CySA+ Study Guide

1.0 Threat and Vulnerability Management

1.1 Explain the importance of threat data and intelligence

Data source types:

- **OSINT (Open Source Intelligence)**: A process of gathering and analyzing information from publicly available sources to extract valuable insights and knowledge for various purposes, such as security, research, or decision-making. It involves using publicly accessible data from the internet, media, and other open sources to piece together relevant information
- **Proprietary, or Closed Source Intelligence:** Information and data that is privately owned, controlled, and restricted from public access, typically utilized by specific organizations or entities for their internal purposes and not openly available to the public

Indicator management:

- **STIX (Structured Threat Information eXpression)**: Standardized language for describing and sharing cyber threat intelligence information, facilitating effective communication and collaboration among cybersecurity professionals and tools
- **TAXII (Trusted Automated eXchange of Indicator Information)**: Protocol that allows for secure and automated sharing of cyber threat intelligence, enabling organizations to exchange relevant information and improve their collective defense against cyber threats
- **OpenIOC (Open Indicators of Compromise)**: An open standard XML-based format used to describe and share indicators of compromise, aiding in the detection and response to cybersecurity threats across different security tools and platforms

Threat classification:

- **Known vs. Unknown Threats:** Known threats refer to cybersecurity risks that have been previously identified and have known signatures or patterns, allowing security measures to be implemented to defend against them, while unknown threats represent new and emerging risks that lack recognizable characteristics, making them more challenging to detect and defend against, requiring advanced security solutions and proactive strategies to mitigate their potential impact
- **Zero-Day Threat**: Exploits a previously unknown software vulnerability, leaving users exposed as there is no fix or patch available from the software vendor at the time of its discovery
- **Advanced Persistent Threat (APT)**: A sophisticated and stealthy cyber attack conducted by skilled and well-funded threat actors, typically targeting specific organizations or individuals over an extended period, with the intent of gaining unauthorized access to sensitive information and maintaining persistence within the target network

Threat actors:

- **Nation-state Actors**: Government-backed entities that engage in cyber attacks to achieve strategic objectives, including espionage, sabotage, or disruption of critical infrastructure, posing significant and highly sophisticated cybersecurity threats
- **Hacktivists**: Individuals or groups who use hacking and digital activism to promote a particular social or political cause, often by targeting websites, organizations, or individuals seen as opposing their beliefs
- **Organized crime**: Profit-driven groups that target Personally Identifiable Information (PII), credit cards, etc.

Insider threat:

- **Intentional**: Disgruntled or profit-driven employee stealing/damaging/exposing internal systems
- **Unintentional**: Personal negligence and/or poor security practices

Intelligence cycle:

- **Planning and Direction**: Establishing intelligence objectives and strategies to guide the collection and analysis efforts in cybersecurity
- **Collection**: Gathering data and information from various sources, such as network logs, threat feeds, and open-source intelligence, to build a comprehensive dataset
- **Processing and Exploitation**: Organizing and converting raw data into a usable format, extracting relevant insights, and identifying potential cybersecurity threats
- **Analysis and Production**: Examining and evaluating the processed data to derive actionable intelligence, such as identifying patterns, trends, and potential attack vectors
- **Dissemination**: Sharing the analyzed intelligence with relevant stakeholders, including security teams and decision-makers, to enable informed responses and proactive measures
- **Feedback**: Assessing the effectiveness of the intelligence gathered and used, refining the intelligence requirements, and integrating lessons learned to improve future intelligence cycles in cybersecurity

- **Commodity Malware**: Widely available paid/free malware used by many threat actors

Information sharing and analysis communities:

- Healthcare: **H-ISAC**, Healthcare Ready
- Financial: **FS-ISAC**
- Aviation: **A-ISAC**
- Government: **EI-ISAC** (elections), **DIB-ISAC** (defense), **NEI** (nuclear)
- Critical infrastructure: **E-ISAC** (electricity), **ONG-ISAC** (oil & gas), **PT-ISAC** (public transit)

1.2 Given a scenario, utilize threat intelligence to support organizational security

Attack frameworks:

- **MITRE ATT&CK**:
  1. **Tactics**: High-level categories representing the goals or objectives of cyber adversaries, such as initial access, execution, persistence, privilege escalation, defense evasion, credential access, discovery, lateral movement, collection, exfiltration, and impact
  2. **Techniques**: Specific methods and procedures employed by attackers to achieve the objectives within each tactic, providing detailed insights into their actions
  3. **Procedures**: Specific instances or examples of how a technique has been observed or documented in real-world attacks
  4. **Software**: Malware, tools, and other software used by adversaries to execute their tactics and techniques
  5. **Groups**: Named threat actor groups or campaigns that exhibit similar behaviors and are associated with specific tactics, techniques, and procedures
  6. **Mitigations**: Recommended defensive measures and best practices to detect, prevent, or respond to attacks utilizing specific tactics and techniques

- **The Diamond Model of Intrusion Analysis**: Graphical depiction of intelligence in relation to network intrusion events using four elements:
  1. **Discovery**: The initial phase where defenders detect potential indicators of compromise (IOCs) or anomalies in the network.
  2. **Collection**: Gathering relevant information and data about the discovered IOCs to build a comprehensive understanding of the threat.
  3. **Processing**: Analyzing and processing the collected data to identify patterns, relationships, and potential attack techniques.
  4. **Dissemination**: Sharing the analyzed threat intelligence with relevant stakeholders and security teams to facilitate effective response and mitigation measures

- **The Lockheed Martin Cyber Kill Chain**:

1. **Reconnaissance**: The initial phase where attackers gather information about the target system and its vulnerabilities
2. **Weaponization**: Creating or obtaining the tools, malware, or exploits necessary to launch the attack
3. **Delivery**: Transmitting the weaponized payload to the target system, often through various means such as email attachments, malicious websites, or compromised software updates
4. **Exploitation**: The stage where the weaponized payload is executed on the target system, taking advantage of vulnerabilities to gain a foothold
5. **Installation**: Establishing a persistent presence on the target system, often achieved through backdoors or other means of maintaining access
6. **Command and Control (C2)**: Setting up communication channels that allow the attacker to control and manage the compromised system remotely
7. **Actions on Objectives**: The final stage where the attacker carries out their primary objectives, which can vary widely depending on their goals (e.g., data theft, system disruption, espionage, etc.)

Threat research and identification:

- **Reputation-based**: Detecting threats with IP/domain/file reputations
- **Behavioral-based**: Detecting threats based on their behavior. Behavioral detection depends on defining normal behavior and identifying deviations from it
- **Signature-based**: Identifying threats by comparing data, files, or network traffic against known patterns or signatures (often MD5 hashes) of known malware or attack techniques
- **Anomaly Detection**: Identifying threats by monitoring for unusual or anomalous activities, deviations from baselines, or statistical outliers that may indicate potential malicious behavior. Anomaly detection seeks to identify patterns that are statistically different from the norm, making it useful for detecting previously unknown or novel threats

- **Heuristic-based**: Identifying threats using rules or algorithms that detect patterns or characteristics commonly associated with known attack techniques, without relying on specific signatures
- **Indicators of Compromise (IoC)**: forensic data that identify potentially malicious activity on systems/networks
- **Common Vulnerability Scoring System (CVSS)**: a standardized system used to assess and communicate the severity of security vulnerabilities in software and systems (AV, AC, Au, C, I, A)

Threat modeling methodologies:

- **Adversary Capability**: Adversarial tool sets, skill sets, evasion techniques
- **Total Attack Surface**: The entirety of all potential points, both digital and physical, that could be exploited by threat actors to compromise an organization's security: network assets, software, hardware, human factors, and external connections
- **Attack Vector**: Any specific path or method used by threat actors to gain unauthorized access or exploit vulnerabilities in a system, network, or application, allowing them to carry out a successful cyber attack
- **Business/Organizational Impact**: An assessment of the severity of a cybersecurity incident on a business's operations and overall well-being
- **Threat Likelihood**: The probability or chance that a specific threat or cyber attack will occur, exploiting vulnerabilities in an organization's systems, networks, or applications

Threat intelligence sharing with supported functions:

- **Incident Response**: The coordinated effort taken by an organization to identify, contain, mitigate, and recover from cybersecurity incidents, minimizing their impact on information systems and assets
- **Vulnerability Management**: Identifying, assessing, prioritizing, and addressing security vulnerabilities in an organization's systems, applications, and network infrastructure to reduce the risk of potential cyber attacks
- **Risk Management**: Identifying, analyzing, evaluating, and mitigating potential threats and uncertainties to achieve an optimal balance between risk and reward for an organization

- **Security Engineering**: Designing and implementing secure systems, applications, and technologies to protect against potential cyber threats and vulnerabilities
- **Detection and Monitoring**: The process of actively observing and analyzing systems, networks, and data to identify and respond to security incidents, potential threats, or suspicious activities in real-time

1.3 Given a scenario, perform vulnerability management activities

Vulnerability identification:

- **Asset Criticality**: Determine the value, function, and impact of each asset on an organization's operations, data, and services. Rank them based on their significance to business objectives and potential consequences of compromise or disruption
- **Active vs. Passive Scanning**: Active scanning involves probing and interacting with the target system. Passive scanning observes and collects information without directly engaging with the system
- **Mapping and Enumeration**: Systematically scanning and probing the network to discover and identify active hosts, services, and devices as an assessment of the network's security posture and potential vulnerabilities

Validation:

- **True Positive**: Scanner correctly identifies existing vulnerability
- **False Positive**: Scanner reports a vulnerability that actually doesn’t exist on the system being scanned (verify patch/versions, or attempt actual attack)
- **True Negative**: Scanner correctly doesn’t alert on non-existent vulnerability
- **False Negative**: Scanner incorrectly doesn’t alert on vulnerability that actually exists on the system being scanned

Remediation/mitigation:

- **Configuration Baseline**: Perform anomaly detection to determine any deviations from the baseline (by using Snort, Suricata, Bro/Zeek, ELK Stack, Splunk, Weka, TensorFlow, etc.)
- **Patching**: Automate software deployments, updates, and security patches to protect the network infrastructure from known threats (using Ansible for Linux, or System Center Configuration Manager (SCCM) for Microsoft)
- **Hardening**: Disable unnecessary ports/services, configure firewalls, encrypt data in transit and data at rest, implement multi-factor authentication (MFA), enforce principle of least privilege, and regularly patch and update
- **Compensating Controls**: If upgrades and/or patches are unavailable, isolate the system and place compensating controls in front
- **Risk Acceptance**: After determining the Annual Loss Expectancy, it might best suit the business to simply accept the risk and refrain from taking any action against it, if the mitigation cost is more than the expected loss
- **Verification of Mitigation**: Audits (formal), assessments (informal), patch consistently, automate repeated vulnerability scans

Scanning parameters and criteria:

- **Risks associated with scanning activities**: Vulnerability scans consume bandwidth and resources, and risk business process interruptions (solution: fine-tune intensity and scan times)
- **Vulnerability feed**: The Security Content Automation Protocol (SCAP) outlines the following standards:
    1. **CCE** (Common Configuration Enumeration) - CCE codes are used to provide a consistent, unique identifier for specific configuration settings
    2. **CPE** (Common Platform Enumeration) - CPE codes are used to represent specific products or platforms
    3. **CVE** (Common Vulnerabilities and Exposures) - CVE standardizes the identification of vulnerabilities across many systems and platforms
    4. **CVSS** (Common Vulnerability Scoring System) - CVSS indicates the level of severity that a given vulnerability can result in
    5. **XCCDF** (Extensible Configuration Checklist Description Format) - XCCDF helps organizations check their compliance against a list of standard security configurations

6. **OVAL** (Open Vulnerability and Assessment Language) - OVAL enables organizations to share detailed information about security vulnerabilities, making it easier to manage security-related information across many environments

- **Scope**: The extent of a vulnerability scan (included systems/networks, host discovery methods, what tests will be conducted against active hosts)
- **Credentialed vs Non-credentialed**: Credentialed scans use valid login credentials to access the target system (enabling deeper and more comprehensive scanning). Non-credentialed scans conduct assessments without authentication, providing limited visibility into the system
- **Server-based vs Agent-based**: Server-based scans are conducted from a centralized server or scanning appliance (scanning multiple targets remotely). Agent-based scans use lightweight software agents installed on individual machines to perform local scans and report back to the central server
- **Internal vs External**: Internal scans are conducted from within the network perimeter, whereas external scans are performed from outside the network, evaluating the organization's internet-facing systems and identifying potential vulnerabilities visible to external attackers

Special considerations for vulnerability management:

- **Types of data**: Protected Health Information (**PHI**), Payment Card Industry (**PCI-DSS**), Personally Identifiable Information (**PII**), Non-public Personal Information (**NPI**), Sensitive Personal Information (**SPI**), and Intellectual Property (**IP**)
- **Technical constraints**: Capabilities of the scanning system; frequency limitations
- **Workflow**: Remediation workflow (detection > remediation > testing)
- **Sensitivity levels**: Minimum severity rating (Low, Medium, High, Critical)
- **Regulatory requirements**: PCI DSS (internal and external) must be performed at least quarterly by a qualified professional or Approved Scanning Vendor (ASV); FISMA - risk assessment, continuous monitoring, and risk mitigation (updated scanning tools, update vulnerability list before/after scan, etc.)
- **Segmentation**: Compliance networks can be segmented to reduce scan scope

- **IPS, IDS, and Firewall settings**: Internal placement is for protecting against insider threats; External placement is for protecting against external attacks

Inhibitors to remediation:

- **Memorandum of Understanding (MOU)**: Formal written agreement between two or more parties that outlines the terms and intentions of a cooperative or collaborative relationship, *without creating a legally binding contract*
- **Service Level Agreement (SLA)**: Formal contract or agreement between a service provider and a customer that outlines the specific services to be provided, performance expectations, and responsibilities of both parties
- **Organizational governance**: Bureaucracy, budget constraints, interdepartmental coordination, risk tolerance, compliance and policy, vendor involvement, etc.
- **Business process interruption**: Sometimes taking down systems can cause a significant interruption to business
- **Degrading functionality**: Service degradation can lead to business process interruption
- **Legacy systems**: End of Life (EoL) unsupported systems cannot receive security updates
- **Proprietary systems**: Different vendors are unable to work with certain systems; some vendors will not have patches/updates

1.4 Given a scenario, analyze the output from common vulnerability assessment tools

Web application scanners:

- **OWASP ZAP (Zed Attack Proxy)**: Open-source security testing tool designed to help identify and remediate vulnerabilities in *web applications*
- **Burp Suite**: A powerful cybersecurity tool designed for *web application* security testing and penetration testing, providing comprehensive scanning, testing, and vulnerability assessment capabilities

- **Nikto**: Open-source *web server* scanner that detects potential security vulnerabilities and misconfigurations, providing valuable insights for cybersecurity assessments
- **Arachni**: Open-source, high-performance security scanner designed to identify vulnerabilities and security issues in *web applications* through automated crawling and analysis

Infrastructure vulnerability scanners:

- **Nessus**: Widely-used *commercial* vulnerability scanner that helps identify security risks and weaknesses in networks, systems, and applications
- **OpenVAS**: Open-source vulnerability scanner that performs comprehensive security assessments of networks and applications, helping identify and address potential vulnerabilities
- **Qualys**: *Cloud-based* cybersecurity and compliance platform that offers a range of *commercial* security solutions: vulnerability management, threat protection, and compliance assessment, helping organizations *proactively* safeguard their digital assets

Software assessment tools and techniques:

- **Static Analysis**: Examines the source code or application *without executing it,* identifying potential vulnerabilities and coding errors
- **Dynamic Analysis**: Evaluates the behavior of an application *while it is running,* examining its interactions and responses, to identify security flaws
- **Reverse Engineering**: The process of analyzing software or hardware to understand its design, functionality, and underlying code
- **Disassembler:** Takes executable machine code (binary code) and converts it back into human-readable assembly language, allowing reverse engineers to analyze and understand the program's instructions, control flow, and logic
- **Fuzzing**: Automated testing technique that involves sending random or malformed data inputs into a program to identify potential software vulnerabilities or crashes
- **Penetration Testing (Pen Test)**: Ethical hackers simulate real-world attacks to identify and exploit vulnerabilities in systems, networks, and applications

- **Code Review**: Manual examination of source code to identify security flaws, coding errors, and potential vulnerabilities
- **Dependency Analysis**: Identifying and analyzing software dependencies to detect vulnerable third-party libraries or components

Enumeration:

- **Nmap**: Returns port listing, services running, MAC addresses, OS/kernel version, network distance, runtime
- **hping**: Packet crafting tool that sends TCP/UDP/ICMP/RAW-IP; can be used for firewall testing, TCP/IP auditing, and network testing
- **Active vs Passive**: Active enumeration involves interacting with the target system or network, while passive enumeration involves observing and collecting information without direct interaction or disruption
- **Responder**: Local Link Multicast Name Resolution (LLMNR) and/or NetBIOS Name Service (NBT-NS) poisoner/rogue authentication server that captures and manipulates authentication requests, allowing attackers to perform attacks like NTLMv1 and NTLMv2 credential harvesting in certain scenarios. Steals NT Lan Manager (NTLM) hashes

Wireless assessment tools:

- **Aircrack-ng**: Suite of WiFi monitoring, attacking, testing, and cracking (WEP/WPA) tools
- **Reaver**: Brute force security testing tool against WPS PINs to recover WPA/WPA2 passphrases. Highlights the insecurity of WPS wireless networks
- **oclHashcat**: GPU-based hash cracker with dictionaries, masks, rules, etc. It’s designed to crack various types of password hashes, including MD5, SHA-1, SHA-256, NTLM, and others

Cloud infrastructure assessment tools:

- **ScoutSuite**: Open-source security auditing tool for security posture in cloud environments, providing comprehensive and automated checks for potential misconfigurations and security risks across various cloud platforms; utilizes “longitudinal survey panels” to track and monitor the cloud environment

- **Prowler**: Open-source security assessment tool designed to audit AWS environments for security best practices, potential vulnerabilities, auditing, hardening, and forensics
- **Pacu**: Open-source penetration testing tool that automates various security testing scenarios and helps assess the security posture of AWS. It’s an exploitation framework with modules to exploit AWS configuration flaws

1.5 Explain the threats and vulnerabilities associated with specialized technology

Specialized technologies:

- **Mobile**: Malware; unpatched devices; jailbreaking; data leaks; OS vulnerabilities
- **Internet of Things (IoT)**: Weak passwords; insecure services; lack of security update; outdated component use; insecure data transfer/storage; lack of secure/physical device management
- **Embedded platforms**: Programming errors; web vulnerability; weak access control/authentication
- **Real-Time Operating System (RTOS)**: Remote Code Execution (RCE); Denial of Service (DoS); information leak; improper access control
- **System on a Chip (SoC)**: Low-level hardware bugs (boot header modification; partition header table parsing)
- **Field-Programmable Gate Array (FPGA)**: Fault injection; hardware trojans; design leaks; potential weaknesses implemented during foundry fabrication
- **Physical access control**: Insufficient access control; lack of training; unattended assets
- **Building automation systems**: Hard-coded secrets; Buffer Overflow errors; Cross-site Scripting (XSS); path traversal; authentication bypass

Vehicles and drones:

- **CAN bus**: Unauthorized remote access, message manipulation, Denial of Service (DoS) in part due to vulnerabilities such as lack of authentication and encryption, potentially leading to vehicle or drone control compromise

- **Workflow and process automation systems**: Third-party platform vulnerabilities (supply chain risks, patch management, etc.); Identity Access Management (IAM) issues with the API (Application Programming Interface)
- **Industrial Control Systems (ICS)**: Improper credentials management; weak firewall rules; network design weaknesses

SCADA:

- **Modbus**: Plaintext transmission; no authentication; command injection; weak sessions

1.6 Explain the threats and vulnerabilities associated with operating in the cloud

Cloud service models:

- **SaaS**: Customer only chooses application; hardware managed by provider; access control
- **PaaS**: Configurable hardware and software/development tools; data protection
- **IaaS**: Configurable hardware; virtual machine management (VM escape; virtual host patching; virtual guest issues [patching]; virtual network issues)

Cloud deployment models and their vulnerabilities:

- **Public**: Public cloud provider sells services to consumers
- **Private**: Internal enterprise sells service to internal customers
- **Community**: Several companies work on same platform
- **Hybrid**: Mix of on-premises, private cloud, and public cloud
- **FaaS/Serverless Architecture**: Apps are hosted by third party; all server software/hardware management is handled by the provider
- **Infrastructure as Code (IaC)**: Managing/provisioning data centers using machine-readable files
- **Insecure API**: Internet-exposed management APIs can have software vulnerabilities (e.g. anonymous access; plaintext authentication; improper authorizations)

- **Improper key management**: Unencrypted; internet-exposed key server; weak/reused key
- **Unprotected storage**: Insider threats; malicious file entry; impersonation; worm that is auto-synced to the cloud, and spread from the cloud to other users

Logging and monitoring:

- **Insufficient logging and monitoring**: Late detection; undetected password spraying; ignored alerts; unidentified suspicious activity
- **Inability to access**: Access logs provide info about failed requests made to the cloud

1.7 Given a scenario, implement controls to mitigate attacks and software vulnerabilities

Attack types:

- **XML attack**: Web Application Firewall (Input validation, parameterized queries, escaping and encoding, enforcing access control policies, error handling, etc.); disable external entities; sensitive data not serialized
- **SQL injection (SQLi)**: Web Application Firewall (Input validation, parameterized queries, escaping and encoding, XML parsers, error handling); input sanitization; least privilege restrictions for databases

Overflow attack:

- **Buffer Overflow**: Address Space Layout Randomization (ASLR); Data Execution Prevention (DEP); No-Execute bit (NX bit); use secure functions; higher-level languages; input validation
- **Integer Overflow**: Range checking; prefer unsigned integers; use safer code implementations
- **Heap Overflow**: Higher-level languages; input validation; safe compilers; proper patching

General attacks:

- **Remote code execution**: Avoid using user input inside evaluated code; strict file upload extensions, etc.
- **Directory traversal**: Ensure user cannot supply entire file path; accept known-good input
- **Privilege escalation**: Avoid using administrative privileges; separate privilege areas
- **Password spraying**: MFA; strong passwords; user training; logging/monitoring
- **Credential stuffing**: MFA; CAPTCHA; unpredictable usernames; check against leaks
- **Impersonation**: Use of session identifiers; packet filtering; Dynamic Address Inspection (DAI); encrypted protocols
- **Man-in-the-middle attack**: Session encryption; ensure only valid certificates are used
- **Session hijacking**: Key/cookie/link encryption; secure and HttpOnly flags for cookies
- **Rootkit**: patching; layered security; heuristic analysis; antivirus

