: sender key
You selected this USER-ID:
"prasad (sender key) <prasadarote27@gmail.com>"
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key F7F06253304D4DCB marked as ultimately trusted
gpg: directory '/home/prasad/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/prasad/.gnupg/openpgp-revocs.d/5AEE69205649A84E549A53B7F7F06253304D4DCB.rev'
public and secret key created and signed.
pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
5AEE69205649A84E549A53B7F7F06253304D4DCB
uid prasad (sender key) <prasadarote27@gmail.com>
sub rsa1024 2023-09-08 [E] [expires: 2023-09-09]
prasad@prasad-VirtualBox:~$


Activities
Terminal
Fri 11:04
prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
Real name: prasad
Email address: prasadarote27@gmail.com
Comment: sender key
You selected this USER-ID:
"prasad (sender key) <prasadarote27@gmail.com>"
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: agent_genkey failed: Timeout
Key generation failed: Timeout
prasad@prasad-VirtualBox:~$ gpg --full-generate-key
gpg (GnuPG) 2.2.4; Copyright (C) 2017 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Please select what kind of key you want:
(1) RSA and RSA (default)
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 1024
Requested keysize is 1024 bits
Please specify how long the key should be valid.
0 = key does not expire
<n> = key expires in n days
<n>w = key expires in n weeks
<n>m = key expires in n months
<n>y = key expires in n years
Key is valid for? (0) 1
Key expires at Saturday 09 September 2023 11:02:23 AM IST
Is this correct? (y/N) y
GnuPG needs to construct a user ID to identify your key.
Real name: prasad
Email address: prasadarote27@gmail.com
Comment: sender key
You selected this USER-ID:
"prasad (sender key) <prasadarote27@gmail.com>"
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O


Activities
Terminal
Fri 11:06
prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key F7F06253384D4DCB marked as ultimately trusted
gpg: directory '/home/prasad/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/prasad/.gnupg/openpgp-revocs.d/5AEE69205649AB4
E549A53B7F7F06253384D4DCB.rev'
public and secret key created and signed.
pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
5AEE69205649AB4E549A53B7F7F06253384D4DCB
uid prasad (sender key) <prasadarote27@gmail.com>
sub rsa1024 2023-09-08 [E] [expires: 2023-09-09]
prasad@prasad-VirtualBox:~$ gpg --gen-key
gpg (GnuPG) 2.2.4; Copyright (C) 2017 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Note: Use "gpg --full-generate-key" for a full featured key generation dialog.
GnuPG needs to construct a user ID to identify your key.
Real name: prasad1
Email address: prasad1@abc.com
You selected this USER-ID:
"prasad1 <prasad1@abc.com>"
Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 666921E76709E947 marked as ultimately trusted
gpg: revocation certificate stored as '/home/prasad/.gnupg/openpgp-revocs.d/37C9ED140CE0043
A1165F558666921E76709E947.rev'
public and secret key created and signed.
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9ED140CE0043A1165F558666921E76709E947
uid prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$


Activities
Terminal
Fri 11:18
prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 666921E76709E947 marked as ultimately trusted
gpg: revocation certificate stored as '/home/prasad/.gnupg/openpgp-revocs.d/37C9ED148CE0043
A1165F558666921E76709E947.rev'
public and secret key created and signed.
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9ED148CE0043A1165F558666921E76709E947
uid prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$ gpg --export -a prasad>senderpublickey
prasad@prasad-VirtualBox:~$ gpg --export-secret-key -a prasad>senderprivatekey
prasad@prasad-VirtualBox:~$ gpg --fingerprint prasad1@abc.com
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 2 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2023-09-09
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9 ED14 8CE0 043A 1165 F558 6669 21E7 6709 E947
uid [ultimate] prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
Terminal
prasad@prasad-VirtualBox:~$ gpg --export -a prasad1>receiverpublickey
prasad@prasad-VirtualBox:~$ gpg --import receiverpublickey
gpg: key 666921E76709E947: "prasad1 <prasad1@abc.com>" not changed
gpg: Total number processed: 1
gpg: unchanged: 1
prasad@prasad-VirtualBox:~$ gpg --list-keys
/home/prasad/.gnupg/pubring.kbx
pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
5AEE69205649A84E549A53B7F7F06253384D4DC8
uid [ultimate] prasad (sender key) <prasadarote27@gmail.com>
sub rsa1024 2023-09-08 [E] [expires: 2023-09-09]
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9ED148CE0043A1165F558666921E76709E947
uid [ultimate] prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$


