TASK 1

Top 10 Most Notorious Hackers of All Time

1. Kevin Mitnick:

   Kevin Mitnick is a renowned figure in the world of hacking. He began his hacking career as a teenager and was initially categorized as a "Black Hat Hacker" due to his involvement in unauthorized computer access and data theft. Mitnick's exploits included hacking into the North American Defence Command (NORAD) and copying software from Digital Equipment Corporation (DEC). However, he never exploited the data he obtained for personal gain, and over time, he shifted his focus to ethical hacking (White Hat). Today, Mitnick is recognized as a cybersecurity consultant and author, helping organizations secure their systems and networks, making a significant transition from his early black hat hacking days.

2. Anonymous:

   The Anonymous group is a loosely organized and decentralized collective of hackers and activists. They are known for their hacktivist activities, often using digital means to protest against perceived injustices or promote various causes. Anonymous hackers are typically categorized as "Grey Hat Hackers" because their actions can range from ethical to potentially illegal. They have been involved in cyberattacks against government websites, corporations, and organizations they view as oppressive or corrupt. Anonymous often operates under a banner of anonymity, using online platforms to communicate and coordinate their actions. Their motives vary, but they are often associated with advocating for free speech, government transparency, and social justice.

3. Adrian Lamo:

   Adrian Lamo was a notable figure in the hacking community, primarily categorized as a "Grey Hat Hacker." He gained recognition for his ability to identify security vulnerabilities in computer systems and websites. While Lamo reported some of these vulnerabilities to the affected organizations, he also faced legal issues for unauthorized access to computer systems, which is why he's considered a Grey Hat Hacker. One of his most prominent actions was reporting the activities of U.S. Army intelligence analyst Chelsea Manning, who leaked classified documents to WikiLeaks. Lamo's actions and ethical stance remain a subject of debate, as he straddled the line between ethical hacking and unauthorized intrusion during his hacking career.

4. Albert Gonzalez:

   Albert Gonzalez was a notorious "Black Hat Hacker" who engaged in cybercriminal activities. He gained infamy for his involvement in high-profile credit card theft and hacking schemes. Gonzalez led a criminal group that targeted major retailers and corporations, compromising their payment card data and causing substantial financial losses. His hacking activities culminated in what is considered one of the largest data breaches in history, affecting companies like TJX, Heartland Payment Systems, and others. Ultimately, Gonzalez's actions led to his arrest, conviction, and a lengthy prison sentence, underscoring his classification as a Black Hat Hacker due to his involvement in illegal and malicious activities in the cyber realm.

5. Matthew Bevan and Richard Pryce:

   Matthew Bevan and Richard Pryce were British hackers known for their activities in the late 1990s. They are often classified as "Grey Hat Hackers" due to the ambiguous nature of their actions. Bevan and Pryce gained attention for hacking into U.S. military and government computer systems, including NASA and the U.S. Air Force. While their motivations were not purely malicious, as they claimed to be conducting these intrusions to expose security vulnerabilities, their actions were illegal and raised significant concerns about national security. Their hacking activities led to investigations and legal consequences, and they remain notable figures in the history of hacking for their high-profile breaches and the ethical debate surrounding their actions.

6. Jeanson James Ancheta:

   Jeanson James Ancheta was a notorious "Black Hat Hacker" known for his involvement in creating and spreading malicious computer worms and botnets. In the mid-2000s, Ancheta gained notoriety for infecting a large number of computers, turning them into zombies and using them to launch distributed denial-of-service (DDoS) attacks on various websites. He also profited from his hacking activities by renting out these botnets for other cybercriminals to use. Ancheta's actions ultimately led to his arrest and conviction for computer crimes, marking him as a classic example of a Black Hat Hacker engaged in illegal and malicious activities in the cyber world.

7. Michael Calce:

   Michael Calce, also known as "Mafiaboy," gained notoriety as a "Black Hat Hacker" during his teenage years in the early 2000s. He orchestrated a series of high-profile cyberattacks, including a major distributed denial-of-service (DDoS) attack against popular websites, including Yahoo!, eBay, and Amazon. Calce's actions disrupted these online services and caused widespread concern about internet security. He was eventually apprehended and convicted of multiple computer-related crimes. Calce's activities exemplify the actions of a Black Hat Hacker, engaged in illegal and disruptive cyber activities for personal gain and notoriety.

