Date : 23/08/2023

## **Task 1**

**Top 10 Hackers of all time**

1. Kevin Mitnik



Mitnick gained unauthorized access to a computer network in 1979, at 16, when a friend gave him the telephone number for the Ark, the computer system that Digital Equipment Corporation (DEC) used for developing its RSTS/E operating system software. He broke into DEC's computer network and copied the company's software, a crime for which he was charged and convicted in 1988. He was sentenced to 12 months in prison followed by three years of supervised release. Near the end of his supervised release, Mitnick hacked into Pacific Bell voicemail computers. After a warrant was issued for his arrest, Mitnick fled, becoming a fugitive for two-and-a-half years

1. Anonymous



Anonymous got its start in 2003 on 4chan message boards in an unnamed forum. The group exhibits little organization and is loosely focused on the concept of social justice. For example, in 2008 the group took issue with the Church of Scientology and begin disabling their websites, thus negatively impacting their search rankings in Google and overwhelming its fax machines with all-black images. In March 2008, a group of "Anons" marched passed Scientology centers around the world wearing the now-famous Guy Fawkes mask.

1. Adrian Lamo



In 2001, 20-year-old Adrian Lamo used an unprotected content management tool at Yahoo to modify a Reuters article and add a fake quote attributed to former Attorney General John Ashcroft. Lamo often hacked systems and then notified both the press and his victims. In some cases, he'd help clean up the mess to improve their security. As Wired points out, however, Lamo took things too far in 2002, when he hacked The New York Times' intranet, added himself to the list of expert sources and began conducting research on high-profile public figures. Lamo earned the moniker "The Homeless Hacker" because he preferred to wander the streets with little more than a backpack and often had no fixed address.

1. Albert Gonzalez



At 22, Gonzalez was arrested in New York for debit card fraud related to stealing data from millions of card accounts. To avoid jail time, he became an informant for the Secret Service, ultimately helping indict dozens of Shadowcrew members.

During his time as a paid informant, Gonzalez continued his criminal activities. Along with a group of accomplices, Gonzalez stole more than 180 million payment card accounts from companies including OfficeMax, Dave and Buster's and Boston Market. The New York Times Magazine notes that Gonzalez's 2005 attack on US retailer TJX was the first serial data breach of credit information. Using a basic SQL injection, this famous hacker and his team created back doors in several corporate networks, stealing an estimated $256 million from TJX alone. During his sentencing in 2015, the federal prosecutor called Gonzalez's human victimization "unparalleled."

1. Mathew Bevan and Richard Pryce



Matthew Bevan and Richard Pryce are a team of British hackers who hacked into multiple military networks in 1996, including Griffiss Air Force Base, the Defense Information System Agency and the Korean Atomic Research Institute (KARI). Bevan (Kuji) and Pryce (Datastream Cowboy) have been accused of nearly starting a third world war after they dumped KARI research onto American military systems. Bevan claims he was looking to prove a UFO conspiracy theory, and according to the BBC, his case bears resemblance to that of Gary McKinnon. Malicious intent or not, Bevan and Pryce demonstrated that even military networks are vulnerable.

1. Jeanson James Ancheta



Jeanson James Ancheta had no interest in hacking systems for credit card data or crashing networks to deliver social justice. Instead, Ancheta was curious about the use of bots—software-based robots that can infect and ultimately control computer systems. Using a series of large-scale "botnets," he was able to compromise more than 400,000 computers in 2005. According to Ars Technica, he then rented these machines out to advertising companies and was also paid to directly install bots or adware on specific systems. Ancheta was sentenced to 57 months in prison. This was the first time a hacker was sent to jail for the use of botnet technology.

1. Michael Calce



Jeanson James Ancheta had no interest in hacking systems for credit card data or crashing networks to deliver social justice. Instead, Ancheta was curious about the use of bots—software-based robots that can infect and ultimately control computer systems. Using a series of large-scale "botnets," he was able to compromise more than 400,000 computers in 2005. According to Ars Technica, he then rented these machines out to advertising companies and was also paid to directly install bots or adware on specific systems. Ancheta was sentenced to 57 months in prison. This was the first time a hacker was sent to jail for the use of botnet technology.

1. Kevin Poulsen



Poulsen hacked a federal computer and dug into files pertaining to the deposed president of the Philippines, Ferdinand Marcos. When discovered by authorities, Poulsen went underground. While he was on the run, Poulsen kept busy, hacking government files and revealing secrets. According to his own website, in 1990, he hacked a radio station contest and ensured that he was the 102nd caller, winning a brand new Porsche, a vacation, and $20,000.

1. Jonathan James



Using the alias cOmrade, Jonathan James hacked several companies. According to the New York Times, what really earned James attention was his hack into the computers of the United States Department of Defense. Even more impressive was the fact that James was only 15 at the time. In an interview with PC Mag, James admitted that he was partly inspired by the book The Cuckoo’s Egg, which details the hunt for a computer hacker in the 1980s. His hacking allowed him to access over 3,000 messages from government employees, usernames, passwords and other sensitive data.

1. Astra



This hacker differs from the others on this list in that he has never been publicly identified. However, according to the Daily Mail, some information has been released about ASTRA. Namely that he was apprehended by authorities in 2008, and at that time he was identified as a 58-year-old Greek mathematician. Reportedly, he had been hacking into the Dassault Group, for almost half a decade. During that time, he stole cutting edge weapons technology software and data which he then sold to 250 individuals around the world. His hacking cost the Dassault Group $360 million in damages. No one knows why his complete identity has never been revealed, but the word 'ASTRA' is a Sanskrit word for 'weapon'.

Date : 24/08/2023

## **Task 2**

**Ports and their vulnerabilities**

1. Port Number: 20

Description: Port 20 is used for the FTP data channel, which is responsible for transferring the actual data files during FTP sessions.

Protocol: TCP (Transmission Control Protocol)

Vulnerability: Since port 20 is used for data transfer in FTP, its main vulnerability would be related to security issues inherent to FTP itself. FTP, in its standard form, is not a secure protocol, as it transmits data, including usernames and passwords, in plain text. This makes it susceptible to eavesdropping and data interception. Additionally, FTP does not inherently support encryption, which further adds to its security concerns.

1. Port Number: 21

Description: Port 21 is used for the FTP control channel. This channel is responsible for sending commands from the client to the server, as well as receiving responses and status messages from the server. It sets up the parameters for the data channel (typically using port 20) for the actual file transfers.

Protocol: TCP (Transmission Control Protocol)

Vulnerability: Port 21 is associated with security vulnerabilities that are inherent to the FTP protocol itself. These vulnerabilities include:

1. Plain Text Transmission: FTP transmits usernames, passwords, and data in plain text, which makes it susceptible to eavesdropping and interception by malicious actors on the network.
2. Weak Authentication: FTP's default authentication relies on usernames and passwords, which can be easily targeted through brute-force attacks or password guessing.
3. No Encryption: FTP does not provide inherent encryption for data transmission, which means that sensitive data can be intercepted and read by attackers.
4. Data Bounce Attacks: Attackers can use FTP servers to initiate connections to other servers, potentially bypassing firewalls and gaining unauthorized access.