Activities
Terminal
Fri 11:20
prasad@prasad-VirtualBox: ~
File Edit View Search Terminal Help
gpg: key 666921E76709E947 marked as ultimately trusted
gpg: revocation certificate stored as '/home/prasad/.gnupg/openpgp-revocs.d/37C9ED148CE0043A1165F558666921E76709E947.rev'
public and secret key created and signed.
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9ED148CE0043A1165F558666921E76709E947
uid prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$ gpg --export -a prasad>senderpublickey
prasad@prasad-VirtualBox:~$ gpg --export-secret-key -a prasad>senderprivatekey
prasad@prasad-VirtualBox:~$ gpg --fingerprint prasad1@abc.com
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 2 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2023-09-09
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9 ED14 8CE0 043A 1165 F558 6669 21E7 6709 E947
uid [ultimate] prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$ gpg --export -a prasad1>receiverpublickey
prasad@prasad-VirtualBox:~$ gpg --import receiverpublickey
gpg: key 666921E76709E947: "prasad1 <prasad1@abc.com>" not changed
gpg: Total number processed: 1
gpg: unchanged: 1
Terminal
sad-VirtualBox:~$ gpg --list-keys
/home/prasad/.gnupg/pubring.kbx
pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
5AEE69205649A84E549A53B7F7F06253384D4DCB
uid [ultimate] prasad (sender key) <prasadarote27@gmail.com>
sub rsa1024 2023-09-08 [E] [expires: 2023-09-09]
pub rsa3072 2023-09-08 [SC] [expires: 2025-09-07]
37C9ED148CE0043A1165F558666921E76709E947
uid [ultimate] prasad1 <prasad1@abc.com>
sub rsa3072 2023-09-08 [E] [expires: 2025-09-07]
prasad@prasad-VirtualBox:~$ gpg --list-keys prasadarote27@gmail.com
pub rsa1024 2023-09-08 [SC] [expires: 2023-09-09]
5AEE69205649A84E549A53B7F7F06253384D4DCB
uid [ultimate] prasad (sender key) <prasadarote27@gmail.com>
sub rsa1024 2023-09-08 [E] [expires: 2023-09-09]
prasad@prasad-VirtualBox:~$


Passphrase:
Please enter the passphrase to unlock the OpenPGP secret key:
"prasad1 <prasad1@abc.com>"
3072-bit RSA key, ID 808AA0E858311DC4,
created 2023-09-08 (main key ID 666921E76709E947).
Password:
Save in password manager
Cancel
OK

CONCLUSION:

We've explored the concepts of private keyrings and public keyrings in GPG. We've also provided commands for key generation, exporting and importing keys, signing messages, and encrypting messages using GPG. These commands are fundamental to using GPG for secure communication and data protection.

ROLL NO: 06

NAME: PRASAD SUNIL AROTE

DATE: 20/09/2023

THEORY ASSIGNMENT 1

1. Explain the padding scheme used in RSA. Why It is used? What is its limitation?

The padding scheme used in RSA (Rivest–Shamir–Adleman) encryption is essential to address some of the security and practical limitations of the RSA algorithm. RSA itself is a public-key cryptosystem widely used for secure communication and digital signatures. However, it has certain vulnerabilities when used without proper padding.

Padding Scheme in RSA:

The most commonly used padding schemes in RSA are PKCS#1 v1.5 padding and OAEP (Optimal Asymmetric Encryption Padding). These padding schemes serve the following purposes:

Security: Padding adds randomization and structure to plaintext data before encryption, making it resistant to certain cryptographic attacks.

Determinism: Without padding, RSA encryption would produce the same ciphertext for the same plaintext every time, which can lead to security issues.

PKCS#1 v1.5 Padding:

PKCS#1 v1.5 padding involves the following steps:

A block of plaintext is padded with random data and formatted to create a consistent-length block.

The padded data is then encrypted using the RSA public key.

The recipient decrypts the ciphertext using the RSA private key and removes the padding to retrieve the original plaintext.

Padding Process:

Determine Block Size:

Calculate the block size in bytes based on the size of the RSA key. For example, for an RSA key of 2048 bits (256 bytes), the block size is typically 256 - 11 = 245 bytes.

Add Padding Bytes:

Generate a random byte string of length block_size - 3 - len(plaintext), where block_size is the calculated block size, and len(plaintext) is the length of the plaintext message.

This random byte string is used to pad the plaintext, ensuring that the overall length of the padded block matches the block size.

Construct the Padded Block:

Create the padded block by concatenating the following elements:

A byte with value 0x00 (0).

A byte with value 0x02 (2). This is called the "block type".

The random padding bytes generated in step 2.

A byte with value 0x00 (0).

The original plaintext message.

Encrypt the Padded Block:

Encrypt the padded block using the recipient's RSA public key. This results in the ciphertext.

Recipient's Decryption Process:

Decrypt with RSA Private Key:

The recipient uses their RSA private key to decrypt the ciphertext, resulting in the padded block.

Verify Padding:

The recipient examines the padded block and checks for specific bytes to ensure it conforms to the PKCS#1 v1.5 padding format.

The block type must be 0x02, and there must be at least eight bytes of padding consisting of non-zero random bytes.

Extract the Plaintext:

Once the padding is verified, the recipient extracts the original plaintext by removing the padding bytes and the additional zero byte.

Limitations of PKCS#1 v1.5 Padding:

Security Vulnerabilities:

PKCS#1 v1.5 padding has been found to be vulnerable to certain attacks, such as the Bleichenbacher attack. This vulnerability has led to security issues in implementations that do not properly validate padding.

Deterministic:

While this padding scheme introduces some randomness due to the padding bytes, it is still somewhat deterministic, which can be a limitation in some situations.

Padding Overhead:

PKCS#1 v1.5 padding adds overhead to the plaintext, reducing the efficiency of encryption for small messages.

OAEP (Optimal Asymmetric Encryption Padding):

OAEP is a more modern and secure padding scheme for RSA encryption. It overcomes some of the limitations of PKCS#1 v1.5 padding. OAEP padding involves the following steps:

A block of plaintext is first padded with a random "seed" and a "mask" to introduce a high degree of randomness.

This padded data is then XORed with the hash value of the recipient's public key to create a masked message.

The masked message is encrypted with the RSA public key.

The recipient decrypts the ciphertext, retrieves the masked message, and applies the reverse operations to obtain the original plaintext.

Advantages of OAEP Padding:

Security Enhancement:

OAEP provides better security by introducing stronger randomization and additional mathematical operations, making it resistant to known cryptographic attacks.

Deterministic Security:

OAEP overcomes the determinism limitation of PKCS#1 v1.5 padding, making it suitable for applications where determinism could pose a security risk.

