**Module 3: CS - Cyber threats & CEH**

**1. What are the Different Types of Hacking Methods?**

**Hacking methods** refer to techniques used by attackers to exploit vulnerabilities in systems, networks, or devices. Understanding these methods is crucial for implementing effective security measures. Here are some common hacking methods:

1. **Phishing Attacks**:
   * **Definition**: Deceptive attempts to obtain sensitive information by masquerading as a trustworthy entity via email, messaging, or websites.
   * **Example**: An attacker sends an email that appears to be from a bank, urging the recipient to verify their account information. The link directs them to a fake website where their login credentials are captured.
2. **Malware Injection**:
   * **Definition**: Inserting malicious software into a system to disrupt operations, gather sensitive information, or gain unauthorized access.
   * **Types**: Viruses, Worms, Trojans, Ransomware, Spyware.
   * **Example**: The **WannaCry ransomware** encrypted data on infected computers and demanded ransom payments in Bitcoin.
3. **SQL Injection**:
   * **Definition**: Inserting malicious SQL code into web application input fields to manipulate the database.
   * **Example**: An attacker enters ' OR '1'='1 in a login field, causing the system to bypass authentication and grant access.
4. **Cross-Site Scripting (XSS)**:
   * **Definition**: Injecting malicious scripts into trusted websites, which then execute in other users' browsers.
   * **Example**: A forum allows users to post comments without proper input validation. An attacker posts a comment with a <script> tag that steals cookies of anyone who views it.
5. **Denial of Service (DoS) and Distributed Denial of Service (DDoS) Attacks**:
   * **Definition**: Flooding a network or server with excessive traffic to make it unavailable.
   * **Example**: A network of infected devices (botnet) sends overwhelming traffic to a website, causing it to crash.
6. **Man-in-the-Middle (MitM) Attacks**:
   * **Definition**: Intercepting communication between two parties to eavesdrop or alter the data.
   * **Example**: An attacker intercepts communication on a public Wi-Fi network, capturing login credentials sent over unsecured connections.
7. **Brute Force Attack**:
   * **Definition**: Attempting all possible password combinations until the correct one is found.
   * **Example**: Using automated software to try millions of password combinations on an account.
8. **Password Spraying**:
   * **Definition**: Using a common password across many accounts before moving on to the next password.
   * **Example**: Trying "Password123" on thousands of user accounts in an organization.
9. **Credential Stuffing**:
   * **Definition**: Using leaked credentials from one breach to access accounts on other services due to password reuse.
   * **Example**: If users use the same email and password on multiple sites, a breach in one service compromises others.

**2. Explain Types of Password Attacks**

Password attacks are methods used by attackers to obtain an individual's password illicitly. Here are the main types:

1. **Dictionary Attack**:
   * **Definition**: Using a list of common words and phrases to guess passwords.
   * **Example**: Trying passwords like "123456", "password", "qwerty", "letmein".
2. **Brute Force Attack**:
   * **Definition**: Systematically checking all possible passwords until the correct one is found.
   * **Example**: An attacker uses a tool to try every combination from "aaaa" to "zzzz" for a 4-letter password.
3. **Rainbow Table Attack**:
   * **Definition**: Utilizing precomputed tables of hashed passwords and their corresponding plaintext to reverse cryptographic hash functions.
   * **Example**: An attacker matches the hashed password stored in a system with a hash in the rainbow table to find the original password.
4. **Phishing**:
   * **Definition**: Trick users into revealing passwords by posing as a legitimate entity.
   * **Example**: Fake emails that prompt users to reset their password via a malicious link.
5. **Keylogger Attack**:
   * **Definition**: Recording keystrokes to gather passwords and other sensitive data.
   * **Example**: Malware installed on a computer that records everything typed, including passwords.
6. **Shoulder Surfing**:
   * **Definition**: Observing someone entering their password.
   * **Example**: Watching over someone's shoulder at an ATM or while they log into their computer.
7. **Credential Stuffing**:
   * **Definition**: Using compromised credentials from one site to access accounts on other sites.
   * **Example**: After a data breach, using the stolen credentials on multiple platforms expecting password reuse.
8. **Password Spraying**:
   * **Definition**: Trying a common password on many different accounts.
   * **Example**: Using “Welcome1” across all user accounts in an organization.

