fPRACTICAL: 1

AIM:

A security training institute is setting up a lab for ethical hacking workshops. The team must decide whether to use Kali Linux or Parrot Security OS, prioritizing ease of installation, hardware requirements, and post-installation configuration for beginner and intermediate students. Evaluate and install the most suitable penetration testing operating system for a security team by exploring the installation processes and user-friendliness of Kali Linux and Parrot Security OS.

THEORY:

Kali Linux *(formerly known as Backtrack Linux)* is an open-source, Debian-based Linux distribution which allows users to perform advanced penetration testing and security auditing. It runs on multiple platforms and is freely available and accessible to both information security professionals and hobbyists.

This distribution has several hundred tools, configurations, and scripts with industry-specific modifications that allow users to focus on tasks such as computer forensics, reverse engineering, and vulnerability detection, instead of dealing with unrelated activities.

Parrot Security (ParrotOS, Parrot) is a Free and Open source GNU/Linux distribution based on Debian Stable designed for security experts, developers and privacy aware people.

It includes a full portable arsenal for IT security and digital forensics operations. It also includes everything you need to develop your own programs or protect your privacy while surfing the net.

CODE:

N/A

OUTPUT:

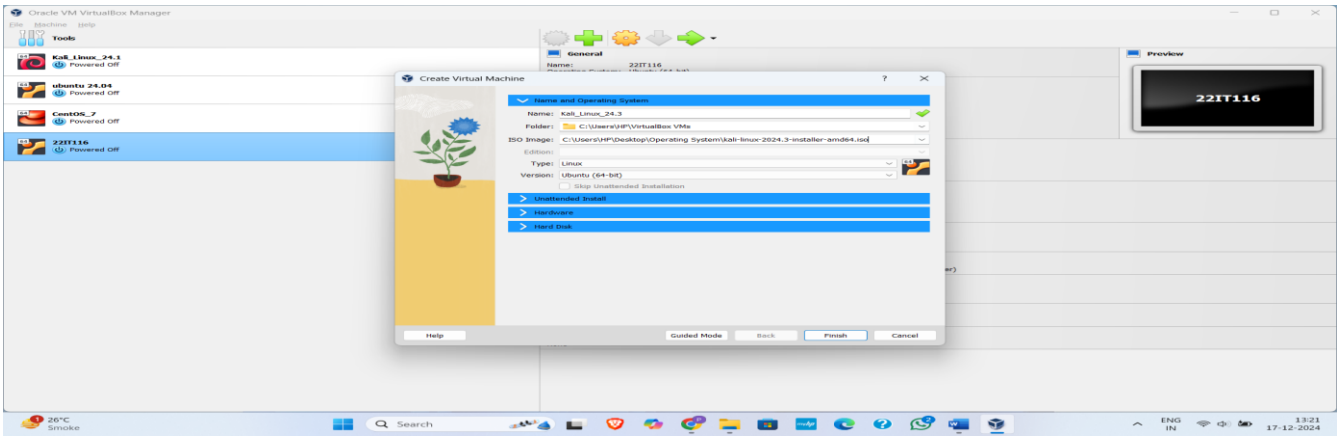


Figure 1: Set the name of your virtual machine and provide iso image

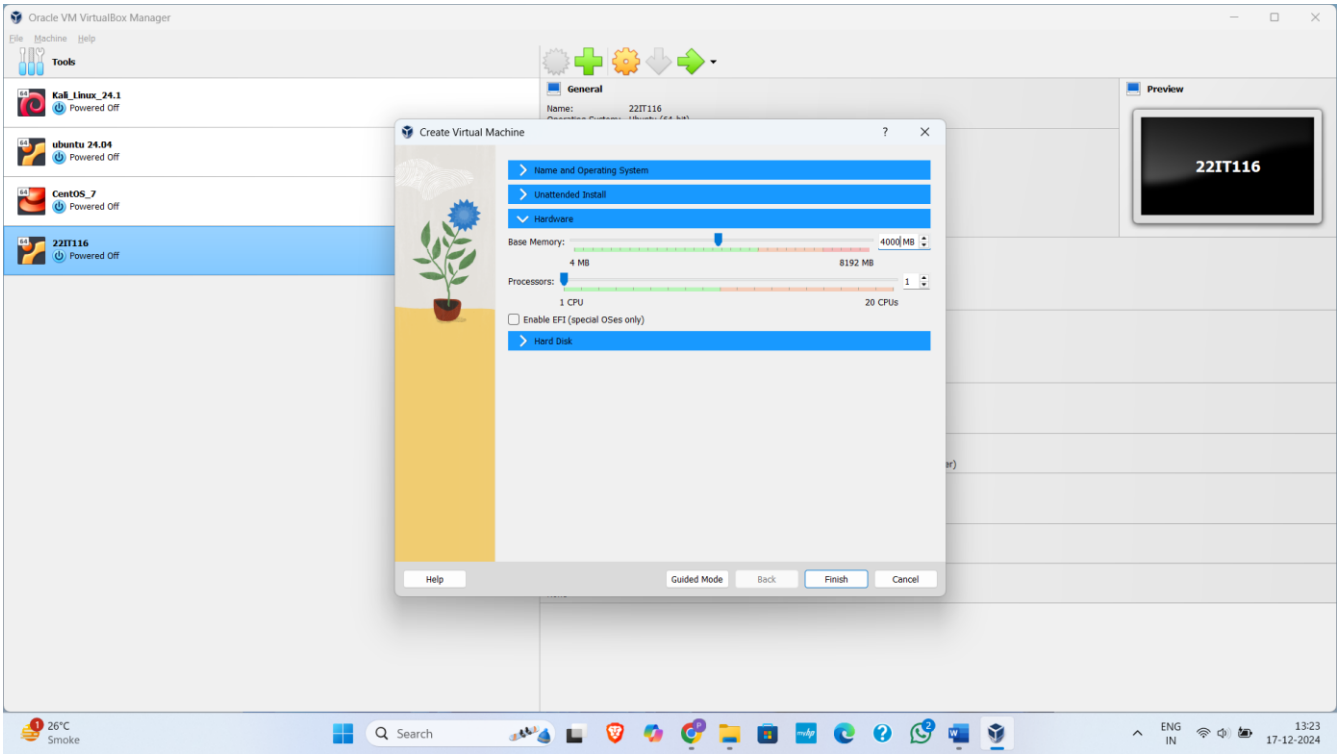


Figure 2:Provide RAM to your VM

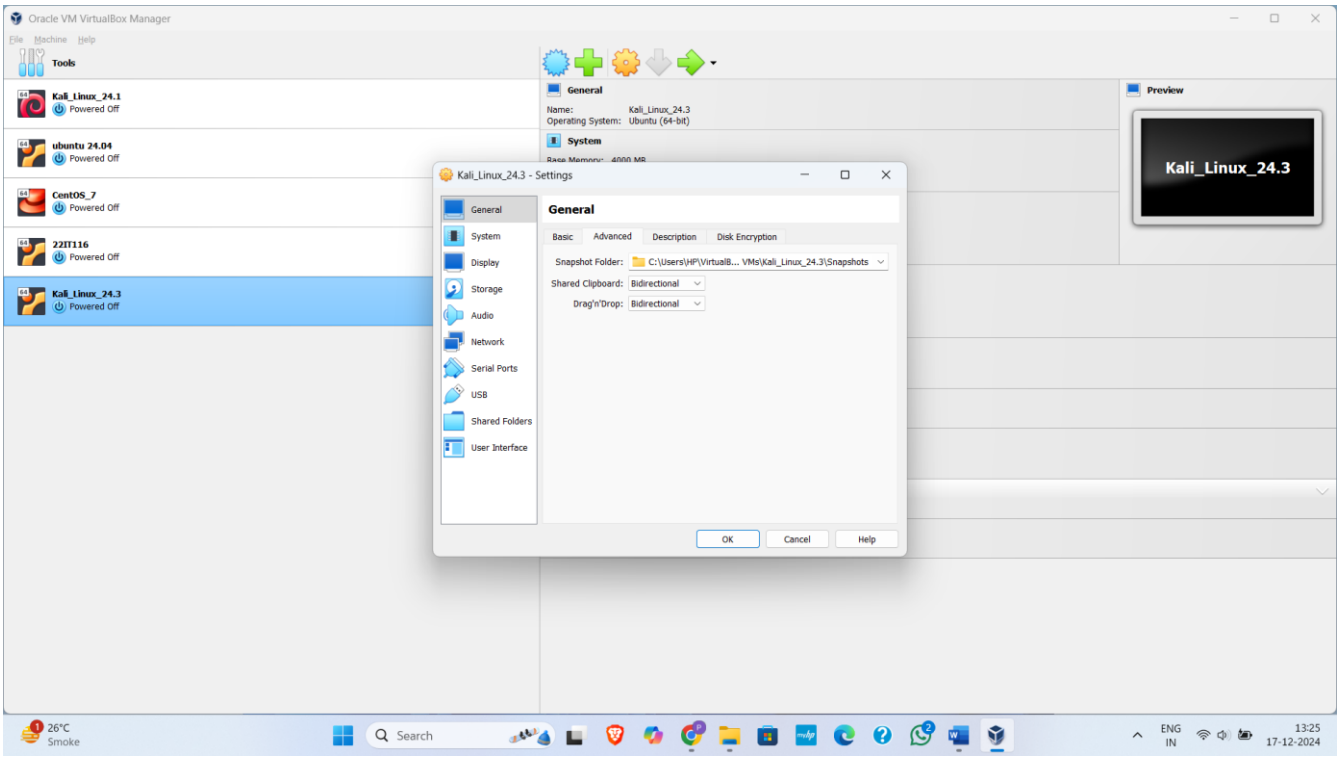


Figure 3: Set Clipboard bidirectional

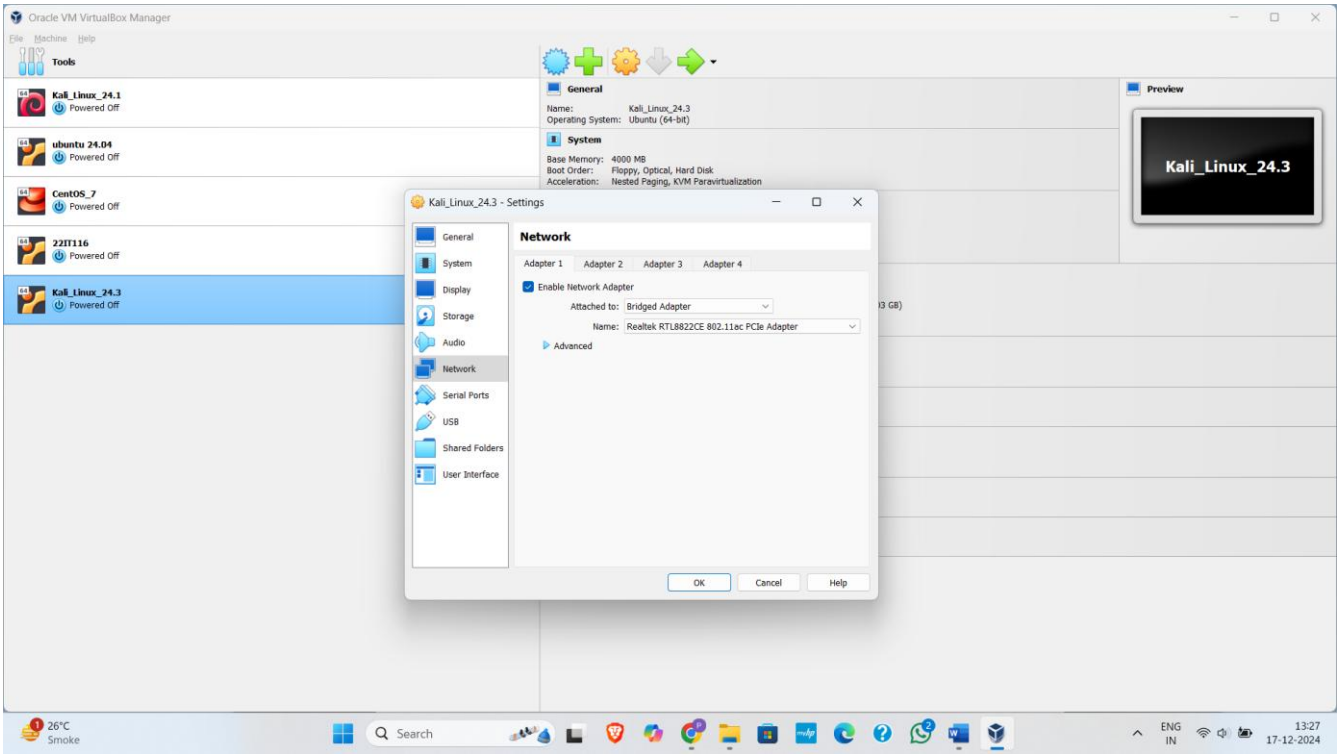


Figure 4: Set network to Bridged adapter

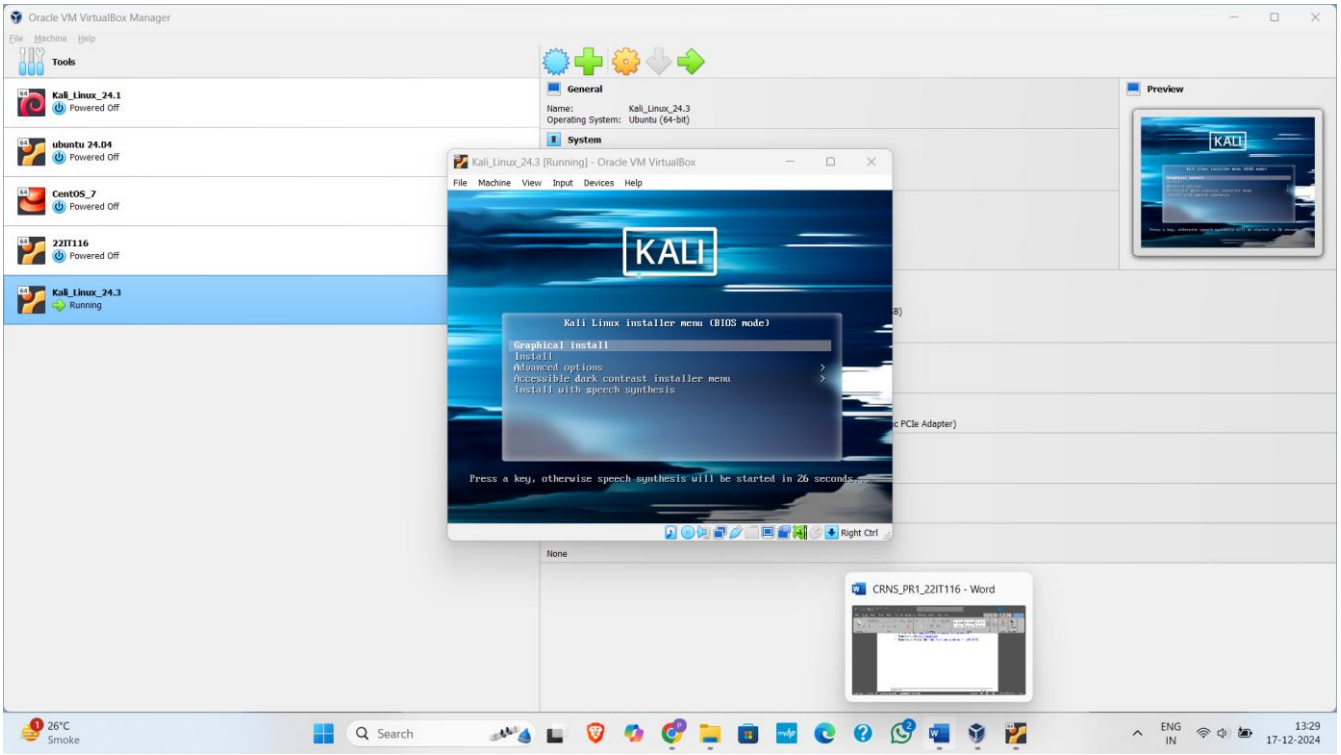


Figure 5:Run VM and install Graphical interface

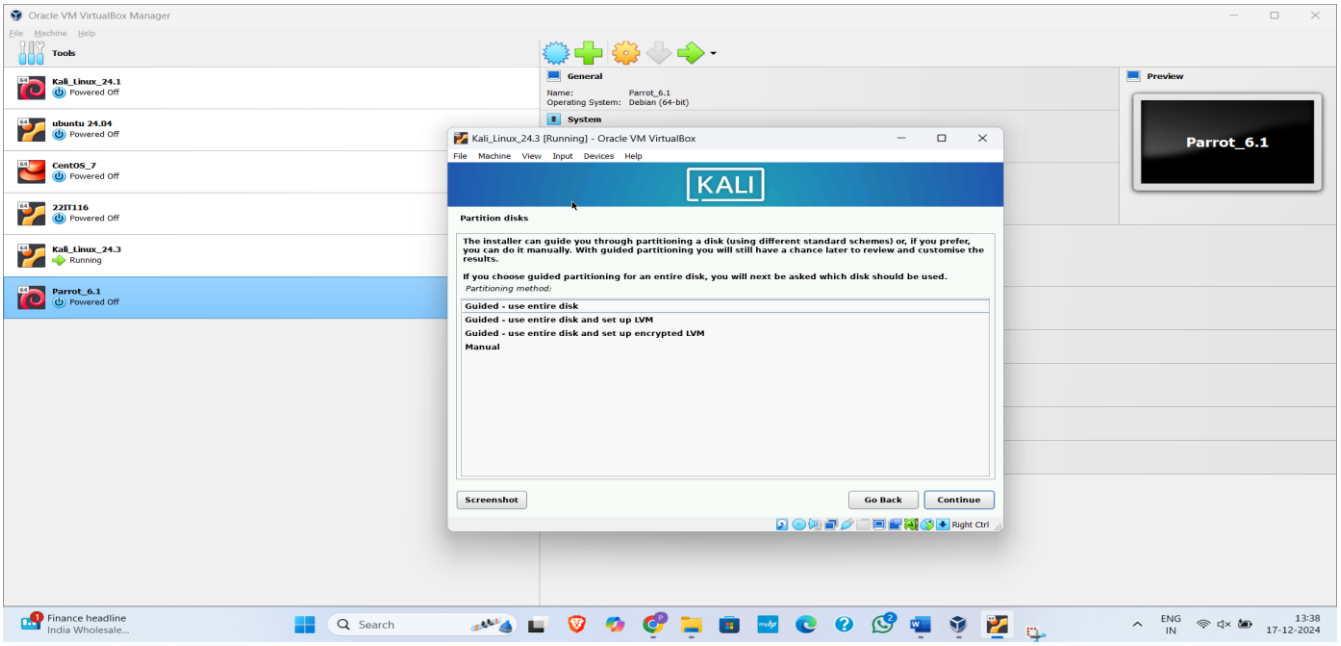


Figure 6:Use entire Disk partition disk

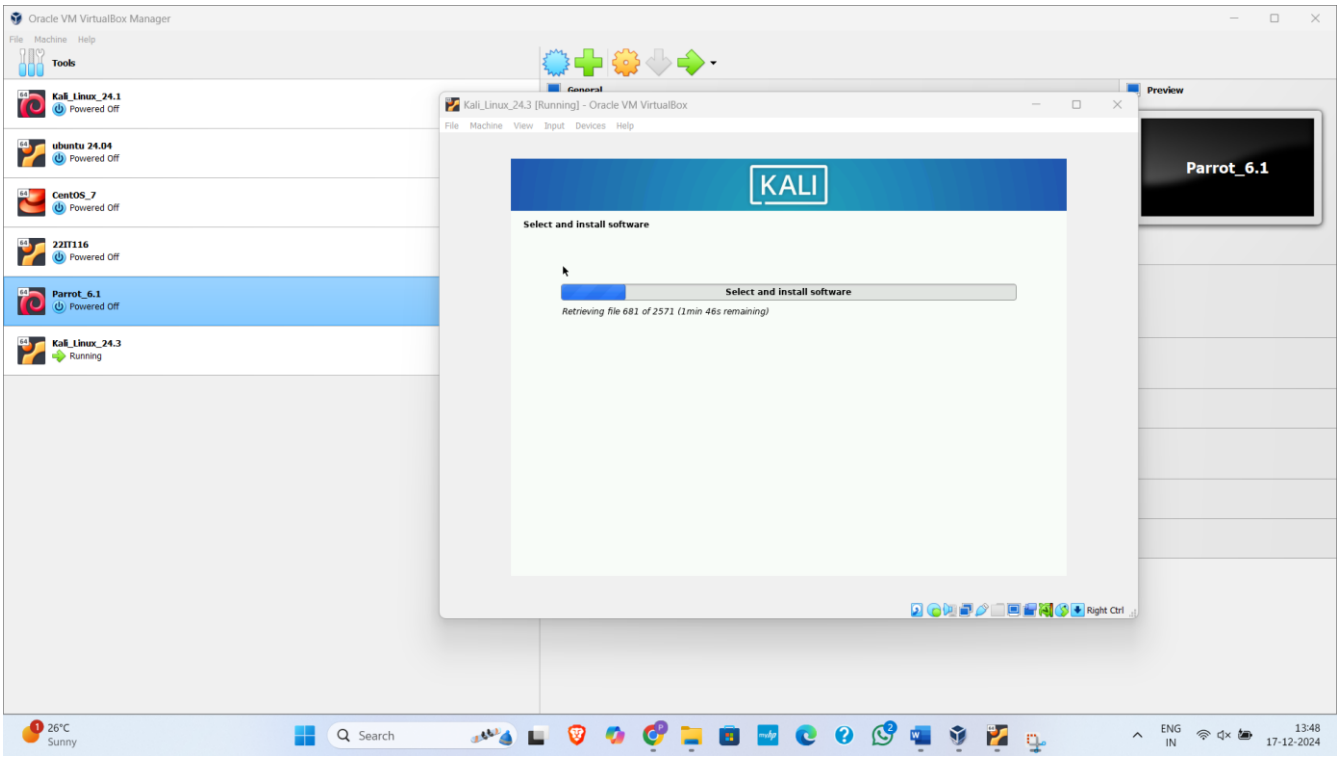


Figure 7:Install software

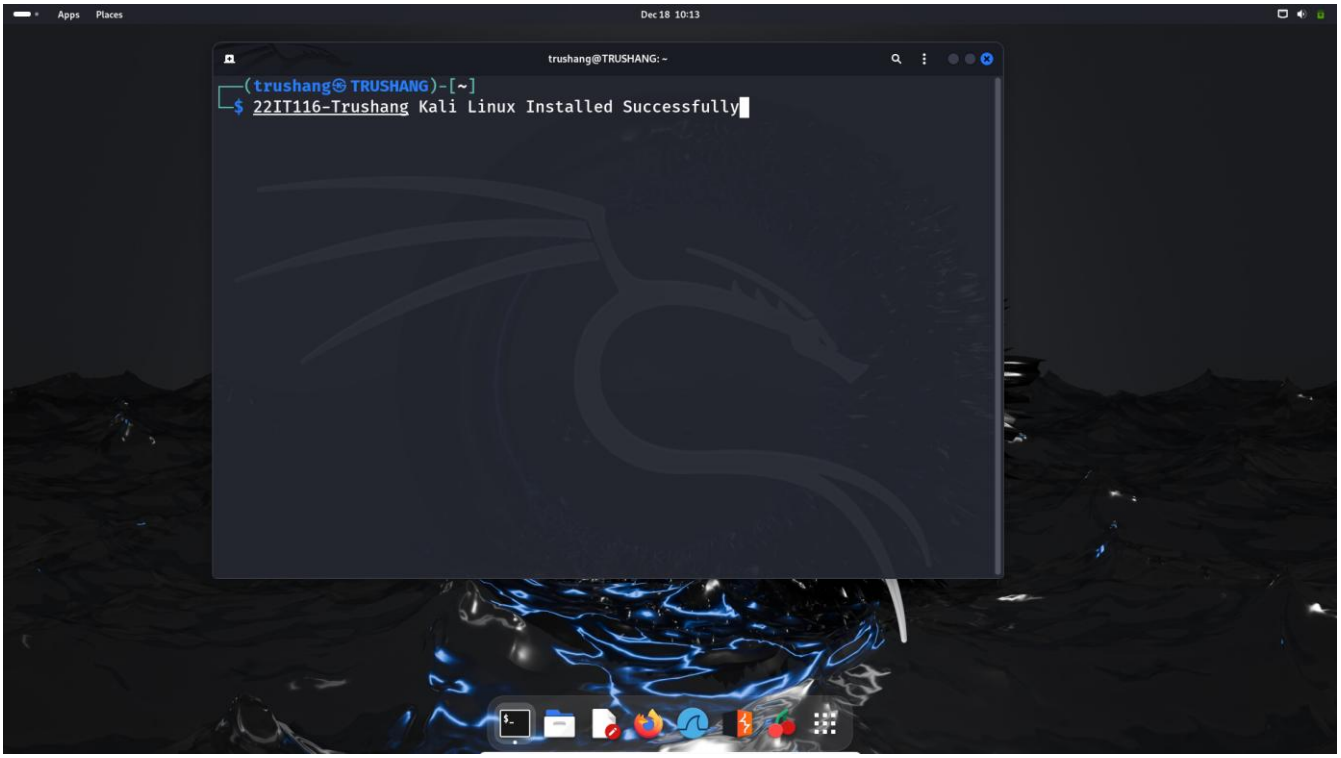


Figure 8: Kali Linux Installed Successfully

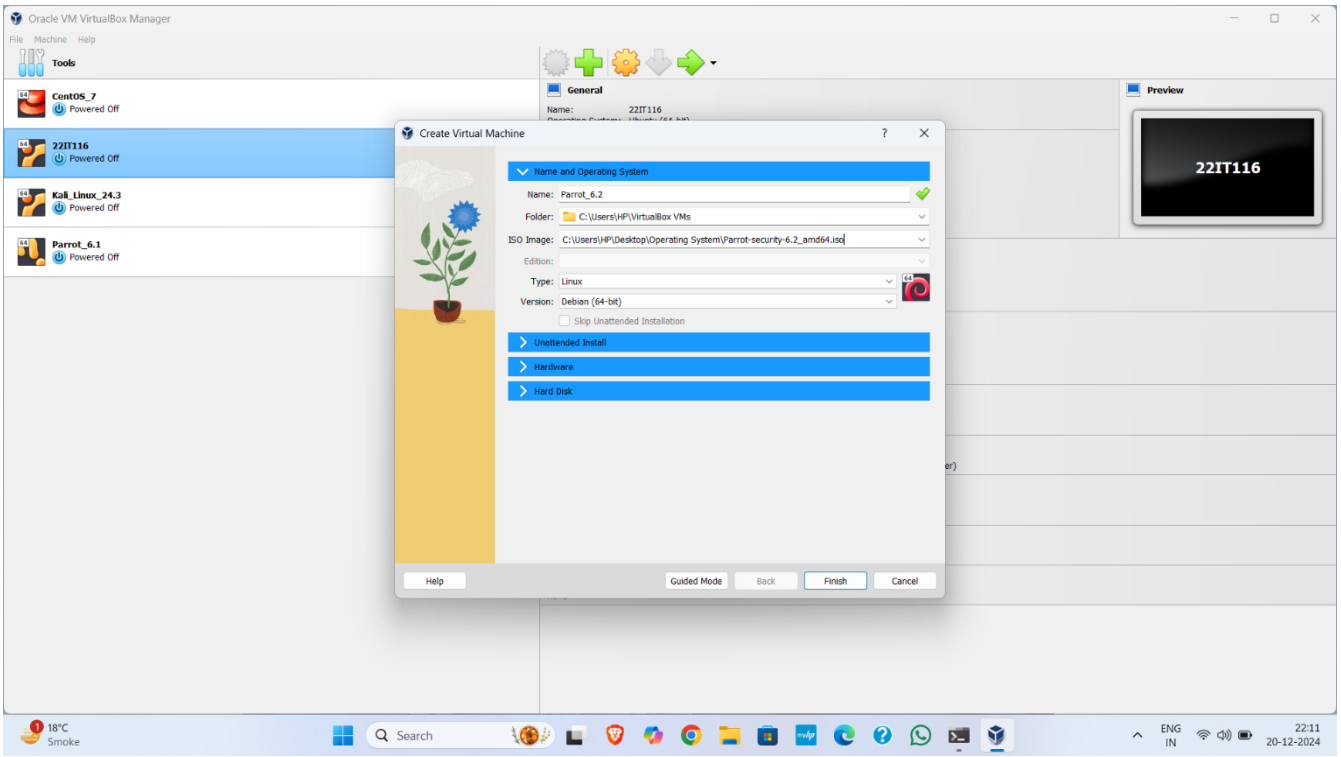


Figure 9:Set the name of your virtual machine and provide iso image

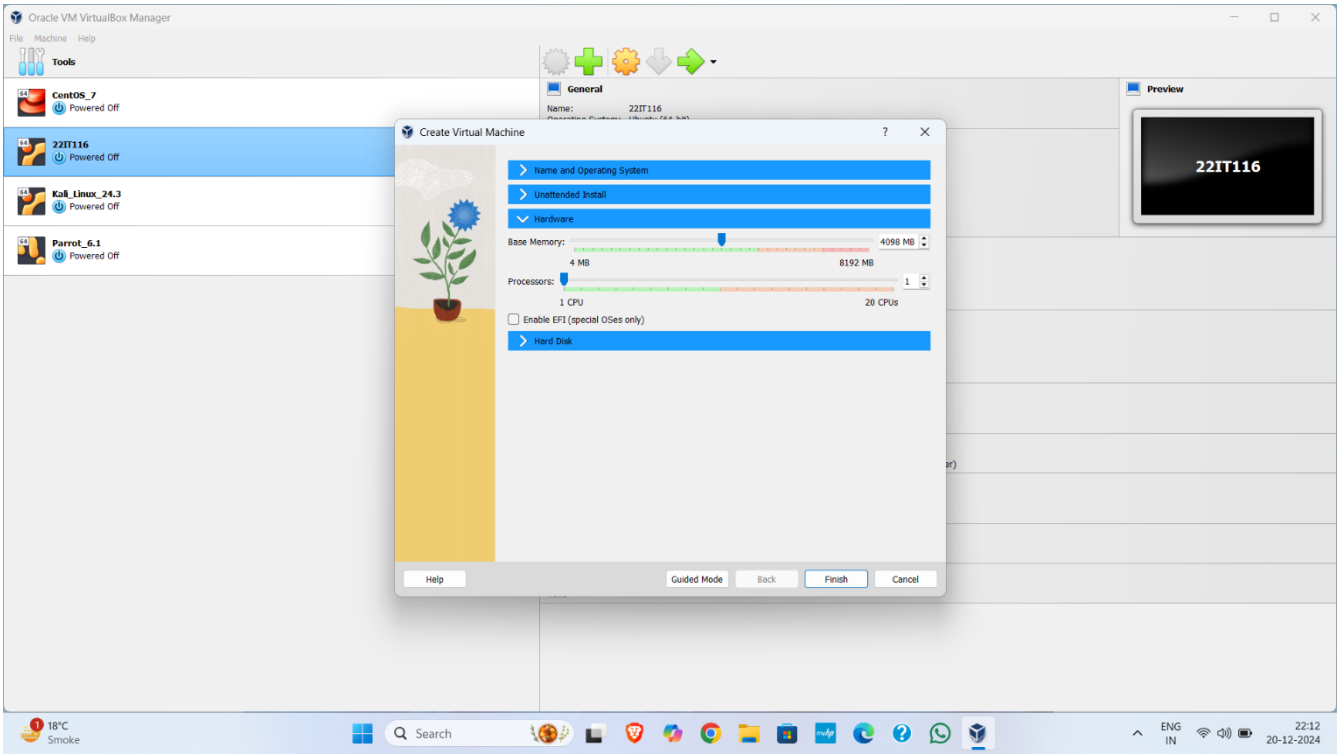


Figure 10:Provide RAM to your VM

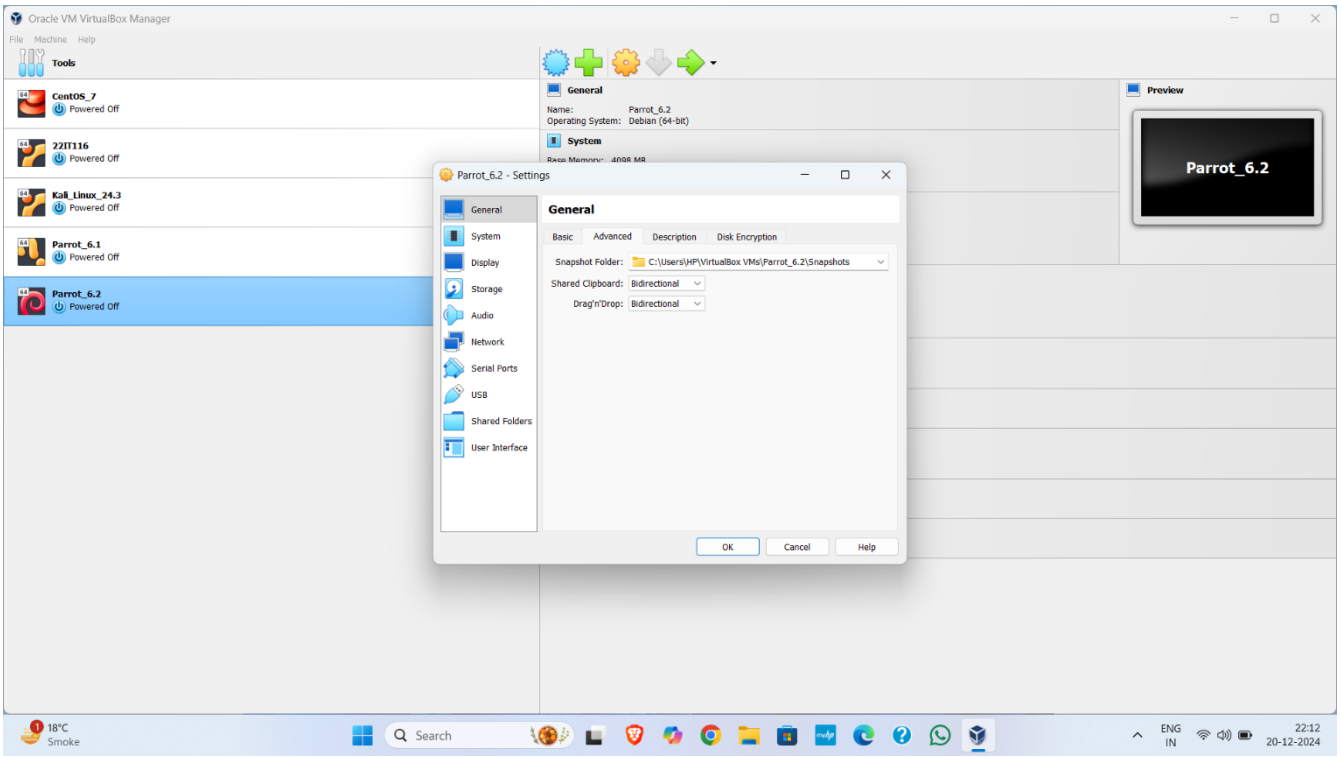


Figure 11:Set Clipboard bidirectional

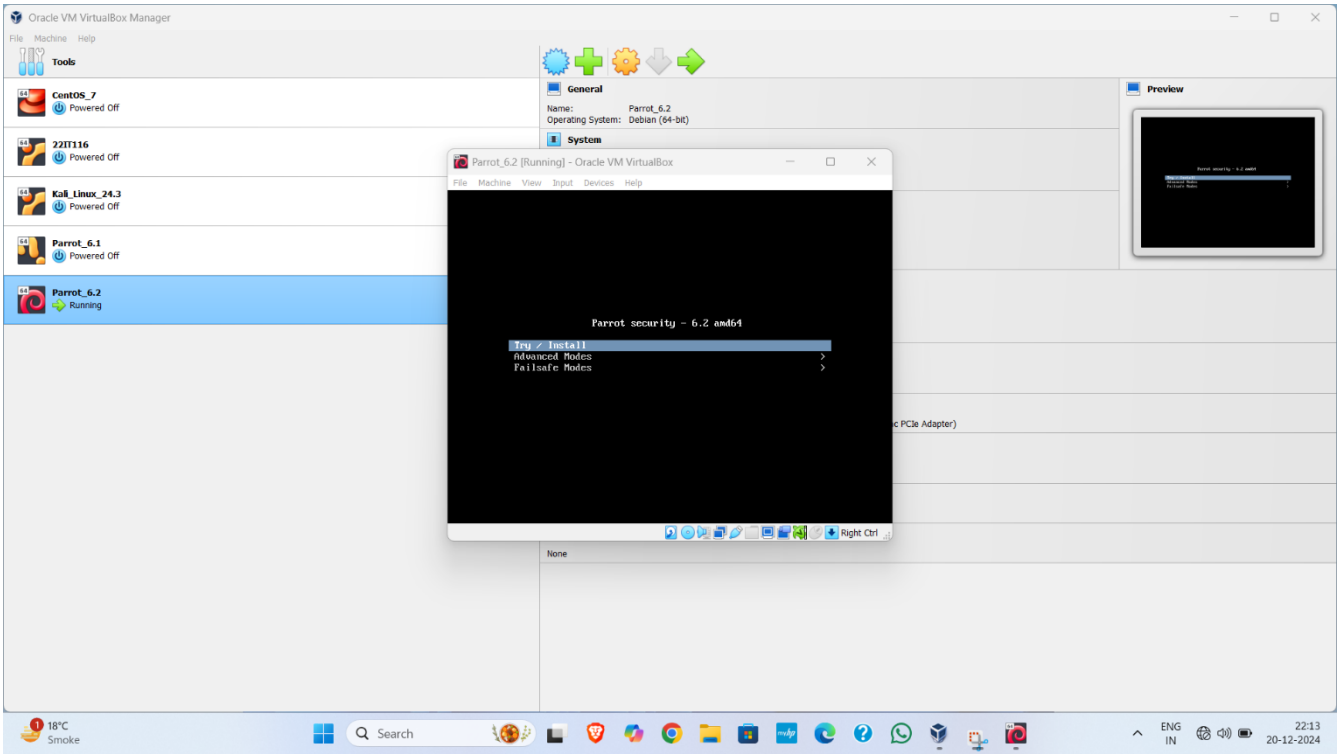


Figure 12:Install parrot

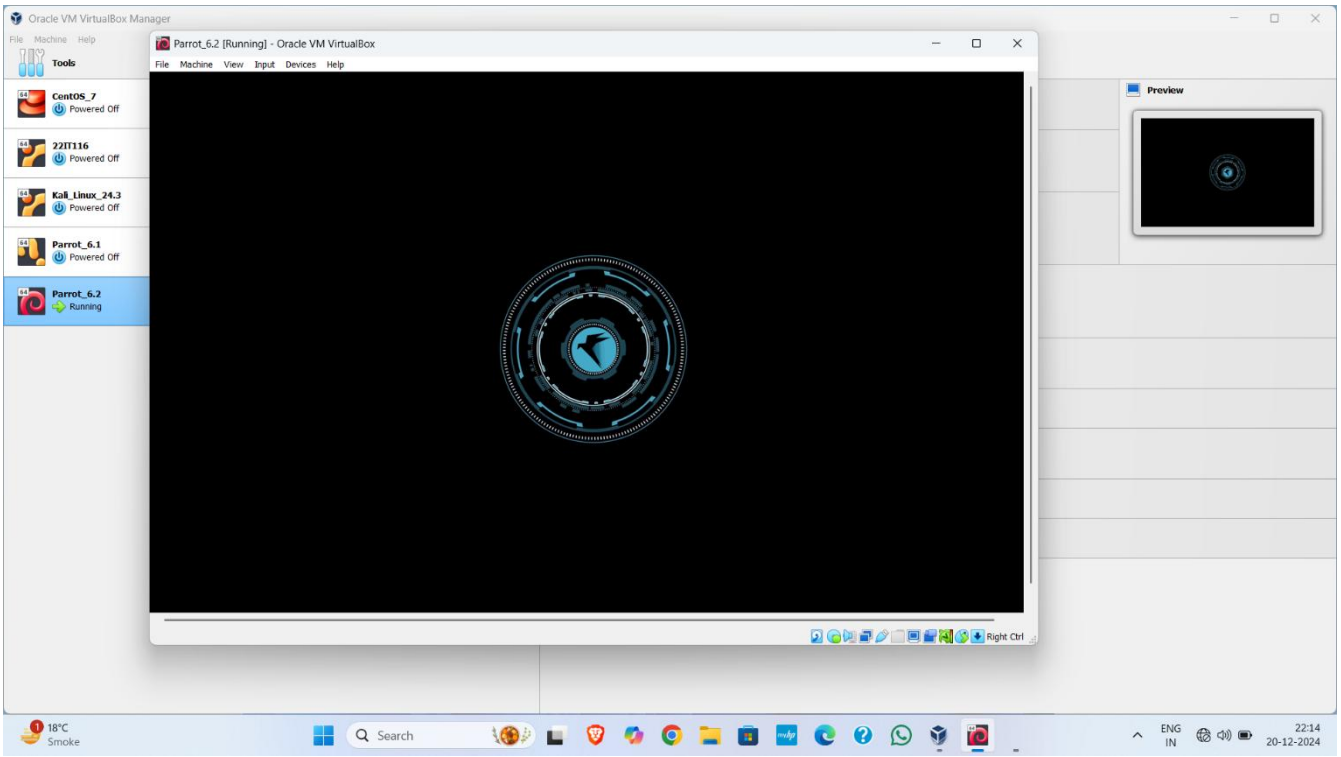


Figure 13: Starting Parrot

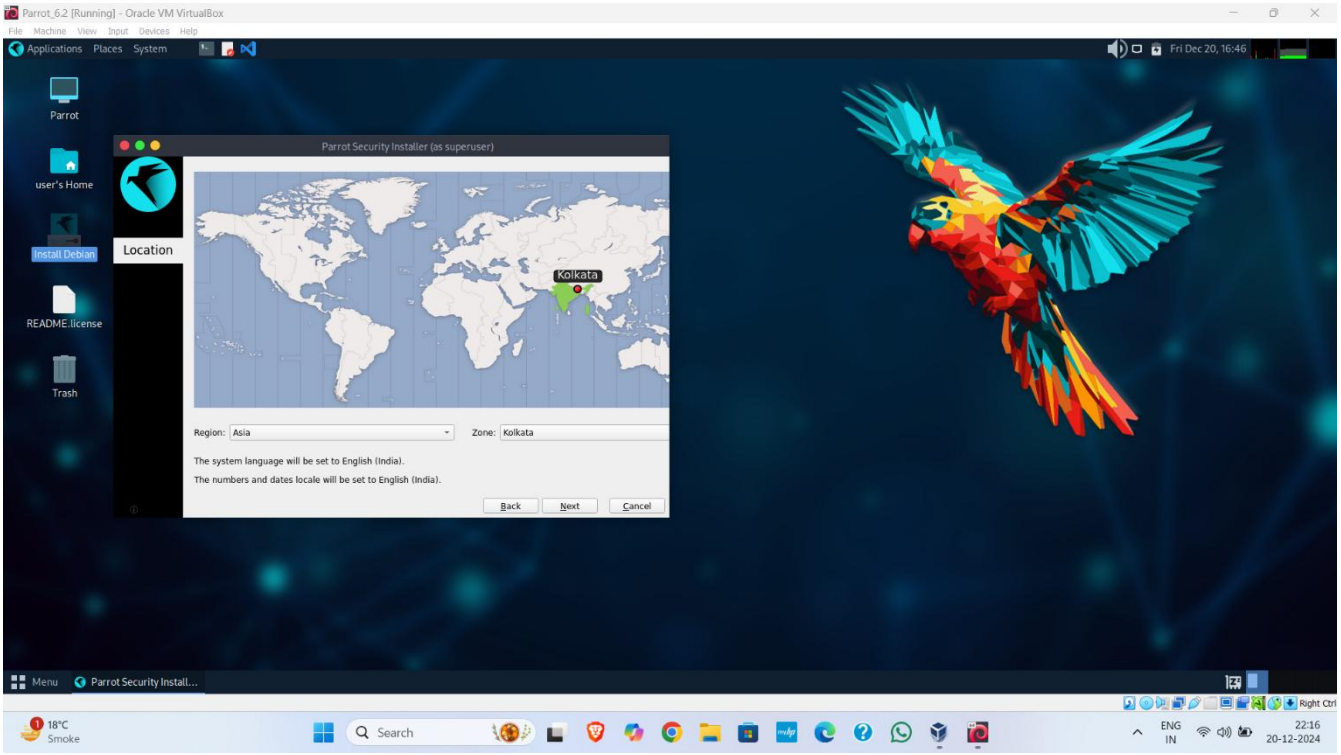


Figure 14: Set Location

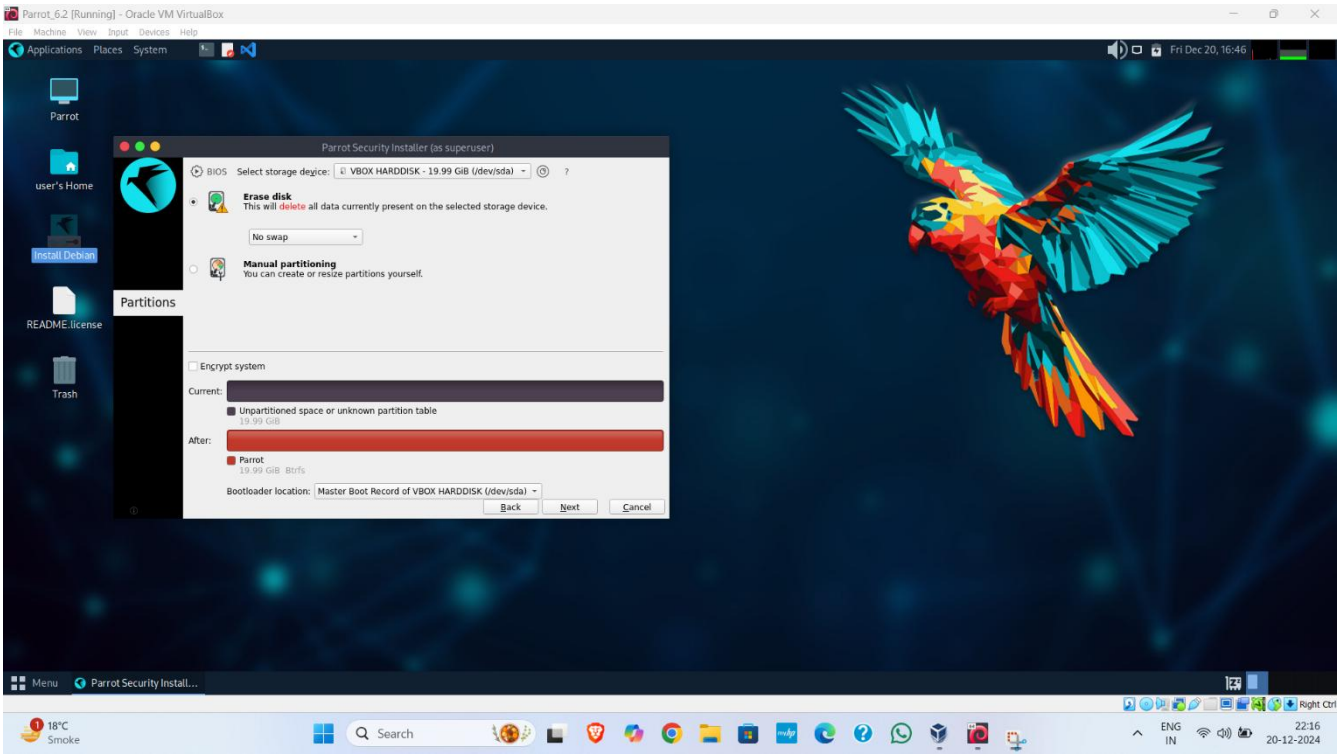


Figure 15 : Set partition

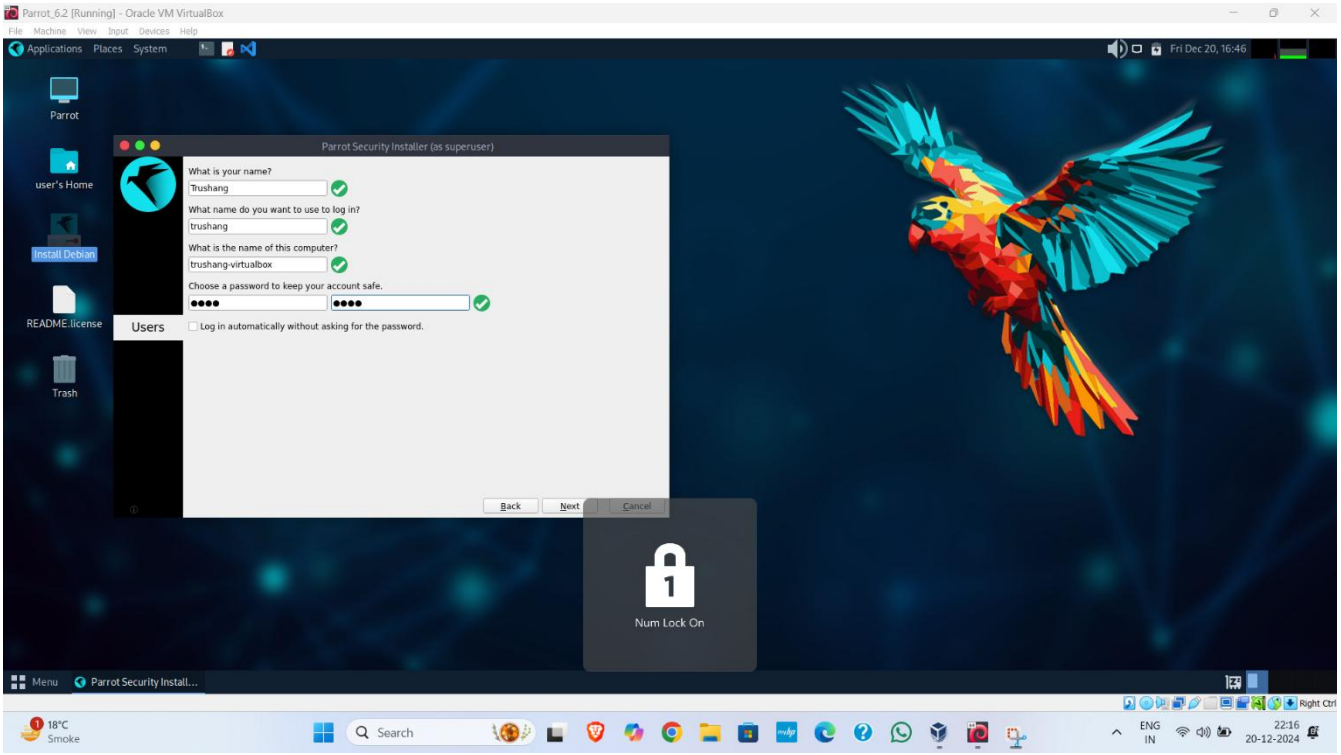


Figure 16: Create user name and password

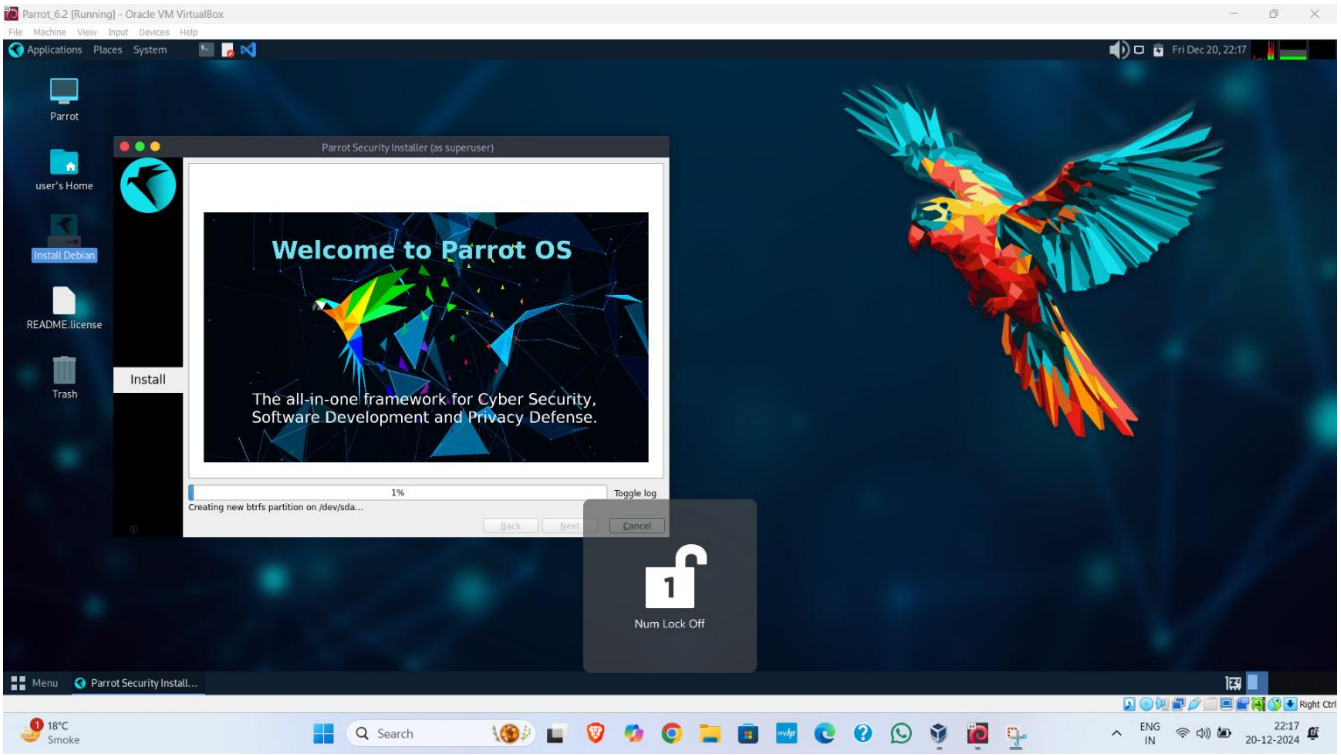


Figure 17: Started installing Parrot OS

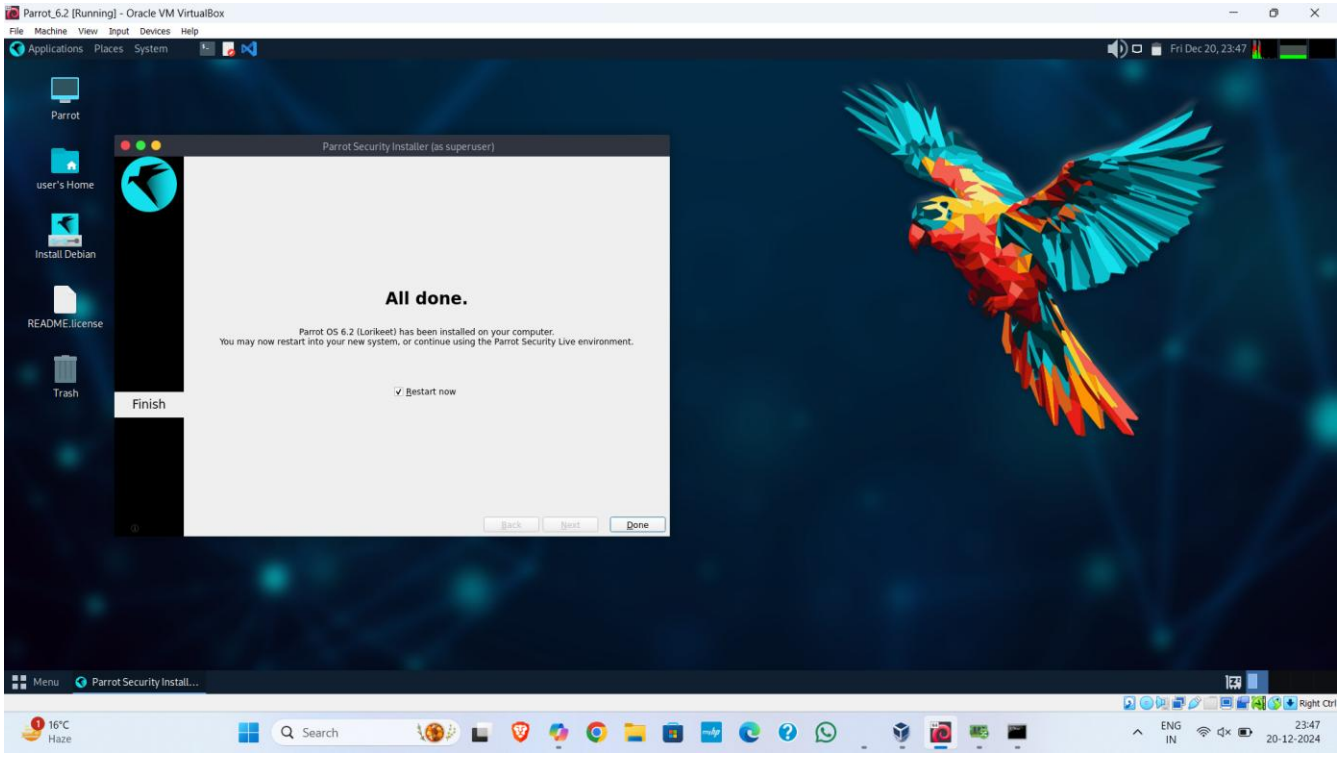


Figure 18: Complete installation of Parrot security OS

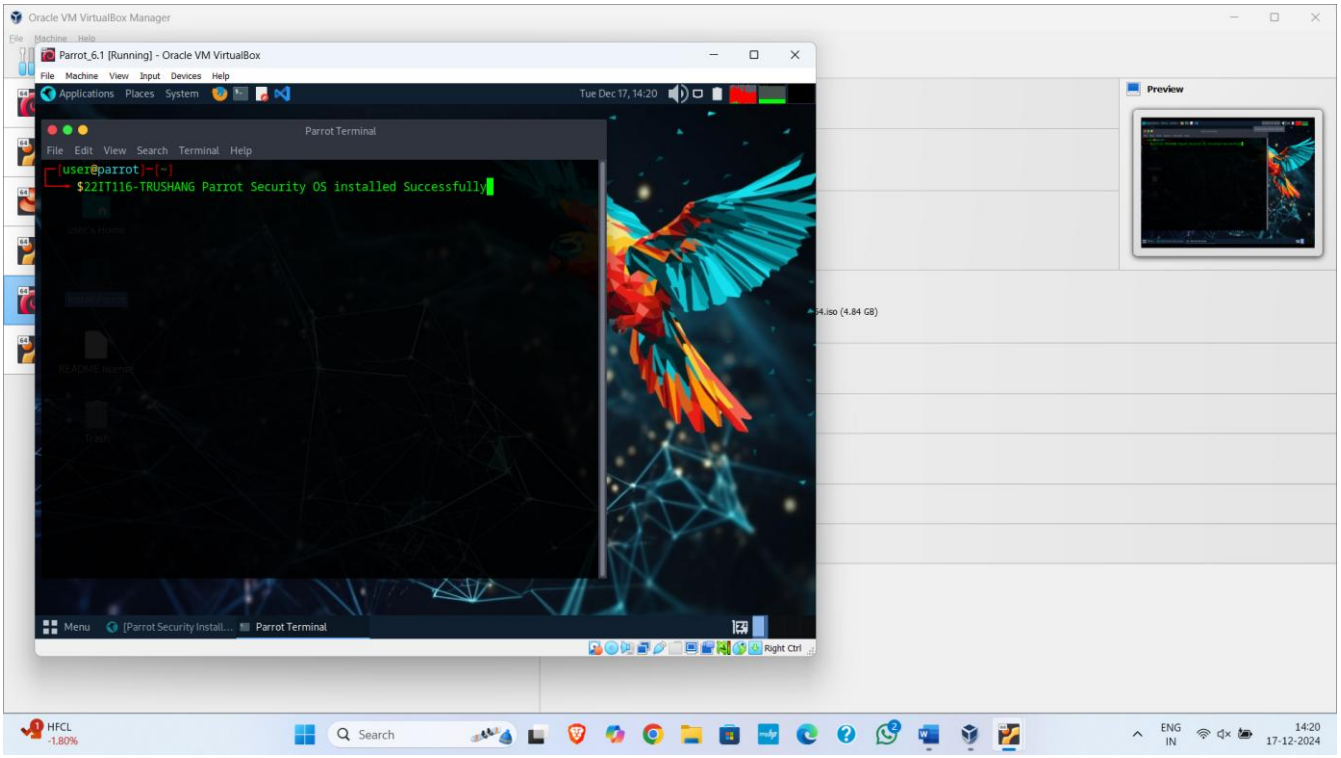


Figure 19: Final screenshot of install parrot

LATEST APPLICATIONS:

Kali Linux:

- Penetration Testing and Ethical Hacking
- Red Team Operations
- Forensics
- Cloud Security

Parrot security OS:

- Cryptography and Encryption
- Forensics and Digital Investigation
- Red and Blue Team Activities
- Web Application Security

LEARNING OUTCOME:

In this practical, we learn how to install Kali Linux and Parrot security OS in VM. We compare kali Linux and parrot security OS, Kali Linux is best for penetration testing and offensive security, offering a comprehensive suite of tools for ethical hackers. Parrot Security OS is ideal for privacy, digital forensics, and defensive security, providing a well-rounded platform for security professionals

REFERENCES:

1. Kali OS: https://www.kali.org/
2. Kali OS Installation: https://www.youtube.com/watch?v=jk2KGdJU2OI
3. Parrot Security OS: https://parrotsec.org/
4. Parrot Security OS Installation: https://www.youtube.com/watch?v=4qvFp99rfXw

PRACTICAL: 2

AIM:

The transmission of information needs to be secure over the communication channel and the data has to be confidential. To do so, steganography is the technique of concealing/hiding a secret file, message, audio, or video in another file format. Study and implement the practical approach for Steganography using the following tools: Steghide, StegoSuite & Xiao Steganography.

THEORY:

What is Steganography?

Steganography is the practice of "hiding in plain sight." Steganography encodes a secret message within another non-secret object in such a manner as to make the message imperceptible to those who aren't aware of its presence. Of course, because of this secrecy, steganography generally requires the recipient to be aware that a message is forthcoming.

How Steganography works

Steganography works by hiding secret information within a medium, such as an image, audio file, video, or even text, in a way that makes it undetectable to anyone who doesn't know where or how to look. The core idea is to conceal the secret data so that it's not obvious or suspicious to an observer. One common method is called **Least Significant Bit (LSB)** steganography, where secret data is hidden in the least important bits of a file, like an image or audio file.

For example, in an image, each pixel is made up of three-color values: red, green, and blue. Each of these values is stored as a byte (a group of 8 bits). In LSB steganography, the last bit of each byte is changed to hide secret information. Since this change is so small, it doesn't noticeably alter the image, making it look almost the same as the original. So, if you want to hide 1 megabyte of data, you would need an image that is 8 megabytes in size.

The same method can also be used for audio and video files, where small changes are made to the sound or visual elements, making it hard for anyone to notice the hidden data.

Types of steganography

From a digital perspective, there are five main types of steganography. These are:

1. Text steganography
2. Image steganography
3. Video steganography
4. Audio steganography
5. Network steganography

Text steganography involves hiding information inside text files. This includes changing the format of existing text, changing words within a text, using context-free grammars to generate readable texts, or generating random character sequences.

Image steganography

This involves hiding information within image files. In digital steganography, images are often used to conceal information because there are a large number of elements within the digital representation of an image, and there are various ways to hide information inside an image.

Audio steganography

Audio steganography involves secret messages being embedded into an audio signal which alters the binary sequence of the corresponding audio file. Hiding secret messages in digital sound is a more difficult process compared to others.

Video steganography

This is where data is concealed within digital video formats. Video steganography allows large amounts of data to be hidden within a moving stream of images and sounds. Two types of video steganography are:

- Embedding data in uncompressed raw video and then compressing it later
- Embedding data directly into the compressed data stream

Network steganography

Network steganography, sometimes known as protocol steganography, is the technique of embedding information within network control protocols used in data transmission such TCP, UDP, ICMP, etc.

Uses of steganography

In recent times, steganography has been mainly used on computers with digital data being the carriers and networks being the high-speed delivery channels. Steganography uses include:

- **Avoiding censorship:** Using it to send news information without it being censored and without fear of the messages being traced back to their sender.
- **Digital watermarking:** Using it to create invisible watermarks that do not distort the image, while being able to track if it has been used without authorization.
- **Securing information:** Used by law enforcement and government agencies to send highly sensitive information to other parties without attracting suspicion.

How to detect steganography

The practice of detecting steganography is called ‘steganalysis’. There are various tools that can detect the presence of hidden data, including StegExpose and StegAlyze. Analysts may use other general analysis tools such as hex viewers to detect anomalies in files.

However, finding files that have been modified through steganography is a challenge – not least because knowing where to start looking for hidden data in the millions of images being uploaded on social media every day is virtually impossible.

CODE:

- nano hide.py
- python hide.py -e '/home/trushang/Desktop/Dog. bmp'
- python hide. py -d ' /home/trushang/Desktop/Dog. bmp'
- sudo apt-get install steghide
- steghide –version
- nano secret.txt
- cat secret.txt
- steghide --embed -ef ' /home/trushang/Desktop/secret .txt' -cf '/home/trushang/Desktop /steghide_ image.jpeg ' -p 22it116
- steghide --extract -sf '/home/trushang/Desktop/steghide_image . jpeg' -p 22it116 -xf '/home/trushang/Desktop/secrets. txt '
- sudo apt-get install stegosuite
- stegosuite gui
- copy /b download .jpg + file. zip image .jpg
- ren image . jpg image. zip

OUTPUT:

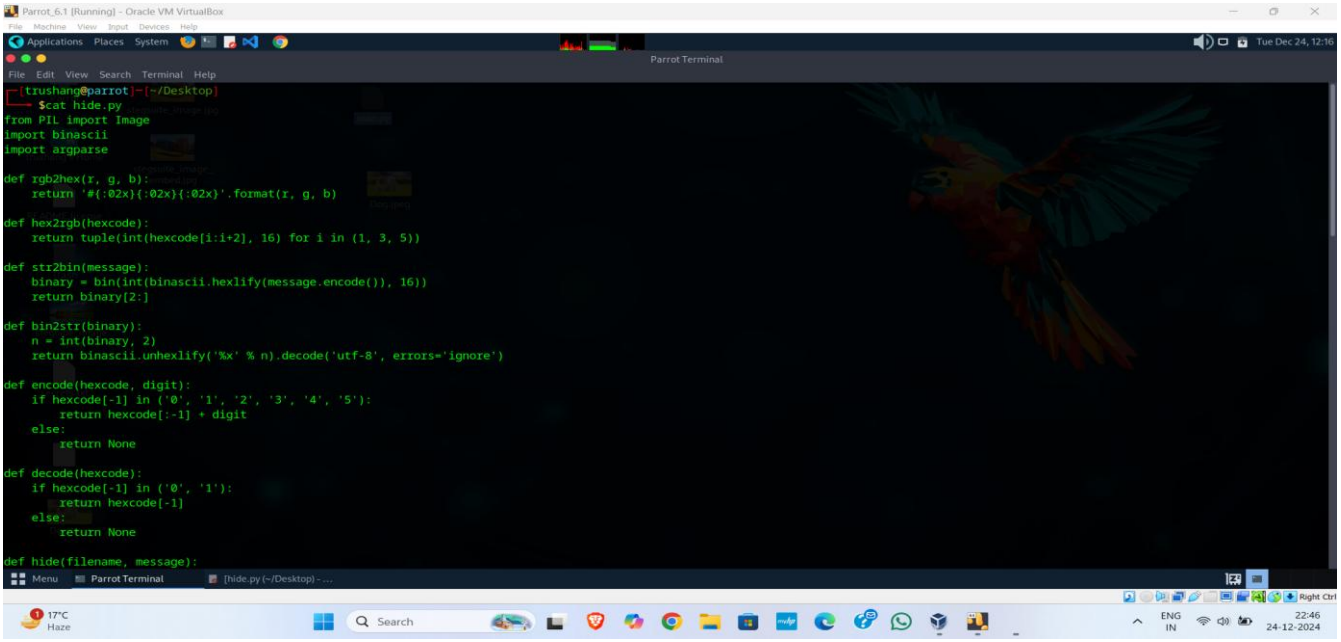


Figure 20:hide.py where we write a logic for hiding data in image

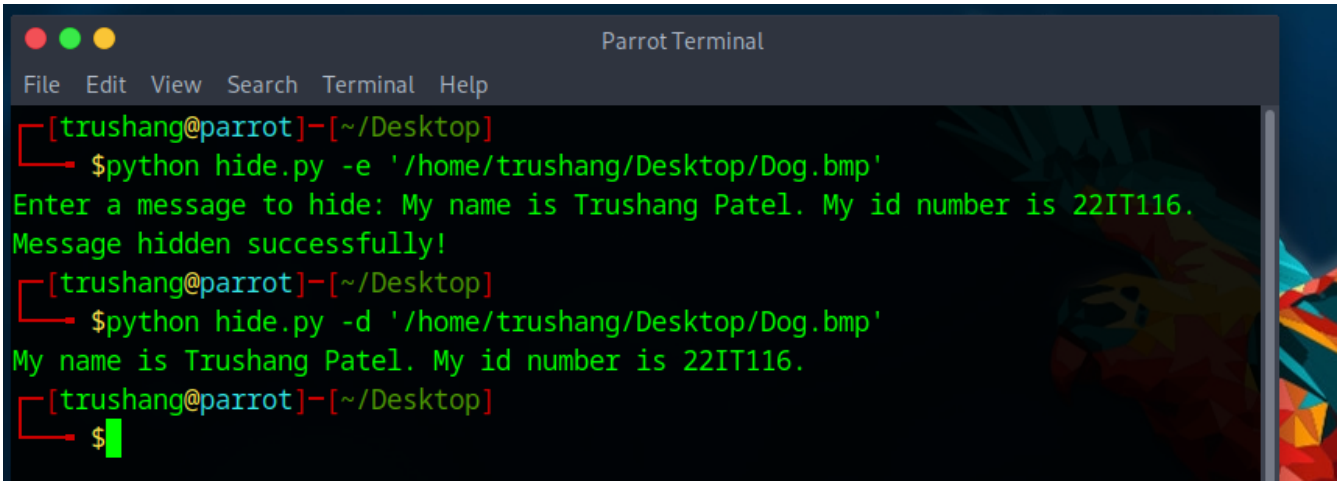


Figure 21:Hide message in image using python code

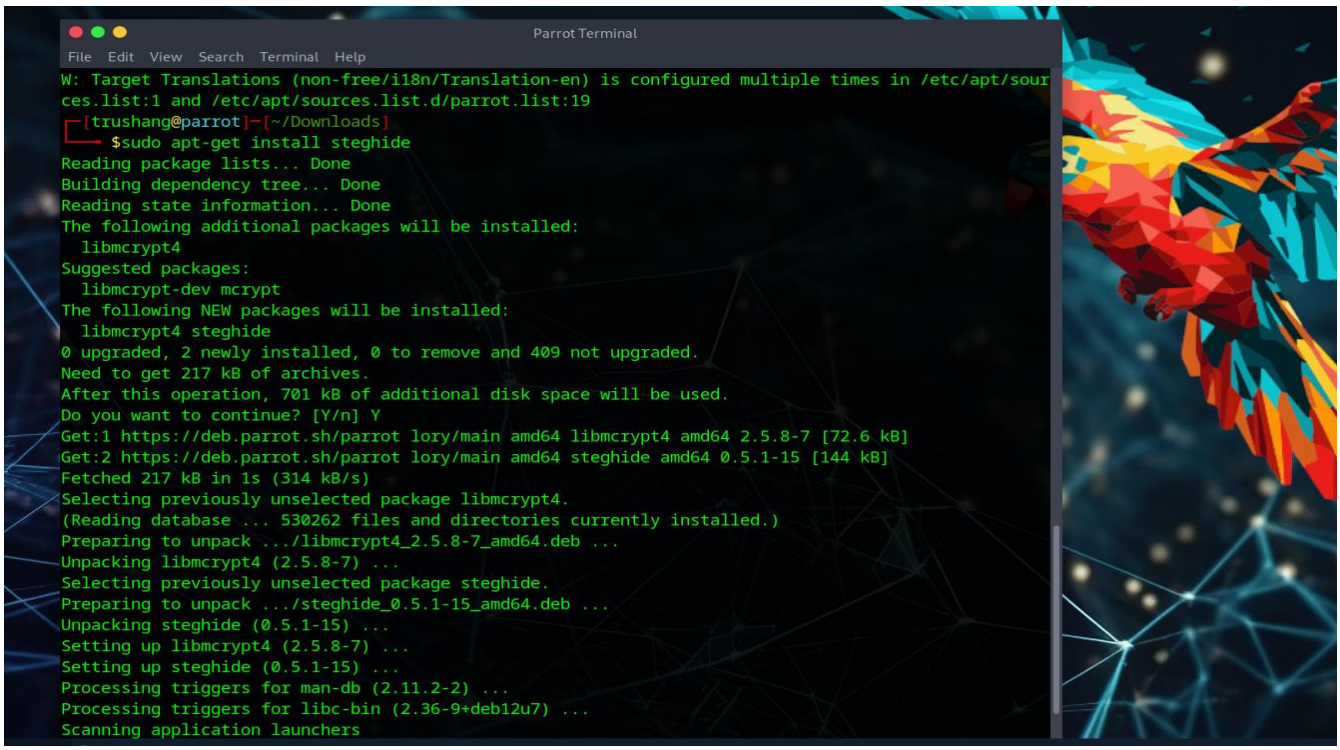


Figure 22:First install steghide in our parrot security os

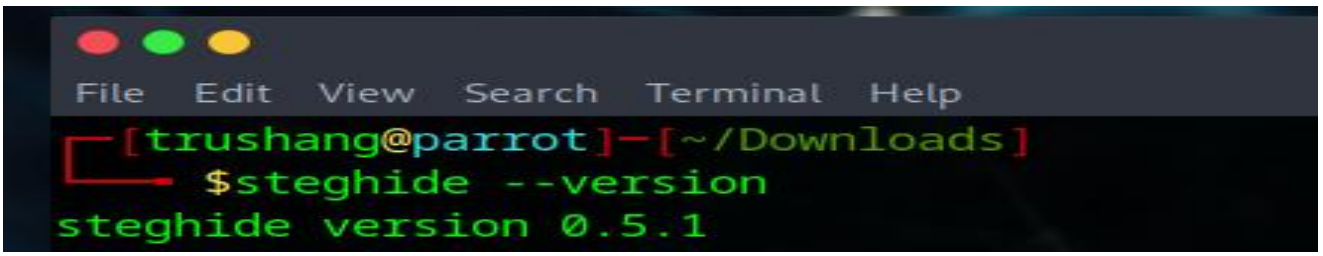


Figure 23:Check the version of steghide

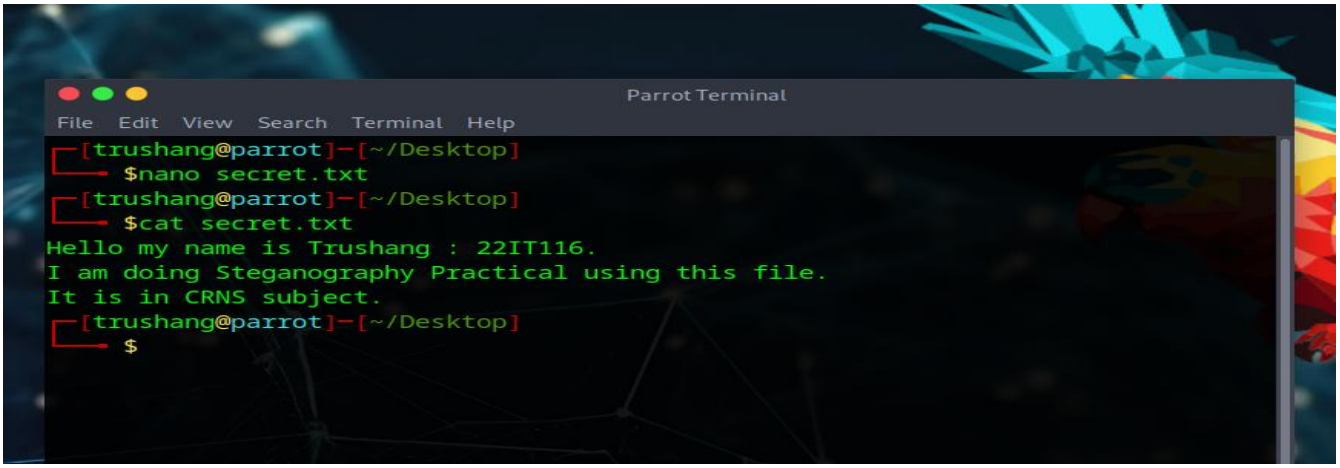


Figure 24: Simple create a secret text file which contains your secret data

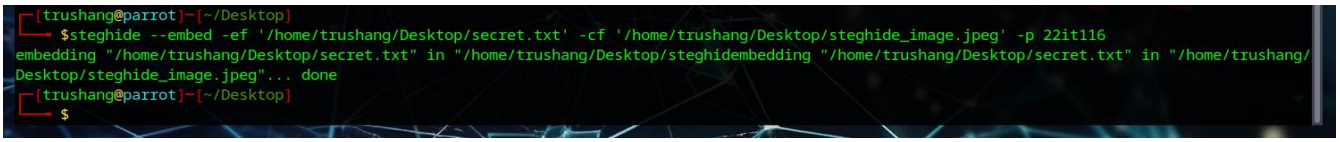


Figure 25: This command hides your secret file into steghide_image file

Figure 26:This command extracts your secret message and store it into secrets.txt