Padding Overhead:

While OAEP introduces additional data, it is more efficient in terms of padding overhead compared to PKCS#1 v1.5 padding.

In summary, padding schemes like PKCS#1 v1.5 and OAEP are used in RSA encryption to enhance security and address vulnerabilities. While PKCS#1 v1.5 is less secure due to certain vulnerabilities, OAEP provides a more robust solution with improved security and resistance to attacks, making it the preferred choice for secure RSA encryption in modern cryptographic applications.

Roll No: 06

NAME: PRASAD SUNIL AROTE

DATE: 23/09/2023

THEORY ASSIGNMENT 2

What is Intrusion Detection System? Explain different types of intrusion detection systems with their working. State the advantages and limitations of each.

An Intrusion Detection System (IDS) is a critical component of network and system security that helps identify and respond to unauthorized or suspicious activities, breaches, and attacks on a computer network or system. It monitors network traffic or system behavior and raises alarms or alerts when it detects potential security threats. IDS plays a crucial role in maintaining the confidentiality, integrity, and availability of data and systems in various environments, including corporate networks, data centers, and cloud infrastructures.

There are two primary types of Intrusion Detection Systems:

Host-based Intrusion Detection System (HIDS):

Working: HIDS monitors and analyzes activities and events on individual host systems (e.g., servers, workstations). It focuses on the internals of a system, including log files, system calls, and file integrity. HIDS agents are typically installed on each host, and they generate alerts or reports when they detect suspicious behavior, such as unauthorized access, file modifications, or unusual system calls.

Advantages:

Granular visibility into individual host activities.

Effective at detecting local threats and insider attacks.

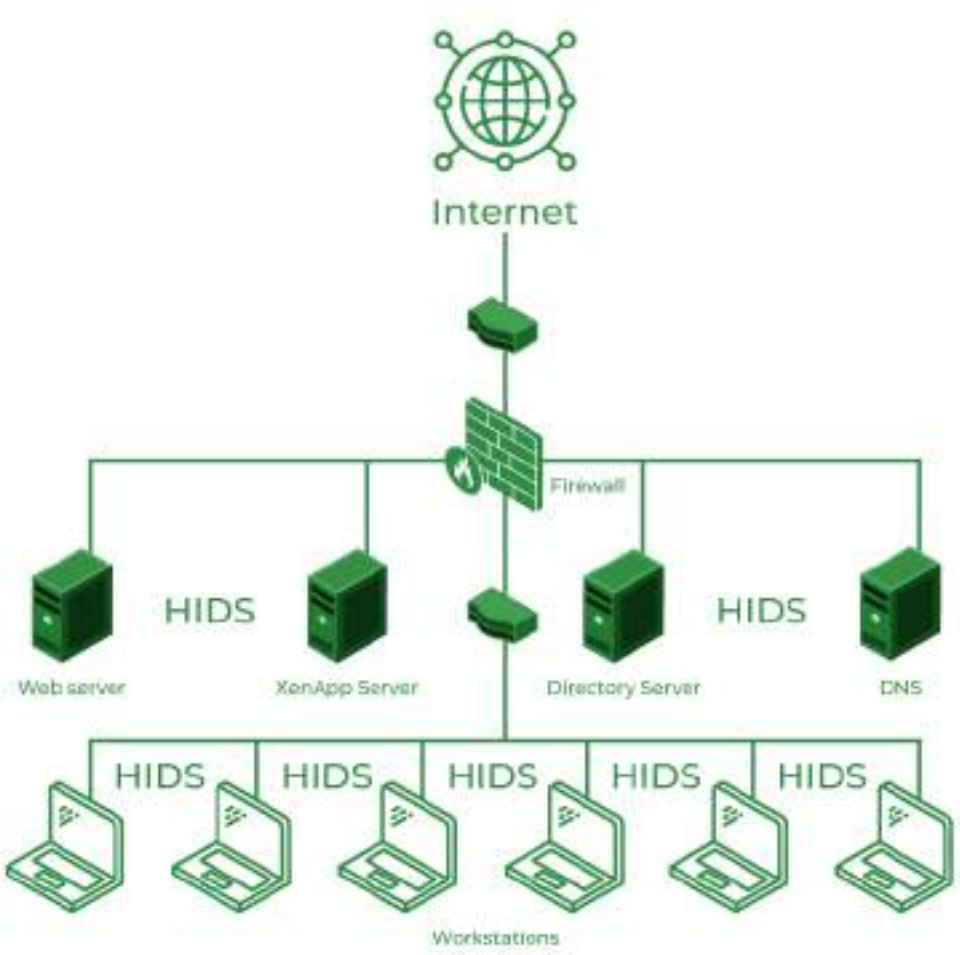
Able to monitor changes in critical system files.

Limitations:

Requires installation and maintenance on each host, which can be resource-intensive.

Limited to monitoring the host it is installed on and may miss network-level attacks.

May generate a high volume of alerts that require manual analysis.



Network-based Intrusion Detection System (NIDS):

Working: NIDS monitors network traffic at various points within a network, such as at the network perimeter or within network segments. It analyzes packets, looking for patterns or signatures that match known attack patterns or anomalies that deviate from established network baselines. NIDS can operate in a passive or active mode, with the passive mode being more common to minimize disruption.

Advantages:

Monitors all traffic on a network segment, making it effective at detecting external threats.

Can identify attacks at the network level, such as port scans, denial-of-service (DoS) attacks, and malware propagation.

