Exploits

- Three Ways to Attack - Social, Physical, Network
    - **Social**
        - Ask the person for access, phishing.
        - Cognitive biases - look at how these are exploited.
        - Spear phishing.
        - Water holing.
        - Baiting (dropping CDs or USB drivers and hoping people use them).
        - Tailgating.
    - **Physical**
        - Get hard drive access, will it be encrypted?
        - Boot from linux.
        - Brute force password.
        - Keyloggers.
        - Frequency jamming (bluetooth/wifi).
        - Covert listening devices.
        - Hidden cameras.
        - Disk encryption.
        - Trusted Platform Module.
        - Spying via unintentional radio or electrical signals, sounds, and vibrations (TEMPEST - NSA).
    - **Network**
        - Nmap.
        - Find CVEs for any services running.
        - Interception attacks.
        - Getting unsecured info over the network.
- Exploit Kits and Drive by Download Attack
- Remote Control
    - Remote code execution (RCE) and privilege.
    - Bind shell (opens port and waits for attacker).
    - Reverse shell (connects to port on attackers C2 server).
- Spoofing
    - Email spoofing.
    - IP address spoofing.
    - MAC spoofing.
    - Biometric spoofing.
    - ARP spoofing.
- Tools
    - Metasploit.
    - ExploitDB.

- Shodan - Google but for devices/servers connected to the internet.
- Google the version number of anything to look for exploits.
- Hak5 tools.

Three Ways to Attack - Social, Physical, Network

In cybersecurity, exploits are actions or tools that attackers use to compromise systems. Attacks can generally be grouped into three categories based on the method of exploitation: **social, physical, and network attacks**. Each type uses different tactics to bypass security, obtain access, or gather sensitive information. Here's a breakdown of each attack category and the specific techniques commonly used within them.

1. Social Attacks

> Social attacks **exploit human psychology and trust rather than technical vulnerabilities**. They rely on manipulating individuals to gain access to systems, data, or credentials.

- Ask the Person for Access: Sometimes attackers simply ask for access, relying on the target's trust or lack of suspicion. Posing as legitimate personnel, they may gain entry by asking directly.
- **Phishing**: Attackers send fraudulent messages that appear to come from a trusted source, tricking users into revealing credentials or clicking malicious links.
- **Cognitive Biases**:
    - Attackers exploit cognitive biases such as authority bias (trusting authority figures), reciprocity (feeling obligated to return favors), and urgency (forcing quick decisions).
- **Spear Phishing**: A targeted form of phishing where attackers **tailor their approach to a specific individual or organization**, making it more believable and effective.
- **Water Holing**: Attackers compromise a website frequented by the target group, planting malware that infects visitors who belong to the target organization.
- **Baiting**: Attackers **leave physical media, such as USB drives or CDs**, in locations where employees will find them, hoping curiosity leads them to plug the media into a company computer.
- **Tailgating**: Attackers **follow legitimate employees into secure areas without providing credentials**, often relying on the social courtesy of holding doors open.

2. Physical Attacks

> Physical attacks **require proximity or direct access to devices**. These attacks focus on tampering with hardware or exploiting the lack of physical security measures.

- **Get Hard Drive Access**: Attackers gain direct access to a hard drive, which might lead to unauthorized data retrieval if it's unencrypted.
- **Boot from Linux**: Attackers with physical access can boot a system from an external Linux drive, bypassing the operating system's security to access files directly.
- **Brute Force Password**: Using software or hardware devices, attackers repeatedly guess passwords until they find the correct one.
- **Keyloggers**: Attackers install keyloggers on a computer to capture keystrokes, allowing them to record login credentials, passwords, and other sensitive information. Keyloggers can be hardware or software-based.
- **Frequency Jamming (Bluetooth/WiFi)**: Attackers **disrupt wireless communications by jamming** Bluetooth or Wi-Fi frequencies, potentially interfering with security devices or other network-dependent systems.

