CompTIA Security+ SY0-701 Certification

Exam Objectives Explained

Thank you for choosing this guide! This comprehensive resource has been designed to summarize and break down all the exam objectives for the CompTIA Security+ SY0-701 exam.

It provides detailed explanations of each topic, referencing the official study materials and including tips for exam success.

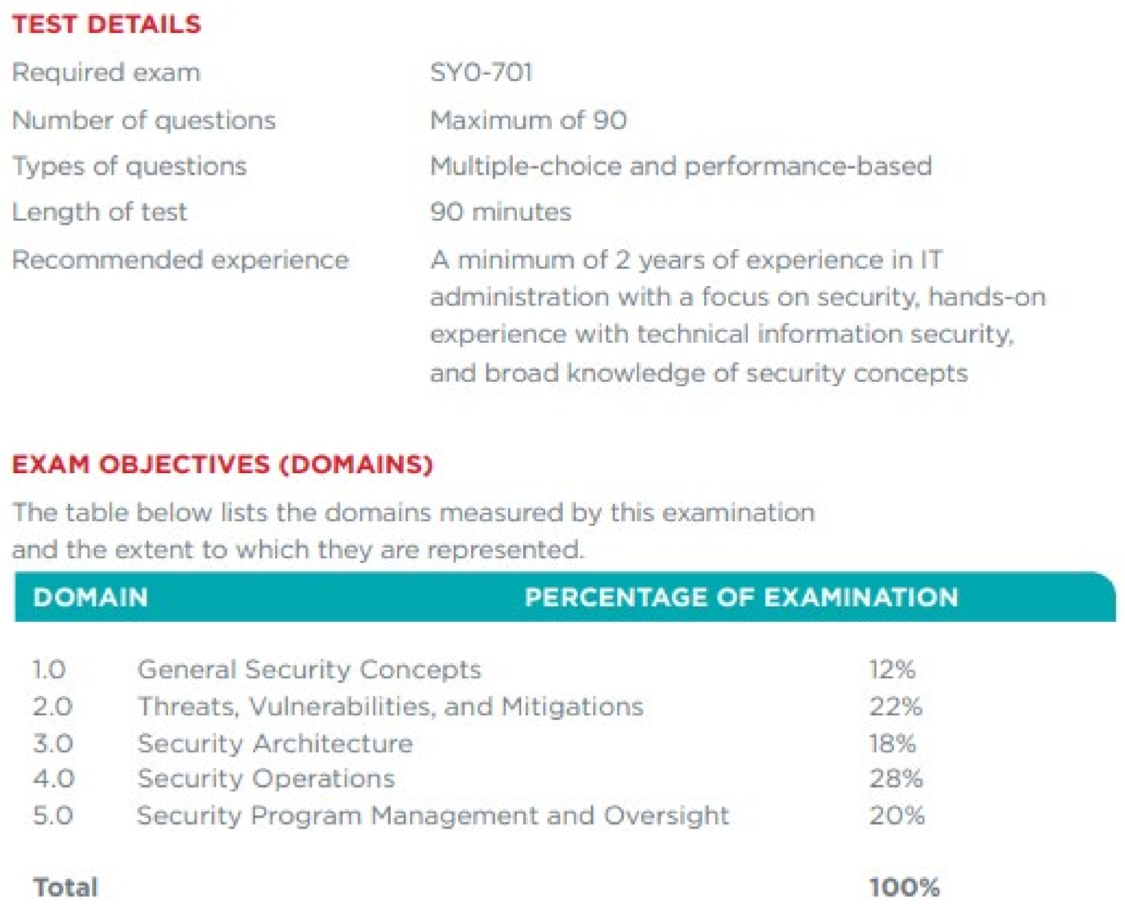
While this guide is an excellent starting point and reference for your exam preparation, please note that it does not substitute the official book and other primary resources. Instead, it serves to clarify and reinforce key concepts, offering a deeper understanding of the exam objectives, backed by official sources and practical advice to help you succeed on your first attempt.

I aim to make these notes accessible to everyone. If you find them helpful, your support through tips/comments or contributions is always appreciated.
It enables me to continue creating high-quality study materials.

✓ Check this post for access to additional valuable resources, such as correct questions with explanations, a glossary, and study methods to help you pass the exam: linkedin.com/in/matteoschirinzi

in Comment & like the post of Sec+ resource with your experience: linkedin.com/in/matteoschirinzi

Thank you for your support and good luck on your journey to earning the CompTIA Security+ certification!



TEST DETAILS

Required exam	SY0-701
Number of questions	Maximum of 90
Types of questions	Multiple-choice and performance-based
Length of test	90 minutes
Recommended experience	A minimum of 2 years of experience in IT administration with a focus on security, hands-on experience with technical information security, and broad knowledge of security concepts

EXAM OBJECTIVES (DOMAINS)

The table below lists the domains measured by this examination and the extent to which they are represented.

DOMAIN		PERCENTAGE OF EXAMINATION
1.0	General Security Concepts	12%
2.0	Threats, Vulnerabilities, and Mitigations	22%
3.0	Security Architecture	18%
4.0	Security Operations	28%
5.0	Security Program Management and Oversight	20%
Total		**100%**

1.0 General Security Concepts

1.1 Compare and contrast various types of security controls

• Categories

- **Technical**: enforce confidentiality, integrity, and availability in the digital space. Examples of technical security controls include firewall rules, access control lists, intrusion prevention systems, and encryption

- **Managerial**: are procedural mechanisms that focus on the mechanics of the risk management process. Examples of administrative managerial controls include periodic risk assessments, security planning exercises, and the incorporation of security into the organization's change management, service acquisition, and project management practices.

- **Operational**: include the processes that we put in place to manage technology in a secure manner. These include user access reviews, log monitoring, and vulnerability management

- **Physical**: are security controls that impact the physical world. Examples of physical security controls include fences, perimeter lighting, locks, fire suppression systems, and burglar alarms

• Control types

- **Preventive**: intend to stop a security issue before it occurs. Firewalls and encryption are examples of preventive controls.

- **Deterrent**: seek to prevent an attacker from attempting to violate security policies. Vicious guard dogs and barbed wire fences are examples of deterrent controls

- **Detective**: identify security events that have already occurred. Intrusion detection systems are detective controls.

- **Corrective**: remediate security issues that have already occurred. Restoring backups after a ransomware attack is an example of corrective control.

- **Compensating**: are controls designed to mitigate the risk associated with exceptions made to a security policy.

- **Directive**: inform employees and others what they should do to achieve security objectives. Policies and procedures are examples of directive controls.

1.2 Summarize fundamental security concepts

• Confidentiality, Integrity, and Availability (CIA): Confidentiality ensures that unauthorized individuals are not able to gain access to sensitive information.
Integrity ensures that there are no unauthorized modifications to information or systems, either intentionally or unintentionally.
Availability ensures that information and systems are ready to meet

• Non-repudiation: means that someone who performed some action, such as sending a message, cannot later deny having taken that action. Digital signatures are a common example of nonrepudiation

- **Authentication, Authorization, and Accounting (AAA):** is a framework used to control access to resources, ensuring security by verifying identities, granting appropriate permissions, and tracking user activities.

  - **Authenticating people**: This process verifies a user's identity before granting access. It typically involves:

    - **Something You Know** (e.g., passwords, PINs)
    - **Something You Have** (e.g., smart cards, hardware tokens)
    - **Something You Are** (e.g., biometrics like fingerprints or facial recognition)
    - **Somewhere You Are** (e.g., geolocation-based authentication)
    - **Something You Do** (e.g., keystroke dynamics)

    Multi-Factor Authentication (MFA) enhances security by requiring two or more factors.

  - **Authenticating systems**: This ensures that devices, applications, and services are verified before they interact with a network. Methods include:

    - **Digital Certificates** (e.g., X.509 certificates used in TLS/SSL)
    - **Public Key Infrastructure (PKI)** for encrypted and trusted communications
    - **Mutual Authentication** (both client and server authenticate each other)
    - **Kerberos & Single Sign-On (SSO)** for secure system authentication

  - **Authorization models**: After authentication, authorization determines what resources a user or system can access. Common models include:

    - **Role-Based Access Control (RBAC)** – Permissions based on roles (e.g., admin, user, guest).
    - **Attribute-Based Access Control (ABAC)** – Access based on attributes (e.g., department, security clearance).
    - **Mandatory Access Control (MAC)** – Centralized control with strict security labels (e.g., military classification).
    - **Discretionary Access Control (DAC)** – Resource owners define access permissions.
    - **Zero Trust Model** – Continuous verification; never assume trust

- **Gap analysis:** During a gap.analysis, the cybersecurity professional reviews the control objectives for a particular organization, system, or service and then examines the controls designed to achieve those objectives. If there are any cases where the controls do not meet the control objective, that is an example of a gap. Gaps identified during a gap analysis should be treated as potential risks and remediate as time and resources permit.

• **Zero Trust:** Unlike traditional "moat and castle" or defense-in-depth designs, Zero Trust presumes that there is no trust boundary and no network edge. Instead, each action is validated when requested as part of a continuous authentication process and access is only allowed after policies are checked, including elements like identity, permissions, system configuration and security status, threat intelligence data review, and security posture.

- **Control Plane**
  - o **Adaptive identity**: leverages context-based authentication that considers data points such as where the user is logging in from, what device they are logging in from, and whether the device meets security and configuration requirements.
  - o **Threat scope reduction**: Limiting the scope of what a subject can do as well or what access is permitted to a resource limit what can go wrong if an issue does occur.
  - o **Policy-driven access control**: **Policy Engines** rely on policies as they make decisions that are then enforced by the Policy Administrator and Policy Enforcement Points.
  - o **Policy Administrator**: are not individuals. Rather they are components that establish or remove the communication path between subjects and resources, including creating session specific authentication tokens or credentials as needed.
  - o **Policy Engine**: make policy decisions based on both rules and external systems (threat intelligence, identity management, and SIEM devices). They use trust algorithm that makes the decision to grant, deny, or revoke access.
- **Data Plane**
  - o **Implicit trust zones**: which allow use and movement once a subject is authenticated by a Zero Trust Policy Engine
  - o **Subject/System**: are the users, services, or systems that request access or attempt to use rights.
  - o **Policy Enforcement Point**: communicate with Policy Administrators to forward requests from subjects and to receive instruction from the policy administrators about connections to allow or end.

• **Physical security:** establish guidelines for securing the physical premises and assets of the organization. This includes security measures like access control systems, surveillance cameras, security personnel, and policies regarding visitor access, protection of sensitive areas, and handling of physical security breaches.

- **Bollards**: are posts or obstacles like that prevent vehicles from moving through an area

- **Access control vestibule** uses a pair of doors. When an individual enters, the first door must be closed and secured before the second door can be opened.

- **Fencing**: *Fences* act as a deterrent by both making it look challenging to access a facility

- **Video surveillance**: allowing security practitioners and others to observe what is happening in real time and to capture video footage of areas for future use

- **Security guard**

- **Access badge**

- **Lighting**: Bright lighting that does not leave shadowed or dark areas is used to discourage intruders and to help staff feel safer

- **Sensors**

  o **Infrared**: They look for changes in infrared radiation in a room or space and alert when that change occurs.

  o **Pressure**: detect a change in pressure. While not commonly deployed in most environments, they may be used when an organization needs to detect an object being moved or when someone is moving through an area using a pressure-plate or pad.

  o **Microwave**: use a baseline for a room or space that is generated by detecting normal responses when the space is at a baseline. When those responses to the microwaves sent out by the sensor change, they will trigger

  o **Ultrasonic**: sensors can be set off by machinery or other vibrations, and they can have environmental effects on human occupants. Ultrasonic sensors are more commonly used in applications where proximity detection is required.

• **Deception and disruption technology:** network-related tools are those intended to capture information about attackers and their techniques and to disrupt ongoing attacks.

- **Honeypot**: A decoy system designed to attract attackers, monitor their activities, and gather threat intelligence. Can be low or high interaction.

- **Honeynet**: A network of multiple honeypots simulating real infrastructure to study advanced attacks.

- **Honeyfile**: A fake document/file placed in storage to detect unauthorized access, often embedded with tracking mechanisms.

- **Honeytoken**: A deceptive credential, API key, or database entry that triggers alerts when used, revealing potential breaches

1.3 Explain the importance of change management processes and the impact to security.

• Business processes impacting security operation

- **Approval process**: ensures only authorized changes are implemented, reducing unauthorized modifications that could introduce security risks.

- **Ownership**: assigns accountability for changes, ensuring proper management and oversight.

- **Stakeholders**: engages relevant teams (IT, security, compliance) to assess risks and maintain security controls.

- **Impact analysis**: evaluates potential security implications, ensuring changes do not create vulnerabilities.

- **Test results**: confirm that changes are tested in a controlled environment to prevent security failures

- **Backout plan** provides a rollback strategy in case the change causes issues, minimizing downtime and risk

- **Maintenance window**: schedules change during low-impact periods to avoid business disruptions.

- **Standard operating procedure**: consistent documents, repeatable processes to ensure compliance and security, best practices.

• Technical implications

- Allow lists/deny lists

- Restricted activities

- Downtime

- Service restart

- Application restart

- Legacy applications

- Dependencies

• **Documentation:** Keeping documentation current is a crucial step when completing a change

- Updating diagrams

- Updating policies/procedures

• **Version control:** ensures that developers and users have access to the latest versions of software and that changes are carefully managed throughout the release process

1.4 Explain the importance of using appropriate cryptographic solutions.

• **Public key infrastructure (PKI):** The major strength of public key encryption is its ability to facilitate communication between parties previously unknown to each other. This is made possible by the *public key infrastructure (PKI)* hierarchy of trust relationships. These trusts permit combining asymmetric cryptography with symmetric cryptography along with hashing and digital certificates, giving us hybrid cryptography.

- **Public key**: A key that is shared openly and used for encrypting data or verifying digital signatures. Anyone can use it to send encrypted messages to the key owner.

- **Private key**: A confidential key, securely held by the owner, used for decrypting data or creating digital signatures. It must remain secret to maintain security.

- **Key escrow**: *Key escrow* systems address this situation by having a third party store a protected copy of the key for use in an emergency. Organizations may have a formal *key recovery* policy that specifies the circumstances under which a key may be retrieved from escrow and used without a user's knowledge.

• **Encryption**

- **Level**

  o **Full-disk**: encrypts the disk and requires that the bootloader or a hardware device provide a decryption key and software or hardware to decrypt the drive for use

  o **Partition**: like FDE but targets a specific partition of a hard drive instead of the entire disk. This allows for more flexibility, as you can choose which parts of your data to encrypt

  o **File**: focuses on individual files. This method allows users to encrypt specific files rather than entire drives or partitions o **Volume**: involves encrypting a set “volume” on a storage device, which could contain several folders and files. This is like a middle ground between partition encryption and file-level encryption

o **Database**: *encryption* targets data at the database level. It's a method used to protect sensitive information stored in a database

o **Record**: is a more granular form of database encryption. It allows individual records within a database

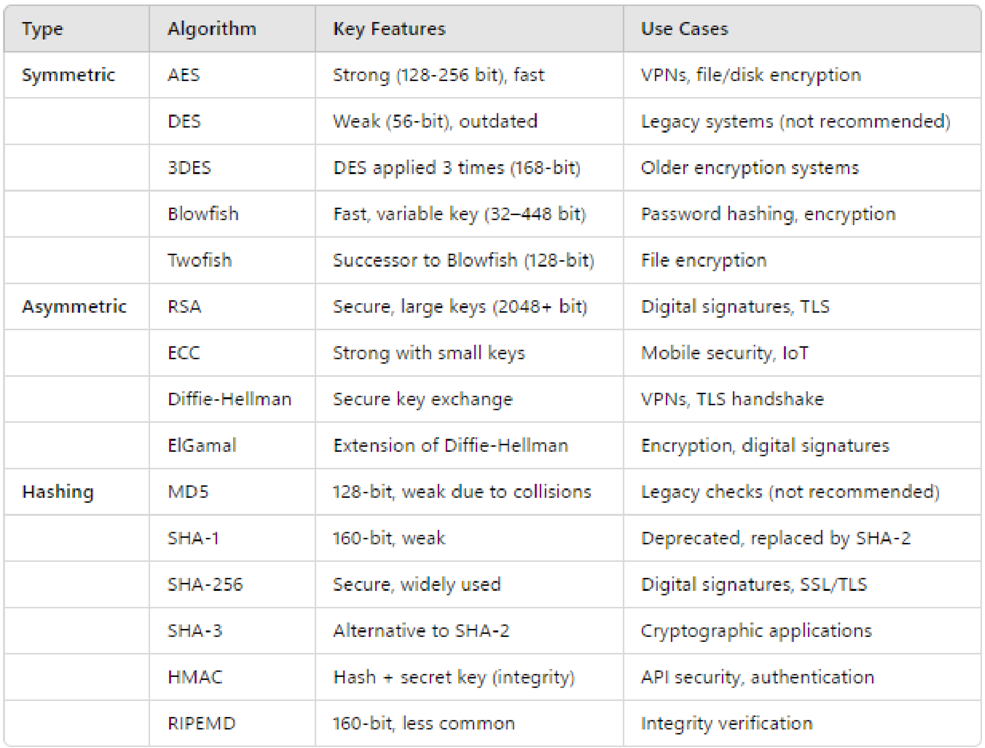
- **Transport/communication**

- **Asymmetric**: also known as *public key algorithms*, provide a solution to the weaknesses of symmetric key encryption. In these systems, each user has two keys: a public key, which is shared with all users, and a private key, which is kept secret and known only to the owner of the key pair. But here's a twist: opposite and related keys must be used in tandem to encrypt and decrypt. In other words, if the public key encrypts a message, then only the corresponding private key can decrypt it, and vice versa.

- **Symmetric**: rely on a “shared secret” encryption key that is distributed to all members who participate in the communications. This key is used by all parties to both encrypt and decrypt messages, so the sender and the receiver both possess a copy of the shared key. The sender encrypts with the shared secret key and the receiver decrypts with it. When large-sized keys are used, symmetric encryption is very difficult to break.

- **Key exchange**: In symmetric algorithms *key exchange* is one of the major problems, The three main methods used to exchange secret keys securely are offline distribution, public key encryption, and the Diffie–Hellman key exchange algorithm.
In asymmetric algorithms users who want to participate in the system simply make their public key available to anyone with whom they want to communicate. There is no method by which the private key can be derived from the public key.

- **Algorithms**



Type	Algorithm	Key Features	Use Cases
Symmetric	AES	Strong (128-256 bit), fast	VPNs, file/disk encryption
	DES	Weak (56-bit), outdated	Legacy systems (not recommended)
	3DES	DES applied 3 times (168-bit)	Older encryption systems
	Blowfish	Fast, variable key (32–448 bit)	Password hashing, encryption
	Twofish	Successor to Blowfish (128-bit)	File encryption
Asymmetric	RSA	Secure, large keys (2048+ bit)	Digital signatures, TLS
	ECC	Strong with small keys	Mobile security, IoT
	Diffie-Hellman	Secure key exchange	VPNs, TLS handshake
	ElGamal	Extension of Diffie-Hellman	Encryption, digital signatures
Hashing	MD5	128-bit, weak due to collisions	Legacy checks (not recommended)
	SHA-1	160-bit, weak	Deprecated, replaced by SHA-2
	SHA-256	Secure, widely used	Digital signatures, SSL/TLS
	SHA-3	Alternative to SHA-2	Cryptographic applications
	HMAC	Hash + secret key (integrity)	API security, authentication
	RIPEMD	160-bit, less common	Integrity verification

- **Key length**: The length of the cryptographic key is perhaps the most important security parameter that can be set at the discretion of the security administrator. It's important to understand the capabilities of your encryption algorithm and choose a key length that provides an appropriate level of protection. This judgment can be made by weighing the difficulty of defeating a given key length (measured in the amount of processing time required to defeat the cryptosystem) against the importance of the data.

• Tools

- **Trusted Platform Module (TPM):** is a hardware-based security feature that helps protect sensitive data by securely storing encryption keys, passwords, and other important information. It provides cryptographic operations like encryption and digital signatures, ensuring that private keys are never exposed to the system’s memory. TPM also helps verify the integrity of the system during boot, protecting against tampering or unauthorized changes. Common uses include full disk encryption (such as BitLocker), secure authentication, and key management, making TPM crucial for enhancing data security and platform integrity.

- **Hardware security module (HSM):** are typically external devices or plugin cards used to create, store, and manage digital keys for cryptographic functions and authentication, as well as to offload cryptographic processing

- **Key management system**: are used to store keys and certificates as well as to manage them centrally. This allows organizations to effectively control and manage their secrets while also enforcing policies.

- **Secure enclave**: a dedicated secure element that is built into a system on chip (SoC) modules. They provide hardware key management, which is isolated from the main CPU, protecting keys throughout their life cycle and usage.

!! Remember that TPMs are used for system security; HSMs are used to create, store, and manage keys for multiple systems; and a KMS is a service used to manage secrets.

- **Obfuscation:** is a concept closely related to confidentiality. It is the practice of making it intentionally difficult for humans to understand data/how code works.

  - **Steganography**: is the practice of using cryptographic techniques to embed or conceal secret messages within another file. It can be used to hide images, text, audio, video, and many other forms of digital content

  - **Tokenization**: replaces sensitive values with a unique identifier using a lookup table. For example, we might replace a widely known value, such as a student ID, with a randomly generated 10-digit number.

  - **Data masking**: partially redacts sensitive information by replacing some or all sensitive fields with blank characters. For example, we might replace all but the last four digits of a credit card number with *

- **Hashing:** uses a hash function to transform a value in our dataset to a corresponding hash value. If we apply a strong hash function to a data element, we may replace the value in our file with the hashed value.

- **Salting:** adds a randomly generated value to each password prior to hashing, prevent Rainbow.table attacks attempt to reverse hashed password values.

- **Digital signatures:** Once you have chosen a cryptographically sound hashing algorithm, you can use it to implement a *digital signature* system. Digital signature infrastructures have two distinct goals:
  - assure the recipient that the message truly came from the claimed sender(**nonrepudiation**)
  - assure the recipient that the message was **not altered** while in transit between the sender and recipient.(**integrity**)

  Digital signature algorithms rely on a combination of the two major concepts already covered in this chapter—public key cryptography and hashing functions.

- **Key stretching:** is used to create encryption keys from passwords in a strong manner. Key stretching algorithms, such as the Password-Based Key Derivation Function v2 (PBKDF2), use thousands of iterations of salting and hashing to generate encryption keys that are resilient against attack.

• **Blockchain:** is a distributed and immutable public ledger, meaning it stores records across many systems worldwide in a way that prevents tampering or destruction. Initially developed for Bitcoin, blockchain allows cryptocurrency transactions to occur without a central authority, with transaction authority distributed among participants.
While cryptocurrency is the most prominent use, blockchain has many other applications. For instance, it could be used to store property ownership records in a transparent, tamper-proof repository, or track supply chains, ensuring product origins and simplifying regulatory oversight for recalls. Blockchain's potential extends far beyond its initial use in cryptocurrencies.

• **Open public ledger:** is a distributed and transparent database where records are stored across multiple systems and are publicly accessible. It allows anyone to view the transactions, but the data is immutable, meaning once recorded, it cannot be altered or deleted. This ensures data integrity and transparency without the need for a central authority.

• **Certificates:** *Digital certificates* provide communicating parties with the assurance that the people they are communicating with truly are who they claim to be. Digital certificates are essentially endorsed copies of an individual's public key.

- **Certificate authorities(CAs):** neutral organizations offer notarization services for digital certificates. To obtain a digital certificate from a reputable CA, you must prove your identity to the satisfaction of the CA.

- **Certificate revocation lists (CRLs):** are maintained by the various certificate authorities and contain the serial numbers of certificates that have been issued by a CA and have been revoked along with the date and time the revocation went into effect

- **Online Certificate Status Protocol (OCSP):** This protocol eliminates the latency inherent in the use of certificate revocation lists by providing a means for real-time certificate verification. When a client receives a certificate, it sends an OCSP request to the CA's OCSP server. The server then responds with a status of good, revoked, or unknown

- **Self-signed**: These certificates won't be trusted by the browsers of external users, but internal systems may be configured to trust the internal CA, saving the expense of obtaining certificates from a third-party CA.

- **Third-party**: is a digital certificate issued by a trusted **Certificate Authority (CA)** that validates the identity of the certificate holder, typically a website or an organization.

- **Root of trust**: Certificate authorities must carefully protect their own private keys to preserve their trust relationships. To do this, they often use an *offline CA* to protect their *root certificate*, the top-level certificate for their entire PKI that serves as the *root of trust* for all certificates issued by the CA.

- **Certificate signing request (CSR) generation**: Once you've satisfied the certificate authority regarding your identity, you provide them with your public key in the form of a Certificate.Signing.Request.(CSR)**.** The CA next creates an X.509 digital certificate containing your identifying information and a copy of your public key. The CA then digitally signs the certificate using the CA's private key and provides you with a copy of your signed digital certificate.

- **Wildcard**: The subject of a certificate may include a wildcard in the certificate name, indicating that the certificate is good for subdomains as well. The *wildcard* is designated by an asterisk character. Wildcard certificates are only good for one level of subdomain. Therefore, the *.certmike.com certificate would not be valid for the www.cissp.certmike.com subdomain

2.0 Threats, Vulnerabilities, and Mitigations

2.1 Compare and contrast common threat actors and motivations.

• Threat actors

- **Nation-state**: government-sponsored cybercriminals, often associated with advanced persistent threats (APTs). APTs are long-term, sophisticated attacks aimed at foreign governments or corporations, utilizing advanced techniques and persistent efforts that can last for years. These attacks typically involve highly skilled attackers with substantial resources—time, money, and labor. The motives behind nation-state attacks are often political or economic. They can involve espionage to gather defense-related information or target intellectual property and other economic assets. The attacks are well-planned and executed by nations with the capabilities to sustain long-term, complex cyber operations.

