CompTIA Security+ SY0-701 exam - Satender Kumar

Threats, Vulnerabilities, and Mitigations

2.1 Compare and contrast common threat actors and motivations

1. Threat Actors

1. **Nation-State**:
    - **Motivation**: Nation-state threat actors are typically motivated by geopolitical objectives, which can include espionage, sabotage, or influencing political outcomes. They often target critical infrastructure, military assets, government institutions, or intellectual property.
    - **Attributes**:
        - **Internal/External**: Can be both internal and external. External actors are typically state-sponsored hackers from rival nations, while insiders can be agents within the government or military working on behalf of their nation.
        - **Resources/Funding**: High resources and funding. Nation-state actors often have the support of their government, which gives them access to advanced tools, training, and human capital.
        - **Level of Sophistication/Capability**: Very high sophistication. These actors typically employ advanced persistent threats (APT), sophisticated malware, and zero-day vulnerabilities to infiltrate and steal sensitive data over extended periods.
2. **Unskilled Attacker (Script Kiddies)**:
    - **Motivation**: Unskilled attackers are typically driven by personal gain, curiosity, or the thrill of disrupting systems. They may also attack to make a name for themselves in online communities.
    - **Attributes**:
        - **Internal/External**: Can be external attackers who use pre-written scripts and tools to exploit vulnerabilities, or insiders who use these tools for malicious purposes.
        - **Resources/Funding**: Low resources. They generally rely on readily available hacking tools and scripts found on the internet.
        - **Level of Sophistication/Capability**: Low sophistication. They lack advanced skills and typically use scripts, which are pre-written programs or tools designed to exploit known vulnerabilities.
3. **Hacktivist**:
    - **Motivation**: Hacktivists are primarily motivated by political, social, or environmental causes. They use cyberattacks to protest or raise awareness about specific issues, such as government surveillance, corporate greed, or human rights violations.
    - **Attributes**:
        - **Internal/External**: External, though insiders with similar motivations can also engage in these attacks.
        - **Resources/Funding**: Moderate to low resources, depending on the group. Hacktivists may be part of larger organized networks like Anonymous or operate alone.
        - **Level of Sophistication/Capability**: Moderate sophistication. Hacktivists can use various techniques, including Distributed Denial of Service (DDoS)

attacks, website defacements, and social media campaigns to achieve their goals.

4. **Insider Threat**:
    - **Motivation**: Insider threats come from individuals within the organization—such as employees, contractors, or partners—who have access to the system and data. These actors are often motivated by personal gain (financial), revenge, or even coercion.
    - **Attributes**:
        - **Internal/External**: Internal, by definition. Insiders have legitimate access to sensitive systems and data, making their attacks harder to detect.
        - **Resources/Funding**: Varies. Insiders often have sufficient access to carry out attacks without the need for external resources.
        - **Level of Sophistication/Capability**: Ranges from low to high sophistication. An insider might use their authorized access to bypass security controls or intentionally leak sensitive information. These threats can be especially damaging because they exploit trusted access.
5. **Organized Crime**:
    - **Motivation**: Organized crime groups typically seek financial gain through illegal activities such as data breaches, ransomware attacks, financial fraud, and identity theft.
    - **Attributes**:
        - **Internal/External**: External. These groups often operate across borders, targeting individuals and organizations to steal financial data or intellectual property.
        - **Resources/Funding**: High resources and funding. Organized crime syndicates often have the resources to purchase advanced attack tools and hire skilled attackers.
        - **Level of Sophistication/Capability**: High sophistication. They may use well-coordinated campaigns involving ransomware, phishing, or large-scale botnet attacks.
6. **Shadow IT**:
    - **Motivation**: Shadow IT refers to the use of unauthorized or unsupported IT systems, applications, and devices within an organization. Employees or departments may use their own solutions to improve productivity, often bypassing IT oversight due to a perceived lack of responsiveness or control from the IT department.
    - **Attributes**:
        - **Internal/External**: Internal. Employees use non-approved tools and applications within their organization.
        - **Resources/Funding**: Low resources, as the tools are usually free or cheap solutions that employees find independently.
        - **Level of Sophistication/Capability**: Low to moderate sophistication. While the tools might not be inherently malicious, they can create significant vulnerabilities in security, as they often lack proper security controls, oversight, or integration with the organization's security infrastructure.

2.2 Attributes of Actors

- **Internal/External**:
    - **Internal**: Insiders, including employees, contractors, or business partners, have access to an organization’s systems. Their actions can either be accidental (negligence) or deliberate (malicious insider). Internal threats are more challenging to detect because they bypass external defenses using legitimate access.

    - **External**: External actors, such as hackers, cybercriminals, and nation-states, do not have direct access to the organization's systems. They must exploit vulnerabilities or use social engineering techniques to gain unauthorized access.
- **Resources/Funding**:
    - Threat actors with **high resources** (like nation-states or organized crime) have access to advanced tools, multiple team members, and continuous funding, making them capable of launching sophisticated attacks such as advanced persistent threats (APTs).
    - **Low-resource actors** (such as unskilled attackers or hacktivists) rely on publicly available tools, social engineering, and opportunistic attacks, targeting weak systems or vulnerabilities without the need for significant investments.
- **Level of Sophistication/Capability**:
    - **High sophistication** means that threat actors possess advanced skills, tactics, and tools that allow them to carry out prolonged, stealthy, and highly targeted attacks (e.g., APTs, zero-day vulnerabilities).
    - **Low sophistication** typically involves unsophisticated attacks such as using publicly available exploit kits, launching basic phishing campaigns, or taking advantage of known vulnerabilities without advanced preparation.

Motivations of Threat Actors

1. Data Exfiltration

- **Definition**: Data exfiltration refers to the unauthorized transfer of sensitive data from a system or network to an external destination, often without the knowledge of the organization.
- **Context in Cybersecurity**:
    - **Primary Goal**: The goal of data exfiltration is to steal intellectual property, confidential business information, trade secrets, or personal data (e.g., PII, credit card details).
    - **Attack Methods**: Exfiltration is commonly achieved via **malware**, **phishing** attacks, or **compromised credentials**. Once an attacker gains access to sensitive data, they may use **C2 servers** (Command and Control) to transfer the data to external locations.
    - **Motivating Threat Actors**: This could be carried out by **nation-states** seeking to steal trade secrets, **hacktivists** wanting to reveal sensitive governmental data, or **cybercriminals** involved in identity theft.
- **Real-World Example**: The **Edward Snowden** case where sensitive National Security Agency (NSA) data was exfiltrated and leaked to the media.

2. Espionage

- **Definition**: Espionage involves the act of spying or gathering secret information, often related to national security, corporate competition, or other sensitive activities.
- **Context in Cybersecurity**:
    - **Motivation**: The goal of espionage is typically to gain an unfair advantage or gather classified intelligence for personal, political, or financial gain.
    - **Attack Methods**: Espionage is often conducted using sophisticated methods such as **Advanced Persistent Threats (APT)**, where attackers maintain long-term access to an organization’s network to monitor communications and extract valuable information.
    - **Motivating Threat Actors**: This is typically a tactic used by **nation-state actors**, though it can also be carried out by corporate competitors or internal actors (**insider threats**).

- **Real-World Example**: The **Chinese cyber-espionage group** APT1, which was found to be conducting large-scale espionage against U.S. corporations.

3. Service Disruption

- **Definition**: Service disruption refers to attacks that interrupt or degrade the services an organization offers, such as **Denial of Service (DoS)** or **Distributed Denial of Service (DDoS)** attacks.
- **Context in Cybersecurity**:
    - **Primary Goal**: The main goal is to render an organization's services or infrastructure unavailable to legitimate users, often with the intent to cause damage or operational interruption.
    - **Attack Methods**: Service disruption is commonly achieved using **DDoS attacks**, where a massive volume of traffic is directed at a service, overwhelming its infrastructure.
    - **Motivating Threat Actors**: Hacktivists and organized crime groups often target high-profile services, while competitors may use this tactic to disrupt business rivals.
- **Real-World Example**: The **Dyn DDoS attack** in 2016, which disrupted internet services by attacking domain name system (DNS) providers.

4. Blackmail

- **Definition**: Blackmail in the cyber world involves threatening to release sensitive information unless the victim complies with the attacker's demands, such as paying a ransom or taking some specific action.
- **Context in Cybersecurity**:
    - **Primary Goal**: Blackmailers typically aim for **financial gain** or **coercion**. The victim is forced to comply with demands to prevent the release of damaging information or the execution of harmful actions.
    - **Attack Methods**: **Ransomware** attacks often serve as a form of cyber blackmail. Attackers encrypt the victim's data and demand payment in exchange for the decryption key.
    - **Motivating Threat Actors**: **Organized crime** syndicates and **cybercriminals** primarily use blackmail techniques, but insiders with access to sensitive data may also carry out blackmail.
- **Real-World Example**: **Ransomware attacks** where attackers threaten to leak or destroy valuable data unless their financial demands are met.

5. Financial Gain

- **Definition**: Financial gain is one of the most common motivations behind cybercrime. Attackers steal money, credit card information, or personal financial details to profit from their activities.
- **Context in Cybersecurity**:
    - **Primary Goal**: The goal is to obtain direct financial benefits through theft or fraud.
    - **Attack Methods**: Common methods include **phishing**, **malware**, and **carding** attacks, which target banking systems and financial institutions to steal money or commit fraud.
    - **Motivating Threat Actors**: **Organized crime**, **cybercriminals**, and **hackers** typically carry out attacks for financial gain.
- **Real-World Example**: The **WannaCry ransomware attack**, which was financially motivated, exploiting unpatched vulnerabilities to demand ransoms from infected systems.

