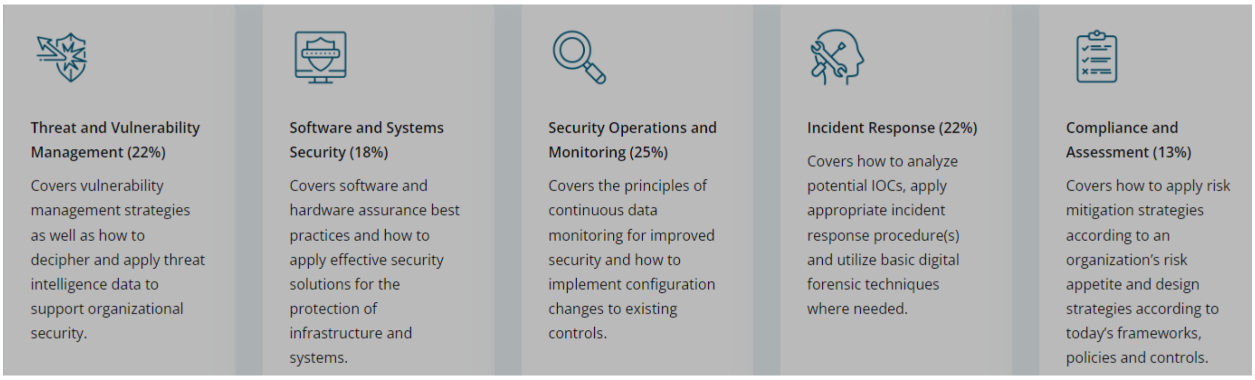

Threat and Vulnerability Management (22%)

Covers vulnerability management strategies as well as how to decipher and apply threat intelligence data to support organizational security.

Software and Systems Security (18%)

Covers software and hardware assurance best practices and how to apply effective security solutions for the protection of infrastructure and systems.

Security Operations and Monitoring (25%)

Covers the principles of continuous data monitoring for improved security and how to implement configuration changes to existing controls.

Incident Response (22%)

Covers how to analyze potential IOCs, apply appropriate incident response procedure(s) and utilize basic digital forensic techniques where needed.

Compliance and Assessment (13%)

Covers how to apply risk mitigation strategies according to an organization's risk appetite and design strategies according to today's frameworks, policies and controls.

Cysa+ Note

Chapter 1:

When print document, take page 272 paper and stick it to page 35 of this note.

-**Confidentiality:** unauthorized disclosure of sensitive information.

-**Integrity**: Altercation of information or unauthorized modification.

-**Availability**: systems are ready for clients to access.

Note: Security and privacy are closed related concepts: There are 3 components of security which's (CIA)

1. Security controls focus on how Org can protect its own information.

2. Privacy controls have a different focus. (It focuses on how Org can use and share info about individuals). The data known as PII.

+ General Accepted Privacy Principals GAPP) outlines 10 privacy practices:

- **Management:** Org document its privacy practices in a privacy policy and related Doc.
- **Notice:** Notify individual about its privacy, type of info being collected, how info is used
- **Choice and Consent:** Org should obtain, direct consent of Individual for storage, use, sharing PII
- **Collection:** Org collect PII only, purpose identified in noticed and consent by individual.
- **Use, Retention, and disposal:** Org should only use info for identify purpose, not use for any other purposes.
- **Access:** Org provide individual, access to any info about that individual record, per request.
- **Disclosure:** Org will disclose info to 3rd party only when consistent with notice and consent.
- **Security:** PII be protected against accurate and complete information
- **Monitoring and enforcement:** put business processes, ensure compliance with its privacy

Formular: Risk = Threat x vulnerability

I. Identify Threat:

Note: NIST (SP) 800-30: risk assessment process

Risk Assessment process by identify type of **Threat**, and then identify **vulnerabilities.** Now you know much risk you have.

- NIST identify 4 different types of threats:
  - **Adversarial Threat:** individuals or group, org attempt to undermine security or Org. **Adversaries** includes **trusted insider**, **competitor, supplier, customers, business partner, nation-states.**
    =>Evaluate adversarial threat: capability of threat actor engage in attack, its **intent, likelihood** threat will target Org.
  - **Accidental Threat:** when individual doing their routine work, form an action that undermine security.

- **Structural Threat:** equipment, software, environmental controls fails due to exhaustion of resource. (Simply fail due to an age).
- **Environmental threat:** natural, disaster occur that are outside the control of the Org.

Note: Risk Assessor: Person who conducts risk assessment.

+ **Risk Assessment** may uses "Quantitative techniques" that numerically assess likelihood and impact of risk.

Vital: Overall risk = likelihood and impact of risk (combine 2 evaluations)

- ➔ Reviewing Controls:
- Risk Acceptance, Risk Avoidance, Risk Mitigation, Risk Transference (will be used during **Risk Assessment**)

Most common way Org manage **security risk**: develop technical and operational control security controls that **mitigate** those risks.

II. Building Secure Network

- ❖ **Network Access Control (NAC):** 802.11x protocol, standard used for NAC. Devices wishes connect to a wireless access point, need network challenge to authenticate using 802.11x.
  - Supplicant (device): communicate with service AKA "**Authenticator",** run WAP or network switch. "**Authenticator"** don't have necessary info to validate **Supplicant**, so pass access request along to "**Authenticator Server"** using "**RADIUS".**
- **NAC** has 2 different major ways:
  - Agent-based **Vs** agentless: **Agent-based** solution (802.11x) requires device request access to the network run special software to communicate with MAC service. **Agentless** approaches to NAC conduct authentication in the web browser, "Do not require special software".
  - In-band **Vs** Out-of-band: **In-band (inline)** NAC solution use dedicated appliance, sit in between devices, resources they wish to access. **Ex. "Captive Portal"**. **Out-of-band** NAC (802.11x) have network devices communicate with Auth server, then reconfigure the network to grant or deny network access (**has to communicate with Auth server**).
- ❖ **Some criteria used by NAC:**
  - Time of day: access network only during specific time
  - Role: be assign to a particular network based on their roles.
  - Location: physical location.
  - System health: NAC has agent running on device, obtain config. Fails to meet requirement, got denied access to network.

III. Firewall and network perimeter security

- ➔ NAC: design to manage system connect directly to an Org wired or wireless network.
- ➔ Firewall: typically configured "Tripled Home" meaning connect to 3 different network.
- ➔ DMZ: network zone designed, receives connections from outside worlds such as web email serves.

Whenever **Firewall** receives request, it evaluates firewall's rules base (ACL). ACL identifies the types of traffic permitted to pass through the firewall.

Note: FTP (20,21), Telnet (23), POP3 (110), IMAP (143), NTP (123), LDAP (289), LDAPs (636), SQL Server (1443), Oracle (1521), PPTP (1723),

- ❖ 4 types of **Firewall:**
  - Packet filtering (Stateless Firewall): check characteristic of each packet against firewall's rules (ACL). (Often found in routers, other network devices)
  - Stateful inspection (Stateful firewall): maintain information about state of each connection passing through firewall.

- Next-generation firewall (NGFWs): contextual info about users, application, business processes.
- Web application firewall (WAFs): designed to protect against web application attack, SWL injection, cross-site scripting.

IV. Network Segmentation

Firewall uses principals known as "Network Segmentation" to separate the network.
+ Connection to **the jump box,** protects with strong multifactor Auth tech**.** It's used to serve a layer of insulation.

- ❖ **DNS Sinkhole:** feel false info to malicious software. Ex. When compromised system attempts to obtain info from DNS Server about it C2C, **DNS Server** detect suspicious request, respond with IP address of a **Sinkhole System** (**Detect and remediate the botnet-infected system**)
- ❖ **Group Policy (GPO):** allows admin define group of security group settings, apply those settings to a group of systems based on roles.
- ❖ **Endpoint Security Software:** Admin choose to install host-based firewall, serves as basic firewall for that individual system, complimenting network-based firewall controls or host intrusion prevention system (HIPSs) that bock suspicious network activities.
- ❖ **Mandatory Access Control (MAC):** based on sensitivity of info. Ex. Classified, secret…etc.
- ❖ **Discretionary Access Control (DAC):** owner of the files control permission.

Note: Security-Enhanced Linux (SELinux): operate by US national security, enforce **MAC.**

V. Penetration Testing

There are 3 rules of engagement to finalize during the "Planning phase".

1. **Timing: When**, test take place, staff be informed, have as little impact on business operation
2. **Scope: What** is the agreed-on scope of pen test? Off-limit to the testers?
3. **Authorization: Who** authorize the pen test to take place?