8. Kevin Poulsen:

   Kevin Poulsen, also known as "Dark Dante," is a unique figure in the hacking world who transitioned from a "Black Hat Hacker" to a "White Hat Hacker." In his early years, Poulsen was involved in various hacking activities, including taking over phone lines for radio station contests and hacking into computer systems. However, he gained notoriety for his involvement in a high-profile hack of telephone company computers to win a Porsche. After serving a prison sentence for his cybercrimes, Poulsen transformed into a cybersecurity journalist and expert. He has since used his extensive knowledge to contribute positively to the field by reporting on cybersecurity issues and helping organizations improve their digital security, firmly placing him in the category of a White Hat Hacker.

9. Jonathan James:

   Jonathan James, also known as "c0mrade," was a "Black Hat Hacker" who gained infamy at a young age. In 2000, at the age of 16, he became the first juvenile to be incarcerated for cybercrimes in the United States. James was involved in various hacking activities, including unauthorized access to government and corporate networks. His most notable breach was infiltrating NASA's computer systems, where he stole software and caused significant disruptions. His actions led to legal consequences, marking him as a Black Hat Hacker engaged in illegal and malicious cyber activities. Tragically, Jonathan James passed away in 2008 at the age of 24.

10. ASTRA:

    The hacker known as ASTRA stands out from the others on this list due to the mystery surrounding their identity. Unlike many hackers, ASTRA has never been publicly identified. However, according to reports, ASTRA was apprehended by authorities in 2008 and was revealed to be a 58-year-old Greek mathematician at the time of his arrest. ASTRA's hacking activities were primarily focused on infiltrating the Dassault Group for nearly half a decade. During this period, he illicitly acquired cutting-edge weapons technology software and data, which he allegedly sold to approximately 250 individuals worldwide. The damages caused by his hacking were estimated to be around $360 million for the Dassault Group. Despite these details, the reason behind the continued secrecy of ASTRA's complete identity remains a mystery. It's noteworthy that "ASTRA" translates to "weapon" in Sanskrit, adding an intriguing layer to the enigma surrounding this hacker. He can be categorised as a Black Hat Hacker.

TASK 2

Ports and Vulnerabilities

Port 20

Data Exposure: Port 20 is used for FTP data transfer. It's vulnerable to data exposure because FTP transfers are often unencrypted, which means that any data sent over this port, including sensitive information like usernames and passwords, can be intercepted and accessed by attackers.

Data Injection: Attackers can exploit vulnerabilities in FTP servers listening on port 20 to inject malicious files onto the server. If the server doesn't properly validate and sanitize incoming data, it could lead to the upload of malicious files that can compromise the server's security.

Port 21

Brute Force Attacks: Port 21 is used for FTP control and authentication. It's susceptible to brute force attacks where attackers try various username and password combinations until they gain unauthorized access to the FTP server.

FTP Bounce Attack: FTP bounce attack is a technique where an attacker uses an FTP server to perform port scans of other servers. By abusing the PORT command, attackers can use the FTP server as a proxy to scan other systems, potentially bypassing firewall rules.

Port 22

SSH Brute force: Port 22 is the default port for SSH (Secure Shell) which provides secure remote access to systems. It's vulnerable to brute force attacks where attackers repeatedly try different username and password combinations to gain unauthorized access.

SSH Protocol Vulnerabilities: The SSH protocol itself may have vulnerabilities that can be exploited by attackers to gain unauthorized access or execute arbitrary code on a target system, especially if the SSH software is not kept up-to-date.

Port 23

Telnet Password Sniffing: Port 23 is used for the Telnet protocol, which transmits data including passwords in plain text. This makes it vulnerable to password sniffing attacks, where attackers can intercept and read sensitive information.

Remote Code Execution: Telnet servers often have vulnerabilities that can be exploited for remote code execution. Attackers can take advantage of these vulnerabilities to execute malicious code on the target system.