Cross-site scripting:

- **Reflected**: WAF; use appropriate response headers; avoid suspicious links
- **Persistent**: WAF; filter input and encode data on output; escape HTML data on arrival
- **Document Object Model (DOM)**: Don’t treat untrusted data as code; delimit untrusted data as quoted strings

Vulnerabilities and their remediations:

- **Improper error handling**: Info leak through over-detailed error messages. Solutions: error handling policy; error logging; graceful handling of all possible errors
- **Dereferencing**: Get value (NULL) in memory pointed by pointer. Solutions: process failure; higher-level programming languages; sanity-check pointers prior to use
- **Insecure object reference (IOR)**: Exposure of reference to internal object. Solutions: user authorization; make objects harder to enumerate

- **Race condition**: Produces unexpected results when timing of actions impact other actions. Solutions: careful programming; locking (at most one thread can modify database)
- **Broken authentication**: Brute-forcing credentials; unexpired session tokens. Solutions: MFA; no default credentials; password policy; delay failed attempts; session management
- **Sensitive data exposure**: Stolen keys; MITM; stolen plaintext data (server/transit/client). Solutions: data classification; secure encryption; key management; salted hashes
- **Insecure components**: Public exploits for known vulnerabilities. Solutions: check product versions; monitor for unmaintained products (virtual patch/WAF)
- **Insufficient logging and monitoring**: Lack of timely response; late detection/monitoring; failure logging. Solutions: centralized logs; tamper prevention; timely incident response
- **Weak or default configurations**: Unpatched flaws; default accounts; unprotected files. Solutions: hardening; minimalistic platforms; segmentation; review and update configurations

Use of insecure functions:

- **strcpy**: Allows buffer overflow; use proper input validation; use secure functions

2.0 Software and Systems Security

2.1 Given a scenario, apply security solutions for infrastructure management

Asset management:

- **Asset Tagging**: Assign labels including classification; unique ID; asset tracking system

Segmentation:

- **Physical**: Placing network devices to control access > new hardware + additional costs
- **Virtual**: VLANs/subnets on top of existing infrastructure > no new hardware/costs
- **Jumpbox**: Intermediary connection point from untrusted to trusted network

System isolation:

- **Air Gap**: Isolate system from other networks/Internet; physical isolation (transfer with USBs or other removable media)

Network architecture:

- **Physical**: Defense-in-depth security appliance; segmentation; physical security
- **Software-Defined Network (SDN)**: Separates the control plane from the data plane, allowing centralized software controllers to dynamically manage network resources and configurations; TLS; secure tunneling; SDN controller hardening; access control
- **Virtual Private Cloud (VPC)**: A virtual network infrastructure within a public cloud platform that allows users to isolate and control their resources and securely deploy applications; traffic/anomaly monitoring; ingress/egress traffic control; secure VPC connections
- **Virtual Private Network (VPN)**: A secure and encrypted connection that enables users to access and transmit data over a public network as if they were directly connected to a private network; strong authentication; avoid DNS leaks; use a kill switch (drop Internet if VPN fails)
- **Serverless**: Network architecture where resources are dynamically provisioned and managed in the cloud, allowing for more flexible, scalable, and cost-efficient network operations; log monitoring, Identity Access Management (IAM), secured secrets, input validation, secure libraries
- **Change Management**: Change identification > request > request review > prioritization > evaluation/impact analysis > approval/rejection > testing > implementation > review

Virtualization:

- **Virtual Desktop Interface (VDI)**: Allows desktop operating systems and applications to run and be managed in virtual machines on centralized servers, enabling remote access and efficient desktop management; easy patching, antivirus, etc.
- **Containerization**: Lightweight virtualization technology that allows applications to be packaged with their dependencies into isolated units; portable, scalable, and easily deployable across various environments

Identity and Access Management:

- **Privilege management**: Least privilege; privileged account usage monitoring; prevent privilege creep; role-based authorization
- **MFA**: Multiple authentication methods (knowledge; possession; biometric; location)
- **SSO**: Authenticate once to use multiple systems; reduces password reuse, resets, and support
- **Federation**: Sharing of customer info to Service Providers (SPs); trust relationship between Identity Provider (IdP), Service Provider and user
- **Role-based**: Access decision is based on roles; permissions assigned to roles not users
- **Attribute-based**: Based on context (e.g. time, location, access frequency, behavior)
- **Mandatory**: End users cannot modify security permissions set by administrators
- **Manual review**: Review of access change logs, alerts, employee profiles, procedures
- **Cloud Access Security Broker (CASB)**: Cloud security solution providing visibility and control over data and activities in cloud environments; helps organizations enforce security policies and protect sensitive data; intermediary between an organization's on-premises infrastructure and cloud service providers
- **Honeypot**: Intentionally vulnerable system that monitors attackers for intentions and blacklists the IP address

- **Monitoring and logging**: Security Information and Event Management (SIEM); privileged use/change/grant, account creation/modification, terminated account usage, account lifecycle events, separation of duty
- **Encryption**: Salted hashes; encrypted traffic; encrypted keys/data/session identifiers
- **Certificate management**: Creation > storage > dissemination > suspension > revocation
- **Active defense**: Identity Provider (IdP) notifies account owners/service providers (SPs); SPs respond to IdP/authorization system/account compromise

2.2 Explain software assurance best practices

Platforms:

- **Mobile**: Secure coding, code review, encryption, secure APIs
- **Web Application**: Input validation, secure authentication, secure session management, regular updates
- **Client/Server Platforms**: Secure communication, secure APIs, least privilege
- **Embedded**: Secure boot, code signing, firmware updates
- **System on a Chip (SoC)**: Hardware security, secure boot
- **Firmware**: Secure development, patch management
- **Software Development Lifecycle (SDLC) Integration**: Requirements/criteria definition; secure design; static analysis and peer code review; testing and analysis + user acceptance testing
- **DevSecOps**: Identify vulnerabilities; find and prioritize risk remediation; secure workflow

Software assessment methods:

- **User Acceptance Testing (UAT)**: Ensures software users are satisfied with the functionalities
- **Stress test application**: Ensures application availability and scalability; maximum load

- **Security regression testing**: Ensures no new vulnerabilities/misconfigurations are introduced by patches/updates; examples include: change control, Version Control System (VCS), Software Configuration Management (SCM)
- **Code review**: Pair programming; over-the-shoulder; pass-around; tool-assisted

Secure coding best practices:

- **Input validation**: Validate all untrusted data; specify character sets + data types/length; whitelist allowed characters; additional controls for hazardous characters
- **Output encoding**: Encode all unsafe characters; sanitize SQL, XML queries and operating system commands
- **Session management**: Short session inactivity timeout; new session identifier generation; logout available from any authorized page; secure session ID algorithms
- **Authentication**: Central, segregated authentication; POST requests; nonspecific error codes; encrypted and securely stored (salted hash) credentials
- **Data protection**: Least privilege; protect/purge sensitive caches; secure encryption; no plaintext password storage; disable client-side caching; access controls for sensitive data
- **Parameterized queries**: Use placeholders to separate query and data > prevents SQL query altering (SQLi)
- **Static analysis tools**: Thorough white-box code review to identify programming errors
- **Dynamic analysis tools**: Test inputs during code execution for complex vulnerabilities
- **Formal methods for verification of critical software**: Fagan inspection (planning > overview > preparation > meeting > rework > follow-up)

Service-oriented architecture:

- **Security Assertion Markup Language (SAML)**: XML-based standard for exchanging authentication and authorization data between different security domains, commonly used in Single Sign-On (SSO) scenarios

- **Simple Object Access Protocol (SOAP)**: Facilitates communication between applications running on different platforms and built using different programming languages
- **Token-based/digest authentication**: Validate digital signatures; encrypt data with keys
- **Representational State Transfer (REST)**: An architectural style for designing networked applications and web services; access and manipulate textual representations of web resources with HTTP
- **HTTPS**: Access control; API keys; whitelist HTTP methods; input validation
- **Microservices**: App is a collection of loosely coupled services; lightweight protocols
- **IAM with OAuth**: Defense-in-depth; use open source crypto libraries; automatic security updates; distributed monitoring/scanning; single point of entry (API gateway)

2.3 Explain hardware assurance best practices

Hardware root of trust:

- **Trusted Platform Module (TPM)**: Generates/stores cryptographic keys; full disk encryption; keeps hardware locked until authentication is complete; motherboard-embedded chip
- **Hardware Security Module (HSM)**: Manage/generate/store cryptographic keys; removable or external device
- **eFuse**: Manufacturer can change circuits on a chip while it is in operation
- **Unified Extensible Firmware Interface (UEFI)**: Secure boot (only signed apps used at boot; operating system needs recognized key in order to boot). Boot phases:
    1. **Security (SEC) Phase:** This initial phase is executed by the firmware on the system's CPU, where the firmware performs essential hardware initialization and integrity checks of the system firmware
    2. **Pre-EFI Initialization (PEI) Phase:** The PEI phase extends the firmware initialization process, enabling access to more hardware devices and providing the necessary services for subsequent phases

3. **Driver Execution Environment (DXE) Phase:** In this phase, UEFI drivers and UEFI applications are executed, allowing for more advanced hardware initialization, configuration, and providing higher-level services for boot and runtime operations
4. **Boot Device Selection (BDS) Phase:** The BDS phase is responsible for selecting the boot device and loading the UEFI Boot Manager
5. **Transient System Load (TSL) Phase:** In this phase, the UEFI Boot Manager loads the operating system bootloader, such as GRUB or Windows Boot Manager, from the boot device
6. **Runtime (RT) Phase:** After the operating system bootloader takes over, the RT phase provides runtime services and interfaces to UEFI applications during the system's runtime

- **Trusted Foundry**: Department of Defense (DoD) program to secure supply chain of hardware for military

Secure processing:

- **Trusted execution**: Assures operating system trust using Trusted Platform Module (TPM); prevents system/BIOS code corruption or platform configuration modification from stealing sensitive data (Intel)
- **Secure enclave**: Separately booted microkernel to store private decryption keys; apps never have direct access to the keys (Apple)
- **Processor security extensions**: Core can switch to secure state (only trusted code can run; can access secure memory; strict security state entry control) (ARM)
- **Atomic execution**: Cannot be interrupted by other threads; thread locking; shared data is always valid > thread safety > prevent race conditions
- **Anti-tamper**: Unusual screws/bolts; secure crypto-processors; zeroize when tampered; chips can’t be accessed externally; fracture when interfered
- **Self-encrypting drive**: User password to decrypt media; encrypt as data is written and decrypt as data is retrieved; encryption is invisible to user (can’t be turned off)
- **Trusted firmware updates**: The ability to update the firmware (software) of a device or component in a secure and trusted manner; this is often used to fix

vulnerabilities, add new features, or apply patches to improve the security, functionality, or performance of the device (Intel)

- **Measured boot and attestation**: Object signature hashes are recorded in Trusted Platform Module (measured boot); host reliably authenticates hardware/software config to remote server to determine level of trust in platform (remote attestation)
- **Bus encryption**: Encrypted instructions in data bus; executed by cryptoprocessor

3.0 Security Operations and Monitoring

3.1 Given a scenario, analyze data as part of security monitoring activities

- **Heuristics**: Detects unknown threats (no signature) based on their behavior
- **Trend analysis**: Identifies unexpected changes that don't match expected growth rates; predicts behaviors based on existing data (e.g. network congestion based on bandwidth)

Endpoint:

Malware:

- **Reverse engineering**: Sandboxing; code detonation; software fingerprinting to compare malware against existing hashes; decompilers/disassemblers
- **Memory**: Monitor process memory consumption and set thresholds; prevent buffer overflow/insufficient memory allocation and memory leaks (causes app/system to crash)

System and application behavior:

- **Known-good behavior**: Establish baselines to compare against for anomalies

- **Anomalous behavior**: Suspicious activity that deviates from the baseline model
- **Exploit techniques**: Memory overflows; Denial of Service (DoS); beaconing (botnet); data exfiltration; privilege escalation; new accounts, etc.
- **File system**: File Integrity Monitoring (FIM); file creation/modification/deletion; prevent drive capacity outage
- **User and Entity Behavior Analytics (UEBA)**: Pattern-based user activity anomaly detection (for insider threats; detecting if attacker has compromised system; breaches; brute-forces; superuser creations)

Network:

URL and DNS analysis:

- **Dynamically generated algorithms**: Malware creates a large number of domain names to connect to C2 servers > harder botnet control; often called Fast Flux; uses datetime, words etc.
- **Flow analysis**: Monitor bandwidth, flow sources, utilization, endpoints, applications

Packet and protocol analysis:

- **Malware**: Check destination IP address/port, protocols, flag fields, sequence number, etc.

Log review:

- **Event logs**: Logins, service start/stop, file activity, rights usage; Windows (application logs, security logs, setup logs, system logs, ForwardedEvents logs)
- **Syslog**: 8 log levels for event notification severity (EACEWNID):
    1. Emergency: System is unusable
    2. Alert: Action must be taken immediately
    3. Critical: Critical conditions
    4. Error: Error conditions
    5. Warning: Warning conditions
    6. Notice: Normal but significant conditions
    7. Informational: Informational messages

8. Debug: Debug-level messages

- **Firewall logs**: Successful/blocked traffic characteristics; threat attempts; bandwidth use
- **Web Application Firewall (WAF)**: Web traffic; scalability thresholds; detailed requests log (e.g. status, header info)
- **Proxy**: User/app requests; user agents; HTTP methods; response length; resource access
- **Intrusion Detection System (IDS) / Intrusion Prevention System (IPS)**: Attack attempts alert; attack types/sources, target devices; trends

Impact analysis:

- **Organizational impact vs Localized impact**: Threat has organizational scope vs local scope
- **Immediate vs Total**: Impact of threat when activated vs. until fully resolved

Security Information and Event Management (SIEM) review:

- **Rule writing**: Take action (e.g. trigger alert) if event occurs > quick incident response
- **Known-bad IP**: Global blacklists of suspected malicious IPs/URLs; reputation analysis
- **Dashboard**: Overview of aggregated info; customize to include important events, graphs, etc.

Query writing:

- **String search**: Searches in (specified) columns and tables for string (wildcards, conditions, etc.)
- **Script**: Use languages to query for items from event logs (according to time, severity, etc.)
- **Piping**: Redirects output as input to the following command for filtering, sorting, aggregating, etc.