6. Philosophical/Political Beliefs

- **Definition**: Some threat actors are driven by a strong sense of philosophy or political ideology, and they attack organizations or governments they perceive as unethical or corrupt.
- **Context in Cybersecurity**:
    - **Primary Goal**: Their aim is not always financial but to promote a cause, disrupt a system, or expose perceived injustices.
    - **Attack Methods**: Hacktivists might engage in **defacement**, **DDoS**, or data leaks as a form of protest or to draw attention to a cause.
    - **Motivating Threat Actors**: This is most often associated with **hacktivists** or other politically motivated groups.
- **Real-World Example**: The **Anonymous group** attacking government and corporate websites in protest against censorship and in support of free speech.

7. Ethical

- **Definition**: Ethical hackers, or "white hats," aim to improve security by identifying and fixing vulnerabilities. However, ethical motivations can also be used as a disguise for malicious intent.
- **Context in Cybersecurity**:
    - **Primary Goal**: The intention is to help organizations by testing systems and reporting vulnerabilities before they are exploited maliciously.
    - **Attack Methods**: Ethical hackers use penetration testing and vulnerability scanning tools, but their actions are conducted within the legal and ethical framework.
    - **Motivating Threat Actors**: Ethical hackers, security researchers, and security consultants.
- **Real-World Example**: Ethical hackers who report zero-day vulnerabilities to the vendor or use bug bounty programs to identify vulnerabilities.

8. Revenge

- **Definition**: Revenge attacks occur when a person or group seeks to harm an organization or individual as retaliation for a perceived wrong or personal grievance.
- **Context in Cybersecurity**:
    - **Primary Goal**: To cause harm to the target organization or individual as a response to mistreatment, betrayal, or injustice.
    - **Attack Methods**: Revenge attacks might include **data breaches**, **system sabotage**, or **releasing sensitive information** to damage the reputation of the victim.
    - **Motivating Threat Actors**: Former employees, disgruntled partners, or anyone with a personal vendetta.
- **Real-World Example**: **Disgruntled employees** or insiders who leak sensitive data or sabotage organizational systems out of revenge.

9. Disruption/Chaos

- **Definition**: Some attackers are motivated by a desire to create disorder or disruption in systems, simply for the chaos or the thrill of seeing the impact.
- **Context in Cybersecurity**:
    - **Primary Goal**: The goal here is to create havoc without specific financial or ideological objectives. It’s often about demonstrating power or ability.

    - **Attack Methods**: **DDoS**, **data manipulation**, and **system takeovers** are common attack methods used to disrupt operations and cause chaos.
    - **Motivating Threat Actors**: **Unskilled attackers**, some **hacktivists**, or even **script kiddies** may engage in these acts for attention or fun.
- **Real-World Example**: **DDoS attacks** on high-profile websites purely for disruption, such as the attack on **GitHub** in 2018.

10. War

- **Definition**: Cyberwarfare refers to the use of cyberattacks as part of a conflict between nation-states or rival powers, aimed at damaging critical infrastructure or stealing sensitive information to undermine national security.
- **Context in Cybersecurity**:
    - **Primary Goal**: The objective is to disrupt or damage the opponent's government, military, economy, or infrastructure.
    - **Attack Methods**: Cyberwarfare typically involves sophisticated **APTs**, **espionage**, **data exfiltration**, and **infrastructure sabotage**.
    - **Motivating Threat Actors**: **Nation-state actors** engaged in cyber espionage or war, such as during conflicts between countries like Russia, the U.S., or China.
- **Real-World Example**: **Stuxnet**, the cyberattack against Iran's nuclear program, which is considered an example of state-sponsored cyber warfare.

2.2 Explain common threat vectors and attack surfaces

1. Message-based Threat Vectors

These involve communication channels where attackers exploit users through messages to gain access or launch attacks.

a. Email

- **Definition**: Email is one of the most common message-based vectors for cyberattacks. Attackers use email to deliver malware, phishing attempts, and scams.
- **Common Attacks**:
    - **Phishing**: Fraudulent emails designed to trick the recipient into revealing sensitive information, like passwords or credit card numbers.
    - **Spear-phishing**: A more targeted form of phishing where attackers customize their message to a specific individual or organization.
    - **Malware**: Attachments or links in emails that, when clicked, download malicious software like ransomware, viruses, or Trojans.
- **Mitigation**: Email filtering, anti-phishing software, and user education on recognizing suspicious emails can help reduce this risk.

b. Short Message Service (SMS)

- **Definition**: SMS (text messages) is another communication vector often targeted by attackers.
- **Common Attacks**:
    - **Smishing (SMS phishing)**: Fraudulent SMS messages attempting to lure victims into revealing sensitive information.
    - **Malware Links**: SMS can also contain links that, when clicked, lead to malicious websites or download malware.

    - **SIM Swapping**: Attackers convince a mobile provider to switch a phone number to a new SIM card, enabling them to intercept two-factor authentication codes.
- **Mitigation**: Avoid clicking on links in unsolicited text messages and be cautious when sharing personal information over SMS.

c. Instant Messaging (IM)

- **Definition**: IM platforms (like WhatsApp, Facebook Messenger) are commonly used for communication, but they are also targeted by attackers.
- **Common Attacks**:
    - **Malware Delivery**: Attackers can send malicious links or files through IM platforms that, when opened, infect the system.
    - **Phishing Links**: IM services can be used to distribute phishing links, leading victims to fake websites designed to steal login credentials or personal data.
- **Mitigation**: Use encrypted IM platforms and educate users on the dangers of opening unknown links or attachments.

2. Image-based Threat Vectors

Images can be embedded with malicious code or metadata, providing another attack vector.

Definition: Images, particularly in formats like JPEG, PNG, or GIF, can carry hidden threats or malicious payloads.

- **Common Attacks**:
    - **Malicious Metadata**: Malicious code can be hidden within image metadata, which is executed when the image is opened by vulnerable software.
    - **Image-Based Exploits**: Attackers exploit vulnerabilities in image processing libraries or viewers to execute malware.
- **Mitigation**: Use updated software to handle image files and avoid opening images from untrusted sources. Tools can also strip metadata from images to reduce this risk.

3. File-based Threat Vectors

Files are commonly used to deliver malware, steal data, or carry out malicious activities.

Definition: Files like executables, documents, or compressed files are frequently used to deliver malicious payloads.

- **Common Attacks**:
    - **Malware-infected Files**: Files such as PDFs, Word documents, or ZIP files can contain malware (e.g., viruses, worms) that are activated once the file is opened or extracted.
    - **Drive-by Downloads**: Visiting malicious websites can trigger automatic downloads of malware-laden files without the user's knowledge.
- **Mitigation**: Use antivirus software, email filters, and sandboxing to prevent malicious files from executing. Educate users to avoid opening files from unknown sources.

4. Voice Call Threat Vectors

Voice communication systems can also be compromised for malicious purposes.

Definition: Voice calls, whether over the phone or Voice over IP (VoIP), can be used by attackers to gather personal information or launch social engineering attacks.

- **Common Attacks**:
  - **Vishing (Voice Phishing)**: Attackers use phone calls to impersonate legitimate entities (banks, government agencies) and trick victims into providing sensitive information.
  - **Caller ID Spoofing**: Attackers spoof caller IDs to make their calls appear legitimate, increasing the likelihood of successful social engineering.
- **Mitigation**: Be cautious of unsolicited calls asking for sensitive information and use call-blocking technologies to prevent suspicious calls.

5. Removable Device Threat Vectors

Removable devices such as USB drives, external hard drives, or SD cards can introduce malware or other security risks to systems.

Definition: Removable devices are easily connected to systems, making them a prime attack vector for malware or data theft.

- **Common Attacks**:
  - **Malicious USB Devices**: Attackers can create USB drives that automatically execute malware when plugged into a system (e.g., **BadUSB**).
  - **Data Exfiltration**: Insiders or external attackers may use removable devices to steal sensitive data from a compromised system.
- **Mitigation**: Disable USB ports, use device encryption, and employ endpoint security tools to monitor and control the use of removable devices.

6. Vulnerable Software Threat Vectors

Vulnerable software represents a significant attack surface that can be exploited if not properly managed and patched.

Definition: Software vulnerabilities can be exploited by attackers to gain unauthorized access or control over systems.

- **Common Attacks**:
  - **Exploiting Software Bugs**: Vulnerabilities in widely used software (e.g., web browsers, email clients) can be exploited to execute code, steal data, or crash systems.
  - **Zero-Day Attacks**: Attackers exploit unknown vulnerabilities before the software vendor has a chance to release a patch (known as a zero-day exploit).
- **Mitigation**: Regular software patching, vulnerability scanning, and use of intrusion detection systems (IDS) can help mitigate this risk.

Client-based vs. Agentless Vulnerabilities

- **Client-based Vulnerabilities**: These are vulnerabilities found in the client software that interacts with a server (e.g., browsers, email clients, FTP clients).
- **Agentless Vulnerabilities**: These involve systems that do not require a traditional client but can be exploited through web-based services, APIs, or automated attack tools.
- **Mitigation**: Using security solutions that focus on both client-based software (through endpoint protection) and server-side vulnerabilities (e.g., firewall protections).

7. Unsupported Systems and Applications

Outdated systems and applications represent a weak point in network security as they often lack necessary security updates.

Definition: Unsupported systems are those for which the vendor no longer provides security updates or patches.