- **Covert Listening Devices**: Attackers plant hidden microphones or bugging devices to record conversations and collect intelligence.
- **Hidden Cameras**: Small cameras can be hidden in inconspicuous places, allowing attackers to monitor physical spaces, observe passwords being typed, or track user behavior.
- **Disk Encryption**: Encryption protects data on drives by making it unreadable without the decryption key. Attackers must circumvent encryption to gain access, often requiring sophisticated techniques.
- Trusted Platform Module (TPM): TPMs are hardware-based security modules embedded in devices to secure encryption keys and other sensitive data, providing additional security against physical attacks.
- **TEMPEST (NSA)**: TEMPEST refers to NSA research into spying via unintended signals (electromagnetic, sound, or vibration) emitted by electronic devices. Attackers may use these signals to gather data remotely from devices.

3. Network Attacks

> Network attacks exploit vulnerabilities within the network to intercept data, discover devices, or execute remote attacks.

- **Nmap**: Nmap is **a network scanning tool used to discover live hosts, open ports, and services on a network**. Attackers use Nmap to map network topologies and identify potential targets.
- **Find CVEs for Any Services Running**: Attackers search for known vulnerabilities (CVE IDs) associated with services running on discovered hosts. These vulnerabilities, if unpatched, can provide entry points into the system.
- **Interception Attacks**: Attackers use methods like man-in-the-middle (MitM) to intercept and read data packets sent over the network, often capturing login credentials and other sensitive information.
- Getting Unsecured Info Over the Network: Attackers monitor network traffic for unencrypted data transmissions, including credentials, files, and messages, which can be easily captured and read.

Summary

- **Social Attacks** manipulate people to gain access, leveraging tactics like phishing, cognitive biases, spear phishing, baiting, and tailgating.
- **Physical Attacks** exploit direct access to devices, using techniques such as booting from external drives, keylogging, frequency jamming, hidden cameras, and covert listening.
- **Network Attacks** involve probing the network for vulnerabilities and intercepting data, employing tools like Nmap, exploiting CVEs, interception techniques, and capturing unsecured data.

By understanding these attack vectors, organizations can implement better security practices, such as physical security controls, encryption, user training, and network monitoring, to defend against these varied attack types.

Exploit Kits and Drive by Download Attacks

Exploit Kits and drive-by download attacks are **methods attackers use to automatically exploit vulnerabilities on a target's device, often without any interaction from the user**. These techniques focus on finding and exploiting weaknesses, typically in web browsers or plugins, to deliver malware or gain unauthorized access.

1. Exploit Kits

- Definition: An exploit kit is **a software toolkit that automates the process of identifying and exploiting vulnerabilities on target systems**. These kits are **commonly hosted on compromised or malicious websites and are designed to exploit common software vulnerabilities** (like those in browsers, Flash, Java, or PDF readers).
- How Exploit Kits Work:
    - **Scanning for Vulnerabilities**: When a user visits a site hosting an exploit kit, the kit scans the user's system for known vulnerabilities based on the browser type, operating system, and installed plugins.
    - **Exploitation**: Once a vulnerability is identified, the kit delivers a payload (often malware) tailored to exploit that specific weakness.
    - **Payload Delivery**: Exploit kits typically **deliver malicious payloads**, such as **ransomware, spyware, or trojans, without user interaction**.
- Popular Exploit Kits:
    - Examples include **Angler, Rig, Nuclear, and Neutrino** exploit kits, which have been widely used to distribute ransomware and other malware.
- Security Implications: Exploit kits pose a significant risk as they automate attacks and target known vulnerabilities, meaning that systems with outdated software are particularly vulnerable. They allow attackers to infect large numbers of users with minimal effort.

