Security + Last minute notes

Security+ Last Minute Notes

1.1 Security Controls

🔷 **Technical = Firewall & Encryption** 🔥🔑(*Blocks hackers & protects data from unauthorized access*)

🔷 **Managerial = Security Policies & Compliance Rules** 📜🔍(*Defines who gets access, password policies, and security training*)

🔷 **Operational = Log Monitoring & Incident Response** 🛠️🚨(*Tracks login attempts, detects threats, and responds to breaches*)

🔷 **Physical = Biometric Scanners & Surveillance** 🔒📷(*Controls entry to data centers and monitors security with cameras*)

Quick Memory Trick :

- **Preventive = Firewall & Antivirus** 🛡️
- **Deterrent = Account Lockout** 🚫
- **Detective = SIEM & IDS** 🔍
- **Corrective = Patching & Backups** 🔧
- **Compensating = MFA & VPN** 🔑
- **Directive = Security Policies** 📜

Quick Memory Tricks for 1.2 🚀

CIA Triad (Confidentiality, Integrity, Availability) 🛡️

🔷 **Lock** 🔒 **- Seal** 📜 **- Key** 🔑

- **Confidentiality = Lock** → Only authorized users access data (**Encryption, MFA**)
- **Integrity = Seal** → Data stays unchanged & trustworthy (**Hashing, Digital Signatures**)
- **Availability = Key** → Systems & data are always accessible (**Backups, Load Balancing**)

Non-repudiation (No Denying Actions) 📝

🔷 **"Signed & Verified"**

- Like a **receipt proving a transaction**, **digital signatures & logs** prove who did what.

AAA (Authentication, Authorization, Accounting) 🔑
🔷 **"Who You Are, What You Can Do, What You Did"**
- **Authentication = Who you are** (MFA, Passwords, Biometrics)
- **Authorization = What you can do** (RBAC, ACLs)
- **Accounting = What you did** (SIEM Logs, Auditing)

Gap Analysis (Find Weak Spots) ⚠️
🔷 **"Spot the Leaks"**
- Compare **security policies vs. actual practices** to find gaps & improve defenses.

Zero Trust Model (Never Trust, Always Verify) 🚦
🔷 **"Checkpoint Everywhere"**
- **Control Plane**: Decides security policies (**Adaptive Identity, Policy Engine**)
- **Data Plane**: Enforces them at access points (**Implicit Trust Zones, Policy Enforcement**)

Physical Security (Prevent Physical Breaches) 🚧
🔷 **"No Easy Entry"**
- **Barriers = Bollards, Fencing** 🚧 (Prevent unauthorized vehicles & people)
- **Detection = Cameras, Guards** 👀 (Monitor for threats)
- **Verification = Access Badge, Vestibule** 🔑 (Only approved people enter)
- **Sensors = Infrared, Pressure, Microwave** 📡 (Detect intrusions)

Deception & Disruption (Trap Hackers) 🎭
🔷 **"Fake Treasure Chest"**
- **Honeypot** 🏺 = Fake system to lure attackers
- **Honeynet** 🌐 = Network of honeypots
- **Honeyfile** 📂 = Fake file to detect unauthorized access
- **Honeytoken** 🔑 = Bait credentials to track intrusions

💡 **Now it's short, visual, and easy to remember!** Let me know if you want tweaks! 🚀🔥

Quick Memory Tricks for 1.3 (Change Management & Security) 🚀

Business Processes Impacting Security 🔄

🔷 **"Plan Before You Change"**

- **Approval Process** 📝 → Get permission first (**Managers sign off before big updates**)
- **Ownership** 🏢 → Who's responsible (**Who fixes issues if something breaks?**)
- **Stakeholders** 👥 → Who's affected (**Users, IT, Security Teams**)
- **Impact Analysis** ⚠️ → What could go wrong? (**Testing before deployment**)
- **Test Results** 📊 → Validate fixes (**If it works in a test, it works in production**)
- **Backout Plan** 🔙 → Rollback strategy (**Undo changes if things go wrong**)
- **Maintenance Window** 🕒 → Scheduled changes (**Update at low-traffic hours**)
- **Standard Operating Procedure (SOP)** 📖 → Follow a guide (**Step-by-step change process**)

Technical Implications ⚙️

🔷 **"Security Rules for Smooth Changes"**

- **Allow Lists / Deny Lists** ✅❌ → Control access (**Only trusted apps/users allowed**)
- **Restricted Activities** 🚫 → No risky changes (**Limit unauthorized modifications**)
- **Downtime** ⏳ → Plan disruptions (**Schedule system updates when traffic is low**)
- **Service Restart** 🔄 → Rebooting needed (**After updates, restart services securely**)
- **Application Restart** 🔁 → Reload apps (**Some updates need a fresh start**)
- **Legacy Applications** 🏚️ → Old software risk (**Outdated apps = security holes**)
- **Dependencies** 🔗 → If A breaks, B fails (**Software updates affect connected systems**)

Documentation 📜

🔷 **"Write It Down, Keep It Safe"**

- **Updating Diagrams** 🖼️ → Keep system maps current (**Know what connects to what**)
- **Updating Policies/Procedures** 📝 → Ensure rules match reality (**Security policies evolve**)

Version Control 🔄

🔷 **"Track Every Change"**

- Keep **backup versions** of software & settings (**Rollback if needed**)
- **Git & Configuration Management Tools** ensure **no changes are lost**

💡 **Quick Recap:**

- **Plan Before You Change** 📝 (Approval, Impact Analysis, Backout Plan)
- **Security Rules for Smooth Changes** 🔄 (Allow Lists, Downtime, Legacy Risks)
- **Write It Down, Keep It Safe** 📜 (Documentation, Policy Updates)
- **Track Every Change** 🔄 (Version Control, Backups)

Quick Memory Tricks for 1.4 (Cryptographic Solutions) 🚀

Public Key Infrastructure (PKI) 🔑

🔷 **"The Lock & Key System"**

- **Public Key** 🔒 → Open for everyone (**Used for encrypting messages**)
- **Private Key** 🔐 → Secret & secure (**Used for decrypting messages**)

- **Key Escrow** 📦 → Backup key storage (**A trusted third party holds copies in case of loss**)

Encryption (Data Protection) 🔐

🔷 **"Layers of Protection"**

- **Full-Disk** 🖥️ → Encrypts the entire system (**BitLocker, FileVault**)
- **Partition** 📂 → Encrypts specific sections (**Linux LUKS**)
- **File** 📄 → Encrypts individual files (**EFS on Windows**)
- **Volume** 📦 → Encrypts storage drives (**Veracrypt, LUKS**)
- **Database** 📊 → Encrypts structured data (**Transparent Data Encryption (TDE)**)
- **Record** 📜 → Encrypts specific database fields (**Credit card details encryption**)
- **Transport** 🔄 → Encrypts data in motion (**TLS, VPN, SSH**)
- **Asymmetric** 🔑 → Uses **two keys** (**SSL/TLS, RSA**)
- **Symmetric** 🔁 → Uses **one key** (**AES, DES**)
- **Key Exchange** 🔄 → Securely share keys (**Diffie-Hellman**)
- **Algorithms** 🔢 → Encryption formulas (**AES, RSA, ECC**)
- **Key Length** 🔑📏 → The longer, the stronger (**256-bit AES is stronger than 128-bit**)

Cryptographic Tools 🛠️

🔷 **"Hardware for Extra Security"**

- **Trusted Platform Module (TPM)** 🔐 → Chip for storing keys securely (**Used in BitLocker**)
- **Hardware Security Module (HSM)** 🏦 → Dedicated device for key management (**Used by banks & enterprises**)
- **Key Management System** 🗄️ → Controls key storage & access (**AWS KMS, Azure Key Vault**)
- **Secure Enclave** 🏰 → Isolated environment for sensitive operations (**Apple Secure Enclave, Intel SGX**)

Obfuscation (Hiding Data) 🎭

🔷 **"Hide & Seek for Security"**

- **Steganography** 🖼️ → Hides data in images/videos (**Secret messages inside pictures**)
- **Tokenization** 🔢 → Replaces sensitive data with tokens (**Credit card tokenization in payment systems**)
- **Data Masking** 🕵️ → Hides real data for testing (**Showing only last 4 digits of SSN: ***-1234**)