- **Common Attacks**:
    - **Exploitation of Known Vulnerabilities**: Attackers exploit unpatched vulnerabilities in legacy systems or unsupported software that cannot be updated.
- **Mitigation**: Replace or upgrade unsupported systems, or isolate them from critical infrastructure to reduce exposure to risks.

8. Unsecure Networks

Attackers often exploit insecure network configurations, whether wireless, wired, or Bluetooth, to gain unauthorized access.

a. Wireless Networks

- **Common Attacks**:
    - **Eavesdropping**: Attackers can intercept unencrypted traffic on public or poorly secured Wi-Fi networks.
    - **Man-in-the-Middle (MITM) Attacks**: Attackers position themselves between the victim and a legitimate network, intercepting or manipulating communication.
- **Mitigation**: Use encryption protocols like **WPA3**, ensure strong passwords, and avoid using public networks for sensitive activities.

b. Wired Networks

- **Common Attacks**:
    - **Physical Network Access**: Attackers may gain physical access to network cables or ports to launch attacks, especially in open office environments.
    - **Sniffing/Interception**: Attackers can intercept unencrypted data sent over the network.
- **Mitigation**: Use network encryption, segment sensitive networks, and restrict physical access to network infrastructure.

c. Bluetooth

- **Common Attacks**:
    - **Bluejacking**: Sending unsolicited messages to nearby Bluetooth devices.
    - **Bluesnarfing**: Gaining unauthorized access to a Bluetooth-enabled device, stealing data.
- **Mitigation**: Disable Bluetooth when not in use, ensure strong authentication, and limit discoverability of Bluetooth devices.

Open Service Ports

Open service ports are essential for communication over a network but also pose significant security risks if not properly managed.

- **Definition**: Service ports are used to allow specific types of communication with servers or devices. For example, **port 80** is used for HTTP, **port 443** for HTTPS, and **port 22** for SSH.
- **Security Implications**:
    - **Unnecessary open ports** can provide attackers with entry points into a system. Even if a port is open, attackers may exploit vulnerabilities in the service listening on that port.
    - Attackers can scan systems using tools like **Nmap** to detect open ports and identify services running on them, searching for known vulnerabilities to exploit.
- **Mitigation**:
    - **Firewalls**: Use firewalls to block unused or unnecessary ports.
    - **Port Scanning**: Regularly scan your systems to ensure only necessary ports are open.
    - **Service Hardening**: Disable or remove unused services that listen on open ports.

Default Credentials

Default credentials are often set by manufacturers for devices, applications, or software and can easily be exploited by attackers if they are not changed.

- **Definition**: Default credentials are pre-configured usernames and passwords provided by the manufacturer (e.g., **admin/admin**).
- **Security Implications**:
    - Attackers often know or can easily guess default credentials, allowing them to gain unauthorized access to devices, networks, or applications.
    - Devices like routers, firewalls, and IoT (Internet of Things) devices often have default credentials that may never be changed by the user.
- **Mitigation**:
    - **Change Default Credentials**: Always change default usernames and passwords to unique, strong ones.
    - **Enforce Strong Password Policies**: Implement password complexity requirements and multi-factor authentication (MFA) wherever possible.
    - **Regular Audits**: Regularly check for default credentials and unauthorized access attempts.

Supply Chain Threats

Supply chain threats involve exploiting vulnerabilities in external providers (e.g., MSPs, vendors, or suppliers) that are integrated into an organization's infrastructure.

Managed Service Providers (MSPs)

- **Definition**: MSPs manage a company’s IT infrastructure and end-user systems, often providing IT support, cloud services, and security management.
- **Security Implications**:
    - **Access to sensitive data**: If an MSP is compromised, attackers could access sensitive information or exploit the MSP's access to infiltrate client systems.
    - **Potential Weak Links**: MSPs might not follow the same security standards as the client organization, making them vulnerable to supply chain attacks.
- **Mitigation**:
    - **Vetting MSPs**: Ensure that the MSP follows strong security practices and complies with relevant standards (e.g., **SOC 2**, **ISO 27001**).

    - **Segregation of Duties**: Limit the access privileges of MSPs to only necessary systems.
    - **Third-Party Audits**: Regularly audit MSPs' security measures and performance.

Vendors and Suppliers

- **Definition**: Vendors and suppliers provide products, services, and components to an organization. These can range from software suppliers to hardware manufacturers.
- **Security Implications**:
    - **Third-Party Software**: Vendors may provide software with vulnerabilities that could lead to a breach.
    - **Compromised Hardware**: Hardware components like chips or IoT devices could be compromised before they even reach the organization.
- **Mitigation**:
    - **Vendor Risk Management**: Establish security assessments for vendors and suppliers, and ensure they meet security requirements before they are allowed to operate with your system.
    - **Software and Hardware Vetting**: Perform thorough vetting of any third-party software or hardware before implementing them into your environment.
    - **Supply Chain Monitoring**: Implement measures to continuously monitor and track components, especially in high-risk areas like hardware and network infrastructure.

Human Vectors/Social Engineering

Human vectors are often the **weakest link** in the security chain. Attackers often manipulate individuals to gain unauthorized access or steal data through various social engineering techniques.

Phishing

- **Definition**: Phishing is a social engineering attack where attackers send fraudulent messages, usually via email, to trick users into revealing sensitive information such as passwords, credit card numbers, or other personal details.
- **Security Implications**:
    - **Credential Theft**: Phishing attacks often lead to compromised user credentials.
    - **Malware Delivery**: Phishing emails may contain links or attachments that, when clicked, download malware.
- **Mitigation**:
    - **Awareness Training**: Educate employees on how to recognize phishing emails and suspicious links.
    - **Email Filtering**: Implement email filtering solutions to block known phishing attempts.
    - **Multi-Factor Authentication (MFA)**: Even if credentials are stolen, MFA provides an additional layer of security.

Vishing (Voice Phishing)

- **Definition**: Vishing is a phishing attack carried out via voice communication, such as phone calls or voicemail, where attackers impersonate legitimate entities to extract sensitive information.

- **Security Implications**:
    - **Impersonation**: Attackers may impersonate banks, government agencies, or trusted companies to gain access to personal or financial information.
- **Mitigation**:
    - **Caller Verification**: Always verify the identity of callers before sharing any sensitive information.
    - **No Disclosure**: Never disclose sensitive information over the phone unless you can verify the caller's identity independently.

Smishing (SMS Phishing)

- **Definition**: Smishing is a phishing attack carried out through SMS (text messages) to lure users into providing sensitive data.
- **Security Implications**:
    - **Malicious Links**: Smishing may contain links leading to fake websites that capture personal data or download malware.
- **Mitigation**:
    - **Avoid Clicking Links**: Do not click on links or download attachments from unsolicited SMS messages.
    - **Mobile Security**: Use mobile security apps that detect malicious links and phishing attempts.

Misinformation/Disinformation

- **Definition**: Misinformation and disinformation are deliberate attempts to mislead or manipulate individuals or the public, often to sway opinions or destabilize organizations.
- **Security Implications**:
    - **Reputation Damage**: Misinformation can damage the reputation of individuals or organizations.
    - **Manipulation**: Disinformation can be used to manipulate public opinion or influence elections or corporate decisions.
- **Mitigation**:
    - **Fact-Checking**: Encourage fact-checking, especially during crises or sensitive situations.
    - **Information Control**: Implement strong internal controls and public communication strategies.

Impersonation

- **Definition**: Impersonation involves an attacker pretending to be someone else, often to gain access to systems, data, or financial resources.
- **Security Implications**:
    - **Unauthorized Access**: Attackers gain access to sensitive systems or information by impersonating authorized individuals.
- **Mitigation**:
    - **Identity Verification**: Use identity verification methods such as strong authentication, biometrics, or security tokens.
    - **Behavioral Analysis**: Monitor for unusual behavior or access patterns.

Business Email Compromise (BEC)

- **Definition**: BEC is a type of social engineering attack that targets businesses to defraud them, typically involving email fraud where attackers impersonate high-ranking executives to authorize financial transactions or gain access to sensitive data.
- **Security Implications**:
    - **Financial Loss**: Companies can lose significant amounts of money if the fraud is successful.
- **Mitigation**:
    - **Internal Controls**: Implement multi-step approval processes for financial transactions.
    - **Email Authentication**: Use email security protocols like **DMARC** to prevent email spoofing.

Pretexting

- **Definition**: Pretexting involves creating a fabricated scenario to obtain information from a target, such as pretending to be a co-worker or IT support.
- **Security Implications**:
    - **Information Theft**: Attackers may gain unauthorized access to confidential data by convincing victims to provide information under false pretenses.
- **Mitigation**:
    - **Security Policies**: Establish strict verification protocols for sharing sensitive information.
    - **Training**: Educate employees on pretexting tactics.

Watering Hole Attack

- **Definition**: A watering hole attack occurs when attackers compromise a website that is frequented by their target audience, hoping to infect the users with malware.
- **Security Implications**:
    - **Targeted Malware Delivery**: Users visiting the compromised site unknowingly download malware, which is then used for data exfiltration or system compromise.
- **Mitigation**:
    - **Website Monitoring**: Regularly monitor and secure frequently visited sites.
    - **Endpoint Protection**: Use antivirus and endpoint protection software that detects and prevents malware downloads.

