CySA+ CS0-002 Master Cheat Sheet

First, let's be clear about what this study guide is NOT. It is NOT comprehensive, and it is NOT intended to be enough for you to study off exclusively.

This guide is brief on purpose. This guide is an excellent resource to review information, to be reminded of terms you've learned about but may have forgotten, and to understand how to mentally organize the information so you can recall it easier.

CySA Broad Strokes

- Cybersecurity Analysis
    - Threat Detection
    - Data Analysis
    - Securing and Protecting Apps and Systems
- Four Domains
    - Threat Management
    - Vulnerability Management
    - Cyber Incident Response
    - Security Architecture and Tools
- CySA is for those who want to stay Technical, rather than the managerial track of CASP
    - This can actually improve the Managerial Track - by ensuring a foundation of technical skills.

Threat Management

Broad Strokes

- Identifying Threats
- Network Security Measures
- Understanding Response and Countermeasures
- Threats, Vulnerabilities, and Risk
- Footprinting and Recon
- Threats to Confidentiality, Integrity, and Availability
- Controls to secure networks and endpoints
- Evaluation of Security Controls
- Information Gathering (passive and active)

CIA Triad

- Seem familiar?
- Confidentiality - Integrity - Availability
- **Confidentiality**
    - How secure is info?
    - How secure does it need to be?
        - Public data should be public, but PII needs to be secure
    - Physical Protection
        - Doors, fences, security guards and cameras
    - Electronic Protection
        - Encryption, passwords, authentication, and firewalls
- **Integrity**
    - How correct is the information?
    - Has it been modified or corrupted?
        - Hashing and checksums help monitor and verify
- **Availability**
    - Is data always user-accessible?
    - Redundancy in storage, power, and transit helps improve availability
    - Backup strategies and disaster recovery alleviate problems
- CIA values don't need to be balanced. They must instead be designed around the needs of the system.
    - Sometimes availability is more important that integrity - such as for public info.
- **Security v. Operations**
    - Security can interfere with functionality.
    - Sometimes you have to increase risk to improve usability - again, focus on the system's needs.
        - An unplugged device in a cement cube in the ocean is secure, but how useful is it, really?

Risk Consideration

- Risk is the centerpoint of another triad: Threats, Vulnerabilities, and Assets.
    - If you have nothing worth stealing, you've got no risk.
    - If your system is flawless (cement cube in the ocean), there are no vulnerabilities.
    - If nobody wants your assets or has the means to go after them, you're free from threats.
- **Assets**
    - Information or Data
    - Network equipment
    - Servers/Computers
    - Software
    - Personnel

    - Processes
- **Vulnerabilities**
    - Any weakness in a system design, such as a bug, that allows an attacker physical or digital access
    - These weaknesses are internal factors. Patches, or additional security guards, can cover those weaknesses.
        - Sometimes a weakness is out of your control - such as when using proprietary software.
        - Your job is to compensate for those weaknesses
- **Threats**
    - Any condition that can cause harm, loss, damage, or compromise of an asset
    - Natural disasters, cyber attacks, or malware
    - These threats do not have to be intentional - mother nature is dangerous as well, and accidents happen.
    - Your job is to cover vulnerabilities appropriate to your threats - NOT to defeat the the threat itself.
        - You don't have to worry about Quantum Computers brute force hacking through your system if… Quantum Computers don't exist.

Risk Assessment

- Should be performed regularly to understand existing threats, vulnerabilities, and the appropriate mitigations.
- **NIST SP 800-30** is a framework to properly perform these assessments based on current technology
    - Prepare for Assessment
    - Conduct Assessment
        - Identify Threats and Events
        - Identify Vulnerabilities
        - Determine Likelihood of Occurrence
        - Determine Magnitude of Impact
        - Determine *Risk*
    - Communicate Results
    - Maintain Assessment

Identifying Threats

- **Adversarial Threats**
    - Consider capability, intent, and likelihood
    - Customers, foreign governments, suppliers, and competitors can all be considered
- **Accidental Threats**

    - Mistakes that hurt the security of the system
    - Fat-fingering (mistyping)
    - Accidentally taking a device home
    - Accidentally hitting a kill-switch, power button, fire-alarm, etc
- **Structural Threats**
    - Equipment, software, or environmental control failure
    - Hard drive failure, overheating, bugs and crashes
    - ST's are the reason redundancy is key!
- **Environmental Threats**
    - Natural or man-made disasters
    - Fires, floods, storms, loss of power, wire-cuts, etc
    - Another good reason for backups and redundancy
- Remember: Threats go beyond "attackers." Disgruntled employees, accidents, and bugs can all cause asset loss or compromise security
- Risks change!
    - Quantum Computers aren't a threat now, but they may be in a few years, and they'll drastically shift vulnerabilities.

Identifying Vulnerabilities

- Largely internal
- If you have a threat without a vulnerability - it isn't a risk.
    - Snowstorms are a threat… but not in Florida.
    - A Windows XP vulnerability is a threat… but not to a company that only uses IOS.

Likelihood, Impact, Magnitude, and Risk

- What's the likelihood a vulnerability will be exploited, and how bad will it be when it is?
    - Low, Medium, High, or Critical
    - These judgements are **qualitative, not quantitative**
        - You can't always put numbers to the issue, but you need to have an idea of how likely, or severe, risk is.
        - **ALE - Annual Loss Expectancy**
            - Cost x Occurences = ALE
            - Calculated per year.
            - What's the cost of mitigating the risk, vs. the ALE?
                - Some losses can be acceptable - if it's more expensive to fix than to just cover.
- How likely are you to be attacked?
- How likely is it that an attack will have a bad impact?

Risk Controls

- **Acceptance**
  - When risk is low, and/or countermeasures are expensive
  - A high risk that's been largely mitigated can then be accepted
- **Avoidance**
  - When risk is too expensive, and you completely avoid vulnerability
  - XP is no longer supported, so you move your company to Windows 10.
- **Mitigation**
  - Minimizing risk down to acceptable levels
  - Closing vulnerable ports, patching bugs, etc.
  - Everytime you drive, you take some risk. But if you wear you're seatbelt and drive the speed limit, you've mitigated the risk.
- **Transference**
  - The risk is unavoidable, but you don't want it
  - Insurance companies, basically.
  - Data Breach Protection Insurance - get paid back if you fall victim of cyber attack
- **Technical Controls**
  - Firewalls, IDS, IPS, antivirus, and endpoint security
- **Operational Controls**
  - Policies, pentest, SOPs, settings, and configurations

Network Perimeter Security

- **Firewalls**
  - Rest at network boundary
  - Triple-Homed Devices - connected to Internet, DMZ, and intranet
  - Filters the information as it passes between each "home"
- **DMZ - Demilitarized Zone**
  - Semi-trusted zone
  - Often home for servers that get traffic from the internet, but prevents internet from communicating directly to the intranet or trusted network
- **ACL - Access Control List**
  - Rules that define what traffic can pass through the firewall
  - Secure posture relies on **Implicit Deny** / **Explicit Allow**
    - All traffic is denied, unless there's a rule that specifically allows it
- **Firewall Types**
  - Packet Filtering
    - Check each packet against ACL rules for IP and Port

    - Stateful Inspection
        - Maintain information about the *state* of each connection
    - **NGFWs - Next Generation Firewalls**
        - Contextual information - recognizes users, apps, and processes to make decisions
        - Layer 7 of OSI model
    - **WAFs - Web Application Firewalls**
        - Protect web apps from SQL and Cross-Site Scripting attacks (SQL/XSS)
        - Placed right in front of web application server
    - *Go review your ports!!!!! [At bottom of doc]*
- **Network Segmentation**
    - Separate networks by security levels
    - Similar to Intranet / Internet / DMZ
    - Can be divided by physical location as well
    - Example: *[You'll wanna look at the graphic in Jason Dion's video]*
        - SharePoint and SQL in a datacenter on segment one
        - *Firewall 1*
        - DMZ and Remote Administration region on segment two
        - *Firewall 1*
        - Intranet Region on segment three (between *Firewall 1, and Firewall 2*)
        - *Firewall 2*
        - DMZ and Web Servers and Database Servers
        - *Firewall 2*
        - Internet

NAC - Network Access Control

- Limits access to authorized individuals and systems
- Ensures systems have proper antiviral, settings, and authentication
- **802.1x**
    - Common protocol for NAC
    - Supplicant ---> Authenticator ---> Radius Server
    - Supplicant <--- Authenticator <--- Radius Server
    - Agent-based, requires supplicant to use special software to communicate
    - Out-of-Band
- **Agentless NAC**
    - Conducted in web browser
    - Puts you in an isolated network segment until you authenticate
    - How Starbucks wifi works lol
- **In-Band**
    - Uses dedicated appliances between your device and the services
- **Out-of-Band**

    - Relies on existing network and has device communicate to authenticate
    - 802.1x
- **NAC Criteria**
    - Time of Day
    - User Role
    - User Location
    - System Health Status
        - Antivirus definitions, security patches, etc

Defense Deception Methods

- Honeypots
    - Designed to look lucrative or vulnerable
    - Wastes attackers time, or gathers their info
    - Can be used to improve defenses
- DNS Sinkholes
    - Provides false DNS info to malicious software
    - Detects suspicious requests, and redirects attacker to a deadend
    - Useful for preventing an infected host on your network from doing more damage

Secure Endpoint Management

- Hardening System Configuration
    - Disable unnecessary services and ports
    - Verify secure configs
    - Centrally control device settings (so user can't mess it up)
- Patch Management
    - Keep patches up to date to stay ahead of attackers
    - **SCCM** Microsoft System Center Configuration Manager
        - Central service that pushes patches to your devices
- Compensating Controls
    - Stop-gap measures
    - If you can't patch a service, you could disable it, or block a relevant port at the firewall
- **GPO - Group Policies / Group Policy Objects**
    - Allows Admins to manage system and security configs across many devices over a network
        - Require firewall usage
        - Run scripts at logic
        - Activate share drive
- **Endpoint Security Software**
    - Software that allows security analysts to enforce security policies across user devices, and often gather data from them as well

    - Antimalware or antivirus
    - HIDS or HIPS
- Additional Controls
    - **MAC - Mandatory Access Control**
        - Sets all permissions centrally, users cannot adjust them
    - **DAC - Discretionary Access Control**
        - Only the owner of a file or resource can control the permissions
    - MAC is secure, but an admin nightmare
        - Good for Need-To-Know style data
        - SE Linux

