Domain 3 **Security Architecture and Engineering**

You may find this domain to be more technical than others, and if you have experience woring in a security engineering role you likely have an advantage. If not, allocate extra time to this domain to ensure you have a good understanding of the topics

3.1 Research, implement, and manage engineering processes using secure design principles

- **Threat modeling**: a security process where potential threats are identified, categorized, and analyzed. It can be performed as a proactive measure during design and development or as an reactive measure once a product has been deployed
    - Threat modeling identifies the potential harm, the probability of occurrence, the priority of concern, and the means to eradicate or reduce the threat
- **Least privilege**: states that subjects are granted only the privileges necessary to perform assigned work tasks and no more; this concept extends to data and systems
    - Limiting and controlling privileges based on this concept protects confidentiality and data integrity
- **Defense in Depth**: AKA layering, is the use of multiple controls in a series, where a single failed control should not result in exposure of systems or data. Layers should be used in a series (one after the other), NOT in parallel. When you see the terms levels, multilevel, layers, classifications, zones, realms, compartments, protection rings etc think about Defense in Depth
- **Secure defaults**: when you think about defaults, consider how something operates brand new, just turned over to you by the vendor
    - e.g. wireless router default admin password, or firewall configuration requiring changes to meet an organization's needs
- **Fail securely**: if a system, asset, or process fails, it shouldn't reveal sensitive information, or be less secure than during normal operation. Failing securely could involve reverting to defaults
- **Separation of duties (SoD)**: separation of duties (SoD) and responsibilities ensures that no single person has total control over a critical function or system; SoD is a process to minimize opportunities for misuse of data or environment damage.
    - e.g. one person sells tickets, another collects tickets and restricts access to ticket holders in a movie theater
- **Keep it simple**: AKA keep it simple, stupid (KISS), this concept is the encouragement to avoid overcomplicating the environment, organization, or product design
- **Zero Trust**: "assume breach"; a security concept and alternative the traditional (castle/moat) approach where nothing is automatically trusted. Instead each request for activity or access is assumed to be from an unknown and untrusted location until otherwise verified;

    - Goal is to have every access request authenticated, authorized, and encrypted prior to access being granted to an asset or resource
    - See my article on an Overview of Zero Trust Basics
- **Privacy by design (PbD)**: a guideline to integrate privacy protections into products during the earliest design phase rather than tacking it on at the end of development;

    - Same overall concept as "security by design" or "integrated security" where security is an element of design and architecture of a product starting at initiation and continuing through the software development lifecycle (SDLC)
    - There are 7 recognized principles to achieve privacy by design:
        - Proactive, preventative: think ahead and design for things that you anticipate might happen
        - Default setting: make private by default, e.g. social media app shouldn't share user data with everybody by default
        - Embedded: build privacy in; don’t add it later
        - Full functionality, positive-sum: achieve both security and privacy, not just one or the other
        - Full lifecycle protection: privacy should be achieved before, during and after a transaction. Part of this is securely disposing of data when it is no longer needed
        - Visibility, transparency, open: publish the requirements and goals; audit them and publish the findings
        - Respect, user-centric: involve end users, providing the right amount of information for them to make informed decisions about their data
- **Trust but verify**: based on a Russian proverb, and no longer sufficient; it's the traditional approach of trusting subjects and devices within a company's security perimeter automatically, leaving an org vulnerable to insider attacks and providing intruders the ability to easily perform lateral movement

- **Shared responsibility**: the security design principle that indicates that organizations do not operate in isolation

    - Everyone in an organization has some level of security responsibility
    - the job of the CISO and security team is to establish & maintain security
    - The job of regular employees to perform their tasks within the confines of security
    - The job of the auditor is to monitor the environment for violations
    - When working with third parties, especially with cloud providers, each entity needs to understand their portion of the shared responsibility of performing work operations and maintaining security. This is often referenced as the **cloud shared responsibility model**

3.2 Understand the fundamental concepts of security modles (e.g. Biba, Star Model, Bell-LaPadula)

Security models:

- Intended to provide an explicit set of rules that a computer can follow to implement the fundamental security concepts, processes, and procedures of a security policy

- Provide a way for a designer to map abstract statements into a security policy prescribing the algorithms and data structures necessary to build hardware and software
- Enable people to access only the data classified for their clearance level
- **Bell-LaPadula**: Model was established in 1973. The goal is to ensure that information is exposed only to those with the right level of classification
    - Focuse is on confidentiality
    - Simple property: No read-up
    - Star (*) property: No write-down (AKA confinement property)
    - Discretionary Security Property: uses an access matrix (need to know in order to access)
    - Doesn't address covert channels
- **Biba**: Released in 1977, this model was created to supplement Bell-LaPadula
    - Focus is on integrity
    - “No read down” (for example, users with a Top Secret clearance can’t read data classified as Secret)
    - “No write up” (for example, a user with a Secret clearance can’t write data to files classified as Top Secret)
    - By combining it with Bell-LaPadula, you get both confidentiality and integrity
- **Take-Grant**:
    - The take-grant model employs a directed graph to dictate how rights can be passed from one subject to another, or from a subject to an object
    - Four rules:
        - take
        - grant
        - create
        - remove
- **Clark-Wilson**:
    - Designed to protect integrity using the access control triplet
    - A program interface is used to limit what is done by a subject; if the focus of an intermediary program between subject and object is to protect integrity, then it is an implementation of the Clark-Wilson model
- **Brewer and Nash Model**:
    - AKA "ethical wall", and "cone of silence"
    - created to permit access controls to change dynamically based on a user's previous activity
- **Goguen-Meseguer Model**:
    - An integrity model
    - Foundation of noninterference conceptual theories
- **Sutherland Model**:
    - Focuses on preventing interference in support of integrity
- **Graham-Denning Model**
    - Focused on the secure creation and deletion of both subjects and objects
    - 8 primary protection rules or actions
        - 1-4:securely create/delete a subject/object
        - 5-8:securely provide the read/grant/delete/transfer access right
- **Harrison-Ruzzo-Ullman Model**:

