STUDY GUIDE NOTES - PART 1 (CH.01-CH.10)

CHAPTER 1: TODAY'S SECURITY PROFESSIONAL

CYBERSECURITY OBJECTIVES:

- Confidentiality: unauthorized individuals are not able to gain access to sensitive info
  - Examples: Firewalls, ACLs (access control lists), encryption
- Integrity: ensuring no unauthorized modifications of data
  - Examples: hashing, integrity monitoring
- Availability: data/systems are readily available
  - Examples: fault tolerance, clustering, backups
- Nonrepudiation: digital signature, cannot deny it was you
- Disclosure: data loss or data exfiltration. Opposite of confidentiality
- Alteration: unauthorized modification of data. Opposite of integrity
- Denial: disruption of authorized users to access data. Opposite of availability

- CIA Triad:
  - Confidentiality
  - Integrity
  - Availability
  - (Nonrepudiation)
- DAD Triad:
  - Disclosure
  - Alteration
  - Denial
- Breach Impacts:
  - Financial Risk
  - Reputational Risk: goodwill of public
  - Identity Theft
  - Strategic Risk: meeting major goals
  - Operational Risk: day-to-day functions
  - Compliance Risk: legal or regulatory requirements

IMPLEMENTING SECURITY CONTROLS:

- Control Objectives: desired security state
- Security Controls: specific measures to achieve control objectives
- Gap Analysis: examining security controls VS control objectives

- Technical Controls: firewall rules, access control lists, IPS, and encryption
- Operational controls (AKA processes): access reviews, log monitoring, vulnerability management
- Managerial controls (AKA mechanisms of risk management): periodic risk assessments, security planning exercises, change management
- Physical controls: fences, lighting, locks, fire suppression, alarms

  - Security Control Categories:
    - Technical Controls
    - Operational controls
    - Managerial controls
    - Physical controls
  - Security Control Types:
    - Preventive Controls
    - Deterrent Controls
    - Detective Controls
    - Corrective controls
    - Compensating controls: PCI DSS
    - Directive controls

DATA PROTECTION:

- Encryption: use math algos to protect data from prying eyes
- DLP (Data Loss Prevention):
- Agent-based DLP: Detecting and monitoring software searching for sensitive info
- Agentless (network-based) DLP: a dedicated devices on network that blocks traffic and auto-applies encryption
- Segmentation: placing sensitive systems on separate networks
- Isolate: cuts system off from access

  - Types of Data States:
    - Data at rest
    - Data in transit
    - Data in use
  - DLP actions:
    - Pattern matching (ex: looking for SSNs or keywords)
    - Watermarking: tagging sensitive data
    - DRM (Digital Rights Management): enforce copyright and data ownership
  - Data Minimization: reduces sensitive data we use regularly
    - Deidentification: removing ability to link data back
    - Hashing

        - Subject to rainbow table attack when attacker forces a collision
    - Tokenization (ex: student ID)
    - Masking
- Access Restrictions:
    - Geographic Restrictions: limit access by geography
    - Permission restrictions: limit access by level of authorization (ex: IAM)

CHAPTER 2: CYBERSECURITY THREAT LANDSCAPE

EXPLORING CYBERSECURITY THREATS:

- Hacker Hats:
    - Black Hat: Unauthorized
    - Gray Hat: semi-authorized
    - White Hat: Authorized
- Threat Actors:
    - Script Kiddie: unskilled attacker
    - Hacktivists
    - Organized Crime (ex: Russian Mafia)
        - Crimes: Cyber-dependent crime (ransomware), child sexual abuse material, online fraud, dark web, cross-cutting crime factors
    - Nation-State Attackers —> APTs (Advanced Persistent Threats)
    - Insider Threat (ex: Shadow IT)
    - Competitors
- Attacker Motivations:
    - Data exfiltration
    - Espionage
    - Service Disruption
    - Blackmail
    - Financial Gain
    - Philosophical/political beliefs
    - Ethical attacks
    - Revenge
    - Disruption/chaos
    - War

THREAT VECTORS:

- Message-based Threat Vectors: SMS, email, voice phishing, social media
- Wired Networks: physical security
- Wireless Networks: bluetooth, unsecured wireless networks
- Systems: OS, legacy apps

- Files & Images: individual files
- Removable devices: USB drives
- Cloud: improper access controls, security flaws, compromised API
- Supply Chain: MSPs (Managed Service Providers)

THREAT DATA AND INTELLIGENCE:

- Threat intelligence: set of activities and resources available to cybersecurity professionals to learn about threat environment
  - Used for predictive analysis
- Vulnerability Databases
- OSINT (Open Source Intelligence)
  - CISA (cybersecurity and infrastructure security agency
  - AIS (Automated Indicator Sharing)
  - Microsoft's tjhrad intelligence
  - Cisco Security Advisories
- WHOIS Lookup: external registries (passive), also called Domain Name lookup.
  - DNS lookup gets the IP, WHOIS or Domain Name lookup just gets the name
- IoCs (Indicators of Compromise): file signatures, log patterns, file and code repositories
- Proprietary and Closed-Source Intelligence
- Threat maps: geographic view of threat intelligence (geo info is notoriously unreliable)
- Confidence score: score orgs give intelligence on how much they can trust it (ex: high, medium, low)

THREAT INDICATOR MANAGEMENT:

- STIX (Structured Threat Information eXpression): XML language describing the attack in a STIX JSON:
- 

- OASIS (Organization for the Advancement of Structured Information Standards): nonprofit that maintains XML & HTML
- TAXII (Trusted Automated eXchange of Intelligence Information protocol): companion to STIX, communication via HTTPS

- Other Threat Intelligence Sources:

- ISACs (Information Sharing and Analysis Centers)
- Vendor sites
- Vulnerability feeds
- Conferences
- Social media accounts
- Academic Journals (RFCs)
- TTPs (Tactics, techniques, and procedures)

CHAPTER 3: MALICIOUS CODE

MALWARE:

- Ransomware
  - IoCs: contact with malicious IP, abnormal behavior, lateral movement, encryption of files, data loss
  - Counter: backup system
- Trojan: disguised as legitimate software
  - RAT (remote access trojans)
  - IoCs: malware signatures, files created on target devices, C&C (command & control) system hostnames
  - Counters: awareness, anti-malware, EDR
- Worm: self-replicating
  - Examples: Student 2010 attack against Iran nuclear plant, Raspberry Robin
  - IoCs: known malicious files, remote systems, C&C with remote systems, use of malicious cmd.exe, hands-on-keyboard attacker activity
  - Counters: Firewalls, IPS, network segmentation, patching
- Virus: require infection mechanism to spread themselves. Typically have a trigger and a payload
  - Types: memory-resident viruses, non-memory resident viruses, boot sector viruses, macro viruses, email viruses
    - Fileless viruses: via spam email or websites, exploit plug-ins or web browsers
  - Counters: anti-malware, IPS, awareness
- Spyware: associated with identity fraud, stalkerware
  - IoCs: remote access indicators, known software signatures, malicious processes, injection attacks against browsers
  - Counters: awareness, antispyware

- Bloatware
- Keylogger
    - IoCs: malicious file hashes and signatures, exfiltration, process names, known malicious URLs
- Rootkit: can infect MBR (master boot record)
    - IoCs: C&Cs, opening ports/proxy tunnels
    - Counters: review code, mount drive on another system
- Logic bomb: functions or code placed inside programs that will activate when conditions are set
    - Counter: code review
- Botnet
    - Use C&C (command & control)
    - Use HTTP connections or IRC (Internet Relay Chat) Port 6667

