HO#2.1: Ethical Hacking, Penetration Testing & Anonymity

The term **Ethical hacking** is the *practice of intentionally probing computer systems, networks, & applications to find and report security vulnerabilities* that could be exploited by malicious hackers. Ethical hackers use their skills to improve the security of these systems by identifying weaknesses and recommending corrective measures. The primary goal of ethical hacking is to ensure the security and integrity of information systems.

A **hacker** is someone skilled in computer systems and programming who uses his expertise to explore, understand, and manipulate technology. Hackers can be categorized based on their intentions and the legality of their actions:

- **White-Hat Hackers** (Ethical Hackers):
  - Intentions: Positive, with the aim of improving security.
  - Activities: Conduct authorized penetration testing, vulnerability assessments, and security audits.
  - Example: Security professionals who work for companies to secure their networks.
- **Black-Hat Hackers** (Malicious Hackers):
  - Intentions: Malicious, often motivated by personal gain, causing harm, or disrupting systems.
  - Activities: Unauthorized access to systems, data theft, malware distribution, and other illegal activities.
  - Example: Cybercriminals who steal credit card information or deploy ransomware.
- **Gray-Hat Hackers**:
  - Intentions: Ambiguous; they may break into systems without permission but without malicious intent.
  - Activities: Often exploit vulnerabilities and then inform the system owner, sometimes expecting a reward.
  - Example: Hackers who find and disclose vulnerabilities in software without explicit permission but without harmful intent.

Vulnerability, Exploit, and Payload

- *A **vulnerability** is a weakness or flaw that can exist in a hardware's firmware, in the OS of computer or router, NW services, libraries and application software that can be exploited by a threat actor to gain unauthorized access, cause damage, or perform unauthorized actions.* A common example of a real-life vulnerability is a house with a weak lock on main door.
- *An **exploit** is a program/software that take advantage of a vulnerability leading to gain initial access or privilege escalation on the target.* A common example of a real-life exploit is the duplicate key using which a robber can enter the house.
- *A **payload** is the malicious code which runs on the compromised system after initial access, enabling actions like remote control, data exfiltration, or persistence.* It can be active (a reverse shell that maintains a live connection) or passive (a keylogger that silently records keystrokes). A common example of a real-life payload is the robber stealing jewellery or cash from the compromised house.

Example: Consider a refrigerator with an electronic lock contains chocolates, fruit trifle, cold drinks etc. Somehow you come to know about its *vulnerability* that it can be unlocked using a PAK123 key, written on the door. You exploit that vulnerability and open/unlock the refrigerator. So, an *exploit* is a code that takes advantage of a *vulnerability*, while a *payload* is the code executed on the target machine once the exploit is successful.

Common OS Vulnerabilities https://www.cve.org/

These vulnerabilities can vary in severity depending on the operating system and its configuration. Addressing them typically involves regular patching, configuring secure settings, and using robust security controls.

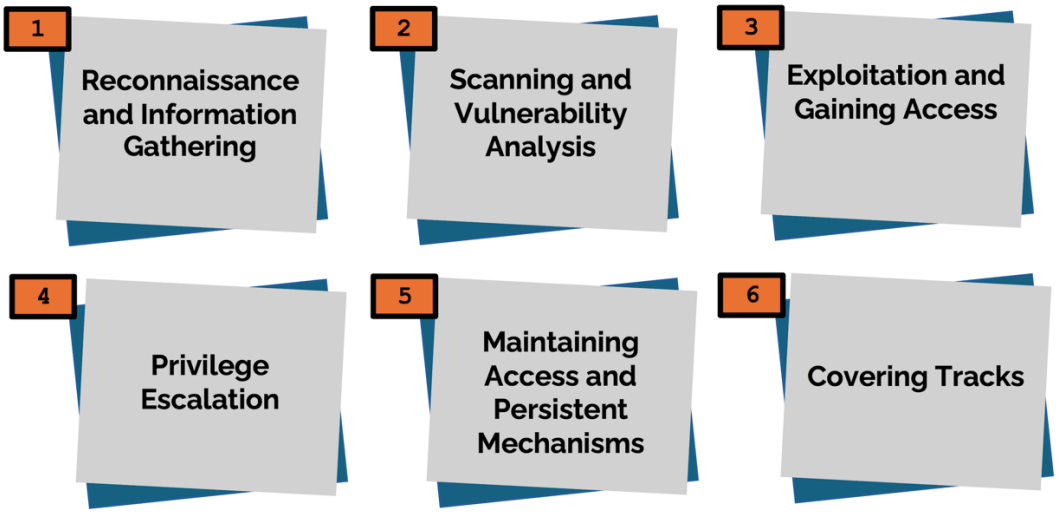
- **Kernel Vulnerabilities:** Flaws in the OS kernel, the core part of the OS, that attackers can exploit to gain root access. May result in a full system compromise.
- **Buffer Overflow:** Occurs when a program writes more data to a buffer than it can hold, leading to memory corruption. It can be exploited to execute arbitrary code or crash the system.
- **Insecure APIs:** Vulnerabilities in application programming interfaces (APIs) that allow attackers to interact with the OS in unintended ways. Data breaches or system manipulation.
- **Race Conditions:** Timing vulnerabilities where two processes try to change the same resource at the same time. Can lead to data corruption or unauthorized access.
- **Unpatched Software:** Failure to apply security patches to the OS or applications. It allows attackers to exploit known vulnerabilities.
- **Weak Access Controls:** Poorly configured permissions, allowing unauthorized users access to sensitive system resources. Unauthorized access, data leakage, or system compromise.
- **Malware Infection:** Malware can exploit OS vulnerabilities to install backdoors, keyloggers, ransomware, or spyware. It can result in system compromise, data theft, or denial of service.
- **Directory Traversal:** An attacker gains access to restricted directories and files outside a web server's root directory by manipulating file paths. Access to sensitive files like password databases.
- **Default Configurations:** OS installations with default settings can expose unnecessary services or weak security configurations. Increases the attack surface.
- **Outdated Cryptography:** Using outdated or weak cryptographic algorithms for data protection. Enables attackers to decrypt sensitive information.
- **Command Injection:** An attacker can execute arbitrary system commands by exploiting input validation flaws in the OS. System compromise and data theft.
- **Weak Session Management:** Poorly managed user sessions that can be hijacked or improperly terminated. Attackers may hijack user sessions and perform unauthorized actions.