Other Cryptographic Concepts 🔄

🔷 **"Extra Security Layers"**

- **Hashing** 🔁 → One-way encryption for integrity (**SHA-256, MD5**)
- **Salting** 🧂 → Adds random values to passwords before hashing (**Prevents rainbow table attacks**)
- **Digital Signatures** ✍️ → Verifies authenticity (**Used in emails, documents, and SSL/TLS certificates**)
- **Key Stretching** 🔄 → Strengthens weak passwords (**PBKDF2, bcrypt, Argon2**)
- **Blockchain** 🔗 → Secure, decentralized ledger (**Used in cryptocurrencies & secure transactions**)
- **Open Public Ledger** 📖 → Everyone can verify transactions (**Bitcoin blockchain transparency**)

Certificates (Digital Identity Verification) 📜

🔷 **"The Online Passport"**

- **Certificate Authorities (CAs)** 🏛️ → Trusted organizations issuing digital certificates (**DigiCert, Let's Encrypt**)
- **Certificate Revocation Lists (CRLs)** 🚫 → List of revoked certificates (**If a cert is compromised**)
- **Online Certificate Status Protocol (OCSP)** 🔍 → Checks certificate validity **in real-time**

- **Self-Signed** 📜 → Made by individuals, **not trusted by browsers**
- **Third-Party** 🤝 → Issued by a trusted CA (**Used for SSL/TLS certificates**)
- **Root of Trust** 🌳 → The **highest authority** in a PKI chain (**Root CA validates all other certs**)
- **Certificate Signing Request (CSR)** ✉️ → A request to get a certificate from a CA
- ***Wildcard Certificate** → Covers **multiple subdomains** under one cert (**Example: *.example.com**)

💡 **Quick Recap:**
🔑 **PKI = Lock & Key (Public, Private, Backup Keys)**
🔐 **Encryption = Layers of Security (Disk, File, Transport, Keys)**
🛠️ **Tools = Extra Security (TPM, HSM, Secure Enclave)**
🎭 **Obfuscation = Hiding Data (Steganography, Tokenization, Masking)**
🔄 **Hashing & Salting = Secure Passwords & Integrity**
✍️ **Digital Signatures = Verify Authenticity**
🔗 **Blockchain = Secure, Decentralized Transactions**
📜 **Certificates = Online Passport (CA, CRL, OCSP, Wildcard Certs)**

2.1 Threat Actors & Motivations (Quick Notes) 🚀

🔷 Threat Actors (Who's Attacking?)

- **Nation-State** 🌍 → Government-backed cyberattacks (espionage, sabotage)
- **Unskilled Attacker** 👶 → "Script kiddie" using premade tools, no deep skills
- **Hacktivist** 🎭 → Motivated by ideology, protests online (Anonymous)
- **Insider Threat** 🏢 → Employee leaks/damages data (malicious or accidental)
- **Organized Crime** 💰 → Cyber gangs targeting businesses for money (ransomware)
- **Shadow IT** 🏴‍☠️ → Unauthorized tech used by employees (risking security)

🔷 Attributes of Attackers

- **Internal vs. External** 🔄 → Employee vs. outsider attack

- **Resources** 💰 → Well-funded (nation-state) vs. low-budget (script kiddie)
- **Skill Level** 🎯 → Low (amateurs) to high (APT – Advanced Persistent Threat)

🔷 **Motivations (Why Attack?)**

- **Data Exfiltration** 📂 → Stealing sensitive info (trade secrets, PII)
- **Espionage** 🕵️ → Spying (nation-state, corporate spying)
- **Service Disruption** 🚧 → DDoS, ransomware (cripple systems)
- **Blackmail** 🤐 → Leaking data unless paid (extortion)
- **Financial Gain** 💵 → Cybercrime for profit (credit card fraud, ransomware)
- **Philosophical/Political Beliefs** ✊ → Hacktivism (Anonymous, Wikileaks)
- **Ethical** 🧐 → White-hat hacking (penetration testing, bug bounties)
- **Revenge** 😠 → Disgruntled employee, personal vendetta
- **Disruption/Chaos** 🎭 → Just to cause damage (trolls, cyber vandals)
- **War** ⚔️ → Cyber warfare between nations

💡 **Skim-friendly Recap:**

- **Who?** Nation-state, Hacktivists, Insiders, Cybercriminals 🎭💰
- **How?** Internal vs. External, Skilled vs. Unskilled 🔄💡
- **Why?** Steal, Spy, Disrupt, Blackmail, Money, Politics, Revenge 🕵️💵🔥

2.2 Threat Vectors & Attack Surfaces (Quick Notes) 🚀

🔷 **Threat Vectors (How Attacks Happen)**

- **Message-Based** ✉️ → Phishing (Email), Smishing (SMS), Vishing (Voice), IM attacks
- **Image-Based** 🖼️ → Malicious images (steganography, exploits)
- **File-Based** 📂 → Infected PDFs, Office macros, executable malware
- **Voice Call** 📞 → Vishing (fraud calls, fake IT support)
- **Removable Device** 💾 → USB drives spreading malware
- **Vulnerable Software** 🖥️ → Outdated, unpatched apps (client vs. agentless risks)

- **Unsupported Systems** 🏚️ → Old OS, end-of-life software (Windows XP, legacy apps)

🔷 **Unsecure Networks (Exposed Entry Points)**

- **Wireless** 📡 → Open Wi-Fi, weak encryption (WEP, WPA2)
- **Wired** 🔌 → Unmonitored LAN ports, rogue devices
- **Bluetooth** 🔵 → Bluejacking, Bluesnarfing (hacking Bluetooth connections)
- **Open Ports** 🚪 → Unused but exposed services (RDP, Telnet)
- **Default Credentials** 🔑 → Factory-set passwords left unchanged

🔷 **Supply Chain Attacks** 🏭

- **MSPs & Vendors** 🤝 → Attackers exploit trusted third-party providers
- **Software Dependencies** 📦 → Injecting malicious code into legit updates

🔷 **Human-Based Attacks (Social Engineering)** 🎭

- **Phishing** 🎣 → Fake emails tricking users into clicking malicious links
- **Vishing** 📞 → Fake phone calls impersonating IT or banks
- **Smishing** 📲 → Fraudulent SMS messages (bank scams)
- **Misinformation** 📰 → Spreading false info to manipulate users
- **Impersonation** 🕵️ → Acting as a trusted person (CEO fraud, tech support scams)
- **Business Email Compromise (BEC)** 📧 → Fake invoices, wire transfer fraud
- **Pretexting** 🎭 → Creating a believable lie to trick someone into sharing data
- **Watering Hole** 🌊 → Hacking a site frequented by the target
- **Brand Impersonation** 🔍 → Fake websites mimicking real companies
- **Typosquatting** ✍️ → Using misspelled domain names (ex: g00gle.com)

💡 **Skim-friendly Recap:**

- **Message Attacks** → Phishing, Smishing, Vishing ✉️📞
- **Exploits** → Files, Images, USBs, Old Software 📂🖼️💾
- **Weak Networks** → Open Wi-Fi, Bluetooth hacks, Default Passwords 📡🔵🔑
- **Supply Chain Risks** → MSPs, Vendor Breaches 🏭📦
- **Social Engineering** → Impersonation, CEO Fraud, Typosquatting 🎭🔍

2.3 Types of Vulnerabilities (Quick Notes) 🚀

🔷 Application Vulnerabilities 🖥️

- **Memory Injection** 🧠 → Injecting malicious code into memory (heap/shellcode attacks)
- **Buffer Overflow** 💥 → Overloading memory to crash or hijack programs
- **Race Conditions** 🏁 → Exploiting timing flaws in execution
  - **TOC/TOU** ⏳ → Changing data between check & execution (time gap attacks)
- **Malicious Update** 🛠️ → Fake software updates installing malware

🔷 OS-Based Vulnerabilities 🏚️

- **Unpatched OS** 🔄 → Missing security updates = easy exploits
- **Privilege Escalation** 🔺 → Gaining admin access through weak permissions

🔷 Web-Based Vulnerabilities 🌐

- **SQL Injection (SQLi)** 💾 → Injecting SQL commands to steal or modify data
- **Cross-Site Scripting (XSS)** 📝 → Injecting malicious scripts into websites

🔷 Hardware Vulnerabilities 💻

- **Firmware Attacks** 🔌 → Exploiting low-level system software
- **End-of-Life Risks** ⚠️ → No security updates for old devices
- **Legacy Systems** 🏛️ → Unsupported tech still in use (Windows XP, old routers)