2. Drive-By Download Attacks

- Definition: A drive-by download attack **occurs when a user unknowingly downloads and executes malware simply by visiting a compromised or malicious website**. Unlike other attacks, no interaction (such as clicking or downloading a file) is required from the user.
- How Drive-By Download Attacks Work:
    - **Compromised Website**: Attackers **inject malicious code into legitimate websites or create fake sites designed to attract users**.
    - **Redirection to Exploit Kit**: Once a user visits the compromised website, they may be redirected to a page hosting an exploit kit. This kit identifies vulnerabilities on the user's system and then delivers a malicious payload.
    - **Automatic Infection**: If a vulnerability is present, the malware downloads and executes without user interaction, often installing spyware, ransomware, or keyloggers.
- Common Infection Points:
    - Drive-by downloads **usually exploit vulnerabilities in web browsers or plugins like Flash Player, Java, and PDF viewers**.
- Security Implications: Drive-by download attacks are **difficult to detect and avoid**, especially for users with outdated software. They leverage the "invisible" nature of the attack, where infection occurs in the background as users simply browse the web.

How Exploit Kits and Drive-By Downloads Work Together

Exploit kits are often used in drive-by download attacks. When a user visits a compromised website, the exploit kit on the site scans for vulnerabilities. If any are found, the kit initiates a drive-by download, delivering malware to the user's device without any action required. This combination makes for a highly effective and automated attack strategy that can infect large numbers of users quickly.

Mitigation Strategies

1. **Keep Software Updated**: Regularly patching operating systems, browsers, and plugins helps close vulnerabilities that exploit kits target.
2. **Use Security Software**: Antivirus and anti-malware tools can detect and block exploit kit traffic and drive-by downloads.
3. **Enable Browser Security Features**: Many modern browsers have built-in protections that can block malicious scripts or warn users before they access unsafe sites.
4. Implement **Network Monitoring**: Monitor network traffic for unusual behavior that could indicate exploit kit activity or unauthorized downloads.
5. **User Education**: Encourage users to avoid suspicious websites and be cautious with unfamiliar links or downloads.

Summary

- **Exploit Kits** are automated tools that scan for vulnerabilities in a user's system and deliver malicious payloads when a weakness is found.
- **Drive-By Download** Attacks exploit these vulnerabilities to infect users automatically as they visit a compromised website.
- Combined, these techniques allow attackers to infect users on a large scale with minimal interaction, making them powerful methods for distributing malware.

Understanding exploit kits and drive-by download attacks is crucial for defending against them, as they can lead to silent infections with severe consequences, such as **data theft, ransomware, and system compromise**.

Remote Control

Remote control techniques **allow attackers to gain control over a compromised system, enabling them to execute commands, exfiltrate data, or install additional malware**. Remote control methods include **remote code execution (RCE), bind shells, and reverse shells**. These techniques are commonly used in the later stages of an attack, once initial access has been established.

1. Remote Code Execution (RCE) and Privilege

- Definition: Remote Code Execution (RCE) occurs when an **attacker is able to remotely execute commands on a target system**, typically through a vulnerability in software or applications. RCE exploits can allow attackers to **run arbitrary code with the same privileges as the application being exploited**.
- Privilege Levels:
  - **User Privileges**: If the code executes with user privileges, the attacker may be **limited** in their ability to access system resources and files.
  - **Administrative Privileges**: If the RCE exploit gives administrative or root privileges, **the attacker gains unrestricted control over the system**, allowing them to install additional tools, manipulate system settings, and move laterally within the network.
- Security Implications: **RCE is a highly dangerous vulnerability**, as it allows attackers to bypass authentication and execute commands directly on the system. The impact of RCE depends on the privilege level attained, with administrative RCE posing the highest risk.

2. Bind Shell

- Definition: A bind shell is a type of shell that **opens a port on the target machine and listens for incoming connections from the attacker**. Once the connection is established, the attacker can issue commands on the compromised system.
- How It Works:
  - **Opening a Port**: The compromised system opens a specific port and waits for an incoming connection.
  - **Attacker Connects**: The attacker connects to the open port using a client (like **Netcat**), gaining command-line access to the compromised system.
- Security Implications: Bind shells **can be detected by monitoring** for unusual open ports or listening services on a system. **Firewalls and intrusion detection systems (IDS) can help detect or block bind shells**, especially if the open port is uncharacteristic for that device.