Common Web Application Vulnerabilities https://owasp.org/Top10/

- **Injection Vulnerabilities:** These occur when untrusted data is sent to an interpreter as part of a query or command, allowing an attacker to manipulate the interpreter's actions.
  - SQL Injection (SQLi): Exploiting unvalidated user input to manipulate SQL queries.
  - Cross-Site Scripting (XSS): Injecting malicious scripts into web pages viewed by other users.
  - Command Injection: Injecting commands to execute on the server via input fields.
  - LDAP Injection: Manipulating Lightweight Directory Access Protocol queries through user input to access unauthorized information.
- **Authentication & Session Management Vulnerabilities:** These vulnerabilities allow attackers to impersonate legitimate users or hijack user sessions.
  - Broken Authentication: Weaknesses in login mechanisms that allow attackers to bypass authentication.
  - Session Hijacking: Stealing or predicting session tokens to impersonate a user.
  - Cross-Site Request Forgery (CSRF/XSRF): Trick a user into submitting a request they did not intend, potentially altering user data.
  - Insecure Direct Object Reference (IDOR): Manipulating URL parameters to access data or resources without proper authorization.
- **Input Validation Vulnerabilities:** These vulnerabilities arise when user input is not properly validated, allowing attackers to submit unexpected or malicious data.
  - Cross-Site Scripting (XSS): Where user input is not sanitized, allowing malicious scripts to be executed in other users' browsers.
  - HTML Injection: Injecting raw HTML into a page, which can manipulate the website's appearance or behaviour.
  - File Upload Vulnerabilities: Uploading malicious files (e.g., scripts or executables) to the server.
  - Remote File Inclusion (RFI): Loading remote files into the web application, leading to arbitrary code execution.
- **Access Control Vulnerabilities:** These are weaknesses in enforcing proper access levels, allowing users to access resources they shouldn't.
  - Broken Access Control: Insufficient restriction on what authenticated users can access.
  - Privilege Escalation: Gaining higher privileges than intended due to improper access control configurations.
  - Directory Traversal: Accessing files outside the intended directory by manipulating file paths.
- **Security Misconfigurations:** These vulnerabilities occur when web servers, applications, or databases are misconfigured, exposing unnecessary or insecure functionality.
  - Unpatched Software: Not applying security updates to web application components.
  - Default Credentials: Using factory-default usernames and passwords in production environments.
  - Excessive Permissions: Overly permissive file or folder access permissions on web servers.
- **Denial of Service (DoS) Vulnerabilities:** These allow attackers to make a web application or website unavailable to legitimate users.
  - Application-Level DoS: Exploiting application flaws (e.g., resource-intensive processes) to exhaust server resources.
  - Distributed Denial of Service (DDoS): Overloading a server with requests from multiple sources to make it unavailable.

Life Cycle of Penetration Testing

Penetration testing (pen testing) is a simulated cyberattack on a computer system, network or applications to find and report security vulnerabilities that could be exploited. It falls into three categories, ***black box***, ***white box***, and ***gray box*** testing. The life cycle of a pen test typically follows a series of phases designed to mimic the steps an attacker might take to compromise a target. Here's an overview of the pen testing life cycle with the key phases:



1. Reconnaissance and Information Gathering

The Information gathering phase (reconnaissance) is the initial step in the penetration testing lifecycle. This phase involves collecting as much public information as possible about the organization, systems, networks, applications, and employees to identify potential vulnerabilities and formulate a strategy for further testing. Passive information gathering (reconnaissance) involves collecting data without directly interacting with the target system, reducing the risk of detection. Gathering information from publicly available sources like news outlets, blogs and social media platforms (Twitter, Facebook, LinkedIn) is named as Open-Source Intelligence (OSINT). The techniques used for OSINT are Web Scraping, Google Dorking, and social media profiling. There are two types of Reconnaissance:

- Passive Reconnaissance: Gathering information without direct interaction with the target, such as using public sources (websites, social media, WHOIS databases).
- Active Reconnaissance: Directly interacting with the target to gather information, such as pinging the network, or using tools like `nmap` for network mapping.

Tools:

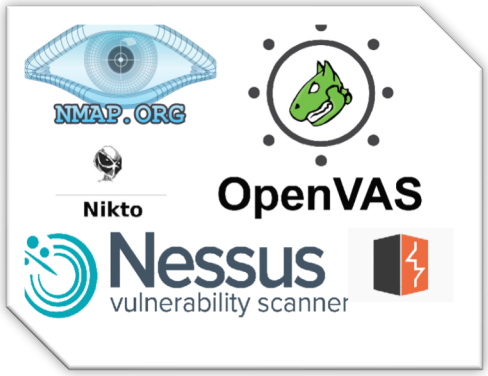
- Netdiscover
- TraceRoute
- Host, nslookup, dig
- Whois
- Whatweb
- TheHarvester
- Knockpy
- Sherlock
- Wafw00f
- Google Dorking/Hacking
- OSINT Framework

2. Scanning & Vulnerability Analysis

Scanning and vulnerability analysis is the second phase of penetration testing whose objective is to discover open ports, services, OS, library versions and other information about the target machine/NW. This information is then used to identify potential vulnerabilities, weaknesses, and misconfigurations that can be exploited to gain unauthorized access to the target machine/NW. You can say that in this phase we perform *Active Reconnaissance*, because the tools used in this phase directly interact with the target network, hosts, ports, employees, and so on to collect data. So DONOT perform active network scanning unless you have written permission of the system owner to perform that testing. Attacker normally perform following activities in this phase:

- Port Scanning: Identifying open ports and the services running on them.
- Network Scanning: Mapping the network to identify active devices and their IP addresses.
- Vulnerability Scanning: Identifying known vulnerabilities in the target's services and applications.

Tools:



- Nmap (https://nmap.org/download)
- Searchsploit (https://www.exploit-db.com/searchsploit)
- Nessus (https://www.tenable.com/products/nessus)
- OpenVAS (https://github.com/greenbone/openvas-scanner)
- MSF (https://www.metasploit.com/download)
- Nikto (https://github.com/sullo/nikto)

3. Exploitation and Gaining Initial Access

In this phase, the pentester take the advantage of the identified weaknesses like vulnerable applications and default configurations/credentials running on the target machine to gain unauthorized entry into the target system. Other than exploiting the known vulnerabilities, and stolen credentials, the pentester may use brute force, social engineering and phishing attacks to gain the initial entry to the target system. It involves the methods and techniques used by a pentester to gain entry into a target network or system. Attacker normally perform following activities in this phase:

- Exploiting Vulnerabilities: Using the information gathered to exploit vulnerabilities in services, applications, or network configurations.
- Brute Force Attacks: Attempting to gain access by guessing passwords or using automated tools to try multiple combinations.

Tools:

- MSF (https://www.metasploit.com/download)
- Exploit DB (https://www.exploit-db.com/)
- SQLmap (https://sqlmap.org/)
- Cobalt Strike (https://www.cobaltstrike.com/)
- Social Engineering Toolkit (https://github.com/trustedsec/social-engineer-toolkit)
- BeEF (Browser Exploitation Framework) (https://beefproject.com/)
- PowerSploit (https://github.com/PowerShellMafia/PowerSploit)

4. Privilege Escalation

After gaining initial access to the target machine, you may find that your session has only limited user rights. This severely limits the actions that one can perform on the remote systems such as dumping passwords, manipulating registry, and installing backdoors or keyloggers. So, Privilege Escalation is a critical phase in penetration testing where the tester attempts to gain elevated access rights beyond initial compromises. Attacker normally perform following activities in this phase:

- Setuid/Setgid Exploits in Linux: Exploiting binaries with the setuid or setgid bit set, which allows them to run with elevated privileges.
- Gaining Root Through Sudo Privileges: If the attacker finds that the compromised user has sudo access, they can use it to escalate privileges.
- Password Hash Dumping: Once an attacker gains access, they can dump password hashes from the system files like `/etc/passwd` or `/etc/shadow` on Linux systems. Later they can attempt to crack them offline and may succeed in cracking passwords of higher-privileged accounts.
- Exploiting Misconfigured Services: Misconfigured services or processes running with elevated privileges may allow attackers to execute code with higher privileges.
- Exploiting Kernel Vulnerabilities: Attackers can exploit vulnerabilities in the operating system's kernel to gain system-level or root privileges.