    - Focuses on the assignment of object access rights to subjects as well as the resilience of those assigned rights
    - HRU is an extension of Graham-Denning model
- **Star Model**:
    - Not an official model, but name refers to using asterisks (stars) to dictate whether a person at a specific level of confidentiality is allowed to write data to a lower level of confidentiality
    - Also determines whether a person can read or write to a higher or lower level of confidentiality

3.3 Select controls based upon systems security requirements

Be familiar with the **Common Criteria (CC)** for Information Technology Security Evaluation

- The CC provides a standard to evaluate systems, defining various levels of testing and confirmation of systems' security capabilities
- The number of the level indicates what kind of testing and confirmation has been performed
- The important concepts:
    - To perform an evaluation, you need to select the **Target of Evaluation (TOE)** (e.g. firewall or an anti-malware app)
    - The evaluation process will look at the **protection profile (PP)**, which is a document that outlines the security needs (customer "I wants"). A vendor might use a specific protection profile for a particular solution
    - The evaluation process will look at the **Security Target (ST)**, specifying the claims of security from the vendor that are built into a TOE (the ST is usually published to customers and partners and available to internal staff)
    - An organization's PP is compared to various STs from the selected vendor's TOEs, and the closest or best match is what the org purchases
    - The evaluation will attempt to gauge the confidence level of a security feature
    - **Security assurance requirements (SARs)** are documented and based on the development of the solution
    - Key actions during development and testing should be captured
    - An **evaluation assurance level (EAL)** is a numerical rating used to assess the rigor of an evaluation. The scale is EAL 1 (cheap and easy) to EAL7 (expensive and complex)
        - EAL1: functionally tested
        - EAL2: structurally tested
        - EAL3: methodically tested and checked
        - EAL4: methodically designed, tested, and reviewed
        - EAL5: semi-formally designed and tested
        - EAL6: semi-formally verified, designed, and tested
        - EAL7: formally verified, designed, and tested
- **Authorization to Operate (ATO)**: official auth to use specific IT systems to perform tasks/accept identified risks

3.4 Understand security capabilities of Information Systems (IS) (e.g. memory protection, Trusted Platform Model (TPM), encryption/decryption)

Security capabilities of information systems include memory protection, virtualization, Trusted Platform Module (TPM), encryption/decryption, interfaces, and fault tolerance

A computing device is likely running multiple applications and services simultaneously, each occupying a segment of memory. The goal of memory protection is to prevent one application or service from impacting another. There are two primary memory protection methods:

- Process isolation: OS provides separate memory spaces for each processes instructions and data, and prevents one process from impacting another
- Hardware segmentation: forces separation via physical hardward controls rather than logical processes; in this type of segmentation, the operating system maps processes to dedicated memory locations

Virtualization: technology used to host one or more operating systems within the memory of a single host, or to run applications that are not compatible with the host OS. The goal is to protect the hypervisor and ensure that compromising one VM doesn't affect others on that host

Trusted Platform Module (TPM): a cryptographic chip that is sometimes included with a client computer or server. A TPM enhances the capabilities of a computer by offering hardware-based cryptographic operations. Many security products and encryption solutions require a TPM

- TPM is both a specification for a cryptoprocessor chip on a motherboard and the general name for implementation of the specification
- A TPM is an example of a **hardware security module (HSM)**
- An HSM is a cryptoprocessor used to manage and store digital encryption keys, accelerate crypto operations, support faster digital signatures, and improve authentication

User interface: a constrained UI can be used in an application to restrict what users can do or see based on their privileges

- e.g. dimming/graying out capabilities for users without the correct privilege

An interface is also the method by which two or more systems communicate Be aware of the common security capabilities of interfaces:

- Encryption/decryption: when communications are encrypted, a client and server can communicate without exposing information to the network; when an interface doesn’t provide such a capability, use IPsec or another encrypted transport mechanism
- Signing: used for non-repudiation; in a high-security environment, both encrypt and sign all communications if possible

Fault tolerance: capability used to enhance availability. In the event of an attack (e.g. DoS), or system failure, fault tolerance helps keep a system up and running

3.5 Assess and mitigate the vulnerabilities of security architectures, designs and solution elements

This objective relates to identifying vulnerabilities and corresponding mitigating contols and solutions. The key is understanding the types of vulnerabilities commonly present in different environments, and their mitigation options