Port 25

SMTP Relay: Port 25 is used for SMTP (Simple Mail Transfer Protocol), and misconfigured SMTP servers can be abused by attackers to relay spam emails, facilitating large-scale email spam campaigns.

Email Spoofing: Attackers can exploit vulnerabilities in email servers on port 25 to spoof email addresses, sending emails that appear to be from a legitimate source but actually contain malicious content or links.

Port 53

DNS Cache Poisoning: Port 53 is used for DNS (Domain Name System) services. DNS cache poisoning attacks can manipulate the DNS records stored on a vulnerable DNS server, redirecting users to malicious websites or intercepting their traffic.

Amplification Attacks: Attackers can abuse poorly configured DNS servers on port 53 to launch DDoS amplification attacks, where a small request to the DNS server triggers a much larger response to the target, overwhelming its resources.

Port 69

TFTP Security: Port 69 is used for the Trivial File Transfer Protocol (TFTP), which lacks authentication and encryption mechanisms. This makes it vulnerable to unauthorized file transfers and exposes data to interception by attackers.

Directory Traversal: Poorly configured TFTP servers on port 69 can be exploited with directory traversal attacks, where attackers manipulate file paths to access files outside the intended directory, potentially exposing sensitive information.

Port 80

HTTP-Based Attacks: Port 80 is the default port for HTTP, and web servers listening on this port are vulnerable to a wide range of attacks, including SQL injection, cross-site scripting (XSS), and remote code execution if not properly secured and updated.

DDoS Attacks: Web servers on port 80 are often targeted in Distributed Denial of Service (DDoS) attacks, where a massive influx of traffic overwhelms the server's resources, causing it to become inaccessible.

Port 110

POP3 Brute Force: Port 110 is used for the POP3 (Post Office Protocol version 3) email retrieval protocol. It's vulnerable to brute force attacks similar to other authentication protocols, where attackers attempt to guess usernames and passwords.

Email Header Injection: Poorly secured POP3 servers can be susceptible to email header injection attacks, where attackers manipulate email headers to forge messages, insert malicious content, or execute phishing attacks.

Port 123

NTP Amplification: Port 123 is used for the Network Time Protocol (NTP). Attackers can abuse misconfigured NTP servers for DDoS amplification attacks, where a small query to the server triggers a much larger response to the target, overwhelming it.

Time-based Attacks: In some cases, vulnerabilities in NTP implementations can be exploited by attackers to manipulate time settings on systems, potentially disrupting services or causing synchronization issues.

Port 143

IMAP Brute Force: Port 143 is used for the IMAP (Internet Message Access Protocol) email retrieval protocol. It can be vulnerable to brute force attacks on email accounts, where attackers attempt various credentials to gain unauthorized access.

Email Content Snooping: If IMAP is not properly configured, attackers might exploit vulnerabilities to gain unauthorized access to users' email contents, potentially exposing sensitive information.

Port 443

SSL/TLS Vulnerabilities: Port 443 is used for HTTPS, which encrypts data transmission. However, improperly configured SSL/TLS certificates or outdated encryption protocols can lead to vulnerabilities that attackers can exploit.

Heartbleed: In the past, a vulnerability known as Heartbleed affected OpenSSL, a widely used SSL/TLS library. This vulnerability allowed attackers to read sensitive information from a server's memory, potentially compromising encrypted data.

TASK 3
Top 5 OWASP

1. CWE: 284: Improper Access Control

   OWASP Category: A01:2021-Broken Access Control

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

   Business Impact: This weakness allows an attacker to bypass intended security restrictions and perform a variety of actions depending on the source of error and functionality of the application. An attacker might be able to perform certain actions by gaining elevated privileges, reading otherwise restricted information, executing commands, bypassing implemented security mechanisms, etc.

2. CWE-326: Inadequate Encryption Strength

   OWASP Category: A02:2021 – Cryptographic Failures

   Description: The product stores or transmits sensitive data using an encryption scheme that is theoretically sound, but is not strong enough for the level of protection required. A weak encryption scheme can be subjected to brute force attacks that have a reasonable chance of succeeding using current attack methods and resources.

