CySA+

Chapter 1: The Importance of Threat Data and Intelligence

* Foundation of Intelligence
  + Network Defense is Knowing
    - what is present on the network
    - what weaknesses exist
    - who might be targeting them
  + Threat Data + Context = Threat Intelligence
    - which leads to an understanding of operational environments
  + Understanding the Primitives of Intelligence Analysis
    - SIGINT
    - MASINT
    - HUMINT
    - GEOINT
    - OSINT
    - All Source
  + The Diamond Model of Intrusion Analysis
* Intelligence Sources
  + - private industry relies on
      * public
      * commercial
      * in-house
  + Open Source Intelligence
    - trend tracking, info sharing, avoiding fingerprinting
      * passive recon
    - Google Dorking
      * misconfigs revealing server side info to internet
        + ex. site: <target domain or URL> intitle:”index of” “parent directory”
      * use pages cached by engines next to link
    - Internet Registries
      * RIRs
      * DNS
      * NROs
    - DNS
      * nslookup, host, dig
      * interrogate DNS
        + DNS Harvesting
      * replicate DNS server contents via zone transfer to create content snapshot
        + name servers, MX records, hostnames
      * vulnerable because default DNS is designed to accept any request for a full transfer from any host on the network
        + important to restrict leakage to prevent DNS poisoning or spoofing
      * This is initiated by client (secondary DNS server or networked host) 🡪 critical only authorized hosts be allowed to request full transfers
        + via ACLs
      * whois
    - Job Sites
      * email harvesting, networking
      * social engineering
    - Social Media
      * indicators of compromise
      * news, artifacts, intel
      * networking
      * social media profiling
        + all used for phishing
* Proprietary/Closed Source Intelligence
  + - multiple artifacts from multiple sources
    - corroborate open source with closed source
      * reduce confirmation bias
  + Internal Network
    - don’t eschew internal threat data for external data
    - events, DNS, VPNs, firewalls, authentication system logs
      * baseline establishment
  + Classified Data
    - data handling, legal stipulations
    - security screening
  + Traffic Light Protocol (TLP)
    - created by UK NISCC
    - Threat-information sharing between organizations
      * color coded
        + RED
        + AMBER
        + GREEN
        + WHITE
    - not a classification/control scheme
      * does not enforce intellectual property terms or how data is used by recipient
    - suggests responsible sharing of sensitive info to correct audience while protecting sources
* Characteristics of Intelligence Source Data
  + - organizations must map threat intel products to a distinct aspect of their threat profile
      * prioritize data by relevance for actionable, timely, and consistent results
    - Intelligence requires context to be specific and capable of providing recommendations
    - Describe threat clearly and consistently
      * illustrate impact
      * recommend actions to mitigate
  + Timeliness
    - time-related stipulations
    - timely-delivery
  + Relevancy
    - internal network data
      * operational levels
        + audience 🡪 focus, limit distractions
  + Accuracy
    - draw reliable conclusions and pursue recommended courses of actions
    - convey facts, even if operational context limits response
* Confidence Levels
  + - estimative language communicates intel while acknowledging gaps in intel
      * judgements 🡪 high, moderate, low
* Indicator Management
  + - indicators are observable network artifacts with context, descriptions, and insinuations
  + Indicator Lifestyle
    - * discovery 🡪 development 🡪 actionable intel
        + vet, apply, and share intelligence
  + Structured Threat Information Expression
    - STIX MITRE
    - STIX Domain Objects (6) and Relationship Objects (2)
      * show relationship between one SDO and another via SROs
        + stored visually or with JSON
    - Attack Pattern
      * class of tactics, techniques, and procedures (TTPs) 🡪 SDO
      * useful for determining attacks such as phishing
        + insight as to how attacks are executed
    - Campaign
      * collection of malicious actor behaviors against a common target over a finite timeframe
      * attribution methods to point to specific threat actors
        + unique indicators, traits
    - Course of Action
      * preventative or responsive action
    - Identity
      * SDO that represents individuals, organizations, or groups
        + specific or broad
        + pattern derivation
    - Indicator
      * observables
        + specific observables or patterns of observables + contextual data
    - Intrusion Set
      * compilation of behaviors, TTPs, or other properties shared by a single entity
        + ID common resources/behaviors within a timeframe restriction
        + multiple campaigns
    - Malware
      * TTP
      * STIX SDO
    - Observed Data
      * collected from a system or network device
        + standalone or aggregate raw data
    - Report
      * threat actors, malware, methodologies
      * object references and connections
    - Threat Actor
      * SDO – individuals/groups
        + campaigns + intrusion set objects + malware + attack pattern TTP objects
      * sophistication, PII, motives
    - Tool
      * software (legit utilities)
      * detection of legit vs illegitimate/malicious use
    - Vulnerability
      * mistakes in software, flaws to be leveraged
    - Relationship
      * connects SDOs together
      * common associations
        + SRC SDO + Relationship SRO + Target SDO
* Trusted Automated Exchange of Indicator Information
  + - data sharing, structure, and messaging
      * in conjunction with STIX
      * hub, source, peer sharing models
      * collections and channels to facilitate exchange models
* OpenIOC
  + Mandiant Framework
    - TTPs + Indicators into machine-readable format for automation
      * metadata 🡪 indexing
      * References 🡪 context of specific orgs
      * Definitions 🡪 content for analysis
        + Boolean Logic
* Threat Classification
  + - incidents describe any action that results in indirect harm to your system or increases likelihood of unauthorized exposure to sensitive data
      * first understand what normal looks like (baselining)
      * Incident Response is often a compulsory part of business
  + Known Threats versus Unknown threats
    - signature vs anomaly based detection which rely on knowledge of threats
      * only as good as historical data collected
      * limited against unknown/changing threats
    - heuristic analysis can observer what an executable is doing
      * invoked commands
      * file writes
      * attempts at concealment
        + all in a sandbox
    - The absence of evidence is not evidence of adsence
  + Zero Days
    - * significant uptick in usage
      * market for exploits vs market for bug bounties to counter them
    - Preparation for Zero Days
      * technical and operational best practices
        + avoid SPOFs
      * proactive and responsive actions
        + Threat Intel Research
        + Control Development
  + Advanced Persistent Threats
    - stealthy and continuous computer hacking efforts by org/govt with significant resources
      * APTs stand out due to their:
        + support infrastructure
        + TTPs
        + Targets
    - Advanced
      * well equipped, formal training, significant funding
      * coordinated, fully supported
    - Persistent
      * directed, purposeful tasks/behaviors
      * focused, ignoring opportunistic targets
        + strict RoE
        + consistency and persistence
    - Threat
      * capability and intent 🡪 technical implementation of political plans
* Threat Actors
  + Nation-State Threat Actors
    - dedicated infrastructure, training, and operational support behind their actions
    - extensive planning and coordination
      * can invoke private business cooperation/support
      * incorporate false flags
      * perspective between adversaries
  + Hacktivists
    - less support but still coordinated
    - available tools, mass participation
      * social media, defacement tactics
      * rarely stealthy
      * typically target system availability (DoS)
        + counter with HA tactics
        + counter social engineering
  + Organized Crime
    - increasingly common utility
    - theft of intellectual property, personal data
      * to make money
    - moderate sophistication
      * cryptojacking, ransomware, bulk data exfiltration techniques
    - low risk, high reward
  + Insider Threat Actors
    - high risk of catastrophic damage due to privileged access
      * bypass perimeter focused defenses
        + counter with least privilege, ACLs
    - Intentional
      * anomalous activity 🡪 high-volume network activity or indiscriminate file access attempts
    - Unintentional
      * lack of education, negligence, human error
      * be wary of attribution of intent, may hide actual root issue
* Intelligence Cycle
  + - * 5-6 step development cycle that transforms raw data into finished intelligence
        + increased situational awareness and delineation of responsibility

continuous and does not require perfect knowledge

* + Requirements
    - Identification
    - Prioritization
    - Refinement
      * of uncertainties
    - Includes key tasks to planning and direction of overall intelligence effort
      * determined by client operations and term capabilities
        + ID + close gaps
  + Collection
    - execution of plan, data is collected to fill gaps
      * technical collection methods
        + network taps, enhanced device logging
      * open source collection
    - Analysis
      * make sense of what is observed
        + prioritize against known requirements with context
        + produce finished, actionable intelligence with minimized biases, judgements, and uncertainty
    - Discrimination
      * the product provided must be useful
        + intelligence should always be actionable
    - Feedback
      * leads to renewed cycle 🡪 new planning and direction of future collaboration efforts
        + review is essential for improvement and maintenance of quality
* Commodity Malware
  + pervasive malware made available to threat actors via sale
    - focus more on operational optimization vs technical sophistication
  + malware-as-a-service
* Information Sharing and Analysis Communities
  + - lower the barrier to entry
    - standardizing sharing of threat information and storage
  + ISAOs
    - without specific industry alignment
  + ISACs
    - Auto-ISAC
    - A-ISAC
    - NCC
    - E-ISAC
    - MS-ISAC
    - EI-ISAC
    - FS-ISAC
    - H-ISAC
    - IT-ISAC

Chapter 2: Threat Intelligence in Support of Organizational Security

* Levels of Intelligence
  + - as the volume of data that traverses the network increases, so do the alerts and logs that have to be triggered, interpreted, and actioned
      * also increases the chances of false alarms
      * tools may lead to alert fatigue
        + mitigate both with threat intelligence program into all aspects of security operations

adds context to decisions

* + - discover the enemys strategy, develop own plan to confront the enemy, and execute with precision
      * strategic-level thinking 🡪 impair adversaries ability
        + neutralize CoGs and Supply Chain
        + products are anticipatory in nature

comprehensive view of environment, ID key actors, and offer glimpse into future

* + - * operational-level thinking 🡪 application of an orgs cybersec strategy
        + determine what is being defended

from whom

for how long

with what methods

* + - * + determine what efforts must be made

with what resources

* + - * + operational threat intelligence products provide insight into specific environmental conditions

determine how to allocate resources

* + - * tactical-level thinking 🡪 how defenders engage attackers
        + countermeasures and their impacts
        + TTP intel
        + highly actionable products
* Attack Frameworks
  + - add structure when thinking about lifecycle of security incident and actors involved
    - broad conceptual understanding, timelines, and motivations
  + MITRE ATT&CK
    - Cybox, CVE, TAXII, STIX
    - ATT&CK 🡪 TTPs
      * + 🡪 Enterprise, PRE-, Mobile
      * Initial Access
      * Execution
      * Persistence
      * PrivEsc
      * Defense Evasion
      * Credential Access
      * Discovery
      * Lateral Movement
      * Collection
      * C2
      * Exfil
      * Impact
    - prioritization of behaviors to focus on
      * compare and contrast TTPs
  + The Diamond Model of Intrusion Analysis
    - emphasizes relationships and characteristics of four basic components
      * Adversary
      * Capability
      * Victim
      * Infrastructure
    - using vertices as connections between each point, can model how an adversary uses an infrastructures capability against a victim
      * quickly pivot between nodes as defenders during an event
    - Model adjusts as the adversary changes TTPs, infrastructure, and targeting
      * requires more attention to follow this model
    - Defined by seven axioms that impact defenders
  + Kill Chain
    - phase-based model that describes the phases of a kinetic operation
      * pinpoint where along the lifecycle an attack is and deploy appropriate countermeasures
        + understand progression and typical behaviors of each stage
    - Recon 🡪 Weaponization 🡪 Delivery 🡪 Exploitation 🡪 Installation 🡪 C2 🡪 Actions on Objectives
      * Defendable
      * Requires Awareness
    - kill chain may miss social engineering, insider threats, and other forms of non-mlaware base intrusions
* Threat Research
  + - * enrich alerts about known threats as well as assist in novel threat detection
        + benign or not?
        + seen before or not?

if so, what has been said?

* + Reputational
    - scores assigned to URLs, domains, and IP addresses across internet
      * CISCO Talos
      * Google Safe Browsing
      * VirusTotal
  + Behavioral
    - isolation environments, executables in sandboxes
      * Cuckoo Sandbox
      * REMnux
  + Indicator of Compromise
    - need two primary components
      * data and context
    - free IOCs sources
      * CIRCL MISPs
      * FBI InfraGard
      * Pyramid of Pain
        + cost of imposition of adversaries when addressing threat vectors
  + Common Vulnerability Scoring System
    - CVSS – accurate quantitative measurement to better understand weakness impacts
* Threat Modeling Methodologies
  + - procedural approach to thinking like the adversary by crating abstractions of a given system
      * profile development
      * weakness awareness
    - perform early and continuously as input directly into software development life cycle (SDLC)
  + Adverse Capability
    - document types of threat actors that are likely threats
      * intent, capabilities, TTPs
  + Total Attack Surface
    - logical and physical
      * infrastructure, services, server rooms, workstations
    - requires system mapping
      * address each component
      * architectural design considerations
  + Attack Vectors
    - most likely path adversary will take to critical assets
      * ID appropriate security controls to mitigate TTPs
        + apply compensating controls
    - red teaming, TTXs
  + Impact
    - damage potential
      * physical, logical, monetary, reputational
    - communicates risk openly
  + Likelihood
    - possibility of successful exploitation
  + STRIDE
    - evaluates systems design using flow diagrams, system entities, events related to a system
      * suitable for both logical and physical systems
        + spoofing, tampering, repudiation, information disclosure, DoS, PrivEsc
  + PASTA
    - risk-centric threat modeling
    - communicate risks to strategic level decision makers
      * bring technical requirements inline with business objectives
      * solicit input from operations, governance, architecture, and development
* Threat Intelligence Sharing with Supported Functions
  + Incident Response
    - rapidly and accurately addressing potentially wide-ranging issues on a consistent basis
      * speed and accuracy
    - response time and proactive measures
      * rely on playbooks
        + threat intelligence is critical for strong defense development
  + Vulnerability Management
    - making risk-based decisions
      * every system and every asset has vulnerabilities and exposures
    - NIST NVD
      * threat intelligence provides awareness about vulnerabilities in an operational context
        + information security sites
        + social media
        + code repos
        + paste sites

all open source resources

* + - Communicate exploits relevant to organization
      * enables prioritization
        + what is being exploited vs what can be exploited
  + Risk Management
    - management of impact and probability
      * three components
        + capability
        + intent
        + opportunity
    - threat intel can describe these three components to create a defense/response
    - risk 🡪 vulnerability, impact, and threat 🡪 capability, intent, opportunity
      * ID what attacks are becoming more/less prevalent
        + likely assets to be targeted
  + Security Engineering
    - rely on threat intelligence to determine countermeasure effectiveness
  + Detection and Monitoring
    - threat intel applied to SecOps can enrich internal alerts with external info and context necessary
      * critical for triage
      * scope determination
    - automated threat intel