🔷 Virtualization Risks 🏗️

- **VM Escape** 🔓 → Breaking out of a virtual machine to access the host
- **Resource Reuse** 🔄 → Leaking sensitive data across VMs

🔷 Cloud-Specific Vulnerabilities ☁️

- **Misconfigured Storage** 📂 → Publicly exposed S3 buckets, Azure blobs
- **Weak API Security** 🔑 → Poorly secured cloud API endpoints

🔷 Supply Chain Attacks 🏭

- **Service Provider Risk** ⚡ → Attackers exploit third-party vendors

- **Hardware/Software Breach** 💾 → Pre-installed malware in devices or software updates

🔷 **Cryptographic Weaknesses** 🔑

- **Weak Encryption** 🔓 → Outdated algorithms (MD5, DES)
- **Key Exposure** 🔑 → Poor key management leading to leaks

🔷 **Misconfiguration Risks** ⚙️

- **Default Settings Left Unchanged** 🔧 → Factory passwords, open admin panels
- **Unrestricted Permissions** 🚀 → Overly broad access to systems & files

🔷 **Mobile Device Vulnerabilities** 📱

- **Side Loading** 🚪 → Installing apps from untrusted sources
- **Jailbreaking/Rooting** 🚨 → Removing OS security controls for full access

🔷 **Zero-Day Vulnerabilities** ⏳

- **Unknown Exploits** 🕵️ → No patch exists yet, making systems vulnerable

💡 **Skim-friendly Recap:**

- **Apps** 🖥️ → Buffer Overflow, Injection Attacks, Malicious Updates
- **OS** 🏚️ → Privilege Escalation, Unpatched Systems
- **Web** 🌐 → SQLi, XSS
- **Hardware** 💻 → Firmware, Legacy Risks
- **Virtualization** 🏗️ → VM Escape, Resource Reuse
- **Cloud** ☁️ → Misconfigured Storage, Weak APIs
- **Supply Chain** 🏭 → Vendor & Software Risks
- **Crypto** 🔑 → Weak Encryption, Key Leaks
- **Misconfiguration** ⚙️ → Default Settings, Open Access
- **Mobile** 📱 → Jailbreaking, Side Loading
- **Zero-Day** ⏳ → Exploits with No Patch

2.4 Indicators of Malicious Activity (Quick Notes) 🚀

🔷 Malware Attacks (Malicious Software) 🦠

- **Ransomware** 💰 → Encrypts files, demands payment
- **Trojan** 🎭 → Disguised as legit software, opens backdoors
- **Worm** 🪱 → Spreads across networks without user action
- **Spyware** 👀 → Secretly collects user data (keyloggers, screen captures)
- **Bloatware** 🛑 → Unwanted, resource-draining software
- **Virus** 🦠 → Self-replicating, spreads through files
- **Keylogger** ⌨️ → Records keystrokes to steal passwords
- **Logic Bomb** 💣 → Triggers at a set condition/time
- **Rootkit** 👑 → Hides deep in OS, gives attackers admin control

---

🔷 Physical Attacks 🏴‍☠️

- **Brute Force** 🔨 → Repeated password guessing
- **RFID Cloning** 📡 → Duplicates keycards/badges
- **Environmental** 🌡️ → Overheating, humidity damage

🔷 Network Attacks 🌐

- **DDoS** 🚀 → Overwhelms a system (Amplified, Reflected)
- **DNS Attacks** 🕵️ → Poisoning, hijacking domain traffic
- **Wireless Attacks** 📶 → Rogue APs, Evil Twin, Deauth attacks
- **On-Path (MITM)** 👂 → Intercepts communications
- **Credential Replay** 🔁 → Reusing stolen credentials
- **Malicious Code** 📜 → Embedded scripts in network traffic

🔷 Application Attacks 🖥️

- **Injection** 💉 → SQLi, Command Injection (malicious input)
- **Buffer Overflow** 💥 → Overloading memory to take control
- **Replay** 🔄 → Repeating valid data to gain access
- **Privilege Escalation** ⬆️ → Gaining unauthorized admin rights
- **Forgery** 📝 → Fake data requests (CSRF, email spoofing)

- **Directory Traversal** 📂 → Accessing restricted files

🔷 Cryptographic Attacks 🔑

- **Downgrade** 📉 → Forces weaker encryption
- **Collision** 🎯 → Two different inputs create the same hash
- **Birthday Attack** 🎂 → Exploits hash probability for cracks

---

🔷 Password Attacks 🔒

- **Spraying** 🎯 → Tries common passwords across many accounts
- **Brute Force** 🔨 → Rapid guessing to crack passwords

🔷 Indicators of Compromise (Red Flags) 🚨

- **Account Lockout** 🚫 → Multiple failed login attempts
- **Concurrent Session Usage** 🖥️🖥️ → Same account logged in from different places
- **Blocked Content** ⛔ → Website or files suddenly inaccessible
- **Impossible Travel** ✈️ → Login from two distant locations in a short time
- **Resource Consumption** 🔋 → Unusual CPU/memory usage
- **Resource Inaccessibility** 🔄 → Server down, restricted access
- **Out-of-Cycle Logging** 📜 → Unexpected log entries
- **Published/Documented** 📰 → Data leaks, public security warnings
- **Missing Logs** ❌ → Evidence tampering

💡 Skim-friendly Recap:

- **Malware** 🦠 → Ransomware, Trojan, Worm, Virus, Keylogger
- **Physical Attacks** 🏴‍☠️ → Brute Force, RFID Cloning
- **Network** 🌐 → DDoS, DNS Poisoning, MITM
- **Apps** 🖥️ → SQLi, Buffer Overflow, Privilege Escalation
- **Crypto** 🔑 → Downgrade, Collision, Birthday

- **Passwords** 🔒 → Spraying, Brute Force
- **Indicators** 🚨 → Lockouts, High Resource Use, Impossible Travel

2.5 Mitigation Techniques (Quick Notes) 🚀

🔷 Malware Mitigation 🦠

- **Endpoint Protection (EPP/EDR)** 🛡️ → Antivirus, behavior-based detection
- **Application Whitelisting** 📜 → Only allow trusted apps
- **Patch Management** 🔄 → Regular updates to fix vulnerabilities
- **User Training** 🎓 → Prevent phishing & social engineering

---

🔷 Physical Security Mitigation 🔒

- **Strong Authentication** 🔑 → Multi-Factor Authentication (MFA) for access
- **RFID Shielding** 🛡️ → Protects ID cards from cloning
- **Environmental Controls** 🌡️ → Cooling, fire suppression systems

---

🔷 Network Mitigation 🌐

- **DDoS Protection** 🚀 → Rate limiting, load balancing, traffic filtering
- **DNS Security** 👮 → DNS filtering, DNSSEC (prevents spoofing)
- **Wireless Security** 📡 → WPA3, MAC filtering, disable SSID broadcast
- **Zero Trust Architecture** 🚦 → Never trust, always verify

🔷 Application Security Mitigation 🖥️

- **Input Validation** 🔪 → Blocks SQLi, XSS attacks
- **Secure Coding Practices** 🛠️ → Follow OWASP guidelines
- **Least Privilege Access** 🔑 → Restrict user permissions
- **Web Application Firewall (WAF)** 🔥 → Protects against web-based attacks

---

🔷 Cryptographic Security 🔑

- **Strong Encryption** 🔪 → Use AES-256, TLS 1.3
- **Regular Key Rotation** 🔄 → Prevents key reuse attacks
- **Hashing & Salting** 🧂 → Secures stored passwords

🔷 **Password Security** 🔒

- **MFA** 🔑 → Extra authentication layer
- **Password Managers** 🗃️ → Strong, unique passwords
- **Account Lockout** 🚫 → Blocks brute force attempts

💡 **Skim-friendly Recap:**

- **Malware** 🦠 → EDR, Patching, User Training
- **Physical** 🔒 → MFA, RFID Shielding, Cooling
- **Network** 🌐 → DDoS Protection, DNS Security, Zero Trust
- **Apps** 🖥️ → Input Validation, Secure Coding, WAF
- **Crypto** 🔑 → AES-256, Key Rotation, Hashing
- **Passwords** 🔒 → MFA, Password Managers, Lockouts