**Protection Measures:**

* Use complex, unique passwords and change them regularly.
* Implement multi-factor authentication (MFA).
* Educate users about phishing and social engineering tactics.
* Use account lockout policies after multiple failed login attempts.

**3. Explain Types of Steganography with QuickStego and Echo**

**Steganography** is the practice of concealing messages or information within other non-secret text or data. It hides the very existence of the message, unlike cryptography, which obscures the content but not the existence.

**Types of Steganography:**

1. **Text Steganography**:
   * **Definition**: Hiding data within text files by altering formatting, using invisible characters, or manipulating text properties.
   * **Methods**:
     + **Line Shift Encoding**: Altering the vertical alignment of text lines.
     + **Word Shift Encoding**: Adjusting spacing between words.
     + **Feature Coding**: Modifying text features like font size or style.
2. **Image Steganography**:
   * **Definition**: Hiding data within image files by modifying pixel values or embedding data in the image's metadata.
   * **Methods**:
     + **Least Significant Bit (LSB) Insertion**: Altering the least significant bits of pixel values.
     + **Masking and Filtering**: Hiding data in more significant areas of an image so it's less detectable.

**Example with QuickStego**:

* + **QuickStego** is a tool that embeds text within an image file.
  + **Process**:
    - **Embedding**: The user inputs text into QuickStego and selects a cover image. The software alters the LSBs of the image pixels to encode the text.
    - **Extraction**: The recipient uses QuickStego to recover the hidden text from the image.
  + **Practical Example**:
    - Alice wants to send Bob a secret message. She writes the message in QuickStego, which embeds it in a photo of a cat. She emails Bob the image. Bob uses QuickStego to extract the message.

1. **Audio Steganography**:
   * **Definition**: Concealing information within audio files by modifying the audio signal.
   * **Methods**:
     + **LSB Coding**: Similar to image steganography but with audio samples.
     + **Echo Hiding**: Introducing echoes with specific delays representing binary data.

**Example with Echo Steganography**:

* + **Echo Steganography Tools** like **DeepSound** use echo hiding.
  + **Process**:
    - **Embedding**: The tool embeds data by adding echoes to the original audio signal. The delays represent binary ones and zeros.
    - **Extraction**: The recipient processes the audio file to detect the echo patterns and retrieve the hidden data.
  + **Practical Example**:
    - A musician hides a secret message in a song by adding subtle echoes. Fans with the decoding tool can extract special messages from the track.

1. **Video Steganography**:
   * **Definition**: Hiding information within video files by exploiting the vast amount of data in videos.
   * **Methods**:
     + Modifying pixel values in select frames.
     + Altering motion vectors in compressed video streams.
2. **Network Steganography**:
   * **Definition**: Embedding data within network protocols and traffic.
   * **Methods**:
     + **Protocol Steganography**: Using fields within network protocols not normally used or altering packet timing.

**4. Define Types of Viruses**

A **computer virus** is a type of malware that, when executed, replicates by modifying other computer programs and inserting its own code. Viruses can spread from one computer to another through network connections, removable media, or email attachments.