Chapter 3 – Vulnerability Management Activities

* Vulnerability Identification
  + - combination of what we want to do, should do, and have to do
      * requirements that must be satisfied
        + external authorities (laws, regulations)
        + internal authorities (policies, directives)
        + best practices
    - vuln scanning
      * lack exploit capitalization functionality (no advanced correlation)
      * must understand how vuln scan results may be combined with threat intel to being planning on attack
        + requiring context
  + Regulatory Environments
    - * laws, rules, regulations with enforcement and noncompliance procedures
    - ISO/IEC 27001 Standard
      * ISMS
      * voluntary security standard concerning development and maintenance of good infosec
        + A.12.6.1 – Vulnerability Management
      * Three Stage Certification
        + desk-side audit
        + implementation audit
        + surveillance audit
    - PCI DSS
      * Requirement 11
        + obligation to regularly test security systems and processes
        + Section 2 – requirements for Vuln Scans
      * Internal and External Scans (ASVs)
      * Require rapid resolution of high-risk discoveries
    - HIPAA
      * safeguard PHI
        + Section 164.308(a)(1)(i)

requires accurate and thorough vulnerability assessments and implementation of security measures

* + Corporate Security Policy
    - * overall general statement produced by senior management that dictates what role security plays within the organization
        + issue-specific (functional)
        + system-specific
      * Issue-specific is common with vuln management and nested within system specific policies
      * must understand organizational security context within which said process occurs
  + Data Classification
    - attached to all metadata to determine protective controls
      * gauge the resources that should go toward protecting each type of data
        + private
        + confidential
        + proprietary
        + public
    - each classification should be unique and separate without overlapping effects
      * should also outline how info is controlled and handled throughout its lifecycle
        + level of damage that could be caused if the data were disclosed
        + level of damage that could be caused if the data were modified or corrupted
        + lost costs incurred if the data is not available or corrupted
        + legal, regulatory, contractual responsibility to protect the data
        + effects on security and age of data
  + Asset Inventory
    - * critical to managing vulns in infosystems
        + central to Center for Internet Security list of Critical Security Controls

inventory of authorized and unauthorized devices

software on said devices

* + - Servers
      * large risk on losing track of a servers purpose on the network and running unnecessary services and open ports
      * misconfiguration of services, unnecessary bonus features
        + must be aware of devices full capabilities
    - Endpoints
      * most common entry point for attackers into networks
        + typically through email and web links
        + lack of up-to-date configs
      * baseline configs help balance functionality with security
    - Critical Assets
      * anything essential to performing the primary functions of your organization
        + require higher degree of attention

focus on thoroughness and frequency of vuln scans

* + - Noncritical Assets
      * still need to include assets in vuln management plan
  + Active vs Passive Scanning
    - * + war driving/dialing
        + host scanning
      * network mapping
      * host/port scanners
      * web app vuln scanners
    - Mapping/Enumeration
      * network mapping 🡪 network topology (perimeters, DMZs, key devices)
      * Topology Discovery
        + ping sweeps
    - Port Scanning
      * probe hosts/servers
        + service discovery
        + OS fingerprinting
    - Web App Vuln Scanning
      * common vuln testing
        + SQLi, Command Inj, XSS, Server Configs
      * important to keep scanner DBs updated
* Scanning Parameters and Criteria
  + - vuln scan process itself is critical
      * preparation and planning
      * frequency, limitations, relevant scope
  + Risks Associated with Scanning Activities
    - risk appetite 🡪 willing to assume and diminishing returns
      * common sense and inexpensive controls
        + expense of risk vs expense of control
        + determined via Risk Deliberation Process
  + Regulatory Requirements
    - PCI DSS Req 11.2 🡪 quarterly and significant changes
    - HIPAA 🡪 none/semiannually
  + Technical Constraints
    - personnel, time, bandwidth, hardware/software, qualifications, capacity
  + Workflow
    - established workflows, personnel availability
    - standardization and enforcement, consistency
  + Sensitivity Level
    - determine configuration and calibration of tools
      * prevent accidental leaks/damage caused by scans
      * prevent disruption of operations
  + Vulnerability Feed
    - frequency of updates should match internal scanning schedule
    - additional RSS feeds
      * NVD
      * NIST
    - determine criticality for out-of-cycle tests
  + Scope
    - series of targeted scans versus whole system scan at once
      * what to include in-scope and what not to
  + Non-Credentialed vs Credentialed
    - perspectives (inside vs outside)
      * pros and cons on completeness vs network load vs realism vs security
        + avoid privileged accounts when possible
  + Server-Based vs Agent-Based
    - Agent-Based 🡪 scans require a running process on every scanned device
    - Server-Based 🡪 Agentless 🡪 consolidates all data and processes on one or small number of scanning hosts
      * agents use less bandwidth and can include mobile devices
  + Internal vs External
    - internal leverages a privileged position whereas external reflects what an attacker is likely to see
      * can be combined to conserve resources
  + Types of Data
    - different requirements with influence on what data can be collected
    - audience of report as well
  + Tools, Updates, and Plug-ins
    - maintain current DBs, lists, plugins
      * Nessus NASL
  + SCAP
    - Security Content Automation Protocol
      * specific standards for the assessment and reporting of vulnerabilities in the Infosys of an organization
        + standardizes ARF, CVE, CVSS
    - NIST Baselines to define minimum standards
  + Special Considerations
    - permissions, network access
    - account 🡪 minimally privileged
    - ACL and Firewall modification
    - NIDS + NIPS awareness
    - Reporting
* Intrusion Prevention System, Intrusion Detection System, and Firewall Settings
  + - * firewalls, IDS/IPs can be issue to detecting weaknesses during vuln scans
        + may block scanners

publish schedule to address

* + Generating Reports
    - critical for vuln management
      * what trends
      * how to automate
      * disseminating timely
    - Automated vs Manual Distribution
      * templates expedite process
        + appropriate info to appropriate audiences
  + Validation
    - analyst review to determine validity and policy exceptions
      * most accurate network info to prioritize responses
    - True Positives
      * verify results
        + determine solutions and ability to apply them
        + group policy, auto remediation
      * Compare to Best Practices or Compliance
        + benchmarks

DISA STIG, NSA, DoD

technical steps to harden networks, devices, endpoints

* + - * Reconcile Results
        + thorough notes

config steps, config validation, operation verification, vuln tests

continuity

* + - * Review Related Logs and/or Other Data Sources
        + compare authorized services with what is actually running

verify logging mechanism

notice changes

determine if vulns are addressed

SIEM use

* + - * Determine Trends
        + track vuln changes over time

context

solution efficacy

trouble ticket software

visibility

* + - False Positives
      * scales in difficulty and impact
      * customized follow-up test to confirm
    - True Negatives
      * hard (impossible) to totally prove
    - False Negatives
      * Type II
      * lack of detection capability, too new, incorrect scan settings
      * worst options
  + Remediation
    - * rapid response
      * continuous examination and remedy confidentially
        + compare consecutive scans
        + document
    - Patching
      * Patch Tuesday and Exploit Wednesday
      * Necessary Evil
    - Prioritizing
      * both alerts and response
        + capabilities and goals
      * Criticality
        + economics (time, money, people)
        + color-coding
    - Hardening
      * requires constant monitoring as both network and local resources are dynamic
      * avoid relying solely on attack detection
      * balance security and usability
        + privileges
        + minimum services necessary
        + patch/update management
  + Compensating Controls
    - achieve goals of a security requirement even if they were unable to meet goals explicitly
      * meet “intent and rigor”
  + Risk Acceptance
    - not ignored, but accepted
      * inventory known open vulns regularly
        + changing priorities over time
  + Verification of Mitigation
    - formal procedure for verification and validation of controls used
      * correctly implemented and actually effective
* Inhibitors to Remediation
  + Memorandum of Understanding
    - clarifies vagaries and how to proceed in a way that is satisfactory to everyone involved
  + Service Level Agreement
    - exist within or between companies
    - roles and responsibilities, and limits
      * remediation must be explicitly stated
  + Organizational Governance
    - corporate governance
      * an organization uses to direct and control its operations
        + balance between competing priorities which may interrupt remediation
    - communication is necessary
      * timely decision-making
  + Business Process Interruption
    - optimization increases risk of disruption
    - unpredictability/instability leads to production system update delays
  + Legacy and Proprietary Systems
    - require constant network monitoring, compensating controls, and multiple layered security mechanisms
    - too big, old, complicated, esoteric
* Ongoing Scanning and Continuous Monitoring
  + automated vuln scans
  + rapid remediation
  + need for scalability and speed

Chapter 4 – Vulnerability Assessment Tools

* Web Application Scanners
  + - understanding why vulnerabilities exist and how they can be exploited can assist in understanding a vuln report
    - typically external scans
    - only target vulns used by/known by plugins
      * outdated server components
      * misconfigured servers
      * secure authentication of users
      * secure session management
      * info leaks
      * XSS vulns, improper HTTPS use
  + OWASP ZAP
  + BurpSuite
  + Nikto
    - web server vuln scanner
      * XSS, SQLi, CMDi, improper server configs
      * not practical for detailed analysis
        + quickly confirms host status
      * very speedy, but narrow focus
        + technical reporting
  + Arachni
    - ruby-based, modular web scanner
    - scans run in parallel
    - learns and self-trains through feedback and meta-analysis to minimize false positives
* Infrastructure Vulnerability Scanners
  + - internal and external perspectives
      * local agents or agentless
  + Nessus
    - 80k plugins in NASL
      * vuln ID, misconfig detection, default password exposure, and compliance determination
    - client runs on web interface and communicates with Nessus server
      * <http://localhost:8834>
    - safe checks option avoids launching potentially disruptive plugins
      * compliance checks
      * custom config files (audits)
    - invasive scans may temporarily reduce functionality
      * all finds are assigned a severity level
  + OpenVAS
    - open source fork of Nessus
      * 47k NVTs
      * handled by a manager
    - <http://localhost:9392>
  + Qualys
    - QualysGuard
    - cloud-based SaaS
      * accessed via web-UI
        + detailed report templates
* Software Assessment Tools and Techniques
  + - static analysis
    - dynamic analysis
    - reverse engineering
    - fuzzing
  + Static analysis
    - can be automated
    - manual is code review
    - “Lapse” tool 🡪 OWASP
      * scalable
    - cannot reveal logic errors
      * combine static analysis with code review
  + Dynamic Analysis
    - sandboxing
      * requests are documented
    - faster, less expertise
      * potentially limited scope
  + Reverse Engineering
    - disassembly and decompiling
    - assembly, compiling
  + Fuzzing
    - sends large amounts of malformed, unexpected, or random data to trigger failures
      * buffer overflows
      * DoS
      * injection flaws, validation flaws
    - American Fuzzy Lop (AFL)
    - often handled with observation and analysis tools
    - Untidy
      * XML fuzzer
        + web application clients and servers
    - Peach Fuzzer
      * fuzzer suite using XML pits that are configurable
    - Microsoft SDL Fuzzers
      * Mini-Fuzz File Fuzzer
      * Regex Fuzzer
* Enumeration Tools and Techniques
  + - network enumeration interrogates a set of hosts
      * horizontal scan 🡪 messages to locate host activity
        + hosts responding to active requests and their corresponding addresses
      * vertical scan 🡪 messages to a set of protocol/port combos to determine what ports are listening
  + NMAP
    - cli + GUI 🡪 Zenmap (W), NmapFE (L), Xnmap (M)
    - precise but limited scope
    - successive, identical scans can help detect changes and inventory assets
  + hping
    - crafts custom packets to assist in detecting network flaws or deliver exploits
      * also tracert and IP fragmentation
  + Passive vs Active Enumeration Techniques
    - * Passive
        + whois
        + nslookup
        + dnsrecon
      * Active
        + port scanning
    - nslookup
      * DNS + IP addr + FQDN
      * MX + CNAME retrieval
    - responder
      * LLMNR/NetBIOS Name Service
        + query LAN to determine if host is unable to resolve hostname via DNS

any host that responds is authoritative if both fail

* + - * responder poisons name services to gather hashes and creds from a LAN
* Wireless Assessment Tools
  + - WLAN Auditing and Analysis
      * WAP is typically master and client is managed
        + Infrastructure Mode manages all aspects of WLAN config
        + Mesh Mode allows for direct device comms

Ad Hoc

* + - * + Monitor Mode

interface sees all available WLANs and characteristics without connecting to any of them

necessary for audit

WLAN Analyzers like Kismet performs this automatically

important to know layout of wireless network

WAPs

Clients

keep track of what normal looks like

record all settings of WAPs

detect rogue Aps or unauthorized clients

need WPA-E or 802.1X

* + Aircrack-ng
    - indiscriminate wireless monitoring
  + Reaver
    - takes advantage of WPS protocol and PIN use
  + oclHashcat
    - GPU optimized version of hashcat
* Cloud Infrastructure Assessment Tools
  + Scout Suite
    - open source auditing tool for verification, auditing, and managed services
      * cloud assets and how they interact with platform API
  + Prowler
    - similar to ScoutSuite, uses CIS best practices for configuring security options for AWS
  + Pacu
    - AWS exploitation framework
    - open source and modular
      * meant for pentesting and compliance
        + RhinoLab Github