First Phase: "Conducting Discovery"

- **Reconnaissance:** gather much info as possible, performing port scans, vulnerable scan, web application testers

  Note: Vulnerable scanning: important component of pen test.
- **Execute a Penetration Test**

NIST attack process:
1st: Initial access to a system
2nd: Seek to escalate
3rd: Install additional penetration testing tools

NIST SP 800-115: Technical Guide to Information Security Testing and Assessment
1st: Discovery Phase (Reconnaissance)
2nd: Gaining Access (make informed attempts to access target)
3rd: Escalating privilege (seek to gain complete control of system-admin access)

- Vertical: attacker gains access to an account with the intent to perform actions as that user
- Horizonal: gain access to account with limited permissions requiring an escalation of privileges

4th: System Browsing (info gathering begins to identify mechanism to gain access to addition system)
5th: Install additional Tools (additional pen test tools are installed to gain additional control)

VI. Training and Exercise

Wargame that pit teams of security professionals against one another in cyber defense.

- ❖ 3 teams are involved in cybersecurity wargame:
  1. **Red Team:** role of attack, reconnaissance and exploitation

2. **Blue Team:** securing targeted environment, building, maintaining, monitoring a comprehensive set of security controls
3. **White Team:** coordinate exercise and servers as referees, monitoring results.

- **Reverse Engineer:** decomposition where reverse engineers start with the finished product and work their way back to its component parts.
- **Compiler:** to convert the source code into binary (0 or 1)
- **Decompiler:** to convert binary back to source codes.
- **Secure Hash Algorithm (SHA):** compute the hash of 2 files, compare the output values

Note: Reverse Engineer may also seek documentation from **Original Equipment Manufacturer (OEM)**

Conclusion:

- **CIA Traid,** includes **identify vulnerabilities, recognize corresponding threats, determine level of risk** that result from vulnerabilities and threat combination.
- **Network Security control:** include NAC, Firewalls, network segmentation.
- **Secure Endpoints control:** include hardened system config, patch management, GPO, endpoint security software.

➔3 objectives of cybersecurity: Confidential (no unauthorize gain sensitive info), Integrity (no unauthorize modification to info), Availability (ready to meet needs)

→**Vulnerability** is a weakness, **Threat** potential event can cause harm or damage, **Risk** likelihood that a threat can exploit a vulnerability.

➔**Adversarial Threat** (Individual & groups attempt to undermine security or Org). **Accidental threats** (Individuals doing their routine, mistakenly perform action undermine security). **Structural threat** (equipment, software, environmental fail due to exhaustion of resources, exceeding operation capabilities – fail due to age). **Environmental threat** (natural disaster)

➔**NAC** helps 2 security objectives: ensure system accessing Org network meet basic requirement, **network firewall** sits at boundaries, provides network perimeter security.

➔Harden config: include disabling any unnecessary services, ensuring that secure config setting exist on devices, centrally controlling device security settings.

➔NIST SP 800-115 (process for penetration testing) divide tests into 4 phases: planning, discovery, attack, reporting.

Chapter 2: Using Threat Intelligence

Threat Intelligence categorize into 3 levels:

1. **Strategic intelligence:** provides broad info about threat and threat actors allowing Org to understand & respond to threat.
2. **Tactical Intelligence:** detail technical & behavior info, directly useful to security professionals and who act as defense.
3. **Operation threat intelligence:** response to a specific threat , include info about where it came from, who create it, how it has changed overtime, how it is delivered about where it spreads, what it attempt to do, how to remove and prevent it.

I. Threat data intelligence

+ **Threat feed**: has detail IP, hostname and domain, email address, URLs, file hashes, file paths, CVE names and others about threats.

Open Source intelligence: acquired from publicly available sources.

+Virustotal: detail about malware

+Spamhaus: focus on blocked list, spam via Spamhaus Block List (SBL)

Threat Connect uses 6 levels confidence:

- **Confirmed (90-100):** independent source or direct analysis, prove threat is real.
- **Probable (70-89):** relies on logical inference, not directly confirm the threat.
- **Possible (50-69)**: use when info is agreed with analysis, but assessment is not confirmed.
- **Doubtful (30-49):** assigned when assessment is possible, not mostly like option, assessment can’t be proven.

  - **Improbable (2-29):** assessment is possible, not most logical option, refuted by other info.
  - **Discredited (1):** used when assessment has been confirmed, inaccurate or incorrect.

+Structured Threat Info Expression (STIX): **XML** language, sponsored by US homeland security. (STIX2.0) current version defines 12 STIX domain objects, include attack pattern, identities, malware, threat actors, and tool.

+Trusted Automated Exchange of Indicator Info (TAXII): allow cyberthreat info to be communicate at the application layer via HTTPs. (Design to support STIX data exchange)

+Open Indicators of Compromise (OpenIOC) format: XML-based framework, developed by Mandiant. Typical IOC includes metadata like authors, name of IOC, description, detail the actual of compromise.

Planning Threat Intelligence:

1. **Requirement Gathering do following**:
- Assess what security breach or compromise you have faced.
- Assess what info, have prevented or limited impact of the breach.
- Assess what control & security measures were not in place that would have mitigate breach.

2. **Data Collection**: once have info requirement, collect data from threat intelligence.
3. **Data Processing and Analysis**: Data, you gather in the Data Collection, will likely be in several formats. So first process data, allow it to be consumed by whatever tools before use.
4. **Intelligence Dissemination**: data is distributed to leadership and operational personnel, use data as part of their security operations role.
5. **Feedback**: gather feedback about the report and data you've gathered. Continuous improvement should create a better requirement to improve overall output of threat intelligence program.

Note: **Information Sharing and Analysis Center (ISACs):** help infrastructure owners and operations share threat information, provides tools and assistant to members.

Threat Actors:

- **Nation-state:** have resource of a country behind them, often associated with APT.
- **Organized crime:** financial gain.
  - **Commodity malware,** written and sold on black market. Sold as stand-alone tool.
- **Hacktivist:** hacking, political or philosophical.
- **Insider threat:** employee or other trusted individuals inside Org, intentional or unintentional.

Threat Classification: known threat, unknown threat (prepare for only through general controls).

- **Microsoft's STRIDE** classification model:
  - **Spoofing user identity**
  - **Tampering**
  - **Repudiation**
  - **Information disclosure**
  - **Denial of services**
  - **Elevation of privilege**
- PASTA (Process for Attack Simulation and Threat Analysis)

Note: Classification Tools provides 2 benefits:

1. Allowing you to use a command framework to describe threats, allowing others to contribute and manage threat information.

2. Models server as a reminder of the type of threats that exist, help analyst perform better threat analysis, by giving them a list of potential threat options.

II. Threat Research and Modeling

Threat modeling takes many factors common elements includes:

- Assessing adversary capabilities, resource, intent, ability of likely threat
- Total attack surface of Org, you are assessing. Any systems, a threat may target
- Listing possible attack vectors, which attackers can gain access to their target.
- The impact if successful
- Likelihood of the tack or threat succeeding

Measure used to assess threat are:

+**Behavioral assessment:** useful for insider threat

+**Indicator of compromise (IOC):** are forensic evidence or data help identify an attack.

- **Attack Framework** is useful help think though what an attacker likely to do, build appropriate defenses against attack.

MITRE's ATT&CK Framework: initial access through execution, persistence, privilege escalation, exhalation.

- ATT&CK: metric include preattack, enterprise matrices focusing on Windows, macOS, Linux, cloud computing, IOS and Andriod, also includes details of mitigations, threat actor groups, software.

III. Diamond Model of Intrusion Analysis

+ **Diamond Model of Intrusion Analysis** describes a sequence where an adversary deploys a capability targeted, infrastructure against a victim.

This model uses a number of specific terms:

- **Core Feature of an event**: adversary capability, infrastructure, and victim.
- **The Meta-Features,** start and end timestamps, phase, result, direction, methodology, and resources. (**Activity Threat**: order events in sequence)
- **Confidence Value**, undefined by model which analyst expected to determine based on their own work.,

➔**Diamond Model** focuses heavily on understanding the attacker, their motivation, use relationships between these elements, allow security analysts understand threat.

Cyber Kill Chain:

1. Reconnaissance: target selection, research, vulnerability identification
2. Weaponization: creation of tools to exploit vulnerabilities. (Phishing email, user clicked)
3. Delivery: Weapon is delivered to target. (Download zip files)
4. Exploitation: Malware program is triggered and exploits vulnerability. (Zip files opens, triggers PS command, get process running)
5. Installation: Remote access tools/backdoor installed (drop files in App/Local/temp)
6. Command and Control (C2): intruder has president assess (communicating to c2c)
7. Action on Objection: intruder takes actions to accomplish their goals: data damage, system change

Describe Cyber kill chain 7 stages in detail:

+**Reconnaissance**

+**Weaponization:** combine malware and exploit into payloads, deliver to target. The model emphasize that defender needs to conduct full malware analysis in this stage to understand, not only what payload is dropped but also how weaponized exploit was made. **Defender**, build detection for weaponizers, timeline of when malware was created.

+**Delivery**: occurs when adversary deploys, tool direct against targets.Ex, email payload, USB stick, via websites they visit. **Defender,** must observer how attack was delivered.

, what was targeted, what attacker intend to accomplish. **Retention Log,** important in this stage, defenders need them to track what occurred.