3. Reverse Shell

- Definition: In a reverse shell, **the compromised system initiates a connection back to the attacker's system**. This allows the attacker to bypass certain firewall restrictions, **as outgoing connections are typically less restricted** than incoming ones.
- How It Works:
  - **Connection Back to C2**: The compromised system connects to a port on the attacker's command-and-control (C2) server.
  - **Attacker Gains Access**: Once connected, the attacker can issue commands on the compromised system, much like with a bind shell.

- Security Implications: Reverse shells are **harder to detect and block than bind shells**, as they exploit outbound connections that are often allowed through firewalls. Network monitoring for unusual outbound connections and IDS rules can help detect reverse shell activity.

Comparing Bind Shells and Reverse Shells

Aspect	Bind Shell	Reverse Shell
Connection	**Target system waits** for an incoming connection	**Target system initiates** a connection to attacker
Firewall Evasion	More easily blocked by firewalls, as it requires an open port	Easier to bypass firewalls, as **outbound connections are often allowed**
Usage	Commonly used in simpler setups; easy to detect	Preferred **for bypassing firewall** restrictions

Summary

- **Remote Code Execution (RCE) enables attackers to execute arbitrary commands remotely**, with the impact dependent on the privileges gained. RCE exploits can grant either user-level or administrative access.
- Bind Shells create a listening service on the target machine, waiting for the attacker to connect, which **can be blocked by firewalls** but provides direct access once established.
- Reverse Shells **initiate a connection from the target** to the attacker's C2 server, allowing attackers to bypass firewall restrictions on incoming connections.

Remote control techniques are powerful tools in an attacker's arsenal, enabling continued control over compromised systems. To defend against these methods, organizations should prioritize **patching** vulnerabilities that allow RCE, use **firewalls** to block unusual connections, and **monitor** network traffic for signs of unauthorized connections, especially to external C2 servers.

Spoofing

Spoofing is a technique where **attackers disguise themselves as trusted sources by faking certain information**, such as an email address, IP address, MAC address, or even biometric data. This tactic is often used to trick users, bypass security measures, or initiate attacks within networks. Here's an overview of various spoofing methods, including email spoofing, IP address spoofing, MAC spoofing, biometric spoofing, and ARP spoofing.

1. Email Spoofing

- Definition: Email spoofing is when attackers **forge the sender's email address** to make their message appear to come from a trusted source.
- Purpose: Commonly used in **phishing attacks**, email spoofing is intended to trick recipients into believing they're receiving an email from a legitimate source, such as a coworker, bank, or trusted service provider.
- Methods:
    - Simple Header Manipulation: Attackers modify the "From" field in the email header.
    - Domain Spoofing: Attackers use a domain that looks similar to the legitimate one (e.g., "@micros0ft.com" instead of "@microsoft.com").
- Security Implications: Email spoofing can lead to phishing, malware distribution, and credential theft, as recipients are more likely to trust and engage with spoofed emails.

2. IP Address Spoofing

- Definition: IP address spoofing is when attackers falsify the source IP address in data packets, making it appear as though the packets are coming from a trusted source.
- Purpose: Often used in **network attacks**, IP spoofing allows attackers to bypass access control lists, launch Distributed Denial of Service (DDoS) attacks, or **evade IP-based security measures**.
- Common Uses:
    - **DDoS Attacks**: Attackers use spoofed IP addresses to flood a target with traffic, **making it difficult to trace the origin** of the attack.
    - **Session Hijacking**: By spoofing a trusted IP address, attackers can attempt to intercept or inject traffic into a trusted session.
- Security Implications: IP spoofing can **lead to significant network disruptions**, as it allows attackers to mask their identity and bypass IP-based security restrictions.

