Well Known Cyber Attacks Explanation:

Phishing Attacks: Phishing attacks involve sending fraudulent emails that appear to be from reputable sources. The goal is to trick recipients into revealing sensitive information or downloading malware. To mitigate phishing attacks, individuals and organizations can educate themselves about phishing techniques, use email filters to block suspicious emails, and implement multi-factor authentication.

Malware Attacks: Malware refers to malicious software designed to disrupt, damage, or gain unauthorized access to computer systems. It can be delivered through email attachments, infected websites, or removable media. Mitigation strategies for malware attacks include using up-to-date antivirus software, regularly patching software vulnerabilities, and practicing safe browsing habits

Ransomware Attacks: Ransomware is a type of malware that encrypts a victim's files and demands a ransom in exchange for the decryption key. It can spread through malicious email attachments, compromised websites, or vulnerable software. To mitigate ransomware attacks, organizations should regularly back up their data, keep software up to date, and educate employees about safe online practices.

Distributed Denial of Service (DDoS) Attacks: DDoS attacks aim to overwhelm a target system or network with a flood of traffic, rendering it inaccessible to legitimate users. Attackers often use botnets to launch these attacks. Mitigation strategies for DDoS attacks include implementing traffic filtering, load balancing, and using DDoS mitigation services.

Insider Threats: Insider threats involve individuals within an organization who misuse their access privileges to steal or compromise sensitive data. These threats can be intentional or unintentional. Mitigation measures include implementing access controls, monitoring user activities, and providing cybersecurity training to employees

Social Engineering Attacks: Social engineering attacks exploit human psychology to manipulate individuals into revealing sensitive information or performing certain actions. Common techniques include pretexting, phishing, and baiting. To mitigate social engineering attacks, individuals should be cautious about sharing personal information and should be trained to recognize and report suspicious activities.

Man-in-the-Middle (MitM) Attacks: MitM attacks involve intercepting and altering communication between two parties without their knowledge. Attackers can eavesdrop on sensitive information or inject malicious content. To mitigate MitM attacks, individuals and organizations should use secure communication protocols (e.g., HTTPS), verify digital certificates, and use virtual private networks (VPNs) when accessing public networks.

SQL Injection Attacks: SQL injection attacks exploit vulnerabilities in web applications that do not properly validate user input. Attackers inject malicious SQL code to manipulate the application's database and gain unauthorized access or extract sensitive information. Mitigation measures include using prepared statements or parameterized queries, input validation, and regularly updating and patching web application frameworks.

Zero-Day Attacks: Zero-day attacks target vulnerabilities in software that are unknown to the software vendor and have no available patch. Attackers exploit these vulnerabilities before they can be fixed, potentially causing significant damage. To mitigate zero-day attacks, organizations should regularly update and patch software, use intrusion detection systems, and employ behavior-based anomaly detection.

Password Attacks: Password attacks involve attempting to discover or guess passwords to gain unauthorized access. Techniques used include brute-forcing, dictionary attacks, and phishing. Mitigation strategies include enforcing strong password policies, implementing multi-factor authentication, and educating users on password security best practices.

Eavesdropping Attacks: Eavesdropping attacks involve intercepting and listening to communication between two parties. Attackers can capture sensitive information, such as usernames, passwords, or confidential data. To mitigate eavesdropping attacks, encryption techniques like secure socket layer (SSL) or transport layer security (TLS) should be used to protect data in transit. Additionally, using secure and trusted communication channels is essential.

Cross-Site Scripting (XSS) Attacks: XSS attacks exploit vulnerabilities in web applications to inject malicious scripts into web pages viewed by other users. These scripts can steal sensitive information or perform unauthorized actions on behalf of the victim. Mitigation measures include input validation and output encoding, using security frameworks, and keeping web application software up to date.