   Business Impact: An attacker may be able to decrypt the data using brute force attacks leading to data breaches which could further lead to legal consequences, reputational damage, financial losses, operational disruptions, competitive disadvantage, loss of intellectual property, and potential lawsuits. Strong encryption and compliance with regulations are crucial to mitigate these risks.

3. CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

   OWASP Category: A03:2021 – Injection

   Description: The product constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended SQL command when it is sent to a downstream component.

   Business Impact: SQL injection vulnerabilities pose significant risks to a system's security. They can result in unauthorized reading of application data, undermining confidentiality, especially when databases store sensitive information. Additionally, these vulnerabilities can enable attackers to bypass protection mechanisms, potentially accessing the system without proper credentials and even modify or delete application data, compromising data integrity.

4. CWE-256: Plaintext Storage of a Password

   OWASP Category: A04:2021 – Insecure Design

   Description: Storing a password in plaintext may result in a system compromise. Password management issues occur when a password is stored in plaintext in an application's properties, configuration file, or memory. Storing a plaintext password in a configuration file allows anyone who can read the file access to the password-protected resource. In some contexts, even storage of a plaintext password in memory is considered a security risk if the password is not cleared immediately after it is used.

   Business Impact: Storing a plaintext password in a configuration file allows anyone who can read the file access to the password-protected resource. This can lead to unauthorized access, identity theft, financial loss, data breaches, and reputational damage. Developers sometimes feel they are helpless in safeguarding the application against someone with configuration file access, but this mindset aids potential attackers.

5. CWE-611: Improper Restriction of XML External Entity Reference

   OWASP Category: A05:2021 – Security Misconfiguration

   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.

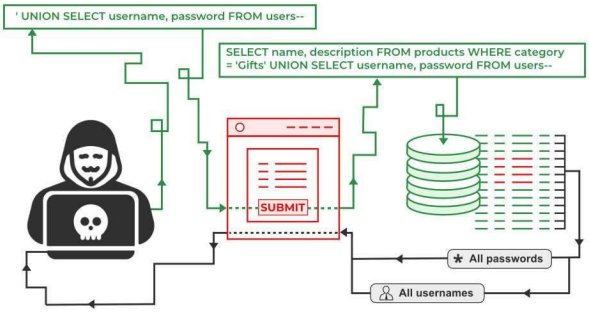
   Business Impact: Improper Restriction of XML External Entity Reference can have a significant impact on security. It can lead to unauthorized access to internal files, potential data exposure, and even denial of service. By exploiting this vulnerability, attackers may manipulate XML parsers to access or modify sensitive information, disrupting system functionality and compromising data integrity and confidentiality. Preventing and mitigating this vulnerability is crucial to maintaining a secure software environment.

TASK 4

Top 10 Web Server Attacks

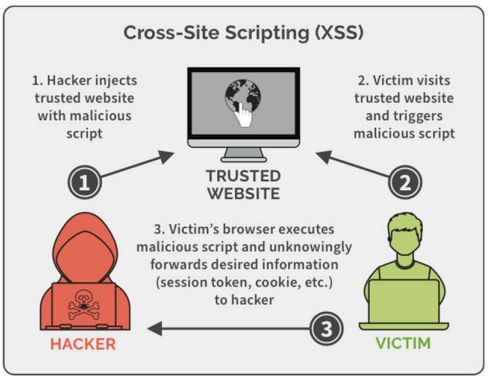
1. SQL Injection (SQLi):

SQL Injection (SQLi) is a malicious attack where attackers inject rogue SQL queries into input fields of a web application. By exploiting weak or unprotected input validation, they can manipulate the database operations. This can lead to unauthorized access, data theft, data manipulation, or even complete control over the database. SQLi attacks can disclose sensitive information, modify or delete records, and potentially grant access to the entire system. Proper input validation and parameterized queries are essential defences against SQL Injection attacks.