Brand Impersonation

- **Definition**: Attackers impersonate legitimate brands to deceive users into believing they are interacting with a trusted organization, often leading to credential theft or financial fraud.
- **Security Implications**:
    - **User Trust Exploitation**: Users trust familiar brands, so impersonating them increases the likelihood of success for social engineering attacks.
- **Mitigation**:
    - **Brand Protection**: Monitor for domain names or accounts that mimic your brand (e.g., **typosquatting** or fake social media profiles).
    - **Awareness**: Educate users on identifying official brand communications.

Typosquatting

- **Definition**: Typosquatting involves registering domain names similar to legitimate ones but with slight misspellings, hoping that users make a typo and visit the fake site.

- **Security Implications**:
    - **Fake Websites**: Users who mistype a URL might end up on a malicious site that looks similar to the real one, leading to credential theft or malware infection.
- **Mitigation**:
    - **Domain Monitoring**: Monitor for suspicious domain registrations that resemble your brand.
    - **User Education**: Encourage users to double-check URLs and avoid clicking on links from untrusted sources.

2.3 Explain various types of vulnerabilities

Vulnerabilities are weaknesses or flaws in a system, application, or process that can be exploited by an attacker. These vulnerabilities can exist in **applications**, **operating systems (OS)**, **web-based platforms**, or **hardware**. Below is a breakdown of each type of vulnerability with explanations, examples, and mitigation strategies.

1. Application-based Vulnerabilities

Application vulnerabilities often occur due to flawed programming or poor design. They can lead to various types of attacks on software applications, databases, and services.

a. Memory Injection

- **Definition**: Memory injection occurs when malicious data is inserted into the memory of a process, allowing an attacker to manipulate the behavior of the application or system.
- **Security Implications**:
    - Attackers can exploit vulnerabilities like buffer overflows to inject code or commands that the application will execute. This could lead to **remote code execution** (RCE), privilege escalation, or system crashes.
    - **Examples**: **Malware** that exploits memory corruption to execute arbitrary code within the system.
- **Mitigation**:
    - Use **Data Execution Prevention (DEP)** and **Address Space Layout Randomization (ASLR)** to make it more difficult for attackers to predict memory locations.
    - **Input Validation**: Ensure that only validated input is allowed to interact with system memory.

b. Buffer Overflow

- **Definition**: A buffer overflow happens when a program writes more data to a buffer than it can hold, causing adjacent memory to be overwritten.
- **Security Implications**:
    - **Attackers** can exploit buffer overflow vulnerabilities to inject malicious code or commands into the program's memory, often resulting in the program executing harmful instructions or crashing.
    - **Examples**: **Code injection attacks** that overwrite function pointers or return addresses in stack-based buffers.
- **Mitigation**:
    - **Bounds Checking**: Always check that the input data fits within the buffer size.
    - Use **Safe Programming Techniques** like stack canaries and bounds-checked libraries to protect against overflows.

c. Race Conditions

- **Definition**: A race condition occurs when two or more processes access shared data or resources concurrently and attempt to change it at the same time, leading to unexpected behavior.
- **Security Implications**:
    - Attackers can exploit race conditions to gain unauthorized access or elevate privileges.
- **Types of Race Conditions**:
    - **Time-of-Check (TOC)**: The time when a system checks the state of a resource before performing an action.
    - **Time-of-Use (TOU)**: The time when the action is performed, typically after the TOC.
- If the resource's state changes between the TOC and TOU, an attacker can manipulate the system's behavior.
- **Mitigation**:
    - **Atomic Transactions**: Use atomic operations and locks to ensure resources are not accessed concurrently.
    - **Proper Synchronization**: Ensure that shared resources are properly locked and synchronized before use.

d. Malicious Update

- **Definition**: Malicious updates involve an attacker delivering a compromised update or patch to software, tricking the user into applying it.
- **Security Implications**:
    - These updates may contain malware or backdoors that allow attackers to compromise the system.
    - **Example**: **APT** groups using fake software updates to install malicious payloads on target systems.
- **Mitigation**:
    - **Digital Signatures**: Ensure that software updates are signed and verified before being applied.
    - **Update Channels**: Use secure and trusted update channels to ensure the authenticity of updates.

2. Operating System (OS)-based Vulnerabilities

OS-based vulnerabilities arise from flaws in the operating system, which can provide attackers with unauthorized access to the system or escalate privileges.

a. OS Misconfigurations

- **Definition**: Incorrect OS configurations can expose services or ports that should be closed, provide weak user permissions, or allow insecure communication.
- **Security Implications**:
    - **Privilege Escalation**: Misconfigured OS settings can allow attackers to gain higher privileges than initially granted.
- **Mitigation**:
    - Regularly **audit system configurations** to ensure compliance with security best practices.
    - Disable unnecessary **services** and **ports**, and use strong access controls.

3. Web-based Vulnerabilities

Web-based vulnerabilities are flaws in websites or web applications that attackers can exploit to gain unauthorized access, steal data, or compromise systems.

a. Structured Query Language Injection (SQLi)

- **Definition**: SQL injection occurs when an attacker can insert or manipulate SQL queries, which the web application executes, potentially exposing or altering the database.
- **Security Implications**:
    - Attackers can retrieve sensitive data (e.g., usernames, passwords), modify records, or execute administrative operations on the database.
    - **Examples**: **Login bypass**, **data extraction**, and **destructive queries**.
- **Mitigation**:
    - Use **prepared statements** and **parameterized queries** to prevent untrusted data from being interpreted as SQL code.
    - **Input Sanitization**: Properly sanitize and validate all user input before using it in database queries.

b. Cross-Site Scripting (XSS)

- **Definition**: XSS is a vulnerability that allows an attacker to inject malicious scripts into web pages viewed by other users.
- **Security Implications**:
    - **Session Hijacking**: Attackers can steal cookies or session tokens.
    - **Malicious Content**: Injected scripts can redirect users to malicious sites or execute arbitrary actions on their behalf.
- **Mitigation**:
    - **Output Encoding**: Encode user input before rendering it in the browser to prevent it from being interpreted as executable code.
    - **Content Security Policy (CSP)**: Implement a CSP to restrict which resources can be loaded and executed by the browser.

4. Hardware-based Vulnerabilities

Hardware vulnerabilities are flaws in the physical components or firmware of a system that can be exploited by attackers.

a. Firmware Vulnerabilities

- **Definition**: Firmware vulnerabilities arise when there are security flaws in the embedded software that controls hardware devices (e.g., routers, IoT devices).
- **Security Implications**:
    - **Backdoor Access**: Attackers can exploit firmware flaws to gain low-level control over hardware.
    - **Example**: **IoT devices** with outdated firmware that can be easily compromised and used as part of a botnet.
- **Mitigation**:
    - Regularly update device firmware and ensure secure, authenticated updates.

- **Secure Boot**: Use secure boot mechanisms to ensure that only trusted firmware is loaded.

b. End-of-Life (EOL) Hardware

- **Definition**: Hardware that has reached its end-of-life is no longer supported by the manufacturer with security updates or patches.
- **Security Implications**:
  - Attackers can exploit **EOL vulnerabilities** because the manufacturer no longer provides fixes for known security issues.
  - **Example**: **Legacy hardware** like old routers or devices that are no longer supported by security patches.
- **Mitigation**:
  - Replace or upgrade hardware as it reaches its end-of-life.
  - Isolate unsupported devices from the rest of the network to reduce exposure.

c. Legacy Hardware

- **Definition**: Legacy hardware refers to old hardware that is still in use but may not meet modern security standards.
- **Security Implications**:
  - **Outdated Security Features**: Legacy hardware often lacks modern security mechanisms like **encryption** or **access controls**.
  - **Example**: Older **network switches** or **routers** that don't support modern encryption protocols.
- **Mitigation**:
  - **Retire or Replace** outdated hardware with devices that support newer security features.
  - Implement additional **network segmentation** to isolate legacy systems from critical infrastructure.

Virtualization Vulnerabilities

Virtualization allows for the creation of virtual instances of servers, storage devices, and networks, improving flexibility, efficiency, and scalability. However, virtualization can also introduce new vulnerabilities.

1. Virtual Machine (VM) Escape

- **Definition**: VM escape refers to an attack where a malicious VM breaks out of its virtualized environment and gains unauthorized access to the host operating system or other VMs running on the same hypervisor.
- **Security Implications**:
  - **Privilege Escalation**: Once a VM escapes, it can potentially access or compromise other VMs or the host machine, leading to full control over the infrastructure.
  - **Examples**: If an attacker successfully escapes a VM, they could exploit vulnerabilities in the hypervisor or guest operating system to attack other VMs, steal data, or execute commands on the host.
- **Mitigation**:
  - Use **VM isolation** to ensure that VMs are properly sandboxed.
  - **Regularly update** the hypervisor to patch vulnerabilities.
  - **Limit guest access** to sensitive resources on the host system.

2. Resource Reuse

- **Definition**: Resource reuse vulnerabilities occur when virtualized environments improperly share resources, leading to unintended access to or leakage of sensitive information.
- **Security Implications**:
    - **Memory Leaks**: If VMs do not properly manage memory, one VM could potentially access another's memory space.
    - **Data Leakage**: Improper isolation of resources, such as CPUs or storage devices, can lead to the leakage of sensitive data between VMs.
- **Mitigation**:
    - Enforce **resource limits** on VMs to prevent them from using excessive resources that could impact the security of other VMs.
    - Use **trusted hypervisors** and **secure configurations** to isolate resources.

Cloud-Specific Vulnerabilities

Cloud computing environments introduce unique challenges for cybersecurity, especially as companies increasingly rely on third-party providers.