3. Port Number: 22

Description: Port 22 is used for SSH, a cryptographic network protocol that allows secure remote access to systems over an unsecured network. It provides a secure channel for various types of interactions, including terminal sessions, file transfers, and remote command execution.

Protocol: SSH (Secure Shell)

Vulnerability: While SSH is designed to be highly secure, there are still potential vulnerabilities that can be exploited:

1. Brute-Force Attacks: Attackers can attempt to guess passwords through brute-force attacks, where they systematically try a large number of possible passwords until they find the correct one.
2. Weak Authentication: If weak passwords are used or if SSH keys are not properly protected, unauthorized individuals could gain access.
3. Key Management: Poorly managed SSH keys can lead to unauthorized access. If private keys are compromised, attackers can impersonate legitimate users.
4. Outdated Software: Using outdated or unpatched SSH software can expose systems to known vulnerabilities that have been patched in newer versions.
5. Misconfigured Access Controls: Incorrectly configured access controls can allow unauthorized users to gain access to SSH services.

4. Port Number: 23

Description: Port 23 is used for Telnet, a protocol that provides a virtual terminal connection over a network. It enables users to remotely access and interact with a host system's command-line interface.

Protocol: Telnet

Vulnerability: Telnet is considered highly insecure due to the following vulnerabilities:

1. Plain Text Transmission: Telnet transmits all data, including usernames, passwords, and commands, in plain text. This makes it extremely susceptible to eavesdropping and interception by malicious actors on the network.
2. Lack of Encryption: Unlike modern secure protocols, Telnet does not encrypt any data, which means that sensitive information can be easily captured by attackers.
3. Authentication Vulnerabilities: Telnet's authentication mechanism is weak, typically relying on plain text passwords. This makes it susceptible to brute-force attacks and password interception.
4. Session Hijacking: Since data is sent in plain text, attackers can easily hijack Telnet sessions, injecting malicious commands or intercepting legitimate commands.
5. No Data Integrity Protection: Telnet lacks mechanisms to ensure the integrity of data. Attackers can modify data in transit without detection.

5. Port Number: 25

Description: Port 25 is the default port used for SMTP (Simple Mail Transfer Protocol). SMTP is responsible for sending email messages from one mail server to another, facilitating the transfer of messages across the internet.

Protocol: SMTP (Simple Mail Transfer Protocol)

Vulnerability: While SMTP itself isn't inherently vulnerable, there are certain security concerns associated with port 25:

1. Spam and Email Relaying: Malicious actors can exploit improperly configured mail servers to use them for sending spam or phishing emails. This is known as an open relay. Proper configuration and security measures are necessary to prevent this.
2. Email Spoofing: Attackers can forge the sender's email address, making it appear as if the email originated from a legitimate source. This can be used for phishing, scams, and other malicious activities.
3. Denial of Service (DoS) Attacks: Attackers can flood a mail server with a large volume of connection requests or emails, overwhelming the server's resources and causing a denial of service.
4. Virus Distribution: Malicious attachments or links in emails can be used to distribute malware or viruses to recipients.
5. Authentication Bypass: Misconfigured servers might allow unauthorized users to use them for sending emails without proper authentication.

6. Port Number: 53

Description: Port 53 is used for DNS (Domain Name System) services. DNS is responsible for resolving domain names to IP addresses and vice versa, enabling users to access resources using easy-to-remember names instead of numerical IP addresses.

Protocol: DNS (Domain Name System)

Vulnerability: There are several vulnerabilities and security concerns associated with DNS and port 53:

1. DNS Spoofing: Attackers can manipulate DNS responses to redirect users to malicious websites or intercept sensitive information, leading to phishing attacks or other malicious activities.
2. DNS Cache Poisoning: Attackers can inject fake DNS records into caching servers, leading legitimate users to malicious sites or services.
3. DNS Amplification Attacks: Malicious actors can exploit misconfigured DNS servers to launch Distributed Denial of Service (DDoS) attacks, amplifying their attack traffic through DNS responses.
4. DDoS Attacks on DNS: DNS infrastructure itself can be targeted in DDoS attacks, leading to service disruptions and making websites and services inaccessible.
5. Zone Transfer Exploits: Misconfigured DNS servers can expose sensitive zone transfer information, allowing attackers to gather information about the domain's structure.

7. Port Number: 69

Description: Port 69 is the default port used for the Trivial File Transfer Protocol (TFTP). TFTP is a simplified version of FTP that lacks many features but is easier to implement and use.

Protocol: TFTP (Trivial File Transfer Protocol)

Vulnerability: TFTP itself is quite minimalistic and doesn't include security features like authentication or encryption, which makes it vulnerable to certain risks:

1. Lack of Authentication: TFTP doesn't include built-in authentication mechanisms. This means that anyone with network access to the TFTP server can potentially upload or download files without restriction.
2. No Encryption: TFTP doesn't support encryption, so data transferred using TFTP is transmitted in plain text. This makes it susceptible to eavesdropping and interception by malicious actors on the network.
3. Data Integrity: TFTP doesn't provide built-in data integrity checks. If data becomes corrupted during transfer, there's no mechanism to ensure its accuracy.
4. No Access Controls: Without access controls and authentication, there's a risk that unauthorized users could overwrite or modify critical files on the TFTP server.

8. Port Number: 80

Description: Port 80 is the default port used for HTTP (Hypertext Transfer Protocol). It is the standard protocol for transferring web content from web servers to web browsers, allowing users to access websites and online resources.

Protocol: HTTP (Hypertext Transfer Protocol)

Vulnerability: There are several vulnerabilities and security concerns associated with port 80 and the HTTP protocol:

1. Cross-Site Scripting (XSS): Attackers can inject malicious scripts into web pages, which are then executed by users' browsers. This can lead to theft of sensitive information, session hijacking, or other malicious actions.
2. SQL Injection: Poorly secured web applications can be vulnerable to SQL injection attacks, where attackers manipulate input fields to execute unauthorized database queries, potentially exposing or modifying sensitive data.
3. Cross-Site Request Forgery (CSRF): Attackers trick users into performing actions they didn't intend, using their authenticated sessions to perform unauthorized actions on a website.
4. Sensitive Data Exposure: If web applications do not handle sensitive data securely, attackers might gain access to usernames, passwords, or other confidential information.
5. Server Misconfigurations: Incorrectly configured web servers or applications can expose directory listings, sensitive files, or other information that should be kept private.
6. Denial of Service (DoS) Attacks: Attackers can flood a web server with excessive requests, overwhelming its resources and causing a denial of service.

9. Port Number: 110

Description: Port 110 is the default port used for the POP3 (Post Office Protocol version 3) protocol. POP3 is a protocol for retrieving email messages from a mail server to a local client device, such as an email client on a computer or smartphone.