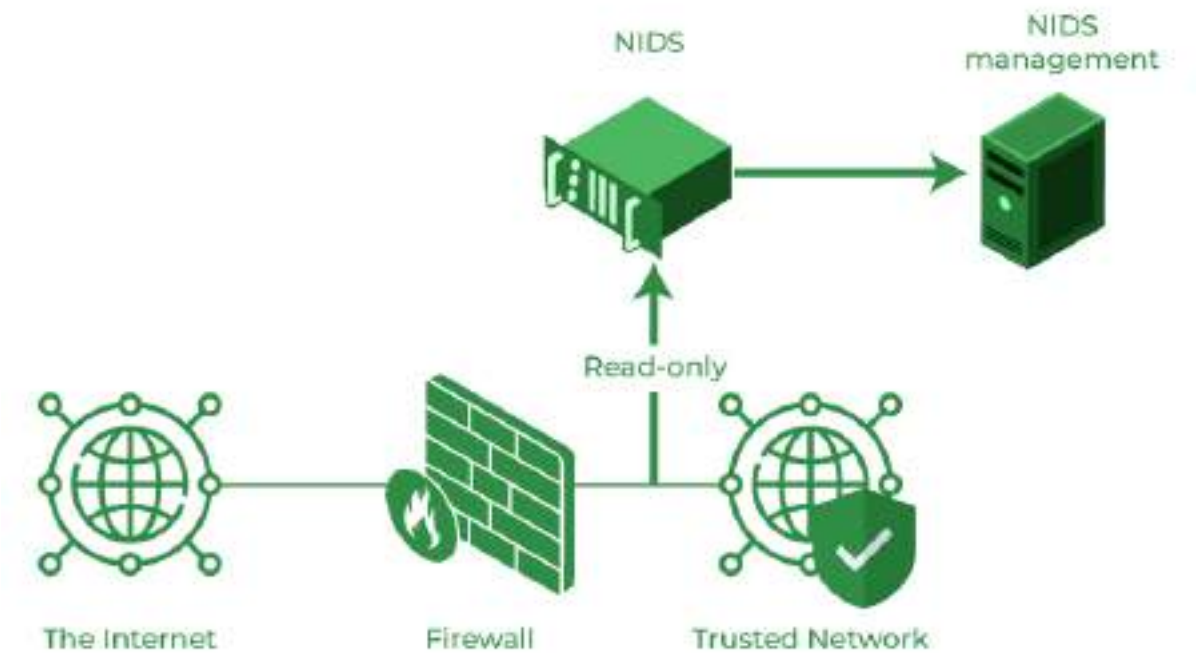
Centralized monitoring simplifies management and alert correlation.

Limitations:

Limited visibility into the internals of individual hosts.

May generate false positives or miss zero-day attacks without updated signatures.

Vulnerable to encrypted traffic that cannot be inspected without decryption.



Protocol-Based Intrusion Detection System (PIDS):

PIDS, a Protocol-based Intrusion Detection System, is a system or agent that resides consistently at the front end of the server to control and interpret the protocol between the user and the server. PIDS is for securing the web server by monitoring the HTTPS protocol stream. A typical use of PIDS is at the front end of the web server, keeping a check on the HTTP or HTTPS stream.

Application Protocol-Based Intrusion Detection System (APIDS):

An application-based intrusion detection system is a system that stays within a group of servers. It identifies the intrusions by monitoring and interpreting the communication on application-specific protocols.

APIDS uses machine language to establish the baseline of the expected system behavior in terms of bandwidth, parts, protocol, and device usage.

Hybrid Intrusion Detection System:

A hybrid intrusion detection system results from two or more approaches to the intrusion detection system. in this, the host agent or the system data is combined with the network

information to develop a complete view of network systems. This system is quite effective in comparison to other IDS.

Signature-Based Intrusion Detection System (SIDS)

A SIDS monitors packets moving through a network and compares them to a database of known attack signatures or attributes. This common type of IDS security looks for specific patterns, such as byte or instruction sequences.

Pros of a SIDS

Works well against attackers using known attack signatures.

Helpful for discovering low-skill attack attempts.

Effective at monitoring inbound network traffic.

Can efficiently process a high volume of network traffic.

Cons of a SIDS

Cannot identify a breach without a specific signature in the threat database.

A savvy hacker can modify an attack to avoid matching known signatures, such as changing lowercase to uppercase letters or converting a symbol to its character code.

Requires regular updates of the threat database to keep the system up to date with the latest risks.

Anomaly-Based Intrusion Detection System (AIDS)

An AIDS monitors ongoing network traffic and analyzes patterns against a baseline. It goes beyond the attack signature model and detects malicious behavior patterns instead of specific data patterns.

This type of IDS uses machine learning to establish a baseline of expected system behavior (trust model) in terms of bandwidth, protocols, ports, and device usage. The system can then compare any new behavior to verified trust models and discover unknown attacks a signature-based IDS cannot identify.

For example, someone in the Sales department trying to access the website's backend for the first time may not be a red flag for a SIDS. For an anomaly-based setup, however, a person trying to access a sensitive system for the first time is a cause for investigation.

Pros of an AIDS

Can detect signs of unknown attack types and novel threats.

Relies on machine learning and AI to establish a model of trustworthy behavior.

Cons of an AIDS

Complex to manage.

Requires more processing resources than a signature-based IDS.

High amounts of alarms can overwhelm admins.