**Types of Viruses:**

1. **Boot Sector Virus**:
   * **Definition**: Infects the master boot record (MBR) of a storage device, ensuring it loads during startup.
   * **Example**: **Michelangelo Virus**
     + Activated on March 6th, it would overwrite critical system data, making the computer unbootable.
2. **File Infector Virus**:
   * **Definition**: Attaches to executable files (.exe, .com) and activates when the file is run.
   * **Example**: **Cascade Virus**
     + Caused letters on the screen to fall into a heap at the bottom (a visual payload).
3. **Macro Virus**:
   * **Definition**: Written in macro languages and targets software applications that use macros (e.g., Microsoft Office).
   * **Example**: **Melissa Virus**
     + Spread through email attachments containing infected Word documents.
4. **Polymorphic Virus**:
   * **Definition**: Changes its code pattern (signature) each time it replicates to avoid detection.
   * **Example**: **Storm Worm**
     + Sent in emails with varied subject lines and content, making it hard to filter.
5. **Resident Virus**:
   * **Definition**: Embeds itself in the system memory, allowing it to infect any file that is accessed.
   * **Example**: **Randex**
     + Resided in memory and allowed attackers to control infected computers remotely.
6. **Multipartite Virus**:
   * **Definition**: Infects in multiple ways, such as both the boot sector and executable files.
   * **Example**: **One\_Half Virus**
     + Encrypted parts of the hard drive and spread through files.
7. **Web Scripting Virus**:
   * **Definition**: Exploits vulnerabilities in web browsers and web pages.
   * **Example**: **JS/SpaceStalk**
     + A JavaScript virus that altered web pages and browser settings.

**Impact on Systems:**

* **Data Loss**: Some viruses delete or corrupt files.
* **Performance Degradation**: Consuming resources, slowing down the system.
* **Security Breaches**: Opening backdoors for other malware or attackers.
* **Spreading to Other Systems**: Through networks or shared files.

**Protection Measures:**

* Use updated antivirus software.
* Keep operating systems and applications patched.
* Avoid opening suspicious emails or downloading from untrusted sources.
* Regularly back up important data.

**5. Explain Any One Antivirus with Example**

Let's explore **Avast Antivirus** as an example.

**Avast Antivirus**

**Overview:**

* **Avast** is a cybersecurity software developed by Avast Software, offering protection against various types of malware and online threats.
* Provides solutions for individuals and businesses.
* Offers both free and premium versions.

**Features:**

1. **Real-Time Protection**:
   * Monitors system activity and blocks malware before it infects the system.
   * Protects against viruses, spyware, ransomware, and phishing attacks.
2. **Smart Scan**:
   * Combines scans for viruses, outdated software, network threats, and performance issues into a single scan.
3. **Behavior Shield**:
   * Monitors the behavior of applications in real-time to detect suspicious activities.
4. **Web Shield**:
   * Checks URLs and certificates to ensure websites are safe before loading.
   * Blocks malicious websites and downloads.
5. **Email Shield**:
   * Scans incoming and outgoing emails for malicious content.
6. **Sandbox**:
   * Allows users to run suspicious files in a safe, isolated environment to observe behavior without risking the main system.
7. **Wi-Fi Inspector**:
   * Scans the network for vulnerabilities and intruders.

**Example Usage:**

* **Scenario**:
  + Emma downloads a PDF attachment from an unknown email.
  + Avast's Real-Time Protection kicks in, scanning the file upon download.
  + The file is flagged as malicious due to embedded malware.
  + Avast moves the file to the **Virus Chest** (quarantine area) and notifies Emma.
  + Emma can choose to delete the file or restore it if it's a false positive.

**Additional Tools:**

* **Ransomware Shield**:
  + Protects personal photos and files from being encrypted by ransomware attacks.
* **Password Manager**:
  + Safely stores passwords and auto-fills them on websites.
* **SecureLine VPN**:
  + Encrypts internet connection for secure and private browsing.

**Benefits:**

* **User-Friendly Interface**: Easy to navigate for users of all technical levels.
* **Regular Updates**: Frequent virus definition updates to protect against new threats.
* **Lightweight**: Optimized to have minimal impact on system performance.

**Limitations:**

* **Pop-ups in Free Version**: Advertisements for premium features can be intrusive.
* **Advanced Features Require Subscription**: Some tools are only available in the paid version.

**Impact on Security:**

* By using Avast Antivirus, users significantly reduce the risk of malware infection.
* Real-time protection and regular scans help maintain system integrity.
* Additional features like Wi-Fi Inspector enhance overall network security.

**Additional Insights and Recommendations**

**Evolving Threat Landscape:**

* Cyber threats are constantly evolving, with attackers developing new methods to bypass security measures.
* Staying informed about the latest threats and updating security practices is essential.