1. Supply Chain Vulnerabilities

- **Definition**: Supply chain vulnerabilities in the cloud arise from external providers (service, hardware, and software) that supply components or services to a cloud environment. These vulnerabilities can be introduced at any point in the supply chain.

a. Service Provider

- **Security Implications**:
    - **Data Breaches**: Cloud service providers may store sensitive data, and any breach in their systems could impact multiple organizations that rely on them.
    - **Service Outages**: Attacks on a service provider could cause downtime or data loss for customers.
- **Mitigation**:
    - **Due Diligence**: Conduct thorough vetting of cloud service providers before entering into contracts.
    - **Service Level Agreements (SLAs)**: Ensure that SLAs specify security expectations and breach notification procedures.

b. Hardware Provider

- **Security Implications**:
    - **Compromised Hardware**: Vulnerabilities in hardware (e.g., compromised chips or devices) can introduce backdoors or weak points in the system that attackers can exploit.
- **Mitigation**:
    - **Trusted Hardware**: Work with verified and trusted hardware providers who comply with industry security standards.
    - **Regular Audits**: Perform regular audits on hardware components for vulnerabilities.

c. Software Provider

- **Security Implications**:

    - **Malware or Backdoors**: If a software provider's product is compromised, it could introduce malware or backdoors into your cloud environment.
- **Mitigation**:
    - **Secure Software Supply Chain**: Use only software from trusted providers, and ensure that software undergoes proper security checks.
    - **Regular Updates and Patching**: Always keep software updated to ensure vulnerabilities are patched.

Cryptographic Vulnerabilities

Cryptography is essential for securing data, but weaknesses in cryptographic algorithms or their implementation can lead to serious security risks.

1. Cryptographic Weaknesses

- **Definition**: Cryptographic vulnerabilities arise when encryption algorithms, keys, or implementations are weak or flawed, allowing attackers to decrypt or manipulate data.
- **Security Implications**:
    - **Weak Algorithms**: Using outdated algorithms like **DES** (Data Encryption Standard) or **MD5** can make the data easily susceptible to attacks like **brute force** or **cryptanalysis**.
    - **Key Management Issues**: Improper key management (e.g., weak key generation, key storage, or lack of key rotation) can expose encrypted data to attacks.
- **Mitigation**:
    - Use strong and up-to-date cryptographic algorithms like **AES** (Advanced Encryption Standard) and **SHA-256**.
    - Implement robust **key management practices**, such as using hardware security modules (HSMs) and enforcing key rotation policies.

Misconfiguration Vulnerabilities

Misconfiguration is one of the most common vulnerabilities in both cloud environments and traditional IT systems. It occurs when systems are not set up or maintained according to best practices or security standards.

- **Definition**: Misconfiguration vulnerabilities occur when systems, networks, or applications are improperly configured, exposing them to unauthorized access or attacks.
- **Security Implications**:
    - **Open Ports and Services**: Misconfigured firewalls may leave ports open or expose services that are unnecessary or vulnerable.
    - **Inadequate Access Controls**: Weak or improperly configured access controls could allow unauthorized users to gain access to critical systems or data.
- **Mitigation**:
    - **Configuration Management**: Follow secure configuration baselines (e.g., **CIS Benchmarks**) and conduct regular configuration audits.
    - **Automated Tools**: Use configuration management tools to automate and standardize the secure setup of systems.

Mobile Device Vulnerabilities

Mobile devices face unique security challenges, as they are often used outside of the organization's controlled network, which can introduce risks related to **side loading**, **jailbreaking**, and other issues.

1. Side Loading

- **Definition**: Side loading occurs when users install applications from unofficial sources, bypassing app store security checks.
- **Security Implications**:
    - **Malware**: Side-loaded apps may contain malware or malicious code that can compromise the device or steal data.
- **Mitigation**:
    - **App Store Restrictions**: Only allow apps from trusted app stores (e.g., Apple App Store, Google Play) to be installed.
    - **Mobile Device Management (MDM)**: Use MDM solutions to control app installations and prevent unauthorized apps from being installed.

2. Jailbreaking

- **Definition**: Jailbreaking refers to the process of removing restrictions on iOS devices, allowing them to run unapproved applications and make system-level changes.
- **Security Implications**:
    - **Loss of Security Features**: Jailbreaking removes security mechanisms like code signing, which can expose the device to malware and unauthorized access.
    - **Void Warranty**: Jailbreaking a device often voids the manufacturer's warranty.
- **Mitigation**:
    - **Do Not Jailbreak**: Avoid jailbreaking mobile devices, and ensure that employees understand the security risks.
    - **Security Policies**: Implement policies to prevent jailbroken devices from accessing the organization's network.

Zero-day Vulnerabilities

A **zero-day vulnerability** is a flaw that is unknown to the software vendor or security community and can be exploited by attackers before it is patched.

- **Definition**: Zero-day vulnerabilities are critical flaws in software or hardware that have not yet been discovered or addressed by the vendor.
- **Security Implications**:
    - **Immediate Exploitation**: Attackers can exploit zero-day vulnerabilities before they are even detected or fixed by the vendor, leading to severe security breaches.
    - **Examples**: **Stuxnet** was an attack that exploited multiple zero-day vulnerabilities in Windows to damage Iran's nuclear program.
- **Mitigation**:
    - **Threat Intelligence**: Use threat intelligence feeds and participate in communities that monitor zero-day threats.
    - **Patch Management**: Quickly apply patches and updates as soon as they are released.
    - **Intrusion Detection Systems**: Implement intrusion detection and prevention systems (IDPS) to monitor for abnormal behaviors that could indicate an exploit.

2.4 Given a scenario, analyze indicators of malicious activity

Malicious activity can take many forms, from malware infections to network attacks. By recognizing indicators of such activities, you can identify and respond to potential threats more effectively. Below, we will discuss the **indicators of various types of attacks**.

Malware Attacks

Malware is software intentionally designed to cause damage, disrupt operations, or gain unauthorized access to systems.

1. Ransomware

- **Definition**: Ransomware is a type of malware that encrypts files or locks users out of their systems, demanding payment (often in cryptocurrency) to restore access.
- **Indicators**:
    - **File encryption**: Files are encrypted with an extension that's not normally seen.
    - **Ransom note**: A message appears on the system demanding payment for decryption keys.
    - **Slow system performance**: The encryption process consumes significant resources, slowing down the system.
    - **Inability to access files**: Affected files cannot be opened or accessed without the decryption key.
- **Mitigation**: Backup data regularly, use anti-ransomware software, and implement strong access controls.

2. Trojan

- **Definition**: A Trojan is a type of malware that disguises itself as a legitimate file or program to gain access to a victim's system.
- **Indicators**:
    - **Suspicious programs**: Unknown programs or files that appear to be legitimate applications but are malicious.
    - **Unexpected system behavior**: A Trojan might open backdoors, allow remote access, or cause unusual system crashes.
- **Mitigation**: Use updated antivirus software, avoid downloading software from untrusted sources, and ensure proper email filtering.

3. Worm

- **Definition**: A worm is a self-replicating piece of malware that spreads across networks without requiring human interaction, unlike viruses.
- **Indicators**:
    - **High network traffic**: Worms often generate significant traffic as they replicate and spread.
    - **Slow system performance**: Systems may slow down due to the worm's self-replication and network activity.
    - **Security tool alerts**: Antivirus tools may flag unusually high network activity as part of a worm's spread.
- **Mitigation**: Patch systems and applications to prevent exploitation of vulnerabilities and use network segmentation to limit worm spread.

4. Spyware

- **Definition**: Spyware is a type of malware that secretly monitors and collects user information, such as browsing habits, login credentials, or other sensitive data.
- **Indicators**:

    - **Unwanted pop-ups**: Frequent ads or pop-ups appear when using the web.
    - **System slowness**: Spyware consumes resources as it runs in the background, leading to slowdowns.
    - **Unexpected toolbars or browser settings changes**: New toolbars or homepages may appear on the web browser.
- **Mitigation**: Use anti spyware software, avoid downloading software from unknown sources, and regularly review browser settings.

5. Bloatware

- **Definition**: Bloatware is unwanted software that consumes system resources but doesn't provide significant value to the user.
- **Indicators**:
    - **System resource usage**: Increased CPU or RAM usage due to unnecessary programs running in the background.
    - **Slow system performance**: Bloatware causes system slowdown by using up system resources.
- **Mitigation**: Regularly uninstall unnecessary applications, particularly pre-installed ones on devices.

6. Virus

- **Definition**: A virus is a type of malware that attaches itself to a legitimate program or file and spreads when the program is executed.
- **Indicators**:
    - **Corrupted files**: Files are either corrupted or cannot be accessed.
    - **Increased system activity**: Unusual file activity, such as files being deleted or modified unexpectedly.
    - **Pop-up messages or strange behaviors**: These might include error messages, unexpected reboots, or system crashes.
- **Mitigation**: Use antivirus software, keep software updated, and avoid opening suspicious attachments.

7. Keylogger

- **Definition**: A keylogger records keystrokes made by the user, allowing attackers to capture sensitive information like usernames, passwords, and credit card details.
- **Indicators**:
    - **Unexpected system behavior**: Programs running in the background without user knowledge.
    - **Unusual network traffic**: Keyloggers may send the captured data to external servers.
- **Mitigation**: Use strong passwords, implement multi-factor authentication (MFA), and install anti-keylogging software.

8. Logic Bomb

- **Definition**: A logic bomb is malware that activates when certain conditions or triggers are met, such as a specific date or action.
- **Indicators**:
    - **Sudden system behavior**: Systems behave abnormally at a particular time or after a specific event.