3. MAC Spoofing

- Definition: MAC spoofing involves changing the Media Access Control (MAC) address of a device to **impersonate another device on the same network**.
- Purpose: MAC spoofing can bypass network filters, gain unauthorized access to restricted networks, or avoid detection by security tools that rely on MAC addresses for device identification.
- Common Uses:
    - **Network Access**: Attackers use MAC spoofing to access restricted networks by impersonating a device that has authorized access.
    - **Evasion**: Attackers change their MAC address to avoid being tracked or detected.

- Security Implications: MAC spoofing can lead to unauthorized network access, making it difficult to identify and block the attacker's device.

4. Biometric Spoofing

- Definition: Biometric spoofing is when **attackers use fake biometric data, such as fingerprints, facial recognition, or iris patterns**, to bypass biometric authentication systems.
- Purpose: This technique is used to gain unauthorized access to systems protected by biometric authentication.
- Methods:
    - **Fingerprint** Spoofing: Creating fake fingerprints using materials like **gelatin or silicon**.
    - **Facial** Recognition Bypass: Using **photos, videos, or 3D models** to trick facial recognition systems.
- Security Implications: Biometric spoofing can bypass otherwise strong security measures, giving attackers access to sensitive systems and data.

5. ARP Spoofing

- Definition: Address Resolution Protocol (ARP) spoofing, also known as **ARP poisoning**, is a technique where attackers **send fake ARP messages to link their MAC address to the IP address of another device on the same network.**
- Purpose: ARP spoofing is commonly used for **man-in-the-middle (MitM) attacks**, where attackers intercept or modify traffic between two devices.
- How It Works:
    - The attacker sends forged ARP responses, associating their MAC address with the IP address of a legitimate device (such as a router).
    - Devices on the network believe the attacker's device is the legitimate device, allowing the attacker to intercept, alter, or reroute traffic.
- Security Implications: ARP spoofing **compromises network integrity**, leading to **data interception, session hijacking, and potentially malware injection**.

Summary

- **Email Spoofing** tricks recipients by disguising the sender's email, leading to phishing and fraud.
- **IP Address Spoofing** allows attackers to mask their origin, evade security controls, and launch DDoS attacks.
- **MAC Spoofing** enables unauthorized network access by changing the device's MAC address to impersonate trusted devices.
- **Biometric Spoofing** bypasses biometric security systems using fake physical characteristics.
- **ARP Spoofing** compromises local network security by rerouting traffic to the attacker's device, enabling MitM attacks.

Each of these spoofing methods allows attackers to gain unauthorized access, evade detection, or manipulate traffic, making spoofing a critical threat. Defending against these techniques **requires multi-layered security measures, including network monitoring, multi-factor authentication, ARP inspection, and regular security awareness training**.

Tools

Here's an overview of some key tools used in cybersecurity for reconnaissance, exploitation, and device scanning, including Metasploit, ExploitDB, Shodan, Google for version-based exploits, and Hak5 tools. These tools are commonly used by both security professionals and attackers for identifying vulnerabilities, testing security defenses, and gathering information on networked devices.

1. Metasploit

- Definition: Metasploit is **an open-source penetration testing framework that allows users to find, exploit, and validate vulnerabilities within systems**. It's widely used by penetration testers, ethical hackers, and security researchers.
- Features:
    - Exploit Modules: Metasploit includes thousands of pre-built exploits that target specific vulnerabilities in software and systems.
    - Payloads: After exploiting a vulnerability, Metasploit can deliver payloads (such as shells) that allow for remote control.
    - Post-Exploitation Tools: These tools help testers perform actions like privilege escalation, persistence, and data exfiltration.
- Use Cases: Metasploit is commonly used to **test system defenses, simulate attacks, and validate the effectiveness of security measures**. Attackers might use it to exploit vulnerabilities on unpatched systems.
- Security Implications: Metasploit provides extensive testing capabilities, but unauthorized use can lead to system compromise, data theft, and network disruption.