Advanced Persistent Threats (APTs): APTs are highly targeted and sophisticated attacks that involve persistent unauthorized access to a network or system over an extended period. Attackers may use various techniques such as social engineering, zero-day exploits, and lateral movement within the network. Mitigation strategies include network segmentation, regular security audits, continuous monitoring, and threat intelligence sharing.

Password Hashes: Windows systems store user passwords in the form of hashed values, rather than plain text. The hash is a fixed-length string generated by applying a cryptographic algorithm to the original password.

Credential Theft: In a Pass the Hash attack, an attacker gains access to the password hashes stored in the Windows system's Local Security Authority Subsystem Service (LSASS) memory or the Active Directory database. This can be achieved through various means, such as using malware, exploiting vulnerabilities, or compromising privileged accounts.

Hash Reuse: Instead of cracking the password hash to obtain the actual password, the attacker uses the stolen hash directly. This allows them to impersonate the legitimate user without needing to know the original password.

Privilege Escalation: Once the attacker has obtained the necessary hash, they can use it to authenticate themselves as the compromised user. This can provide them with elevated privileges and access to sensitive resources and systems within the network.

Directory Traversal attack: A Directory Traversal attack, also known as a Path Traversal attack, is a type of HTTP exploit in which a threat actor uses the software on a web server to access data in a directory other than the server's root directory. If successful, the attacker can view restricted files or execute commands on the server. This attack involves manipulating variables that reference file paths within web applications, allowing the attacker to move upwards in the directory structure or traverse to different directories. By using specific sequences (like **../** or **..**) in Unix and Windows systems, respectively, the attacker can access files and directories stored outside the intended folder, including critical system files, application source code, and configuration files. The goal of a Directory Traversal attack is to access files and directories that the attacker should not have access to, potentially leading to the compromise of the server or web application.

File inclusion attack: A File Inclusion attack, also known as a Remote File Inclusion (RFI) or Local File Inclusion (LFI) attack, is a type of vulnerability found in web applications that rely on a scripting runtime. This attack occurs when an application includes or loads files based on user-supplied input without proper validation or sanitization. In an RFI attack, the attacker exploits the web application's dynamic file inclusion functionality to upload and execute malicious external files or scripts from a remote location. By manipulating user input, such as URLs or parameter values, the attacker can trick the application into including and executing their malicious code.On the other hand, an LFI attack involves exploiting insecure local file upload functions. When user-supplied input is not properly validated, attackers can use malicious characters to upload and execute a directory traversal exploit. This allows them to access and execute files on the web server, potentially gaining unauthorized access to sensitive information or compromising the server's security. The consequences of a successful file inclusion attack can be severe, including unauthorized access, data theft, server compromise, or even complete site takeover. To prevent file inclusion attacks, it is crucial to implement proper input validation, sanitization, and secure coding practices. Whitelisting and maintaining an allowed list of files that can be included can also help mitigate the risk.

Overflow attack: A Buffer Overflow attack occurs when a program or application writes more data to a buffer (a temporary storage area) than it can handle, causing the excess data to overflow into adjacent memory locations. This overflow can overwrite important data or even executable code, leading to unpredictable behavior and potential security vulnerabilities. Attackers exploit buffer overflow vulnerabilities by intentionally sending excessive data to a program's buffer, taking advantage of the program's failure to properly validate or limit the input. By overflowing the buffer, attackers can overwrite critical data, manipulate program

execution, or inject malicious code into the system. There are two main types of buffer overflow attacks: stack-based and heap-based. In a stack-based buffer overflow, the attacker overflows the buffer located in the stack memory, which is used to store local variables and function call information. In a heap-based buffer overflow, the attacker targets the buffer located in the heap memory, which is used for dynamic memory allocation.

Cookies & Attachment attack:

Cookies Stealing Attack:

Cookie stealing, also known as session hijacking, is a type of cyber-attack where hackers gain unauthorized access to a user's account or sensitive information by stealing their session cookies.

Attackers can use Trojans, which are a type of malware, to gain access to a user's computer and steal cookies and other sensitive data.