- Analyzing malware:
    - VirusTotal
    - Sandbox
    - Code analysis

CHAPTER 4: SOCIAL ENGINEERING AND PASSWORD ATTACKS

SOCIAL ENGINEERING AND HUMAN VECTORS:

- Phishing: fraudulent acquisition of information
    - Smashing
    - Vishing
    - Spear phishing: targeted phishing
    - Whaling: targeting whales
- MDM: abbreviation
    - Misinformation: incorrect information from getting facts wrong not on purpose
    - Disinformation: incorrect information on purpose to serve a goal
    - Maliformation
- Impersonation
- Brand Impersonation
- BEC (Business Email Compromise): compromised accounts, spoofed email, typo squatting domains, malware
- Pretexting: made-up scenario to justify
- Watering Hole Attack

- Typo squatting
    - Pharming: redirects victim to lookalike site by attacking system's hosts file

- Key Principles:
    - Authority
    - Intimidation
    - Consensus-based
    - Scarcity
    - Familiarity
    - Trust
    - Urgency
- CISA recommends "TRUST" model to counter:
    - Tell your story
    - Ready your team
    - Understand and assess MDM
    - Strategize response
    - Track outcomes

PASSWORD ATTACKS:

- Brute-force attacks:
    - Password spraying attack: one password, many accounts
    - Dictionary attacks: using list of words for attacks (ex: John The Ripper)
- Rainbow attacks: creating a hash collision
- OWASP (Open Worldwide Application Security Project): maintains cheat sheet of secure password storage

---

CHAPTER 5: SECURITY ASSESSMENT AND TESTING

VULNERABILITY MANAGEMENT:

- NVD (National Vulnerability Database)
- SCAP (Security Content Automation Protocol): standardized communication approach for security info created by NIST
    - CCE (Common Configuration Enumeration): systems and configuration issues
    - CPE (Common Platform Enumeration): product names and versions
    - CVE (Common Vulnerability & Exposures): security flaws
    - CVSS (Common Vulnerabilities Scoring System): measuring and describing severity

    - XCCDF (Extensible Configuration Checklist Description Format): reporting checklist results
    - OVAL (Open Vulnerability and Assessment Language): low-level testing procedures

- Vulnerability scanning types:
  - Intrusive plug-ins vs non-intrusive plug-ins
  - Credential scanning (read-only)
  - Server-based scanning
  - Agent-based scanning
- Asset inventory and asset criticality
  - Example: Nessus scan
  - Requirements: regulatory, technical, business, licensing
- Scan perspectives: external (via internet) & internal
  - Both are required by PCI DSS
  - Must be an ASV (Approved Scanning Vendor)
- Examples of Network vulnerability scanners:
  - Nessus
  - Qualys
  - Rapid7's Expose
  - OpenVAS
- Application Testing:
  - Static testing: analyzes code without executing it
  - Dynamic testing: executes code as part of test
  - Interactive testing: combines static and dynamic (analyzes code while testers test)
- Web Application Scanning:
  - Example: Nikto, Nslookup for DNS
  - Testing for: XSS, CSRF, SQL injection
- CVSS Qualitative Severity Rating Scale:
  - CVSS Vector: single-line format used to convey ratings
    - Attack Vector, Attack Complexity, Privileges Required, User Interaction, Scope, Confidentiality, Integrity, Availability
  - 0.0: None
  - 0.1-3.9: Low
  - 4.0-6.9: Medium
  - 7.0-8.9: High
  - 9.0-10.0: Critical
- Reconciling Scann Results:
  - Log reviews: servers, app, network devices
  - SIEM (Security information and event management)
  - Configuration management systems

VULNERABILITY CLASSIFICATIONS:

- Patch Management
- Legacy Platforms
- Weak configurations:
  - Default passwords
  - Unsecured accounts
  - Open ports
  - Open permissions
- Insecure Protocols:
  - FTP
  - Telnet
- Weak Encryption
  - Two choices of encryption:
    - Algorithm to perform encryption/decryption
    - Encryption key
- Debug Modes: error information when troubleshooting

PENETRATION TESTING:

- Pen testing: most effective way to gain a complete picture of security vulnerabilities
  - Attackers only need to win once. Cybersecurity need to win every time
  - Benefits of Pen Testing: first-hand knowledge, constructive feedback, focused information on a specific attack targets
- SOCs (Security Operations Centers)
- Threat Hunting: looking for attacks hiding in secret

- Penetration Test Types:
  - Physical penetration Testing
  - Offensive Penetration Testing: exploiting vulnerabilities like a hacker
  - Defensive Penetration Testing: evaluating defenses
  - Integrated penetration testing: comprehensive test
- Pen Test Environments:
  - Known environment: full info
  - Unknown environment
  - Partially known environment
- RoE (Rules of Engagement):
  - Timeline
  - Technical constraints: locations, systems, applications, or other potential targets
  - Data handling requirements: during and after pen test

    - Behaviors
    - Resources: administrators, developers, time, etc
    - Legal concerns
    - Communications: # of updates
- Stages of Pen Testing:
    1. Reconnaissance: Active vs Passive
        - Examples: war driving, foot printing, war walking, war flying
    2. Running the test:
        - Initial access
        - Privilege escalation
        - Pivoting or lateral movement
        - Persistence

AUDITS & ASSESSMENTS:

- Responsible Disclosure Programs: Bug bounty programs
- Security Assessments: comprehensive review of the security of a system. Internal use only
- Security Audit: independent auditors. Potentially public (government, boards, etc)
    - Attestation: formal statement saying its working properly
    - Three types: internal, external, and 3rd party
- Internal Audits: internal staff and internal audience only.
    - Compliance obligations
    - Self-assessment
- External Audits: outside firm but request comes from internal
    - Examples: big four audit firms —> East & Young, Deloitte, PwC, KPMG
    - Financial audits are usually external
- Independent Third-Party Audits: request comes from external like regulator
- COBIT (Control Objectives for Information and related Technologies): auditing standards
    - Maintained by ISACA (Information Systems Audit and Control Association)

- Security assessment program:
    - Security tests
    - Security assessments
    - Security audits
- Security Tests:
    - Availability
    - Criticality
    - Sensitivity

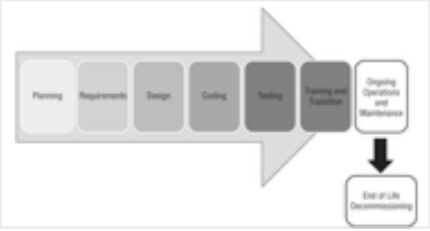
- Likelihood of technical failure
- Likelihood of misconfig
- Risk of attack
- Rate of change of control config
- Other changes
- Difficult and time
- Impact on business operations

VULNERABILITY LIFE CYCLE:

1. Vulnerability Identification: scans, pen tests, bug bounty, audits
2. Vulnerability analysis: CVSS, exposure factor, environmental variables, risk tolerance
3. Vulnerability Response and Remediation: patch, segmentation, firewalls, cyber insurance
4. Validation of Remediation
5. Reporting