E-mail analysis:

- **Malicious payload**: Antivirus + email gateway (Machine Learning + real-time IP reputation) + attachment scanning (sandboxing; behavior-based analysis)

- **Domain Keys Identified Mail (DKIM)**: Receiver checks that domain owner indeed sent/authorized the email + assure message/attachments weren't modified (encrypted signature)
- **Domain-based Message Authentication, Reporting, and Conformance (DMARC)**: Prevents spam/spoofing/phishing through DMARC policies; defines email authentication, actions on failed emails, reporting (XML statistics; message copies)
- **Sender Policy Framework (SPF)**: Prevents spammers sending emails on behalf of domain; publishes authorized mail servers (allowed to send on behalf of domain); gives receivers trust information on email origin
- **Phishing**: Source IP; URLs; attachments; typosquatting; Sender Policy Framework (SPF)
- **Forwarding**: Compromised inbox can automatically forward received email to attacker
- **Digital signature**: Ensures sender authenticity + prevents message tampering (unique)
- **Email signature block**: Customizable text at bottom of email (not unique)
- **Embedded links**: URL analysis to identify known spam/threat against blacklist
- **Impersonation**: Prevent spoofing (SPF/DKIM/DMARC) + user education (check address)
- **Header**: Fields (e.g. Received, Reply-To, Return-Path, SPF, X-Mailer, X-Distribution)

3.2 Given a scenario, implement configuration changes to existing controls to improve security

- **Permissions**: Discretionary Access Control (DAC) - end users can delegate/control permissions); Mandatory Access Control (MAC) - end users cannot modify permissions; Role-based Access Control (RBAC) - rights granted to roles > limits access/functions
- **Whitelisting**: Only allows specific IP/MAC addresses, apps, files, emails (more strict)
- **Blacklisting**: Prevents specific IP/MAC addresses, apps, files, emails (simple, less secure)

- **Firewall**: Add stateful filtering rules/Access Control Lists (ACLs); prevent traffic based on 5-tuple or L7 content
- **Intrusion Prevention System (IPS) rules**: Connection-based block; rules to identify known attack signatures > action
- **Data Loss Prevention (DLP)**: Detects/prevents sensitive data exfiltration; compliance; data tracking/visibility
- **Endpoint Detection and Response (EDR)**: Detects endpoint activities/events for visibility (signature-based; behavioral analysis) + context with threat intelligence > quick incident response
- **Network Access Control (NAC)**: 802.1x; agent-based (requesting devices needs special software) or agentless (web browser authentication); in-band (dedicated appliances) or out-of-band (existing network infrastructure)
- **Sinkholing**: DNS requests for known malicious domains are redirected to a controlled, benign IP address, often hosted by a security organization or an Internet service provider; remediates botnet-infected system looking for C2 server

<u>Malware signatures:</u>

- **Development/rule writing**: Record malware identifiers (e.g. unique strings, malware families, resources within malware, sometimes called function bytes)
- **Sandboxing**: Detects unknown malware based on behaviors, not signatures, then isolates it in a sandbox, where its activities can be safely observed
- **Port security**: Restricts source MAC addresses that can connect to port; static or dynamic filtering (set a maximum number of MAC addresses, move MAC address to a different port, block MAC addresses, etc.)

3.3 Explain the importance of proactive threat hunting

- **Establishing a hypothesis**: Intelligence-driven (Tactics, Techniques, and Procedures [TTPs] via Indicators of Compromise [IOCs]); awareness-driven (network changes, most important assets); analytics-driven (models to avoid bias)

- **Profiling threat actors and activities**: Motivations, objectives, targets, geolocations, languages, budget, technical skills > relevance to organization and threat severity

Threat hunting tactics:

- **Executable process analysis**: Behavior anomaly analysis (execution path, parent name)
- **Reducing the attack surface area**: Eliminate complexity; attack simulation; endpoint visibility + network policies; network segmentation; assessments and traffic analysis
- **Bundling critical assets**: Assets grouped together for ease of management and control
- **Attack vectors**: How an attacker compromises systems through exploiting vulnerabilities
- **Integrated intelligence**: Knowledge + info + collaboration > rapid actionable intelligence
- **Improved detection capabilities**: detect unidentified threat activity based on Tactics, Techniques, and Procedures (TTP) analysis

3.4 Compare and contrast automation concepts and technologies

- **Workflow orchestration**: Scalable cloud resource provisioning to achieve business targets
- **Scripting**: Programming languages to automatically manage tasks, e.g. configure devices
- **API integration**: Controller interaction with systems; seamless connectivity between apps
- **Automated malware signature creation**: Inbound unknown file monitoring for file behavior and content classifiers; signature generated based on malware classification
- **Data enrichment**: Adding context to data (e.g. asset inventory tools, third-party databases) > meaningful insights + threat prioritization + quick investigation and action

- **Threat feed combination**: Combine machine data from many sources to Security Information and Event Management (SIEM) and User and Entity Behavior Analytics (UEBA)
- **Machine Learning**: Finds patterns in data; threat anomaly monitoring; detects unidentified malware; analyzes encrypted traffic; makes predictions based on activity and behavior

Use of automation protocols and standards:

- **Security Content Automation Protocol (SCAP)**: Security automation with languages (OVAL), enumeration (CVE, CPE, CCE), metrics (CVSS), integrity (Trusted Model for Security Automation Data [TMSAD] for authentication and traceability of security data), etc.
- **Continuous Integration**: frequent code commits; automatic code testing; master code branch remains production-ready
- **Continuous Deployment**: Code changes are automatically and instantly deployed to production without human intervention; identical development + test + production environment configuration
- **Continuous Delivery**: Code changes are automatically prepared for deployment and are maintained in a deployable state, but the actual release to production is triggered manually; identical development + test + production environment configuration

4.0 Incident Response

4.1 Explain the importance of the incident response process

Communication plan:

- **Limiting communication to trusted parties**: Law enforcement, Information Sharing and Analysis Center (ISAC) partners, vendors/manufacturers, actual/potential victims, media only on a need-to-know basis (or as policy or regulation requires)

- **Disclosing based on regulatory/legislative requirements**: Data breach notification laws
- **Preventing inadvertent release of information**: Always consult legal counsel/public relations before communicating with law enforcement, media, public, etc.
- **Using a secure method of communication**: Security-tested messaging platforms; consider message retention, monitoring, and response
- **Reporting requirements**: Regulations; classification; storage; retention; expiration policies

Response coordination with relevant entities:

- **Legal**: Ensures team complies with laws/policies/regulations + leader compliance advice
- **Human resources**: Investigates potential employee malfeasance
- **Public relations**: Coordinate communications with the media and the public
- **Internal and External**: Within team for rapid response, and externally for advice/regulatory
- **Law enforcement**: When incident has criminal nature, investigation cooperation is necessary
- **Senior leadership**: Makes critical business decisions; allocates budget and staff policies
- **Regulatory bodies**: Provides advice/guidance on regulatory/legal compliance

Factors contributing to data criticality:

- **Personally Identifiable Information (PII)**: Info which can distinguish an individual's identity (name, SSN, DoB, addresses, etc.)
- **Protected Health Information (PHI)**: HIPAA-regulated health info (medical records, health conditions, etc.)
- **Sensitive Personal Information (SPI)**: Doesn't necessarily identify individual, but is private/can harm person if made public
- **High value asset**: Critical information with a serious impact to an organization's business and/or mission capability

- **Financial information**: Private information about assets, payments, cards, accounts, etc.
- **Intellectual property (IP)**: Proprietary product development plans, formulae, trade secrets, etc.
- **Corporate information**: Sensitive info such as corporate accounting, mergers and/or acquisitions

4.2 Given a scenario, apply the appropriate incident response procedure

Preparation:

- **Training**: Appropriate training on roles and responsibilities; incident preparation
- **Testing**: Incident response drill scenarios, mock data breaches > IR plan evaluation
- **Documentation of procedures**: Tactical details prepared and used during incidents

Detection and analysis:

- **Characteristics contributing to severity level classification**: Functional impact, economic impact, recoverability effort, data (information) impact rating
- **Downtime**: Amount of time that service is unavailable; time until recovery
- **Recovery time**: Possibility/predictability of recovery time; resource requirements
- **Data integrity**: Modification or deletion of sensitive, proprietary, regulatory, or legal information
- **Economic**: Financial losses classified according to thresholds
- **System process criticality**: Prioritize systems based on how vital it is to operation
- **Reverse engineering**: Analyze malware, identify how it works and then establish Indicators of Compromise (IOCs) for rules
- **Data correlation**: Information from multiple sources should be centrally analyzed to identify attacks

Containment:

- **Segmentation**: Network segmentation combined with firewalls; isolate attacker to quarantine network (strictly controlled VLAN for compromised host analysis)
- **Isolation**: Allow attacker access to systems (quarantine network via Internet) but restrict access to other systems (sandbox, honeypot, etc.)