Cookie stealing attacks can occur through various methods, including email attachments or infected downloads.

Cookie Poisoning Attack:

Cookie poisoning refers to various cyber-attacks that aim to manipulate or forge HTTP cookies, potentially leading to session hijacking, exposure of sensitive information, or taking over a victim's account.

In a narrow sense, cookie poisoning involves directly modifying existing cookie values, which can be done manually or through JavaScript.

Attackers may exploit vulnerabilities like cross-site scripting (XSS) to access and manipulate cookie data.

Cookie Hijacking:

Cookie hijacking, also known as session hijacking, occurs when an attacker intercepts or steals a user's session cookie to gain unauthorized access to their account or sensitive information on a web application.

Once an attacker obtains a valid session cookie, they can impersonate the user and perform actions on their behalf without consent.

Cookie hijacking can be accomplished through various methods, such as packet sniffing, man-in-the-middle attacks, or exploiting vulnerabilities like cross-site scripting (XSS) or buffer overflow.

Attachment Attacks:

Attachment attacks involve malicious actions related to email attachments or file downloads.

Attackers may send emails with unencrypted attachments to the wrong recipients, potentially exposing sensitive information.

Users may unknowingly download malware or execute malicious code by opening email attachments or visiting infected websites.

Session hijacking attack: Session hijacking, also known as session hijack or cookie hijacking, is a type of cyber-attack where an attacker gains unauthorized access to a user's session on a web application or website. The attacker aims to take control of the user's session by stealing or manipulating the session identifier, often in the form of a session cookie. During a session hijacking attack, the attacker intercepts or steals the session identifier, which is used to authenticate and maintain the user's session on the server. By obtaining the session identifier, the attacker can impersonate the user and gain access to their account, sensitive information, or perform actions on their behalf.

There are various methods used in session hijacking attacks, including:

Packet Sniffing: Attackers use network sniffing tools to capture and analyze network traffic, intercepting the session identifier sent between the user and the server.

Man-in-the-Middle (MitM) Attacks: In a MitM attack, the attacker positions themselves between the user and the server, intercepting and manipulating the session identifier exchanged during the communication.

Session Side jacking: This attack involves stealing the session identifier from an insecure or unencrypted network connection, such as public Wi-Fi, where the attacker can eavesdrop on the user's traffic.

Race Conditions attack: A race condition attack, also known as a Time of Check to Time of Use (TOCTTOU) attack, takes advantage of the need for computing systems to execute tasks in a specific sequence. In any such sequence, there is a small period of time when the system has carried out the first task but has not started on the second. Attackers exploit this time gap to interfere with processes and gain unauthorized access to secure areas or content.

For example, consider a security system that checks a user's username and password against a database before allowing access. During the authentication process, there is a small window of time between the system receiving the credentials and checking them against the database. An attacker can manipulate the system during this time gap to bypass the authentication and gain unauthorized access. Race condition attacks can have serious consequences, such as unauthorized access to sensitive data, privilege escalation, or the ability to execute malicious code on a system.

Memory Vulnerabilities Attack:

Memory vulnerabilities refer to security weaknesses that arise from improper handling or manipulation of memory in software applications. These vulnerabilities can be exploited by

attackers to gain unauthorized access, execute arbitrary code, or cause system crashes. Two common types of memory vulnerabilities are buffer overflows and memory corruption.

Buffer Overflows: A buffer overflow occurs when a program writes data beyond the boundaries of a buffer, overwriting adjacent memory locations. This can lead to the corruption of critical data structures, execution of malicious code, or system crashes. Buffer overflows often occur due to programming errors, such as improper input validation or incorrect memory allocation.