CHAPTER 6: APPLICATION SECURITY

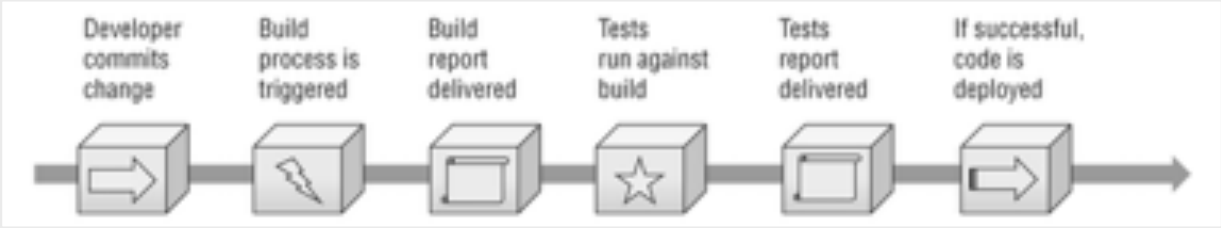
SOFTWARE ASSURANCE BEST PRACTICES:



- SDLC (Software development lifecycle):
  1. Planning: initial effort, considering alternative solutions
  2. Requirements: important to include security requirements definition
  3. Design: functionality, architecture, integration points, data flows
  4. Coding: small unit testing
  5. Testing: UAT (user acceptance testing) occurs

6. Training and Transition (AKA acceptance, installation, and deployment): end users trained
7. Ongoing Operations and Maintenance: ideally longest phase
8. End of Life/Decommissioning

- Code Development Environments:
  - Development environments: where builders build. Fix bugs in this stage too
  - Test environment: AKA QA (quality assurance)
  - Staging environment: transition
  - Production: live system

- DevOps: combines software development + IT operations
- DevSecOps: combines software + IT + security
- CI/CD (Continuous Integration/Continuous deployment or delivery): consistently checking code & deploying
- 

  - Requires: continuous validation & continuous monitoring

DESIGNING AND CODING FOR SECURITY:

- OWASP (Open Worldwide Application Security Project): hosts community-developed standards/best guides
  - Define Security Requirements
  - Leverage Security Frameworks and Libraries
  - Secure Database Access
  - Encode and Escape Data
  - Validate All Inputs
  - Implement Digital Identity
  - Enforce Access Controls
  - Protect Data Everywhere
  - Implement Security Logging and Monitoring
  - Handle All Errors and Exceptions
- APIs (Application programmable interfaces) security: relies on rate limiting, input filtering, appropriate monitoring

- Software Security Testing: Veracode's 2023 metric shows 74% of apps still have 1 CVE
  - Static Code Analysis (AKA source code analysis): source code

without executing, known environment with full visibility
    - Dynamic Code Analysis: execution of code
    - Fuzz testing or fuzzing: testing codes ability to handle random data —> only for simple problems

INJECTION VULNERABILITIES:

- Injection Vulnerabilities: primary attack for web applications
- SQL Injection attacks
- Blind SQL attacks:
    - Blind content-based SQL Injection: no data shown but website displays true or false
    - Blind timing-based SQL injection: show password using boolean commands (ex: If the first letter of the first database's name is an 'A', wait for 10 seconds)
- Code Injection Attacks: seeking to insert malicious code
    - LDAP (lightweight directory access protocol) injection attack
    - XML injection attack
    - DLL injection attack
    - XSS (uses HTML)
- Command Injection Attacks:
- Exploiting Authentication Vulnerabilities:
    - Exploiting password authentication: social engineering, spying, credential harvesting, default password
    - Session Attacks:
        - Session hijacking
        - Cookies stealing: digital version of a badge. Methods: Eavesdropping, malware, on-path attack
        - Session replay attack: literally replaying your session as you
            - Use secure cookies
        - NTLM pass-the-hash attacks: steals the hash tries to unlock stuff with it
        - Unvalidated Redirects: insecure URL redirects
            - Counter: only allow validated redirects
- Exploiting Authorization Vulnerabilities
    - IDOR (Insecure direct object reference): when a web app provides direct access to something by modifying the URL
        - Example: changing it from 123 to 124 to 125
    - Directory Traversal: navigating directory paths to somewhere else on the server (ex: using the ".." In the header
    - Locale file inclusion: tricks web application to running code contained within a malicious file
        1. Locale file inclusion: executing code stored locally on the web

server
2. Remote file inclusion: tricks web app to run file on a remote server (even worse)
- Privilege Escalation
- Exploiting Web Application Vulnerabilities:
  - XSS (cross-site scripting): attacker uses HTML injection into a web app
  - Non-persistent XSS (Reflected): injecting HTML code into the error message and the website unknowingly spits it right back
  - Stored/Persistent XSS: waiting in the site's database for your to interact with it (ex: in a blog's comments)
  - DOM Based XSS: written deep in JS code, look for eval() method
  - Blind XSS: sending a hidden payload that collects victims info like cookies, credentials, etc. XSS Hunter Express is a good tool
- Counters to XSS: input validation, don't allow for things like <SCRIPT>
- Request Forgeries: exploit trust relationships
  1. CSRF/XSRF (Cross-Site Request Forgery): AKA Sea Surf or Session Riding. Uses one session to hack another site's session (ex: like an open banking tab)
  2. SSRF (Server-side Request Forgery): tricking a server to visit a URL based on user-supplied input. Possible when web app accepts URLs as input

APPLICATION SECURITY CONTROLS:

- Input Validation: must be server side
  - Allow listings: dev spells out the exact type of input expected from the user —> most secure
  - Deny listing: tricky
  - Parameter Pollution: sending web app more than one value for the same input variable
    - Example: =12345&account=12345′ OR 1=1;- -
- WAFs (Web Application Firewalls): firewall specific too the application layer (L7), sits in front of web server, looks at input/performs its own input validation
  - Used in addition to input validation
- Parameterized Queries: sends parameters not code to databases to prevent injection
  - Example: Stored procedures
- Sandboxing: controlled test environment
- Code Security:
  - Code Signing: digitally signing code with their private key, browser has public key. Protects against malicious updates

    - Code Reuse: should be tested (ex: SDKs)
    - Software Diversity: avoid SPOF
    - Code Repositories: centralized location for storage and management of application source code, version control, track changes, prevents dead code
    - Integrity measurement: using hash to match approved code with code rolling into production
    - Application Resilience: resilient in the face of changing demand
        - Scalability: support demand as needed
        - Elasticity: provision/deprovision resources automatically

SECURE CODE PRACTICES:

- Source Code Comments: dev-to-dev comments. But be careful with explaining too much —> remove from production code
- Error Handling: write code that handles errors well. Prevents something dangerous
- Hard-Coded Credentials:
    - Backdoor vulnerability: allows dev to gain access to code with UN/PW
    - Accidental disclosure: secret UN/PW or API keys is accidentally exposed when code is published on a repository
- Package Monitoring: keep track of all 3rd libraries/packages
- Memory management
- Resource exhaustion
    - Memory leaks: when an application fails to return memory that it no longer needs —> can crash the system over time
- Pointer Dereferencing: when a memory pointer (AKA cache) goes to something that is empty or malicious. Best outcome is that it crashes. Worst outcome is that its compromised
- Buffer Overflows: placing more data into a memory area than it can handle —> goal is a memory injection attack
- Race Conditions:
    - TOC (Time-of-Check): instance when systems verifies permissions
    - TOU (Time-of-use): when moment when system accesses the resource
    - TOE (Target of Evaluation): being evaluated for potential vulnerabilities
    - TOCTTOU or TOC/TOU (time of check to time of use): if someone is logged on already and permission is removed...well too bad. They have that resource FOREVER
- Unprotected APIs