- **Client-based systems**: client computers are the most attacked entry point
    - Compromised client computers can be used to launch other attacks
    - Productivity software and browsers are constant targets
    - Even patched client computers are at risk due to phishing and social engineering vectors
    - Mitigation: run a full suite of security software, including anti-virus/malware, anti-spyware, and host-based firewall
- Server-based systems:
    - Data Flow Control: movement of data between processes, between devices, across a network, or over a communications channel
    - Management of data flow seeks to minimize latency/delays, keep traffic confidential (i.e. using encryption), not overload traffic (i.e. load balancer), and can be provided by network devices/applications & services
    - While attackers may initially target client computers, servers are often the goal
    - Mitigation: regular patching, deploying hardened server OS images for builds, and use host-based firewalls
- **Database systems**: databases often store a company's most sensitive data (e.g. proprietary, CC info, PHI, and PII)
    - Attackers may use inference or aggregation to obtain confidential information
    - **Aggregation attack**: process whereby SQL provides a number of functions that combine records from one or more tables to produce potentially useful info
    - **Inference attack** involves combining several pieces of nonsensitive info to gain access to that which should be classified at a higher level; inference makes use of the human mind’s deductive capacity rather than the raw mathematical ability of database platforms
- **Cryptographic systems**: the goal of a well-implemented cryptographic system is to make compromise too time-consuming and/or expensive. Each component has vulnerabilities:
    - **Kerckhoff's Principle** (AKA Kerckhoff's assumption): a cryptographic system should be secure even if everything about the system, except the key, is public knowledge
    - Software: used to encrypt/decrypt data; can be a standalone app, command-line, built into the OS or called via API. Like any software, there are likely bugs/issues, so regular patching is important
    - Keys: dictate how encryption is applied through an algorithm. A key should remain secret, otherwise the security of the encrypted data is at risk
        - **Key space**:represents all possible permutations of a key
        - Key space best practices:
            - key length is an important consideration; use as long of a key as possible (your goal is to outpace projected increase in cryptanalytic

capability during the time the data must be kept safe); longer keys discourage brute-force attacks
        - a 256-bit key is typically minimum recommendation for symmetric encryption
        - 2048-bit key typically the minimum for asymmetric
    - always store secret keys securely, and if you must transmit them over a network, do so in a manner that protects them from unauthorized disclosure
    - select the key using an approach that has as much randomness as possible, taking advantage of the entire key space
    - destroy keys securely, when no longer needed Always base key length on your requirements and sensitivity of the data being handled
  - Algorithms: choose algorithms (or ciphers) with a large key space and a large random **key value** (key value is used by an algorithm for the encryption process)
    - Algorithms themselves are not secret, but instead well-known with extensive public details about history and how they function
- **Industrial control systems (ICS)**: ICS is a form of computer-management device that controls industrial processes and machines, also known as operational technology (OT)

  - **Supervisory control and data acquisition (SCADA)**: systems used to control physical devices such as those found in an electrical power plant or factory. SCADA systems are well suited for distributed environments, such as those spanning continents
  - Some SCADA systems still rely on legacy or proprietary communications, which put them at risk, especially as attackers gain knowledge of such systems and their vulnerabilities
  - SCADA risk mitigations:
    - isolate networks
    - limit access physically and logically
    - restrict code to only essential apps
    - log all activity
- **Cloud-based systems**: on-demand access to computing resources available from almost anywhere

  - Cloud's primary challenge: resources are outside the org’s direct control, making it more difficult to manage risk

  - Orgs should formally define requirements to store and process data stored in the cloud

  - Focus your efforts on areas that you can control, such as the network entry and exit points (i.e. firewalls and similar security solutions)

  - All sensitive data should be encrypted, both for network communication and data-at-rest

  - Use centralized identity access and management system, with multifactor authentication

    - Customers shouldn’t use encryption controlled by the vendor, eliminating risks to vendor-based insider threats, and supporting destruction using
        - **cryptographic erase**: methods that permanently remove the cryptographic keys
    - Capture diagnostic and security data from cloud-based systems and store in your security information and event management (SIEM) system
    - Ensure that your cloud configuration matches or exceeds your on-premise security requirements
    - Understand the cloud vendor's security strategy
    - Cloud shared responsibility by model:
        - Software as a Service (SaaS):
            - the vendor is responsible for all maintenance of the SaaS services
        - Platform as a Service (PaaS):
            - customers deploy apps that they’ve created or acquired, manage their apps, and modify config settings on the host
            - the vendor is responsible for maintenance of the host and the underlying cloud infrastructure
        - Infrastructure as a Service (IaaS):
            - IaaS models provide basic computing resources to customers
            - customers install OSs and apps and perform required maintenance
            - the vendor maintains cloud-based infra, ensuring that customers have access to leased systems
- **Distributed systems distributed computing environment (DCE)**: a collection of individual systems that work together to support a resource or provide a service
    - DCEs are designed to support communication and coordination among their members in order to achieve a common function, goal, or operation
    - Most DCEs have duplicate or concurrent components, are asynchronous, and allow for fail-soft or independent failure of components
    - DCE is AKA concurrent computing, parallel computing, and distributed computing
    - DCE solutions are implemented as client-server, three-tier, multi-tier, and peer-to-peer
    - Securing distributed systems:
        - in distributed systems, integrity is sometimes a concern because data and software are spread across various systems, often in different locations
        - Client/server model network is AKA a distributed system or distributed architecture
            - security must be addressed everywhere instead of at a single centralized host
            - processing and storage are distributed on multiple clients and servers, and all must be secured
            - network links must be secured and protected