Chapter 5 – Threats and Vulnerabilities Associated with Specialized Technology

* Access Points
  + - cutting of usual means of exploitation is typically sufficient, unless specifically targeted
      * move onto other lower hanging fruit
    - common means of exploitation
      * missing patches/updates
        + sometimes for a legit reason

should still be noted and tracked and compensated for

* + - * misconfigured firewall rules
      * weak passwords
  + WAPs
    - BYOB challenges to security
    - Rogue Aps
      * counter with 802.1x NACs
* Virtual Private Networks
  + segregates traffic from underlaying network
    - IPSec L2TP
    - TLS
    - DTLS (Cisco)
  + potentially allow untrusted, unpatched, infected hosts
    - BYOB devices over VPN increases chances
    - patched/update state
      * NACs mitigate by quarantining and determining if a device is being monitored/compromised
* Mobile Devices
  + - greater density of stored personal info
      * three chief threat vectors
        + network
        + device
        + application
      * always assume a device will be stolen/lost eventually
  + Network Vulnerabilities
    - flaws of telecom backbone
    - flaws of devices comms networks
  + Device Vulnerabilities
    - entire functionality, context, and associated personal user data at risk
      * force charge spillage in DRAM in a predictable way
        + Rowhammer Technique
  + Operating System Vulnerabilities
    - window of vulnerability following patch/vuln releases
  + App Vulnerabilities
    - user security/privacy issues occur at app level
      * security secondary to functionality
    - Improper Platform Usage
      * shortcuts in development cycle
        + incorrectly crafted API calls or incorrect implementation of builtin security controls
    - Insecure Data Storage
      * improper data handling, storage, transfer fails to ensure confidentiality and integrity
        + world readable files, improper logging, surreptitiously collected analytics data
    - Insecure Authentication
      * locally stored passwords/secrets without security
        + authentication bypass
      * apps that have lax password policies
    - Insecure Authorization
      * affects the mechanism that ensures a user gains access to the correct resources
        + verifies roles and permissions only through back-end systems that are not user accessible
      * authorization vulns can also be authentication vulns
        + resource requests with out-of-order authentication
    - Code Quality Vulnerabilities
      * + code reuse, negligence to correct known errors, risk acceptance
        + modification 🡪 binary level
        + third-party library dependence
      * mitigate with manual secure code review and designed to detect runtime code manipulation to ensure correct error reaction and prevent malicious injection
* Internet of Things
  + - increased attack surface
      * essential to understand what is connected to what
        + patch management can be difficult
        + default/static logins

botnets

bulk exploitations

* + Mirai Botnet
* Embedded Devices
  + - written instructions into firmware/hardware
      * special challenge to proper monitoring
  + Real-Time Operating Systems
    - low-latency responses to input that excel at scheduling tasks to avoid delays
* System on a Chip
  + single integrated circuit and processor
  + similar to microcontrollers, higher likelihood of system-wide impact due to density
    - critical to ensure hardware verification is performed alongside any software
* Field Programmable Gate Array
  + - programmable chip that allows hardware itself to be configured to accommodate new software functionality
      * targeted by attacks that disrupt functionality at level its written
        + HDLanguage

target for malware injection

* Physical Access Control
  + - rely on software/hardware combo and vulns lay between hardware and presentation layers
      * disassembly can reveal weaknesses
      * susceptible to replay or cloning spoofing attacks
        + emulates captured signals
* Connected Vehicles
  + CANBus
    - Controller Area Network Bus
      * defines how independent components of vehicular control systems communicate with one another without a central controller
        + no protections for transmitted comms

MiTM

* Drones
  + - low-cost surveillance, signal-collection, tracking, ID functionality
  + Hardware Security
    - physical access is the best kind of access
      * prevent devices from getting into the wrong kind of hands
  + Communication Channel Security
    - unencrypted, open ports
    - subject to DoS
  + Web Portal Security
    - loss of control of devices following web portal compromise
      * no authen/author needed
    - dependency on 3rd parties
* Industrial Control Systems
  + - RTUs, PLCs 🡪 Level 1
    - HMIs 🡪 Level 2
  + traditionally separated from traditional L3/L4 IT
    - now commonly bridged
  + burned-in firmware limits patching mechanisms and management
    - weak passwords as well, default passwords
      * often static
* SCADA Devices
  + - ICS that can monitor and control devices throughout large geographic regions
    - two additional vulnerabilities
      * long-distance comms links
      * reliance on isolated and unattended facilities
  + Modbus
    - similar to CAN
      * de facto PLC standard
      * no built-in security
* Process Automation Systems
  + - workflow automation systems
      * automate day-to-day business practices
        + suffer when processes are complex
* Review
  + understand specialized tech, map attack surfaces and ID attack vectors

Chapter 6 – Threats and Vulnerabilities Associated with Operating in the Cloud

* Cloud Service Models
  + - ease of provisioning, setup, and teardown
    - shared responsibility model of security
      * SaaS
      * PaaS
      * IaaS
  + Shared Responsibility Model
    - * company is always responsible for the security of deployments and associated data
        + trust but verify
        + manage/secure risks in a manner that compliments cloud provider security
  + Software as a Service
    - most commonly 🡪 access applications and functionality directly from a service provider
      * highest level of technology stack
    - hardware provision, user experience
      * must understand how these services protect company data at every level
        + burden on costumer to design/enforce protective controls at data-handling level

hard to maintain visibility on what is being used

* + - Complexity Identity and Access Management (IAM)
      * friction between security and usability
        + IAM Services 🡪 set access rights and centralize auditing reports

hard to understand and manage

* + - once data has left the system, it is impossible to bring it back
      * no technical controls 🡪 need layered approach between policy and technical
        + DLP and Encryption
        + Auditing and Compliance
    - SECaaS
  + Platform as a Service
    - * offers direct access to a development environment to enable organizations to build their own solutions without need for own development infrastructure
        + value on software development

source code

* + - * role-based controls
      * auditing and account management tools
        + protect admin access
        + physical infrastructure is managed by provider

maintenance and security

physical, monitoring, DDoS prevention

STRIDE assessment between provider and organization

* + Infrastructure as a Service
    - effective and affordable way of managing own hardware without overhead
    - PaaS and SaaS vulns may also exist in IaaS as well as unique IaaS vulns
      * hardware issues/attacks
      * misconfigs client-side
* Cloud Deployment Models
  + - public cloud 🡪 possibility of spillage, questions of data permeance, post-deletion remainder
  + Public
    - vendor owns server, infrastructure, network, and hypervisor
      * organization borrows portion of shared infra
  + Private
    - organization owns hardware and underlying services
      * wholly responsible for operation and upkeep of network, infrastructure, hypervisors, systems/devices, configs, ID and access management, data
  + Community
    - infrastructure shared across organizations with a common interest in how data is stored and processed
      * typically some regulatory environments
        + operational consistency and lower cost
        + more complex

vulns often process and policy related

* + Hybrid
    - on-prem infra with public cloud
      * focus on how data/apps use solutions to achieve goals
        + public failover for private infra
* Serverless Architecture
  + - services without infra setup required
    - strict functionality focus
      * mass scaling and HA
    - harder to secure
      * app and function level only
  + Function as a Service
    - developer focus on action and scalability (like PaaS)
      * code run by provider as needed (as long as code is compat with provider infra)
        + event-driven, serverless computing
* Infrastructure as Code
  + - provision hardware and system configs automatically, granularity in a repeatable manner in human and machine-readable code
      * minimize environment drift during development
        + predefined environments
        + continuous integration/delivery (CI/CD)

automated SDLC

* + - Imperative Approach 🡪 explicit instructioning
    - Declarative Approach 🡪 implicit instructioning
  + ensure consistent environments with up-to-date patching
    - spin-up vulns 🡪 insecure service calls; insecure service qualities
* Insecure Application Programming Interface
  + - APIs are exposed – used for provisioning, management, orchestration, and monitoring
      * API and credentials must be protected
      * Secure Web Interfaces
        + API Security Project
  + Broken Object Level Authorization
    - attackers access resources without proper auth
      * object-level auth check should always be implemented and access granted based on specific user role
      * randomized UUIDs
  + Broken User Authentication
    - mandate authen to all APIs
    - short-lived tokens
    - mandate 2FA, rate-limiting, and password strength policy
  + Excessive Data Exposure
    - prevent objects from returning their full contents and relying on user filtering
      * specific as possible
      * complete as possible
      * monitor response data
        + separate API calls for special data
  + Lack of Resources and Rate Limiting
    - throttling to prevent resource exhaustion
    - payload limitations, pagination use
  + Broken Function Level Authorization
    - restrictive default deny approach to access and apply permissions to roles and groups during evaluation
  + Mass Assignment
    - field guessing
    - server-side variable changing
      * validate inputs and restricting users to read-only
  + Security Misconfiguration
    - default configs, open access, open cloud storage, misconfigured headers, verbose error messages
      * use automated hardening tools and checklists
      * minimize features
  + Injection
    - SQL, LDAP, OS commands, XML
      * input validation, sanitization, and filtering
      * API output limiting
  + Improper Asset Management
    - deprecated, testing, staging API environment that have been forgotten or gone unhardened
      * inventory all API endpoints
        + retire unused ones
        + limit access to training/testing resources
  + Insufficient Logging and Monitoring
    - visibility is key in detecting API usage deviation
      * log failed attempts, denied access, input validation failures, etc
        + relevant security events
        + formatted and normalized
        + protected from tampering
* Improper Key Management
  + - reliable encryption
      * decentralized clouds make key management difficult
        + vendor/provider-based; HSM use
      * encrypt data at customer interface
        + upfront costs and increased computational load
      * risk of unauthorized log access
* Unprotected Storage
  + server misconfigs, altered permissions, flaws in business processes/practices
    - full visibility into real-time state cloud resources and the ability to remediate quickly
* Logging and Monitoring
  + application stack (IaaS, SaaS, PaaS) will modulate what is visible to an organization
    - lose of visibility 🡪 loss of monitoring
  + policy and technical controls to ID and authenticate users correctly
    - assign correct access
    - create/enforce access controls

Chapter 7 – Mitigating Controls for Attacks and Software Vulnerabilities

* Attack Types
  + - proactive security habits in addition to reactive procedures
  + Injection Attacks
    - RCE
    - XML Attacks
      * XML Parser Manipulation
      * XML bombs and XXE attacks
    - SQLi
    - XSS
      * persistent
      * nonpersistent
      * DOM-based
        + impossible to detect server-side
    - Directory Traversal
      * poorly configured permissions
        + unix-compliant directory traversal character sequences or encoded variation
  + Buffer Overflow Attacks
    - disrupt delicate arrangement of elements in memory space
      * buffer overflow exceeds capacity of the buffer
        + memory written to other memory spaces

stack-based 🡪 LIFO

heap-based 🡪 dynamic, random

* + - Stack-Based Attacks
      * most common
    - Heap-Based Attacks
      * difficult to implement and dynamically allocated
        + exhaust memory space for a specific program

likely to corrupt as to exploit

* + - Integer Attacks
      * value submitted is larger than 4-byte limit
      * difficult to detect but can lead to other exploits
  + Privilege Escalation
    - Vertical
    - Horizontal
  + Authentication Attacks
    - * fabricate info used by system to authen a user
    - Password Spraying
      * one password against multiple systems
        + avoid single system acct lockout
        + detectable with SIEM, MFA, password policies
    - Credential Stuffing
      * stolen creds used against other systems, often via automated processes
        + counter with MFA
    - Impersonation
      * service impersonation for harvesting/interception
        + server key theft
        + CA impersonation
    - MiTM
      * two-way impersonation
        + proxy-relay
        + even encrypted connections
    - Session Hijacking
      * stealing and replaying valid session info
        + traffic capture, MiTM, session token prediction
    - Rootkits
      * kernel-level
* Vulnerabilities
  + - how rules are written, how a mechanism is implemented, or what functions are called
      * cause unexpected/undesirable behavior
      * counter with secure coding practices and thoughtful review
  + Improper Error Handling
    - * too much info disclosed to outside users
        + reveal details about internal processes, directory structure

policy on error handling should be documented

* + Dereferencing
    - null point dereference occurs when software attempts to access a value in memory that does not exist
      * immediate crash/instability
  + Insecure Object Reference
    - reveal info/pattern in underlaying or back-end technologies
      * avoid by using indirect reference maps, using random values
      * enforce access controls via server-side validations
  + Race Conditions
    - creates an unstable quality in the operation of a program arising from timing variables
      * time of check to time of use (TOCTOU) is common attack vector
    - bypass ACLs
  + Sensitive Data Exposure
    - * cleartext transmission
      * outdated encryption
      * default crypto keys
      * enforced encryption
      * proper server cert verifications
    - follow baseline countermeasures
      * ID data and sensitivity levels
      * collect and store only minimum amount of data needed to fulfill a business requirement
        + strong encryption, careful key storage

TLS, PFS, HSTS

* + Insecure Component
    - overreliance on 3rd party and open source software
      * need to know what is being used is legit and secure
  + Insufficient Logging and Monitoring
    - need logging at key points throughout network
      * networked devices, OS’s, hosts
    - ingest, understand, and operationalize easily
      * input validation failures
      * output validation failures
      * authentications
      * access control events
      * session management failures
      * connectivity, malware detection
      * config mods, startups/shutdowns
      * admin account actions
    - ensure logging compliance with laws and regulations
  + Weak or Default Configurations
  + Use of Insecure Functions
    - implementation/inherent qualities
      * gets(), strcpy(), strncpy()
        + leads to overflow
* Chapter Review
  + access controls
  + security appliances/software
  + proxies, filters
  + code review and secure design
  + logging, detection, response