- **Unskilled attacker**: "Script kiddies" are attackers with limited skills who use automated tools to exploit vulnerabilities. They often target systems opportunistically using tools for DoS attacks, viruses, and ransomware. Despite their lack of expertise, they pose a threat because these tools are easily accessible online, and they target vulnerable systems indiscriminately. Their motivations usually involve proving their skills, and they often target school or university networks. Although they lack resources, their sheer numbers make them a concern for security

- **Hacktivist**: Hacktivists are individuals or groups who use hacking techniques to promote a political or activist cause, such as defacing websites or attacking networks to protest policies they disagree with. They believe their actions serve a greater good, even if illegal. Hacktivists may be skilled or unskilled, and some even work as cybersecurity professionals by day. While many act alone with limited resources, groups like Anonymous coordinate attacks on high-profile targets. These groups are difficult to trace due to their anonymous, decentralized nature

- **Insider threat**: occur when an employee, contractor, vendor, or someone with authorized access intentionally harms an organization. These attacks often aim to disclose confidential information, alter data, or disrupt operations. Insiders can be of any skill level, from unskilled to highly technical, and their motivations vary. Some act for activist reasons, financial gain, or personal grievances like being overlooked for promotion. While insiders typically work alone with limited resources, their access and knowledge of the organization's systems give them an advantage. Behavioral assessments can help identify unusual actions, enabling cybersecurity teams to intervene before the attack escalates.

- **Organized crime**: involves groups motivated by illegal financial gain, such as traditional crime families, outlaw gangs, and the Russian mafia. Unlike hacktivists, these groups are not driven by political causes or proving skills—they aim to generate profit while staying under the radar. Organized crime groups are active in areas like ransomware, data breaches, DDoS attacks, online fraud, and dark web activities. They range from moderately skilled to highly skilled attackers and often have more resources, both time and money, compared to other attackers. Their larger investments are made with the goal of high returns, making them a significant threat in cybercrime.

- **Shadow IT:** occurs when employees use unauthorized technology or services to achieve their goals, often without malicious intent. For example, employees may use personal cloud services like Dropbox to sync work content between devices for increased productivity. While the intent is usually positive, shadow IT poses security risks by placing sensitive information under the control of external vendors. Cybersecurity teams should monitor for their use, as their presence indicates unmet business needs. Engaging with shadow IT users can help identify secure alternatives that meet both business and security requirements.

• Attributes of actors

- **Internal/external**: We most often think about the threat actors who exist outside our organizations: competitors, criminals, and the curious. However, some of the most dangerous threats come from within our own environments

- **Resources/funding**: they also vary in the resources available to them. Highly organized attackers sponsored by organized crime or national governments often have virtually limitless resources, whereas less organized attackers may simply be hobbyists working in their spare time

- **Level of sophistication/capability**: Threat actors vary greatly in their level of cybersecurity sophistication and capability. They range from the unsophisticated/unskilled attacker simply running code borrowed from others to the advanced persistent threat (APT) actor exploiting vulnerabilities discovered in their own research labs and unknown to the security community

• Motivations

- **Data exfiltration**: motivated by the desire to obtain sensitive or proprietary information, such as customer data or intellectual property

- **Espionage**: motivated by organizations seeking to steal secret information from other organizations. This may come in the form of nation-states attacking each other or corporate espionage.

- **Service disruption**: seek to take down or interrupt critical systems or networks, such as banking systems or health-care networks.

- **Blackmail**: seek to extort money or other concessions from victims by threatening to release sensitive information or launch further attacks.

- **Financial gain**: attacks are motivated by the desire to make money through theft or fraud. Organized crime is generally motivated by financial gain, as are other types of attackers.

- **Philosophical/political beliefs**: attacks are motivated by ideological or political reasons, such as promoting a particular cause or ideology. Hacktivists are generally motivated by philosophical or political beliefs.

- **Ethical**: or white-hat hacking, are motivated by a desire to expose vulnerabilities and improve security. These attacks are often carried out by security researchers or ethical hackers with the permission of the organization being tested

- **Revenge**: are motivated by a desire to get even with an individual or organization by embarrassing them or exacting some other form of retribution against them.

- **Disruption/chaos**: motivated by a desire to cause chaos and disrupt normal operations.

- **War**: Military units and civilian groups may use hacking to disrupt military operations and change the outcome of an armed conflict

2.2 Explain common threat vectors and attack surfaces.

• **Message-based:** exploit various communication channels, including email, SMS, and instant messaging (IM), to launch cyberattacks.

- **Email** is a primary attack vector, often used for phishing, spam, and other malicious messages. Attackers exploit the high volume of emails sent daily, needing only one successful breach to compromise an organization.
- **SMS** (Short Message Service) is another attack medium where attackers send fraudulent texts, often impersonating legitimate entities to steal credentials or distribute malware.
- **Instant Messaging (IM)** platforms are also targeted, with attackers using deceptive messages to lure victims into revealing sensitive information or downloading malicious files.

• **Image-based:** Images may contain hidden payloads or exploit vulnerabilities in image-processing software to execute malicious actions.

• **File-based:** Malicious Files: Attackers disguise harmful code in documents, PDFs, or executables, tricking users into opening them and triggering malware infections.

• **Voice call:** Voice calls may also be used to conduct vishing (voice phishing) attacks.

• **Removable device:** Attackers use removable media, such as USB drives, to spread malware. They may leave infected USB sticks in public places, relying on human curiosity to prompt users to plug them in. Once connected, the device executes malicious code, compromising the system and giving attackers control.

• **Vulnerable software**: Software installed on a system may contain known or undetected vulnerabilities

• **Unsupported systems and applications**

• **Unsecure networks**: Attackers exploit unsecure networks to gain unauthorized access:

- **Wired Networks**: Bold attackers may physically enter facilities and connect to unsecured network jacks or terminals to infiltrate an organization's network.

- **Wireless Networks**: Poorly secured Wi-Fi allows attackers to access the network remotely, even from outside the premises.
- **Bluetooth Connections**: Misconfigured Bluetooth devices without proper security settings can be exploited for unauthorized access.

• **Open service ports:** Misconfigured systems can expose unnecessary **open service ports**, creating security risks. Attackers exploit these ports to gain unauthorized access, especially when:

- **Unused or unnecessary** ports remain open, increasing the attack surface.
- **Default credentials** on services are left unchanged, making them easy targets.
- **Legacy systems** with outdated software lack security patches, leaving vulnerabilities exposed.

• **Default credentials**

• **Supply chain**: Attackers exploit vulnerabilities in an organization's supply chain by targeting **hardware providers, software vendors, and managed service providers (MSPs).**

- **Hardware Tampering**: Attackers may insert backdoors into devices during manufacturing or transit, compromising security before deployment.
- **Software Compromise**: Malicious actors may inject vulnerabilities or backdoors into software updates and patches before release.
- **MSP Exploitation**: Since MSPs have access to multiple client networks, attackers who breach an MSP can leverage that access to infiltrate customer systems.

• **Human vectors/social engineering**

- **Phishing**: is the fraudulent attempt to acquire sensitive information, such as credentials and credit card details, often through email, but also via smishing (SMS) and vishing (phone calls). Phishing can be targeted:

  - **Spear phishing** focuses on specific individuals or groups.
  - **Whaling** targets high-level executives like CEOs, referred to as "big fish."

  Awareness training is a key defense, teaching employees to recognize and respond to phishing attempts. Additionally, technical defenses like filtering, keyword matching, and reputation tools help detect and prevent phishing attacks.

- **Vishing**: is phishing carried out through voice or voicemail messages. Attackers use phone calls to manipulate targets into revealing personal, financial information, or transferring funds. Common vishing scams include requests for help with relatives abroad, tax scams, threats of legal action, or impersonating senior executives. Like other social engineering attacks, vishing often creates a sense of urgency, urging the target to act quickly to resolve a supposed issue. Attackers often pose as authority figures to gain trust and manipulate the victim.

- **Smishing**: is phishing conducted via text messages. It often involves sending links to fake websites that capture credentials, infect devices with malware, or request MFA information like SMS codes. Like other phishing attacks, smishing uses social engineering tactics, such as building trust, creating urgency, or impersonating authority, to deceive victims into clicking links or disclosing sensitive information.

- **Misinformation/disinformation**: Online Influence Campaigns are increasingly used in cyberwarfare and traditional warfare, often targeting social media, email, and other online platforms to spread misinformation and disinformation.

    - **Misinformation** refers to incorrect information, often unintentionally shared.

    - **Disinformation** is deliberately false information spread to influence public opinion or serve a specific agenda.

A high-profile example is the 2016 and 2020 U.S. elections, where influence campaigns played a significant role in shaping political outcomes. While advertising campaigns are a type of influence effort, most in the context of Security+ focus on campaigns meant to mislead or manipulate through false or misleading information.

- **Impersonation**: is a common social engineering technique where an attacker pretends to be someone else to gain access, data, or other desired outcomes. This often involves exploiting the target's willingness to trust the impersonator. Identity fraud or identity theft occurs when an attacker uses someone else’s identity, typically for financial gain. In some cases, impersonation can also be part of penetration tests or security assessments. In less specific instances, attackers may simply pretend to be delivery drivers or service providers without assuming a specific identity.

- **Business email compromise (BEC):** involves attackers using seemingly legitimate email addresses to conduct scams, such as invoice fraud, gift card scams, data theft, and account access attacks. Common techniques include:

- Compromising accounts
- Sending spoofed emails
- Using similar-looking fake domains
- Deploying malware

Mitigation methods include multifactor authentication, awareness training, and implementing policies to ensure proper email usage and behavior.

- **Pretexting**: involves fabricating a story or scenario to justify approaching someone, often used as part of impersonation tactics. The goal is to make the attacker appear more credible and gain access to sensitive information. An aware target can prevent this by asking questions or requiring verification, such as making a verification call, to expose and thwart the attack

- **Watering hole**: involve targeting websites that are frequently visited by specific victims. Attackers compromise these sites, knowing that their targets will visit them. The attackers then deploy malware through the site, often using methods like exploiting vulnerabilities or through advertising networks.

- **Brand impersonation**: is a phishing attack where emails appear to come from a legitimate brand, using recognizable logos and email templates. These emails often trick users into logging into their accounts, providing sensitive information, making payments, or downloading malware. The quality of these emails varies, from convincing replicas to poorly crafted scams.

- **Typo squatting**: involves creating misspelled URLs like legitimate sites, capitalizing on users' typos to drive traffic, sales, or ads. Organizations can combat this by registering common typos of their domains. A related attack, pharming, alters a system's hosts file or DNS settings to redirect users to fake websites. Unlike typosquatting, pharming involves more technical manipulation, such as malware or file modification, to mislead users.

2.3 Explain various types of vulnerabilities.

• Application

- **Memory injection**: The goal is to overwrite other information in memory with instructions that may be executed by a different process running on the system

- **Buffer overflow**: attempt to use more space than is allocated for a purpose and allow the attacker to perform memory injection, inserting their own content into sensitive memory locations.

- **Race conditions**: *Race conditions* occur when the security of a code segment depends upon the sequence of events occurring within the system

  o **Time-of-check (TOC):** is the instance when a system verifies access permissions or other security controls o **Time-of-use (TOU):** is the moment when the system accesses the resource or uses the permission that was granted

o **Target of Evaluation (TOE)** refers to the particular component, system, or mechanism being evaluated or tested for potential vulnerabilities, such as the system's method of managing and validating access permissions

- **Malicious update**: attempts to deploy a fake patch that actually undermines the security of an application or operating system

• **Operating system (OS)-based:** Securing operating systems is crucial for organizational security. Key vulnerabilities include:

- **Unpatched OS Flaws**: Attackers exploit system vulnerabilities, emphasizing the need for regular patching and minimizing exposed services.
- **Default Settings**: Insecure defaults, such as default passwords, create easy attack paths. Security baselines help mitigate these risks.
- **Weak Configurations**: Poorly designed configurations introduce security gaps, highlighting the importance of access control measures.
- **Misconfigurations**: Human errors can weaken system security, making proper security practices and monitoring essential.

• **Web-based**

- **Structured Query Language injection (SQLi):** is an attack where a user inputs malicious SQL code into a web application's input field, aiming to manipulate the backend database. For instance, instead of a simple search, an attacker could insert code causing the web server to execute additional, unauthorized SQL commands. If successful, this could expose sensitive data such as customer names and credit card numbers. Even when results aren't directly visible, attackers can use blind SQL injection techniques, either content-based or timing-based—to exploit the flaw without immediate feedback

- **Cross-site scripting (XSS):** Cross-site scripting (XSS) attacks occur when web applications allow an attacker to perform HTML injection, inserting their own HTML code into a web page.

• **Hardware**

- **Firmware**: is embedded software that controls a device's hardware and can often be updated manually. Firmware attacks can occur through malicious updates, compromised downloads, or remote updates. These attacks are dangerous because malicious firmware persists even after an OS reinstall, as seen with the **MoonBounce** malware in 2022. In 2023, millions of Android devices were sold with malware in the firmware. **Firmware validation** and techniques like **trusted boot** are critical for security.

- **End-of-life(EOL)**: hardware raises concerns due to lack of support. Once a device or system reaches end-of-life, manufacturer support ends, including security updates. Without these updates, security issues cannot be directly addressed, requiring compensating controls that may not always be suitable or effective for organizations.

- **Legacy**: This term is less well defined but typically is used to describe hardware, software, or devices that are unsupported

• Virtualization

- **Virtual machine (VM) escape**: attacks occur when an attacker breaks through the hypervisor's restrictions to access resources assigned to other virtual machines. This is a critical vulnerability in virtualized environments. VM Sprawl refers to the creation of unused virtual machine instances that are neglected, leading to security risks and increased costs. Organizations should maintain control and awareness of their virtual instances to avoid this issue.

- **Resource reuse**: it happens when cloud providers reassign hardware resources from one customer to another. If data is not properly erased from the hardware, the new customer may unintentionally gain access to data from the previous customer.

• Cloud-specific: Cloud providers offer native and third-party security solutions to help organizations secure their cloud infrastructure. Cloud-native controls integrate directly with the provider's services, often being cost-effective and user-friendly. Third-party solutions, while more expensive, allow for multicloud management.

Cloud Access Security Brokers (CASBs): act as intermediaries between cloud users and providers to monitor activity and enforce policies. There are two types:

- **Inline CASBs**: Intercept requests before reaching the cloud service, blocking policy violations.
- **API-based CASBs**: Interact directly with the cloud provider, monitoring activity without blocking requests in real time.

Resource Policies: cloud providers offer policies to limit users' actions, reducing risks from accidental commands, compromised accounts, or malicious insiders. An example policy restricts access to certain regions and limits the use of large instances to control costs.

Secrets Management: Hardware Security Modules (HSMs) manage encryption keys securely, preventing exposure to unauthorized parties. Cloud providers offer HSM services to manage customer keys securely.

• **Supply chain:** sophisticated attackers may target an organization's IT supply chain, including hardware, software, and service providers, to indirectly attack the organization.

- **Hardware Attacks**: Attackers may tamper with devices during manufacturing or while in transit, inserting backdoors that provide control once the device is installed on the organization's network.

- **Software Attacks**: Attackers can insert vulnerabilities or backdoors into software during development or through official updates and patches.

- **Managed Service Provider (MSP) Attacks**: Attackers infiltrating an MSP can gain access to the MSP's clients' systems and networks.

Other risks in the supply chain include vendors failing to support critical systems, providing necessary integrations, or ensure adequate security for outsourced code development or data storage.

• Cryptographic

- **Brute Force**: This attack involves trying every possible key until the correct one is found. While it is guaranteed to work, it can take an impractically long time for strong encryption algorithms.

- **Frequency Analysis**: This method looks for common patterns in encrypted messages, such as frequently used letters or words, to deduce the encryption method. It works on historical ciphers but not modern ones.

- **Known Plain Text**: The attacker uses known plaintext and corresponding ciphertext pairs to deduce the key. This method worked for cracking the German Naval Enigma code during World War II.

- **Chosen Plain Text**: The attacker selects specific plaintexts and obtains their corresponding ciphertexts, attempting to deduce the encryption key. Advanced methods like differential cryptanalysis fall under this category.

- **Related Key Attack**: Like chosen plain-text attacks, but the attacker gets ciphertexts encrypted with two different keys, helping to deduce the key used in both encryptions.

- **Birthday Attack**: This attack targets cryptographic hashes and uses the "birthday paradox." It exploits the probability that two inputs will produce the same output (a hash collision). For example, with MD5, a collision can occur with significantly fewer inputs than expected.

- **Downgrade Attack**: This attack forces a system to use weaker cryptographic methods by tricking the user or system into shifting to a less secure version of a protocol (e.g., TLS), making it easier to break.

- **Misconfiguration:** unlike configuration and defaults, occurs when a mistake is made. Human error remains a consistent way for attackers to successfully overcome default operating system and application security.

- **Mobile device**

  - **Side loading**: is the process of transferring files, such as applications, to a mobile device using methods like USB, MicroSD cards, or Bluetooth, outside of the official app store. While more common on Android, it can also occur on iOS devices. Sideloading allows users to install apps not available in their region or apps that are not signed, often for legitimate purposes. However, it is commonly prohibited by organizations as part of their security policies due to potential risks.

  - **Jailbreaking**: exploits vulnerabilities in a mobile device's operating system to escalate privileges and gain root access, providing the user with more control than typically allowed. After jailbreaking, users can install apps not available in official stores, modify restricted settings, or add custom elements to the operating system. While both sideloading and jailbreaking can have legitimate uses, they can also be exploited for malicious purposes, which is why they are often restricted in organizational security policies.

- **Zero-day:** refer to attacks that exploit vulnerabilities in software or systems that are unknown to the vendor or security community, meaning no patches or fixes are available. **Advanced Persistent Threat (APT)** attackers often discover these vulnerabilities through their own research and keep them in a repository for future use, making them highly dangerous. These attacks are particularly effective because they target flaws that have not been publicly identified or addressed. A notable example is **Stuxnet**, which exploited zero-day vulnerabilities to infiltrate the control systems of an Iranian uranium enrichment facility. The attack, attributed to the U.S. and Israeli governments, illustrates the potential of zero-day vulnerabilities for sophisticated, targeted attacks

2.4 Given a scenario, analyze indicators of malicious activity.

- **Malware attacks:** is a broad term for software created with the intent to harm systems, networks, or users. It can steal information, grant unauthorized access, or cause various other malicious actions that the system owner doesn't consent to

  - **Ransomware**: is malware that encrypts files or locks systems, demanding a ransom for release. It can also threaten exposure of sensitive data or illegal activities. Commonly delivered via phishing, it can also exploit direct attack methods like RDP or vulnerable services. **Indicators of compromise (IoCs)** include:

- Malicious IP connections
- Abnormal use of legitimate tools
- Lateral movement within networks
- File encryption and ransom demands
- Data exfiltration

Defense includes backup systems to store files separately and decryption tools. Antivirus and anti-malware solutions also offer protection against ransomware.

- **Trojan**: are malware disguised as legitimate software, tricking users into running them. Once activated, they provide attackers access to systems. A common example is the Triada Trojan, which is distributed through modified apps like WhatsApp, gathering device information to enable further malicious activities. Indicators of compromise (IoCs) for Trojans:

  - Malware signatures or downloadable files
  - Command and control system IP addresses
  - Unusual files or folders created on devices

Remote Access Trojans (RATs) provide attackers with remote access to systems, often using legitimate tools, making them hard to identify. Mitigation involves user awareness, controlling software downloads, and using anti-malware, EDR, and behavioral detection tools to prevent or identify Trojans and RATs.

- **Worm**: self-replicating malware that spread without user interaction, exploiting vulnerabilities in networks, email attachments, file shares, IoT devices, and more. Unlike Trojans, worms install themselves automatically, making them highly dangerous.
Notable Worms:

  - Stuxnet (2010): A nation-state cyber weapon targeting Iranian nuclear facilities, spreading via USB drives and exploiting industrial control systems (ICSs).

  - Raspberry Robin: A modern worm linked to pre-ransomware activity, spreading via infected USB drives and using Windows tools for persistence.

Indicators of Compromise (IoCs):

- Malicious files or downloads from remote systems
- Command and control (C2) communication
- System command misuse (e.g., cmd.exe, msiexec.exe)
- Hands-on-keyboard attacker activity

Mitigation Strategies:

- Network Controls: Firewalls, IPS, and segmentation to prevent spread
- Patching: Keeping systems updated to close vulnerabilities
- EDR & Anti-malware: Detecting and removing infections
- Reinstallation: In severe cases, full system restoration may be required

- **Spyware**: malware that secretly collects user data, often for identity theft, fraud, ads, or surveillance (e.g., stalkerware). Indicators of Compromise (IoCs):

- Remote access activity
- Disguised system processes
- Browser injection attacks

Mitigation:

- User awareness to prevent installation
- Restrict unapproved apps
- Antispyware tools and configurations

- **Bloatware**: it refers to preinstalled, often unnecessary applications on new devices. While not inherently harmful, it can slow performance, send data externally, or introduce security risks.
Removing bloatware manually or using a clean OS improves security and performance. Since it's not malicious, it lacks traditional Indicators of Compromise (IoCs).

- **Virus**: are malicious programs that self-replicate but require user interaction or an infected file to spread. Unlike worms, they don't spread automatically through networks. Types of Viruses:

- Memory-resident: Stay active in system memory.
- Non-memory-resident: Execute and shut down after spreading.
- Boot sector viruses: Infect the boot sector of storage devices.
- Macro viruses: Exploit macros in documents.
- Email viruses: Spread via infected email attachments.
- Fileless viruses: Operate in memory without leaving files on disk.

Mitigation:

- Keep browsers, plug-ins, and software updated.
- Use antimalware tools to detect malicious scripts.
- Employ network defenses like IPS and reputation-based filtering.
- Remove viruses using antivirus tools, but severe infections may require a full system wipe and reinstallation.

- **Keylogger**: programs designed to capture keystrokes and other user inputs, such as mouse movements and touchscreen interactions. They operate through various methods, including kernel-level access, API hooks, or memory monitoring, all aimed at stealing sensitive information. Mitigation Strategies:

  - Prevent installation: Keep systems patched and use antimalware tools.

  - Use multifactor authentication (MFA): Limits damage even if credentials are stolen.

  - Bootable USB drives: Prevent reliance on compromised operating systems in high-risk environments.

Indicators of Compromise (IoCs):

- Known malware signatures and file hashes.
- Suspicious data exfiltration to remote servers.
- Unusual running processes.

Hardware keyloggers, which are physical devices attached to keyboards, can be used to steal credentials. These have been found in academic settings to alter grades and compromise accounts

- **Logic bomb**: Unlike standalone malware, logic bombs are hidden code within legitimate programs that activate when specific conditions are met. They can be part of other malware or embedded in software by insiders, posing a significant risk in software development and system management.

Detection & Mitigation:

- Code review and auditing: Regularly inspect software for hidden malicious logic.
- Change monitoring: Track modifications to critical applications.
- Access controls: Restrict who can alter software code to reduce insider threats.

Since logic bombs are embedded in programs, they lack common IoCs and require proactive security practices for detection.

- **Rootkit**: Rootkits are a type of malware designed to grant attackers persistent, hidden access to a system. They often use advanced techniques to evade detection, such as modifying system processes, hooking filesystem drivers, or infecting the Master Boot Record (MBR).
Detection & Challenges:

- Difficult to detect: Since rootkits hide within the system, infected machines cannot always be trusted for analysis.

- Detection techniques: Use integrity checking, behavior-based analysis, and specialized anti-rootkit tools.

- External analysis: Testing from a separate, trusted device can help uncover hidden rootkits.

Mitigation & Prevention:

- System hardening: Apply patches, enforce privilege management, and enable Secure Boot.
- Regular monitoring: Watch for suspicious activity, such as unauthorized file changes or unexpected open ports.
- Restoration over removal: Due to the complexity of rootkits, reimaging the system or restoring from a clean backup is often the best solution.

Because rootkits are designed to avoid detection, proactive security practices are crucial to prevent infections before they occur.

• Physical attacks

- Brute force: include breaking down doors, cutting off locks, or other examples of the simple application of force or determination to physical entry.

- **Radio frequency identification (RFID) cloning**: work by cloning an RFID tag or card. This can be difficult to catch if the RFID is the only identifier used. Without physical observation or automated systems that pay attention to unusual activity and access and flag it for review, RFID cloning may go unnoticed

- **Environmental**: include attacks like targeting an organization's heating and cooling systems, maliciously activating a sprinkler system, and similar actions. These are more likely to be detected as issues or problems than as attacks and determining if issues were caused by a malicious attack can be difficult.

• Network attacks

- **Distributed Denial-of-Service (DDoS):** Overload a system or network with traffic from multiple sources, making detection and mitigation difficult. Types of DDoS:

- Volume-Based: Aim to saturate available bandwidth.
- UDP Floods: Send large amounts of UDP traffic without handshake.
- ICMP Floods: Flood the network with ICMP requests (ping).

Protocol-Based: Exploit vulnerabilities in network protocols.

- SYN Floods: Send TCP requests without completing the handshake, exhausting system resources.
- Ping of Death, Smurf Attack, Xmas Attack: Manipulate packets to cause malfunctions.

Amplification/Reflection: Use legitimate services to amplify traffic towards the target:

- **Amplified**: Exploit protocols that respond with much larger data than the initial request (e.g., DNS, NTP). The attacker sends a small request with a spoofed victim IP, causing a massive response directed at the target.
- **Reflected**: Use legitimate services to reflect traffic towards the victim. The attacker sends requests with the victim's IP to various servers, which then respond directly to the victim, making it harder to identify the attacker.

- **Domain Name System (DNS) Attacks**: DNS attacks allow attackers to manipulate network traffic without removing encryption. Types of DNS Attacks:

- **Domain Hijacking**: Attackers take control of a domain by modifying its settings, often through registrar vulnerabilities, social engineering, or domain renewal lapses.
- **DNS Poisoning**: The attacker inserts false DNS responses, redirecting traffic to malicious sites. This can occur via on-path attacks, protocol vulnerabilities, or DNS cache poisoning.
- **URL Redirection**: Modifying the hosts file on a device forces redirection to harmful IP addresses.

Defenses Against DNS Attacks:

- DNSSEC: Protects the integrity and authenticity of DNS responses.
- Hosts File Monitoring: Antimalware tools can detect suspicious changes.
- Domain Reputation Services: Tools like Cisco Talos or McAfee WebWasher evaluate the reliability of domains and IPs, protecting against malicious sources.