- **Internet of things (IoT)**: a class of smart devices that are internet-connected in order to provide automation, remote control, or AI processing to appliances or devices
    - An IoT device is almost always a separate/distinct hardware that is used on its own or in conjunction with an existing system
    - IoT security concerns often relate to access and encryption
    - IoT is often not designed with security as a core concept, resulting in security breaches; once an attacker has remote access to the device they may be able to pivot
    - Securing IoT:
        - Deploy a distinct network for IoT equipment, kept separate and isolated (known as **three dumb routers**)
        - Keep systems patched
        - Limit physical and logical access
        - Monitor activity
        - Implement firewalls and filtering
        - Never assume IoT defaults are good enough, evaluate settings and config options, and make changes to optimize security while supporting business function
        - Disable remote management and enable secure communication only (such as over HTTPS)
        - Review IoT vendor to understand their history with reported vulnerabilities, response time to vulnerabilities and their overall approach to security
        - Not all IoT devices are suitable for enterprise networks
- **Microservices**: a feature of web-based solutions and derivative of SOA
    - A microservice is simply one element, feature, capability, business logic, or function of a web application that can be called upon or used by other web applications
    - Microservices are usually small and focused on a single operation, designed with few dependencies, and are based on fast short-term development cycles (similar to Agile)
    - Securing microservices:
        - using HTTPS only
        - encrypt everything possible and use routine scanning
        - closely aligned with microservices is the concept of shifting left, or addressing security earlier in the SDLC; also integrating it into the CI/CD pipeline
        - consider the software supplychain or dependencies of libries used, when addressing updates and patching
- **Containerization**: AKA OS virtualization is based on the concept of eliminating the duplication of OS elements in a virtual machine; instead each application is placed into a container that includes only the actual resources needed to support the enclosed application, and the common or shared OS elements are then part of the hypervisor
    - Containerization is able to provide 10 to 100 x more application density per physical server compared to traditional virtualization

    - Vendors often have security benchmarks and hardening guidelines to follow to enhance container security
    - Securing containers:
        - container challenges include the lack of isolation compared to a traditional infrastructure of physical servers and VMs
        - scan container images to reveal software with vulnerabilities
        - secure your registries: use access controls to limit who can publish images, or even access the registry; require images to be signed
        - harden container deployment including the OS of the underlying host, using firewalls, and VPC rules, and use limited access accounts
        - reduce the attack surface by minimizing the number of components in each container, and update and scan them frequently
- **Serverless architecture** (AKA **function as a service (FaaS)**): a cloud computing concept where code is managed by the customer and the platform (i.e. supporting hardware and software) or servers are managed by the CSP

    - Applications developed on serverless architecture are similar to microservices, and each function is created to operate independently and automonomously
    - A serverless model, as in other CSP models, is a shared security model,and your organization and the CSP share security responsibility
- **Embedded systems**: any form of computing component added to an existing mechanical or electrical system for the purpose of providing automation, remote control, and/or monitoring; usually including a limited set of specific functions

    - Embedded systems can be a security risk because they are generally static, with admins having no way to update or address security vulnerabilities (or vendors are slow to patch)
    - Embedded systems focus on minimizing cost and extraneous features
    - Embedded systems are often in control of/associated with physical systems, and can have real-world impact
    - Securing embedded systems:
        - ebedded systems should be isolated from the internet, and from a private production network to minimize exposure to remote exploitation, remote control, and malware
        - use secure boot feature and physically protecting the hardware
- **High-performance computing (HPC)** systems: platforms designed to perform complex calculations/data manipulation at extremely high speeds (e.g. super computers or MPP); often used by large orgs, universities, or gov agencies

    - An HPC solution is composed of three main elements:
        - compute resources
        - network capabilities
        - storage capacity
    - HPCs often implement real-time OS (RTOS)
    - HPC systems are often rented, leased or shared, which can limit the effectiveness of firewalls and invalidate air gap solutions
    - Securing HPC systems:

        - deploy head nodes and route all outside traffic through them, isolating parts of a system
        - "fingerprint" HPC systems to undersatnd use, and detect anomalous behavior
- **Edge computing**: philosophy of network design where data and compute resources are located as close as possible, at or near the network edge, to optimize bandwidth use while minimizing latency

    - Securing edge computing:
        - this technology creates additional network edges that result in increased levels of complexity
        - visibility, control, and correlation requires a Zero Trust access-based approach to address security on the LAN edge, WAN edge and cloud edge, as well as network management
        - edge-based computing devices,especially IoT devices, are often produced with limited security forethought
        - devices on your network, no matter where they reside, need to be configured, managed, and patched using a consistent policy and enforcement strategy
        - use intelligence from side-channel signals that can pick up hardware trojans and malicious firmware
        - attend to physical security
        - deploy IDS on the network side to monitor for malicious traffic
        - in many scenarios, you are an edge customer, and likely will need to rely on a vendor for some of the security and vulnerability remediation
- **Virtualized systems**: used to host one or more OSs within the memory of a single host computer, or to run apps not compatible with the host OS

    - Securing virtualized systems:
        - the primary component in virtualization is a hypervisor which manages the VMs, virtual data storage, virtual network components
        - the hypervisor represents an additional attack surface
        - in virtualized environments, you need to protect both the VMs and the physical infrastructure/hypervisor
        - hypervisor admin accounts/credentials and service accounts are targets because they often provide access to VMs and their data; these accounts should be protected
        - virtual hosts should be hardened; to protect the host, avoid using it for anything other than hosting virtualized elements
        - virtualized systems should be security tested via vulnerability assessment and penetration testing
        - virtualization doesn't lessen the security management requirements of an OS, patch management is still required
        - be aware of VM Sprawl and Shadow IT
        - **VM escape**: occurs when software within a guest OS is able to breach the isolation protection provided by the hypervisor
        - VM escape minimizaton:
            - keep highly sensitive systems and data on separate physical machines

        - keep all hypervisor software current with vendor-released patches
        - monitor attack, exposure and abuse indexes for new threats to virtual machines (which might be better protected). Often, virtualization administrators have access to all virtual