Eradication and recovery:

- **Vulnerability mitigation**: Perform vulnerability scans; protect systems against future attacks
- **Sanitization**: Clear (sanitize against simple recovery, factory reset); purge (prevent even laboratory recovery, e.g. degaussing); destroy (unable to be re-used ever, e.g. melting)
- **Reconstruction/reimaging**: All compromised hosts should be rebuilt from scratch or a known trusted backup; ensure backups don't re-introduce the vulnerability
- **Secure disposal**: Encrypt and delete encryption key (cryptographic deletion), or physically destroy media and use a third-party collector
- **Patching**: Patch directly involved systems, indirectly involved systems, and any other adjacent systems
- **Restoration of permissions**: Perform account review; check for principle of least privilege violations; ensure only authorized user accounts exist on every system
- **Reconstitution of resources**: Rebuild systems and apply updates and patches
- **Restoration of capabilities and services**: Bring affected systems back into production
- **Verification of logging/communication to security monitoring**: Configured to meet logging policy requirements; check centralized log receipt; log automation

Post-incident activities:

- **Evidence retention**: Follow retention policies (no court use); consult legal counsel before discarding (prosecution); US government agencies must retain records for 3 years

- **Lessons learned report**: Evaluates how incident response was performed; suggest improvements in the future; evaluate plan/procedure effectiveness
- **Change control process**: Document emergency changes that bypassed normal configuration management/change control process (return to them post-incident for review)
- **Incident response plan update**: Find plan deficiencies; make changes to IR plan
- **Incident summary report**: Useful in new security control development/training; legal record; previously undetected deficiencies; event timeline + root cause + evidences + actions and their reasons + validation results + lessons learned
- **Indicators of Compromise (IOCs) generation**: IOCs based on network/host artifacts, addresses, hashes, tools, TTPs, etc.
- **Monitoring**: Full network visibility; continuous monitoring for future persistent attack

4.3 Given an incident, analyze potential indicators of compromise

Network-related:

- **Bandwidth consumption**: Causes service outages/disruptions > flow data tools, threshold-based alarms, real-time graphs, Simple Network Management Protocol (SNMP) device-level load monitoring
- **Beaconing**: HTTP/S traffic sent to C2 server from a botnet system > IDS/IPS with known C2 server rules, behavior-based analysis, outbound traffic analysis
- **Irregular peer-to-peer communication**: P2P botnets > DNS lookup anomaly detection
- **Rogue device on the network**: Wired and/or wireless rogues > validate MAC addresses to whitelist, Organizationally Unique Identifier (OUI) checking, network scans, site surveys, traffic analysis, port security/Network Access Control (NAC)
- **Scan/sweep**: Port scanning, repeated requests, etc. > IDS/IPS + Security Information and Event Management (SIEM) > attack correlation
- **Unusual traffic spike**: Scan traffic and attack traffic > anomaly/heuristics detection; protocol analysis

- **Common protocol over non-standard port**: Exploit/exfiltration route or vulnerable service

Host-related:

- **Processor consumption**: New software/process or Denial of Service (DoS) > CPU utilization or processes using CPU runtime; spike monitoring
- **Memory consumption**: Insufficient memory allocation/memory leaks (can lead to crash) > memory consumption/processes monitoring, thresholds and alarms, periodic restarts
- **Drive capacity consumption**: Outage > real-time disk utilization monitoring, such as Systems Center Operations Manager (SCOM) or Nagios (open-source), daily reports via System Center Configuration Manager (SCCM)
- **Unauthorized software**: remediate with System Center Configuration Manager (SCCM) - central installation management/reporting, antimalware, file blacklisting/app whitelisting (limit installations)
- **Malicious process**: Compromised host > antimalware, process monitoring
- **Unauthorized change**: File creation, setting changes > logs, SIM/SIEM, File Integrity Monitoring (FIM), monitoring
- **Unauthorized privilege**: Privilege use attempts, escalation > SIM/SIEM, log + analysis
- **Data exfiltration**: Large outbound communications > anomaly detection, outbound IDS/IPS rules, Data Loss Prevention (DLP)
- **Abnormal OS process behavior**: Unusual process/command execution > attacker use of system (data exfiltration, privilege escalation, remote execution, enumeration, etc.)
- **File system change or anomaly**: New, removed, or modified files (e.g. malware) > File Integrity Monitoring (FIM)
- **Registry change or anomaly**: Malware persistence on reboot (auto-start) > utilize RegMon (registry monitoring program)
- **Unauthorized scheduled task**: Adware, persistence > remediate with Task Scheduler and/or event monitoring

Application-related:

- **Anomalous activity**: Log analysis, baseline anomaly detection, File Integrity Monitoring (FIM), user training
- **Introduction of new accounts**: Check admin account creation approvals and change management workflows, user creation logs, granted privileges tracking
- **Unexpected output**: Improper output/errors/issues > manual output validation by an admin
- **Unexpected outbound communication**: Beaconing, data exfiltration > network monitoring, outbound IDS/IPS rules, pattern-based behavior analysis
- **Service interruption**: Application or server restart, Denial of Service (DoS) > app/service status monitoring
- **Application log**: Windows app log via Systems Center Operations Manager (SCOM), check /var/log for Linux, transactional logs, and error messages

4.4 Given a scenario, utilize basic digital forensics techniques

Network:

- **Wireshark**: Graphical User Interface (GUI) tool to apply filters, reassemble streams, search captured packets
- **tcpdump**: Command Line Interface (CLI) tool for capturing and analyzing packet capture (PCAP) traffic + advanced header filtering

Endpoint:

- **Disk**: Registry, autorun keys, Master File Table (MFT), event logs, index files, change logs, volume shadow copies, user artifacts, Recycle Bin, hibernation files/memory dumps, temporary directories, app logs, removable devices
- **Memory**: Linux kernel tools like fmem and LiME (memory forensics: access to physical memory and copy data); Windows DumpIt (copy physical memory to USB) and crash dump (%SystemRoot%\MEMORY.DMP, live memory); Volatility Framework (used for extracting encryption keys, user activity/rootkit analysis)
- **Mobile**: Physical (acquire SIM card, memory cards, backups); logical (image of logical storage volumes); manual access (reviewing/recording an unlocked phone); filesystem (deleted files and existing files details, such as search histories, messages, call records, etc.)