Attack Indicators: Signs of compromise include changes in DNS resolution and domain hijacking. Active monitoring is essential to prevent and detect these attacks.

- **Wireless**

- **On-Path: A man-in-the-middle (MitM) attack** occurs when an attacker intercepts and modifies traffic between two parties.
SSL Stripping and Defenses: SSL stripping removes TLS encryption to read data in plaintext, often on open Wi-Fi networks. Defenses include:

- HSTS: Forces the browser to use HTTPS.
- Certificate Verification: Prevents insecure connections.
- HTTPS Everywhere: Forces HTTPS connections.

MitB and Attack Indicators: A man-in-the-browser attack uses trojans to manipulate browser data, bypassing TLS. Effective defenses include antivirus software and network monitoring. Attack indicators may include changes in gateways or traffic routes.

- **Credential Replay**: Credential replay attacks capture and reuse valid authentication data, such as credential hashes, to gain unauthorized access. However, modern security measures, like unique session IDs and encryption, mitigate these attacks. Attack Indicators: Signs of credential replay include changes to network gateways or traffic routes, often related to on-path attacks.

- **Malicious Code**: Malicious code attacks include worms, backdoors, viruses, trojans, and ransomware that spread via networks.
Attack Indicators: Signs of malicious activity include signatures recognized by IDS/IPS systems and scans on ports and protocols used by worms.

• Application attacks

- **Injection**: occur when an attacker sends malicious data to an application, causing it to execute unintended commands or access unauthorized data. Common types include:

  - **SQL Injection**: Inserting malicious SQL queries into input fields, allowing attackers to manipulate or extract data from the database.
  - **Command Injection**: Injecting system commands through vulnerable inputs, enabling attackers to execute arbitrary commands on the server.
  - **XML Injection**: Manipulating XML data to alter the structure or logic of an application, leading to unauthorized actions.
  - **LDAP Injection**: Inserting malicious LDAP queries to bypass authentication or retrieve sensitive information.

These attacks exploit improper validation of user inputs and can lead to unauthorized access, data leakage, or system compromise. Proper input validation and parameterized queries can help defend against them.

- **Buffer overflow**: occurs when an attacker forces a program to write more data than allocated in memory, overwriting adjacent data with executable code. This allows for memory injections and unauthorized execution. They remain common, with some vulnerabilities lasting over a decade. An example is integer overflow, where an arithmetic operation stores too large an integer. Analysts should patch identified vulnerabilities during scans to prevent exploitation.

- **Replay**: occurs when an attacker steals a session cookie, allowing them to impersonate the legitimate user and bypass authentication. This can happen through eavesdropping on unencrypted connections, installing malware, or tricking the user with a fake login page. The attacker then uses the stolen cookie to gain unauthorized access, known as session replay. To protect against this, web developers can mark cookies with the SECURE attribute, ensuring they are only transmitted over encrypted connections. Additionally, the NTLM pass-the-hash attack is another form of replay attack, where attackers use stolen password hashes to gain access to Windows systems.

- **Privilege escalation**: exploit vulnerabilities to gain administrative privileges (vertical or lateral escalation) on a system. A well-known example is Dirty COW (2016), which allowed root access on Linux. Prevention includes security updates, restricting privileges, and monitoring suspicious activity.

- **Forgery**: exploit trust relationships and aim to make users unknowingly execute commands on a remote server. There are two main types:

    - **Cross-Site Request Forgery (CSRF/XSRF):** exploit the trust that remote websites have in a user's browser. They work by assuming that users are often logged into multiple websites at once. An attacker can embed malicious code in a website, which sends a request to a second site. If the user is logged into that second site, the request could succeed. For example, an attacker might post a link on a forum that transfers funds from an online banking account to the attacker's account.
    Protection:
        - Use secure tokens that attackers cannot predict.
        - Check the referring URL to ensure requests originate from the same site.
    - **Server-Side Request Forgery (SSRF):** trick a server into making requests for a URL specified by a user. These attacks are possible when a web application accepts user-supplied URLs and retrieves information from them. If the server has access to non-public URLs, the attacker can gain access to this private information.

- **Directory traversal**: exploit web server misconfigurations that allow users to access files outside of the designated web directories. These attacks use operators like .. to navigate up the directory structure. For example, an attacker may manipulate the URL (e.g., www.mycompany.com/../../../etc/shadow) to access sensitive files, such as the shadow password file. If successful, this can expose hashed passwords for brute-force attacks. Proper configuration of access controls and validation of input paths are essential to prevent such attacks.

• Cryptographic attacks

- **Downgrade**: sometimes used against secure communications such as TLS to get the user or system to inadvertently shift to less secure cryptographic modes. The idea is to trick the user into shifting to a less secure version of the protocol, one that might be easier to break.

- **Collision**: occurs when two different inputs produce the same hash output. This undermines the integrity of a hash function, as it allows attackers to substitute one input for another without detection. Collision attacks are particularly relevant to algorithms like MD5 and SHA-1, which have known vulnerabilities that make finding collisions easier than brute-forcing all possible inputs. The birthday paradox is often used to explain the likelihood of collisions, where the number of attempts needed to find a collision is much smaller than the total number of possible hash values.

- **Birthday**: targets cryptographic hash functions, leveraging the birthday paradox. The paradox states that with 23 people in a room, there is a 51% chance that two people share the same birthday, even though there are 365 days in a year. This principle can be applied to hash functions, where a collision occurs when two different inputs produce the same output.
For a hash like MD5 (with 128-bit output), a brute-force approach would require trying 2^128 possible inputs to guarantee a collision. However, the birthday paradox suggests that with around 2^64 (about 31.4 trillion) different inputs, there is a 51% chance of finding a collision, which is much more feasible than trying every possible input.

• Password attacks

- **Spraying**: are a form of brute-force attack that attempts to use a single password or small set of passwords against many accounts. This approach can be particularly effective if you know that a target uses a specific default password or a set of passwords

- **Brute force**: which iterate through passwords until they find one that works. Actual brute-force methods can be more complex than just using a list of passwords and often involve word lists that use common passwords, words specifically picked as likely to be used by the target, and modification rules to help account for complexity rules. Regardless of how elegant or well thought out their input is, in the end, brute force is simply a process that involves trying different variations until it succeeds

- **Dictionary**: there is yet another form of brute-force attack that uses a list of words for their attempts. Commonly available brute force dictionaries exist, and tools like John the Ripper, a popular open-source password cracking tool, have word lists (dictionaries) built in.

• Indicators

1. Account.lockout, which is often due to brute-force login attempts or incorrect passwords used by attackers.
2. Concurrent.session.usage when users aren't likely to use concurrent sessions. If a user is connected from more than one system or device, particularly when the second device is in an unexpected or uncommon location or the application is one that isn't typically used on multiple devices at once, this can be a strong indicator that something is not right.
3. Blocked.content is content that the organization has blocked, often via a DNS filter or other tool that prohibits domains, IP addresses, or types of content from being viewed or accessed. If this occurs, it may be because a malicious actor or malware is attempting to access the resource.
4. Impossible.travel, which involves a user connecting from two locations that are far enough apart that the time between the connections makes the travel impossible to have occurred, typically indicates that someone else has access to the user's credentials or devices.
5. Resource.consumption like filling up a disk or using more bandwidth than usual for uploads or downloads, can be an indicator of compromise. Unlike some of the other IoCs here, this one often requires other actions to become concerning unless it is much higher than usual.
6. Resource.inaccessibility can indicate that something unexpected is happening. If a resource like a system, file, or service isn't available identifying the underlying cause and ensuring that the cause isn't malicious, it can be important.
7. Out_of_cycle.logging occurs when an event that happens at the same time or on a set cycle occurs at an unusual time. This might be a worker logging in at 2 a.m. who normally works 9–5, or a cleanup process that gets activated when it normally runs once a week.
8. Missing.logs may indicate that an attacker has wiped the logs to attempt to hide their actions. This is one reason that many organizations centralize their log collection so that a protected system will retain logs even if they are wiped on a server or workstation.

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

• Segmentation: involves dividing a network into smaller, manageable segments to enhance security, performance, and control. Common methods include **Virtual Local Area Networks (VLANs)**, which create separate broadcast domains at the Data Link layer. Segmentation reduces network noise and limits the impact of broadcasts.

Key network segmentation concepts:

- **Screened Subnets (DMZs):** Zones for less-trusted systems, often used for web servers or Internet-facing devices.
- **Intranets**: Internal networks accessible only to authorized users.
- **Extranets**: Networks for external users like partners or customers.

Zero Trust networks emphasize security for both internal and external traffic, requiring protection for communications within segments, not just at boundaries. East-West traffic refers to communications between systems within the same zone, and monitoring it is crucial in Zero Trust models.

• Access control

- **Access control list (ACL):** are rules used to permit or deny specific actions, often like firewall rules. ACLs can vary in complexity, from simple statements to advanced configurations, including time-based or dynamic ACLs with specific conditions. For example, a Cisco IP-based ACL might look like:
access-list 100 permit tcp any host 10.10.10.1 eq http.

Cloud services also offer ACLs to control traffic, such as for Virtual Private Clouds (VPCs), often managed through security groups.

- **Access Control Schemes** define how users, services, and programs are granted access to system resources. Key schemes include:

  - **Mandatory Access Control (MAC):** The OS enforces access policies set by a security administrator. Users cannot change these policies. Common in high-security systems like SELinux and Windows MIC.
  - **Discretionary Access Control (DAC):** Owners of objects (e.g., files) assign access rights to them. Users can control who can read, modify, or execute these objects, like Linux file permissions.
  - **Role-Based Access Control (RBAC):** Access is granted based on roles (e.g., cashier or database administrator). Three primary rules: role assignment, role authorization, and permission authorization.

  - **Rule-Based Access Control (RuBAC):** Access is granted based on a set of rules, like firewall rules or ACLs.
  - **Attribute-Based Access Control (ABAC):** Access is determined by policies based on user attributes. It allows complex, flexible rules but can be difficult to manage.

- **Filesystem Permissions**: control which accounts, users, groups, or services can perform actions (read, write, execute) on files. There are different permission schemes for Linux and Windows systems.

  - **Linux Permissions**: Represented as drwxrwxrwx, where the first character indicates whether it's a directory or file, and the following three sets of rwx show permissions for the user, group, and others. Numeric representations (e.g., chmod 755) are used for shorthand to change permissions.
  - **Windows Permissions**: These can be set through the GUI or command line. Permissions include:
    - Full control (similar to rwx in Linux)
    - Modify (view and change files)
    - Read & Execute (run files, but not modify)

• Application allow/deny list: Application **allow lists** (sometimes referred to as whitelisting) list the applications and files that are allowed to be on a system and prevent anything that is not on the list from being installed or run.

Application **deny lists** or block lists (referred to as blacklists) list applications or files that are not allowed on a system and will prevent them from being installed or copied to the system.

• Isolation: or **quarantine** solutions can place files in a specific safe zone. Antimalware and antivirus often provide an option to quarantine suspect or infected files rather than deleting them, which can help with investigations

• Patching: Keeping systems and software up to date is essential for endpoint security, as it removes known vulnerabilities. However, patching must be carefully managed to avoid introducing new flaws.

- **Automated Updates**: Many applications and operating systems have built-in update tools, but enterprises often use centralized patch management solutions.

- **Patch Delays & Risk Management**: Organizations may delay patches to monitor for issues, balancing the risk of unpatched exploits against potential patch failures.

- **Enterprise Patch Management**: Tools like Microsoft's Configuration Manager and third-party solutions help manage updates across multiple systems and applications.

- **Mobile Device Patch Challenges**: Mobile Device Management (MDM) solutions assist in tracking and enforcing updates for mobile software and OS security.

• **Encryption:** Disk encryption protects data from unauthorized access if a system or disk is lost or stolen.

- **Full-Disk Encryption (FDE):** Encrypts the entire disk, requiring a key to decrypt. Transparent encryption makes it seamless for users but leaves systems vulnerable if accessed while unlocked.
- **Volume & File Encryption**: Encrypts specific volumes, files, or folders, allowing different security levels and secure data transfer.
- **Self-Encrypting Drives (SEDs):** Hardware-based encryption requiring a key to boot, offering strong protection but still vulnerable if the system is already logged in.
- **Risks**: Lost encryption keys can render data unrecoverable, and troubleshooting encrypted drives can be complex.

• **Monitoring:** is a key part of containment and mitigation efforts because security professionals and system administrators need to validate their efforts. Monitoring a system, service, or device can provide information about whether there are still issues or the device remains compromised. Monitoring can also show other actions taken by attackers after remediation is completed, helping responders identify the rest of the attacker's compromised resources.

• **Least privilege:** asserts that individuals should only be given the minimum permissions needed to perform their job functions. While the concept is simple, it can be difficult to implement effectively. It requires careful management of permissions and regular review to avoid security risks. One common issue is the privilege creep, which happens when employees transition between roles but retain permissions from previous positions without having them revoked, leading to unnecessary access over time

• **Configuration enforcement:** Once baselines are set, tools support configuration enforcement, a process that not only monitors changes but makes changes to system configurations as needed to ensure that the configuration remains in its desired state.

• **Decommissioning:** is the process of removing outdated systems and devices from service, ensuring that sensitive data is securely erased. This process involves several steps:

1. Removing from service and inventory: **Retiring** the device or system.
2. **Data removal**: Wiping or destroying storage media to prevent data recovery.

- **Wiping**: Overwriting data, often with multiple passes (e.g., DBAN), though SSDs require secure erase commands due to wear-leveling.
- **Destroying**: Physically shredding or incinerating media to ensure data is unrecoverable.

For high-security environments, using full-disk encryption and discarding the encryption key is an alternative method for ensuring data can't be accessed. Certification processes are used to document proper disposal, often through third-party services that provide certificates of destruction.
A real-world example highlights the risks of not properly wiping systems, as sensitive data remained accessible on a system passed through multiple users without being wiped, leading to a potential security incident.

• Hardening techniques

- Installation of endpoint protection

- Host-based firewall

- Host-based intrusion prevention system (HIPS)

- Disabling ports/protocols

- Default password changes

- Removal of unnecessary software

- System Hardening: Involves modifying system settings to enhance security and reduce vulnerability. Key methods include:

- Reducing the attack surface
- Using tools and benchmarks (e.g., CIS, NIST)
- Key Security Measures: Disabling unnecessary services, ports, and protocols, changing default passwords, and removing unneeded software.

- **Service Hardening**:

- Disable unneeded services and ports to minimize the attack surface.
- Port scanners help identify vulnerabilities.
- Windows and Linux offer tools to start/stop services.

- **Network Hardening**: Use VLANs to segment and secure network traffic, particularly for IoT and guest networks.

- **Default Passwords**: Always change default passwords to mitigate risks from publicly available default credentials.

- **Operating System Hardening**: Modify system settings using benchmarks (e.g., CIS) to secure the OS. Example: Setting password policies, disabling reversible encryption, and applying Group Policy for Windows.

- **Windows Registry & Group Policy Hardening**: Secure Registry by limiting access and configuring permissions. Use Group Policy Objects (GPOs) for system-wide settings management.

- **Linux Hardening (SELinux):** SELinux enforces mandatory access control, improving security by defining strict user and resource permissions.

- **Configuration Management**: Use tools (e.g., Jamf Pro, CFEngine) to enforce security baselines and automate configuration.

- **Patching and Patch Management**: Regularly update systems and software to fix vulnerabilities. Use patch management tools to automate and track updates.

- **Encryption**: Full-disk and volume encryption protect data, particularly in case of theft or loss. Self-encrypting drives (SEDs) provide hardware-based encryption, ensuring data security with minimal user intervention.

3.0 Security Architecture

3.1 Compare and contrast security implications of different architecture models

• Architecture and infrastructure concepts

- **Cloud**: Cloud computing involves delivering computing services (e.g., storage, applications, servers) over the Internet. Providers like Google and AWS offer these services to customers, enabling access from anywhere.
Core Features:

  - Ubiquitous Access: Cloud services are available wherever there’s Internet access.
  - On-demand: Users can quickly provision or deprovision resources.
  - Multitenancy: Multiple users share the same cloud infrastructure, maximizing efficiency.
  - Configurable Resources: Users can customize resources like infrastructure, platforms, or applications.

Benefits:

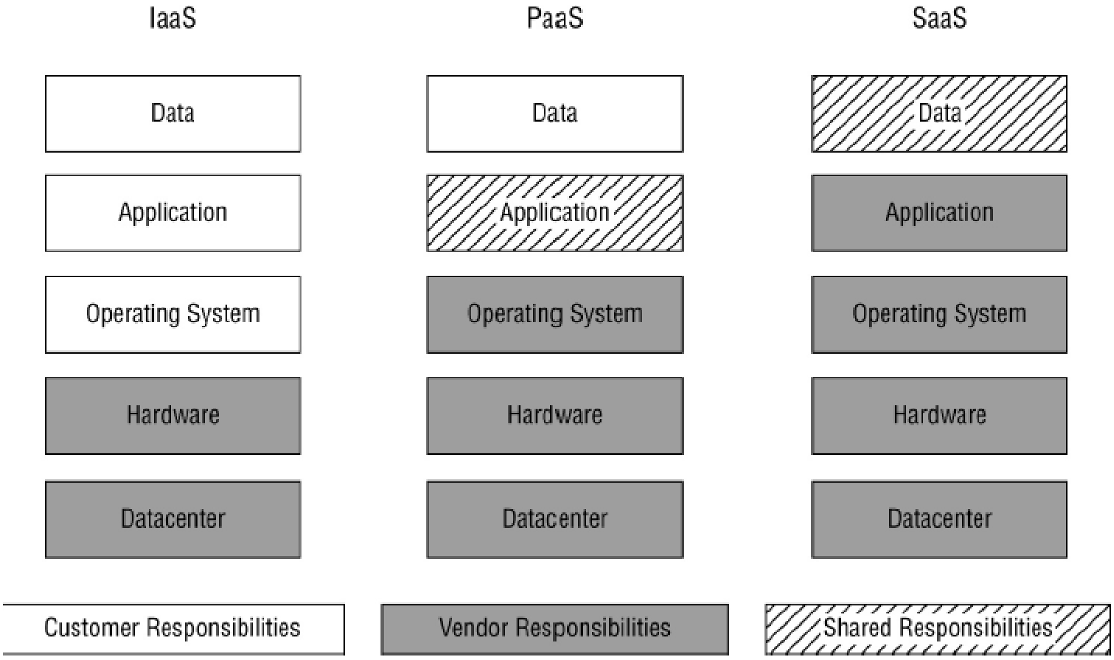
- **On-demand** Self-service: Resources are available whenever needed, enhancing flexibility.
- **Scalability**: Resources can be increased or decreased based on demand, through:
    - **Vertical Scaling**: Increasing server capacity (e.g., adding CPU or memory).
    - **Horizontal Scaling**: Adding more servers to handle increased load.
- **Elasticity**: Capacity automatically adjusts to meet demand, optimizing costs.
- **Measured Service**: Usage is tracked, and customers pay based on their actual consumption.
- **Agility and Flexibility**: Cloud services allow quick testing, deployment, and iteration, enabling rapid innovation

Service Model:

- **XaaS (Anything as a Service):** Cloud services are often categorized as "XaaS," where "X" represents the type of service provided.
- **Infrastructure as a Service (IaaS)**: IaaS provides basic infrastructure components like computing, storage, and networks. Customers can configure and manage these services based on their needs while the cloud provider manages the physical hardware and underlying security.
- **Software as a Service (SaaS)**: SaaS offers fully managed applications in the cloud. The provider handles everything, including the infrastructure and application performance. Customers only manage limited configuration and data access, paying typically through a subscription. Examples: Web-based email, ERP, and CRM systems.
- **Platform as a Service (PaaS):** PaaS provides a platform for customers to run their own applications, with the provider managing the infrastructure. It includes tools, libraries, and services that facilitate code execution.
- **Function as a Service (FaaS),** a type of PaaS, allows customers to run event-driven code without managing the servers (serverless computing). Example: AWS Lambda, which executes user-uploaded code in response to events.
- **Infrastructure as Code (IaC):** IaC is a practice of managing and provisioning cloud infrastructure using code rather than manual processes. With IaC, cloud resources (like servers, networks, and storage) can be defined and deployed automatically using configuration

files or scripts. It enables repeatable and consistent infrastructure setups, reduces human errors, and integrates well with DevOps practices. Tools: Terraform, AWS CloudFormation, Ansible.

o **Responsibility matrix**: The Shared Responsibility Model divides the cybersecurity and operational responsibilities between the cloud service provider and the customer. The division of these responsibilities varies depending on the cloud service model being used (IaaS, PaaS, or SaaS). The primary goal is to ensure that both parties understand their roles in securing the environment.



This model ensures that both the customer and the provider collaborate to maintain a secure cloud environment, with the provider handling the lower layers of the stack and the customer focusing on the higher layers where their data and applications reside.

o **Hybrid considerations**

o **Third-party vendors**: Cloud governance ensures that IT activities align with organizational strategy, policy, and security standards, particularly in managing third-party vendor relationships.

- **Vendor Vetting**: Organizations need to carefully evaluate potential cloud vendors before establishing partnerships, ensuring they align with business needs and security requirements.
- **Vendor Relationship Management**: Ongoing monitoring of vendor performance and stability is crucial to identify potential risks early,

ensuring that the vendor remains reliable and meets agreed-upon standards.

- **Portfolio Oversight**: Managing an organization's overall cloud activities, including vendor relationships, helps ensure compliance with strategic goals and effective resource use.
- **Auditability**: Cloud contracts should include provisions that allow customers to audit vendors directly or through a third party to ensure that the vendor is secure and meets data protection commitments. Serverless

- **Microservices**: cloud service offerings that provide very granular functions to other services, often through a function-as-a-service model. These microservices are designed to communicate with each other in response to events that take place in the environment

- **Network infrastructure**

  - **Physical isolation**: refers to separating devices with no network connection, often referred to as an air-gapped design. This method requires physical presence to transfer data, preventing remote attacks. However, air gaps can fail, as seen with the Stuxnet malware, which used infected USB drives to bypass the air gap and infect isolated systems. Physical separation alone isn't foolproof and must be combined with other security controls to prevent vulnerabilities
  - **Logical segmentation**: is achieved through software or settings, rather than physical separation. A common method is VLANs (Virtual Local Area Networks), where VLAN tags are applied to packets to create separate virtual network segments. Systems on these segments perceive them as distinct physical segments. However, attacks targeting logical segmentation aim to bypass the software controls, enabling traffic to be sent or received from other segments, thus compromising the separation.
  - **Software-defined networking (SDN):** allows dynamic control of networks using software, with controllers managing devices and configurations. It enables flexible, performance-based network adjustments and can be used for security, such as dynamically isolating systems or creating security zones.
    SD-WAN is a virtual WAN that combines various connectivity methods (e.g., MPLS, 4G/5G, broadband) to enhance performance and reduce costs. It routes traffic based on application needs and is often used in place of MPLS for cost-effective, high-availability solutions.

Secure Access Service Edge (SASE) combines SD-WAN, VPNs, and cloud-based security tools (like firewalls, CASBs, and zero-trust networks) to secure access and data, regardless of device location. It ensures secure, policy-enforced access to cloud resources, leveraging tools like Cloud Access Security Brokers (CASBs) to manage cloud security.

- **On-premises**: Unlike on-premises hardware acquisition, you can provide cloud services yourself without dealing with account representatives and order processing times

- **Centralized vs. decentralized**: Another common reason for using hybrid cloud environments is a desire to move away from a *centralized* approach to computing that places a significant portion of an organization's infrastructure within a single environment toward a *decentralized* approach that reduces single points of failure by spreading technology components across multiple providers.

- **Containerization**: Containerization provides application-level virtualization, allowing applications to be packaged and run as portable units across different operating systems and hardware platforms. Unlike virtual machines, containers share the host operating system but remain isolated from each other.
Containers package an application with all its dependences, enabling it to run consistently across environments without needing a full operating system for each instance.

Platforms like Docker provide standardized interfaces for containers to interact with operating system resources.

Security Considerations: Containers must be isolated from each other to prevent one from affecting another, both operationally and in terms of security. Container platforms share security concerns similar to virtualization platforms.

Security Best Practices for Containers: Use container-specific host operating systems with reduced features to minimize the attack surface. Segment containers based on risk profiles and purposes. Use container-specific security tools for vulnerability management.

Containers enhance portability and scalability, but like virtual machines, they require proper isolation and security to prevent potential issues.

- **Virtualization**: Cloud computing providers utilize virtualization technology to maximize the efficiency of their datacenters by allowing multiple virtual machines (VMs) to share the same physical hardware. This is accomplished using a hypervisor,

which is responsible for managing and isolating VMs from each other.
Hypervisor:

- **Function**: The hypervisor is responsible for isolating virtual machines, ensuring that each VM has the illusion of its own dedicated hardware, even though the underlying resources are shared.
- **Security Role**: It ensures that VMs cannot interfere with or access the data or resources of other VMs.

Types of Hypervisors:

- **Type I (Bare metal) Hypervisor**: Runs directly on physical hardware. More efficient and commonly used in datacenters for server virtualization.
- **Type II (Hosted) Hypervisor**: Runs as an application on top of an existing operating system. Less efficient and generally used for personal virtualization needs, such as on personal computers.

These hypervisor models are crucial for creating scalable and secure virtualized environments, allowing cloud providers to efficiently manage resources and deliver services.

- **IoT**: refers to network-connected devices used for automation, sensors, and security. IoT devices raise several security concerns, including:

  - Poor Security Practices: Weak defaults, lack of encryption, and insecure data storage.
  - Short Support Lifespans: Limited patching and updates, leaving devices vulnerable.
  - Vendor Data Issues: Concerns about data ownership and unauthorized access.

Despite these issues, IoT devices, like wearables, continue to grow in use. Security professionals must address both company and personal devices.