Tools:

- MSF (msfconsole, msfvenom) (https://www.metasploit.com/download)
- Privilege Escalation Scripts(e.g., powersploit, LinEnum) (https://github.com/PowerShellMafia/PowerSploit)
- Password Cracking Tools (e.g., John the Ripper, Hashcat) (https://github.com/openwall/john)
- LinPEAS (Linux Privilege Escalation Awesome Script) (https://github.com/peass-ng/PEASS-ng)

5. Maintaining Access and Persistent Mechanisms

Maintaining access and persistent mechanisms are crucial in penetration testing for ensuring that an attacker can retain control over a compromised system and re-access it even after initial detection or remediation efforts. Some common techniques to maintain access are installing a backdoor, keylogger, or a Remote Access Trojan (RAT), create rogue accounts, or modify system configurations to allow re-entry even if the system is rebooted or patched. Attacker normally perform following activities in this phase:

- Installing Backdoors, rootkits, keyloggers, and reverse shells on target system to maintain access.
- Uploading web shells, which are malicious scripts that are uploaded on a web server.
- Creating new user accounts with administrative privileges.
- Setting up persistence mechanisms that re-establish access after reboots.

Tools:

- Metasploit Persistence Module
- Cobalt Strike
- PowerSploit
- Cron Jobs

6. Covering Tracks

Covering tracks is important in penetration testing as it demonstrates the methods attackers use to evade detection and hide their activities. To do this the attacker delete or modifies log entries, or do log spoofing (creating false trails). The attacker may also clear command history, perform time stomping, and erase evidence of persistence mechanisms. Attacker normally perform following activities in this phase:

- Log Clearing: Deleting or modifying system logs to remove traces of the attack.
- File Removal: Removing any files or tools that were uploaded during the test.
- Hiding Evidence: Using tools and techniques to conceal the presence of malware or changes made to the system.

Tools:

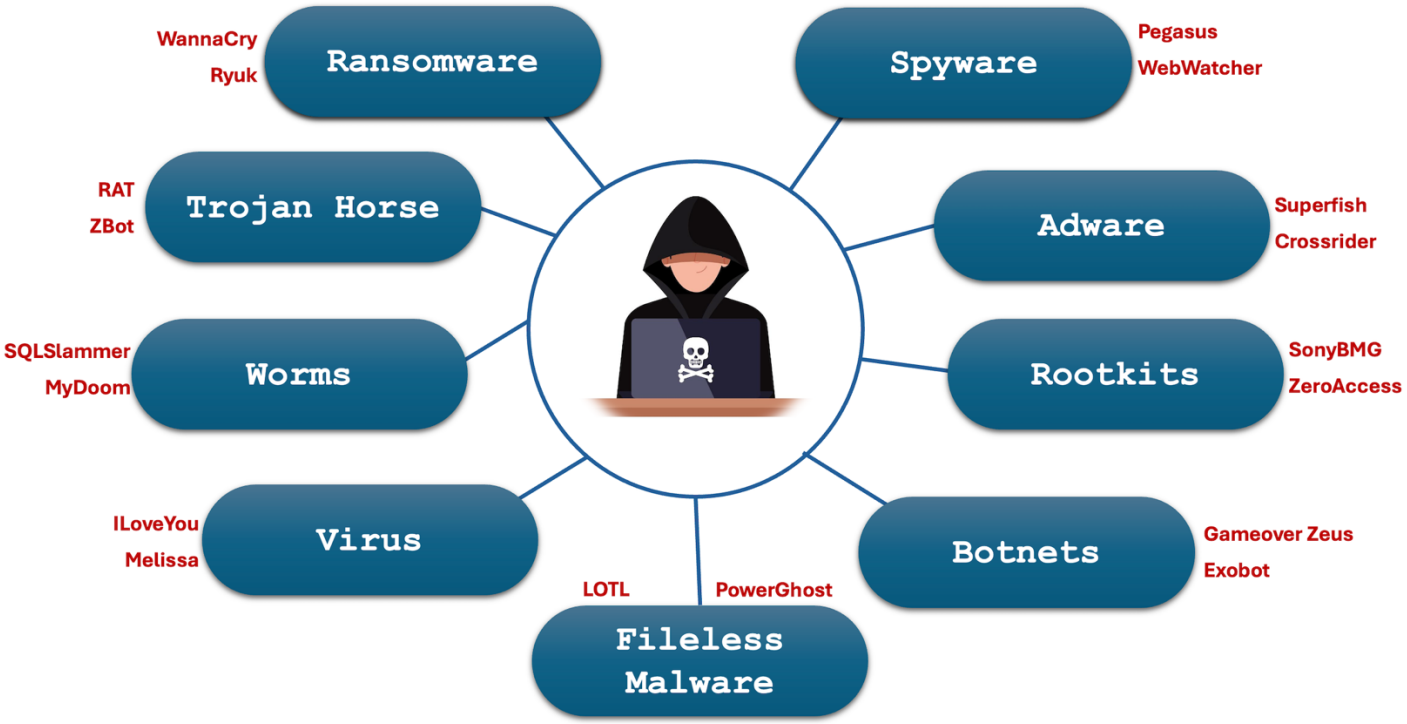
- Metasploit's Meterpreter `clearev` command
- Log-cleaning scripts
- Secure file deletion tools (e.g., `shred`, `sdelete`)
- Rootkits that hide files and processes
- Timestomping, i.e., modifying timestamps of files to avoid detection

Cyber Attacks and Malware Types

*A **cyber-attack** is a malicious attempt by individuals or groups to compromise, disrupt, damage, or gain unauthorized access to computer systems, networks, or data.* These attacks can target a variety of entities, including individuals, organizations, or government institutions, and can have a wide range of motives such as financial gain, espionage, political activism, or simply causing disruption. Some of the impacts of cyber-attacks have been listed below:

- **Financial Loss**: Costs associated with recovery, fines, and loss of business.
- **Reputation Damage**: Loss of customer trust and confidence.
- **Data Breach**: Loss of sensitive information, including personal and financial data.
- **Operational Disruption**: Interruption of services and business operations.

In order to launch a cyber-attack, attackers use malwares. ***A malware is a malicious software designed to harm, exploit or compromise systems, network or data.*** Following image gives an overview of different categories of malware, with two examples of each.