AUTOMATION AND ORCHESTRATION:

- SOAR (Security orchestration, automaton, and response): automating responses, learn of emerging threats,
    - Scripting to automate log analysis, scans, and other responses

- Automation & Scripting use cases:
    - User provisioning
    - Resource provisioning
    - Guard rails: automation employed to enforce policy controls and prevent violations of security protocols
    - Security groups
    - Ticket creation
    - Escalation
    - Enabling/disabling services and access
    - Continuous integration and testing
    - Integrations and API
- BENEFITS:
    - Achieving efficiency and time savings
    - Enforcing baselines
    - Standardizing infrastructure configurations
    - Scaling securely
    - Retaining employees: increase job satisfaction
    - Reducing reaction time
    - Increase team's workload capacity
- CONS:
    - Complexity
    - Cost: implement automation and scripting involves upfront costs, invent in tools, training, and potentially new staff members with expertise
    - SPOF (single point of failure)
    - Technical debt: automated scripts become outdated over time
    - Ongoing supportability

CHAPTER 7: CRYPTOGRAPHY AND PKI

AN OVERVIEW OF CRYPTOGRAPHY:

- Cryptography: practice of encoding information that it cannot be decoded with the required decryption key
    - Encryption: plaintext —> cipher text via encryption key
    - Decryption: cipher text —> plaintext via decryption key

    - Sometimes cryptography/cryptology used interchangeably
- Substitution cipher: ciphering that substitutes one character for another
    - Oldest one was called Caesar cipher (used by the man JC himself)
    - ROT13: shifts letters 13 places to the left (ex: A becomes an N)
- Polyalphabtic Substitution: shifting letters around even more
    - Polyalphabtic substitution ciphers
    - Vigenere cipher: keyword to lookup cipher text
    - Transposition Ciphers: scrambling letters in a certain manner
    - Enigma Machina: Nazi's encryption broken by Alan Turing
- Steganography: art of using to cryptographic techniques to embed secret messages in another file
    - Alters the least significant bits that make up image files —> not even perceivable to human eye (AKA hiding in plain sight)
    - Example: Digital watermark, OpenStego creates digital watermarks in image files

- Cryptography goals:
    - CIA triad except A stands for AUTHENTICATION and not AVAILABILITY
    - Nonrepudiation
- Historical Cryptography:
    - First cryptographic effort was 4000 years ago
    - Cipher: method to obfuscate characters
    - Ciphering: act of obfuscation
    - Non-mathematical cryptography substitution and transportation

GOALS OF CRYPTOGRAPHY:

- Confidentiality
- Integrity: ensures data is not altered without authorization
    - Via digital signatures
- Authentication: verifies the claimed identity
    - Via challenge-response authentication technique
- Non-repudiation: prevents sender from saying that they never sent the message

- Two main types of crypto systems:
    - Symmetric crypto systems: shared key available to all users of the crypto system
    - Asymmetric cryptosystems: combo of public and private keys for each user
- Three types of data sets:
    - Data at rest

    - Data in transit: AKA data on the wire, uses TLS
    - Data in use
- Types of Encryption:
    - FDE (Full-disk encryption): all files on a hard rive automatically encrypted (including OS and system files)
    - Partition encryption: targets specific portions of the disk
    - File-level encryption: individual files
    - Volume encryption: volume on a storage device, in between partition encryption and file-level encryption

CRYPTOGRAPHIC CONCEPTS:

- P: Plaintext
- C: cipher text
- Cryptographic Keys: every crypto algorithm relies on keys to maintain their security
    - Key Space: range of values that are valid for the key to use for an algorithm AKA all the possibilities
    - Key Length: number of binary bits in the key
    - Kerckhoff's Principle/assumption: the enemy knows the system (not all agree)
        - In private/secret cryptosystems, all participants use single shared key
        - In public key cryptosystems, all participants use their own pair of keys
    - Cyptovariables: another term for cryptographic keys
    - Cryptography: creating and implementing secret codes and ciphers
    - Cryptanalysis: the study of methods to defeat codes and ciphers
    - Cryptology: cryptanalysis + cryptography
    - Cryptosystems: specific implementation of code or cipher in hardware
    - Ciphers: algorithms to perform encryption and decryption
    - Cipher suites: sets of ciphers and key lengths to support a system
        - Block ciphers: apply encryption algorithm
        - Stream ciders: one character or a bit at a time (ex: Caesar cipher)

MODERN CRYPTOGRAPHY:

- Symmetric key encryption algorithms, asymmetric key encryption algorithms, and hashing algorithms
- Modern cryptosystems: secret of one or more cryptographic keys to personalize the algorithm
- Columnar transposition (also known as permutation cipher): scrambles the position of the characters without changing the characters themselves

- DES (Data Encryption Standard): 56-bit key created decades ago (insecure)
  - Modern cyrptographic systems need at least 128-bit keys
- AES (Advanced Encryption Standard): for symmetric keys, current version is 256 bit
- Symmetric Key Algorithms: also known as secret key cryptography and private key cryptography
  - Drabacks: key exchange is dangerouso, no non-repudiation, not scalable, must be respawned
  - Asymmetric keys are sometimes use to lock symmetric keys
  - Distribution methods:
    - Physical
    - Public Key Encryption
    - DHA: key exchange algorithm
  - Storage Methods:
    - Never store it where the data resides
    - Split knowledge: two people, two halves -> like mission impossible
- Asymmetric:
  1. Digitally sign with hash using private key for integrity
  2. Encrypt message with recepients public key. Now, not even the sender can decrypt it without the receiptints private key
  3. Recipient receives data and decrypts with their private key
  4. Receipient CAN choose to use their private key to ensure the integrity and non-repudiation
  - Number of Keys (symmetric) = (n (n-1)) / 2
    - Aymmetric is always just 2x the number of users
  - Strengths: addition of new users is easy, users can be removed easily too, key regeneration only required when private key is compromised, integrity and non-repudiation, simple process, no preexisting communications links needed
- Key Escrow: third party store to protect the key
- RSA: public key algorithm for asymmetric keys only
- Key Length: longer keys are more secure but take more resources. But still need to compete with Moore's law
- ECC (Elliptic curve cryptography): harder to solve than RSA
  - Elliptic curve group

HASH FUNCTIONS:

- Hash function: takes a message and outputs a unique output value
  - Message digest: the output value of a hash function. Original message must be EXACTLY the same for the message digest to match

        - Also known as hash, hash value, hash total CRC, fingerprint, checksum, and digital ID
        - Message digests can be used for digital signatures
    - 5 requirements of a hash function:
        - Accept any input of any length
        - Always product a fixed length output (regardless of the input)
        - Hash value is easy to compute
        - Hash function is a one-way function (hard to review)
        - Collision Free
- SHA (Secure Hash Algorithm): SHA-1, SHA-2, SHA-3
    - NIST publishes the SHS (Secure Hash Standard) also know as FIPS 180
    - SHA-1 considered insure, modern hash uses SHA-2 and SHA-3 (made in 2015)
    - MD5: made in 1991. Insecure, lots of collisions
- Digital Signatures: Enforce non-repudiation & integrity
- HMAC (Hash-Based Message Authentication Code): partial digital signature —> guarantees integrity but not non-repudiation, has as shared secret key

PUBLIC KEY INFRASTRUCTURE:

- PKI (Public Key Infrastructure)
- Digital certificates: provide assurance people are who they claim to be —> endorsed copies of a individuals public key
    - Has to be signed with a CA (certificate authority)
    - X.509 Standard:
        - Current version of X.509 (V3) —> V3 supports tracking
        - Serial number from certificate creator
        - Signature algorithm identifier
        - Issuer name
        - Validity period
        - Subject's Common Name (CN)
        - SANs (Subject Alternative Name)
        - Subject's public key
    - Certificates assure the following: computers/machines, individual users, email address, developers (code-signing certificates)
    - Wildcard Certificate: designated by an "*" sign to apply for other subdomains. Example for *.certmike.com for only ONE level of subdomain
        - Certmike.com
        - www.certmike.com
        - mail.certmike.com

        - Secure.certmike.com
            - NOT!! www.cissp.certmike.com
- CAs (certificate authorities): neutral organizations offer notarization services for digital signatures
    - Examples: (can be internal & self-signed)
        - IdenTrust
        - Amazon Web Services
        - DigiCert Group
        - Sectigo/Comodo
        - Global Sign
        - Let's Encrypt
        - GoDaddy
    - RAs (Registration authorities): help CAs verify identities before digital signing
    - Root CAs protected by offline CA
        - Uses intermediate CAs that serve the online CAs (Like proxy servers)
    - Certificate chaining: verifies intermediate CA —> root CA (CA trust model)

- Certificate Generation and Destruction:
    1. Enrollment:
        - Enrollment: Appearing before the CA (sometimes in person)
        - CSR (certificate signing request): providing CA with your public key to initiate the CSR
        - CA issues you a X.509 containing your info + public key
        - CA signs your X.509 with their private key
        - Now you have a certificate you can use anywhere
            - DV (Domain Validation): CA verifies use subject has control over the domain name
            - EV (Extended Validation): higher level of assurance —> more security steps from CA
    2. Verification:
        - When another subject presents you their certificate to talk, you first verify by checking their certificate's private CA signature with the CA's public key
        - Check the certificate is not on CRL (certificate revocation list) or OCSP (Online Certificate Status Protocol)
            - Other checks:
                - Trust the CA?
                - Certificate is not listed on the CRL
                - Certificate contains legitimate data

        - Only trust whats contained in the certificate (and no other information)
    - Certificate pinning: attach a certificate to a subject for an extended time period
        - More secure than stapling because there is no availability for MITM
3. Revocation:
    - Reasons: certificate was compromised, mistakenly issued, subject's info changed, security associate changed
    - CRLs (Certification revocation lists): newlu revoked certificates. Must be downloaded tho
    - OCSP (Online Certification Status Protocol): faster and real-time verification
    - Certificate Stapling: extension of OCSP, website contacts OSCP first and gets it approval —> then staples it to the certificate for speed

- Certificate Formats: binary and text-based
    - DER (Distinguished Encoding Rules) format: binary file stored in .der .crt or .cer extension
    - PEM (Privacy Enhanced Mail): text-version of the DER format, stored in either .pem or .crt extension
        - Check file contents for differences in .crt files
    - Windows:
        - PFX (Personal Information Exchange): format for windows systems using .pfx or .p12 file
        - P7B certificates in text format

CRYPTOGRAPHIC ATTACKS:

- Brute Force
- Frequency Analysis: looking for patterns in encrypted messages
- Known Plain Text: example would be "Heil Hilter"
- Chosen Plain Text: selecting text to use a cipher
- Related Key Attack: obtaining plaintext and cipher text and trying to derive a key
- Birthday Attack: collision attack using the same hash (Same birthday theory)
- Downgrade Attack: trick the user in shifting to less secure cryptographic mode
- Rainbow table: attempts to reverse a hash value by taking common passwords, making hashes out of them, and seeing if they match
    - Salting: adds random generated value to each password PRIOR to hashing

    - Key Stretching: thousands of iterations of salting and hashing (ex: PBKDF2)
- Weak Keys/Protocols: WEP (Wireless Equivalent Privacy) uses RC4 encryption algorithm
- Human errors:
    - Unencrypted AKA in the clear

- Asymmetric Key Management:
    - Do no use “black box” systems
    - “Sufficient Entropy”: truly random
    - Keep private keys secret!
    - Rotate and backup private keys
    - HSMs (Hardware Security Modules): store and manage encryption keys so that humans never work directly with them
        - Provided by Amazon and Microsoft

EMERGING ISSUES IN CRYPTOGRAPHY:

- Tor (formely known as the The Orion Router): anonymously routing traffic across the internet using encryption and a set of relay nodes
    - Perfect Forward Security: layers of encryption that prevents nodes in the relay chain from reading messages and only lets them forward traffic
    - Allows for: anonymously browsing standard internet + hosting completely anonymous sites on the Dark Web
- Blockchain: open public ledger, creates a data store that nobody can tamper or destroy
    - Cryptocurrency: currency with no central regulator. Only one application of blockchain
    - Other applications: property ownership records, track supply chains
- Lightweight Cryptography: specialized hardware that can minimize power consumption
    - Examples: satellites, Smartcards, VPN hardware device
- Homomorphic Encryption: encrypts data but stills lets you do computations on it —> protects privacy
- Quantum Computing: quantum mechanics to use superposition to perform computing and communication skills

---

CHAPTER 8: IDENTITY AND ACCESS MANAGEMENT

IDENTITY:

- AAA (Authentication, Authorization, and Accounting)
  - Authentication: verifies claimed identity
  - Authorization: verifies resources you have access to
  - Accounting: keeps track of time, bandwidth, or CPU utilization
- Attributes: can be changeable things, like subjects title or address
- Traits: inherent to the subject —> height, eye color, place of birth
- Usernames: associated with an identity but not authentication factor themselves
- Certificates: stored on a system or paired with a storage device
- Tokens: physical device that may generate a code, plug in via USB, or connect via Bluetooth or other means to present a certificate
- SSH keys: cryptographic representations of keys that replace UN/PW
- Smartcards: cryptographic smartcards generate pairs on the card itself via embedded chip (via contactless and physical chip-reader)
  - Smartcards usually develop key pairs on the physical card itself

AUTHENTICATION AND AUTHORIZATION:

- Authentication methods:
  - SSO (single sign-on): authentication protocol
  - EAP (Extensible Authentication Protocol):
    - EAP-TLS
    - LEAP
    - EAP-TTLS
  - CHAP (Challenge Handshake Authentication Protocol): encrypted challenge + 3-way handshake
- 802.1X: IEEE standard for NAC (network access control)
- RADIUS (Remote Authentication Dial-In User Service): most common AAA system for networks, system, etc. Sends passwords via shared secret and MD5 hashed passwords.
  - UDP or TCP
  - Uses MD5 hash to encrypt passwords (not very secure)
  - Also uses IPSec tunnels
- TACACS+ (Terminal Access Controller Access Control System Plus): provides AAA via TCP, allows individual commands. Designed by Cisco
- Kerberos: authentication service ticketing request system for between hosts and untrusted networks
- SSO: main tradeoff is no further verification (easy as clicking)
  - LDAP (Lightweight directory access protocols): identity management infrastructure organizational hierarchy
    - Ous: security and Human Resources