+**Exploitation**: use software or hardware, human vulnerability to gain access. This can involve 0 day, may either use adversary-triggered exploits or victim-triggered exploit. **Defender,** against this focus on user awareness, secure coding, vulnerability scanning, pen testing, endpoint hardening, ensure that Org has strong security posture, limiting attack surface.

+**Installation**: persistent backdoor for attackers. **Defenders** must monitor for typical artifacts of a persistent remote shell, remote access method.

+**Command and control (C2): Defenders,** seek to detect to C2 infrastructure, hardening network, ensure new C2 models and technology.

+**Action on Objectives:** final stage, adversaries collect credentials, escalate privileges, pivot, move laterally. **Defenders,** establish their incident response playbook, detect the actions of attacker and capture data, respond to alerts, assess the damage the attackers have caused.

The Unified Kill Chain: combines cyber kill chain and MITRE's ATT&CK. It uses 18 phases both inside and outside a defended network, addressing complaints.

Note: Cyber Kill Chain has been criticized for including actions that outside of the defended network (Reconnaissance and weaponization)

➔**Common Vulnerability Scoring System (CVSS):** help describe vulnerabilities using a numerical score.

IV. Proactive Threat Hunting

Search for threats proactively rather than reactively.

+**Proactive threat**, triggered by new data, inspire threat analysts to establish a hypothesis about new theat.

+Threat hunter, once you have a hypothesis, investigate the threat. Attack vector (malware, virus)

➔***<u>Proactive threat hunting activities into few bullets:</u>***

- Establish hypothesis. Hypothesis is needed to test and should have actionable results.
- Profiling threat actors and activities, ensure may be a threat, what typical actions and process are.
- Threat hunting tactics: skill, techniques, and procedures, this step includes executable process analysis.
- Reducing the attack surface area: making protection more manageable
- Bundling critical assets into groups and protection zones helps with managing attack surface area.
- Attack vector must understand assessed, addressed based on analysis of threat.
- Integrated intelligence (SIEM)
- Improving detection capabilities: continuous process

Note: Analyzing threat, Diamond Model, and Cyber Kill chain.

Conclusion:

+**Assessing intelligence:** source based on timelines, relevant the data, accurate, which dataset to use and rely on.

+**Using Standardize (STIX & TAXII):** information exchange and management. **OpenIOC** provide Org- a framework to agree on rating for threats, communication detail of compromise.

+**Threats classified,** known Vs Unknown, 0-day exploits and advanced persistent threats.

+**Threat intelligence cycle:** collection data, analysis, communication and dissemination (spread info widely) and gathering feedback.

+**Using threat model:** help fully understand identifying gap.

+**Threat Intelligence major role:** is to be in risk assessment, influence probability and impact assessment, provide useful info about risk due to specific threat. Proactive, critical part of threat intelligence and management activities.

NOTE: In order to identify insider threat, use behavioral assessment.

Chapter 3

+Reconnaissance & intelligence gathering = gather information

I. Mapping and Enumeration

First step when gather Org intelligence is identify technical footprint.

➔**Host enumeration** used to create a map of an Org's network, system. Typically accomplish by combining info gathering tools with manual research to identify the network and system.

- **Active Reconnaissance:** use host scanning to gather info about system, services and vulnerabilities. Ex. Nmap, Zenmap

+**Scanning a network** or **system** cause a problem of devices that are scanned.

Before conducting active reconnaissance, ask for approval first (**called: Get out of jail free cards**)

- **Mapping Network and Discovering Topology**

=>**Active scans** provide info about network design & topology, helps testers guess about topology based on TLL of packet it receives, **Traceroute info,** response from network and security devices.

=>**Router or gateway** centrally connected, easily see where group of hosts connect.

The system that **Nmap** runs from becomes center initial scan and show its local loopback address (127.0.0.0).

A number of hosts appear on second network segment behind the 10.0.2.1 router.

Note: Firewall may stop **Nmap** from scanning traffic.

- Security and network devices can cause differences in TTL and traceroute info, resulting in incorrect or missing data.

Pinging Hosts:

➔**Ping command,** a low-level network command, send packet called "**echo request**" to a remote IP address. If remote system, receives the **request,** it responds with "**echo reply**" indicating that it is up running, meaning communicating path is valid. **Ping use ICMP.**

Note: The lack of respond does not necessarily mean that remote system is not active.

Many Firewall block ping requests and individual systems may be configured to ignore **echo request packet**.

- Hping -p 80: specify TCP port 80
- Hping -S: indicate tCP SYN flag should be set, indicating a request to open a connection.

II. Port Scanning and Service Discovery technique Tool

Port Scan often 1st step in **Active Reconnaissance**. Port Scan feature include:

- Host Discovery
- Port Scanning and service identification
- Service version ID
- Operating System

Well-known port: 0-1023

Registered port: 1024-49151

- **Nmap -PO:** skip pinging the system before scanning
- **Nmap -sS:** perform TCP SYN scan, send connection attempts to each port.
- **Nmap -**O= scan Operating system
- **Nmap -sV:** grabs banners and perform other service version identification step to capture more info, checks against a database services.

Important: Time it takes to run and how many hops there are to the host (port scanning). Scan completes less than 2 seconds, indicates host respond quickly, host was one hop away, directly accessible from the scanning host.

- **OS Fingerprinting:** ability to identify OS based on the network traffic that it sends **AKA (operating system fingerprinting).**
  - Typically done using TCP/IP stack fingerprinting technique, focus on comparing response to TCP and UDP packets sent to remote host.

    - Differences in how OS, what TCP options they support, what order they send packets in, a host of other details.
- **Service and version identification:** ability to identify service, info about potential vulnerabilities, as well verify the service is responding to given port matches, service typical use that port.
    - Service ID is usually done by **connecting** and **grabbing the banner** or **connection info** by comparing its response to the signature of known services.
    - Service name and version, help identify service, are running on nonstandard port.

Common Microsoft port includes, 135, 129, and 445, running Microsoft Remote Procedure called (MSRPC), NetBIOS, and Microsoft domain services (useful indicator that a remote system is a Windows Hosts).

➔TCP scan: popular scan method, verify a service response, quick and unobstructive.

➔**Nmap** has GUI, **Zenmap** provide additional visualization capabilities including topology view mode.

➔**Angry IP scanner** a multiplatform (port scan), with GUI.

- Angry IP scan do not provide detailed ID or services and OS. Angry IP scan requires Java.

Passive Footprinting: rely on info available through social media, crawling website, without performing your own probe. Reply on stored data, meaning "out-of-date".

Note: Active scan interact with hosts, **passive scan** observes activities and draw conclusion.

III. **Log and configuration Analysis**

Network device information includes network device log, network device config files and network flow.

→Most managed network also send network logs to a central log server using **syslog** utility.

SNMP: send device info to a central control system.

- ❖ Network device log, not as useful as device config data when focus on intelligence gathering, although they provide some assistance with topology discovery.
    - During pen test, security operation, network device log, provides useful warning of attack or reveal config or system issue.

Network device Configuration

Configuration files invaluable when mapping network topology, it includes detail of network, routes, systems devices interact with, syslog and SNMP servers, administrative and user account info.

➔SNMP community strings, the contract for devices, as well what traps are enabled and where they are sent.

Note: Trap Operation allow SNMP agent to send a synchronous notification of the occurrence of event.

- ❖ **Netflow (Cisco network protocol):** collect IP traffic info, allowing network traffic monitoring.
    - Include IP, port source, destination for the traffic, class of service.

Netstat: can provide info as below:

- Active TCP and UDP connection, filtered by each if major protocols: tcp, udp, ICMP, IP, IPv6. Linux: **Netstat -ta** (showing active TCP connections), **Netstat -u** (shows active UDP connections), **-w** (show RAW)**, -X** (show unix socket), **-o (**identify process numbers).
- **Windows**: **Netstat -e (**show interface statistic), **Netstat -nr (**shpw destination network, netmask, gateway, interface the route is associate, metric for route that capture link speed).

<u>DHCP Logs and DHCP Server Configuration Files</u>

DHCP: client/server protocol, provides IP address, default gateway, subnet mask for network segment that host will reside on.

- When conducting passive reconnaissance, DHCP logs from DHCP server, provide identify many of the hosts on network.
- Combine DHCP logs, firewall logs, can determine which hosts provides with dynamic IP and which host uses static IP.
- System with fixed DHCP address often **server or system** need to have a known IP address for a specific function.
- DHCP logs for **Linux** found in **/var/log/dhcpd.log** use **journalctl** command to view log.
- DHCP log provide MAC address and IP address

Firewall log and configuration files

- Analyzing router and firewall access control list and logs, provide useful info what traffic is allowed and help topological mapping by identify where system are based on traffic allowed through or blocked.
    - Firewall logs allow pen test to reverse-engineer firewall rules based on the content.