Penetration Testing

- Simulated cyber attack to test your defenses and vulnerabilities
- Goal is to gain access to your systems and report findings
- Pentesters
    - Company may have a dedicated **Red Team**
    - External Consultants
    - Requires highly skilled individuals
    - Time intensive and costly
- Phases
    - **Planning**
        - Read through resumes
        - News articles, open source content, etc
        - Do not touch the network
        - Figure out Timing, Scope, and Authorization with the owner of the network you're testing
    - **Discovery**
        - Port scanning, enumeration, vulnerability scanning, web app scanning
        - Plan around vulnerabilities
    - **Attack (Exploitation)**
        - Exploit vulnerabilities, loop back to discovery for further vulnerabilities
        - Gain Access
        - Escalate Privileges
        - Jump from System to System
        - Install Additional Tools
    - **Reporting**
        - Explain your findings after you've gone deep as you can
        - Describe successful tests, and possible solutions
        - List secure assets as well!
        - Prioritized based on risk posed by vulnerability

- Don’t PenTest without permission unless you wanna go to prison, dummy

Security Exercises and Testing

- Similar to pentest, but in simulated environment
- Trains practitioners on both Red Team and Blue Team
  - Red practices accelerating exploits
  - Blue practices blocking them out and keeping system up
- Can go much further than PenTesters would be willing to go in production
  - Deleting hosts, compromising apps, etc
- White Team
  - Plays as referee, arbitrates disputes, enforces rules, maintains environment

Reverse Engineering

- Taking a finished product and dismantling it until you understand its inner workings and components
- **Dynamic Analysis**
  - Launch malware in virtualized environment and see what it does
  - What ports does it communicate on?
  - What websites does it reach for?
  - Some Automated systems can use dynamic analysis to check for malware in attachments, emails, etc
  - Quickest way to discover the EFFECTS of malware
- **Static Analysis**
  - Analysis of the code of the malware
  - Easy if the code is in interpreted language like Python or Ruby
  - Difficult with compiled code like C/C++ or Java
    - Requires a decompiler or binary for compiled code
- **Hardware Reverse Engineering**
  - Difficult due to device firmware
  - Usually use dynamic analysis
  - Hardware should come from a trusted source to ensure security
  - Refurbished or second-hand devices can be compromised with bad firmware

Recon and Intelligence

- Several security standards and laws, such as PCI-DSS, that require regular vulnerability scanning and informationg gathering

Footprinting the Network

- Create a map of network, systems, and company infrastructure
- NIST SP 800-115 and **OSSTMM - Open Source Security Testing Methodology Manual** house instructions on this process
- **Active Reconnaissance**
    - Utilizes host scanning tools such as **NMAP** to gather info about systems, services, and vulnerabilities in network
    - NMAP can rely on responses to TCP/IP stack fingerprints to generate an Operating System Fingerprint
        - Identifying an OS by its response to TCP calls
    - Only identification, not methods of exploitation
    - Permission must be sought before conducting active recon
        - Scans can indicate an attack
        - Contract with proper scope-of-work is your protection
- **Network Mapping**
    - Utilizes **TTL- Time To Live**, **traceroute** and other responses to gather information
    - **Zenmap** converts NMAP info into graphical data
- **Challenges to net mapping**
    - Firewalls and Layer 3 Switch ACLs can block Nmap queries such as ping
    - Wireless devices can pop in and out at different locations
    - Virtualized devices may be hidden behind the physical device
    - Cloud services could be unscannable
- **Port Scanning**
    - Host Discovery
    - Port and Service identification
    - Service Version ID
    - Operating System ID
    - Useful for inventory tasks and security audits - confirming what's actually running, accessible, etc
- **Service Scanning**
    - Looks at the banners or packet responses of data to identify what is running and on which port
    - Judging by responses to known signatures, you can sometimes identify specific versions of individual services
    - Review your port info again! (bottom of the doc)
        - *Specifically "well-known" vs. "registered" ports*
    - Where you scan from matters - external scans will be blocked from more issues by NAT, ACLs, and firewalls
        - Pentests are best conducted from the outside
        - Vulnerability tests are best conducted from inside

    - The more information you get - such as version of the service - you can isolate specific vulnerabilities much easier.
- **Alternate Port Scanners**
    - Angry IP
        - Multiplatform
        - Graphical
        - Provides less OS and service info by default than NMAP/Zenmap
    - Metasploit
    - Qualys Vulnerability Management
    - Tenable's Nessus Vulnerability Scanner
- Great Reference Guide for Common NMAP commands:
    - https://hackertarget.com/nmap-cheatsheet-a-quick-reference-guide/
    - SYN scans - Fast scans usually use -sS but require raw socket access
    - TCP scans- -sT scans don't require raw access
    - UDP scans- -sU

Passive Reconnaissance

- More difficult than active recon, requires more data-digging
    - Utilizes logs, packet captures, etc
    - Data may be out of date
    - Useful for responding to a cyber incident without alerting the attacker that you're analyzing the attack
- Log and Configuration Analysis
    - Local data can be read through, or parsed, to create a network map
    - Log files, system config files, etc
    - Much of this is manual labor
- **Network Devices**
    - Log activities, status, and events
    - Include traffic patterns and utilization
        - You can potentially see where the attacker is, and what they're doing
    - Helpful to have a centralized location that gathers the log files from all available devices, possibly through SNMP
- Logging Fault Levels - **Log Levels**
    - 0 - Emergencies - Failure causing a shutdown
    - 1 - Alerts - Immediate Action Required (Overheating)
    - 2 - Critical - Software Failure
    - 3 - Errors - Interface up/down
    - 4 - Warning - Config Change
    - 5 - Notifications - Protocol up/down
    - 6 - Information - ACL violation

    - 7 - Debugging
    - The higher level you log, the more data you'll fill up, but the more you'll have to work with when troubleshooting an incident
    - **Remember: 0 is worst, 7 is informational**
- Go get an example of reading through a log!!
    - **Important Info to Parse:**
        - Allow or Deny
        - Type of Traffic: TCP/UDP
        - Source
        - Destination
- **Configuration Files**
    - Identify all routes and devices in detail
    - Provides details of SNMP and SYSLOG servers on the network
    - User and Admin accounts
    - ACLs
- **Netflow** Data
    - Captures IP traffic info to provide flow and volume
    - IP, source, destination, and class of service
    - Doesn't provide packet details, but lets you easily monitor changes.
        - You can sometimes identify an attack if your data volume triples all of the sudden for no known reason
- **Netstat**
    - Identify active TCP/UDP connections
    - Identify process using a connection
    - Stats on sent/received data
    - Route Table Info
    - **Netstat - a**
        - Provides active TCP/UDP connections
    - **Netstat -0**
        - Identify process using a connection
        - Allows you to correlate malicious process to malicious host
    - **Netstat -e**
        - Volume of flow over ethernet
    - **Netstat -r**
        - Routing table information
- **DHCP Logs**
    - Dynamic Host Configuration Protocol
    - IP Address, Default Gateway, Subnet Mask, and DNS server
    - Combined with firewall logs and network logs, you can tell which hosts are using which IPs, and how often those IPs are changing
- **Firewall Logs and Configs**
    - Configs

- What’s allowed, what’s blocked
- Clearer than log files
- Logs
    - Use levels to categorize info and debug messages
    - Date/Time Stamp & Details
    - Designed for human readability
        - Kinda
    - https://www.cisco.com/c/en/us/about/security-center/identify-incidents-via-syslog.html

- **System and Host Log Files**
    - Provide info on system config, applications, and user accounts
    - You need access to the system to gather these logs
    - **Windows System Logs**
        - Application Logs
        - Security Logs
            - Login events, resource usages, files created/deleted
        - Setup Logs
            - Installs
        - System Logs
            - Events from Windows OS
        - ForwardedEvent Logs
            - Any activity performed remotely
    - **Linux System Logs**
        - /var/log directory
- **DNS Harvesting**
    - DNS info is publically available
    - **Whois** can tell you who registered a domain, their address, email, etc
    - **Hostnames** can tell you about the server itself
    - **Nslookup**
        - Provides IP addresses
    - **DNS Records**
        - MX - Mail records
        - A - Address Records
        - AAAA - IPv6 address
        - C - Canonical Records
        - PTR - Pointer Records
    - **Tracert - Trace Route**
        - Shows each hop from a host to a destination
- **Domain Names and IP Range Review**
    - Human Readable names used to locate servers
    - Top level domains
        - .com .net .org .edu .mil .goc

  - Country Code domains
    - .com.uk .edu.it
  - Five Regional Authorities
    - AFRINIC - Africa
    - ARIN - US/Canada/Antarctica/Caribbean
    - APNIC - Asia/Australia/New Zealand
    - LACNIC - LatAm, Caribbean
    - RIPE - Europe/Russia/Middle East
    - Each authority provides Whois services for their IP space
- **DNS Zone Transfers**
  - Replicate DNS databases between two DNS servers
  - This is vulnerable, so most only allow zone transfers to trusted servers
  - **Dig** can allow you to perform the transfer
    - **'Dig axfr'** is the command for a zone transfer - LOTS of data
    - Digininja proves a few DNS servers that allow zone transfers for testing
      - Nsztm1.digi.ninja and nsztm2.digi.ninja
  - **DNS Brute Forcing**
    - Using manual or scripted DNS queries for each IP of an organization to gather data
- **Whois and Host commands**
  - Whois searches a database for domain names and IP blocks
    - Provides detailed registration information such as names, phone numbers, and physical addresses
    - This information can be blocked, generalized, or falsified
    - Historical whois data can be helpful
  - **Host**
    - Provides info about a systems ipv4 and ipv6 addresses and servers
    - Search domain, get related info

Info Gathering and Aggregation

- Packet Captures
  - Requires an intruder to gain access to a network first
  - Treasure trove of info
  - Utilizes tools like Wireshark
- theHarvester
  - Gathers emails, domains, hostnames, employee names, etc across a network
- Maltego
  - Build relationship maps between people using facebook and other publicly available info
  - Allows you to use social engineering to phish up and up a company's ladder
- Shodan

    - Helps locate internet-of-things devices and identifies their vulnerabilities
    - If attacker can identify a vulnerable device of someone they've identified, they have a toehold
    - Can they access someone's webcam and watch them type a password?