1. Virus

A computer virus is a type of malicious software (malware) that infects a computer system and spreads by *replicating itself within files or by attaching itself to executable programs*. Viruses replicate by inserting their code into other programs or documents. When these infected programs or documents are executed, the virus activates and may further spread to other files on the system. Virus can have following impact on the victims:

- Data Corruption: Viruses can modify or delete files, leading to data loss and potential damage to critical information.
- System Instability: Some viruses consume system resources, slowing down or crashing the victim's computer.
- Privacy Breach: Certain viruses can capture sensitive information such as passwords or credit card details, exposing victims to identity theft.

Examples:

- **ILOVEYOU Virus**:
    - Release: May 2000
    - Propagation: Spread via email as a love letter with an attachment named "LOVE-LETTER-FOR-YOU.TXT.vbs."
    - Impact: Infected millions of computers worldwide, causing extensive damage by overwriting files and stealing passwords. It forced organizations to shut down their email systems temporarily.
    - Link: ILOVEYOU virus
- **Melissa Virus**:
    - Release: March 1999
    - Propagation: Disseminated via infected Microsoft Word documents attached to emails with enticing subject lines.
    - Impact: Rapidly spread through email, causing email servers to crash due to the high volume of emails sent. It disrupted email communication globally, incurred high costs.
    - Link: Melisssa virus

2. Worms

A computer worm is a type of malicious software (malware) that, unlike viruses, can self-replicate and *spread independently without needing to attach itself to other programs or files*. Worms typically exploit vulnerabilities in network protocols or operating systems to propagate quickly across interconnected computers or devices. Key characteristics and behaviors of computer worms are:

- Self-Replication: Worms can create copies of themselves and spread autonomously to other computers or devices on a network.
- Network Propagation: They exploit security vulnerabilities in network services or operating systems to spread rapidly across interconnected systems.
- No Host Attachment: Unlike viruses, worms do not need to attach themselves to executable files or documents to spread.

Examples:

- **SQL Slammer:**
  - Release: January 2003
  - Propagation: Exploited a vulnerability in Microsoft SQL Server and Desktop Engine (MSDE), spreading rapidly across the Internet.
  - Impact: Infected hundreds of thousands of systems worldwide within minutes, causing massive network congestion and disruptions to Internet services.
  - Link: SQL Slammer
- **MyDoom:**
  - Release: January 2004
  - Propagation: Spread via email attachments and peer-to-peer (P2P) file sharing networks.
  - Impact: Became one of the fastest-spreading email worms at the time, launching Distributed Denial of Service (DDoS) attacks against targets and causing significant disruption to Internet services. It also installed back doors on infected systems.
  - Link: Mydoom

3. Trojan Horse

A Trojan horse, often referred to simply as a **Trojan** is a type of malicious software (malware) that disguises itself as legitimate software or files to deceive users into executing or installing it on their systems. Unlike viruses or worms, Trojans do not self-replicate. Instead, they *rely on social engineering tactics to trick users into unwittingly installing them*. Here are key characteristics and behaviors of Trojan horses:

- Disguised as Legitimate Software: Trojans often masquerade as harmless or useful software, files, or documents to trick users into downloading and executing them.
- No Self-Replication: Unlike viruses and worms, Trojans do not propagate by themselves. They require user interaction to be installed.
- Malicious Payload**:** Trojans can carry a wide range of malicious payloads, including backdoors, keyloggers, ransomware, spyware.

Examples:

- **Zbot:**
  - Type: Banking Trojan
  - Impact: Zeus targeted financial institutions, capturing login credentials and financial information from infected systems.
  - Link: Zbot
- **Remote Access Trojan (RAT):**
  - Type: Remote Access Trojan
  - Function: RATs provide remote control and administrative access to the attacker over the infected system.
  - Impact: Attackers can perform various malicious activities, such as spying on users through webcams or microphones, stealing files, or launching further attacks from the compromised system.
  - Link: RAT

4. Ransomware

Ransomware is a type of malicious software (malware) designed to encrypt files on a victim's computer or entire network, rendering them inaccessible until a ransom is paid. It is a form of extortion where attackers demand payment in exchange for decrypting the files and restoring access to the affected system. Here are key characteristics of ransomwares:

- Encryption: Ransomware encrypts files using strong encryption algorithms, making them unreadable without the decryption key held by the attacker.
- Ransom Demand: Attackers typically display a ransom note demanding payment (usually in cryptocurrency) in exchange for the decryption key.
- Propagation: Ransomware can spread through malicious email attachments, compromised websites, or vulnerabilities in software and operating systems.

Examples:

- **Ryuk:**
  - Release: August 2018
  - Target: Primarily aimed at businesses and large organizations.
  - Impact: Ryuk encrypts files and demands high ransom payments, often customized based on the victim's perceived ability to pay. It has been linked to financially motivated cybercrime groups and has caused significant financial losses.
  - Link: Ryuk
- **WannaCry:**
  - Release: May 2017
  - Propagation: Exploited a vulnerability in Microsoft Windows known as EternalBlue, which allowed it to spread rapidly across networks.
  - Impact: WannaCry infected hundreds of thousands of computers worldwide within days, affecting hospitals, government agencies, and businesses. It encrypted files and demanded ransom payments in Bitcoin.
  - Link: Wannacry

5. Spyware

Spyware is a type of malicious software (malware) designed to secretly gather information about a user's activities on their computer or device without their knowledge or consent. The purpose of spyware ranges from tracking browsing habits for advertising purposes to stealing sensitive information such as passwords, credit card numbers, and personal data. Here are key characteristics and behaviors of spyware:

- Information Gathering: Spyware monitors and collects various types of information, including keystrokes, personal information, browsing history.
- Remote Access: Some advanced spyware variants allow attackers to remotely access and control the infected device, enabling them to perform malicious activities discreetly.
- Data Transmission: Spyware transmits the collected information to remote servers controlled by the attackers or third parties for exploitation or resale.

Examples:

- **Pegasus:**
  - Type: Advanced spyware developed by NSO Group, an Israeli Cyber-arm company.
  - Target: Pegasus is known for targeting mobile devices (iOS and Android) and has been used to monitor journalists, activists, and political targets globally.

- Capabilities: Pegasus can intercept communications, track location, access contacts, messages, and capture audio and video from the device's microphone and camera.
- Link: Pegasus

- **WebWatcher:**
  - Type: Commercial spyware marketed as a parental control and employee monitoring solution.
  - Function: WebWatcher monitors computer and smartphone activities, including websites visited, emails, social media interactions, and keystrokes.
  - Link: Web Watcher

6. Adware

Adware is a type of software designed to display advertisements on a user's device, often in a disruptive or intrusive manner. While not always malicious, adware can become a significant nuisance, affecting user experience and system performance. Adware is often bundled with free software and gets installed without the user's full awareness or consent. Some characteristics of adware are listed below:

- Advertisement Display: Adware primarily functions to display advertisements, which can appear as pop-ups, banners, or within the software interface.
- User Tracking: Many adware programs track user behavior, such as browsing habits, search queries, and site visits, to deliver targeted advertisements.
- Bundling with Freeware: Adware is commonly bundled with free software downloads, often as part of the installation process, where users may inadvertently agree to install it.