Figure 27:Output of secrets.txt which is hide by sender

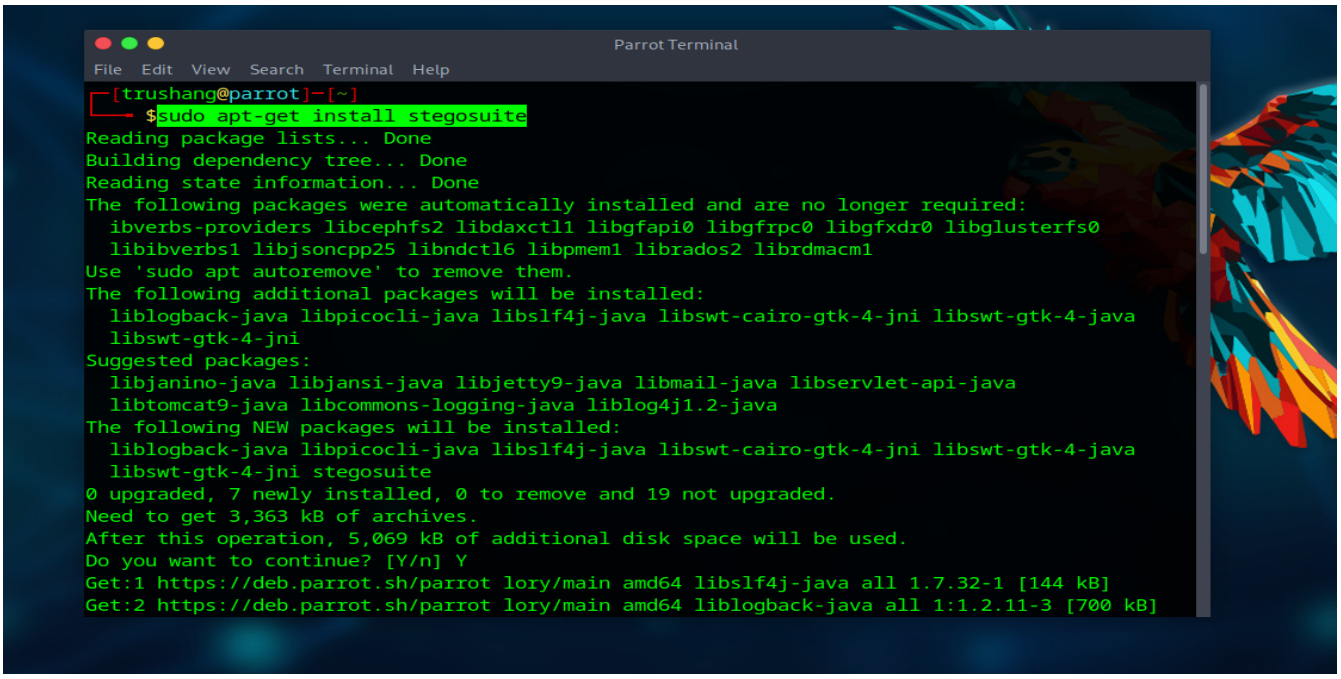


Figure 28:Install Steg suite in your system

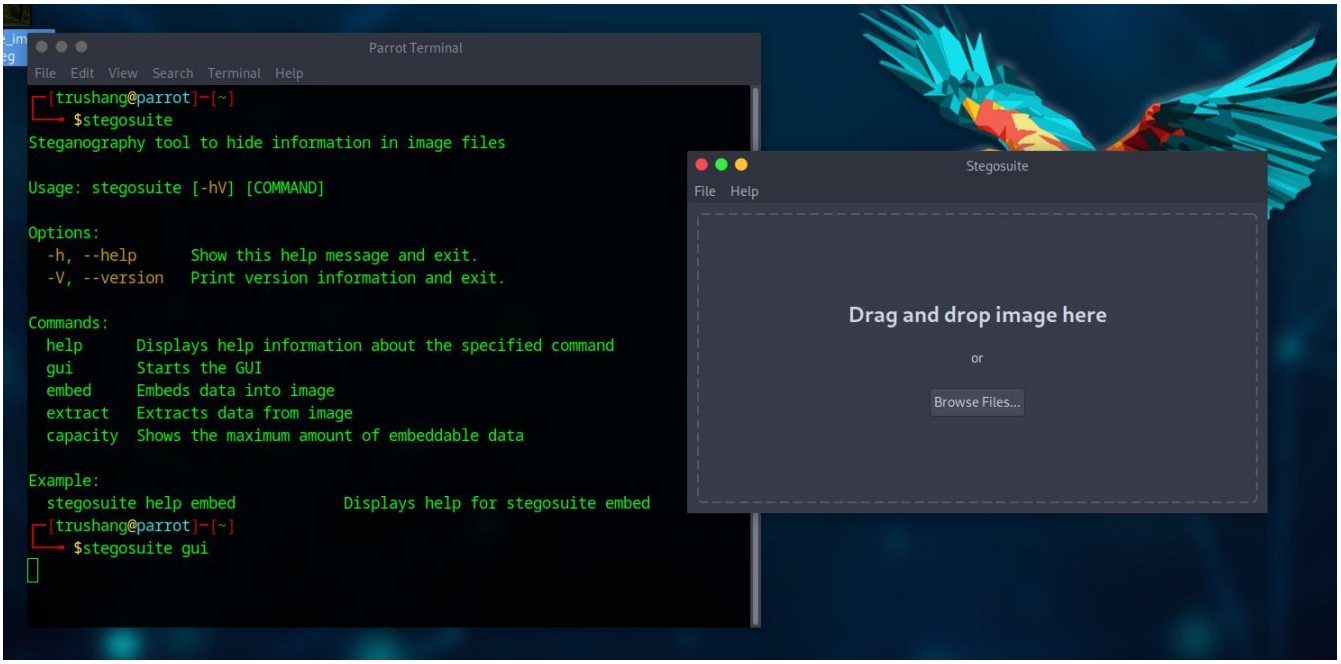


Figure 29:Open stego suite GUI

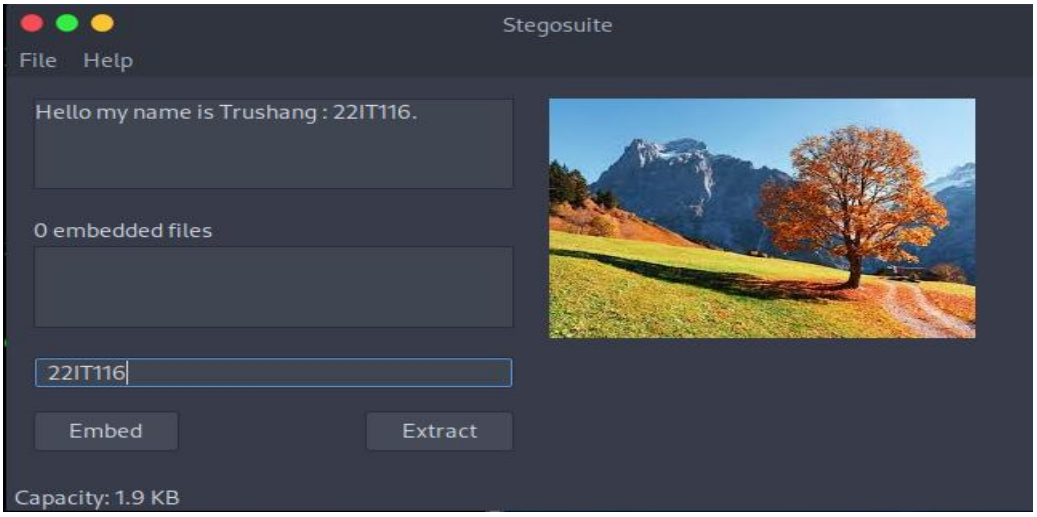


Figure 30:Drag file and set secret message and password

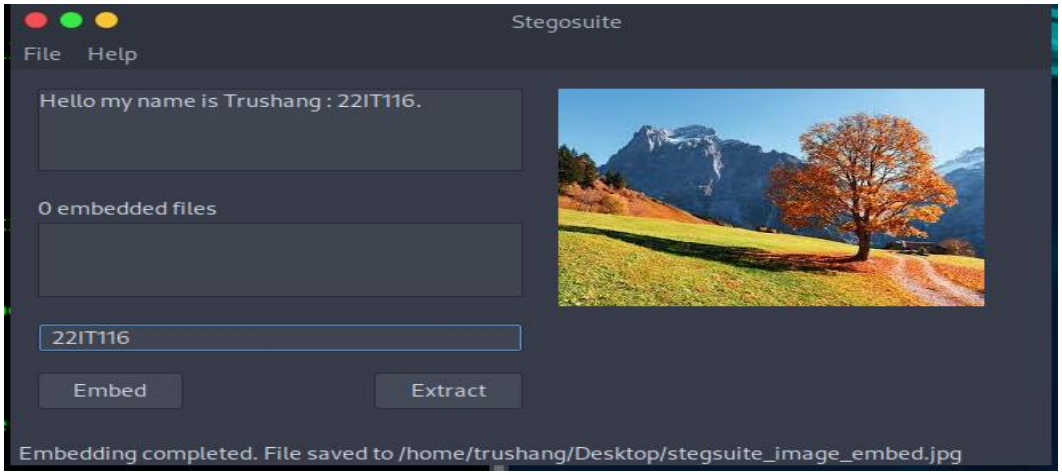


Figure 31:After clicking Embed we create new file with hiding message

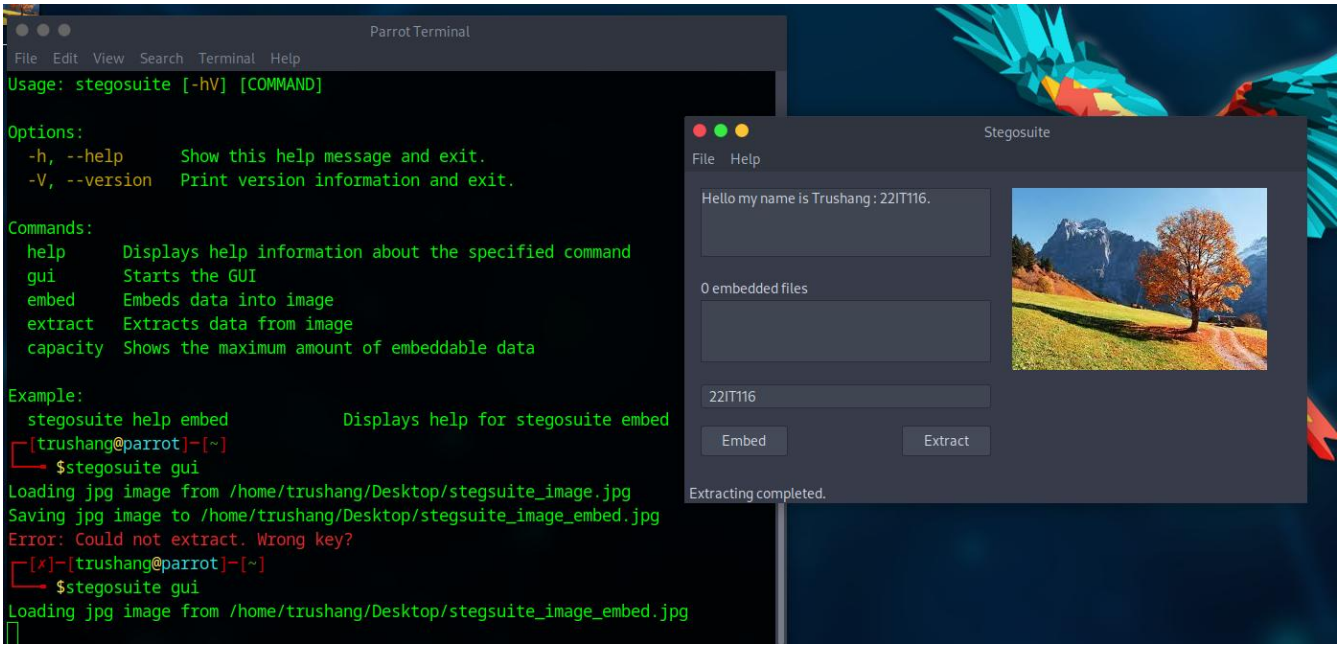


Figure 32: Now Extract the hide data from image

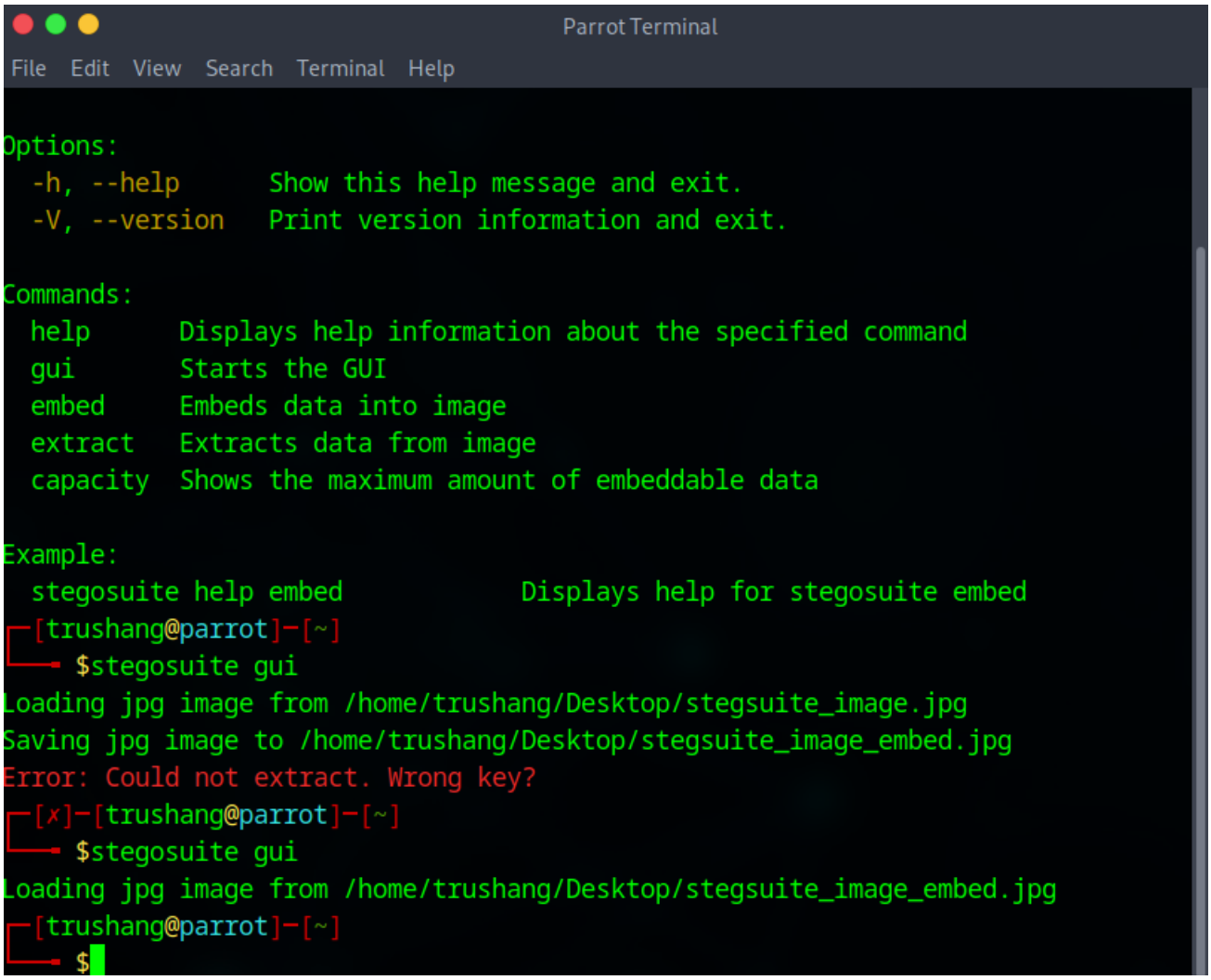


Figure 33:CLI for Steg suite

Figure 34:After installing the xiao Steganography

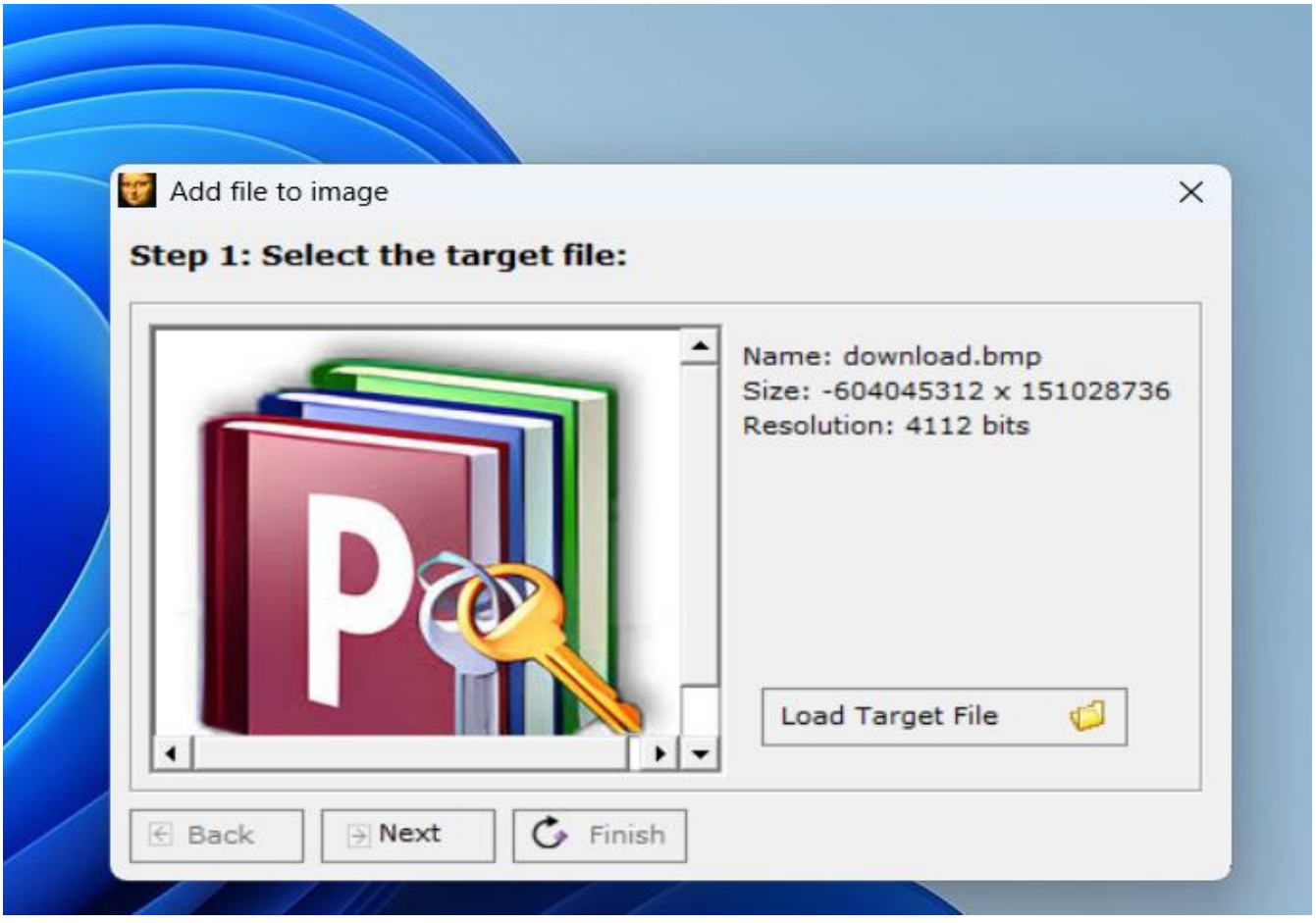


Figure 35:Add image into xios Steganography

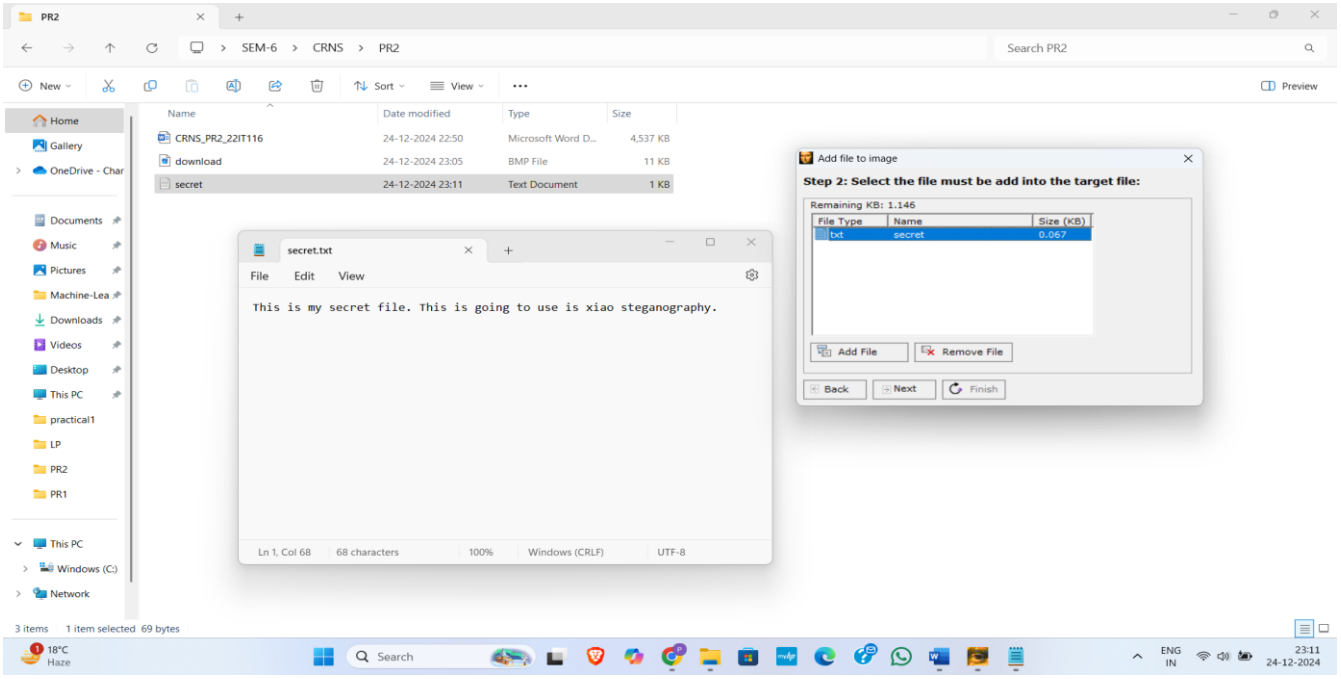


Figure 36: Add secret file to the xiao steganography

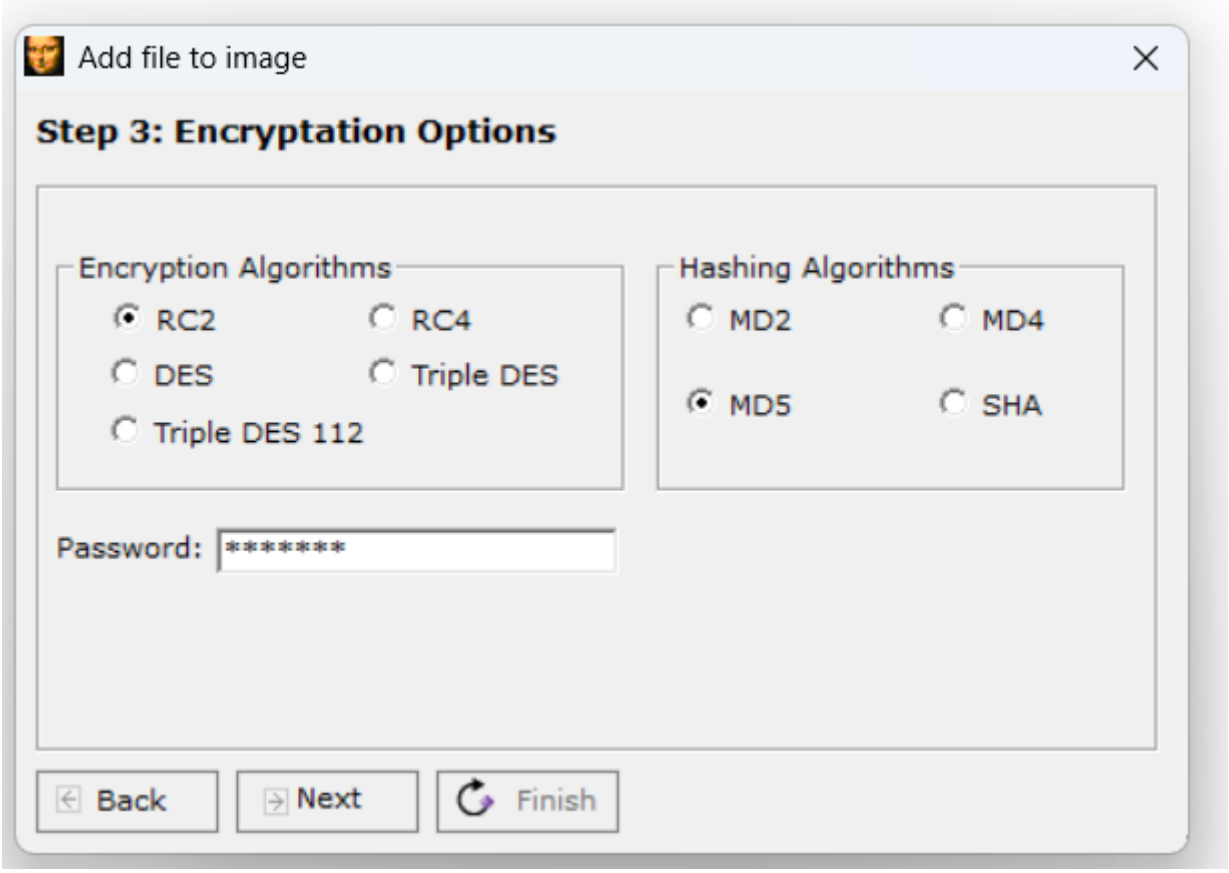


Figure 37:Choose encryption options and set password

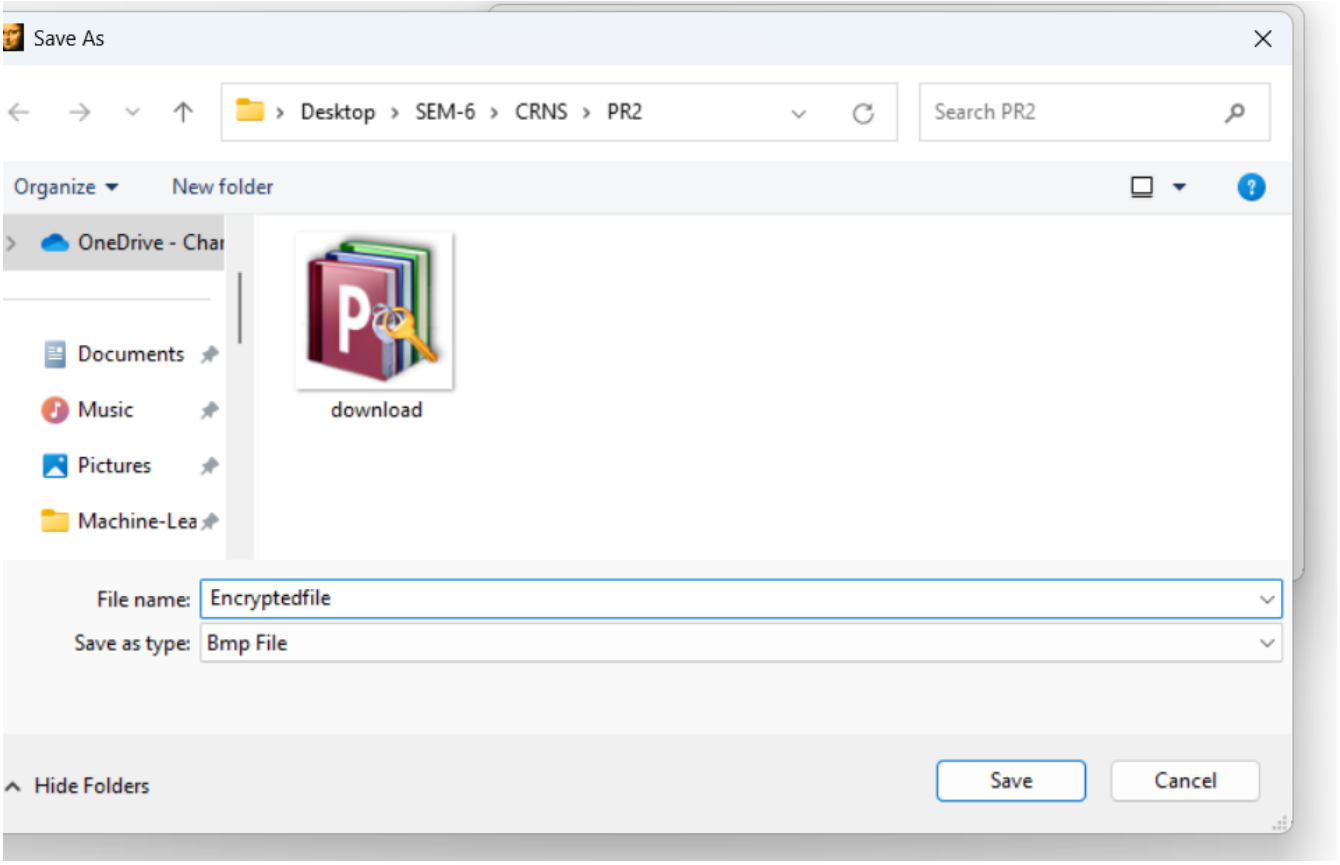


Figure 38:Save file in our local system

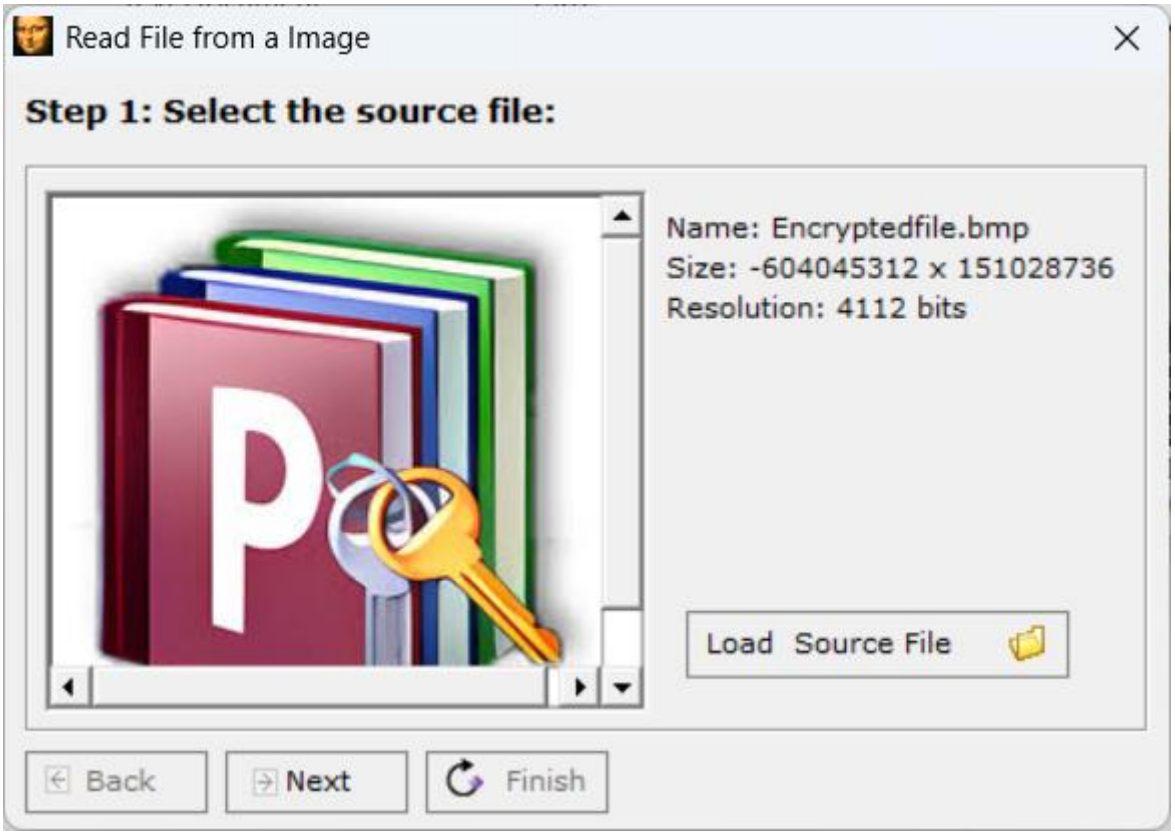


Figure 39: Select stegno file for decryptions

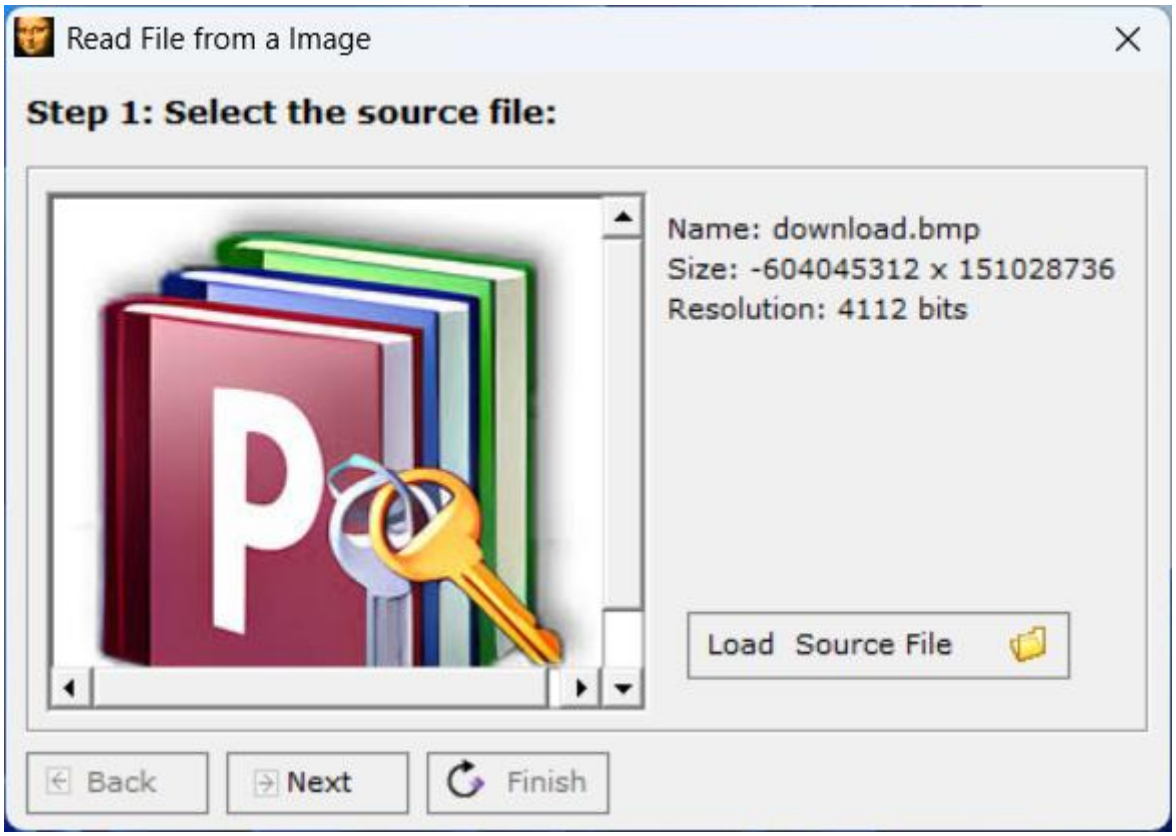


Figure 40:To extraction we use same file as we use for original image

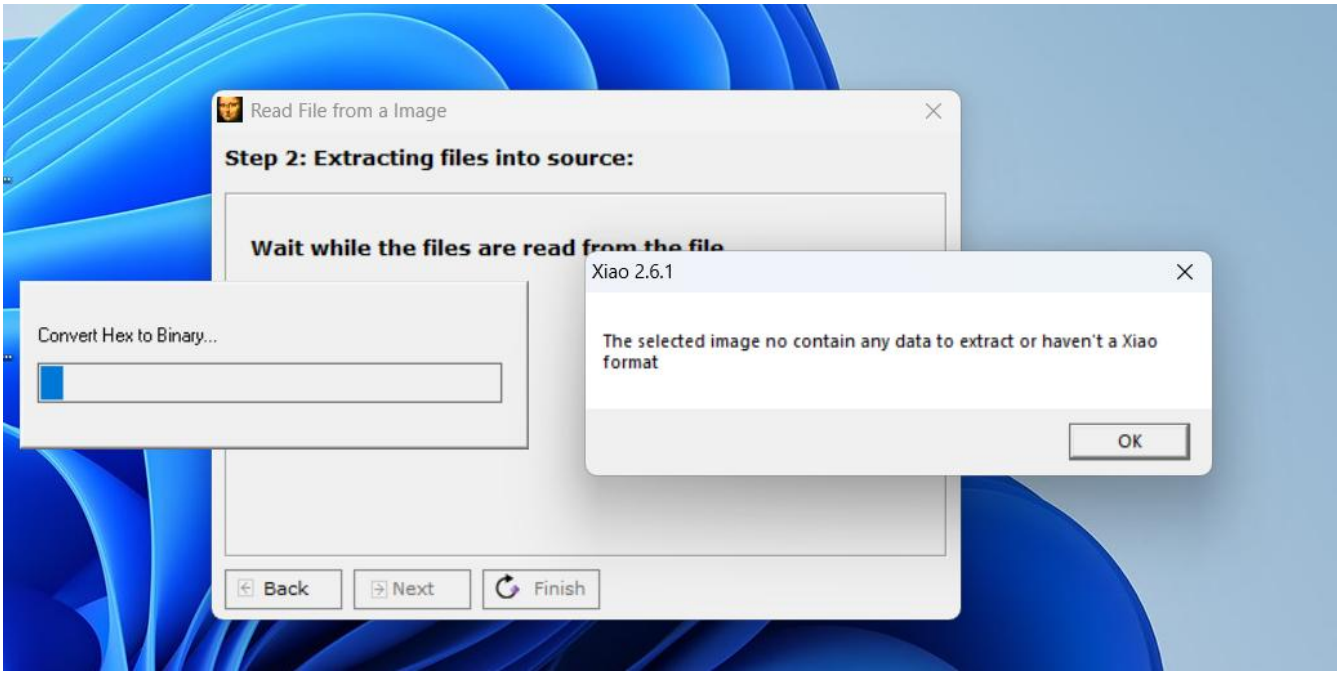


Figure 41:We uploaded original file without steganography so we get output like the selected image no contain any data to extract or haven't a Xiao

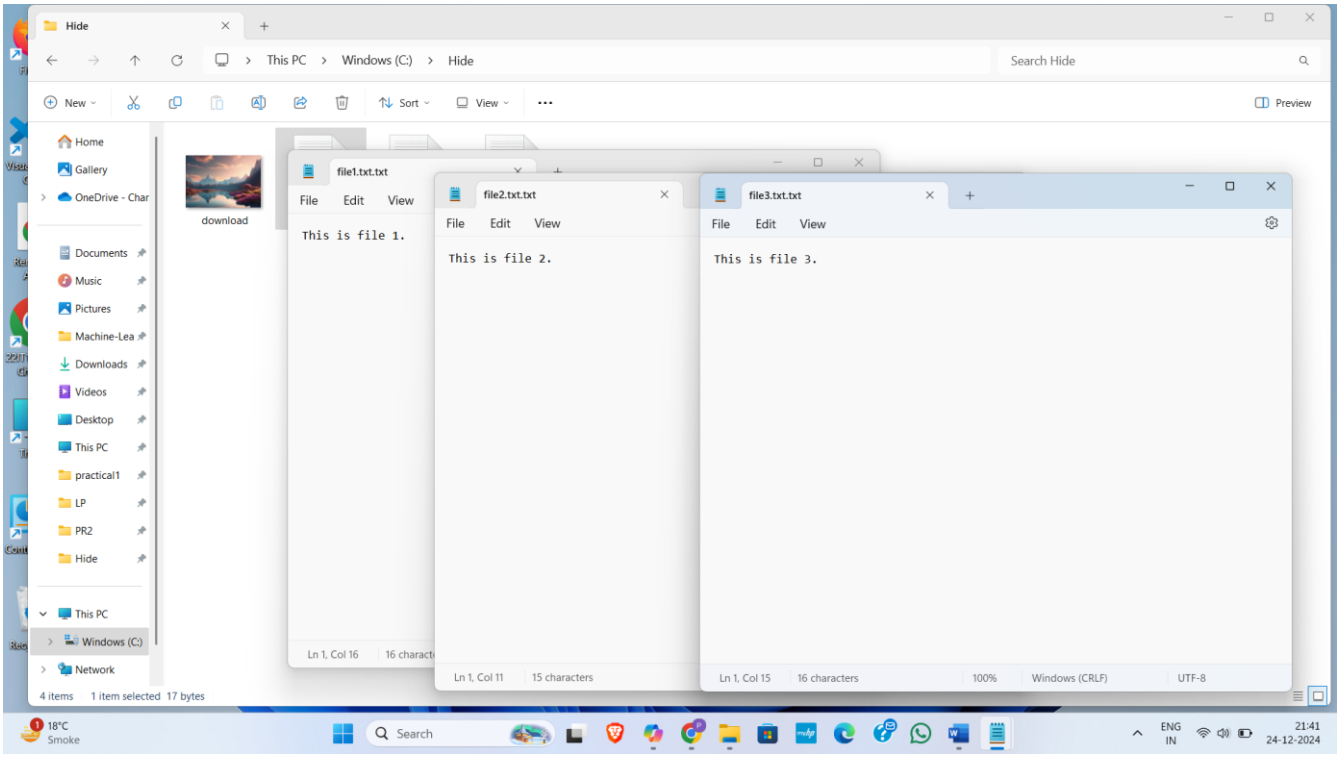


Figure 42: Created a folder in C drive name hide and we have an image name download and three file which we have to hide under image

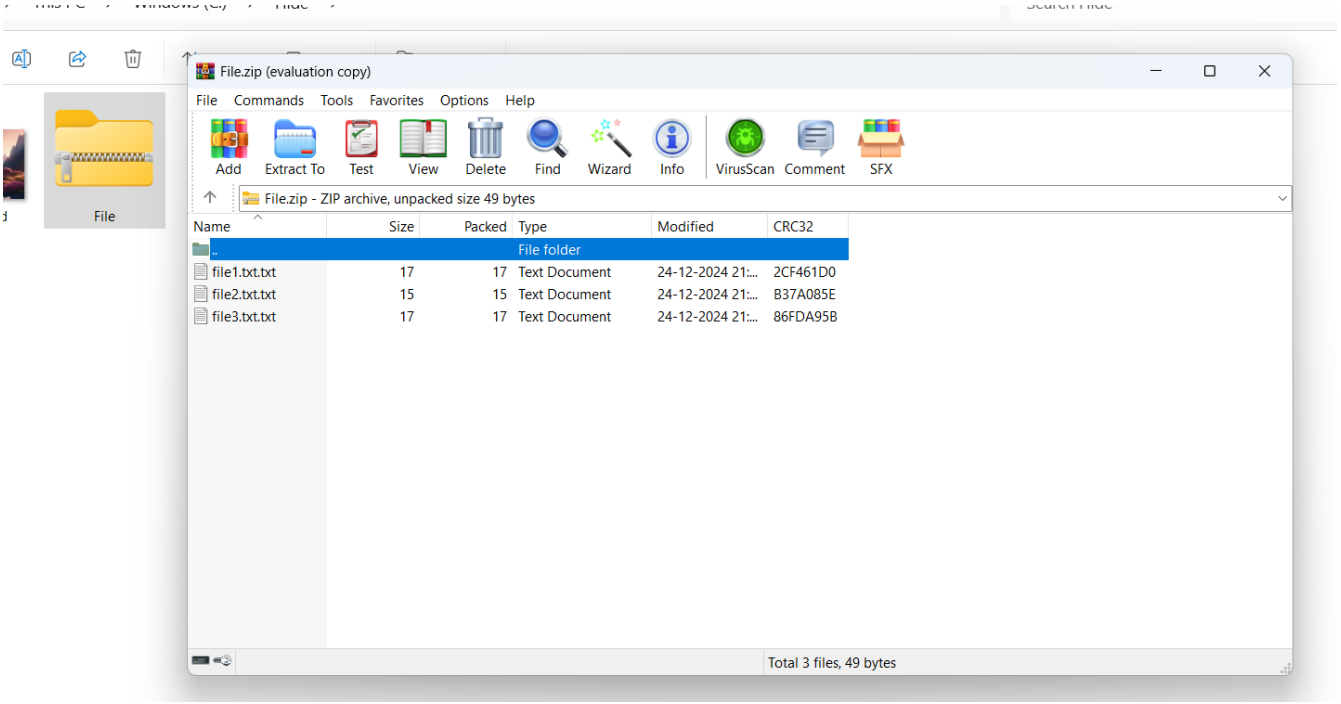


Figure 43: Create ZIP file which you want to hide in image

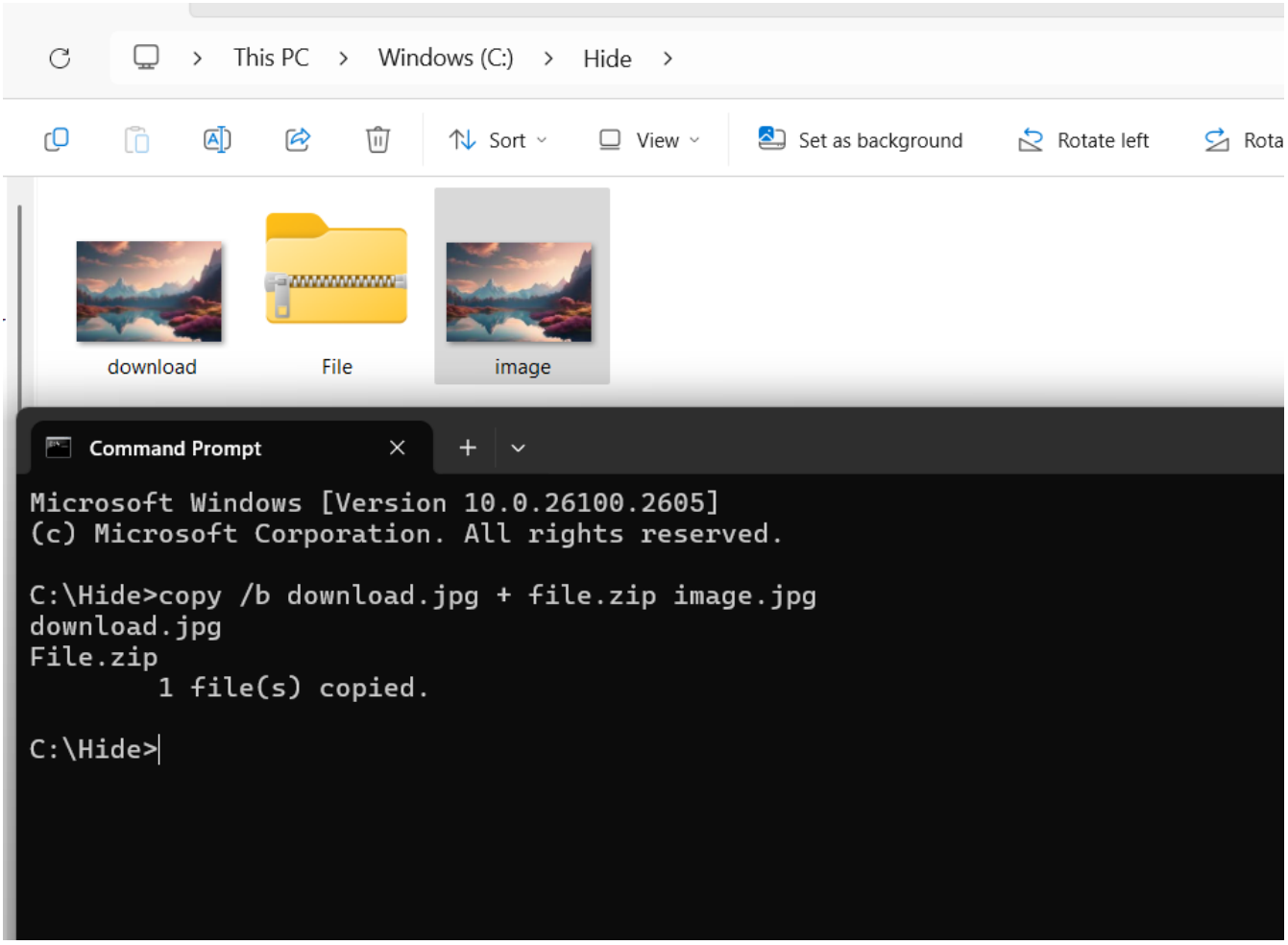


Figure 44:After performing command we create new image with file.zip

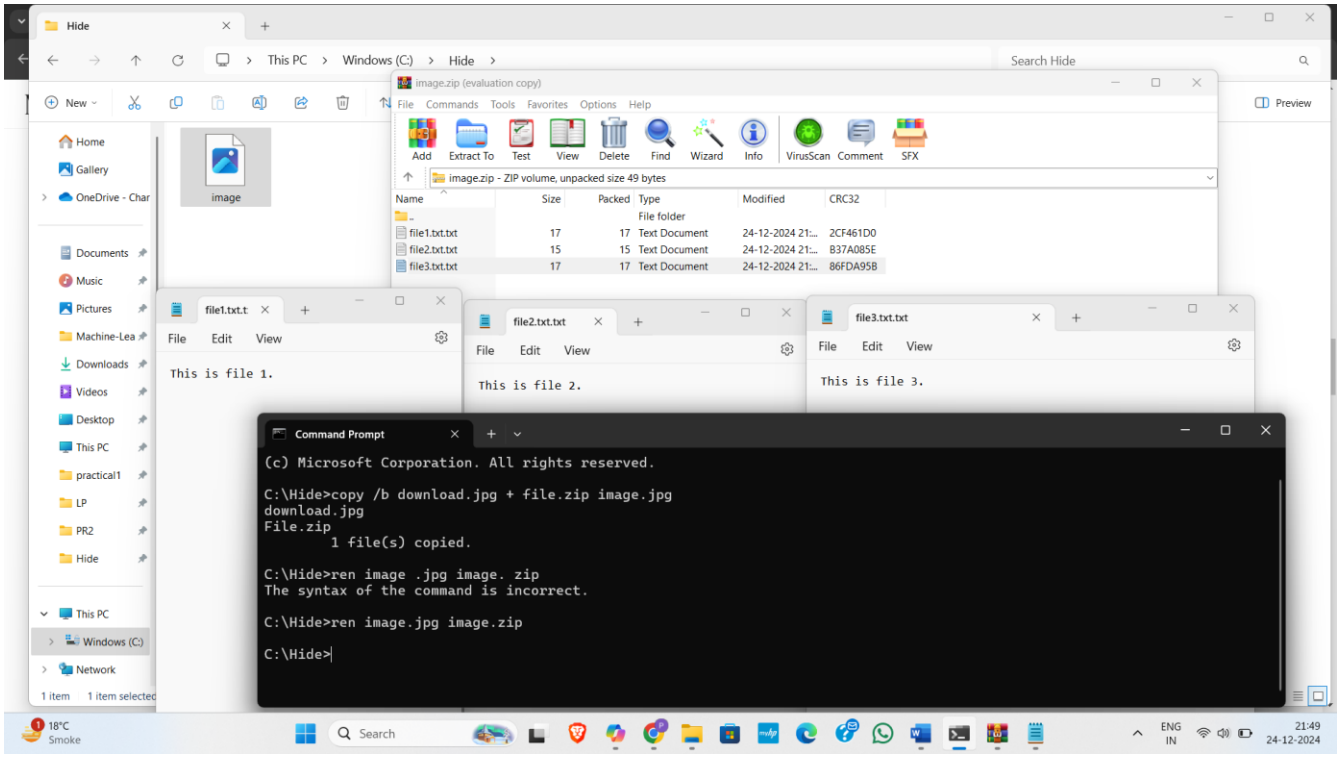


Figure 45:After change image type we can get our zip file with data

LATEST APPLICATIONS:

- E-commerce skimming
- SolarWinds
- Industrial enterprises
- Blockchain and Cryptocurrency
- Social Media and Messaging Apps

LEARNING OUTCOME: In this practical, we learn about Steganography and tools for steganography like steghide, Steg Suite, Xiao Steganography, using simple CLI and also write python code for steganography.

REFERENCES:

1. EC-council : https://www.eccouncil.org/Steganography
2. Kaspersky: https://www.kaspersky.com/Steganography
3. GeeksforGeeks: https://www.geeksforgeeks.org/Stegosuite-in-linux/
4. YouTube: https://www.youtube.com/watch?v=xepNoHgNj0w

PRACTICAL: 3

AIM:

Footprinting is the process of accumulating data regarding a specific network environment, usually to find ways to intrude into it. It can reveal system vulnerabilities and improve the ease with which they can be exploited. Footprinting is also known as reconnaissance. Study a practical approach to implementing Footprinting: Gathering Target Information using the OSINT Framework.

THEORY:

OSINT framework focused on gathering information from free tools or resources. The intention is to help people find free OSINT resources. Some of the sites included might require registration or offer more data for $$$, but you should be able to get at least a portion of the available information for no cost.

The five tools we will use in this exercise are:

1. WhatsMyName
2. Web Archive (Wayback Machine)
3. URLScan.io
4. DNSDumpster
5. OpenCorporates

1. WhatsMyName: Gathering DNS and Subdomain Information

Overview: WhatsMyName is an OSINT tool designed to perform DNS lookups and provide information about domain names. It helps identify various domain records associated with a target, including subdomains, IP addresses, and mail servers.

Steps:

- Go to the **WhatsMyName** website.
- Enter the domain name (e.g., "example.com") of the target you wish to investigate.
- The tool will return DNS records, which may include:
    - **A Records**: IP addresses associated with the domain.
    - **MX Records**: Mail servers linked to the domain.
    - **Subdomains**: Any additional domains or subdomains tied to the target.
    - **Nameservers**: Information about the authoritative nameservers for the domain.

This information helps us build a map of the target's domain infrastructure. It can reveal other systems or services the target might be running or any exposed services like mail servers or web applications.

2. Web Archive (Wayback Machine): Exploring Historical Web Data

Overview: The Wayback Machine, part of the Internet Archive, allows you to access historical versions of websites. It provides snapshots of websites taken at various points in time, enabling you to uncover potentially exposed information that has since been removed or changed.

Steps:

- Visit the **Wayback Machine** at archive.org/web.
- Enter the target's URL (e.g., "example.com") and press "Browse History."
- You will be shown a calendar of snapshots taken by the Wayback Machine, which you can explore by clicking on different dates.

Reviewing past versions of the target's website might reveal old content that could have been inadvertently exposed, such as outdated documents, forms, or login pages.

This can help you identify misconfigurations or legacy systems that have since been updated or removed from the live site.

3. URLScan.io: Scanning and Analyzing Web Pages

Overview: URLScan.io is a security analysis tool that scans websites and provides detailed reports about how a site behaves. It highlights external domains, embedded JavaScript, network activities, and potential security issues that might not be immediately visible from a standard web visit.

Steps:

- Go to **URLScan.io**.
- Enter the target URL (e.g., "example.com") in the search bar.
- URLScan.io will scan the website and return a report detailing:
    - External domains the site is connected to (e.g., third-party services, trackers).
    - Embedded JavaScript files and other resources.
    - Security headers and potential vulnerabilities.

This tool helps identify connections to external servers or suspicious domains that the target website might be communicating with.

The scan results can reveal potential security risks, such as unprotected scripts or malicious activity linked to the domain.

4. DNSDumpster: Comprehensive DNS and Network Mapping

Overview: DNSDumpster is a free tool that maps out DNS records for a given domain. It can provide insights into a target's DNS infrastructure, uncovering subdomains, associated IP addresses, and DNS record details that might help in understanding the target's network architecture.

Steps:

- Visit **DNSDumpster.com**.
- Enter the target domain name (e.g., "example.com") in the search bar and click "Search."
- The tool will display:
    - **Subdomains**: A list of subdomains linked to the target domain.
    - **DNS Records**: Information such as A, MX, and CNAME records.
    - **IP Addresses**: The range of IP addresses associated with the target's services.
    - **Geolocation Information**: The physical locations of the associated servers.

Use the subdomain and IP information to uncover additional services or infrastructure related to the target.

Mapping out DNS records helps you identify vulnerable or exposed systems, such as mail servers, that could be potential entry points.

5. OpenCorporates: Investigating Corporate Data

Overview: OpenCorporates is a database that aggregates public company registration data from jurisdictions around the world. It provides insights into the legal and business structure of organizations, which can be invaluable for foot printing a target company.

Steps:

- Go to **OpenCorporates** at opencorporates.com.
- Enter the target company name (e.g., "Example Corp.") or its business registration number.
- OpenCorporates will return:
    - **Corporate Registration Information**: Data about the company's registration, including jurisdiction and date of incorporation.
    - **Corporate Structure**: Information about subsidiaries, parent companies, and related entities.
    - **Directors and Key Personnel**: Publicly available data about the company's executives and board members.

This tool helps us to understand the organizational structure of the target, revealing potential subsidiaries, associated businesses, or directors who may have connections to other vulnerable assets.

The company's legal records could also provide insights into its history, legal standing, and potential financial issues, all of which can be valuable when assessing security risks.

CODE:

N/A

OUTPUT:

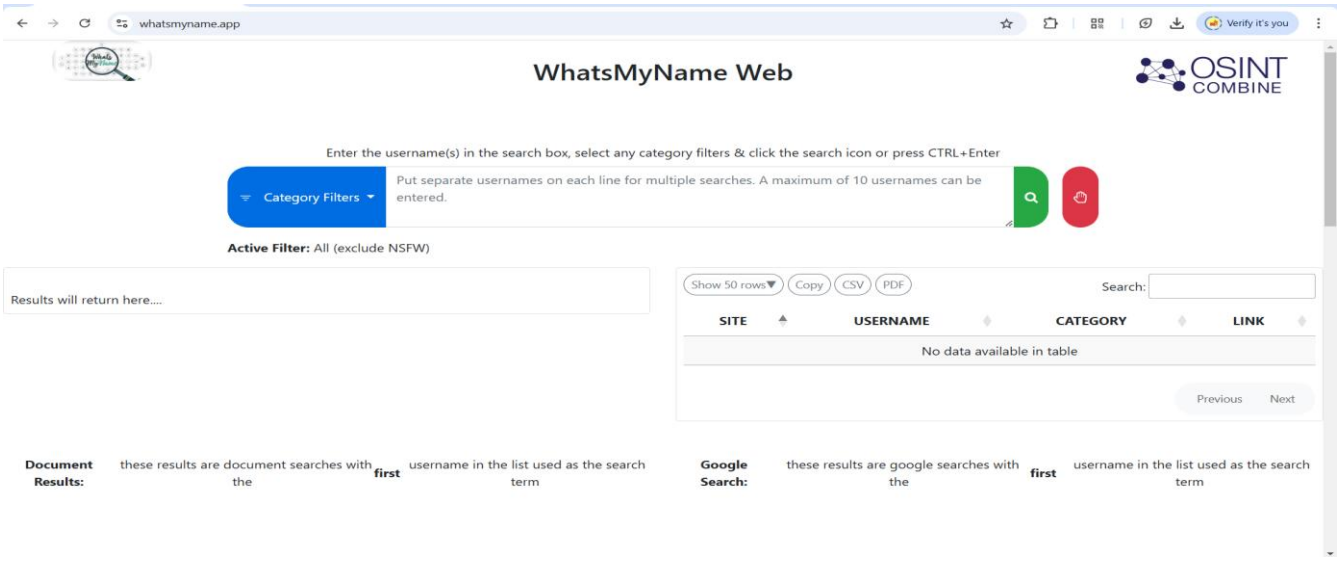


Figure 46:OSINT whatsMyName Web tool

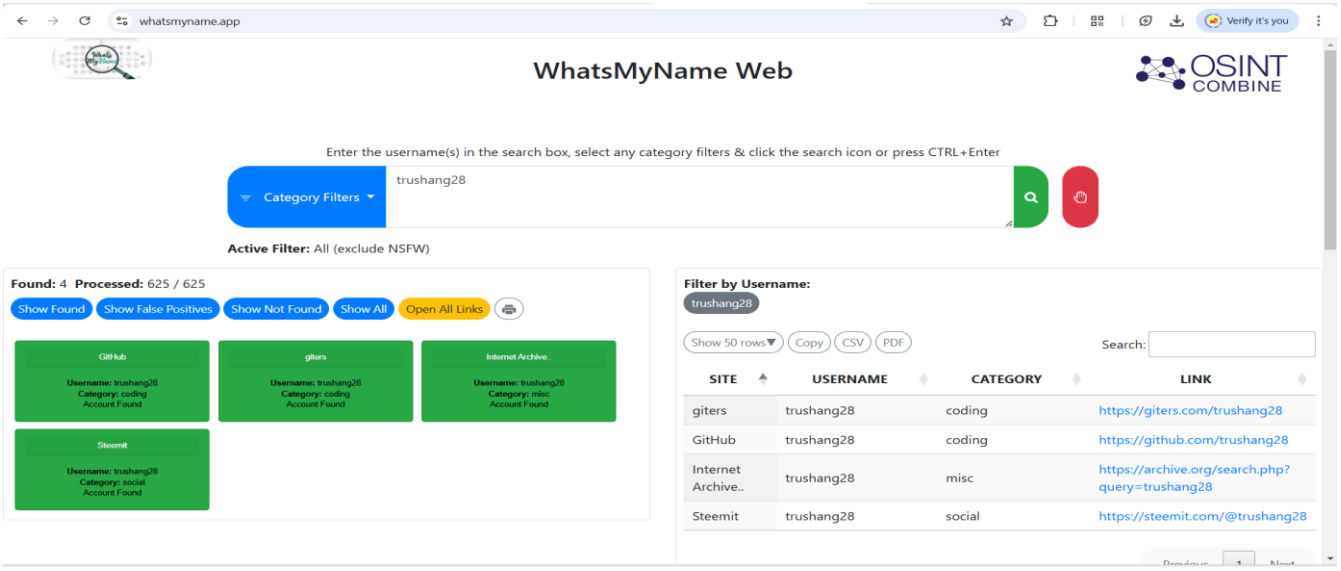


Figure 47:We provide our username it will check 625 site and found 4 as this username

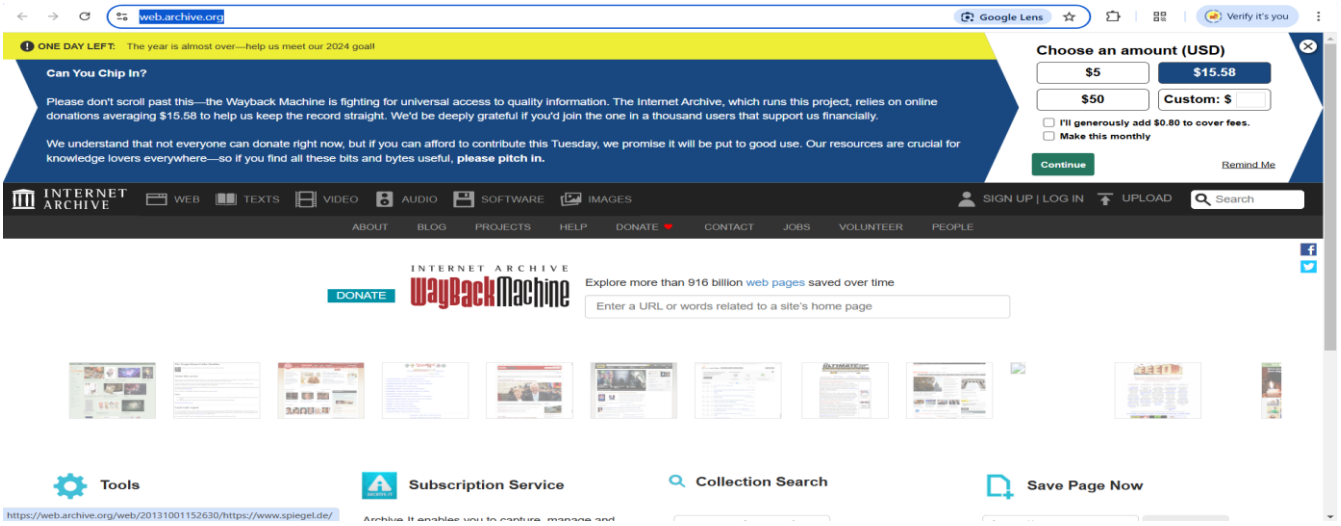


Figure 48:Second tools are web archive

Figure 49:This is change in website month wise

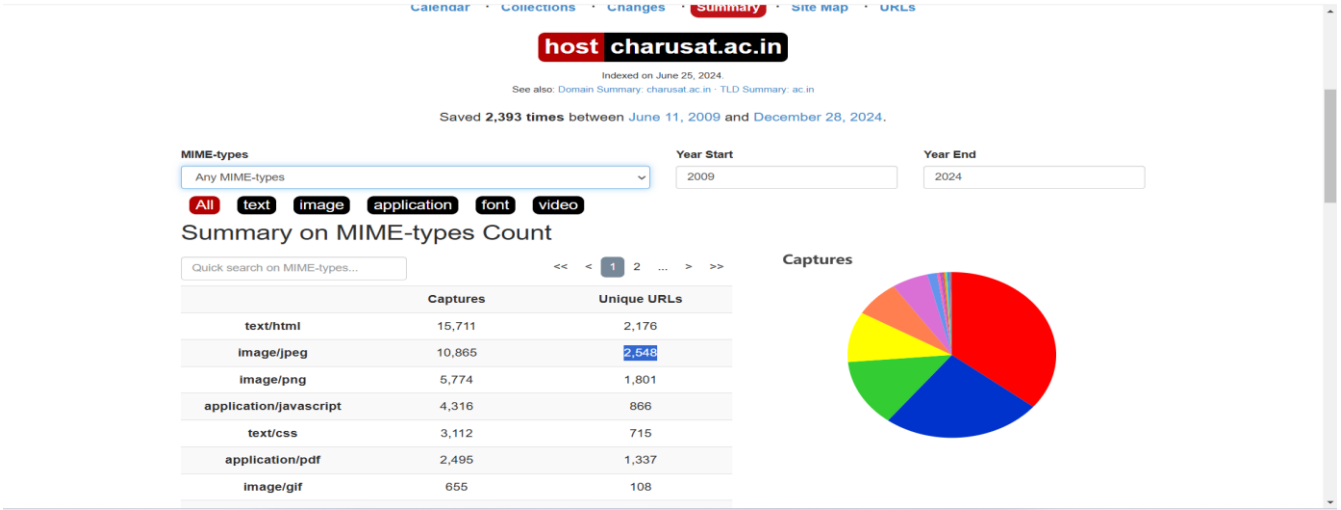


Figure 50:This is summary of charusat.ac.in

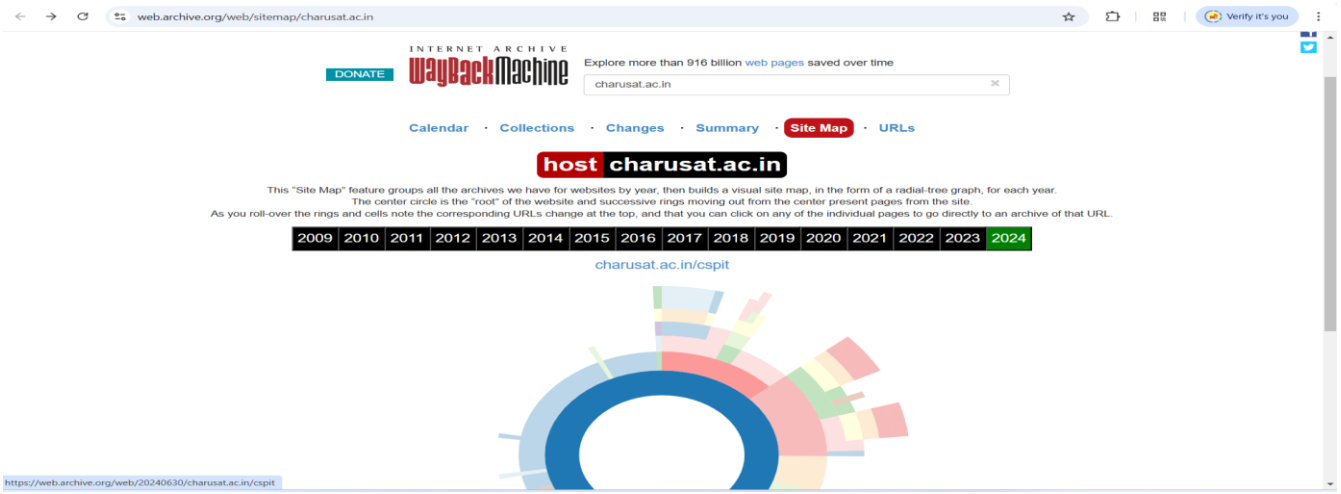
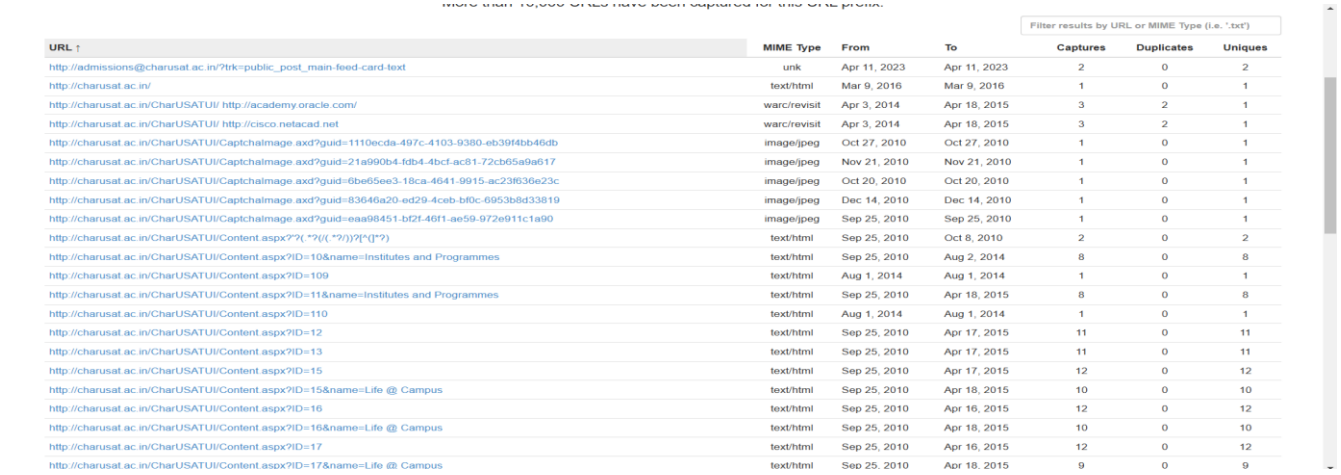


Figure 51:This is site map for charusat.ac.in of year 2024



Filter results by URL or MIME Type (i.e. ".txt")

URL ↑	MIME Type	From	To	Captures	Duplicates	Uniques
http://admissions@charusat.ac.in/?trk=public_post_main-feed-card-text	unk	Apr 11, 2023	Apr 11, 2023	2	0	2
http://charusat.ac.in/	text/html	Mar 9, 2016	Mar 9, 2016	1	0	1
http://charusat.ac.in/CharUSATUI/ http://academy.oracle.com/	warc/revisit	Apr 3, 2014	Apr 18, 2015	3	2	1
http://charusat.ac.in/CharUSATUI/ http://cisco.netacad.net	warc/revisit	Apr 3, 2014	Apr 18, 2015	3	2	1
http://charusat.ac.in/CharUSATUI/Captchalmage.axd?guid=1110ecda-497c-4103-9380-eb39f4bb46db	image/jpeg	Oct 27, 2010	Oct 27, 2010	1	0	1
http://charusat.ac.in/CharUSATUI/Captchalmage.axd?guid=21a990b4-fdb4-4bcf-ac81-72cb65a9a617	image/jpeg	Nov 21, 2010	Nov 21, 2010	1	0	1
http://charusat.ac.in/CharUSATUI/Captchalmage.axd?guid=6be65ee3-18ca-4641-9915-ac23f636e23c	image/jpeg	Oct 20, 2010	Oct 20, 2010	1	0	1
http://charusat.ac.in/CharUSATUI/Captchalmage.axd?guid=83646a20-ed29-4ceb-bf0c-6953b8d33819	image/jpeg	Dec 14, 2010	Dec 14, 2010	1	0	1
http://charusat.ac.in/CharUSATUI/Captchalmage.axd?guid=eaa98451-bf2f-46f1-ae59-972e911c1a90	image/jpeg	Sep 25, 2010	Sep 25, 2010	1	0	1
http://charusat.ac.in/CharUSATUI/Content.aspx?"?(.*?/(/(.*?/))?[^(]*?)	text/html	Sep 25, 2010	Oct 8, 2010	2	0	2
http://charusat.ac.in/CharUSATUI/Content.aspx?ID=10&name=Institutes and Programmes	text/html	Sep 25, 2010	Aug 2, 2014	8	0	8
http://charusat.ac.in/CharUSATUI/Content.aspx?ID=109	text/html	Aug 1, 2014	Aug 1, 2014	1	0	1
http://charusat.ac.in/CharUSATUI/Content.aspx?ID=11&name=Institutes and Programmes	text/html	Sep 25, 2010	Apr 18, 2015	8	0	8
http://charusat.ac.in/CharUSATUI/Content.aspx?ID=110	text/html	Aug 1, 2014	Aug 1, 2014	1	0	1
http://charusat.ac.in/CharUSATUI/Content.aspx?ID=12	text/html	Sep 25, 2010	Apr 17, 2015	11	0	11
http://charusat.ac.in/CharUSATUI/Content.aspx?ID=13	text/html	Sep 25, 2010	Apr 17, 2015	11	0	11
http://charusat.ac.in/CharUSATUI/Content.aspx?ID=15	text/html	Sep 25, 2010	Apr 17, 2015	12	0	12
http://charusat.ac.in/CharUSATUI/Content.aspx?ID=15&name=Life @ Campus	text/html	Sep 25, 2010	Apr 18, 2015	10	0	10
http://charusat.ac.in/CharUSATUI/Content.aspx?ID=16	text/html	Sep 25, 2010	Apr 16, 2015	12	0	12
http://charusat.ac.in/CharUSATUI/Content.aspx?ID=16&name=Life @ Campus	text/html	Sep 25, 2010	Apr 18, 2015	10	0	10
http://charusat.ac.in/CharUSATUI/Content.aspx?ID=17	text/html	Sep 25, 2010	Apr 16, 2015	12	0	12
http://charusat.ac.in/CharUSATUI/Content.aspx?ID=17&name=Life @ Campus	text/html	Sep 25, 2010	Apr 18, 2015	9	0	9