Organizational Intelligence (Social Engineering and Real World Monitoring)

- Locations of facilities and buildings
    - Physical security posture from google earth
    - Business hours
- Work routine of organization
- Organizational charts
    - Departments and hierarchies
- Documents
    - Metadata - author's name and software version
    - EXIF Data - geolocation coordinates on photos
    - Scrub metadata and exif before making anything public
    - Outdated docs can be found on the internet archive, time travel service, google cache, etc
        - These docs may still contain vulnerable metadata
- Financial Data
- Employee personal info
    - **Immersion**
        - MIT app that allows you to go through emails to figure who someone talks to, how often they talk to them, etc.
    - Social Media
        - People are too free with info, and too willing to make friends
    - Public Record Search
        - Credit scores, background checks, home address, etc
- All this data can be used to improve attacker's credibility during phishing and social engineering exploits

Detecting, Preventing, and Responding to Recon

- Recon doesn't guarantee a successful attack
- **Detecting Recon**
    - Perform Data Collection for analysis
    - **Anomaly Analysis**
        - What's different, what's abnormal?
    - **Trend Analysis**

        - Identify future problems based on past data
    - **Signature Analysis**
        - Fingerprint or hash used to detect threats
    - **Heuristic Analysis**
        - Detect threats based on behavior
        - Identify unknown threats
    - **Manual Analysis**
        - Human expertise trolling through logs
- **Preventing Recon**
    - Employ firewalls and network defenses
    - Limit external exposure of services
    - Utilize an IPS to limit or stop scanners
    - Utilize monitors and alerts for signature/heuristic/anomalous activity
    - Control info release
    - Blacklist abusive services
    - Use CAPTCHAs to prevent scripts and bots
    - Utilize third-party registration for your domains and IPs
    - Set rate limits for lookups and searche
    - Avoid publishing zone files
    - Educate your users about social media risks

Domain Vulnerability Management

- Broadly: How do you scan for vulnerabilities, and respond to them within the context of your company.
- “Identification, prioritization, and remediation of vulnerabilities before a threat can exploit them.”

Regulatory Requirements

- Ooh boy isn’t this a fun one
- HIPAA, GLBA, FERPA - govern info storage and processing
- PCI DSS, FISMA - require vulnerability management program
- All set out requirements based on the kind of business you do and the data you handle
- Corporate Policy can mandate additional requirements

PCI DSS - Payment Card Industry Data Security Standard

- Security controls for credit card processors and merchants

- Most specific of any vulnerability management
- Vendor-driven, not legally mandated
- EX:
    - Internal and external scans must be conducted
    - Scanned at least quarterly, and after all major changes
    - Internal scans by qualified personnel
    - External scans by approved scanning vendor
    - High-risk vulnerabilities must be remediated until a “clean” report is achieved
- **PoS Malware - Point of Sale Malware**
    - Malware that circumvents encryption by stealing credit card info directly from memory

FISMA - Federal Information Security Management Act

- Security controls for govt, or anyone handling govt data
- Systems are classified as low, moderate, or high impact
    - Requirements based on those classifications
- Objectives designed around CIA Triad
- EX:
    - Scan system when new threats emerge
    - Utilize interoperable tools and techniques
    - Analyze scan reports from assessments
    - Remediate vulnerabilities based on risk
    - Share findings with other agencies to mutually eliminate vulnerabilities

Scanning

- **Scan Targets**
    - What do you scan, and why?
        - All systems, or merely critical?
        - Time and effort
    - What tools do you use?
        - **QualysGuard** can be used to build an automatic **asset inventory**
        - Admins can take that asset inventory and set priorities for scans
- **Scan Frequency**
    - Continuous, daily, weekly, yearly?
    - Determined by goals, requirements, and capacity
    - Automated reports can save time and effort
        - **Tenable’s *Nessus* Vulnerability Scanner**
        - Automatically identifies vulnerabilities
- **Scanning Tools**
    - **QualysGuard**

        - Port scans, vulnerability scans, scheduling, asset management, etc
    - **Nessus**
        - Port scans, vulnerability scans, scheduling, asset management, etc
        - **H**as default policies to meet certain regulatory requirements
    - **Nexpose**
        - Port scans, vulnerability scans, scheduling, asset management, etc
    - **OpenVAS**
        - Open-source, low cost, good for home network security
    - **Nikto**
        - Web Application Scanner
        - Other tools are good on the database and network, but Nikto supplements by looking at the code of the app
    - **Microsoft's Baseline Security Analyzer**
        - Client-side to monitor for updates, registry changes, firewall, hashing, etc
        - Home use, not effective for central network scan
    - *Be prepared to recognize these products, more than their details!*
- **Scanning Scope**
    - What networks and systems are covered
    - What tests are performed on each asset
    - Staff and management should know what/when is being scanned
    - **Minimizing the Scope**
        - Network segmentation allows you to scan smaller clusters to achieve regulation compliance
            - If only one machine is processing credit cards, only it needs to meet PCI DSS standards **IF** you segment it properly
        - Increases security, and decreases labor
- **Configuring Scans**
    - Scheduling
    - Producing reports
    - Authenticated Access for Scans
    - Plugins and Scan Agents
    - Scan perspective: internal v external
- **Scanning Sensitivity**
    - Anyone else think scope, targets, and sensitivity overlap hard?
    - Certain targets, such as production environments, require lighter scans or **safe scans** to prevent taking them down during production hours
    - Some scans can disrupt systems or cause loss of data
    - **Plugins** can be grouped by "family" to focus on certain environments
    - **Templates** can be used to group settings and plugins for certain situations/environments/times.
        - Useful if you have a light weekly scan, and a heavier monthly scan, etc
        - Prevents config errors

    - **Nessus** has default policies to meet certain regulatory requirements
- **Scanning Perspective**
    - Insider threat viewpoint
    - Attacker threat viewpoint
    - Different perspectives may highlight different issues
    - Some regulatory bodies require both internal AND external scans
    - Useful to get the internal worked out before you hire an external group
- **Authenticated/Credentialed Scanning**
    - What Rights does the scanner have while traversing servers, apps, firewalls, etc
    - Without credentials, scanner can appear as an attacker
    - Scanner should have **read only** rights so that if it becomes compromised, it's still limited
    - **Agent-Based Scanning**
        - Install agent on each client to provide an "inside-out" perspective of vulnerabilities
        - Data then sent to centralized server for review
        - Can be resource intensive, but provides a very detailed view
- **Maintaining Scanners**
    - Scanners *must* be updated before use
    - They can become vulnerable themselves, but also need the latest signatures to catch up-to-date threats
    - Vulnerabilities are unavoidable, but can be managed
- Other Considerations:
    - **Organizational Risk Appetite**
        - How much risk are you willing to handle?
        - Determines scan frequency
    - What Regulatory Requirements do you have?
    - How long does a scan take can determine how often you can scan.
    - Scans may be limited to evening hours due to network latency during business hours
    - Licensing Limitations: How many scanners are you paying for?
- Best Practices:
    - Start small - core assets
    - Expand slowly - add scope and monitor the expense
    - Prevent overwhelming the enterprise system, network, and sysadmin team

Remediation

- **Standardizing Vulnerabilities**
    - **SCAP - Security Content Automation Protocol**
        - Led by NIST SP 800-117
        - **CCE -** Common ***Configuration*** Enumeration
            - Naming convention for system config

- **CPE** - Common ***Platform*** Enumeration
  - Standard names for products and versions
- **CVE - Common Vulnerabilities and Exposure**
  - Standard names for security-related software flaws
- **CVSS - Common Vulnerability Scoring System**
  - Standard approach for categorizing severity of software flaws
  - 10 = most critical
  - 1 = least critical
- **XCCDF** - Extensible Configuration ***Checklist*** Description Format
  - Checklist and reporting standards
- **OVAL -** Open Vulnerability and Assessment Language
  - Low-level testing procedures for XCCDF checklist

- **Workflow for Remediation**
  - **Vulnerability Management Lifecycle:** ...Detection -> Testing -> Remediation -> Detection...
  - Continuous Monitoring provides for early detection
  - Automation
    - Many scanner products can automatically create tickets for remediating detected vulnerabilities, and automatically close when vulnerability is fixed
- **Vulnerability Reporting**
  - Analysts need to communicate known issues to SysAdmins
  - Scanners provide detailed reporting that can automatically alert sysadmins periodically
  - Low-priority vulnerabilities can wait, but critical must be communicated immediately
  - **Dashboards** provide a high-level summary that's easy to understand at a glance
    - Can indicate priorities, trends, etc
    - Host Overview allows you to see which hosts are most vulnerable
      - Useful for allocating remediation resources
    - Overview of Criticality shows the worst vulnerabilities at the top
- **Remediation Priority**
  - CVSS scores and priorities can help you know what vulnerabilities are worst, but you can't fix everything, and some fixes cost more time, money, and resources
  - How critical is the system and the information it contains?
    - If a system has a lot of PII, financial, or classified data, it needs fixed.
    - If all the data is encrypted, it might be less dangerous if it's accessed.
  - How difficult is it to fix the vulnerability?
    - If you can fix four vulnerabilities for the same cost as one, well… prioritize
  - How Severe is the vulnerability?
    - CVSS score helps here.
  - How exposed is a server to that vulnerability?
    - If an external facing server has a moderate vulnerability, it might pose a greater risk than a critical vulnerability on an internal server.
  - A lot of these are judgment calls, rather than clear-cut.

- **Implementing and Testing**
    - Vulnerability Analysts don’t implement fixes
        - Sysadmins do.
        - Larger fixes need to be run by a Change Control Board
        - Fixes must be tested in a lab environment to ensure they don’t break things worse
    - Analysts view fixes as the highest priority, but not everyone does
        - Fixes often must avoid causing service degradation or breaking promises to customers
        - Scanning, patching, etc can slow, or take down systems
        - Operations and Security are always in a struggle of balance
    - MOUs and SLAs
        - Security team needs to be involved in their formulation
        - Must address scope of security needs
        - Times for scanning, patches, etc
    - IT Governance
        - Even in an emergency, it can be necessary to get higher-ups to approve actions which will affect production to push fixes
- **Go Practice with a product like Nessus**

Analyzing Vulnerability Scans

- While scanners identify vulnerabilities, analysts must interpret those results
    - Eliminate False Positives
    - Identify root causes
    - Prioritize Remediation
- Parsing Reports
    - Identities
    - Synopsis
    - Description
    - See Also
        - References on the vulnerability
    - Solution
        - List of patches or contingencies for if your system is unsupported
    - Risk Factor
    - CVSS Score
        - 3.0 is newer, and not addressed on the exam. Recognize 2.0
    - STIG Severity
        - Military - cat 1 is critical, cat 3 is informational
    - References
        - Related vulnerabilities to the plugin
    - Exploitable With

        - Good way to know how prevalent the methods of attack are
    - Plugin Info
        - When the plugin to scan the vulnerability was made
    - Hosts
        - Where the vulnerability exists