PART II

Chapter 8 – Security Solutions for Infrastructure Management

* Cloud vs On-Premises Solutions
  + - lack of awareness of network topology and shift from physical to virtual perimeters
      * prioritize knowledge to most relevant
        + architecture, asset and change management, ID/authen/author system uses
    - critical security controls are different between the two
    - similar challenges
      * misconfigs
      * encryption
      * insecure usage
    - unique challenges
      * ICS
      * highly classified data
      * sunk costs
* Network Architecture
  + - nodes on a computer network and the manner in which they are connected to another
      * often illustrated, by subnet
      * determines how things should be
      * hybrid constructs on physical, logical, virtual, and cloud
  + Physical Network
    - determine where traffic flows, develop policies and ACLs accordingly
    - limited in flexibility
  + Software-Defined Network
    - software defines how to best route data (control layer) and actually moving data (data layer)
      * more dynamic than physical architecture
    - SDN decouples data forwarding from decision-making functions
      * allow holistic and adaptive control
  + Virtual Private Cloud Network
    - * shared resources 🡪 risk
        + VPC privatized resources within a public cloud environment

create, isolate physical/SDN environment

* + - * + owned by enterprise in private data center (diff from private cloud)
  + Virtual Private Network
    - * secure tunnel
        + persistently or on demand
        + traffic goes through same controls that protect onsite users
      * can act as bottlenecks
        + counter with VPN split tunnel (only some traffic)
        + custom traffic rules
  + Serverless Network
    - single clients requiring specific cloud-resource functions (FaaS)
      * thick local clients that communicate via serverless architecture
      * broken down into functions hosted by a CSP
* Virtualization
  + - creation/utilization of varied instances of operating systems and applications on an ad hoc basis
  + Hypervisors
    - manage resources across multiple virtual instances
      * Type 1 🡪 bare-metal
      * Type 2 🡪 within present OS
  + Virtual Desktop Infrastructure
    - VDI 🡪 separates physical devices that users are touching from systems hosting desktops/apps/data
    - thin clients
      * RDP
      * DaaS
    - useful in regulated environments, mobility
      * data retention
      * config management
      * incident response
      * reduce downtime
        + dependent on network connectivity
  + Containerization
    - sit on top of OS’s and share resources of host OS
      * abstracts kernel of OS for the applications running above
        + whole VM isn’t needed for one service
      * sandbox environment, solely API interaction
* Network Segmentation
  + - subnetworks with switches and routers
    - VPCs and security groups (cloud)
    - can be implemented at physical layer and all the way up to application layer
  + VLANs
    - network switches apply tags to each from received at a port
      * VLAN ID based on sender endpoint
        + determines which port frame is to be forwarded to
        + dest port removes tag and then sent to dest host
    - segmenting a network at one layer doesn’t carry over to higher layers
      * VLANs by themselves are insufficient
  + Physical Segmentation
    - firewalls, traditional switching
    - simple, static, difficult to reconfig
  + Jump Boxes
    - jumping off points for external users to access parts of a network
      * chokepoint for outside users who want access to a protected network
    - high level logging
    - between firewall and network
      * disable unnecessary services, strict ACL use, patch management, MFA
  + System Isolation
    - policies on network devices in addition to segmentation plan, ACLs
    - air gaps
* Honeypots and Honeynets
  + - virtualization has made this process much easier
      * prolonged interaction within these systems indicates compromise
        + less likely to be a false positive
    - intense logging and monitoring
* Asset Management
  + - assets exist within a given context
      * created/used at a particular point for a specific process/purpose
    - track asset lifecycle
      * creation/acquisition 🡪 use 🡪 deletion
      * ITAM
        + determines location, configuration, and users of each other
  + Asset Inventory
    - critical to managing vulns
      * CIS Control 1 – inventory and control of authorized and unauthorized hardware assets
      * CIS Control 2 – software running on said devices
    - hardware and relevant software
      * licenses, cloud assets/data
    - discover continuously and track changes
  + Asset Tagging
    - UID on assets 🡪 part of acquisition/creation process
      * continuously updated
      * automated controls
        + barcodes, QR code, RFID

geofencing with RFID

* Change Management
  + - right changes at the right time to ensure HA and efficient
      * ID – Analyze – Implement changes to address both business and security requirements
    - three common key roles
      * requestor
      * review board
      * owner
    - routine change requests should be addressed regularly
      * emergency procedures in place
        + validate and analyze risk to consider controls and costs
* Identity and Access Management
  + - prove ID and rights to access managed resources
      * must track activities afterward for accountability
    - ID and Authentication = credentials, which are checked for authorization
    - IAM 🡪 user account management, access control, credential management, SSO, rights and permissions management, auditing and monitoring
      * create and manage digital identity lifecycles
  + Privilege Management
    - authorization creep
      * priv management is a critical security function
      * least privilege
    - prevented with regular review
      * minimize priv accts
      * ensure each acct is unique
      * minimize duration of elevated privileges
      * accurate and update account inventory
      * monitor and log all privileged actions
      * enforce MFA
    - ensure proper deprovisioning occurs as part of standard change management process
  + MFA
  + SSO
    - minimize password management, increased efficiency
      * Google SAML – user, ID Provider, Service Provider
      * access controlled via decision-making
    - requires greater focus on protecting user credentials
      * MFA and context
      * centralized authen server becomes critical asset
  + Identity Federation
    - federated identity manager is the only source that requires authen
    - OpenID
      * open standard for SAML-like functions
      * creds maintained by 3rd party
        + offload need for authen system
      * end user 🡪 relaying part 🡪 OpenID Provider
  + Role-Based Access Control
    - * role or group 🡪 implicit
        + no need for an explicit denial to a resource for a given role
        + minimize number of roles
  + Attribute-Based Access Control
    - can belong to subjects, objects, actions, contexts
      * subjects – clearance, position, dept, exp, certs, location, membership
      * objects – classification, file relevance, HR, location, security
      * actions – review, approve, comment, archive, configure, restart
      * context – time of day, project status, fiscal year, audits
    - most granular but can lead to policy-bloat
  + Mandatory Access Control
    - additional levels of scrutiny
      * requires explicit authorization
      * additional levels and security labels
  + Manual Review
    - review both successful and unsuccessful login attempts
    - combine actions and context to determine trust
* Cloud Access Security Broker
  + - track multiple different cloud services that are in use by an organization
      * CASB 🡪 sits between users and cloud services to
        + monitor all activity
        + enforce policies
        + alerts
    - Four Pillars
      * Visibility – who is doing what, where
      * Threat Protection – detect/block malware, threats (IPS/IDS)
      * Compliance – regulated environment ops
      * Data Security
* Monitoring and Logging
  + - ensure Infosys remains available and operating within expected performance
    - Real User Monitoring (RTM)
      * passive monitoring using agents to capture metrics from users perspective
        + produces noisy data
        + req back-end analysis predictability and regularly
    - Synthetic Interactions with systems while under observation
      * precise but limited in complexity
    - Logging helps determine what is out of place
      * what log data is generated, where it is sent, how it is utilized
    - post-collection, must normalize raw data into a common format for analysis
      * pattern recognition
      * classification
      * predictive analytics
* Encryption
  + - avoid unauthorized disclosure over unsecure channels using cryptosystems
      * algorithms and keys
  + Symmetric Cryptography
    - fast but risky
      * key distribution problem
  + Asymmetric Cryptography
    - digital signatures
      * comparison between public and private key to ensure accountability and non-repudiation
  + Asymmetric vs Symmetric Cryptography
    - key length and decryption time
      * Symm 🡪 256 bits, random, faster
      * Asymm 🡪 2048 bits, not random
    - Asymm to set up encrypted comms, symm to perform encrypted comms
* Certificate Management
  + - addresses concern over authenticity of public keys
      * web of trust 🡪 PGP
      * CAs 🡪 digital certs, X.509 (IETF RFC 5280)
        + validity period
      * CRLs maintained by Ras
* Active Defense
  + adaptive measures aimed at increasing effort required to compromise systems
    - Moving Target Defense (MTD)
      * frequently change attack surface
    - Honeypots/Nets
* Chapter Review
  + asset management
  + change management
  + identity and access management

Chapter 9 – Software Assurance Best Practices

* Platforms and Software Architectures
  + - consider three fundamental tasks any non-trivial application must do
      * interact with another part to give/receive data
      * do something useful with data
      * store data
    - Presentation Layer
      * Business Logic
        + Data

Files

DBs

* + - If all three are done in one platform
      * standalone application
        + increasingly rare
    - vs. Distributed Application
      * most popular
  + Client/Server
    - distributed application system
      * client makes requests
      * server receives them
        + n-tier client/server systems
        + two-tier architecture
  + Web Application
    - specific type of client/server system
      * require the most focus in secure development
        + OWASP Top Ten
  + Mobile
    - OS-specific coded apps
    - typically distributed systems
    - authentication and access control on mobile is not as robust
  + Embedded
    - self-contained computer system designed with specificity in mind
      * built around microcontrollers
    - must ensure security of software in them as network connectivity is increasingly common
      * full audit if possible, ensure all data traffic is encrypted at minimum
    - difficult to update and patch securely
  + System on a Chip
    - single integrated circuit
      * fast, low power consumption
      * lack of modularity negates repair option
  + Firmware
    - read-only, non-volatile memory storage
    - first stage in complex systems, only stage in simpler ones
    - some ROM is truly read-only, preventing updates without physical replacement
      * EEPROM has an update/upgrade functionality
        + addresses vulns due to patch management issues
        + otherwise isolate or replace
* Service-Oriented Architecture
  + - focus on functionality/services provided and not care where it is hosted 🡪 SOA
    - SOA 🡪 interconnected but self contained components that communicate with each other and clients via standardized protocols (APIs)
      * each corresponds to specific functions
      * Three Characteristics
        + self-contained components
        + standardized protocols 🡪 SOAP, REST
        + components that implement business functions
  + Simple Object Access Protocol
    - XML over HTTP to enable clients to invoke processes on a remote host without platform
      * Linux and Windows can work together
    - Three Main Components
      * message envelope
      * encoding rules to define data types
      * remote procedure conventions to interpret responses
    - Envelop allows requestor to describe actions expected from nodes
      * enables fine control of access and recovery from failures
        + SOAP is complex and cumbersome
  + Representational State Transfers
    - very common, leverage statelessness and standard ops of HTTP for performance and reliability
      * standard unit is a resource (URI)
      * must implement client/server architecture
      * stateless
      * responses indicate if they are/aren’t cacheable
      * must have unfirm interface
  + Microservices
    - small, decentralized, individually deployable services built around business capabilities
      * loosely coupled, minimizing dependencies between services
        + quick to test, develop, deploy
    - requires log aggregation to properly track malicious behaviors
  + Security Assertion Markup Language
    - specific SOA implementation for authentication and authorization
      * SSO
    - defines Service Provider (SP) and Identity Provider (IP) roles
      * user 🡪 SP 🡪 IdP 🡪 SP (SAML Assertion) 🡪 control decisions
* The Software Development Lifecycle
  + - ID/track needs
    - design, build, and test
    - produce
    - support
    - retire safely
  + Requirements
    - ID and track
      * functional – what
      * nonfunctional – how
        + main input to software architecture
      * security
  + Development
    - design nonfunctional architecture to support fulfillment of functional requirements
    - testing code to ensure security
    - unit testing
    - integration testing
    - system testing
    - acceptance testing
  + Implementation
    - friction between development and operations teams if no integration pre-release
      * ensure proper integrations through DevOps or DevSecOps
  + Operations and Maintenance
    - majority of cost of ownership
      * correcting vulns
      * support
* DevOps and DevSecOps
  + - prevent friction between devs and ops by integrating communication and teamwork
      * devs, IT, and QA
* Software Assessment Methods
  + - software developments and QA
      * IT ops and security teams to ensure CIA
  + User Acceptance Testing
    - before end of devphase or end of implementation phase
    - continuous engagement
  + Stress Testing
    - DoS conditions
    - too little as well as too much
      * resource starvation
    - graceful degradation vs total fail
  + Security Regression Testing
    - fix flaws and vulns before production
      * process must be standardized to ensure thoroughness
  + Code Review
    - QA and Secure Coding Review as a systematic process
      * mature software development process
        + proofreading
    - coding standards, reviewers, standardization of coding
  + Static Analysis Tools
    - integrated into development
  + Dynamic Analysis Tools
    - integrated into development
      * debuggers
      * execution flow mappers
    - consider code coverage
      * what degree of code is reviewable
        + ~75-100% depending on size
  + Formal Methods of Verifying Critical Software
    - mathematical
      * formal specifications
        + consistency and completeness
      * formal intense testing
        + specific mission-critical components
* Secure Coding Best Practices
  + - Quality
    - Input Validation
    - Output Encoding
    - Data Protection
    - Session Management
    - Authentication
    - Data at Rest
      * encryption
    - Data in Motion
      * TLS encrypt, IPSec, VPNs
    - Data in Use
      * patch/detect vulns in processing
      * whole memory encryption
    - Parameterized Queries
* Chapter Review
  + - attend meetings, ask questions, sharing thoughts
    - DevSecOps
      * web app vuln scans
      * stress testing
      * fuzzing
  + context-sensitive whitelisting
    - client and server side
  + HTTPS use always, and hard to guess session IDs
  + MFA, strong passwords, avoid plaintext transmissions
  + account lockout policies, log all authentication attempts