Research on Computer Network Information Security System Based on Big Data

Gengyi Xiao
Department of Mathematics and Computer Technology, Guilin Normal College, Guilin, China
*Corresponding author e-mail: xiao6169@126.com

***Abstract*—In order to effectively improve the computer network security defence capability in the era of big data, a comprehensive analysis of the functions of big data centre applications is performed to create a comprehensive computing network security defence system. First of all, a comprehensive analysis of the hidden dangers of modern computer network security is carried out, and then corresponding technologies such as modern network security technology and solutions, intrusion detection technology are introduced to realize the design of computer network security defence system in the context of the big data era. After the system design is implemented, the system is tested accordingly. According to the test results, the computer network security defence system designed in this paper can actively discover and effectively prevent security threats in the network, thereby ensuring that the network can Normal and safe operation. The computing network security defence system can also provide effective ideas for future network security protection and achieve further expansion of security defence.**

Keywords—Big data era, network security, defence system, intrusion detection.

I. INTRODUCTION

As an iconic technology for human beings entering the 21st century, computer network technology has been continuously improved in China in the past decades. This has also provided a favourable background for the development of information technology. Different types of information methods are continuously integrated into people's daily life and production, bringing great convenience to people's lives, and higher industrial production efficiency. It can be said that the arrival of computer network technology has achieved social change. However, in the context of big data, computer network security has also received widespread attention, and information security issues are very serious. Personal information is transferred to big data, and anyone can query personal privacy information, which affects users. This requires the strengthening of computer network information security protection in the context of big data to protect user information interests.

II. OVERVIEW OF BIG DATA

A. Basic overview of big data

Since 2012, the term big data has been in people's field of vision and attracted constant attention. In the current period, Internet information technology is constantly expanding, and various information resources are flooding every corner of our lives, posing severe challenges for the development of enterprises in the future. Therefore, network information occupies a vitally important position in the operation and management of an enterprise [1]. In the context of the era of big data, information processing models have ushered in new changes, and are constantly being updated. Network information resources have been characterized by diversification and complexity. Therefore, major companies have contended for the market of network information resources, making it a unique advantage for their own development, and thus enhancing their competitiveness in development. It not only breaks the previous limitation of time and space, but also provides a broader information exchange platform for the operation and development of enterprises, and has become a treasure trove of resources that can be continuously tapped by enterprises in the development process. Therefore, constructing a perfect network information security system and continuously developing and using network information technology are important directions for the development of various enterprises in the current period.

B. Analysis of hidden dangers in big data security

The processing of big data includes processes such as generation, transmission, storage, analysis, push, and application. It involves data producers, software developers, distribution links, processors, and users. The information uploaded to the network is divided into structured data and There are two types of unstructured data. Structured data is stored in a relational database structure system. It has an obvious logical structure and is represented by a traditional two-dimensional table. In the era of big data, open data platforms are exposed to the eyes of professionals and various non-professionals. Anyone can send information to the server without strict review. Unstructured accounts for a larger proportion, including office documents in all formats. Comments, text, pictures, subsets of the standard universal mark-up language XML, HTML, various reports, images, audio, video, location information, and other media, the length of the field is variable, field duplication is legal, and You can set subfields and multivalued fields. The structuredness of big data makes the representation of data more difficult, and uniqueness and precision cannot be achieved. Traditional relational database management systems can only manage the structured part, but they are powerless and unstructured in the face of unstructured data. Although data management software has emerged, it has not yet reached a perfect level. There are loopholes in data protection, which provides hacking, information leakage, and Trojan horse penetration. It is difficult to trace the source according to system logs [2].

978-1-7281-7005-3/20/$31.00 ©2020 IEEE

Dalian, China
June 27-29, 2020

C. Big data security risks

1) Privacy data leakage.

With the increase in the level of informatization, people's dependence on the Internet has become more serious. During online shopping, medical treatment, deposit and withdrawal, and social networking, the filled-in form contains a lot of private information, such as bank card account number, password, medical history, home address, ID card, and mobile phone number. Some are encrypted and some are stored in plain text. These data for merchants to analyse customer behaviour, targeting target groups and market predictions provide great convenience, but in the era of big data, the number of customers has become the main indicator of the potential value of the merchant. Driven by huge benefits, the phenomenon of buying and selling customer information is sometimes occurred. At the same time, due to technical reasons and non-standard prevention management, there is a possibility of hacking into the system, leading to the outflow of sensitive information.

2) Technical information leaked.

Enterprise technical documents, R & D data, and software source programs are stored in CRM, ERP, and OA systems. They are the core secrets of the enterprise. If they are not strictly controlled and intercepted by competitors or hackers through VPN, it will inevitably bring economic losses and because Vicious competition affects the production and operation of enterprises.

3) Government industry data outflow.

Household registration files, social security, provident funds, savings, personnel and other information are the guarantee of national security. If obtained by illegal invaders, it will not only pose a threat to the residents, but also cause social instability. In peacetime, government data and information agencies are often targeted by terrorist groups and extremists.

4) Internal data leakage.

Any website, computer information system and database have an administrator role, which can not only perform system management, but also directly enter the background to modify data. Driven by interests or personal purposes, they use administrator permissions to copy or illegally tamper with internal data, and very concealed and difficult to find [3].

III. The Importance of Computer Network Information Security

Due to the characteristics of openness, interconnection, diversity of connection methods, and uneven distribution of terminals, computer networks are vulnerable to computer viruses, hackers, or malicious software. In the face of various threats to network security, it is necessary to consider the crucial issue of network security. Taking enterprise information as an example, as enterprises pay more and more attention to the application of information data, they will inevitably establish websites or information platforms to collect and integrate information related to enterprise production. It directly induces security risks and brings certain negative effects to the enterprise. Therefore, attaching importance to computer network information security, actively finding potential threats in computer network information security, and formulating countermeasures are the key to the development and application of computer network technology in the context of current big data.