- **Cloud**: Determine contract info regarding investigations > legal recourse with vendor > identify data and their availability > work with vendor
- **Virtualization**: Easy disk and/or memory images taken with snapshots; dead vs. live forensic analysis (somewhat comparable to static vs. dynamic code analysis)
- **Legal hold**: Either pre-emptive (voluntary) or mandatory obligation to preserve electronic data for legal investigation
- **Procedures**: Form problem statement > determine required data and their locations > document and review plan > acquire and preserve evidence > initial analysis > track actions > deeper investigation and review missing/additional data > report on findings

Hashing:

- **Changes to binaries**: Compare hashes (MD5/SHA1) to ensure integrity (chain of custody)
- **Carving**: Extract files from unallocated space via “magic numbers” (fixed byte patterns that indicate the type and format of a file); cluster-based (file start near FAT/NTFS cluster boundary), sector-based (de-clustered files), byte-based (file in file)
- **Data acquisition**: Copies all (used, slack, unallocated) spaces; dd/FTK Imager + write blocker; forensic copy devices (duplicate) > compare both hashes, maintain proper chain of custody protocols

5.0 Compliance and Assessment

5.1 Understand the importance of data privacy and protection

- **Privacy vs security**: Personal data collection/sharing vs. protecting data against illegal access

Non-technical controls:

- **Classification**: Classification schema based on risk after breach (e.g. secret, sensitive, confidential, internal use only, etc.)
- **Ownership**: Ownership of information created/used by organization; owner must protect data
- **Retention**: What information is maintained; length of time which data categories are retained for
- **Data types**: Regulatory (PII, PHI, credit cards, etc.), Intellectual Property (IP), corporate confidential information, customer data, employee data, research data, marketing data, legal data, log data and auditing trails, supplier or vendor data, financial records, etc.
- **Retention standards**: Depends on law/regulation/industry category, but there is a possibility for global compliance (depending on the data type)
- **Confidentiality**: Prevent unauthorized access/disclosure/theft of privacy information
- **Legal requirements**: Privacy Act of 1974, Family Educational Rights and Privacy Act (FERPA), Health Insurance Portability and Accountability Act (HIPAA), Payment Card Industry Data Security Standard (PCI DSS), Gramm-Leach-Bliley Act (GLBA), Sarbanes-Oxley Act of 2002 (SOX), and relevant notification laws
- **Data sovereignty**: The concept that data is subject to the laws and regulations of the country or jurisdiction where it is stored, processed, or transmitted, emphasizing the control and/or ownership of data by the respective nation
- **Data minimization**: Collected data should not be held or used unless clearly stated, as required by the General Data Protection Regulation (GDPR)
- **Purpose limitation**: Data collected for specified, legitimate, explicit purposes and not further processed in a way not compatible with the purposes, as required by the General Data Protection Regulation (GDPR)
- **Non-Disclosure Agreement (NDA)**: Legal contract that prevents sharing confidential data (such as intellectual property) with third parties

Technical controls:

- **Symmetric Encryption**: Cryptographic method where the same secret key is used for both encryption and decryption of data
- **Asymmetric Encryption**: Cryptographic method that uses a pair of unique keys: a public key for encryption and a private key for decryption, enabling secure communication and data exchange without sharing the private key
- **Data Loss Prevention (DLP)**: Detects/prevents sensitive data exfiltration; compliance; data tracking/visibility
- **Data masking**: Structurally similar but inauthentic version of data; sometimes used for testing and training
- **De-identification**: Separation of Personally Identifiable Information (PII) from Protected Health Information (PHI), for example; de-identified data can still be HIPAA non-compliant
- **Tokenization**: Data security technique that involves substituting sensitive data with randomly generated tokens, often consisting of alphanumeric characters to obfuscate the original data, but can also include punctuation (#$@<&*%., etc.)

Digital Rights Management (DRM):

- **Watermarking**: Steganographically implemented in video/audio; often used to preserve integrity, ownership, copyright, or to indicate a licensed user
- **Geographic access requirements**: Checks geolocation with system/IP address or GPS
- **Access controls**: Authentication & authorization, logging, least privilege, Multi-factor Authentication (MFA), MAC/DAC/RBAC, etc.

5.2 Given a scenario, apply security concepts in support of organizational risk mitigation

- **Business impact analysis**: Identify critical technologies/processes, prioritization, recovery time objectives, financial/operational/legal impact, requirements for recovery
- **Risk identification process**: Determine, document, and/or communicate potential risks

Risk calculation:

- **Probability**: Likelihood that a threat will successfully execute an attack, combined with the potential adverse impact or consequences resulting from it
- **Magnitude**: The adversity of the impact the risk has on the organization
- **Communication of risk factors**: Consult stakeholders; decision makers often avoid risky practice

Risk prioritization:

- **Security controls**: Prioritize upon manageability (risk control vs. risk occurrence time)
- **Engineering tradeoffs**: Risk mitigation costs vs Annual Loss Expectancy (ALE); based on risk appetite
- **Systems assessment**: Prioritize assets, identify vulnerabilities, assess controls and impact
- **Documented compensating controls**: Mitigates risk for non-compliant exceptions

Training and exercises:

- **Red Team**: Offensive attacker attempting to gain access to protected network
- **Blue Team**: Defensive security team guarding the environment to keep the red team out
- **White Team**: Coordinate/maintain/referee the wargame, and monitor results
- **Capture the Flag (CTF)**: A cybersecurity competition where participants must solve a series of challenges, often involving hacking, reverse engineering, cryptography, and other cybersecurity-related tasks.
- **Tabletop Exercises**: Discussions or role-playing exercises involving key stakeholders to simulate how they would respond to a cybersecurity incident, test incident response plans, and improve coordination.
- **Cybersecurity Drills**: Practice sessions that simulate real-world cyber incidents to test the readiness and effectiveness of incident response teams.
- **Penetration Testing (Pen Testing)**: Ethical hacking exercises conducted to identify vulnerabilities and weaknesses in an organization's systems and networks.

- **Phishing Simulations**: Controlled phishing attacks performed to train employees in recognizing and responding to phishing attempts.

Supply chain assessment:

- **Vendor due diligence**: Evaluate risks involved in partnership with potential vendor
- **Hardware source authenticity**: NSA certified, Trusted Foundry secure production via Original Equipment Manufacturers (OEMs)

5.3 Explain the importance of frameworks, policies, procedures, and controls

Frameworks:

- **Risk-based**: Focus on understanding and managing risks based on the organization's individual risk landscape
- **Prescriptive**: Provide specific guidelines and controls to achieve compliance or meet industry standards

Policies and procedures:

- **Code of conduct/ethics**: Employee accountable for own behavior; support values, principles, standards; ethical/legal decision making; restricted info disclosure
- **Acceptable Use Policy (AUP)**: Clear directions on permissible uses of resources
- **Password policy**: Password length/complexity requirements, reuse limitation
- **Data ownership**: States the ownership of the info created/used by the organization
- **Data retention:** What information is maintained and length of time which certain categories are retained for
- **Account management**: Account lifecycle (provision > active use > decommission); also referred to as onboarding and offboarding

- **Continuous monitoring**: How monitoring is performed; monitoring technology usage
- **Work product retention**: The practice of preserving and keeping documents, artifacts, or outputs generated during the course of a project or work activity

Control types:

- **Managerial**: Security assessment, planning, risk identification, evaluation of controls
- **Operational**: Practices and procedures that follow security requirements
- **Technical**: Systems, devices, software, settings, etc. which enforce requirements
- **Preventative**: Proactive measures to prevent incidents (firewalls, training, etc.)
- **Detective**: Detects and captures information on incidents (alarms, notifications, etc.)
- **Deterrent**: Discourages potential attackers or adversaries from attempting to exploit vulnerabilities or conduct malicious activities
- **Responsive**: Responds to breach and restores initial behaviors of systems (e.g. backups)
- **Corrective**: Remediates incident or limits damage (e.g. patching, antimalware, etc.)
- **Physical**: Security measures and safeguards to protect the physical assets, premises, and resources of an organization

Audits and assessments:

- **Regulatory**: PCI DSS - internal and external vulnerability scanning by professional or and Approved Scanning Vendor (ASV)
- **Compliance**: HIPAA, GLBA, SOX, FERPA, FISMA, data breach notification laws