Chapter 10 – Hardware Assurance Best Practices

* Hardware Root of Trust
  + - need to establish to be sure systems on which software is being run are secure
      * foundation of secure systems
    - trusted execution environment with cryptographic functions for data protections
      * resist tampering
    - integrated right into motherboard or added as module later
      * IoT and embedded devices often lack Root of Trust
  + Trusted Platform Module
    - SoC dedicated to security functions such as crypto keys and digital certs, encryption, and hashing
      * protected/encapsulated microcontroller used as a safe haven
        + improves HRoT, effective detection of malicious config changes
    - TPM Uses
      * bind hard disk drives to a particular computing system
        + encrypted and wrapped in encryption again

importance to escrow a backup in case of TPM failure

* + - * sealing a systems state to a particular hardware and software configuration
        + deter tampering via file hash comparison
      * TPM uses RAM (non-volatile)
        + persistent – static – EK pair, SRK master
        + versatile – dynamic – PCRs, ALKs, Storage Keys
  + Hardware Security Module
    - removable expansion card that offloads cryptographic functions
      * US FIPS 140-2 evaluates HSMs
* eFuse
  + single bit of nonvolatile memory that is One-Time Programmable Memory (OTP)
    - disables access to certain chip functionalities
    - stores data such as firmware, cryptokeys, and OS integrity checking
* Firmware
  + - start section of non-volatile memory containing instructions and data
  + Unified Extensible Firmware Interface
    - UEFI
      * + partitioning over 2TB
        + boot from disks, drives, LAN, HTTP
        + remote recovery and management
      * Sec Phase (SEC)
      * Pre-EFI Initialization (PEI)
      * Driver Exe Environment (DXE)
      * Boot Device Select (BDS)
      * Transient System Load (TSL)
      * Runtime (RT)
    - figure out what code is needed, load it into memory, and execute it
      * can do secure boot
        + RoT in firmware via root CA and x.509 certs
        + blacklisting capability
  + Measured Boot and Attestation
    - * starts with firmware RoT but relies on hash comparison rather than Cas
        + audit trail of code
      * attestation sends these hashes to a management station
        + TPM is used to securely compute, store, and transmit the hashes needed for secure boot and attestation
  + Trusted Firmware Updates
    - firmware overwrites 🡪 risky
    - use firmware with inbuilt updating ability
      * trusted firmware update system
* Self-Encrypting Drive
  + - full-disk encryption (FDE)
      * data at rest
    - self-encrypting drives
      * hardware-based approach, crypto module integrated with storage media
        + TCG Opal 2.0 Standard
      * symmetric encryption, AES-128/256
        + dynamically
        + faster than software-based
        + DEK stored in non-volatile memory and encrypted with user password
      * SED wipes are almost instant via new DEK creation
* Bus Encryption
  + SEDs decrypt before transit
    - bus between drive to MB, in memory, and bus between memory and CPU
      * vulnerable
  + bus encryption encrypts instructions everywhere except in processor
    - requires cryptoprocessor
    - common in ATMs, satellite TVs, military, smart cards
* Secure Processing
  + - processors typically require unencrypted instructions
      * to protect
        + isolate processing area for only trusted apps
        + build extensions for processor to create mini environments
        + use apps that lock processor until runtime-completes
  + Trusted Execution Environment
    - * run in parallel with untrusted Rich Execution Environments
        + apps undergo rigorous checks
      * common in embedded/IoT for CIA
    - creates trust boundary and controls REE interaction via API
      * common to run apps that are non-cryptographic
    - starts with secure boot, HRoT
  + Processor Security Extensions
    - provide security features inside CPU in support of TEE
      * designate memory regions as encrypted and private
      * dynamically decrypted by CPU
  + Atomic Execution
    - control manner in which certain sections of a program run to prevent interruptions between start and end of section
      * places lock around it, each given code segment
      * avoid using this regularly
    - protects against TOCTOU attacks
* Trusted Foundry
  + - counterfeit products introduce vulns into networks
    - ensure mission critical systems can be developed and fielded using a hardened supply chain
* Anti-Tamper Techniques
  + - generation of random signals
    - bogus circuit mapping
    - micro-mesh to detect breaches
      * automatic zeroization of non-volatile memory/firmware
* Chapter Review
  + HRoT
  + leverage trusted platform modules and processor security extensions

Part III: Security Ops and Monitoring

Chapter 11 – Data Analysis in Security Monitoring Activities

* Security Data Analytics
  + - not if, but when
    - ID relevant data for analysis 🡪 collected, normalized, and visualized
      * generate security decisions, auditing, and compliance
        + info 🡪 knowledge 🡪 action
    - data nodes as numerous as system devices, all generating log data
  + Data Aggregation and Correlation
    - * compare similar data types regardless of source
        + log manager 🡪 timeline data
      * start with tools at hand or with observations
        + tool workflow

filtering 🡪 limited range

* + - * + comprehensive tools

SIEMs 🡪 Splunk, ELK

Big Data Solutions 🡪 Built on top of SIEMs

Local Analytic solutions 🡪 inhouse scripting

* + - * observation
        + hypothesis

prove/disprove

* Data Analysis
  + - trend analysis to determine network change for countermeasures and compensating controls
  + Trend Analysis
    - threat management context
      * predictive analytics
    - Internal Trends
    - Temporal Trends
      * trends over time
    - Spatial Trends
      * specific geographic regions and response behaviors
  + Historical Analysis
    - reference point creation/use
    - baseline development, model design
      * must consider context of changes over time
* Behavioral Analysis
  + - network behavior anomaly analysis
      * multi-point connection between data points
  + Heuristics
    - approached based on experience
      * practical detection methods
      * malware detection
        + NGFs test malware in VM sandboxes
  + Anomaly Analysis
    - first requires baselining
      * individuals and collections
      * statistically significant duration combined with context
* Endpoint Security
  + - network security 🡪 thorough inspection without impacting speed
    - host security is important too, as it receives the malicious data
  + Malware
    - * recognizing malware 🡪 common binaries, software deconstruction
    - Fingerprinting/Hashing
      * integrity checking, VirusTotal
    - Decomposition
      * decompiling and disassembly
        + of binary executables

windows PE format with 2-byte sequence

5A 4D

4D 5A

linux executable and linkable format (ELF) with 4-byte sequence

7F 45 4C 46

* + - Computer Generations
      * 1st – ML
      * 2nd – AL
      * 3rd – HLP
    - disassemblers are more useful than decompilers
  + Detect and Block
    - * signature-based
      * behavior-based
    - try and minimize false positives
  + Fileless Malware
    - in-memory malware
      * doesn’t leave behind traces for forensics
        + can use pre-existing software to carry out malicious actions
  + Sandbox
    - testing unknown files in real, but restricted, environments
  + Cloud-Connection Protection
    - rapid file reputation determination and behavioral statistics
    - automatic sharing of threat details across network
  + User and Entity Behavior Analytics
    - applied machine learning techniques to detect anomalous behavior quickly when there are deviations from normal patterns without predefined rules
      * first collects trends and behaviors for baseline collection
        + then performs continuous monitoring
* Network
  + - common sources available to help ID security events faster and more accurately
  + Domain Name System Analysis
    - recursive DNS resolution can obfuscate true source IP addresses when analyzing traffic
      * intermediate sources falsely flagged
        + pivots increase log complexity
    - modern DNS servers have enhanced logging for SIEM use
      * increased visibility
    - re-architect network so originating client info is piped direct into SIEM or detected by security devices before request reaches resolver
      * tons of storage required
      * compare domains/IPs resolved against a blacklist or malformed requests
  + Domain Generation Algorithms
    - generate domains rapidly using random yet predictable processes
      * C2 servers
        + cut link to prevent successful use
    - quickly switching to new domains rather than rely on static C2 targets
      * to balance predictability with randomness, DGA uses a seed
        + linear congruential generator to create domains
    - consider cost, privacy, and reliability of registration infrastructure
    - machine learning can help begin to counter DGAs
  + Flow Analysis
    - track traffic origination and termination point
      * real-time combined with automation to minimize false positives
    - also track aggregate volume
      * Etherape can assist in depicting all known endpoints by relational sizr
    - Netflow Analysis
      * Cisco System
        + network traffic statistics
      * packets grouped into flows via common
        + arrival interface
        + source and destination IP addresses
        + source and destination ports
        + IP protocol, service type
      * aggregate statistics about flow such as duration, number of packets, number of bytes
        + analysis console enables analysist to examine data and turn it into actionable info
      * data always available only after flow end
        + good for early stages of investigation
  + Packet Analysis
    - * re-create timeline of events using strategically places sensors
        + lots of storage
    - Filters
      * for capture and for display
        + balance between storage, performance, and visibility
    - TCP Streams
      * wireshark allows for reconstruction of streams of TCP data
    - Encryption
      * content will be incomprehensible
        + use HTTPS proxies that terminate TLS/SSL connections to capture normally encrypted context

use of a CA

* + - Protocol Analysis
      * way in which the packets conform to protocols they are implementing
        + detect ICMP tunneling
        + determine security of a given protocol
        + how protocols are expressed on networks
  + Malware
    - anomalous network traffic activity
      * unusual patterns and volumes and sensitive data leaving network
    - analyze traffic for requests to suspicious domains to ID behavior associated with preparation for malware delivery
      * C2 detection
* Log Review
  + - knowing how to review
      * gather, correlate, and analyze info in a central location 🡪 SIEM
    - human review is essential
  + Packet Capture
    - connections and ensuing exchanges
      * manual review with automated tools
  + System Logs
    - * end systems have richer logging potential as they are easier to observer
        + additionally, user impersonation minimizes network traffic effectiveness
      * windows event logs
      * syslog
    - Event Logs
      * Event Viewer 🡪 Filter Current Log
        + Event IDs
    - Syslog
      * messaging protocol
        + linux/unix syslogd
      * syslog hierarchy, can be used locally or centrally
        + syslog server 🡪 UDP 514/TCP 514

analysis of aggregated syslog data

facility code and severity level

origination source

* + Firewall Logs
    - * flow restriction via packet examination to ensure compliance
        + provide ample logs if configured so
    - Web Application Firewall
      * WAFs 🡪 App Layer 🡪 monitor/block destructive traffic
        + user, session, and application aware
    - Proxy Logs
      * ideal position to log all web traffic originating from inside the network
        + ID anomalous browsing behavior (C2)
  + IDS/IPS
    - * powerful rule languages
      * NGFs combine traditional firewalls and IPSs
        + measured by focus and approach

host-based

network-based

stand-alone or integrated

signature-based

anomaly-based

* + - Snort
      * open source NID/NIPS
        + rich and abundant rules
      * snort rules are structured using a header and options
        + specify action – header
        + display messages, where to look, etc
    - Zeek
      * signature and anomaly-based
        + creates events that tell user something has happened

session tracking, IPS

* + - * + compared with policies to determine best course of action

may lead to different response options

can extract executables for sandboxing mid-stream

* + - Suricata
      * multithreaded snort
      * can use GPU, extract files, and be used as IPS
  + Authentication Logs
    - both successful and unsuccessful attempts should be logged
      * linux – auth\_log
    - combine with behavioral context
* Impact Analysis
  + - tangible and intangible costs
      * broken down into phases that mirror incident response
        + triage 🡪 immediate
        + reaction 🡪 localized
        + auditing 🡪 total
        + repair 🡪 future
  + Auditing Analysis
    - determine likelihood that systems will be available to authorized users
      * DDoS mitigation
      * at what point will local controls fail
    - determine consequences of the loss of a given asset or set of assets
      * ID key assets
    - resource-monitoring tools
      * monitor metrics and endpoints
* Security Information and Event Management Review
  + - collect, store, analyze, and report
      * normalize data format
      * optimized for quick retrieval
    - ELK-stack
      * visible diagrams to illustrate network activity
    - craft correlation rules to derive meaningful info from observed patterns
* Query Writing
  + - data searching
      * search languages
        + SPL
        + KQL
        + Apache Lucene
      * string searches and Boolean searches
    - scripting
      * shell, bash, Perl, Python
        + use appropriate working directories and correct environments, argument usage
    - Splunk 🡪 REST and Python Enterprise SDK
      * script at the system level and interact directly with platforms
        + via APIs, cURL, etc

as long as necessary permissions have been granted

ELK and cURL with console or specific indexes

* + - * piping is useful by enabling stdout of a command to be connected to stdin
* Email Analysis
  + - phishing
  + Malicious Payloads
    - concealed inside other files
      * word docs, ZIP files, media, PDFs,
    - social engineering
      * script/macro enabling
      * malicious URLs
  + DomainKeys Identified Mail
    - email msg verification
      * ID verification
      * determine ID as known/unknown
      * determine ID as trusted/untrusted
    - key per usage
      * public published to DNS records
      * sender includes special signature header in all outgoing messages
        + protect against spam and spoofing, alert recipients of possible tampering
  + Sender Policy Framework
    - enables domain owners to prevent such email spoofing using domains by leveraging DNS functionality
    - SPF TXT records
      * list authorized mail servers, verified via DNS
        + otherwise marked suspicious, for further review
  + Domain-Based Message Authentication, Reporting, and Conformance
    - * DMARC
        + give email domain owners ability to prevent spoofing and reduce spam that appears to originate from said domain

entity created in domain DNS record

determine how apparently specific messages should be handled

* + - * + uses SPF and DKIM 🡪 essential to correctly configure both

msgs that fail SPK/DKIM or domain tests may invoke DMARC policy

DMARC reports issued

* + Header
    - * details contained here can be used for analysis
        + SPF and DKIM verdicts
        + messages journey
  + Phishing
  + Forwarding
    - dedicated inboxes to review suspicious emails that have bypassed filters
      * help counter phishing, assist in aggregating data for policy changes
  + Digital Signatures and Encryption
    - * sender verification
      * message integrity
      * nonrepudiation
    - uses public/private cryptokeys
      * commercial - S/MIME – encryption, key exchange, message signing
      * private – PGP
        + crypto privacy and authentication
  + Embedded Links
    - real-time analysis for presence of URLs and domains and disable/redirect to valid domains instead
      * SaaS platforms often offer this feature
  + Impersonation (Social Engineering)
    - counter with training and awareness
* Chapter Review
  + analyze collected data and turn it into information

Chapter 12 – Implementing Configuration Changes to Existing Controls to Improve Security