- **Industrial control systems (ICS)/ supervisory control and data acquisition (SCADA):** are essential for automating and managing industrial and manufacturing processes. SCADA is used to monitor large-scale facilities, such as those for power or water distribution. These systems integrate sensors, devices like Programmable Logic Controllers (PLCs), and Remote Telemetry Units (RTUs) to collect and send data, allowing operators to monitor and control processes in real-time.
From a security perspective, these systems are complex to protect because they are often not designed with security in mind. Introducing protective measures can interfere

with their functioning, so one of the most effective solutions is isolating them from other networks to prevent external attacks.

ICS vs. SCADA: ICS is a broader term encompassing all industrial automation systems, while SCADA specifically refers to large-scale systems used for monitoring and controlling widespread processes, such as utilities.

System Components: SCADA systems use a combination of RTUs and PLCs to manage and control processes. RTUs are used for remote data collection, and PLCs manage industrial devices like machinery. ICS systems might also include these components but may involve other forms of industrial control as well.

Security Challenges: ICS and SCADA systems are often designed without security in mind, making them vulnerable. Security measures, such as isolation from other networks, are crucial to prevent interference and external threats.

- **Real-time operating system (RTOS):** is an operating system that is used when priority needs to be placed on processing data as it comes in, rather than using interruptions for the operating system or waiting for tasks being processed to be handled before data is processed. Since embedded systems are widely used for industrial processes where responses must be quick, real-time operating systems are used to minimize the amount of variance in how quickly the OS accepts data and handles tasks

- **Embedded systems**: Embedded systems have unique constraints that require special attention for security:

  - Limited Resources: Low computational power, memory, and storage make it difficult to run traditional security tools (e.g., firewalls, antimalware).
  - Network Connectivity Issues: Some embedded systems may lack network access or have limited connectivity, making remote monitoring and patching challenging.
  - Authentication Challenges: Lack of authentication or the inability to authenticate may be necessary for usability, requiring alternative security models for authorization.
  - High Cost: While individual devices may be low-cost, they can be expensive components of larger systems, making replacement difficult. Special security designs may be needed to address vulnerabilities.
  - Implied Trust: Many embedded systems rely on physical access and assume operators are trusted, which can introduce security risks if not properly managed.

These limitations require tailored security strategies for each device to prevent vulnerabilities

- **High availability(HA)**: refers to a system's ability to remain operational without downtime, even during upgrades, failures, or changes in load. HA designs aim to meet availability targets using solutions like clustering, load balancing, and proxies. The focus is on ensuring reliable switching between components, eliminating single points of failure, and quickly detecting and addressing issues to maintain service continuity.

• Considerations

- **Availability**: targets should be set and designed based on organizational requirements balanced against the other considerations.

- **Resilience**: which is a component of availability that determines what type and level of potential disruptions the service or system can handle without an availability issue

- **Cost**: including financial, staffing, and other costs

- **Responsiveness**: or the ability of the system or service to respond in a timely manner as desired or required to function as designed

- **Scalability**: either vertically (bigger) or horizontally (more) as needed to support availability, resilience, and responsiveness goals.

- **Ease of deployment**: which describes the complexity and work required to deploy the solution that often factors into initial costs and that may have impacts on ongoing costs if the system or service is frequently redeployed.

- **Risk transference**: through insurance, contracts, or other means is assessed as part of architectural design and cost modeling.

- **Ease of recovery** is considered part of availability, resilience, and ease of deployment as complex solutions may have high costs that mean additional investments should be made to avoid recovery scenarios

- **Patch availability**: and *vendor support* are both commonly assessed to determine both how often patching will be required and if the vendor is appropriately supporting the solution

- **Inability to patch** is a consideration when high availability is required and other factors like scalability do not allow the system to be patched without downtime or other interruptions

- **Power** consumption drives ongoing costs and is considered part of datacenter design

- **Compute** requirements also drive ongoing costs in the cloud and up-front and recurring replacement costs for on-premises solutions.

3.2 Given a scenario, apply security principles to secure enterprise infrastructure.

• Infrastructure considerations

- **Device placement**: Devices may be placed to secure a specific zone or network segment, to allow them to access traffic from a network segment, VLAN, or broader network, or may be placed due to capabilities like maximum throughput. Common placement options include at network borders, datacenter borders, and between network segments and VLANs, but devices may also be placed to protect specific infrastructure.

- **Security zones**: are network segments or infrastructure components separated from less secure areas through logical or physical means. They are often based on trust or data sensitivity levels but can be created for any security need. Examples include segregated guest networks, internet-facing networks for hosting public services, and management VLANs used for network device management.

- **Attack surface**: The organization's attack surface, or a device's attack surface, consists of the points at which an unauthorized user could gain access. This includes services, management interfaces, and any other means that attackers could obtain access to or disrupt the organization. Understanding an organization's attack surface is a key part of security and infrastructure design.

- **Connectivity**: involves aspects such as the organization's connection to the internet, redundancy, connection speed, security controls from the provider, types of connectivity (fiber optic, copper, wireless), and whether connection paths are physically separated or use different providers to prevent disruption from a single event.

- **Failure modes**: refer to how an organization handles security device failures. The two main states are **fail-closed** (where no traffic passes through the device if it fails) and **fail-open** (where all traffic passes through). The choice between these modes depends on the organization's business objectives, weighing the risk of being unable to conduct business (fail-closed) versus the risk of lacking security controls (fail-open).

- **Device attribute**: **Network taps** are devices used to monitor or access network traffic and can be **active** (powered) or **passive** (unpowered). Passive taps are more reliable since they can't lose power, eliminating a failure risk. They can be set up either

in-line (where traffic flows through them) or as **monitoring taps** (where they copy traffic without interacting with the original flow).

- **Network appliances**

  o **Jump server**: are secure, monitored systems that allow administrators to access different security zones with varying security levels. They are typically accessed via SSH, RDP, or other remote methods and are equipped with the necessary administrative tools. Jump servers should maintain a secure **audit trail** for investigation purposes, with copies stored in a separate environment for incident analysis.

  o **Proxy server**: centralize requests by accepting and forwarding them, allowing for filtering, modification, and caching of data. They can also enforce access restrictions, such as by IP address.

  There are two types of proxy servers:

  1. **Forward proxies**: Positioned between clients and servers, they forward client requests to servers, anonymizing traffic or bypassing IP/geographic restrictions.

  2. **Reverse proxies**: Positioned between servers and clients, they help with load balancing and caching, allowing clients to query a single system while distributing traffic across multiple systems.

  o **Intrusion prevention system (IPS)/ intrusion detection system (IDS):** are used to detect and block potential threats on networks. They use two main detection methods:

  1. **Signature-based detection**: Detects threats by matching known hashes or signatures.

  2. **Anomaly-based detection**: Flags behavior that deviates from established baselines.

  While **IPS** is deployed in-line to block threats, both **IDS** and **IPS** can also be used in passive mode to detect but not block threats. These systems can be hardware, software, virtual, or cloud-based, with key selection criteria including throughput, detection methods, update availability, and detection rates.

  o **Load balancer**: distribute traffic across multiple systems, providing redundancy and enabling easier upgrades and patching. They are commonly used in web service infrastructures and typically present a **virtual IP (VIP)** for clients to send service requests, which are then distributed to a pool of servers.

There are two main types of load balancer designs:

1. **Active/active**: Multiple systems are online and share the load.

2. **Active/passive**: Backup systems are brought online when active systems fail, commonly used for disaster recovery.

Common load balancing algorithms include:

- **Round-robin**: Distributes traffic equally.

- **Least connection**: Directs traffic to the server with the fewest connections.

- **Agent-based adaptive**: Adjusts traffic distribution based on server conditions.

- **Source IP hashing**: Uses a hash of the client's IP to distribute traffic.

- **Weighted algorithms**: Factor in server capabilities, like **weighted least connection** or **weighted response time**.

o **Sensors**

- **Port security**: helps protect network switches by limiting the number of MAC addresses allowed on a single port, thus preventing issues like MAC address spoofing and CAM table overflows. There are two primary types:

  - **Dynamic Locking**: Sets a maximum number of MAC addresses allowed.
  - **Static Locking**: Restricts access to only specific MAC addresses.

While useful, port security doesn't prevent all forms of unauthorized access and should be combined with other methods like Network Access Control (NAC) for better security.

802.1X: This is a protocol that enforces port-level security by authenticating devices before allowing them to connect to the network. It is commonly used in conjunction with NAC to ensure only authorized devices gain access to network resources. 802.1X is particularly useful in environments with high-security needs, requiring devices to authenticate via methods like usernames, passwords, or certificates before access is granted.

Both **Port Security** and **802.1X** enhance network security at the device connection level, but 802.1X provides a more robust solution by requiring authentication, whereas port security primarily limits traffic based on MAC addresses. These tools are essential in protecting against unauthorized access, CAM table attacks, and network configuration issues.

o **Extensible Authentication Protocol (EAP):** is an authentication framework that is commonly used for wireless networks. Many different implementations exist that use the EAP framework, including vendor-specific and open methods like EAP-TLS, LEAP, and EAPTTLS

- **Firewall types**

o **Web application firewall (WAF):** is designed to monitor and control web traffic, specifically for web applications. It examines traffic, including database queries and API calls, and applies rules to block attacks like SQL injections or cross-site scripting (XSS). It acts like a firewall and intrusion prevention system combined, offering deeper inspection and real-time protection for web servers.

o **Unified threat management (UTM):** devices provide an all-in-one security solution, combining a range of features such as firewalls, IDS/IPS, antimalware, VPN, email filtering, and more. They are typically deployed for smaller to mid-sized organizations for easier, out-of-the-box setup and management. UTM devices are central to network security but may offer less specialized throughput compared to NGFWs.

o **Next-generation firewall (NGFW):** go beyond traditional firewalls by incorporating deep packet inspection, IDS/IPS capabilities, antivirus, and other advanced features. They provide application-level awareness and can detect and block application-specific attacks, but they require more configuration and expertise than UTMs. NGFWs are optimized for high throughput and detailed traffic analysis. Next-generation firewalls (NGFWs) combine both Layer 4 and Layer 7 capabilities, allowing for enhanced security but requiring more CPU, memory, and complex rules to manage application-level traffic effectively.

o **Layer 4/Layer 7**: **Layer 4 firewalls** (Transport Layer) operate at the TCP/UDP level, focusing on IP addresses, protocols, and ports. They filter traffic based on these parameters and are less capable of analyzing the actual content of the traffic.

Layer 7 firewalls (Application Layer) have the ability to inspect traffic more deeply by understanding the context of applications, protocols, and content. This enables them to detect and block application-specific attacks, such as SQL injection or cross-site scripting (XSS).

• Secure communication/access

- **Virtual private network (VPN) & Remote access**: VPNs create secure connections across public networks, allowing endpoints to behave as if they are on the same network. There are two major types of VPNs:

1. **IPSec VPNs**: Operate at layer 3, requiring a client. They can function in **tunnel mode** (encrypting entire packets) or **transport mode** (only encrypting the payload). They are often used for site-to-site connections or for handling non-web traffic.

2. **SSL VPNs**: Use TLS (instead of SSL) and can be accessed via a web portal or a tunnel. SSL VPNs don't require client installation and are favored for their ability to provide segmented access without complex configurations.

VPNs can be categorized into:

- **Remote-access VPNs**: For traveling or remote workers, typically turned on as needed.

- **Site-to-site VPNs**: Used to connect two or more sites, often running continuously with automatic reconnection in case of failure.

- **Tunneling**: VPNs can be configured as **split tunnel** or **full-tunnel**:

- **Full-tunnel VPN**: Sends all network traffic through the VPN, ensuring security for all data, especially when using untrusted networks (e.g., public Wi-Fi). This method provides complete protection but uses more bandwidth.

- **Split-tunnel VPN**: Only sends traffic destined for the remote network through the VPN tunnel, while other traffic bypasses the VPN and goes through the user's local internet connection. This reduces bandwidth usage but leaves traffic outside the VPN unprotected and unmonitored.

Full-tunnel VPNs are more secure, especially when using untrusted networks, while **split-tunnel** VPNs are more bandwidth-efficient but expose some traffic to potential vulnerabilities.

- **Software-defined wide area network (SD-WAN):** is a virtual network solution that integrates multiple connectivity services, such as MPLS, 4G/5G, and broadband, to optimize network performance. SD-WAN enhances **high availability** by dynamically routing traffic based on application needs, and **cost efficiency** by using less expensive connection options when feasible.

SD-WAN is often paired with **Multiprotocol Label Switching (MPLS)**, which uses data labels for routing traffic, ensuring low-latency paths for real-time services like voice and video, while best-effort paths are used for less time-sensitive traffic like email. Despite MPLS being more expensive, SD-WAN is enabling many organizations to move away from MPLS due to its flexibility and cost-effectiveness.

- **Secure access service edge (SASE):** combines **virtual private networks (VPNs)**, **SD-WAN**, and **cloud-based security tools** (like firewalls, cloud access security brokers (CASBs), and zero-trust networks) to offer secure access for devices regardless of their location. It ensures **endpoint security**, protects **data in transit**, and enforces **policy-based security** across an organization's infrastructure and services. SASE is designed to streamline secure access, making it easier to manage security as an integrated service in the cloud.

• Selection of effective controls: *Selection of effective controls* is a key component in securing networks and requires both an understanding of threats and the controls that can address them.

3.3 Compare and contrast concepts and strategies to protect data.

• Data types

- **Regulated**: information includes any data that is governed by laws or regulations. This includes the regulations discussed earlier (HIPAA, GLBA, and PCI DSS), as well as any other rules governing different categories of information

- **Trade secret**: included in intellectual property

- **Intellectual property**: includes *trade secrets*, which encompass proprietary business information that provides a company with a competitive edge, such as formulas, manufacturing processes, strategies, or any other confidential information.

- **Legal information**: *information* includes documents, communications, and records that are related to legal proceedings, contracts, or corporate governance. This might include attorney-client privileged communications, contracts, legal opinions, court records, and regulatory filings

- **Financial** information: includes any personal financial records maintained by the organization. Some of these records may be subject to the provisions of the Gramm–Leach–Bliley Act (GLBA) and/or the Payment Card Industry Data Security Standard (PCI DSS).

- **Human- and non-humanreadable**

• Data classifications

- **Sensitive**: information that requires protection but disclosure would not damage national security. Protection: Restricted access and careful management.

- **Confidential**: information requires some protection. The unauthorized disclosure of Confidential information could reasonably be expected to cause identifiable damage to national security.

- **Public**: information intended for public release with no privacy or security risks. No special protection, accessible to all.

- **Restricted**: less sensitive than critical data but still requires protection to prevent moderate damage. Limited access and controlled management

- **Private**: personal data protected by privacy laws, such as identifiable information. Strong protection and access control.

- **Critical**: essential data for the organization or national security; loss causes severe damage. High protection, encryption, backup, and recovery plans

• General data considerations

- **Data states**

  o **Data at rest**: is stored data that resides on hard drives, tapes, in the cloud, or on other storage media. This data is prone to theft by insiders or external attackers who gain access to systems and are able to browse through their contents.

  o **Data in transit**: is data that is in motion/transit over a network. When data travels on an untrusted network, it is open to eavesdropping attacks by anyone with access to those networks

  o **Data in use**: is data that is actively in use by a computer system. This includes the data stored in memory while processing takes place. An attacker with control of the system may be able to read the contents of memory and steal sensitive information

- **Data sovereignty**: refers to the legal principle that data is subject to the laws of the jurisdiction where it is collected, stored, or processed. This becomes a concern in cloud computing, where data may be distributed across multiple countries and jurisdictions. If customers are unaware of where their data is stored, they may inadvertently fall under the legal requirements of foreign jurisdictions.

To manage this, security professionals need to ensure they understand where and how their data is stored, processed, and transmitted. They may also opt to encrypt data with keys controlled outside the provider's reach, maintaining full control over the data. Some cloud providers allow customers to choose specific geographic regions for data storage, offering more control over data sovereignty.

- **Geolocation**: GPS is a key technology for determining the location of devices, often used in geolocation services like location-aware authentication and geofencing. It is commonly combined with other location data, such as Wi-Fi networks, Bluetooth, and cellular connections, to provide more accurate information. Tools like Apple's Find My use GPS along with Wi-Fi, Bluetooth, cellular data, and sensors to locate devices, while AirTags use nearby Apple devices to assist in tracking

• Methods to secure data

- Geographic restrictions

- Encryption

- Hashing

- Masking

- Tokenization

- Obfuscation

- Segmentation

- Permission restrictions

3.4 Explain the importance of resilience and recovery in security architecture.

• High availability

- Load balancing vs. clustering:

Load Balancing: Distributes network traffic across multiple systems or services, ensuring redundancy and improved performance by spreading the load. It can also enable system upgrades by temporarily redirecting traffic from systems undergoing maintenance.

Clustering: Involves grouping multiple computers to perform the same task, making them act as a single larger system. Clusters provide redundancy and scalability, often used for tasks like web hosting or high-performance computing.

- **Site considerations**

  - **Hot**: have all the infrastructure and data needed to operate the organization. Because of this, some organizations operate them full time, splitting traffic and load between multiple sites to ensure that the sites are performing properly. This approach also ensures that staff are in place in case of an emergency

  - **Cold**: have space, power, and often network connectivity but they are not prepared with systems or data. This means that in a disaster an organization knows they would have a place to go but would have to bring or acquire systems. Cold sites are challenging because some disasters will prevent the acquisition of hardware, and data will have to be transported from another facility where it is stored in case of disaster. However, cold sites are also the least expensive option to maintain of the three types

  - **Warm**: *sites* have some or all of the systems needed to perform the work required by the organization, but the live data is not in place. Warm sites are expensive to maintain because of the hardware costs, but they can reduce the total time to restoration because systems can be ready to go and mostly configured. They balance costs and capabilities between hot sites and cold sites

  - **Geographic dispersion**: ensures that a single disaster, attack, or failure cannot disable or destroy them. For datacenters and other facilities, a common rule of thumb is to place datacenters at least 90 miles apart, preventing most common natural disasters from disabling both (or more!) datacenters. This also helps ensure that facilities will not be impacted by issues with the power grid, network connectivity, and other similar issues

- **Platform diversity:** or diversity of technologies and vendors, is another way to build resilience into an infrastructure. Using different vendors, cryptographic solutions, platforms, and controls can make it more difficult for a single attack or failure to have system- or organization-wide impacts. There is a real cost to using different technologies, such as additional training, the potential for issues when integrating disparate systems, and the potential for human error that increases as complexity increases

- **Multi-cloud systems:** can also help address this resilience need. Large-scale organizations that need continuous operations may opt to use multiple cloud vendors to help ensure that their systems will continue to operate even if a cloud vendor has a problem. That's a level of investment that's beyond most organizations, but it is becoming more accessible as multicloud management and deployment tools mature.

- **Continuity of operations:** or ensuring that operations will continue even if issues ranging from single system failures to wide-scale natural disasters occur, is a design target for many organizations. Not every organization or implementation will use all, or even many, of these design elements. Each control adds complexity and expense

- **Capacity planning:** Resilience requires capacity planning to ensure that capacity including staff, technology, and infrastructure is available when needed.

  - **People**: Ensuring adequate staffing with the right skills to handle increased demand or emergencies. This includes maintaining coverage through global or remote staffing and using third-party support like consultants or cloud services when needed.

  - **Technology**: Involves planning for the scalability of deployed technologies, such as web servers, load balancers, and storage systems. This ensures that technology can meet growing demands without performance degradation.

  - **Infrastructure**: Ensures that physical and network infrastructure (e.g., connectivity, throughput, storage) can be scaled to meet changing loads and support disaster recovery and business continuity efforts.

- **Testing**

  - **Tabletop exercises**: use discussions between personnel assigned roles needed for the plan to validate the plan. This helps to determine if there are missing components or processes. Tabletop exercises are the least potentially disruptive of the testing methods but also have the least connection to reality and may not detect issues that other methods would.

  - **Fail over**: test full failover to an alternate site or system, and they have the greatest potential for disruption but also provide the greatest chance to fully test in a real-world scenario

  - **Simulation**: are drills or practices in which personnel simulate what they would do in an actual event. It is important to ensure that all staff know that the exercise is a simulation, as performing actual actions may cause disruptions.

  - **Parallel processing**: move processing to a hot site or alternate/backup system or facility to validate that the backup can perform as expected. This has the potential for disruption if the processing is not properly separated and the parallel system or site attempts to take over for the primary's data processing

• Backups

- **Onsite/offsite:**

  - **Onsite backups** are quickly accessible and provide immediate retrieval, making them ideal for responding to operational issues. They are often used for daily operational backups to maintain business continuity.

  - **Offsite backups**, which can include tape, disk, or cloud storage, offer the advantage of geographic diversity, reducing the risk of total data loss from a localized disaster. Offsite backups may be stored in secure, controlled environments, either managed by the organization or through third-party services.

  - **Cloud backups** offer a flexible and cost-effective solution, though they can have time and cost limitations for data retrieval. For example, services like Amazon S3 Glacier or Google Coldline provide low-cost, long-term archival storage but slower access times.

  - **Nearline backups** are storage systems that are not immediately available but can be accessed within a reasonable time without human intervention, such as with tape robots or some cloud services.

- **Frequency**: Backup frequency is determined by factors like data change rates, tolerance for data loss, and restoration efforts. Some data, such as database transactions, may require continuous backups, while others may be scheduled daily, weekly, or monthly. The chosen frequency impacts the effort required for restoration, as frequent incremental backups may require a full backup followed by several incremental restorations to recover the most recent data state. The cost and business need also influence the decision on backup frequency.

- **Encryption**

- **Snapshots**: A snapshot captures the full state of a system or device at a specific moment in time. They're commonly used for virtual machines (VMs), allowing restoration to a previous system state or point in time. Snapshots are useful for cloning systems, rolling back before upgrades or patches, and restoring a system to its previous state. Since they are taken live, they can be done with minimal performance impact. While similar to full backups, snapshots can consume significant space, but compression and deduplication technologies often help optimize space usage.

- **Recovery**: Recovery from backups is guided by **Recovery Point Objectives (RPOs)** and **Recovery Time Objectives (RTOs)**. RPOs define how much data loss is acceptable, influencing the frequency of backups and storage costs. Shorter RTOs require faster recovery times, which may increase costs and necessitate designs for rapid restoration. Both RPOs and RTOs are essential for determining the recovery process and its impact on the organization's ability to minimize downtime and data loss.

- **Replication**: focuses on using either synchronous or asynchronous methods to copy live data to another location or device. Unlike backups that occur periodically in most designs, replication is always occurring as changes are made. Replication helps with multisite, multisystem designs, ensuring that changes are carried over to all systems or clusters that are part of an architecture. In synchronous replication designs, that occurs in real time, but a backup cluster may rely on asynchronous replication that occurs after the fact, but typically much more regularly than a backup. In either design or replication

- **Journaling**: creates a log of changes that can be reapplied if an issue occurs. Journaling is commonly used for databases and similar technologies that combine frequent changes with an ability to restore to a point in time. Journaling also has a role to play in virtual environments where journal-based solutions allow virtual machines to be restored to a point in time rather than to a fixed snapshot.

• Power

- **Generators**: systems that are used to provide power for longer outages; and design elements, such as *dual-supply* or multisupply hardware, ensures that a power supply failure won't disable a server

- **Uninterruptible power supply (UPS):** systems that provide battery or other backup power options for short periods of time

Managed.power.distribution.units.(PDUs*)* are also used to provide intelligent power management and remote control of power delivered inside server racks and other environments.

4.0 Security Operations

4.1 Given a scenario, apply common security techniques to computing resources

• **Secure baselines:** Baseline configurations are an ideal starting place to build from to help reduce complexity and make configuration management and system hardening possible across multiple machines even thousands of them, three phases of a baseline's life cycle:

- **Establish**: a baseline, which is most often done using an existing industry standard like the CIS benchmarks with modifications and adjustments made to fit the organization's needs

- **Deploy**: the security baseline using central management tools or even manually depending on the scope, scale, and capabilities of the organization

- **Maintain**: the baseline by using central management tools and enforcement capabilities as well as adjusting the organization's baseline if required by functional needs or other changes

• Hardening targets

- **Mobile devices**: Hardening mobile devices is more challenging than desktop hardening due to limited security options and central management capabilities. However, best practices include:

  - Regular **OS updates and patching**
  - **Enabling remote wipe** for lost or stolen devices
  - **Requiring strong passcodes** and automatic screen locks
  - **Setting wipe thresholds** for excessive failed login attempts
  - **Disabling unused connectivity** options like Bluetooth

  Security benchmarks for iOS and Android are available from the **Center for Internet Security (CIS)** to guide hardening efforts.

- **Workstations**

- **Switches**

- **Routers**

- **Cloud infrastructure**: Organizations use **cloud-native controls**, **third-party solutions**, or both to secure cloud infrastructure.

- **Cloud-native controls** integrate directly with cloud providers, making them cost-effective and user-friendly.
- **Third-party solutions** are often more expensive but provide flexibility for **multicloud environments**.

Both approaches aim to **harden cloud infrastructure** against attacks.

- Servers

- ICS/SCADA

- Embedded systems

- RTOS

- IoT devices

• Wireless devices

- **Installation considerations**: Effective Wi-Fi network design requires proper WAP placement, configuration, and tuning to ensure usability, performance, and security.

- **Site Surveys**: Assess existing networks, physical structures, and signal strength to determine optimal access point (AP) placement.
- **Heat Maps**: Visually represent signal coverage and strength, helping identify dead zones and channel interference.
- **Channel Planning**: Avoids overlapping channels, especially in dense environments, to reduce interference and optimize performance (e.g., using channels 1, 6, and 11 in 2.4 GHz networks).
- **Network Optimization**: Advanced Wi-Fi management tools can automatically adjust channels and power levels to enhance connectivity and security.
- **Wi-Fi Analyzers**: Help map networks, conduct speed tests, and detect rogue devices for better planning and troubleshooting.

Key takeaway: Site surveys, heat maps, and proper channel management are crucial for a high-performing, interference-free Wi-Fi network.