**Best Practices for Security:**

1. **Use Strong, Unique Passwords**:
   * Incorporate uppercase, lowercase, numbers, and special characters.
   * Consider using a reputable password manager.
2. **Enable Multi-Factor Authentication (MFA)**:
   * Adds an extra layer of security beyond just a password.
3. **Regular Software Updates**:
   * Keep operating systems, software, and antivirus programs up to date to patch vulnerabilities.
4. **Educate Yourself and Others**:
   * Awareness of phishing tactics and social engineering reduces the risk of falling victim.
5. **Backup Important Data**:
   * Regular backups ensure data can be recovered in case of ransomware attacks or system failures.
6. **Secure Networks**:
   * Use strong encryption (WPA3) for Wi-Fi networks.
   * Change default router passwords.

**Understanding the Ethics of Security Tools:**

* Tools like steganography software and network utilities have legitimate uses in security testing and privacy protection.
* However, they can be misused for malicious purposes.
* It's important to use these tools responsibly and legally.

**The Role of Antivirus Software:**

* An essential component of a multi-layered security strategy.
* Not a standalone solution; should be combined with safe browsing habits and other security measures.
* **Explain Password Cracking Tools: pwdump7,Medusa and Hydra**

Password cracking tools are essential in penetration testing for assessing the strength of passwords and identifying vulnerabilities in authentication systems. Here are explanations of pwdump7, Medusa, and Hydra:

**pwdump7**

**pwdump7** is a password dumping tool used to extract password hashes from the Windows operating system's Security Account Manager (SAM) database. It is particularly useful for penetration testers who need to analyze the security of Windows systems.

**Key Features:**

* **SAM Database Extraction:** pwdump7 can extract password hashes from the SAM database, which is a critical component for Windows authentication.
* **Compatibility:** It supports various versions of Windows, including Windows NT, 2000, XP, and Server 2003.
* **Hash Extraction:** The tool can extract hashes in formats compatible with popular password cracking tools like John the Ripper and Hashcat.
* **Offline Analysis:** pwdump7 can be used to perform offline analysis of the extracted hashes, making it a versatile tool for security assessments.

**Usage Example:**

pwdump7.exe -s 127.0.0.1 -u -p

This command extracts password hashes from the local machine (127.0.0.1) and saves them to a file.

**Medusa**

**Medusa** is a parallel, modular, and fast login brute-forcer. It is designed to test various services for weak passwords and is widely used in penetration testing for assessing the security of authentication mechanisms.

**Key Features:**

* **Parallel Execution:** Medusa can perform multiple login attempts simultaneously, making it efficient for brute-force attacks.
* **Modular Design:** It supports a wide range of protocols, including HTTP, HTTPS, FTP, SSH, and more.
* **Customizable:** Users can create custom modules to test specific services or protocols.
* **Output Options:** Medusa provides various output options, including plain text, XML, and JSON, making it easy to integrate into other tools and workflows.

**Usage Example:**

medusa -h 192.168.1.1 -u admin -P /path/to/passwords.txt -M http

This command attempts to brute-force the HTTP login for the user "admin" on the target IP 192.168.1.1 using a list of passwords from the specified file.

**Hydra**

**Hydra** is another popular password cracking tool known for its speed and flexibility. It supports a wide range of protocols and is often used in penetration testing to identify weak passwords.

**Key Features:**

* **Speed:** Hydra is known for its high speed due to its parallel processing capabilities.
* **Protocol Support:** It supports a wide range of protocols, including HTTP, HTTPS, FTP, SSH, Telnet, and more.
* **Customizable:** Users can create custom modules to test specific services or protocols.
* **Output Options:** Hydra provides various output options, including plain text, XML, and JSON.

**Usage Example:**

hydra -l admin -P /path/to/passwords.txt 192.168.1.1 http-post-form "/login.php:username=^USER^&password=^PASS^:F=incorrect"

This command attempts to brute-force the HTTP login for the user "admin" on the target IP 192.168.1.1 using a list of passwords from the specified file. The command specifies the login form parameters and the failure condition.