3.6 Select and determine cryptographic solutions

- Cryptographic lifecycle (e.g., keys, algorithm selection)
    - Keep **Moore's Law** in mind (processing capabilities of state-of-the-art microprocessors double about every 2 years), and have appropriate governance controls in place to ensure that algorithms, protocols, and key lengths selected are sufficient to preserve the integrity of the cryptosystems for as long as necessary to keep secret information safe
    - Specify the cryptographic algorithms (such as AES, 3DES, and RSA) acceptable for use in an organization.
    - Identify the acceptable key lengths for use with each algorithm based on the sensitivity of the information transmitted
    - Enumerate the secure transaction protocols (such as TLS) that may be used
    - As computing power goes up, the strength of cryptographic algorithms goes down. Keep in mind the effective life of a certificate or certificate template, and of cryptographic systems
    - Beyond brute force, you have other issues to consider, such as the discovery of a bug or an issue with an algorithm or system
    - NIST defines the following terms that are commonly used to describe algorithms and key lengths:
        - approved (a specific algorithm is specified as a NIST recommendation or FIPS recommendation),
        - acceptable (algorithm + key length is safe today),
        - deprecated (algorithm and key length is OK to use, but brings some risk),
        - restricted (use of the algorithm and/or key length is deprecated and should be avoided),
        - legacy (the algorithm and/or key length is outdated and should be avoided when possible), and
        - disallowed (algorithm and/or key length is no longer allowed for the indicated use)
- Cryptographic methods (e.g., symmetric, asymmetric, elliptic curves, quantum)
    - **Symmetric** encryption: uses the same key for encryption and decryption
        - symmetric encryption uses a shared secret key available to all users of the cryptosystem
        - symmetric encryption is faster than asymmetric encryption because smaller keys can be used for the same level of protection
        - downside is that users or systems must find a way to securely share the key and hope the key is used only for the specified communication
        - primarily employed to perform bulk encryption and provides only for the security service of confidentiality -"same" is a synonym for symmetric
        - "different" is a synonym for asymmetric

        - total number of keys required to completely connect n parties using symmetric cryptography is given by this formula:
            - **(n(n - 1)) / 2**
    - **Asymmetric** encryption: uses different keys for encryption and decryption
        - Asymmetric (AKA public key, since one key of a pair is available to anybody) algorithms provide convenient key exchange mechanisms and are scalable to very large numbers of users (addressing the two most significant challenges for users of symmetric cryptosystems) - Asymmetric cryptosystems avoid the challenge of sharing the same secret key between users, by using pairs of public and private keys to allow secure communication without the overhead of complex key distribution
        - Besides the public key, there is a private key that should remain private and protected
        - While asymmetric encryption is slower, it is best suited for sharing between two or more parties
        - Most common asymmetric cryptosystems in use today:
            - Rivest-Shamir-Adleman (RSA)
            - Diffie-Hellman
            - ElGamal
            - Eliptical Curve Cryptography (EEC)
- **Public Key Infrastructure (PKI)**: hierarchy of trust relationships permitting the combination of asymmetric and symmetric cryptography along with hashing and digital certificates (giving us hybrid cryptography)

    - A PKI issues certificates to computing devices and users, enabling them to apply cryptography (for example, to send encrypted email messages, encrypt websites or use IPsec to encrypt data communications)
    - Many vendors provide PKI services; you can run a PKI privately and solely for your own org, you can acquire certificates from a trusted third-party provider, or you can do both (which is common)
    - A PKI is made up of
        - **certification authorities (CAs)**: servers that provide one or more PKI functions, such as providing policies or issuing certificates
        - certificates: issued to other certification authorities or to devices and users
        - policies and procedures: such as how the PKI is secured, and
        - templates: a predefined configuration for specific uses, such as a web server template
    - There are other components and concepts you should know for the exam:
        - A PKI can have multiple tiers:
            - single tier means you have one or more servers that perform all the functions of a PKI.
            - two tiers means you often have an offline root CA (a server that issues certificates to the issuing CAs but remains offline most of the time) in one tier, and issuing CAs (the servers that issue certificates to computing devices and users) in the other tier
            - servers in the second tier are often referred to as intermediate CAs or subordinate CAs.

            - three tier means you can have CAs that are responsible only for issuing policies (and they represent the second tier in a three-tier hierarchy)
                - in such a scenario, the policy CAs should also remain offline and be brought online only as needed
        - Generally, the more tiers, the more security (but proper configuration is critical)
            - the more tiers you have, the more complex and costly the PKI is to build and maintain
        - A PKI should have a certificate policy and a certificate practice statement (CSP)
            - certificate policy: documents how your org handles items like requestor identities, the uses of certificates and storage of private keys
            - CSP: documents the security configuration of your PKI and is usually available to the public
        - Besides issuing certificates, a PKI has other duties:
            - a PKI needs to be able to provide certificate revocation information to clients
            - if an administrator revokes a certificate that has been issued, clients must be able to get that information from your PKI
            - storage of private keys and information about issued certificates (can be stored in a database or a directory)
        - PKI uses LDAP when integrating digital certificates into transmissions
- **Key management practices** include safeguards surrounding the creation, distribution, storage, destruction, recovery, and escrow of secret keys

    - Cryptography can be used as a security mechanism to provide confidentiality, integrity, and availability only if keys are not compromised
    - Three main methods are used to exchange secret keys:
        - offline distribution
        - public key encryption, and
        - the Diffie-Hellman key exchange algorithm
    - Key management can be difficult with symmetric encryption but is much simpler with asymmetric encryption
    - There are several tasks related to key management:
        - Key creation
        - **Key distribution**: the process of sending a key to a user or system; it must be secure and it must be stored in a secure way on the computing device
        - Keys are stored before and after distribution; when distributed to a user, it can't hang out on a user's desktop
        - Keys shouldn't be in cleartext outside the crypography device
        - Key distribution and maintenance should be automated (and hidden from the user)
        - Keys should be backed up!
        - **Key escrow**: process or entity that can recover lost or corrupted cryptographic keys
            - **multiparty key recovery**: when two or more entities are required to reconstruct or recover a key