• Mobile solutions

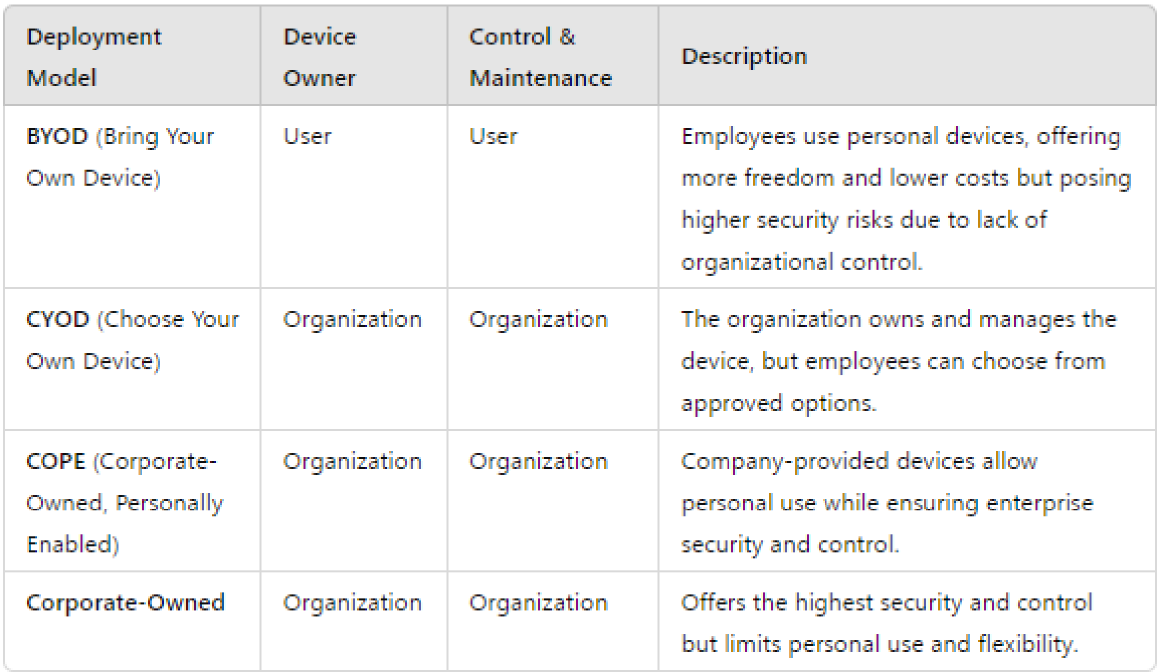
- **Mobile device management (MDM):** Managing mobile devices is challenging due to **hardware variability, OS limitations, and carrier settings**. To address these challenges, organizations use **Mobile Device Management (MDM) and Unified Endpoint Management (UEM) tools** for security and control.

Key Security Features:

- **Application & Content Management**: Controls app installation, deployment, and secure access to organizational data.
- **Remote Wipe**: Ensures lost or stolen devices can be wiped, protecting business data.
- **Geolocation & Geofencing**: Restricts device usage based on location; enables lost device tracking.
- **Authentication & Access Control**: Uses **passwords, biometrics, and context-aware authentication** for security.
- **Containerization & Storage Segmentation**: Separates personal and business data to prevent cross-contamination.
- **Full Device Encryption (FDE)**: Protects data in case of theft, especially when combined with strong authentication.
- **Push Notifications**: Enables remote alerts, security warnings, and communication with lost or stolen devices.
- **Firmware & Device Control**: Monitors **software updates**, detects **rooted/jailbroken devices**, and **restricts unauthorized apps**.
- **Connectivity & Peripheral Restrictions**: Controls **Wi-Fi, Bluetooth, NFC, cameras, and external media** to prevent security risks.

Modern **MDM and UEM solutions provide extensive security controls**, ensuring compliance and protection for mobile devices in corporate environments.

- **Deployment models**: When organizations use mobile devices, one important design decision is the deployment and management model that will be selected. The most common options are:



Deployment Model	Device Owner	Control & Maintenance	Description
BYOD (Bring Your Own Device)	User	User	Employees use personal devices, offering more freedom and lower costs but posing higher security risks due to lack of organizational control.
CYOD (Choose Your Own Device)	Organization	Organization	The organization owns and manages the device, but employees can choose from approved options.
COPE (Corporate-Owned, Personally Enabled)	Organization	Organization	Company-provided devices allow personal use while ensuring enterprise security and control.
Corporate-Owned	Organization	Organization	Offers the highest security and control but limits personal use and flexibility.

- **Connection methods**: Designing a secure network often starts with a basic understanding of the type of network connectivity that you will be deploying or securing

- **Cellular**: Cellular networks enable mobile connectivity through a network of towers and cells. Modern technologies like LTE (4G) and 5G offer increased bandwidth, but 5G requires denser antenna placement. Cellular networks are managed by carriers rather than organizations, making security a third-party concern.

- **Wi-Fi**: Wi-Fi provides wireless networking using 2.4 GHz, 5 GHz, and 6 GHz frequency bands. Various Wi-Fi standards (e.g., Wi-Fi 4, Wi-Fi 5, Wi-Fi 6) offer increasing speeds and improved security, such as WPA2 and WPA3 encryption. Wi-Fi can be deployed in **ad hoc mode** (device-to-device) or **infrastructure mode** (through access points).

- **Bluetooth**: A short-range, low-power wireless technology operating in the 2.4 GHz band. Devices pair using PIN-based authentication. Bluetooth has four security modes, with **Security Mode 4 (SSP)** being the most secure. However, it remains vulnerable to attacks like eavesdropping due to ease of discovery and fixed PIN weaknesses.

• Wireless security settings

- **Wi-Fi Protected Access 3 (WPA3):** is the successor to WPA2 and became mandatory for all Wi-Fi devices by mid-2020. It enhances security, offering improvements in both Personal and Enterprise modes. WPA3-Personal uses Simultaneous Authentication of Equals (SAE) to replace preshared keys and strengthen password-based authentication, making brute-force attacks more difficult. It also implements Perfect

Forward Secrecy (PFS) to protect traffic even if a password is compromised. WPA3-Enterprise introduces stronger encryption and key management features, including an optional 192-bit security mode and improved frame protection. Additionally, Open Wi-Fi networks are upgraded with Wi-Fi Enhanced Open (OWE), using encrypted connections for open networks. Overall, WPA3 provides stronger security compared to WPA2.

- **AAA/Remote Authentication Dial-In User Service (RADIUS):** is a widely used AAA (authentication, authorization, and accounting) system for network and wireless services. It operates in a client-server model over TCP or UDP, with passwords obfuscated using a shared secret and MD5 hash, though its password security is relatively weak. RADIUS traffic is typically encrypted using IPSec or similar protections. In an AAA system, users authenticate with credentials, are authorized based on policies, and their actions are tracked for accounting purposes, such as resource usage.

- **Cryptographic protocols**

- **Authentication protocols**

  - **PEAP (Protected Extensible Authentication Protocol)**: PEAP is an extension of EAP that encapsulates EAP within a secure TLS tunnel, providing better security for wireless networks. It helps protect user credentials during authentication by encrypting the communication between the client and the authentication server, making it more secure than other EAP methods that do not offer encryption.
  - **EAP (Extensible Authentication Protocol)**: EAP is an authentication framework used in network access systems, particularly in wireless networks. It supports various authentication methods like certificates, tokens, and passwords. EAP is commonly used in WPA2 and WPA3 for stronger security in wireless networks, providing flexibility in how users are authenticated.

• Application security

- **Input validation**: is crucial in cybersecurity to prevent attacks like injection and cross-site scripting. It involves verifying user input to ensure it matches expected parameters, reducing vulnerabilities. **Allow listing** is the most effective approach, where only predefined, valid input is accepted. For example, a form may require an age within a specific range. **Deny listing** can also be used when it's difficult to predict all valid inputs, blocking potentially harmful content like SQL commands or HTML tags.

Input validation should always be performed **server-side** to prevent bypass through client-side methods. A key challenge is balancing security and usability, such as ensuring valid names with apostrophes don't trigger security measures.

Parameter Pollution is an attack method where multiple values are sent for the same input variable, aiming to bypass input validation and execute malicious commands. This works if the application only validates the first input value, leaving the second vulnerable to attack. Modern platforms typically protect against this, but vulnerabilities can still occur with unpatched systems or insecure code.

- **Secure cookies**: Web developers can protect against cookie theft by marking cookies with the SECURE attribute. *Secure cookies* are never transmitted over unencrypted HTTP connections. Both servers and web browsers understand that they must only be sent over encrypted channels to protect against session replay attacks

- **Static code analysis** involves reviewing an application's source code to identify vulnerabilities, without running the program. It allows testers to detect issues that other testing methods may overlook, such as logic errors or internal business logic problems. This method can be done manually through "code understanding" or using automated tools. Automated analysis is effective for finding known issues, while manual review helps identify errors introduced by developers. Static code analysis is valuable for uncovering hidden flaws and improving security before code deployment. OWASP offers tools for static code analysis.

- **Code signing** allows developers to confirm their code's authenticity by digitally signing it with a private key. Users can verify the signature with the developer's public key to ensure the code hasn't been tampered with. This process helps prevent running inauthentic or malicious code, such as fake patches intended to compromise security. If only digitally signed updates are accepted, any malicious update will fail verification and be rejected by the system.

- **Sandboxing:** is the practice of running an application in an isolated environment to prevent it from negatively affecting other system resources or applications. It limits the application's permissions, restricting its ability to interact with files, the operating system, or other software. This reduces the risk of malicious or compromised applications causing harm. Sandboxing is useful for testing untrusted software or debugging code in a safe space. It can also be used to contain attackers in a controlled environment, making it appear as if they are attacking a real system while they are isolated.

- **Monitoring:** Implement Security Logging and Monitoring This helps detect problems and allows investigation after the fact

4.2 Explain the security implications of proper hardware, software, and data asset management.

- **Acquisition/procurement process:** Acquisition and procurement processes should involve security best practices and assessment to ensure that assets have appropriate security controls or features, that the companies that provide them have appropriate controls and practices themselves, and that contracts and agreements support the acquiring organization's needs

- **Assignment/accounting**

  - **Ownership**: typically includes identifying owners or managers for devices, systems, software, and data

  - **Classification**: It may also include classification efforts, particularly for data and systems that contain data that the organization considers more sensitive or valuable

- **Monitoring/asset tracking**

  - **Inventory**: Once assets have been acquired, they need to be added to asset inventories and tracked through their lifespan

  - **Enumeration**: is typically associated with scanning to identify assets, and some organizations use port and vulnerability scans to help identify systems that aren't part of their inventory.

- **Disposal/decommissioning:** When systems and devices reach the end of their lifecycle, organizations must establish a decommissioning process. This involves removing the device from service, updating inventories, and ensuring no sensitive data remains. Key steps include securely wiping or destroying storage media. For magnetic media, degaussing can be used to scramble data, while for SSDs or flash drives, secure erase tools are preferred due to wear-leveling concerns. Wiping overwrites data to make it unrecoverable, but full-disk encryption and discarding the encryption key may be a better option. Alternatively, physical destruction of the media (e.g., shredding or incinerating) ensures data cannot be recovered. Certification of decommissioning, including certificates of destruction, helps ensure proper disposal of assets.

  - **Sanitization**: At the end of their life cycle, when devices are retired or fail, or when media needs to be reused, sanitization procedures are employed to ensure that remnant data doesn't leak

  - **Destruction**

  - **Certification**

- **Data retention**: At the end of the life cycle, the organization should implement *data retention* standards that guide the end of the data life cycle. Data should only be kept for as long as it remains necessary to fulfill the purpose for which it was originally collected. At the conclusion of its life cycle, data should be securely destroyed

4.3 Explain various activities associated with vulnerability management.

• Identification methods

- **Vulnerability scan**

- **Application security**

  o **Static analysis**: involves reviewing an application's source code to identify potential issues without running the program. This method allows testers to detect problems that other testing methods might miss, such as hidden logic errors or internal issues. Static analysis can be done using automated tools or manually, known as "code understanding." Automated analysis is effective for finding known issues, while manual review helps uncover errors introduced by developers. Unlike other testing methods, static analysis focuses on understanding the program's structure and intended behavior rather than executing it

  o **Dynamic analysis**: involves executing the code and providing input to test the software's behavior in real-time. It is typically done using automated tools due to the large volume of tests required. Unlike static analysis, which reviews the code without execution, dynamic testing runs the code to identify vulnerabilities through interactions with the user interfaces. It helps find issues that occur during execution, while static analysis highlights vulnerabilities in the code itself, often with remediation suggestions.

  o **Package monitoring**: involves tracking third-party libraries and packages used in development to ensure they are secure. This includes understanding their functionality, staying updated on potential vulnerabilities, and regularly updating to the latest secure versions. Automated tools can help identify outdated or insecure dependencies. It's also crucial to source packages from trusted repositories, as using untrusted sources can introduce vulnerabilities. Developers should investigate any suspicious activity related to packages to maintain the security of their applications.

- **Threat feed**

  o **Open-source intelligence (OSINT):** is publicly available threat information that helps organizations stay informed about cybersecurity risks. While it has become widely accessible, selecting reliable, up-to-date sources and using them effectively is key. Some notable sources include Senki.org, AT&T’s Open Threat Exchange, MISP Threat Sharing, Threatfeeds.io, and government agencies like CISA. These sources support collaborative efforts in identifying and addressing cybersecurity threats.

o **Proprietary/third-party**: involves threat data created and maintained by commercial vendors, government agencies, and security organizations using custom methods. This intelligence is often more reliable and curated than open-source feeds, but it can be expensive and exclusive. It's important to use multiple reliable feeds to avoid outdated or inaccurate data. Threat maps offer geographic views of cyber threats, though geographic attribution can be unreliable due to attackers using anonymizing services.

o **Information-sharing organization**: include organizations like Information Sharing and Analysis Centers (ISACs), which help critical infrastructure sectors share information about cyber and physical threats. Established in 1998 under Presidential Decision Directive-63, ISACs operate on a trust model, offering 24/7 support for incident response and threat analysis. Similar agencies exist globally, such as the UK National Protective Security Authority, providing threat intelligence and resources to government, industry, and academia. These centers help improve resilience by fostering cooperation and information sharing.

o **Dark web**

- **Penetration testing**: involves authorized attempts to exploit an organization's security controls, mimicking real-world attackers. It is a time-consuming process requiring skilled professionals to identify vulnerabilities. While challenging, it is one of the most effective ways for organizations to assess their overall security posture and uncover weaknesses.

- **Responsible disclosure program**: enable security researchers to report vulnerabilities in products to the vendors in a secure, collaborative manner. This helps organizations quickly address security issues with the assistance of the cybersecurity community. **Bug bounty programs** are a type of responsible disclosure initiative that offers financial rewards to researchers for finding vulnerabilities, encouraging them to report issues rather than exploit them

- **System/process audit**

• Analysis

- **Confirmation** involves cybersecurity analysts verifying vulnerabilities identified by scanners. These tools may produce false positives or false negatives.

  - **False Positives** occur when a scanner reports a vulnerability that doesn't actually exist. This can happen due to insufficient system access or errors in the scanner's plugin.
  - **False Negatives** happen when a scanner fails to report an existing vulnerability.

Cybersecurity analysts need to confirm each reported vulnerability by cross-referring with external data, using their expertise, and consulting with other specialists in the organization to validate whether a report is a true or false positive/negative.

- **Prioritizing and categorizing** the vulnerability using tools such as CVSS and CVE that provide an external assessment of the vulnerability

- **Common Vulnerability Scoring System (CVSS):** provides a standardized approach for measuring and describing the severity of security-related software flaws

- **Common Vulnerabilities and Exposures (CVE):** provides a standard nomenclature for describing security related software flaws

- **Vulnerability classification**: vulnerability scanners detect various vulnerabilities, from major issues like SQL injections to minor information disclosure. Familiarity with common vulnerabilities and categories is essential for cybersecurity analysts.

  - **Patch Management** is vital yet often neglected due to limited resources. Vulnerabilities such as outdated operating systems or applications are common scan results, which can be fixed by applying security patches.
  - **Legacy Platforms** pose significant risks when vendors discontinue support, leaving organizations without updates for critical vulnerabilities. If unsupported systems must be used, they should be isolated with compensating controls.

Weak Configurations and Error Messages:

- **Weak Configurations** may include default settings, unsecured accounts, open ports, or excessive permissions. These vulnerabilities require careful examination of vulnerability scan reports.
- **Error Messages** in debug mode can provide attackers with critical information. It's important to disable debug mode on public-facing systems to avoid such risks.

Insecure Protocols and Weak Encryption:

- **Insecure Protocols** (e.g., Telnet, FTP) often lack encryption and expose sensitive data. Replacing them with secure alternatives like SSH and SFTP is essential.
- **Weak Encryption** can be exploited if insecure algorithms (e.g., RC4) or weak keys are used. Switching to stronger ciphers like AES is recommended for securing sensitive data.

- **Exposure factor**: Determine the amount of damage that will occur to the asset if the risk materializes. This is known as the exposure factor (EF) and is expressed as the percentage of the asset expected to be damaged. The exposure factor of a risk that would destroy an asset is 100 percent, whereas a risk that would damage half of an asset has an EF of 50 percent.

- **Environmental variables**

- **Industry/organizational impact**

- **Risk tolerance**: An organization's *risk tolerance* is its ability to withstand risks and continue operations without any significant impact

• **Vulnerability response and remediation:** The outcome of the vulnerability analysis should guide the organization to identify the vulnerabilities that are most in need of *remediation*.

- **Patching**: Apply a patch or other corrective measure to correct the vulnerability

- **Insurance**

- **Segmentation**

- **Compensating controls**: Compensating controls balance the fact that it simply isn't possible to implement every required security control in every circumstance with the desire to manage risk to the greatest feasible degree. In many cases, organizations adopt compensating controls to address a temporary exception to a security requirement. In those cases, the organization should also develop remediation plans designed to bring the organization back into compliance with the letter and intent of the original control

- **Exceptions and exemptions**: **Exceptions** are temporary deviations from security policies, usually granted with compensating controls, and are reviewed periodically. **Exemptions** are permanent exclusions from policies granted for specific reasons, and typically do not require compensating controls.

• **Validation of remediation:** After completing a vulnerability remediation effort, cybersecurity professionals should perform a validation that the vulnerability is no longer present. This is typically done by rescuing the affected system and verifying that vulnerability no longer appears in scan results. In the case of more serious vulnerabilities, internal or external auditors may perform this validation to provide independent assurance that the issue is resolved.

• **Reporting:** The final stage of the vulnerability life cycle is reporting, which involves sharing the findings, actions, and lessons with stakeholders. It includes:

- Summarizing identified vulnerabilities, their severity, and impact.
- Detailing remediation actions like patches, compensating controls, and risk acceptance.
- Highlighting trends, patterns, or areas needing attention.
- Offering recommendations for improving the vulnerability management process, policies, or training.

Regular reporting helps demonstrate cybersecurity commitment and supports continuous improvement by identifying gaps and addressing risks proactively.

4.4 Explain security alerting and monitoring concepts and tools.

• Monitoring computing resources

- **Systems**: are typically done via system logs as well as through central management tools, including those found in cloud services. System health and performance information may be aggregated and analyzed through those management tools in addition to being gathered at central logging servers or services

- **Applications**: may involve application logs, application management interfaces, and performance monitoring tools. This can vary significantly based on what the application provides, meaning that each application and application environment will need to be analyzed and designed to support monitoring.

- **Infrastructure** devices can also generate logs. SNMP and syslog are both commonly used for infrastructure devices. In addition, hardware vendors often sell management tools and systems that are used to monitor and control infrastructure systems and devices

• Activities

- **Log aggregation**: involve collecting and matching various data points, such as event times, systems, user accounts, and other details, to investigate incidents. **SIEM** tools automate this process by correlating data and identifying indicators of compromise, building a complete dataset for analysis. While SIEM tools are common, other centralized logging tools like **syslog-ng** and **rsyslog** also enable log aggregation and analysis. These systems often overlap with other security tools like **SOAR**, creating a comprehensive approach to security response and analysis.

- **Alerting**

- **Scanning**

- **Reporting**: involves identifying trends and providing visibility into changes in logs that could indicate issues or require oversight.

- **Archiving**: ensures logs are retained but not actively used, helping to manage storage space in systems like SIEM. Organizations typically set retention periods (e.g., 30, 60, 90, or 180 days) before archiving or deleting logs. This process ensures logs are manageable for analysis while maintaining the necessary data for compliance and historical review.

- **Alert response and remediation/validation**

    - **Quarantine**: refers to isolating potential threats or incidents detected by alerts to prevent further damage.

    - **Alert tuning**: involves fine-tuning alerts to ensure only critical events trigger notifications, reducing noise from false positives or non-urgent issues. Proper tuning helps prevent **alert fatigue**, where analysts become desensitized to frequent or irrelevant alerts, increasing the risk of missing actual threats. By setting appropriate thresholds and refining alerts, organizations can focus on genuine security incidents.

• Tools

- **Security Content Automation Protocol (SCAP)** led by NIST, standardizes how security-related information is communicated to automate interactions between security components. SCAP includes:

1. **CCE (Common Configuration Enumeration)**: Standardizes system configuration issues.

2. **CPE (Common Platform Enumeration)**: Standardizes product names and versions.

3. **CVE (Common Vulnerabilities and Exposures)**: Standardizes software flaw descriptions.

4. **CVSS (Common Vulnerability Scoring System)**: Standardizes severity measurement of security flaws.

5. **XCCDF (Extensible Configuration Checklist Description Format)**: Specifies checklists and their results.

6. **OVAL (Open Vulnerability and Assessment Language)**: Specifies low-level testing procedures for checklists.

- **Benchmarks**: are used to configure systems to a known security standard. They include guidelines for log settings, ensuring that systems are configured to log critical events and alerts. A well-designed benchmark helps ensure that all endpoints or servers within an organization are set up to log important information effectively, supporting security operations and aiding in consistent monitoring and incident response across systems, services, and devices at scale.

- **Agents/agentless**

- **Security information and event management (SIEM):** are central tools for security monitoring in many organizations. They collect and aggregate log data from various sources (e.g., systems, network devices) and perform correlation, analysis, and alerts using rules, AI, and machine learning. SIEMs can analyze user behavior and even perform sentiment analysis on textual data. They can also capture and analyze raw packet data, aiding incident analysis. SIEM tools often include alerting, reporting, and response capabilities, helping organizations track and address security issues. The term **SIEM** has largely replaced older terms like **SIM** (Security Information Management) and **SEM** (Security Event Management), though some tools still use these terms for specialized functionalities.

- **Antiviruses** are essential in defending against malware in enterprise environments. These tools use several methods to detect and prevent malicious software:

1. **Signature-based detection** identifies known malware by matching its hash or pattern, though this is less effective against evolving malware techniques like polymorphism.

2. **Heuristic-based detection** focuses on the behavior of malware, identifying it based on actions rather than signatures.

3. **AI and Machine Learning** are increasingly used to enhance detection, combining heuristic, signature, and other techniques.

4. **Sandboxing** isolates and runs suspicious software in a controlled environment to observe its behavior and analyze potential threats.

Antimalware tools are deployed on various devices (e.g., desktops, mobile devices, email systems) and remain a critical defense layer, despite challenges from advanced malware that seeks to bypass these protections. Effective deployment involves considering threat types, central management, integration with other security tools, and using multiple technologies to improve detection rates.

- **Data loss prevention (DLP)** tools are essential for safeguarding organizational data from theft or accidental exposure. These tools are deployed across endpoints, networks, and servers to ensure data is protected throughout its lifecycle.
Key features of DLP systems include:

- **Data classification**: Identifies which data needs protection.
- **Labeling/tagging**: Supports classification and management.
- **Policy management and enforcement**: Ensures data handling follows organizational standards.
- **Monitoring and reporting**: Alerts administrators about potential issues.

Some DLP systems also offer encryption for data sent outside protected areas, and tools to safely share data without violating policies (e.g., tokenization or data wiping). DLP systems may track user behavior to detect mistakes or inappropriate data handling, such as over-permissioning or sharing sensitive files.
While DLP tools are vital, they are part of a layered security approach, as techniques like screenshots or data copying can bypass them. Effective DLP requires combining technical tools with policies and awareness to mitigate risks.

- **Simple Network Management Protocol (SNMP) traps**: When a device configured to use SNMP encounters an error, it sends a message known as a *SNMP trap*. Unlike other SNMP traffic, SNMP traps are sent to a SNMP manager from SNMP agents on devices when the device needs to notify the manager. SNMP traps include information about what occurred so that the manager can take appropriate action.

- **NetFlow** is a network protocol developed by Cisco to collect and monitor network traffic data. It provides insights into network flow information, such as the source, destination, and type of data being transmitted. NetFlow helps in identifying traffic patterns, analyzing performance, detecting anomalies, and enhancing security by monitoring the flow of data across a network. It is widely used in network performance monitoring and security operations, including identifying potential attacks or unauthorized activities.

- **Vulnerability scanners**

4.5 Given a scenario, modify enterprise capabilities to enhance security

- **Firewall:** Firewalls are key components in network security, deployed as appliances or on individual devices. There are two main types:

1. **Stateless Firewalls**: These filter packets based on information like IP address, port, and protocol from packet headers. They are basic and review every packet individually.
2. **Stateful Firewalls**: These track the state of traffic and can allow entire traffic flows once a conversation is approved. They use a state table to provide more context for making security decisions, making them more efficient than stateless firewalls.

- **Rules**: Firewalls use rules to determine what traffic is allowed or blocked. These rules typically specify the source (IP, hostname, or domain), destination (IP, hostname, or domain), ports, protocols, and whether traffic is allowed or denied. An example rule might allow traffic from a specific subnet to a web server on a specific port.

Additionally, firewalls can create **screened subnets**, which involve three interfaces: one connecting to the untrusted network (Internet), one creating a secured area, and one forming a public area (DMZ). This setup helps secure and segment the network.

- **Access lists**

- **Ports/protocols**

- **Screened subnets**: often called DMZs, or demilitarized zones, are network zones that contain systems that are exposed to less trusted areas. Screened subnets are commonly used to contain web servers or other Internet-facing devices but can also describe internal purposes where trust levels are different.