Protocol: POP3 (Post Office Protocol version 3)

Vulnerability: There are several vulnerabilities and security concerns associated with port 110 and the POP3 protocol:

1. Plain Text Transmission: POP3 transmits authentication credentials, email headers, and message contents in plain text, which makes it susceptible to eavesdropping and interception by malicious actors on the network.
2. Lack of Encryption: POP3 doesn't inherently support encryption, so sensitive information, including passwords and email contents, can be captured and read by attackers.
3. Credential Theft: Since authentication is based on usernames and passwords, attackers can attempt to guess passwords through brute-force attacks or password guessing.
4. Message Download: POP3 typically downloads email messages to the client device, which means that once downloaded, the messages are stored locally and not backed up on the server.

11. Port Number: 123

Description: Port 123 is the default port used for the NTP (Network Time Protocol) protocol. NTP is essential for synchronizing the clocks of computers and network devices to a common time reference.

Protocol: NTP (Network Time Protocol)

Vulnerability: While NTP itself is not inherently insecure, there are certain vulnerabilities and security concerns associated with port 123 and the NTP protocol:

1. DDoS Amplification Attacks: Attackers can exploit improperly configured NTP servers to amplify DDoS attacks. By sending small requests to open NTP servers, attackers can cause them to respond with much larger responses to a target, overwhelming the target's resources.
2. NTP Reflection Attacks: Similar to DDoS amplification attacks, NTP reflection attacks involve sending requests to NTP servers, which then respond to a victim's IP address with larger responses, causing congestion and potential service disruption.
3. Time Spoofing: If attackers gain control of a NTP server or manipulate network traffic, they can potentially spoof time information, which could lead to authentication failures, incorrect transaction records, or other time-dependent security issues.
4. Unauthorized Access: Poorly configured NTP servers could potentially allow unauthorized access to their time synchronization services, leading to potential misuse or unauthorized time adjustments.

12. Port Number: 143

Description: Port 143 is the default port used for the IMAP (Internet Message Access Protocol) protocol. IMAP is used to retrieve and manage email messages stored on a mail server, providing more advanced features than POP3.

Protocol: IMAP (Internet Message Access Protocol)

Vulnerability: There are several vulnerabilities and security concerns associated with port 143 and the IMAP protocol:

1. Plain Text Transmission: IMAP transmits authentication credentials and email content in plain text, making it susceptible to eavesdropping and interception by malicious actors on the network.
2. Lack of Encryption: If used without encryption, IMAP traffic can expose sensitive information, including usernames, passwords, and email contents.
3. Brute-Force Attacks: Attackers can attempt to guess passwords through brute-force attacks or password guessing.
4. Email Hijacking: If an attacker gains access to an email account via compromised credentials, they can potentially access, modify, or delete email messages.
5. Man-in-the-Middle Attacks: Attackers can intercept unencrypted IMAP traffic, potentially altering the communication between the client and the server.

13. Port Number: 443

Description: Port 443 is the default port used for the HTTPS (Hypertext Transfer Protocol Secure) protocol. It's used for secure communication between web browsers and web servers, ensuring that data transmitted is encrypted and confidential.

Protocol: HTTPS (Hypertext Transfer Protocol Secure)

Vulnerability: While HTTPS itself is designed to provide a high level of security, there can still be vulnerabilities associated with port 443 and the HTTPS protocol:

1. SSL/TLS Vulnerabilities: Weak or outdated SSL/TLS (Secure Sockets Layer/Transport Layer Security) versions and configurations can expose systems to vulnerabilities like the Heartbleed bug or POODLE attack.
2. Certificate Issues: Improperly configured or expired SSL/TLS certificates can lead to security warnings, potentially allowing attackers to perform man-in-the-middle attacks.
3. Mixed Content: Loading insecure (HTTP) content on an HTTPS page can introduce security risks, as attackers might be able to manipulate the insecure content.
4. Phishing: Attackers can create fake websites with valid SSL/TLS certificates to trick users into sharing sensitive information.
5. Insecure Cipher Suites: Weak encryption algorithms and cipher suites can compromise the security of the HTTPS connection.

Date : 25/08/2023

## **Task 3**

**The top 10 OWASP vulnerabilities in 2021:**

1. Injection

CWE : CWE-20, Improper Input Validation

Description : The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly.

1. Broken Authentication

CWE : CWE-287, Improper Authentication

Description : When an actor claims to have a given identity, the product does not prove or insufficiently proves that the claim is correct.

1. Sensitive Data Exposure

CWE : CWE-200, Exposure of Sensitive Information to an Unauthorized Actor  
Description : The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.

1. XML External Entities (XXE)

CWE : CWE-611, Improper Restriction of XML External Entity Reference

Description : The product processes an XML document that can contain XML entities with URIs that resolve to documents outside of the intended sphere of control, causing the product to embed incorrect documents into its output.

1. Broken Access control

CWE : CWE-284, Improper Access Control

Description : The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor.

1. Security misconfigurations

CWE : CWE-264, Improper Restriction of Functionality

Description : The product does not properly restrict the functionality that is available to users.

1. Cross-Site Scripting (XSS)

CWE : CWE-79, Improper Neutralization of Input During Web Page Generation

Description : The product does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users.

1. Insecure Deserialization

CWE : CWE-502, Deserialization of Untrusted Data

Description : The product deserializes untrusted data without sufficiently verifying that the resulting data will be valid.

1. Using Components with known vulnerabilities

CWE : CWE-937, Using Components with Known Vulnerabilities

Description : The product is developed using an outdated version of the component, or when the component is not properly patched.