Linux system log: are in **/var/log. Windows** provides several types of event logs:

- **Application log:** programs or application
- **Security log:** resources, right usage, file being opened, created or deleted.
- **Setup logs:** application are setup
- **System logs:** logged by Windows components. Present as part of windows.
- **Forwarded events logs:** setup using event subscription**,** contain events collected from remote computers.
- Traceroute starts by passing author's home router, follow a path through Comcast's network with stops int eh south Bend area, then chigago.

Global IP manages by IANA. IANA manages the **DNA Root Zone,** handles both gTLD (region TLD), and ccTLD (Country TLD).

Note: AD4 mostly likely target for active directory-based, exploit window server-specific scan, whereas hostname that reflect a specific app or service can provide both target info and a clue for social engineering.

DNS Discovery: External DNS info, provide part of Whois. Additional DNS server may be identified as part of active scanning or passive scanning based on network traffic or logs by reviewing Org doc.

- Scan be done using port scan and search for system that provide DNS services- on TCP or UDP port 53.
- Once found DNS server, query it is using **dig** or other DNS lookup command (Test to see if support **Zone transfer)**

Zone Transfer: intended to be used to replicate DNS database between DNS Servers, make it a powerful info-gathering tool if a target's DNS servers allow a zone transfer.

- Most DNS server set to prohibit zone transfer to server that aren't their trusted DNS peer.

There are 2 ways of checking zone transfer:

- Host -t axfr domain.name dns-server
- Dig axfr @dns-server domain.name

Running this against DNS server that allow zone transfer will result in a large file (dump file digi.ninja), a site that allows practice zone transfer for security practitioners.

Note: Zone transfer are turn off for most DNS servers.

- **DNS Brute forcing**: simply sending a manual or scripted DNS query for each IP. Solution: Can be prevent by IDS or IPS, with a rule preventing DNS brute force. **However,** sending queries at a slow rate can bypass most prevention methods.
- **Whois**: allow to search database of registered users of domains and IP address blocks.
- **Responder:** a python script, hybrid between active and passive info gathering.

    - Begins by passively monitoring the network, waiting for other systems to send out broadcast request intended for devices running networked services.
    - Once responder passively identified one of these requests, switched into active mode and responds, attempting to hijack the connection and gather info from the broadcasting system and its users.
- **Harvester**: gathering emails, domain info, hostnames, employee names, and open ports, banner using search engine. (**Maltego: build relationship between people and ties to others)**
- **Shodan:** search engine for internet-connect devices and their vulnerabilities.

Note: A capture from a host, can tell what systems are on a given network by capturing broadcast packets, OS fingerprinting can give you remote host's OS.

- **Exif data:** photo's GPS metadata into an online mapping application.

➔**Detecting reconnaissance is to capture data.** Monitoring at the connection points between network zone, where data sensitivity or privilege zones meet.

Typical Data source:

- IDS, IPS, HIDs, NIDs, firewall and other security devices has these capabilities:
    1. Packet **analysis at individual packet**, **detect issue** with **content of packets, signatures of attack contain in them.**
    2. **Protocol analysis examines traffic.**
    3. Traffic and flow analysis monitor behavior based on historic traffic pattern.
- Traceroute: determine path and latency to remote host.
- Creepy: geolocation tool, gather data from social media and geotagging.

IV. Data Analysis methods

- ***Anomaly analysis***: look for differences, establish patterns or expected behaviors. **"Normal"** to identify differences to build a base model.
- ***Trend analysis:*** predicting behavior based on existing data, identify future outcome such as network congestion based on usage patterns and observed growth. (help guarantee availability of service)
- ***Signature analysis:*** user fingerprint or signature to detect threats or other events.
- ***Heuristic or behavioral analysis:*** detect threats based on their behavior, detect unknown threats. (good for detecting Zero day exploitation).
- ***Manual analysis:*** frequently performed. Human expertise and instinct, detect sth not be seen.

Note: **Passive Reconnaissance** and **Social Engineering** less likely to be dealt with a network security-centered defensive design.

V. Preventing Active Reconnaissance

Active reconnaissance prevention relied on a few common defenses as below:

- Limiting external exposure of services, know external footprinting.
- Using IPS or others, limit or stop probes to preventing scanning.
- Using monitoring and alerting systems (SIEM)

VI. Preventing Passive Information Gathering

- **DNS harvesting:** used by domain registrars. Help prevent misuse, include the following:
    - Blacklisting systems or network that abuse the services.
    - Using CAPTCHA to prevent bots.
    - Providing privacy services that use 3rd party registration info instead of actual person or Org registering domain.
    - Not publishing zone files, if possible, but gTLP required to publish their zone files, ccTLDs only publish some.
- **Netstat:** show TCP connection executable, associate with them and route table info
- **Security logs:** show files, open, create, delete, sign in

- **Metadata scrubbing:** remove hidden info about files such as creator, creation time, system used to create files.
- **Prevent DNS harvesting**: Captcha, rate limit, blacklisting network or system

Conclusion:

- **Active Reconnaissance** involves probing systems. **Port scanning,** first step during reconnaissance, **Nmap** used for system, port, OS and service discovery for part scanning.
- **Passive Reconnaissance,** provide info without active probe. Relies on data gathering.
    - Log files, config files, publish data from DNS and Whois queries can provide valuable data without sending traffic to a system or network.
    - **Packet Capture:** useful, understand network and help document active reconnaissance activities.
- **Detecting Reconnaissance** relies on capture evidence of intelligence gathering activities. Done using IDS, IPSs, and log analysis by correlational info using SIEM system.
    - **Automated data analysis** look for anomalies, tends, signatures, and behaviors.
- **Preventing and responding to Reconnaissance:** require limiting Org footprint, use IPS, firewall. Proactive measure such as pen testing and self-testing.

Take away: a proper zone transfer will only allow secure zone transfer to specific permitted peer DNS servers.

Chapter 4

HIPPA: handle protected health information.

GLBA (Gramm-Leach-Biley Act): govern how financial institutions handle customer financial records.

Neither of these 2 compliances above specifically requires Org conduct Vulnerable scanning.

PCI DSS prescribe details of vulnerabilities scans includes the following:

- Org run both internal and external vulnerability scans
- Org must run scan on at least a quarterly basic, "after any significant change" in network (new system install, or changes)
- Internal scans must be conducted by qualified personnel
- Org must remediate any high-risk vulnerabilities and repeat scans until they receive a "Clean" report
- External scan must be conducted by an "**Approved Scanning Vendor (ASV)"** authorized by PCI SSC

➔**Federal Information security management Act (FISMA**): requires that government agencies and other Org comply with a series of security standards.

- **Confidential:** prevent unauthorized access to user info.
    - Low: unauthorized disclosure of info expected to have **Limited** impact on Org
    - Moderate: unauthorized disclosure of info expected to have **Serious** impact
    - High: unauthorized disclosure of info expected to have **Severe or catastrophic**
- **Integrity:** unauthorized modification to info
    - Low: unauthorized modification expect **Limited** impact on Org
    - Moderate: unauthorized modification expect **Serious** impact on Org
    - High: unauthorized modification expect **Severe or catastrophic** impact on Org
- **Availability:** Ensuring reliable access
    - Low: disruption of access expect **Limited** impact on Org
    - Moderate: disruption of access expect **Serious** impact on Org
    - High: disruption of access expect **Severe or catastrophic** impact on Org
- **FISMA** requires vulnerable mgmt program for all federal info system regard their impact
- **FISMA** require government agency conduct vulnerable scan
- **Control enhance #4** require that Org determine what info about system is discovered by adversaries (only apply to **FISMA High Impact**)

All federal info must meet basic requirements for vulnerability scan found on NIST Special Publication 800-53 (Security and Privacy Controls for Federal Information Systems and Organizations)

→These require each Org subject to **FISMA** do the following:

- Scan for vulnerabilities in the info system and hosted app, new vulnerability affecting system/app are identified, report them.
- Analyze vulnerability scan reports and result from security control assessments
- Remediate legitimate vulnerabilities
- Share info obtained from vulnerabilities scan process and security control assessment to help eliminate similar vulnerabilities in other info system

→**NIST 800-53** describes 8 controls enhancements, requires depending on circumstances:

1. Org employs vulnerability scanning tools includes capability to readily update info system vulnerabilities to be scanned.
2. Org updates info vulnerabilities scanned prior to a new scan when new vulnerabilities are identified and reported.
3. Org employes vulnerabilities scanning procedures, can identify the breadth and depth of coverage.
4. Org determines what info about info system is discoverable by adversaries and subsequently Org define-corrective actions.
5. Info implements privilege access authorization to info system component for selected vulnerabilities scanning activities.
6. Org employs automated mechanisms to compare the results of vulnerable scan over time to determine trend in info system vulnerabilities.
7. Requirement 7 was withdrawn.
8. Org reviews historic audit logs to determine if a vulnerability has been previously exploited.
9. Requirement 9 was withdrawn.
10. Org correlates the output from vulnerabilities scanning tools to determine the presence of multi-vulnerability/multi-hop attack vector.