• IDS/IPS

- Trends

- Signatures

• Web filter: sometimes called content filters, are centralized proxy devices or agent-based tools that allow, or block traffic based on content rules. These can be as simple as conducting *Uniform Resource Locator (URL) scanning* and blocking specific URLs, domains, or hosts, or they may be complex, with pattern matching, IP reputation, and other elements built into the filtering rules. Like other technologies, they can be configured with allow or deny lists as well as rules that operate on the content or traffic they filter

- **Agent-based**: In agent-based deployments, the agents are installed on devices, meaning that the proxy is decentralized and can operate wherever the device is rather than requiring a network configured to route traffic through the centralized proxy

- **Centralized proxy**: When deployed as hardware devices or virtual machines, they are typically a *centralized proxy*, which means that traffic is routed through the device.

- **Universal Resource Locator (URL) scanning**

- **Content categorization**: they typically provide *content categorization* capabilities that are used for URL filtering with common categories, including adult material, business, child-friendly material, and similar broad topics.

- **Block rules:** *block rules* that stop systems from visiting sites that are in an undesired category or that have been blocked due to reputation, threat, or other reasons

- **Reputation**

• Operating system security

- **Group Policy**: is a feature in Microsoft Windows operating systems that allows administrators to define and control security settings, configurations, and permissions across multiple computers within an Active Directory environment. It helps enforce security policies consistently throughout the organization.

- **SELinux** (Security-Enhanced Linux): is a security module for Linux systems that provides a mechanism for enforcing access control policies. It uses mandatory access controls (MAC) to restrict the actions of users and processes based on predefined policies, enhancing the security of the system.

• Implementation of secure protocols is a common part of ensuring that communications and services are secure

- **Protocol Selection**: Most organizations prioritize using secure protocols when available (e.g., HTTPS over HTTP) to protect data during transmission. Insecure protocols are seen as a risk and should be avoided when possible.
- **Port Selection**: While protocols generally have default ports (e.g., port 80 for HTTP, port 443 for HTTPS), some services use the same port for both secure and insecure versions. For example, Microsoft's SQL Server uses TCP 1433, with secure connections relying on client requests.
- **Transport Method**: Choosing the right transport method, including the version of the protocol (e.g., TLS), is crucial. Insecure protocol versions or downgrade attacks can lead to data vulnerabilities, so enforcing secure protocol versions is key to protecting data.

- **DNS filtering:** helps organizations block access to malicious domains by using a list of prohibited domains, subdomains, and hosts. When a user attempts to access a blocked domain, the DNS response is redirected to a trusted internal site that notifies the user of the block. This method is particularly effective against phishing campaigns, as it allows quick blocking of phishing domains and redirection to a warning site. DNS filters are often updated through threat intelligence feeds, leveraging community knowledge about malicious domains.

- **Email security**

  - **Domain-based Message Authentication Reporting and Conformance (DMARC):** uses SPF and DKIM to authenticate emails. It helps determine whether to accept, reject, or quarantine messages from senders based on their alignment with SPF and DKIM records published in DNS. This protocol enhances email security by allowing domains to specify policies for handling unauthenticated messages.

  - **DomainKeys Identified Mail (DKIM):** adds a digital signature to email headers, ensuring the authenticity of the message's origin. This signature can be verified using the public key stored in the sender's DNS records, providing a method to confirm that the email hasn't been tampered with during transit.

  - **Sender Policy Framework (SPF):** allows organizations to publish a list of authorized email servers in DNS records. It ensures that only these servers can send emails on behalf of the domain, rejecting emails sent from unauthorized sources.

  - Gateway: **Email Security Gateway (Gateway)**: These devices filter both inbound and outbound emails, providing security services such as phishing protection, encryption, attachment sandboxing, ransomware defense, URL analysis, and supporting DKIM, SPF, and DMARC checks. They are crucial for protecting email communication and preventing email-based attacks.

- **File integrity monitoring:** such as Tripwire, help detect unauthorized changes to configuration files and systems by creating signatures or fingerprints of files and monitoring them for alterations. These tools focus on unexpected changes, while allowing normal activities like patching or user interaction. Though essential for system security, FIM tools can be challenging to configure and maintain, as they may generate significant alerts due to frequent file changes. Effective setup and ongoing management are crucial to minimize noise and ensure accurate monitoring.

- **DLP:** Data Loss Prevention (DLP) systems are used to enforce information handling policies and prevent data theft or loss by monitoring systems for sensitive information and network traffic for potential breaches. DLP operates in two modes:

1. **Agent-based DLP**: Involves software agents installed on systems to detect sensitive data (e.g., credit card numbers) and monitor system configuration and user actions. It can block actions like accessing USB drives to prevent data theft.
2. **Agentless (Network-based) DLP**: Devices placed on the network to monitor outbound traffic for unsecured sensitive data, blocking or encrypting transmissions that violate policy.

DLP uses pattern matching to detect sensitive data (e.g., Social Security numbers) and watermarking to tag sensitive documents for tracking. These tools can also automatically encrypt content, particularly in email, to prevent unauthorized sharing.

• Network access control (NAC): is a security measure designed to regulate access to networks by verifying whether a device or system should be allowed to connect. NAC can be agent-based (requiring software installation to perform detailed security checks) or agentless (no installation required, typically browser-based), with agent-based systems providing more comprehensive security validation, such as checking patch levels, antivirus versions, and security settings.

NAC can enforce policies by either denying access to non-compliant devices or placing them in quarantine for remediation. Access checks can occur before (pre-admission) or after (post-admission) network connection, with pre-admission offering higher security by preventing non-compliant devices from connecting.

NAC is often used in combination with 802.1X, a standard for authenticating devices on both wired and wireless networks. It uses centralized authentication (often through EAP) to verify devices before granting network access, which may involve placing the device in a specific network zone or VLAN based on security posture.

• Endpoint detection and response (EDR)/ extended detection and response (XDR): Endpoint Detection and Response (**EDR**) tools are advanced security solutions designed to detect and respond to threats that traditional antimalware tools cannot address. EDR systems monitor endpoint devices using software agents and network monitoring, collecting and analyzing event data. They can detect anomalies and indicators of compromise (IoCs) through automated detection engines, allowing both automated and manual investigation. EDR is effective in handling large volumes of security data and supporting incident response.

Extended Detection and Response (**XDR**) builds on EDR by extending its scope beyond endpoints to encompass the entire technology stack of an organization, including cloud services, email, and other components. XDR tools aggregate logs and data from various sources, applying detection algorithms, AI, and machine learning to analyze and respond to security threats more comprehensively. XDR provides a broader, more integrated view of security across an organization's infrastructure.

• User behavior analytics

4.6 Given a scenario, implement and maintain identity and access management

• Provisioning/de-provisioning user accounts are crucial stages in the user account life cycle.

- **Provisioning** involves creating an account, assigning resources, permissions, and other attributes, and often includes identity proofing to ensure the person requesting the account is who they claim to be. This typically happens during employee onboarding, ensuring the right permissions are granted based on the individual's role, while following the principle of least privilege to minimize unnecessary access.
- **Deprovisioning** refers to the process of terminating an account when a user no longer needs access. This includes removing the account, its permissions, and any related data or artifacts. Deprovisioning helps mitigate the risk of dormant accounts being exploited by attackers. While some organizations may disable accounts temporarily, fully deleting accounts is generally preferred to prevent security risks from re-enabled accounts.

A critical aspect of both processes is preventing permission creep, where users accumulate excessive permissions over time, which can be difficult to manage and lead to security vulnerabilities. Effective permission management, especially with the least privilege approach, is vital to maintaining security within an organization.

• Permission assignments and implications

• Identity proofing: is the process of ensuring that the person who the account is being created for is the person who is claiming the account

• Federation: involves linking identity providers (IdPs) with relying parties (RPs) to allow secure, trusted authentication across different services. In federated identity systems, the IdP validates a user's identity and attests that the user is who they claim to be, enabling access to services provided by service providers (SPs). The attestation process is the formal verification of a user's identity. Key components in federated environments:

- **Principal**: Typically the user.
- **IdPs**: Provide identity and authentication services.
- **SPs**: Offer services to users after authentication.
- **Relying Party (RP):** A service or entity that trusts the IdP for user authentication.

Federation commonly uses standards like SAML, OAuth, and OpenID to ensure interoperability between different systems and services. This approach is particularly useful in cloud services, where federated identity management allows for streamlined user management across multiple platforms.

• **Single sign-on (SSO):** allows users to log in once and access multiple systems or services without needing to reauthenticate. This simplifies user interactions but may compromise security boundaries, necessitating additional authentication in high-security environments. Commonly used in both personal and enterprise applications, SSO enhances user experience while managing identity across various platforms.

- **Lightweight Directory Access Protocol (LDAP)** is a directory service protocol used to manage hierarchical information about users and organizational resources. It provides essential data for identity management and is often integrated with SSO systems to facilitate directory-based authentication. Due to its significance, LDAP must be secured, especially when exposed publicly for service access.
- **Security Assertion Markup Language (SAML)** is an XML-based standard for exchanging authentication and authorization information between identity providers and service providers. It is commonly utilized in federated environments, enabling service providers to accept SAML assertions from multiple IdPs, thus streamlining access across web applications.
- **Open Authorization (OAuth)** is an open standard that enables users to grant third-party applications limited access to their resources without sharing their credentials. OAuth allows users to specify permissions for applications, enhancing security and user control over personal data.

Together, these technologies form the backbone of many web-based SSO and federation systems, promoting efficient and secure access management across various platforms.

• **Interoperability:** Interoperability between identity and authorization systems is enabled by standards and shared protocols, making federation possible

• **Attestation:** Attestation provides validation that a user or identity belongs to the user claiming it

• **Access controls**

- **Mandatory**
- **Discretionary**
- **Role-based**

- Rule-based

- Attribute-based

- Time-of-day restrictions

- Least privilege

• Multifactor authentication: One way to ensure that a single compromised factor like a password does not create undue risk is to use *multifactor authentication (MFA)*.

- **Implementations**

  o **Biometrics**: use unique physical traits for authentication, including:

  - **Fingerprint scanning**
  - **Retina and Iris recognition**
  - **Facial recognition**
  - **Voice recognition**
  - **Vein recognition**
  - **Gait analysis**

  Biometric systems are evaluated by **False Rejection Rate (FRR)** and **False Acceptance Rate (FAR)**, with acceptable standards set by organizations like the FIDO Alliance. User acceptance is crucial, as some systems may not work for everyone, requiring backup methods for authentication.

  o **Hardware/software authentication tokens**

  o **Security keys**

- **Factors**

  - Something.you.know, including passwords, PINs, or the answer to a security question.
  - Something.you.have.like a smartcard, USB or Bluetooth token, or another object or item that is in your possession, like the Titan security key o Something you are
  - Something.you.are which relies on a physical characteristic of the person who is authenticating themselves. Fingerprints, retina scans, voice prints, and even your typing speed and patterns are all included as options for this type of factor

- Somewhere.you.are, sometimes called a location factor, is based on your current location. GPS, network location, and other data can be used to ensure that only users who are in the location they should be can authenticate.

• Password concepts

- **Password best practices**

1. **Length**: Passwords should be long to prevent brute-force attacks, with a focus on length over complexity.
2. **Complexity**: While complexity was historically emphasized, current practices suggest reducing complexity requirements and not requiring special characters. Larger character sets are still useful for preventing attacks.
3. **Reuse**: Limit password reuse to prevent users from selecting compromised passwords.
4. **Expiration**: Passwords should have expiration dates, but organizations with **multifactor authentication (MFA)** may not require frequent changes.
5. **Age**: Password age settings prevent users from resetting passwords repeatedly to bypass reuse limitations.

NIST recommends reducing complexity, allowing ASCII and Unicode characters, using password manager and securing passwords with salting and hashing methods.

- **Password managers**: like 1Password and Bitwarden, help users create, store, and manage secure passwords, along with related data such as notes and URLs. They are commonly available across platforms like **Windows** (Windows Credential Manager) and **macOS** (Apple's Keychain, which syncs with iCloud). Password managers reduce password reuse and encourage the use of complex, long passwords, making them a common recommendation for both individuals and organizations to improve security.

- **Passwordless**: is gaining popularity, relying on factors such as something you **have** (security tokens, one-time password apps, certificates) or something you **are** (biometric data like fingerprints or voice). One common option is a **security key**, which supports protocols like **FIDO** and **U2F** and connects via USB or Bluetooth. These keys interact with systems, requiring a PIN or fingerprint for access instead of a password.

FIDO2 is an open authentication standard that uses key pairs for secure authentication without passwords. This method reduces risks and friction typically associated with traditional password-based systems.

• Privileged access management tools

- **Just-in-time permissions** are permissions that are granted and revoked only when needed. This is intended to prevent users from having ongoing access when they don't need that access on an ongoing basis. Users will typically use a console to "check out" permissions, which are then removed when the task is completed, or a set time period expires. This helps to prevent privilege creeping but does add an additional step for use of privileges

- **Password vaulting**: is commonly used as part of PAM environments to allow users to access privileged accounts without needing to know a password. Much like JIT permissions, password vaulting often allows privileged credentials to be checked out as needed while creating a logged, auditable event related to the use of the credentials. Password vaults are also commonly used to ensure that passwords are available for emergencies and outages

- **Ephemeral credentials** are temporary accounts with limited lifespans. They may be used for guests or for specific purposes in an organization when a user needs access but should not have an account on an ongoing basis. Setting an appropriate lifespan and ensuring that the account is deprovisioned is key to the successful implementation of ephemeral accounts

4.7 Explain the importance of automation and orchestration related to secure operations.

• Use cases of automation and scripting: automation and scripting are powerful tools that can significantly improve efficiency and security.

- **User provisioning**: Automated scripts can handle the process of adding, modifying, or removing user access to systems and networks, reducing manual efforts and human error

- **Resource provisioning**: Scripts can automate the allocation and deallocation of system resources, ensuring optimal performance and reducing the burden on IT staff

- **Guard rails**: Automation can be employed to enforce policy controls and prevent violations of security protocols

- **Security groups**: Automated processes can manage security group memberships, ensuring users have appropriate

- **Ticket creation**: Automation can streamline the ticketing process, enabling immediate creation and routing of issues to the right teams.

- **Escalation**: In case of a major incident, scripts can automate the escalation process, alerting key personnel quickly

- **Enabling/disabling services and access**: Automation can be used to turn services or access on or off based on certain triggers or conditions.

- **Continuous integration and testing**: Scripts can automate the build and test process, ensuring faster and more reliable software delivery.

- **Integrations and Application programming interfaces (APIs):** Automated processes can handle data exchange between different software applications through APIs, enhancing interoperability.

• Benefits

- **Efficiency/time saving**: Automation reduces manual tasks, allowing team members to focus on higher-level tasks.

- **Enforcing baselines**: Automation ensures consistent application of security baselines across systems and networks

- **Standard infrastructure configurations**: Scripts can automate the process of configuring systems, ensuring uniformity and reducing errors.

- **Scaling in a secure manner**: Automation supports rapid scaling of infrastructure while maintaining security controls.

- **Employee retention**: Automation of mundane tasks can increase job satisfaction and employee retention

- **Reaction time**: Automated alerts and responses can significantly reduce the time to react to security incidents

- **Workforce multiplier**: Automation increases the capacity of your team by handling repetitive tasks, effectively acting as a force multiplier.

• Other considerations

- **Complexity**: While automation can simplify many processes, the development and management of automation scripts can be complex and require a high level of technical skill

- **Cost**: Implementing automation and scripting often involves upfront costs, including investment in tools, training, and potentially new staff members with specific expertise

- **Single point of failure**: Over-reliance on automation might lead to a single point of failure where one malfunctioning script or process could impact a significant part of your operations

- **Technical debt**: Over time, as systems evolve and change, automated scripts might become outdated or inefficient, creating a form of "technical debt" that needs to be addressed

- **Ongoing supportability**: Maintaining and updating scripts to ensure they remain effective and compatible with your systems is a continual task that requires dedicated resources

4.8 Explain appropriate incident response activities.

• Process

- **Preparation**: In this phase, you build the tools, processes, and procedures to respond to an incident. That includes building and training an incident response team, conducting exercises, documenting what you will do and how you will respond, and acquiring, configuring, and operating security tools and incident response capabilities.
– **Detection**: This phase involves reviewing events to identify incidents. You must pay attention to indicators of compromise, use log analysis and security monitoring capabilities, and have a comprehensive awareness and reporting program for your staff

- **Analysis**: Once an event has been identified as potentially being part of an incident, it needs to be analyzed. That includes identifying other related events and what their target or impact is or was.

- **Containment:** Once an incident has been identified, the incident response team needs to contain it to prevent further issues or damage. Containment can be challenging and may not be complete if elements of the incident are not identified in the initial identification efforts. This can involve *quarantine*, which places a system or device in an isolated network zone or removes it from a network to ensure that it cannot impact other devices.

- **Eradication**: The eradication stage involves removing the artifacts associated with the incident. In many cases, that will involve rebuilding or restoring systems and applications from backups rather than simply removing tools from a system since proving that a system has been fully cleaned can be very difficult. Complete eradication and verification are crucial to ensuring that an incident is over.

- **Recovery**: Restoration to normal is the heart of the recovery phase. That may mean bringing systems or services back online or other actions that are part of a return to operations. Recovery requires eradication to be successful, but it also involves implementing fixes to ensure that whatever security weakness, flaw, or action that allowed the incident to occur has been remediated to prevent the event from immediately occurring

- **Lessons learned**: These sessions are important to ensure that organizations improve and do not make the same mistakes again. They may be as simple as patching systems or as complex as needing to redesign permission structures and operational procedures. Lessons learned are then used to inform the preparation process, and the cycle continues.

• **Training:** Appropriate and regular training is required for incident responders to be ready to handle incidents of all types. Organizations often invest in training for their staff, including incident response certifications

• **Testing**

- **Tabletop exercise**: are used to talk through processes. Team members are given a scenario and are asked questions about how they would respond, what issues might arise, and what they would need to do to accomplish the tasks they are assigned in the IR plan. Tabletop exercises can resemble a brainstorming session as team members think through a scenario and document improvements in their responses and the overall IR plan

- **Simulation**: can include a variety of types of events. Exercises may simulate individual functions or elements of the plan or only target specific parts of an organization. They can also be done at full scale, involving the entire organization in the exercise. It is important to plan and execute simulations in a way that ensures that all participants know that they are engaged in exercise so that no actions are taken outside of the exercise environment

• **Root cause analysis** is conducted after mitigating an issue to identify its underlying cause and prevent recurrence. It helps organizations fix vulnerabilities and address systemic issues. Common RCA techniques include:

- Five Whys – Repeatedly asking "why" to uncover the root cause.
- Event Analysis – Evaluating events to determine if they are causes or effects.
- Cause-and-Effect Diagramming – Using tools like fishbone diagrams to map relationships.

RCA is a crucial part of incident response, feeding into the preparation phase to prevent similar future incidents. Effective mitigation and recovery also involve allow/deny lists, isolation, quarantine, and continuous monitoring.

• Threat hunting: supports the detection and analysis phases of incident response by identifying Indicators of Compromise (IoCs) associated with attacks. Key IoCs include:

- **Account lockout** – Often caused by brute-force attempts.
- **Concurrent session usage** – Multiple logins from different locations may signal credential compromise.
- **Blocked content – Access** attempts to restricted domains could indicate malware activity.
- **Impossible travel – Logins** from geographically distant locations within a short time suggest credential theft.
- **Unusual resource consumption – Excessive disk or bandwidth** use may signal compromise.
- **Resource inaccessibility – Unexpected** loss of access might indicate an attack.
- **Out-of-cycle logging** – Unusual login times or process executions may be suspicious.
- **Missing logs** – Attackers may delete logs to cover their tracks.

IoCs are often analyzed collectively to detect incidents. Published/documented IoCs from threat feeds and information-sharing organizations help in proactive defense. Understanding the incident response process and IoC analysis is critical for security professionals.

• Digital forensics

- **Legal hold**: is a notice requiring an organization to preserve relevant data, preventing **spoliation of evidence** (altering, destroying, or withholding information). Mishandling data after receiving a legal hold can have serious legal consequences.

- **Chain of custody**: Throughout acquisition and the forensic life cycle, maintaining a chain of custody helps ensure that evidence is admissible in court.
To determine if evidence is admissible, criteria such as the relevance and reliability of the evidence, whether the evidence was obtained legally, and whether the evidence is authentic are applied

- **Acquisition**: Forensic data acquisition follows the **order of volatility**, capturing the most perishable data first. The hierarchy includes:

1. **CPU cache & registers** – Highly volatile, rarely captured.

2. **Ephemeral data** – Process tables, ARP cache, and kernel stats.
3. **RAM** – Stores encryption keys and application data.
4. **Swap & pagefile** – Supplements RAM and provides process insights.
5. **Disk files & data** – Less volatile but crucial for investigations.
6. **OS artifacts** – Windows Registry and system logs.
7. **Devices & firmware** – Includes mobile, IoT, and specialized systems.
8. **VM snapshots & network logs** – Useful for reconstructing events.
9. **Physical artifacts** – Printouts, media, and external evidence.

Preventing Malicious Data Acquisition

- **USB Data Blockers** prevent unauthorized data access via charging ports.
- **Firmware analysis** detects tampering and forensic artifacts.

Maintaining proper forensic procedures ensures data integrity and legal compliance, crucial for court-admissible evidence.

- **Reporting**: **forensic report** is the key product of an investigation, summarizing findings clearly without excessive technical detail. A typical report includes:

- **Summary** of the investigation and key findings.
- **Outline of the forensic process**, including tools used and assumptions.
- **Detailed findings** for each examined device or drive.
- **Accurate conclusions** backed by evidence.
- **Recommendations** based on findings.

- **Preservation**: **Validation and preservation of forensic data is a key part of the forensic process.** Hashing drives and images ensures that the acquired data matches its source. Preservation requires following chain-of-custody processes as well as forethought about the use of write blockers, forensic copies, and documented processes and procedures.

- E-discovery: is the electronic process of gathering and analyzing evidence in legal cases, public records requests, and investigations. It follows the **Electronic Discovery Reference Model (EDRM)**, which includes:

1. **Information Governance** – Managing data before a case arises.
2. **Identification** – Locating relevant electronically stored information (ESI).

3. **Preservation** – Preventing data from being altered or deleted.
4. **Collection** – Gathering the necessary data.
5. **Processing** – Filtering and preparing data for analysis.
6. **Review** – Ensuring only relevant data is included.
7. **Analysis** – Identifying key topics, individuals, and trends.
8. **Production** – Providing data to involved parties.
9. **Presentation** – Using the data in court or expert analysis.

Preserving electronic data can be challenging, especially with frequently modified information.

4.9 Given a scenario, use data sources to support an investigation

• Log data

- **Firewall logs**: can provide information about blocked and allowed traffic, and with more advanced firewalls like NGFW or UTM, devices can also provide application-layer details or IDS/IPS functionality along with other security service–related log information.

- **Application logs**: for Windows include information like installer information for applications, errors generated by applications, license checks, and any other logs that applications generate and send to the application log. Web servers and other devices also generate logs like those from Apache and Internet Information Services (IIS), which track requests to the web server and related events. These logs can help track what was accessed, when it was accessed, and what IP address sent the request. Since requests are logged, these logs can also help identify attacks, including SQL injection (SQLi) and other web server and web application–specific attacks.

- **Endpoint logs**: such as application installation logs, system and service logs, and any other logs available from endpoint systems and devices.

- **OS-specific security logs**: for Windows systems store information about failed and successful logins, as well as other authentication log information. Authentication and security logs for Linux systems are stored in /var/log/auth.log and /var/log/secure

- IPS/IDS logs: provide insight into attack traffic that was detected or, in the case of IPS, blocked

- **Network logs**: can include logs for routers and switches with configuration changes, traffic information, network flows, data captured by *packet analyzers* like Wireshark

- **Metadata**: is **data about data** and is valuable for **incident response** and **forensic investigations**. It is generated as part of system operations and communications. Common types include:

- **Email Metadata** – Contains sender, recipient, timestamps, routing details, antispam data, and attachment info.
- **Mobile Metadata** – Includes call logs, SMS records, GPS locations, and cellular tower data, offering geospatial tracking.
- **Web Metadata** – Found in websites as metatags, headers, cookies, and tracking data for SEO, advertising, and functionality.
- **File Metadata** – Provides details on creation/modification, author, GPS location, camera settings, and device information

• Data sources

- **Vulnerability scans**: Organizations regularly conduct vulnerability scans designed to identify potential vulnerabilities in their environment. One of these scans might identify a server that exposes TCP port 22 to the world, allowing brute-force SSH attempts by an attacker

- **Automated reports**: Other useful data can be found in *automated reports* from various systems and services and *dashboards* that are available via management tools and administrative control panels

- **Dashboards**

- **Packet captures**: can be used to review network traffic as part of incident response or troubleshooting activities.

5.0 Security Program Management and Oversight

5.1 Summarize elements of effective security governance

• Guidelines: provide advice to organizations seeking to comply with the policy and standards**.**

• Policies

- **Acceptable use policy (AUP):** provides network and system users with clear direction on permissible uses of information resources

- **Information security policies**: provides high-level authority and guidance for the security program

- **Business continuity & Disaster recovery**: outline the procedures and strategies to ensure that essential business functions continue to operate during and after a disaster, and that data and assets are recovered and protected

- **Incident response**: describes how the organization will respond to security incidents

- **Software development lifecycle (SDLC):** *policy* that establishes the processes and standards for developing and maintaining software, ensuring that security is considered and integrated at every stage of development

- **Change management** and.change.control.policies that describe how the organization will review, approve, and implement proposed changes to information systems in a manner that manages both cybersecurity and operational risk

• Standards

- **Password**: set forth requirements for password length, complexity, reuse, and similar issues

- **Access control**: describe the account life cycle from provisioning through active use and decommissioning. This policy should include specific requirements for personnel who are employees of the organization as well as third-party contractors. It should also include requirements for credentials used by devices, service accounts, and administrator/root accounts

- **Physical security** standards establish guidelines for securing the physical premises and assets of the organization. This includes security measures like access control systems, surveillance cameras, security personnel, and policies regarding visitor access, protection of sensitive areas, and handling of physical security breaches.