3.1 Security Architecture Models (Quick Notes) 🚀

🔷 **Architecture & Infrastructure Models** 🏗️

- **Cloud** ☁️ → Shared responsibility, security varies by provider (AWS, Azure, GCP)
    - **Hybrid** 🤝 → Mix of cloud & on-premises
    - **Third-party Vendors** 🔄 → Risk from external service providers
- **Infrastructure as Code (IaC)** 📜 → Automates infrastructure (Terraform, CloudFormation)
- **Serverless** ⚡ → No traditional servers, security depends on cloud provider (AWS Lambda)
- **Microservices** 🔗 → Small, independent app components (API security is key)
- **Network Infrastructure** 🌐 →

    - **Physical Isolation (Air-gapped)** 🔌 → No external connectivity (nuclear plants)
    - **Logical Segmentation** 🔄 → VLANs, firewalls to separate traffic
    - **Software-Defined Networking (SDN)** 🖧 → Centralized network control (programmable)
- **On-Premises** 🏢 → Full control but high maintenance costs
- **Centralized vs. Decentralized** ⚖️ → One control center vs. multiple independent nodes
- **Containerization** 📦 → Isolated applications (Docker, Kubernetes)
- **Virtualization** 🖥️ → Multiple virtual machines on one physical host (VMware, Hyper-V)
- **IoT** 📡 → Networked smart devices (security concerns: weak passwords, unpatched firmware)
- **ICS/SCADA** ⚡ → Industrial control systems (power grids, water plants) with critical security needs
- **RTOS** ⏳ → Real-time OS used in medical, automotive, and robotics (low latency, high security)
- **Embedded Systems** 🔧 → Fixed-function devices (routers, smart appliances)
- **High Availability** 🔄 → Redundant systems for **zero downtime**

🔷 **Security Considerations** 🔍

- **Availability** ⏳ → Ensure systems stay online (redundancy, failover)
- **Resilience** 💪 → Ability to **withstand & recover** from attacks

---

💡 **Skim-friendly Recap:**

- **Cloud** ☁️ → Shared security, Hybrid risks, Third-party concerns
- **IaC & Serverless** 📜⚡ → Automation, but cloud-dependent security
- **Microservices & Containers** 📦 → API security, isolation risks
- **Network** 🌐 → Segmentation, SDN, Air-gapped for security
- **IoT & ICS** 📡⚡ → Unpatched firmware, high-risk industrial systems
- **RTOS & Embedded** 🔧 → Specialized systems with unique security needs
- **Availability & Resilience** 🔄💪 → Uptime & recovery-focused design

3.2 Securing Enterprise Infrastructure (Quick Notes) 🚀

🔷 Infrastructure Considerations 🏗️

- **Device Placement** 📍 → Keep critical systems in **secured areas** (data center, locked cabinets)
- **Security Zones** 🔒 → DMZ for external services, **segmented networks** for internal security
- **Attack Surface** 📉 → Reduce exposure by **disabling unnecessary services & ports**
- **Connectivity** 🌐 → Secure wired/wireless links (VPN, encryption)

🔷 Failure Modes 🔄

- **Fail-Open** 🚪 → System **remains open** during failure (risk: security bypass)
- **Fail-Closed** 🔐 → System **shuts down** when failure occurs (risk: availability issues)

🔷 Device Attributes ⚙️

- **Active vs. Passive** 📡 →
  - **Active** (inline) → Filters traffic (IPS, Firewalls)
  - **Passive** (tap/monitor) → Only **observes traffic** (IDS, Logging tools)

🔷 Network Appliances 🌐

- **Jump Server** 🚀 → Secure entry point for admin access (bastion host)
- **Proxy Server** 🔄 → Filters web traffic (anonymity, security)
- **IPS/IDS** 🛡️ → **IPS (blocks), IDS (detects)** threats
- **Load Balancer** ⚖️ → Distributes traffic for **availability**

- **Sensors** 📡 → Monitor traffic, detect anomalies

🔷 **Port Security** 🚪

- **802.1X** 🏢 → Authenticates devices before network access
- **EAP** 🔑 → Secure authentication for Wi-Fi

🔷 **Firewall Types** 🔥

- **WAF** 🌐 → Blocks web-based attacks (SQLi, XSS)
- **UTM** 🏰 → All-in-one security (Firewall, IDS, AV)
- **NGFW** 🚀 → Deep packet inspection, application-layer filtering
- **Layer 4 vs. Layer 7** 📶 →
  - **L4 (Transport)** → Blocks traffic by IP/Port
  - **L7 (Application)** → Blocks based on content (HTTP, DNS filtering)

🔷 **Secure Communication/Access** 🔐

- **VPN** 🛡️ → Encrypts remote access traffic
- **TLS/IPSec** 🔄 → **TLS for web**, **IPSec for network encryption**
- **SD-WAN** 🌍 → Secure, flexible network for remote offices
- **SASE** 🏢 → Cloud-based **security & networking**

🔷 **Selecting Effective Controls** ✅

- Use **firewalls, IPS, WAFs** to **filter traffic**
- Apply **zero-trust principles** (Verify first, trust never)
- **Segment networks** to reduce lateral movement

💡 **Skim-friendly Recap:**

- **Device Placement & Zones** 📍🔒 → Secure hardware & segment networks
- **Failure Modes** 🚪🔐 → **Fail-open (risky)**, **Fail-closed (secure but disrupts)**
- **Network Security** 🌐 → IPS, Jump Server, Load Balancer, Proxy
- **Firewalls** 🔥 → WAF (Web), UTM (All-in-One), NGFW (Advanced Filtering)

- **Communication** 🔐 → VPN, TLS/IPSec, SD-WAN, SASE

3.3 Data Protection Strategies (Quick Notes) 🚀

🔹 Data Types 🗂️

- **Regulated** 🟫 → GDPR, HIPAA (Legal requirements)
- **Trade Secret** 🔑 → Proprietary business info (Formulas, strategies)
- **Intellectual Property (IP)** 🏛️ → Copyrights, patents, trademarks
- **Financial** 💰 → Banking records, credit card info
- **Human/Non-Human Readable** 🧐 → Encrypted vs. plain text data

🔹 Data Classifications 🔄

- **Sensitive** 🔐 → Needs encryption (SSNs, medical records)
- **Confidential** 🤫 → Internal-only (Company plans)
- **Public** 🌍 → No security needed (Website content)
- **Restricted** 🚫 → Limited access (Government secrets)
- **Private** 🏠 → Personal, customer-related data
- **Critical** 🚨 → Affects operations if lost (Production databases)

🔹 Data States & Protection 🛡️

- **At Rest** 🛑 → Stored data (Encrypt with AES-256)
- **In Transit** 🚀 → Moving data (Secure with TLS, VPN)
- **In Use** 🏗️ → Being processed (Memory encryption, access controls)

🔹 Security Methods 🔄

- **Geographic Restrictions** 🌍 → Block access from certain locations
- **Encryption** 🔑 → Protects data in all states (AES, RSA, ECC)
- **Hashing** #️⃣ → Ensures data integrity (SHA-256)
- **Masking** 🎭 → Hides sensitive info (e.g., credit card ****1234)
- **Tokenization** 🔄 → Replaces real data with tokens

- **Obfuscation** 🎭 → Scrambles data to make it unreadable
- **Segmentation** 🔀 → Separate networks for different data types
- **Permission Restrictions** 🔒 → Least privilege, role-based access

---

💡 **Skim-friendly Recap:**

- **Data Types** 📂 → Regulated, IP, Financial, Confidential
- **Classification** 🔐 → Public, Private, Restricted, Critical
- **States** 🔄 → **At Rest (AES), In Transit (TLS), In Use (Access Control)**
- **Protection** 🛡️ → **Encrypt, Hash, Mask, Tokenize, Restrict**

3.4 Resilience & Recovery in Security Architecture (Quick Notes) 🚀

🔷 **High Availability (HA)** 🔄

- **Load Balancing** ⚖️ → Distributes traffic to prevent server overload
- **Clustering** 🔗 → Multiple systems work as one (failover protection)

🔷 **Site Considerations** 🏢

- **Hot Site** 🔥 → Fully operational backup (minimal downtime)
- **Warm Site** ♨️ → Partial backup (needs some setup)
- **Cold Site** ❄️ → Just space & power, requires full setup
- **Geographic Dispersion** 🌍 → Backups in different locations for disaster recovery

🔷 **Platform Diversity** 🖥️

- **Multi-Cloud** ☁️ → AWS + Azure + GCP for redundancy
- **Hybrid Systems** 🔄 → Combining on-prem & cloud for flexibility