PCI DSS and FISMA involve in retail transaction and operating government system.

Note: Scan schedule determines by Org risk appetite regulatory requirement, technical constrains, business constrains, licensing limitation. (Willing to tolerate risk = **Risk Appetite)**

Many different factors influence how often Org decides to conduct vulnerability scans against its system

- Org risk appetite is its willingness to tolerate risk with Org. If Org is extremely risk averse, choose to conduct scan more often
- PCI DSS and FISMA, dictate minimum frequency for vulnerability scans.
- Technical Constraints may limit the frequency of scanning. (disrupt work, slow.ect)
- Business constraints may limit Org from conducting resource-intensive scan during period of high business activities.
- Licensing limitation may curtail the bandwidth consumed by the scanner.

Active and Passive Scanning

→**Active vulnerability scanning** interacts with scanned host to identify open service and check for possible vulnerabilities, it provides high-quality results, but comes with some drawbacks:

- Active scanning is noisy, likely to be detect by admin
- Active scanning has potential to accidentally exploit vulnerabilities and interfere the functioning of production systems.
- Active scanning may mis some systems if blocked by firewalls, IPS, VPN.

→**Passive vulnerability scanning** probing systems for vulnerabilities, monitor networks, similar to technique used by IDS, but instead of watching for intrusion attempts, they look for telltale signatures of outdates systems and pp reporting result to admin.

- **Passive scan** capable of detecting vulnerabilities. Reflected in network traffic. They are not a replacement for active scanning, a very strong complement to periodic active scan.

Configuring and Executing Vulnerability Scan

I. **Scoping Vulnerability Scan**

- What systems and networks will ne included in the vulnerability scan.
- What technical measure will be to test whether systems are present on the network.
- What test will be performed against systems discovered by a vulnerability scan.

Vital: Scoping reduce problems to a manageable size. Segmentation reduces the scope of PCI DSS much smaller isolated network that is dedicated to payment card processing.

➔Info maybe gathered in 2 ways to supplement remote scan with trusted info about server config:

1st, admin provides scanner with credentials, allows scanner to connect to the target server and retrieve config info.

- This info determines whether vulnerability exists, improving the scan's accuracy over non credentialed alternative.
- **If a vulnerability scan detects a potential issue that can be corrected by OS update, credentialed scan can check whether the update is installed on the system before reporting a vulnerability.**

(**Credentialed Scan** only retrieves info from target server, don't make changes to server itself. Thus, admin enforce principal of least privilege by providing the scanner with a read-only account on the server.)

II. **Scan Perspective**

Controls that might affect scan results include: (Because it might block the scan)

- Firewall settings
- Network segmentation
- Intrusion Detection system (IDS)
- Intrusion prevention system (IPS)

Vulnerability Plug-in Feeds

Security researchers discover new vulnerabilities every week, and vulnerability scanners can be effective against these vulnerabilities only if they receive frequent updates to their plug-in.

- Security Content Automation Protocol (SCAP): an effort by security community led by NIST to create a standardized approach for communicating security-related info.
- Common Configuration Enumeration (CCE): provides standard nomenclature for discussing system config issues.
- Common Platform Enumeration (CPE): provides standard describing product names and version
- Common Vulnerabilities and Exposure (CVE): provide standard describing security-related software flaw.
- Common Vulnerability Scoring System (CVSS): provide standard approach for measuring and describing the severity of security-related software flaw.
- Extensible Config Checklist Description Format (XCCDF): a language specifying checklists and reporting checklist results.
- Open Vulnerability and Assessment Language (OVAL): a language specifying low-level testing procedure used by checklist.

Prioritizing Remediation:

Important factor remediation prioritizing decision-making process include:

- **Criticality** of system and info affected by vulnerability: taking into account of CIA.
  - If vulnerability allow DOS attack, analyst should consider the impact of Org if system becomes unusable due to an attack.
  - If vulnerability allows theft to store info from database, analyst should consider the impact

- **Difficulty** of remediating the vulnerability: if fixing vulnerability will requires human and financial resources too much, should be factored into decision-making process.
    - Analyst should fix the issue rate from 2-6 in priority orders.
- **Severity** of vulnerability: The more severe the issue, the more important to correct that issue.
    - Analyst should turn to CVSS to provide relative severity rankings for different vulnerabilities. (**CSVV is a component of SCAP)**
- **Exposure** of the vulnerability: analyst also consider how exposed vulnerability is to potential exploitation (Risk).
    - If internal server has a serious SQL injection, but server is accessible only from internal networks (Less severe than the one facing the interne).

Testing and Implementing Fixes

Before deploying remediation, should rest planned fixes on Sandbox, reduce likelihood disrupt business operation.

- After deploying a fix, verify that mitigation was effective. Involve repeating vulnerabilities scan that initially identified issue does not appear in the new scan result

Delayed Remediation Options

Incase, you can't correct the problem immediately, 2 basic options

1. **Compensating control:** security measure, address vulnerability without remediating underling issue. Ex. Web application vulnerable to SQL injection, you can't correct web app, use web application firewall to block SQL injection attack.
2. **Risk Acceptance:** continue business as usual, acknowledge risk and move on

Overcoming Risks of vulnerability Scanning

1. **Service degradation:** consume network bandwidth system functionality, interrupting business processes.
    - Risk increase, involve legacy systems or proprietary systems, exhibit unexpected behavior.
2. **Customer Commitments:** if scanning negatively impact the Org, ability to meet customer commitments, customers may need participate in decision making process.
3. **IT Governance and Change Management Processes:** create bureaucracy to make the config change require to support scanning.

Vulnerability Scanning Tools

→Vulnerability scanner often leveraged for preventing scanning and testing, also found in pen testers toolkit, help identify systems that testers can exploit.

III. Infrastructure Vulnerability Scanning Tool:

1. **Tenable's Nessus:** well-known and widely respected network vulnerability
2. **Qualys's vulnerability canner:** using a software (SaaS), located both in on-premises datacenters and in the cloud.
3. **Rapid7's Nexpose**: offers capabilities similar to Nessus and Qualys.
4. **Open source OpenVAS**: offers a free alternative to a commercial one

Web Application Scanning

Web Application scanner are specialized tools use to examine the security of web applications.
→These tools use to test SQL injection, XXs, CSRF. It works by combining traditional network scans of web servers with detail probing, sending known malicious input sequences and fuzzing in attempts to break the application.

- **Nikto:** web application scanner, are required to knowledge for the Cysa+ exam
- **Arachni:** another web application scanner

Interception Proxy: valuable tools for pen testers, seeking to evaluate security of web application.

- Run on the test's system and intercept requests being sent from the web browsers to the web server before released onto the network. This allows tester manipulate the request to attempt the injection of an attack
- **Zed Attack Proxy (ZAP):** OWASP
- **Burp Proxy:** is a commercial web application security.

Wireless Assessment Tool:

- **Aircrack-ng:** wireless network testing. Capture packets from wireless networks, conducts packet injection attacks,, crack preshared key used on WEP,WPA, WPA2 network.
- **Reaver:** used to find WPA and WPA2 passphrase specifically on Wi-Fi Protected Setup (WPS)
- **Hashcat:** password cracking tool used on wireless network.

Conclusion: Prioritize remediation is focused on 4: Criticality, Difficulty, Severity, Exposure.

- Criteria for scanning target: select scan target, data classification, system exposure, services offered, status of the systems as a test, production environment.

Chapter 5: Analyzing Vulnerability Scans

I. Understanding CVSS

CVSS, an industry standard for assessing the severity of security vulnerabilities, provides a technique for scoring each vulnerability on variety of measures.

- Analyst scoring new vulnerabilities on 8 differences measures. Each measure is given descriptive rating and a numeric score.
- **First 4 measures, evaluates exploitability of the vulnerability, the last 3 evaluates the impact of the vulnerability**.
- 8 metric discusses the scope of the vulnerability.

Attack Vector Metric: describe how adversary exploit the vulnerabilities.

- **Physical (P)**: attacker physically touch vulnerable device. (Score: 0.20)
- **Local (L)**: attacker has physical or logical access to affected systems. (Score: 0.55)
- **Adjacent Network (A)**: attacker has access to the local network that affected system connect to. (Score: 0.62)
- **Network (N):** attacker can exploit vulnerability remotely over a network (Most severe) (Score: 0.85)

Attack Complexity Metric: describe the difficulty of exploiting the vulnerabilities.

- **High (H):** vulnerability requires "specialized" condition would be difficult to find. (Score:0.44)
- **Low (L):** vulnerability does not require any specialized conditioned. (Score: 0.77)

Privilege Require Metric: describe type of account access that attacker need to exploit.

- **High (H):** attacker **requires admin privileges** to conduct the attack. (Score: 0.270-0.50)
- **Low (L):** attack requires **basic user privileges** to conduct attack. (Score: 0.62-0.68)
- **None (N):** attack do not need to authenticate to exploit the vulnerability. (Score: 0.85)

User Interaction Metric: describe whether attacker needs to involve another human in the attack.