CVSS Scores - Common Vulnerability Scoring System

- Wow these acronyms are horrible
- Industry standard for identifying severity of a vulnerability
- Measured in six categories, based on exploitability and impact
- Each of the below categories is assigned a value - these values are used in a formula to create the ultimate value between 0-10, 10 being the highest vulnerability
    - Exploitability
        - **AV Access Vector Metric**
            - **L**ocal
            - **A**djacent Network
            - **N**etwork (Remote Access)
        - **AC Access Complexity Metric**
            - High - Specialized conditions
            - Medium - “somewhat specialized”
            - Low - no specialized conditions
        - **Au Authentication Metric -** number of times attacker has to authenticate on access route
            - Multiple
            - Single
            - None
    - Impact
        - **C Confidentiality Metric**
            - None
            - Partial
            - Complete (loss)
        - **I Integrity Metric**
            - None
            - Partial
            - Complete (loss)
        - **A Availability Metric**
            - None
            - Partial
            - Complete (loss)
    - CVSS metric shows all of these values on a single line
        - AV:N = Access Vector, Network

- CVSS Score Categories *memorize this*
    - < 4.0 = low
    - > 4.0 and < 6.0 = medium
    - > 6.0 and < 10.0 = high
    - 10.0 = Critical
- **CVSS Temporal Score**
    - Base scores stay the same, but the temporal score changes as vulnerabilities are addressed and mitigated
    - **Exploitability**
        - U - Unproven
        - P - Proof-of-Concept
        - F - Functional
        - H - High
            - Typically when it can be automated, or is super easy
        - ND - Not Defined
    - **RL - Remediation Level Metric**
        - O - Official Fix
        - T - Temporary Fix
        - W - Workaround
        - U - Unavailable
        - ND - Not Defined
    - **RC - Report Confidence**
        - UC - Unconfirmed
        - UR - Uncorroborated
        - C - Confirmed
        - ND - Not Defined

Validation of Results

- Some vulnerabilities are false positives, documented exceptions, or informational
- **False Positives**
    - False Positive Error Rate
    - If a report says a patch is missing, check
    - Verify configs
- **Documented Exceptions**
    - Known-issues that you don’t plan to deal with or have properly mitigated
    - Implement exceptions in the scan so it doesn’t keep firing the alerts
- **Informational Results**
    - Not everything reported is a vulnerability
    - Some configs just allow attacker to perform some recon
- Compare results with other sources, like log files, config files, SIEM utilities

- Conduct Trend Analysis
  - Understand why you sometimes find more vulnerabilities (patch tuesday?)
  - Notice when sudden issues appear

Common Vulnerabilities

- There are thousands
- Five Basic Categories (the next headers)

Server and Host Vulnerabilities

- **Missing Patches**
  - If nobody installs a patch, it can't do anything
- **Unsupported Software**
  - Keep track of software EOL - End of Life
  - After that date, nothing gets patched (usually)
- **Buffer Overflows**
  - Tricking systems to run code or release data by forcing data into a section of memory
- **Privilege Escalation**
  - Pushing access from normal user, to root, admin, or superuser
  - An attacker can use a toehold to gain full control
- **Arbitrary Code Execution**
  - Allows attacker to run software on a system
- **Insecure Protocol Use**
  - Using unsecured protocols like FTP or Telnet
- **Debugging Modes**
  - Allows attackers to get a lot of background information for other exploits
  - Debug mode should be left off
- **PoS Malware - Point of Sale Malware**
  - Malware that circumvents encryption by stealing credit card info directly from memory

Network Vulns

- Missing Firmware Updates
- **SSL and TLS Issues**
  - Must use TLS 1.2 or newer
  - Must use current, secure ciphers
  - Certificates must remain valid and uncompromised
- **Domain Name Server Issues**
  -

- **Internal IP Disclosure**
  - Bad packet headers revealing information that should be hidden by NAT
- **VPN Issues**
  - Protocols, encryption tunnels can be vulnerable

Virtualization Vulns

- **VM Escape**
  - Break from the virtual machine, and reach the hypervisor (host)
  - Uncommon, but very dangerous
- **Management Interface Access**
  - Access to the configuration utility for virtual machines
- **Virtual Host Patching**
  - Host, and all guests must be patched
- **Virtual Guest Issues**
  - Vulnerability scans can't stop at the host, need to check the guests as well
- **Virtual Network Issues**
  - Virtual firewalls, routers, and switches must be patched and scanned

Web Application Vulns

- **Injection Attacks**
  - SQL inject is most common
  - Send commands through web server backend to bypass normal controls
  - Input Validation
- **XSS Cross Site Scripting**
  - When a website secretly executes a script
  - User is the usual victim, but this can open further vulnerabilities
- **CSRF Cross Site Request Forgery**
  - Cause a user to perform actions on a website they're already authenticated on
    - A website sees you're logged into your bank, and sends the command to your bank to transfer money out without you knowing
- **Web Application Scans**
  - Go beyond Nessus and Qualysguard - try Nikto within Kalilinux

IoT - Internet of Things Vulns

- **Smart Devices**
  - TVs, thermostats, or google homes that connect to the internet and can be hard to patch and secure
- **SCADA - Supervisory Control and Data Acquisition Systems**
  - Used in powerplants and such
  - Sensors and agents can remain unsecured

        - These entities should be on an isolated network segment
- **ICS - Industrial Control Systems**
    - Pumps, valves, and pressures controlled by a network computer
    - Firmware must be updated and secured

Cyber Incident Response

- Response to a security incident or event
    - Understand the incident
    - Mitigate negative effects
    - Plan the recovery
    - Investigate Root Cause

Security Incidents

- **Event -** Any observable occurrence in a system
    - Can be good or bad - such as logon event, or incorrect password event
- **Adverse Event** - Any event that has a negative consequence
- **Incidents** - An imminent threat of violation, or a violation, of a security policy, acceptable use policy, or security standard practice
    - Coworker logging in under your account credentials, against policy
    - Coworker downloads and installs malware
- **CSIRT - Computer Security Incident Response Team**

Incident Response Team - CSIRT

- Cybersecurity professionals with incident response experience
    - Temporarily include experts for specific subjects such as database trouble
    - Smaller orgs assign CSIRT roles as secondary roles
- **Management's Role**
    - Ensure funding, resources, and expertise available
    - Make critical business decisions
        - Mitigation vs. Production
    - Communicate with legal or news media
    - Communicate with Stakeholders
- Who?
    - Leader is skilled in incident response
    - SMEs (Subject Matter Experts)

    - IT Support Staff
    - Legal Counsel
    - Human Resource Staff
    - PR and Marketing Staff
- CSIRT can be **outsourced**
    - Must understand the third-party's guarantees, response time, and expenses
    - Third-party must be trusted
    - Scope must be clearly articulated
        - You could outsource analysis, but remediate in-house
- **Scope of Control**
    - What triggers CSIRT?
    - Who authorizes the activation?
    - What is each CSIRT's focus?
    - Can CSIRT talk to law enforcement?
    - Can CSIRT talk to media?
    - How does CSIRT escalate the issue?
- **Testing the Team**
    - Train through the plan
    - Simulate and pentest

Incident Response Phases

- NIST SP 800-61 is a handy guide, but not mandated
- Four phases are cyclical and feed into one another

Preparation

- Requires proper policy foundation
- Useless without existing, proper defenses
- Includes training with proper response tools
- **Preparation Tool Kits**
    - Digital Forensic Workstation
    - Forensic Software
    - Packet Capture Devices
    - Spare Servers/Network Gear
    - Backup Devices
    - Blank Removable Media
    - Collection, analysis, and reporting laptops
    - Portable Printers
    - Evidence Collection Materials
- Tools should be tested regularly

Detection and Analysis

- Hardest to Standardize
- Tools are helpful, but skilled analyst is necessary
- Analysts shift from detection, to validation, back to detection
- **Alerts**
    - IDS/IPS, SIEM, Anti-virus, etc
- **Logs**
    - From OS, services, apps, network devices, and network flows
- **Publicly Available Info**
    - News, media, and open-source info
- **People**
    - Reports from admins and users
- **Best Practices**
    - Understand Baseline
    - Create good logging practices
    - Conduct Event Correlation
        - Keep device times synced
    - Maintain Organization Knowledge Base
    - Capture network traffic ASAP during incident
    - Filter information to reduce confusion
    - Know when to consult experts

Containment, Eradication, and Recovery

- Stop the spread, remove it from network, and recover
- **Five Steps**
    - **Pick Containment Strategy**
        - Isolation, or shutdown?
    - **Limit Damage**
    - **Gather Evidence for legal action**
    - **Identify Attacker/Attacking System**
    - **Remove Effects of Incident, Recover normal Actions**

Post-Incident Activity

- Recreate a timeline of the incident
- Identify root cause of intrusion/incident
- Consult with sysadmins and management on findings
    - Utilize timeline and root-cause report to address vulnerabilities, improve response time, evaluate response successfulness, and prevent future attacks
    - What did we do well?
    - What can we do better?
- Evidence Retention

  - Understand legal requirements for what data you retain
  - Archive whatever you need to keep, usually 2-3 years minimum

Incident Response Policies and Procedures

- Foundation of Orgs Incident Response Program
  - Provides authority for response efforts
  - Approved by CEO/CIO
  - Should be fairly timeless and rarely need updating
- Contents
  - Statement of management commitment
  - Purpose
  - Objectives
  - Scope
  - Definitional Terms
  - Roles, responsibilities, and authority
  - Incident prioritization
  - Performance Measure for CSIRT
  - Reporting Requirements
  - Contact Info - by position, not name
- Incident Response Procedures
  - Detailed Info
  - Step-by-step guidelines
- **Playbook**
  - Describes response to high severity incidents
    - Data breach
    - Phishing Attacks
    - Web server compromise
    - Loss of corporate laptop
    - Network Intrusion
- Incident Response Checklist
  - You can just use the one in NIST SP 800-61
- **Communication**
  - How does CSIRT communicate among each other?
  - How will management communicate with employees?
  - Out-of-Band Communication is important if network is compromised
    - Don’t use a system that’s compromised
    - Sometimes, you just have a guy run down the hallway
  - When/How will you communicate with law and media?
  - Consider
    - Law Enforcement
    - Information Sharing Partners

        - Vendors
        - Collaterally Affected Orgs
        - Media or Public or Customers