IV. Network Security Défense System Design

A. System requirements

The article uses a school as an example to implement the design of a computer network security defence system. The school campus network mainly includes 4 security levels, of which the first level of security requirements mainly includes the requirement for Internet access security; the system has the ability to restore; the identity authentication system is implemented. Design; the second-level security requirements include the interconnection of different sub-networks; the third-level security requirements are mainly to achieve secure access to services and intrusion detection. Figure 1 shows the network plan of the campus network [4].

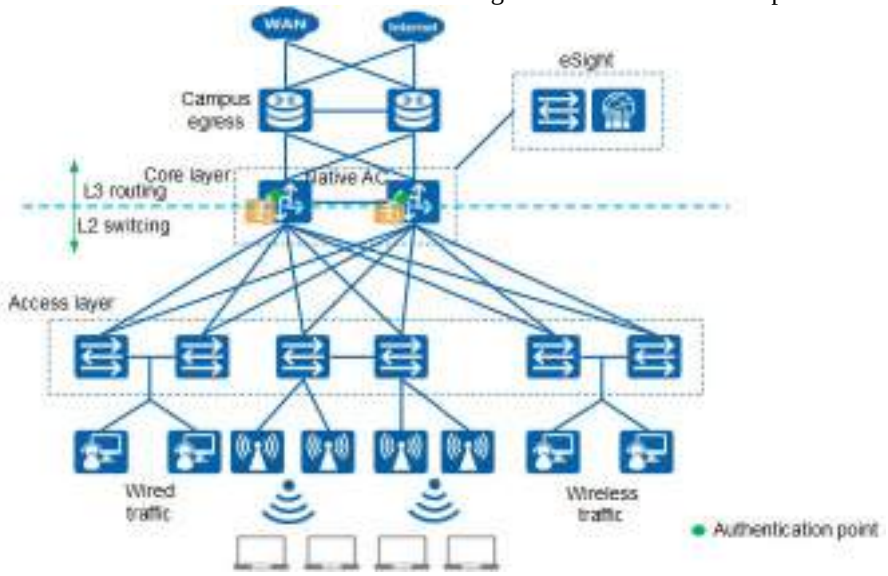


Figure 1. Overall campus network planning

The boundary firewall divides the network area and the campus network area. The network area mainly includes subnets for external services. The campus network mainly includes the three subnets of the department office, administrative office, and student computer room. Through the analysis of modern computer information network

system investigation and campus information system security requirements analysis, it can be said that the computer network security defence system requirements are mainly: desktop system security requirements, virus protection requirements, identity requirements, access control requirements, encryption requirements, security audits Requirements, intrusion security detection system requirements, vulnerability scanning requirements, security management requirements, and physical security measures.

B. Design of Network Security System

1) Security Défense Function.

The attack threats in the use of big data can be spread using multiple channels such as computers and mobile terminals. The latency period of Trojans and viruses is relatively long, which expands the scope of hacker's damage. In order to effectively improve the defence capabilities of big data application centres, you can create an active defence system, thereby further improving the ability of network security operations. Figure 2 shows the security defence system of a big data application centre.

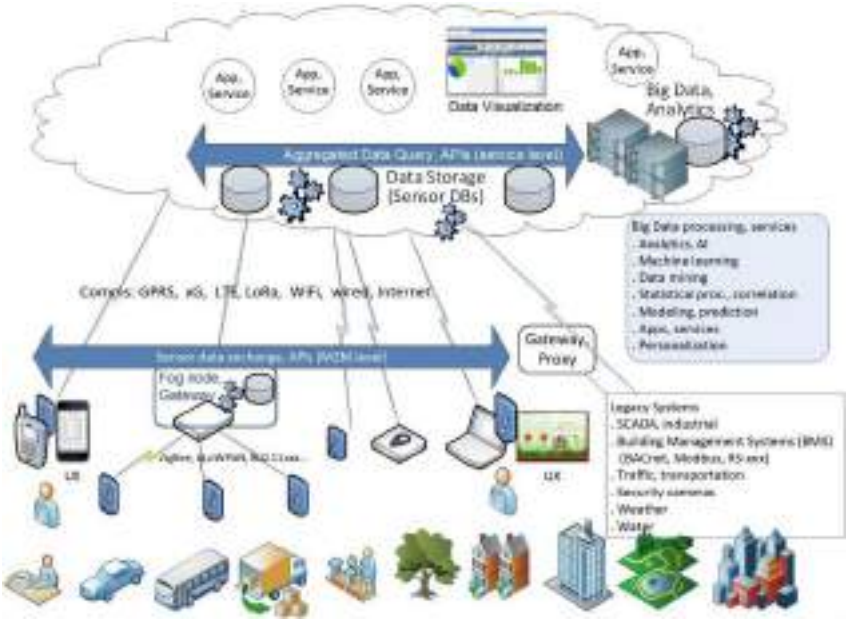


Figure 2. Security Défense System of Big Data Application Centre

2) Security protection.

At present, most computer network security defence systems use firewalls, antivirus software, and other content to achieve security protection. These software's are deployed individually or integrated, thereby effectively improving the integrity of big data application centres. In the process of the continuous promotion and popularization of modern big data application centres, the security defence measures also use digital signature defence technology to avoid repudiation in data communication. Therefore, the system designed in this article combines multiple defence technologies to prevent network data from being infected and attacked. Figure 3 shows the design of the security protection module [5].