🔷 **Continuity of Operations** 🔄

- **Disaster Recovery Plan (DRP)** 📜 → Step-by-step plan for recovery
- **Business Continuity Planning (BCP)** 🏢 → Keeping operations running after a disaster

🔷 **Capacity Planning** 📏

- **People** 👥 → Staff availability for emergencies
- **Technology** 💻 → Sufficient resources (servers, storage)
- **Infrastructure** 🏗️ → Power, cooling, networking

🔷 **Testing & Validation** ✅

- **Tabletop Exercises** 📝 → Simulated discussions on disaster response
- **Failover Testing** 🔄 → Switching systems to test redundancy
- **Simulation** 🎭 → Running disaster drills to check preparedness

🔷 **Backup Strategies** 💾

- **Onsite vs. Offsite** 🔄 → Local backups vs. cloud storage
- **Backup Frequency** ⏳ → Daily, Weekly, Incremental, Differential
- **Encryption** 🔐 → Protect backups from theft
- **Snapshots** 📸 → Instant state copies for fast recovery
- **Replication** 🏃 → Continuous data syncing to another system
- **Journaling** 📜 → Keeps logs of all transactions for rollback

🔷 **Power Management** ⚡

- **Generators** 🏭 → Backup power for long outages
- **Uninterruptible Power Supply (UPS)** 🔋 → Short-term power for safe shutdowns

---

💡 **Skim-friendly Recap:**

- **High Availability** 🔄 → Load Balancing, Clustering
- **Sites** 🏢 → Hot (Fast), Warm (Partial), Cold (Setup Needed)
- **Diversity** 🖥️ → Multi-Cloud, Hybrid
- **Continuity Planning** 📜 → DRP, BCP, Capacity Planning
- **Testing** ✅ → Tabletop, Failover, Simulation
- **Backups** 💾 → Onsite/Offsite, Encryption, Snapshots, Replication

- **Power** ⚡ → Generators, UPS

4.1 Security Techniques for Computing Resources (Quick Notes) 🚀

🔷 Secure Baselines 🏗️

- **Establish** 📜 → Define security settings (OS hardening, configurations)
- **Deploy** 🚀 → Apply baselines to systems (servers, workstations, cloud)
- **Maintain** 🔄 → Regular updates & security checks

🔷 Hardening Targets 🔐

- **Mobile Devices** 📱 → Encrypt, enable remote wipe, disable USB transfer
- **Workstations** 💻 → Disable unnecessary services, enforce MFA
- **Switches/Routers** 🌐 → Secure SSH access, disable unused ports
- **Cloud** ☁️ → IAM roles, encryption, security groups
- **Servers** 🖥️ → Patching, least privilege, log monitoring
- **ICS/SCADA** ⚡ → Air-gapped networks, firewall rules
- **Embedded/IoT** 📡 → Change default credentials, update firmware
- **RTOS** ⏳ → Real-time security policies (medical, automotive)

🔷 Wireless Devices 📶

- **Installation Considerations** 🏢 →
  - **Site Surveys** 🔍 → Optimize Wi-Fi placement
  - **Heat Maps** 🔥 → Detect weak signal areas

🔷 Mobile Security 📲

- **MDM (Mobile Device Management)** 🛡️ → Enforce security policies remotely
- **Deployment Models** 🚀 →

    - **BYOD** → Employee's device (high risk)
    - **COPE** → Company-owned, personal use allowed
    - **CYOD** → Employee chooses from approved devices
- **Connection Methods** 🔗 → Secure **Wi-Fi, Cellular, Bluetooth**

🔷 Wireless Security Settings 🔒

- **WPA3** 🏆 → Strongest Wi-Fi encryption
- **RADIUS** 🔑 → Centralized authentication
- **Cryptographic Protocols** 🔐 → TLS, AES for secure connections
- **Authentication Protocols** 🎟️ → EAP, PEAP, 802.1X

🔷 Application Security 🛠️

- **Input Validation** 💉 → Prevents SQLi, XSS
- **Secure Cookies** 🍪 → Stops session hijacking
- **Static Code Analysis** 📜 → Detects vulnerabilities before deployment
- **Code Signing** ✍️ → Ensures software authenticity

🔷 Additional Security Measures 🏗️

- **Sandboxing** 🏖️ → Isolates untrusted applications
- **Monitoring** 📊 → Logs, SIEM, anomaly detection

💡 Skim-friendly Recap:

- **Baselines** 🏗️ → Define, Deploy, Maintain
- **Hardening** 🔐 → Mobile, Workstations, Cloud, ICS, IoT
- **Wireless** 📶 → WPA3, RADIUS, Site Surveys
- **Mobile Security** 📲 → MDM, BYOD, COPE, CYOD
- **Apps** 🛠️ → Input Validation, Secure Cookies, Code Signing
- **Extras** 🏗️ → Sandboxing, Monitoring

4.2 Hardware, Software, and Data Asset Management (Quick Notes) 🚀

🔷 Acquisition & Procurement 📦

- **Vet Vendors** 🕵️ → Ensure trusted suppliers (no pre-installed malware)
- **Security Requirements** 🔒 → Compliance (ISO, NIST, GDPR) before purchase

---

🔷 Assignment & Accounting 📜

- **Ownership** 👤 → Track device responsibility
- **Classification** 🔍 → Label as **Public, Private, Confidential, Restricted**

🔷 Monitoring & Asset Tracking 🔄

- **Inventory Management** 📋 → Track all hardware/software
- **Enumeration** 🔢 → Identifying and logging active devices

🔷 Disposal & Decommissioning 🚮

- **Sanitization** 🧹 → Secure wiping before disposal
- **Destruction** 💥 → Physical shredding of storage devices
- **Certification** 📜 → Proof of proper disposal
- **Data Retention** 🗄️ → Define policies for how long data is stored

---

💡 Skim-friendly Recap:

- **Procurement** 📦 → Secure vendors & compliance
- **Tracking** 🔍 → Ownership, classification, inventory
- **Disposal** 🚮 → Wipe, Destroy, Certify

4.3 Vulnerability Management (Quick Notes) 🚀

🔷 Identification Methods 🔍

- **Vulnerability Scans** 🖥️ → Detects outdated software & weak settings
- **Application Security Testing** 🛠️ →
  - **Static Analysis** 📜 → Checks source code for flaws
  - **Dynamic Analysis** 🔄 → Tests running apps for security holes
  - **Package Monitoring** 📦 → Tracks third-party dependencies
- **Threat Feeds & Intelligence** 🕵️ →
  - **OSINT** 📰 → Publicly available data (threat reports)
  - **Proprietary Threat Feeds** 🔐 → Paid services with latest threats
  - **Info-Sharing Groups** 🤝 → ISACs, government alerts
- **Penetration Testing** 🎯 → Simulated real-world attacks
- **Bug Bounty Programs** 💰 → Rewards for ethical hackers
- **Audits & System Reviews** 📊 → Ensuring policy compliance

🔷 Vulnerability Analysis 📉

- **False Positives** 🚫 → Alerts that aren't real threats
- **False Negatives** ⚠️ → Missed threats
- **Prioritization** 🎯 → High-risk vulnerabilities get patched first
- **Scoring Systems** 📊 →
  - **CVSS** 🏆 → Rates vulnerability severity (Critical, High, Medium, Low)
  - **CVE** 📂 → Catalogs known vulnerabilities

🔷 Response & Remediation 🔄

- **Patching** 🔧 → Fix vulnerabilities with software updates
- **Segmentation** 🔀 → Isolate vulnerable systems
- **Compensating Controls** 🛡️ → Extra security while awaiting patches
- **Exception Handling** ⚠️ → When patches can't be applied

🔷 **Validation of Fixes** ✅

- **Rescanning** 🔍 → Ensure patching was successful
- **Audits & Reports** 📋 → Verify security improvements

💡 **Skim-friendly Recap:**

- **Identify** 🔍 → Scans, Pen Testing, Bug Bounties
- **Analyze** 📉 → False Positives, Prioritization (CVSS, CVE)
- **Respond** 🔧 → Patch, Segment, Apply Controls
- **Validate** ✅ → Rescan, Audit

4.4 Security Alerting & Monitoring (Quick Notes) 🚀

🔷 **Monitoring Computing Resources** 📊

- **Systems** 🖥️ → Servers, endpoints
- **Applications** 📦 → Web, databases
- **Infrastructure** 🌐 → Network, cloud, IoT