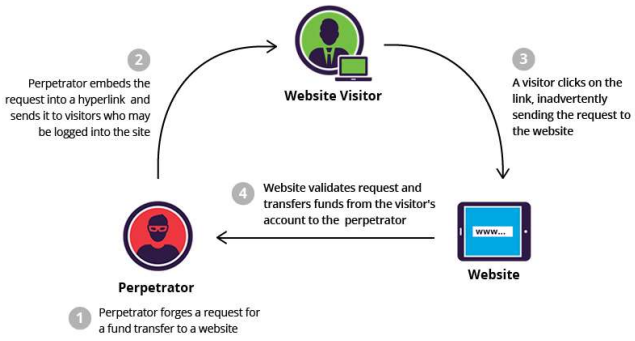
2. Cross-Site Scripting (XSS):

Cross-Site Scripting (XSS) is a harmful attack where malicious scripts are injected into a web application, typically through input fields or other vulnerable areas. When other users access the affected application, these scripts execute in their browsers within the context of the application. This allows attackers to steal cookies, session tokens, or even perform actions on behalf of the user. XSS attacks can have various consequences, including data theft, defacement of websites, spreading malware, or redirecting users to malicious sites. Prevention involves input validation, proper encoding of user inputs, and adopting security headers like Content Security Policy (CSP).



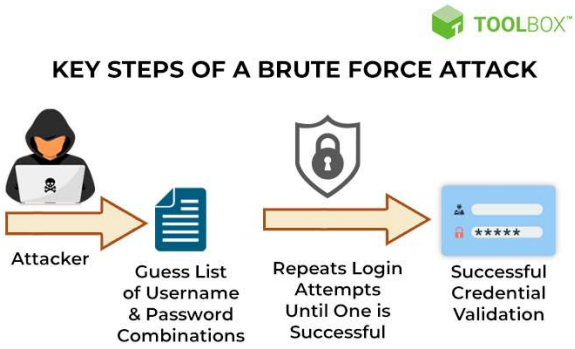
3. Cross-Site Request Forgery (CSRF):

Cross-Site Request Forgery (CSRF) is an attack where an authenticated user is tricked into unknowingly executing actions on a different website where they are authenticated. The attacker creates a malicious request, often using an image or a link, which the victim is induced to click. If the victim is authenticated on the targeted website, the malicious request executes actions as the victim, potentially modifying data, changing passwords, or performing unauthorized transactions. To prevent CSRF attacks, developers implement measures like unique tokens per session, validating the Referer header, or implementing anti-CSRF tokens in forms.



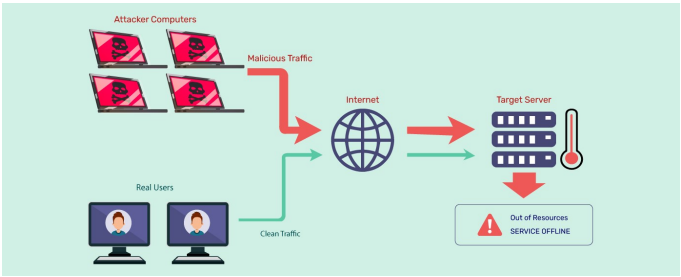
4. Brute Force Attack:

A Brute Force Attack is a malicious activity where automated tools systematically attempt a vast number of username and password combinations to gain unauthorized access to a system or account. Attackers use these tools to "guess" the correct login credentials by trying various combinations until successful entry is achieved. This attack relies on the sheer volume of attempts and is often used to compromise weak or easily guessable passwords. Countermeasures include enforcing strong password policies, implementing account lockout policies, and utilizing multi-factor authentication to enhance security and deter brute force attacks.



5. Distributed Denial of Service (DDoS):

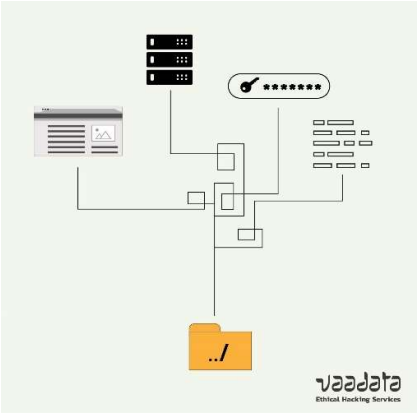
A Distributed Denial of Service (DDoS) attack is a malicious assault in which a multitude of compromised devices, often forming a botnet, flood a web server or network with an overwhelming volume of traffic. The sheer volume of traffic exhausts the server's resources, making it slow or completely unavailable to genuine users. Attackers exploit this tactic to disrupt the targeted website, online service, or network, causing financial losses or hindering operations. Mitigating DDoS attacks involves implementing robust network security measures, utilizing DDoS mitigation services, and having a DDoS response plan in place to minimize the impact of such attacks.