1. Insufficient logging and monitoring.

CWE : CWE-778, Insufficient Logging and Monitoring

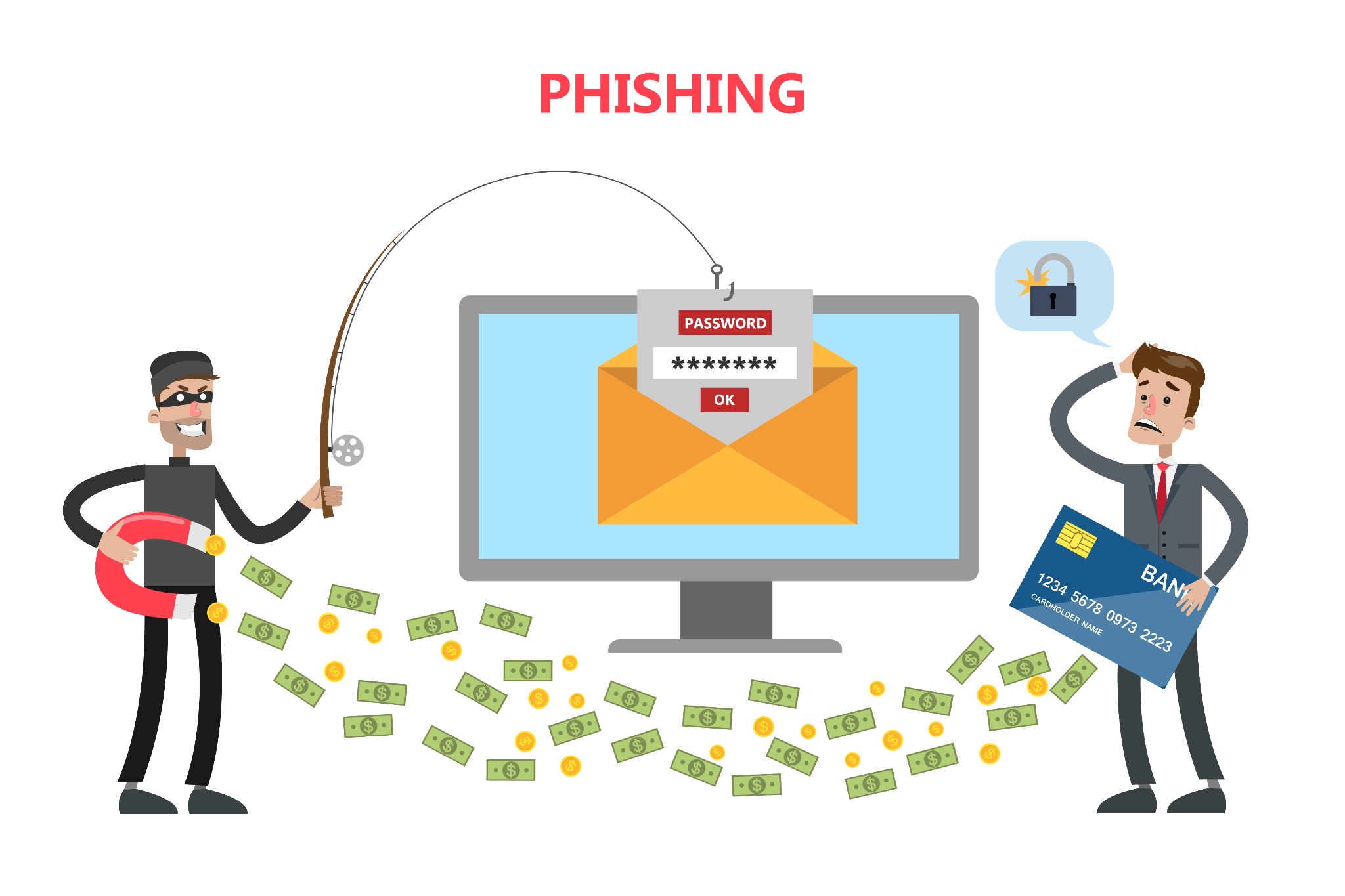
Description : When a security-critical event occurs, the product either does not record the event or omits important details about the event when logging it.

Date : 28/08/2023

## **Task 4**

**Non Top-10 web application vulnerabilities**

1. Phishing attack



This type of attack sends fraudulent emails or text messages that appear to be from a legitimate source. The emails or text messages often contain a link that, when clicked, takes the victim to a fake website that looks like the real website. Once the victim enters their login credentials on the fake website, the attacker can steal them.

1. Zero-day attack



This type of attack exploits a vulnerability in software that the software vendor is not aware of. Zero-day attacks are often very difficult to defend against because there is no patch available to fix the vulnerability.

DNS spoofing: This type of attack redirects traffic from a legitimate website to a fake website. This can be used to steal login credentials or other sensitive information.

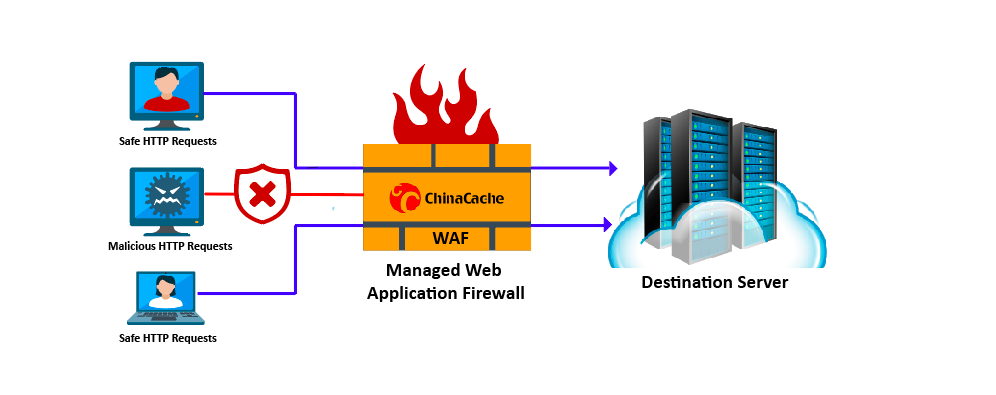
1. Botnet attack



This type of attack uses a network of compromised computers, called a botnet, to carry out malicious tasks. Botnet attacks can be used to send spam, launch DDoS attacks, or spread malware.

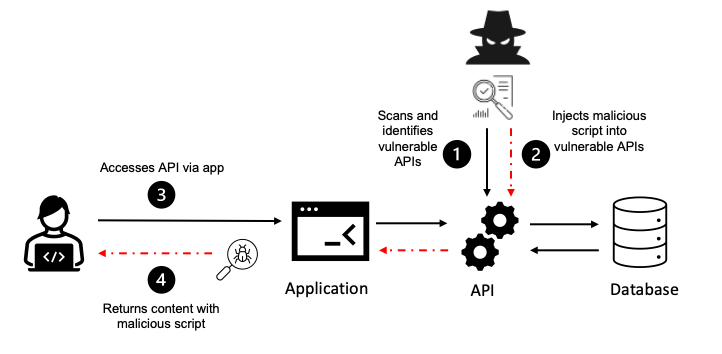
Web defacement attack: This type of attack changes the content of a website. This can be done to damage the reputation of the website or to spread propaganda.

1. Web application firewall (WAF) bypass attack



This type of attack exploits vulnerabilities in a WAF to bypass its security controls.

1. Insecure coding practices attack



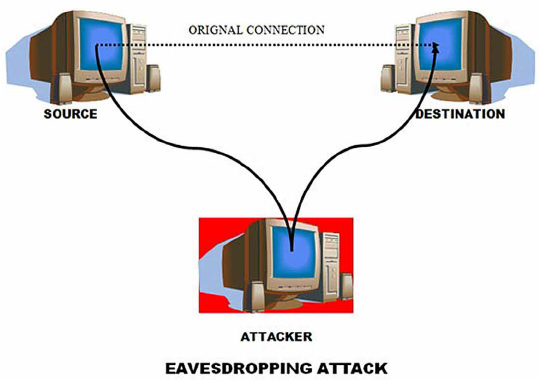
This type of attack exploits insecure coding practices in the web application code. These insecure coding practices can allow attackers to gain unauthorized access to the web application or to its data.