Memory Corruption: Memory corruption vulnerabilities encompass a range of issues, including buffer overflows, dangling pointers, and other scenarios where memory is incorrectly allocated or accessed. These vulnerabilities can be exploited by attackers to manipulate the execution flow of a program, modify data, or gain unauthorized access to sensitive information. Memory vulnerabilities are a significant concern in software security. In fact, a Microsoft security engineer reported that approximately 70% of all security vulnerabilities are caused by memory safety issues.

These vulnerabilities can have severe consequences, including unauthorized access to data, system crashes, and the execution of malicious code.

Code Execution attack: A code execution attack refers to the ability of an attacker to run malicious code on a target system or application. This type of attack allows the attacker to gain unauthorized access, manipulate the behavior of the system, or perform actions beyond the intended functionality of the application. Code execution attacks can have severe consequences, including data theft, system compromise, service disruption, and the deployment of additional malware. The attacker can exploit vulnerabilities in software or its environment to execute arbitrary code or commands, taking control of the targeted system or application. One common example of a code execution attack is through injection vulnerabilities, such as SQL injection or command injection. These vulnerabilities occur when an application fails to properly validate and sanitize user input before using it in a command or query. An attacker can inject malicious code or commands into the input, which is then executed by the application, leading to unauthorized access or manipulation of data. Another example is remote code execution (RCE), where an attacker can remotely execute malicious code on a target system or application. RCE vulnerabilities can be particularly dangerous as they can be exploited even if the attacker has no prior access to the system. The impact of an RCE vulnerability can range from malware execution to complete compromise of the affected system.

Data Poisoning attack: Data poisoning is a type of attack that involves the deliberate and malicious contamination of data used to train machine learning (ML) and artificial intelligence (AI) systems. The goal of a data poisoning attack is to compromise the performance and integrity of the ML or AI model by introducing corrupted or misleading data during the training phase.

Unlike other adversarial techniques that target the model during inference, data poisoning attacks specifically target the training data. By injecting manipulated or malicious data into the

training dataset, attackers aim to influence the learning process and manipulate the behavior of the ML or AI model.

The impact of data poisoning attacks can be significant. They can lead to incorrect predictions, biased outcomes, or even complete compromise of the ML or AI system. For example, an attacker could manipulate training data used for email filtering, causing the system to misclassify spam emails as legitimate or vice versa.

Third Party Code attack:

Interception Proxies attacks: Interception proxies, also known as intercepting proxies, are tools used to analyze, modify, and intercept network traffic between a client and a server. These proxies sit between the client and the server, allowing them to intercept and inspect the communication passing through. Interception proxies can be used for various purposes, including security testing, traffic analysis, and debugging. However, they can also be exploited for malicious purposes, leading to interception proxy attacks. Interception proxy attacks involve unauthorized access, monitoring, or manipulation of the intercepted communication. Attackers can exploit interception proxies to gain access to sensitive information, modify data in transit, or impersonate one or both parties involved in the communication. Some common interception proxy attacks include:

1. **Man-in-the-Middle (MitM) Attacks**: In a MitM attack, the attacker positions themselves between the client and the server, intercepting and potentially modifying the communication. This allows the attacker to eavesdrop on sensitive information, such as passwords or financial data, or even impersonate one or both parties involved.
2. **SSL/TLS Stripping**: SSL/TLS stripping is an interception attack that targets the HTTPS communication protocol. The attacker tricks users into connecting to insecure HTTP websites instead of the intended secure HTTPS sites. This allows the attacker to intercept and manipulate the traffic, potentially compromising the confidentiality and integrity of the communication.
3. **Evil Twin Attacks**: In evil twin attacks, attackers set up rogue Wi-Fi access points that mimic legitimate networks. Unsuspecting users connect to these rogue access points, allowing the attackers to intercept and manipulate their network traffic.

Industrial Control Systems attacks: Industrial Control Systems (ICS) attacks are cyber threats targeted at the systems that manage and control industrial processes. These systems operate some of the nation's most critical infrastructures, including energy transmission and distribution plants, oil refineries, wastewater treatment facilities, transportation systems, and more1.