Examples:

- **Super fish:**
  - Description: Superfish was an adware program that came pre-installed on some Lenovo laptops. It injected advertisements into web pages viewed by the user.
  - Impact: Superfish raised significant security concerns because it used a self-signed root certificate to intercept HTTPS traffic, potentially exposing users to man-in-the-middle attacks.
  - Link: Super fish
- **Crossrider:**
  - Description: Crossrider is a development platform for creating browser extensions and toolbars. Some of the extensions developed using Crossrider were classified as adware because they displayed intrusive ads and collected user data.
  - Impact: Crossrider-based adware often altered browser settings, injected advertisements, and tracked user browsing activities without proper consent.
  - Link: Crossrider

7. Rootkits

A rootkit is a type of malicious software designed to gain unauthorized access to and maintain control over a computer system while *hiding its presence from the user and security software*. Rootkits achieve this by altering the operating system or other software to mask their activities. They are often used by attackers to maintain long-term control over infected systems, allowing them to execute further malicious activities undetected. Rootkits characteristics are listed below:

- Stealth and Evasion: Rootkits are highly effective at hiding their presence from users and security tools by concealing their files, processes, and network connections.
- Persistence: They are designed to survive reboots and persist on the system, providing attackers with continued access.
- Privilege Escalation: Rootkits often gain root or administrative privileges, allowing them to make significant changes to the system and bypass security mechanisms.

Examples:

- **Sony BMG:**
  - Description: In 2005, Sony BMG (Bertelsmann Music Group) included a rootkit as part of its digital rights management (DRM) software on music CDs to prevent unauthorized copying.
  - Impact: The rootkit was installed without user consent and hid itself deeply within the operating system. It caused significant controversy when it was discovered, as it introduced security vulnerabilities and exposed users to additional malware.
  - Link: Sony BMG

- **Zero Access Rootkit:**
  - Description: It is a series of php scripts kept on attacker's website. Victim machine gets infected on visiting the website. The malware escalate privileges and gets installed on the system. The malware disables security services and turn the system into a bot.
  - Impact: Zero Access achieved high levels of stealth and persistence, infecting millions of computers worldwide. It disabled security software and redirected web traffic to generate fraudulent ad clicks.
  - Link: Zero Access

8. Botnets

A botnet is a network of compromised computers, known as "bots" or "zombies," that are controlled remotely by an attacker, often called a "botmaster" or "bot herder." Botnets are typically used to conduct large-scale cyber-attacks and other malicious activities.

- Remote Control: Botnets allow attackers to control multiple compromised systems from a central point, coordinating their activities for various malicious purposes.
- Large Scale: Botnets can range in size from a few hundred to millions of infected devices, providing significant computing power and bandwidth to the attacker.

Examples:

- **GameOver Zeus:**
  - Description: GameOver Zeus (GOZ) is a sophisticated botnet primarily used for banking fraud and the distribution of ransomware. It is a variant of the Zeus malware.

- Impact: Gameover Zeus infected hundreds of thousands of computers worldwide, stealing banking credentials and facilitating fraudulent transactions. The affected systems were MS Windows 95, 98, Me, 2000, XP, Vista, 7 and 8. It also affected MS Server2003, Server2008, Server2008R2 and Server2012.
- Link: Gameover Zeus

- **Exobot:**
  - Description: Exobot is an Android banking trojan that also functions as a botnet, enabling attackers to control infected Android devices.
  - Impact: Exobot steals banking credentials by overlaying banking apps with fake login screens. It also can intercept SMS messages, allowing attackers to bypass two-factor authentication.
  - Link: Exobot

9. Fileless Malware

Fileless malware is a type of malicious software that does not rely on traditional files to infect a computer. Instead, it exploits existing, legitimate tools and features within the operating system to carry out malicious activities. This makes fileless malware particularly *difficult to detect and remove*, as it leaves few traces. Some characteristics of fileless malwares are listed below:

- Leveraging Legitimate Tools: It exploits built-in system tools and utilities, such as PowerShell, Windows Management Instrumentation (WMI), and other scripting frameworks, to execute malicious activities.
- Persistence: Fileless malware can achieve persistence by modifying registry settings, scheduling tasks, or exploiting legitimate system processes to restart upon reboot.

Examples:

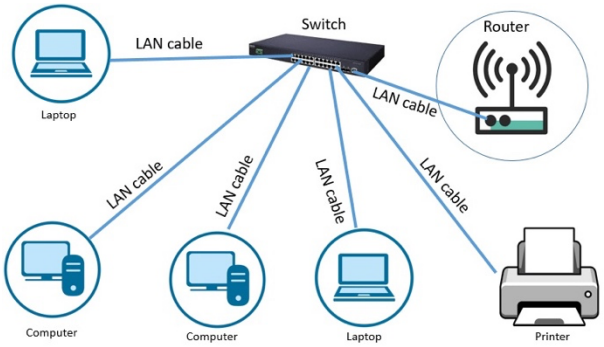
- **PowerGhost:**
  - Description: PowerGhost is a type of fileless malware that primarily targets enterprise networks. It spreads through networks using exploits and weak credentials.
  - Impact: PowerGhost is used to mine cryptocurrency on infected systems, consuming significant CPU and memory resources, leading to degraded system performance.
  - Link: PowerGhost
- **Living of the Land:**
  - Description: Living off the land refers to a technique used by attackers where they utilize legitimate tools and processes already present in the operating system to conduct malicious activities.
  - Impact: By using trusted system tools like PowerShell, WMI, and PsExec, attackers can perform a variety of malicious activities, such as data exfiltration, lateral movement, and command and control operations.
  - Link: LOTL

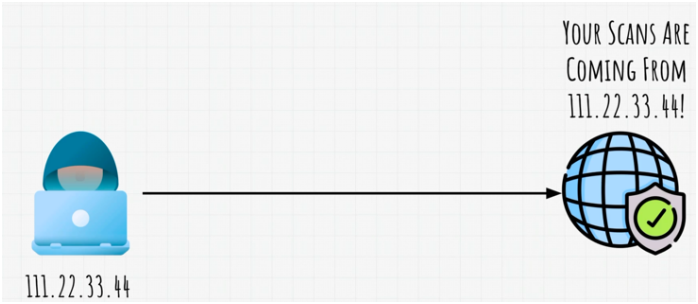
Achieving Anonymity

Anonymity *refers to the state of being unidentified or untraceable within a digital environment*. This concept is often associated with privacy and security measures designed to protect an individual's identity and activities from being discovered or tracked by others, including service providers, other users, and malicious entities.

Connecting to the Internet can be done in various ways, each offering different levels of privacy, security, and complexity. Here's a detailed explanation of each approach:

1. Directly Connected via LAN





Description: Connecting directly to the Internet through a Local Area Network (LAN) typically involves using an Ethernet cable or a Wi-Fi connection provided by a router. This is the most common method for home and office environments.