1. Drive-by download attack



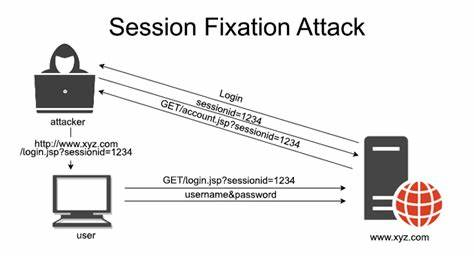
This type of attack downloads malware to the victim's computer when they visit a compromised website.

1. Eavesdropping attack



This type of attack intercepts communication between two parties. This can be done to steal sensitive information, such as passwords or credit card numbers.

1. Session fixation attack



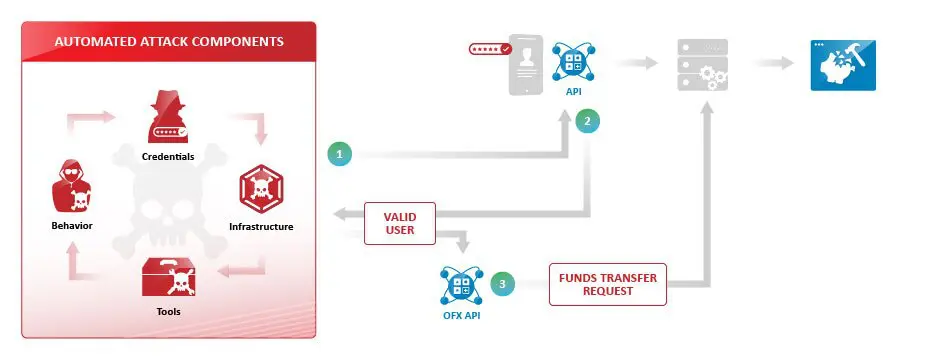
This type of attack forces the victim to reuse their session ID, allowing the attacker to impersonate that user.

1. Zero-click attack



This type of attack does not require the victim to take any action. The attack can be carried out simply by visiting a compromised website or opening an infected attachment.

1. API abuse attack



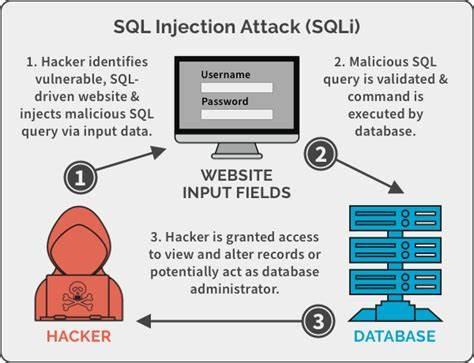
This type of attack exploits vulnerabilities in application programming interfaces (APIs). APIs are used to allow different software applications to communicate with each other.

Date : 29/08/2023

## **Task 5**

**Web server attacks**

1. SQL Injection



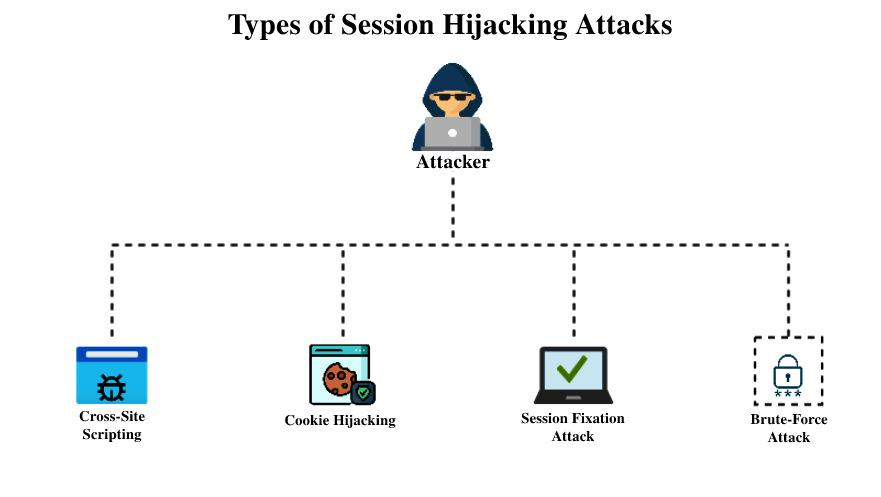
SQL injection is a type of code injection attack where malicious code is injected into a SQL query to manipulate the query and gain unauthorized access to a database. This can be done by exploiting vulnerabilities in the web application's input validation.

To put it simply, SQL injection is a way to trick a web application into executing unintended SQL commands. This can be done by inserting malicious code into a form field or URL parameter. Once the malicious code is executed, the attacker can gain access to sensitive data, such as passwords, credit card numbers, or other personal information.

SQL injection attacks are a serious threat to web applications, and it is important to take steps to prevent them. Some of the most common ways to prevent SQL injection attacks include:

1. Using prepared statements
2. Escaping all user input
3. Validating all user input
4. Using a web application firewall (WAF)