Incident Classification

- Methods of Attack
    - Removable Media
    - Attrition
        - Brute-force
    - Web
        - Web app or website
    - Email
        - Attachments or Spoofing
    - Impersonation
        - Spoofing, SQL Inject
    - Improper Usage
        - Violation of Policy
    - Loss or Theft of Equipment
    - Unknown
    - Other
        - Known origin, but not quite a category
    - **APT - Advanced Persistent Threat**
        - Highly funded/skilled attackers that are willing to work overtime, or wait
        - Could have access that they aren't actively exploiting
- Severity
    - Functional Impact
        - none, low, medium, or high
    - Economic Impact
        - none, low, medium, or high
    - Recoverability Impact
        - Regular, supplemented, extended, or not-recoverable
    - Informational Impact
        - None, privacy breach, proprietary breach, integrity loss
        - OR - Regulated info breach, intellectual property breach, confidential proprietary breach

Network Event Monitoring

- Gather, correlate, and analyze data across systems
- **Router-Based Monitoring**
    - Provides data flow and status

  - Relies on capturing data *about* the traffic
- **Network Flows**
  - Netflow, sFlow, J-Flow
    - Samples traffic to find out connection types and speeds of data
  - RMON
    - Operates at layers 1-4 of OSI
    - Client/Server model with probes
    - Statistics, history, alarms, and events, reported to **MIB - Management Information Base**
  - SNMP - Simple Network Management
    - Collects info about routers/switches and centralizes them
    - Gathers info about *devices*, not *flow*
    - Only V3 is secure
- *Review Ports again!*
- **Active Monitoring**
  - Request is sent to a remote system which responds by sending data to a central location
    - Availability
    - Routes
    - Packet Delays
    - Packet Loss
    - Bandwidth
  - Ping/ICMP
  - iPerf
    - Measures max bandwidth of a network
    - Remote testing of a link
    - Useful for determining network baseline
- **Passive Monitoring**
  - Uses a network tap to copy all traffic between two devices
  - Useful between router and firewall
  - Useful for after-the-fact analysis
    - Rates of traffic
    - Protocols used
    - Content

Network Monitoring Tools

- **Wireshark**
  - Passive monitoring and packet capture
  - Useful for packet analysis
- **SolarWinds**
  - Netflow Traffic Analyzer
    - Demo.solarwinds.com

        - Great for seeing where flow is heaviest
    - Network Performance Monitor
        - What's up, what's down, what has errors
- **PRTG - Paessler Router Traffic Grapher**
    - Similar to solarwinds, but free
    - Packet sniffing
    - Flows
    - SNMP
    - WMI - Windows Management Instrumentation
- **Nagios**
    - GUI for network and system log monitoring
    - What's up, what's down, flow monitoring, etc
    - Nagios uses "criticals" that are not similar to NIST, but set by the user
- **Cacti**
    - SNMP polling of network devices

Detecting Network Events

- Analysis of logs and other data will allow analyst to determine when an event becomes an incident
- **Beaconing**
    - Significant warning of a malware or botnet infection
    - Pings/heartbeats that send consistently to a command and control center of an attacker
    - Typically over HTTP or HTTPS
    - Patterns can vary and be difficult to detect
    - Very common
- **Unusual Bandwidth Consumption**
    - Could be service issues, or a sign of larger trouble
    - First step is attempting to identify the cause of the spike
    - Compare unusual data to baseline network data
- **Link and Connection Failures**
    - Usually hardware, firmware, or software issues such as broken cable or unplugged connector
    - Could be a symptom of a Denial of Service attack
- **Unexpected Traffic**
    - Detected by IDS/IPS, traffic monitoring, or manual observation
    - Understand your baseline to identify what's unusual
    - Not all unexpected traffic is malicious, but should be investigated
        - Unusual country traffic
        - Unusual service traffic like VPN
    - **Baseline or Anomaly Based**

        - Baseline must be indicative of usual traffic - preferably over a long period of time
    - **Heuristic or Behavior Based**
        - Utilizes signatures or defined rules to identify unusual issues
    - **Protocol Analysis**
        - Is a protocol being used that you don't run usually?

Network Probes and Attacks

- Most incident handling is responding to reconnaissance probes like port scans
- **Denial of Service**
    - Ping of death, or other techniques to overwhelm your system
    - Block the source of the attack
- **Distributed Denial of Service**
    - Many sources attempting to overwhelm your system
    - Detectable by coming from known botnets, or unusual system data
    - **Defense**
        - Block the type of traffic
        - Utilize a distribute network, so you can take down an impacted segment
- **Detecting Rogue Devices**
    - Validate by Mac Address for familiar devices
    - Scan network to identify devices
    - Conduct physical site inspection
- **Rogue Wired Devices**
    - Something plugged into your network illegitimately
    - Block unused ports and validate MAC addresses with Network Access Control
- **Rogue Wireless Devices**
    - These devices can be hard to find
    - **Evil Twin** devices used to trick users to connect to them instead of your network
    - Utilize wireless surveys and maps to identify them

Server and Host Events

- Monitor CPU, memory, and drive usage
- Some attacks simply cause memory leaks to crash a server
- Windows resource monitor
    - **Resmon**
    - **Perfmon**
        - Allows remote tracking
- For Linux
    - **Ps**

        - CPU and memory utilization
    - **Top**
        - Similar to PS, but sorted
    - **Df**
        - Disk Usage
    - **W**
        - Account monitoring
- **Malware and Unsupported Software**
    - Use centralized management tools to inventory software and control installs
    - Antivirus and antimalware
    - Blacklist bad software and files
    - Could whitelist only the apps that you want
- **Unauthorized Access, Changes, and Privileges**
    - SIM/SIEM correlate logs for analysis
        - Authentication logs, user creation logs, security event logs, etc
        - Enables you to ensure nobody is using privileges or access they should have

Service and Application Events

- Are the apps up, running, responding, and logging properly?
- **Non-security issues**
    - Authentication issues
    - Permission issues
    - Services not starting on boot
    - Service failures
    - These issues can lead to security issues or be signs of one
- Windows
    - Services.msc or **sc** in the command line
    - Windows event viewer for logs
- Linux
    - Service -status-all
    - /var/log, tail the log files
- Service Application Behavior
    - Get yo baseline
    - Log/alert anything outside the baseline
- **Anomalous Activity**
    - Investigate and solve, identify as known-good, or known-bad
- **New Accounts**
    - Ensure they were authorized
- **Unexpected output**
    - Improper output or garbage output

- **Unexpected outbound communication**
  - Is a service reaching outside the network which shouldn't be?
- **Service Interruption**
  - Simple issue, or DDoS?
- **Memory Overflows**
  - Causes OS errors and crashes
  - Easier to analyze afterwords than to detect

Digital Forensics

- Determine changes, activities, or actions that have occured on a system
- Allows incident responders to determine what occured by collecting info
- **Documentation**
  - Must follow chain of custody and be properly handled
  - Chain of Custody is easier to maintain by having a second tech validate actions
  - Any data needs date, time, and method of collection
  - Proper handling is essential in case the incident is reported to law enforcement
- Forensic personnel should be trained and CERTIFIED for their evidence to be admissible in court

Forensic Toolkits

- Special software and hardware for disk imaging and analysis
- Free open source versions, or very expensive versions
- **Digital Forensic Workstation**
  - Powerful computer for data capture *and* analysis
  - 16++ gigs of ram
  - Lots of storage, preferably RAID
    - Must be capable of containing images of MANY computers
  - Powerful CPU
- **Forensic Investigation Software**
  - Software to capture and analyze forensic images
  - Forensic Toolkit (FTK)
  - EnCase
  - SANS Investigative Forensic Toolkit (SIFT)
  - The Sleuth Kit (TSK)
- **Write Blocker**
  - Could be hardware or software
  - Ensures integrity of captured disk by preventing its data from being written to or changed
  - Hashing improves this integrity
  - Hardware write blockers can be expensive, but are more secure

- **Forensic Drive Duplicator**
    - Simply copies a drive perfectly without wasting the energy of a workstation
    - Useful to have multiple if you've got a lot of big drives to copy
- **Wiped Drives or removable media**
    - Clean drives ready to receive disk images
- **Cables and Drive Adapters**
    - Everything's gotta get plugged in, and you don't know what you'll have on-site
    - Prepare for old tech and new tech
- **Digital Camera**
    - Document system layout and config, labels, etc
    - Good for fixing something if you have to make hasty changes during an attack
    - Pictures back up written documentation
- **Label Maker and Labels**
    - You can't just unhook stuff without keeping track of what it is, where it went, etc
- **Documentation and Checklists**
    - Playbooks, incident response forms, custody forms, checklists, etc
- **Mobile Forensic Tools**
    - **SIM Card Extractor**
    - **Connection Cables**
        - Lightning, 30-pin, USB-c, USB micro, an array of proprietary cables
    - **Mobile Forensic Software**

Forensic Software

- Imaging
    - FTK or EnCase or dd
    - FTK is free, and even documents chain of custody, hashes, and creates metadata tags for analysis
    - Always create a hash, and log it, immediately after capturing an image
    - Bit by bit copies preserve slack, or blank, space, preserving file layout and partitions
- Analysis
    - Creates timeline of system changes including hidden files and metadata changes
    - Validates files against known-good
    - Registry Analysis
    - Log file parsing and analysis
- Hashing/Validation
    - Chain of custody file integrity check
    - Should use MD5 or SHA1/SHA256
- Process and Memory Dumps
    - State of OS and currently running processes from memory
    - Difficult to collect without changing the contents

- Can capture decryption keys
- Hibernation files and crash dumps contain similar info
- **Tools**
  - Fmem and LiME (linux)
  - DumpIt (windows)
  - Volatility Framework (Any)
  - EnCase or FTK

- Password Cracking
  - Tools like John The Ripper or Cain and Abel
  - Some passwords can take forever to be cracked
  - DOC, XLS, PPT, and ZIP files have specialized tools that can crack those passwords

Forensic Process

1. What are you trying to find out?
2. Where would that information be?
3. Document your plan.
4. Acquire/preserve the relevant evidence
5. Perform initial analysis (log actions)
6. Conduct deeper analysis (log actions)
7. Report your findings

- **Order of Volatility**
  - CPU Cache, Registers, Running Processes, and Memory
  - Network Traffic
  - Hard Disk Drives and USB Drives
  - Backups, Printouts, Optical Media
- What do you do when you find something you didn't expect?
  - Evidence of illegal activities, or activities against policy
  - Stop everything
  - Call either management, or law, if relevant.
  - Seek guidance

Target Locations

- Windows Registry
  - Information about files and services, locations of deleted files, evidence of applications run
- Autorun keys
  - Programs set to run at startup
- MFT - Master File Table
  - Details of inactive/removed records
- Event Logs