6. Path Traversal:

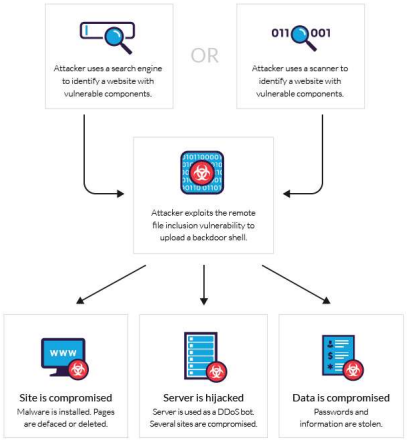
Path Traversal, also known as Directory Traversal, is a web server attack that occurs when an attacker exploits vulnerable input fields to navigate through directories and access files and directories they are not supposed to. By manipulating input parameters containing file paths, attackers attempt to break out of the intended directory and access sensitive files or execute unauthorized actions on the server. This attack is often facilitated through "../" or similar constructs to navigate up the directory tree.

Preventing Path Traversal attacks involves thorough input validation, restricting access rights, and properly configuring server permissions to limit file system access. Additionally, implementing security controls and employing a secure coding approach helps mitigate the risk of successful path traversal attacks.



7. Remote File Inclusion (RFI):

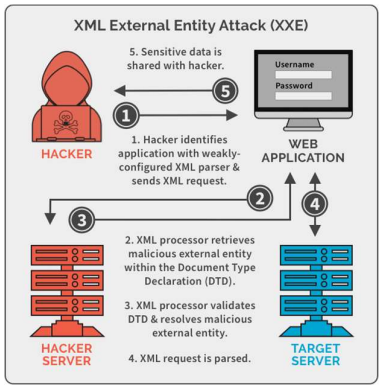
Remote File Inclusion (RFI) is a type of web server attack where an attacker exploits vulnerabilities in a web application to include files hosted on a remote server. Attackers inject malicious code or provide a URL that points to a malicious script hosted on an external server. When the application processes this input without proper validation or sanitization, it includes and executes the external script, potentially leading to unauthorized access, data breaches, or other malicious actions.



To prevent RFI attacks, developers should avoid allowing user input to dictate file paths or URLs directly. Instead, they should validate and sanitize input rigorously and employ a whitelist approach to ensure that only trusted, specific files or URLs can be included. Proper server configurations and regular security assessments are crucial to mitigate the risk of RFI attacks.

8. XML External Entity (XXE) Attack:

XML External Entity (XXE) Attack is when attackers manipulate vulnerable XML parsers by injecting malicious XML entities. These entities can disclose internal files, perform remote requests, or even execute commands on the server. To prevent XXE attacks, applications should disable external entity processing and validate/sanitize XML input effectively.



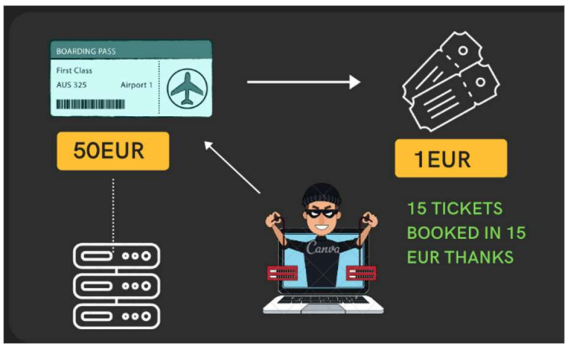
9. Server-Side Request Forgery (SSRF):

Server-Side Request Forgery (SSRF) is a web server attack where an attacker manipulates a web application into making requests to internal or external resources. By exploiting this vulnerability, attackers can make the application act on their behalf, potentially accessing internal services, bypassing firewalls, or retrieving sensitive information. SSRF poses a serious threat as it can lead to data exposure and compromise of internal systems.