Pros:

- **Simplicity**: Easy to set up and use with minimal configuration required.
- **Speed**: Generally, offers high-speed Internet access, depending on the quality of the ISP and network infrastructure.
- **Stability**: Provides a stable connection with low latency.

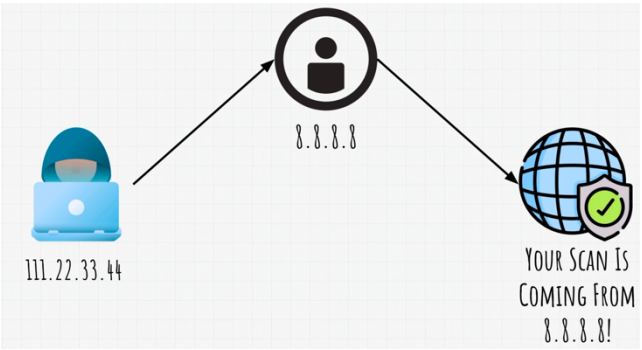
Cons:

- **Lack of Privacy**: Using a browser like Google chrome provides almost no privacy or anonymity at all. These browsers and many other applications using which you access the Internet collect your private data in order to show you personalized ads. Similarly, when doing penetration testing, e.g., doing a *nmap* scan of a network, your IP address and other information is exposed to the target machine running websites and online services and therefore making it easy to track you and log your activities.
- **Security Risks**: Without additional security measures, such as a firewall or antivirus software, the system may be vulnerable to attacks.

Check your IP: Open your browser and type "*what is my ip*" to check the IP, the location, and the name of the service provider ☺. Or visit https://www.dnsleaktest.com/

```
$ curl ifconfig.me
$ curl ipinfo.io/ip
$ curl icanhazip.com
```

2. Use a Proxy Server



Description: A proxy server acts as an intermediary between the user's device and the Internet. When you connect to a website using a proxy, the website only sees the IP address of the proxy server, thus providing some level of anonymity. Links for some free proxy services that you can use are given below:

- https://hideme.com/lander
- https://www.proxfree.com/

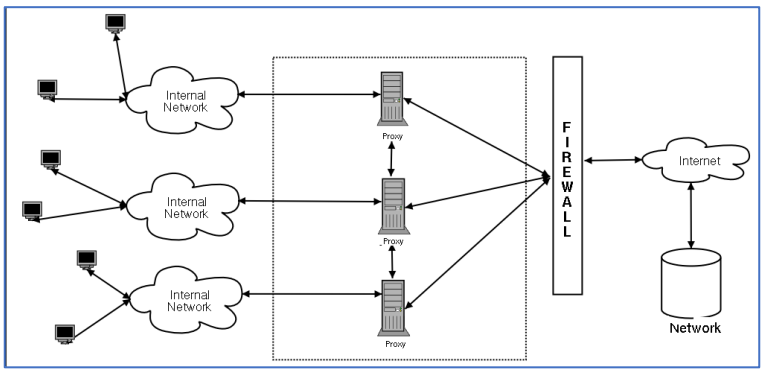
Pros:

- **Anonymity**: Hides the user's IP address, making it harder to track online activities.
- **Access Control**: Can be used to bypass geo-restrictions and access blocked content.
- **Content Filtering**: Often used in organizations to monitor and control employee Internet usage.

Cons:

- **No Encryption**: Many proxies do not encrypt traffic, leaving it vulnerable to interception.
- **Logging**: Proxy servers can log user activity, potentially compromising privacy.
- **Performance**: Can introduce latency, slowing down Internet speed.

3. Use Multiple Proxies



Description: Using **multiple proxies** means manually switching between different proxy servers for each session or connection. You are not routing your traffic through several proxies in a sequence for the same connection, but rather using one proxy at a time and changing it for each new request or session.

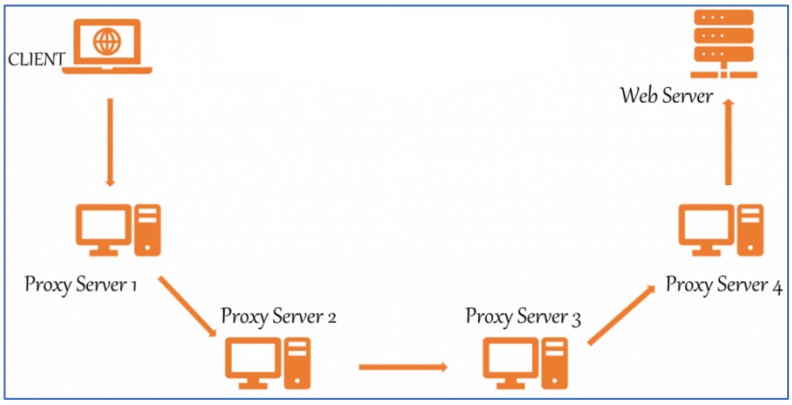
Pros:

- **Increased Anonymity**: Enhanced privacy compared to a single proxy by adding more layers of obfuscation.
- **Bypassing Censorship**: Makes it harder for authorities or attackers to block or track Internet access.

Cons:

- **Complex Setup**: Configuring and maintaining multiple proxies can be complex.
- **Performance Degradation**: Increased latency due to multiple hops, potentially slowing down the connection significantly.
- **Reliability Issues**: If one proxy in the chain fails, it can disrupt the entire connection.

4. Use Proxy Chains



Description: A **proxy chain** refers to routing your Internet traffic through **multiple proxies** in sequence (i.e., chaining proxies). Each proxy server forwards your traffic to the next one in the chain before it reaches the destination. This makes it harder to trace the origin of the traffic because each hop hides the previous one. Proxy chains come pre-installed in Kali Linux and can be configured from the `/etc/proxychains.conf` file.

Pros:

- **Enhanced Anonymity**: Routes traffic through multiple proxies, increasing anonymity.
- **Versatility**: Can configure different types of proxies in a chain, including public and private proxies.
- **Integration with Tools**: Useful for penetration testing tools and anonymizing traffic.

Cons:

- **Performance**: Can significantly slow down the Internet connection due to multiple hops.
- **Configuration Complexity**: Requires careful configuration to ensure all proxies are functioning correctly.
- **Potential Logging**: Each proxy in the chain can log traffic, potentially compromising privacy.