        - CN: common names
    - SAML (Security Assertion Markup Language): XML-based open standard for exchanging authentication and authorizing information, used for identity providers
    - OpenID: open standard for decentralized authentication, “sign in with google”
        - IdP (OpenID identity providers): OpenID providers (ex: google, Facebook, amazon, etc)
        - RP (relying parties): redirect it to the IdPs
    - OAuth: open standard for authorizing websites via SSO (ex: web conferencing tools using Google calendar)
        - Handles authorization of access to protected resources
- Federation: group of trusted IdPs relaying information. Many CSPs use this
    - Principal: user
    - IdP: via attestation (verifying who the IdP is)
        - Attestation: formal verification that something is true
    - SPs (service providers): provide services to IdPs who have been attested to. An SP allows users to access their services
    - RP (relaying party)

AUTHENTICATION METHODS:

- Passwordless Authentication: security tokens, one-time passwords, certificates
    - Security Key: hardware devices
    - FIDO
    - U2F (Universal 2nd Factor)
    - FIDO2: key pair, private and public. Supports W3C Web Authentication and CTAP (Client to Authenticator Protocol)
- MFA (Multifactor Authentication): something you know, something you have, something you are, somewhere you are
- OTP (One Time Passwords): combating password theft, makes brute force attacks harder. Dynamically made
    - TOTP (Time-based One Time Password): uses algorithms to derive an OTP and then moves on
        - Example: Authenticator app
        - Can be stolen via social engineering or stealing phone
    - HMAC (Hash-based message authentication codes)
    - HTOP (HMAC One Time Passwords): generate code token from last known token
        - Example: SMS code. But susceptible to SIM cloning or VoIP network

        - Example: push notifications.
    - Static codes: algorithmically generated, stored in a secure location, but can be compromised
    - Biometrics: something you are (physiology) like fingerprints, retina scans, facial recognition, voice recognition, vein recognition, gait analysis (how a person walks)
        - FRR (False Rejection Rate): FIDO sets their standards for 3% of attempts
        - FAR (False Acceptance Rate): FIDO sets their standards at 0.01% for FAR
        - ROC (Receiver Operating Characteristic)
        - IAMPR (Imposter Attack Presentation Match Rate)

- Passwords are the most common but also most susceptible
- NIST password recommendations:
    - Using “Show Password” to show typos
    - Using password managers
    - Ensuring secrets are stored using slating and secure casing methods
    - Locking accounts after multiple attempts
    - MFA
    - Reducing password complexity in favor of password length
    - Not requiring special characters
    - Allowing ASCII and Unicode in passwords
    - Allowing pasting passwords for password managers to work
    - Monitoring new passwords to endure compromised passwords were not used
    - Eliminating password hints
- Common password settings:
    - Length
    - Complexity
    - Reuse limitations
    - Expiration dates
    - Age settings
- Password Managers: 1Password, Bitwarden, Window’s Credential Manager, Apple’s Keychain

ACCOUNTS:

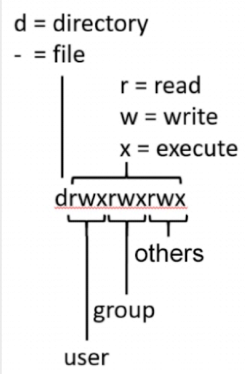
- Account Types:
    - User Account
    - Privileged or administrative accounts (AKA root counts)
        - Linux, Unix, Windows default Administrator
    - Shared and generic accounts or credentials

    - Guest accounts
    - Service accounts: associated with applications and services
- Provisioning and Deprovisioning Accounts:
    - Identity Proofing: process of ensuring that the person who the account is being created for is claiming the account
        - Examples: Government IDs, personal information
    - Permission Creep: when an employee gains a new positions and keeps all of the new and existing permissions with themself
    - Least Privilege is the key
    - Deprovisioning: terminating account, removing permissions, data, etc
    - PAM (Privileged Access Management): tools for ensuring least privilege
        - JIT (Just-in-time) permissions: granted and revoked only when needed
        - Password vaulting: access privileged accounts without knowing the password
        - Ephermeral accounts: temp accounts with a limited lifespan
- Access Control Schemes: what users, services, and programs can access various files
    - MAC (Mandatory access control): OS sets security policy, users do not have the ability to change security settings (rare setting)
        - Found in SELinux and MIC (Mandatory Integrity Control)
    - DAC (Discretionary Access Control): more common, access control scheme to control home PCs.
        - Examples: Linux file permissions
    - RBAC (ROLE-based access controls): roles are matched with privileges, popular with enterprises (ex: cashier, database admin)
        - Role assignment: subjects can only use permission that they have been assigned
        - Role authorization: subject's active role must be authorized for subject
        - Permission authorization: subjects can only use permissions that their active role is allowed to use
    - RuBAC (RULE-based access control): set of rules that apply to various objects or resources (ex: firewall ruleset)
    - ABAC (Attribute-based access control): relies on policies that are driven by attributes of the users. Complex to manage
    - Time-of-day restrictions: limits who activities can occur
    - Least privilege: concept that says users should only be given the minimum set of permissions and capabilities they need to perform their job
    - Filesystem Permissions: which users can perform actions like reading,

writing, and executing
  - Windows file permissions set in GUI
  - Linux set in command line
  - Filesystem permissions often exploited by attackers

CHMOD LINUX COMMANDS:

NUMERIC REPRESENTATION	PERMISSION	LETTER REPRESENTATION
0	No permission	- - -
1	Execute	- - x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read + Write	rw-
7	Read + Write + Execute	rwx



Numeric representation	Permission	Letter representation
0	No permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read + Write	rw-
7	Read + Write + Execute	rwx

CHAPTER 9: RESILIENCE AND PHYSICAL SECURITY

RESILIENCY AND RECOVERY IN SEC ARCHITECTURE:

- Continuity of Operations
  - Factors: cost, maintenance requirements, suitability
  - Redundancy is the most common solution for resiliency

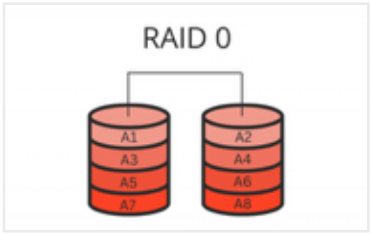
- Redundancy Factors:
  - Geographic dispersion
  - Separate servers and PDUs (power distribution units)
  - Multiple network paths (multipath)
  - Redundant network devices:

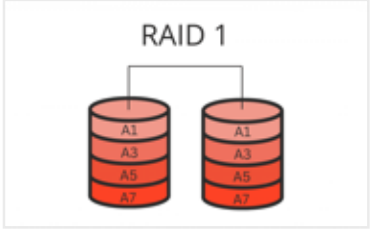
    - Load balancing: allows multiple systems to appear like a single resource
    - Clustering: group of computers provide the same task
  - Power protection:
    - UPS (uninterruptible power supply)
    - Generator
    - Dual-supply
    - PDUs (managed power distribution units)
  - System storage redundancy
  - Platform diversity: different tech & vendors

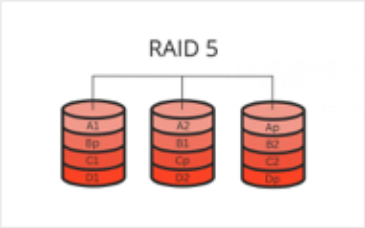
ARCHITECTURAL CONSIDERATIONS AND SECURITY:

- Availability
- Resilience: how much disruption a service/system can handle without availability issue
- Cost
- Responsiveness: responding in a timely manner
- Scalability: vertically (bigger) or horizontally (more)
- Ease of deployment
- Risk transference: insurance, contracts, etc
- Easy of recovery
- Patch availability & vendor support
- Inability to patch
- Power consumption
- Compute requirements