  - Logins, services start/stop, evidence of apps being run
- INDX Files and Change Logs
  - Evidence of deleted files, Mac timestamps
- Volume Shadow Copies
  - Point-in-time information from prior copies
- User directories and files
- Recycle bin contents
- Hibernation files and memory dumps
  - Artifacts of run commands, possible encryption keys
- Temporary directories
  - Artifacts of software installs, user temporary file storage
- Removable Drives
  - System logs may indicate drives were plugging in
  - USB Historian

Incident Containment

- Containment can be quick and dirty
  - Can cause loss of business functionality
  - Coordinate with stakeholders to perform risk analysis - but quickly
- Segmentation
  - Isolate infected network segments, and try to cut them off from unaffected segments
  - Routers and firewalls are typically the delineation marks
- Isolation or Removal
  - Remove infected segments entirely
  - Recognize you lose their function and perform cost benefit analysis
  - You can isolate segments by allowing them to continue to work, while disconnecting them from the rest of the network
- Objective of Containment
  - Is it worse to take a system offline, or leave it running to spread infection or allow an attacker to move further?
- Identifying Attackers
  - Is this important?
  - It might not matter as much as stopping the attack
  - It might be too expensive and difficult to be worth pursuing - especially if that’s not your business goal
  - Law enforcement might be willing to pursue it further, using the data you collected

Eradication and Recovery

- **Remove any artifacts of the incident**
  - Revert all changes

  - Restore from good backup, or rebuild system
- Restore network to full functionality
- Correct security deficiencies
- Remove malicious code, sanitize compromised media, and fix affected user accounts
- **Not**
  - Rebuilding the whole network
  - Buying all new equipment
- **Reconstruction and Reimaging**
  - Reimage or restore to before the attack, because you don't know what an attacker might have hidden
  - Consider if root cause might affect other systems
- Rescan and patch all systems
- **Sanitization and Disposal**
  - Clear - write all data to 0s
  - Purge - degauss or overwrite with 35x 0s
  - Destroy - This really speaks for itself
  - Take the more extreme measures if your data is very secure, and very vulnerable
- **Validation Effort**
  - Check everything against baselines
  - Ensure only authorized user accounts remain
  - Verify permissions for each user
  - Verify all system logging
  - Verify vulnerability scans are properly configured

Finishing the Response

- Change Management Process
  - Many changes may have been made quickly or hastily
  - Make sure you go back through those changes and document properly according to CM
- Lessons-Learned meeting
  - Document details, root cause, and solution
  - Conduct these meetings immediately after incident
  - Identify needed changes and plan to implement
    - This might require seeking permissions, funding, and resources
- Final Lessons Report
  - Established organizational "memory" for future techs to review
  - Useful for further legal action
  - Should Include:
    - Timeline
    - Root Cause
    - Location and description of evidence collected

- Actions taken for containment, eradication, and recovery
- Est. Impact to org in time and money
- Post-recovery validation results
- Documentation of Lessons-learned

Security Architecture and Tool Sets

- So many tools and not even one goddam hammer

Policy Frameworks

Policy Documents

- High level statements of intent
- Broad statements of security objectives
- Basically a catch-phrase.
- **Policy Examples**
    - Information Security Policy
    - Acceptable Use
    - Data Ownership
    - Data Classification
    - Data Retention
    - Account Management
    - Password
- Policy usually approved by the C-Suite or management
- **Standards**
    - Mandatory Actions, steps, or rules
    - Approved below C-Suite
    - Standards also exist across the industry, so can be borrowed
- **Procedures**
    - Step-by-step instructions to perform an action
    - Creates consistent methods and outcomes for security objectives
- **Guidelines**
    - Recommendations, not requirements
    - Flexible so users can adapt to unique sitations
    - Easily, quickly changed
- **Exceptions**
    - Framework should have method for granting “exceptions” to rules
    - Usually signed by higher managers, indicated within framework
    - Should understand:

- What rule is being broken
- Why its being broken
- Scope and duration
- Risks associated
- Risk Mitigations

Standard Frameworks

- Your company/team doesn’t need to build everything out manually - frameworks exist to simplify this process
- **NIST - National Institute Standard of Technology**
    - Describe Current Posture
    - Describe Desired State
    - Identify/Prioritize areas for improvement
    - Assess progress toward desired state
    - Communicate risk among stakeholders
    - EXCELLENT Overview of NIST
    - **Tiers**
        - Partial
            - Informal, Reactive
        - Risk Informed
        - Repeatable
            - Understands dependencies and partners
        - Adaptive
            - Formal, well-thought-out, good with partners, etc
        - TLDR: How well prepared your company is
        - https://www.cciitool.info/section/tier
    - **Risk Assessment**
        - Threats
        - Vulnerabilities
        - Likelihood
        - Impact
- **ISO 27001**
    - Used to be most common standard
    - International
    - Regulated companies are required to use this, but many switching to NIST
    - 14 Categories, go look ‘em up
- **ITIL - Information Technology Infrastructure Library**
    - Security Management Meets Service Business Needs
- **COBIT - Control Objectives for Information and Related Technologies**
    - Plan and Organize
    - Acquire and Implement

        - Deliver and Support
        - Monitor And Evaluate
    - Less popular than the others
- **TOGAF - The Open Group Architecture Framework**
    - 4 Domains: Business, Application, Data, and Technology - working together in harmony
        - ...but everything changed when the application nation attacked
        - **Technical Architecture** supports the other domains
        - **Business Architecture** defines governance and organization
        - **Application Architecture** includes the apps and systems
        - **Data Architecture** is company's approach to storing and managing assets
- **SABSA - Sherwood Applied Business Security Architecture**
    - Similar to TOGAF
    - Uncommon

Policy-Based Controls

- Physical Controls
- Logical Controls
- Administrative Controls
- Combining Control Objectives is obviously better
- No I'm not detailing these didn't you take Sec+?

Audits, Assessments, Laws, and Regulations

- Guidelines are worthless if you're not inspecting and enforcing them
- **Audit** - Formal, usually internal, review of security guidelines and procedures
    - Checks specifically to make sure things have been done right
- **Assessment** - Informal review of controls and procedures
    - Mostly asks about stuff, instead of checking specifically
- Confirm **Compliance** with an regulatory body over your particular data and systems
    - **HIPPA - Healthcare**
        - If you secure any info for patients or healthcare products, this is you
    - **GLBA - Gramm-Leach-Bliley Act**
        - At least the name is memorably
        - Financial controls and security programs
        - Designates a "responsible" individual - usually CFO or CIO
    - **SOX - Sarbanes-Oxley Act**
        - Involves the security around financial systems for publicly traded companies
        - Mostly to make sure companies can be audited properly without "accidentally" destroying their own info

- **FERPA - Family Education Rights and Privacy Act**
  - Privacy controls for educational records
  - Only students or teacher can access a students info
- **PCI DSS - Payment Card Industry Data Security Standard**
  - Contractual obligation, not a law
  - How you secure and handle credit services and data
  - Requires external audits of compliancy
- Data Breach Notifications
  - Usually state law
  - Reporting standards to customers so they can protect themselves
- Really just know the category of each acronym

Defense In Depth

- Security must be redundant and varied, to prevent any single point of failure and to slow attackers long enough to rebuff them
- **Layered Security Defense**
  - Data > Application > Endpoint Security > Network > Perimeter
    - Perimeter as outermost layer
  - Difficult to design without affecting business needs
- Four Design Models
  - **Uniform Protection**
    - Same level of protection for all systems
    - Best for smaller networks
    - Expensive for large networks
  - **Protected Enclaves**
    - Higher protection for more secure data
    - Credit ops has more than internal network, which has more than web server
  - **Risk or Threat Based**
    - Employing specific controls based on the threats and risks you're most worried about
  - **Information-Classification Based**
    - Map data protection to different classes of information
    - Secret, Classified, Top Secret, etc
    - Higher classifications get additional attention and security controls
  - **Combining Design Models**
    - WOW SURPRISE YOU WANNA USE THEM ALL

Types of Controls

- Controls prevent, detect, counteract, or limit security risks

- **Technical Controls**
    - Firewalls, IDS/IPS, Authentication Systems, Network Segmentation
- **Administrative/Procedural Controls**
    - Security through policies and procedures
    - Incident Response Plans
    - User Awareness Training
    - Account Creation Policies
    - Acceptable Use Policy
    - Legal Controls
- **Physical Controls**
    - Gates, fences, mantraps, and fire suppression systems
- **Preventative Controls**
    - Proactive measures
    - Stop an incident before it happens
    - Security Guards, antivirus, training
- **Detective Controls**
    - Designed to detect when an incident occurs, capture details about it, and send an alarm
- **Corrective Controls**
    - Reactive - incident response
    - Fix an issue when it occurs
    - Patching, backups, etc
- **Compensating Control**
    - Minimize threat to acceptable levels
    - Blocking ports on an insecure OS
    - Segmenting vulnerable software that you can't replace into a distinct network segement

Layered Network Defense

- AHHH THIS ENTIRE COURSE IS JUST EXCUSES TO PUT FIREWALLS IN DIFFERENT CATEGORIES
- Can be accomplished through
    - Network Segmentation
        - Compartmentalization (synonyms are FUN)
        - Increases availability and efficiency
        - Makes it harder for incidents to spread
        - Implemented through *firewalls,* routers, switches, and VLANs
    - *Firewalls*
        - **Single *Firewall* or Router**
            - Isolates a segment into a DMZ
            - Router must have good ACL
        - **Multiple Interface *Firewall***

      - Different ACL and rulesets applies to each interface, creating multiple network segments
      - Requires a fancy expensive *firewall*
    - **Multi-Firewall**
      - Different *firewalls* at each control point
      - Allows for more stringent controls
      - Can use multiple cheap firewalls, instead of an expensive one
  - Outsourcing Network Segments
    - Remote Services
      - Saas or PaaS rely on provider's security
    - Directly Connected Remote Network
      - Acts as an extension of your intranet
      - IaaS with direct point-to-point VPNs
      - Seems like its just part of your network, but really uses someone else's secured system

Layered Host Security

- Password and authentication
- Encryption
  - Data at rest
  - Security keys and passwords must be secured
  - Hashing required to maintain integrity
- Host based firewalls or IPS
- Data Loss Prevention software
- White-lists/black-lists
- Patch management
- Antivirus
- System hardening
- Configuration management
- File Integrity Monitoring
- Logging
  - Logs should be centrally stored
  - SIEM can help

Data Analytics

- We've been through this, scroll up man
- Be ready to correlate data from multiple systems to understand what's happening
  - Look at the timestamps, bruh
- **Splunk**
  - Syslogs, auth logs, app logs, event logs, and others combined
- **Trend Analysis**

    - WE HAVE BEEN THROUGH THIS
- **Historical Analysis**

Personnel Security

- Humans will ruin everything
- **Separation of Duties**
    - Each person can only do/access so much
        - One person authorizes a payment, someone else signs it
    - Makes it harder to commit fraud
- **Dual Control**
    - Two people need to perform a single action
    - Check requires two signatures
    - Safe requires two people's keycards
- **Succession Planning**
    - Fleshy meatbags die, which fleshbag moves into their place?
    - OH and employees can quit, too. Who else knows how to do their job?
    - Don't allow an employee to be a single point of failure, no matter their position
- **Cross Training**
    - Ensure people know more than just their own job
    - If someone quits, make sure you have people to cover
    - If a project gets too big, make sure people can help
- **Background Checks**
    - Make sure people aren't hidden criminals and in millions of dollars of debt
- **Mandatory Vacation**
    - It's hard to run fraud if you're not there
    - Also its a good test to make sure the company can run without you
- **Termination**
    - Make sure people can't burn the place down on their way out
    - Recover all of their devices
    - Disable all their accounts
    - Change any codes that they know
    - Make a checklist so this is the same procedure everytime

Outsourcing Concerns

- If you think YOUR humans are a risk, other people's humans are scarier
- Proper Vetting:
    - What background checks do you perform on the service provider?
    - What background checks does the provider use on their employees?
    - How do they handle internal issues and personnel?