Installation and Configuring Proxy Chains:

```
$ sudo apt-get install proxychains4 -y
$ sudo vim /etc/proxychains4.conf
```

Just go through the above configuration file. At the very top you can find the types of proxies like http, socks4, socks5, and so on (Do read about these). In above configuration file remove the hash symbol before **`dynamic_chain`** and comment the **`strict_chain`** and at the very bottom of the file just add a new line that will add socks5 support **`socks5 127.0.0.1  9050`**

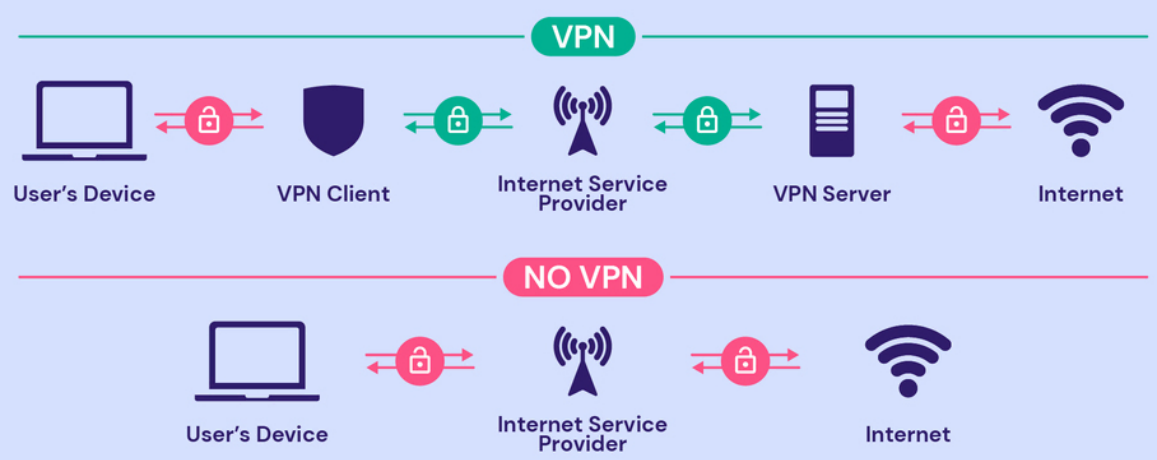
```
$ curl icanhazip.com
139.135.32.190
$ proxychains curl icanhazip.com
185.220.101.37
```

If `proxychains` do not work, you may need to start the `tor` service using `systemctl` command. Finally, to test give the following command to run `firefox` browser using `proxychains`

```
$ proxychains firefox
```

Now visit https://www.dnsleaktest.com/ inside Firefox running under **proxychains**, and you will see most probably, you will get a new IP every five minutes ☺.

5. Use Virtual Private Network (VPN) Service



Description: A VPN server encrypts the user's Internet traffic and routes it through a remote server, masking the user's IP address and providing a secure connection. Moreover, it works at OS level and changes the IP address of all NW applications. There exist free as well as paid services like OpenVPN and ProtonVPN respectively. VPNs are widely used for enhancing privacy, security, and bypassing geo-restrictions. Links for some free proxy services that you can use are given below:

https://protonvpn.com/
https://www.tunnelbear.com/

Pros:

- **Strong Encryption**: Provides secure, encrypted connections, protecting data from interception.
- **Privacy**: Masks the user's IP address, making it difficult to track online activities.
- **Bypass Geo-restrictions and Censorship**: Allows access to geo-blocked content and services.

Cons:

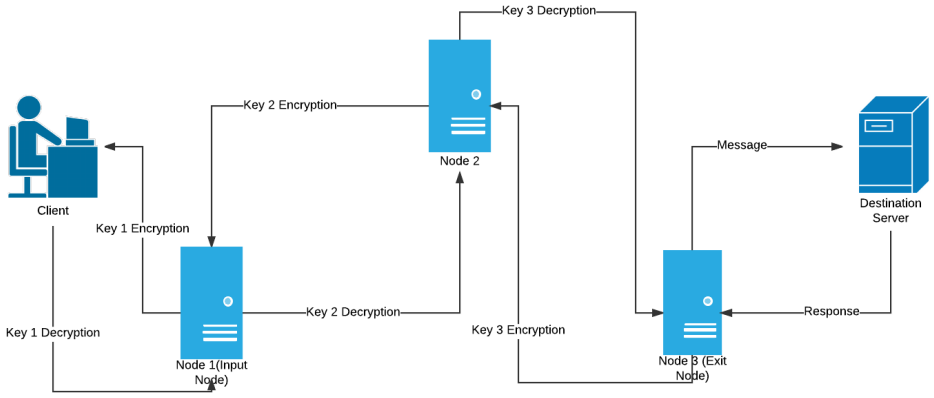
- **Trust in Provider**: Users must trust the VPN provider not to log or misuse their data.
- **Cost**: High-quality VPN services typically require a subscription fee.
- **Speed Reduction**: Encryption and routing through remote servers can reduce connection speeds.

Using Proton VPN:

You can create a login on the Proton VPN by visiting https://protonvpn.com/ and then use it from your Kali terminal as shown below:

```
$ protonvpn-cli login <your-user-name>
$ protonvpn-cli connect
$ curl ifconfig.me
$ protonvpn-cli disconnect
```

6. Use Tor Browser (The Onion Router) https://www.torproject.org/



Description: ***The Onion Router (TOR)*** *is privacy focused web browser that routes your Internet traffic through multiple volunteer-operated servers (nodes) across the globe to anonymize your online activity*. The data is sent through at least three nodes. The *Entry node*, which knows your IP but not your destination. The *Middle node* passes encrypted data. The *Exit node* decrypts the final layer and sends the request to destination website, which sees the Exit node's IP address and not yours. The TOR browser is used to access all the **`.onion`** websites and the deep web.

Pros:

- **Strong Anonymity**: Traffic is encrypted and routed through multiple nodes, thus making your traffic anonymous on the Internet and difficult to trace.
- **Access to .onion Sites**: Enables access to hidden services and websites on the dark web.
- **Censorship Resistance**: Helps bypass Internet censorship and restrictions.

Cons:

- **Performance**: Can be slow due to multiple layers of encryption and routing.
- **Blocked Sites**: Some websites block traffic from Tor nodes.
- **Reputation**: Associated with illegal activities on the dark web, although it also supports legitimate uses.

Installation and using Tor Browser:

```
$ sudo apt update
$ sudo apt-get install tor torbrowser-launcher
$ sudo systemctl status/start/restart/stop tor
$ torbrowser-launcher
```

- The last command will launch your Tor browser, and for the first time it may take a while. Once you see the GUI of Tor browser, the next step is to verify that the tor service is running on your OS. For this just type the following address inside the address bar of the Tor browser https://torbrowser.project.org
- Once the index page is displayed, it will display the public IP through which your browser is connected to the Internet, which will be different from the public IP of your Linux machine and you can confirm this by checking your public IP in another browser tab by typing **what is my ip** OR by visiting https://www.dnsleaktest.com/
- Please make time and open the settings of Tor browser, especially Privacy & Security settings, and over there specially the Security Level that can be set to Standard/Safer/Safest. ☺

Visit Onion Websites: Many popular websites and services offer onion services (also known as TOT hidden services) to provide secure, private access through the TOR network. These onion services are typically used for enhanced privacy and anonymity, and they can only be accessed via TOR browser or any browser that routes traffic through the TOR network. The URLs of onion websites often change, so you can always search for "*hidden wiki URLs*" on your TOR browser to search from a large list of links of `.onion` domains. (https://thehidden-wiki.org/wiki/index.php/Main_Page)

- **DuckDuckGo:** DuckDuckGo is a privacy-focused search engine that doesn't track users. Its onion service allows you to search anonymously via TOR. Open the TOR browser, and enter the onion URL into the address bar.