    - **Delayed activation**: The malware may remain dormant for a long time before activating.
- **Mitigation**: Perform regular system audits and use endpoint protection solutions to detect abnormal behavior.

9. Rootkit

- **Definition**: A rootkit is malware that gains privileged access to a system and hides its existence by modifying the system's kernel or operating system.
- **Indicators**:
    - **Unusual file activity**: Files are hidden or altered without the user's knowledge.
    - **Unresponsive system**: The system may become unresponsive or perform unusually slow.
    - **Anti-virus failure**: Traditional antivirus software may fail to detect rootkits because they operate at the kernel level.
- **Mitigation**: Use specialized rootkit detection tools and implement strict access controls and monitoring.

Physical Attacks

Physical attacks involve attackers directly interacting with hardware or systems to gain unauthorized access.

1. Brute Force

- **Definition**: A brute-force attack involves an attacker trying all possible password combinations until the correct one is found.
- **Indicators**:
    - **Multiple failed login attempts**: A sudden increase in failed logins could indicate a brute-force attack.
    - **Slow system response**: Systems might slow down as they handle many failed authentication attempts.
- **Mitigation**: Implement account lockout policies, use CAPTCHA mechanisms, and enforce strong password policies.

2. Radio Frequency Identification (RFID) Cloning

- **Definition**: RFID cloning involves duplicating the information stored on an RFID-enabled device, such as an access card, and using it to gain unauthorized access.
- **Indicators**:
    - **Unexplained access events**: Access logs may show entries that don't match expected users.
    - **Cloning attempts**: Detection systems might flag attempts to access RFID readers at unusual times.
- **Mitigation**: Use encrypted RFID tags, implement multi-factor authentication for access, and limit the range of RFID readers.

3. Environmental

- **Definition**: Environmental attacks exploit physical factors like temperature, humidity, or physical disruptions to damage hardware or data.
- **Indicators**:

    - **Overheating hardware**: Sudden temperature changes or failure of cooling systems might indicate an environmental attack.
    - **Physical damage**: Signs of physical tampering with equipment.
- **Mitigation**: Use environmental monitoring systems and ensure proper physical security measures (e.g., access control).

Network Attacks

Network attacks involve manipulating network traffic to gain unauthorized access or disrupt services.

1. Distributed Denial-of-Service (DDoS)

- **Definition**: A DDoS attack involves overwhelming a target system with traffic from multiple sources, making the system or network unavailable to legitimate users.
- **Indicators**:
    - **Sudden surge in network traffic**: A sharp increase in inbound traffic from multiple sources can indicate a DDoS attack.
    - **Slow system performance**: Legitimate users may experience slow or no access to the system.
- **Types of DDoS Attacks**:
    - **Amplified**: The attacker sends small requests to a server, which responds with larger amounts of data, overwhelming the target.
    - **Reflected**: The attacker spoofs the victim's IP address and causes other servers to send traffic to the victim, increasing the attack's scale.
- **Mitigation**:
    - Implement **rate-limiting** and use **content delivery networks (CDNs)** to absorb traffic spikes.
    - Use **DDoS protection services** (e.g., Cloudflare, Akamai).

2. Domain Name System (DNS) Attacks

- **Definition**: DNS attacks involve manipulating the DNS records to redirect traffic or deny access to websites.
- **Indicators**:
    - **Unusual redirects**: Users trying to access a website may be redirected to a malicious site.
    - **DNS resolution failure**: Legitimate websites fail to resolve, and users are unable to access them.
- **Mitigation**:
    - Use **DNSSEC** (Domain Name System Security Extensions) to secure DNS transactions.
    - Regularly monitor DNS logs for unauthorized changes.

3. Wireless Attacks

- **Definition**: Wireless attacks target the vulnerabilities inherent in wireless networks, such as Wi-Fi or Bluetooth.
- **Indicators**:
    - **Unusual wireless network activity**: New, unrecognized devices connecting to the network or unusual traffic patterns.

    - **Weak encryption**: Devices connecting using outdated encryption protocols (e.g., WEP).
- **Mitigation**:
    - Use **WPA3** encryption for Wi-Fi and implement **strong network access controls**.

4. On-path Attacks

- **Definition**: On-path (formerly man-in-the-middle) attacks involve intercepting and potentially altering communication between two parties without their knowledge.
- **Indicators**:
    - **Unusual certificate errors**: Users may see warnings about invalid certificates when accessing secure websites.
    - **Unexpected redirects or behaviors**: Users may experience unexpected redirects to malicious websites.
- **Mitigation**:
    - Implement **TLS** (Transport Layer Security) and **SSL** for encryption, and use **certificate pinning**.

5. Credential Replay

- **Definition**: Credential replay attacks involve capturing and reusing valid credentials to gain unauthorized access to a system or network.
- **Indicators**:
    - **Suspicious login activity**: Logs showing successful login attempts from unusual IP addresses or times.
    - **Unexplained access events**: Users logging in at times or locations that don't align with normal behavior.
- **Mitigation**:
    - Use **multi-factor authentication (MFA)** and implement **session expiration** to limit the impact of stolen credentials.

6. Malicious Code

- **Definition**: Malicious code includes any software or script designed to harm or exploit systems.
- **Indicators**:
    - **Antivirus alerts**: Detection of known malicious code signatures.
    - **Unusual system behavior**: Unexpected changes in file integrity, new processes running in the background.
- **Mitigation**:
    - Regularly scan for malware, use **intrusion detection systems (IDS)**, and apply patches to known vulnerabilities.

Application Attacks

Application attacks exploit weaknesses in software applications, either by manipulating input or taking advantage of flaws in the application's code.

1. Injection

- **Definition**: Injection attacks occur when an attacker injects malicious code into an application that the application then executes.

- **Common Types**:
    - **SQL Injection (SQLi)**: The attacker inserts malicious SQL code into an input field, which is then executed by the database.
    - **Command Injection**: Malicious commands are injected into the application, which then execute on the server.
    - **XML Injection**: Malicious XML data is injected to manipulate an application.
- **Indicators**: Unexpected results, database errors, or slow performance due to excessive resource use.
- **Mitigation**: Use **parameterized queries**, **input validation**, and **escape special characters** to prevent injection attacks.

2. Buffer Overflow

- **Definition**: A buffer overflow occurs when an application writes more data to a buffer than it can handle, causing data to overwrite adjacent memory, potentially allowing attackers to execute arbitrary code.
- **Indicators**: Application crashes, system slowdowns, or unexpected behaviors like unauthorized access.
- **Mitigation**: Use **bounds checking**, **safe programming techniques**, and modern compiler security features like **stack canaries** to prevent overflows.

3. Replay

- **Definition**: A replay attack occurs when an attacker intercepts valid data transmissions and replays them to gain unauthorized access or perform malicious actions.
- **Indicators**: Unusual or duplicate transactions occurring within a short time frame.
- **Mitigation**: Implement **timestamps**, **nonces**, and **encryption** to ensure data is not replayed successfully.

4. Privilege Escalation

- **Definition**: Privilege escalation involves exploiting a vulnerability to gain higher privileges than originally assigned, often leading to unauthorized access to critical resources.
- **Indicators**: Users or processes gaining access to data or functions they shouldn't have, unusual account activities, or system changes.
- **Mitigation**: Implement **least privilege access**, **regular audits**, and **secure coding practices** to minimize privilege escalation risks.

5. Forgery

- **Definition**: Forgery refers to the creation of fraudulent data, transactions, or documents to deceive the system.
- **Common Examples**: **Email forgery** (spoofing), **web forgery** (creating fake webpages), and **transaction forgery** (fake financial transactions).
- **Indicators**: Suspicious user actions, mismatched or unexpected document or transaction data.
- **Mitigation**: Use **digital signatures**, **email authentication** protocols (e.g., **DKIM**, **SPF**), and **two-factor authentication** (2FA) to prevent forgery.

6. Directory Traversal

- **Definition**: Directory traversal allows an attacker to access files and directories that are outside the intended scope of an application by manipulating file paths.

- **Indicators**: Attempted access to restricted files or directories, unexpected file system access.
- **Mitigation**: Validate input to ensure file paths are restricted and use **chroot** or similar methods to restrict file access.

Cryptographic Attacks

Cryptographic attacks exploit weaknesses in cryptographic algorithms, protocols, or key management.

1. Downgrade

- **Definition**: A downgrade attack forces the system to use a weaker version of a protocol or encryption, allowing attackers to exploit vulnerabilities in the older version.
- **Indicators**: Unexpected fallbacks to less secure cryptographic protocols (e.g., TLS 1.2 instead of TLS 1.3).
- **Mitigation**: Use **cryptographic version negotiation**, enforce **strong protocols** (e.g., TLS 1.3), and **disable old cipher suites**.

2. Collision

- **Definition**: A collision attack occurs when two different inputs produce the same hash value, undermining the integrity of a cryptographic function.
- **Indicators**: Unexpected hash matches for different data sets.
- **Mitigation**: Use **strong hash functions** (e.g., **SHA-256**, **SHA-3**) and avoid deprecated hash algorithms (e.g., **MD5**).

3. Birthday

- **Definition**: A birthday attack is based on the **birthday paradox**, where finding two inputs that hash to the same value becomes more likely with a larger number of inputs.
- **Indicators**: Detection of hash collisions or suspicious changes in data integrity.
- **Mitigation**: Use **stronger hash functions** with larger hash lengths (e.g., **SHA-256** or **SHA-512**) and implement **salt** to increase security.