- Access Control
    - What can they touch?
    - How is your data kept separate from another company's?
- Data Ownership and Control
    - Who owns the data?
    - How is it encrypted?
    - Does the service provider have direct access to that data or the keys?
- Incident Response and Notification Processes
    - What happens during an incident?
    - Will the provider notify you?
    - Will the provider handle it, or just call you in?

User Awareness Training

- Train your users
    - AUP
    - Threats faced by organization
        - Like phishing
    - How to report a security issue
    - Physical security concepts
        - Don't lend badges, don't let people piggyback through mantraps
    - BYOD Policy
    - Data handling requirements
        - What can they print
        - How do they dispose of datas, disks, etc
    - Best practices for passwords, emails, remote work, secure web browsing, etc

Analyzing Secure Architectures

- Attackers are always looking for a flaw
- Pentesters are always looking for single points of failure
- Understand goals and requirements, and check if controls meet those
- **Reviewing Architecture**
    - **Operational View**
        - How a function is performed or what its supposed to accomplish
    - **Technical View**
        - Focuses on technologies, configs, and settings
    - **Logical View**
        - Focuses on the connections and paths of the network
- **Common Issues**
    - **Single point of failure**
        - If this one thing breaks, does everything break?

-
    - **Data Validation and Trust**
        - Don’t assume data, both incoming and resting, remains valid
        - Integrity checks on data at rest with hashing
        - Data validation on any user generated data to prevent SQL injections
    - **Users**
        - Mistakes and abuse cause faults
        - Automate monitoring on users
        - Constrain user access to what they need
            - Users don’t need CLI access
        - Implement checks and balances on all permissions and accounts
        - User awareness training
    - **Authentication and Authorization**
        - Multifactor auth
        - Centralized account and privilege management
            - With checks and balances!
        - Monitor privileged accounts
        - User awareness training
- Analyze through each goal and view
- Identify and report issues
- **Maintaining Secure Architecture**
    - Conduct scheduled reviews
    - Continually improve and stay up on best practices
    - Retire processes that are outdated
    - Reassess how processes work together as they change

Identity

- **Identity =** User info, rights, credentials, group memberships, and roles
- Name, address, title, contact info, id number, etc
- **AAA**
    - Authentication
        - Prove you are who you say you are
    - Authorization
        - What are you allowed to access?
    - Accounting
        - A record of what you access and do
        - Logssss
- **Account Lifecycle**
    - Create -> Provision -> Modify/Maintain -> Disable -> Retire/Delete
    - Must Utilize Least Privilege
        - Users with too much access are both threats, and vulnerable

- Privilege creep
    - validate accounts have the correct rights
    - If someone keeps moving job positions, promotions, etc, they may end up with permissions to a dozen places, which means they can do shady stuff
- **Identity Lifecycle Management**
    - Centrify, Okta, Ping Identity
        - Help you create, manage, monitor, and report on accounts

Identity Systems

- **IAM - Centralized Identity Access Management**
    - Create, store, and manage identity info
    - Includes group membership, roles, permissions
    - Used for:
        - Provisioning accounts
        - Authentication
        - Single-sign-on
        - LDAP
        - Account Maintenance
        - Reporting
        - Monitoring
        - Logging
        - Auditing
- **Directory Services**
    - **LDAP - Lightweight Directory Access Protocol**
        - Hierarchical structure
            - dc = domain name
            - Ou = organizational unit
            - Cn = common unit
        - Securing LDAP
            - Enable and require TLS for queries
            - Set password storage to salted hash
            - Disable unauthenticated or anonymous modes
            - Replicate to a redundant server to prevent Denial of Service
            - Strong ACLs to limit access to non-privileged users
        - **LDAP Injection**
            - Similar to SQL inject
            - Secure web apps and validate queries and input
    - Provides info about systems and users
    - Useful for email and other programs like address books

- **Authentication Protocols**
    - TACACS+
        - TCP to provide AAA services
        - Lacks integrity checking
        - Encryption flaws
        - Bad
    - RADIUS - Remote Authentication Dial-In User Service
        - Common AAA service
        - Password security isn't great by default
        - Requires IPSec encryption on traffic
    - Kerberos
        - Designed with security in mind
        - Encrypts all data sent
        - Principles (users)
            - Primary - Username
            - Instance - Unique ID
            - Realm - Groups
        - Replaced NTLM for windows domains
        - Review Kerberos ticket system
- **Single-Sign-On SSO**
    - Users authenticate once and gain access to multiple services
    - LDAP
    - **CAS** - Central Authentication Service
    - Reduces password reuse, and less password resets and support calls
    - **Shared Authentication**
        - OpenID
            - Open source standard for decentralized authentication
            - Sign in through google, access everything that relies on them
        - OAuth
            - User shares elements of their info but doesn't need an account
        - OpenID Connect
            - Uses OAuth info but adds authentication
        - Facebook Connect
            - Basically OpenID but for facebook instead of google

Identity System Threats

- Logon Exploits
- Credential Handling
- Authorization Process
- Target Account Lifecycle
    - Create credentials

    - Escalate privileges
    - Prevent credential removal
- Phishing
- **Personnel-based Threats**
    - Usually phishing or other social engineering
    - Train your users not to share their passwords
    - Insider threats
- **Endpoint Threats**
    - Local exploits on laptop
    - Keyloggers
    - Password stores and tokens
    - Anti-malware and anti-virus and strong authentication will defend against these threats
- **Other Threats**
    - Server-based threats
        - Attacks server to interfere with AAA
    - Application/Service Threats
    - Roles, Rights, and Permission Threats
        - Giving users/accounts additional roles, rights, and permissions

Attacking AAA Protocols and Systems

- Directories, AAA and SSO systems are high-profile targets
- **Attacking LDAP**
    - Target unencrypted LDAP traffic
        - Attempting replay attacks
    - Target improper ACLs to harvest info or modify directory
    - Perform LDAP injection against web apps
    - Denial-of-Service
- **Attacking RADIUS**
    - Replay attacks
    - Compromised shares secret key off client machines
    - Brute-force secret keys from stolen passwords
    - Denial of Service
- **Attacking Kerberos**
    - Secure, but popular so high profile target
    - Compromise of Key Distribution Center **KDC** allows attacker to impersonate anyone
    - Stealing Kerberos Tickets allows attacker to impersonate specific user
    - **Ticket Granting Tickets** are especially vulnerable, because these allow an attacker to do pretty much anything on your system
- **Attacking Active Directory**
    - Many existing exploits against clients, servers, and AD domain

    - Many AD domains are outdated and unpatched
    - Malware focused on stealing credentials
    - Attacking older services like NTLM, LANMAN, NetBIOS, unsigned LDAP, and SMB
    - Privilege creep
        - validate accounts have the correct rights
    - Overuse of admin credentials - only use them when you need to perform specific functions
    - Privilege escalation
        - Setup user accounts and admin accounts and name them properly
        - If a user account has admin rights, you have a problem
- Attacking OAuth, OpenID, and OpenID Connect
    - Each service provider implements them uniquely and improper configs can make data vulnerable
    - Original account info - google ID for example - won't be compromised, but you may be redirected improperly, or attackers may be able to get in unvalidated
    - Early versions of these protocols may be more vulnerable
- **Identity Exploits**
    - Impersonation Attacks
    - Usually credential theft, or OAuth abuse
    - **Session Hijacking**
        - Attacker acquires, or guesses session key
        - Prevented through TLS encryption sessions
    - **Man-in-the-Middle**
        - When attacker taps the data flow and listens in, or takes over
    - **Privilege Escalation**
    - **Rootkits**
        - Uses malware to give attacker access to a server/client continually
    - **SMS Vulnerabilities**
        - Multi-authentication systems that rely on SMS are vulnerable to VoIP attacks
- **Credential Theft**
    - **Phishing**
    - **Compromise other websites**
        - Abusing reused passwords and credentials
        - Dual-authentication prevents this risk
    - **Brute-force-attack**
        - Could take millions of years, but it gets easier all the time
        - Captchas and limited login attempts prevent this

Securing Authentication and Authorization

- Strong passwords
- Password management

  - SSO
  - Token-based multifactor
  - Password safes (LastPass, Dashlane)
- Encrypt Communication between clients and authenticators
- ACLs to match users with proper right and privileges
- Policies to control right distribution
- Management oversight for approval
- **Securing Auth (Admin)**
  - Privileged Users must be managed and monitored
  - Additional monitoring and logging
  - Separation of Duties
  - Training
  - Prevent admin accounts from being used as daily accounts
- **Multifactor Auth**
  - Knowledge, possession, biometric, location
- **Context Auth**
  - Time of day
  - IP Address
  - Frequency of Access
  - Location
  - Type of Device

Identity as a Service (IDaaS)

- Cloud Based AAA
- Make sure you trust your provider
- Will you configure internal/external database?
- Where do you authenticate?
- Where do you store your credentials?
- **Benefits**
  - Can be more secure, capable, and better managed

Detecting Identity Attacks

- **SIEM - Security Information and Event Management**
  - Can alert if new privileged accounts are made
  - If privileges change
  - If terminated accounts are restored
  - If unused accounts are lingering
  - Violations of Separation of Duties
  - Can monitor patterns to identify abnormalities
  - **Best part about SIEM: It’s easier to read**

- Humans have to analyze these trends and data to see if abnormalities are actually bad

Federated Identity Systems

- Moves the trust boundary to a third party like Google or Facebook
- **IDP - Identity Provider**
  - Third party that houses the identity
- **RP Relying Party or SP Service Provider**
  - You, usually, requesting the identity
  - Members of the federation the provide services to the user when identified
- **Consumer or User**
- **Fourish steps**
  - Discovery
  - Validate
  - Register RP Attributes
  - Federation Protocol
- Federated Identity System Technologies
  - **SAML - Security Assertion Markup Language**
    - XML-based
    - Enables SSO for web apps and services
  - **OAuth 2.0**
    - Developed by IETF - Internet Engineering Task Force
    - Designed for HTTP based services
    - Open source
  - **Flickr**
  - **ADFS - Active Directory Federation Services**
    - Microsofts
    - Similar to OAuth
- **Risks**
  - If Facebook gets hacked, how do you respond?
  - If someone's facebook account gets hacked, how vulnerable does that make your system?
  - If you stop using a federated system, will you lose your users?