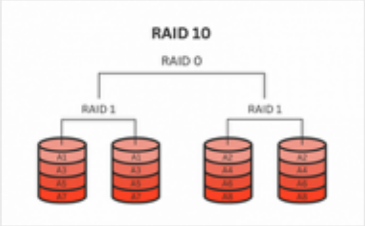
STORAGE RESILIENCY:

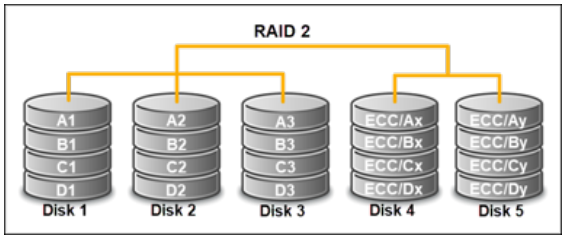
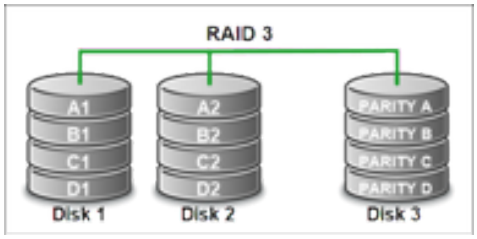
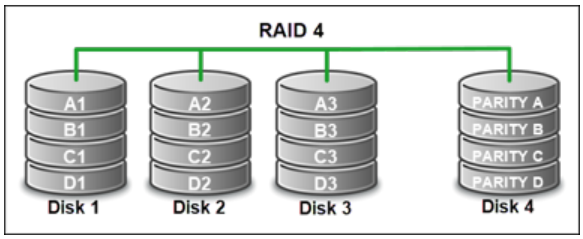
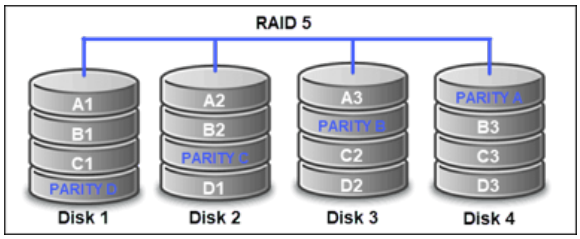
- RAID: Redundant Array of Independent Disks
- 

- 


RAID 5


RAID 6


RAID 10
RAID 0
RAID 1
RAID 1

- 
RAID 2
A1
B1
C1
D1
A2
B2
C2
D2
A3
B3
C3
D3
ECC/Ax
ECC/Bx
ECC/Cx
ECC/Dx
ECC/Ay
ECC/By
ECC/Cy
ECC/Dy
Disk 1
Disk 2
Disk 3
Disk 4
Disk 5
- 
RAID 3
A1
B1
C1
D1
A2
B2
C2
D2
PARITY A
PARITY B
PARITY C
PARITY D
Disk 1
Disk 2
Disk 3
- 
RAID 4
A1
B1
C1
D1
A2
B2
C2
D2
A3
B3
C3
D3
PARITY A
PARITY B
PARITY C
PARITY D
Disk 1
Disk 2
Disk 3
Disk 4
- 
RAID 5
A1
B1
C1
PARITY D
A2
B2
PARITY C
D1
A3
PARITY B
C2
D2
PARITY A
B3
C3
D3
Disk 1
Disk 2
Disk 3
Disk 4

RAID description	Description	Advantage	Disadvantage
RAID 0 - Striping	Data is spread across all drives	Best I/O performance (speed); all capacity used.	Not fault tolerant—all data lost if a drive is lost.
RAID 1 - Mirroring	All data is duplicated to another drive or drives	High read speeds from multiple drives; data available if a drive fails.	Uses twice the storage for the same amount of data.
RAID 5 - Striping with parity	Data is striped across drives, with one drive used for parity (checksum) of the data. Parity is spread across drives as well as data. Parity drives can restore lost data	Data reads are fast; data writes are slightly slower. Drive failures can be rebuilt as long as only a single drive fails. More efficient than RAID 1	Can tolerate only a single drive failure at a time. Rebuilding arrays after a drive loss can be slow and impact performance.
RAID 10 - Mirroring and striping	Requires at least four drives, with drives added in pairs. Data is mirrored, then striped across drives.	Combines the advantages and disadvantages of both RAID 0 and RAID 1.	Combines the advantages and disadvantages of both RAID 0 and RAID 1. Sometimes written as RAID 1+0.

- Backup Types:
  - Full Backup: copies the entire device or storage system
  - Incremental backup: captures changes since last incremental backup
    - Fatest to backup
    - Slowest to recover
  - Differential backup: captures changes since last full backup
    - Slow to backup
    - Fast to recover
  - Replication: synchronous (real-time) or asynchronous (after-the-fact) methods of copying data
  - Journaling: creates log of changes that can be replies of an issue

occurs —> restoring to a fixed snapshot
    - Journal also needs to be stored somewhere
  - Snapshot: captures full state of a system at the time the backup is completed (common for VMs)
    - Captured live
    - Can consume a lot of storage
  - Images: complete copy of a server or a drive down to each bit. Backup method of choice for complex servers
  - Gold Master Image: best and final version of a VDI (Virtual Desktop Infrastructure), System, or Server
- Recovery Process:
  - RPO (Recovery Point Objectives): how much data loss is acceptable
  - RTO (Recovery Time Objectives): how long the recovery can take
- Backup Medias: encrypt in storage and transit
  - Tape: lowest cost, still in use
  - Disks: HDD/SSD, NAS or SANs, more expensive than tape
  - Optical Media: Blu-Ray, DVDs —> not common
  - Flash Media: SD cards, thumb drives
  - Nearline backups: not immediately available but can be retrieved
    - Faster than offsite, but slower than onsite
    - Examples: Amazon's S3 Glacier, Google's Coldline storage
  - Off-Site Storage: Iron Mountain
    - Quick restoration is not possible
    - Takes a long time and higher cost
    - Overestimated reliability
    - New security models require for backups
  - New code backups: industry becoming software-defined infrastructure model, only code that defines cloud being backed up

RESPONSE AND RECOVERY CONTROLS:

- Key Resiliency Factors:
  - Geographic dispersal
  - Power protection
  - Technology diversity
  - Backups: tape, disk, 3rd party, full backup, differential backup, incremental backup, snapshots
  - Capacity planning
  - DRP: disaster recovery planning
- Nonpersistence: ability to have systems or services that are spun up and shut down as needed
- Documented Restoration order: helps ensure that systems and services have dependencies start in the right order

- Last-known good configuration: windows system build this in the patching process
  - Live boot media: bootable operating system that can run from removable media like thumb drive or DVD
- Scalability: services are designed to scale across many servers instead of requiring one large server to handle more workload
  - Vertical Scalability: all tasks or functions need to be handled on the same system
  - Horizontal Scaling: using smaller systems but adding more of them. Elastic, transparent upgrades, patching
- Recovery Sites:
  - Hot site: basically operated full-time
  - Warm Site: have systems but no live data
  - Cold Site: only bare metal infrastructure
  - Multi-cloud: business will continue even if one cloud vendor has a problem