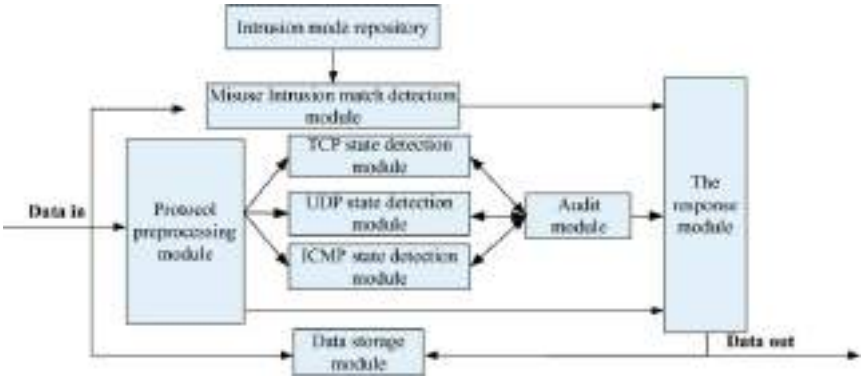


Figure 3. Security module design

C. System test

Through the system designed in this paper, and the deployment of linkage devices is scanned using vulnerabilities, the school network security and the school intranet security can be effectively guaranteed. In the process of implementing department planning, it is necessary to be rational, so as to achieve the uniformity of department configuration strategies, to provide a basis for the configuration of subordinate departments and network management application service systems, and to ensure the network security of all departments. Figure 4 shows the deployment plan of the system [6].

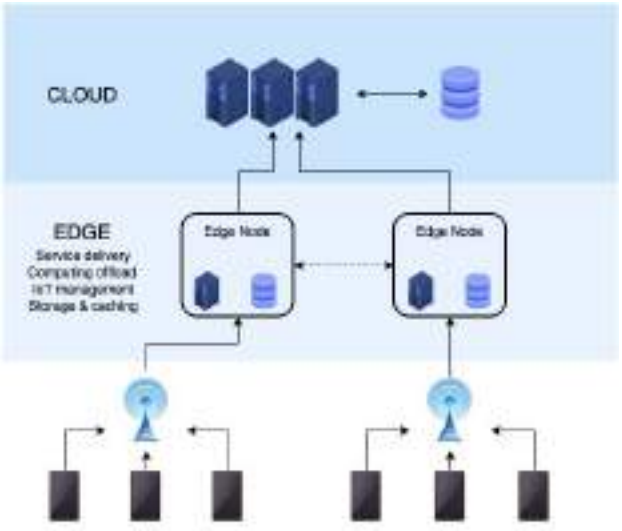


Figure 4. System test deployment scheme

V. Measures to Prevent Computer Network Information Security Risks in the Context of Big Data

A. Strengthening Account Password Security

When using a computer network system, various accounts such as computer system accounts, online banking accounts, and email accounts are inevitably involved. These accounts involve the privacy of the user. Once the account password is leaked, it will inevitably cause some damage to normal life. Impact. Specific methods to strengthen account password security include: setting difficult and complicated passwords, using numbers and letters as much as possible to reduce the possibility of password cracking; avoiding the use of numbers from ID cards or other documents as passwords; avoiding the use of multiple websites The same set of user names and passwords, especially accounts involving personal property such as online banking; to avoid leaking personal information, try not to visit illegal websites.

B. Improve the environment for system operation

For some important hardware equipment, maintenance and treatment should be carried out regularly. If problems are found, they should be handled in time to ensure the normal use of the entire hardware equipment. In addition, relevant technical staff should also be provided with special technical training. The training content includes data security prevention knowledge and the characteristics of the Internet to improve their comprehensive professional quality. In addition, the registration management system should be improved. For example, for the maintenance of some equipment, registration work should be done to ensure that the system configuration, technical parameters, management and maintenance, and data performance of the social security system are in the best state.

C. Check the security performance of the terminal in time

Basic-level terminal equipment is the target of hacking in the actual use process, and is easily affected by viruses and Trojan horse programs. In this regard, the security of the access terminal needs to be strictly checked, especially the update and installation of virus software on the terminal, and the operating system patches must be updated in time to eliminate potential security risks in the terminal.

D. Implementing Intrusion Detection

Intrusion detection is a network prevention method using network communication technology, artificial intelligence technology and other monitoring methods. It is divided into two methods: statistical analysis method and signature analysis method. The statistical analysis method is based on statistical principles, and comprehensively analyses the system's action mode under normal conditions to determine whether there is an abnormality in the real-time action of the system. The signature analysis method is based on the known vulnerabilities of the system to prevent security. Through template matching, the existing attack mode A problem was found in the signature. Intrusion detection can effectively detect computer network information security risks, and certain measures can be taken in time to avoid the risks.

VI. Conclusion

During the continuous development of big data technology, the channels of network attacks are constantly changing, and the latency period of network attacks is not only increasing, but the speed of security threat infection is also getting faster and faster. Influence. Therefore, it is necessary to regularly use advanced security countermeasures to effectively improve the security defence capabilities of big data application centres, thereby achieving in-depth defence of network threats.

Acknowledgments

This work was financially supported by Guangxi university scientific research fund: Research on the key technologies of vehicle-mounted network based on CPS (2013YB286).

References

[1] S. Vijayakumar Bharathi. Prioritizing and ranking the big data information security risk spectrum. Global Journal of Flexible Systems Management, 18(2) (2017) 183-201.

[2] James T. Graves, Alessandro Acquits, & Nicolas Christin. Big data and bad data: on the sensitivity of security policy to imperfect information. University of Chicago Law Review, 83(1) (2016)117-137.