Password Attacks

Passwords are often the primary method of authentication, making them a frequent target for attackers. Understanding password attacks is critical for securing systems.

1. Spraying

- **Definition**: Password spraying involves using a small set of commonly used passwords against a large number of accounts to avoid account lockouts.
- **Indicators**: Unusual login activity, many failed login attempts across multiple accounts, or increased failed logins from a single IP address.
- **Mitigation**: Implement **account lockout policies**, use **multi-factor authentication (MFA)**, and enforce **strong password policies**.

2. Brute Force

- **Definition**: A brute-force attack involves systematically trying all possible password combinations until the correct one is found.

- **Indicators**: Excessive login attempts, slow system performance due to authentication overload, or a flood of failed login attempts.
- **Mitigation**: Use **strong passwords**, **account lockout mechanisms**, **MFA**, and **CAPTCHA** to prevent automated attacks.

Indicators of Malicious Activity

Understanding **indicators of malicious activity** helps in detecting and mitigating attacks early. Here are some key indicators to watch for:

1. Account Lockout

- **Definition**: Multiple failed login attempts may trigger an account lockout mechanism, preventing further login attempts for a specified period.
- **Indicators**: Accounts locking out after multiple failed login attempts, especially for critical systems or services.
- **Mitigation**: Use **account lockout policies**, and monitor logs for suspicious login patterns.

2. Concurrent Session Usage

- **Definition**: Multiple sessions being used simultaneously by the same user account from different locations or devices.
- **Indicators**: Unexpected concurrent sessions, particularly when users are supposed to be logged in from a single location.
- **Mitigation**: Monitor **session logs**, enforce **IP-based session control**, and use **MFA**.

3. Blocked Content

- **Definition**: Certain content or activities may be blocked by security software, indicating malicious attempts to execute or access restricted resources.
- **Indicators**: Alerts from firewalls, proxies, or endpoint protection software blocking suspicious content or network traffic.
- **Mitigation**: Implement **web filtering**, **email filtering**, and use endpoint protection software to block malicious content.

4. Impossible Travel

- **Definition**: Impossible travel occurs when a user's account is used from two geographically distant locations within a short period, making it impossible for the same user to be at both locations.
- **Indicators**: Login events from geographically distant locations in a short time span.
- **Mitigation**: Implement **geofencing** and monitor login locations to detect and alert on impossible travel scenarios.

5. Resource Consumption

- **Definition**: Excessive consumption of system resources (CPU, memory, bandwidth) can indicate a malware infection or a denial-of-service attack.
- **Indicators**: Unexplained spikes in CPU usage, disk space usage, or network bandwidth.
- **Mitigation**: Use **resource monitoring tools** and **intrusion detection systems** to detect abnormal consumption.

6. Resource Inaccessibility

- **Definition**: Resources (e.g., files, databases) becoming inaccessible due to malicious activities like ransomware or privilege escalation.
- **Indicators**: Inability to access or retrieve important files, or suspicious behavior around resource permissions.
- **Mitigation**: Implement **backup** and **disaster recovery plans**, use **file integrity monitoring** systems, and enforce **strong access controls**.

7. Out-of-Cycle Logging

- **Definition**: Logging events occurring outside the expected intervals, possibly due to attackers attempting to cover their tracks or trigger specific actions.
- **Indicators**: Logs being generated at unusual times or containing suspicious activities, like failed login attempts or system changes.
- **Mitigation**: Implement **continuous monitoring**, maintain proper **log management** practices, and use **SIEM** systems to analyze logs.

8. Published/Documented

- **Definition**: Published or documented vulnerabilities, such as those listed in CVEs (Common Vulnerabilities and Exposures), can be exploited by attackers if not patched in time.
- **Indicators**: Unpatched systems or outdated software with publicly known vulnerabilities.
- **Mitigation**: Stay up-to-date with **security patches** and **vulnerability management** practices to address documented vulnerabilities.

9. Missing Logs

- **Definition**: The absence of logs that should normally be generated, possibly indicating attempts to cover tracks after malicious activity.
- **Indicators**: Missing logs or gaps in the timeline of security events, which could indicate tampering.
- **Mitigation**: Use **centralized logging** solutions and implement **log integrity monitoring** to ensure logs are intact and secure.

2.5 Explain the purpose of mitigation techniques used to secure the enterprise

1. Segmentation

Definition: Network segmentation involves dividing a network into smaller, isolated segments to improve security, control traffic, and limit the reach of attacks.

- **Purpose**: Segmentation helps restrict access to sensitive data and applications, minimizing the attack surface by limiting the number of systems that can communicate directly with each other.
- **Examples**:
    - **VLANs (Virtual Local Area Networks)**: Used to segment traffic between different departments or business units to prevent unauthorized access.
    - **DMZ (Demilitarized Zone)**: A network area where publicly accessible services (like web servers or email servers) are placed, separated from the internal network.
- **Mitigation Benefits**:
    - **Containment of Attacks**: If an attacker compromises one segment, the damage can be contained within that segment, preventing lateral movement within the network.

    - **Improved Control**: More granular control over which systems can communicate with each other.

2. Access Control

Definition: Access control is a fundamental security measure that restricts access to resources based on predefined policies. It ensures that only authorized users or systems can access specific data or services.

a. Access Control List (ACL)

- **Definition**: An ACL is a set of rules used to control access to a system, file, or network resource based on the source IP address, protocol type, or other factors.
- **Purpose**: ACLs provide fine-grained control over network access by specifying which users or systems are allowed or denied access to resources.
- **Mitigation Benefits**:
    - **Granular Control**: Provides more detailed control over network traffic and access to sensitive resources.
    - **Prevents Unauthorized Access**: Ensures that only authorized users or systems can access critical resources.

b. Permissions

- **Definition**: Permissions determine what actions users or systems can perform on a given resource, such as reading, writing, or executing files.
- **Purpose**: By applying permissions to files, folders, or systems, organizations can ensure that only users with the appropriate rights can perform certain actions.
- **Mitigation Benefits**:
    - **Access Restriction**: Prevents unauthorized users from modifying or accessing sensitive data.
    - **Accountability**: Helps in tracking user actions and determining whether any malicious activities occurred.

3. Application Allow List

Definition: An application allow list (also called a "whitelist") specifies a list of trusted applications that are allowed to run on a system, preventing unauthorized or potentially harmful applications from executing.

- **Purpose**: This mitigation technique helps ensure that only known, approved applications are allowed to execute, reducing the risk of malicious software running on the system.
- **Mitigation Benefits**:
    - **Prevents Malicious Software**: Limits the execution of unauthorized or untrusted software that could be harmful or malicious.
    - **Control Over Installed Applications**: Helps organizations ensure that only necessary and secure applications are installed.

4. Isolation

Definition: Isolation refers to creating a secure environment where systems, applications, or processes are separated to prevent them from affecting each other.

- **Purpose**: Isolation helps reduce the impact of a compromise by ensuring that even if one system or application is attacked, it doesn't affect others.
- **Examples**:
    - **Virtualization**: Running multiple virtual machines (VMs) on the same hardware to isolate applications and workloads.
    - **Containerization**: Isolating applications and their dependencies in containers, preventing them from interfering with each other.
- **Mitigation Benefits**:
    - **Containment of Attacks**: An attack in one isolated environment cannot spread to others.
    - **Increased Security**: Limits the potential damage from security vulnerabilities by segregating critical resources.

5. Patching

Definition: Patching involves updating software, firmware, or operating systems to fix vulnerabilities, improve functionality, and prevent exploits.

- **Purpose**: Patching ensures that known vulnerabilities are addressed promptly, minimizing the risk of exploitation by attackers.
- **Mitigation Benefits**:
    - **Prevents Exploits**: Regular patching reduces the chances of vulnerabilities being exploited by attackers.
    - **System Stability**: Ensures that software and systems run smoothly and securely by fixing bugs and security flaws.
- **Best Practices**:
    - **Automated Patching Systems**: Implement automated systems to deploy patches across the enterprise.
    - **Regular Patch Management**: Establish a patch management schedule to ensure timely updates for critical systems.

6. Encryption

Definition: Encryption is the process of converting data into a secure format that can only be read by authorized users with the correct decryption key.

- **Purpose**: Encryption protects data at rest (stored data) and in transit (data being transmitted) by making it unreadable to unauthorized users.
- **Mitigation Benefits**:
    - **Confidentiality**: Ensures that sensitive data, such as financial records or personal information, remains confidential.
    - **Data Integrity**: Prevents unauthorized modifications to data during transmission.
- **Examples**:
    - **SSL/TLS**: Encrypted web traffic, ensuring secure communications between a browser and a server.
    - **Full Disk Encryption (FDE)**: Encrypts the entire disk on a device, ensuring that data is protected even if the device is lost or stolen.

7. Monitoring

Definition: Monitoring refers to the continuous surveillance of systems, networks, and applications to detect security events, vulnerabilities, or anomalies.

- **Purpose**: Monitoring helps organizations detect malicious activity early, allowing for prompt response and mitigation.
- **Mitigation Benefits**:
    - **Early Detection**: Enables the identification of security incidents or unusual activity before they escalate into serious threats.
    - **Continuous Improvement**: Helps security teams understand normal behavior and identify deviations from the norm that could indicate a breach.
- **Examples**:
    - **Intrusion Detection Systems (IDS)**: Monitors network traffic for signs of malicious activity.
    - **Security Information and Event Management (SIEM)**: Aggregates and analyzes security data to provide real-time alerts.