- Testing Resilience and Recovery Controls and Designs:
  - Tabletop exercises
  - Simulation exercises
  - Parallel processing exercises: validating hot site will work
  - Failover exercises: testing that involves an actual failure to another site
- Physical Security Controls:
  - Site Security: plan that looks at entire facility
  - Fences
  - Bollards
  - Lighting
  - Drone Defense (UAV)
  - Access badges (RFID)
  - Access control vestibules: prevents tailgating/piggybacking
  - Alarms
  - Fire Suppression
  - Locks: “locks keep honest people honest”
  - Security guards
  - Video surveillance: motion recognition, object detection, CCTV (closed-circuit television)
  - Sensors:
    - infrared sensors (smaller indoor spaces)
    - pressure sensors (less common)
    - microwave sensors (expensive and more error-prone than infrared)

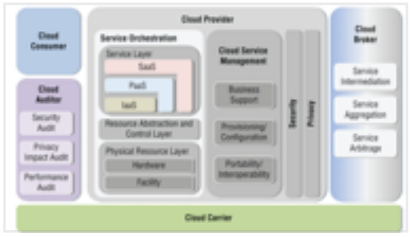
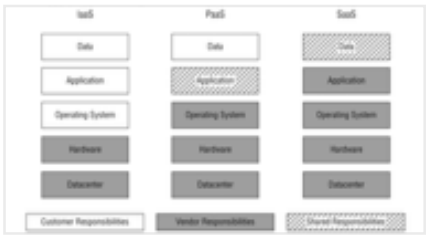
        - Ultrasonic sensors (proximity)
- Detecting Physical Attacks:
    - Physical Brute-force attacks
    - RFID Cloning: cloning RFID tag or card
    - Environmental attacks: attacking HVAC, sprinkler, etc

CHAPTER 10: CLOUD AND VIRTUALIZATION SECURITY

EXPLORING THE CLOUD:

- CSA (Cloud Security Alliance): made the CCM (Cloud Controls Matrix) to understand appropriate use of cloud security controls
- Edge Computing: IoT devices that preprocess data before shipping it back to the cloud
    - I.e. at the very edge
- Fog computing: IoT sensors—> local gateway —> cloud
    - Layer between edge computing and server

- Cloud Computing: delivers computing services over the internet that scale to business needs
    - Examples: Gmail, AWS
    - Convenient
    - On-demand
    - Oversubscription/multitenancy: same physical environment
    - Rapid provisioning and deprovisioning
- Cloud Benefits:
    - On-demand self-service computing
    - Scalability: Vertical vs Horizontal
    - Elasticity
    - Measured service: pay as you go
    - Agility and flexibility
- Cloud Roles:
    - CSPs (Cloud service providers)
    - Cloud consumers
    - Cloud partners/cloud brokers: software development, integration services, cloud services
    - Cloud Auditors: 3rd party assessment
    - Cloud carriers: delivery of cloud services from providers to consumers
- Cloud Service Models:

    - IaaS: bare metal computing. Dominated by AWS, Azure, and GCP
    - SaaS: fully managed app
    - PaaS: in between SaaS and SaaS. Customers may run applications that they have developed themselves
        - FaaS (serverless computing): customers write code functions, providers execute. (Ex: AWS Lambda). Not exposed directly to servers
    - XaaS
    - MSPs (Managed Service Providers): capable of working customer's total environment, both on-premises and cloud
    - MSSPs (Managed Security Service Providers): security monitoring, vulnerability management, incident response, and firewall management
- Cloud Deployment Models:
    - Public Cloud: any customers (ex: AWS, Microsoft Azure, GCP)
    - Private Cloud: for one customer (ex: CIA)
    - Community cloud: private & public —> for a specific community
    - Hybrid cloud: public/private/community —> decentralized
        - Cloud bursting: when. Private cloud leverages public cloud when demand exceeds
        - Example: AWS Outposts —> AWS equipment but privately enabled
- Shared Responsibility Matrix/Model
    - IaaS: Hardware and datacenter
    - PaaS: Hardware, Datacenter, and OS
    - SaaS: Hardware, Datacenter, OS, and Application
    - 
    - 

VIRTUALIZATION:

- Virtualization: allowing multiple guests (multitenancy) with the same hardware
    - Runs on hypervisor —> OS (Windows/Linux)
- Hypervisors: isolates virtual machines. Must present illusion of being

completely separate physical environment
    - Type 1 Hypervisor: bare-metal hypervisors, operate directly on top of hardware
    - Type 2 Hypervisor: runs on top of existing operating system. Common for devs
- VMs (Virtual Machines): VMs are the building block of cloud, cost of a server based on an hourly rate of computing resources
    - Microsoft Remote Desktop tool: RDP (Remote Desktop Protocol) + Windows IaaS
- Containers: application-level virtualization (ex: Docker), interface is the same regardless of hardware/OS, can shift between systems as needed
    - Security issues: isolation
    - NIST recommendations: container-specific OS, segmenting containers based on risk profile, container-specific vulnerability management tools

- Cloud Storage Resources:
    - Block Storage: large volumes of storage for a virtual instance
        - Example: AWS's EBS (Elastic Block Storage). You pay for the capacity allocated no matter if you use it
    - Object Storage: treat each object independently, can be accessed via web or API. You pay as you go
        - Example: AWS S3 storage bucket
    - Security concerns: set permissions properly, consider high availability, use encryption

CLOUD NETWORKING:

- SDN (Software-Defined Networking): allows engineers to interact and modify cloud resources via APIs
    - SDV (Software-Defined Visibility): traffic insight on virtual networks

- Security Groups: CSPs do NOT provide customers with access to their firewalls, instead they provide security groups that define permissions
- VPC (Virtual Private Clouds): virtual segmentation —> designate subnets as private or public
    - Segmentation: physical —> VLAN. Virtual —> VPC
    - VPC Endpoints: connects VPCs to each other
    - Cloud Transit Gateways: connects VPCs to VLANs
- DevOps and Cloud Automation:
    - DevOps: development + operation teams
    - IaC (Infrastructure as Code): deployment of infrastructure services via code rather than humans (sometimes offered via CSP or 3rd

party)
    - Examples: AWS's CloudFormation -> devs customize infrastructure requirements (i.e. JS, JSON, YAML)
    - APIs
    - Microservices: CSP offering that provide granular functions

CLOUD SECURITY ISSUES:

- Availability: many regions on each major continent (high availability might be a purchasable upgrade tho)
- Data Sovereignty: data is subject to legal restrictions based on where it is collected, stored, or processed
- Virtualization Security:
    - VM Escape
    - VM sprawl: accumulation of VMs overtime and forgetting about them
    - Resource reuse: new customer might accidentally gain data from old customer using the same hardware
- Application Security:
    - API inspection: scrutinizes API requests for security issues
    - SWGs (Secure Web Gateways): application security for cloud-dependent orgs. Monitor web requests, blocking suspicious requests
- Governing & Auditing of Third-Party Vendors:
    - Vetting vendors for cloud partners
    - Managing vendor relationships
    - Overseeing cloud activities
    - Audibility: customer's right to audit

HARDENING CLOUD INFRASTRUCTURE:

- Native cloud controls: cost-effective and user-friendly
- 3rd-party solutions: costly but mitigate risk
- CASB (Cloud Access Security Brokers): software tools in-between cloud users and providers
    - Inline CASB solutions: physically inline between users and providers, sees requests before they go to the cloud
    - API-based CASB solutions: can only monitor and detect API requests
- Resource policies: limits the actions that users of their accounts may take
- HSMs: expensive/complicated to operate but can create and manage encryption keys without exposing them to a single human being.