2. ExploitDB

- Definition: ExploitDB (Exploit Database) is **an online repository of publicly disclosed exploits and vulnerabilities**. It includes scripts, code, and detailed information about vulnerabilities.
- Features:
    - **Exploit Code**: Ready-to-use scripts and code that target specific vulnerabilities.
    - **Detailed Vulnerability Descriptions**: Each entry includes information about the affected software, vulnerability details, and steps for exploitation.
    - **Searchable Database**: Users can search by software, CVE ID, or vulnerability type to find relevant exploits.
- Use Cases: ExploitDB is a go-to resource for penetration testers and researchers who need exploit code for specific vulnerabilities. Attackers may also use it to find exploits for unpatched systems.
- Security Implications: ExploitDB is valuable for understanding how vulnerabilities work, but attackers can use its scripts to target vulnerable systems if proper defenses are not in place.

3. Shodan

- Definition: Shodan is **a search engine for internet-connected devices**, allowing users to find servers, routers, webcams, IoT devices, and more, based on their IP addresses and exposed services.
- Features:
    - **Device Search**: Shodan indexes devices connected to the internet, showing open ports, protocols, and available services.

    - **Vulnerability Information**: Shodan flags devices that are exposed to known vulnerabilities, especially if they're running outdated software versions.
    - **Geolocation and Device Metadata**: Users can see device locations, banners, and metadata.
- Use Cases: Security professionals use Shodan for **reconnaissance**, finding devices on their network, and identifying potential exposure points. Attackers use it to locate exposed, vulnerable devices to target for exploitation.
- Security Implications: Shodan makes it easy to discover vulnerable devices, highlighting the importance of securing internet-exposed systems with strong configurations and regular patching.

4. Google for Version-Based Exploits

- Definition: Googling the version number of software or hardware alongside keywords like "exploit" or "vulnerability" is a simple yet effective way to discover known vulnerabilities.
- How It Works:
    - Users search for specific software versions (e.g., "Apache 2.4.49 exploit") to find information on vulnerabilities and available exploits.
    - This technique can reveal CVEs, forum discussions, exploit scripts, and mitigation strategies.
    - Use Cases: This method is used by security researchers for quick information gathering on potential vulnerabilities associated with specific software. Attackers may use it to identify unpatched systems they can target.
- Security Implications: With many vulnerabilities documented online, it's easy to find exploits for unpatched versions of software, stressing the importance of keeping software updated and monitoring for known issues.

5. Hak5 Tools

- Definition: Hak5 is **a company that produces hardware and software tools for penetration testing and security research**, popular for their ease of use and specialized capabilities.
- Popular Hak5 Tools:
    - **WiFi Pineapple**: A tool for testing Wi-Fi security, capable of performing man-in-the-middle attacks, network monitoring, and more.
    - **USB Rubber Ducky**: A USB device that functions as a keystroke injector, executing preloaded commands or payloads on a target device.
    - **LAN Turtle**: A covert network implant that allows for remote access and network traffic monitoring.
- Use Cases: Hak5 tools are designed for security professionals conducting penetration tests, network assessments, and Wi-Fi audits. Attackers, however, can use these tools for unauthorized access, surveillance, and data theft.
- Security Implications: Hak5 tools make network assessment easy but also accessible to malicious actors. Security teams should be aware of these tools and monitor for behaviors or devices consistent with their use.

Summary

- **Metasploit**: **A penetration testing framework** that provides a large library of exploits and payloads for testing system defenses.
- **ExploitDB**: **An online database** of public exploits and vulnerability details, useful for understanding vulnerabilities and finding exploit code.

- **Shodan**: **A search engine** for internet-connected devices, making it easy to locate vulnerable and exposed systems.
- **Google for Version-Based Exploits**: **Searching** for vulnerabilities by software version helps identify known exploits and weaknesses for specific versions.
- **Hak5 Tools**: **Specialized hardware and software tools** for penetration testing, such as WiFi Pineapple and USB Rubber Ducky, which can also be misused by attackers.

Each of these tools provides valuable resources for security testing and research, but they also come with risks if used maliciously. **Organizations can mitigate these risks by securing devices, patching known vulnerabilities, and monitoring for signs of unauthorized tools or suspicious network activity**.