2) Session Hijacking

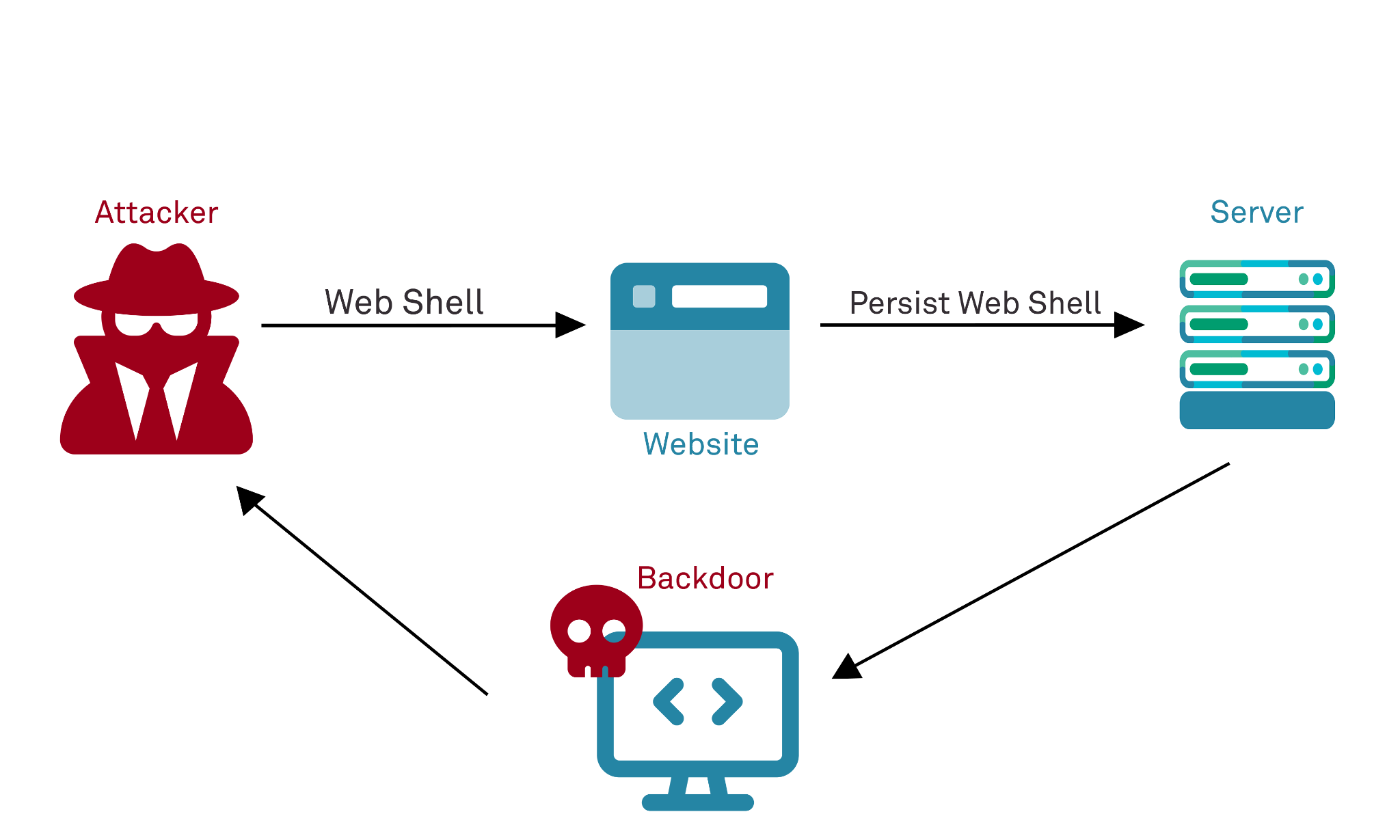


Session hijacking is a type of attack where the attacker steals the victim's session token. This token is used to authenticate the victim to the web application. Once the attacker has the token, they can impersonate the victim and access their account.

To put it simply, session hijacking is a way for an attacker to take over an existing session between a user and a web application. This can be done by stealing the session token, which is a unique identifier that is used to authenticate the user. Once the attacker has the session token, they can use it to access the user's account as if they were the legitimate user.

Session hijacking attacks can be prevented by using strong session tokens and by implementing security measures to protect the tokens from being stolen.

3) File upload vulnerabilities

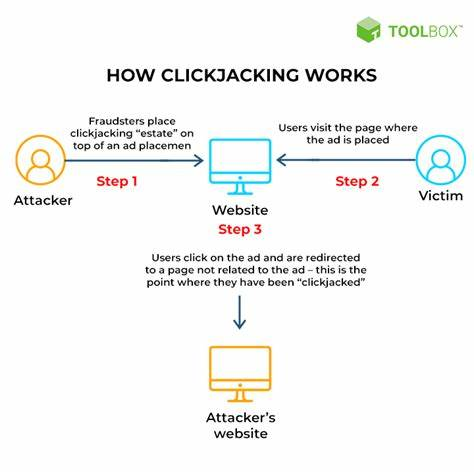


File upload vulnerabilities are when a web server allows users to upload files to its filesystem without sufficiently validating things like their name, type, contents, or size. This can be exploited by attackers to upload malicious files, such as viruses or backdoors, to the server.

To put it simply, file upload vulnerabilities are security flaws in web applications that allow attackers to upload malicious files to the server. These files can then be used to gain unauthorized access to the server, steal data, or disrupt operations.

File upload vulnerabilities can be prevented by carefully validating all uploaded files and by restricting the types of files that can be uploaded.

4) Clickjacking

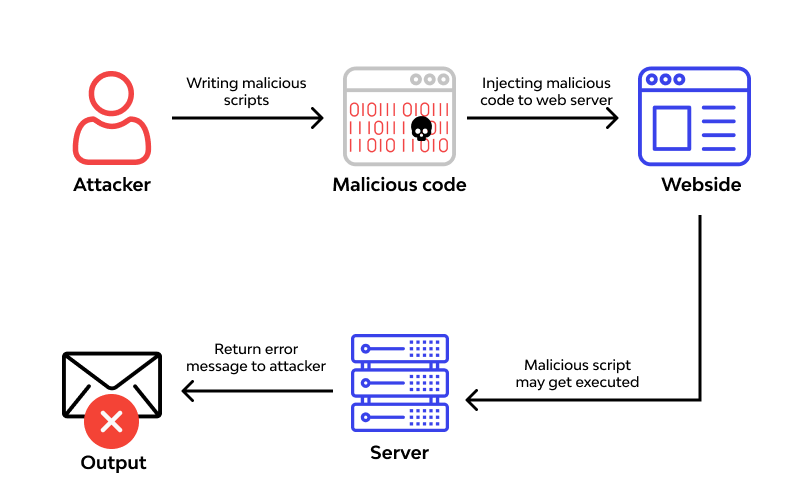


Clickjacking, a deceptive attack, involves overlaying malicious elements on a website with benign content. Users, perceiving only the benign content, unwittingly interact with the hidden elements.

These interactions can trigger unintended actions, like sharing sensitive information or making unauthorized transactions, all without the user's awareness. The attack capitalizes on users' trust in what they see, exploiting their actions for malicious purposes.

To prevent clickjacking, developers must use techniques like frame-busting scripts and implement security headers (like X-Frame-Options) to deny malicious sites the ability to load their content within frames, thus safeguarding users from this kind of manipulation.

5) Remote Code Execution

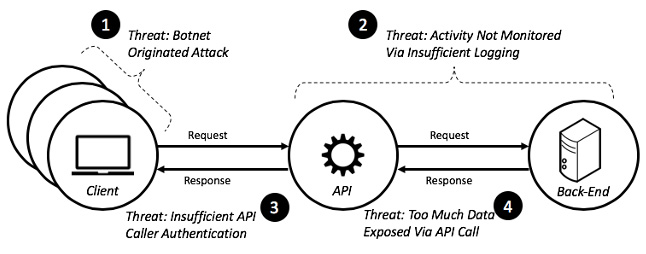


Remote Code Execution (RCE) is a critical web application vulnerability allowing attackers to inject and execute malicious code on a targeted server.

Often stemming from inadequate input validation or insecure deserialization, attackers exploit these vulnerabilities to send and execute arbitrary commands remotely. This can lead to a complete compromise of the system, data breaches, unauthorized access, or further attacks like lateral movement within a network. Prevention involves input validation, secure coding practices, and regular security patches.

Employing strong access controls, applying the principle of least privilege, and monitoring system behavior for anomalies are essential. Rapid detection and response, along with thorough security testing, are vital to minimizing the potential damage inflicted by RCE attacks.