            - **m of n control**: you designate a group of (n) people as recovery agents, but only need subset (m) of them for key recovery
            - **split custody**: enables two or more people to share access to a key (e.g. for example, two people each hold half the password to the key)
        - Key rotation: rotate keys (retire old keys, implement new) to reduce the risks of a compromised key having access
        - Key states:
            - suspension: temporary hold
            - revocation: permanently revoked
            - expiration
            - destruction
        - See NIST 800-57, Part 1
- Digital signatures and digital certificates
    - **Digital signatures**: provide proof that a message originated from a particular user of a cryptosystem, and ensures that the message was not modified while in transit between two parties
        - Digital signatures rely on a combination of two major concepts — public key cryptography, and hashing functions
        - Digitally signed messages assure the recipient that the message truly came from the claimed sender, enforcing nonrepudiation
        - Digitally signed messages assure the recipient that the message was not altered while in transit; protecting against both malicious modification (third party altering message meaning), and unintentional modification (faults in the communication process)
        - Digital signature process does not provide confidentiality in and of itself (only ensures integrity, authentication, and nonrepudiation)
    - Non-repudiation
        - Here non-repudiation refers to methods ensuring certainty about data origins
        - Most common method of non-repudiation is digital signatures
        - Digital signatures rely on certificates
        - If a digital signature was verified with the public key of the sender, then we know that it was created using the sender's private key
        - Private key should only be known to the sender, so the verification proves to the recipient that the signature came from the sender, providing origin authentication
        - The recipient (or anyone else) can demonstrate that process to a third party providing nonrepudiation
        - Data encryption provides confidentiality
    - Integrity (e.g., hashing)
        - Hash Functions have a very simple purpose — they take a potentially long message and generate a unique output value derived from the content of the message called a **message digest**

        - hash function implements encryption with a specified algorithm, but without a key
        - used to ensure message sent by the originator is the same one received by recipient
        - input can be of any length
        - output has a fixed length
        - the hash function is relatively easy to compute for any input
        - the hash function is one-way, meaning it is extremely difficult to determine the input given the hash function output
        - the hash function should be collision-resistant, meaning it is extremely hard to find two messages that produce the same hash value output
        - hashes are used for storing passwords, with email, and for file download integrity verification
    - Hashing and integrity: if the hash generated by sender, and separately by the receiver match, then we have integrity

3.7 Understand methods of cryptanalytic attacks

- **Brute force**: an attack that attempts every possible valid combination for a key or password
    - They involve using massive amounts of processing power to methodically guess the key used to secure cryptographic communications
- **Ciphertext only**:an attack where you only have the encrypted ciphertext message at your disposal (not the plaintext)
    - If you have enough ciphertext samples, the idea is that you can decrypt the target ciphertext based on the ciphertext samples
    - One technique proves helpful against simple ciphers is frequency analysis (counting the number of times each letter appears in the ciphertext)
- **Known plaintext**:in this attack, the attacker has a copy of the encrypted message along with the plaintext message used to generate the ciphertext (the copy); this knowledge greatly assists the attacker in breaking weaker codes
- **Frequency analysis**:an attack where the characteristics of a language are used to defeat substitution ciphers
    - For example in English, the letter "E" is the most common, so the most common letter in an encrypted cyphertext could be a substitution for "E"
    - Other examples might include letters that appear twice in sequence, as well as the most common words used in a language
- **Chosen ciphertext**: in a chosen ciphertext attack, the attacker has access to one or more ciphertexts and their plaintexts; i.e. the attacker has the ability to decrypt chosen portions of the ciphertext message, and use the decrypted portion to discover the key
    - **Differential cryptanalysis**, a type of chosen plaintext attack, is a general form of cryptanalysis applicable primarily to block ciphers, but also to stream ciphers and cryptographic hash functions; in the broadest sense, it is the study of how

differences in information input can affect the resultant difference at the output advanced methods such as differential cryptanalysis are types of chosen plaintext attacks;
    - as an example, an attacker may try to get the receiver to decrypt modified ciphertext, looking for that modification to cause a predictable change to the plaintext
- **Implementation attack**: attempts to exploit weaknesses in the implementation of a cryptography system
    - Focuses on exploiting the software code, not just errors or flaws but the methodology employed to program the encryption system
    - In this type of attack, attackers look for weaknesses in the implementation, such as a software bug or outdated firmware
- **Side-channel**: these attacks seek to use the way computer systems generate characteristic footprints of activity, such as changes in processor utilization, power consumption, or electromagnetic radiation to monitor system activity and retrieve information that is actively being encrypted
    - Similar to an implementation attack, side-channel attacks look for weaknesses outside of the core cryptography functions themselves
    - A side-channel attack could target a computer’s CPU, or attempt to gain key information about the environment during encryption or decryption by looking for electromagnetic emissions or the amount of execution time required during decryption.
    - Side-channel characteristics information are often combined together to try to break down the cryptography
    - Timing attack is an example
- **Fault-Injection**: the attacker attempts to compromise the integrity of a cryptographic device by causing some type of external fault
    - For example, using high-voltage electricity, high or low temperature, or other factors to cause a malfunction that undermines the security of the device
- **Timing**: timing attacks are an example of a side-channel attack where the attacker measures precisely how long cryptographic operations take to complete, gaining information about the cryptographic process that may be used to undermine its security
- **Man-in-the-middle (MITM) (AKA on-path)**: in this attack a malicious individual sits between two communicating parties and intercepts all communications (including the setup of the cryptographic session)
    - Attacker responds to the originator's initialization requests and sets up a secure session with the originator
    - Attacker then establishes a second secure session with the intended recipient using a different key and posing as the originator
    - Attacker can then "sit in the middle" of the communication and read all traffic as it passes between the two parties
- **Pass the hash (PtH)**: a technique where an attacker captures a password hash (as opposed to the password characters) and then simply passes it through for

authentication and potentially lateral access to other networked systems

    - The threat actor doesn't need to decrypt the hash to obtain a plain text password
    - PtH attacks exploit the authentication protocol, as the passwords hash remains static for every session until the password is rotated
    - Attackers commonly obtain hashes by scraping a system's active memory and other techniques
- Kerberos exploitation:

    - **Overpass the Hash**: alternative to the PtH attack, used when NTLM is disabled on the network (AKA pass the key)
    - **Pass the Ticket**:in this attack, attackers attempt to harvest tickets held in the lsass.exe process
    - **Silver Ticket**: a silver ticket uses the captured NTLM hash of a service account to create a ticket-granting service (TGS) ticket (the silver ticket grants the attacker all the privileges granted to the service account)
    - **Golden Ticket**: if an attacker obtains the hash of the Kerberos service account (KRBTGT), they can create tickets at will within Active Directory (this provides so much power it is referred to as having a golden ticket)
    - **Kerberos Brute-Force**: attackers use the Python script kerbrute.py on Linux, and Rubeus on Windows systems; tools can guess usernames and passwords
    - **ASREPRoast**: ASREPRoast identifies users that don't have Kerberos preauthentication enabled
    - **Kerberoasting**: kerberoasting collects encrypted ticket-granting service (TGS) tickets
- **Ransomware**: a type of malware that weaponizes cryptography

    - using many of the same techniques as other types of malware, ransomware gens an encryption key, and encrypts critical files
    - this encryption renders the data inaccessible to the authorized user or anyone else other than the malware author - often threatning to publically release sensitive data if ransome is not paid - 2020 study, 56% of orgs suffered a ransomeware attack, 27% of orgs who reported an attack chose to pay, on average ~$1.1m
    - seek legal advice prior to engaging with ransomware authors

3.8 Apply security principles to site and facility design

- **Secure facility plan**: outlines the security needs of your organization and emphasizes methods or mechanisms to employ to provide security, developed through risk assessment and critical path analysis
    - **critical path analysis (CPA)**: a systematic effort to identify relationships between mission-critical applications, processes, and operations and all the necessary supporting components
    - During CPA, evaluate potential **technology convergence**: the tendency for various technologies, solutions, utilities, and systems to evolve and merge over time, which can result in a single point of failure
    - A secure facility plan is based on a layered defense model

    - Site selection should take into account cost, location, and size (but security should always take precedence), that the building can withstand local extreme weather events, vulnerable entry points, and exterior objects that could conceal break-ins
- Facility Design:
    - the top priority of security should always be the protection of the life and safety of personnel
    - in the US, follow the guidelines and requirements from Occupational Safety and Health Administration (OSHA), and Environmental Protection Agency (EPA)
    - **Crime Prevention Through Environmental Design (CPTED)**: a well-established school of thought on "secure architecture"
        - core principle of CPTED is that the design of the physical environment can be managed/manipulated, and crafted with intention in order to create behavioral effects or changes in people present in those areas that result in reduction of crime as well as a reduction of the fear of crime
        - CPTED stresses three main principles:
            - **natural access control**: the subtle guidance of those entering and leaving a building
                - make the entrance point obvious
                - create internal security zones
                - areas of the same access level should be open, but restricted/closed areas should seem more difficult to access
            - **natural surveillance**: any means to make criminals feel uneasy through increased opportunities to be observed
                - walkways/stairways are open, open areas around entrances
                - areas should be well lit
            - **natural territorial**: reinforcement: attempt to make the area feel like an inclusive, caring community
    - Overall goal is to deter unauthorized people from gaining access to a location (or a secure portion), prevent unauthorized personnel from hiding inside or around the location, and prevent unauthorized from committing crime
    - There are several smaller activities tied to site and facility design, such as upkeep and maintenance: if property is run down, unkempt or appears to be in disrepair, it gives attackers the impression that they can act with impunity on the property

3.9 Design site and facility security controls

- Note that alghough the topics in this section cover mostly interior spaces, physical security is applicable to both interior and exterior of a facility

- **Wiring closets/intermediate distribution facilities (IDF)**: A wiring closet or IDF is typically the smallest room that holds IT hardware

    - Wiring closet is AKA premises wire distribution room, main distribution frame (MDF), intermediate distribution frame (IDF), and telecommunications room, and it is referred to as an IDF in (ISC)^2 CISSP objective 3.9.1
    - Usually includes telephony and network devices, alarm systems, circuit breaker panels, punch-down blocks, WAPs, video/security
    - May include a small number of servers

    - Access to the wiring closest/IDF should be restricted to authorized personnel responsible for managing the IT hardware - Use door access control (i.e. electronic badge system or electronic combination lock)
    - From a layout perspective, wiring closets should be accessible only in private areas of the building interiors; people must pass through a visitor center and a controlled doorway prior to be able to enter a wiring closet
- **Server rooms/data centers**: server rooms, data centers, communication rooms, server vaults, and IT closets are enclosed, restricted, and protected rooms where mission critical servers and networks are housed

    - A server room is a bigger version of a wiring closet, much smaller than a data center
    - A server room typically houses network equipment, backup infrastructure and servers (more archaic versions include telephony equipment)
    - Server rooms should be designed to support optimal operation of IT infrastructure and to block unauthorirzed human access or intervention
    - Server rooms should be located at the core of the building (avoid ground floor, top floor, or in the basement)
    - Server rooms should have a single entrance (and an emergency exit)
    - Server room should block unauthorized access, and entries and exits should be logged
    - Datacenters are usually more protected than server rooms, and can include guards and mantraps
    - Datacenters can be single-tenant or multitenant
- **Media storage facilities**:often store backup tapes and other media, and should be protected just like a server room