* Permissions
  + - users and service accounts
      * better to focus on groups with minimum permissions assigned
  + Users
    - * Local group Policy Editors
      * Local Group Security Policy Editors
      * Default Domain Policy Group Policy Object
        + Windows 10 🡪 lusrmgr.msc utility

disable Guest account

* + - * + Red Hat Linux 🡪 lslogins cmd

provides info about UID, username, and recent login activity

* + Groups
    - inherits group privileges, assigned by necessity
      * W10 🡪 lusrmgr.msc
        + most OS default
      * Red Hat 🡪 getent group cmd
        + groups cmd
* Blacklisting
  + - relies heavily on up-to-date listing of known threats
      * not really effective due to explicit deny
* Whitelisting
  + - allowing only known good applications with implicit deny
      * improves stability
      * improves auditing and compliance
    - Windows 🡪 configure application whitelisting
      * blocks every app from running by default
        + Security Policy Editor (local)
        + Group Policy Editor (domains)
      * Software Restriction Policies
        + define security level

disallowed level

* + - * + add application paths
* Firewalls
  + - many set to deny all incoming connections and allow all outgoing connections by default
    - Enable
      * Set Rules in Correct Order
        + Linux ufw 🡪 uncomplicated firewall

iptables interface

* + - * + begin with explicit deny all incoming that does not have pre-established connections

allow all outgoing connections

set additional rules

call utility

provide rule

set direction

* + - * See full list of rules
        + sudo ufw status verbose
        + sudo iptables -S
    - each rule in a list is interpreted in order as a packet comes in for inspection
      * sequential order in iptables
    - ufw uses /etc/services file to determine ports associated with services
    - safer to deny all traffic by default and allow only the specific communications that you need to pass through
  + Web Proxies
    - intercepts and forwards web traffic between clients and servers
      * common for content filtering
      * block traffic, screen traffic
      * provide logs, monitor bandwidth
      * caching servers
    - ensure all clients trust CA signing internal certs
      * proxys CA must be trusted in both directions for it to work
  + Web Application Firewalls
    - mediates external traffic to a protected server
      * configured specific to what apps are being protected
  + Operating System Firewalls
    - should be used in addition to network-based firewalls
      * Windows Defender
        + managed via GroupPolicies
      * MacOS
        + accessed via Security + Privacy in System Preferences

CA whitelisting

works with IP-based firewalls

* Intrusion Prevention System Rules
  + - matching previously identified malicious activity or by analyzing network traffic for indicators
      * some (IPSs) will react and intervene
        + network-based
        + host-based
    - common deployments
      * Zeek
      * Suricata
      * Snort
  + Snort Rule Building
    - * not automatic logging
      * highly customizable rules
        + <snort action> <protocol> <src IP> <src port> <direction> <dst IP> <dst port> (msg:” “; <optional classtype>; <optional snort ID>; <optional revision number>; )
      * make as focused as possible
        + scope down to specific traffic
      * PulledPork and OinkMaster
        + auto install latest ruleset based on Snort version

Perl-based, cron-job -able

* + - * + configs carry over between updates
  + Zeek Logs
    - event capture and analysis via scripts for anomalies
      * react with alerts, config changes, and low-level pattern matching
  + Suricata Rule-Building
    - high performance IPD/IDS and netmonitoring tool
      * inspection, detection, netcapture
      * high scalable via Lua scripting
    - rules consist of three components
      * actions that will occur with pattern match
        + pass, drop, reject, alert
      * header identifies protocol, addrs, ports, direction of rule
      * rule options component is used to define rules further
  + Host-Based IPS
    - specific host network interfaces
      * finely tune traffic per devices
      * incorporates behavioral or heuristic approaches
    - limitations on resources, manpower demand
      * counter with centralized management and monitoring capabilities
* Data Loss Prevention
  + - balance between protecting data at rest, transit, and use while allowing unencumbered flow of data to support business operations
      * SaaS often offers DLP solutions
    - difficult to detect complex patterns
      * config rules in relation to account context
* Endpoint Detection and Response
  + - monitoring and detection, investigation, response, and mitigation
      * Monitor 🡪 log/aggregate endpoint activity to facilitate trend analysis
      * Detect 🡪 find threats with continuous analysis of monitored data
      * Respond 🡪 address malicious activity on the network by stopping the offending asset
    - protect endpoints
      * highly scalable, lightweight, cross-platform
    - tremendous data collection, manpower demand
      * must determine if prevention is priority over remediation
* Network Access Control
  + - deeper endpoint visibility, policy enforcement checks before network connection
      * RBAC, 802.1x (inflexible)
      * endpoint malware protection, version checking
        + transparent remediation for noncompliant devices
    - resource intensive, data collection, and privacy concerns
    - Modern NACs
      * multiple framework support
  + Time-Based
    - fixed intervals, timeframes, time limits
  + Rule-Based
    - query-hosts 🡪 version, security software, prohibited data, criteria, configs
      * can be passive
  + Role-Based
    - limit interactions between nodes to prevent unauthorized data disclosure
      * relies on directory services
      * process of locating sensitive information across various parts of the network is much faster
  + Location-Based
    - relying on VPNs
    - geofencing
* Sinkholing
  + - routing suspicious traffic or dropping altogether
      * provides a response to a DNS query that does not resolve to the actual IP address
        + DNS sinkholes target addresses known malicious domains
    - allow log checking to determine what local machines are attempting to contact a malicious endpoint
* Malware Signatures
  + signature-based
    - IoCs
    - byte strings
    - YARA rules
      * string definition
      * condition
    - ExploitDB
* Snadboxing
  + dynamic analysis
    - monitor traffic
    - examine processes
    - compare behaviors
    - memory artifacts, log files, system settings
  + can be costly, time-intensive, can be detected
* Port Security
  + unnecessary services, default credentials
    - easier to admin fewer services
* Chapter Review
  + layered defense
  + comprehensive approach that uses the principle of least privilege, enhanced logging, and smart whitelisting

Chapter 13 – The Importance of Proactive Threat Hunting

* Establishing a Hypothesis
  + - mindset that the attacker is already in your system 🡪 assume breach
      * make fewer potentially damaging assumptions about security posture
    - active defense
      * see beyond just alerts and dig deep to find malicious actors
      * human with machine learning and user and entity behavior analytics (UEBA)
    - data and analytical skills
      * requires familiarity with processes and skills associated
        + diverse experience
      * enhanced network awareness, accurate logging, established baselines
        + add context to raw data
    - ID and distinguish patterns of attack
      * track target activity throughout environment
        + remove threat after preparation of tools/plans and follow up

purpose, outcome

where, what is needed

who is involved

* + - * Generate Hypothesis
      * Investigate via Tools
      * Discover Patterns
      * Inform Operations
    - ideas about threats in the environment and how to likely ID them
      * observable and testable
        + must communicate process to stakeholders clearly and concisely
      * Analytics Driven
      * Situation Driven
      * Intelligence Driven
      * Experience Driven
* Profiling Threat Actors and Activities
  + - ATT&CK
      * other frameworks
* Threat Hunting Tactics
  + - automatics and specialized tooling
      * + data sifting
      * firewalls
      * endpoint protection software
      * IPSs
      * SIEMs and visualization tools
      * spreadsheets, ATT&CK Navigator
  + High Impact TTPs
    - go through phases of an attack model and determine areas of greatest concern
      * Initial Access and Discovery
        + enumeration

artifacts

* + - * Persistence
        + scheduled tasks, registry
        + legit processes, malicious injection

DLLi

* + - * + MacOS LaunchAgent
      * Lateral Movement and Privilege Escalation
        + PtH, RDP
      * Command and Control
        + blended into routine net traffic

use deep inspection

* + - * + uncommon ports
      * Exfiltration
        + DNS Tunneling

abnormal queries/volume

* + - * + High Entropy Domains

sometimes used by Cloud CDNs

* + - * Searching
        + data that matches hypothesis via collection/aggregation

Querying

* + - * Clustering
        + used in statistics to ID groups of data points

determine high interclass similarity

* + - * Grouping
        + categorize similar data by taking set of unique features and determining artifacts that fit criteria

require defining features be described in advance

* + - * Stacking
        + ID data outliers
        + most useful with limited datasets
* Delivering Results
  + - well-defined criteria for success and knowing when to stop
      * hard to determine
        + science and art
    - periodic reviews
  + Documenting the Process
    - short and highly scoped hunts
      * document hypothesis
      * why was/wasn’t correct
      * inhibitors and lessons learned
  + Reducing the Attack Surface Area and Bundling Critical Assets
    - SANS Threat Hunting Survey to ID trends
      * quality and coverage of detection
      * reduction of attack surface exposure
    - device hardening and network architecture, asset management, vuln scanners
      * network diags, asset catalogs
  + Attack Vectors
    - macro 🡪 malware, vuln exploits, social engineering, insider threats
    - malware 🡪 polymorphism
    - vuln exploits 🡪 zero days, brute-forcing
    - social engineering 🡪 mislead/confuse
    - insider threats 🡪 incredibly difficult to detect
  + Integrated Intelligence
    - Hunters and Analysts
      * TTPs and Infrastructure
      * feedback loop, intel loop
  + Improving Detection Capabilities
    - TTP understanding, detection logic, improvements, policy changes, infrastructure modification
    - increase network visibility, config tailoring
      * updated rules for IDS/IPS
      * updated logic for SIEM
      * updated logic for EDR
      * improved sensor placement
      * improved asset visibility
    - improved development process
    - security training
    - changes to QA and QC
* Chapter Review
  + threat hunting looks for malicious activity that may have slipped past detection and prevention mechanisms
    - assume compromise
  + repeatable and scalable
    - context and data 🡪 pursue signs of intrusion

Chapter 14 – Automation Concepts and Technologies

* Workflow Orchestration
  + - create conditions that enable the analyst to focus on complex tasks
      * repeat security testing cases
      * quickly flag suspicious behaviors
        + error conditions
        + automated alerting and report delivery mechanisms
    - scripting at scale and in mixed environments
      * orchestration 🡪 instrumentation and management of automation
        + APIs
  + Security Orchestration, Automation, and Response Platforms
    - * getting disparate security systems to work nicely together
        + SOAR tools

collect security threats data and alerts from different sources

* + - * define, prioritize, and drive standardized incident response
        + often in tandem with SIEM
        + Splunk Phantom (Phantom Cyber)

back-end integration

front-end presentation

* + Orchestration Playbooks
    - workflows for visualization and execution of processes
      * formulaic, scalable
    - requires initial condition (proactive/reactive)
      * followed by process steps 🡪 to react accordingly
        + finally reaches an end state (remediation/reporting/auditing)

may chain to another initial condition

* + - SOARs allow for playbook design both textually and visually
* Data Enrichment
  + - leads to insights into data, tasks to draw on resources
      * automate for useful context, informed decision making
* Scripting
  + - cronjobs 🡪 crontab
    - SSH to issue CLI cmds
      * scp for secure copy
  + Python Scripting
    - multiplatform language
      * human legible
    - combined with regex
  + Powershell Scripting
    - automation framework
      * obtain info from endpoints and servers
* Application Programming Interface Integration
  + - primary user/machine interface mechanism
      * improve comms while maintaining security
      * effective at data sharing
      * high volume, scalable, and repeatable
    - rely on SOAP and REST
  + Representational State Transfer
    - RESTFUL APIs
      * General Architectural Style
        + Client/Server
        + Stateless
        + Cacheable
        + Uniform Interface

ID resources via reqs

manipulation of resources through representation

self-descriptive messages

hypermedia as the engine of application state

* + - * layered system
      * code on demand
    - use a partition of HTTP response message to provide feedback to a requestor
      * status codes
        + 1xx – info
        + 2xx – success
        + 3xx – redirect
        + 4xx – client error
        + 5xx – server error
  + Automating API Calls
    - craft a call and convert into code
      * Postman, Insomnia, Swagger Codeger
* Automated Malware Signature Creation
  + - YARA
      * strings definition
      * condition
    - scalable Booleans
      * yarGen 🡪 YARA rule generation software
        + searches for strings found in malware files
        + comes with default database 🡪 point to path to examine

python yarGen.py -m <path>

* + - * + outputs generated rules containing meta, strings, and condition fields
    - BASS – open source, similar to YARA
      * low memory profile and performance demand implemented via Docker containers
        + generates pattern based signatures in a scalable way
* Threat Feed Combination
  + - sheer number of observables, indicators, and contextual pieces
      * internal and external sources, all requiring abstraction
    - formatting and tuning feeds to relevancy, with processing and normalization being capable of being automated
      * threat feed extraction, data parsing
  + improve firewalls, SIEMs, and SOARs
* Machine Learning
  + - uncover patterns, assist in decision-making
      * algorithm-based for data-mining
      * botnet and malware detection
        + behavior-pattern recognition, deviation from normality
    - tailored algs are better than general purpose ones
* Use of Automation Protocols and Standards
  + - DISA STIG
      * drive strategy
      * technical measures
      * minimize exposure
      * improvement network resilience
  + Security Content Automation Protocol and NIST SP 800-126
    - * assessment and reporting of vulns
      * languages 🡪 standard vocab/conventions
        + XCCDF
        + OVAL
        + OCIZ
      * reporting formats
        + ARF
        + AID
      * identification schemes
        + CPE
        + SWID
        + CCE
        + CVE
      * measurements and scoring systems
        + CVSS
        + CCSS
      * integrity
        + TMSAD
    - OpenSCAP
      * hardening guides, config baselines
      * widespread organization adaptability
        + use CIS OVAL DB for scanner definitions
        + common in RedHat
        + results can be used manually in a script or SOAR
* Software Engineering
  + - integrate security into development, to ensure security measures keep pace with software development
      * DevOps/SecDevOps
      * consistent cloud development environments
  + Continuous Integration
    - merging changes early and often for quick validation and production
      * frequent security testing
      * high usability and security
  + Continuous Development
    - previous changes are incorporated and released automatically
      * automated runtime security and compliance checks, disable unnecessary services and ports, remove devtools, enable security mechanisms, enforce audit and logging
* Chapter Review
  + well-planned approach to ID and deploy technical, operational, managerial steps with repeatability and scalability, reliably

Part IV

Chapter 15 – The Importance of the Incident Response Process

* Establishing a Communication Process
  + - * minimize likelihood of an incident by hardening the network and addressing vulns
        + outline exact steps to address incidents when they occur