- **Encryption** *standards* specify the requirements for encrypting data both in transit and at rest. This includes the selection of encryption algorithms, key management practices, and the conditions under which data must be encrypted to protect the confidentiality and integrity of information

• Procedures: *Procedures* are detailed step-by-step processes that individuals and organizations must follow in specific circumstances. Like checklists, procedures ensure a consistent process for achieving a security objective

- **Change management**: describe how the organization will perform change management activities that comply with the organization's change management policy, including the possible use of version control and other tools

- **Onboarding/offboarding**: describe how the organization will add new user accounts as employees join the organization and how those accounts will be removed when no longer needed

- **Playbooks**: describe the actions that the organization's incident response team will take when specific types of incidents occur

• **External considerations:** As you prepare the documents in your policy framework, you should not only take into account your organization's business objectives but also consider external considerations that may impact your policies:

- **Regulatory & Legal requirements** that mandate the use of certain controls

- **Industry**: specific considerations that may alter your approach to information security

- **Local/regional/National/Global**: Jurisdiction-specific considerations based on global, national, and/or local/regional issues in the areas where you operate

• **Monitoring and revision:** Policy monitoring is a continuous process to evaluate the effectiveness of an organization's information security policies. Using tools like SIEM systems, audits, and assessments, organizations ensure compliance with security needs, regulations, and technological changes.
When gaps or inconsistencies are found, policy revision is necessary. Updates should be communicated promptly, and training may be required for effective compliance. Regular monitoring and timely updates help maintain a strong and adaptive security posture.

• **Types of governance structures**

- Boards & Committees: In addition to using a formal board of directors, governance structures may incorporate a variety of internal committees consisting of subject matter experts(SMEs) and managers

- **Government entities**: such as regulatory agencies, may also play a role in the governance of some organizations. For example, banks may be regulated by the U.S. Treasury Department or similar agencies in other countries.

- Centralized.governance.models use a top-down approach where a central authority creates policies and standards, which are then enforced throughout the organization.

- Decentralized.governance.models use a bottom-up approach, where individual business units are delegated to achieve cybersecurity objectives and then may do so in the manner, they see fit.

• Roles and responsibilities for systems and data

- **Owners**: One of the most important things that we can do to protect our data is to create clear *data ownership* policies and procedures. Using this approach, the organization designates specific senior executives as the data owners for different data types. For example, the vice president of Human Resources might be the data owner for employment and payroll data, whereas the vice president for Sales might be the data owner for customer information.

- Data.subjects are individuals whose personal data is being processed. This can include customers, employees, and partners. Data subjects often have rights regarding their data, such as the right to access, correct, or request the deletion of their data.

- Data.controllers are the entities who determine the reasons for processing personal information and direct the methods of processing that data. This term is used primarily in European law, and it serves as a substitute for the term data owner to avoid a presumption that anyone who collects data has an ownership interest in that data.

- Data.processors are service providers that process personal information on behalf of a data controller. For example, a credit card processing service might be a data processor for a retailer. The retailer retains responsibility as the data controller but uses the service as a data processor.

- Data.stewards are individuals who carry out the intent of the data controller and are delegated responsibility from the controller.

- Data.custodians are individuals or teams who do not have control or stewardship responsibility but are responsible for the secure safekeeping of information. For example, a data controller might delegate responsibility for securing PII to an information security team. In that case, the information security team serves as a data custodian.

5.2 Explain elements of the risk management process.

• Risk identification: involves recognizing threats and vulnerabilities in an organization's environment. Risks can originate from various sources, including hackers, natural disasters, and internal failures. Key risk categories include:

- **External risks** (cyber threats, natural disasters)
- **Internal risks** (insider threats, user errors, equipment failures)
- **Multiparty risks** (shared risks, such as SaaS breaches)
- **Legacy system risks** (outdated systems with unpatchable vulnerabilities)

- **Intellectual property (IP)** theft risks (exposure of proprietary information)
- **Software compliance/licensing risks** (violations of usage agreements leading to financial or legal consequences)

Identifying and mitigating these risks is crucial for maintaining security and business continuity.

• **Risk assessment:** evaluates risks based on likelihood (probability of occurrence) and impact (magnitude of consequences) to determine their severity. Key Concepts:

- **Likelihood**: The probability of a risk occurring within a specific timeframe.
- **Impact**: The consequences if the risk materializes (e.g., financial loss, legal penalties).
- **Risk Severity**: Determined by combining likelihood and impact.

Types of Risk Assessments:

1. **One-time**: Conducted at a specific moment for a snapshot of risk.
2. **Ad hoc**: Performed in response to a specific event or change.
3. **Recurring**: Conducted periodically (e.g., annually, quarterly) to track risk evolution.
4. **Continuous**: Ongoing monitoring with automated tools for real-time risk detection.

Laws and regulations, such as GDPR, can significantly influence the impact assessment of risks. Regular risk assessments help organizations adapt to evolving threats and compliance requirements.

• **Risk analysis**: is a structured approach to prioritizing risks, using **quantitative** and **qualitative** methods. Organizations often combine both to communicate risk factors effectively.

1. Quantitative Risk Analysis

Uses numerical data to assess risk severity and prioritize mitigation efforts. Key steps include:

1. **Asset Value (AV):** Determine the financial worth of the asset.
2. **Annualized Rate of Occurrence (ARO):** Estimate how often the risk will happen per year.
3. **Exposure Factor (EF):** Assess the percentage of asset damage if the risk occurs.
4. **Single Loss Expectancy (SLE):** SLE=AV×EFSLE = AV \times EFSLE=AV×EF (expected loss per event).

5. **Annualized Loss Expectancy (ALE):** ALE=SLE×AROALE = SLE \times AROALE=SLE×ARO (expected yearly loss).

Example: If a **DoS attack** occurs **3 times a year** and results in a **$2,700 loss per attack**, the **ALE** would be **$8,100 per year**. Organizations use this to decide if mitigation strategies (e.g., DoS protection services) are cost-effective.

2. Qualitative Risk Analysis

Uses subjective ratings (e.g., **Low/Medium/High**) instead of numerical values. This is useful for risks that are hard to quantify, such as **reputational damage or employee morale**.

A risk matrix visually prioritizes risks based on **likelihood and impact**. Example: If **stolen unencrypted devices** and **spear-phishing attacks** have **high probability and high impact**, they should be prioritized over less severe risks.

Key Takeaway:

- **Quantitative analysis** provides clear financial impact estimates.
- **Qualitative analysis** helps evaluate risks that lack precise numerical data.
- A combined approach helps organizations **prioritize mitigation efforts efficiently**.

- **Risk register**: Risk managers use a **risk register** to track risks faced by the organization. It contains detailed information about each risk, but for senior leaders, a **risk matrix** (or heat map) is often used to provide a quick summary, highlighting the most significant risks.

Key Elements of a Risk Register:

1. **Risk Owner:** individual or entity responsible for managing and monitoring risks, including implementing necessary controls and actions to mitigate them
2. **Risk Threshold:** is related to its risk appetite, but it is a more specific term. The risk threshold is the specific level at which a risk becomes unacceptable. It is the actual boundary that, when crossed, will trigger some action or decision. The risk threshold is usually more quantitative, defining clear points or values.
3. **Key Risk Indicators (KRIs):** are metrics used to measure and provide early warning signals for increasing levels of risk. These indicators help in tracking the effectiveness of risk mitigation efforts and make sure that the residual risk stays within the risk appetite.

- **Risk tolerance:** An organization's *risk tolerance* is its ability to withstand risks and continue operations without any significant impact.

- **Risk appetite**: An organization's **risk appetite** defines how much risk it is willing to accept to achieve its goals. The more risk an organization is willing to take, the higher the potential reward, but also the greater the chance of failure. Conversely, organizations with a lower risk appetite generally have more stable success but fewer rewards.

Types of Risk Appetite:

1. **Expansionary Risk Appetite:** Willing to take higher risks for potentially higher rewards, suitable for aggressive growth, innovation, or market capture.
2. **Neutral Risk Appetite:** Takes a balanced approach, accepting moderate risks for steady growth and returns.
3. **Conservative Risk Appetite:** Avoids high risks, focusing on stability and asset protection, often common in regulated industries.

• Risk management strategies

- **Transfer**: **Risk transference** involves shifting the impact of a risk from the organization to another entity, typically through purchasing insurance. For example, by paying a premium to an insurance provider, the organization transfers the financial burden of specific risks, such as theft, to the insurer.

  - **Example**: Property insurance might cover the cost of a stolen laptop.
  - **Cybersecurity risks**: General business policies often exclude cybersecurity risks, so organizations should purchase separate **cybersecurity insurance** for coverage against incidents like DDoS attacks, which may cover recovery costs and lost revenue.

- **Accept**: Risk acceptance should not be confused with neglecting the risk. It should be a conscious decision to deliberately choosing to take no other risk management strategy and to simply continue operations as normal in the face of the risk

  o **Exeption**: if a particular risk does not align with the established policy but the cost of mitigation is too high, an *exception* can be granted for that specific case. This means the organization acknowledges the risk and has decided to accept it for certain reasons.

  o **Excemption**: *Exemptions* are like exceptions but are generally more formal. They may require a higher level of approval and are often documented to ensure that there is a record of the decision-making process. Exemptions might also have an expiration date and need to be reviewed periodically.

- **Avoid**: is a strategy that seeks to eliminate the potential for a risk by changing business practices. While it may seem appealing to avoid risks entirely, it often has significant negative impacts on the organization.

- **Example 1**: To avoid laptop theft, the organization could ban laptops, but this would harm productivity and be unpopular with employees.
- **Example 2**: To avoid a DDoS attack, the organization could shut down its website entirely, but this would be impractical and defeat the purpose of having an online presence.

Risk avoidance often leads to undesirable consequences that outweigh the benefits of eliminating the risk.

- **Mitigate**: involves applying security controls to reduce the likelihood and/or impact of risks. This strategy is the most used by security professionals, focusing on the design, implementation, and management of security measures.

  - **Example 1**: To mitigate the risk of laptop theft, options like cable locks and tamperproof registration tags can be used to reduce the probability of theft and its impact if it occurs.
  - **Example 2**: To mitigate the risk of a DDoS attack, increasing bandwidth or using third-party DDoS mitigation services can reduce the attack's impact or prevent it altogether.

Risk mitigation typically involves a combination of security controls to address both the probability and magnitude of risks.

• Risk reporting: is a critical aspect of risk management, involving the communication of the status and evolution of risks to stakeholders. It ensures that decision-makers are informed and can prioritize risk mitigation strategies effectively. Risk reporting include:

- **Regular Updates**: Routine reports on risk status, control effectiveness, and recent developments.
- **Dashboard Reporting**: Visual aids (e.g., graphs, charts) for quick, real-time understanding of key risk indicators.
- **Ad Hoc Reports**: Produced as needed for specific events or in-depth analysis.
- **Risk Trend Analysis**: Analyzes historical data to identify trends and predict future risks.
- **Risk Event Reports**: Documents specific incidents and their impacts.

Reports should be tailored to the audience and provide clear, concise information, ensuring that they align with the organization's risk appetite and support informed decision-making.

• Business impact analysis: is a formal process used to identify mission-critical functions and the systems supporting them. It helps evaluate the potential impact of disruptions and plan for recovery. Key metrics in BIA include:

- **Mean Time Between Failures (MTBF)**: Measures system reliability, indicating the average time between system failures.
- **Mean Time to Repair (MTTR)**: The average time to restore a system after failure.
- **Recovery Time Objective (RTO)**: The maximum acceptable downtime before a system must be repaired.
- **Recovery Point Objective (RPO)**: The maximum amount of data loss an organization can tolerate during an outage.

Additionally, organizations must identify **single points of failure**, such as systems or devices that, if they fail, could cause a disruption. Redundancy helps mitigate these risks (e.g., adding a backup power supply or server).

5.3 Explain the processes associated with third-party risk assessment and management.

• Vendor assessment: is an ongoing process to ensure vendors maintain the required security, performance, and compliance standards. Key methods include:

- **Penetration Testing**: Simulated attacks on a vendor's systems to identify vulnerabilities and weaknesses.
- **Right-to-Audit Clause**: A clause in vendor agreements allowing the organization to perform or commission audits to verify compliance with terms and regulations.
- **Evidence of Internal Audits**: Requesting documentation of audits conducted by the vendor, which provide insights into internal controls and risk management practices.
- **Independent Assessments**: Involving third-party experts to objectively evaluate the vendor’s security, systems, and certifications (e.g., ISO 27001, SOC reports).
- **Supply Chain Analysis**: Assessing the vendor's suppliers and the associated risks that could impact service delivery, including interdependencies that could affect performance.

Additionally, organizations can use tailored questionnaires to regularly gather information on vendor practices, security, and business continuity.

• Vendor selection

- **Due diligence**

- **Conflict of interest**

• Agreement types

- **Service level agreement (SLA):** there are written contracts that specify the conditions of service that will be provided by the vendor and the remedies available to the customer if the vendor fails to meet the SLA. SLAs commonly cover issues such as system availability, data durability, and response time.

- **Memorandum of agreement (MOA):** is a formal document that outlines the terms and details of an agreement between parties, establishing a mutual understanding of the roles and responsibilities in fulfilling specific objectives. MOAs are generally more detailed than MOUs and may include clauses regarding resource allocation, risk management, and performance metrics.

- **Memorandum of understanding (MOU)**: is a letter written to document aspects of the relationship. MOUs are an informal mechanism that allows the parties to document their relationship to avoid future misunderstandings. MOUs are commonly used in cases where an internal service provider is offering a service to a customer that is in a different business unit of the same company

- **Master service agreement (MSA):** provide an umbrella contract for the work that a vendor does with an organization over an extended period of time. The MSA typically includes detailed security and privacy requirements. project-specific details and references the MSA.

- **Work order (WO)/statement of work (SOW):** Each time the organization enters into a new project with the vendor, they may then create a *work order (WO)* or a *statement of work (SOW)* that contains

- **Non-disclosure agreement (NDA):** are legal contracts that require employees to protect confidential information they access during their employment. Organizations typically have new employees sign an NDA upon hire and periodically remind them of their responsibilities. During offboarding, exit interviews often include a final reminder that employees must continue to adhere to the NDA terms even after leaving the organization.

- **Business partners agreement (BPA):** exist when two organizations agree to do business with each other in a partnership. For example, if two companies jointly develop and market a product, the BPA might specify each partner's responsibilities and the division of profits

• **Vendor monitoring**: is vital for managing third-party risks by continuously tracking a vendor's performance and compliance with contractual obligations. Key aspects include:

- **Rules of Engagement**: Define boundaries, communication protocols, responsibilities, and issue resolution processes.
- **Performance Monitoring**: Establish and track key performance indicators (KPIs) to assess vendor performance.
- **Security Monitoring**: Ensure vendors maintain proper security practices and comply with security standards.
- **Compliance Monitoring**: Verify vendor compliance with legal, regulatory, and industry requirements, including certifications.
- **Financial Monitoring**: Assess the vendor's financial health to ensure long-term viability.
- **Issue Resolution**: Address any identified issues with formal meetings, corrective actions, or contract termination if needed.

• **Questionnaires:** Organizations can employ *questionnaires* to collect information regarding the vendor's practices and performance regularly. These questionnaires can be tailored to focus on specific areas of concern, such as security policies, data handling procedures, and business continuity planning.

• **Rules of engagement(RoE):** are essential in defining the scope and guidelines for penetration testing. Key elements include:

1. **Timeline**: Set the testing schedule, considering critical and non-critical times.
2. **Scope**: Define which locations, systems, applications, and third-party providers are included or excluded.
3. **Data Handling**: Specify confidentiality and disposal requirements for sensitive information gathered during the test.
4. **Expected Behaviors**: Clarify defensive measures that may affect test outcomes, such as shunning or blocking penetration attempts.
5. **Resource Commitment**: Identify necessary resources like time from administrators or developers for testing.
6. **Legal Concerns**: Address applicable laws and remote locations involved in the test.
7. **Communication**: Define when and how updates will occur during the test, especially in case of a compromise discovery.

8. **Permission**: Ensure proper authorization is obtained, with documentation protecting against legal repercussions.
9. **Scoping Agreements**: Document test limitations and disclaimers about its validity and methodology.
10. **Problem Handling**: Establish communication and escalation procedures for dealing with disruptions or issues during testing.

Clear RoE ensures the penetration test runs smoothly, minimizes risks, and defines responsibilities and expectations for both parties.

5.4 Summarize elements of effective security compliance

• **Compliance reporting**: involves both **internal** and **external** processes to ensure adherence to regulations and maintain transparency.

1. **Internal Compliance Reporting**:
   - Focuses on internal management, providing regular reports to leadership (management or board).
   - These reports highlight compliance status, identify gaps, and recommend improvements.
   - Crucial for decision-makers to allocate resources, align compliance with strategic goals, and ensure security posture.
2. **External Compliance Reporting**:
   - Mandated by regulatory bodies or contractual obligations.
   - Involves submitting documentation and evidence to demonstrate compliance with laws and regulations.
   - Essential for maintaining good standing with authorities, avoiding penalties, and building trust with customers and partners (e.g., PCI SSC for credit card data, GDPR compliance).

Both types of reporting are vital for maintaining regulatory adherence and trust.

• **Consequences of non-compliance**: Failure to comply with regulations can have serious repercussions, including:

1. **Fines**:
   - Regulatory bodies can impose substantial fines for noncompliance, such as up to 4% of annual global turnover or €20 million under GDPR.

2. **Sanctions**:
    - Noncompliance can result in nonfinancial sanctions, like restrictions on business operations or revocation of licenses. For example, a financial institution could lose its banking license for breaching anti-money laundering regulations.
3. **Reputational Damage**:
    - Publicized noncompliance, especially data breaches or privacy violations, can severely damage an organization's reputation, leading to loss of customer trust and business.
4. **Loss of License**:
    - Critical business licenses may be suspended or revoked, impacting core operations and business viability.
5. **Contractual Impacts**:
    - Noncompliance can lead to the termination of contracts, causing lost revenue and additional costs in establishing new partnerships.

Overall, noncompliance can result in financial loss, legal issues, and long-term damage to the organization's reputation and business operations. Regular audits, training, and adherence to regulations are vital to avoid these consequences.

• Compliance monitoring: Effective compliance monitoring ensures adherence to legal, regulatory, and contractual obligations. Key elements include:

1. **Due Diligence/Care**:
    - **Due Diligence** involves continuous research to understand evolving laws and regulatory requirements.
    - **Due Care** ensures that policies and controls are effective and regularly updated to maintain compliance.
2. **Attestation and Acknowledgement**:
    - **Acknowledgment** ensures that employees and partners are aware of compliance requirements.
    - **Attestation** confirms that they adhere to these requirements in practice.
3. **Internal and External Monitoring**:
    - **Internal Monitoring** involves audits and checks within the organization to ensure policy adherence and legal compliance.

  - **External Monitoring** includes third-party assessments to provide an unbiased view of compliance status.

4. **Automation**:
   - Automated compliance solutions track regulatory changes, monitor violations, and ensure consistent application of policies, saving time, reducing errors, and supporting detailed reporting for further improvements.

Overall, continuous diligence, both internal and external, combined with automation, is crucial to maintain compliance effectively.

• **Privacy**: Cybersecurity professionals must protect the confidentiality, integrity, and availability of information, particularly personally identifiable information (PII). Improper disclosure of PII can lead to privacy breaches, which pose significant risks:

1. **Impact on Individuals**: Privacy breaches can expose individuals to identity theft and personal risks.
2. **Organizational Consequences**: Businesses may suffer reputational damage, fines, and the loss of intellectual property (IP) to competitors.
3. **Legal Implications**: Understanding local, regional, national, and global privacy laws is essential when evaluating privacy risks.
4. **Privacy Practices**: Organizations may adopt privacy notices to outline commitments, with some laws requiring these notices. Privacy statements may also be included in terms of agreement with customers and stakeholders.
5. **Data subject**
6. **Controller vs. processor**
7. **Ownership**
8. **Data inventory and retention**: To manage sensitive and personal data effectively, organizations must create a data inventory that includes:
   - **Personally Identifiable Information (PII)**: Any data that uniquely identifies an individual.
   - **Protected Health Information (PHI)**: Medical records subject to HIPAA regulations.
   - **Financial Information**: Personal financial records, which may fall under GLBA or PCI DSS.

- **Intellectual Property**: Proprietary business information, such as trade secrets and strategies.
- **Legal Information**: Documents related to legal proceedings, contracts, or corporate governance.
- **Regulated Information**: Data governed by laws and regulations (e.g., HIPAA, GLBA, PCI DSS).

The inventory should include all forms of data, whether human-readable or not, ensuring protection against loss or theft. This includes binary files containing sensitive information, such as PII, even if they require specialized software to read.

9. **Right to be forgotten** allows individuals to request the deletion of their personal data under certain conditions, particularly under the EU's GDPR. Individuals can request erasure if:

   - The data is no longer necessary for its original purpose.
   - Consent is withdrawn.
   - There is no legitimate reason to continue processing after objection.
   - The data was processed unlawfully.
   - There is a legal requirement to erase the data.

   This right gives individuals control over their personal information, addressing concerns about outdated or incorrect data. However, implementing this right can be technically and procedurally challenging for organizations, which must have systems in place to identify and erase data when requested.

5.5 Explain types and purposes of audits and assessments.

• **Attestation:** One of the primary outcomes of an audit is an *attestation* by the auditor. This is a formal statement that the auditors have reviewed the controls and found that they are both adequate to meet the control objectives and working properly.

• **Internal**: Internal audits are conducted by an organization's audit staff, typically for internal purposes. The audit team operates independently from the functions they evaluate, with the chief audit executive (CAE) often reporting directly to senior leadership (e.g., CEO) or the governing board's audit committee. These audits are crucial for ensuring **compliance** with internal and external obligations. They also include **self-assessments** to identify potential control gaps before formal external audits. This process helps management, and the board verify the organization’s compliance and readiness for more extensive external evaluations.

- **External**: are conducted by independent third-party audit firms to assess an organization's compliance and operations. These audits carry significant credibility due to the independence of the auditors, with firms like Ernst & Young, Deloitte, PwC, and KPMG being highly regarded.

  - **Regulatory Audits**: Conducted by regulatory bodies that may have the authority to audit firms based on legal or contractual requirements.
  - **Examinations**: These may be part of the regulatory audits or separate checks to ensure compliance with laws.
  - **Assessment**: External audits assess a company's performance, financial health, and adherence to regulations.
  - **Independent Third-Party Audits**: These audits are initiated by external entities like regulators or customers, and often aim to ensure that organizations meet specific standards. The SSAE 18 standard, particularly in SOC audits, helps streamline multiple third-party audits for service organizations, reducing the audit burden on them.

- **Penetration testing**: is an authorized, in-depth attempt to identify and exploit vulnerabilities in an organization's security controls, simulating real-world attacks. The process requires skilled testers who adopt a "hacker mindset," focusing on identifying a single vulnerability that could compromise security, rather than defending against all potential threats. Key Points:

  - **Hacker Mindset**: Penetration testers think like attackers, aiming to exploit overlooked or weak security controls. They focus on discovering flaws that can be leveraged to bypass defenses.
  - **Purpose**: Penetration tests complement existing security measures by providing visibility into vulnerabilities that may not be detected through traditional security tools. They provide insights into the effectiveness of defenses from an attacker's perspective.
  - **Benefits**:
    - Offers knowledge about the organization's security posture and its ability to withstand attacks.
    - Provides actionable remediation plans if vulnerabilities are exploited.
    - Helps in assessing specific systems, especially new ones, to ensure robust security before deployment.

Penetration testing enhances overall security by highlighting potential weaknesses and offering clear strategies to strengthen defenses against real-world attacks.
Type of PenTest:

- **Physical Penetration Testing**: Focuses on vulnerabilities in physical security, such as breaking into buildings, bypassing access controls, or compromising surveillance systems to assess physical security effectiveness.

- **Offensive Penetration Testing**: Involves security professionals acting as attackers to identify and exploit vulnerabilities in systems, networks, and applications, simulating real-world cyberattacks.

- **Defensive Penetration Testing**: Assesses an organization's ability to defend against cyberattacks by evaluating the effectiveness of security policies, procedures, and technologies in detecting and mitigating threats.

- **Integrated Penetration Testing**: Combines offensive and defensive testing approaches, involving close collaboration between experts to evaluate the security posture from both attack and defense perspectives.

- **Known Environment**: Testers have full knowledge of the system, including configurations, network diagrams, and credentials. This allows for a more thorough and complete test but may not accurately reflect the perspective of an external attacker.

- **Partially Known Environment**: A mix of both known and unknown environments, where testers are provided some information but not full access. This provides a balance between realism and efficiency in testing.

- **Unknown Environment**: Testers have no prior knowledge and must gather information and discover vulnerabilities like an external attacker. This type offers a more realistic test of security but is time-consuming and dependent on the skill of the testers.

- **Passive Reconnaissance**: Involves gathering information without directly interacting with the target system, typically through publicly available data.

- **Active Reconnaissance**: Involves directly engaging with the target system to gather data, such as scanning for vulnerabilities or attempting to probe systems for weaknesses.

5.6 Given a scenario, implement security awareness practices.

• Phishing: Phishing attacks often target users at all levels of the organization, and every security awareness program should include specific **antiphishing campaigns** designed to help users **recognize suspicious messages and respond to phishing attempts** appropriately. These campaigns often involve the use of *phishing simulations*, which send users fake phishing messages to test their skills. Users who click on the simulated phishing message are sent to a training program designed to help them better recognize fraudulent messages

• Anomalous behavior recognition: is also an important component of security awareness training. Employees should be able to recognize when **risky**, **unexpected**, and/or **unintentional** behavior takes place. The insider threat posed by employees with legitimate access permissions is significant, and other employees may be the first to notice the signs of anomalous behavior that could be a security concern.