🔷 **Security Activities** 🔄

- **Log Aggregation** 📂 → Centralized logging (SIEM)
- **Alerting** 🚨 → Notifications on suspicious activity
- **Scanning** 🔍 → Continuous vulnerability scans
- **Reporting** 📋 → Security dashboards & insights
- **Archiving** 🏛️ → Store logs for forensic analysis
- **Incident Response** 🚑 →
    - **Quarantine** 🛑 → Isolate infected machines
    - **Alert Tuning** 🔧 → Reduce false positives

🔷 **Monitoring Tools** 🛠️

- **SCAP** 📜 → Automates security benchmarks
- **Agents vs. Agentless** 🔍 → On-device monitoring vs. network scans
- **SIEM** 📊 → Logs & correlates security events (Splunk, Microsoft Sentinel)
- **Antivirus** 🛡️ → Detects known malware
- **DLP (Data Loss Prevention)** 🔐 → Stops unauthorized data transfers
- **SNMP Traps** 📡 → Detects network changes
- **NetFlow** 🌐 → Monitors traffic patterns
- **Vulnerability Scanners** 🔎 → Identifies weak spots

💡 **Skim-friendly Recap:**

- **What's Monitored?** 📊 → Systems, Apps, Network
- **Security Activities** 🔄 → Logging, Alerts, Scanning, Reports
- **Monitoring Tools** 🛠️ → SIEM, Antivirus, DLP, NetFlow

4.5 Identity & Access Management (IAM) (Quick Notes) 🚀

🔷 **Firewalls & Access Control** 🚪

- **Firewall Rules** 📜 → Allow/block traffic based on policies
- **Access Lists (ACL)** 🛑 → Restrict access to specific IPs/ports
- **Ports/Protocols** 🌐 → Secure unused ports (disable Telnet, open SSH)
- **Screened Subnets** 🏢 → DMZ for public-facing services

🔷 Intrusion Detection & Prevention 🛡️

- **IDS (Detects)** 🔍 → Monitors & alerts but doesn't block
- **IPS (Prevents)** 🚨 → Blocks suspicious activity automatically
- **Signature-Based IDS** 📝 → Detects known threats
- **Behavior-Based IDS** 🤖 → Detects anomalies (zero-day attacks)

🔷 Web & Email Security 📧

- **Web Filtering** 🌍 → Blocks malicious sites
- **Centralized Proxy** 🔄 → Controls internet access
- **URL Scanning** 🔎 → Checks links before opening
- **Content Categorization** 📂 → Filters based on topic (social media, gambling)

🔷 Operating System Security 🖥️

- **Group Policy** 🎛️ → Enforce security settings in Windows
- **SELinux & AppArmor** 🔐 → Linux access controls

🔷 Secure Protocols 🔄

- **Protocol Selection** 📜 → Use SSH, SFTP over Telnet, FTP
- **Port Selection** 🚪 → Close unused ports
- **Transport Methods** 🚀 → TLS, IPSec for encrypted traffic

🔷 Advanced Security Tools 🛠️

- **DNS Filtering** 🛑 → Prevents access to malicious domains
- **Email Security** 📧 →
  - **DMARC** 📜 → Protects against spoofing
  - **DKIM** 🔑 → Email signing for authenticity
  - **SPF** ✅ → Verifies sender identity
- **File Integrity Monitoring (FIM)** 🔍 → Detects unauthorized file changes

- **DLP (Data Loss Prevention)** 🔐 → Blocks data leaks
- **NAC (Network Access Control)** 🌍 → Ensures only authorized devices connect
- **EDR/XDR** 🛡️ → Endpoint/Extended Detection & Response
- **User Behavior Analytics (UBA)** 👮 → Detects unusual user actions

---

💡 **Skim-friendly Recap:**

- **Firewall Rules** 🔥 → ACLs, Screened Subnets
- **IDS/IPS** 🛡️ → Detects & Blocks Threats
- **Web & Email Security** 📧 → Filtering, URL Scanning
- **OS Security** 🖥️ → Group Policy, SELinux
- **Secure Protocols** 🔄 → TLS, IPSec, SSH
- **Advanced Security Tools** 🛠️ → DNS Filtering, DLP, NAC, EDR

4.6 IAM & Access Controls (Quick Notes) 🚀

🔷 User Management & Authentication 🔑

- **Provisioning & De-Provisioning** 🔄 → Adding/removing users securely
- **Permissions** 🏷️ → Role-based access control (RBAC)
- **Identity Proofing** 🆔 → Verifying users before granting access
- **Federation** 🌍 → Single sign-on (SSO) across multiple organizations
- **SSO (Single Sign-On)** 🔗 → One login for multiple services

---

🔷 Access Control Models 🏗️

- **Mandatory Access Control (MAC)** 🚔 → Strict rules (Government systems)
- **Discretionary Access Control (DAC)** 🎛️ → Owners decide permissions
- **Role-Based Access Control (RBAC)** 👥 → Permissions based on job roles
- **Rule-Based Access Control** ⚖️ → Dynamic rules (Time-based access)
- **Attribute-Based Access Control (ABAC)** 🏷️ → Access based on attributes (device, location)
- **Time-Based Restrictions** ⏳ → Blocks access outside working hours
- **Least Privilege Principle** 🔐 → Only allow **minimum** access needed

---

🔷 Multi-Factor Authentication (MFA) 🔑

- Factors: Something You...
  - Know (Password) 🔤
  - Have (Smart Card, OTP) 📱
  - Are (Biometrics) 🧬
  - Somewhere You Are (Geo-Location) 📍

🔷 Password Security 🔒

- **Best Practices** 🏆 → Length > Complexity
- **Password Managers** 🗂️ → Generate & store securely
- **Passwordless Authentication** 🚀 → Biometrics, Security Keys

---

🔷 Privileged Access Management (PAM) 🛡️

- **Just-in-Time Permissions** ⏳ → Grant access only when needed
- **Password Vaulting** 🔐 → Secure storage for admin credentials
- **Ephemeral Credentials** 🤖 → Temporary access keys

💡 Skim-friendly Recap:

- **User Management** 🔄 → Provisioning, SSO, Federation
- **Access Models** 🏗️ → MAC, DAC, RBAC, ABAC
- **MFA** 🔑 → Password, OTP, Biometrics, Geo-Location
- **Password Security** 🔒 → Managers, Passwordless Auth
- **PAM** 🛡️ → Just-in-Time, Vaulting, Ephemeral Keys

4.7 Incident Response Process (Quick Notes) 🚀

🔷 Incident Response Steps 🔄

1. **Preparation** 🏗️ → Policies, training, tools in place

2. **Detection & Analysis** 🔍 → Identify the incident, assess impact
3. **Containment** 🚧 → Stop the spread (Network isolation, account lockout)
4. **Eradication** 🛠️ → Remove the threat (Patching, malware removal)
5. **Recovery** 🔄 → Restore systems, monitor for re-infection
6. **Lessons Learned** 📜 → Improve processes, update security measures

---

🔷 **Roles & Responsibilities** 👥

- **Incident Response Team (IRT)** 🚑 → Handles incidents
- **Management** 📊 → Approves actions
- **Legal & Compliance** ⚖️ → Ensures reporting requirements
- **PR/Communications** 🗣️ → Manages public disclosures

🔷 **Incident Types** 🚨

- **Malware** 🦠 → Virus, Ransomware
- **Unauthorized Access** 🔑 → Hacking, Credential Theft
- **Insider Threat** 🏢 → Employee misuse
- **Data Breach** 📂 → PII, IP theft
- **Denial-of-Service (DoS/DDoS)** 🚧 → Service disruption

💡 **Skim-friendly Recap:**

- **Steps** 🔄 → Prepare, Detect, Contain, Eradicate, Recover, Review
- **Roles** 👥 → IRT, Management, Legal, PR
- **Incident Types** 🚨 → Malware, Unauthorized Access, Insider Threats

🔥 **Quick & structured for fast review!** 🚀

4.8 Digital Forensics (Quick Notes) 🚀

🔷 **Forensic Process** 🔍

1. **Identification** 👮 → What data is relevant?
2. **Collection** 📩 → Securely gather evidence
3. **Preservation** 🔐 → Maintain chain of custody
4. **Examination** 🔎 → Analyze logs, files, and metadata
5. **Analysis** 📊 → Correlate findings
6. **Reporting** 📜 → Document conclusions