Incident Response Process

* + - * well documented, tailored, and well understood
        + well-designed IR process makes it more likely organization moves efficiently from initial detection to resumption to normal operations
  + Internal Communications
    - process of keeping internal parties informed
      * war room
      * info clearing house
  + External Communications
    - must be carefully controlled, professionally
      * Govt entities
        + legal and compliance
      * Clientele
        + regulatory environments
        + transparency, media relations
      * Key Partners
  + Response Coordination with Relevant Entities
    - * effective inter-role comms
        + determine roles beforehand
        + escalation models
      * management involvement
        + contractor/external involvement

govt involvement

* + - * + stakeholders
    - Legal Counsel
      * Gov/LE involvement
      * regulatory considerations
        + HIPAA, PHI, PCI
    - Human Resources
      * insider involvement, accidental/intentional
      * general purpose
    - Public Relations
      * comms with customers, investor
      * planning/practice
        + laymans terms
    - Internal Staff
      * response team staff and composition
      * wargaming, chain of command
    - Contractors and External Parties
      * requires significant prior coordination and communication
        + planning/testing cooperation
    - Law Enforcement
      * law adherence, cooperation
        + mitigation/recovery – ops
        + business continuity – management
        + evidence preservation – LEA
    - Senior Leadership
      * include senior leaders for support
        + minimize disruptions, address regulatory issues, interface with leadership
      * scenario testing, trust building
* Factors Contributing to Data Criticality
  + Personally Identifiable Information
    - unique or context-based
      * US Privacy Act of 1974
    - sensitive personal information
  + Personal Health Information
    - HIPAA 1996
      * past, prevent, future physical/mental health conditions
  + High-Value Assets
    - info/systems critical to functionality
  + Payment Card Information
    - GDPR (EU)
    - GLBA (US)
    - FTC FPR
    - PCI DSS
  + Intellectual Property
    - prevent unauthorized disclosure
      * legal guidance in IRP
    - patents, copyrights, trademarks, trade secrets
  + Corporate Confidential Information
    - * internal ops, proprietary info
    - Account Data
      * treated like PHI
    - Mergers and Acquisitions
      * + insider trading, fraud
      * US SEC’s Fair Disclosure
* Chapter Review
  + Proper IR planning
    - ID scope of damage quickly
      * ID data requiring special handling
      * regulatory requirements
    - communication 🡪 smooth recovery

Chapter 16 – Appropriate Incident Response Procedures

* Preparation
  + - difference between *events* (observed, verified, documented) and *incidents* (one or more related negative events that cause compromise)
      * IR negates these effects
    - using intelligence cycle
      * used with a framework to digest and parse data efficiently
        + Incident Response Cycle

preparation

identification

understanding/analysis

response

recovery and growth

* + - reduce uncertainty via best practices and prioritization
      * proactivity and reactively at all levels
    - training, testing, and documentation
    - cooperation
  + Training
    - fundamentals, technical/nontechnical, identification/awareness, communication
  + Testing
    - practice makes perfect
      * evaluate organizations IR plans, procedures, and capabilities
        + walkthroughs – familiarization
        + tabletops – discussion/hypotheticals
        + functional exercises – ID and execute tasks to master
        + full-scale – real-time, detailed performance measurement
    - clearly defined rules, management involvement
  + Documentation
    - referencing, sharing, improving
    - details beyond steps
      * anything useful/relevant
* Detection and Analysis
  + - identification
      * automated techniques
        + scalable/reliable
  + Characteristics of Severity Level Classification
    - * clear criteria to determine how to classify a security event
        + scope of impact 🡪 deviation sufficient to warrant response

document legit anomalies

* + - Downtime
      * foundation of all other network metrics
        + feasibility of total network shutdown to handle IR
      * ID and inventory critical operational assets
        + max tolerable downtime (MTD)

nonessential – 30d

normal – 7d

important – 72hr

urgent – 24hr

critical - ~1hr

* + - * + prioritize
    - Recovery Time
      * Key Performance Indicators (KPIs)
      * Completeness over Speed
        + Recovery Time Objective (RTO)

smaller than MTD

* + - Data Integrity
      * + obvious attacks
        + insidious attacks
      * data backups
        + segregated from network
        + verify functionality periodically
    - Economic Impact
      * fines, fallout
      * reputational
      * asset value 🡪 determine security effort
    - System Process Criticality
      * tasks that must be accomplished with consistency
        + not just technical
      * criticality and probability
    - Data Correlation
      * SIEMs
        + what is/has happened, how and how to recover

what stage of attack cycle

* + Reverse Engineering
    - malware
      * dynamic
      * static
  + Containment
    - deny threat ability/means to enact further damage
      * prevent/reduce spread
        + based on nature of attack, assets targeted, and criticality
      * proactive/reactive
  + Segmentation
    - subnetting
      * physically
      * logically
    - ACLs
  + Isolation
    - isolation VLANs
      * receive compromised hosts for observation/testing
        + IOCs 🡪 CERT and ISACs
  + Removal
    - * + powered on and isolated
        + shut down and preserved
        + destroyed and rebuilt
      * Threat Intel Value (TTPs)
      * Crime Scene Evidence 🡪 forensic images
      * Ability to Restore 🡪 backed up data or not
    - Process must be well documented
* Eradication and Recovery
  + - gather evidence beforehand then rebuild system
      * having ID’d attack vectors and implemented countermeasures
  + Vulnerability Mitigation
    - applying lessons learned, preventing recurrences via vuln management process
  + Sanitization
    - NIST SP-800-88 Rev 1
      * cursory
      * sophisticated
    - Beyond reformatting a drive
      * overwriting – run and rerun
      * encryption – delete key after
      * degaussing
      * physical destruction
  + Reconstruction
    - rebuild host to pristine state
      * gold masters
      * data backups
  + Secure Disposal
    - physical destruction via shredding, pulverizing, disintegrating, or incineration
  + Patching
    - management, zero day discovery
      * BYOD concerns, centralization, NACs
  + Restoration of Permissions
    - * reliable backup solutions was implemented successfully, having been tested and verified regularly
    - Validating Permissions
      * correct elevated perms
        + permissions creep
      * excessive priv acct usage
      * poor management of inactive/ineligible accounts
  + Restoration of Services and Verification of Logging
    - * develop/execute network service validation
      * ensure necessity validation and logging
* Post-Incident Activities
  + Lessons-Learned Report
    - AARs
      * issue – sentence
      * discussion – paragraph
      * recommendation – report
    - discuss all inputs
      * refine and update
  + Change Control Process
    - recommendations from AAR sent to change control board
      * careful consideration
  + Updates to Response Plan
    - IP is a process needing constant review incorporating various perspectives
  + Summary Report
    - consider audience and purpose
  + IOC Generation
    - leverage and incorporate IoCs generated during response/containment phase
      * share internally/externally
  + Monitoring
    - update and implement following response changes to organization operations
* Chapter Review
  + IR is interpersonal
    - preparation, practice, execution, and communication
  + Prep 🡪 detect and analyze 🡪 contain 🡪 eradicate 🡪 recovery 🡪 post-incident activity

Chapter 17 – Analyze Potential Indicators of Compromise

* Network Related Indicators
  + - arrive at relevant facts to exact root cause of an issue
      * important symptoms
    - start from the outside-in (network-sensors) to interrogate hosts and processes
  + Bandwidth Utilization
    - patterns of utilization
      * amounts, endpoints involved, and directionality of connection
  + Beaconing
    - maintaining contact with compromised hosts, outgoing signals that are permitted by firewalls
      * contacting a C2 server
        + detected by periodicity and destination
    - endpoint analysis for regularity via internal source address and external destination address and time
  + Irregular Peer-to-Peer Communication
    - * most network traffic is typically client/server
        + less common for peer-to-peer connections
        + lateral movement

SMB, PsExec, SSH

* + - * unprivileged accounts connecting to other hosts
      * privileged accounts connecting from regular hosts
      * repeated failed remote logins
  + Rogue Devices on the Network
    - hardware/software asset management
      * help determine who/what doesn’t belong
    - use NACs to counter
      * granular control, centralized logs
    - centralize log collection from APs to check for MACs
      * can be spoofed though
  + Scan Sweeps
    - one host generating an abnormally large number of connection attempts
      * but not full connections
    - look for ARP messages, large number generated during scan sweep
      * need a sensor in every subnet to detect ARPs
  + Common Protocol over a Non-Standard Port
    - 8080 vs 80
    - 53, 443 🡪 assumed lack of visibility on traffic
      * encrypted and/or necessary
* Host-Related Indicators
  + - check suspicious host to determine benign/malicious cause
  + Capacity Consumption
    - memory, CPU cycles, disk space, local bandwidth
      * when/where these spike should/shouldn’t occur
        + system baselines
    - how to ID anomalous
    - Warning Signs
      * memory – low memory available despite low system usage
        + high memory consumption despite no active user/tasks
      * drive capacity – sudden drop in available free space with increased network activity
      * processor – prolonged periods of high processor consumption
        + unusual consumption from unfamiliar tasks
      * network – periods of extremely high network throughput despite no active user/tasks
  + Unauthorized Software
    - * convenient/effective vs memory-only/fileless
      * can bypass firewalls using code-obfuscation
      * counter with whitelisting programs
    - Whitelisting
      * effective but unpopular with everyday user as acquisition process is delayed
      * should inventory all software
  + Malicious Processes
    - * examine running processes when responding to an incident
        + top, ps in linux
        + windows task manager
      * must known what is normal (baseline), any changes to common/normal systems
        + check what resources are being utilized
        + netstat

windows 🡪 netstat ano

macOS 🡪 netstat -v

linux 🡪 netstat -nap

* + - Memory Contents
      * volatile memory tool

time/labor intensive

* + - * + AccessData FTK Imager
        + Hal Pomeraniz Linux Memory Grabber
      * analyze with Volatility
  + Unauthorized Changes
    - * DLL changes
        + detect with logging (object access auditing)
        + hash comparison for files that are expected to never change
      * File System
        + high priority during incident response for artifact collection
    - Unauthorized Privileges
      * PrivEsc measures
        + acquiring privileges credentials, exploiting software flaws, exploiting misconfigs
      * Disable compromised accounts, confine hosts to VLANs, monitor behaviors
  + Data Exfiltration
    - * ID data and sensitivity to design appropriate controls
    - search for exfiltration staging areas
      * Irregular Peer-to-Peer Comms
        + legit connection with irregular volume and endpoints
        + NetFlow Analysis, DLPs
  + Registry Change or Anomaly
  + Unauthorized Scheduled Tasks
    - Task Scheduler
    - cron, anacron, at cmd
      * crontab -e
    - LaunchAgent, LaunchDaemon, cron, Login Items, kernel extensions
* Applied Related Indicators
  + Anomalous Activity
    - unusual behavior
      * frozen pages, rapidly changing URLs, browser restarts
    - malicious emails
    - unresponsive apps, flashing windows, pop-ups
  + Introduction of New Accounts
    - * privileged domain accounts
        + remove/disabled
  + Unexpected Output
    - unexpected pop ups (UACs for ex)
    - cert warnings
    - navigation confirmation pop ups
  + Unexpected Outbound Communication
    - * host-based sensor, IDS to counter
    - applications needlessly communicating over a certain port
  + Service Interruption
    - sudden starting, stopping, restarting, or crashing
      * review resource manager and log files
  + Memory Overflows
    - memory errors that disrupt/terminate processes, display error messages
      * memory dumps
  + Application Logs
    - OS application logs, applied SIEM logs
      * important for response efforts
        + log volume issue, lack context and require effort to analyze
* Chapter Review
  + network 🡪 host 🡪 application

Chapter 18 – Utilize Basic Digital Forensic Techniques

* Phases of Investigation
  + - collect/analyze data to determine how/why
      * treat every forensic investigation as if its going to end up in court
    - security/collection of digital evidence cant affect integrity
      * using trained persons
        + all documented, preserved, and available for review
  + Seizure
    - * prevent changes to evidence
        + control the crime scene
      * secure/document environment
        + acquire data, photograph, tag/label/inventory
      * proper packaging/handling/transportation
    - Chain of Custody
  + Data Acquisition
    - * extract digital content from evidence
        + forensic imaging to examine copies of original, not actual
      * Prepare the Destination Media/Medium
        + overwriting
      * Prevent Changes to Original
        + write-protection mechanisms like hardware write blockers
      * Hash the Original Evidence
        + dd utility in linux
      * Verify the Acquisition
        + hash comparison
      * Safeguard Original Evidence
  + Analysis
    - determine significance of data
      * timeframe
      * data hiding
      * applications and files
      * ownership and possession
    - timeframe
      * date and time
      * time zone
      * source
      * item name
      * item location
      * description
    - take copious notes
  + Reporting
    - narrative, notes, conclusions
* Network
  + - header captures and full packet captures
      * significant data management
    - both require legal compliance and governance
  + Networking Tap
    - tap hardware capture traffic between endpoints
      * passive 🡪 no additional power, passive tap splits cooper line for power to sniff packets
        + can disrupt Gb lines
      * active 🡪 used on Gb speed lines
        + sends copy of signal to interface, moves OG signal to forwarder
  + Hub
    - capture traffic directly as all traffic is shared equality in a hub
  + Switches
    - in-built port mirroring functionality options
      * SPAN port
  + Wireshark/tshark
    - TCPdump
    - Windump
* Endpoints
  + - don’t power anything off unless data is being destroyed by a software
      * keep on for memory forensics to track fileless/memory-only
    - notes and photos of surrounding environment
      * computer screen
      * peripherals
      * immediate surroundings
      * proximate surroundings
  + Servers
    - forced to perform live forensics
      * cap memory contents and files of interest
    - storage capacity increases analytics times
      * RAIDs require special tools
    - consider particular architectures
  + OS and Process Analysis
    - * computer resources like memory, CPU, and disks
        + registry and event logs

autorun locations

most recently used lists

wireless networks

* + - * + event viewer
      * linux plaintext files
        + dd, sha1sum, ps
    - Log Viewers
      * Splunk
      * SolarWinds Event Log Consolidator/Manager
      * Ipswitches WhatsUp
* Mobile Device Forensics
  + - still connected to the network unless powered off
    - require special tools and cables
      * data is also not designed for acquisition
        + need to load alternate OS via custom bootloader
        + miniDBMs also require special syntax
* Virtualization and the Cloud
  + - benefits of virtualization
      * everything exists in the form of files
      * snapshots
        + VMX, VMDK, NVRAM, VMEM
    - critical to ID cloud computing asap
      * ID and track data associated with virtualized devices and functionality
      * physical location of data can pose legal challenges
        + cloud services often employ strong logging that can be relied upon
* Procedures
  + - maintaining data integrity
  + Building a Forensic Kit
    - * jump bag 🡪 bugout bag for IR
    - Live Response Tools
      * collect live volatile data quick from a system using removable media
    - Write Blockers and Drive Adapters
      * prevent modification to storage during acquisition
        + consult asset list to determine interface capability
    - Cables
      * of all types
    - Wiped Removable Media
      * maximum capacity should exceed devices listed
        + portable RAIDs