- **None (N):** successful exploitation **doesn't require action** by any users. (Score: 0.85)
- **Required (R):** successful **exploitation does require** action by any by a user. (Score: 0.62)

Confidentiality Metric: describe type of info disclosure might occur if attacker successfully exploit.

- **None (N):** there is **no confidentiality impact.** (Score: 0.00)
- **Low (L): access to some info, but attacker doesn't have control over** what info is compromised. (Score: 0.22)
- **High (H):** all info is compromised. (Score: 0.56)

Integrity Metric: describe type of info alteration might occur if an attacker exploit.

- **None (N):** no integrity impact. (Score: 0.00)
- **Low (L):** modification of some info possible, attacker doesn't have control over what info is modified. (Score: 0.22)
- **High (H):** integrity of system is totally compromised. (Score: 0.56)

Availability Metric: describe type of disruption night occur if attacker successfully exploit.

- **None (N):** no availability impact. (Score: 0.00)
- **Low (L):** performance of system is degraded. (Score: 0.22)
- **High (H):** completely shut down. (Score: 0.56)

Scope Metric: describe whether vulnerabilities can affect system component beyond scope of vulnerability.

Note: scope metric doesn't contain score information.

- **Unchanged (U):** exploited vulnerability only affect resources managed by the same security authority.
- **Changed (C):** exploited vulnerability affects resources beyond scope of security authority managing component containing the vulnerability**.**
- Current version of CVSS is 3.1

II. **Interpreting CVSS Vector:** CVSS vector use a single-line format convey rating of a vulnerability on 8 of the metrics.

CVSS: 3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

1 / 2 /3 / 4/ 5 /6 / 7/8 / 9

Vector contains 9 components.

1st component: CVSS:3.0 (refer to the version)
2nd component: AV:N (attack vector: Network, score 0.85)
3rd component: AC:L (Attack complexity: Low, score 0.77)
4th component: PR:N (Privilege Required: None, score 0.85)
5th component: UI:N (User Interaction: None, score 0.85)
6th component: S:U (Scope: Unchanged)
7th component: C:H (Confidentiality: High, score 0.56)
8th component: I:N (Integrity: None, score 0.00)
9th component: A:N (Availability: None, score 0.00)

➔**Summarizing CVSS Scores**: CVSS vector provides info on risk posed by vulnerability, analyst can calculate the CVSS base score, representing overall risk. It **requires "exploitability, impact score, impact function"**.

Use the number above to calculate this.

ISS = 1-[1-Confidentiality)x)1-Integrity)x(1-Availability)]

So ISS = 1-[1-0.56)x(1-0.00)x(1-0.00)]

ISS = 0.56

→Calculate impact score: take value of scope (6.42, unchanged) multiple with ISS.

Impact = 6.42 * ISS = 6.42 * 0.56 = 3.60

➔**Calculating Exploitability Score:**

Exploitability = 8.22 x attackvectorxattackcomplexityxPrivilegerequiredxUserInteraction

Exploitability = 8.22 x 0.85 x 0.77 x 0.85 x 0.85 = 3.89

Calculate base score: determine cvss based scores following rule:

- If impact is 0, base score is 0
- If scope metric is Unchanged, calculate base score by adding **(Impact + Exploitability)**
- **Scope metric is Changed, take (impact + Exploitability) x 1.08**
- Highest possible base score is 10. If calculate value is greater than 10, set to 10.

Vital: Cysa+ objective, able to validate scan result. Understand true positive, false positive, true negative, false negative)

→Verifying SQL injection vulnerability may require attempting attack web application and verifying the result in the backend database.

III. Reconciling scan results with other data sources

Valuable info sources for this process includes:

- Logs from servers, network devices, might contain info attempts to exploit vulnerability.
- SIEM, correlating log entries from multiple sources
- Config management system provides info on OS and application installed on system.

Trend Analysis: Analyst should watch **new vulnerabilities arising over time**, age of existing vulnerabilities**, time required to remediate vulnerabilities**.

Note: Lack of support implies that no new security patches will be released.

- **Buffer Overflow:** attacker manipulates program into placing more data into area of memory than its allocated for that program's use.
  - Stack Overflow: target the stack which stores variable values, managed by OS.
  - Heap Overflow: target heap which stores object created by codes, must be managed by application developer.

- **Integer Overflow:** variant of buffer overflow, attempt to store an integer that is too large to fit in specific buffer.

4 numbers followed by CVE indicates year.

Privilege Escalation: seek to increase level of access that an attacker has to target the system. Allows normal users account into a more privileged account such as the root superuser account.

Arbitrary Code Execution: allows attacker run software of their choice on the target system.

- Attacker run the code with admin privileges.

Remote Code Execution: *subset of code execution* vulnerabilities because attackers exploit vulnerability over a *network connection without having physical or logical access to the target system.*

- **Telnet** (insecure protocol) port 23, used to gain command-line access to remote server.

  →Use **SSH** to replace Telnet.
- **FTP** port 21, transfer file between systems, does not incorporate security features.

  →Use **FTPS** to replace FTP.
- **SFTP:** SSH extension (remote access)

→Developer should conduct testing only on system dedicated to that purpose, on a production system, disable "Debug Mode".

Missing Firmware Update: OS and application aren't the only device required security updates, network device also do.

TLS version 1.2 is a replacement for SSL (heart bleed attack)

IV. Insecure Cipher Use

SSL and TLS commonly describe as cryptographic algorithms, in fact not the case.

- SSL and TLS describe how cryptographic ciphers used to secure network communications, not cryptographic themselves.
- **A client and a sever communicates using SSL/TLS they exchange, a list of cipher, each system supports and agree on a mutually acceptable cipher.**

Certificate Problem: SSL and TLS rely on digital certificates to validate the identity of the server. Common errors for SSL/TLS below:

1. Mismatch between name of the certificate and server: may indicate use of certificate taken from another site. Its's the digital equivalent using fake ID.
2. Expiration of digital certificate: when see expired cert, mean that server admin fails to renew the certificate.

3. Unknown CA: anyone can create digital cert, but digital cert useful only if recipient of a cert trust the entity that issued it. OS and browser contain instructions to trust well-known Cas but will show error if they encounter a cert issued by unknown or untrusted CA.

DNS provides a translation service between domain names and IP addresses.
Scenario: attacker sends spoof DNS requests to a DNS server, designed to elicit response that are much larger in size than original requests.

- These larger responses then goes to spoofed address where DNS server believes that query originated.
- IP address used in the spoof request actually target of DOS and bombarded by large response from DNS servers from over the world.

Internal IP Disclosure: Sever is accessible over the internet must have public IP address, but that address managed by firewall that use NAT to map that public IP address to the server's true (NAT table), private IP address.

- Sever not properly config, leak private IP addresses to remote systems. This can occur when **system includes its own IP address in the header info** return in the response to HTTP request.
- Server is not aware that NAT is in use, so use private address in its response. Attacker use this info to learn more about internal config of a firewall network.

Virtualization Vulnerability: virtual machine then run on top of the virtual infrastructure provides by hypervisor, running standard OS such as windows and linux.

- These virtual machines may not be aware that they are running in a virtualization environment.

Vital: *hypervisor runs directly on top of physical hardware*, is known as bare-metal virtualization.
Hosted virtualization, where a host OS sits between the hardware and the hypervisor.

- ❖ **VM Escape:** the attacker has access to a single virtual host and then manage to leverage access to intrude on the resources assigned to a different VM.
- **Solution:** Hypervisor prevent this by restricting VM's access to only those resources assigned to that machine.

Internet of things (IOT):
Supervisory Control Access and Data Acquisition (SCADA) system, Industrial control system (ICSs) and other example of IOT. These systems allow connection of physical devices and processes to network.

- ICS relies on sensors and controllers distributed throughout Org collecting info and controlling activities.
- **PLC** (Programmable Logic Controller) specialized hardware controller designed to operate in IOT environments.
  - **PLC** often uses Modbus to communicate with sensors and other IOT component over wired serial interface.

Some other technologies component that made up IOT:

- **Embedded system:** computers integrated into operation or another device such as vehicle, camera, multifunction printer.
- **Real-time OS (RTOS):** slimmed-down OS, designed to work quickly on IOT in a low power environment
- **System on Chip (SoC):** an entire embedded system package onto a single chip, includes a processor, memory, network interface
- **Field-programmable gate arrays** (FPGA): are computer chip allows end users to reprogram their function, for embedded system
- **Controller Area Network Bus (CAN Bus):** network, specialized network designed to facilitate communicate network embedded systems without overhead of TCP/IP network.

Injection Attacks: attack send commands through web servers to a backend system, bypassing normal security controls and fooling the backend system into believing the request came from the web server.

- ❖ Web Application often receives input from users and use it to compose a database query that provides results that are sent back to a user.

2 best ways to protect SQL injections, are **Input Validation** and **least privilege restrictions** on database access.