🔷 Forensic Artifacts 🔄

- **Memory Dumps** 💾 → Captures live RAM data
- **Disk Imaging** 💿 → Exact copy of storage
- **Log Analysis** 📂 → SIEM logs, network traffic
- **Metadata** 🏷️ → Timestamps, file access history

🔷 Integrity & Legal Considerations ⚖️

- **Chain of Custody** 🔗 → Document every evidence transfer
- **Hashing** 🧱 → Verify file integrity (SHA-256, MD5)
- **Legal Compliance** 📜 → GDPR, HIPAA, PCI-DSS regulations

💡 Skim-friendly Recap:

- **Process** 🔍 → Identify, Collect, Preserve, Analyze, Report
- **Artifacts** 📂 → Memory, Disk, Logs, Metadata
- **Legal** ⚖️ → Chain of Custody, Hashing, Compliance

🔥 **Concise & easy to memorize!** 🚀

4.9 Security Governance & Risk (Quick Notes) 🚀

🔷 Security Policies 📜

- **Acceptable Use Policy (AUP)** ✅ → Defines proper system use

- **Data Classification Policy** 🗂️ → Labels sensitive data
- **Access Control Policy** 🔐 → Who can access what?
- **Incident Response Policy** 🚨 → How to handle security breaches
- **Disaster Recovery (DRP)** 🏢 → Ensures business continuity

---

🔷 Risk Management 🎯

- **Risk Assessment** ⚠️ → Identify threats & vulnerabilities
- **Risk Mitigation** 🛡️ → Controls to reduce risk
- **Risk Acceptance** 🤷 → Live with it if low impact
- **Risk Transfer** 📜 → Buy cyber insurance
- **Risk Avoidance** ❌ → Eliminate risky activities

🔷 Compliance & Frameworks 🏛️

- **ISO 27001** 🔐 → Global security standard
- **NIST** 📖 → US security guidelines
- **GDPR** 🇪🇺 → EU data protection law
- **HIPAA** 🏥 → US healthcare security law
- **PCI-DSS** 💳 → Payment card security standard

💡 Skim-friendly Recap:

- **Policies** 📜 → AUP, Data Classification, DRP
- **Risk** 🎯 → Assess, Mitigate, Accept, Transfer, Avoid
- **Compliance** 🏛️ → ISO, NIST, GDPR, HIPAA, PCI-DSS

5.1 Effective Security Governance (Quick Notes) 🚀

🔷 Governance Framework 📜

- **Guidelines** 📖 → General best practices (Not strict rules)
- **Policies** 📜 → High-level rules (What to do, not how)
  - **AUP** ✅ → Defines acceptable system use
  - **InfoSec Policies** 🔐 → Data protection guidelines
  - **BCP/DRP** 🏢 → Plans for keeping business running after incidents
  - **Incident Response** 🚨 → Steps to handle breaches
  - **SDLC** 🛠️ → Secure software development
  - **Change Management** 🔄 → Controls for system modifications

---

🔷 Security Standards 🏛️

- **Password Rules** 🔑 → Complexity, expiration
- **Access Control** 🚪 → Least privilege, role-based access
- **Physical Security** 🔒 → Badges, cameras, guards
- **Encryption** 🔐 → Data protection rules (AES, TLS)

🔷 Security Procedures 🛠️

- **Change Management** 🔄 → Review, approve, implement changes
- **Onboarding/Offboarding** 👥 → Grant/remove system access
- **Playbooks** 📂 → Step-by-step guides for security events

🔷 External Considerations 🌍

- **Regulatory** 📜 → GDPR, HIPAA, PCI-DSS compliance
- **Legal** ⚖️ → Laws governing data protection
- **Industry Standards** 🏢 → NIST, ISO 27001
- **Geographic Rules** 🌍 → Local, national, global security laws

---

🔷 **Governance Structures** 🏢

- **Boards** 🏛️ → High-level decision-making (CISOs, execs)
- **Committees** 🤝 → Security working groups
- **Government Entities** 🏛️ → Regulatory bodies
- **Centralized vs. Decentralized** ⚖️ → One authority vs. distributed control

---

🔷 **Roles & Responsibilities** 👥

- **Owners** 🔑 → Decide how data is used
- **Controllers** 🛠️ → Define security measures
- **Processors** ⚙️ → Handle data on behalf of owners
- **Custodians/Stewards** 📂 → Maintain and protect data

💡 **Skim-friendly Recap:**

- **Framework** 📜 → Policies, Standards, Procedures
- **Compliance** 🌍 → Regulatory, Legal, Industry, Geographic
- **Governance** 🏢 → Boards, Committees, Centralized/Decentralized
- **Roles** 👥 → Owners, Controllers, Processors, Custodians

5.2 Risk Management Process (Quick Notes) 🚀

🔷 **Risk Identification & Assessment** 🔍

- **Risk Identification** ⚠️ → Find potential threats & vulnerabilities
- **Risk Assessment** 📊 → How often do we assess?
  - **Ad Hoc** 🛠️ → One-time, no set schedule
  - **Recurring** 🔄 → Periodic reviews (monthly, yearly)
  - **One-Time** 🕒 → Major project-specific assessments
  - **Continuous** 🔍 → Ongoing monitoring for real-time risks

---

🔷 Risk Analysis 📉

- **Qualitative** 📜 → Subjective (high, medium, low risk)
- **Quantitative** 💰 → Uses **numbers & financial impact**
- **Single Loss Expectancy (SLE)** 💵 → Cost of **one** incident
- **Annualized Rate of Occurrence (ARO)** 🔄 → **How often** it happens per year
- **Annualized Loss Expectancy (ALE)** 🏦 → **SLE × ARO** (Yearly financial impact)
- **Probability** 🎲 → How likely is the risk?
- **Likelihood** 📉 → Frequency of occurrence (low, medium, high)
- **Exposure Factor (EF)** ⚡ → % of asset loss per incident
- **Impact** 🚨 → How severe is the damage? (Financial, reputational, operational)

🔷 Risk Register 📋

- **Key Risk Indicators (KRIs)** 🚦 → Warning signs of potential threats
- **Risk Owners** 👥 → Assigned person/team responsible for managing the risk
- **Risk Threshold** 🎯 → Maximum risk level before action is taken

---

🔷 Risk Tolerance vs. Risk Appetite ⚖️

- **Risk Tolerance** 🎛️ → How much risk can we handle per project?
- **Risk Appetite** 🍽️ → Overall company approach to risk:
  - **Expansionary** 📈 → Willing to take more risks
  - **Conservative** 🛑 → Avoids risk as much as possible
  - **Neutral** ⚖️ → Balanced approach

🔷 Risk Management Strategies 🚀

- **Transfer** 📜 → Buy insurance, outsource risk to third parties
- **Accept** 🤷 → Decide to live with the risk
  - **Exemption** 🚫 → Formal approval to operate despite risk
  - **Exception** ❗ → Temporary bypass of risk controls
- **Avoid** ❌ → Stop risky activities altogether
- **Mitigate** 🛡️ → Reduce risk impact with security controls (firewalls, encryption)

🔷 **Risk Reporting & Business Impact Analysis (BIA)** 📊

- **Risk Reporting** 📜 → Document & communicate risk status
- **Business Impact Analysis (BIA)** 🏢 → Evaluates how risks impact business operations
    - **Recovery Time Objective (RTO)** ⏳ → Max downtime allowed before recovery
    - **Recovery Point Objective (RPO)** 🔄 → Max data loss allowed (how often backups are needed)
    - **Mean Time to Repair (MTTR)** 🔧 → Avg time to **fix a failure**
    - **Mean Time Between Failures (MTBF)** 🔄 → Avg **time between system failures**

💡 **Skim-friendly Recap:**

- **Assessment** 🔍 → Ad Hoc, Recurring, One-Time, Continuous
- **Analysis** 📉 → Qualitative (subjective) vs. Quantitative (numbers)
- **Risk Factors** 🎲 → SLE, ALE, ARO, EF, Impact
- **Register** 📋 → KRIs, Risk Owners, Thresholds
- **Risk Tolerance** ⚖️ → Expansionary, Conservative, Neutral
- **Strategies** 🚀 → Transfer, Accept, Avoid, Mitigate
- **BIA** 📊 → RTO (Downtime), RPO (Data Loss), MTTR (Fix Time), MTBF (Failure Gaps)