The impacts from these attacks can range from disruption to operational productivity to serious harm to human life and the surrounding environment1. For instance, adversaries may try to

interrupt critical service delivery by disrupting industrial processes or cause physical damage to equipment.

Here are some examples of ICS attacks:

The Ukrainian grid experienced cyberattacks that shut down power over short periods in 2015 and 2016.

The Colonial Pipeline in the US was targeted by a ransomware attack in May 2021, causing an acute fuel shortage.

CPC Corp. Taiwan, a state-owned petroleum and natural gas company, saw its payment system crippled by a ransomware attack in May 2020.

A highly sophisticated malware attack, known as **Triton**, was intended to target ICS and cause physical damage to critical infrastructure.

IoT attacks: Internet of Things (IoT) attacks are cyber threats targeted at internet-connected devices, such as smart home devices, industrial control systems, and medical devices1. Attackers may gain control of the device, steal sensitive data, or use the device as a part of a botnet for other malicious purposes1. IoT devices are particularly vulnerable to network attacks such as data thefts, phishing attacks, spoofing, and denial of service attacks (DDoS attacks).

Here are some examples of IoT attacks:

A Russian hacker group known as 'Strontium' used IoT connected devices like wireless printers to bypass safety protocols and gain access to sensitive networks.

The Apache Log4j Vulnerability affected hundreds of millions of devices and allowed an unauthenticated user to control the connected system for anything from data theft to cryptomining.

Philips disclosed a vulnerability in its TASY Electronic Medical Record (EMR) HTML5 system, where a successful SQL injection attack can result in confidential patient data being exposed.

Embedded Systems attacks: Embedded Systems attacks are cyber threats targeted at systems that are designed to perform a specific function within a larger system. These systems are often found in devices such as microwave ovens, cars, and aircraft. Due to their specific functionality and the data they generate, process, and transmit, they are popular targets for hacking.

Common attacks on Embedded Systems include:

Network-based attacks: These occur through network-based vulnerabilities. Hackers exploit these vulnerabilities to carry out attacks such as signals jamming, session hijacking, DNS poisoning, and man-in-the-middle attacks.

Packet injection: This involves the introduction of malicious or misleading packets into a network to disrupt or take control of a system.

Man in the Middle (MITM) attack: This is where the attacker secretly intercepts and potentially alters the communication between two parties who believe they are directly communicating with each other.

Replay attacks: This involves the malicious or fraudulent repeat of a valid data transmission.

Exploitation Frameworks attacks:

Exploitation Frameworks attacks involve the use of software packages that contain reliable exploit modules and other useful features, such as agents used for successful repositioning. These frameworks allow attackers to use different exploit payloads and other unique options to obfuscate shellcode and network traffic in order to avoid detection. They provide a centralized repository of exploits and attack tools, saving time and effort by eliminating the need to develop custom exploits for each vulnerability.

Common exploitation frameworks include Metasploit and BeEF (Browser Exploitation Framework). Metasploit is a penetration testing framework that helps you find and exploit vulnerabilities in systems. It gives you everything you need from scanners to third-party integrations that you will need throughout an entire penetration testing lifecycle. BeEF focuses on the web browser and allows the professional penetration tester to assess the actual security posture of a target environment by using client-side attack vectors.

Cross-Site Request Forgery (CSRF) Attacks: CSRF attacks exploit the trust that a website has in a user's browser. Attackers trick users into unknowingly performing unwanted actions on a targeted website while authenticated. Implementing measures such as anti-CSRF tokens, input validation, and session management techniques can help mitigate CSRF attacks.

Advanced Evasion Techniques: Advanced evasion techniques (AETs) are used to bypass network security devices and intrusion detection systems (IDS/IPS). Attackers modify or obfuscate their malicious code to evade detection. Mitigation strategies include keeping security devices up to date, employing deep packet inspection techniques, and using behavior-based analysis to detect and block AETs.