Preventing SSRF involves validating and restricting input that can be used to make requests, employing proper access controls, and configuring the server to block requests to internal/private IP addresses. It's essential to educate developers about SSRF risks and ensure they implement secure coding practices to mitigate this attack.

10. Insecure Direct Object Reference (IDOR):

Insecure Direct Object Reference (IDOR) is an attack where a user can manipulate input like URLs or parameters to access or modify unauthorized resources. It's a security vulnerability that occurs when an application does not properly validate user permissions, allowing access to restricted data or actions. Prevention involves enforcing strict access controls and ensuring that users can only access authorized resources.



TASK 5
CIS Policies

Basic Controls:

1. **Inventory and Control of Hardware Assets:** Establish and maintain an organized inventory of all hardware devices within the organization's network. This ensures visibility and control over authorized and unauthorized devices.
2. **Inventory and Control of Software Assets:** Maintain an up-to-date inventory of all authorized software across the organization, enabling efficient management, updates, and license compliance.
3. **Continuous Vulnerability Management:** Regularly identify, evaluate, and mitigate vulnerabilities within systems and software to minimize potential security risks and maintain a secure environment.
4. **Controlled Use of Administrative Privileges:** Implement strict controls and oversight over administrative access to systems, limiting access to authorized personnel and reducing the risk of misuse.
5. **Secure Configuration for Hardware and Software:** Establish and enforce secure configurations for hardware and software on various devices (mobile, laptops, workstations, and servers) to minimize security risks and vulnerabilities.
6. **Maintenance, Monitoring, and Analysis of Audit Logs:** Continuously monitor, review, and analyze system audit logs to detect and respond to security incidents, unauthorized activities, or policy violations effectively.

Foundational Controls:

7. **Email and Web Browser Protections:** Employ security measures to protect email and web browsers from potential threats like phishing, malware, and other cyber-attacks to ensure a secure browsing and communication environment.
8. **Malware Defenses:** Implement robust defenses against malware, including antivirus software, intrusion detection systems, and regular updates, to prevent and detect malicious software.
9. **Limitation and Control of Network Ports, Protocols, and Services:** Restrict and manage network ports, protocols, and services to reduce the organization's attack surface and minimize potential vulnerabilities.
10. **Data Recovery Capabilities:** Establish and maintain data recovery processes and capabilities to ensure the organization can swiftly and effectively recover critical data in the event of a security breach or data loss.
11. **Secure Configuration for Network Devices:** Configure network devices such as firewalls, routers, and switches securely to protect against unauthorized access, data breaches, and other network-based attacks.
12. **Boundary Defense:** Implement defenses at network boundaries to monitor and control traffic entering and leaving the organization's network, enhancing security posture.

13. **Data Protection:** Safeguard sensitive data through encryption, access controls, and monitoring to maintain data integrity, confidentiality, and compliance with relevant regulations.
14. **Controlled Access Based on Need to Know:** Ensure that individuals are granted access only to the data and systems necessary for their specific roles, limiting potential exposure and enhancing security.
15. **Wireless Access Control:** Implement controls to secure and manage wireless access to the network, protecting against unauthorized access and potential security threats.
16. **Account Monitoring and Control:** Continuously monitor and manage user accounts, ensuring appropriate access levels and prompt action against any suspicious activities to prevent unauthorized access.

Organizational Controls:

17. **Implement a Security Awareness and Training Program:** Develop and conduct a comprehensive security awareness program to educate employees about security policies, procedures, and potential threats, fostering a security-conscious culture within the organization.
18. **Application Software Security:** Integrate security measures into the software development lifecycle to identify and address security vulnerabilities in applications, reducing the risk of exploitation.
19. **Incident Response and Management:** Establish an incident response plan and team, enabling the organization to effectively detect, respond to, and recover from security incidents while minimizing damage and downtime.
20. **Penetration Tests and Red Team Exercises:** Conduct penetration tests and red team exercises to simulate real-world attacks, identify vulnerabilities, and enhance incident response strategies and overall security posture through proactive measures.