5.3 Third-Party Risk Assessment & Management (Quick Notes) 🚀

🔷 **Vendor Assessment** 🔍

- **Penetration Testing** 🎯 → Test vendor security by simulating attacks
- **Right-to-Audit Clause** 📜 → Allows company to inspect vendor security practices
- **Internal Audit Evidence** 📋 → Vendor must show proof of security assessments
- **Independent Assessments** 👨‍⚖️ → Third-party security evaluations
- **Supply Chain Analysis** 📦 → Ensuring all vendors & suppliers follow security standards

---

🔷 Vendor Selection 🤝

- **Due Diligence** 🏛️ → Background checks, financial stability, security policies
- **Conflict of Interest** ⚖️ → Ensure vendor does not have competing interests

---

🔷 Agreement Types 📜

- **Service-Level Agreement (SLA)** 📏 → Defines service expectations (uptime, performance)
- **Memorandum of Agreement (MOA)** ✍️ → Formal contract with legal obligations
- **Memorandum of Understanding (MOU)** 🤝 → Non-binding agreement (partnership details)
- **Master Service Agreement (MSA)** 📑 → Long-term contract covering multiple projects
- **Statement of Work (SOW)** 📋 → Details specific work to be done under an agreement
- **Non-Disclosure Agreement (NDA)** 🔒 → Prevents sharing of sensitive information
- **Business Partners Agreement (BPA)** 💼 → Defines responsibilities between business partners

---

🔷 Vendor Monitoring 🔄

- **Ongoing Evaluations** 🔍 → Regular security reviews & compliance checks
- **Security Questionnaires** 📄 → Vendors answer security-related questions to assess risk
- **Rules of Engagement** 🚦 → Defines how security testing & audits are conducted

---

💡 Skim-friendly Recap:

- **Vendor Security** 🔍 → Pen Testing, Audits, Supply Chain Analysis
- **Selection** 🤝 → Due Diligence, Conflict of Interest

- **Agreements** 📜 → SLA (Performance), NDA (Confidentiality), MSA/SOW (Contracts)
- **Monitoring** 🔄 → Questionnaires, Ongoing Audits, Rules of Engagement

5.4 Security Compliance (Quick Notes) 🚀

🔷 Compliance Reporting 📜

- **Internal** 📊 → Reports for internal audits, risk teams, and executives
- **External** 🌍 → Regulatory filings for **GDPR, HIPAA, PCI-DSS, ISO 27001**

🔷 Consequences of Non-Compliance ⚠️

- **Fines** 💰 → Heavy financial penalties (GDPR fines up to 4% of revenue)
- **Sanctions** 🚫 → Restrictions imposed by regulators
- **Reputational Damage** 📰 → Loss of customer trust
- **Loss of License** 📉 → Revoked business certifications (PCI-DSS, financial services)
- **Contractual Impacts** 🤝 → Breach of service agreements = lawsuits

---

🔷 Compliance Monitoring 🔍

- **Due Diligence/Care** ⚖️ → Ensuring security compliance at all levels
- **Attestation & Acknowledgment** 📝 → Employees/vendors confirm adherence
- **Internal & External Audits** 📋 → Continuous security checks
- **Automation** 🤖 → Compliance tools (SIEM, GRC platforms) for real-time monitoring

---

🔷 Privacy & Legal Compliance 🔐

- **Legal Implications** ⚖️ →
  - **Local/Regional** 🏛️ → State/country-specific laws (CCPA, GDPR)

    - **National** 🌎 → Federal laws (HIPAA, SOX)
    - **Global** 🌍 → International data laws (GDPR for EU, Data Sovereignty)
- **Data Subject** 👤 → Person whose data is collected
- **Controller vs. Processor** 🔄 →
    - **Controller** → Owns & decides how data is used
    - **Processor** → Processes data on behalf of the controller
- **Ownership** 🔑 → Who controls, stores, and secures the data
- **Data Inventory & Retention** 📦 → Tracking where data is stored & retention policies
- **Right to be Forgotten** 🛑 → Individuals can request data deletion under GDPR

💡 **Skim-friendly Recap:**

- **Reporting** 📜 → Internal (Audits) vs. External (Regulators)
- **Non-Compliance Risks** ⚠️ → Fines, Sanctions, Reputation Loss
- **Monitoring** 🔍 → Audits, Automation, Due Diligence
- **Privacy & Legal** 🔐 → GDPR, Controllers, Processors, Right to be Forgotten

5.5 Audits & Assessments (Quick Notes) 🚀

🔹 Attestation 📜

- **Formal certification** that security controls are in place & working

---

🔹 Internal Audits 🔍

- **Compliance** 🏢 → Ensures company follows policies & regulations
- **Audit Committee** 📋 → Oversees internal security reviews
- **Self-Assessments** ✅ → Internal teams evaluate their own security

---

🔹 External Audits 🌍

- **Regulatory** ⚖️ → Compliance checks by government agencies (GDPR, HIPAA)
- **Examinations** 🏛️ → In-depth security reviews (banks, healthcare)
- **Assessments** 📊 → Security posture evaluations by external firms
- **Independent Third-Party Audits** 👨‍💼 → Performed by external security auditors (SOC 2, ISO 27001)

🔷 **Penetration Testing** 🎯

- **Physical** 🏢 → Testing locks, security cameras, badge access
- **Offensive** 🚀 → Simulating hacker attacks to find vulnerabilities
- **Defensive** 🛡️ → Blue team responding to attacks
- **Integrated** 🔄 → Combines offensive (red team) & defensive (blue team) strategies

🔷 **Testing Environments** 🏗️

- **Known Environment** 👨‍💻 → Tester has full system knowledge
- **Partially Known Environment** 🕵️ → Tester has **some** system knowledge
- **Unknown Environment** 🎭 → Blind test (black-box testing)

🔷 **Reconnaissance (Info Gathering)** 🔎

- **Passive** 📖 → Observing publicly available data (OSINT, social media)
- **Active** 🎯 → Actively probing systems (network scans, phishing)

💡 **Skim-friendly Recap:**

- **Audits** 📜 → Internal (Self, Compliance), External (Regulatory, Third-Party)
- **Pen Testing** 🎯 → Physical, Offensive, Defensive, Integrated
- **Environments** 🏗️ → Known, Partially Known, Unknown
- **Recon** 🔍 → Passive (OSINT), Active (Network Scans)

5.6 Security Awareness Practices (Quick Notes) 🚀

🔷 Phishing Awareness 📧

- **Campaigns** 🎯 → Simulated phishing tests to educate employees
- **Recognizing Phishing** 🔍 → Suspicious links, urgent messages, unknown senders
- **Response to Phishing** 🚨 → Report, quarantine, block sender

---

🔷 Anomalous Behavior Recognition 🔄

- **Risky** ⚠️ → Login from unusual locations, mass file downloads
- **Unexpected** 🎭 → Employees accessing data outside their role
- **Unintentional** 🛑 → Accidental data sharing, weak passwords

---

🔷 User Training & Best Practices 🎓

- **Policies & Handbooks** 📜 → Documented security guidelines
- **Situational Awareness** 👀 → Recognizing threats in daily activities
- **Insider Threats** 🏢 → Employees misusing data (malicious or accidental)
- **Password Security** 🔑 → MFA, unique passwords, password managers
- **Removable Media & Cables** 💾 → Avoid USBs from unknown sources
- **Social Engineering** 🎭 → Phone scams, pretexting, tailgating prevention
- **Operational Security (OPSEC)** 🔐 → Protecting company secrets (avoid oversharing)
- **Hybrid/Remote Work** 🏠 → VPN use, secure Wi-Fi, webcam cover

---

🔷 Reporting & Monitoring 📊

- **Initial Reports** 📝 → New security concerns reported immediately
- **Recurring Monitoring** 🔄 → Regular audits of security behavior

---

🔷 Awareness Development & Execution 🚀

- **Development** 🏗️ → Build training programs based on threats
- **Execution** 🎯 → Workshops, phishing tests, security drills

---

💡 **Skim-friendly Recap:**

- **Phishing** 📧 → Simulated campaigns, Recognizing, Reporting
- **Behavior Recognition** 🔍 → Risky, Unexpected, Unintentional actions
- **User Training** 🎓 → Passwords, Social Engineering, OPSEC, Remote Work
- **Monitoring & Reporting** 📊 → Initial & Recurring logs
- **Execution** 🚀 → Hands-on training, phishing tests, awareness drills

🔥 **Quick, structured, and ready for last-minute review!** 🚀