  `https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/`

- **The New York Times:** The New York Times provides an onion site to offer secure and anonymous access to its journalism, especially for those in countries with media censorship Open the Tor Browser, and enter the onion URL into the address bar.

  `https://www.nytimes3xbfgragh.onion/`

- **The BBC:** The BBC provides an onion version of its website to make its news accessible in regions where it might be censored.

  `https://www.bbcnewsv2vjtpsuy.onion/`

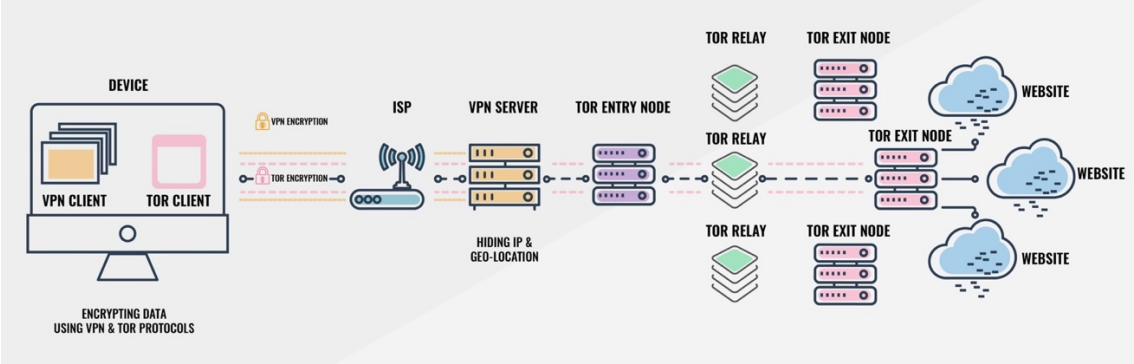
- **The Facebook:** The Facebook offers an onion version of its website to allow users in restrictive environments to access the social network securely. Note that Facebook still collects user data, even though TOR.

  `https://www.facebookcorewwwi.onion/`

Popular TOR Browsers:

- The Tor Browser: https://www.torproject.org/
- Brave Browser: https://brave.com/
- Onion Browser for iOS: https://onionbrowser.com/
- Tor Browser for Android: https://play.google.com/store/

7. TOR Browser + VPN



Description: Combining TOR with a VPN provides an additional layer of security and privacy. Users connect to a VPN before accessing the Tor network, adding an extra hop between their device and the Tor entry node.

Pros:

- **Enhanced Privacy**: Encrypts traffic through both the VPN and Tor network, providing strong anonymity.
- **Hidden Tor Usage**: VPN hides the use of Tor from the ISP, adding an extra layer of obfuscation.
- **Security**: Provides defense-in-depth by combining the benefits of both technologies.

Cons:

- **Significant Speed Reduction**: Can be very slow due to double encryption and multiple hops.
- **Complexity**: More complex to set up and manage compared to using either technology alone.
- **Cost**: Requires a VPN subscription in addition to using Tor.

Note: You can use a layered approach by using TOR + VPN + ProxyChains

8. Use a Persistent USB Containing Portable OS

Description: Using a USB drive with a portable operating system (such as Tails https://tails.net/ or Whonix https://www.whonix.org/) allows users to boot in a secure, isolated environment that leaves no trace on the host machine. This method is ideal for maintaining high levels of privacy and security as these OSs use layered approach by using TOR+VPN+ProxyChains.

Pros:

- **High Level of Anonymity**: Does not leave any traces on the host computer, ensuring complete privacy.
- **Portable**: Can be used on any compatible computer, providing a consistent and secure environment.
- **Pre-configured Tools**: Often includes pre-configured privacy and security tools, such as Tor and secure communication software.

Cons:

- **Convenience**: Requires carrying and securely storing the USB drive.
- **Compatibility Issues**: May not work seamlessly with all hardware configurations.
- **Learning Curve**: May require users to learn and adapt to a different operating system and tools.

BONUS: The Surface Web vs The Deep Web vs The Dark Web

The Deep Web

Definition: The deep web refers to all parts of the Internet that are not indexed by traditional search engines like Google, Bing, or Yahoo. This means that these pages cannot be found through standard search queries.

Scope:

- **Size**: The deep web is estimated to be significantly larger than the surface web, possibly hundreds of times greater in terms of data and content.
- **Content**: Includes private databases, academic and scientific research, medical records, legal documents, financial records, and internal organizational sites. It also encompasses online banking, subscription-only websites, and any other content that requires authentication or is behind a paywall.

Accessibility:

- **Access**: Accessible to anyone with the correct URL and permissions. For example, accessing your email, logging into your online banking account, or reading a private academic journal all involve entering the deep web.
- **Purpose**: Provides a means for secure, private information exchange and storage. It is used by individuals, businesses, and governments to protect sensitive information and ensure privacy.

Examples:

- Academic databases like JSTOR or IEEE Xplore.
- Government databases and resources not intended for public access.
- Corporate intranets and internal systems.

The Dark Web

Definition: The dark web is a small portion of the deep web that has been intentionally hidden and is inaccessible through standard web browsers. It requires specific software, configurations, or authorization to access.

Scope:

- **Size**: A much smaller subset of the deep web, but it is the most notorious due to its association with illegal activities.
- **Content**: Includes marketplaces for illicit goods and services (such as drugs, weapons, and stolen data), forums for criminal activity, and other illegal content. However, it also hosts sites for whistleblowing, political activism, and secure communication.

Accessibility:

- **Access**: Requires special software like Tor (The Onion Router) or I2P (Invisible Internet Project) to access. These tools anonymize user activity by routing traffic through multiple nodes.
- **Purpose**: Provides a platform for anonymous communication and transactions. It is used by those seeking privacy and anonymity, such as political activists, journalists, and individuals living under oppressive regimes. It is also exploited by criminals for illicit activities.

Examples:

- **Marketplaces**: Silk Road (defunct), AlphaBay (defunct), and others that emerge frequently.
- **Communication Platforms**: SecureDrop for whistleblowers to share information with journalists.
- **Forums**: Dark web forums for hackers and cybercriminals.

Key Differences

- **Indexing**:
  - **Deep Web**: Not indexed by search engines but accessible with correct permissions.
  - **Dark Web**: Not indexed and intentionally hidden; requires special software to access.
- **Purpose and Content**:
  - **Deep Web**: Legitimate content including private and sensitive information.
  - **Dark Web:** Mixed content with many illegal activities alongside legitimate uses.
- **Access**:
  - **Deep Web**: Accessible through standard web browsers with proper credentials.
  - **Dark Web**: Accessible only through special browsers like TOR, ensuring anonymity.
- **Security and Privacy**
  - **Deep Web**: Generally secure and used for legitimate purposes. Standard security measures apply, such as strong passwords and encryption.
  - **Dark Web**: High-risk area; users must take extra precautions like using VPNs, avoiding malicious links, and being aware of the potential for scams and law enforcement activity.