- Input Validation ensures users don't provides unexpected text to the web server.

Cross-Site Scripting (XSS): **attack embed scripting commands** on a website that will ***be later executed*** **by an unsuspecting visitor** accessing the site. The idea is to **trick a user visiting a trusted site** into **executing malicious code by an untrusted 3rd party**.

- **Persistent XSS:** attacker actually store attack code on a server, code remains on the server, waiting for a user to request the affected content. (AKA "Stored XSS attack")
- **Reflected XSS:** occur when attacker tricks users into sending attack to the server as part of a query strings or other contents. Server then sends the attack back to the user (reflecting it), casing code to execute.
- **Document Object Model (DOM):** based on XSS attack occur within a database maintain by user's web browser. These attacks particularly insidious because they occur entirely on the user's computer, never seen by remote servers.

Note: Correcting issue with web app vulnerabilities is required rewriting code and rescanning vulnerable application.

Directory Traversal: **attacker insert filesystem path value into a query string**, seeking to navigate to a file located in an area, not normally authorized for public access. (Occur when file name includes in the query strings).

Ex. //www.myserver.com/policy?document='aup.pdf'

- Web application might see this query as string, then go to policy server, retrieve a doc called aup.pdf. If attack know policy server is located on the name server as payroll records, might try following query string.

Ex. //www.myserver.com/policy?document='../payroll/mike.pdf'

- This query seeks to traverse the directory structure of the storage server, navigate up to parent directory of the policy folder, then down into the payroll directory.

Solution:

- Filename in user-manipulatable fields, such as query strings
- Input validation prevent use of special characters.
- Access controls on storage server should restrict the web server's access to files authorized for public access.

Authentication Vulnerabilities: methods to targeting identity access management by password spraying, credential stuffing, impersonation, MITM, session hijacking attack.

Password Reuse: 2 common authentication vulnerabilities:

- **Password spray:** attacker use list of common passwords and attempts to log into many different user accounts.
- **Credential Stuffing:** attacker takes a list of usernames and password were stolen in the compromise of one website and use them to attack to gain access to a different, potential unrelated, website.
  - ▪ Credential stuff attack are successful when user use the same passwords.

Impersonation: occur when attacker takes on identity of a legitimate user. Preventing impersonation requires stronger session handling.

MITM: occur when attacker able to interfere communication flow between 2 systems.

Session Hijacking: focus on taking over already existing session, either by acquiring session key or cookies used by remote server to validate the session by causing the session to pass through a system the attacker controls, allowing them to participate in the system.

Conclusion:

+Common server and endpoints vulnerabilities: missing patches, unsupported OS, app, buffer overflow, privilege escalation, arbitrary code **execution, insecure protocol usage, present of debugging modes, DNS misconfig, internal IP disclosure, VPN issues.**

+ Virtualization vulnerabilities: virtual machine escape, management interface access, missing patches on virtual host, security misconfig on virtual guests and virtual network.

Chapter 6: Cloud Security

The "On demand" nature of cloud computing means, customers requests are fulfilled immediately.

- Benefits of cloud computing:
- Cloud computing environ shared pool of resources, used an approach called "Multitenancy" where many different customers share access to same physical resources.
  - Cloud provider responsible for implementing "Isolation" control that prevent actions of one customer from interfering with or accessing data from another customer.
- Cloud Environ is configurable. Customer tailor to their specific business objective.
- Cloud resource not only rapidly provisioned, also rapidly released. When customer lo longer requires cloud resource, can release resource, stop paying immediately.
- Cloud requires minimal management from customer, core benefit transfer many responsibilities from own IT Team to cloud provider.

The Case of Cloud Computing

- **Agility and Elasticity:** when IT resources required, can be provision at a moment's notice. When no longer needed, can be released, stopped the billing.
  - **Agility**: the ability to scale quickly.
  - **Elasticity**: **automatically scaling up or down resources to meet user demands**
- **Scalability:** solutions built on cloud architecture limitless scalability, computing, storage, networking resources may be added and removed as needed.
  - **Scalability:** system's ability to scale (allocate/deallocate) resources**.**

Cloud Service Model

- **SaaS:** offering customer complete application and maintained by service provider, run in an infrastructure operated or procured by service provider.
  - Offering typical access service through web browser & perform limited application config.
  - Almost all responsibilities for operating service rests in hand of CSP.
  - SaaS example is Google's Gmail, Office 365
- **IaaS**: offer building blocks that customers use to design and implement their own service offerings. (more responsibility than SaaS)
  - Include, computing process, storage, networking.
  - **Customer retains more control** of infrastructure than **SaaS**, Thus, has greater responsibilities include monitoring, management, security.
  - **Customer** responsible for maintaining, upgrading OS, and software running.
  - **IaaS CSP** responsible for maintain physical environment, enforcing isolation, OS underlying cloud infrastructure.
- **PaaS:** middle group between SaaS and IaaS.
  - **CSP** operates infrastructure fully managed and configurable to run customer application
  - **Customer** deploys application either developed themselves or purchase from vendors onto the service provides' platform.
  - **Paas** supports variety of application and programming language.

  - **PaaS** allows customer to execute code without managing the underlying server, but customers need to be involved in provisioning appropriate level of infrastructure, by specifying number of servers they wish to use.
  - **PaaS** offers billed based on provisioning capacity.
- Function as a Service (**FaaS)** for this developer: allows serverless application architecture. Developers write function in common programming then configure that **FaaS** to execute (trigger), functions in response to events. Functions deployed, units of code, easily scaled to millions of executions per day.
  - FaaS offer Amazon's Lambda services, Python, Node.js, C#, Ruby, Go, and Powershell. Developers uses Lambda Runtime API to execute code, includes Microsoft Azure Functions and Google Cloud Functions.
  - Billing computing power and memory consumed during execution.

Public Cloud: make it accessible to customers, take advantage of a multitenant model.

- Cloud Public may offer IaaS, PaaS, SaaS, and FaaS services to their customers.
- Key distinction, those services don't run on infrastructure dedicate to a single customer, but rather on infrastructure that is available to general public.

Private Cloud: provisioned for used by a single customer. Maybe be built and managed by the Org, will be using infrastructure or be built and managed by 3rd party. (**Only one customer use this environment**)

Community Cloud: shares both public and private models.

- Community membership defines based on shared mission, similar security & compliance requirements.

Hybrid Cloud: describes cloud deployments that blend public, private, community cloud services together.

- Hybrid Cloud require use of technology that unify different cloud into a single coherent platform.
- Firm operate their own private cloud for the majority of their workloads and their leverage public cloud capacity when demand exceeds the capacity of their private cloud infrastructure.

  (this approach known as "**Public Cloud Bursting**")

Operating in the cloud: Cloud customers must divide responsibilities between one or more services providers and customers' own cybersecurity teams.

IaaS: CSP (Datacenter->Hardware) Customer (Operating System->Application->Data)
PaaS: CSP (Datacenter->Hardware->Operating System) Application is Both Customer (Data)
SaaS: CSP (Datacenter->Hardware->Operating System->Application) Customer (Data)

By nature, cloud providers always responsible for the security of both hardware and the physical data center environment.

- **IaaS:** customer takes over security responsibility for everything, isn't infrastructure.
- **PaaS:** vendor responsibility for operating systems, hardware, datacenter, whereas customer retains responsibility for data being placed into environment.
- **SaaS:** provider takes on all security responsibilities, whereas customer retains some shared control over data.

DevOps Strategies

Separating development and operating world provides technologies with comfortable working environment. Taks clearly defined and are surrounded by community peer. However, brings significant disadvantages following:

- Isolating operations teams from development process inhibits their understanding of business requirement.
- Isolating developers from operational considerations leads to design that wasteful in term of processor, memory, networking consumption.

- Requiring clear hand-offs from development to operations reduce agility & flexibility by requiring a lengthy transition phase.
- Increasing the overhead associated with transition encourages, combining fixes & enhancement into one major release, increasing the time to requirement satisfaction.

-->**DevOps** bring together development & operation team in a unified process, work together in a agile approach to software development.

→ Software testing and release process highly automated and collaborative.

Infrastructure as Code (IaC): one of key enabling tech behind DevOps, crucial to advantage of cloud computing solutions.

- **IaC**: process of automating the provisioning, management and deprovisioning of infrastructure services through scripted than human intervention.
- **IaC**: key feature of all major **IaaS** including AWS, Azure, google cloud platforms.

→**AWS** offer service called "**CloudFormation"**, allows developers to specification in several format, including JavaScript Object Notation (JSON) and Yet Another Markup Language (YAML).

Major advantages to using **IaC** approach are:

- Increase reusability of code.
- Reducing time spent by operation teams creating infrus components.
- Increase speed of infrus creation.
- Reducing likelihood of config errors by leveraging common templates.

Application Programming Interfaces:

IaC requires developer interact directly with a cloud service through their code than individual to work within a web interface.