8. Least Privilege

Definition: The principle of least privilege states that users, applications, and systems should have the minimum level of access necessary to perform their job functions.

- **Purpose**: Reduces the risk of unauthorized access to sensitive resources by ensuring users only have access to what they absolutely need.
- **Mitigation Benefits**:
    - **Minimise Attack Surface**: Limits the potential damage an attacker can cause if they compromise a user account.
    - **Better Accountability**: Easier to track and control what actions users or processes can perform.
- **Examples**:
    - **Role-Based Access Control (RBAC)**: Grants permissions based on a user's role within the organization.
    - **Temporary Elevation**: Allowing users to elevate privileges only when absolutely necessary, and for a limited time.

9. Configuration Enforcement

Definition: Configuration enforcement involves ensuring that systems and applications are configured securely and that configurations adhere to organizational or industry best practices.

- **Purpose**: Ensures that systems are set up and maintained with secure settings, preventing security vulnerabilities due to misconfiguration.
- **Mitigation Benefits**:
    - **Consistency**: Ensures that systems are configured securely across the enterprise, reducing the risk of insecure configurations.
    - **Compliance**: Helps organizations meet regulatory requirements by ensuring systems are configured to specific standards.
- **Examples**:
    - **Configuration Management Tools**: Tools like **Puppet** and **Ansible** enforce security configurations across all systems.
    - **Automated Compliance Checks**: Regular scans to ensure systems comply with secure baselines.

10. Decommissioning

Definition: Decommissioning refers to the process of securely retiring systems, software, or hardware when they are no longer needed or are being replaced.

- **Purpose**: Ensures that no sensitive data remains on systems that are no longer in use and that they are disposed of securely.
- **Mitigation Benefits**:
    - **Prevents Data Exposure**: Ensures that any residual data on decommissioned systems is securely erased to prevent data breaches.
    - **Reduces Attack Surface**: Retired systems are no longer vulnerable to attacks.
- **Examples**:
    - **Data Wiping**: Using specialized tools to completely erase all data from hard drives and storage devices before disposal.
    - **Secure Disposal**: Physically destroying hardware (e.g., shredding hard drives) to prevent data retrieval.

Hardening Techniques

Hardening refers to the process of securing a system by reducing its surface of vulnerability and increasing its resistance to attacks. The following techniques are part of system hardening to make it more robust against security threats.

1. Encryption

Definition: Encryption is the process of converting data into a secure format that can only be read or decrypted by authorized parties with the correct decryption key.

- **Purpose**: Encryption protects data at rest (stored data) and in transit (data being transmitted over networks), ensuring that sensitive information, such as passwords, personal data, and financial records, is protected from unauthorized access or interception.
- **Types of Encryption**:
    - **Symmetric Encryption**: Uses the same key for both encryption and decryption (e.g., **AES**, **3DES**).
    - **Asymmetric Encryption**: Uses a pair of keys, one for encryption (public key) and one for decryption (private key) (e.g., **RSA**, **ECC**).
- **Use Cases**:
    - **Full Disk Encryption (FDE)**: Encrypts the entire hard drive to protect sensitive data in case the device is lost or stolen.
    - **Secure Communication**: Protocols like **SSL/TLS** ensure that data transmitted over the internet is encrypted, protecting it from interception.
- **Mitigation Benefits**:
    - **Confidentiality**: Ensures sensitive data remains private and protected.
    - **Data Integrity**: Ensures that the data has not been tampered with during transmission.
- **Advanced Concepts**:
    - **Key Management**: Proper management of encryption keys is crucial. If encryption keys are poorly managed, they can be exposed, rendering the encryption ineffective.
    - **End-to-End Encryption (E2EE)**: Ensures that data is encrypted at the source and only decrypted by the intended recipient, offering robust protection against unauthorized access.

2. Installation of Endpoint Protection

Definition: Endpoint protection involves securing end-user devices like laptops, desktops, mobile phones, and servers from security threats by installing software solutions that detect and block malware, ransomware, viruses, and other types of attacks.

- **Purpose**: Protects endpoints (the entry points to the network) from malware, malicious activity, and unauthorized access. Endpoints are often targeted by attackers, making them crucial components of network security.
- **Components of Endpoint Protection**:
    - **Antivirus/Antimalware**: Software designed to detect and remove malicious software, such as viruses and trojans.
    - **Firewall**: Monitors and controls incoming and outgoing network traffic based on predetermined security rules.
    - **Behavioral Analysis**: Monitors applications and processes for abnormal behavior indicative of malware or malicious activities.
    - **Application Control**: Allows only approved applications to run on the endpoint, preventing malicious software from executing.
- **Mitigation Benefits**:
    - **Malware Detection and Removal**: Scans and removes malware from endpoints, protecting against infections.
    - **Proactive Threat Defense**: Uses heuristics and behavioral analysis to detect new or unknown threats.
- **Advanced Concepts**:
    - **Cloud-Based Endpoint Protection**: Centralized endpoint protection services that can be easily updated and managed across a large enterprise.
    - **Zero Trust Architecture**: Endpoint protection becomes more effective within a **zero trust** model where no device is inherently trusted, and each one must be verified continuously.

3. Host-Based Firewall

Definition: A host-based firewall is a security system installed on individual devices (hosts) that monitors and controls incoming and outgoing network traffic to and from the device based on predefined security rules.

- **Purpose**: It provides an additional layer of security by filtering network traffic before it can reach the system's resources. Host-based firewalls are especially useful for protecting endpoints from external threats, such as unauthorized access.
- **How It Works**:
    - **Packet Filtering**: Inspects packets of data based on rules set by the system administrator.
    - **Stateful Inspection**: Monitors the state of active connections and allows or blocks traffic based on connection state and rules.
- **Mitigation Benefits**:
    - **Blocking Unauthorized Access**: Prevents unauthorized applications or services from communicating with the device.
    - **Granular Control**: Provides fine-grained control over which applications and services can communicate over the network.
- **Advanced Concepts**:
    - **Adaptive Firewalls**: Firewalls that can learn from traffic patterns and automatically adjust rules to block unusual or dangerous traffic.

4. Host-Based Intrusion Prevention System (HIPS)

Definition: HIPS is a security tool designed to monitor the behavior of systems and networks, identifying malicious activities such as unauthorized access or abnormal system behavior.

- **Purpose**: HIPS actively monitors and prevents potentially harmful activities on the host system, such as exploits or malware executions.
- **How It Works**:
    - **Behavioral Analysis**: HIPS detects suspicious behavior by analyzing patterns that differ from the norm, such as excessive CPU usage or file system modifications.
    - **Signature-Based Detection**: Identifies known threats by matching them to predefined signatures.
    - **Heuristic Analysis**: Identifies unknown threats by evaluating suspicious behavior rather than relying solely on known signatures.
- **Mitigation Benefits**:
    - **Real-time Protection**: Actively prevents threats as they attempt to exploit vulnerabilities.
    - **Comprehensive Defense**: Provides protection against a wide range of threats, including exploits, malware, and unauthorized access attempts.
- **Advanced Concepts**:
    - **Network-based vs. Host-based HIPS**: Network-based HIPS focuses on monitoring network traffic, while host-based HIPS focuses on protecting individual systems.

5. Disabling Ports/Protocols

Definition: Disabling unnecessary ports and protocols means turning off or blocking unused communication channels and network protocols to reduce the system's exposure to potential vulnerabilities.

- **Purpose**: Minimizes the attack surface by eliminating unused or unnecessary services and ports that attackers might exploit.
- **Common Ports to Disable**:
    - **Telnet**: An old protocol for remote communication that transmits data in plain text.
    - **FTP**: Unencrypted file transfer protocol, which is vulnerable to interception.
    - **SMBv1**: An outdated protocol that can be exploited in **EternalBlue** attacks.
- **Mitigation Benefits**:
    - **Reduce Attack Surface**: Decreases the number of potential access points available to attackers.
    - **Prevent Unnecessary Exploits**: Many attacks target vulnerable or outdated protocols that have been disabled in a hardened system.
- **Advanced Concepts**:
    - **Port Scanning**: Regularly scan systems to identify open ports and close those that aren't needed.
    - **Access Control Lists (ACLs)**: Use ACLs to control which ports are accessible based on source, destination, and protocol.

6. Default Password Changes

Definition: Default passwords are set by manufacturers and vendors to allow initial access to devices, applications, or systems. Changing these default passwords is a critical security practice.

- **Purpose**: Default passwords are often publicly known or easily guessable. Changing them prevents attackers from gaining unauthorized access to systems.
- **Mitigation Benefits**:

    - **Prevents Unauthorized Access**: Default passwords are often weak and known to attackers, making them a common entry point.
    - **Improves Accountability**: Custom passwords help ensure that access is tied to specific individuals or roles.
- **Advanced Concepts**:
    - **Password Management**: Use password managers to generate and securely store complex passwords for critical systems.

7. Removal of Unnecessary Software

Definition: Unnecessary software refers to applications or programs that are not needed for the system's operation. Removing them reduces the risk of attack.

- **Purpose**: Unnecessary software, especially outdated or unsupported applications, can introduce vulnerabilities or be exploited by attackers.
- **Mitigation Benefits**:
    - **Minimise Attack Surface**: The fewer programs running on a system, the fewer opportunities for attackers to exploit vulnerabilities.
    - **Simpler Patch Management**: Reduces the complexity of keeping systems updated and patched.
- **Advanced Concepts**:
    - **Application Whitelisting**: Allows only approved applications to run, preventing unauthorized or unnecessary software from executing.