Cryptojacking: Cryptojacking involves unauthorized use of a victim's computing resources to mine cryptocurrencies. Attackers infect systems with malware, such as browser-based JavaScript miners or malicious software, to exploit system resources for mining purposes. To

Zero-Trust Attacks: Zero-trust attacks exploit vulnerabilities in zero-trust architecture, where trust is not automatically granted to any user or device. Attackers may exploit misconfigurations, weak authentication mechanisms, or compromised credentials to gain unauthorized access. To mitigate zero-trust attacks, organizations should implement robust identity and access management.

Password Spray Attack

Password spraying is a type of brute force attack that targets multiple user accounts with a few commonly used passwords. Unlike traditional brute force attacks that attempt many passwords against a single account, password spraying attempts a few passwords against many accounts. This approach helps attackers evade detection mechanisms that lock out accounts after multiple failed login attempts. Here's how a password spray attack works:

Account Enumeration: Attackers first gather a list of valid usernames or email addresses associated with a target system or application. They may obtain this information through various means, such as reconnaissance, social engineering, or data breaches.

Password Selection: Instead of using a large number of passwords, attackers choose a small set of commonly used or easily guessable passwords. Examples include "123456," "password," or variations of "admin" or "1234."

Credential Stuffing: Attackers then systematically try these selected passwords against multiple user accounts. They often distribute their attempts over a large number of accounts to avoid triggering account lockouts or detection mechanisms.

Successful Authentication: If the attacker successfully authenticates with any of the targeted accounts, they gain unauthorized access and can further exploit the compromised account for malicious activities.

Log and Monitor Authentication Attempts: Implement logging and monitoring mechanisms to capture and analyze authentication attempts. This can help identify patterns, detect suspicious activities, and trigger appropriate response measures.

Watering Hole Attacks: Watering hole attacks target a specific group of users by infecting websites they frequently visit. Attackers compromise these legitimate websites and inject malware to exploit vulnerabilities in visitors' systems. Mitigation involves keeping systems and software up to date, using web filters, and practicing safe browsing habits.

File less Malware Attacks: File less malware attacks do not rely on traditional malicious files. Instead, they leverage legitimate system tools or scripts to carry out malicious activities directly in memory. Mitigation includes using endpoint protection solutions, monitoring system behavior, and implementing security measures to detect and block such attacks.

Physical Attacks: Physical attacks involve gaining unauthorized access to physical devices, systems, or locations. This may include theft, tampering, or unauthorized access to sensitive infrastructure. Mitigation strategies include implementing physical security measures such as access controls, surveillance systems, and security guards.

Supply Chain Attacks: Supply chain attacks target the software supply chain to compromise systems further downstream. Attackers infiltrate trusted software vendors or suppliers and inject malicious code or backdoors into the software or hardware. To mitigate supply chain attacks, organizations should conduct due diligence on vendors, perform security assessments, and maintain visibility and control over their supply chain.

IoT-Based Attacks: Internet of Things (IoT)-based attacks exploit vulnerabilities in connected devices to gain access to networks or compromise data. Insecure default settings, weak passwords, and outdated firmware are common entry points for attackers. Mitigation strategies include changing default settings and passwords, keeping devices updated with the latest security patches, and segmenting IoT devices from critical networks.

Social Media Attacks: Social media attacks involve the exploitation of social media platforms to spread malware, steal personal information, or launch phishing campaigns. Attackers may create fake profiles, send malicious links, or engage in social engineering to deceive users. Mitigation involves being cautious of unknown or suspicious links, using privacy settings effectively, and staying informed about social media security risks.

DNS Spoofing: DNS spoofing is a technique where attackers redirect DNS queries to malicious websites by tampering with DNS responses. This can lead to users unknowingly visiting fraudulent websites or falling victim to phishing attacks. To mitigate DNS spoofing, organizations should implement DNSSEC (DNS Security Extensions), monitor DNS traffic, and use DNS filtering services.