[3] Santosh Aditham, & Nagarajan Ranganathan. A system architecture for the detection of insider attacks in big data systems. IEEE Transactions on Dependable and Secure Computing, 15(6) (2018) 974-987.

[4] Xiaoming Wang, Carolyn Williams, Zhen Hua Liu, & Joe Croghan. Big data management challenges in health research. Briefings in Bioinformatics, 20(1) (2017) 1-12.

[5] Wu, Z., Niu, F., Pan, D., & Lei, J. Authority for swim based on attribute encryption., 43(3) (2017) 350-357.

[6] Liu, Y., Wang, X., Zhang, J., Zhang, M., Peng, L., & Xu, A. W. An improved security 3d watermarking method using computational integral imaging cryptosystem., 12(2) (2016) 1-21.

[7] Sharma Kartik; Aggarwal Ashutosh; Singhania Tanay; Gupta Deepak; Khanna Ashish (2019). Hiding Data in Images Using Cryptography and Deep Neural Network. Journal of Artificial Intelligence and Systems, 1, 143–162.

[8] G. H. Rosa, J. P. Papa (2019). Soft-Tempering Deep Belief Networks Parameters Through Genetic Programming. Journal of Artificial Intelligence and Systems, 1, 43–59.

Research on Computer Network Information Security System Based on Big Data

Altaf Alam - 02
Prasad Arote - 06
Krish Chaurasiya - 12
Yash Dave - 19

Introduction to Computer Network Technology

1. Improvement in computer network technology.
2. Favors the development of information technology.
3. Enhances convenience and industrial production efficiency.
4. Achieves social change .
5. Emphasize the Importance of Computer network security.

Basic Overview of Big Data

1. Emergence of the term "Big Data" since 2012.
2. Rapid expansion of internet information technology.
3. Proliferation of information resources.
4. Challenges for future enterprise development.
5. Vital role of network information in enterprise operations.

Hidden Dangers Big Data Security

1. Big Data includes
   - Structured Data
   - Unstructured Data

2. Traditional Relational Databases cannot be used for unstructured data.

3. Data Management Software are emerging but they have some issue in data protection.

4. Thus the openness and complexity of Big Data can create security risks.

Big Data Security Risks

1.Privacy Data Leakage:

- Data used for analysis and target marketing.
- Increased Security Risks such as data breaches, hacking

2. Technical Information Leakage:

- Sensitive(core data) of enterprise stored in critical systems(ERP,CRM etc).
- Vulnerabilities via VPNs by competitors or hackers.

3. Government Data Outflow:

- Security of government databases is essential for national security.
- Illegal access to information poses threat to citizens and can lead to social instability.

4. Internal Data Leakage:

- Computer Information Systems have administrative roles with significant privileges.
- Driven by personal interests the administrators may copy or tamper the data.

Importance of Computer Network Information Security

A. Vulnerability of Computer Networks:

- Openness, interconnection, diverse connection methods, and uneven terminal distribution make computer networks susceptible to threats.
- Threats include computer viruses, hackers, and malicious software.

B. Impact on Enterprises:

- Enterprises emphasize information data application and often establish websites and information platforms for data integration.
- This integration introduces security risks and can have negative effects on businesses.

` C. Crucial Network Security:

- Given the growing importance of information data, network security becomes paramount.
- Actively identifying potential threats and formulating countermeasures are essential to protect businesses in the era of big data.

Measures to prevent Security Risks

A. Intrusion Detection:

- Network Defense: Intrusion detection utilizes network communication and AI.
- a) Statistical Analysis: Detect anomalies in system behavior.
- b) Signature Analysis: Identify known vulnerabilities.
- Effective Risk Mitigation: Detect and respond to network security threats promptly.

B. Terminal Security:

- Hacker Targets: Basic terminals are frequent hacker targets.
- Security Measures: Rigorously inspect access terminals, keep antivirus software updated, and maintain current OS patches to eliminate security risks.

Measures to prevent Security Risks

C. Enhancing System Operation:

- Regular Maintenance: Ensure hardware operates smoothly.
- Technical Training: Educate staff on data security and internet characteristics.
- Improved Registration: Implement equipment registration for optimal system setup, parameters, and performance.

D. Strengthening Account Password Security:

- Importance: Protect privacy across various accounts (e.g., systems, online banking, email).
- Password Tips: Create complex, alphanumeric passwords, Avoid common choices (e.g., ID card numbers), Use unique login credentials for each account, Refrain from visiting unsafe websites.

NETWORK SECURITY DÉFENSE SYSTEM DESIGN

1. System Requirements:

- Level 1 focuses on internet access security, data restoration, and user identity verification.
- Level 2 addresses the interconnection of different subnetworks.
- Level 3 ensures secure access to services and intrusion detection.

2. Security Defense Function:

- An active defense system is proposed to enhance network security and proactively safeguard against a range of threats, including viruses and Trojans, known for their potential to cause substantial damage with prolonged latency periods.
- This system is designed to actively improve network security operations by countering diverse threats and attacks.

NETWORK SECURITY DÉFENSE SYSTEM DESIGN

3. Security Protection:

- Common security measures like firewalls, antivirus software, and digital signatures work together to strengthen the integrity of the network.
- The system utilizes a blend of defense technologies to protect against data infection and attacks, ensuring the security of data during communication.

4. System Testing:

- Employing vulnerability scans to identify and address network weaknesses.
- The deployment plan plays a pivotal role in securing the entire school network, ensuring consistent configurations, and facilitating effective network management.

References

Reference to IEEE paper- https://drive.google.com/file/d/1f2XOniVxYm6afWEDXAqt0HkhprrVfCzo/view?usp=drive_link