Figure 52:This are URL where charusat.ac.in is used as prefix

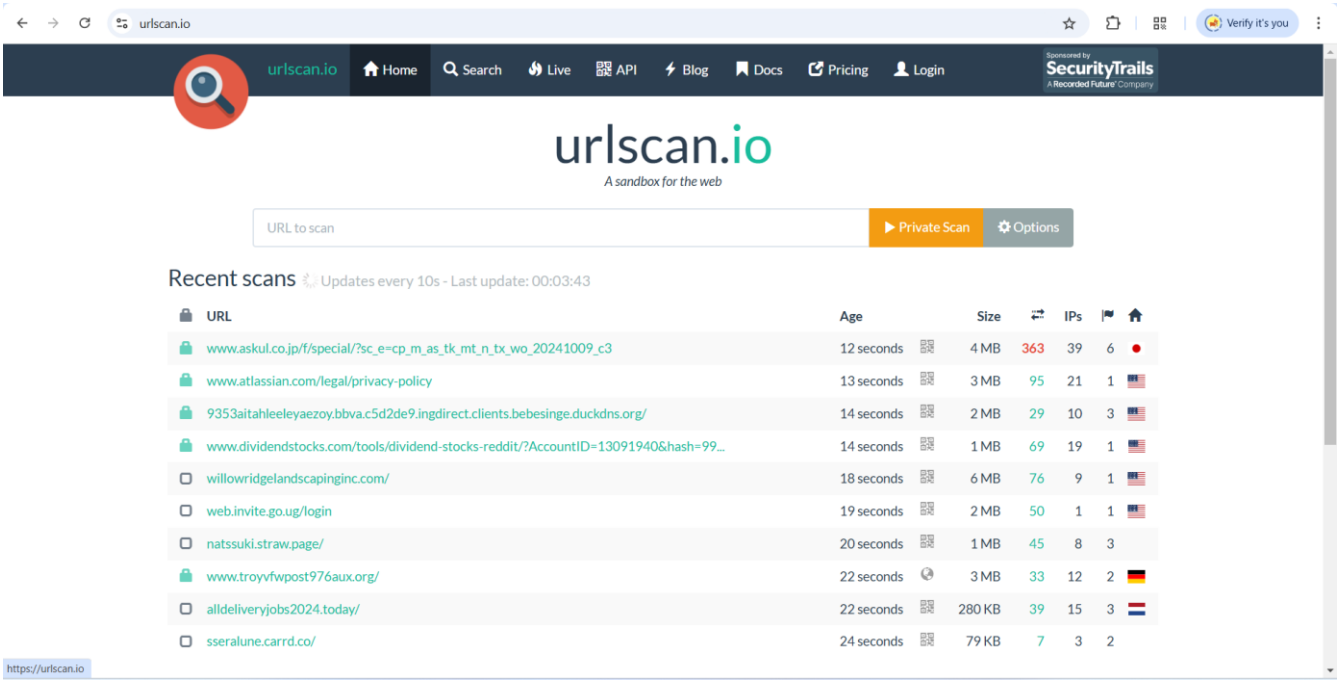


Figure 53:Third tool are urlscan.io

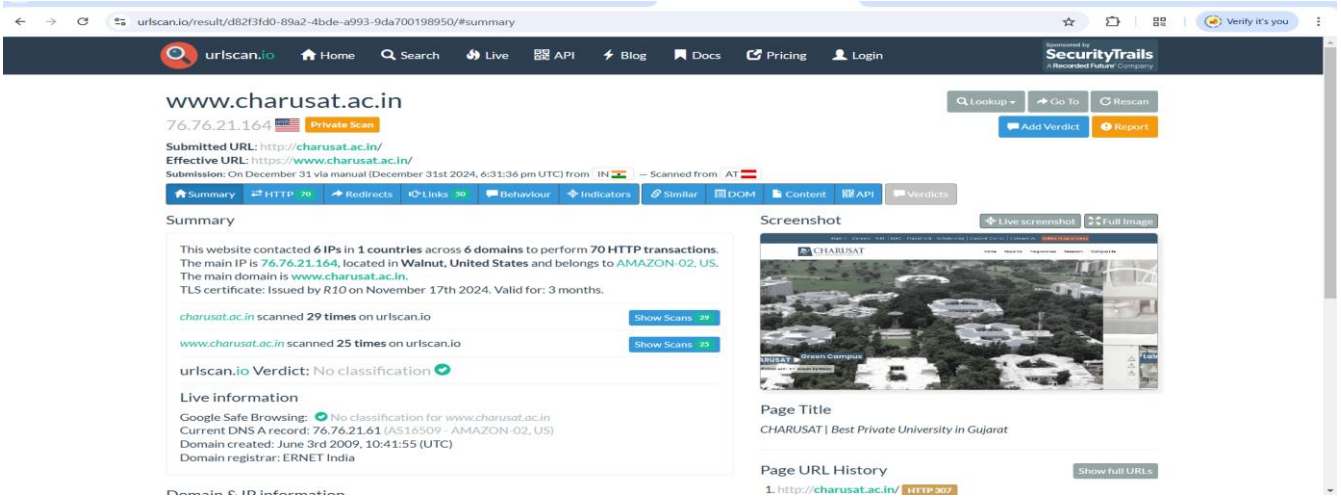


Figure 54:We scan charusat.ac.in privately and we scan charusat.ac.in 25 times

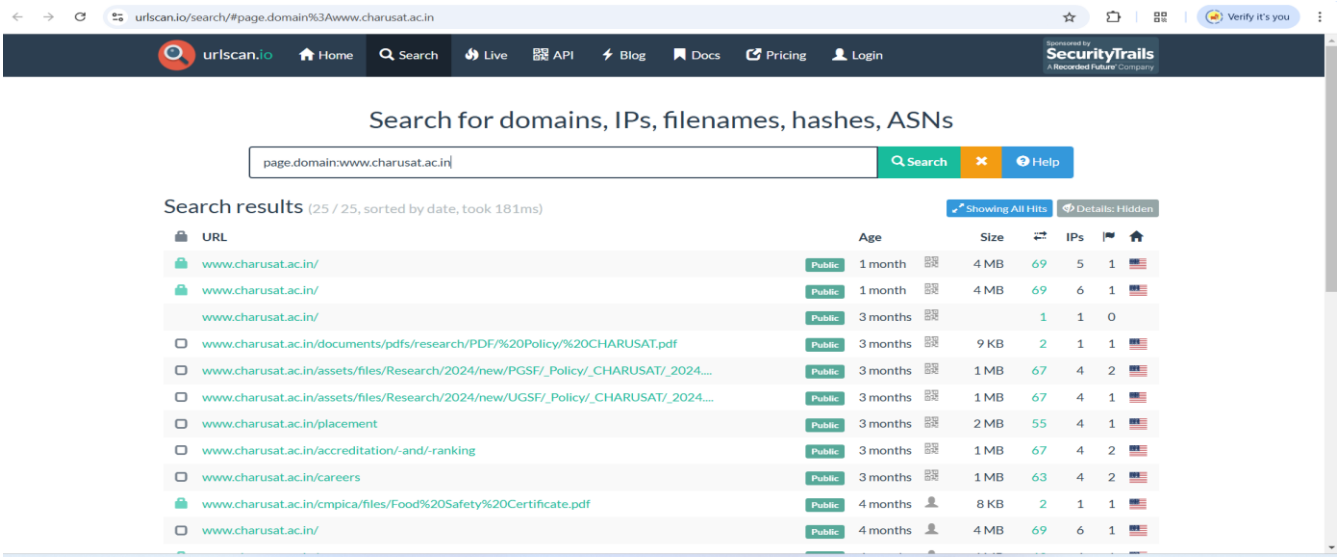


Figure 55:We can see which sub domain URL scan happened

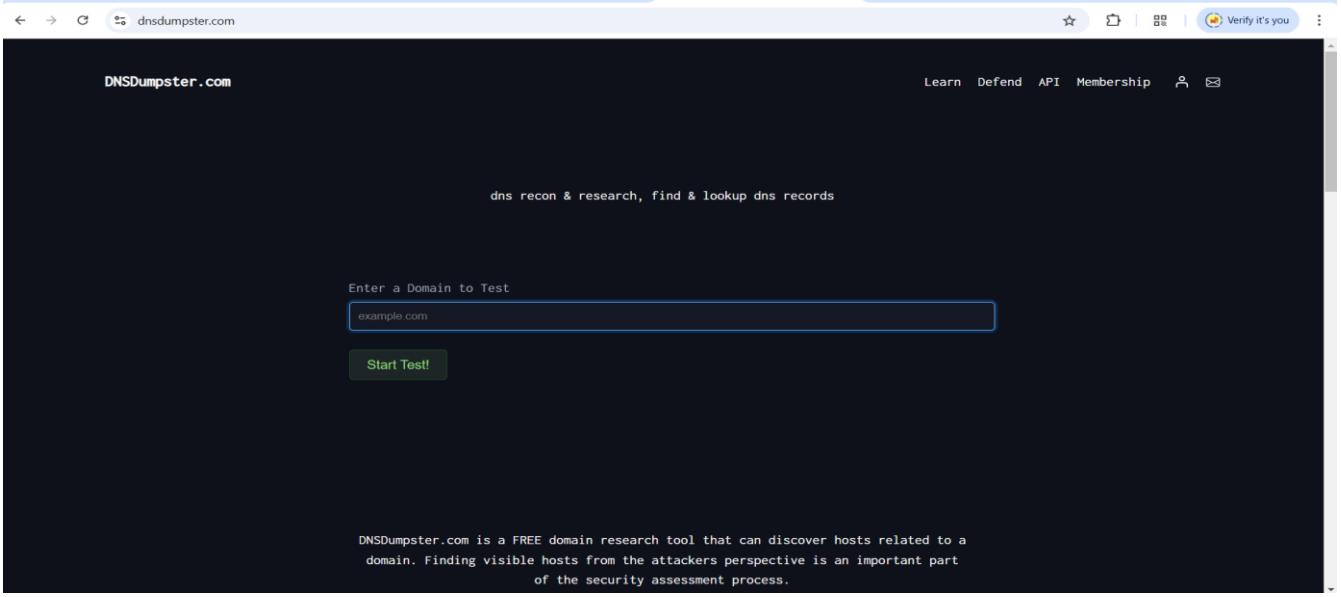


Figure 56:4th tool is DNSDumpster.com

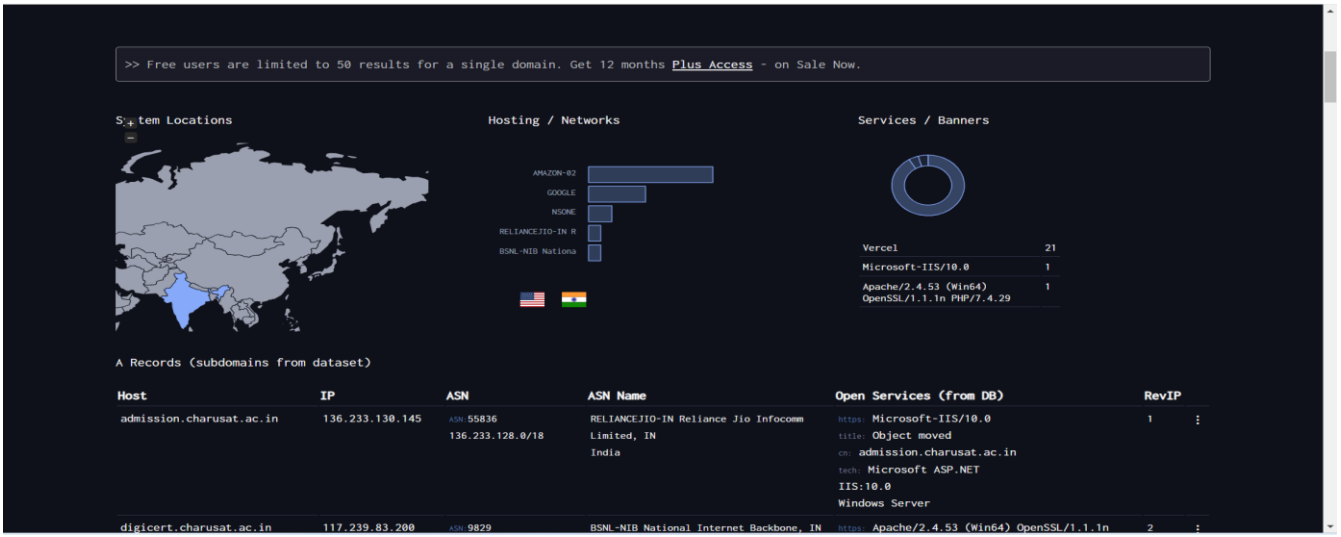


Figure 57:This is a result of dnsdumpster.com

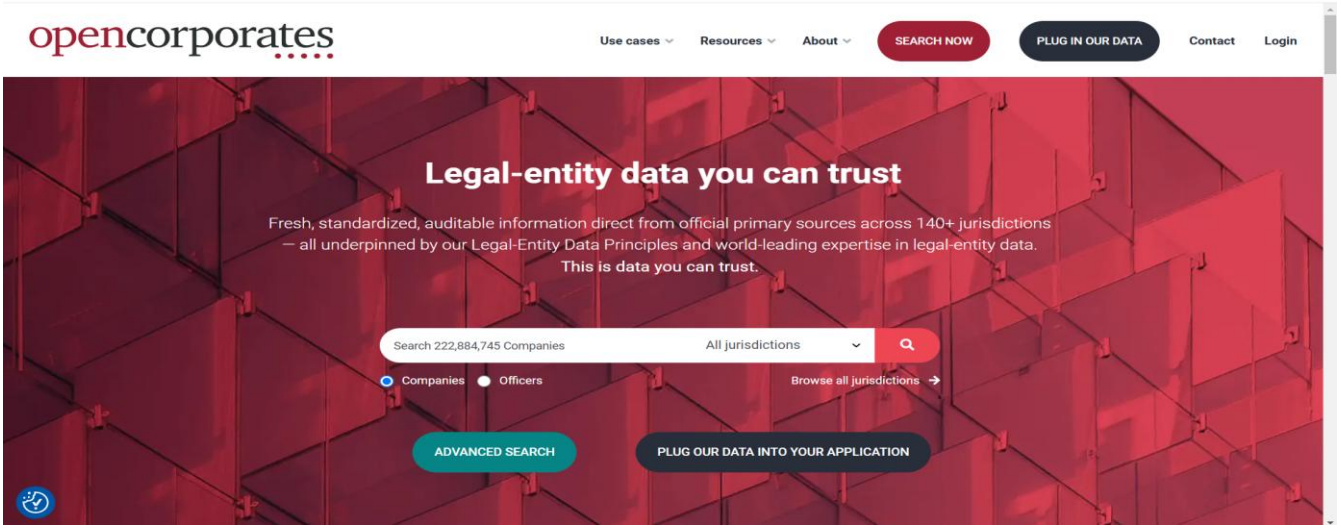


Figure 58:5th tool are opencorporates

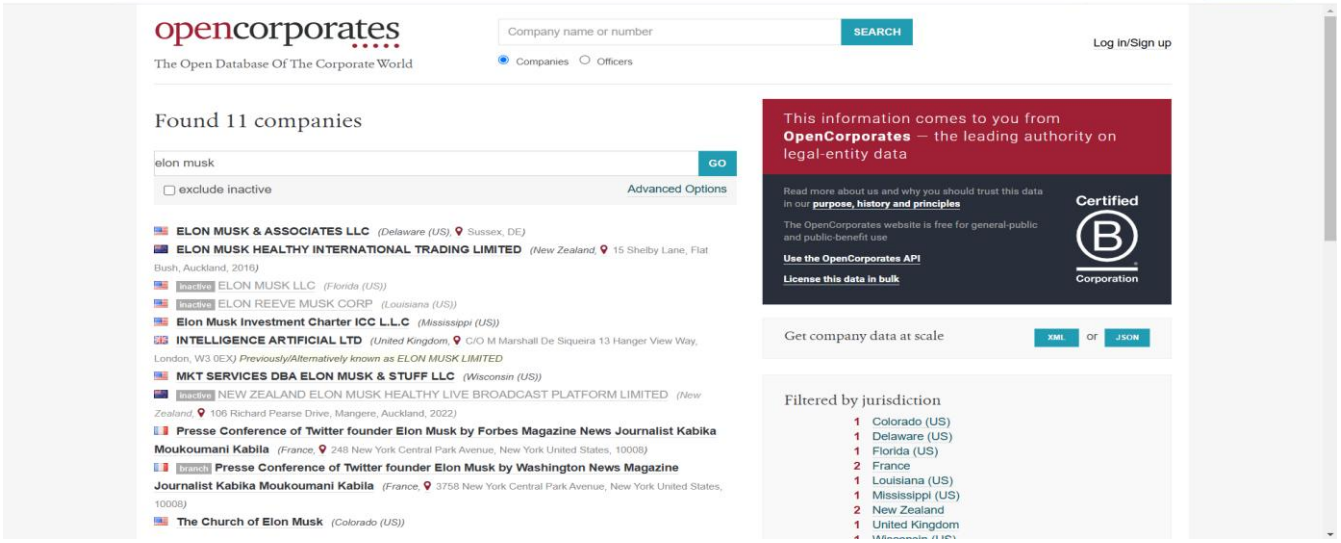


Figure 59:It will find 11 companies for Elon musk

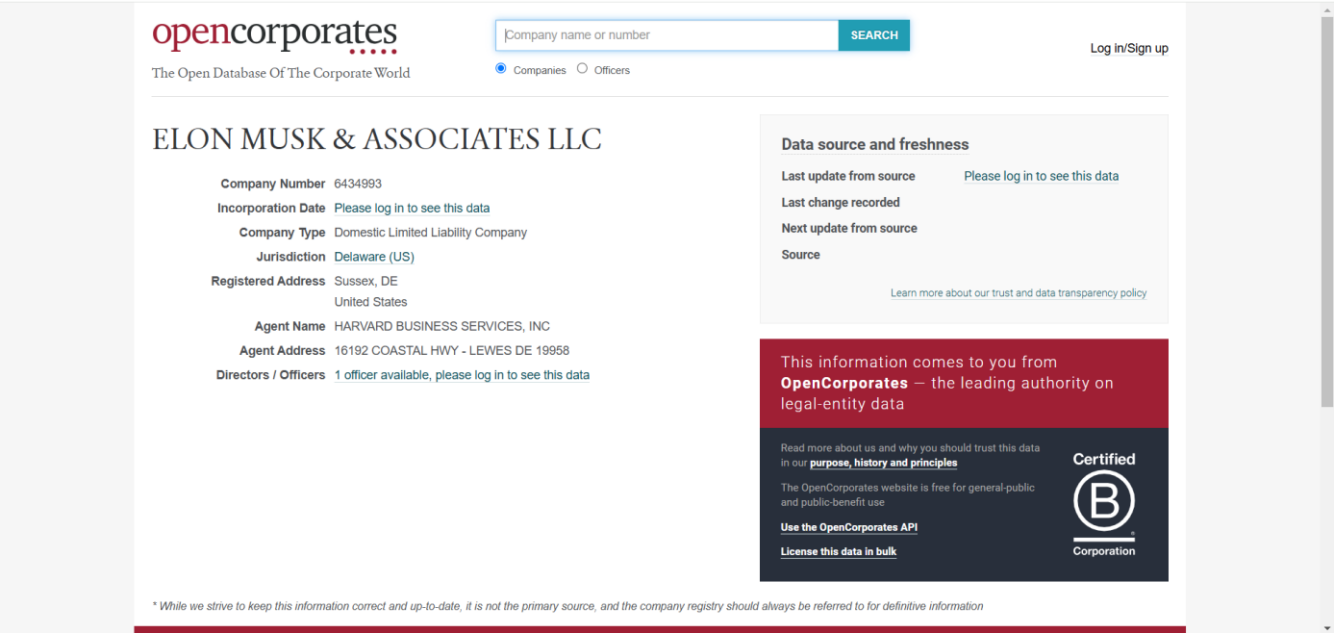


Figure 60:This is a details of Elon musk company

LATEST APPLICATIONS:

- Reconnaissance for Phishing Campaigns
- Cyber Threat Intelligence (CTI) Analysis
- Digital Forensics and Incident Response (DFIR)
- Website Threat Intelligence
- Vulnerability Discovery in Cloud Infrastructure
- Corporate Governance and Insider Threat Analysis
- Mergers and Acquisitions (M&A) Risk Assessment

LEARNING OUTCOME:

In this practical we learn about foot printing and reconnaissance using various OSINT frameworks tool

REFERENCES:

5. OSINT Framework: https://osintframework.com/
6. WhatsMyName: https://whatsmyname.app/
7. Urlscan.io: https://urlscan.io/
8. Wayback Machine: https://web.archive.org/
9. DNSDumpster: https://dnsdumpster.com/
10. opencorporates: https://opencorporates.com/

PRACTICAL: 4

AIM:

Port scanning is a method for determining open ports and services available on a network or a host. It involves connecting with TCP and UDP ports on the system once you find the IP addresses of a target network or host using the Footprinting technique. You have to map the network of this targeted organization. Nmap (Network Mapper) is a powerful, flexible, open- source, easy-to-use port scanning tool available for both Linux and Windows-based operating systems. Study practical approaches to implementing scanning and enumeration techniques using Nmap.

THEORY:

Nmap ("Network Mapper") is a free and open-source utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts. Nmap runs on all major computer operating systems, and official binary packages are available for Linux, Windows, and Mac OS X. In addition to the classic command-line Nmap executable, the Nmap suite includes an advanced GUI and results viewer (Zenmap), a flexible data transfer, redirection, and debugging tool (Ncat), a utility for comparing scan results (Ndiff), and a packet generation and response analysis tool (Nping).

Nmap is ...

Flexible: Supports dozens of advanced techniques for mapping out networks filled with IP filters, firewalls, routers, and other obstacles. This includes many port scanning mechanisms (both TCP & UDP), OS detection, version detection, ping sweeps, and more. See the documentation page.

Powerful: Nmap has been used to scan huge networks of literally hundreds of thousands of machines.

Portable: Most operating systems are supported, including Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS X, HP-UX, NetBSD, Sun OS, Amiga, and more.

Easy: While Nmap offers a rich set of advanced features for power users, we can start out as simply as "nmap -v -A targethost". Both traditional command line and graphical (GUI) versions are available to suit your preference. Binaries are available for those who do not wish to compile Nmap from source.

Free: The primary goals of the Nmap Project are to help make the Internet a little more secure and to provide administrators/auditors/hackers with an advanced tool for exploring their networks. Nmap is available for free download, and also comes with full source code that you may modify and redistribute under the terms of the license.

Well Documented: Significant effort has been put into comprehensive and up-to-date man pages, whitepapers, tutorials, and even a whole book! Find them in multiple languages here.

CODE:

- nmap -v
- nmap localhost
- nmap 172.16.3.129 --disable-arp-ping
- nmap -v 172.16.3.129 --disable-arp-ping
- nmap -sA 172.16.3.129
- nmap -Pn 172.16.3.129
- nmap -F 172.16.3.129
- nmap –iflist
- nmap -o 172.16.3.129
- nmap -A 172.16.3.129
- nmap 172.16.3.1-255
- nmap charusat.edu.in
- nmap -sS 172.16.3.129
- nmap 172.16.3.129/24 --disable-arp-ping

OUTPUT:

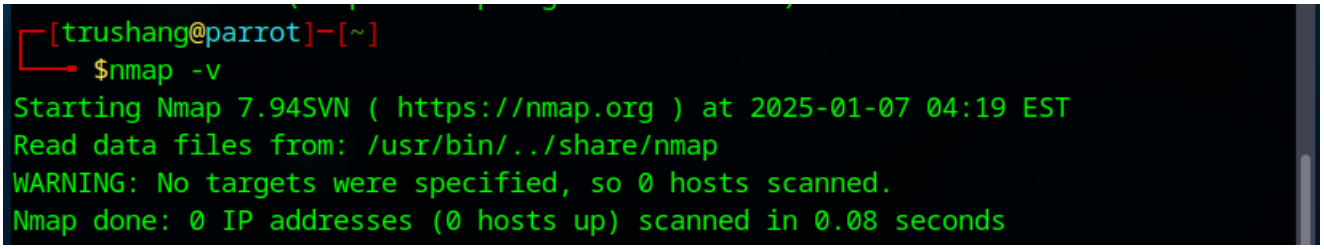


Figure 61:Check the version of Nmap

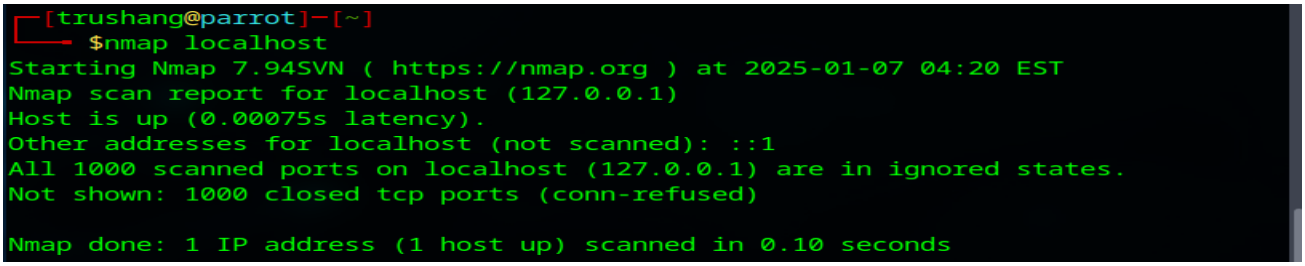


Figure 62:Scan using Nmap

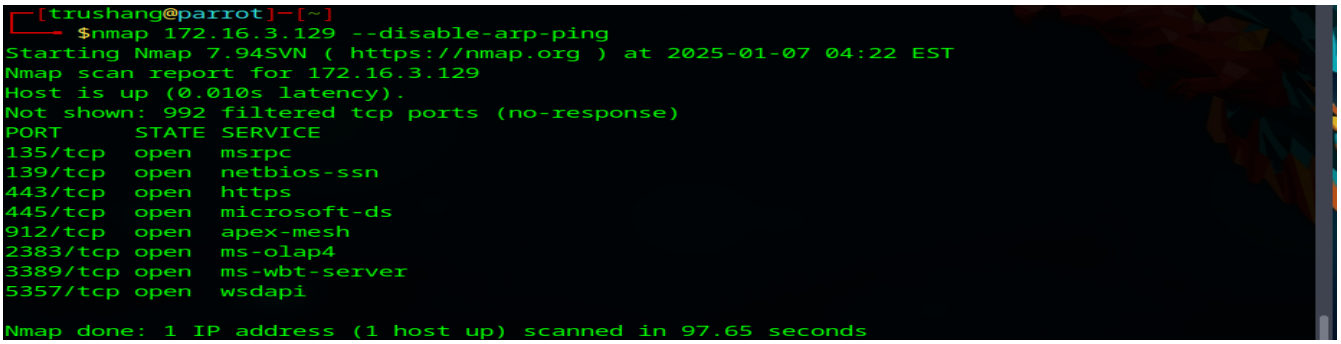


Figure 63:Scan using ip address

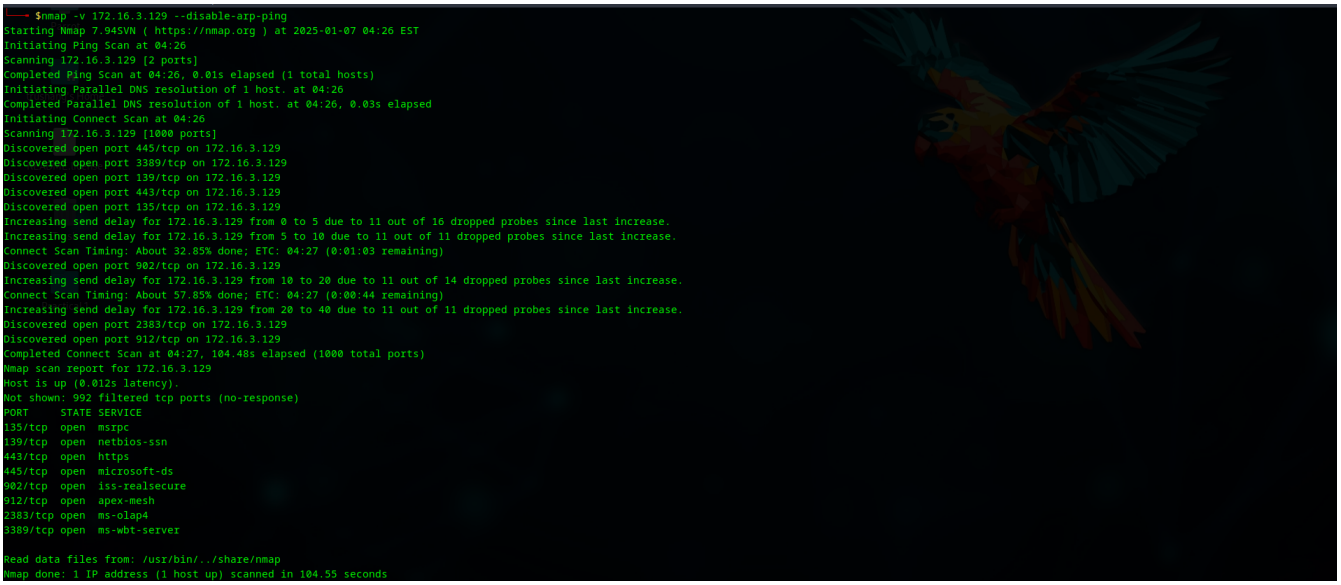


Figure 64:Get a more detailed output of the scan, such as status updates on scanning the host and ports

```
┌─[root@parrot]─[/home/trushang]
└──╼ #nmap -sA 172.16.3.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 04:39 EST
Nmap scan report for 172.16.3.129
Host is up (0.0023s latency).
All 1000 scanned ports on 172.16.3.129 are in ignored states.
Not shown: 1000 unfiltered tcp ports (reset)

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

Figure 65:TCP ACK port scan

```
┌─[trushang@parrot]─[~]
└──╼ $nmap -Pn 172.16.3.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 04:48 EST
Nmap scan report for 172.16.3.129
Host is up (0.0085s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
902/tcp  open  iss-realsecure
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 96.81 seconds
```

Figure 66:Scan a host to detect firewall

```
┌──[trushang@parrot]─[~]
└─$nmap -F 172.16.3.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 04:54 EST
Nmap scan report for 172.16.3.129
Host is up (0.012s latency).
Not shown: 96 filtered tcp ports (no-response)
PORT     STATE SERVICE
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
5357/tcp open  wsdapi

Nmap done: 1 IP address (1 host up) scanned in 6.12 seconds
```

Figure 67:Perform fast scan

```
┌──[trushang@parrot]─[~]
└─$nmap --iflist
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 05:02 EST
************************INTERFACES************************
DEV    (SHORT)  IP/MASK                    TYPE     UP MTU   MAC
lo     (lo)     127.0.0.1/8                loopback up 65536
lo     (lo)     ::1/128                    loopback up 65536
enp0s3 (enp0s3) 10.0.2.15/24               ethernet up 1500  08:00:27:72:FD:6F
enp0s3 (enp0s3) fe80::8474:225:82b7:85eb/64 ethernet up 1500  08:00:27:72:FD:6F

*************************ROUTES*************************
DST/MASK                    DEV    METRIC GATEWAY
10.0.2.0/24                 enp0s3 100
0.0.0.0/0                   enp0s3 100    10.0.2.2
::1/128                     lo     0
fe80::8474:225:82b7:85eb/128 enp0s3 0
fe80::/64                   enp0s3 1024
ff00::/8                    enp0s3 256
```

Figure 68:Print Host interface and route

```
┌──[root@parrot]─[/home/trushang]
└─#nmap -O 172.16.3.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 05:06 EST
Nmap scan report for 172.16.3.129
Host is up (0.0069s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed por
t
Device type: phone|general purpose|switch
Running (JUST GUESSING): Nokia Symbian OS (87%), Linux 1.0.X (86%), Cisco embedded (85%)
OS CPE: cpe:/o:nokia:symbian_os cpe:/o:linux:linux_kernel:1.0.9 cpe:/h:cisco:catalyst_1900
Aggressive OS guesses: Nokia 3600i mobile phone (87%), Linux 1.0.9 (86%), Cisco Catalyst 1900 switch
 (85%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.92 seconds
┌──[root@parrot]─[/home/trushang]
```

Figure 69:Remote OS detection using TCP/IP stack fingerprinting

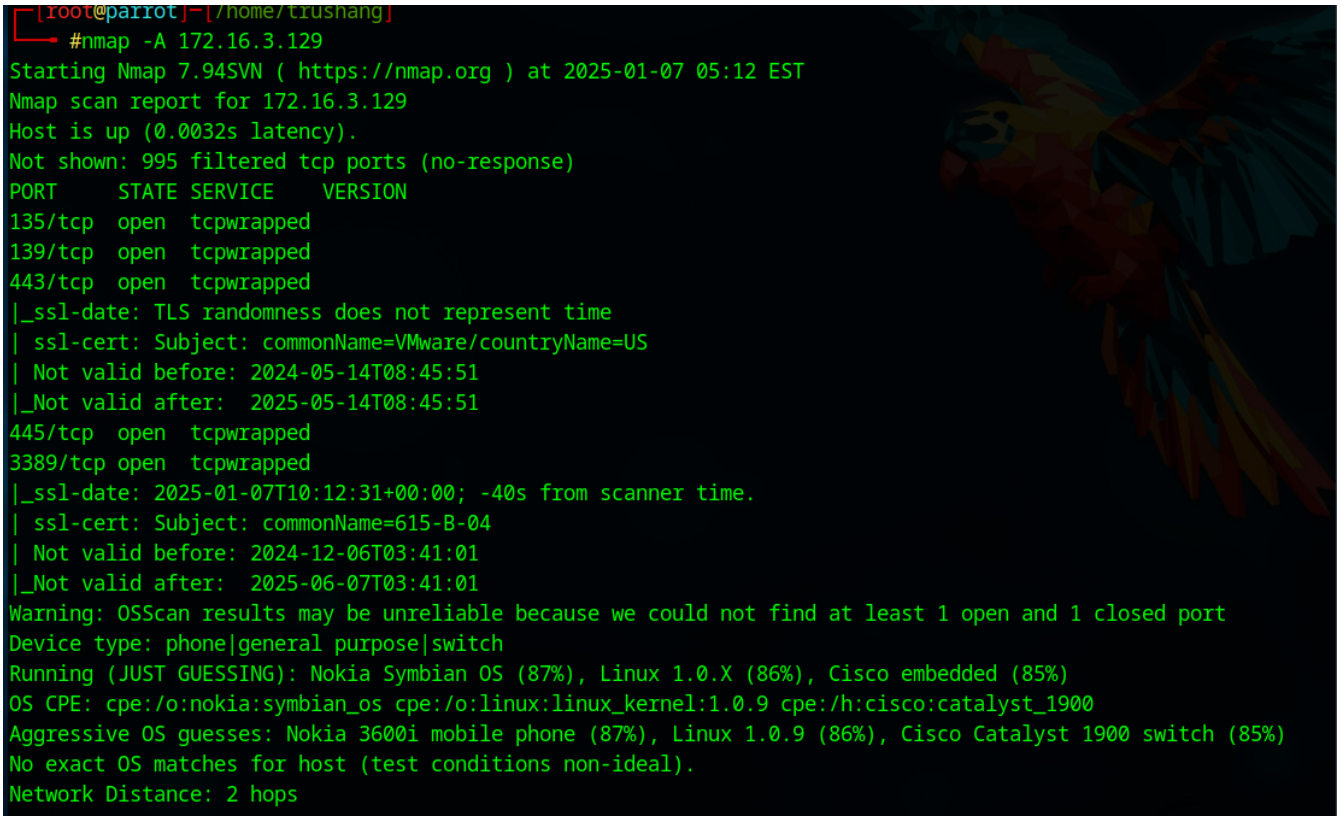


Figure 70: Enables OS detection, version detection, script scanning, and traceroute

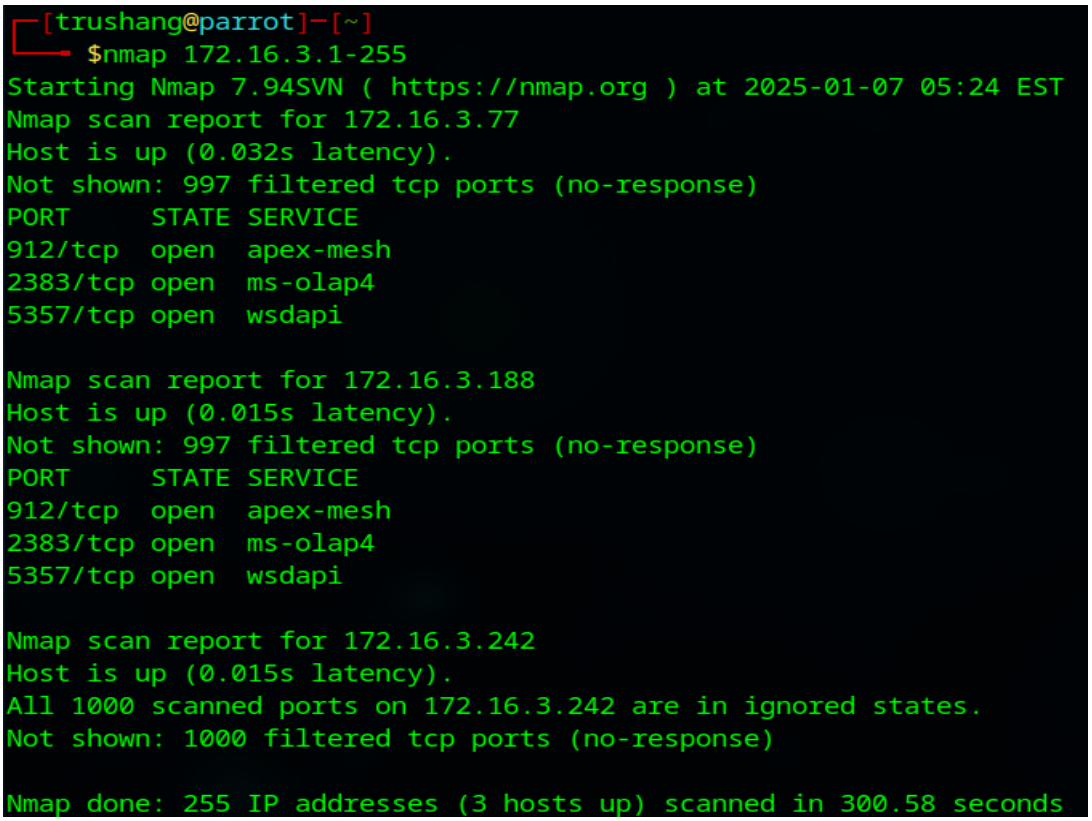


Figure 71:Scan a range of Ip range

```
┌──[✗]─[trushang@parrot]─[~]
└──╼ $nmap charusat.edu.in
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 05:33 EST
Nmap scan report for charusat.edu.in (117.239.83.200)
Host is up (0.011s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
443/tcp  open  https
911/tcp  open  xact-backup
4443/tcp open  pharos
4444/tcp open  krb524
9000/tcp open  cslistener

Nmap done: 1 IP address (1 host up) scanned in 38.39 seconds
```

Figure 72:Scan a domain

```
┌──[root@parrot]─[/home/trushang]
└──╼ #nmap -sS 172.16.3.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 05:48 EST
Nmap scan report for 172.16.3.129
Host is up (0.035s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT     STATE SERVICE
2383/tcp open  ms-olap4

Nmap done: 1 IP address (1 host up) scanned in 21.08 seconds
```

Figure 73:TCP SYN port scan (Default)

```
┌──[✗]─[trushang@parrot]─[~]
└──╼ $nmap 172.16.3.129/24 --disable-arp-ping
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 05:49 EST
Nmap done: 256 IP addresses (0 hosts up) scanned in 104.57 seconds
```

Figure 74:Scan using CIDR notation

LATEST APPLICATIONS:

- Vulnerability Scanning & Exploitation
- Cloud Security
- IoT Device Discovery and Security
- Automated Network Discovery and Asset Management
- Remote Monitoring and Incident Response
- Automating Network Scans with APIs

LEARNING OUTCOME:

In this practical, we learn that port scanning with Nmap is used to identify open ports and services on a network. By combining foot printing and enumeration techniques, Nmap helps security professionals map networks and assess vulnerabilities.

REFERENCES:

1. YouTube: https://www.youtube.com/watch?v=fp1042XK4A8
2. Nmap: https://nmap.org/

PRACTICAL: 5

AIM:

RSA algorithm is a public key encryption technology. It is considered the most secure way of encryption to secure sensitive data, particularly when sent over an insecure network such as the Internet. The public and private key generation algorithm is the most complex part of the RSA algorithm. The strength of RSA is the difficulty of factoring large integers that are the product of two large prime numbers, which is considered infeasible due to the time it would take using even today's highly configured computers. Implement the RSA algorithm.

THEORY:

RSA (Rivest–Shamir–Adleman) is a public-key cryptosystem, one of the oldest widely used for secure data transmission. The initialism "RSA" comes from the surnames of Ron Rivest, Adi Shamir and Leonard Adleman, who publicly described the algorithm in 1977. An equivalent system was developed secretly in 1973 at Government Communications Headquarters (GCHQ), the British signals intelligence agency, by the English mathematician Clifford Cocks. That system was declassified in 1997

In a public-key cryptosystem, the encryption key is public and distinct from the decryption key, which is kept secret (private). An RSA user creates and publishes a public key based on two large prime numbers, along with an auxiliary value. The prime numbers are kept secret. Messages can be encrypted by anyone, via the public key, but can only be decrypted by someone who knows the private key

The security of RSA relies on the practical difficulty of factoring the product of two large prime numbers, the "factoring problem". Breaking RSA encryption is known as the RSA problem. Whether it is as difficult as the factoring problem is an open question. There are no published methods to defeat the system if a large enough key is used

RSA is a relatively slow algorithm. Because of this, it is not commonly used to directly encrypt user data. More often, RSA is used to transmit shared keys for symmetric-key cryptography, which are then used for bulk encryption–decryption.

The RSA algorithm involves four steps: key generation, key distribution, encryption, and decryption.

A basic principle behind RSA is the observation that it is practical to find three very large positive integers e, d, and n, such that for all integers m (0<= m< n), both $(m^e)^d$ and m have the same remainder when divided by n (they are congruent modulo n):

$$(m^e)^d \equiv m \pmod{n}.$$

Key generation

The keys for the RSA algorithm are generated in the following way:

1. Choose two large prime numbers *p* and *q*.
    - To make factoring harder, *p* and *q* should be chosen at random, be both large and have a large difference. For choosing them the standard method is to choose random integers and use a primality test until two primes are found.

- *p* and *q* are kept secret.

2. Compute n = pq.
   - n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.
   - n is released as part of the public key.
3. Compute λ(n), where λ is Carmichael's totient function. Since n = pq, λ(n) = lcm(λ(p), λ(q)), and since p and q are prime, λ(p) = φ(p) = p − 1, and likewise λ(q) = q − 1. Hence λ(n) = lcm(p − 1, q − 1).
   - The lcm may be calculated through the Euclidean algorithm, since lcm(a, b) = |ab|/gcd(a, b).
   - λ(n) is kept secret.
4. Choose an integer e such that 1 < e < λ(n) and gcd(e, λ(n)) = 1; that is, e and λ(n) are coprime.
   - e having a short bit-length and small Hamming weight results in more efficient encryption – the most commonly chosen value for e is 216 + 1 = 65537. The smallest (and fastest) possible value for e is 3, but such a small value for e has been shown to be less secure in some settings.
   - e is released as part of the public key.
5. Determine d as d ≡ e−1 (mod λ(n)); that is, d is the modular multiplicative inverse of e modulo λ(n).
   - This means: solve for d the equation de ≡ 1 (mod λ(n)); d can be computed efficiently by using the extended Euclidean algorithm, since, thanks to e and λ(n) being coprime, said equation is a form of Bézout's identity, where d is one of the coefficients.
   - d is kept secret as the private key exponent.

Key distribution

Suppose that Bob wants to send information to Alice. If they decide to use RSA, Bob must know Alice's public key to encrypt the message, and Alice must use her private key to decrypt the message.

To enable Bob to send his encrypted messages, Alice transmits her public key (n, e) to Bob via a reliable, but not necessarily secret, route. Alice's private key (d) is never distributed.

Encryption

After Bob obtains Alice's public key, he can send a message M to Alice.

To do it, he first turns M (strictly speaking, the un-padded plaintext) into an integer m (strictly speaking, the padded plaintext), such that 0 ≤ m < n by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c, using Alice's public key e, corresponding to

$$c \equiv m^e \pmod{n}.$$

This can be done reasonably quickly, even for very large numbers, using modular exponentiation. Bob then transmits *c* to Alice. Note that at least nine values of *m* will yield a ciphertext *c* equal to *m*, but this is very unlikely to occur in practice.

Decryption

Alice can recover *m* from *c* by using her private key exponent *d* by computing

$$c^d \equiv (m^e)^d \equiv m \pmod{n}.$$

Given *m*, she can recover the original message *M* by reversing the padding scheme.

CODE:

```
import sympy
import random
import psutil
import time

def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

def extended_gcd(a, b):
    old_r, r = a, b
    old_s, s = 1, 0
    old_t, t = 0, 1

    while r != 0:
        quotient = old_r // r
        old_r, r = r, old_r - quotient * r
        old_s, s = s, old_s - quotient * s
        old_t, t = t, old_t - quotient * t

    return old_r, old_s, old_t

def mod_inverse(a, m):
    gcd_value, x, y = extended_gcd(a, m)
    if gcd_value != 1:
        raise ValueError(f"{a} and {m} are not coprime, modular inverse does not exist.")
    else:
        return x % m

def generate_large_prime(decimal_digits):
    return sympy.randprime(10**(decimal_digits-1), 10**decimal_digits)

def generate_public_exponent(phi_n, decimal_digits):
    e = random.randint(10**(decimal_digits-1), 10**decimal_digits - 1)
    while gcd(e, phi_n) != 1:
        e = random.randint(10**(decimal_digits-1), 10**decimal_digits - 1)
    return e

def generate_keys(decimal_digits):
    cpu_before, ram_before = get_system_utilization()
    p = generate_large_prime(decimal_digits)
    q = generate_large_prime(decimal_digits)
    n = p * q
```

```
    phi_n = (p - 1) * (q - 1)
    e = generate_public_exponent(phi_n, decimal_digits)
    cpu_during_gen, ram_during_gen = get_system_utilization()
    d = mod_inverse(e, phi_n)
    cpu_after, ram_after = get_system_utilization()
    print("\nKey Generation Utilization:")
    print(f"CPU Before Key Generation: {cpu_before}% | RAM Before Key Generation:
{ram_before}%")
    print(f"CPU During Key Generation: {cpu_during_gen}% | RAM During Key Generation:
{ram_during_gen}%")
    print(f"CPU After Key Generation: {cpu_after}% | RAM After Key Generation: {ram_after}%")

    return ((n, e), (n, d), p, q)

def encrypt(message, pub_key):
    n, e = pub_key
    cipher_text = [pow(ord(char), e, n) for char in message]
    return cipher_text

def decrypt(cipher_text, priv_key):
    n, d = priv_key
    message = ''.join([chr(pow(char, d, n)) for char in cipher_text])
    return message

def get_system_utilization():
    cpu = psutil.cpu_percent(interval=1)
    ram = psutil.virtual_memory().percent
    return cpu, ram

def rsa_algorithm(message, decimal_digits):
    start_time = time.time()

    pub_key, priv_key, p, q = generate_keys(decimal_digits)

    cpu_before_enc, ram_before_enc = get_system_utilization()

    cipher_text = encrypt(message, pub_key)

    cpu_during_enc, ram_during_enc = get_system_utilization()

    decrypted_message = decrypt(cipher_text, priv_key)

    cpu_after_enc, ram_after_enc = get_system_utilization()

    end_time = time.time()

    print(f"\nOriginal Message: {message}")
    print(f"Ciphertext: {cipher_text}")
    print(f"Decrypted Message: {decrypted_message}")

    print("\nSystem Utilization During Encryption/Decryption:")
```

```
    print(f"CPU Before Encryption: {cpu_before_enc}% | RAM Before Encryption:
{ram_before_enc}%")
    print(f"CPU During Encryption: {cpu_during_enc}% | RAM During Encryption:
{ram_during_enc}%")
    print(f"CPU After Encryption: {cpu_after_enc}% | RAM After Encryption: {ram_after_enc}%")

    print(f"Execution Time: {end_time - start_time:.6f} seconds")
    print(f"Generated Prime p: {p}")
    print(f"Generated Prime q: {q}")
    print(f"Public Exponent e: {pub_key[1]}")
    print(f"Private Exponent d: {priv_key[1]}")

if __name__ == "__main__":
    message = input("Enter the message to encrypt: ")
    decimal_digits = int(input("Enter the number of decimal digits for p, q, and e: "))
    rsa_algorithm(message, decimal_digits)
```

OUTPUT:

```
PROBLEMS  COMMENTS  PORTS  DEBUG CONSOLE  OUTPUT  TERMINAL

C:\Users\HP\Desktop\SEM-6\CRNS\PR5>python3 Practical-7.py
Enter the message to encrypt: Trushangkumar sumanbhai patel
Enter the number of decimal digits for p, q, and e: 10

Key Generation Utilization:
CPU Before Key Generation: 5.7% | RAM Before Key Generation: 83.5%
CPU During Key Generation: 4.4% | RAM During Key Generation: 83.5%
CPU After Key Generation: 8.7% | RAM After Key Generation: 83.9%

Original Message: Trushangkumar sumanbhai patel
Ciphertext: [4453532882523342408, 6399500532632042452, 1440956188479928763, 604955520960206071, 5159882049969110120, 2069423041551465604, 935424833
3781236276, 2260547122366746860, 8317109596080542601, 1440956188479928763, 3969103403446493610, 2069423041551465604, 6399500532632042452, 2925968607955983467, 6049555209602060
07955983467, 604955520960206071, 1440956188479928763, 3969103403446493610, 2069423041551465604, 935424833781236276, 3905712531264677894, 515988204
9969110120, 2069423041551465604, 6577943649377056241, 2925968607955983467, 541775565799116804, 2069423041551465604, 5669203044497453248, 339769707
0105920668, 3925536414680111882]
Decrypted Message: Trushangkumar sumanbhai patel

System Utilization During Encryption/Decryption:
CPU Before Encryption: 5.9% | RAM Before Encryption: 83.3%
CPU During Encryption: 7.8% | RAM During Encryption: 83.3%
CPU After Encryption: 5.4% | RAM After Encryption: 83.2%
Execution Time: 6.045281 seconds
Generated Prime p: 1562305163
Generated Prime q: 6239118947
Public Exponent e: 1452316119
Private Exponent d: 9201074065718868455

C:\Users\HP\Desktop\SEM-6\CRNS\PR5>
```

Figure 75: 10 Decimal Places for Prime Numbers in RSA: Key Generation, Encryption, and Decryption Performance

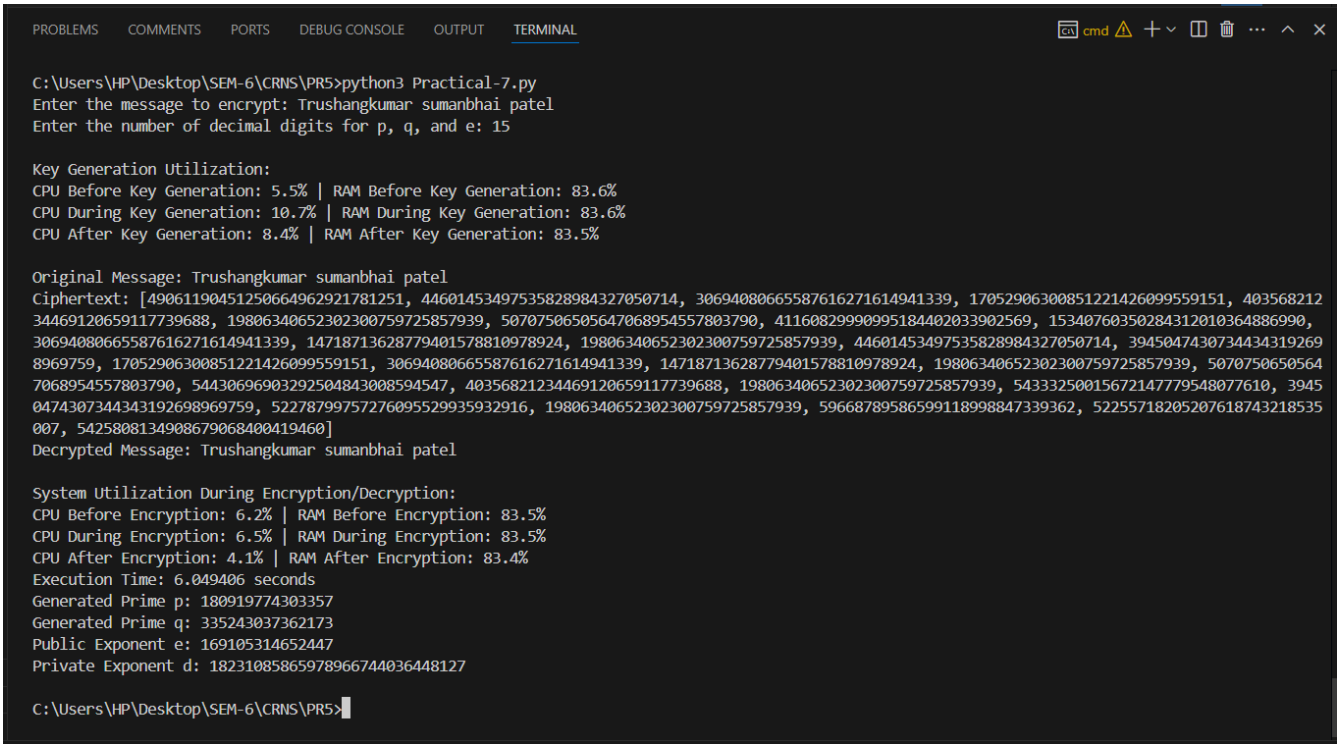


Figure 76: 15 Decimal Places for Prime Numbers in RSA: Key Generation, Encryption, and Decryption Performance

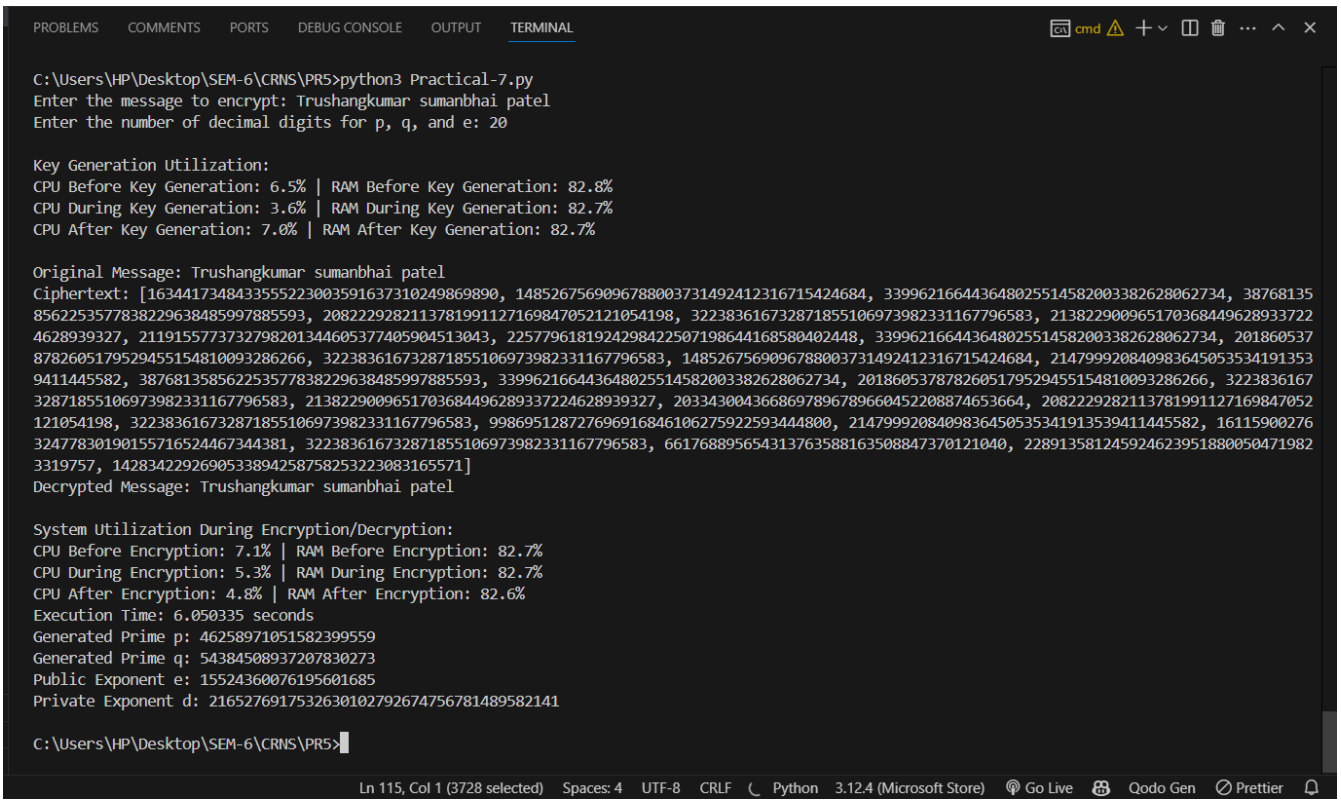


Figure 77: 20 Decimal Places for Prime Numbers in RSA: Key Generation, Encryption, and Decryption Performance

Decimal Places	Stage	CPU Utilization (%)	RAM Utilization (%)
10	Before Key Generation	5.7	83.5
	During Key Generation	4.4	83.5
	After Key Generation	8.7	83.9
	Before Encryption	5.9	83.3
	During Encryption	7.8	83.3
	After Encryption	5.4	83.2
15	Before Key Generation	5.5	83.6
	During Key Generation	10.7	83.6
	After Key Generation	8.4	83.5
	Before Encryption	6.2	83.5
	During Encryption	6.5	83.5
	After Encryption	4.1	83.4
20	Before Key Generation	6.5	82.8
	During Key Generation	3.6	82.7
	After Key Generation	7	82.7
	Before Encryption	7.1	82.7
	During Encryption	5.3	82.7
	After Encryption	4.8	82.6

LATEST APPLICATIONS:

- Secure Web Communications (SSL/TLS)
- Digital Signatures
- Cryptographic Key Management
- Email Encryption (S/MIME)
- Blockchain & Cryptocurrencies
- Authentication Systems

LEARNING OUTCOME:

In this practical, I tested the RSA algorithm with primes of 10, 15, and 20 decimal places to assess system utilization. CPU and RAM usage remained stable, with slight increases during key generation. The highest resource usage occurred during key generation. Overall, RSA demonstrated efficient performance with minimal impact on system resources, even with larger primes.

REFERENCES:

3. YouTube: https://www.youtube.com/watch?v=VF3AHG0T9ec
4. Wikipedia: https://en.wikipedia.org/wiki/RSA_(cryptosystem)
5. ChatGPT: https://chatgpt.com/

PRACTICAL: 6

AIM:

A medium-sized enterprise is concerned about the security posture of its internal network after a recent breach. The IT security team uses OpenVAS to conduct a comprehensive vulnerability assessment of the network's servers and devices to identify exploitable vulnerabilities. Perform a vulnerability scan on a network using OpenVAS and analyse the results to identify and mitigate potential security risks.

THEORY:

OpenVAS is an open-source vulnerability scanning and management tool that helps to identify security issues like misconfigurations, outdated software, and weak passwords that could be exploited by attackers. OpenVAS is widely used by security professionals to assess and improve the security posture of their networks and is known for its effectiveness and flexibility.

Working of OpenVAS

OpenVAS consists of a server and various client-side tools for scanning and reporting. It uses a regularly updated database of known vulnerabilities and checks systems against these to detect potential weaknesses. The tool performs a comprehensive scan of the specified targets, identifying potential vulnerabilities such as outdated software, misconfigurations, and weak passwords and generates comprehensive reports detailing the identified vulnerabilities and provide recommendations for remediation.

A vulnerability assessment tool works in the following way as follows.

1. Classifies the system resources.
2. Allocates the enumerable values to the classified resources.
3. Detects the possible threats (vulnerabilities) in each resource.
4. Eliminates the vulnerabilities on a priority basis.

Components of OpenVAS architecture

- **OpenVAS Scanner**:
  - The primary engine that performs the actual scanning of target systems. It uses Network Vulnerability Tests (NVTs) to detect security vulnerabilities.
- **OpenVAS Manager**:
  - Manages scan configurations, schedules, and stores scan results. It acts as an intermediary between the scanner and the user interfaces, handling scan requests and processing results.

- **Green bone Security Assistant (GSA)**:
  - A web-based graphical user interface (GUI) that allows users to manage scans, configure settings, and view scan results. It provides an easy-to-use platform for interacting with OpenVAS.
- **OpenVAS CLI**:
  - A command-line interface for users who prefer scripting and command-line operations. It enables management of scans, targets, and results through commands and scripts.
- **Green bone Security Feed (GSF)**:
  - A continuously updated feed that provides the latest Network Vulnerability Tests (NVTs) and security information. It ensures OpenVAS can detect the most recent vulnerabilities.
- **OpenVAS Libraries**:
  - These libraries provide essential functionalities required by the scanner and manager, such as network communication, data storage, and cryptographic operations.
- **Database**:
  - The database stores scan results, configurations, and other essential data. It ensures data persistence and retrieval for analysis and reporting purposes.

CODE:

```
• sudo su
• apt update && apt upgrade
• apt-get update  --fix-missing
• apt install gvm
```

OUTPUT:

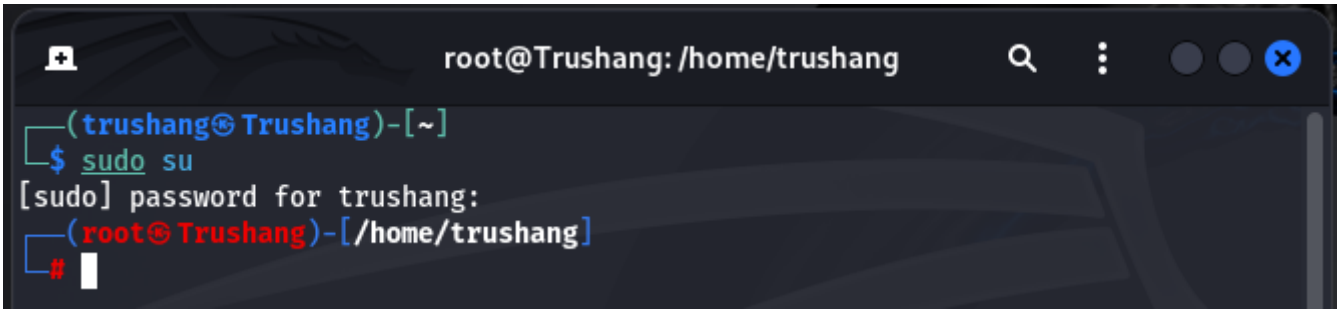


Figure 78: Gain root access

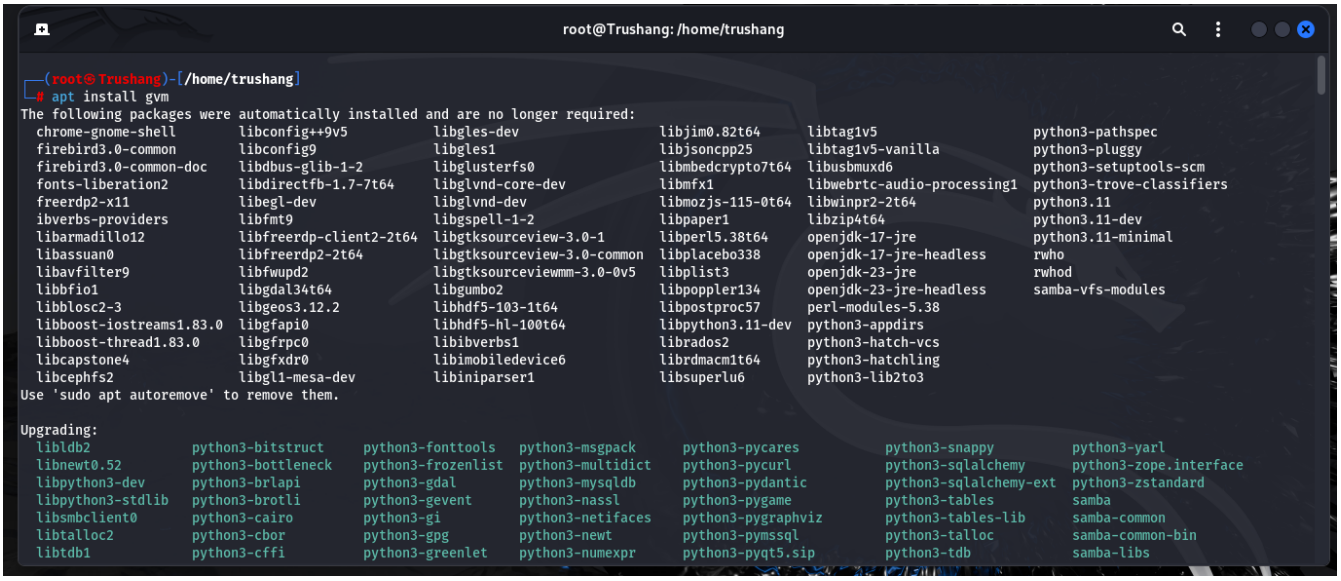


Figure 79: Install gvm in your system

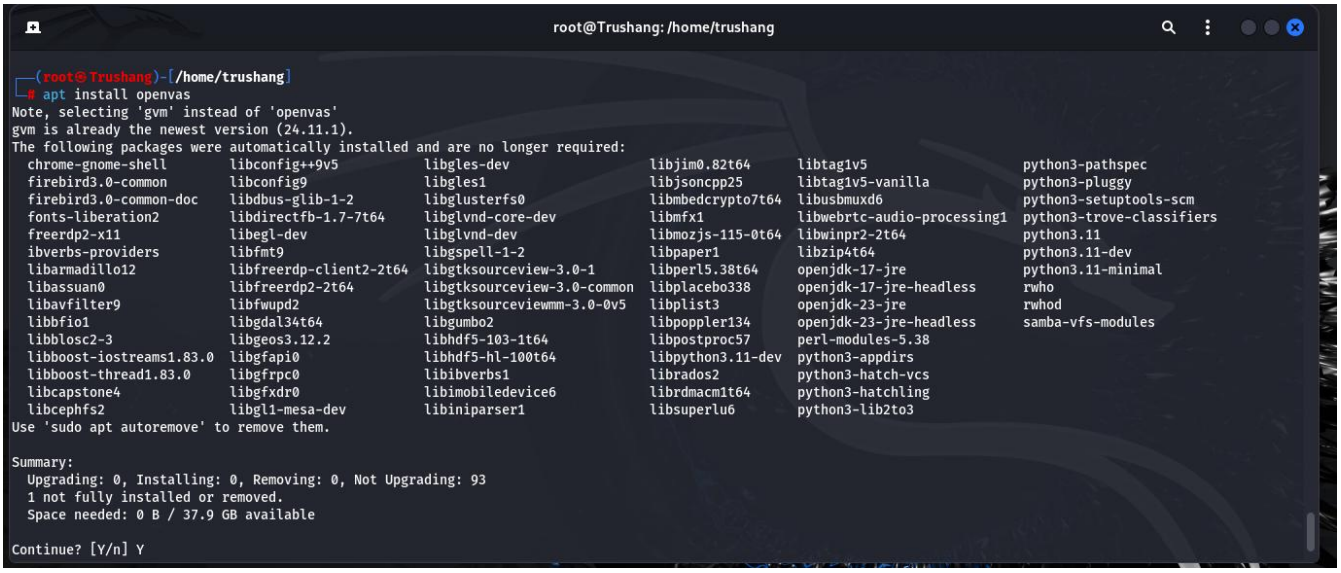


Figure 80: Install OpenVAS in your system

Figure 81: We have to change the port of PostgreSQL

Figure 82: Delete PostgreSQL 16 and 17 and then create 17 main cluster

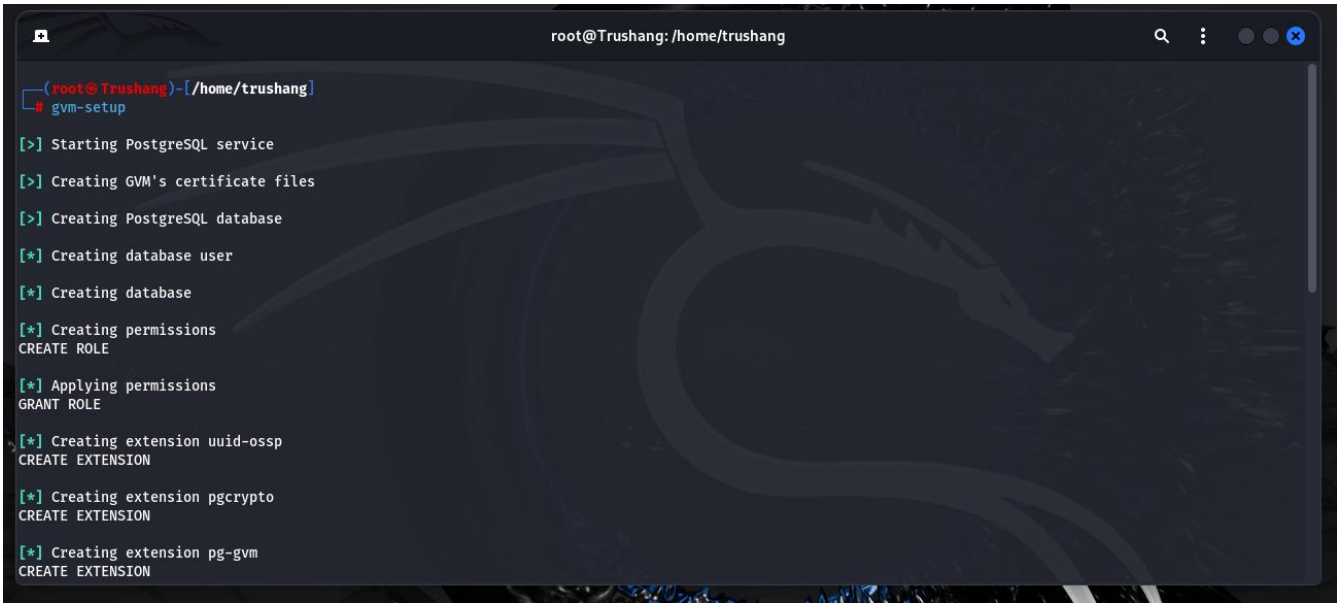


Figure 83: Start GVM setup

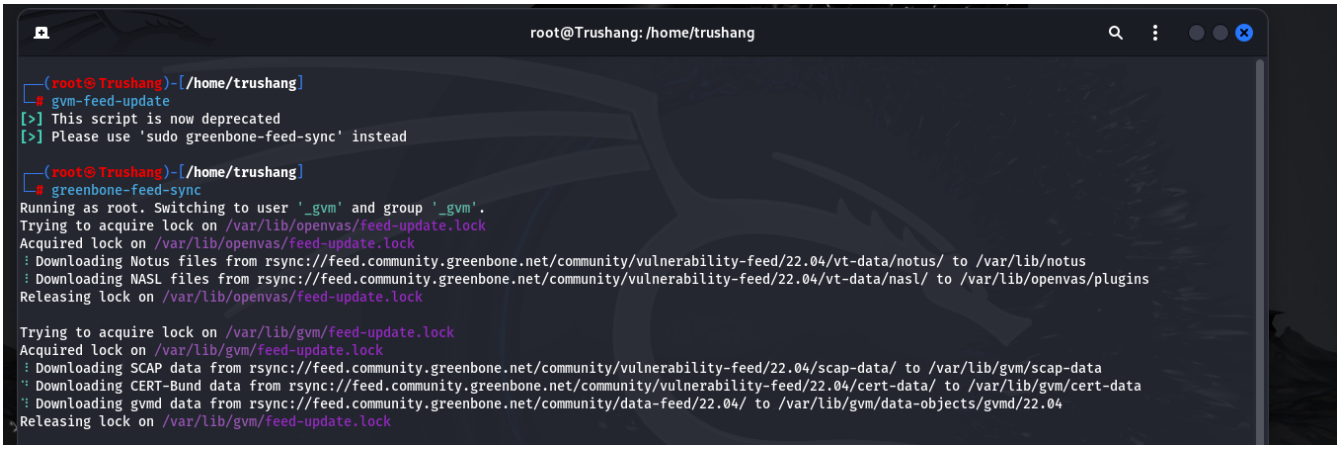


Figure 84: Sync NVT, SCAP, CERT, GVMD_DATA

Figure 85: Start GVM

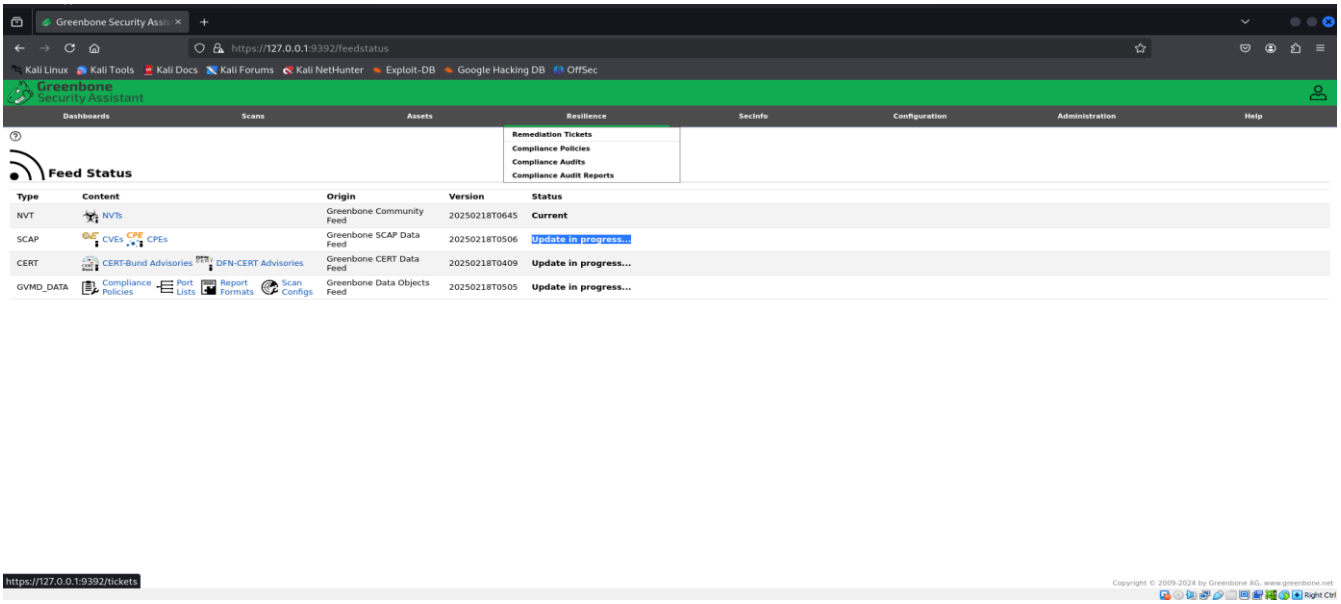


Figure 86: Check feed status

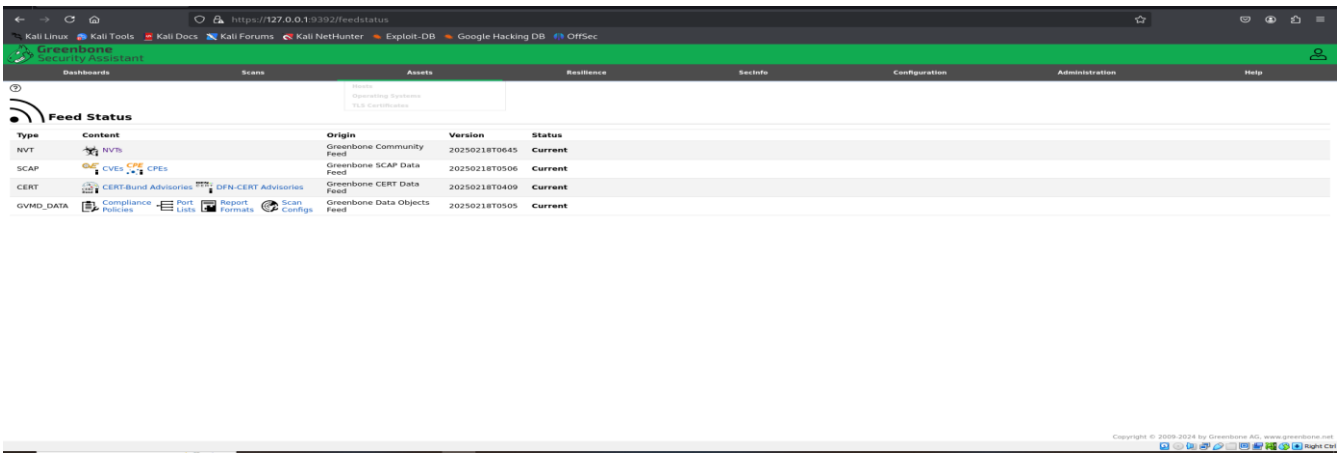


Figure 87: Feed status is now current

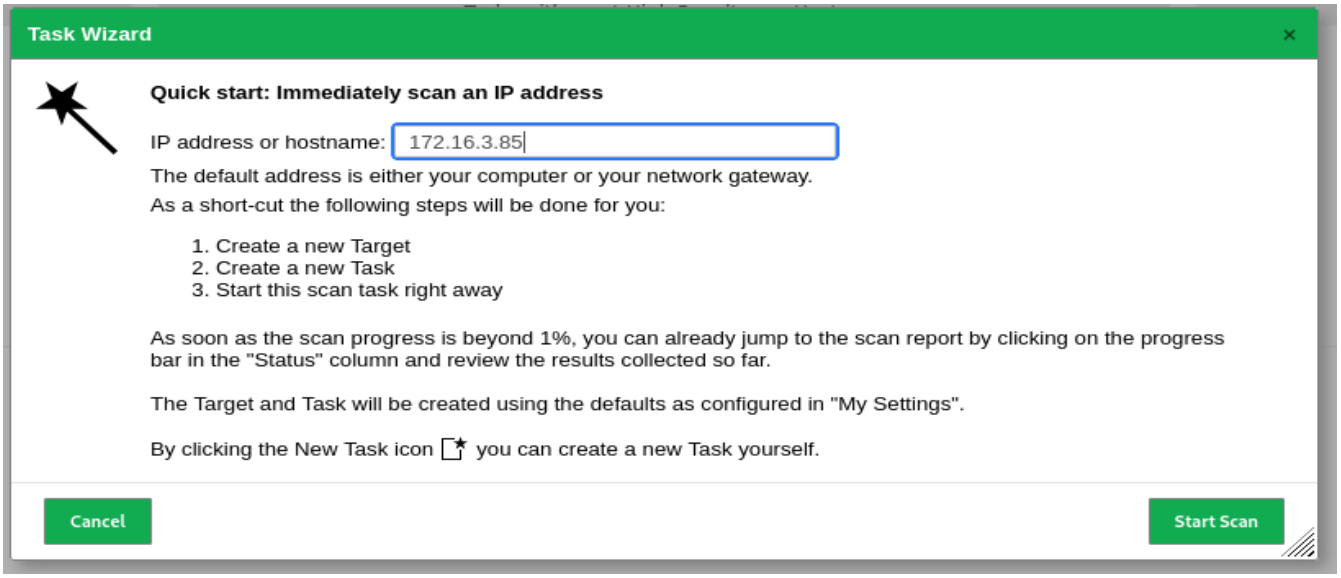


Figure 88: Write IP address for scan

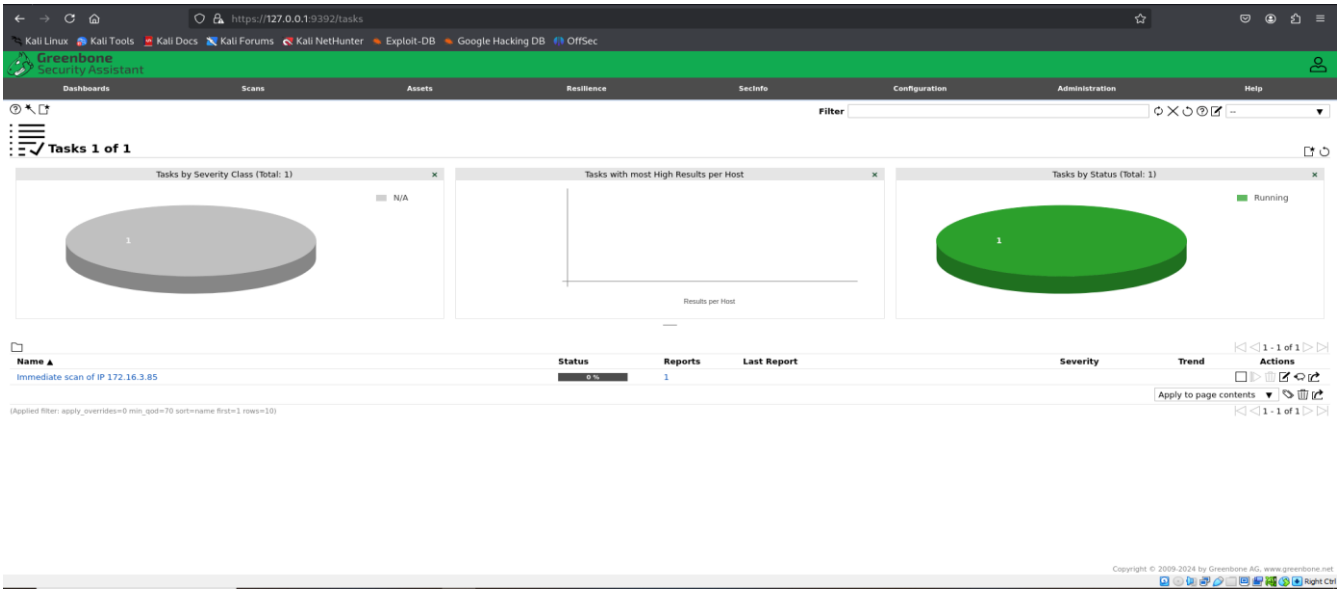
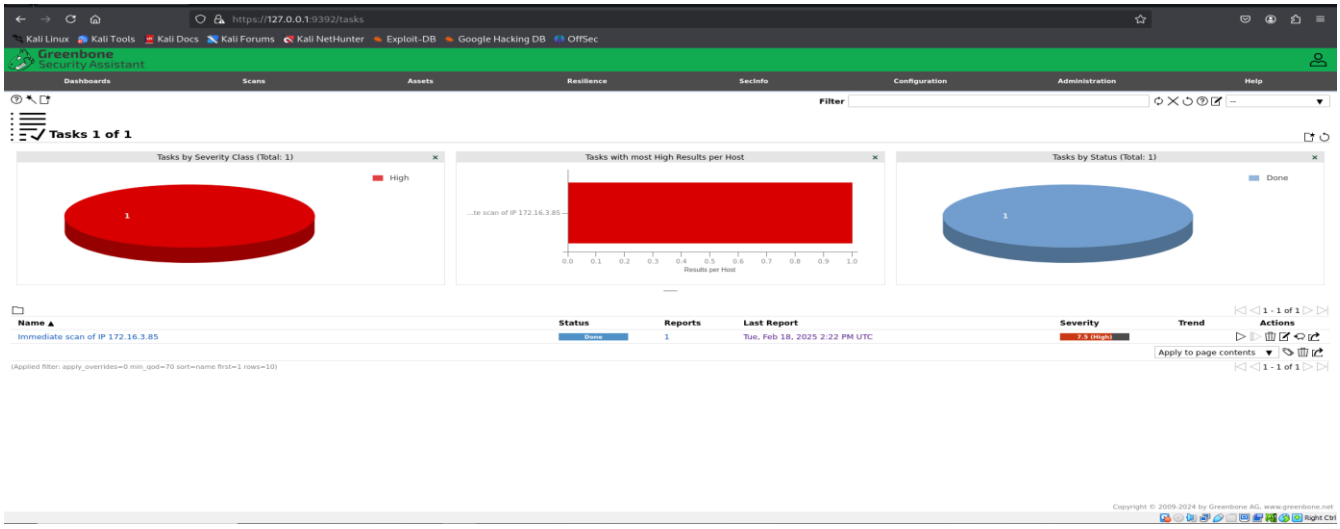
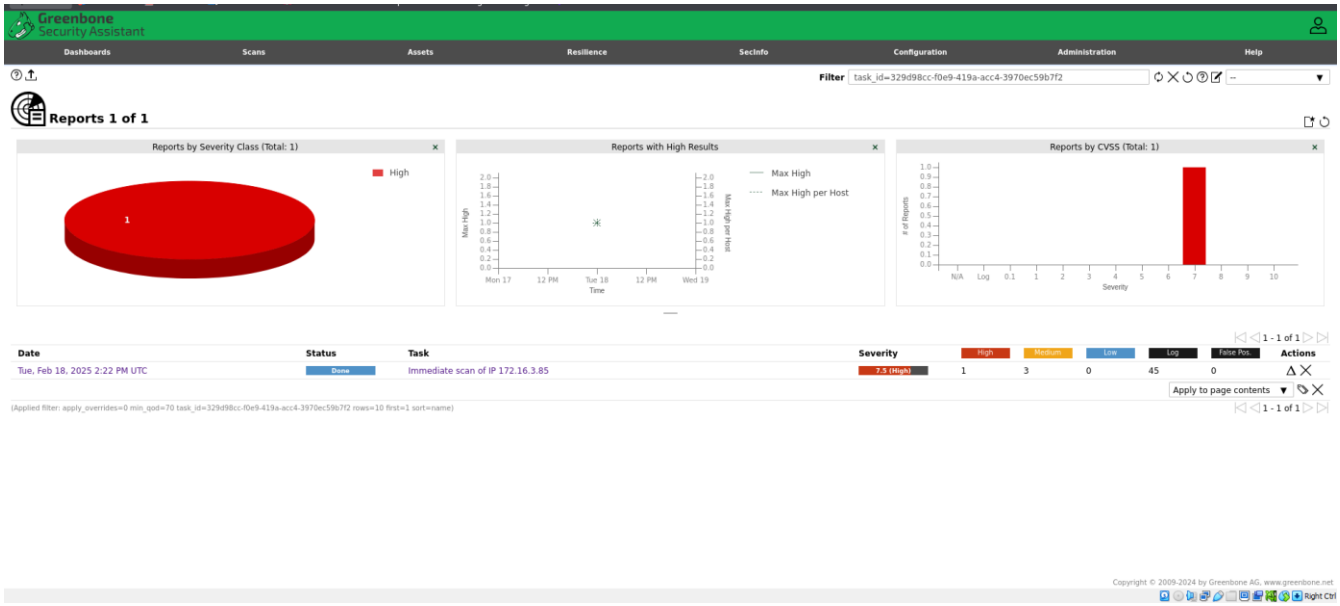
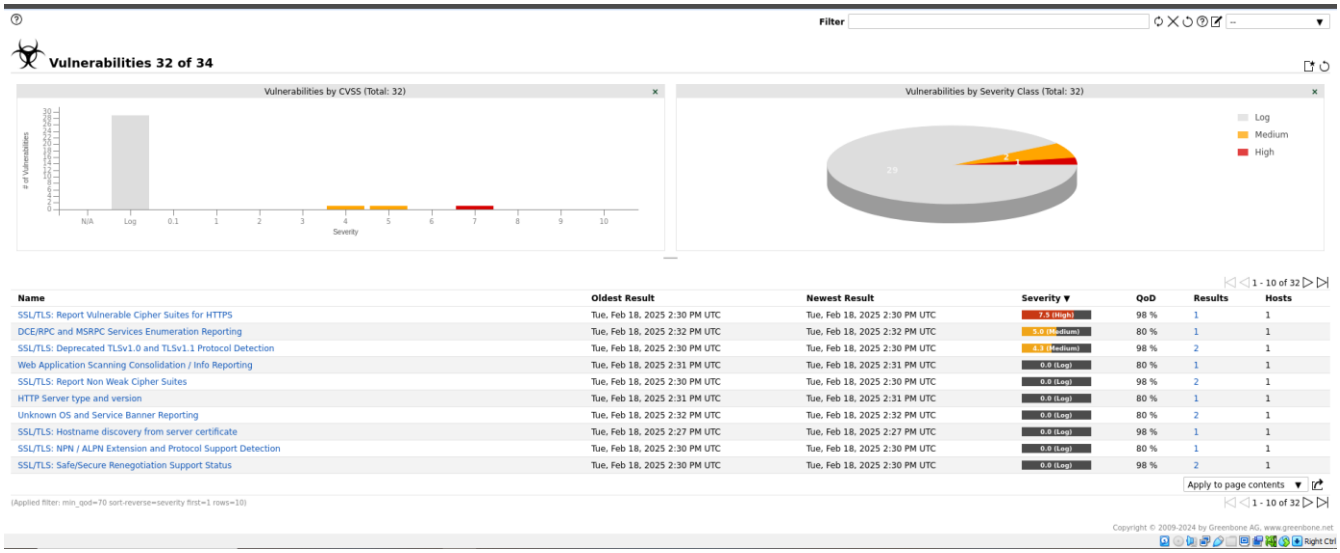
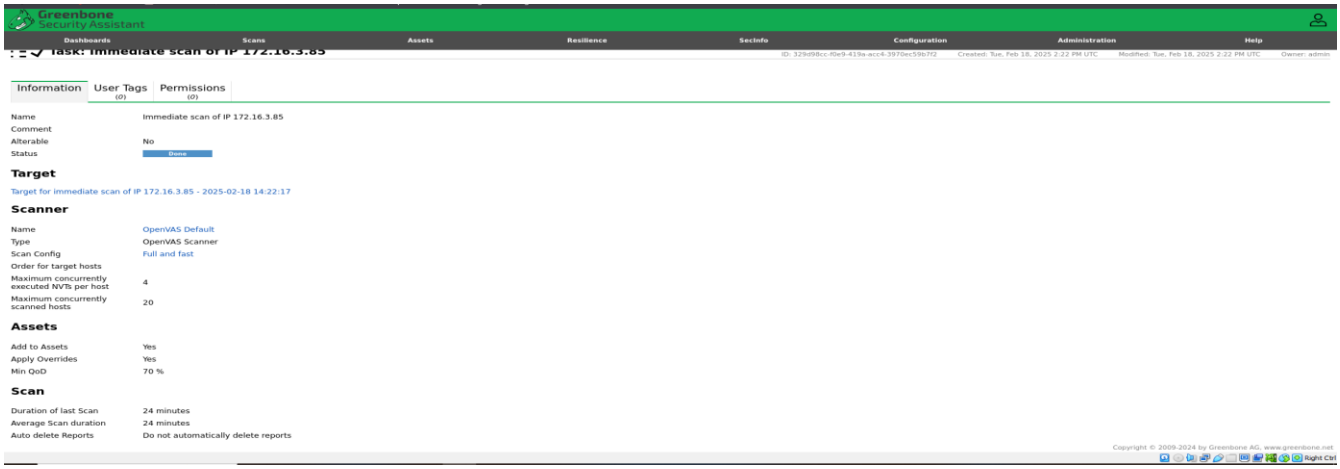


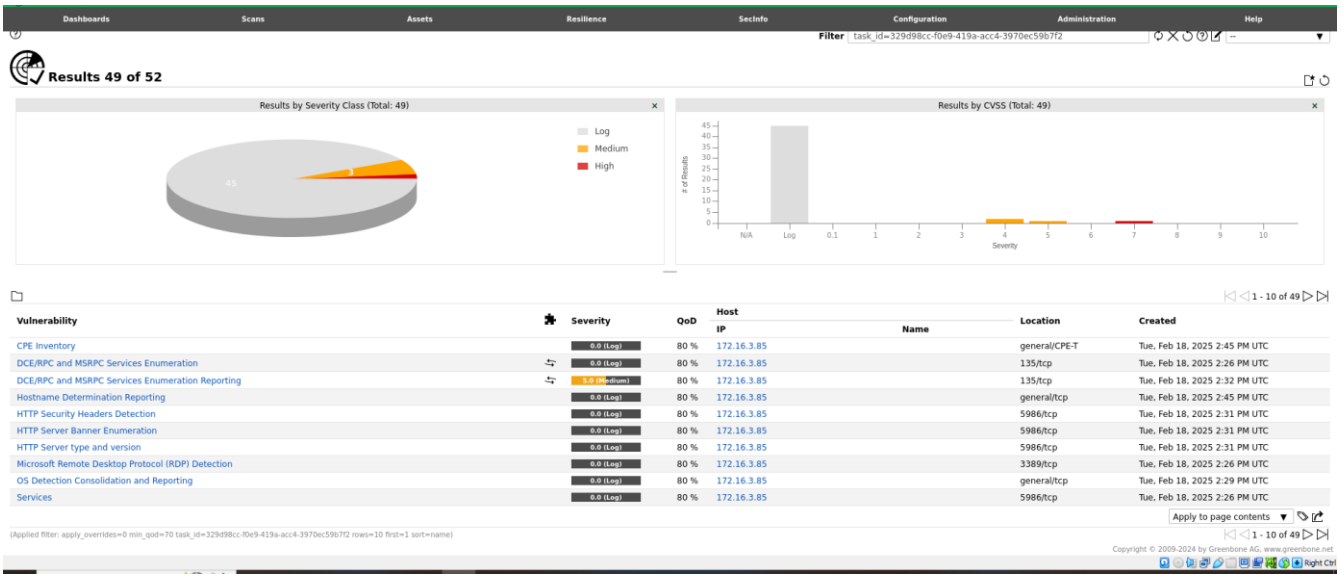
Figure 89: Scanning IP address

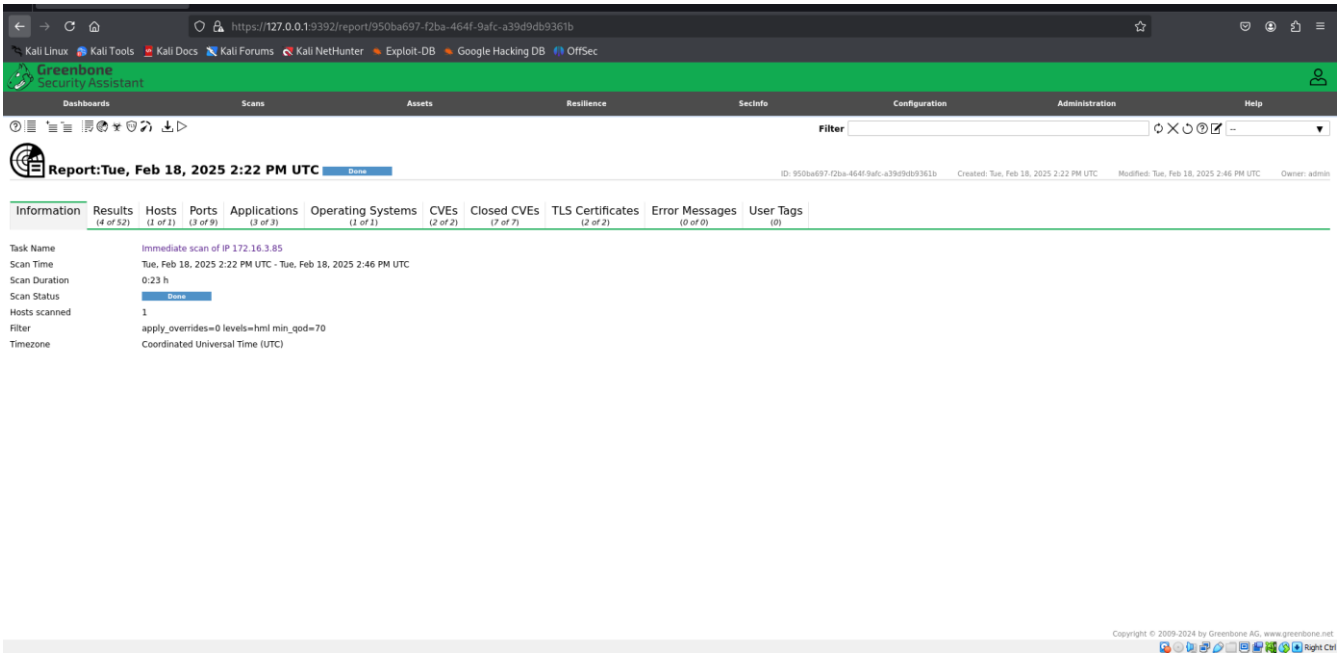


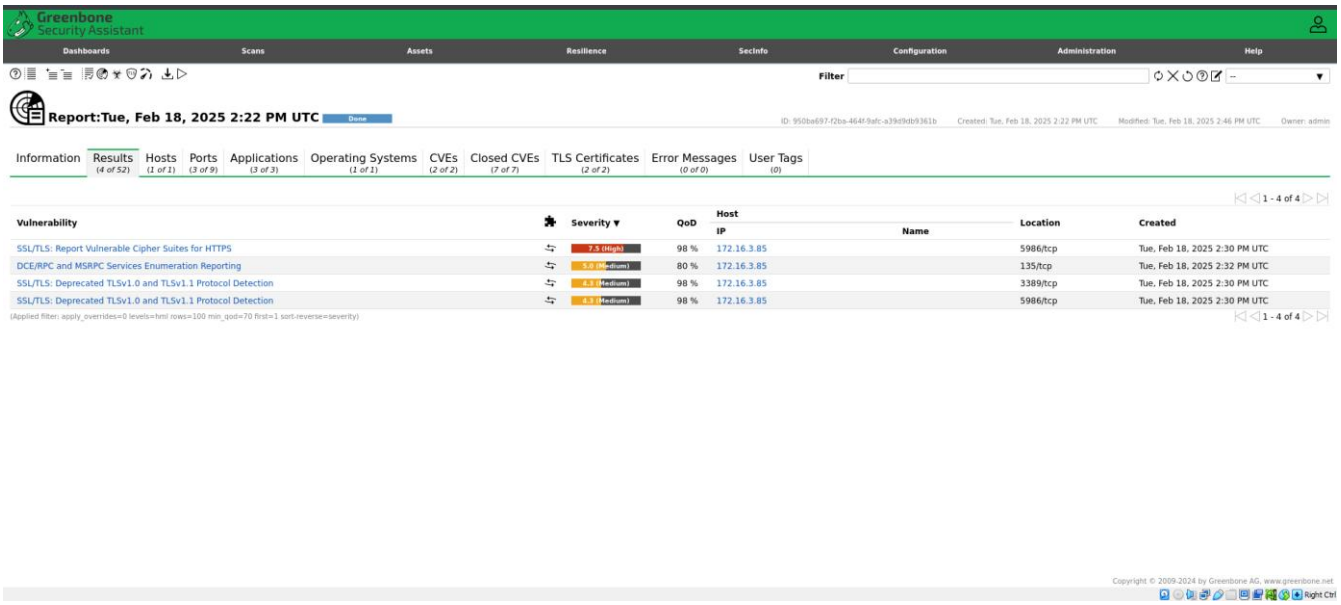

Greenbone Security Assistant
Dashboards Scans Assets Resilience SecInfo Configuration Administration Help
Filter task_id=329d98cc-f0e9-419a-acc4-3970ec59b7f2
Reports 1 of 1
Reports by Severity Class (Total: 1)
High
Reports with High Results
Max High
Max High per Host
Reports by CVSS (Total: 1)
Date Status Task Severity High Medium Low Log False Pos. Actions
Tue, Feb 18, 2025 2:22 PM UTC Done Immediate scan of IP 172.16.3.85 7.5 (High) 1 3 0 45 0
Apply to page contents
1 - 1 of 1

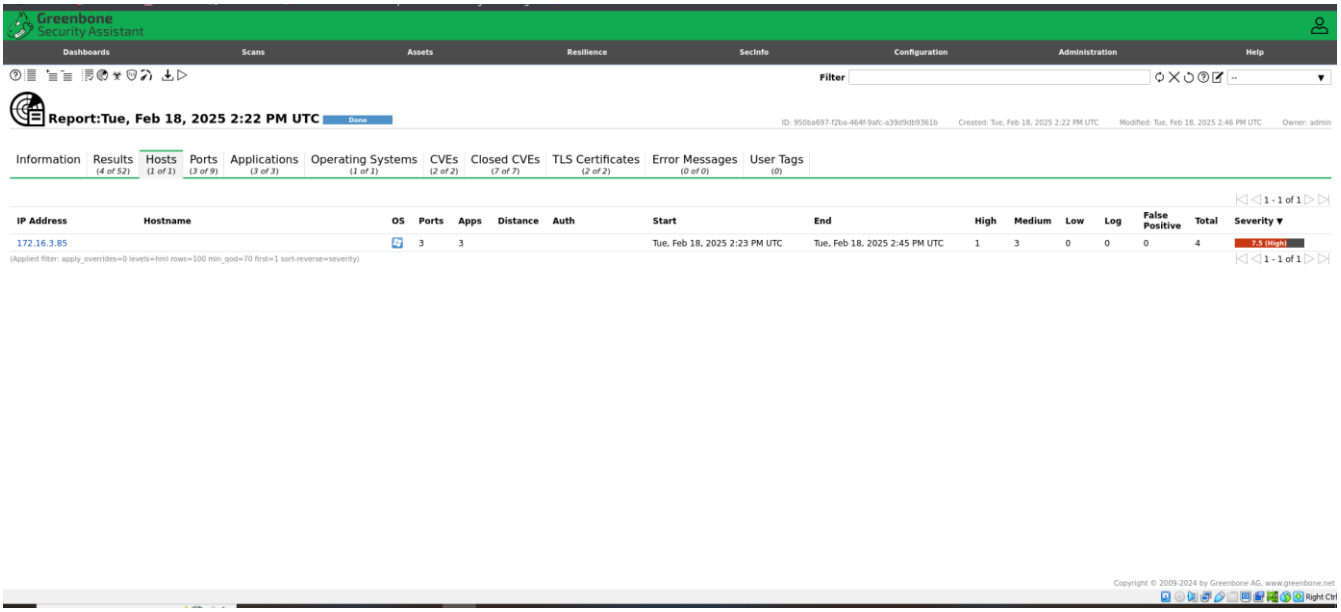

Filter
Vulnerabilities 32 of 34
Vulnerabilities by CVSS (Total: 32)
Vulnerabilities by Severity Class (Total: 32)
Log
Medium
High
Name Oldest Result Newest Result Severity QoD Results Hosts
SSL/TLS: Report Vulnerable Cipher Suites for HTTPS Tue, Feb 18, 2025 2:30 PM UTC Tue, Feb 18, 2025 2:30 PM UTC 7.5 (High) 98 % 1 1
DCE/RPC and MSRPC Services Enumeration Reporting Tue, Feb 18, 2025 2:32 PM UTC Tue, Feb 18, 2025 2:32 PM UTC 5.0 (Medium) 80 % 1 1
SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection Tue, Feb 18, 2025 2:30 PM UTC Tue, Feb 18, 2025 2:30 PM UTC 4.3 (Medium) 98 % 2 1
Web Application Scanning Consolidation / Info Reporting Tue, Feb 18, 2025 2:31 PM UTC Tue, Feb 18, 2025 2:31 PM UTC 0.0 (Log) 80 % 1 1
SSL/TLS: Report Non Weak Cipher Suites Tue, Feb 18, 2025 2:30 PM UTC Tue, Feb 18, 2025 2:30 PM UTC 0.0 (Log) 98 % 2 1
HTTP Server type and version Tue, Feb 18, 2025 2:31 PM UTC Tue, Feb 18, 2025 2:31 PM UTC 0.0 (Log) 80 % 1 1
Unknown OS and Service Banner Reporting Tue, Feb 18, 2025 2:32 PM UTC Tue, Feb 18, 2025 2:32 PM UTC 0.0 (Log) 80 % 2 1
SSL/TLS: Hostname discovery from server certificate Tue, Feb 18, 2025 2:27 PM UTC Tue, Feb 18, 2025 2:27 PM UTC 0.0 (Log) 98 % 1 1
SSL/TLS: NPN / ALPN Extension and Protocol Support Detection Tue, Feb 18, 2025 2:30 PM UTC Tue, Feb 18, 2025 2:30 PM UTC 0.0 (Log) 80 % 1 1
SSL/TLS: Safe/Secure Renegotiation Support Status Tue, Feb 18, 2025 2:30 PM UTC Tue, Feb 18, 2025 2:30 PM UTC 0.0 (Log) 98 % 2 1
Apply to page contents
1 - 10 of 32

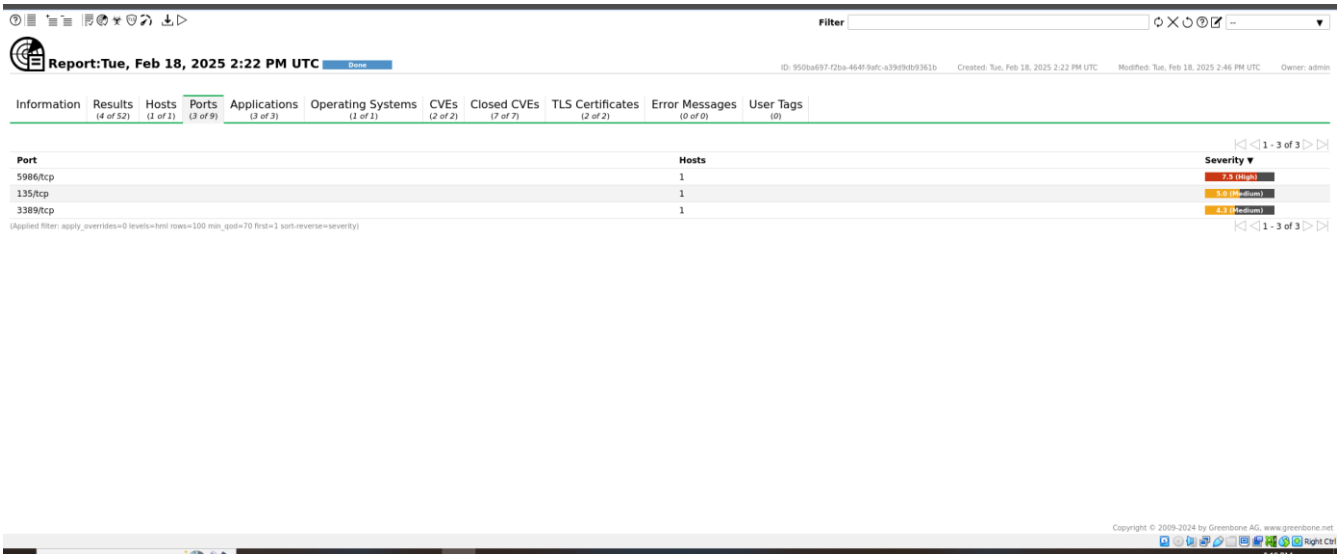

Greenbone Security Assistant
Dashboards Scans Assets Resilience SecInfo Configuration Administration Help
Information User Tags (0) Permissions (0)
Name Immediate scan of IP 172.16.3.85
Comment
Alterable No
Status Done
Target
Target for immediate scan of IP 172.16.3.85 - 2025-02-18 14:22:17
Scanner
Name OpenVAS Default
Type OpenVAS Scanner
Scan Config Full and fast
Order for target hosts
Maximum concurrently executed NVTs per host 4
Maximum concurrently scanned hosts 20
Assets
Add to Assets Yes
Apply Overrides Yes
Min QoD 70 %
Scan
Duration of last Scan 24 minutes
Average Scan duration 24 minutes
Auto delete Reports Do not automatically delete reports

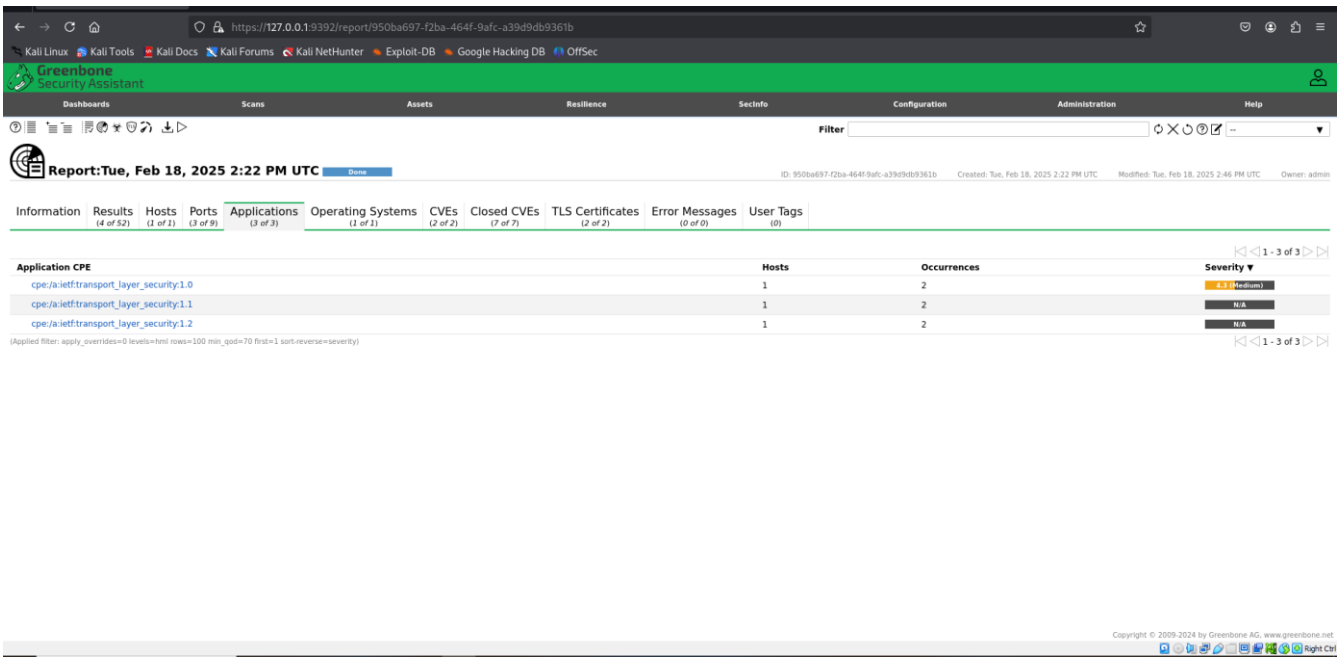

Dashboards
Scans
Assets
Resilience
SecInfo
Configuration
Administration
Help
Filter task_id=329d98cc-f0e9-419a-acc4-3970ec59b7f2
Results 49 of 52
Results by Severity Class (Total: 49)
Log
Medium
High
Results by CVSS (Total: 49)
Severity
1 - 10 of 49
Vulnerability
Severity
QoD
Host
IP
Name
Location
Created
CPE Inventory 0.0 (Log) 80 % 172.16.3.85 general/CPE-T Tue, Feb 18, 2025 2:45 PM UTC
DCE/RPC and MSRPC Services Enumeration 0.0 (Log) 80 % 172.16.3.85 135/tcp Tue, Feb 18, 2025 2:26 PM UTC
DCE/RPC and MSRPC Services Enumeration Reporting 5.0 (Medium) 80 % 172.16.3.85 135/tcp Tue, Feb 18, 2025 2:32 PM UTC
Hostname Determination Reporting 0.0 (Log) 80 % 172.16.3.85 general/tcp Tue, Feb 18, 2025 2:45 PM UTC
HTTP Security Headers Detection 0.0 (Log) 80 % 172.16.3.85 5986/tcp Tue, Feb 18, 2025 2:31 PM UTC
HTTP Server Banner Enumeration 0.0 (Log) 80 % 172.16.3.85 5986/tcp Tue, Feb 18, 2025 2:31 PM UTC
HTTP Server type and version 0.0 (Log) 80 % 172.16.3.85 5986/tcp Tue, Feb 18, 2025 2:31 PM UTC
Microsoft Remote Desktop Protocol (RDP) Detection 0.0 (Log) 80 % 172.16.3.85 3389/tcp Tue, Feb 18, 2025 2:26 PM UTC
OS Detection Consolidation and Reporting 0.0 (Log) 80 % 172.16.3.85 general/tcp Tue, Feb 18, 2025 2:29 PM UTC
Services 0.0 (Log) 80 % 172.16.3.85 5986/tcp Tue, Feb 18, 2025 2:26 PM UTC
Apply to page contents
1 - 10 of 49

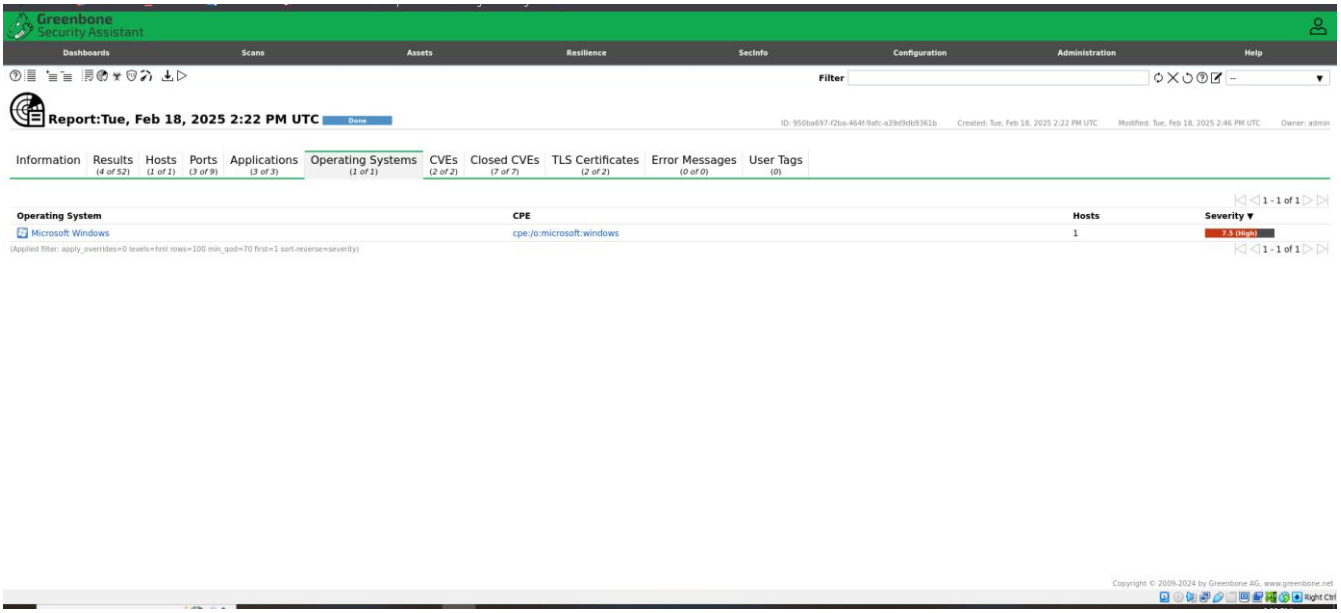

https://127.0.0.1:9392/report/950ba697-f2ba-464f-9afc-a39d9db9361b
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Greenbone Security Assistant
Dashboards
Scans
Assets
Resilience
SecInfo
Configuration
Administration
Help
Filter
Report:Tue, Feb 18, 2025 2:22 PM UTC
Done
Information
Results (4 of 52)
Hosts (1 of 1)
Ports (3 of 9)
Applications (3 of 3)
Operating Systems (1 of 1)
CVEs (2 of 2)
Closed CVEs (7 of 7)
TLS Certificates (2 of 2)
Error Messages (0 of 0)
User Tags (0)
Task Name Immediate scan of IP 172.16.3.85
Scan Time Tue, Feb 18, 2025 2:22 PM UTC - Tue, Feb 18, 2025 2:46 PM UTC
Scan Duration 0:23 h
Scan Status Done
Hosts scanned 1
Filter apply_overrides=0 levels=hml min_qod=70
Timezone Coordinated Universal Time (UTC)

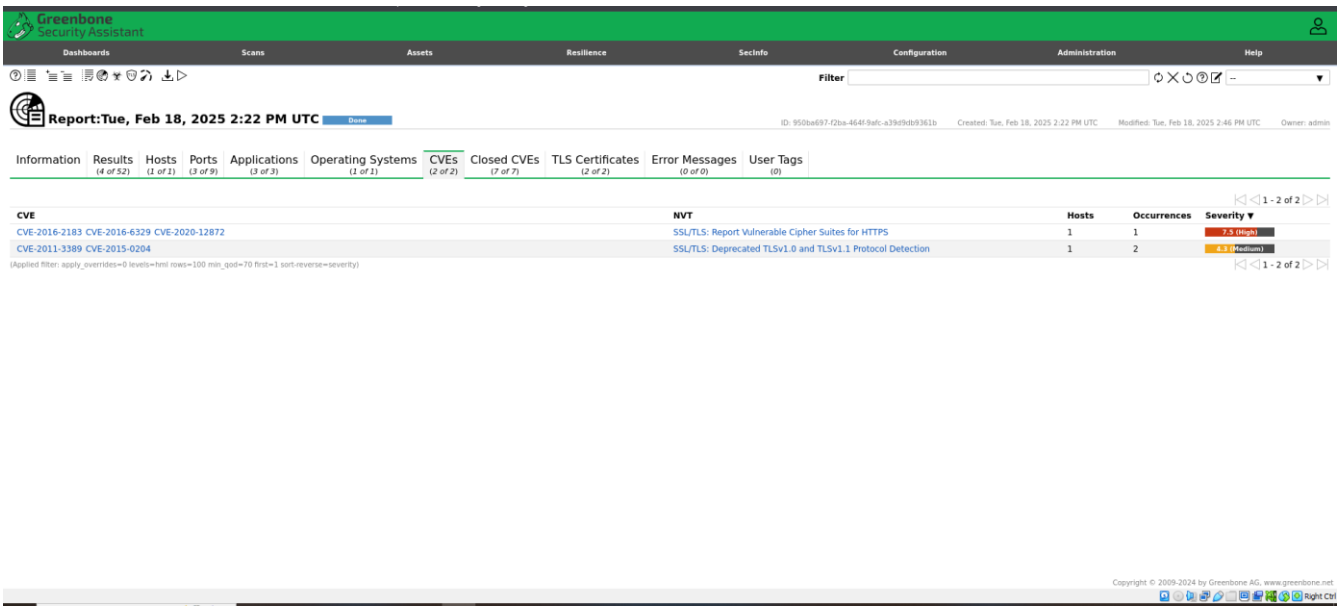

Greenbone Security Assistant
Dashboards
Scans
Assets
Resilience
SecInfo
Configuration
Administration
Help
Filter
Report:Tue, Feb 18, 2025 2:22 PM UTC
Done
Information
Results (4 of 52)
Hosts (1 of 1)
Ports (3 of 9)
Applications (3 of 3)
Operating Systems (1 of 1)
CVEs (2 of 2)
Closed CVEs (7 of 7)
TLS Certificates (2 of 2)
Error Messages (0 of 0)
User Tags (0)
1 - 4 of 4
Vulnerability
Severity
QoD
Host
IP
Name
Location
Created
SSL/TLS: Report Vulnerable Cipher Suites for HTTPS 7.5 (High) 98 % 172.16.3.85 5986/tcp Tue, Feb 18, 2025 2:30 PM UTC
DCE/RPC and MSRPC Services Enumeration Reporting 5.0 (Medium) 80 % 172.16.3.85 135/tcp Tue, Feb 18, 2025 2:32 PM UTC
SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection 4.3 (Medium) 98 % 172.16.3.85 3389/tcp Tue, Feb 18, 2025 2:30 PM UTC
SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection 4.3 (Medium) 98 % 172.16.3.85 5986/tcp Tue, Feb 18, 2025 2:30 PM UTC
1 - 4 of 4

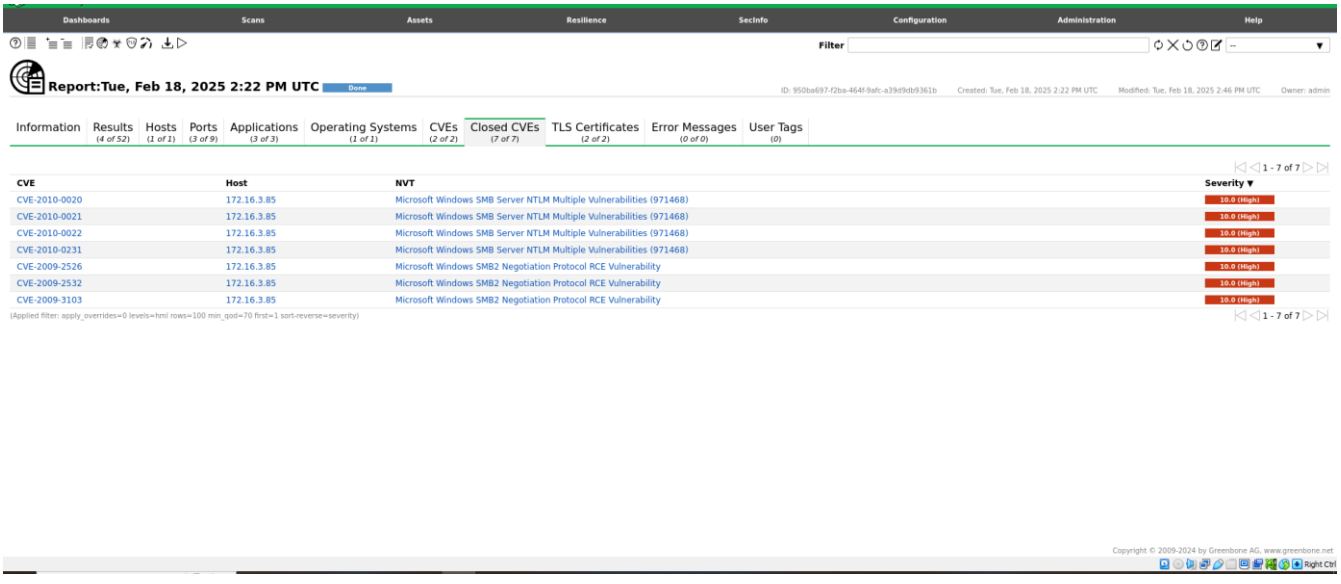

Greenbone Security Assistant
Dashboards Scans Assets Resilience SecInfo Configuration Administration Help
Filter
Report:Tue, Feb 18, 2025 2:22 PM UTC
Done
ID: 950ba697-f2ba-464f-9afc-a39d9db9361b Created: Tue, Feb 18, 2025 2:22 PM UTC Modified: Tue, Feb 18, 2025 2:46 PM UTC Owner: admin
Information Results (4 of 52) Hosts (1 of 1) Ports (3 of 9) Applications (3 of 3) Operating Systems (1 of 1) CVEs (2 of 2) Closed CVEs (7 of 7) TLS Certificates (2 of 2) Error Messages (0 of 0) User Tags (0)
1 - 1 of 1
IP Address Hostname OS Ports Apps Distance Auth Start End High Medium Low Log False Positive Total Severity
172.16.3.85 3 3 Tue, Feb 18, 2025 2:23 PM UTC Tue, Feb 18, 2025 2:45 PM UTC 1 3 0 0 0 4 7.5 (High)
(Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)
1 - 1 of 1
Copyright © 2009-2024 by Greenbone AG, www.greenbone.net

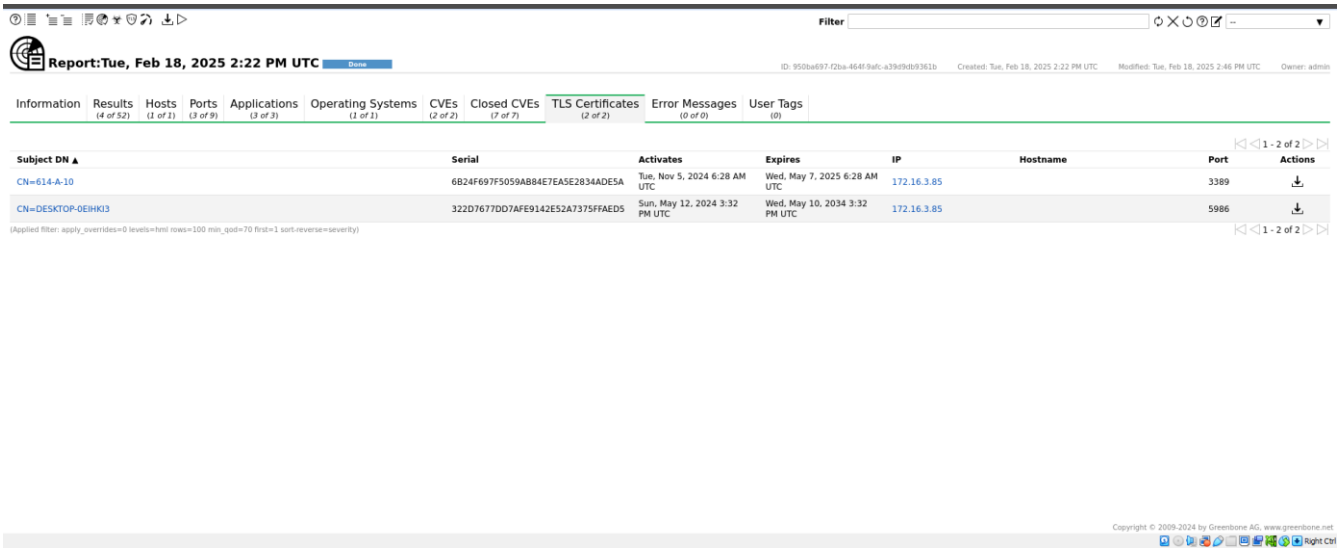

Filter
Report:Tue, Feb 18, 2025 2:22 PM UTC
Done
ID: 950ba697-f2ba-464f-9afc-a39d9db9361b Created: Tue, Feb 18, 2025 2:22 PM UTC Modified: Tue, Feb 18, 2025 2:46 PM UTC Owner: admin
Information Results (4 of 52) Hosts (1 of 1) Ports (3 of 9) Applications (3 of 3) Operating Systems (1 of 1) CVEs (2 of 2) Closed CVEs (7 of 7) TLS Certificates (2 of 2) Error Messages (0 of 0) User Tags (0)
1 - 3 of 3
Port Hosts Severity
5986/tcp 1 7.5 (High)
135/tcp 1 5.0 (Medium)
3389/tcp 1 4.3 (Medium)
(Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)
1 - 3 of 3
Copyright © 2009-2024 by Greenbone AG, www.greenbone.net


https://127.0.0.1:9392/report/950ba697-f2ba-464f-9afc-a39d9db9361b
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Greenbone Security Assistant
Dashboards Scans Assets Resilience SecInfo Configuration Administration Help
Filter
Report:Tue, Feb 18, 2025 2:22 PM UTC
Done
ID: 950ba697-f2ba-464f-9afc-a39d9db9361b Created: Tue, Feb 18, 2025 2:22 PM UTC Modified: Tue, Feb 18, 2025 2:46 PM UTC Owner: admin
Information Results (4 of 52) Hosts (1 of 1) Ports (3 of 9) Applications (3 of 3) Operating Systems (1 of 1) CVEs (2 of 2) Closed CVEs (7 of 7) TLS Certificates (2 of 2) Error Messages (0 of 0) User Tags (0)
1 - 3 of 3
Application CPE Hosts Occurrences Severity
cpe:/a:ietf:transport_layer_security:1.0 1 2 4.3 (Medium)
cpe:/a:ietf:transport_layer_security:1.1 1 2 N/A
cpe:/a:ietf:transport_layer_security:1.2 1 2 N/A
(Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)
1 - 3 of 3
Copyright © 2009-2024 by Greenbone AG, www.greenbone.net


Greenbone Security Assistant
Dashboards Scans Assets Resilience SecInfo Configuration Administration Help
Filter
Report:Tue, Feb 18, 2025 2:22 PM UTC Done
Information Results (4 of 52) Hosts (1 of 1) Ports (3 of 9) Applications (3 of 3) Operating Systems (1 of 1) CVEs (2 of 2) Closed CVEs (7 of 7) TLS Certificates (2 of 2) Error Messages (0 of 0) User Tags (0)
1 - 1 of 1
Operating System CPE Hosts Severity
Microsoft Windows cpe:/o:microsoft:windows 1 7.5 (High)
(Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)
1 - 1 of 1


Greenbone Security Assistant
Dashboards Scans Assets Resilience SecInfo Configuration Administration Help
Filter
Report:Tue, Feb 18, 2025 2:22 PM UTC Done
Information Results (4 of 52) Hosts (1 of 1) Ports (3 of 9) Applications (3 of 3) Operating Systems (1 of 1) CVEs (2 of 2) Closed CVEs (7 of 7) TLS Certificates (2 of 2) Error Messages (0 of 0) User Tags (0)
1 - 2 of 2
CVE NVT Hosts Occurrences Severity
CVE-2016-2183 CVE-2016-6329 CVE-2020-12872 SSL/TLS: Report Vulnerable Cipher Suites for HTTPS 1 1 7.5 (High)
CVE-2011-3389 CVE-2015-0204 SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection 1 2 4.3 (Medium)
(Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)
1 - 2 of 2


Dashboards Scans Assets Resilience SecInfo Configuration Administration Help
Filter
Report:Tue, Feb 18, 2025 2:22 PM UTC Done
Information Results (4 of 52) Hosts (1 of 1) Ports (3 of 9) Applications (3 of 3) Operating Systems (1 of 1) CVEs (2 of 2) Closed CVEs (7 of 7) TLS Certificates (2 of 2) Error Messages (0 of 0) User Tags (0)
1 - 7 of 7
CVE Host NVT Severity
CVE-2010-0020 172.16.3.85 Microsoft Windows SMB Server NTLM Multiple Vulnerabilities (971468) 10.0 (High)
CVE-2010-0021 172.16.3.85 Microsoft Windows SMB Server NTLM Multiple Vulnerabilities (971468) 10.0 (High)
CVE-2010-0022 172.16.3.85 Microsoft Windows SMB Server NTLM Multiple Vulnerabilities (971468) 10.0 (High)
CVE-2010-0231 172.16.3.85 Microsoft Windows SMB Server NTLM Multiple Vulnerabilities (971468) 10.0 (High)
CVE-2009-2526 172.16.3.85 Microsoft Windows SMB2 Negotiation Protocol RCE Vulnerability 10.0 (High)
CVE-2009-2532 172.16.3.85 Microsoft Windows SMB2 Negotiation Protocol RCE Vulnerability 10.0 (High)
CVE-2009-3103 172.16.3.85 Microsoft Windows SMB2 Negotiation Protocol RCE Vulnerability 10.0 (High)
(Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)
1 - 7 of 7



LATEST APPLICATIONS:

- Enterprise IT Security
- Cloud Security
- IoT (Internet of Things) Security
- Compliance Auditing
- Penetration Testing and Ethical Hacking
- Managed Security Service Providers (MSSPs)
- Incident Response and Forensics
- Security Automation Platforms
- Educational Institutions and Training
- Government and Defense Organizations

LEARNING OUTCOME:

In this practical, we use OpenVAS to perform vulnerability scans, analyze results, and apply security measures to protect systems from threats.

REFERENCES:

5. Open VAS: https://www.openvas.org/
6. GFG: https://www.geeksforgeeks.org/security-assessment-openvas/
7. Green bone: https://greenbone.github.io/docs/latest/22.4/kali/index.html

PRACTICAL: 7

AIM:

A digital forensics team is investigating a case involving encrypted files and documents critical to their investigation. The team must recover the passwords for various applications, including archived files and PDF documents, using tools like Passware Password Recovery Kit Forensic, Advanced Archive Password Recovery, and Advanced PDF Password Recovery. Recover application passwords using specialized tools to demonstrate password recovery techniques and evaluate the efficiency of each tool in real-world scenarios.

THEORY:

Passware Password Recovery Kit Forensic:

Passware is one of the most well-known tools used in digital forensics for password recovery. It supports a wide range of file formats, including encrypted Office documents (Word, Excel, PowerPoint), PDFs, and more. This tool uses several attack methods to recover passwords:

- **Brute-Force Attack**: This method attempts every possible combination of characters to find the correct password. It can be effective for short or simple passwords but is time-consuming for longer, complex passwords.
- **Dictionary Attack**: This attack uses a pre-compiled list of common passwords and attempts to match the password against entries in the list. It is faster than brute-force and works well when the password is a common word or phrase.
- **Mask Attack**: If the investigator has partial knowledge of the password (e.g., the password's length or the characters it contains), they can use a mask attack to limit the number of combinations attempted, significantly speeding up the process.

Passware is also capable of **GPU acceleration**, which allows the use of graphics cards (GPUs) to speed up password recovery, especially for more complex encryption.

Advanced Archive Password Recovery (AAPR):

AAPR specializes in recovering passwords from encrypted archive files, such as ZIP, RAR, and 7z formats. Archive files are often used for compressing multiple files into a single package, making them a common target in digital investigations.

- **Brute-Force Attack**: AAPR will try every possible combination of characters in an effort to find the password.
- **Dictionary Attack**: Similar to Passware, AAPR can use a wordlist to attempt known passwords, speeding up the recovery process for simple passwords.
- **Mask Attack**: If partial knowledge of the password is available (such as length or known characters), AAPR can use a mask attack to reduce the recovery time.

AAPR is particularly useful for encrypted archive files and can handle various compression formats. It is known for its speed and efficiency when compared to other password recovery tools.

Advanced PDF Password Recovery (APDFPR):

PDF documents are commonly used to store critical data, and they are often protected by passwords. APDFPR specializes in recovering passwords for encrypted PDF files. Similar to the other tools mentioned, it offers several attack methods:

- **Brute-Force Attack**: APDFPR attempts all possible combinations until the correct password is found.
- **Dictionary Attack**: The tool uses a wordlist of common passwords, which can be particularly effective if the password is a commonly used word or phrase.
- **Mask Attack**: If the investigator has knowledge about the length of the password or other patterns (such as specific characters), a mask attack can significantly reduce the time needed to crack the password.

APDFPR also supports the removal of **password restrictions** on PDF files, such as restrictions on printing or copying the document, even if the password to open the file is not recovered.

CODE:

N/A

OUTPUT:

Figure 90: Start Installing of Advanced Archive Password Recovery Setup

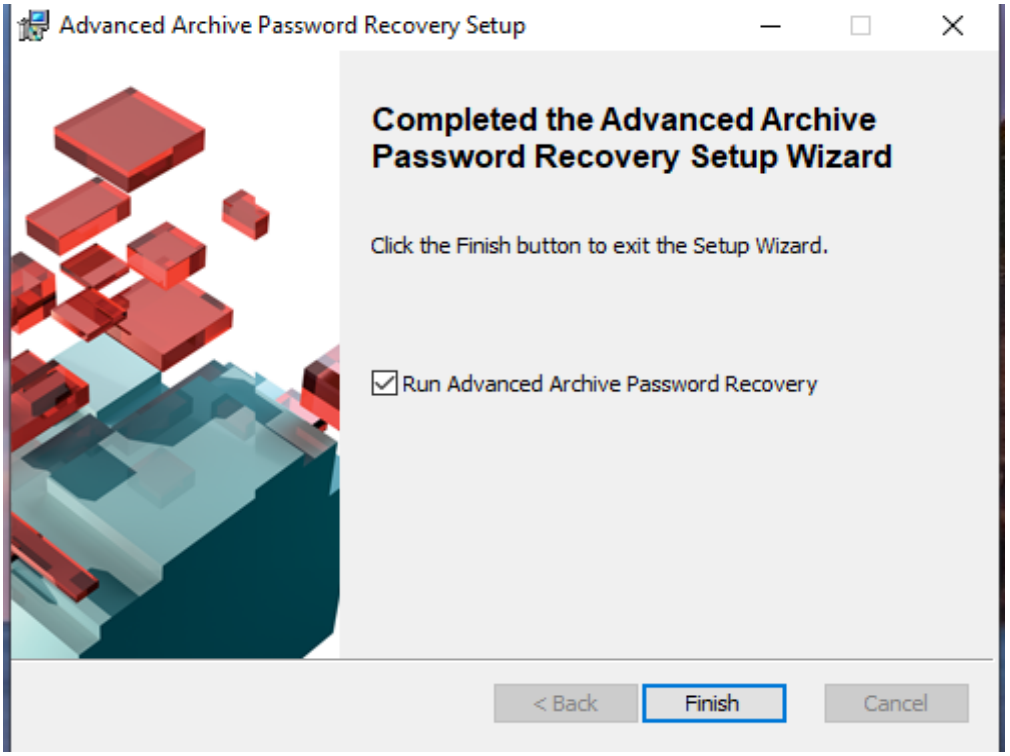


Figure 91: Complete Installing of Advanced Archive Password Recovery Setup

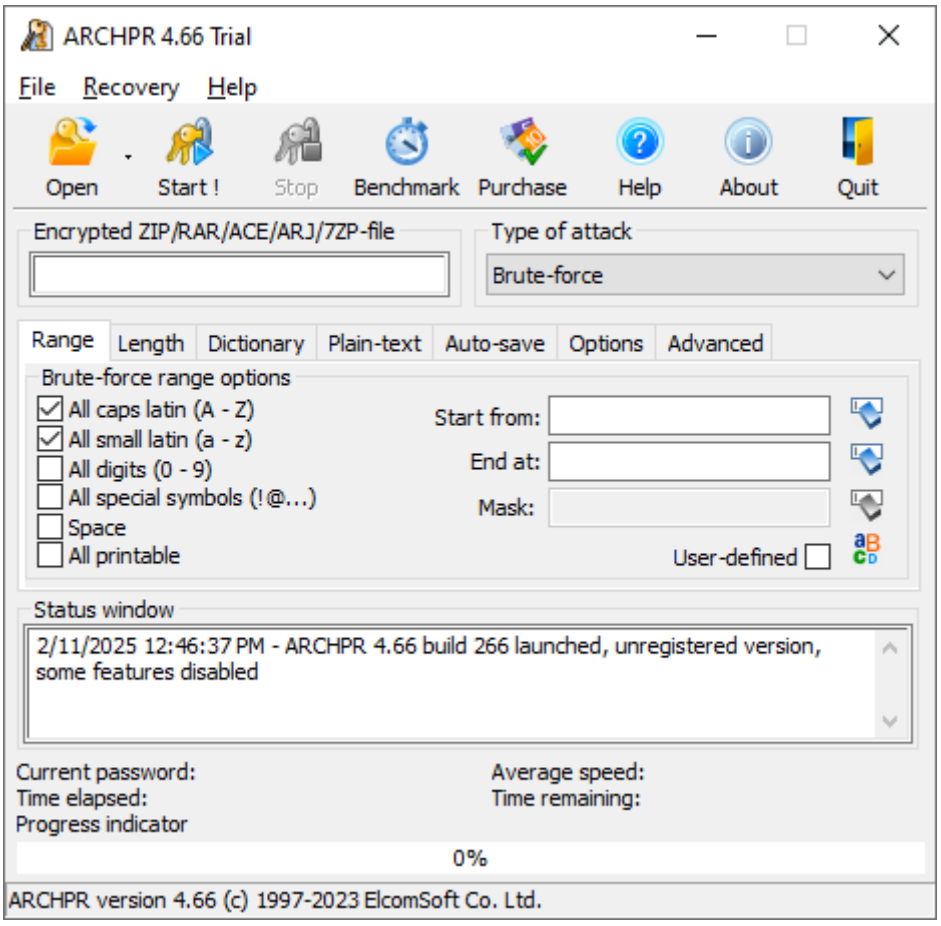


Figure 92: Open Advanced Archive Password Recovery Setup

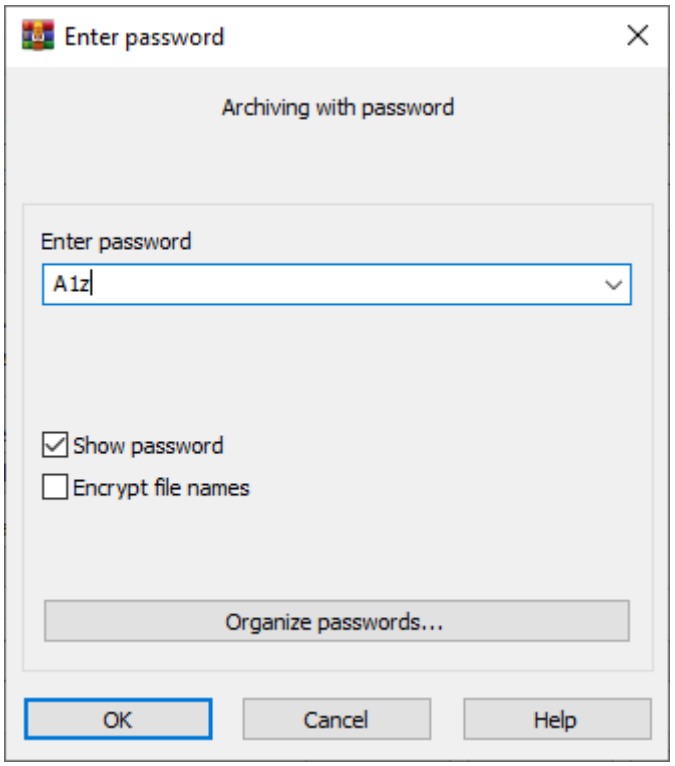


Figure 93:Set password to .rar file

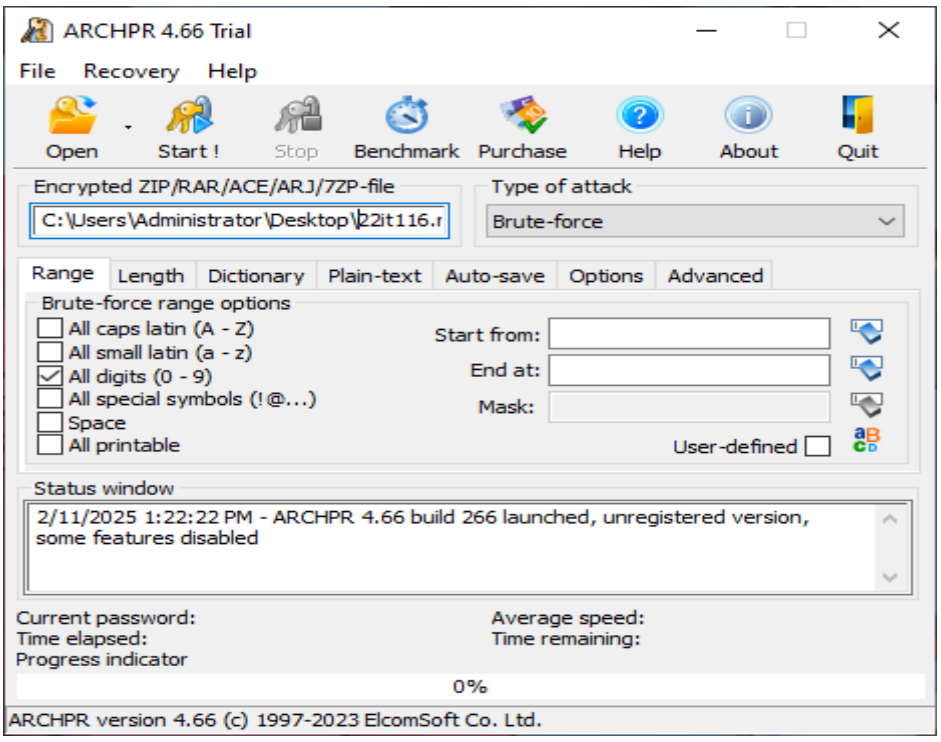


Figure 94:Set Encrypted .rar file to open it and set another parameter

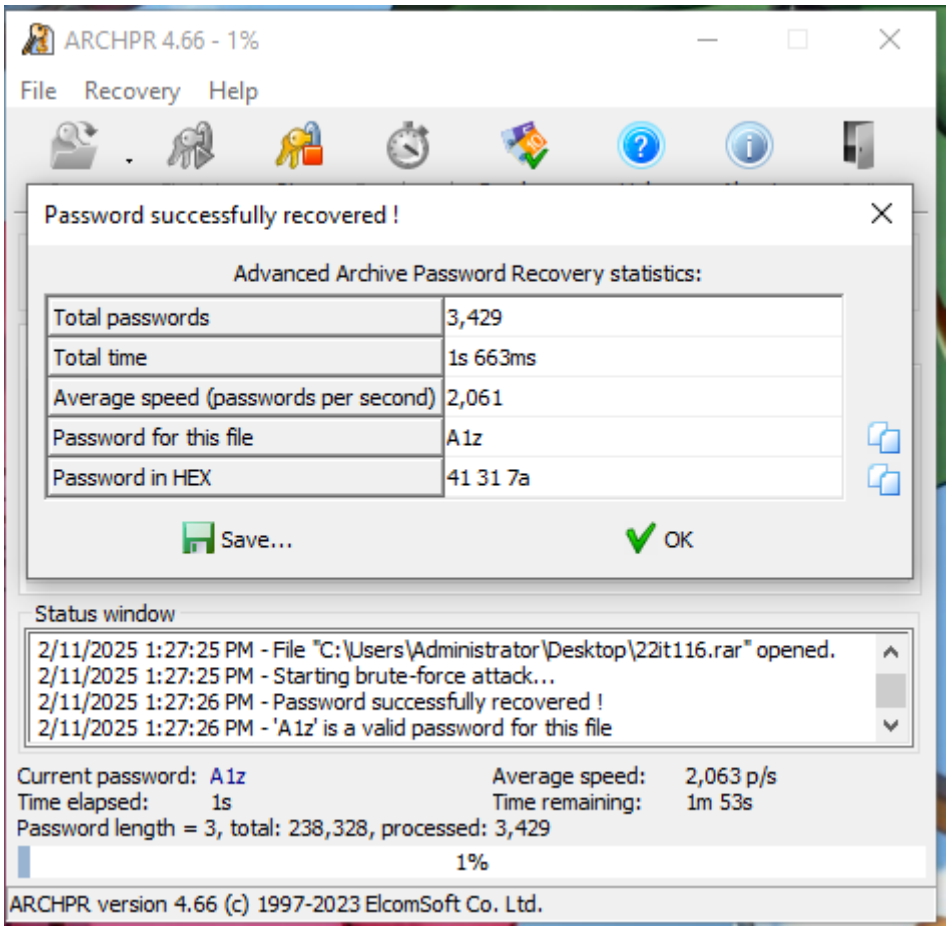


Figure 95: We got our password from Advanced Archive Password Recovery Setup

Figure 96: Start installing Advanced PDF Password Recovery Setup

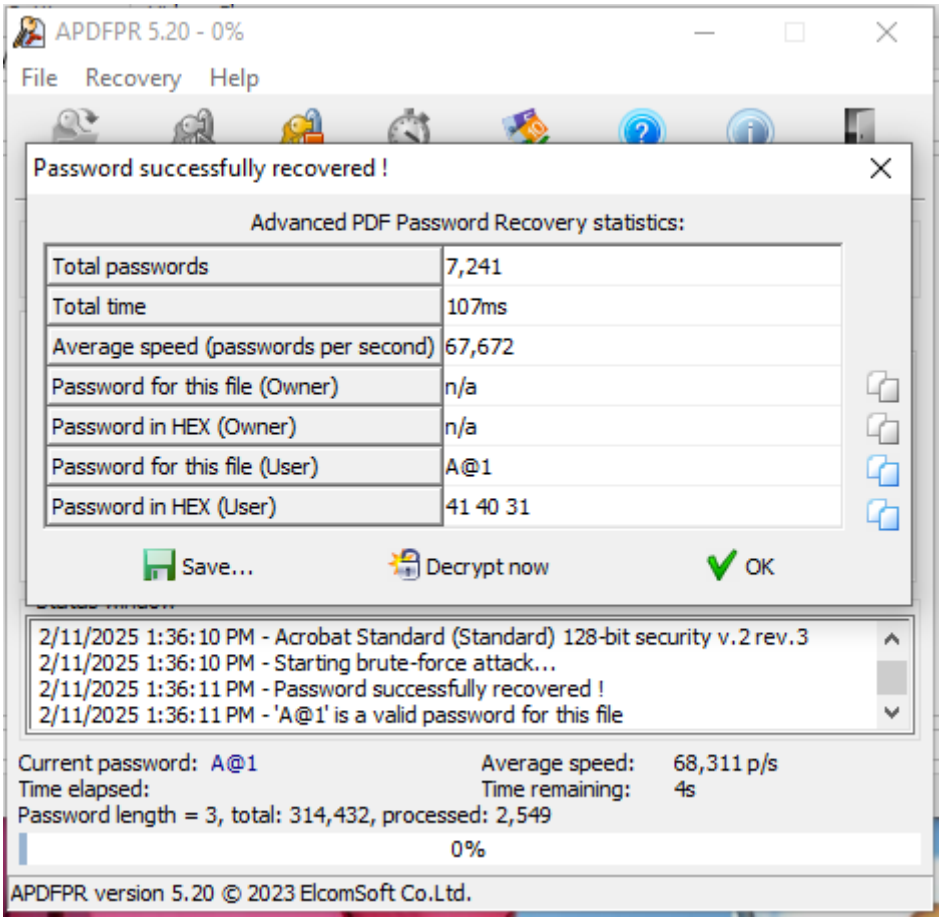


Figure 97:: We got our password from Advanced PDF Password Recovery

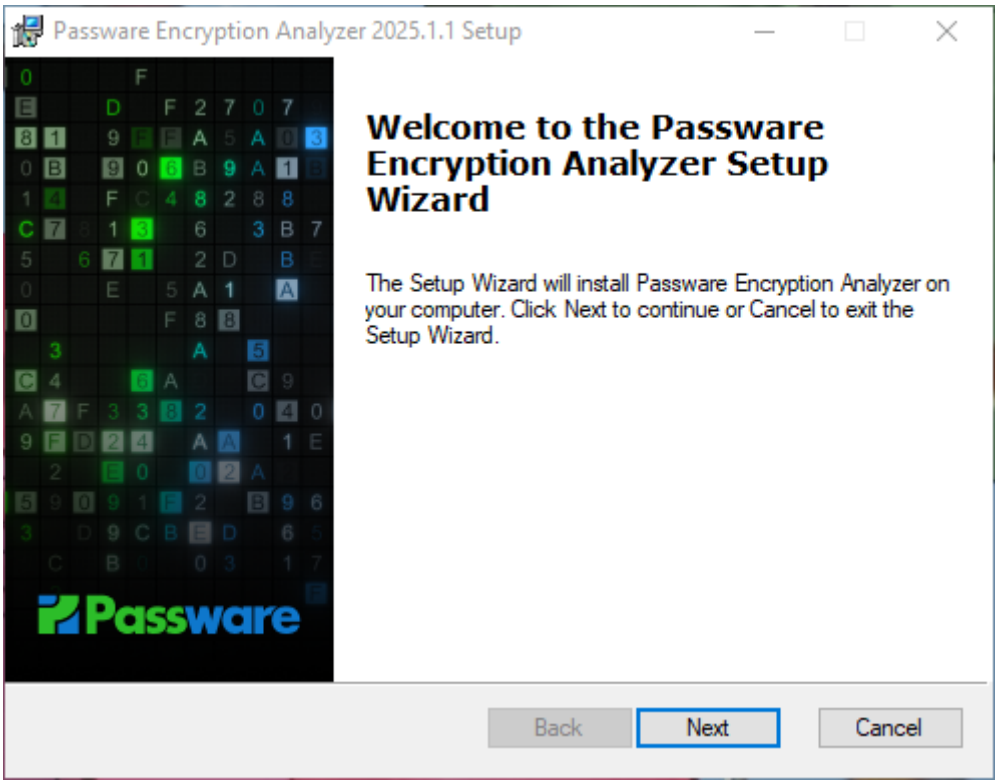


Figure 98:Start Passware Encryption Analyzer

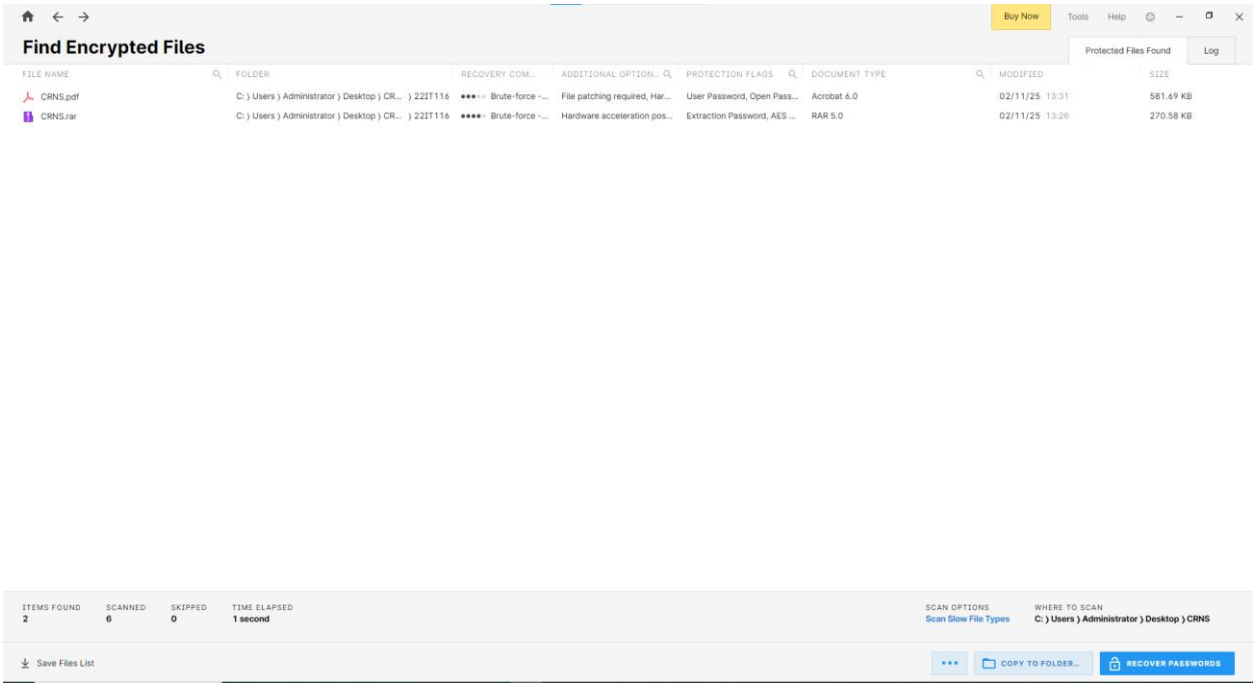


Figure 99:Scan encrypted files

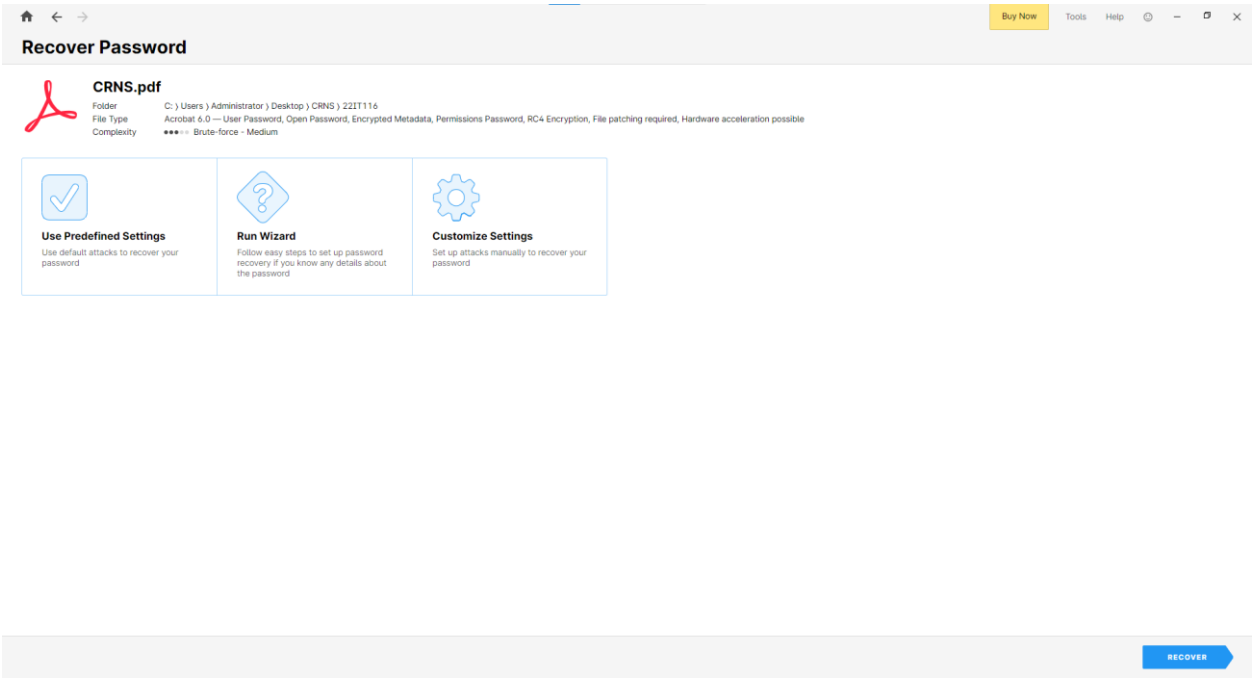


Figure 100:Click recover password for pdf file

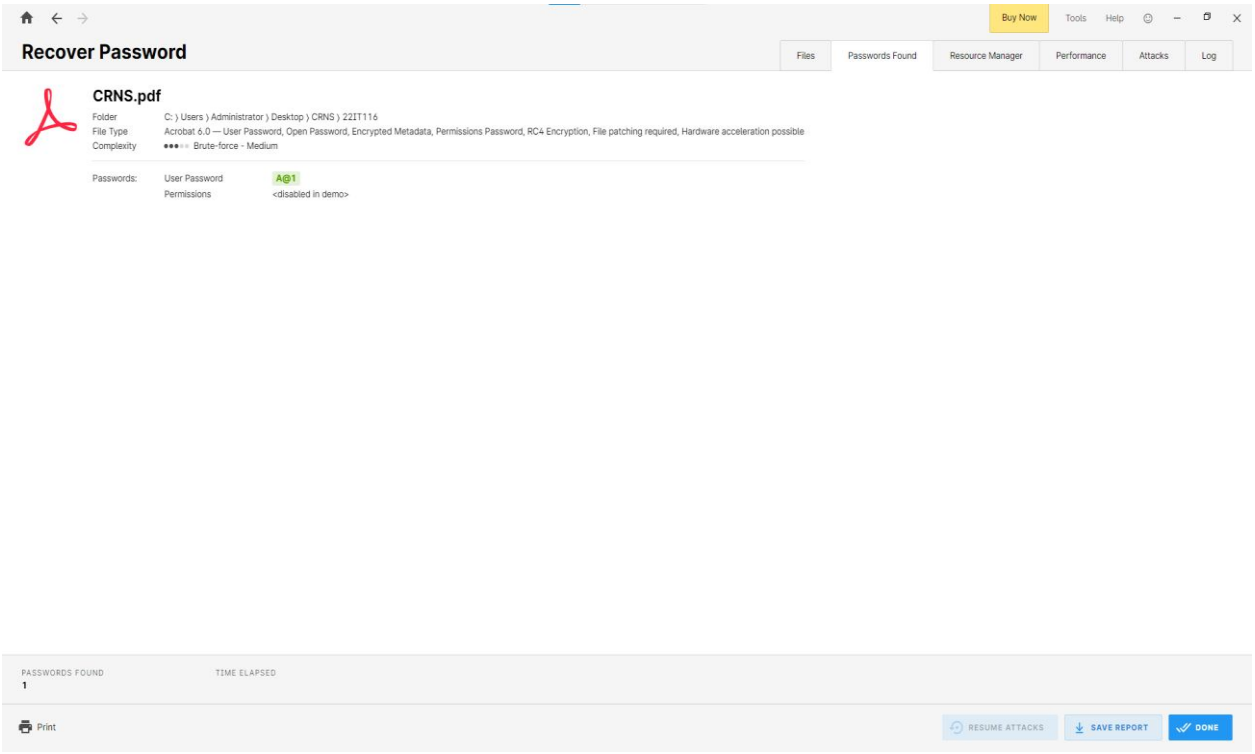


Figure 101: Apply Use Predefined Settings attack

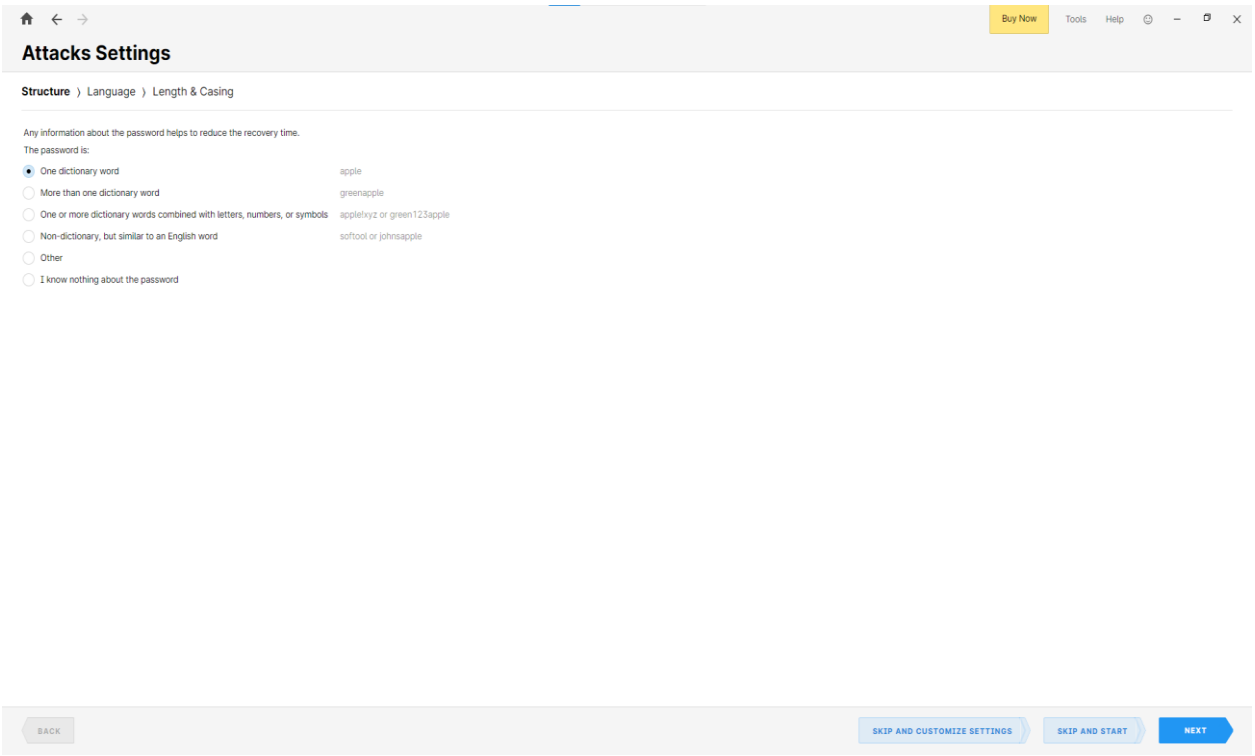


Figure 102:Apply Run Wizard

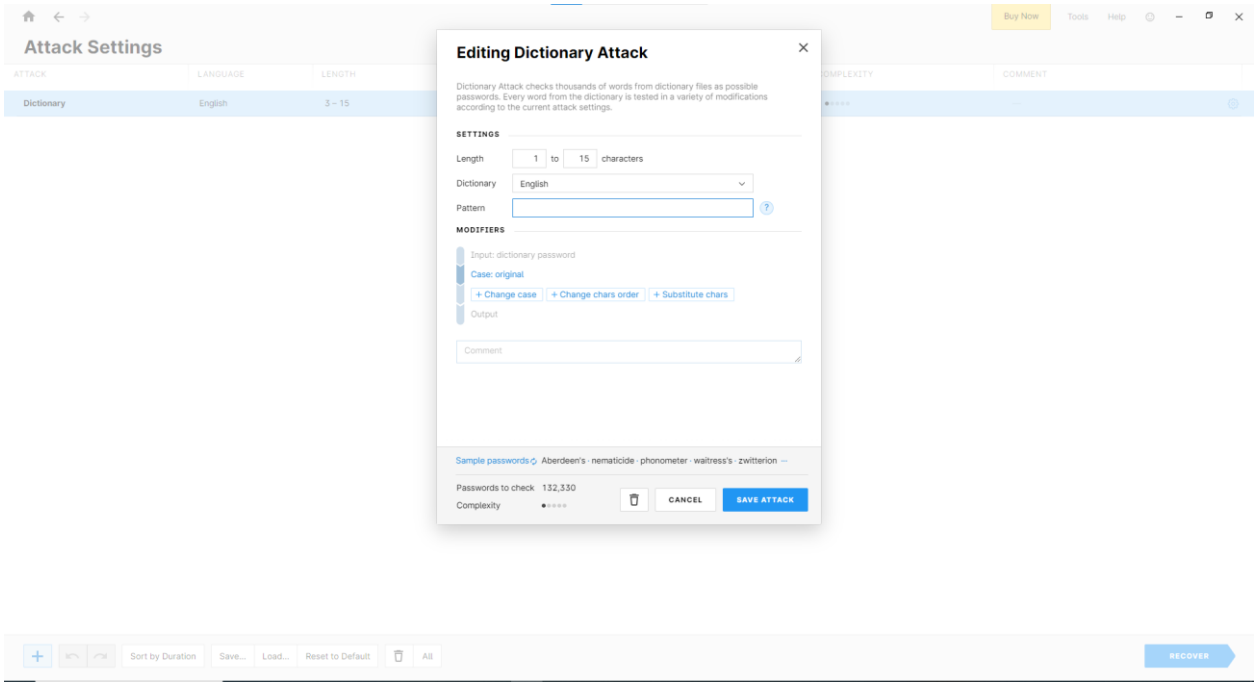


Figure 103:Apply Customize Settings

LATEST APPLICATIONS:

- Hashcat
- Passware Kit 2025 v1
- Kon-Boot
- Active@ Password Changer
- NirSoft Utilities

LEARNING OUTCOME:

In this practical, I gained a deeper understanding of different password recovery methods, including brute-force, dictionary, and mask attacks, and their effectiveness with various encrypted files. Using tools like Passware, AAPR, and APDFPR, I explored how each method performed depending on password complexity and file type.

REFERENCES:

6. Advanced PDF Password Recovery : https://www.elcomsoft.com/apdfpr.html
7. Advanced Archive Password Recovery: https://www.elcomsoft.com/archpr.html
8. Passware Password Recovery Kit Forensic: http://passware.com/
9. ChatGPT : https://chatgpt.com/

PRACTICAL: 8

AIM:

An organization has transferred sensitive financial documents between departments over a network and wants to ensure that the files remain unchanged during transit. The IT team is tasked with using HashCalc and MD5 Calculator to confirm the integrity of these files by comparing hash values generated before and after the transfer. Verify the integrity of critical data files by generating and comparing hash values using HashCalc and MD5 Calculator, ensuring no tampering or corruption has occurred during transmission or storage.

THEORY:

The MD5 (message-digest algorithm) hashing algorithm is a one-way cryptographic function that accepts a message of any length as input and returns as output a fixed-length digest value to be used for authenticating the original message.

The MD5 hash function was originally designed for use as a secure cryptographic hash algorithm for authenticating digital signatures. But MD5 has been deprecated for uses other than as a noncryptographic checksum to verify data integrity and detect unintentional data corruption.

Although originally designed as a cryptographic message authentication code algorithm for use on the internet, MD5 hashing is no longer considered reliable for use as a cryptographic checksum because security experts have demonstrated techniques capable of easily producing MD5 collisions on commercial off-the-shelf computers. An encryption collision means two files have the same hash. Hash functions are used for message security, password security, computer forensics and cryptocurrency.

Ronald Rivest, founder of RSA Data Security LLC and professor at Massachusetts Institute of Technology, designed MD5 in 1991 as an improvement to a prior message-digest algorithm, MD4. Describing it in Internet Engineering Task Force (IETF) Request for Comments (RFC) 1321, "The MD5 Message-Digest Algorithm," he wrote:

The algorithm takes as input a message of arbitrary length and produces as output a 128-bit 'fingerprint' or 'message digest' of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be 'compressed' in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA.

IETF suggests MD5 hashing can still be used for integrity protection, noting: "Where the MD5 checksum is used in line with the protocol solely to protect against errors, an MD5 checksum is still an acceptable use." However, it added that "any application and protocol that employs MD5 for any purpose needs to clearly state the expected security services from their use of MD5."

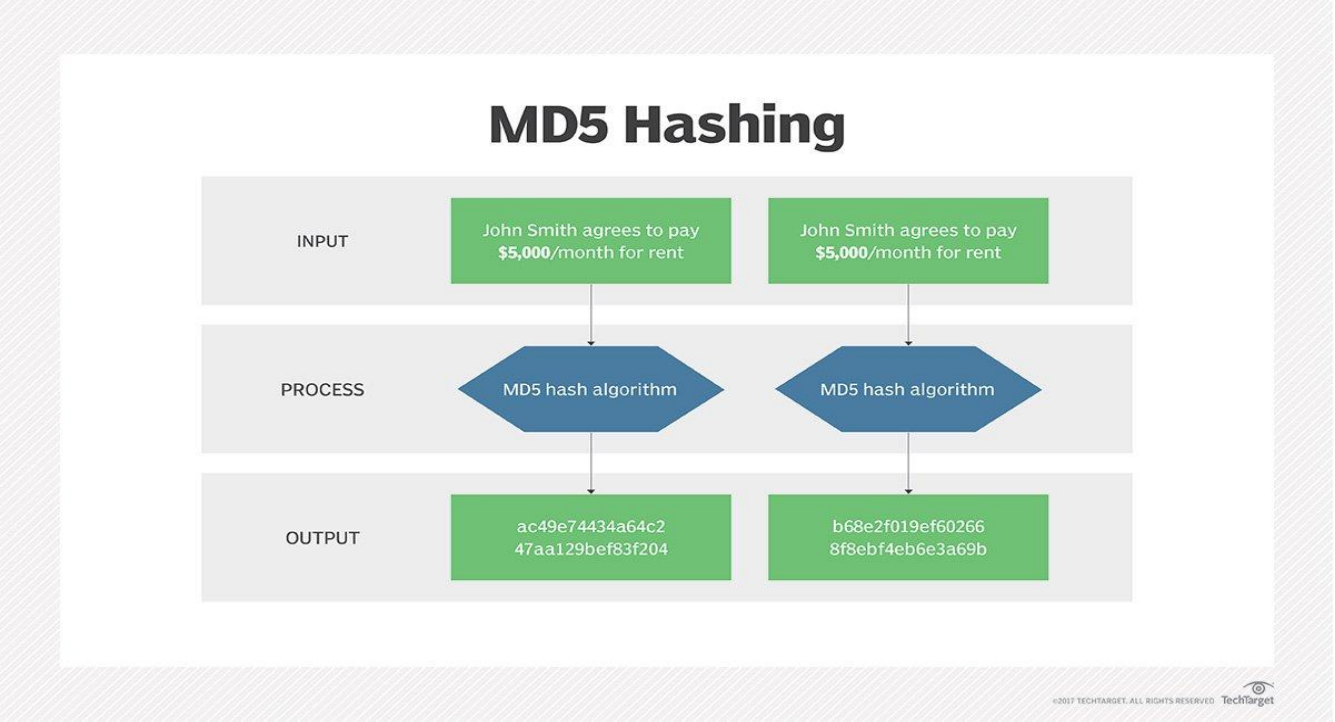


Figure 104: MD5

Message digests, also known as *hash functions*, are one-way functions; they accept a message of any size as input and produce as output a fixed-length message digest.

MD5 is the third message-digest algorithm Rivest created. MD2, MD4 and MD5 have similar structures, but MD2 was optimized for 8-bit machines, in comparison with the two later algorithms, which are designed for 32-bit machines. The MD5 algorithm is an extension of MD4, which the critical review found to be fast but potentially insecure. In comparison, MD5 is not quite as fast as the MD4 algorithm, but offered much more assurance of data security.

The MD5 message-digest hashing algorithm processes data in 512-bit strings, broken down into 16 words composed of 32 bits each. The output from MD5 is a 128-bit message-digest value.

Computation of the MD5 digest value is performed in separate stages that process each 512-bit block of data along with the value computed in the preceding stage. The first stage begins with the message-digest values initialized using consecutive hexadecimal numerical values. Each stage includes four message-digest passes, which manipulate values in the current data block and values processed from the previous block. The final value computed from the last block becomes the MD5 digest for that block.

The goal of any message-digest function is to produce digests that appear to be random. To be considered cryptographically secure, the hash function should meet two requirements:

1. It is impossible for an attacker to generate a message matching a specific hash value.
2. It is impossible for an attacker to create two messages that produce the same hash value.

MD5 hashes are no longer considered cryptographically secure methods and should not be used for cryptographic authentication, according to IETF.

CODE:

N/A

OUTPUT:

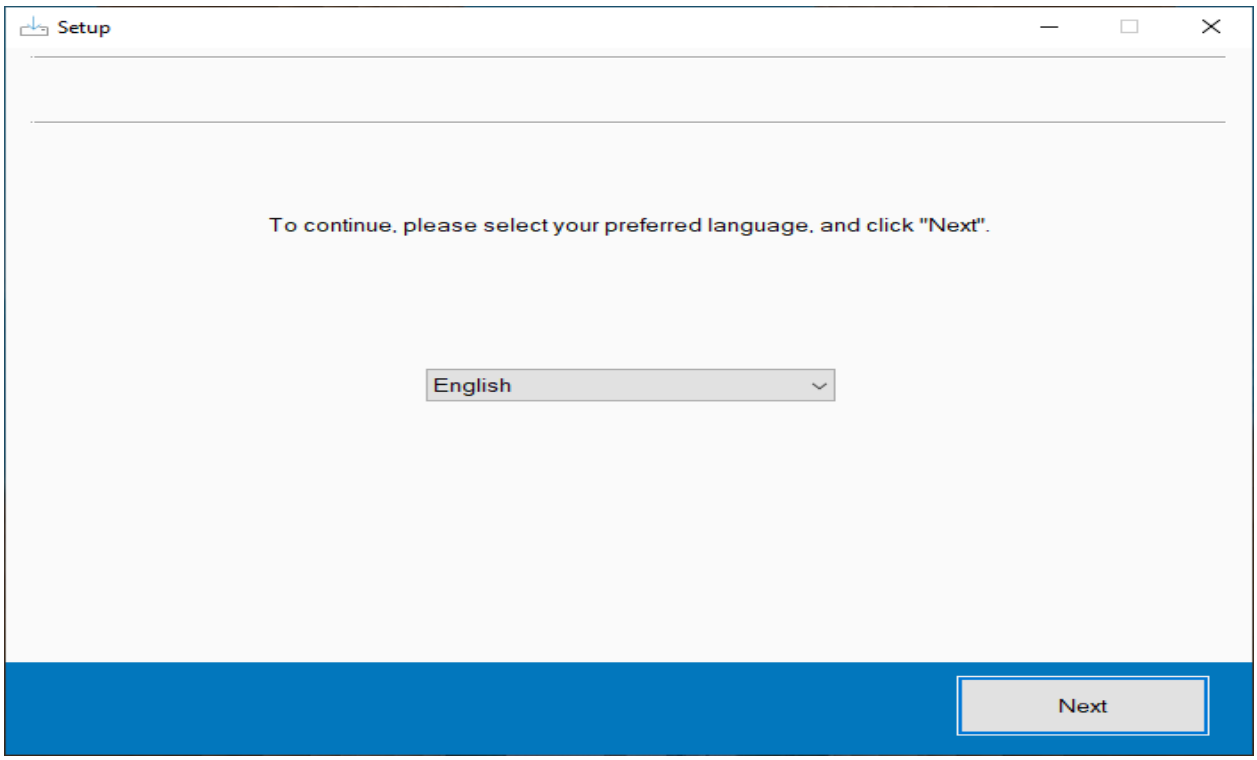


Figure 105:Start setting up HashCalc

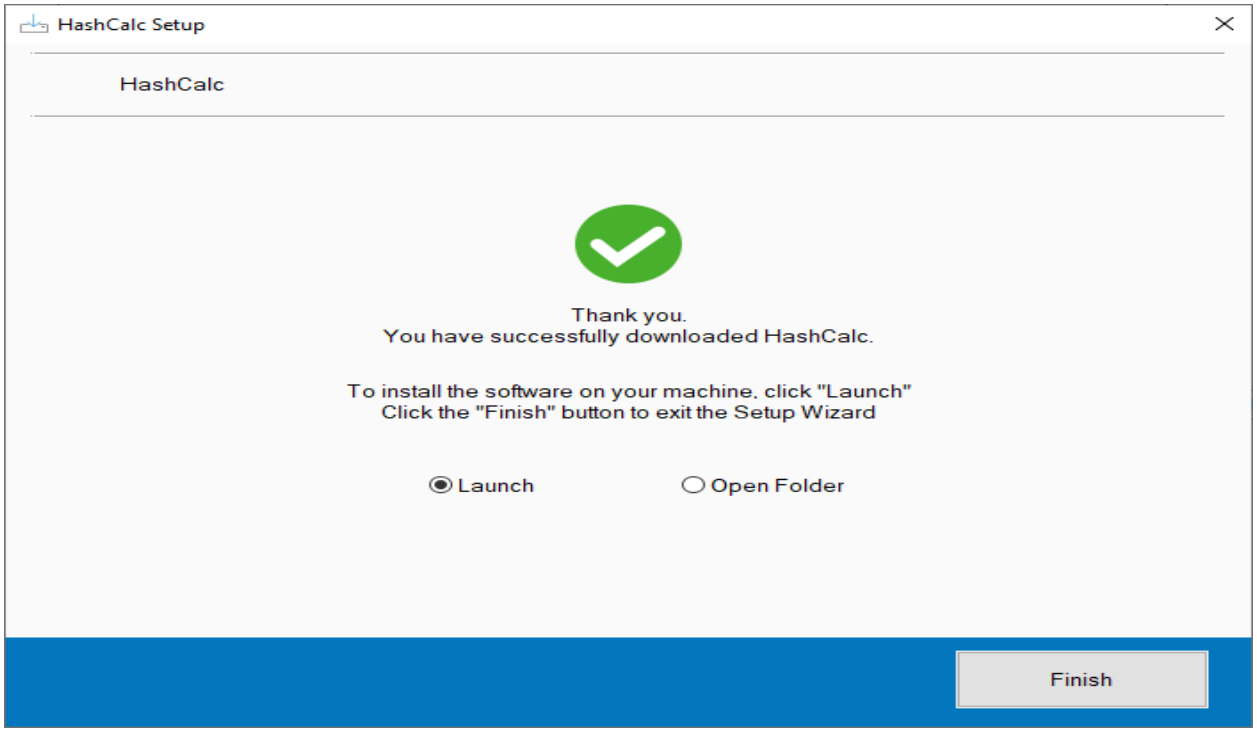


Figure 106:Finish setup of HashCalc and launch it

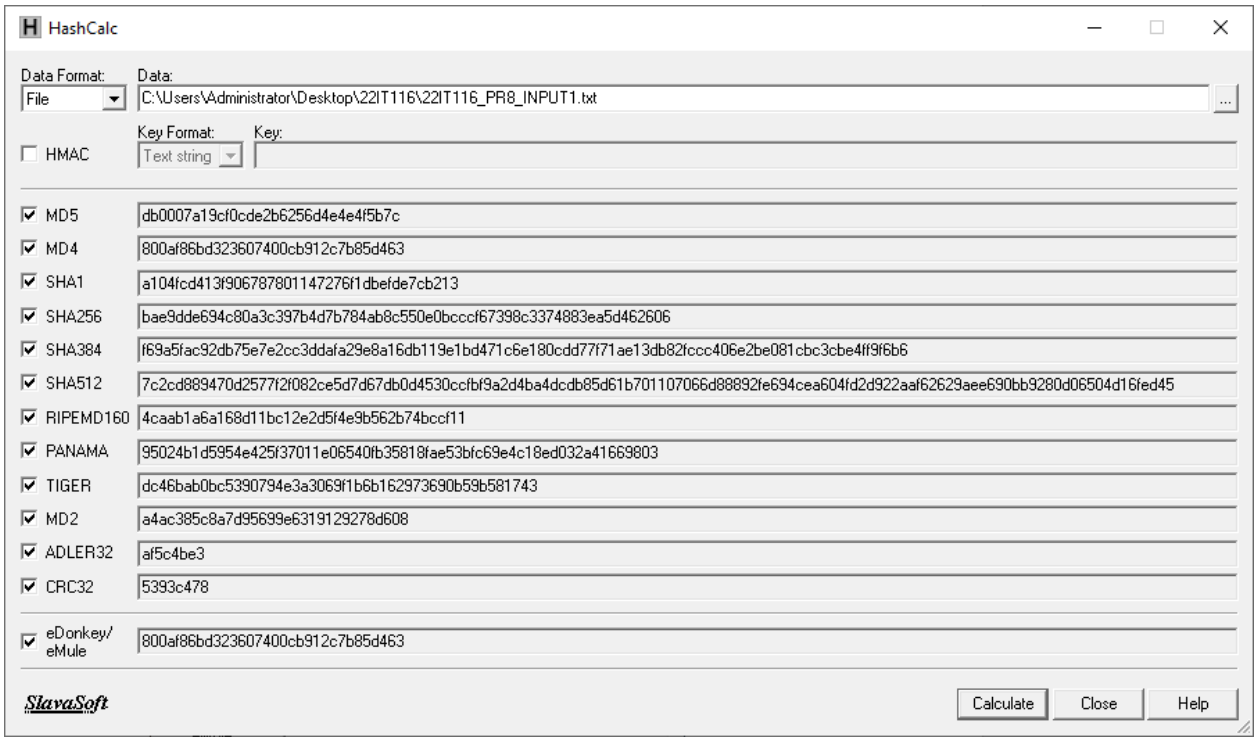


Figure 107:Hash values for Input1.txt file

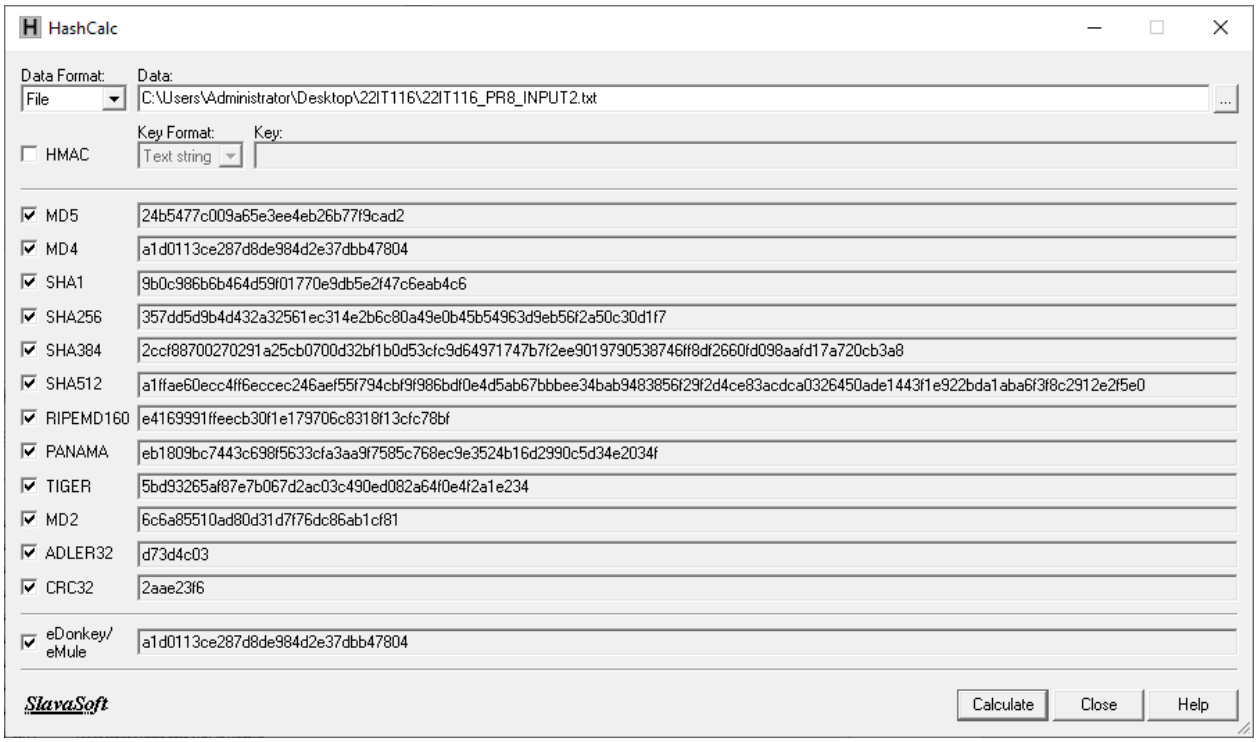


Figure 108:Hash values for Input2.txt file

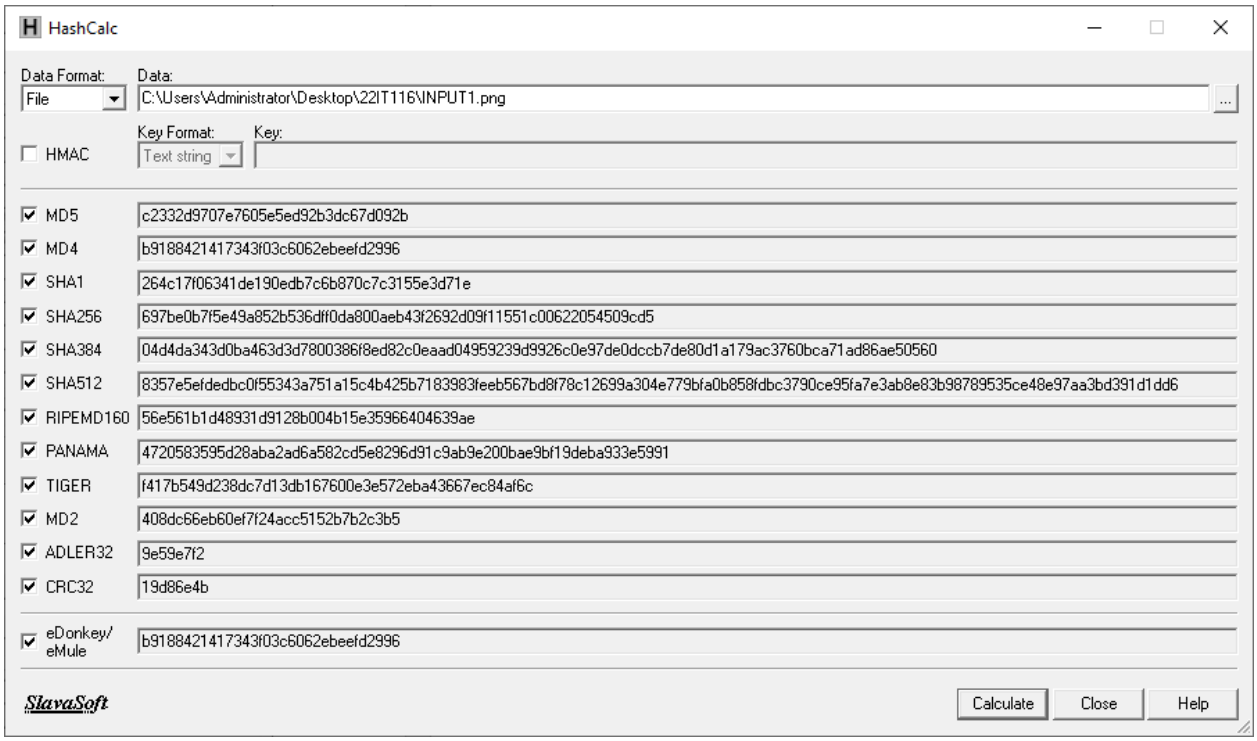


Figure 109:Hash values for INPUT1.png photo

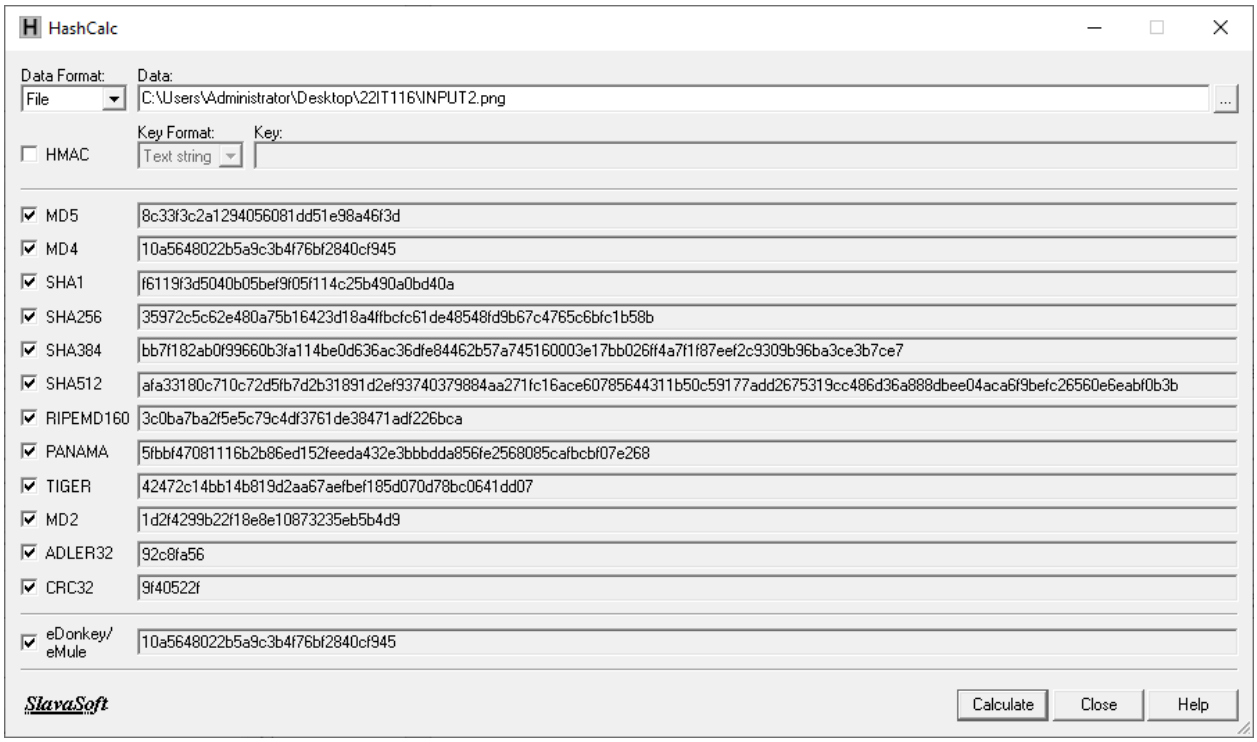


Figure 110:Hash values for Input2.png photo

LATEST APPLICATIONS:

- Hashi Corp Vault
- Acronis True Image
- VeraCrypt
- GnuPG (GPG)
- OpenSSL
- AWS S3

LEARNING OUTCOME:

In this practical, I will learn how to verify the integrity of sensitive financial documents transferred between departments using HashCalc and MD5 Calculator. I will generate hash values before and after the transfer, comparing them to ensure the files remain unchanged. Through this process, I will understand how hashing algorithms help detect data tampering or corruption, practice secure file transfer methods, and troubleshoot integrity issues, ensuring critical files are securely transmitted and stored.

REFERENCES:

8. HashCalc: https://download.cnet.com/hashcalc/3000-2250_4-10130770.html
9. MD5: https://www.techtarget.com/searchsecurity/definition/MD5
10. ChatGPT: https://chatgpt.com/

PRACTICAL: 9

AIM:

A cybersecurity training program is preparing students to understand and apply cryptographic techniques in real-world scenarios. The students are tasked with using CrypTool to study various cryptographic algorithms and simulate encryption and decryption processes to gain practical insights into data security mechanisms. To explore and analyses cryptographic algorithms using CrypTool to understand their functionality, strengths, and weaknesses in securing sensitive information.

THEORY:

Cryptography is the process of hiding or coding information so that only the person a message was intended for can read it. The art of cryptography has been used to code messages for thousands of years and continues to be used in bank cards, computer passwords, and ecommerce.

Modern cryptography techniques include algorithms and ciphers that enable the encryption and decryption of information, such as 128-bit and 256-bit encryption keys. Modern ciphers, such as the Advanced Encryption Standard (AES), are considered virtually unbreakable.

Monoalphabetic Cipher is a part of the substitution technique in which a single cipher alphabet is used per message (mapping is done from plain alphabet to cipher alphabet). Monoalphabetic cipher converts plain text into cipher text and re-convert a cipher text to plain text. Monoalphabetic Cipher eliminates the brute-force techniques for cryptanalysis. Moreover, the cipher line can be a permutation of the 26 alphabetic characters.

In Cryptography, various encryption techniques are used to provide data security. The classical Encryption technique is categorized into two divisions:

1. Substitution Cipher Technique

- Caesar Cipher
- Monoalphabetic Cipher
- Polyalphabetic Cipher
- Playfair Cipher
- Hill Cipher
- One-time Pad

2. Transposition Cipher Technique

The techniques which come under this division are as follows -

- Rail Fence.
- Row Column Transposition.

Types of Monoalphabetic Substitution Ciphers

Additive Cipher

It is also Known as Shift Cipher which shifts plain text to form Cipher-text.

- Mathematical Expression:
    - For Encryption: C=(P + K) mod 26 where 'P' is the character in plain text, 'K' is the key, and 'C' is the Cipher.
    - For Decryption: P=(C-K) mod 26.
- Example : Input : P= GEEKS ,Key= 4 . Output : C=KIIOW

Caesar Cipher

A type of Addictive Cipher but the value of key is always '3' here.

- Mathematical Expression:
    - For Encryption: C=(P + K) mod 26 where 'P' is the character in plain text, 'K' is the key, and 'C' is the Cipher.
    - For Decryption: P=(C-K) mod 26
- Example: Input: P=GEEKS , Output : C=JHHNV

Multiplicative Cipher

Letters are changed here using a multiplication key.

- Mathematical Expression:
    - For Encryption: C=(P * K) mod 26 where, 'P' is the character in plain text, 'K' is the key, and 'C' is the Cipher.
- Example: Input : P=VMH , Key= 3 . Output : C=HEL

Affine Cipher

A mathematical function is used to convert plain text into cipher text.

- Mathematical Expression:
    - For Encryption : C=(P * K1+K2) mod 26 where, 'P' is the character in plain text, 'K1' is the multiplicative key, 'K2' is the additive key and 'C' is Cipher.
    - For Decryption : P=((C-K2)/ K1) mod 26.
- Example : Input : P=ARM , Key1=3,Key2=5 . Output : C=HEL

Hill Cipher

Hill cipher is a polygraphic substitution cipher based on linear algebra. Each letter is represented by a number modulo 26. Often the simple scheme A = 0, B = 1, …, Z = 25 is used, but this is not an essential feature of the cipher. To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n × n matrix, against modulus 26. To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption. The matrix used for encryption is the cipher key, and it should be chosen randomly from the set of invertible n × n matrices (modulo 26).

Examples:

Input : Plaintext: ACT

Key: GYBNQKURP

Output : Ciphertext: POH

Playfair cipher, type of substitution cipher used for data encryption.

In cryptosystems for manually encrypting units of plaintext made up of more than a single letter, only digraphs (pairs of letters) were ever used. By treating digraphs in the plaintext as units rather than as single letters, the extent to which the raw frequency distribution survives the encryption process can be lessened but not eliminated, as letter pairs are themselves highly correlated. The best-known digraph substitution cipher is the Playfair, invented in 1854 by Sir Charles Wheatstone but championed at the British Foreign Office by Lyon Playfair, the first Baron Playfair of St. Andrews. Below is an example of a Playfair cipher, solved by Lord Peter Wimsey in Dorothy L. Sayers's *Have His Carcase* (1932). Here, the mnemonic aid used to carry out the encryption is a 5 × 5-square matrix containing the letters of the alphabet (I and J are treated as the same letter). A key word, MONARCHY in this example, is filled in first, and the remaining unused letters of the alphabet are entered in their lexicographic order:

M	**O**	**N**	**A**	**R**
C	**H**	**Y**	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Plaintext digraphs are encrypted with the matrix by first locating the two plaintext letters in the matrix. They are (1) in different rows and columns; (2) in the same row; (3) in the same column; or (4) alike. The corresponding encryption (replacement) rules are the following:

1. When the two letters are in different rows and columns, each is replaced by the letter that is in the same row but in the other column; i.e., to encrypt WE, W is replaced by U and E by G.
2. When A and R are in the same row, A is encrypted as R and R (reading the row cyclically) as M.
3. When I and S are in the same column, I is encrypted as S and S as X.
4. When a double letter occurs, a spurious symbol, say Q, is introduced so that the MM in SUMMER is encrypted as NL for MQ and CL for ME.

5. An X is appended to the end of the plaintext if necessary to give the plaintext an even number of letters.

Encrypting the familiar plaintext example using Sayers's Playfair array yields:

```
Plaintext: WE ARE DISCOVERED  SAVE YOURSELFX
Cipher:    UG RMK CSXHMUFMKB  TOXG CMVATLUIV
```

If the frequency distribution information were totally concealed in the encryption process, the ciphertext plot of letter frequencies in Playfair ciphers would be flat. It is not. The deviation from this ideal is a measure of the tendency of some letter pairs to occur more frequently than others and of the Playfair's row-and-column correlation of symbols in the ciphertext—the essential structure exploited by a cryptanalyst in solving Playfair ciphers. The loss of a significant part of the plaintext frequency distribution, however, makes a Playfair cipher harder to cryptanalyze than a monoalphabetic cipher.

substitution cipher, data encryption scheme in which units of the plaintext (generally single letters or pairs of letters of ordinary text) are replaced with other symbols or groups of symbols.

The ciphertext symbols do not have to be the same as the plaintext characters in a substitution cipher, as illustrated in Sir Arthur Conan Doyle's *Adventure of the Dancing Men* (1903), where Sherlock Holmes solves a monoalphabetic substitution cipher in which the ciphertext symbols are stick figures of a human in various dancelike poses.

The simplest of all substitution ciphers are those in which the cipher alphabet is merely a cyclical shift of the plaintext alphabet. Of these, the best-known is the Caesar cipher, used by Julius Caesar, in which A is encrypted as D, B as E, and so forth. As many a schoolboy has discovered to his embarrassment, cyclical-shift substitution ciphers are not secure, nor is any other monoalphabetic substitution cipher in which a given plaintext symbol is always encrypted into the same ciphertext symbol. Because of the redundancy of the English language, only about 25 symbols of ciphertext are required to permit the cryptanalysis of monoalphabetic substitution ciphers, which makes them a popular source for recreational cryptograms. The explanation for this weakness is that the frequency distributions of symbols in the plaintext and in the ciphertext are identical, only the symbols having been relabeled. In fact, any structure or pattern in the plaintext is preserved intact in the ciphertext, so that the cryptanalyst's task is an easy one.

There are two main approaches that have been employed with substitution ciphers to lessen the extent to which structure in the plaintext—primarily single-letter frequencies—survives in the ciphertext. One approach is to encrypt elements of plaintext consisting of two or more symbols; e.g., digraphs and trigraphs. The other is to use several cipher alphabets. When this approach of polyalphabetic substitution is carried to its limit, it results in onetime keys, or pads.

CODE:

N/A

OUTPUT:

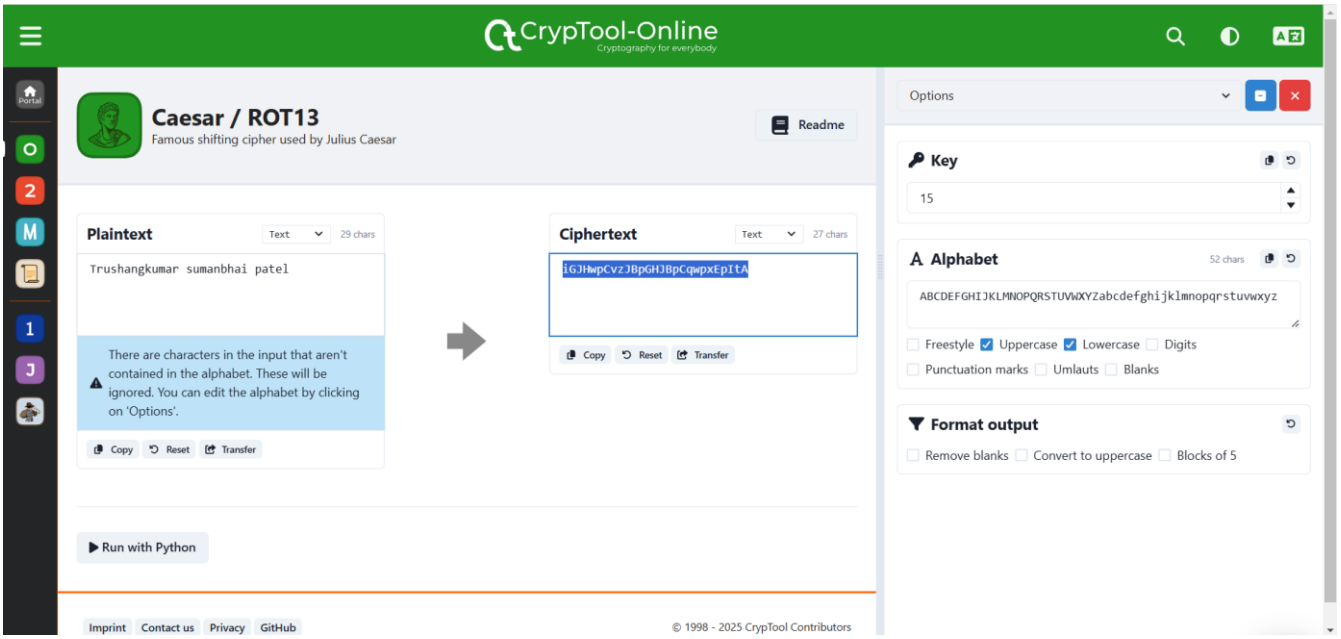


Figure 111: Encrypt the plain text using caser/Additive/shift cipher where k=15 without considering space

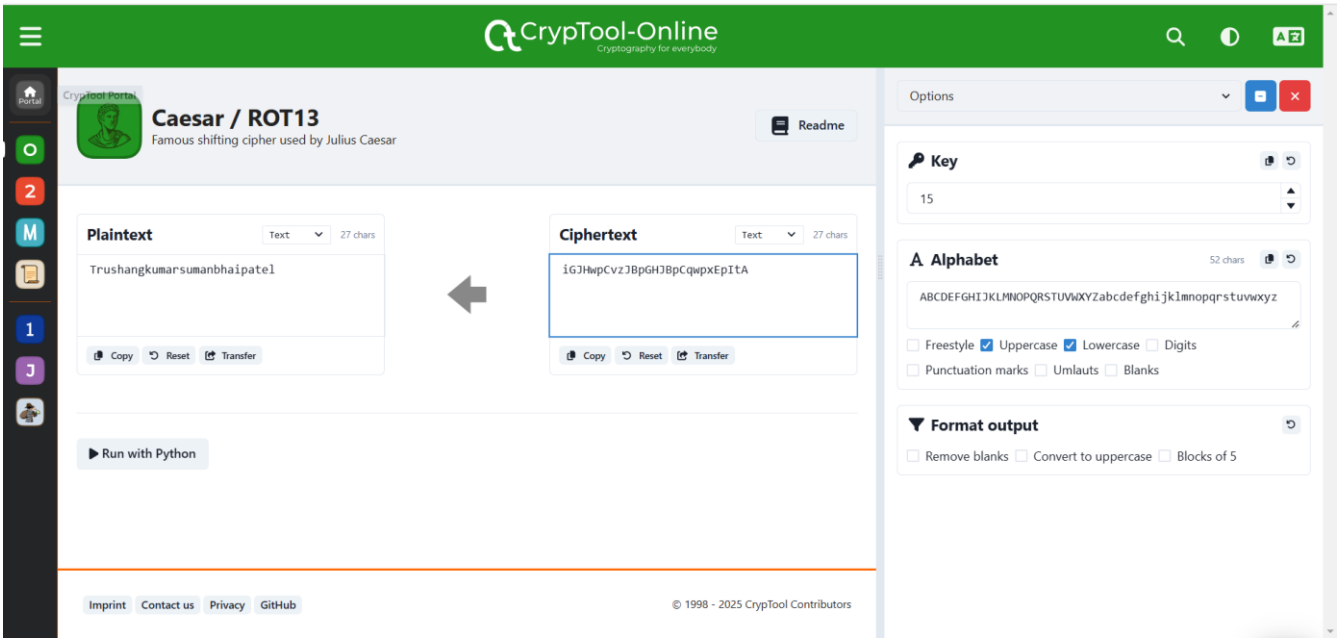


Figure 112:Decrypt the cipher text using caser/Additive/shift cipher where k=15 without considering space

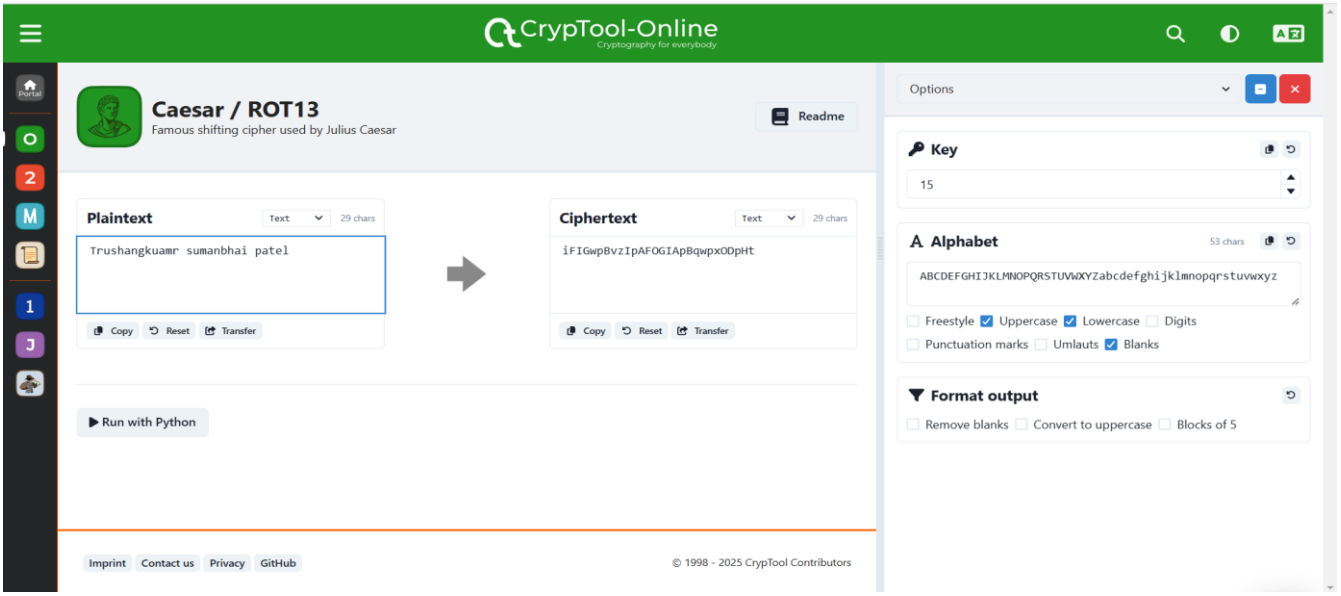


Figure 113:Encrypt the plain text using caser/Additive/shift cipher where k=15 with considering space

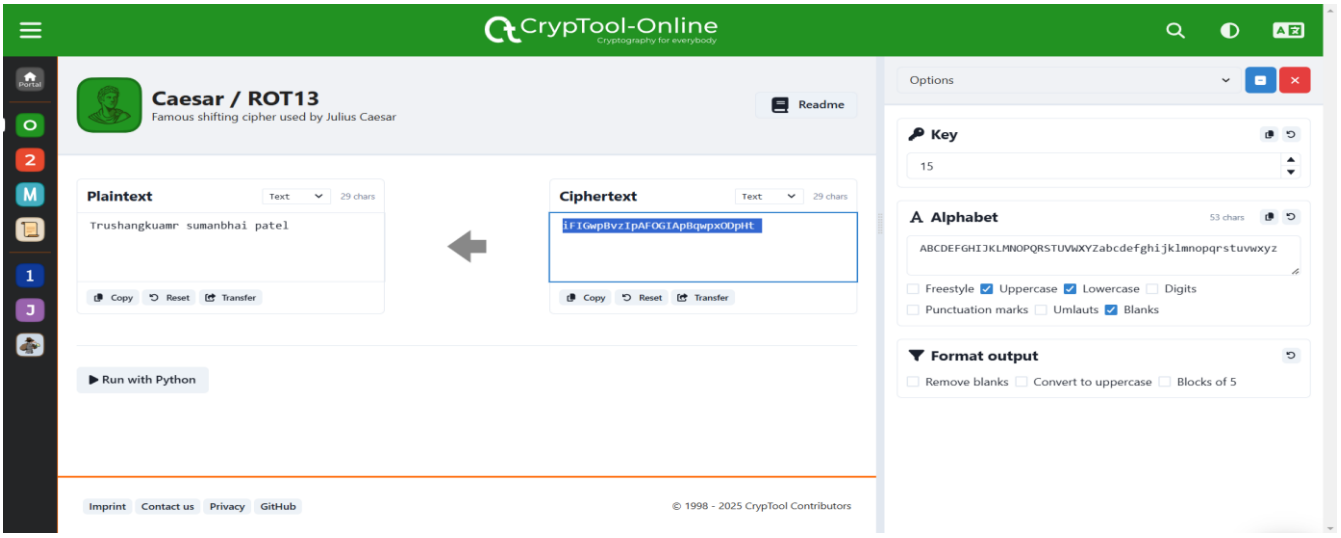


Figure 114:Decrypt the cipher text using caser/Additive/shift cipher where k=15 with considering space

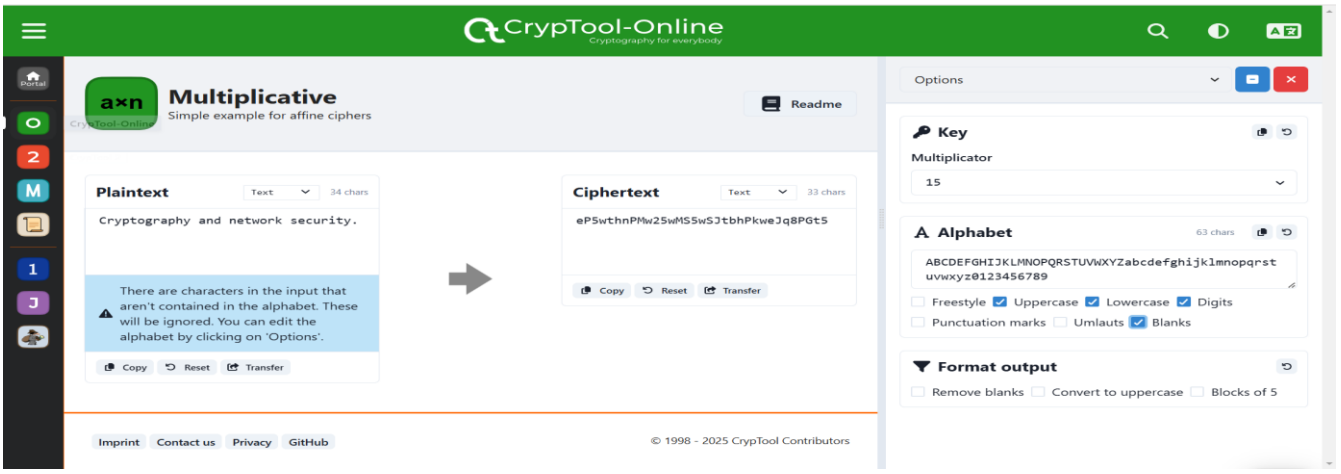


Figure 115:Encrypt message using Multiplicative cipher considering Uppercase and lowercase and key=15

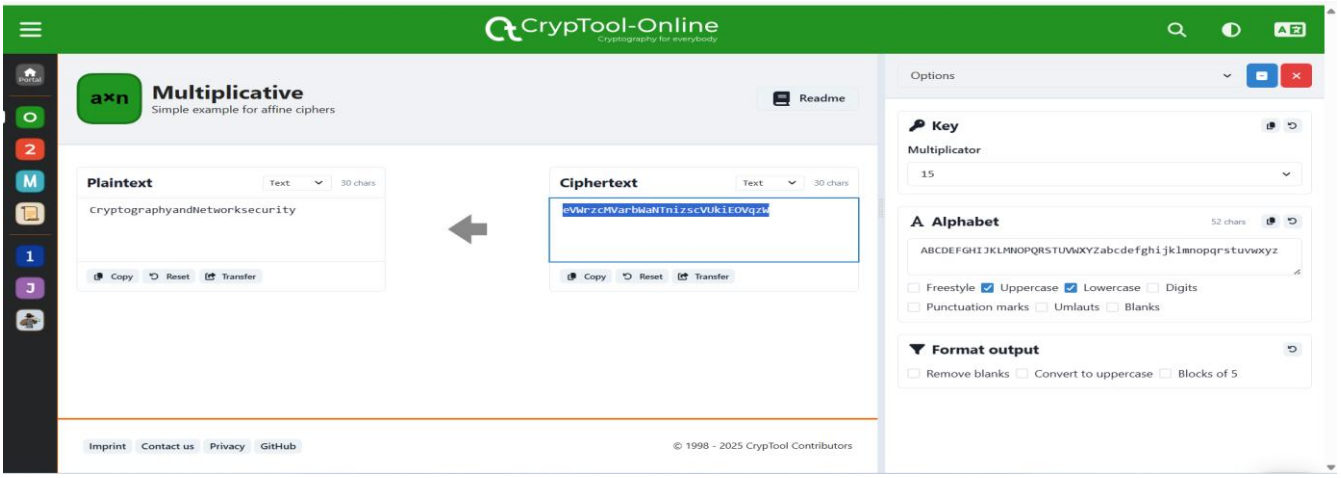


Figure 116:Decrypt message using Multiplicative cipher considering Uppercase and lowercase and key=15

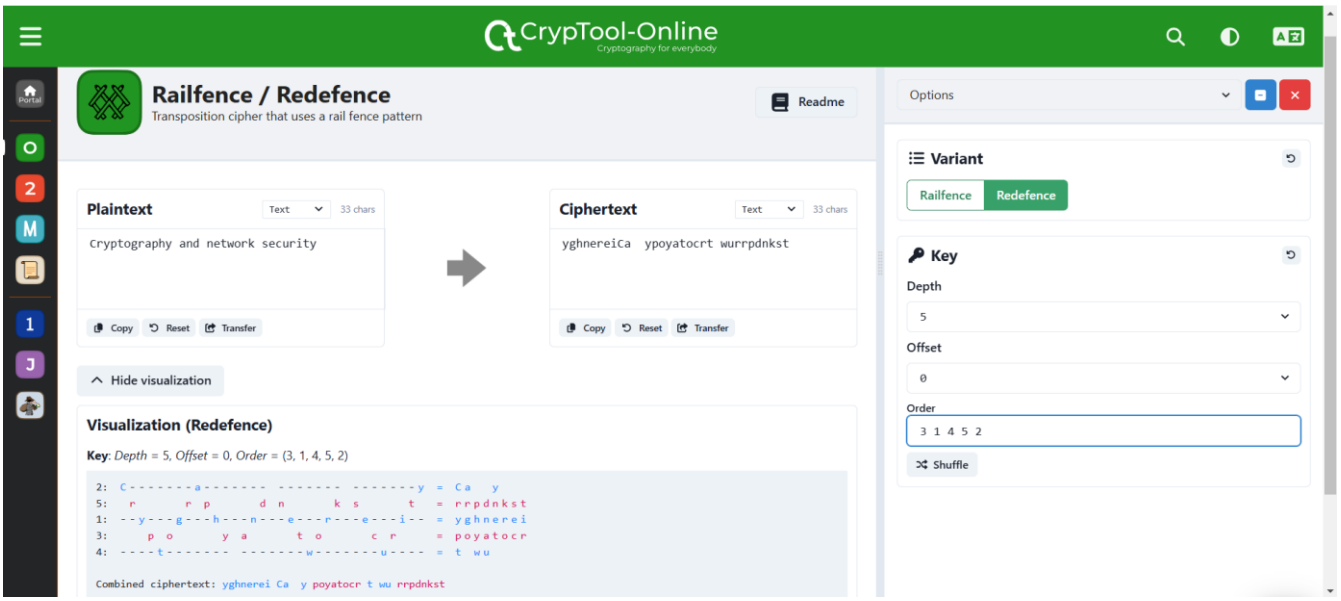


Figure 117:Encrypt the plaintext using key:34152 in transposition cipher

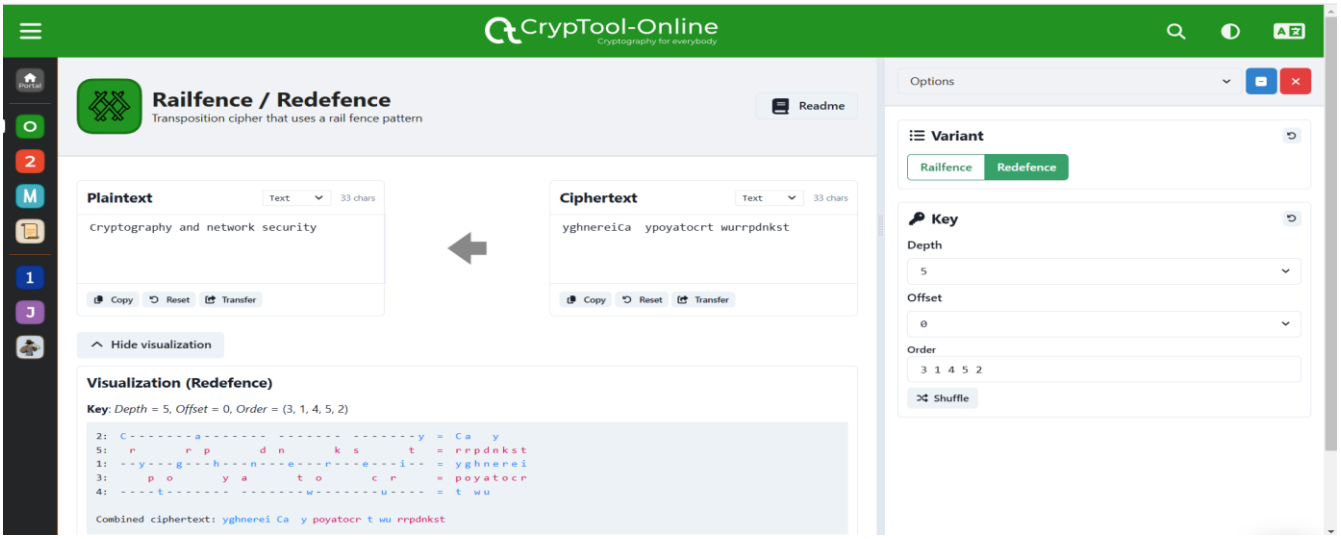


Figure 118:Decrtpt the Ciphertext using key:25134 in transposition cipher

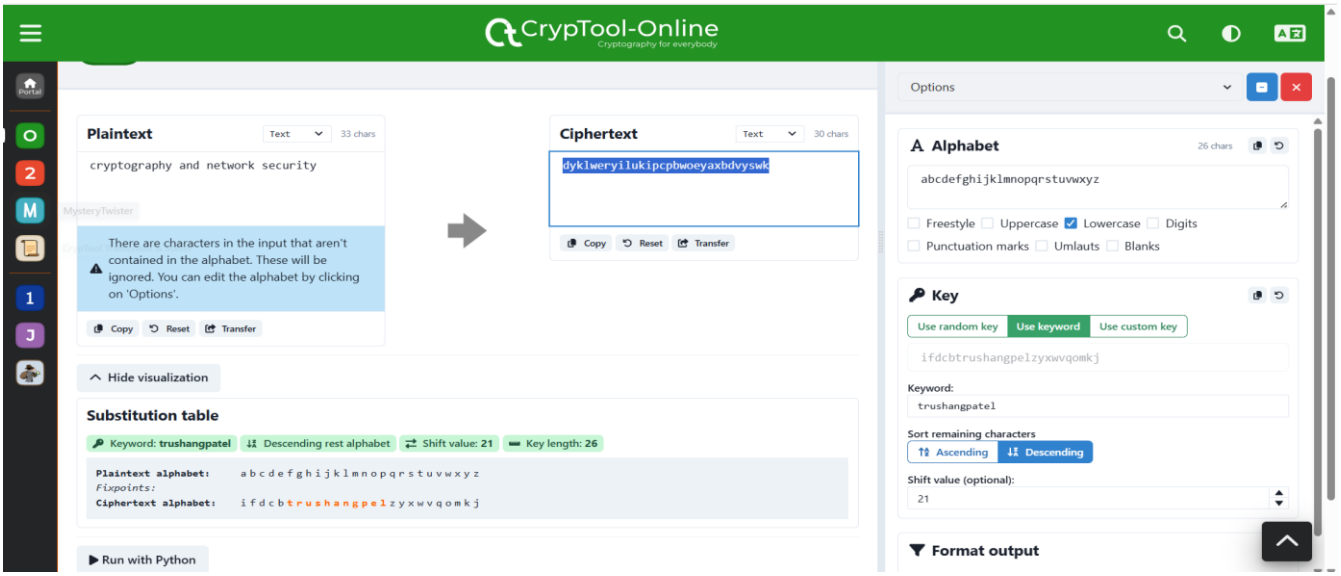


Figure 119:Encrypt the message using Monoalphabetic and key= trushangpatel

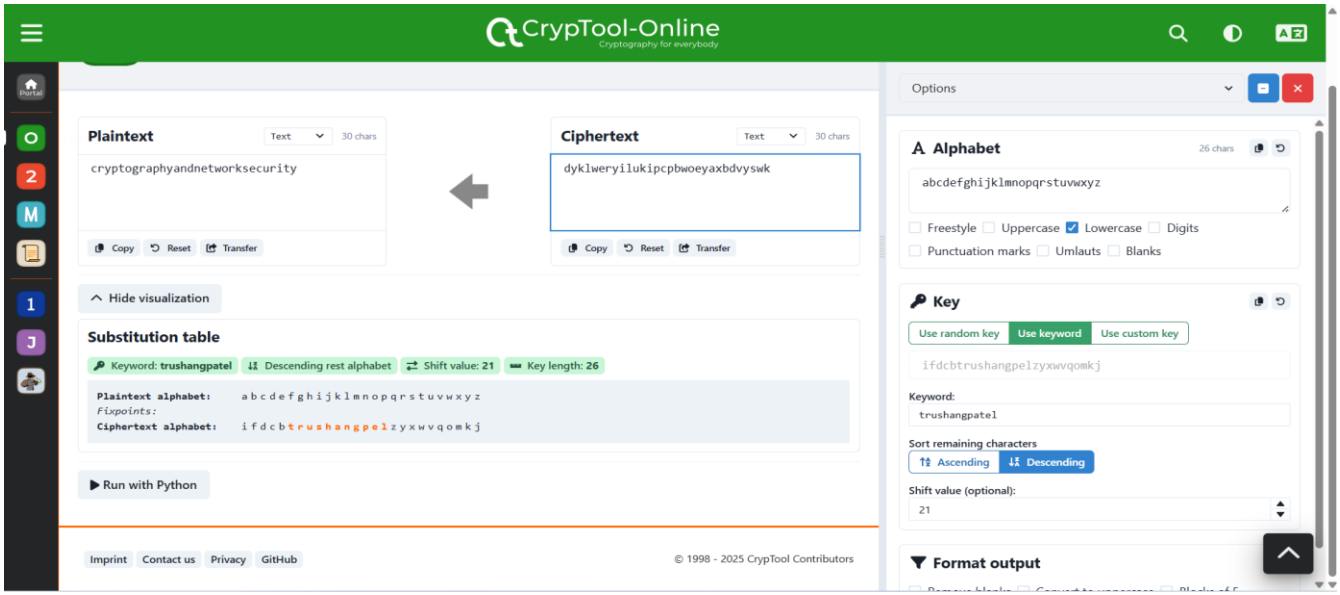


Figure 120:Decrypt the message using Monoalphabetic and key= trushangpatel

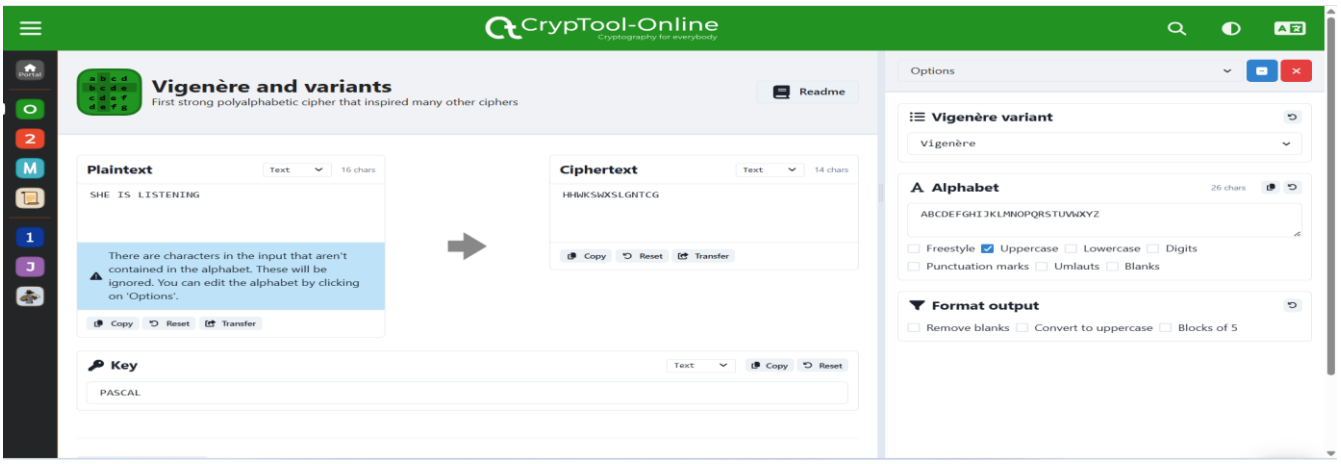


Figure 121:Encryption the message using Vigenère with key=PASCAL

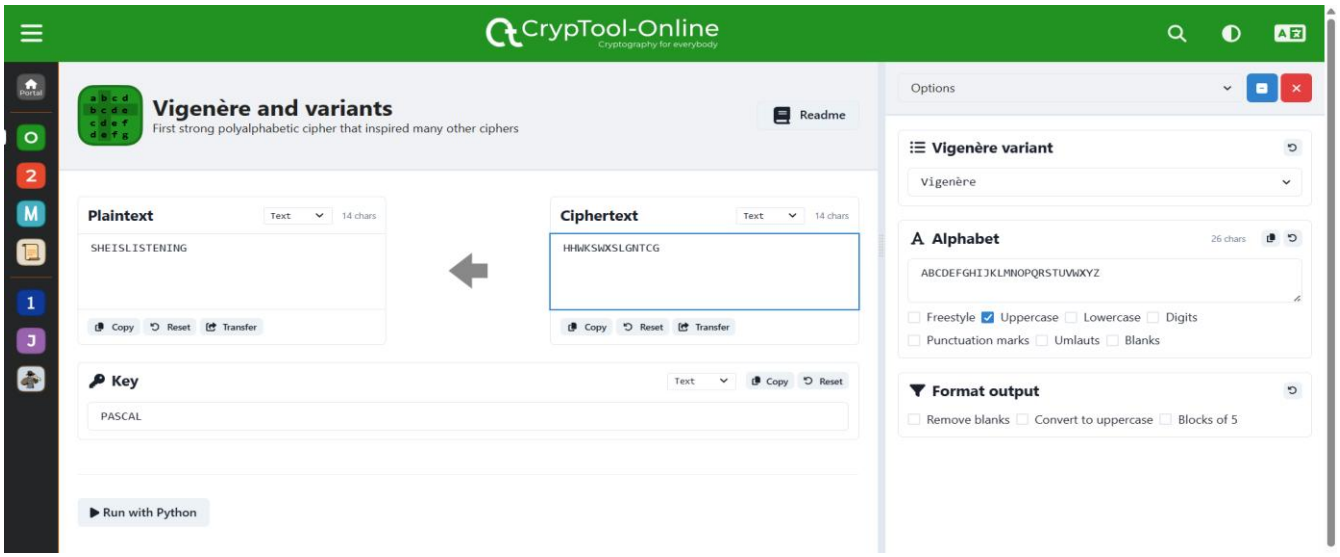


Figure 122: Decryption the message using Vigenère with key=PASCAL

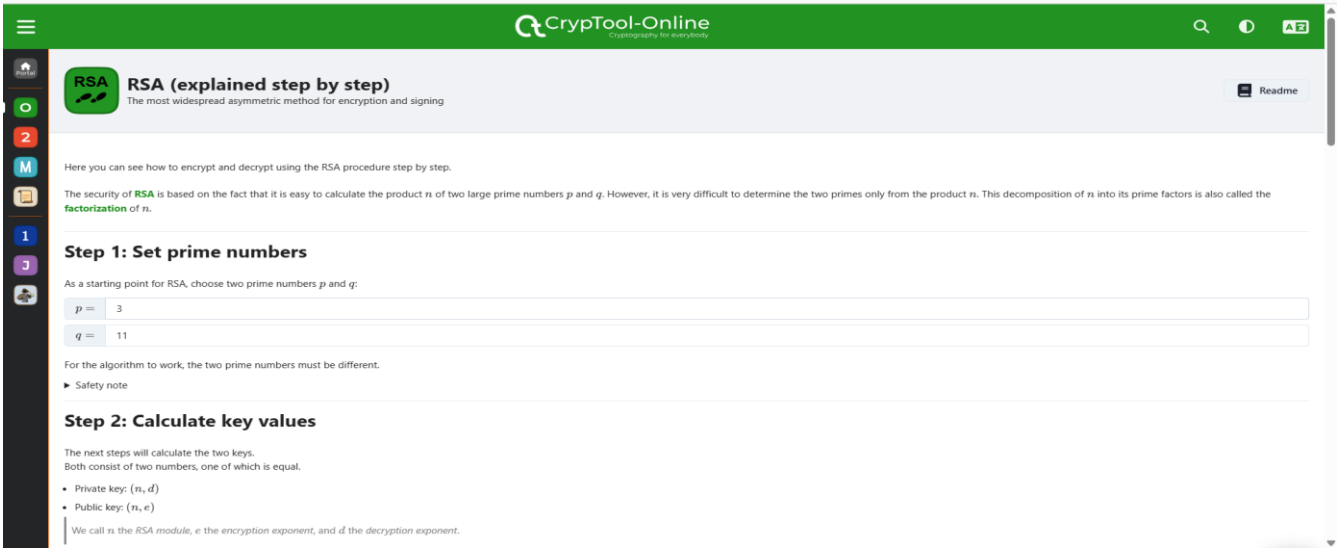


Figure 123:Pick two prime number

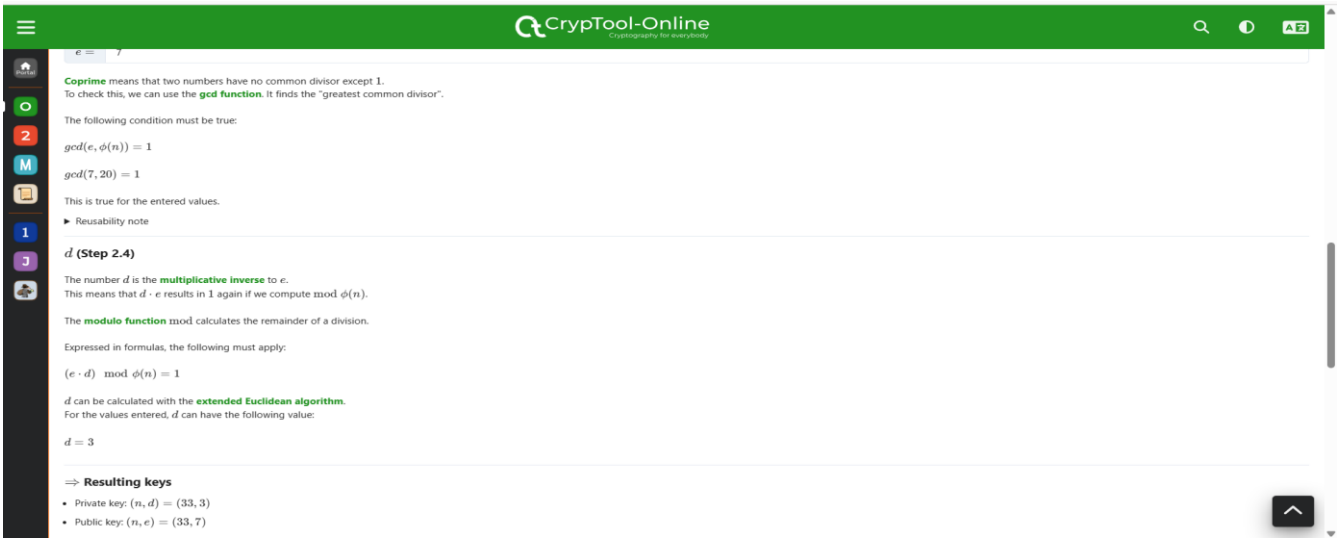


Figure 124:Generate the keys

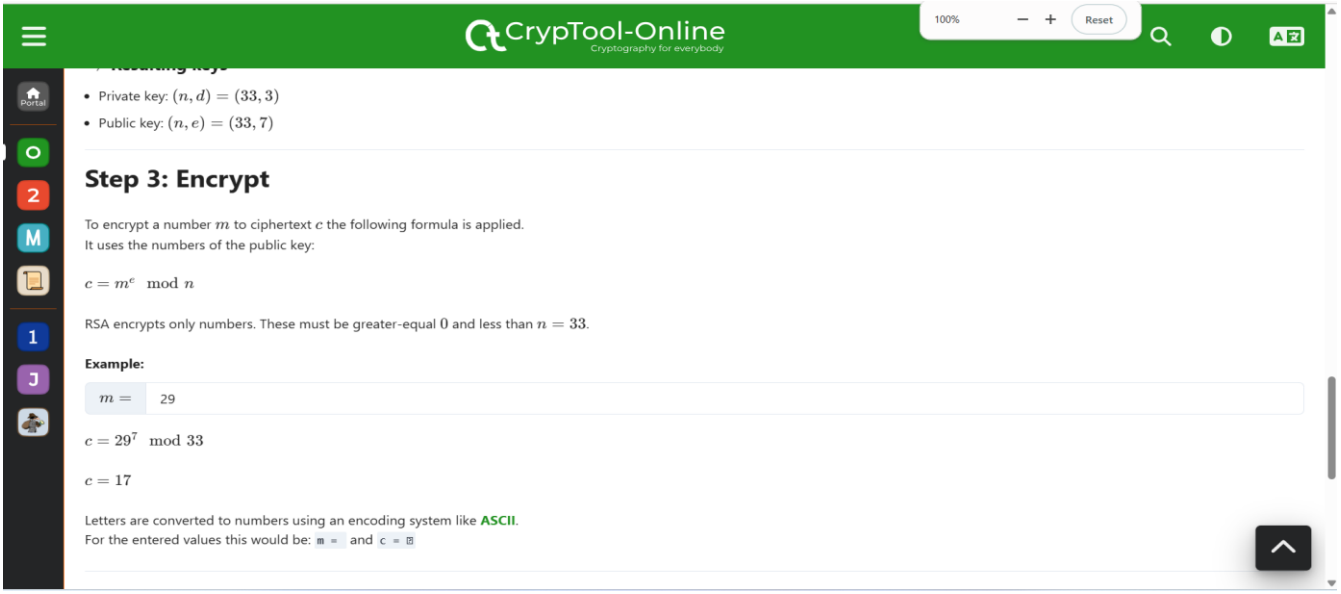


Figure 125:Encrypt the message

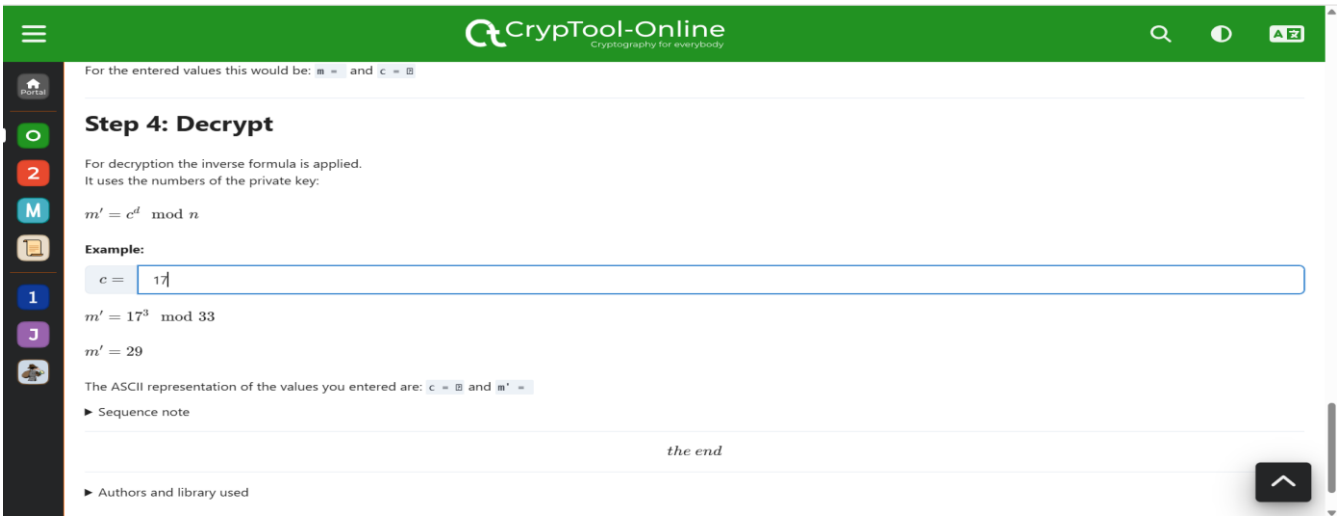


Figure 126:Decrypt the message

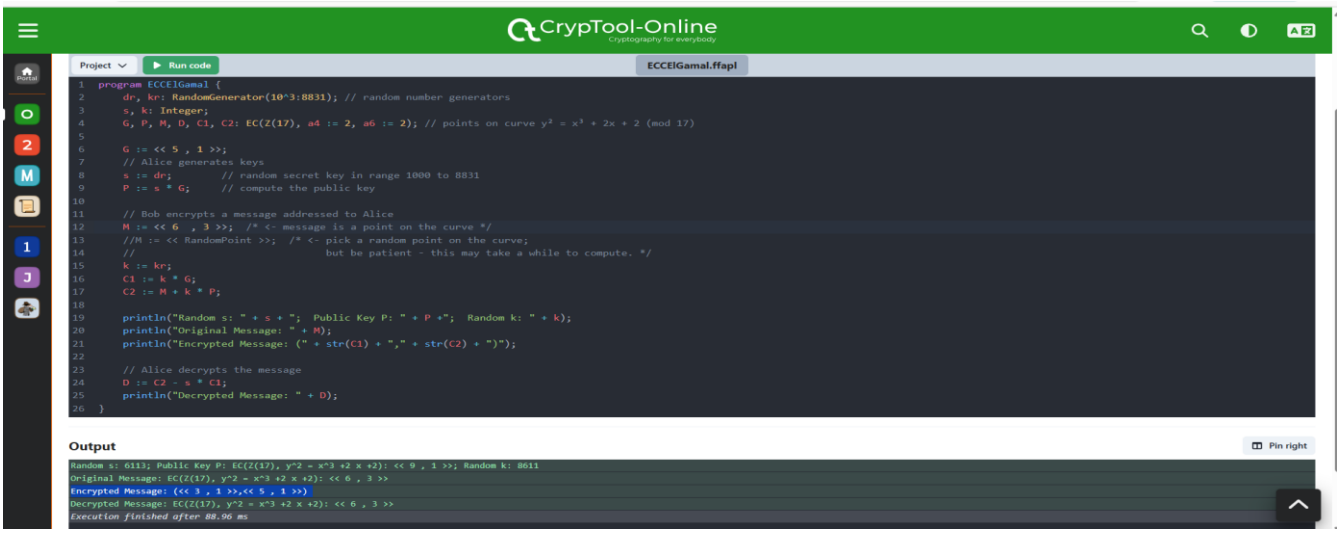


Figure 127:Use of Elgamal Cryptosystem

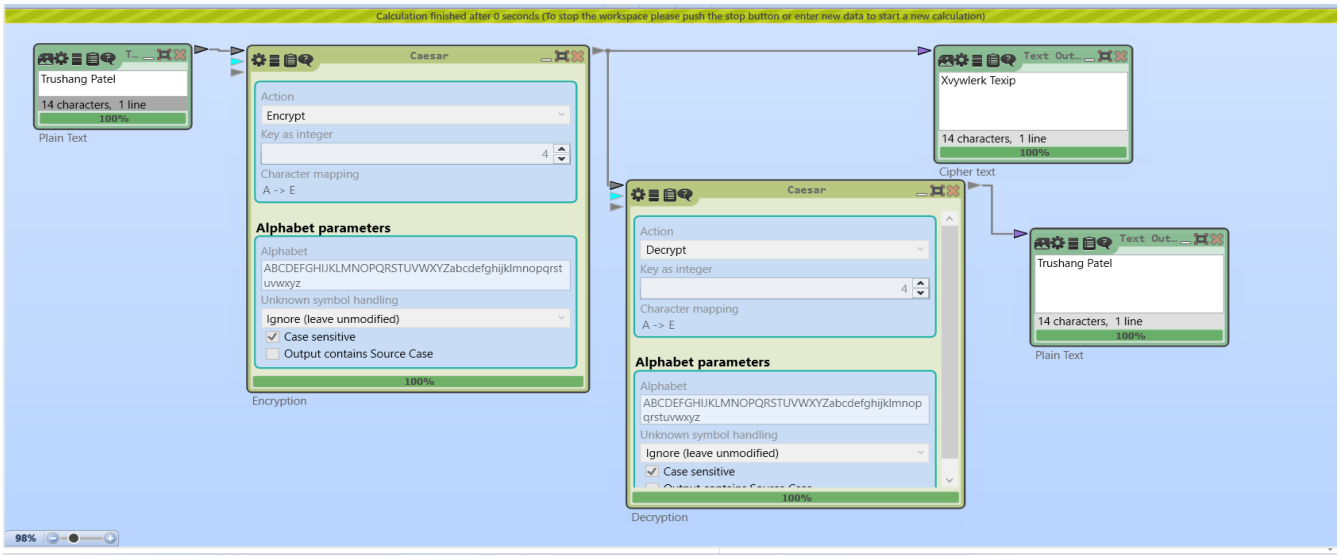


Figure 128:Encrypt and Decrypt the message using Caesar

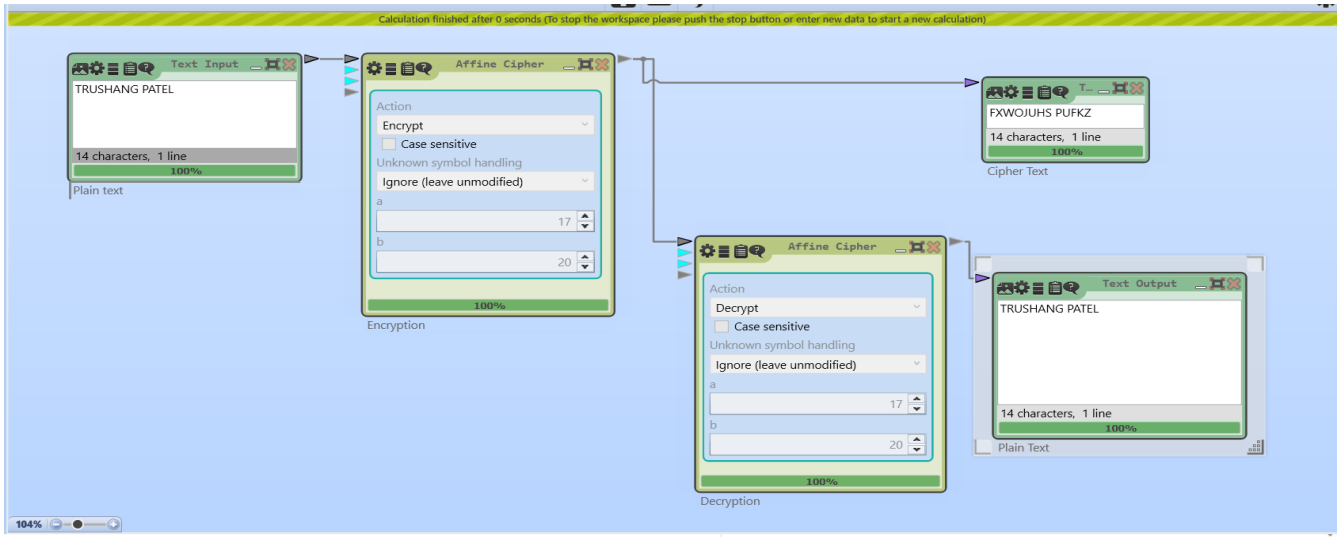


Figure 129:Encrypt and Decrypt the message using Affine Cipher

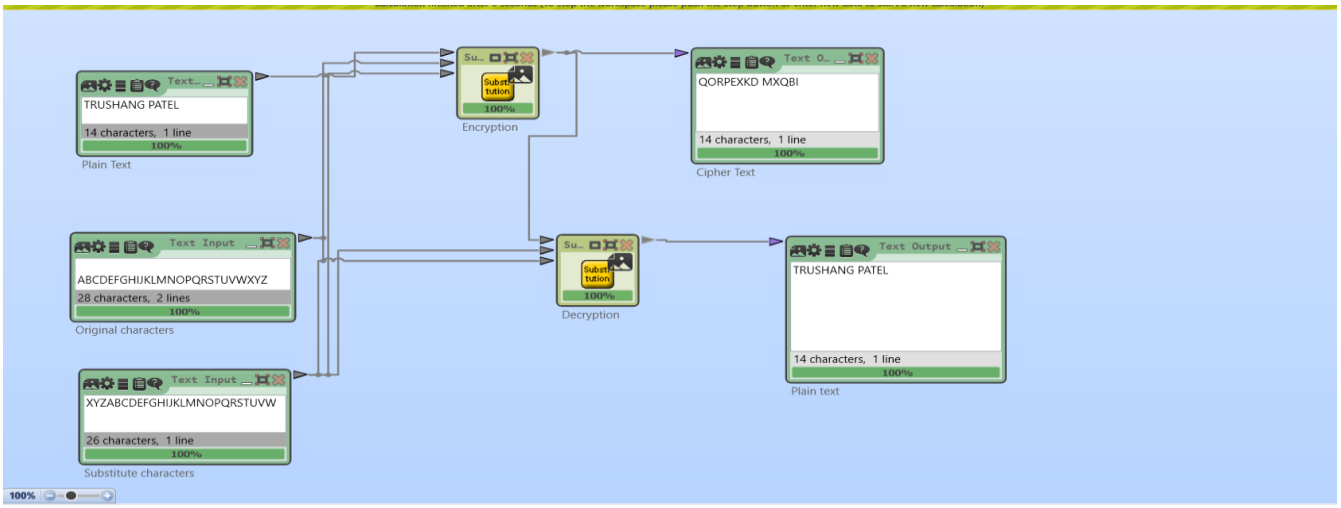


Figure 130:Encryption and decryption using substitution

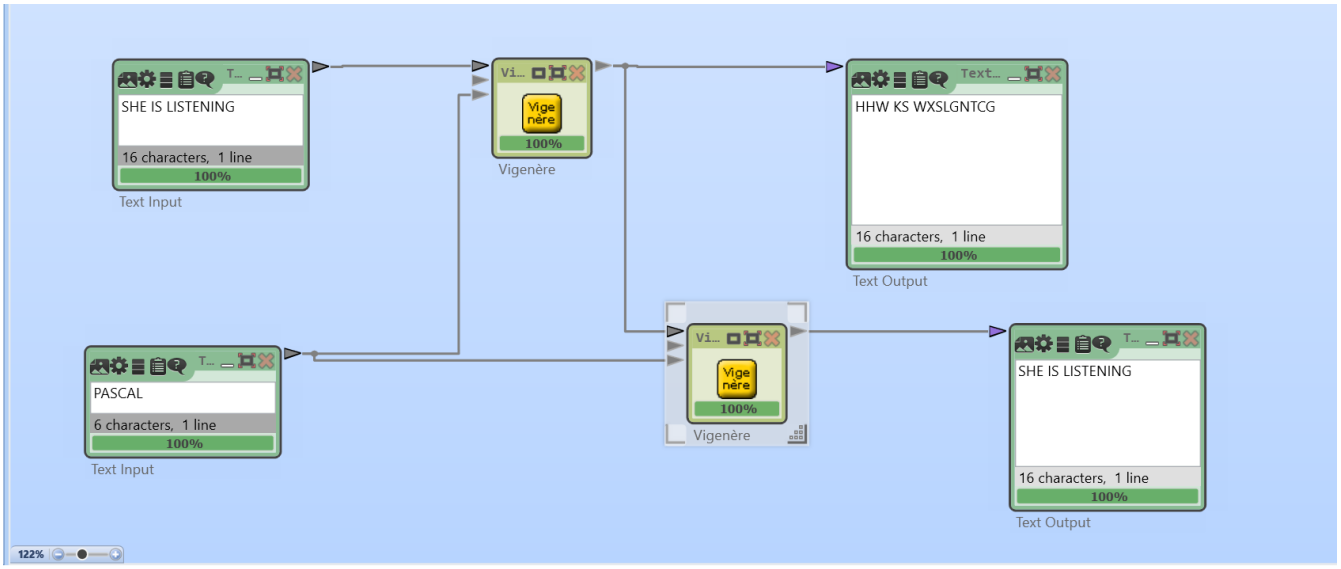


Figure 131:Encryption and Decryption using Vigenère cipher

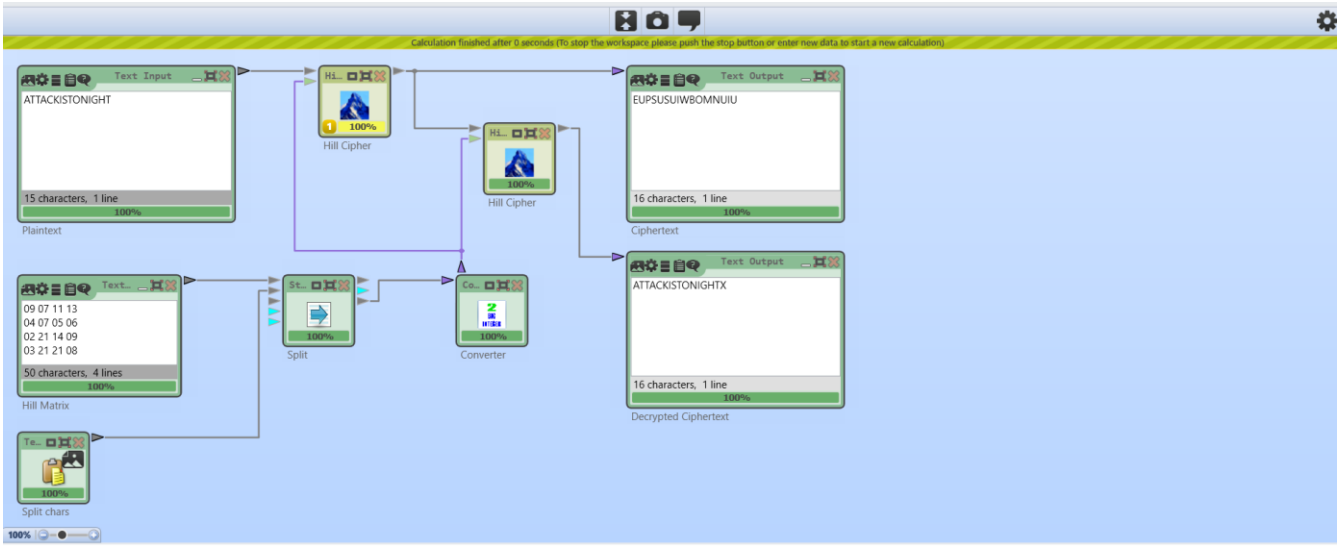


Figure 132:Encryption and Decryption using Hill cipher where=ATTACKISTONIGHT

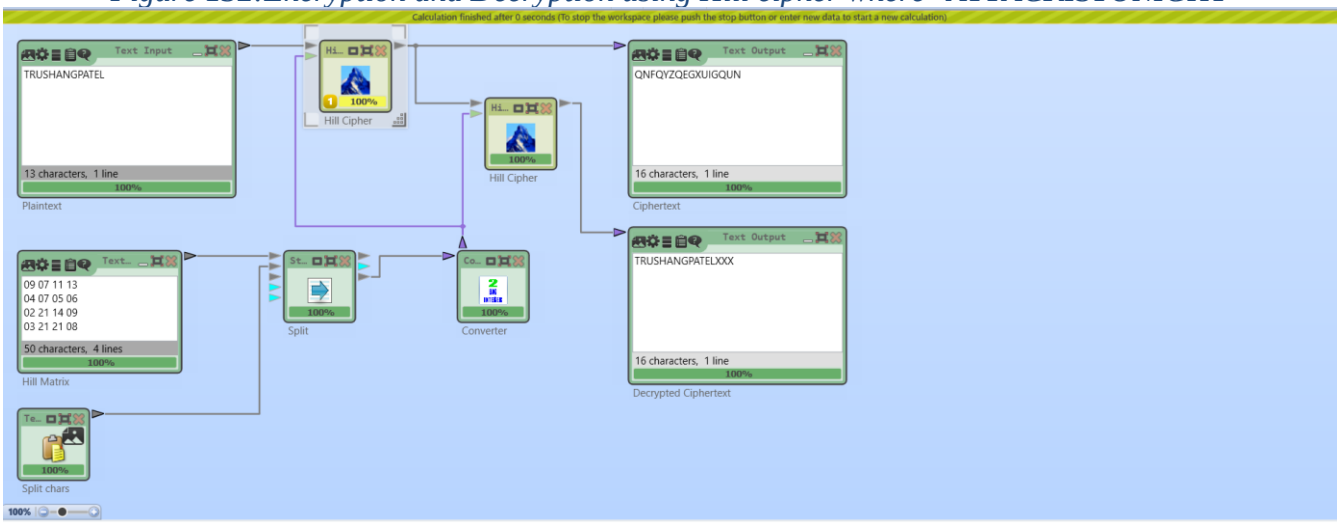


Figure 133:Encryption and Decryption using Hill cipher where plaintext = TRUSHANGPATEL

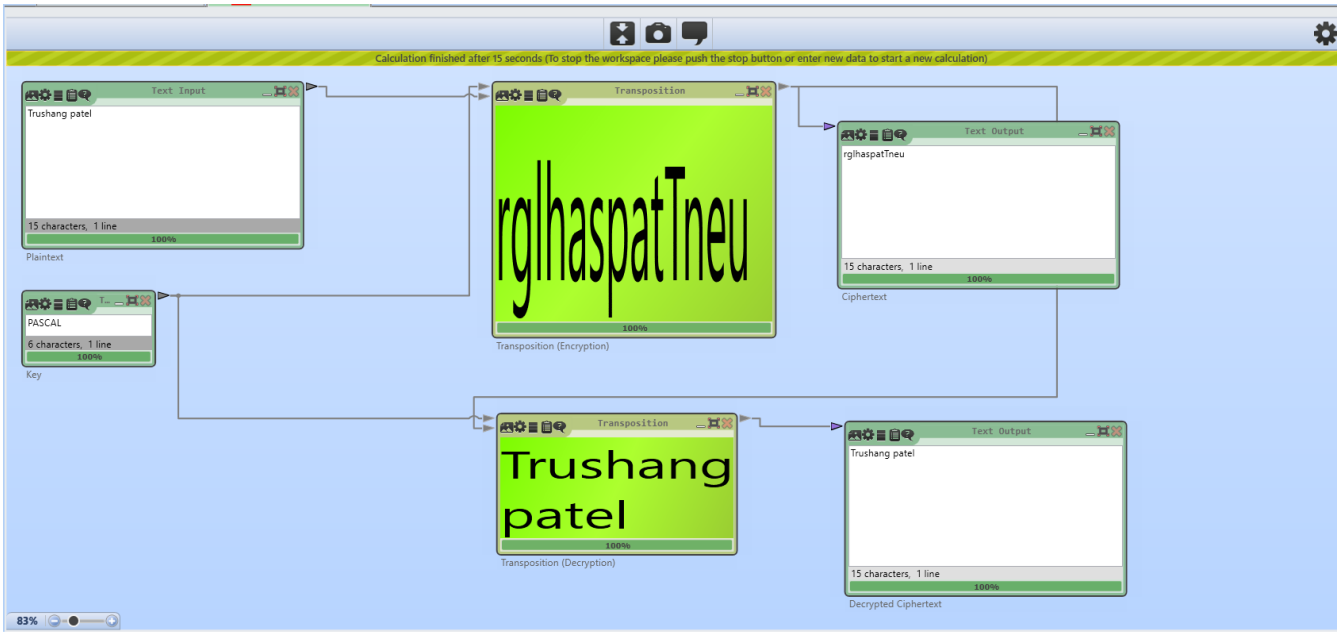


Figure 134:Encryption and Decryption using transposition cipher where plaintext = Trushang patel

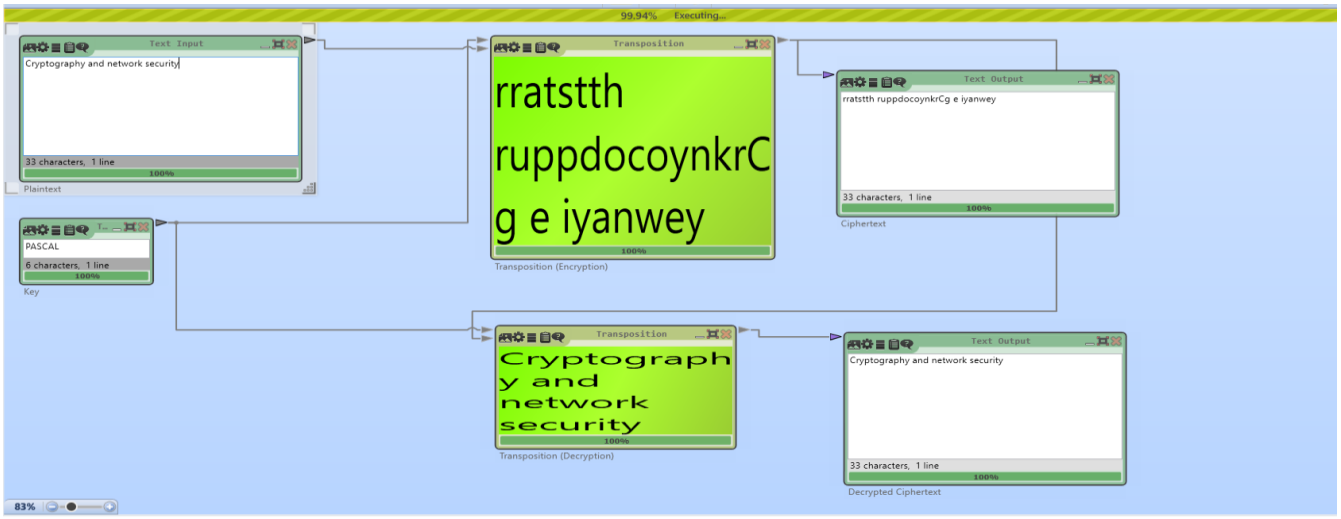


Figure 135Encryption and Decryption using transposition cipher where key = PASCAL

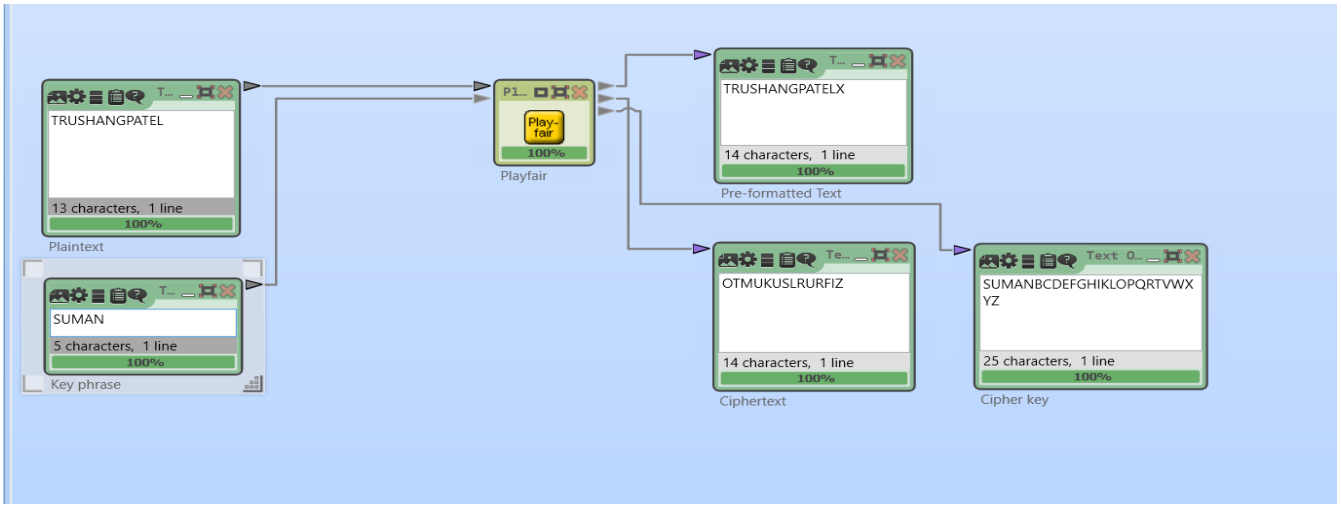


Figure 136:Example of Playfair where key = SUMAN

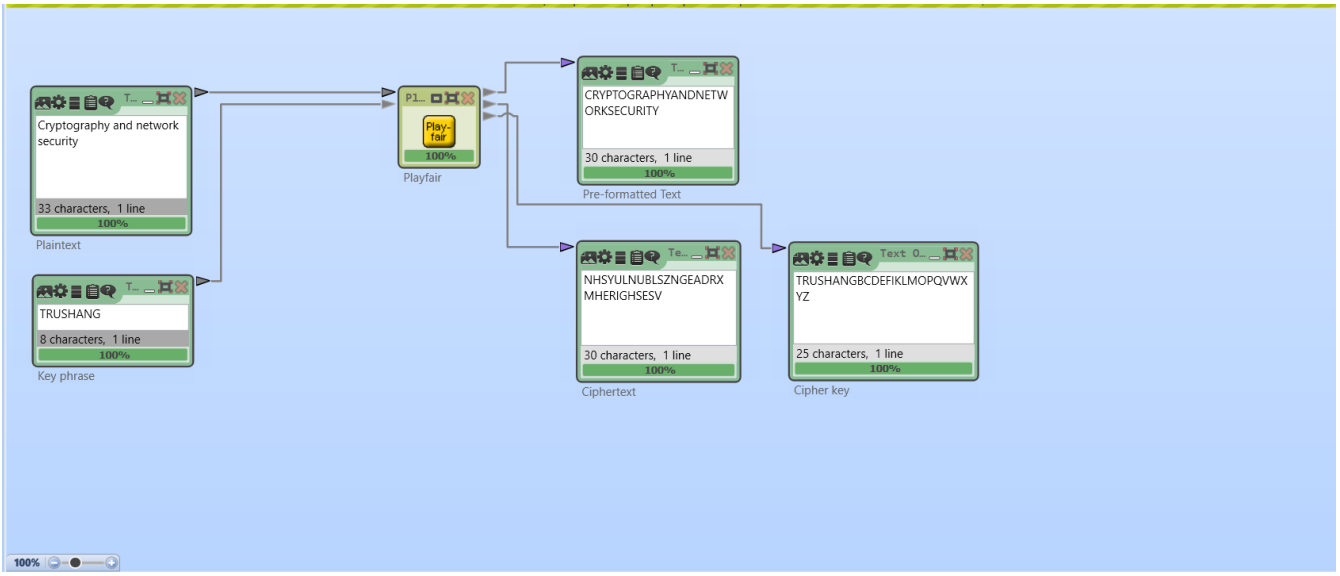


Figure 137:Exmaple of Playfair where key=TRUSHANG

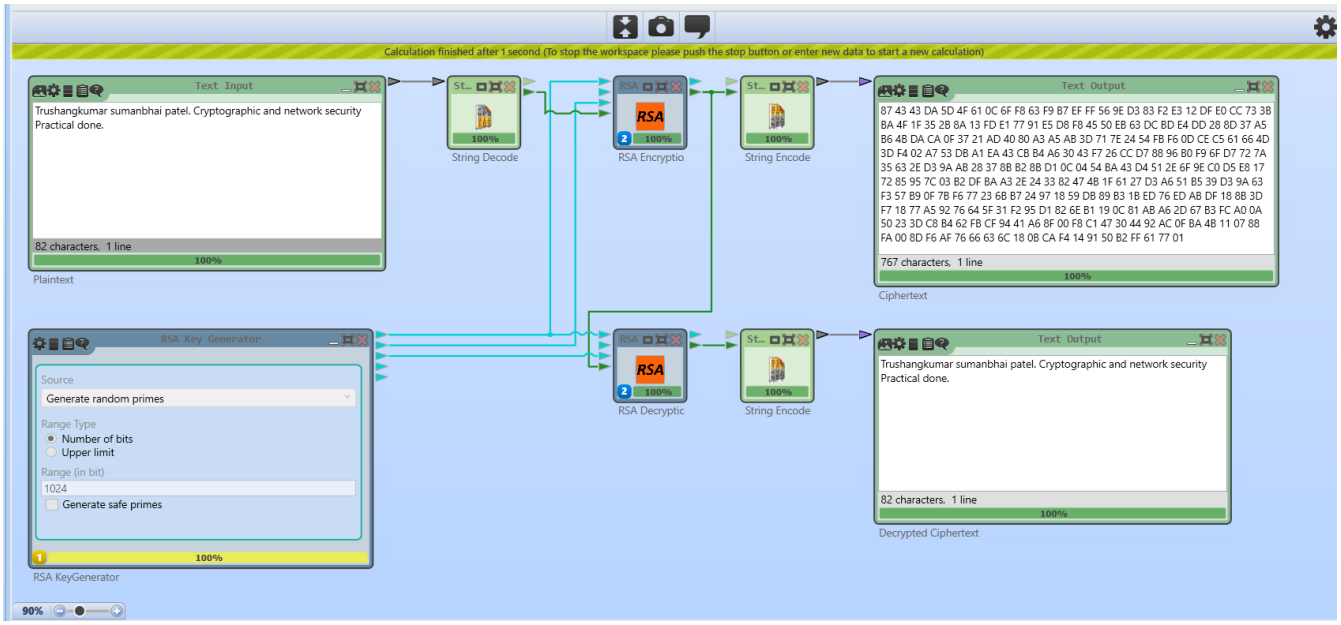


Figure 138:Encryption and Decryption using RSA algorithm

Figure 139:Plaintext of JavaCrypTool practical

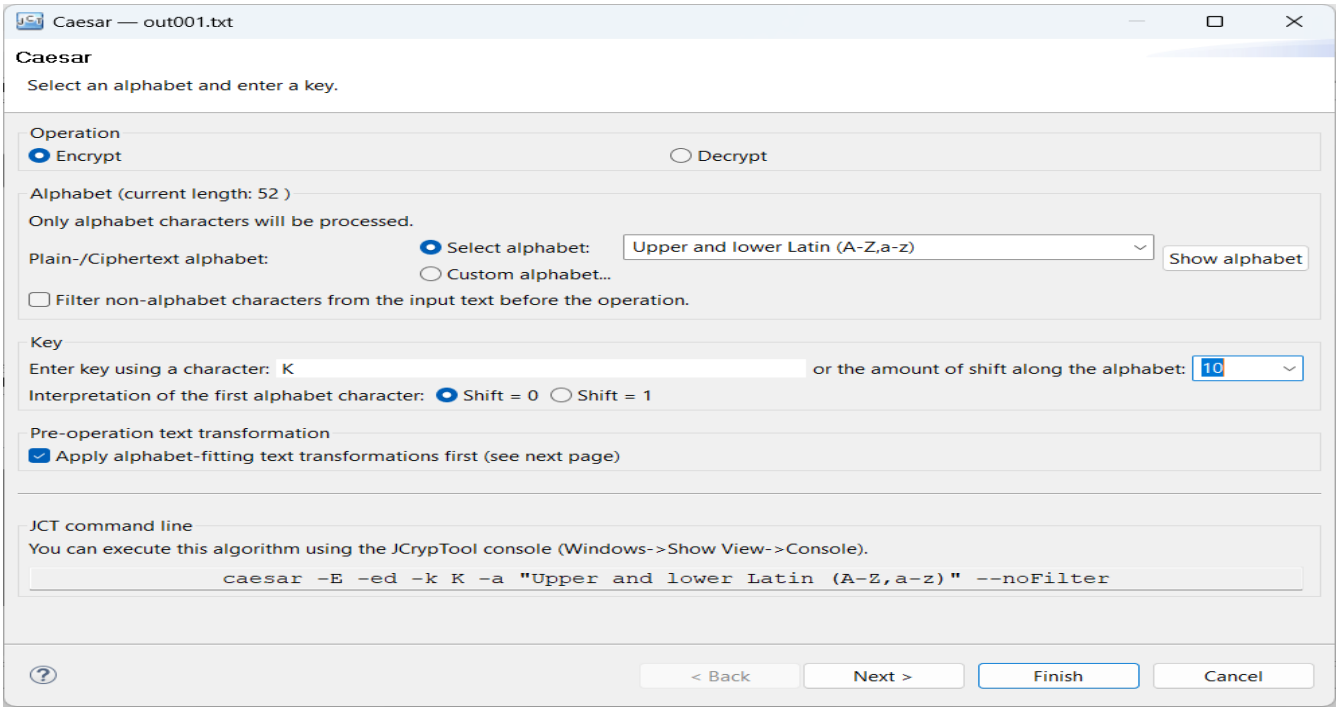


Figure 140:Encrypt Using Caser cipher

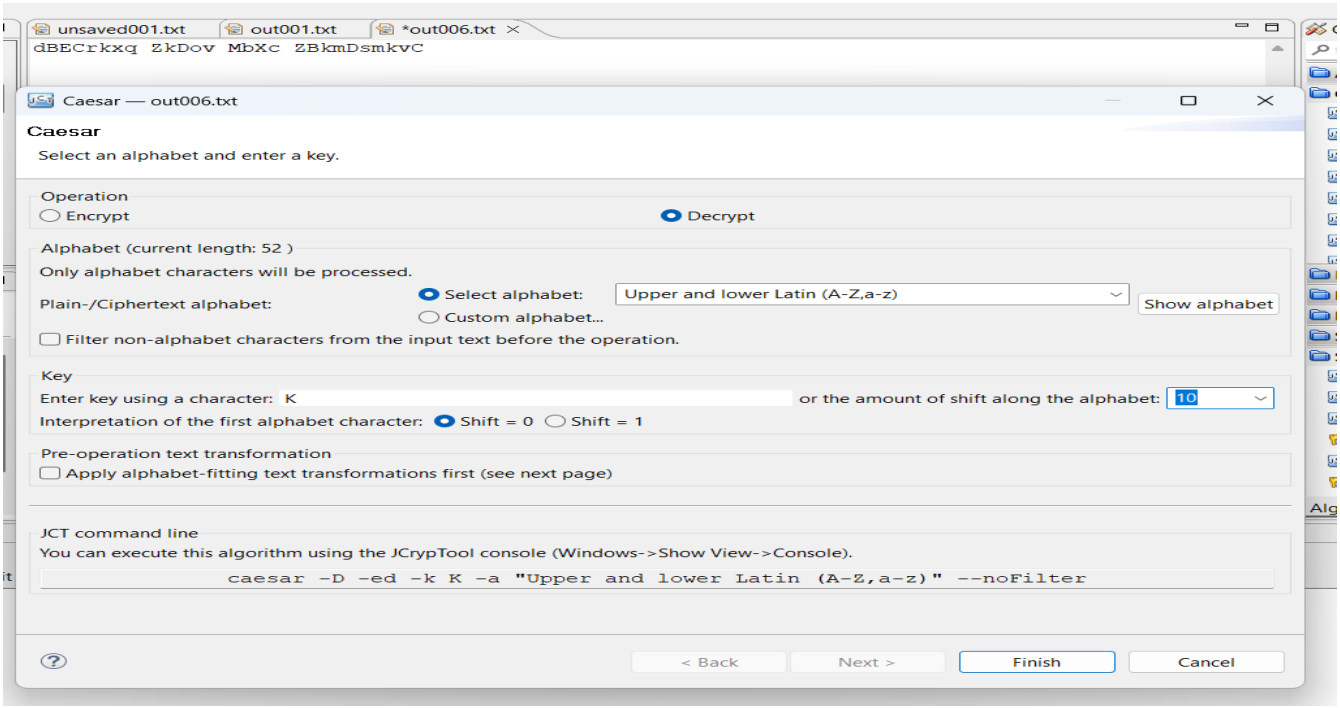


Figure 141:Decrypt using caser cipher

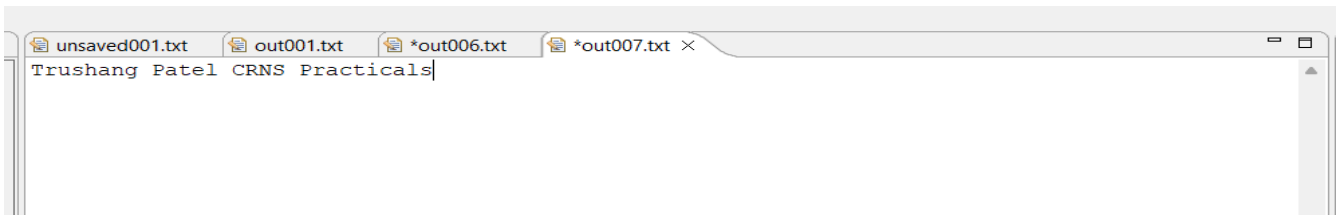


Figure 142:After Decrypting output

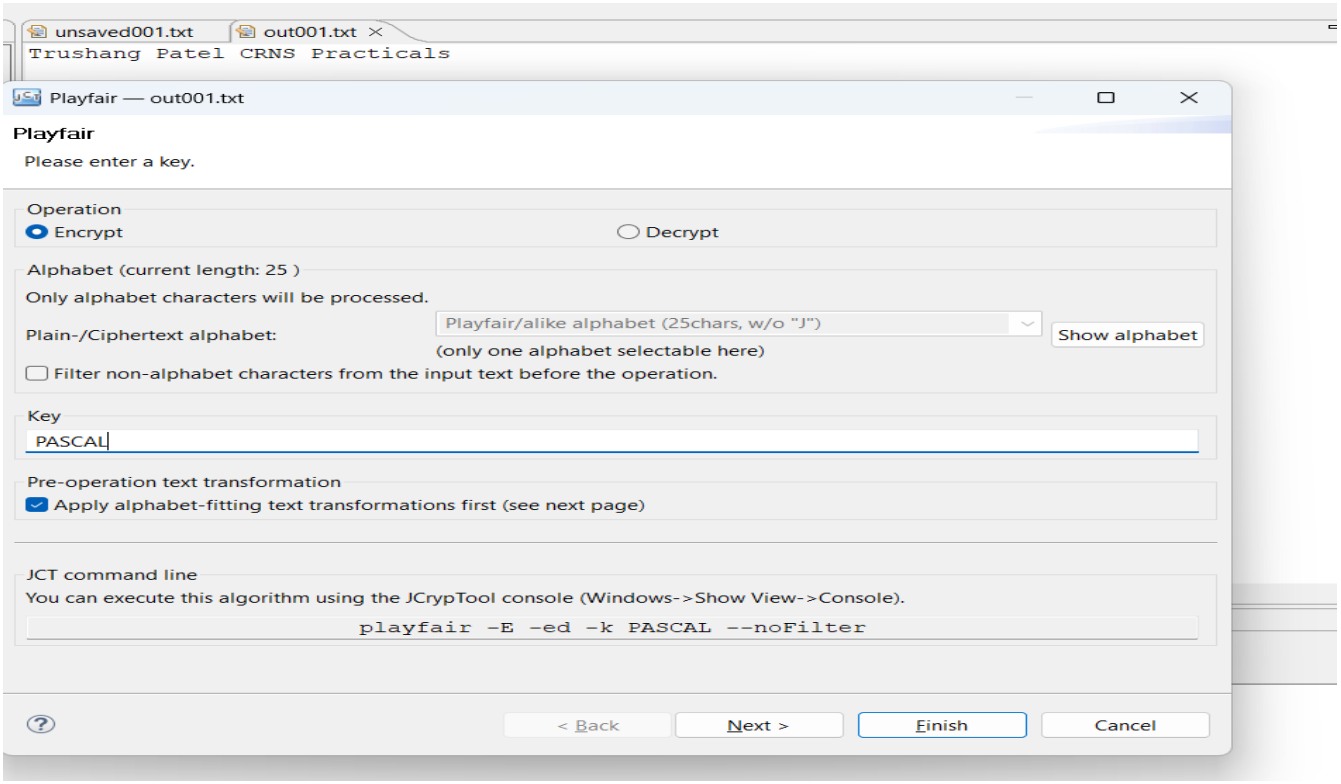


Figure 143:Encrypt using Playfair cipher

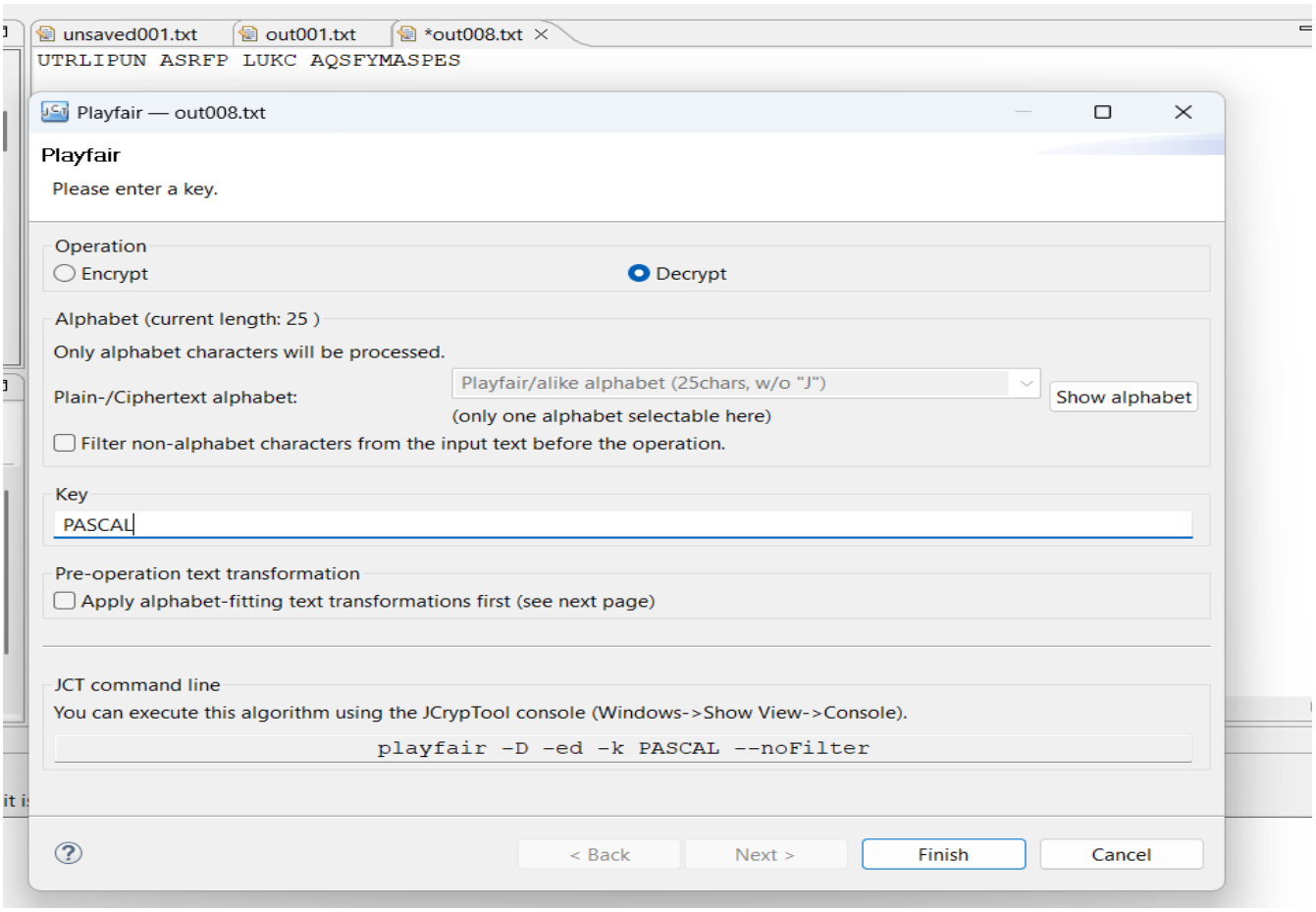


Figure 144:Decrypt using Playfair cipher

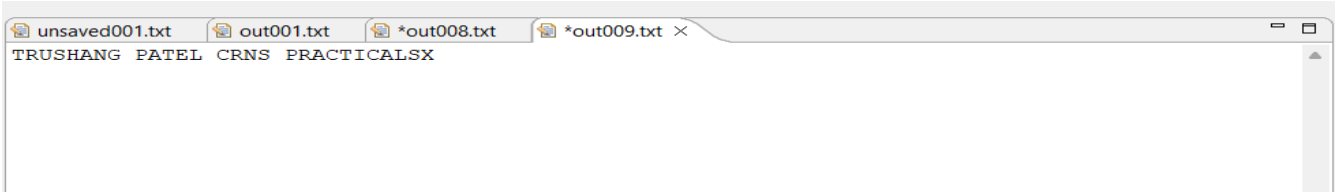


Figure 145:Output of Playfair cipher

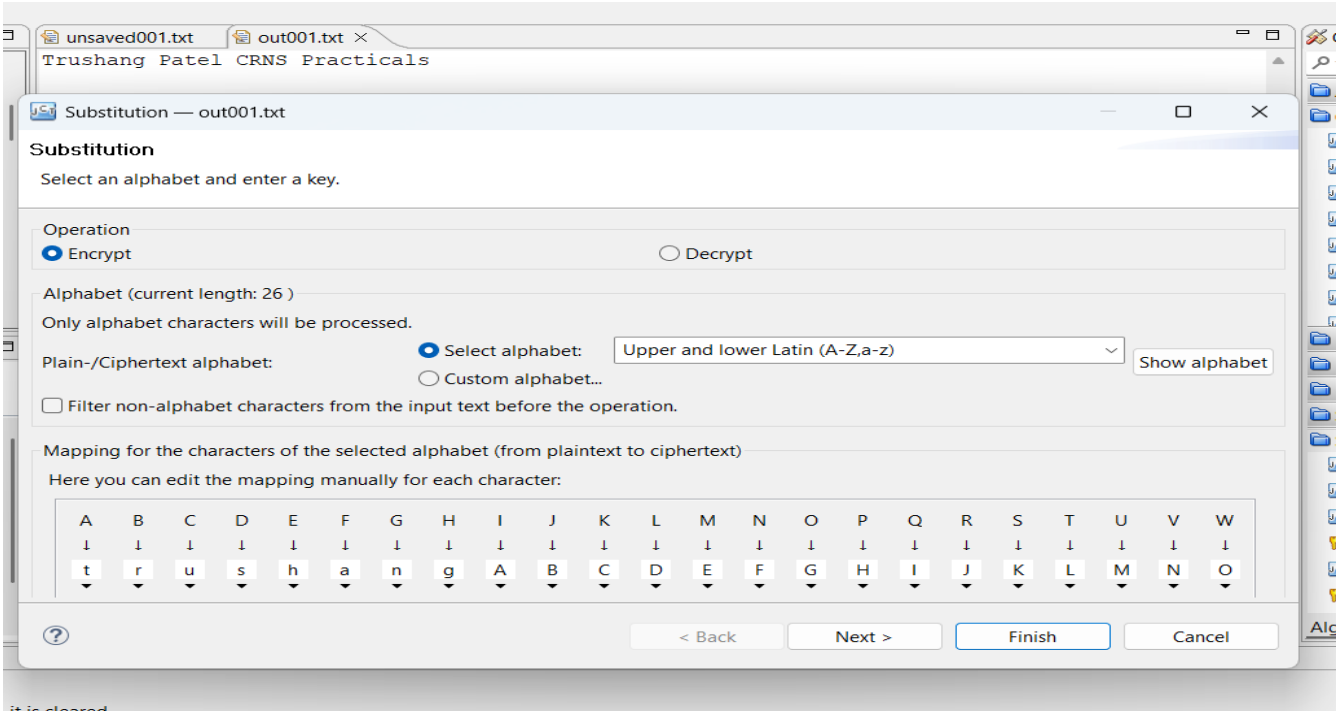


Figure 146:Encryption using substitution cipher

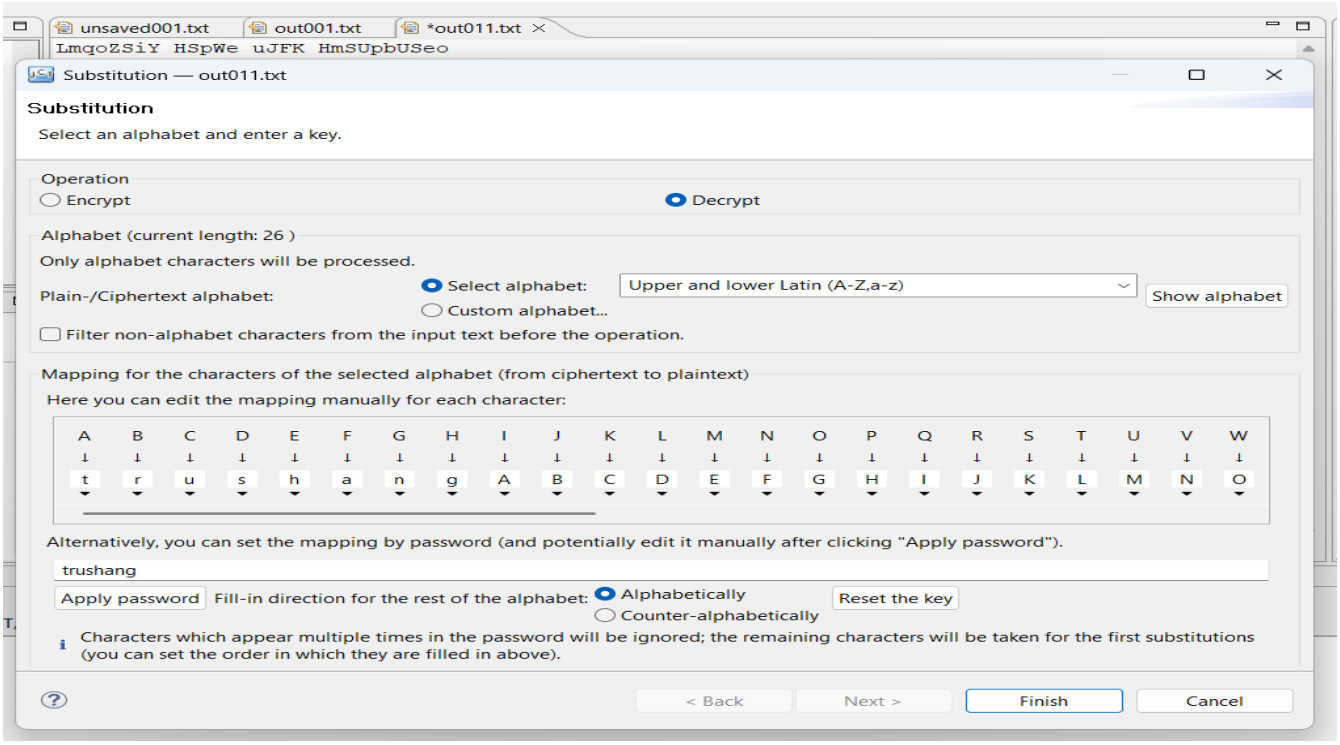


Figure 147:Decrypt using Substitution cipher

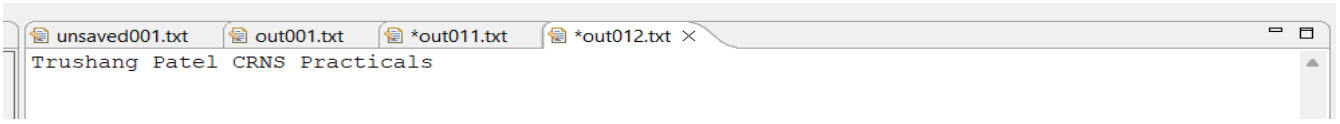


Figure 148: Output of the Substitution

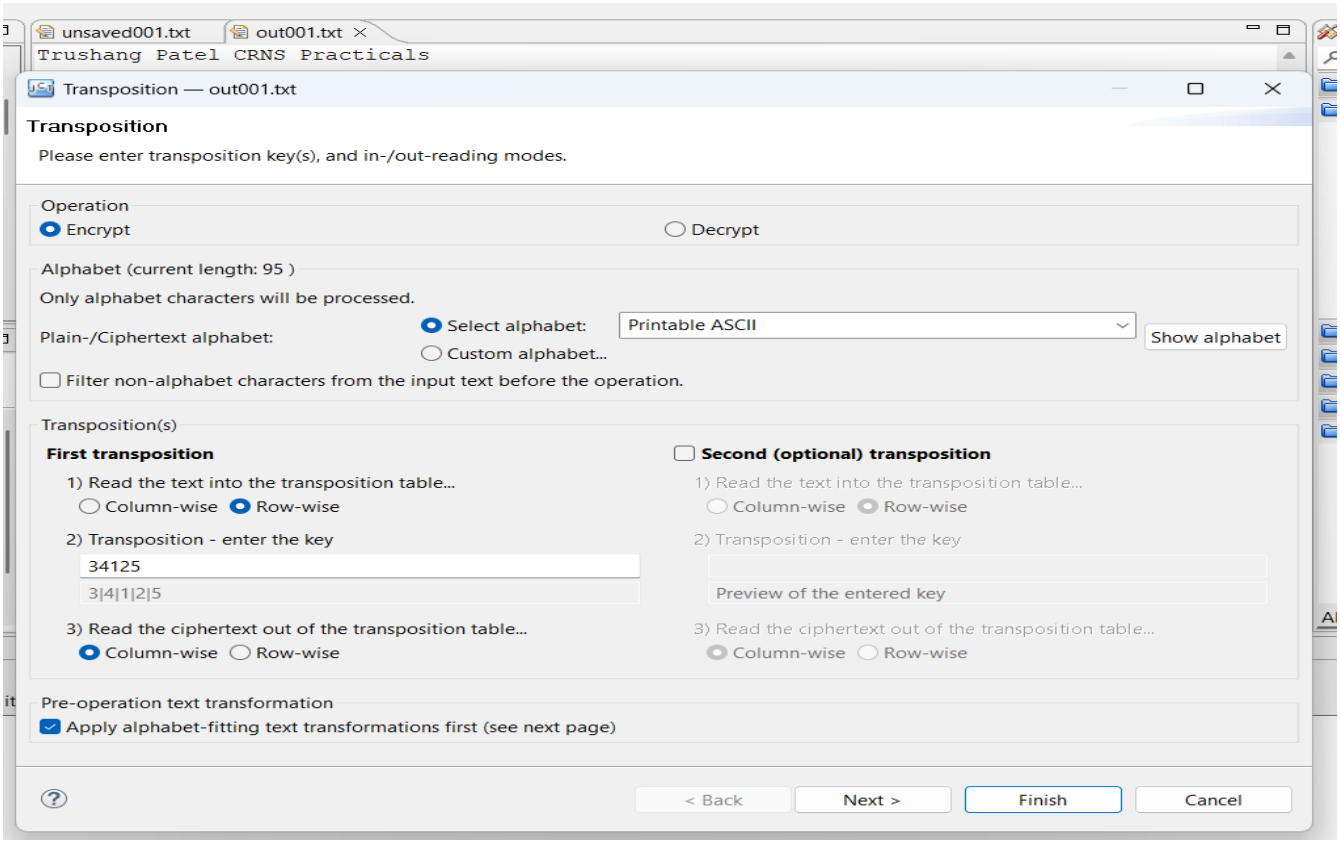


Figure 149: Encryption using Transposition cipher

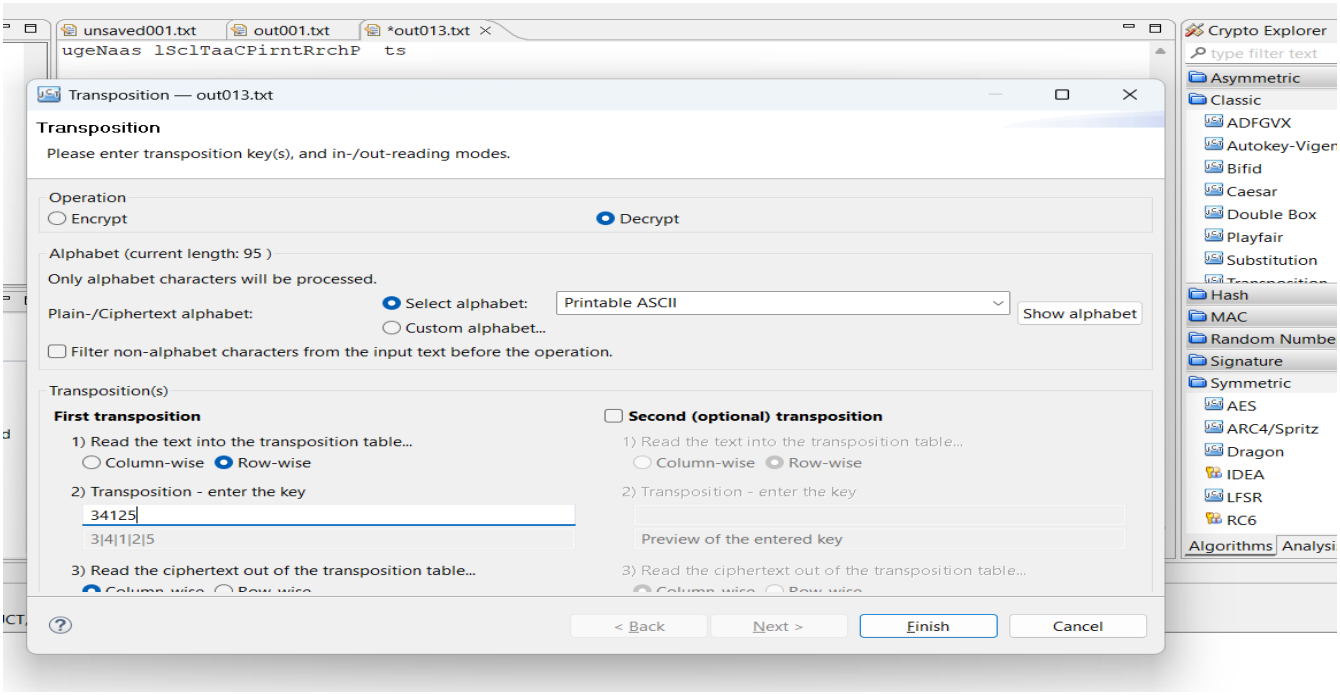


Figure 150:Decryption using Transposition cipher

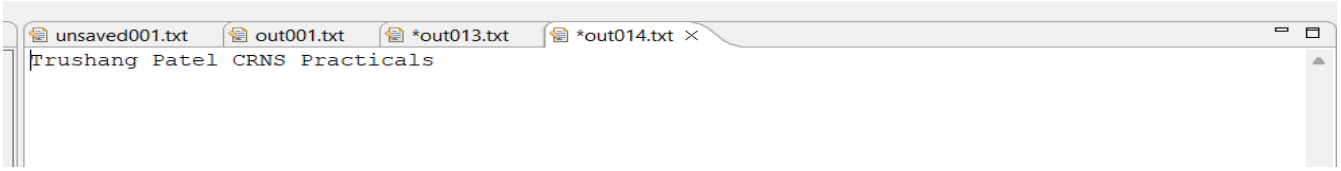


Figure 151:Output of the transposition cipher

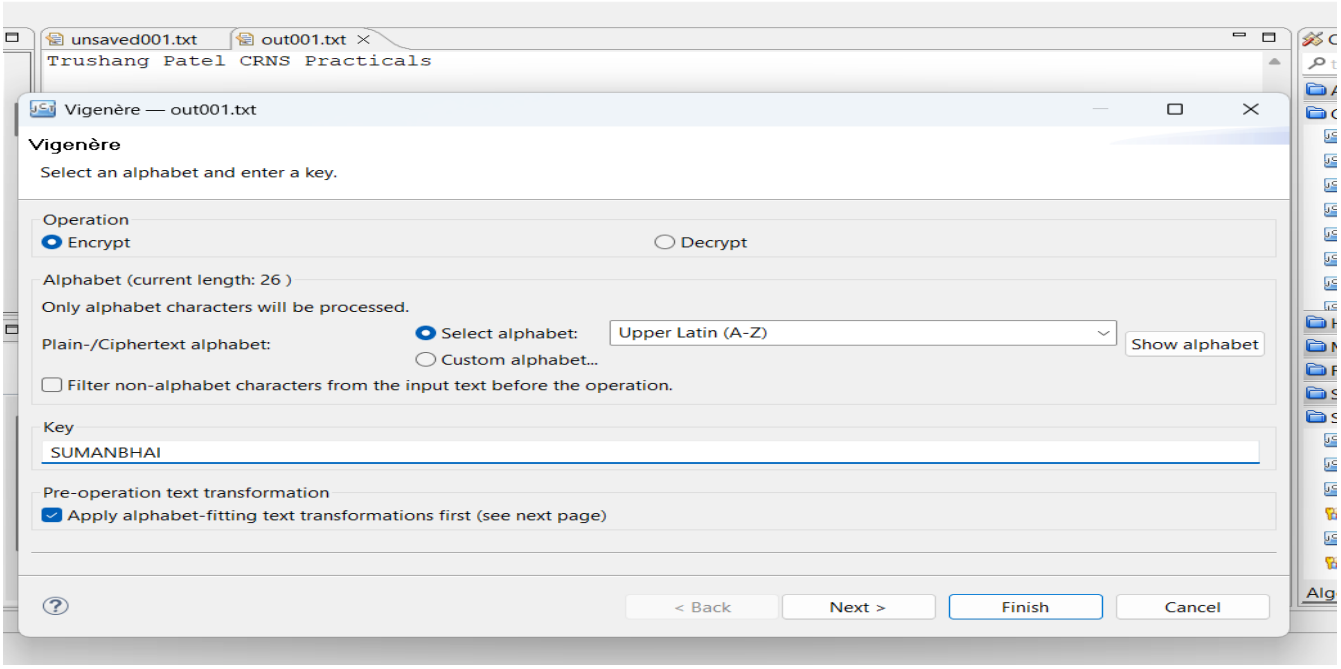


Figure 152:Encryption using Vigenère cipher

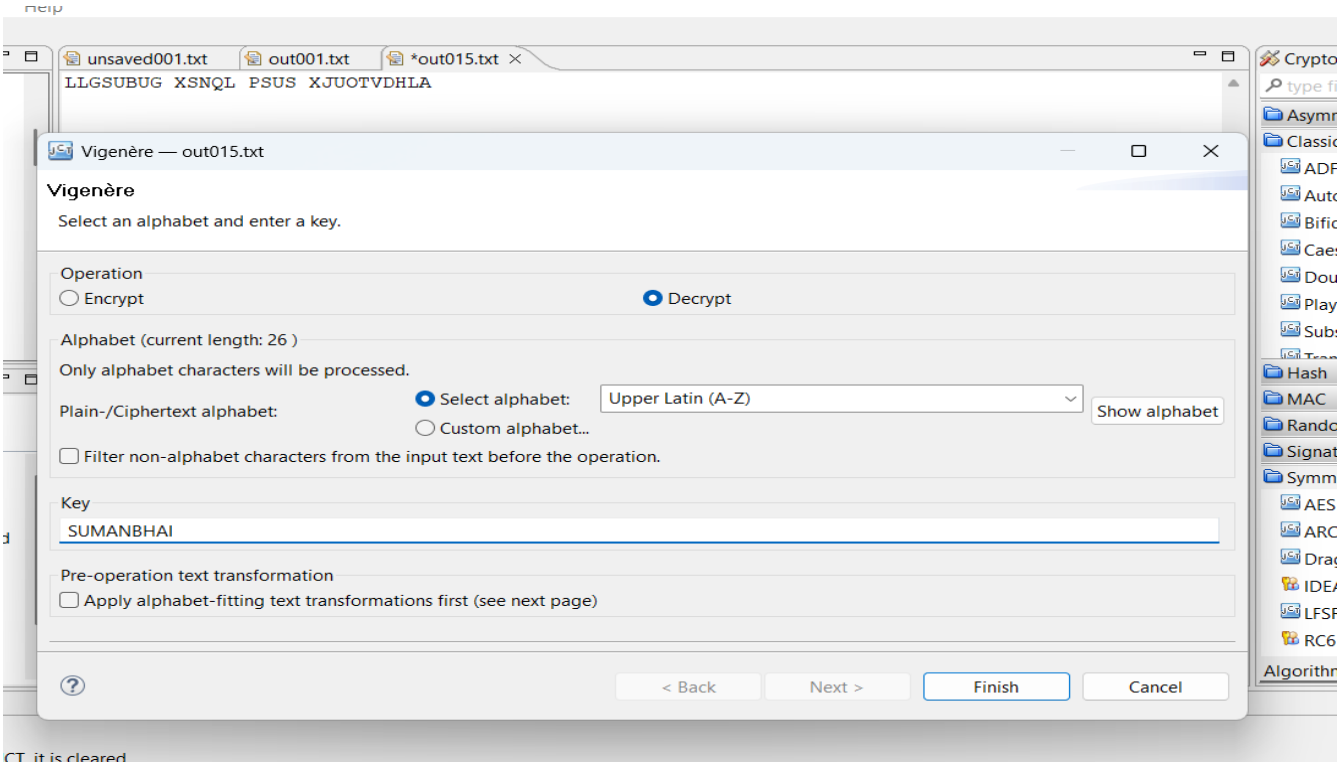


Figure 153:Decryption using Vigenère cipher

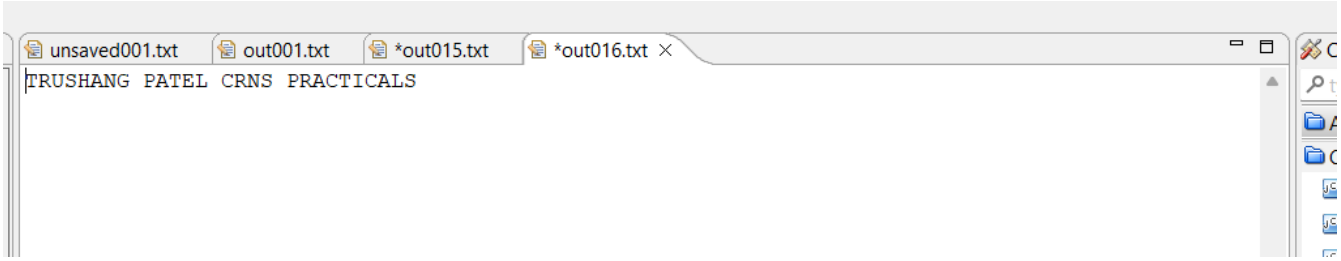


Figure 154:Output of Vigenère cipher

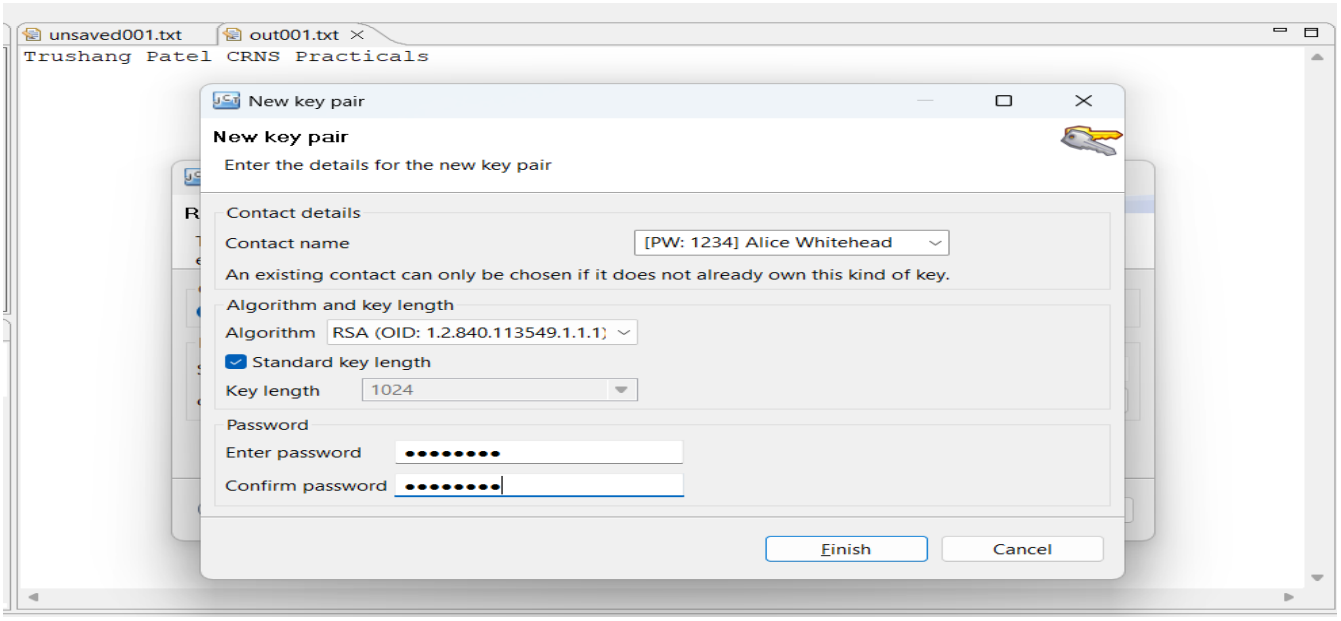


Figure 155:Generate key for RSA

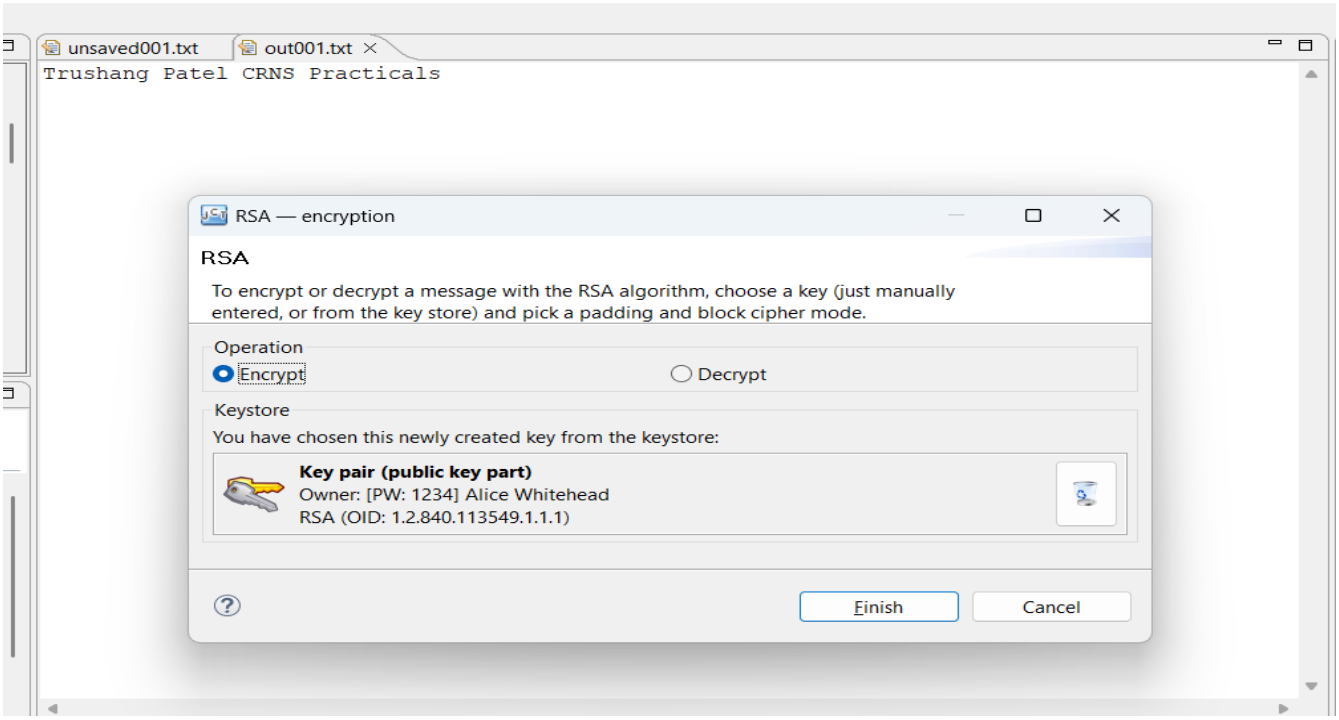


Figure 156:Encryption using RSA

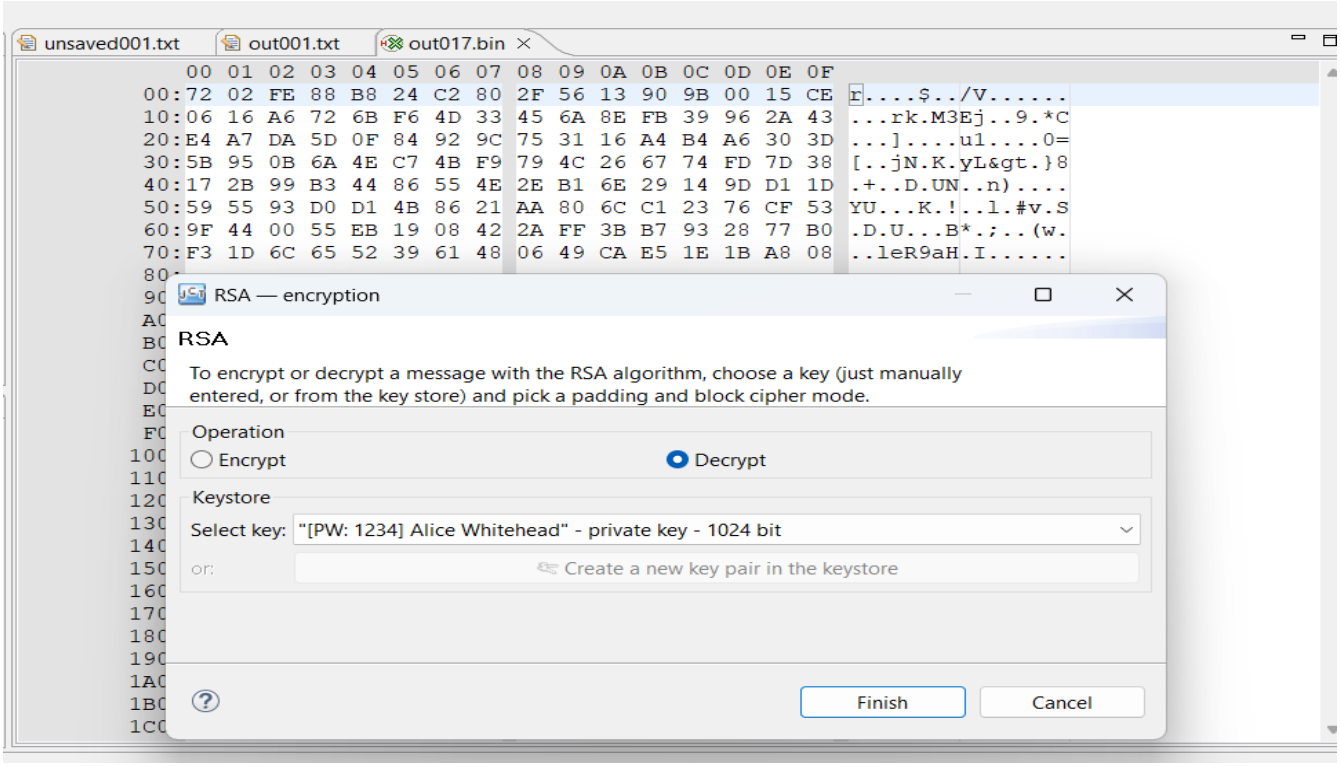


Figure 157:Decryption of the RSA

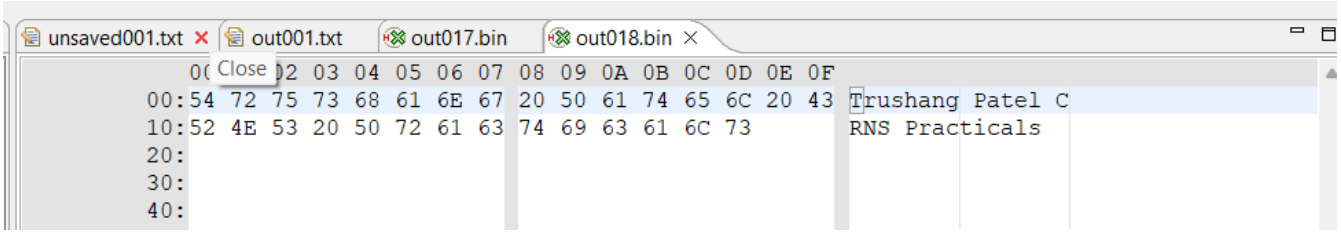


Figure 158:Output of the RSA

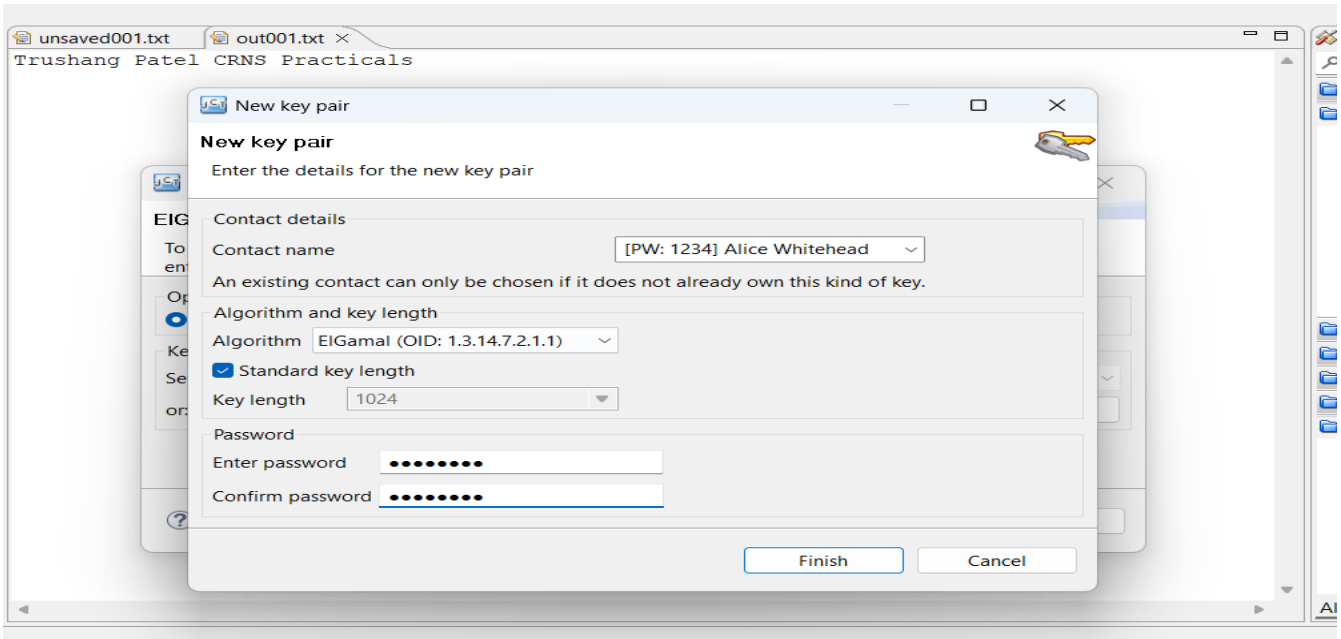


Figure 159:Key Generation using ElGamal

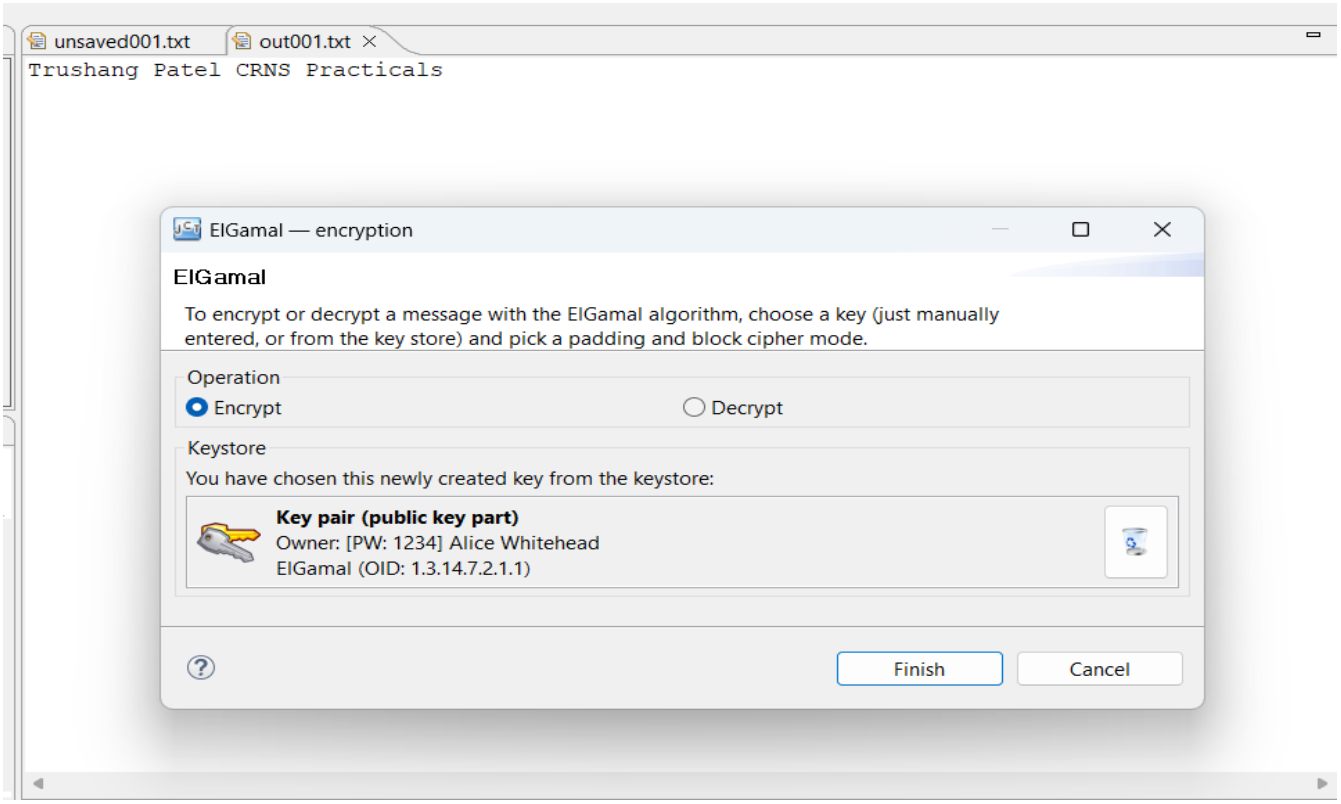


Figure 160:Encryption using ElGamal

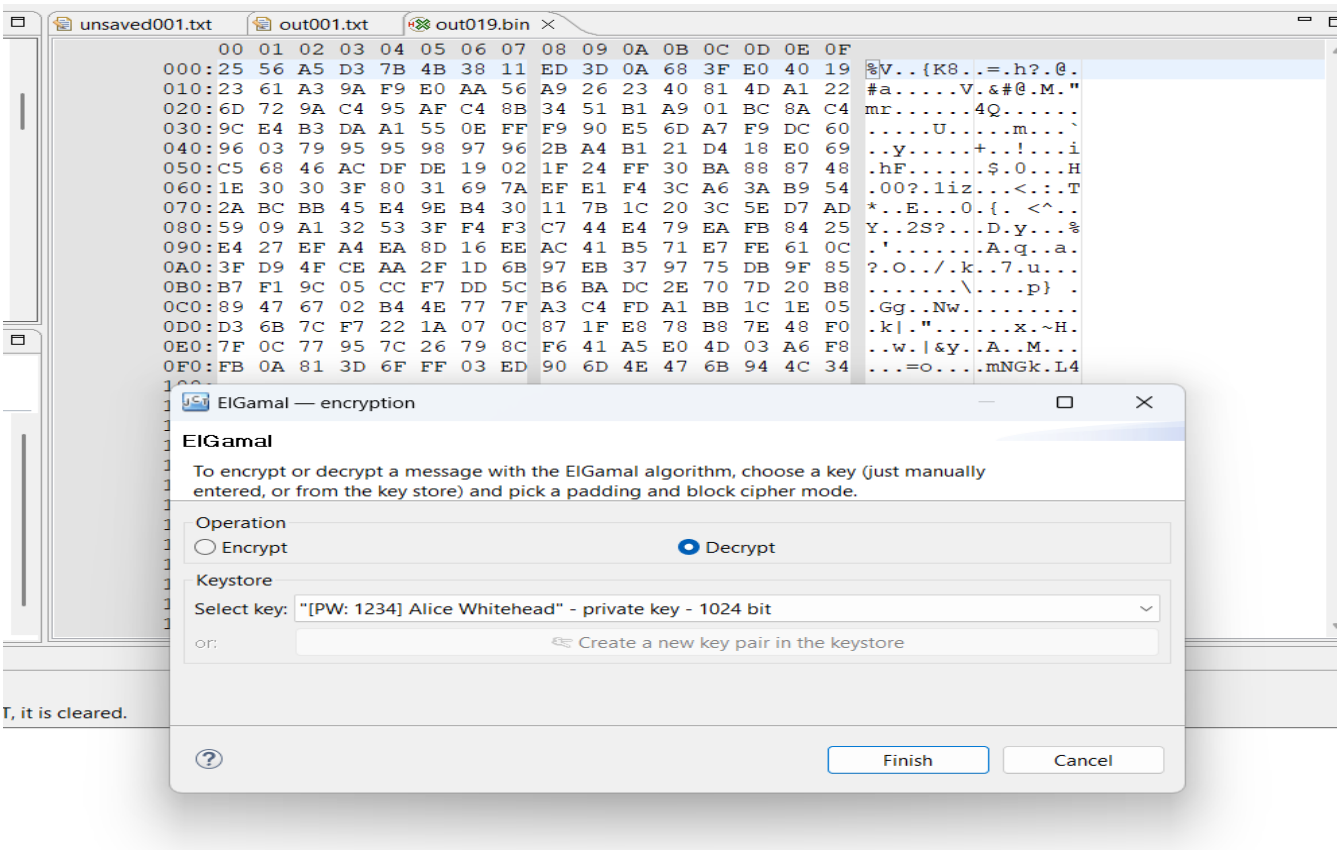


Figure 161:Decryption using ElGamal

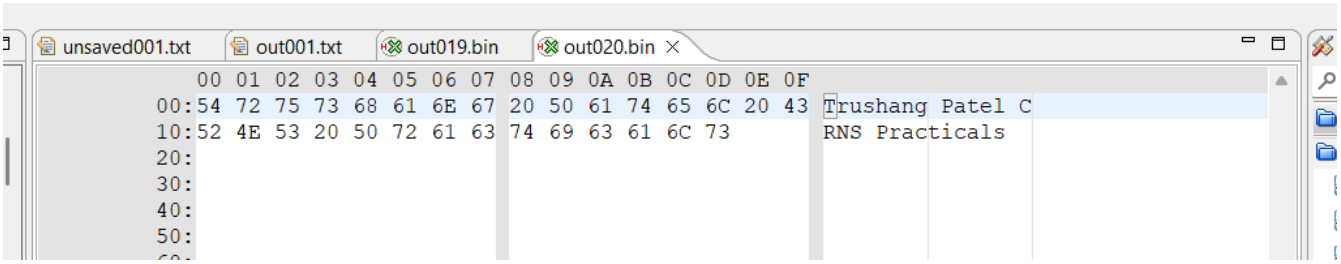


Figure 162:Output of ElGamal

LATEST APPLICATIONS:

- Post-Quantum Cryptography Standards
- Integration of PQC in Messaging Platforms
- Quantum-Resistant Encryption in VPN Services
- Development of Mix Networks for Anonymity

LEARNING OUTCOME:

In this practical, I learned how encryption and decryption work within a cryptosystem and tested their application using various cryptographic algorithms.

REFERENCES:

11. Crypto Tool : https://www.cryptool.org/en/cto/
12. Britannica : https://www.britannica.com/topic/Caesar-cipher
13. GeeksforGeeks : geeksforgeeks.org/transposition-cipher-techniques-in-cryptography/
14. ChatGPT: https://chatgpt.com/

PRACTICAL: 10

AIM:

A mid-sized company plans to enhance network security by deploying a robust firewall solution. The IT security team is tasked with studying and testing various firewall software options to determine the most suitable one based on functionality, ease of use, and performance. To evaluate and compare different firewall software solutions for securing network traffic and preventing unauthorized access in a simulated enterprise environment.

THEORY:

A firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. Firewalls are essential for protecting systems and networks from unauthorized access, malware, and other malicious activities. They can be implemented in both hardware and software and serve as a barrier between trusted internal networks and untrusted external networks, like the internet.

There are several types of firewalls, and the choice of which one to deploy depends on the organization's needs:

- **Packet-Filtering Firewalls**: These are the simplest form of firewalls. They inspect packets of data and determine whether to allow or block them based on IP addresses, port numbers, and protocols. However, they do not track the state of connections, making them less secure.
- **Stateful Inspection Firewalls**: These are more advanced than packet-filtering firewalls, as they track the state of active connections and only allow packets that match an established connection's state. They are more secure and provide more comprehensive monitoring of traffic.
- **Proxy Firewalls**: Proxy firewalls act as intermediaries between the internal network and the external network. They can provide enhanced security by hiding the internal network's details, but they may introduce latency because they must forward requests from the internal network to external destinations and vice versa.
- **Next-Generation Firewalls (NGFW)**: NGFWs combine the features of traditional firewalls with additional layers of security, such as intrusion prevention, application awareness, and advanced threat protection. They are capable of detecting more sophisticated attacks, such as zero-day exploits, and they often include VPN support.

Key Functions of Windows Defender Firewall

1. **Traffic Filtering**:
    - Windows Defender Firewall filters traffic based on rules you set. It decides which network traffic can pass through (allow) and which should be blocked (deny).
    - It manages both **inbound traffic** (data coming into your system) and **outbound traffic** (data leaving your system).
2. **Security**:
    - It is primarily designed to prevent unauthorized access to your computer from external sources, such as hackers or malicious software attempting to exploit your computer.

- It blocks connections to untrusted devices or websites and ensures only authorized apps, services, or users can communicate with your computer.

3. **Application Control**:
   - The firewall works in conjunction with specific applications on your system. It can block or allow access to individual programs, ensuring that only approved applications can connect to the network.

Components of Windows Defender Firewall

1. **Inbound Rules**:
   - Inbound rules control incoming traffic (from the internet or local network) trying to access services, programs, or resources on your computer.
   - For example, if someone tries to access a web server running on your machine, inbound rules decide whether that traffic should be allowed or blocked.
2. **Outbound Rules**:
   - Outbound rules control traffic leaving your computer. These rules govern which applications or services on your system are allowed to send data to external networks.
   - For example, you can configure outbound rules to block specific apps (such as a torrent client or browser) from accessing the internet.
3. **Connection Security Rules**:
   - These rules are used to establish and maintain secure connections. They manage **IPsec (Internet Protocol Security)**, a set of protocols that encrypt and authenticate data packets.
   - IPsec is used to secure communication between devices by ensuring that only authorized devices can communicate with each other.
4. **Profiles**:
   - Windows Defender Firewall uses **profiles** to manage how the firewall behaves depending on the type of network your computer is connected to. The three types of network profiles are:
     - **Domain Profile**: Applied when your computer is connected to a corporate network that is part of a domain.
     - **Private Profile**: Applied when your computer is connected to a private, trusted network, like a home Wi-Fi network.
     - **Public Profile**: Applied when your computer is connected to a public network, such as a café or airport Wi-Fi. The strictest rules are typically applied in public networks.
5. **Logging**:
   - Windows Defender Firewall can log events to help administrators track traffic and identify potential issues or threats. This can include failed attempts to connect, blocked traffic, or system events.
   - Logs are stored in the **Event Viewer** and can be analyzed to identify patterns of suspicious activity.

How Windows Defender Firewall Works

1. **Filtering Based on Rules**:
   - When a packet of data is sent to or from your system, the firewall evaluates it against the existing rules. These rules specify whether certain types of traffic should be allowed or denied.

   - **Rules can be customized** by the user or administrator. For example, you could allow incoming traffic for a web server on port 80 (HTTP) or block outgoing connections for a specific application.
2. **Profile-Based Filtering**:
   - Depending on the network your computer is connected to, the firewall switches between different profiles, adjusting security levels based on the environment.
   - **Domain Profile** has more relaxed rules compared to **Public Profile**, as the latter is typically used in public places where more stringent security is required.
3. **Default Behavior**:
   - By default, Windows Defender Firewall **blocks all inbound traffic** that is not explicitly allowed by a rule.
   - For outbound traffic, it is typically allowed unless specified otherwise in the firewall rules.
4. **Allowing and Blocking Applications**:
   - Applications and services on your computer can request permission from the firewall to access the network. You can choose whether to allow or block specific applications, such as web browsers, email clients, or even background services.
5. **Stateful Packet Inspection (SPI)**:
   - Windows Defender Firewall performs Stateful Packet Inspection, which means it tracks the state of network connections. It looks at the entire context of the traffic, not just individual packets.
   - For instance, if you initiate a connection to a server, the firewall will "remember" that connection and allow the response from the server to reach you, but it will block any unsolicited responses.

Types of Rules in Windows Defender Firewall

1. **Inbound Rules**:
   - Govern the traffic coming into your system. For example, you can set up inbound rules for allowing access to services such as **HTTP**, **FTP**, or **RDP** (Remote Desktop Protocol).
   - Common scenarios for inbound rules:
     - Allowing a web server (port 80/443) to receive incoming web traffic.
     - Blocking all incoming traffic on an unused port to reduce the attack surface.
2. **Outbound Rules**:
   - Control which applications or processes can send data out of your computer to external networks.
   - Common scenarios for outbound rules:
     - Allowing your browser to access the internet but blocking other apps (like a game) from accessing it.
     - Preventing malware or unauthorized applications from making outbound connections.
3. **Connection Security Rules**:
   - These are used to configure security measures like **IPsec**, which secures traffic between devices on a network.
   - Connection security rules are typically used in corporate or enterprise environments to ensure secure communications.

Creating and Managing Firewall Rules

1. **Create a Rule**:
    - Open **Windows Defender Firewall with Advanced Security** (can be accessed by searching for it in the Start menu).
    - Choose either **Inbound Rules** or **Outbound Rules** on the left sidebar, then select **New Rule**.
    - You can create rules based on:
        - **Program**: Allow or block specific applications.
        - **Port**: Allow/block specific ports.
        - **Predefined**: Choose predefined rules for specific services (e.g., **File and Printer Sharing**).
        - **Custom**: Create highly specific rules based on multiple criteria (program, port, IP address).
2. **Configuring a Rule**:
    - After selecting the rule type, you'll specify what action to take (allow or block).
    - You can also specify the conditions for when the rule should apply (e.g., only on a private network, only for specific IP addresses, etc.).
3. **Managing Rules**:
    - You can enable, disable, or delete rules by selecting them in the list of inbound or outbound rules.
    - You can also prioritize the rules based on their importance.

Best Practices for Windows Defender Firewall

1. **Use Default Settings Where Possible**:
    - For most users, the default firewall settings provide a good level of protection. Don’t disable the firewall unless absolutely necessary.
2. **Enable Logging**:
    - Enable logging to help you track unusual network activity and identify potential threats.
3. **Review and Update Firewall Rules Regularly**:
    - Periodically review the rules to ensure they're up to date and that no unnecessary open ports or services exist.
    - Remove any unused or redundant rules to minimize potential vulnerabilities.
4. **Use Profiles Based on Your Network**:
    - Configure the firewall profiles properly based on your network type (home, work, public). Use stricter rules on public networks to reduce exposure to threats.

Some of the top firewall include:

1. **pfSense** (Open-source):

    - Advantages: Highly customizable and flexible, ideal for tech-savvy organizations.
    - Features: Stateful firewall, VPN support, and advanced logging capabilities.
    - Ideal For: Companies with an in-house IT team that can manage and configure it.

2. **FortiGate** (Commercial, Next-Generation Firewall):

    - Advantages: Strong performance, high availability, and advanced security features like IDS/IPS and DDoS protection.
    - Features: NGFW with centralized management, high throughput, and scalability.
    - Ideal For: Medium to large enterprises looking for an all-in-one security solution.

3. **Check Point Quantum Firewall**:

    - Advantages: Strong integration with cloud environments, centralized management, and ease of use.
    - Features: VPN support, threat prevention, and application control.
    - Ideal For: Enterprises needing a combination of strong network security and simplified management.

4. **Palo Alto Networks PA-Series**:

    - Advantages: Highly detailed traffic analysis and real-time protection.
    - Features: Advanced threat protection, application visibility, and user identification.
    - Ideal For: Organizations with high-performance needs and complex network environments.

5. **Cisco ASA**:

    - Advantages: Proven track record in enterprise environments with strong support and extensive documentation.
    - Features: VPN, content filtering, and support for a variety of security protocols.
    - Ideal For: Enterprises already using Cisco infrastructure.

ESET's personal firewall, integrated within ESET Internet Security and ESET Smart Security Premium, offers advanced features beyond those of the built-in Windows Defender Firewall. While Windows Defender Firewall provides essential inbound and outbound traffic filtering based on user-defined rules, ESET's firewall enhances this with interactive user prompts for outbound connections, allowing for more granular control over application network access.

Additionally, ESET's firewall includes Intrusion Detection System (IDS) capabilities, which monitor network traffic for suspicious activities and potential threats—a feature not natively available in Windows desktop versions. It's important to note that running two firewalls simultaneously can lead to conflicts; therefore, upon installation, ESET's firewall automatically disables Windows Defender Firewall to prevent such issues.

CODE:

```
N/A
```

OUTPUT:

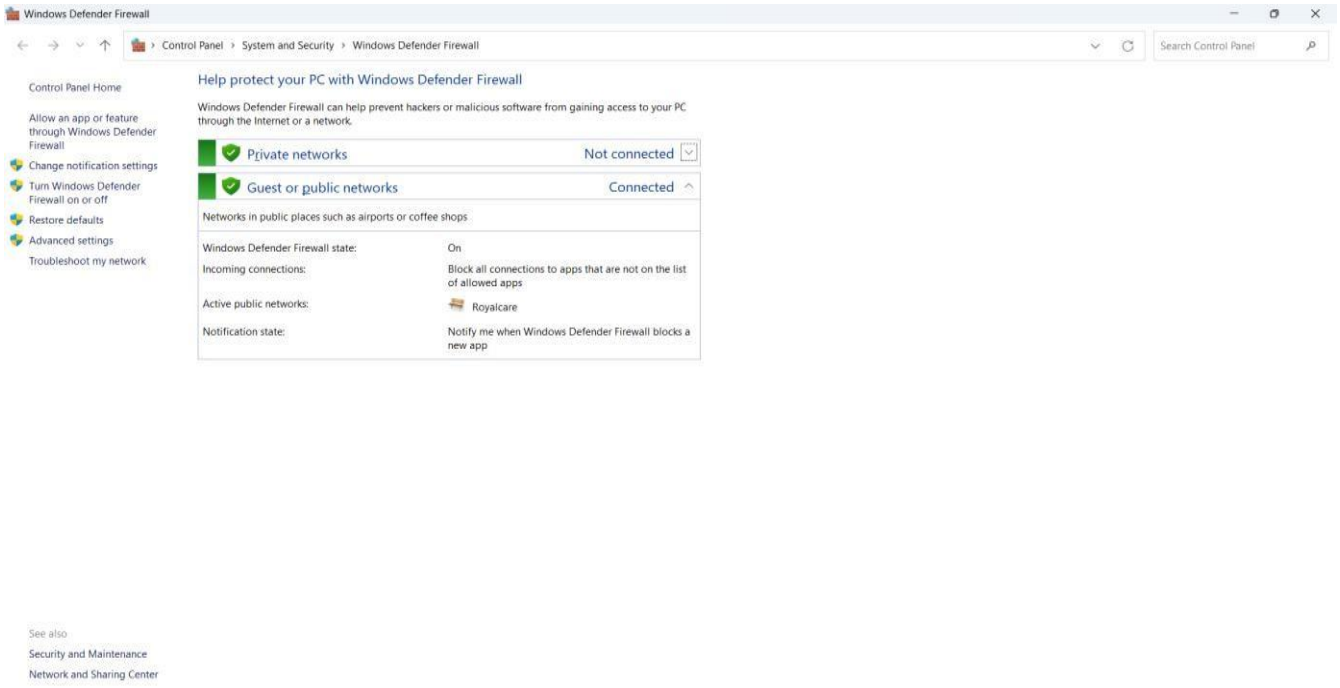


Figure 1: Windows Defender Firewall main control panel

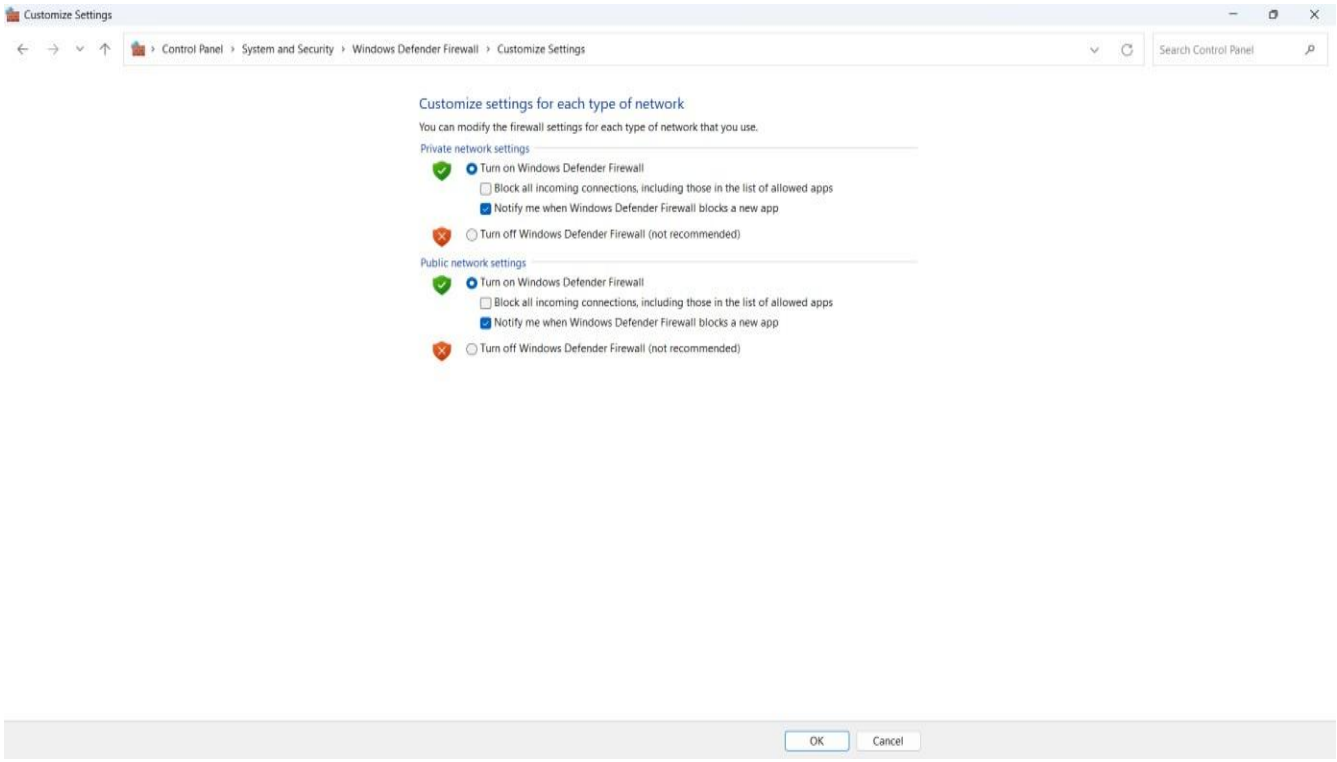


Figure 2: Enable or disable Windows Defender Firewall for different network profiles.

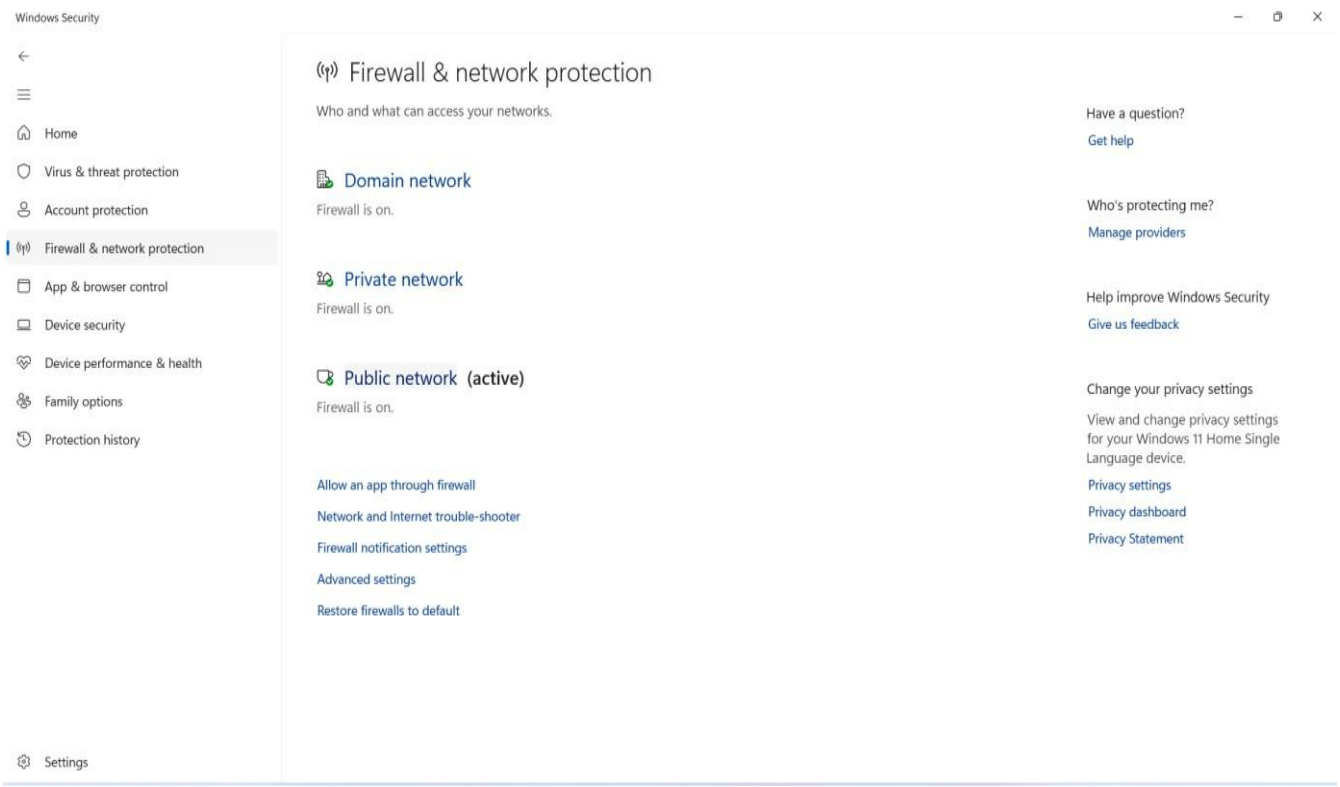


Figure 3: Display of active firewall status

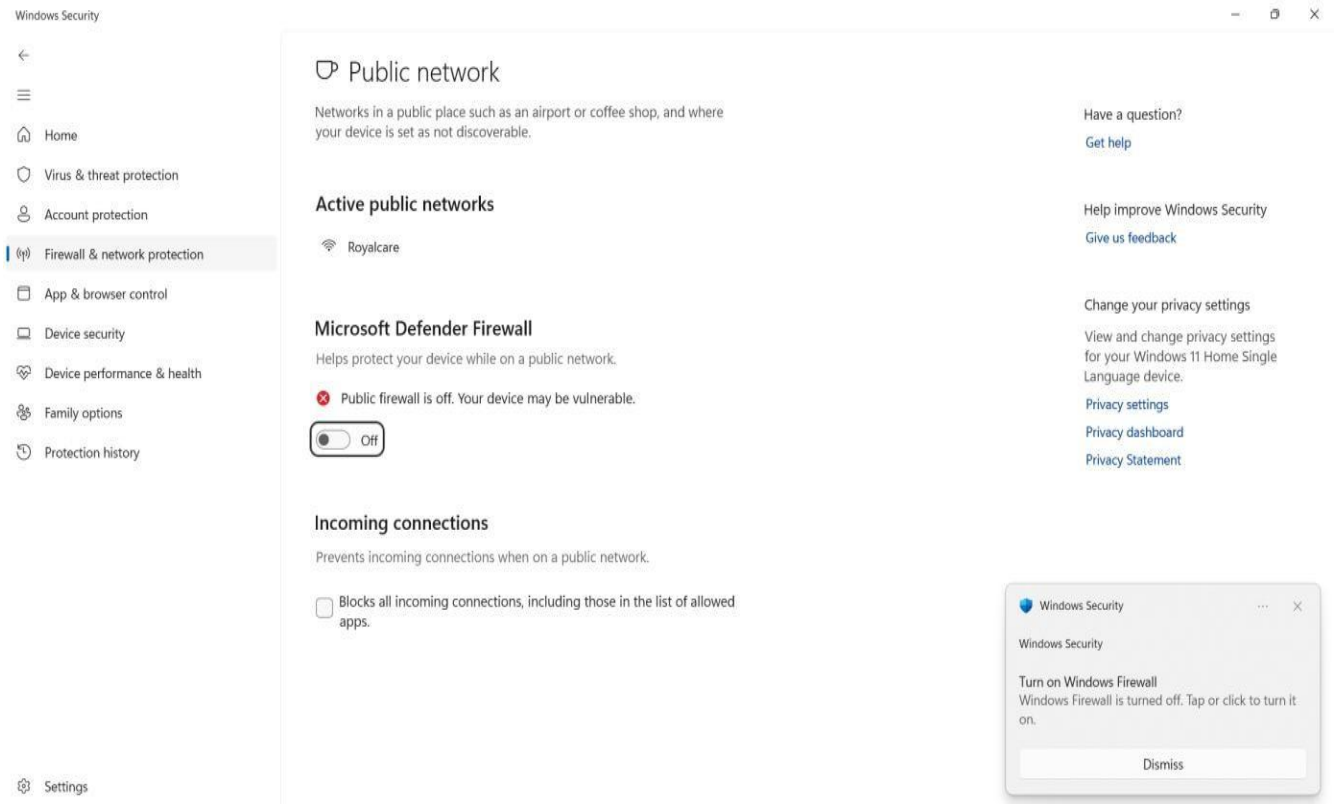


Figure 4: Warning prompt when the firewall is turned off

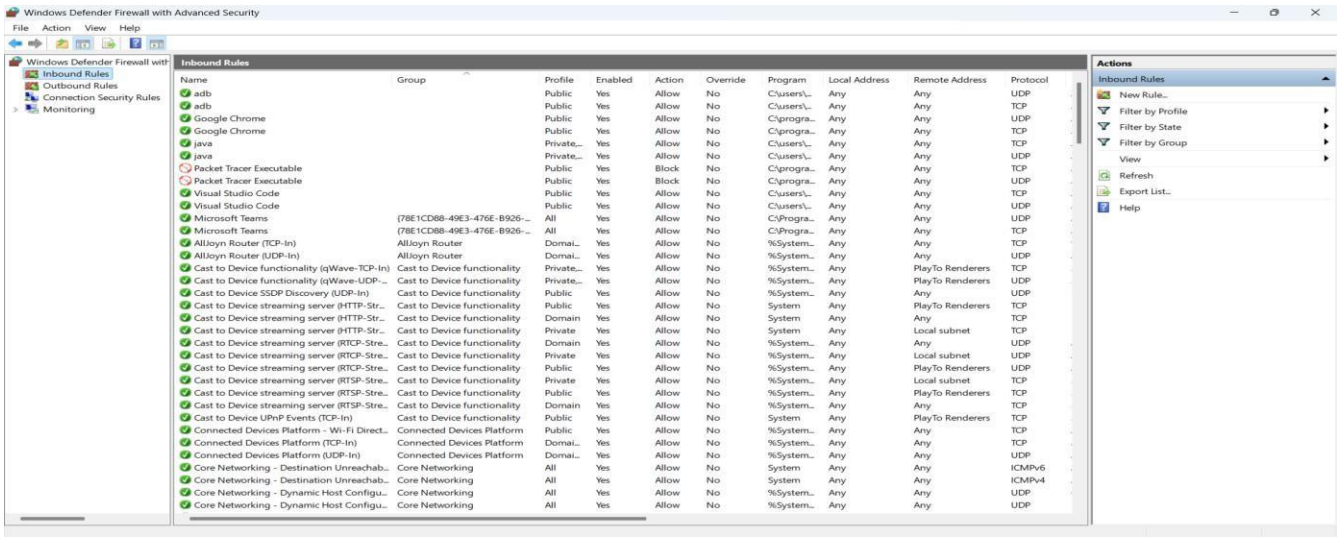


Figure 5: Inbound rule configuration for managing incoming traffic

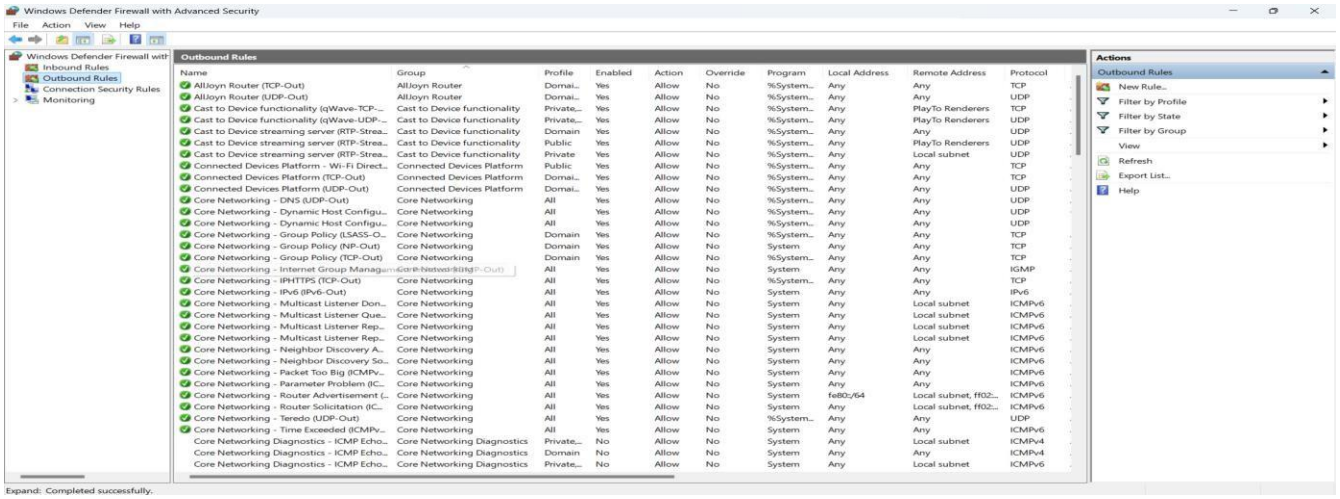


Figure 6: Outbound rule configuration for controlling outgoing traffic

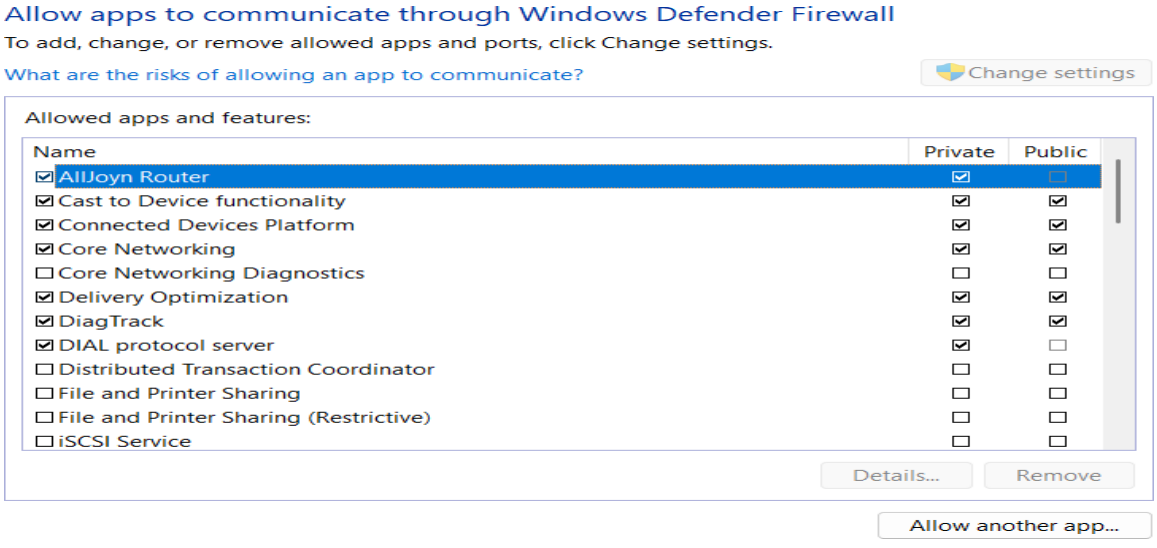


Figure 7: Selecting which applications can communicate through the firewall

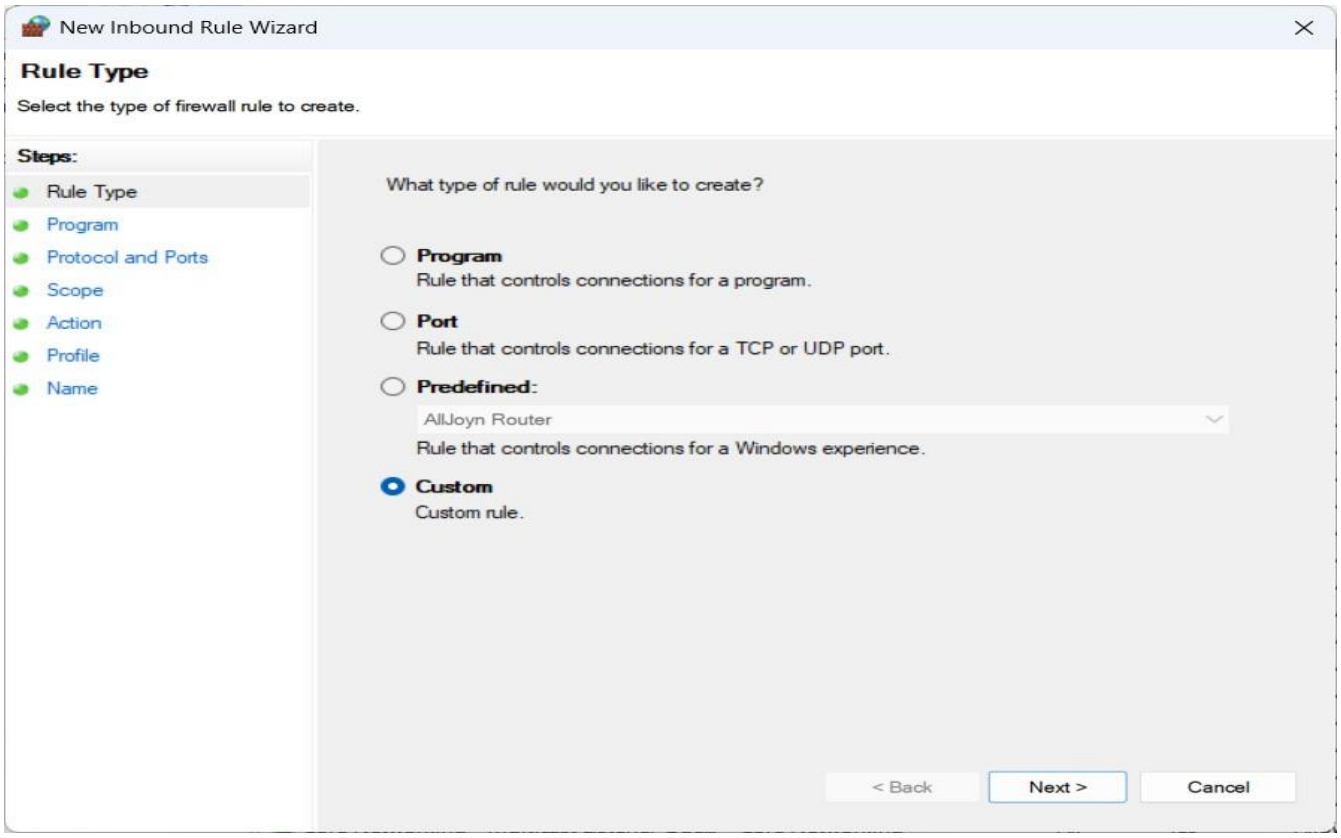


Figure 8: Creating a custom inbound rule with tailored settings

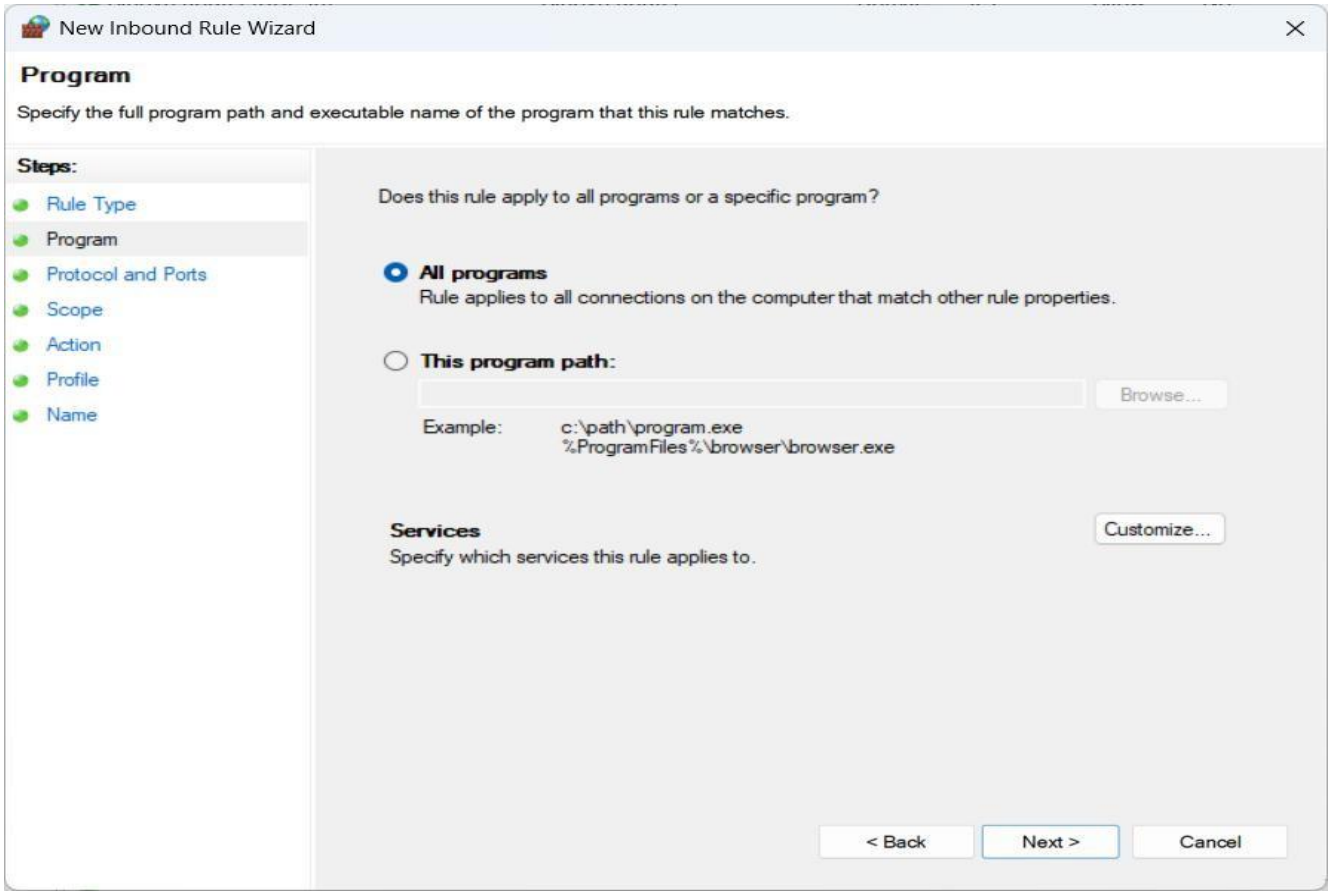


Figure 9: Define rule criteria for specific programs or all programs

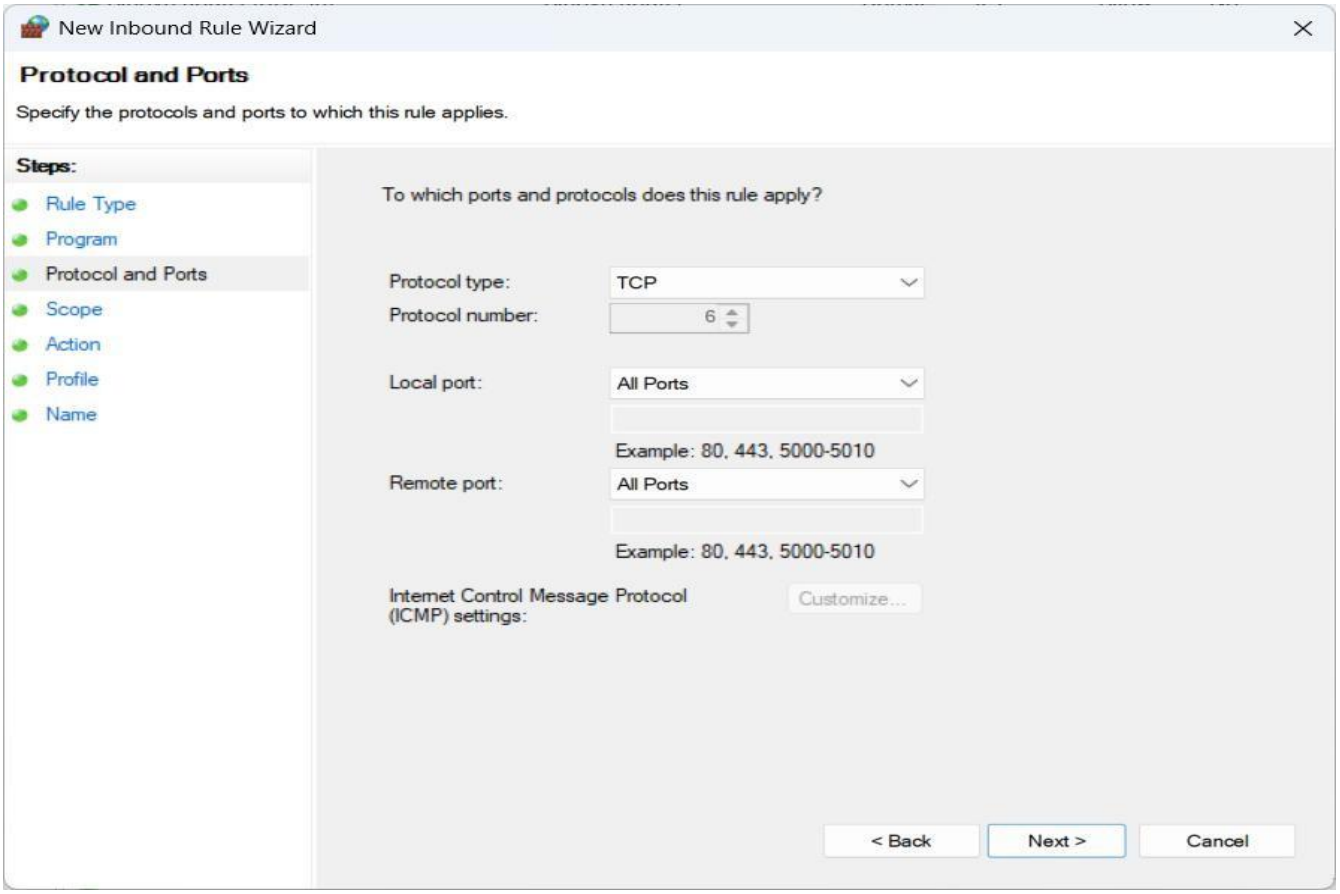


Figure 10: Set protocols and ports for firewall rules

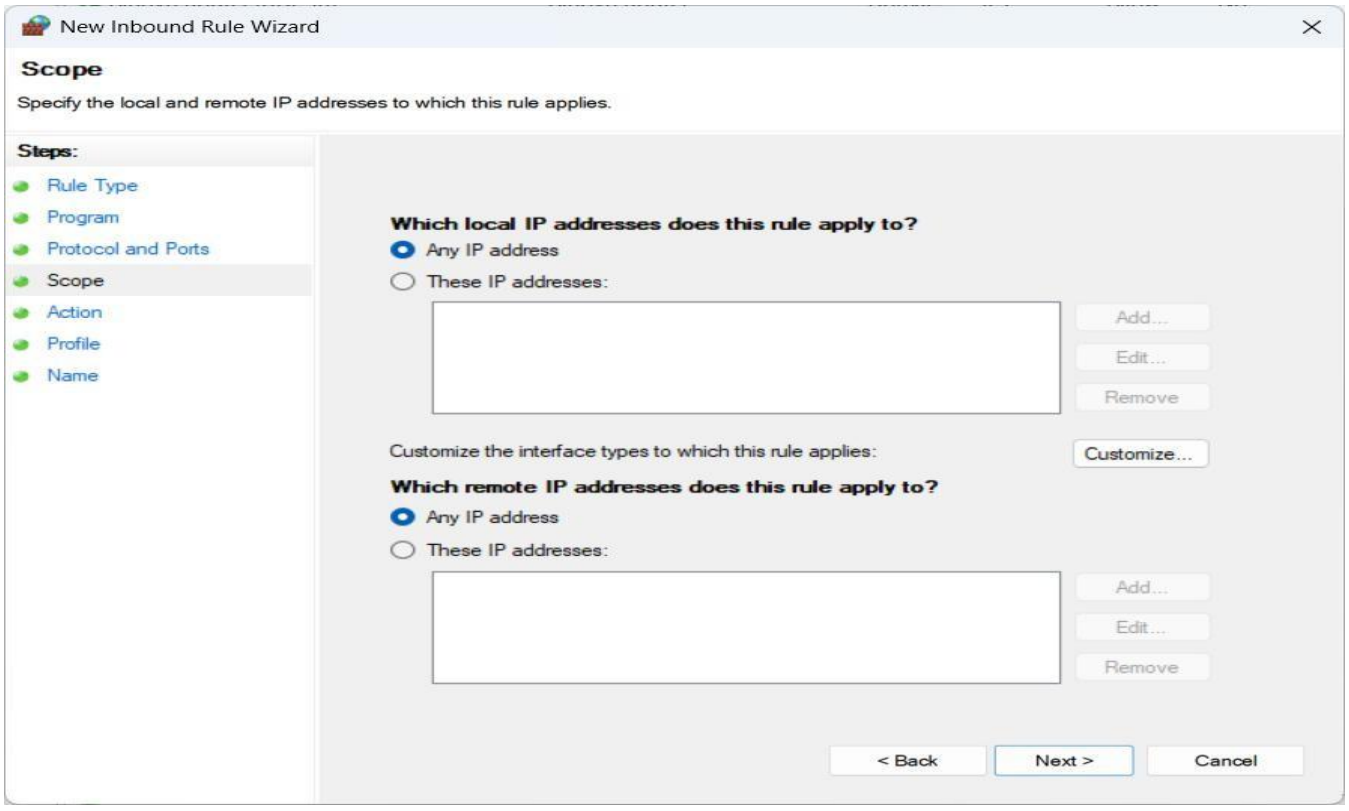


Figure 11: Define IP addresses to target specific connections

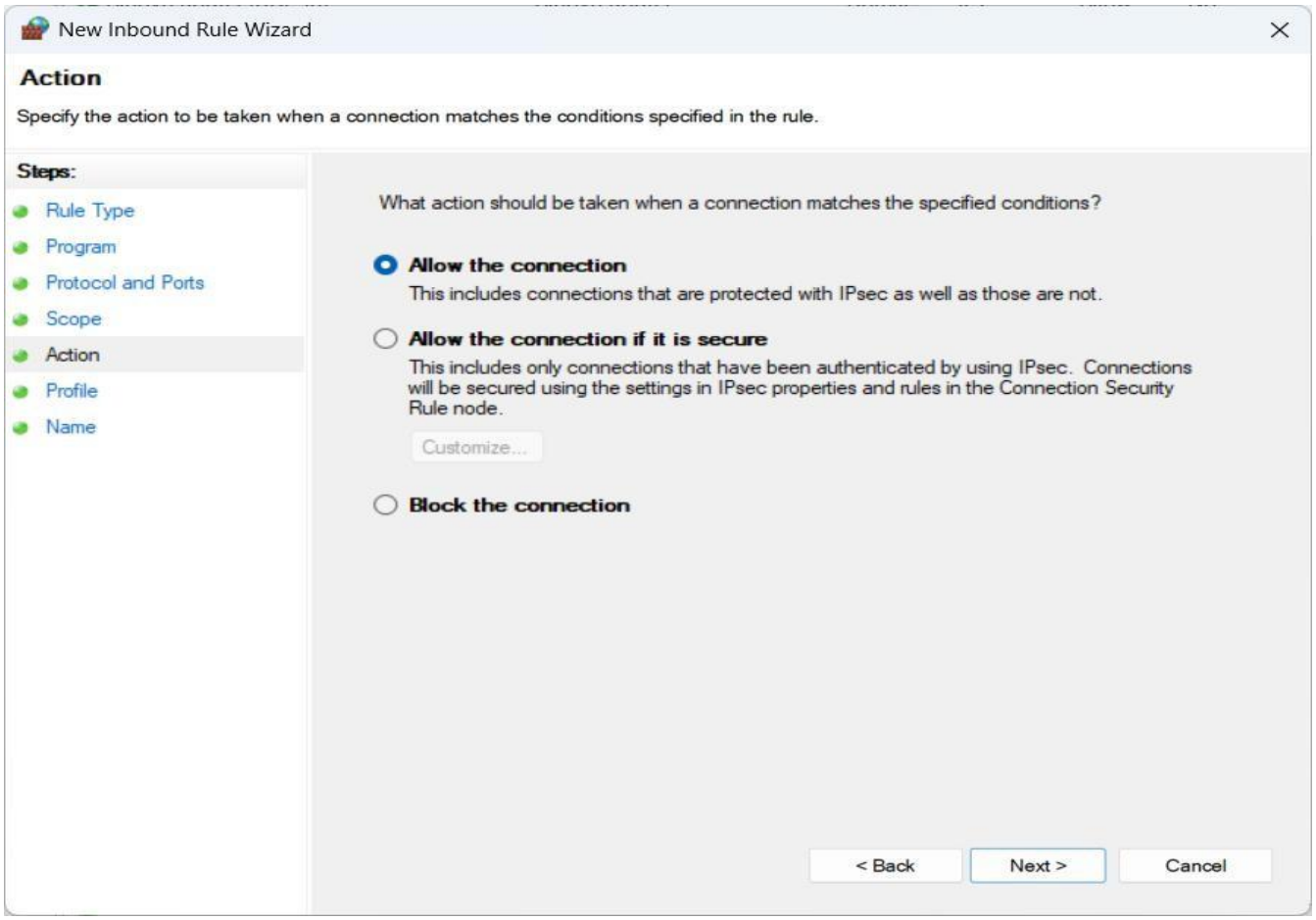


Figure 12: Configure actions — allow or block connections

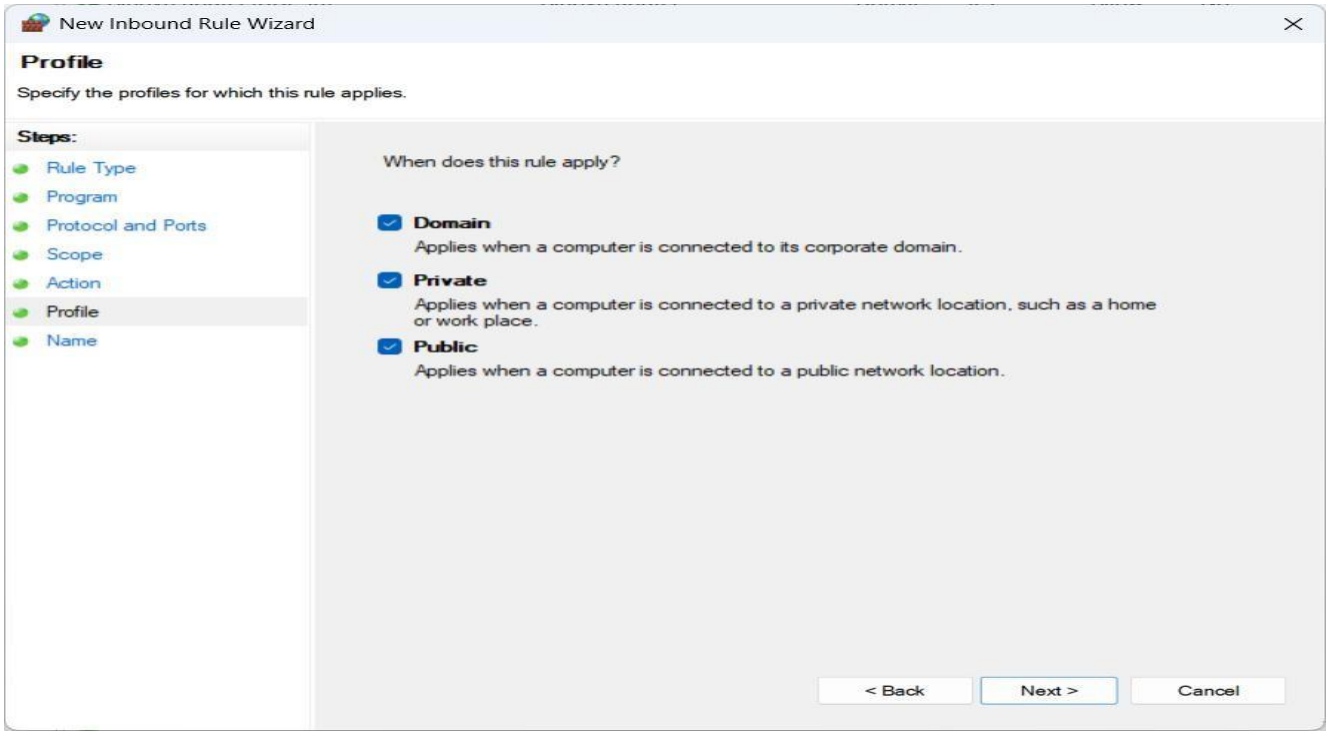


Figure 13: Apply rules to specific network profiles

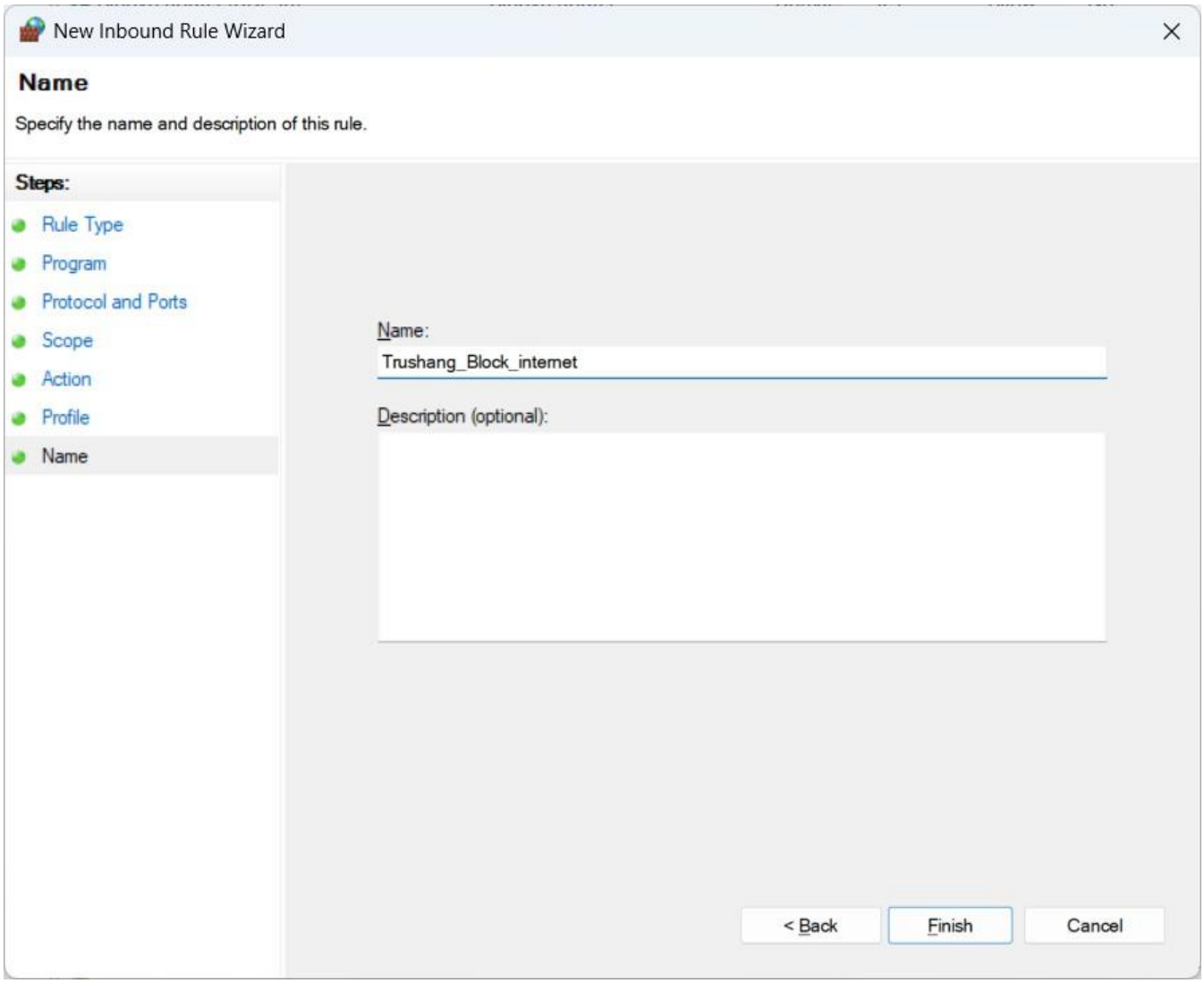


Figure 14: Naming the firewall rule for better management

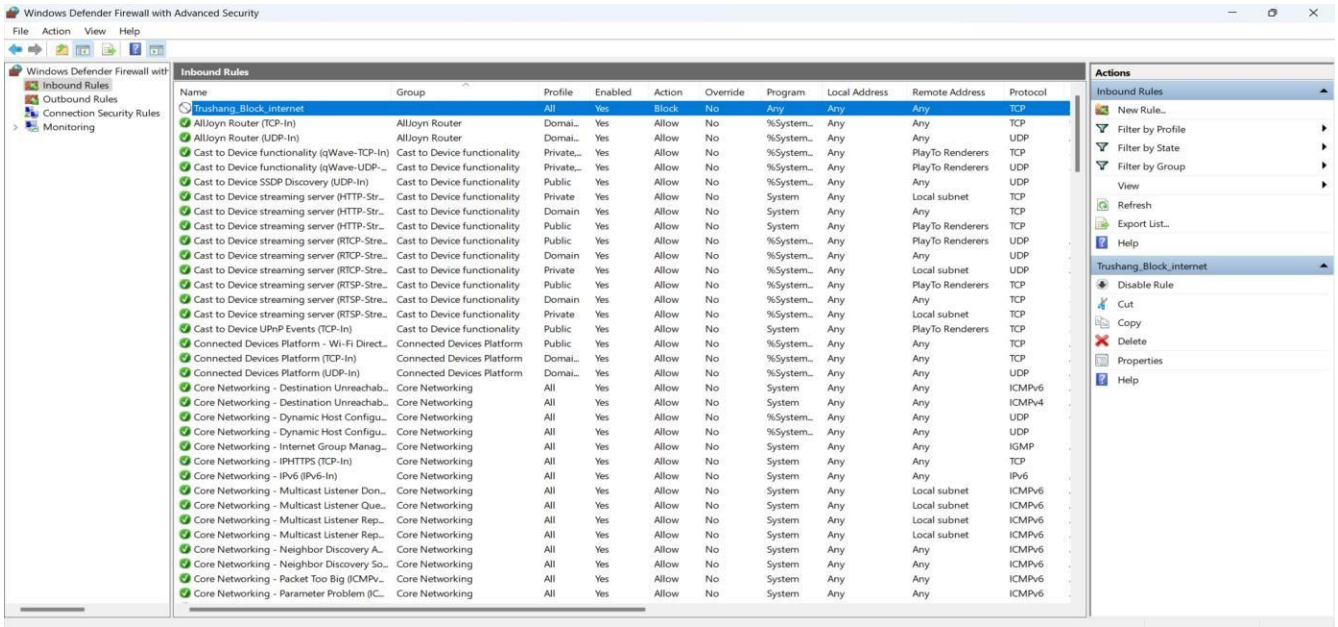


Figure 15: Example of creating an inbound rule to block internet access

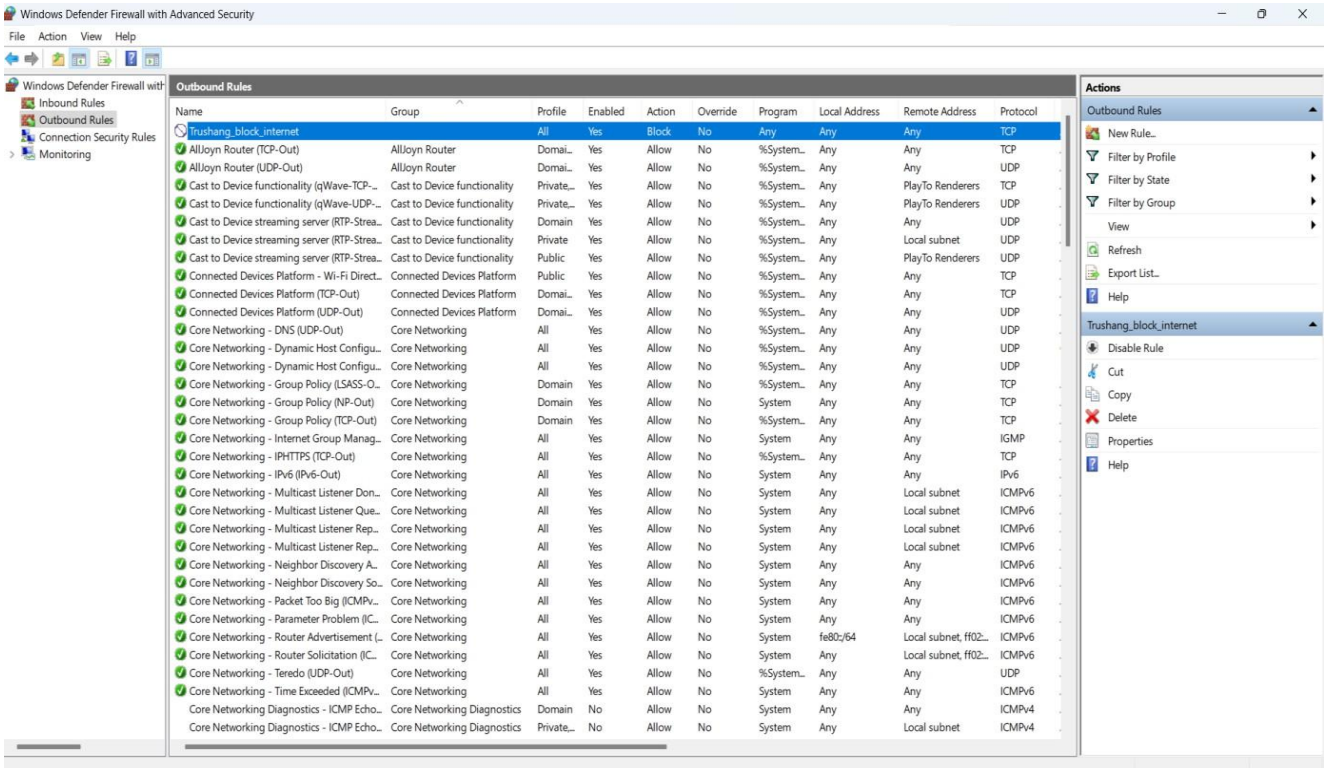


Figure 16: Example of creating an outbound rule to block internet access

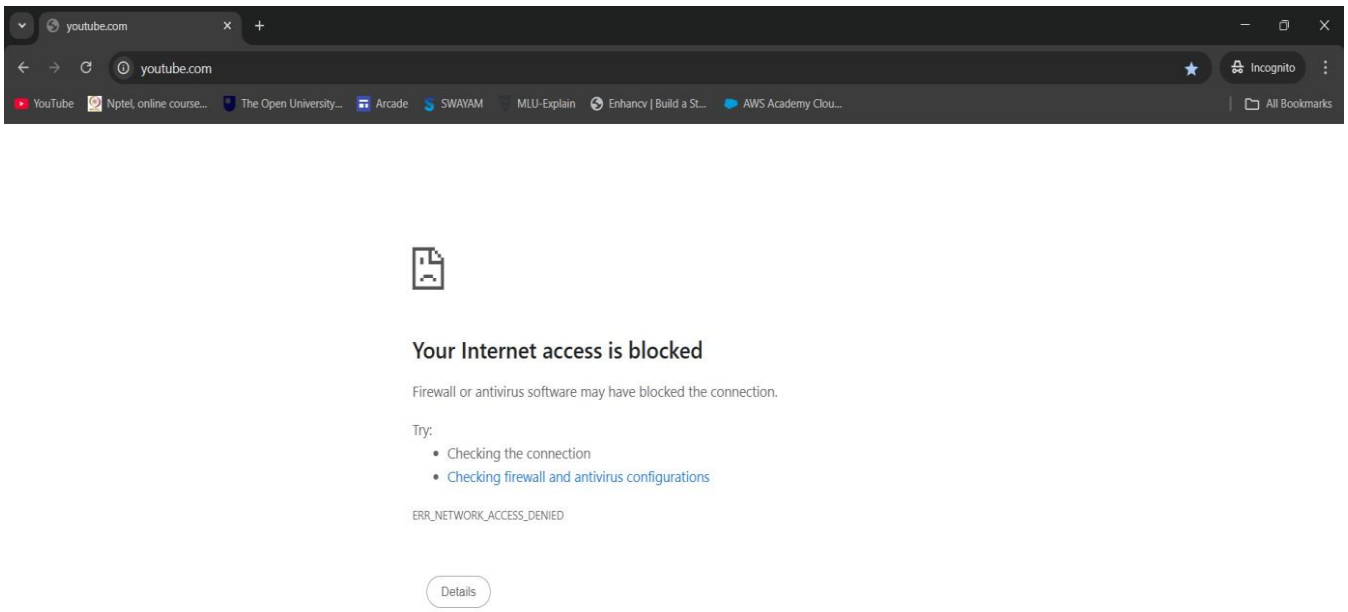


Figure 17: Verification — internet access is successfully blocked

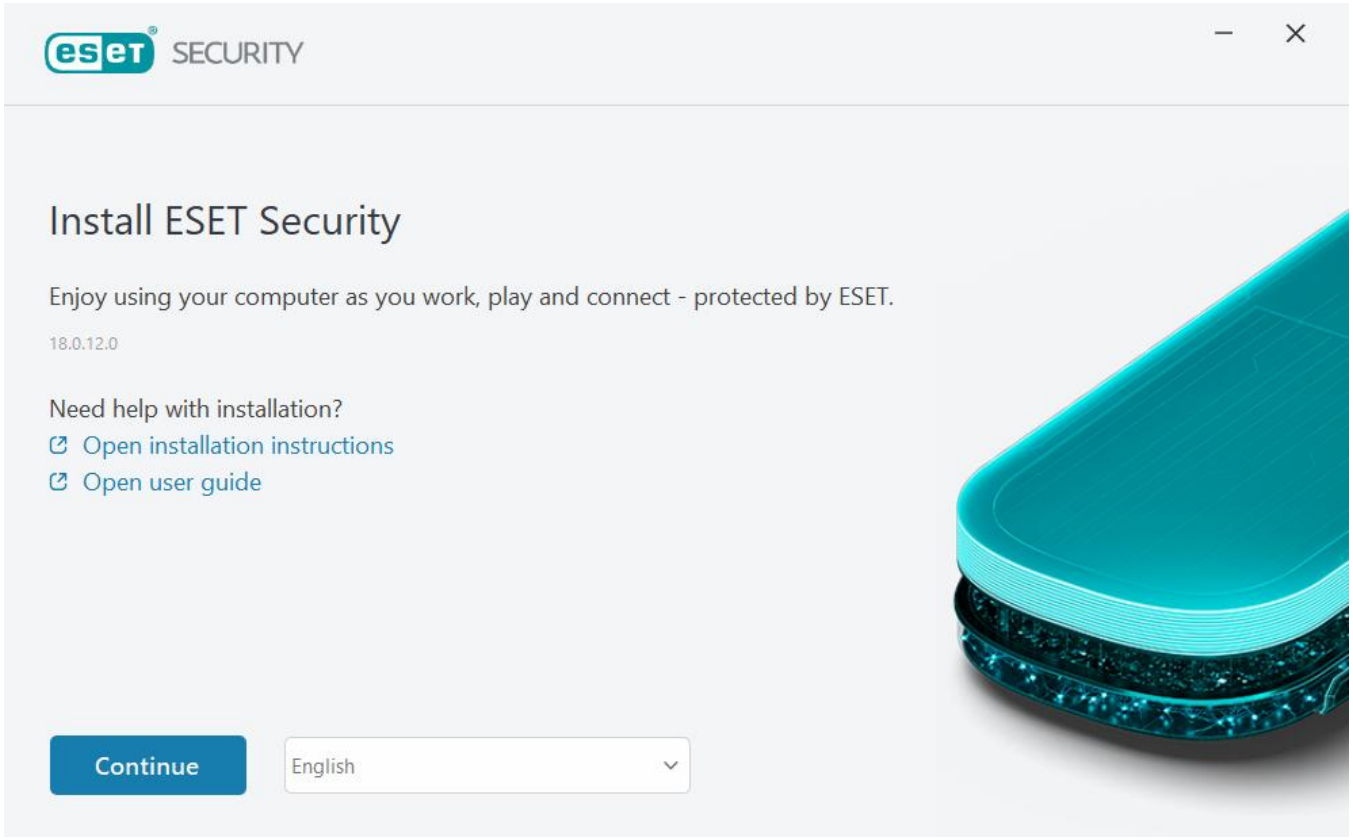


Figure 18: Installing ESET Personal Firewall trial version

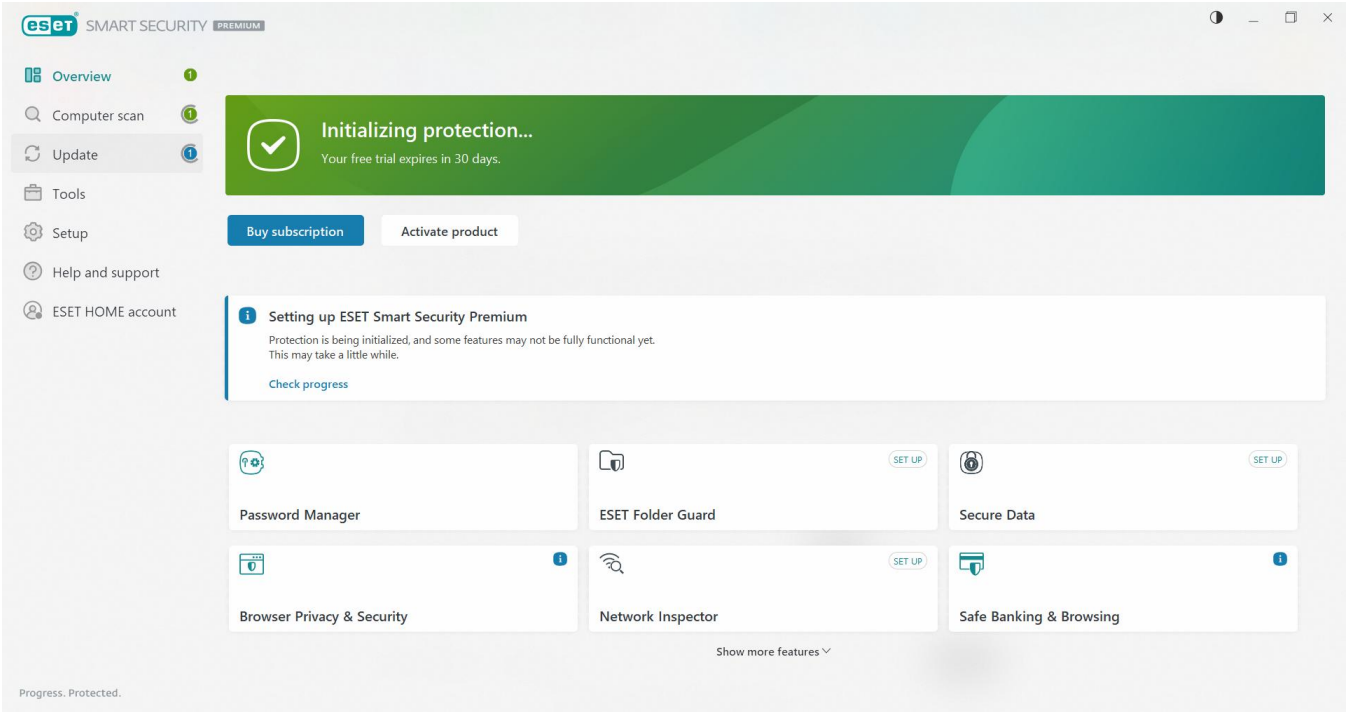


Figure 19: ESET security dashboard overview

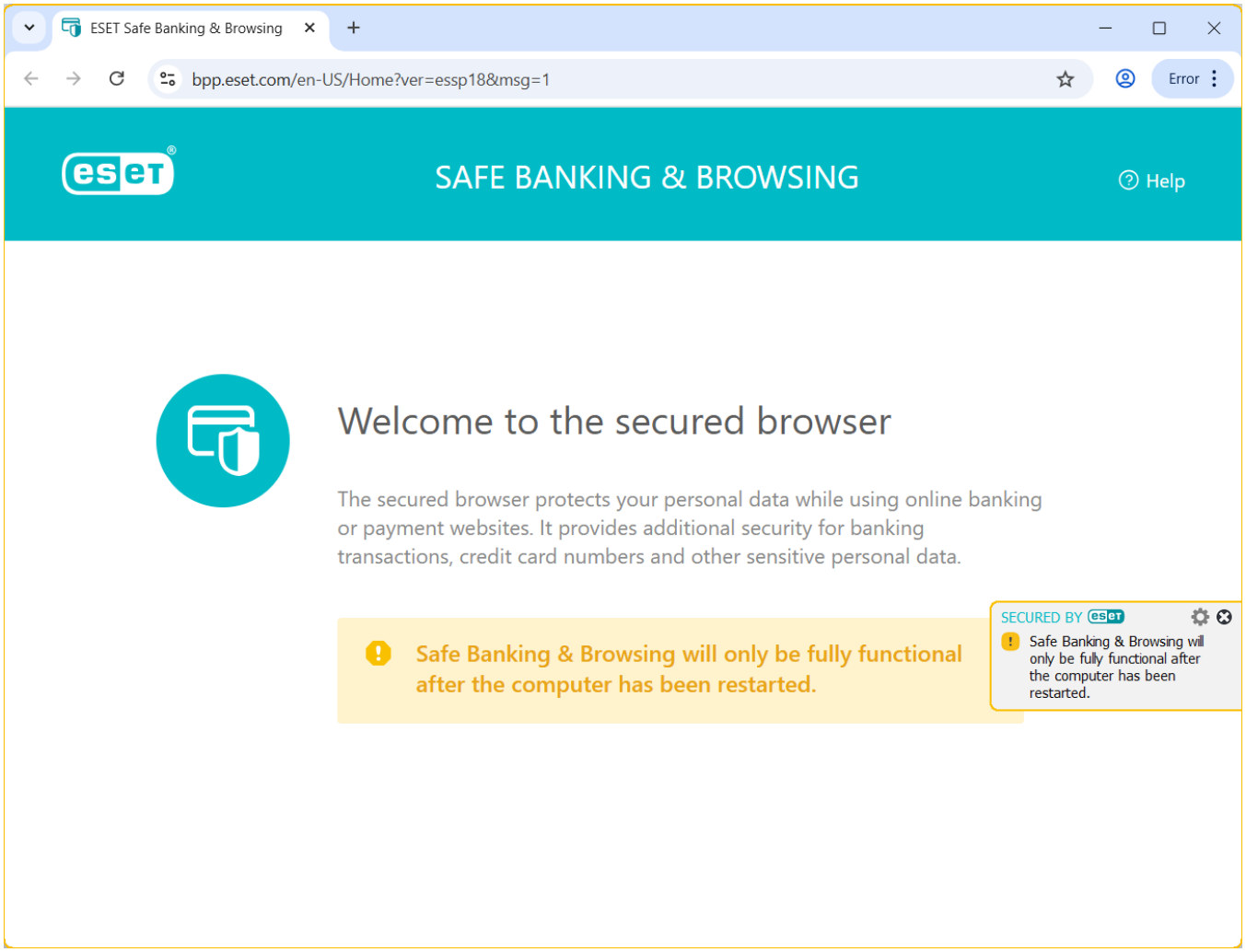


Figure 20: ESET's secure browser for safe banking transactions

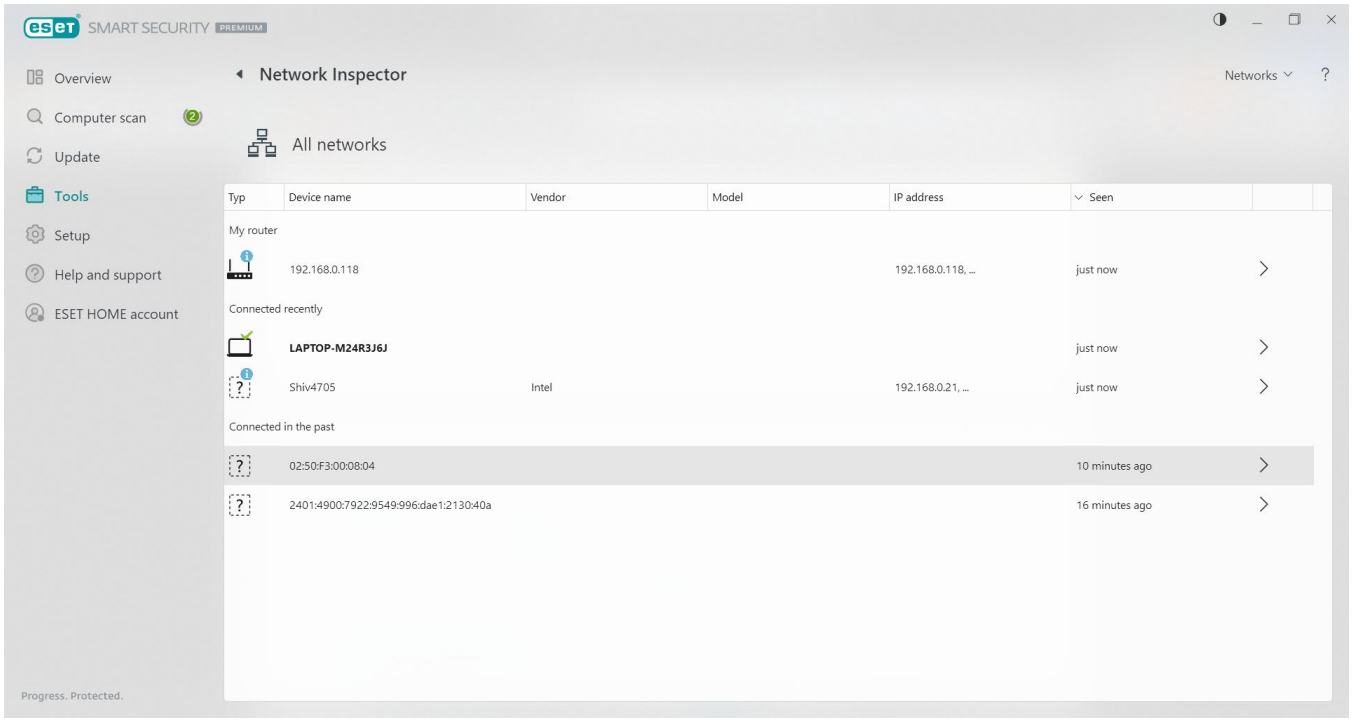


Figure 21: ESET Network Inspector showing connected devices

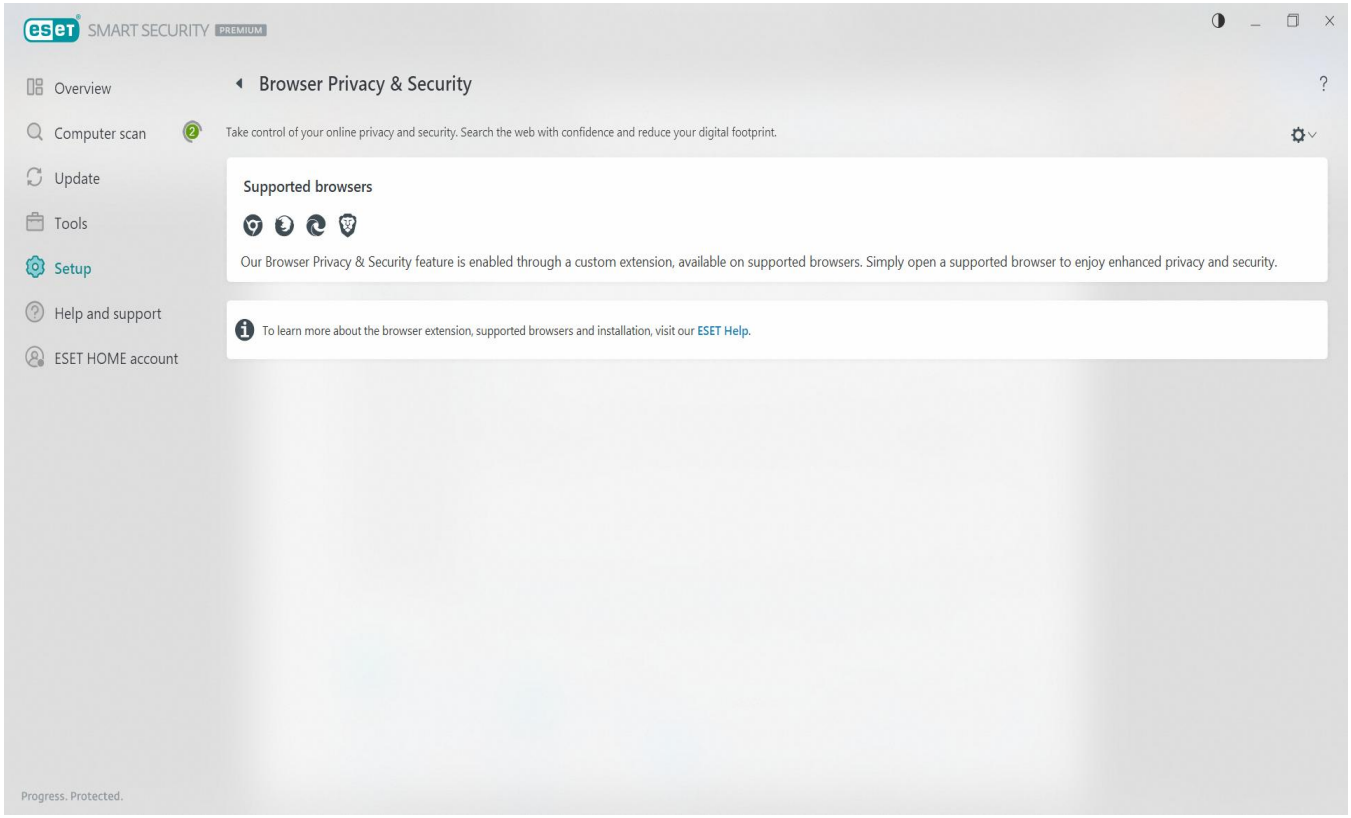


Figure 22: ESET browser security feature in action

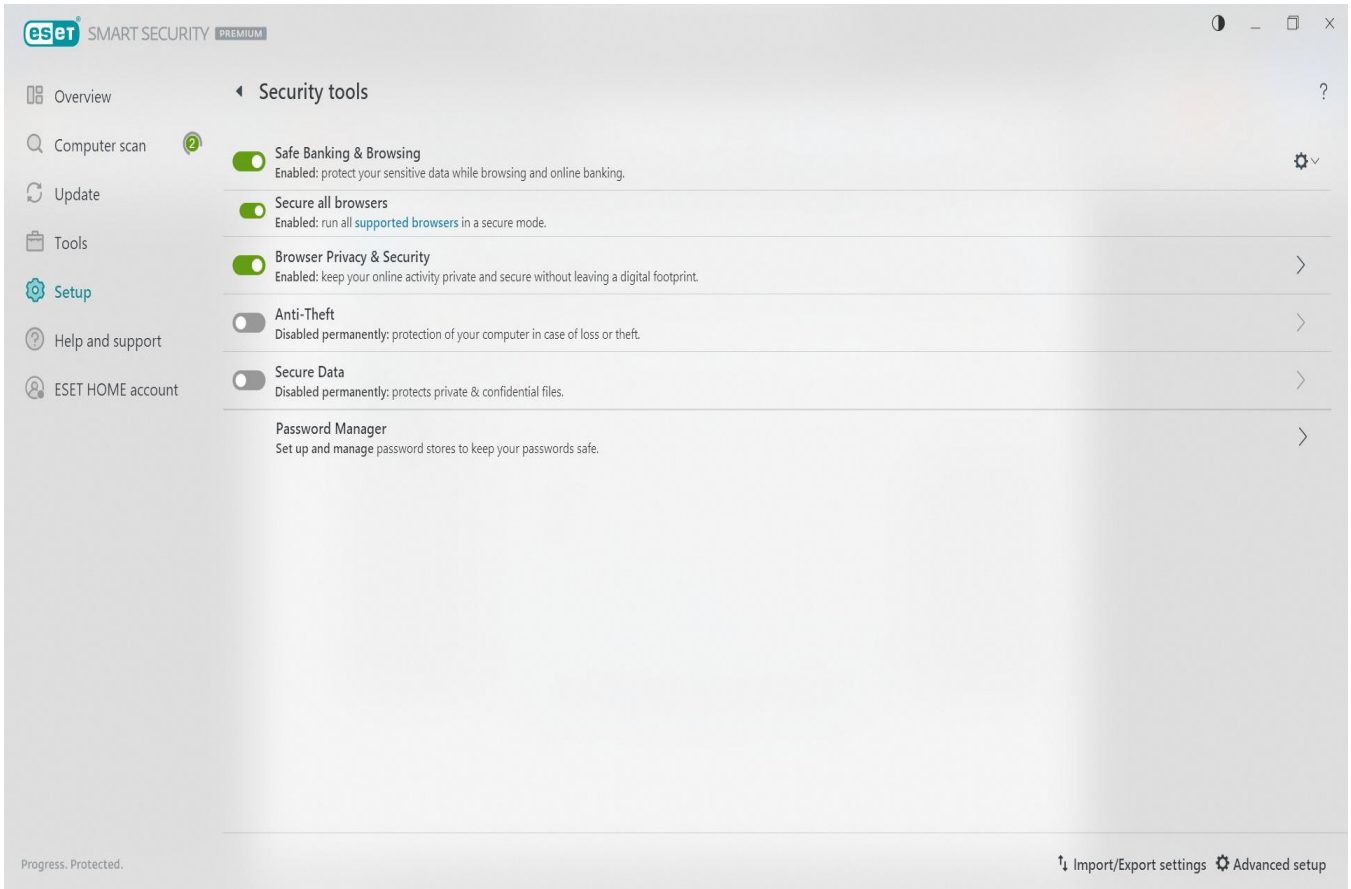


Figure 23: Overview of ESET's security tools

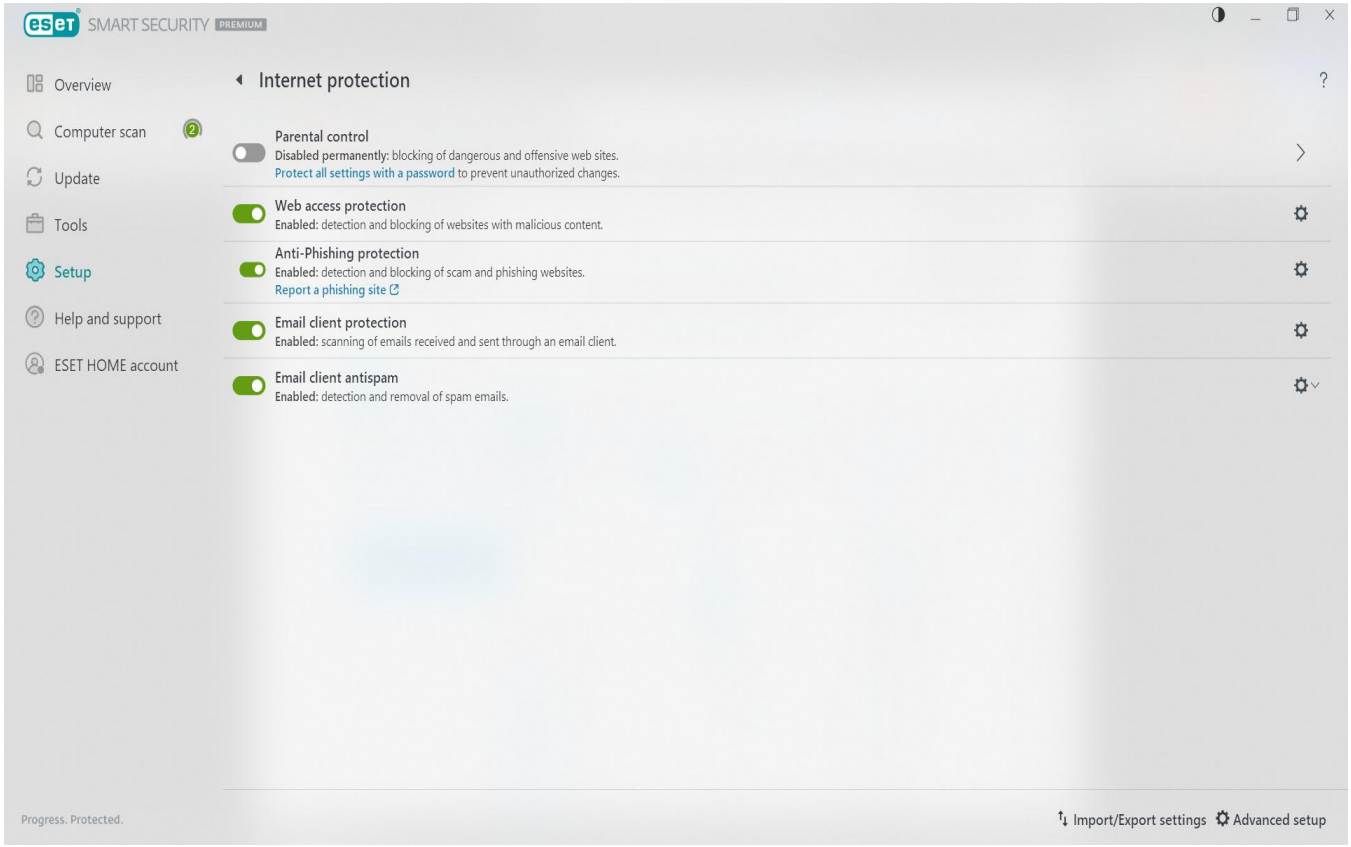


Figure 24: Internet protection settings in ESET

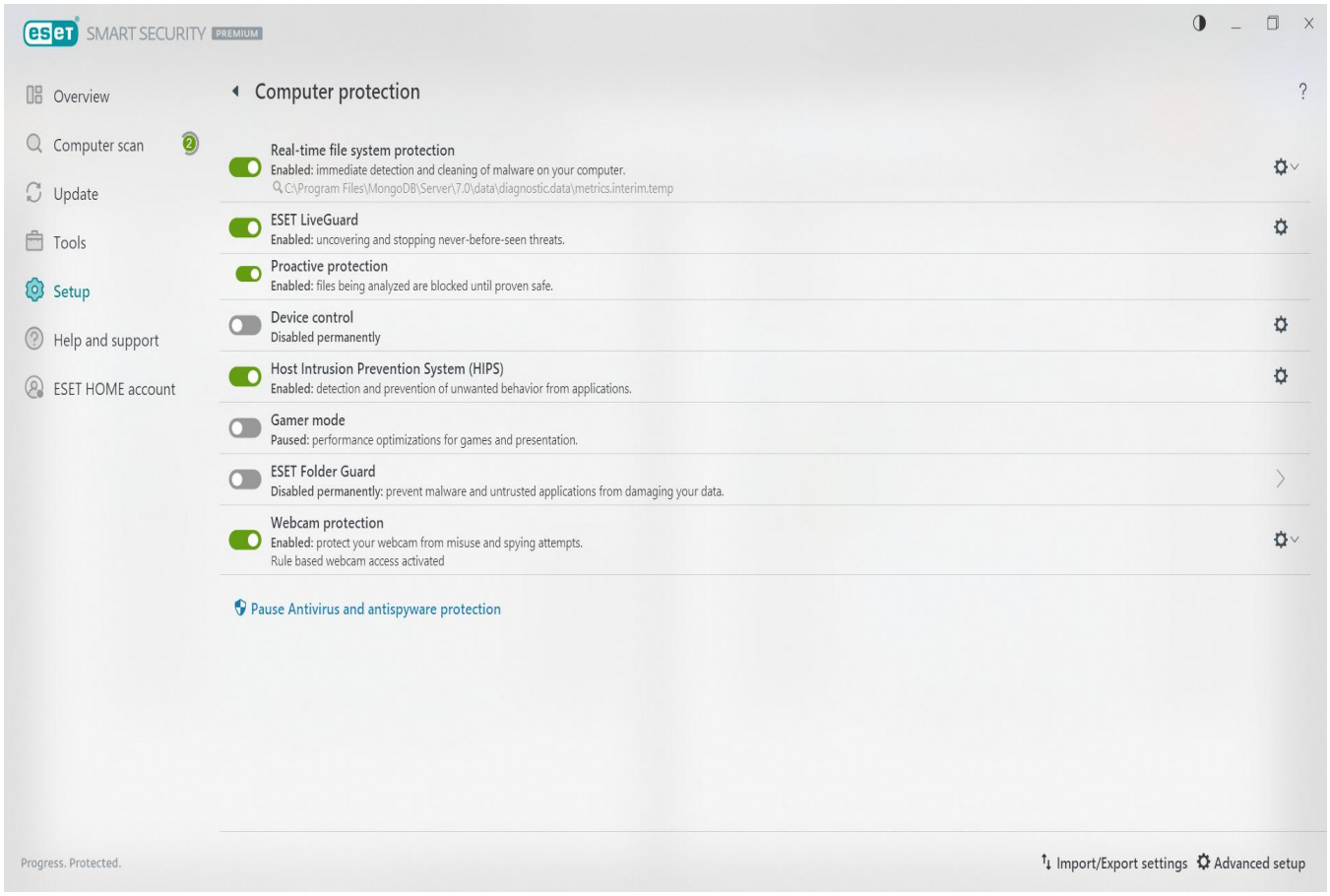


Figure 25: Computer protection settings in ESET

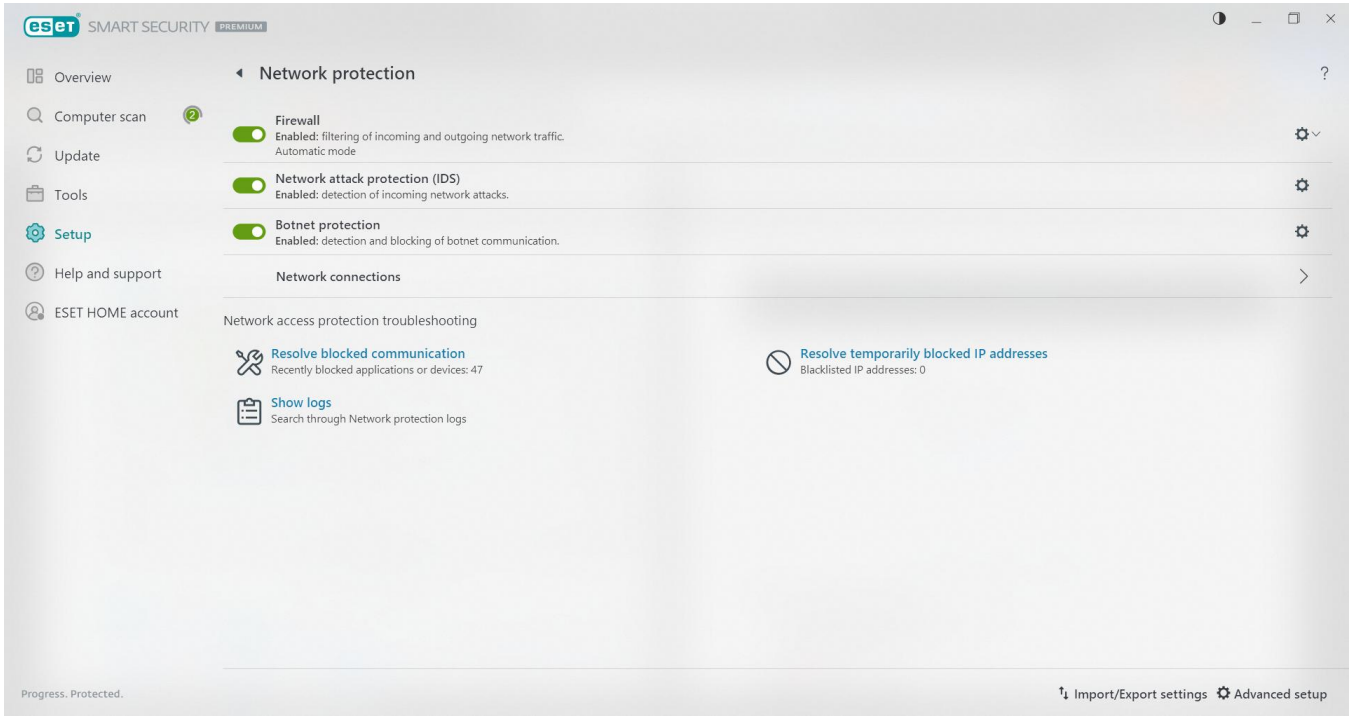


Figure 26: Network protection settings in ESET

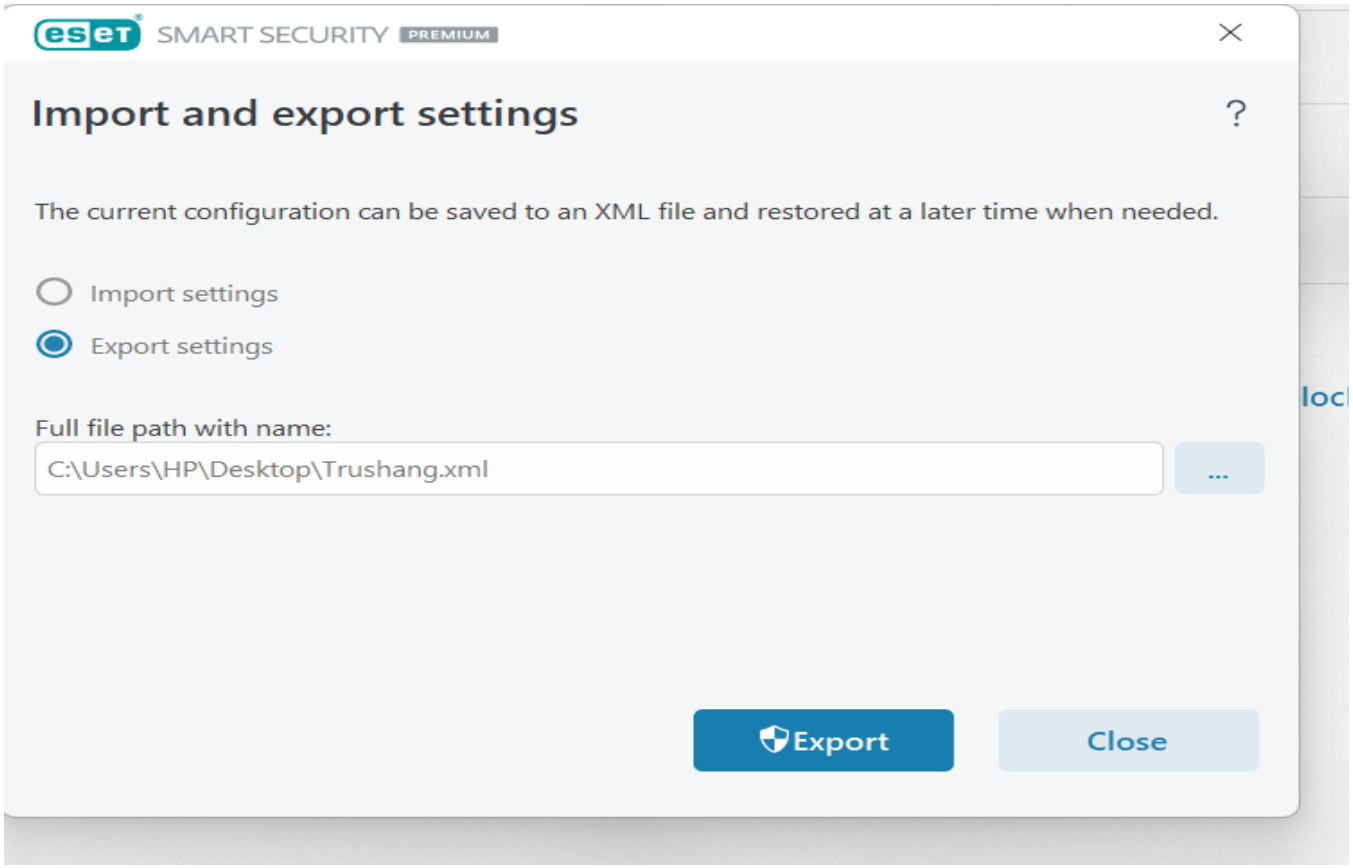


Figure 27: Import/export feature for ESET configuration settings

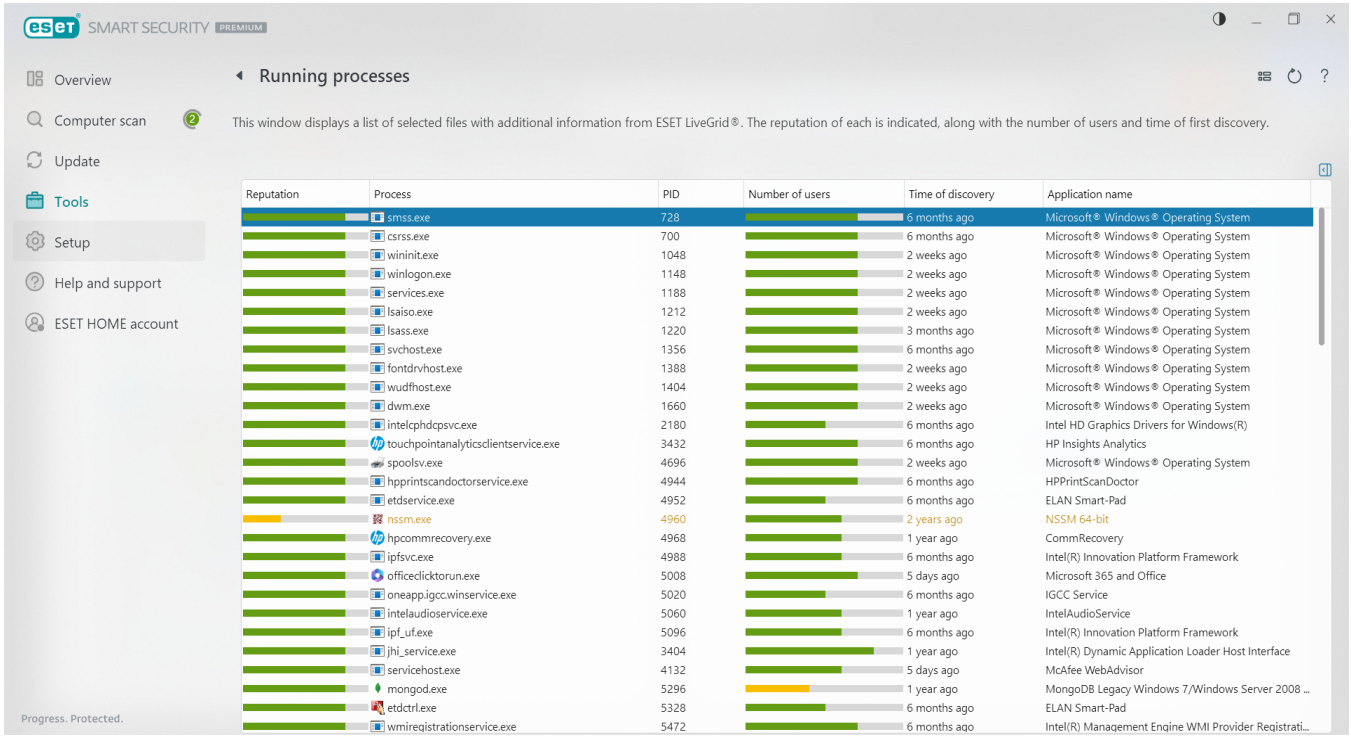


Figure 28: View of running processes monitored by ESET

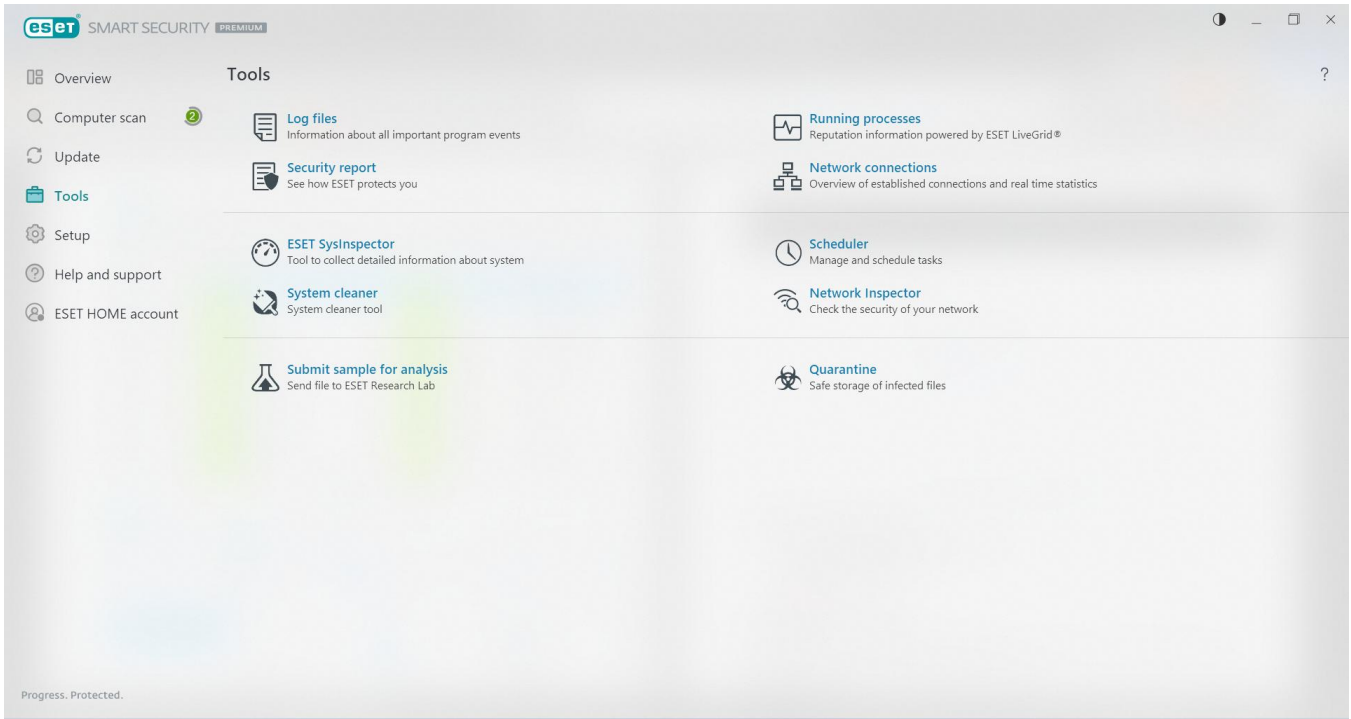


Figure 29: Additional ESET security tools

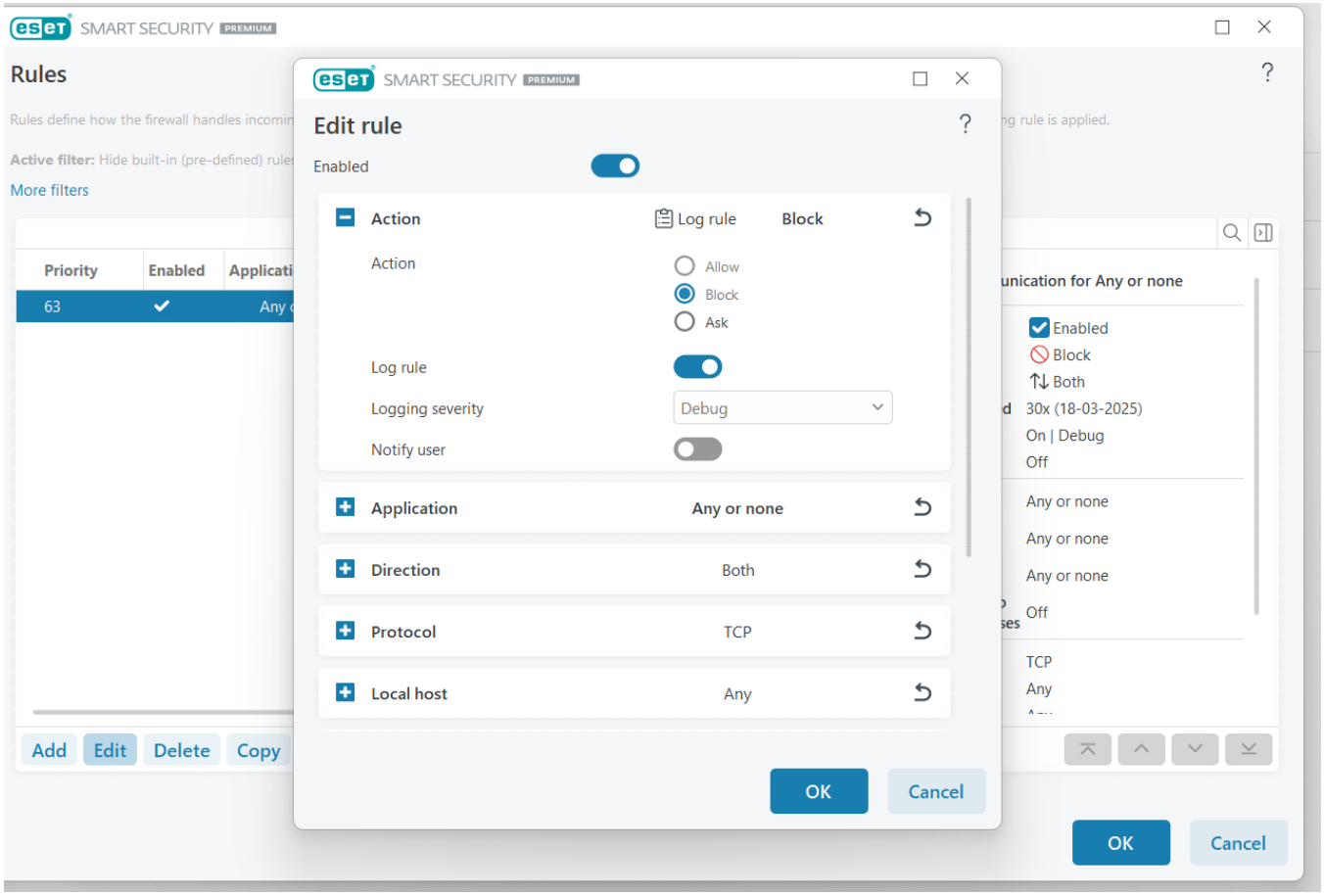


Figure 30: Creating a custom firewall rule in ESET to block internet access

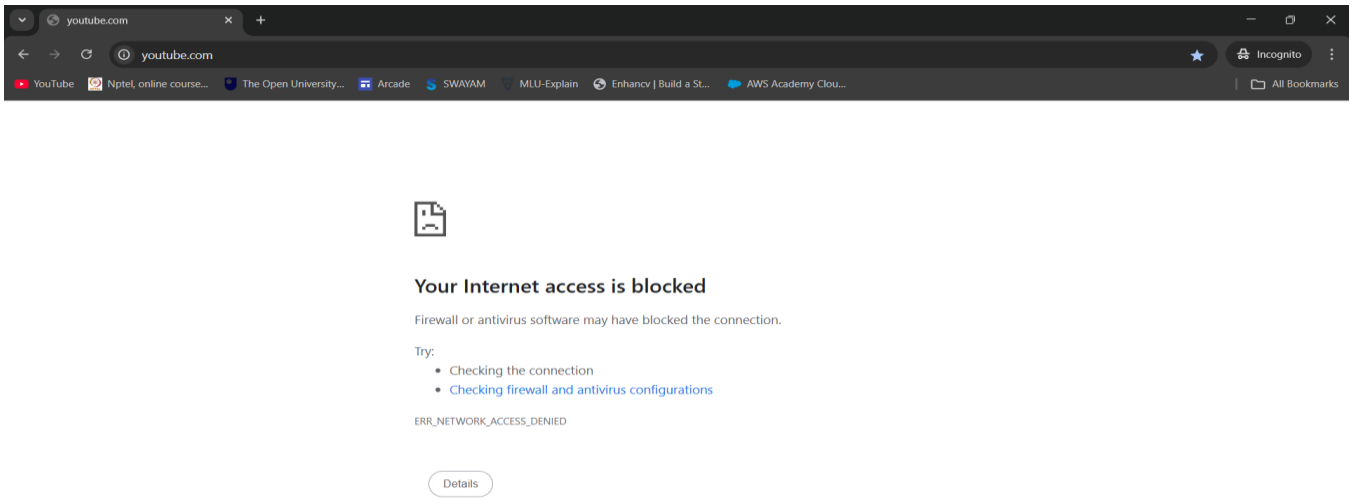


Figure 31: Demonstration of internet access blocked by ESET Firewall

LATEST APPLICATIONS:

- FortiGate Next-Generation Firewall (NGFW)
- Check Point Quantum Firewall
- Cisco Secure Firewall
- pfSense
- ESET

LEARNING OUTCOME:

In this practical, I gained a comprehensive understanding of configuring, managing, and evaluating both Windows Defender Firewall and ESET Personal Firewall. I learned how to create, customize, and apply firewall rules to control network traffic, ensure secure communication, and block unauthorized access. Additionally, I explored advanced security features like intrusion detection, secure browsing, and network monitoring, enhancing my ability to assess and implement firewall solutions in an enterprise environment.

REFERENCES:

1. YouTube : https://www.youtube.com/watch?v=pP7_nFBNR-M
2. ChatGPT: https://chatgpt.com/
3. ESET forum : https://forum.eset.com/topic/25222-eset-firewall-vs-windows-10-firewall/

PRACTICAL: 11

AIM:

Wireshark is an open-source tool for profiling network traffic and analysing packets. It is often referred to as a network analyzer, network protocol analyzer, or sniffer. Wireshark intercepts traffic and converts that binary traffic into a human-readable format. Network administrators, Network security engineers, QA engineers, Developers, and other people who troubleshoot network problems, examine security problems, verify network applications, debug protocol implementations, and learn network protocol internals, respectively, can use it. A practical approach to study Wireshark from network security concept.

THEORY:

Wireshark is a powerful, open-source network protocol analyzer that allows users to capture, analyze, and troubleshoot network traffic. It is a widely used tool in the field of cybersecurity, providing valuable insights into network activity and potential security threats.

Network security is a critical aspect of modern computing, as the increasing reliance on interconnected devices and networks has made organizations more vulnerable to various security threats. Understanding the fundamentals of network security is essential for effectively detecting and mitigating these threats.

Common Network Security Threats

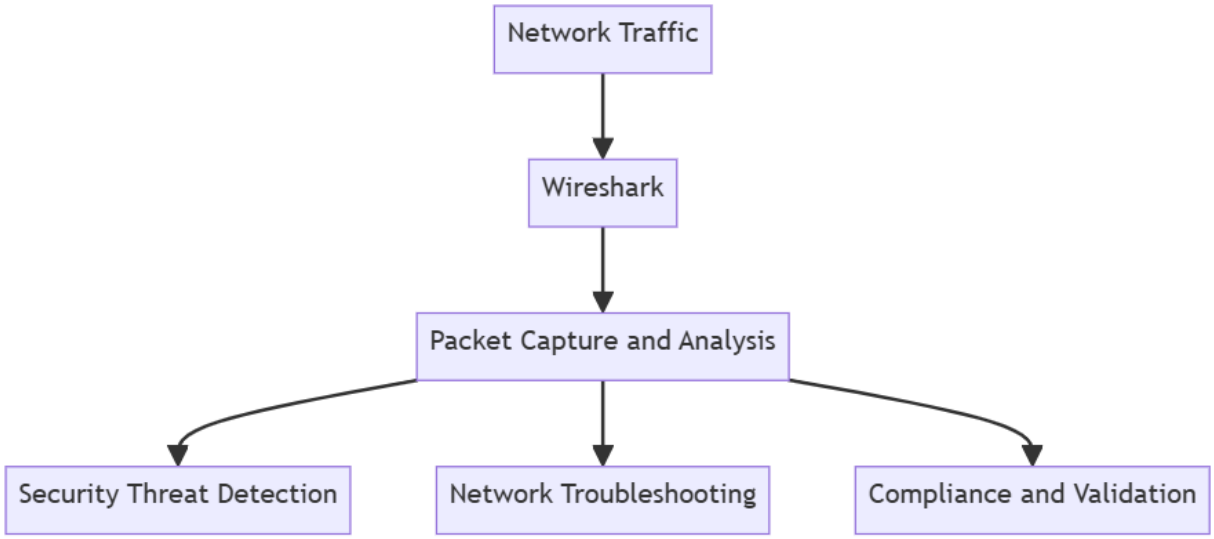
- Unauthorized access: Attackers attempting to gain unauthorized access to network resources or systems.
- Eavesdropping: Intercepting and monitoring network traffic to gather sensitive information.
- Denial-of-Service (DoS) attacks: Attempts to disrupt or overwhelm a network or system, rendering it unavailable to legitimate users.
- Malware propagation: The spread of malicious software, such as viruses, worms, or trojans, through the network.

Effective network monitoring and analysis are crucial for identifying and addressing security threats. By analyzing network traffic, security professionals can detect anomalies, identify potential vulnerabilities, and take appropriate actions to mitigate risks.

Wireshark's Role in Cybersecurity

Wireshark is a valuable tool for cybersecurity professionals, as it provides a comprehensive view of network activity, allowing them to:

- Identify and analyze network security incidents
- Detect and investigate suspicious network traffic
- Troubleshoot network issues and performance problems
- Validate the effectiveness of security controls and policies



By understanding the fundamentals of network security and mastering the use of Wireshark, cybersecurity professionals can effectively detect and respond to network security threats, ensuring the overall security and integrity of their organization's network infrastructure.

CODE:

```
N/A
```

OUTPUT:

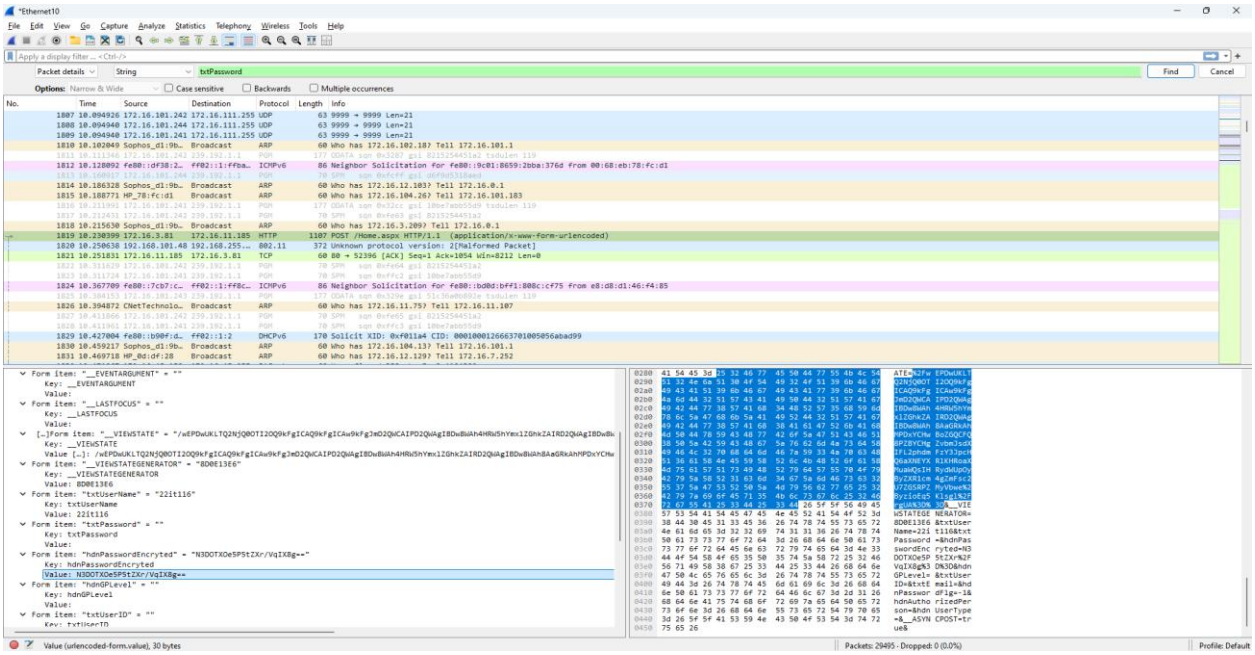


Figure 163: Getting Hash Value of E-Governance login password