    - Depending on requirements a cabinet or safe could suffice
    - New blank media, and media that is reused (e.g. thumb drives, flash memory cards, portable hard drives) should be protected against theft and data remnant recovery
    - Other recommendations:
        - employ a media librarian or custodian
        - use check-in/check-out process for media tracking
        - run a secure drive sanitization or zeroization when media is returned
    - Note: a safe is a movable secured container that's not integrated into a building's construction; a vault is a permanent safe integrated into construction
- **Evidence storage**: as cybercrime events continue to increase, it is import to retain logs, audit trails, and other records of digital events; the evidence storage exists to preserve chain of custody

    - A key part of incident response is to gather evidence to perform root cause analysis
    - An evidence storage room should be protected like a server room or media storage facility
    - An evidence storage room can contain physical evidence (such as a smartphone) or digital evidence (such as a database)

- **Restricted and work area security**: covers the design and configuration of internal security, including work and visitor areas
    - Includes areas that contain assets of higher value/importance which should have more restricted access
    - Restricted work areas are used for sensitive operations, such as network/security ops
    - Protection should be similar to a server room, but video surveillance is typically limited to entry and exit points
- Utilities and heating, ventilation, and air conditioning (HVAC)
    - Power management in ascending order: surge protectors, power/power-line conditioner, uninterruptible power supply (UPS), generators
    - Types of UPS:
        - double conversion: functions by taking power from the wall outlet, storing it in a battery, pulling power out of the battery and feeding that power to the device/devices
        - line-interactive: has a surge protector, battery charger/inverter and voltage regulator positioned between the grid power source and the equipment (battery is not in line under normal conditions)
    - Commercial power problem types:
        - **fault**: momentary loss of power
        - **blackout**: complete loss of power
        - **sag**: momentary low voltage
        - **brownout**: prolonged low voltage
        - **spike**: momentary high voltage
        - **surge**: prolonged high voltage
        - **inrush**: initial surge of power associated with connecting to a power source
    - Think through types of physical controls for HVAC:
        - restrict duct space continuity to controlled areas
        - use separate and redundant HVAC systems for computer equipment
    - Datacenter:
        - should be on different power circuits from occupied areas
        - common to use a backup generator
- Environmental issues
    - Environmental monitoring is the process of measuring and evaluating the quality of the environment within a given structure (e.g. temperature, humidity, dust, smoke), using things like chemical, biological, radiological, and microbiological detectors
    - Halon starves a fire of oxygen by disrupting the chemical reaction of combustion, but degrades into toxic gases at 900 degrees Fahrenheit, and is not environmentally friendly
    - If water-based sprinklers are used for fire suppression, damage to electronic equipment is likely; automate the shutoff of electricity prior to sprinkler trigger
    - Other environmental issues include earthquakes, power outages, tornados and wind

    - Secondary facilities should be located far enough away from the primary to ensure they won't be damaged by the same event
- Fire prevention, detection and suppression
    - Protecting personnel from harm should always be the most important goal of any security or protection system!
    - In addition to protecting people, fire detection and suppression is designed to keep asset damage caused by fire, smoke, heat, and suppression materials to a minimum
    - **Fire triangle**: three represent fuel, heat, and oxygen; the center of the triangle represents the chemical reaction among these three elements
        - if you can remove any one of the four items from the fire triangle, the fire can be extinguished
    - Fire suppression mediums:
        - water suppresses temperature
        - soda acid and other dry powders suppress the fuel supply
        - carbon dioxide (CO2) suppresses the oxygen supply
        - halon substitutes and other nonflammable gases interfere with the chemistry of combustion and/or suppress the oxygen supply
    - Fire stages:
        - **Stage 1**: incipient stage: at this stage, there is only air ionization and no smoke
        - **Stage 2**: smoke stage: smoke is visible from the point of ignition
        - **Stage 3**: flame stage: this is when a flame can be seen with the naked eye
        - **Stage 4**: heat stage: at stage 4, there is an intense heat buildup and everything in the area burns
    - Fire extinguisher classes:
        - **Class A**: common combustibles
        - **Class B**: liquids
        - **Class C**: electrical
        - **Class D**: metal
        - **Class K**: cooking material (oil/grease)
    - Four main types of suppression:
        - **wet pipe system**: (AKA closed head system): is always filled with water. water discharges immediately when suppression is triggered
        - **dry pipe system**: contains compressed inert gas
        - **preaction system**: a variation of the dry pipe system that uses a two-stage detection and release mechanism
        - **deluge system**: uses larger pipes and delivers larger volume of water
    - Note: Most sprinkler heads feature a glass bulb filled with a glycerin-based liquid; this liquid expands when it comes in contact with air heated to between 135 and 165 degrees; when the liquid expands, it shatters its glass confines and the sprinkler head activates
- Power (e.g., redundant, backup)
    - Consider designing power to provide for high availability
    - Most power systems have to be tested at regular intervals

- As part of the design, mandate redundant power systems to accommodate testing, upgrades and other maintenance
- Additionally, test failover to a redundant power system and ensure it is fully functional
- The International Electrical Testing Association (NETA) has developed standards around testing power systems
- Battery backup/fail-over power (including UPS/generators):
    - this is a system that collects power into a battery but can switch over to pulling power from the battery when the power grid fails
    - generally, this type of system was implemented to supply power to an entire building rather than just one or a few devices