• User guidance and training

- **Security Policies and Handbooks** Provide users with information about where they can find critical security documents.
- **Situational Awareness** Update users on the security threats facing the organization and how they can recognize suspicious activity.
- **Insider Threats** Remind users that employees, contractors, and other insiders may pose a security risk and that they should be alert for anomalous behavior.
- **Password Management** Educate users about your organization's password standards and the importance of not reusing passwords across multiple sites.
- **Removable Media and Cables** Inform users of the risks associated with the use of USB drives, external hard drives, and other removable media, as well as unfamiliar cables. Educate them on the policies for using these devices and the importance of scanning for malware before accessing files.
- **Social Engineering** Train users to recognize and respond to social engineering attacks. Teach them to be skeptical of unsolicited communications, especially those that create a sense of urgency or require sensitive information.
- **Operational Security** Educate users on the importance of protecting sensitive information during day-to-day operations. This includes understanding the importance of access controls, not discussing sensitive information in public or unsecured areas, and being vigilant about who has access to sensitive data.
- **Hybrid/Remote Work Environments** Instruct users on best practices for securing data and maintaining privacy when working remotely or in hybrid environments. This includes the use of VPNs, secure Wi-Fi networks, ensuring physical security of devices, and understanding the specific policies and procedures that are in place for remote work.

• Reporting and monitoring:

- **Initial Reporting**: At the start, administrators should track participation in training programs and assess user knowledge through quizzes and feedback. These initial reports help gauge the effectiveness of the training and identify areas for improvement.

- **Recurring Reporting**: Ongoing monitoring involves regularly providing decision-makers with detailed reports for technical stakeholders and high-level trends for management. This ensures that both day-to-day and strategic decisions can be informed by accurate, timely data. Regular reports also help to analyze trends in knowledge and security incidents, offering insights into the long-term impact of training programs.

2. Regular Review and Updates:

- The training materials should be reviewed periodically to ensure they remain relevant. Changes in the security landscape or within the organization may necessitate updates to keep the training content fresh and aligned with current risks and business needs.

This ongoing monitoring and reporting process helps to ensure that security training programs are effective and continue to address emerging threats and organizational changes.

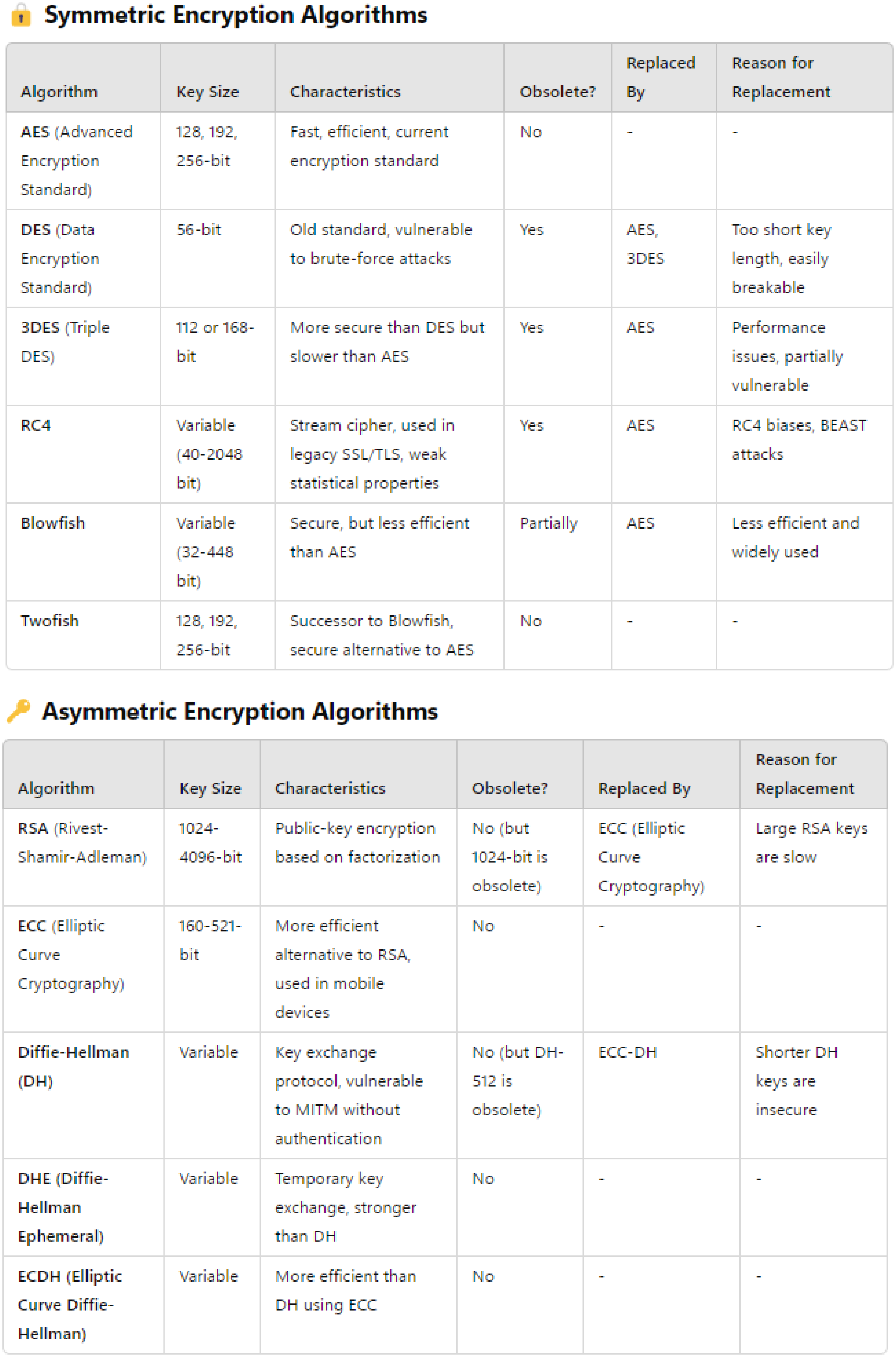
• Development

- Security training begins with a comprehensive assessment of the organization's security landscape, identifying potential risks and threats.
- Based on this analysis, training content is tailored to address the unique challenges of the organization.
- Real-world examples and interactive elements should be incorporated to engage participants, while aligning the training with organizational policies and procedures for consistency and relevance.

• Execution

- The execution phase includes diverse training methods like workshops, e-learning modules, and simulations to accommodate various learning preferences.
- Ensuring accessibility and regular training for all employees is crucial. This includes initial training for new recruits and periodic refreshers to keep knowledge up-to-date and relevant.

Other useful tips

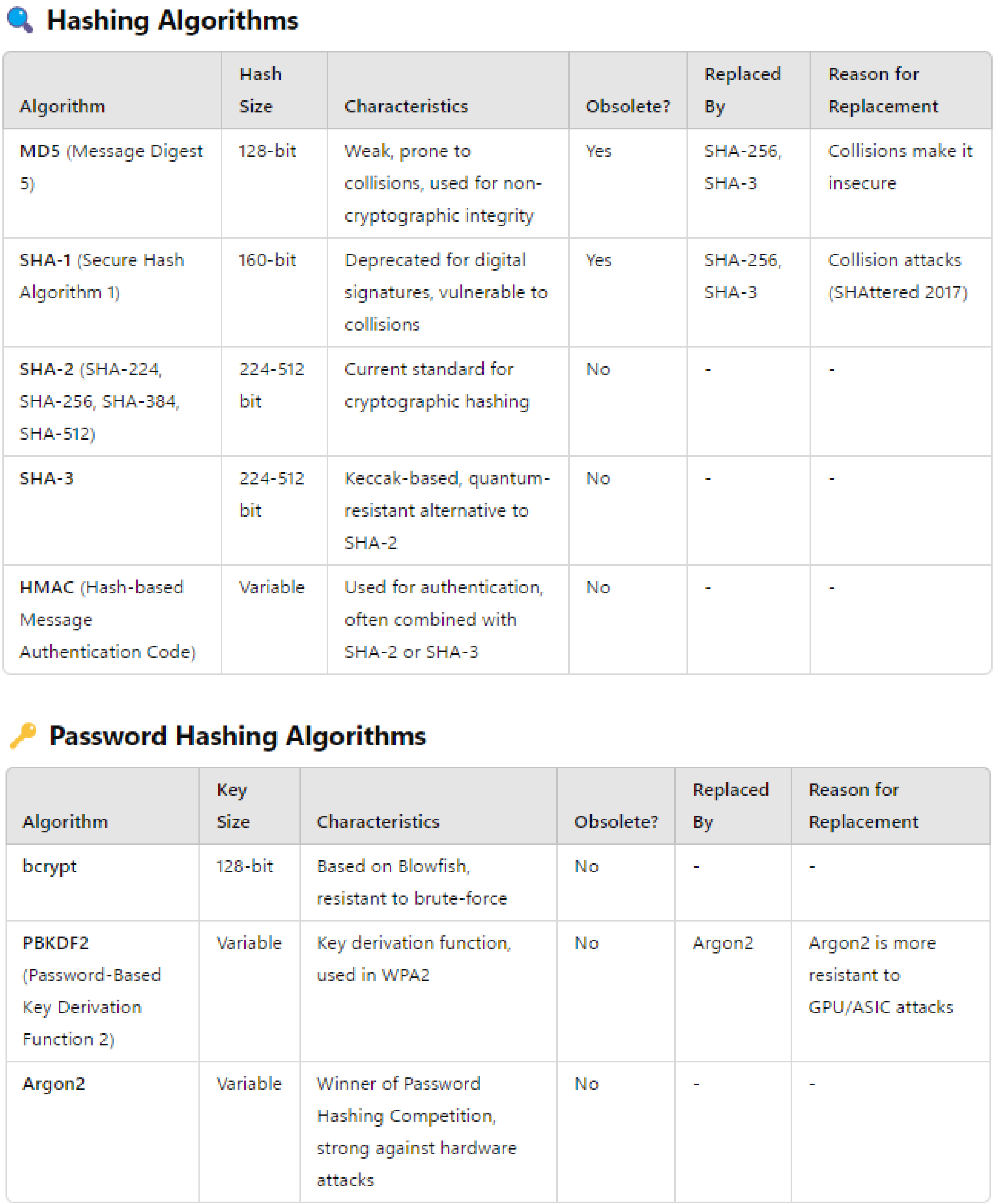


🔒 Symmetric Encryption Algorithms

Algorithm	Key Size	Characteristics	Obsolete?	Replaced By	Reason for Replacement
AES (Advanced Encryption Standard)	128, 192, 256-bit	Fast, efficient, current encryption standard	No	-	-
DES (Data Encryption Standard)	56-bit	Old standard, vulnerable to brute-force attacks	Yes	AES, 3DES	Too short key length, easily breakable
3DES (Triple DES)	112 or 168-bit	More secure than DES but slower than AES	Yes	AES	Performance issues, partially vulnerable
RC4	Variable (40-2048 bit)	Stream cipher, used in legacy SSL/TLS, weak statistical properties	Yes	AES	RC4 biases, BEAST attacks
Blowfish	Variable (32-448 bit)	Secure, but less efficient than AES	Partially	AES	Less efficient and widely used
Twofish	128, 192, 256-bit	Successor to Blowfish, secure alternative to AES	No	-	-

🔑 Asymmetric Encryption Algorithms

Algorithm	Key Size	Characteristics	Obsolete?	Replaced By	Reason for Replacement
RSA (Rivest-Shamir-Adleman)	1024-4096-bit	Public-key encryption based on factorization	No (but 1024-bit is obsolete)	ECC (Elliptic Curve Cryptography)	Large RSA keys are slow
ECC (Elliptic Curve Cryptography)	160-521-bit	More efficient alternative to RSA, used in mobile devices	No	-	-
Diffie-Hellman (DH)	Variable	Key exchange protocol, vulnerable to MITM without authentication	No (but DH-512 is obsolete)	ECC-DH	Shorter DH keys are insecure
DHE (Diffie-Hellman Ephemeral)	Variable	Temporary key exchange, stronger than DH	No	-	-
ECDH (Elliptic Curve Diffie-Hellman)	Variable	More efficient than DH using ECC	No	-	-

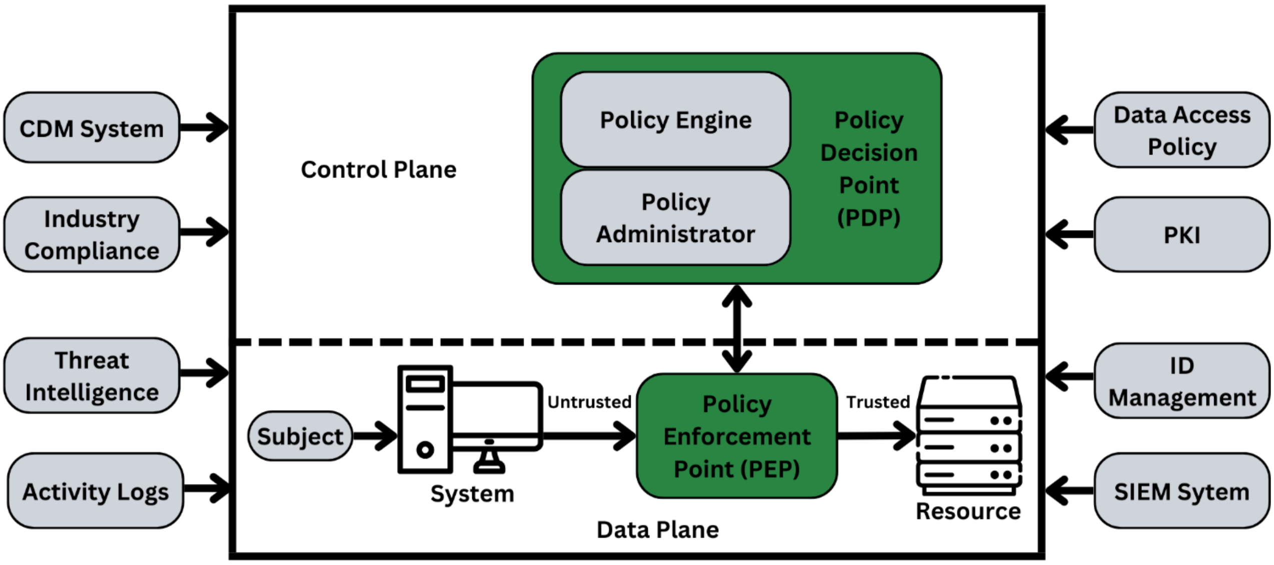


🔍 Hashing Algorithms

Algorithm	Hash Size	Characteristics	Obsolete?	Replaced By	Reason for Replacement
MD5 (Message Digest 5)	128-bit	Weak, prone to collisions, used for non-cryptographic integrity	Yes	SHA-256, SHA-3	Collisions make it insecure
SHA-1 (Secure Hash Algorithm 1)	160-bit	Deprecated for digital signatures, vulnerable to collisions	Yes	SHA-256, SHA-3	Collision attacks (SHAttered 2017)
SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512)	224-512 bit	Current standard for cryptographic hashing	No	-	-
SHA-3	224-512 bit	Keccak-based, quantum-resistant alternative to SHA-2	No	-	-
HMAC (Hash-based Message Authentication Code)	Variable	Used for authentication, often combined with SHA-2 or SHA-3	No	-	-

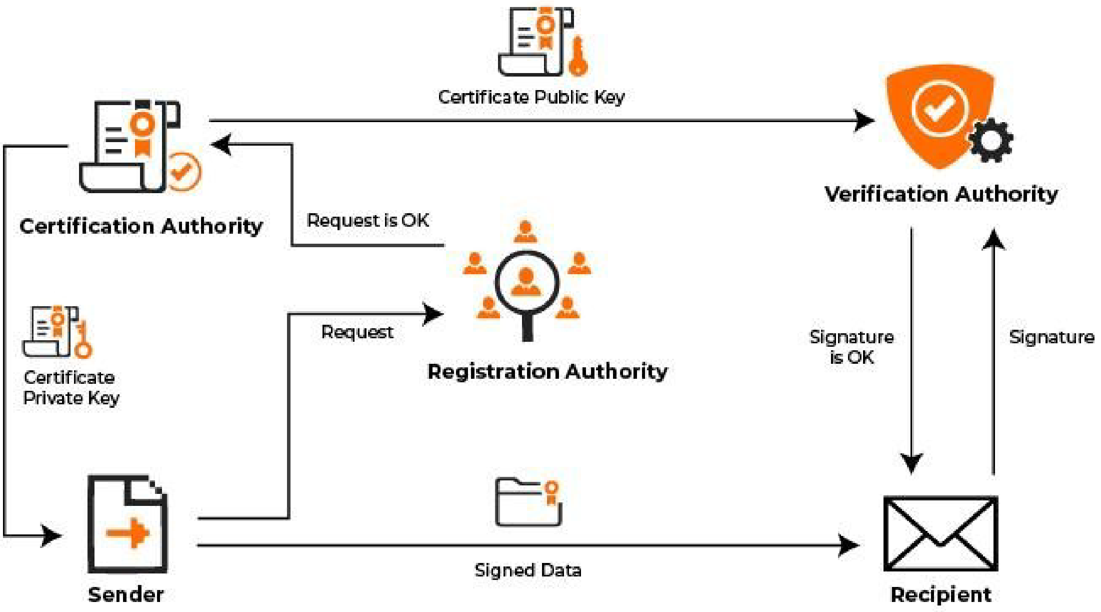
🔑 Password Hashing Algorithms

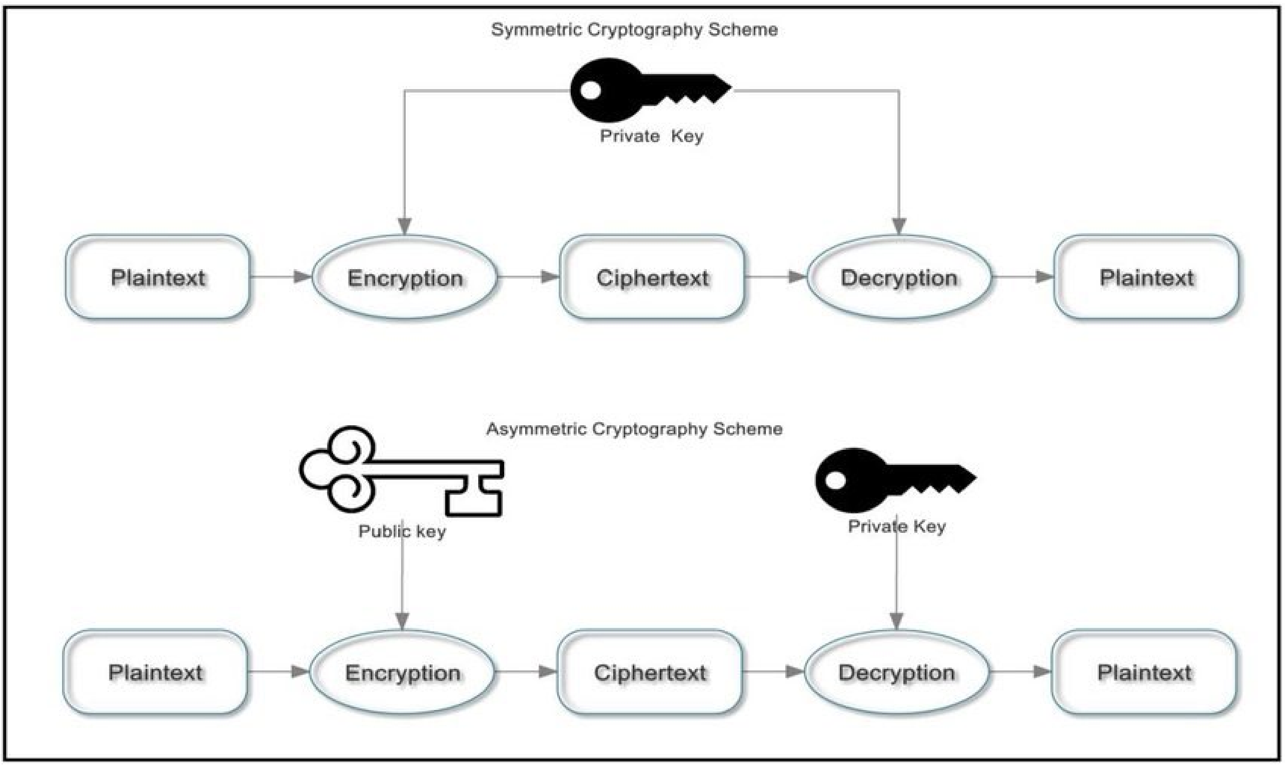
Algorithm	Key Size	Characteristics	Obsolete?	Replaced By	Reason for Replacement
bcrypt	128-bit	Based on Blowfish, resistant to brute-force	No	-	-
PBKDF2 (Password-Based Key Derivation Function 2)	Variable	Key derivation function, used in WPA2	No	Argon2	Argon2 is more resistant to GPU/ASIC attacks
Argon2	Variable	Winner of Password Hashing Competition, strong against hardware attacks	No	-	-



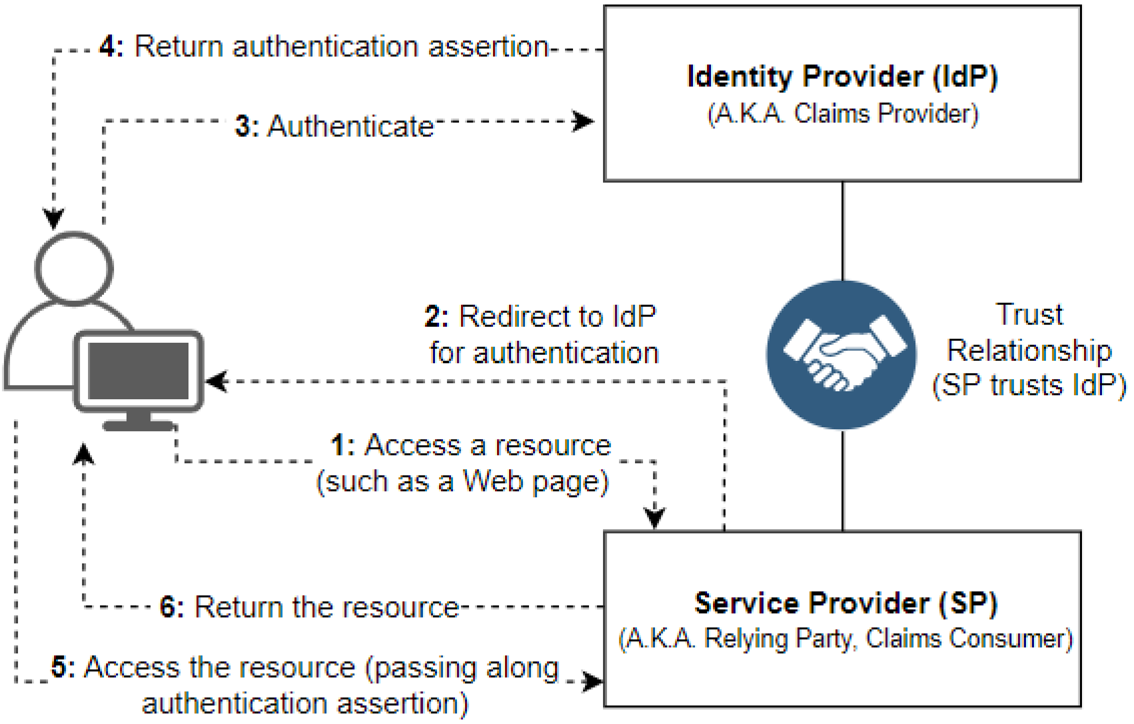
Zero Trust Core Logical Components

Public Key Infrastructure

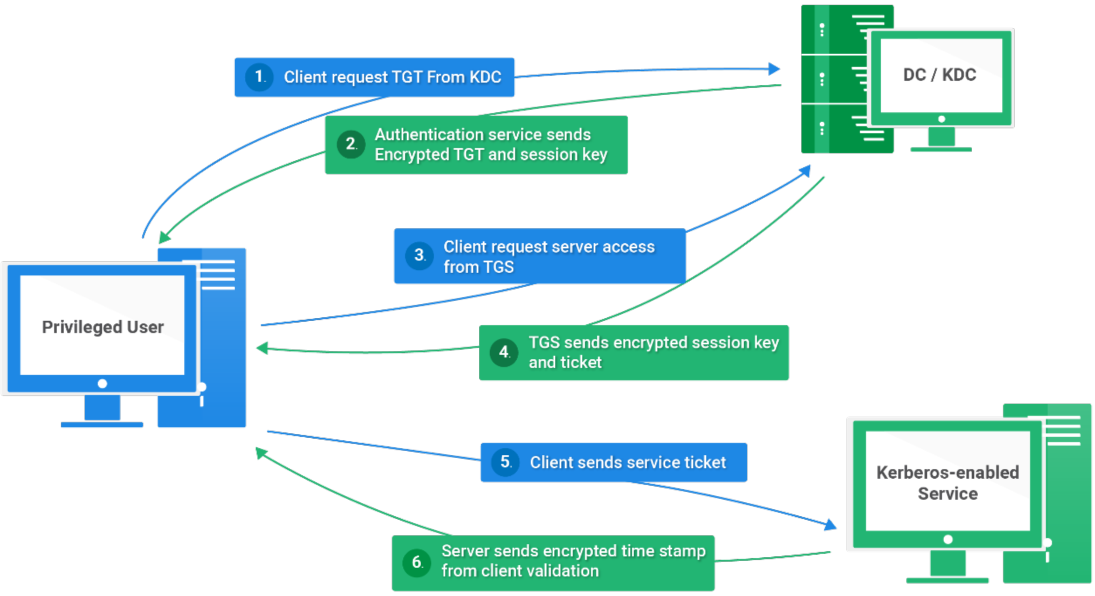




FEDERATION SCHEME


4: Return authentication assertion
3: Authenticate
Identity Provider (IdP)
(A.K.A. Claims Provider)
2: Redirect to IdP for authentication
Trust Relationship (SP trusts IdP)
1: Access a resource (such as a Web page)
6: Return the resource
5: Access the resource (passing along authentication assertion)
Service Provider (SP)
(A.K.A. Relying Party, Claims Consumer)

KERBEROS AUTHENTICATION


DC / KDC
1. Client request TGT From KDC
2. Authentication service sends Encrypted TGT and session key
3. Client request server access from TGS
Privileged User
4. TGS sends encrypted session key and ticket
5. Client sends service ticket
Kerberos-enabled Service
6. Server sends encrypted time stamp from client validation