6) API Attacks

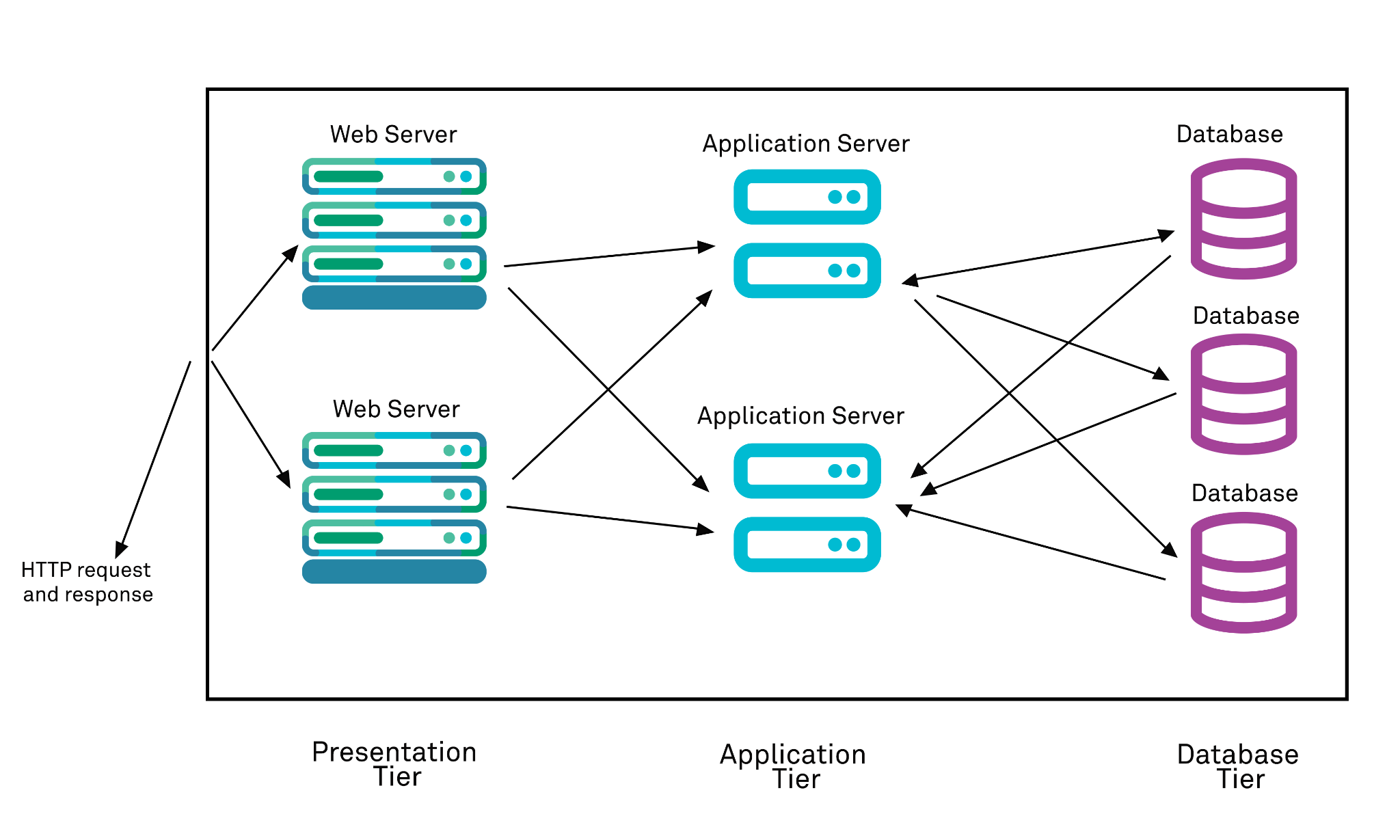


API attacks target vulnerabilities in the interfaces that allow different software components to communicate.

Attackers exploit weaknesses in input validation, authentication, or authorization to gain unauthorized access, exfiltrate sensitive data, or disrupt services. Common types include injection attacks, like SQL or XML injections, and improper use of APIs, like using unauthenticated endpoints.

To mitigate API attacks, implement strong authentication and authorization mechanisms, validate and sanitize input data, employ encryption for sensitive information, and use security tokens. Regular security assessments and monitoring API traffic for suspicious patterns are essential to identify and prevent potential API vulnerabilities and breaches.

7) HTTP Parameter Pollution

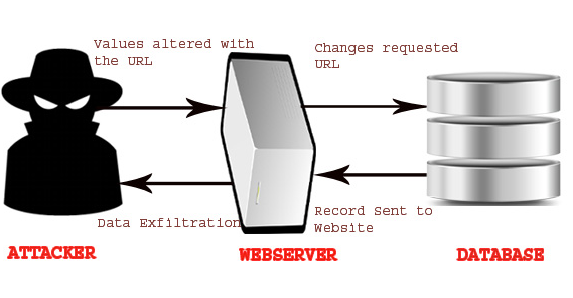


HTTP Parameter Pollution (HPP) is a web attack where an attacker manipulates a web application's URL parameters or HTTP headers.

By injecting additional parameters or values, the attacker confuses the application's logic, potentially bypassing security controls or causing unintended behavior. HPP attacks can lead to data exposure, privilege escalation, or application crashes.

Mitigation involves proper input validation, ensuring that multiple parameters with the same name are handled correctly, and employing security mechanisms such as web application firewalls. Regular security testing and staying updated on emerging HPP attack vectors are crucial to safeguarding web applications from this manipulation.

8) Insecure Object References



Insecure Object References occur when a web application doesn't properly validate user access to objects or resources.

Attackers exploit this vulnerability by manipulating input to access unauthorized data or functionality. For instance, they might modify URLs to access other users' profiles or sensitive information.

Insecure Object References can lead to data exposure, privilege escalation, and compromise of confidential information. Preventive measures include strict authorization checks, employing unique identifiers rather than predictable ones, and implementing access controls at both the application and database levels. Regular security audits and thorough testing help identify and rectify these vulnerabilities before they are exploited.