**Comparison**

* **pwdump7** is specialized for extracting password hashes from Windows systems, making it useful for post-exploitation activities.
* **Medusa** and **Hydra** are general-purpose brute-forcing tools that support a wide range of protocols and are commonly used for initial access and credential testing.
* **Perform Practical on key logger tool.**

here are the steps to perform a practical exercise on a keylogger tool. For this example, we'll use **Keylogger-ng**, a popular open-source keylogger. Please note that using keyloggers without proper authorization is illegal and unethical. This exercise should only be performed in a controlled, authorized environment, such as a penetration testing lab or a personal machine for educational purposes.

**Steps to Use Keylogger-ng**

1. **Install Keylogger-ng:**
   * **On Linux:**

sudo apt-get update

sudo apt-get install keylogger-ng

* + **On Windows:** Download the latest release from the [Keylogger-ng GitHub page](https://github.com/keylogger-ng/keylogger-ng) and follow the installation instructions.

1. **Configure Keylogger-ng:**
   * Open a terminal or command prompt.
   * Navigate to the Keylogger-ng directory if necessary.
   * Run the configuration command:

keylogger-ng -c

* + Follow the prompts to configure the keylogger. You can set options like the log file location, email notifications, and more.

1. **Start Keylogger-ng:**
   * Start the keylogger with the following command:

keylogger-ng

* + The keylogger will begin capturing keystrokes.

1. **Test Keylogger-ng:**
   * Open a text editor or any application where you can type.
   * Type some test keystrokes to ensure the keylogger is capturing them.
   * Check the log file specified in the configuration to see if the keystrokes are being recorded.
2. **Stop Keylogger-ng:**
   * To stop the keylogger, you can simply terminate the process in the terminal or command prompt:
3. **Analyze the Logs:**
   * Open the log file specified during the configuration to analyze the captured keystrokes.
   * Ensure that sensitive information is not logged unless explicitly authorized.

**Additional Tips**

* **Ethical Considerations:** Always obtain proper authorization before using keyloggers on any system. Unauthorized use is illegal and unethical.
* **Legal Compliance:** Ensure that your activities comply with all applicable laws and regulations.
* **Security Measures:** Be aware of the security implications of using keyloggers. Ensure that the logs are stored securely and that access to them is restricted.
* **Steps to Create a Simple HTTP RAT Using NanoCore RAT**

1. **Download NanoCore RAT:**
   * Download the latest version of NanoCore RAT from a trusted source. For example, you can download it from [GitHub](https://github.com/NanoCoreRAT/NanoCore).
2. **Extract the Downloaded File:**
   * Extract the downloaded ZIP file to a directory of your choice.
3. **Open the Builder:**
   * Navigate to the extracted directory and open the **NanoCore.exe** file.
4. **Configure the Builder:**
   * In the NanoCore Builder, you will see various options to configure your RAT.
   * **IP Address:** Enter the IP address where you want to receive the connections.
   * **Port:** Enter the port number on which the RAT will listen for connections.
   * **Password:** Set a password for authentication.
   * **Install:** Check the "Install" option if you want the RAT to install itself on the target machine.
   * **Persistence:** Check the "Persistence" option if you want the RAT to start automatically on system boot.
   * **Melt:** Check the "Melt" option if you want the RAT to delete itself after execution.
5. **Build the RAT:**
   * Click the "Build" button to generate the RAT executable.
   * Choose a location to save the generated executable.
6. **Test the RAT:**
   * Transfer the generated executable to a test machine in a controlled environment.
   * Run the executable on the test machine.
   * Check your configured IP address and port to see if the RAT connects back to your machine.
7. **Analyze the Connection:**
   * Use a tool like Wireshark or a simple netcat listener to analyze the incoming connections.
   * Ensure that the RAT is functioning as expected and that you can control the target machine.

**Important Notes**

* **Legal and Ethical Considerations:** Creating and distributing malware is illegal and unethical. This guide is for educational purposes only and should be used in a controlled, authorized environment.
* **Security Measures:** Be aware of the security implications of using RATs. Ensure that the RAT is used responsibly and that access to it is restricted.
* **Compliance:** Ensure that your activities comply with all applicable laws and regulations.