Forensic Data Monster

* + - * + NAS
    - Camera
    - Crime Scene Tape
    - Tamper Proof Seals
    - Documentation and Forms
      * Chain of Custody form
      * Incident Response Plan
      * Incident Log
      * Call/Escalation List
  + Cryptographer Tools
    - compression utilities with passwords
    - VeraCrypt
* Acquisition Utilities
  + - ensuring admissibility of evidence, analysis, and conclusions in court
  + Forensic Duplicates
    - * don’t rely on file system operations, allowing capture of MFT and inode table
        + slack and free space
      * dd utility
      * FTK Imager
        + forensically sound acquisition
  + Password Crackers
    - Password Forensic Kit
    - JtR and Hashcat/oclHashcat
  + Hashing Utilities
    - default in most ISs
      * md5, sha1sum
      * Microsoft File Checksum Integrity
      * Verifier
  + Forensic Suites
    - EnCase
      * forensic acquisition, analysis, and report generation
    - FTK
      * create forensic images of hard drive with built-in logging features
        + FTK Imager
    - Cellebrite
      * handheld Universal Forensic Extraction Device (UFED)
        + mobile phones
  + File Carving
    - fully recover partially recovered files or those discovered to be damaged
      * doesn’t depend on file system in use
    - PhotoRec
* Chapter Review
  + discipline and fixed adherence to established processes
    - ID quick response scenarios and when they require escalation
  + decide what to do next

Part V

Chapter 19 – The Importance of Data Privacy and Protection

* Privacy vs Security
  + - control over how others view, share, or use personal info vs protection of info assets against unauthorized access, modification, or destruction
* Types of Data
  + - * compliance and assessment implications of date privacy and protection
    - Personal Data
    - Personal Health Data
    - Financial Data
    - Copyrighted Data
  + Legal Requirements for Data
    - * multiple local, national, and international laws
        + assess risk, implement controls that reduce risk
    - GDPR
    - HIPAA
    - PCI DSS
    - US Copyright Law
      * USC 17 ch 1-8 + 10-12
      * Copyright act of 1976
* Nontechnical Controls
  + - policies and procedures, admin/soft controls
  + Data Ownership
    - possession and responsibility
      * management due care responsibilities
  + Data Classification
    - private/public
    - valuation and sensitivity/criticality
  + Data Confidentiality
    - NDA
  + Data Sovereignty
    - counting which data is collected has supreme legal authority over it
      * tricky with cloud resources
  + Data Minimization
    - can acquire and retain only the minimum amount of data required to satisfy the specific purpose for which owner has authorized use of data
  + Data Purpose Limitation
    - data can only be used for the purpose for which it was collected
  + Data Retention
    - * deliberate preservation and protection of digital data in order to satisfy business or legal requirements
        + possess and follow an established data retention policy
    - Data Retention Standards
      * + what is kept, for how long, and where

manner in which data is kept

* + - * + taxonomy
        + normalization
        + indexing
      * archiving in a centralized, regimented, and homogenous manner
* Technical Controls
  + - need to understand context in which specific controls are implemented
  + Access Controls
    - Kerberos, biometrics, PKI, RADIUS, TACACS+, cards/chips
  + Encryption
  + Sharing Data While Preserving Privacy
    - * limited access to just enough data
        + data masking
        + deidentification
        + tokenization
    - Data Masking
      * cover/replace parts of sensitive data with data that isn’t sensitive
    - Deidentification
      * render near impossible to determine individual to whom a specific data record belongs
        + anonymization
        + pseudorandom identifiers

tokenization

* + - * + can be reversed with sufficient data correlation
    - Tokenization
      * mapping sensitive and de-sensitized versions of a dataset together
  + Digital Rights Management
    - * tech used for controlling access to copyrighted media
        + product keys
    - watermarking
      * embedded data bits
        + steganography, must be robust to resist editing/removal attempts
    - geographic access requirements
      * geoblocking
        + local or server based
        + software/firmware
      * can be countered with VPNs, proxies
  + Data Loss Prevention
    - * sensitive data, concerned with external parties in an authorized manner
      * integrate DLP into risk management process
    - Data Inventories
      * find and characterize all data
        + data classification nontechnical control
    - Data Flows
      * specific network pathways, sensors throughout network
        + not just perimeter
    - Implementation, Testing, and Tuning
      * verify allows authorized data processing and to ensure prevents unauthorized data processing
        + continuous maintenance and improvement
    - Network DLP
      * apply DLP to data in motion
        + perimeter of network
        + internal subnet boundaries
        + additional security modules
    - Endpoint DLP
      * data at rest and in use
        + software-based DLP agent
      * data is observable at point of creation
        + detect possible removable media violations
    - Hybrid DLP
      * NDLP and EDLP
* Chapter Review
  + coordination of technical and nontechnical controls
    - use classification standards to ID, handle, and protect data
      * according to sensitivity/criticality
  + legal, regulatory, operational reasons
    - limit what data is acquired, who has access, and for how long

Chapter 20 – Security Concepts in Support of Organizational Risk Mitigation

* Business Impact Analysis
  + - understand the business of the organization and determine how it relies on the information systems to determine risks
      * select and implement context-specific controls to mitigate
    - functional analysis to develop a hierarchy of functions and apply a classification scheme
      * consider
        + max downtime/disruption
        + operational disruption/productivity
        + financial considerations
        + regulatory responsibilities
        + reputation
    - gather info from proper/qualified sources
      * people part of BIA data – gathering sessions
      * process flow diagrams
        + criticality classification
    - ID required assets and respective vulnerabilities and threats
* Risk Assessment
  + - ID vulns and threats, assess their possible impacts
      * ensure security is cost-effective, relevant, timely, and responsive
        + too much, too little, wrong type
    - determine probability, potential impact, and provide economic balance between impact of threat and countermeasure
  + Risk Identification Process
    - Cyber Threat Intelligence
    - Vulnerability Assessment
    - Cybersecurity Operations
    - Brainstorming Sessions
    - Threat Working Group (TWG)
      * Risk Registers
        + owners assigned responsible for mitigating risk
  + Risk Calculation
    - * quantitative 🡪 rigorous; monetary/numeric values
      * qualitative 🡪 softer; categorically describe risk elements
        + risk likelihood/impact matrix
      * Delphi/Hybrid
    - Probability
      * quantitative
      * qualitative
      * important to consider context of datasets
    - Magnitude
      * quantitative approach
      * qualitative 🡪 delphi, open discussion TWG, hybrid
      * impact of a risk
        + impact interpretation map
      * Putting It All Together
      * Communication of Risk Factors
        + present factors to stakeholders

document and communicate

* + Risk Prioritization
    - transfer, avoid, reduce, accept
    - insurance, terminate risky action, reduce, final step
  + Security Controls
    - * administrative 🡪 soft controls 🡪 policies, procedures, training
      * technical 🡪 logical 🡪 software/hardware components
      * physical 🡪 guards, locks, fencing, lighting
    - Assign Risks
      * transfer, avoid, mitigate, accept
        + apply correct control type
    - Technical Control Review
      * is control version up-to-date
      * configured properly
      * proper access controls
      * updated licenses/controls
    - Administrative Control Review
      * consistent with laws, regulations, policies, directives
      * relevant members aware
      * onboarding and refresher controls
      * staff abiding by controls
  + Engineering Tradeoffs
    - balance impact on performance of systems
      * deliberate balance of security and performance
        + neither optimal, both acceptable
* Documented Compensating Controls
  + - compensate for lack of a direct control
      * documented separately or included
* Systems Assessment
  + - trust but verify
    - gather info and make determinations
      * audits, vuln scans, pentests, red teams, risk assessments, tabletops
* Supply Chain Risk Assessment
  + scope of assessments and possible actionable measures
    - supply chain map
  + Vendor Due Diligence
    - exercise of reasonable expectations of care to prevent liability
    - due diligence of researching a vendor or partner before creating supply chain
      * assess risk of a business relationship
  + Hardware Source Authenticity
    - prevent malicious features and poor quality performance
      * authorized retailers
      * serial number checking
      * price/quality continuum
* Training and Exercises
  + - fast, reflexive
    - slow, analytical
    - develop and maintain skillsets, knowledge, attributes via exercises and training
      * achieve operational and managerial goals
  + Types of Exercises
    - Tabletop Exercises
      * TTXs
        + executive or team level
        + test procedures and intended outcomes
        + likely scenarios, branch scenarios, sequels must remain focused and controlled
    - Live-Fire Exercises
      * LFX
        + real/simulated defense scenarios

red vs blue

* + - * virtualized copies to practice, still lack pattern of life, significant resources
        + no POL realism

require another part to create traffic or traffic injector/autonomous agents

* + Red Team
    - require highly skilled members to be effective
      * costs, availability
  + Blue Team
    - require proper members, not just those who are available
      * executive involvement and support
  + White Team
    - plan, document, assess, moderate exercise
      * after action review
* Chapter Review
  + risk must be revisited periodically and deliberately/formally
    - vuln assessments, personnel preparedness, and training

Chapter 21 – The Importance of Frameworks, Policies, Procedures, and Controls

* Security Frameworks
  + - two types of frameworks
      * risk-based
      * non-risk based/prescriptive
        + overarching focus
        + specific control focus
  + NIST
    - * Risk Management Framework (SP 800-37)
      * Cyber Security Framework (CSF)
      * NIST SP 800-53
    - Risk Management Framework
      * Three Tiers to Risk Management
        + Organization Tier

business as a whole

governance, program-related risk management

* + - * + Mission/Business Processes Tier

risk to major organizational processes

cross-organizational process risk common to multiple business units

* + - * + Information Systems Tier

most common area of focus

* + - * Multiple Layers of abstraction to describe issues at different levels
        + four components

frame risk 🡪 context 🡪 assumptions, constraints, priorities, tolerance

assess risk 🡪 most critical 🡪 must assess properly

respond to risk 🡪 matching limited resources with prioritized set of controls

monitor risk – monitor effectiveness against risks designed for

* + - * + align tasks with appropriate components
    - Framework for Improving Critical Infrastructure Cybersecurity
      * + flexible, repeatable, and cost-effective
      * Framework Core
        + five functions, 22 categories, 98 subcategories
      * Implementation Tiers
        + rigor, sophistication of cybersec practices
      * Framework Profile
        + state of an org in regards to CSF cats/subcateg

as-is vs to-be

* + - * Core Practices
        + ID
        + Detect
        + Protect
        + Respond
        + Recover
    - SP 800-53
      * NIST
      * outlines Federal Information Processing Standards
        + select appropriate security control baselines
        + tailor baselines
        + document security control selection process
        + apply controls
      * categorize infosystems based on criticality/sensitivity and CIA
        + which of the 965 controls should be applied

document the selection process, necessary to assess control effectiveness and auditing purposes

* + ISO/IEC 27000 series
    - ISO 27001
      * formalization of security activities through the creation of an info security management system
        + 114 controls/14 domains, similar to SP 800-53 but also with overarching concerns
    - ISO 27005
      * similar to RMF but no specific control methods, use with 27001
      * risk management process 🡪 context
  + Center for Internet Security Controls
    - * 20 critical security controls to mitigate majority of attacks
        + prescriptive controls/grouping

171 subcontrols

* + - * Divided Into Three
        + Basic – minimum essentials
        + Foundational – best practices
        + Organizational – people and processes
* Policies and Procedures
  + - beginning of a framework
      * scope defined by management via regulations, laws, and liability
      * long term strategy
    - detailed the better
  + Ethics and Code of Conduct
    - tone at the top
    - mission/vision statements
  + Acceptable Use Policy
    - use of information systems available to employers
      * first line of defense
  + Password Policy
    - most visible of security policies
      * generation, duration, use
        + min length
        + special characters
        + no reuse or common words
        + min/max age
  + Data Ownership
    - combined with data classification policies
      * roles/responsibilities (due care)
        + management
      * necessary controls, access, usage criteria
        + personal data use on site premises
  + Data Retention
    - legal and regulatory requirements
      * adhere to a formal policy, confirm with audits
      * balance storage costs, e-discovery orders, data-specific time requirements
        + avoid one size fits all
  + Work Product Retention
    - materials collected/prepared in anticipation of litigation
      * refer to data retention policy, consult legal counsel
  + Account Management
    - strong authentication
    - limit private account usage
    - monitor creation, modification, misuse/use of accounts
    - well-defined onboarding process
      * expiration dates
      * password policy
      * info access/user permissions
    - change-control process, elevated privilege minimization, account suspension
  + Continuous Monitoring
    - SP 800-137
      * ongoing awareness of infosec, vulns, and threats
    - process of collecting/analyzing info to maintain situational awareness
      * leads to creation/enacting remediation plan
* Control Types
  + - * reduce risk via managerial, technical, and operational
        + managerial 🡪 planning, risk assess, sec assess, systems acquisition process
        + technical 🡪 software/hardware
        + operational 🡪 people
      * also physical security controls
      * defense-in-depth
        + firewalls, IDS/IPS, AV, ACLs, encryption
      * all mapped to relevant threats and asset value
    - Control Functions
      * Preventative
      * Detective
      * Corrective
      * Deterrent
      * Compensating
* Auditing and Assessments
  + - ensure proper implementation
      * cybersec assessment
    - audits systematically inspected by a 3rd party
  + Standards Compliance
    - reduce risk, establish due diligence
    - meet pre conditions, compliances
      * ISO27000 and FedRAMP
  + Regulatory Compliance
    - threshold requirements
  + SOX 🡪 integrity protections, digital record keeping
  + PCI DSS
  + GLBA 🡪 protect consumer personal financial info
    - safeguards rule
  + FISMA 🡪 Infosys belonging/operated by federal govt and contractors
    - risk assessments, sec awareness training, incident response, ops continuity
  + HIPAA 🡪 security and privacy rules
* Chapter Review
  + develop formal documents that will ensure that the entire organization is working towards the correct goal

END