9) Unvalidated Redirects and Forwards

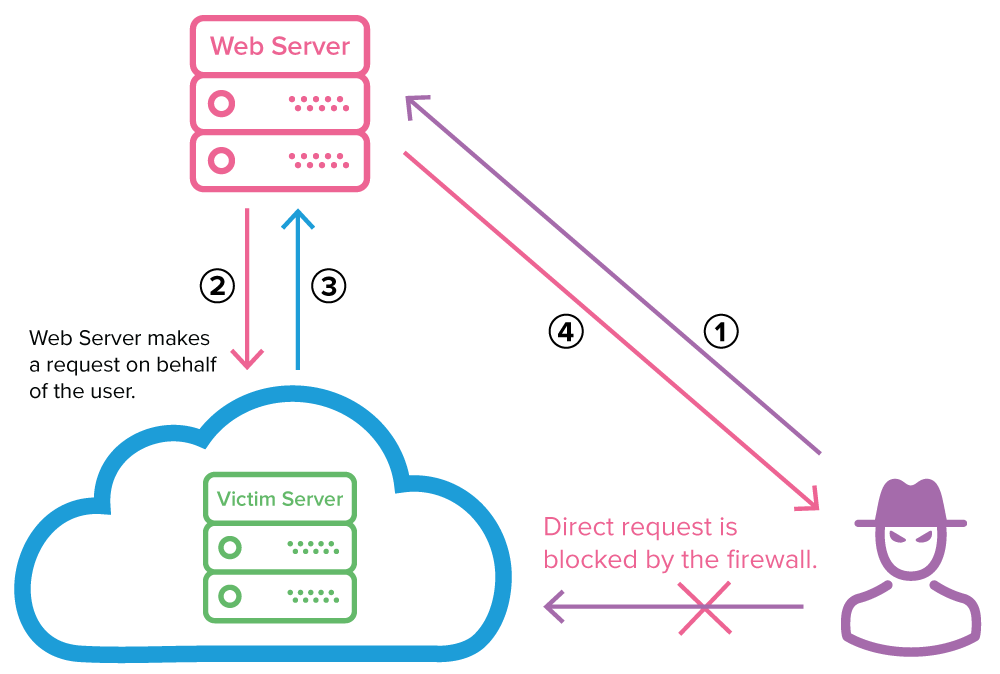


Unvalidated Redirects and Forwards are web vulnerabilities where attackers manipulate URLs to redirect users to malicious sites or unauthorized pages.

By exploiting improper validation of user-provided input, attackers deceive users into clicking on seemingly legitimate links that lead to harmful destinations. This can facilitate phishing attacks, data theft, or the injection of malicious content.

Mitigation involves validating and sanitizing user input, using server-side controls to manage redirects, and applying security mechanisms like security tokens to ensure authorized navigation. Regular security assessments and staying informed about emerging attack techniques are vital to thwart potential unvalidated redirect and forward vulnerabilities.

10) Server-Side Request Forgery



Server-Side Request Forgery (SSRF) is a web vulnerability where attackers manipulate a web application into making unauthorized requests to internal or external resources.

By exploiting inadequate input validation, attackers trick the server into fetching data from unintended locations, potentially leading to data exposure, remote code execution, or the discovery of sensitive information. Attackers might target internal databases, exploit cloud metadata endpoints, or perform port scanning on internal networks.

To prevent SSRF, input validation and whitelisting of allowed resources are crucial. Employing security controls, like network firewalls, to restrict external access from within the application, and using specialized libraries for URL handling, can further mitigate the risk of SSRF attacks.

Date : 30/08/2023

## **Task 6**

**Understanding CIS Policies**

The Center for Internet Security (CIS) publishes the CIS Critical Security Controls (CSC) to help organizations better defend against known attacks by distilling key security concepts into actionable controls to achieve greater overall cybersecurity defense.

Formerly the SANS Critical Security Controls (SANS Top 20) these are now officially called the CIS Critical Security Controls (CIS Controls).

CIS Controls Version 8 combines and consolidates the CIS Controls by activities, rather than by who manages the devices. Physical devices, fixed boundaries, and discrete islands of security implementation are less important; this is reflected in v8 through revised terminology and grouping of Safeguards, resulting in a decrease of the number of Controls from 20 to 18.

The CIS Critical Security Controls (CIS Controls) are a set of best practices that organizations can use to improve their cybersecurity posture. The CIS Controls are divided into 8 groups, each of which contains a number of specific controls.

The top 18 CIS Controls are:

1. Inventory and Control of Hardware Assets

2. Inventory and Control of Software Assets

3. Data Protection

4. Secure Configuration of Enterprise Assets and Software

5. Account Management

6. Access Control Management

7. Continuous Vulnerability Management

8. Audit Log Management

9. Email and Web Browser Protections

10. Malware Defenses

11. Data Recovery

12. Network Infrastructure Management

13. Network Monitoring and Defense

14. Security Awareness and Skills Training

15. Service Provider Management

16. Application Software Security

17. Incident Response Management

18. Penetration Testing

**Explanations**

1. **Inventory and Control of Hardware Assets** : This control requires organizations to have a comprehensive inventory of all of their hardware assets, including computers, laptops, printers, and mobile devices. This inventory should be updated regularly to ensure that it is accurate.
2. **Inventory and Control of Software Assets** : This control requires organizations to have a comprehensive inventory of all of their software assets, including operating systems, applications, and firmware. This inventory should be updated regularly to ensure that it is accurate.
3. **Data Protection** : This control requires organizations to protect their data from unauthorized access, use, disclosure, disruption, modification, or destruction. This can be achieved through a variety of measures, such as encryption, access control, and backup and recovery.
4. **Secure Configuration of Enterprise Assets and Software** : This control requires organizations to configure their assets and software in a secure manner. This includes setting strong passwords, disabling unnecessary ports and services, and applying security patches.
5. **Account Management** : This control requires organizations to manage their user accounts effectively. This includes creating and managing strong passwords, deactivating unused accounts, and monitoring for suspicious activity.
6. **Access Control Management** : This control requires organizations to control access to their systems and data. This can be achieved through a variety of measures, such as role-based access control, multi-factor authentication, and least privilege.
7. **Continuous Vulnerability Management** : This control requires organizations to scan their systems and software for vulnerabilities on a regular basis. Vulnerabilities can be exploited by attackers to gain access to systems and data, so it is important to identify and fix them promptly.
8. **Audit Log Management** : This control requires organizations to collect and analyze audit logs from their systems and applications. Audit logs can provide valuable information about security incidents, so it is important to keep them secure and accessible.
9. **Email and Web Browser Protections** : This control requires organizations to implement security measures to protect their users from malicious emails and websites. This includes using spam filters, antivirus software, and pop-up blockers.
10. **Malware Defenses** : This control requires organizations to implement security measures to protect their systems from malware. This includes using antivirus software, firewalls, and intrusion detection systems.
11. **Data Recovery** : This control requires organizations to have a plan in place to recover data in the event of a security incident or natural disaster. This plan should include steps for backing up data, restoring data, and testing the recovery process.
12. **Network Infrastructure Management** : This control requires organizations to manage their network infrastructure effectively. This includes configuring network devices securely, monitoring network traffic, and implementing security policies.
13. **Network Monitoring and Defense** : This control requires organizations to monitor their networks for suspicious activity. This can be done by using intrusion detection systems and other security tools.
14. **Security Awareness and Skills Training** : This control requires organizations to train their employees on security best practices. This training should cover topics such as password security, phishing attacks, and social engineering.
15. **Service Provider Management** : This control requires organizations to manage their relationships with third-party service providers. This includes ensuring that these providers have adequate security measures in place.
16. **Application Software Security** : This control requires organizations to secure their application software. This can be achieved through a variety of measures, such as code reviews, penetration testing, and vulnerability scanning.
17. **Incident Response Management** : This control requires organizations to have a plan in place to respond to security incidents. This plan should include steps for identifying, containing, and recovering from incidents.
18. **Penetration Testing** : This control requires organizations to have their systems and applications penetration tested by a qualified security professional. This testing can help to identify and fix security vulnerabilities.

The CIS Controls are a comprehensive set of best practices that can help organizations to improve their cybersecurity posture. By implementing these controls, organizations can reduce their risk of being attacked and help to protect their data and systems.

## 