Software Development Life Cycle

- I really don't understand why this is in this course at all
  - Sure, software needs security too? But that should be a course to itself
- SDLC - Software Development Life Cycle is applicable to other things
- Planning for security earlier makes it easier!

- Phases
    - **Planning**
        - Initial investigations into the effort
        - Feasibility analysis
        - Alternate solutions?
        - Move forward, or buy an off-the-shelf solution
    - **Requirements**
        - Gain stakeholder/customer feedback to determine required functionality
        - What should the program do?
        - What does current program NOT do?
        - This is where CySAs become important - it must be safe and secure
    - **Design**
        - Functionality, architecture, data flows, processes, etc
        - Basically whiteboarding/flowcharting
    - **Coding**
        - if(code == code)
    - **Testing**
        - Coders already did some testing, now shareholders join in
        - Try to break it
        - See if users are happy with it
        - Ensure security!
    - **Training and Transition**
        - Make sure users can use it
    - **Operations and Maintenance**
        - Longest phase of SDLC
        - Patches, updates, mods, and support
        - Most expensive to update Security at this point
    - **End of Life**
        - How long does software get support?
        - At EoL, support ends, security suffers
        - Users must migrate to new software to maintain security
        - Remember: Nobody wanted to leave Windows XP for Vista, cause it sucked

Software Development Models

- **Waterfall Model**
    - Linear model, pretty loose
    - **Requirements - > Design -> Implementation -> Verification -> Maintenance**
- **Spiral Model**
    - Iterative adaptation of Waterfall
    - Revisits phases over and over again throughout prototype stages

  - Faster to Minimum Viable Product
- **Agile**
  - Iterative and incremental
  - Function over Documentation
  - Customer collab over contract negotiation
  - Responding fast is better than planning
  - *Seems like a total fucking mess*
  - **Terms**
    - Backlogs - features to complete
    - Planning Poker - Estimation tool for planning
    - Timeboxes - Agreed upon time to work on specific goal
    - User Stories - High Level User Reqs
    - Velocity Tracking - Add up estimates for current sprint efforts and compare to completion rates
- **RAD - Rapid Application Development**
  - Informal, iterative process, focused on modules and prototypes
  - Highly responsive, no planning phase
  - **Terms**
    - Business Modeling - Understand business process
    - Data Modeling - analyze datasets and their relationships
    - Process Modeling - Define the processes and data flows
    - Application Generation - Convert processes into code
    - Testing and Turnover - Do the inputs and outputs work?
- **Big Bang SDLC Method**
  - One guy in his basement banging it out
  - Wtf, really?
- **V Model**
  - Adaption of Waterfall
  - Waterfall down, test back up
  - Costly and time consuming, but quality

Coding For Security

- Security isn't an afterthought - it should be in the requirements
- Security is built during design and coding
- Security is tested in prototypes AND finals
- **Secure Coding Practices**
  - Have an organizational secure code policy
  - Conduct risk assessments to prioritize issues
  - User Input Validation
    - Don't trust anything you get from the user

  - Consider error messages - how much do you reveal?
  - Database security in application and database
  - Secure data-in-motion
  - Encrypt Stored information
  - Hash passwords
  - Design for availability and scalability
    - Prevent DDoS
  - Conduct Monitoring and Logging
  - If possible, utilize multifactor authentication
  - Code for secure sessions
    - Prevent session hijacking
  - Secure your cookies
  - Encrypt Network Traffic
    - HTTPS or TLS tunnel
  - Secure underlying infrastructure
    - Good code on a bad system can still be vulnerable
- **OWASP - Open Web Application Security Project**
  - Community hosted standards, guides, best practices and tools
  - Proactive controls for testing web app security
  - Top 10 Vulnerability List
  - **ZAP - Zed Attack Proxy**
    - Security Scanner
- **Secure Code Management**
  - GitHub
  - Check-in, check-out
  - Revision history

Testing Application Code

- Scanning With a Tool
- Automated Vuln Scanning Tools
- Manual Pentest
- **Code Review**
  - Share knowledge with others
  - More experience is learned across team
  - Detect problems and enforce good coding
  - Agile and formal models
  - **Pair Programming**
    - 2devs1workstation
    - Its like getting a running commentary on your work
    - Costly

    - **Over-The-Shoulder**
        - One dev codes, another dev shows up to see if it makes sense
    - **Pass-Around**
        - Dev codes, then pass it… around
        - Documentation is essential
    - **Tool-Assisted**
        - Formal or informal
        - Marks up code and provides feedback
        - **Codacy**
    - **Fagan**
        - Formal code review by a team of reviewers
        - Specifies input/output for each process
        - More costly and difficult, but effective
- OWASP consider code review the best option
    - 360 review - code review AND pent test, then review code again

Finding Security Flaws

- Static Analysis
    - Code review or scanning
    - Requires access to source code
- Dynamic Analysis
    - Code is executed with specific input and analyzed
    - Automated tools help
    - Test Types:
        - **Fuzzing**
            - Sends invalid data to test ability to handle unexpected data
            - Use large datasets from automated tool
            - Tests for logic issues, memory leaks, error handling, input validation
        - **Fault Injection**
            - Tests error handling functions
            - **Compile-Time Injection**
            - **Protocol Software Injection**
                - Sending FTP instead of HTTP data, etc
            - **Runtime Injection**
                - Insert data into running memory of the program
        - **Mutation Testing**
            - Make small changes to program to determine if it causes a failure
            - What can bad guys do to the code? (with malware, typically)
        - **Stress Testing**
            - Can app support production load?

                - How does it respond to worst-case scenarios, traffic spikes, DDoS?
            - **Security Regression Testing**
                - Ensures that changes made do not create new problems
                - Patch testing, basically
                - Scan, patch, scan

Web Application Vulnerability Scanners

- CySA are usually gonna be dealing with web app code
- Dedicated web app scanners do better than Nessus, Nexpose, and OpenVAS
- Identify app problems, web server, database, and infrastructure problems
- Examples:
    - Acunetix WVS
    - Archni
    - ***Burp Suite**
    - IBM App Scan
    - Netsparker
    - QualysGuard Web
    - W3AF
- Better at finding issues with forms, SQL injections, etc
- Manual Scanning
    - Use an interception proxy to capture communication between browser and server
    - Modify data sent and receives
    - Examples:
        - Tamper Data
        - HttpFox
        - Fiddler
        - ***Burp Suite**
    - Allows you to manually try cross-site scripting attacks, injection attacks, etc
- Definitely learn more about Burp Suite - common, powerful, etc

Performanced Based Question Review

- Get familiar with Vulnerability Scan Results
    - Google: Nessus Report Examples
- Analyze Event Logs
- Analyze Server Logs
    - Look for unusual things
    - Lots of invalid logon attempts from the same IP
    - Anything mentioning money, paypal, payment

- .exe files on webservers, etc
- Exam will be largely obvious - limiting the logs specifically around the nefarious activity
    - Look for repeating IPs, keywords, common ports, server names for clues
    - Look for escalating privileges, admin/system/root accounts
    - Look for users being added to groups, esp. Admin groups

Very Cool CyberSecurity Stuff

- **Unified Kill Chain** - an adaptation of the Cyber Kill Chain with a few other defense frameworks to help visualize complete threat-vectors and defense strategies.
    - This Document is long and dense, but chock full of useful information
    - If you read nothing else, look at the case studies starting on page 36

Ports!

- Well-known/System Ports
  - 0-1023
- User Ports/Registered Ports
  - 1024-49151
- Dynamic/Private/Ephemeral Ports
  - 49152-65535
- Port Numbers and their Applications
  - 20 - FTP (Send file data)
  - 21 - FTP (Session info)
  - 22 - SSH, SFTP, SCP
  - 23 - Telnet
  - 25 - SMTP
  - 49 - TACACS+
  - 53 UDP/TCP - DNS
  - 67 UDP - DHCP and BOOTP
  - 69 - TFTP
  - 80 - HTTP
  - 88 - Kerberos
  - 110 - POP3
  - 119 - NNTP (Network News Transfer Protocol)
  - 123 - NTP (Network Time Protocol)
  - 137,138,139 - NetBIOS
  - 143 - IMAP
  - 161 - SNMP (Agents receive requests)
  - 162 - SNMP (Controller receives data)
  - 389 - LDAP Lightweight Directory Access - 389
  - 443 - HTTPS (over TLS/SSL)
  - 443 - SSTP (Over TLS/SSL) Secure Socket Tunneling Protocol
  - 445 - SMB/SAMBA Server Messaging Block - 445(Also, 137,138,139)
  - 465 - SMTP/s Secure Mail Transfer Protocol
  - 500 - IKE Internet Key Exchange

636 - LDAPS w/ TLS	
~~989/~~990 - FTPS	
1433 - SQL	
1701 - L2TP, L2F	Layer 2 Tunneling Protocol - 1701
1720 - H.323	
1723 - PPTP	Point to Point Transfer Protocol - 1723
1812,1813 - RADIUS	RADIUS - 1813,1812
2427 - MGCP	Media Gateway Control Protocol - 2427
2727 - MGCP	
3389 - RDP	Remote Desktop Protocol - 3389
5004 - RTP	Real-time Transport Protocol - 5004
5005 - RTP (Default)	
5060 - SIP (unencrypted)	Session Initiation Protocol - 5060
5061 - SIP (encrypted with TLS)	
Protocol IDs	
PID 50 - ESP IPsec	IPsec
PID 51 - AH IPsec	Authentication Headers