```
encrypted_message = "N3DOTXOe5P5tZXr/VqIX8g=="  # Replace with your actual encrypted message
decrypted_message = decrypt_message(encrypted_message, secret_key)
print("Decrypted message:", decrypted_message)
```

```
Collecting pycryptodome
  Downloading pycryptodome-3.22.0-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.4 kB)
Downloading pycryptodome-3.22.0-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.3 MB)
   ---------------------------------------- 2.3/2.3 MB 14.5 MB/s eta 0:00:00
Installing collected packages: pycryptodome
Successfully installed pycryptodome-3.22.0
Decrypted message: Egovernance@28
```

Figure 164: Decrypt the Hash Password

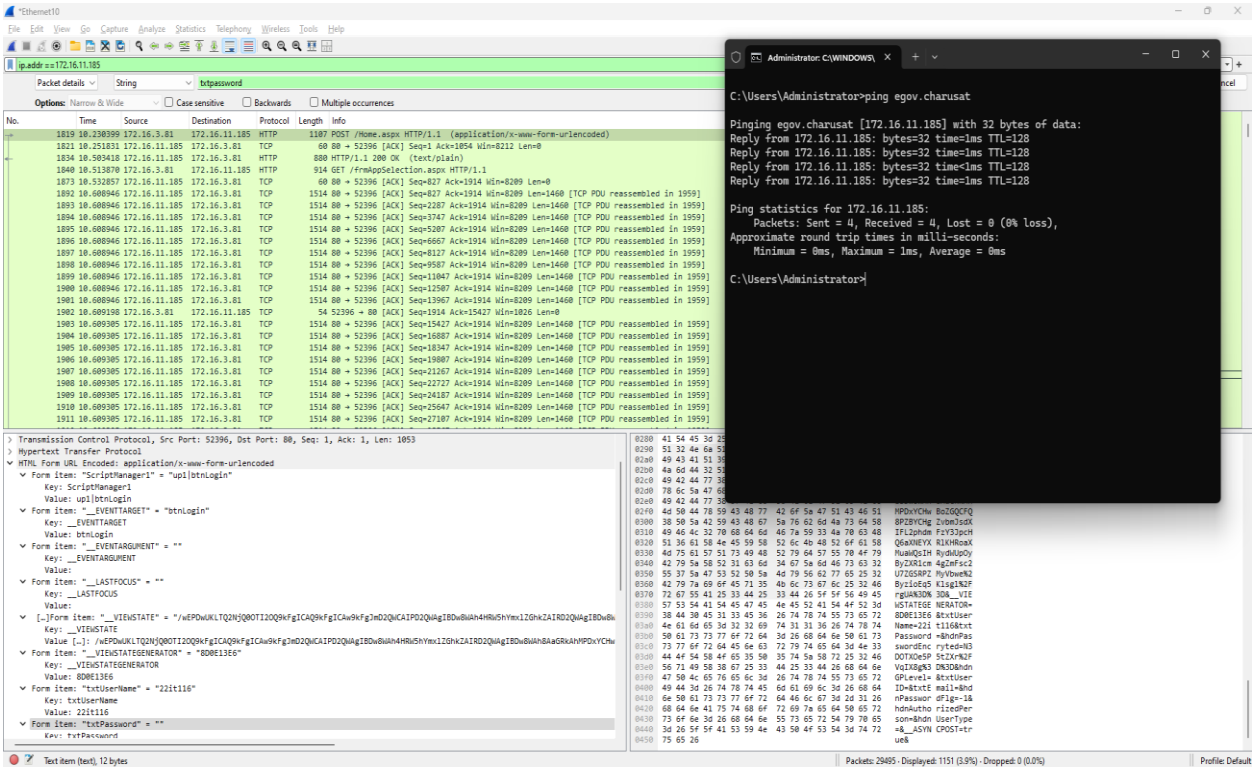


Figure 165: Flitter packet by IP address

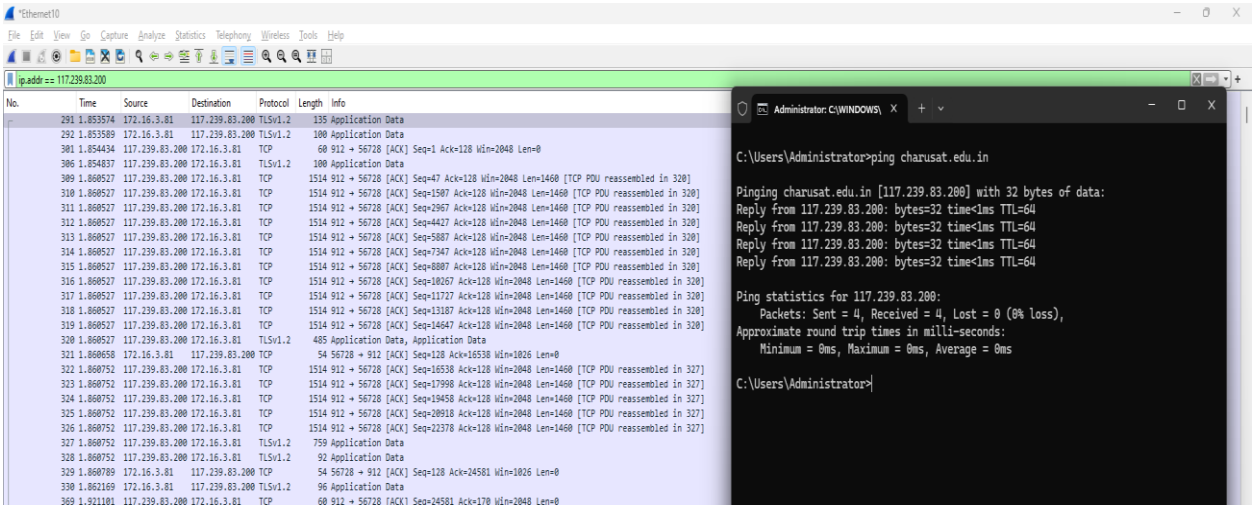


Figure 166: Check the E governance Host website IP and Apply filter for IP address

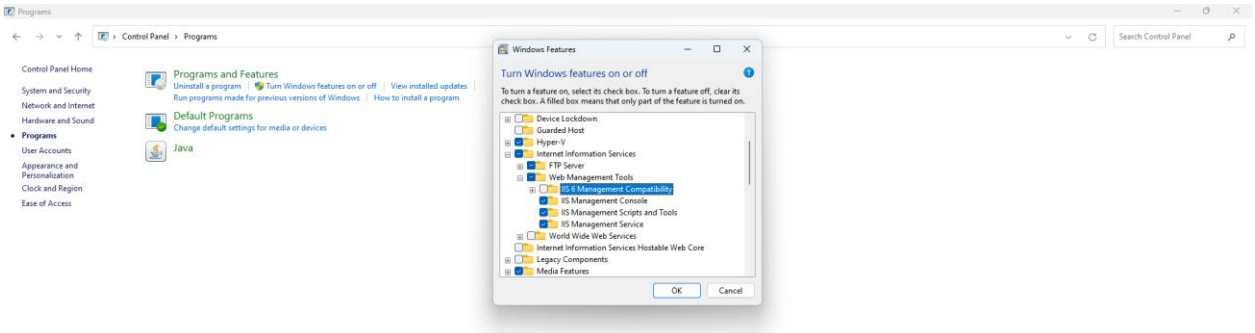


Figure 167: Open Windows Features and Check FTP service (Control Panel -> Programs -> Turn Windows features on or off -> FTP Services

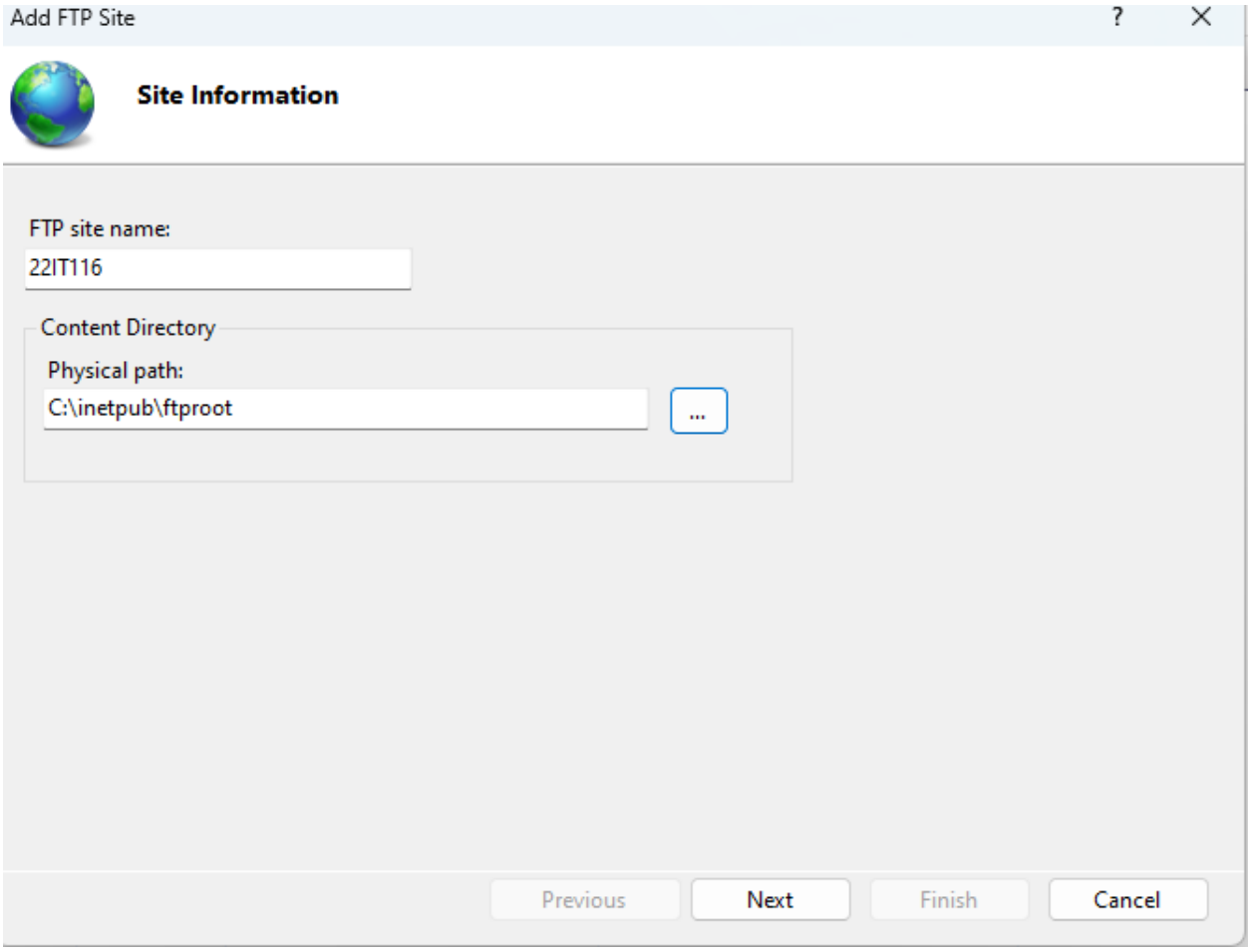


Figure 168: Enter a name for FTP site and specify the path to the folder that you want to use for FTP

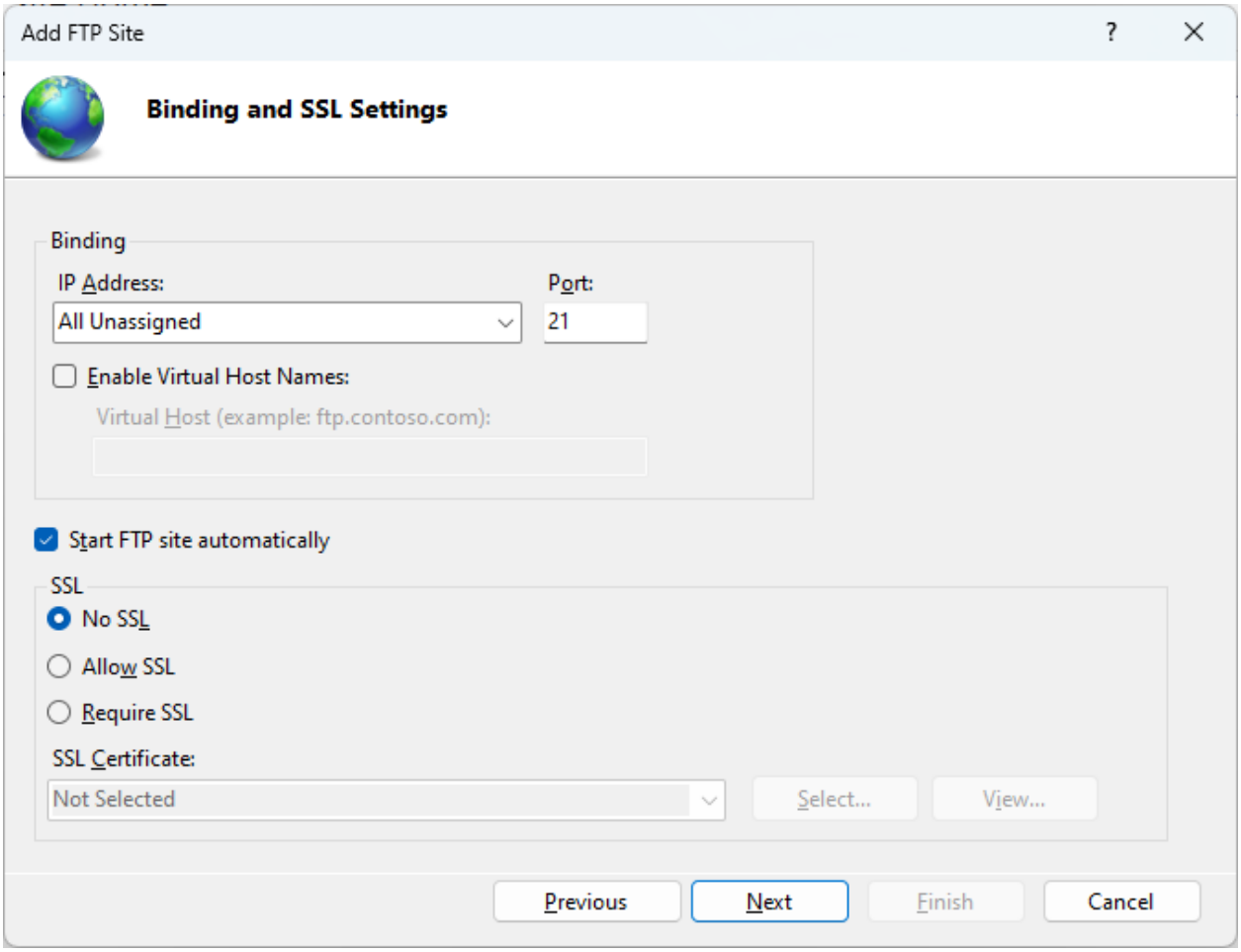


Figure 169: Configure Binding and SSL

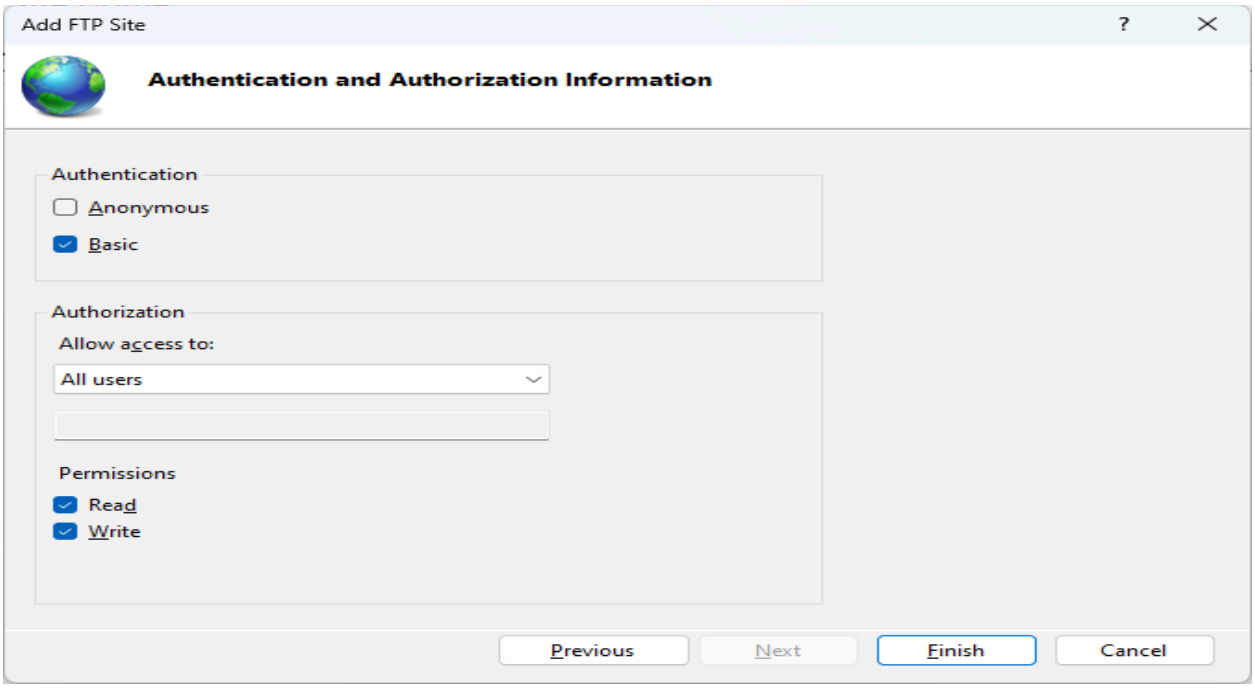


Figure 170: Click Finish to create the FTP site and start it

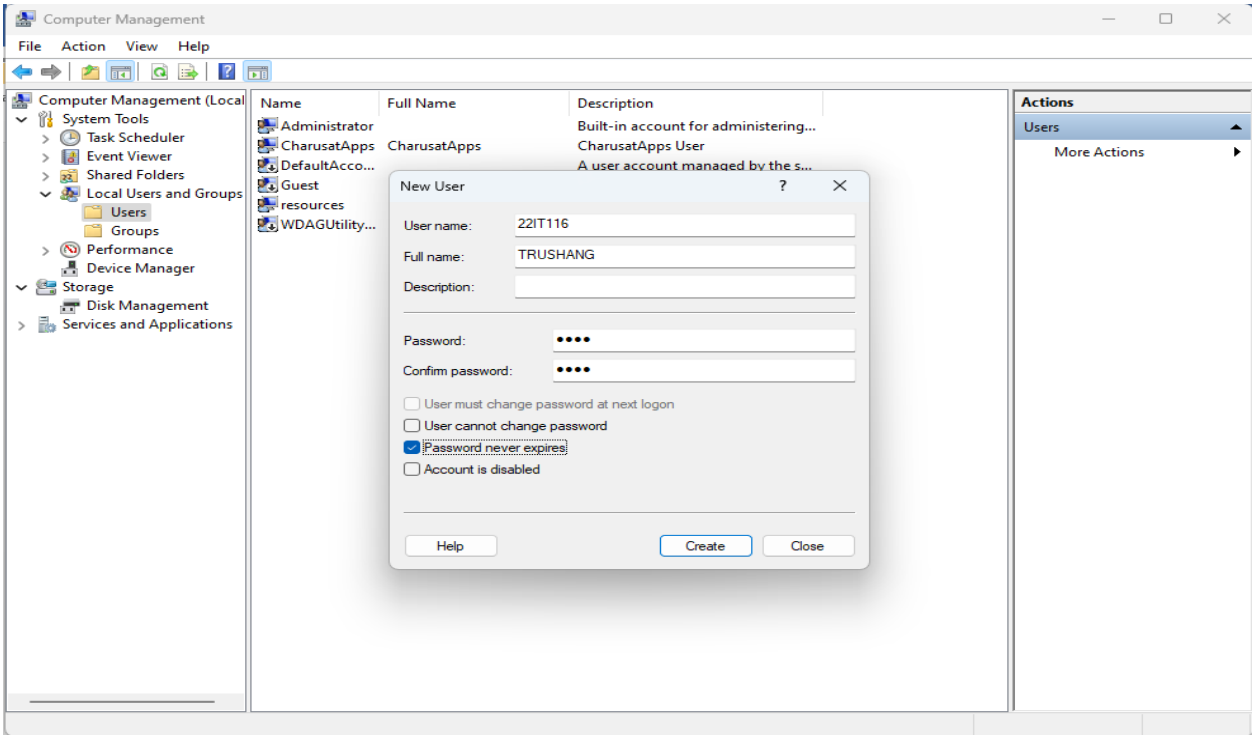


Figure 171: Create a user in computer management (Computer management ->Local Users & Groups -> User-> New User)

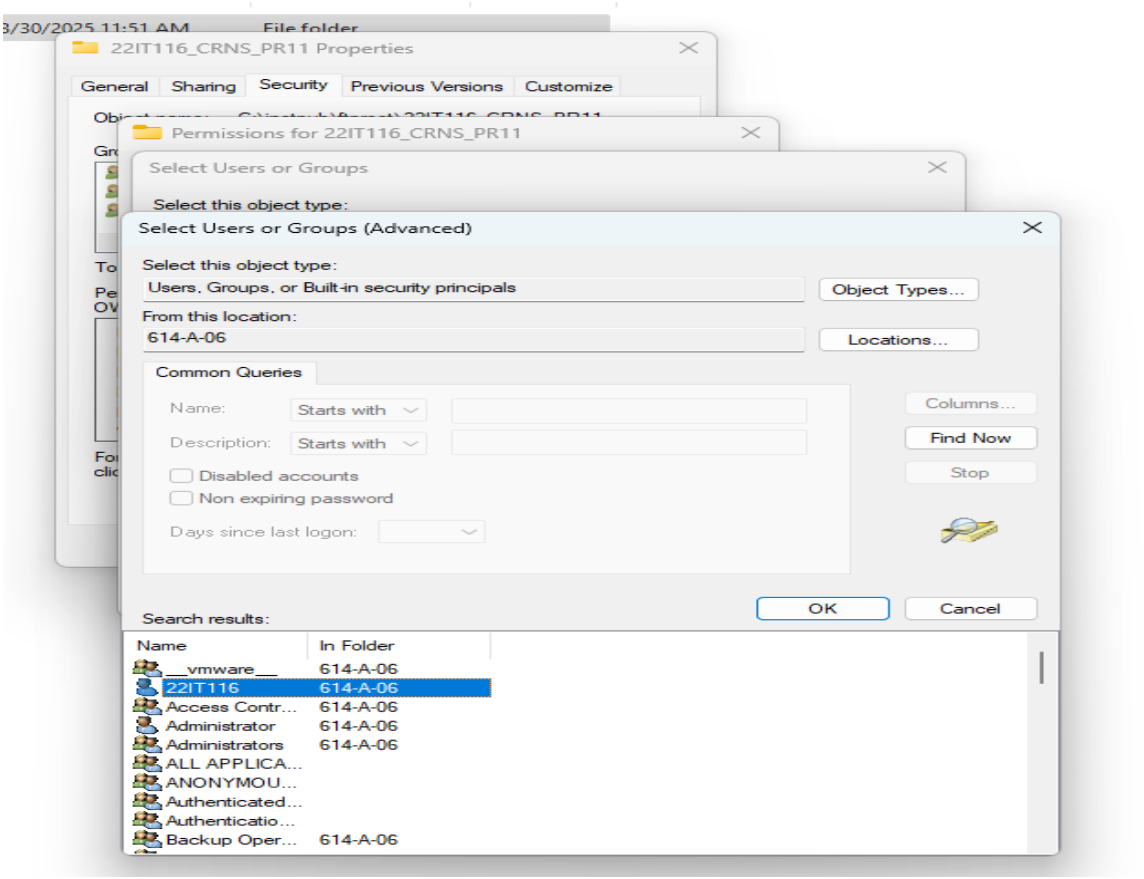


Figure 172: Create a folder and Give permission to user which created in previous step

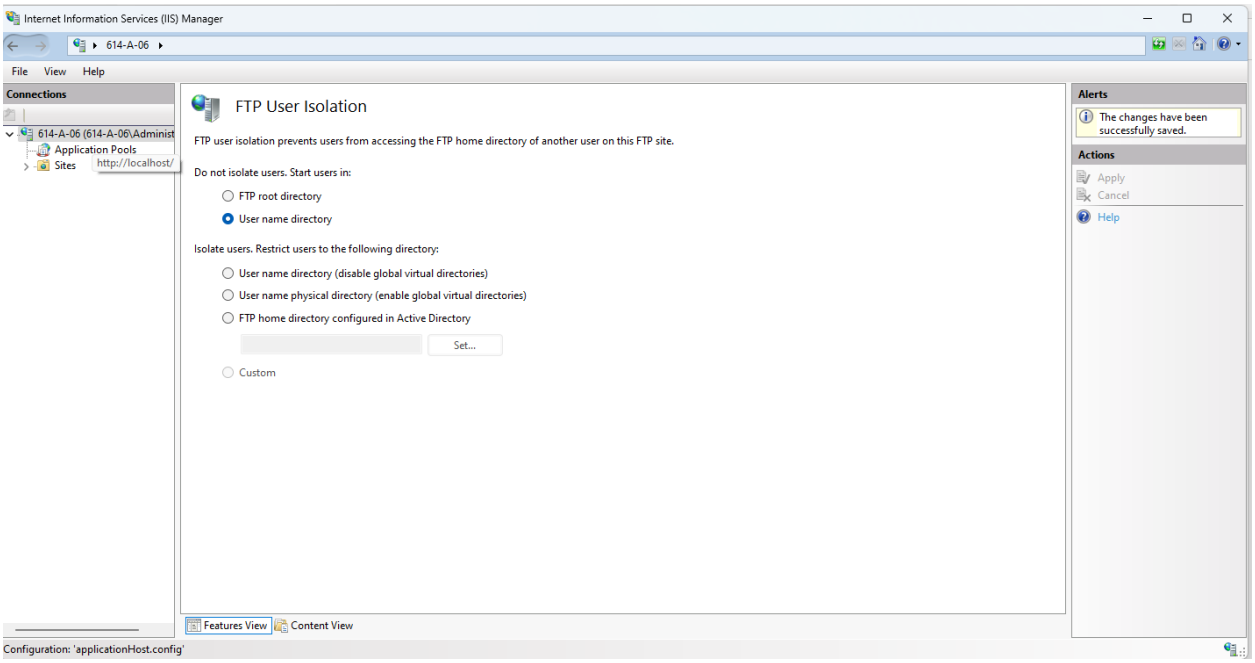


Figure 173:FTP isolation

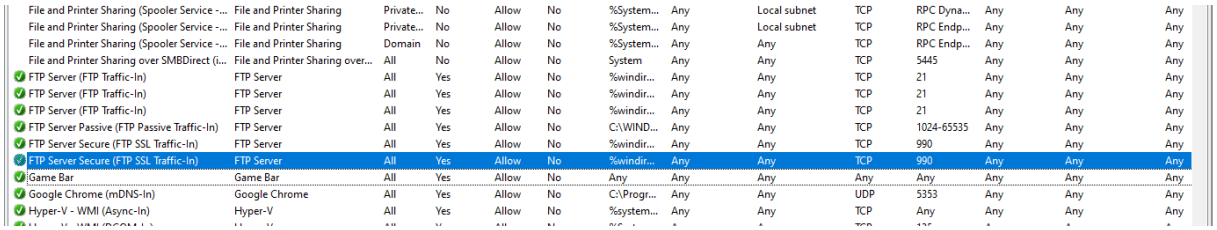


Figure 174: Enable FTP server rule

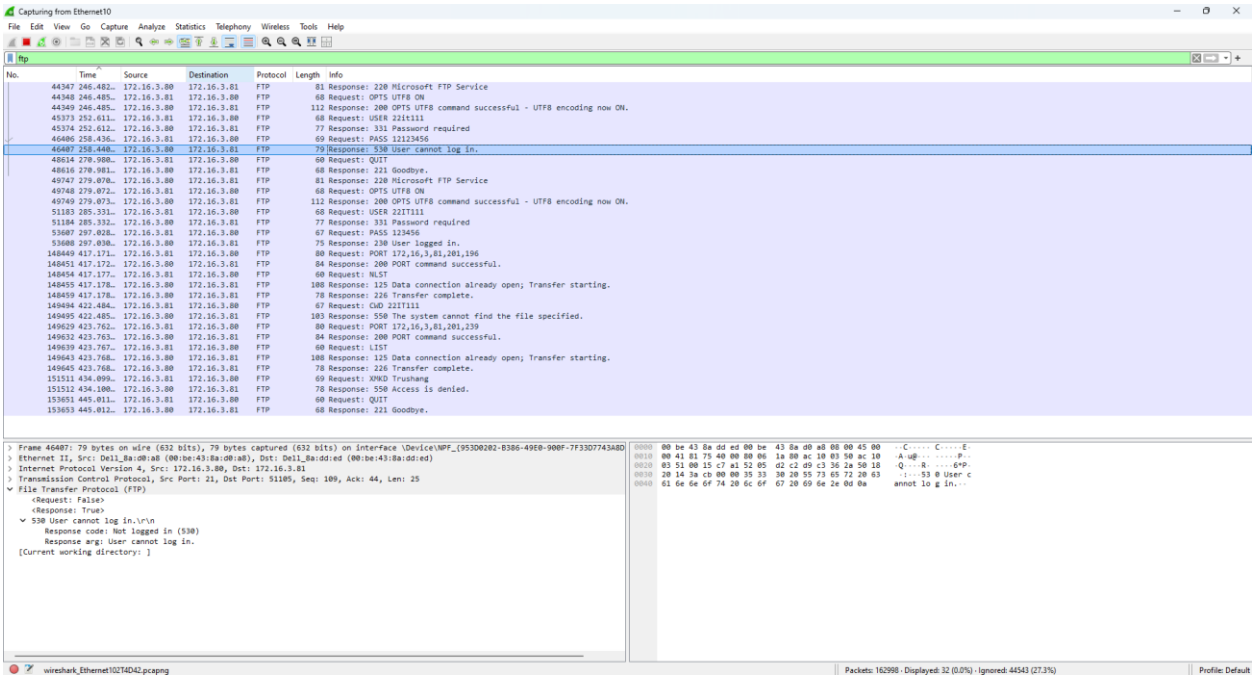


Figure 175:Get Password of FTP using Wireshark

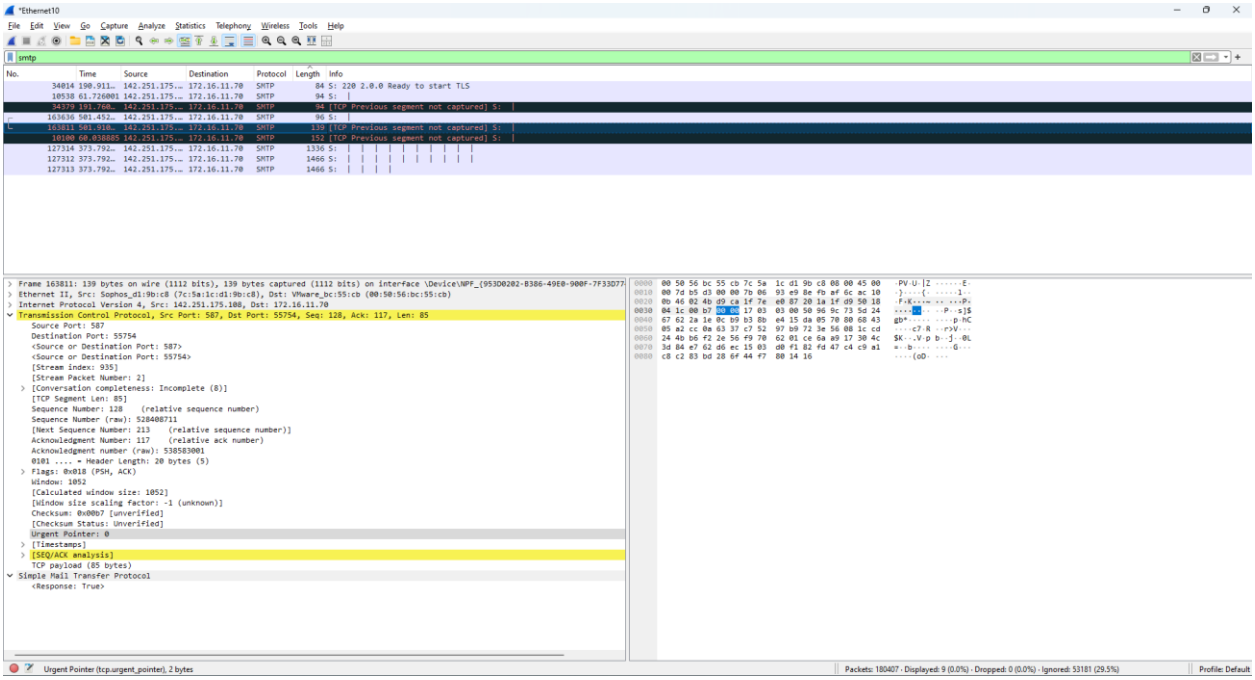


Figure 176:Check SMTP packet in Wireshark

LATEST APPLICATIONS:

- Advanced Network Traffic Analysis:
- Intrusion Detection and Prevention:
- Malware Analysis:
- Network Forensics:
- Vulnerability Analysis
- Security Monitoring and Incident Response

LEARNING OUTCOME:

In this practical, I learned how to utilize Wireshark, a network protocol analyzer, to capture and analyze network traffic. By observing packets transmitted over the network, I gained insights into the behavior of various protocols such as HTTP, TCP, and FTP. This hands-on experience enhanced my understanding of network operations and the importance of monitoring for security purposes.

REFERENCES:

4. YouTube : https://www.youtube.com/watch?v=yC0e0bSSleo
5. WireShark : https://www.wireshark.org/docs/wsug_html_chunked/
6. ChatGPT: https://chatgpt.com/