- Developers may wish to write codes that execute in their own environment and still interacts with cloud service. (That's where Application programming interface (API) come into play)
- **API** are standard interface used to interact with web-based services. **CSP** create APIs and expose them *to customers to allow customer code to provision, manage and deprovision service*. (Ex. Bitwarden being provisioning in AD Azure Enterprise Application)

Note: when CSP expose APIs, must ensure that users requesting action through an API are authorized to do so.

- **APIs** manage this through use of API keys, are similar to password.
  - When a user request API, they also send their API keys to authenticate the request.
  - CSP validates the API key and checks that user, system, application associated with that key is authorized to perform the request action.

Insecure API, one of key risk associated with operating in the cloud. CSP manages their API well to enforce security requirement, but security of a user's account depends on the security of their API key.

Cloud customers best practices with API keys:

- Limit exposure of API key
- Use different API key for different users, and services
- Restrict the right associate with each API key to specific right needed by user, application or service
- Never transmit API keys cover unencrypted channels
- Never store API keys in unencrypted form
- Ensure that API keys, removed from my code that is placed in public code repositories

CASB (Cloud Access Security Broker): tool can help organize against cloud-specific risks.

- In a private datacenter, firewall restrict direct access to storage, limiting the exposure of unprotected file to user, who already has access to datacenter.
- In the public cloud, am improperly managed storage bucket may be exposed to the entire internet with only a few clicks.

Cloud Infrastructure Security Tools:

Cloud Infrastructure assessment tools reach into cloud environments, retrieve security info & deliver a report showing relative security of environment.

- They might detect issue that not appear on other vulnerability scan. Ex cloud-focus tool might able reach into cloud provider's API and identify fact that a security key has not been rotated for year.
- Tool that might be able to retrieve a list of all security group applied to an instance & determine the instance's network exposure without conducting port scan.

ScoutSuite: multicloud auditing tool reaches into user's account with CSP and retrieve config info using those services' APIs.

- Capable of auditing accounts with AWS, Azure, Google cloud, Alibaba cloud, Oracle cloud.
- ScoutSuite probe service config and search for potential security issues.

Pacu: (not a scanning tool, rather a cloud-focus exploitation framework)—Pen testing tool.

- Work with only AWS account, designed to help attacker determine what they can do with access have on existing AWS account.

Prowler: security config testing tool, similar to ScoutSuite. Deeper testing, limited to scanning AWS environment.

Cloud Access Security Broker (CASB): software tool servers between cloud service user & CSP. This positioning allows them to monitor user activities & enforce policy requirement.

CASBs operate 2 different approaches:

- **Incline CASB:** physically or logically reside in connection path between user and service. They may do this through hardware appliance or endpoint agent that route requests through CASB.
  - Requires config of the network and/or endpoint devices
  - Provides advantages of seeing requests before, they are sent to cloud service, allowing CASB to block request that violate policy.
- **API-based CASB:** do not interact directly with user, rather interact direct with CSP through provider's API.
  - Provide direct access to cloud service, doesn't require user device config. However, don't allow CASB to block request that violate policy.
  - API-based CASB: limited to monitoring user activities and reporting – or correcting policy violation after the fact.

Note: Incline CASB: will block request that violate policy whereas API-based CASB limited to blocking.

Conclusion:

++Public cloud: open to all customer.

++Private cloud: restrict use to single customer.

++Community cloud: offer service to members of defined, closed community.

++Hybrid environment: combines aspects of public, private, and/or community into a unified platform.

➔Cloud environment: embrace automation to improve agility.

- DevOps: unified software development & technology operation.
- Cloud serices enables DevOps by offering **IaC** capabilities through **API.**

ScoutSuite: probe security config multi vendors, **Prowler** same function, but only for AWS. **Pacu** pentesting tool for AWS.

Chapter 7: Infrastructure Security and Controls

I. **Layered Security**: most crucial concept of defense-in-depth, is the idea of *layer of security*.

→Data security is the core where policies practices, and data classification would be implemented.

→Config differ based on specific needs found at that layer in security mode. Ex. Data security require fill disk encryption for laptop and mobile devices, whereas laptop might only leverage data loss prevention software. **Layer Security Network Design** as below:

1. Data Security
2. Application Security
3. Endpoint Security
4. Network Security
5. Perimeter Security

+**Defense-in-depth** should address 3 CIA triad. **Depth in availability** take the form of *redundant systems or multiple network paths and providers or backup power systems.* **Integrity defense** may include *file integrity checking or system config validation tool*s.

Zero Trust: remove trust used to be placed in systems services and individuals inside security boundaries. In 0 trust environment, each action requested and allowed must be verified and validated before being allowed to occur.

- Zero trust move away from strong perimeter as primary security layer and instead moves even further toward a deeply layered security model where individual devices and application as well as user account, part of security design.
- Implementing **Zero Trust** require blend of technologies, *process, policy to manage, monitor, assess and maintain a complex environment*.

→***Providing layered of defense involves "segmentation and separation***".

- Physical segmentation: running *separate physical infrastructure or network*.
- System isolation is handled ensure infrastructure is separated, can go as far using **"Air Gap"** ensure *no connection at all between infrastructures*.
- Virtual segmentation: take advantages of virtualization capabilities to separate function to virtual machine to containers although some implement of segmentation for virtualization also run on separate physical servers in addition to running separate virtual machines.

Network Segmentation provides several advantages:

- Attack surface can be reduce by compartmentalizing systems and network.
- *Limit the scope of regulatory compliance* efforts by placing the systems, data or unit that must be compliant in more easily maintained environ separate from the rest of Org.
- Segmentation help increase availability by *limiting the impact of issue or attack*.
- Segmentation used to increase efficiency of a network. Large number of system in a single segment lead to network congestion, making segmentation attractive as networks increase in size.

Note: Physical and virtual solutions in form of isolation, focus on ***Air Gap, and a Jump Box*** as an access method.

For security reason, firewall designed ruleset typically used between network segments with different levels of trust or functional requirement.

- **Network segmentation** *relies on routes and switches that support VLAN tagging*. (**Segmentation** is *handled using only routers or switches*).

One common solution for access into segmented environ, the use of a ***jump box,*** sometimes called ***jump server,*** a system that resides in a segmented environ, used to access and manage devices in the segment resides.

- Jump boxes span 2 different security zones, carefully secured, managed, and monitored.

Physical Network Architectures: composed of the routers, switches, security devices, cabling. Common elements of a security design include the following:

- Firewall that controls traffic flow between networks or systems.
- **IPSs** detect and stop attacks, **IDSs** only alarm or notify when attack is detected.

- Content filtering and caching devices that are used to control what info pass through to protected devices.
- **NAC** technology controls which devices able to connect to the network, may assess security state of devices or require info allowing connection.
- **Network Scanner** identify systems and gather info including services that running, patch level and other details about systems.
- **Unified Threat Management (UTM):** combine a number of these services often including firewalls IDSs/IPSs. Content filtering and security features.

Software-Defined Network (SDN): make networks programmable. *Using SDN*, you can *control network centrally*, which allows management of network resources and traffic with more intelligence than a traditional physical network infrastructure.

- SDN provides info and controls via APIs like OpenFlow, mean that network monitoring and management can be done across disparate hardware and software vendors.
- **SDN-WAN** are SDN-driven service model where providers use SDN tech to provide network services.
    - **SDN-WAN** *provides encryption*, but introduce risks, including vulnerabilities of SDN orchestration platform, risks related to multivendor network paths and control, availability and integrity risks as traffic flows through multiple paths.

Virtualization: used to implement *virtual desktop infrastructure (VDI),* run desktop OS like windows 10 on central hardware and streams the desktop across the network to system.

Containerization: an *alternative to virtualizing an entire system*, and instead *permits application to be run in their own environ with their own required components such as library, config files, other dependencies, in a dedicated container*.

- Many containers run on the same servers threats to the host OS can impact many containerized services.
    - Tools exists to sign container images and to monitor and patch containers.
    - When addressing containerized systems, remember it shares underlying host as well as the rapid deployment model, typically used with containers.
    - Serverless computing sometimes called "FaaS".
        - Serverless computing relies on a system that executes function as they are call

→Networking in virtual environ combines traditional physical networks between virtualization or containerization host, then virtual networking that runs inside the virtual environ.

Logging, Monitoring, and Validation

- Logging system that well-design should ensure that logs are secure and available centrally for monitoring.
- Monitoring system need to have appropriate alert threshold set and notification capability.
- Monitoring and logging capability, config management and validation also important.

Certificate Management: The basic concepts:

- *Keeping private keys and passphrases secure*
- Ensure system use and respect *certificate revocations is necessary*.
- *Managing Cert life cycles* prevent expired certificates ensure, they're replaced when necessary

Active Defense: describe offensive actions taken to counter adversaries, include active response to incursion to limit damage by shutting down systems, removing account.

- *Threat intelligence, threat hunting, honeypot, and countermeasure design and implementation are part of active defense*.
- **Active defense** doesn't include "attack against adversaries".

Virtual Private Cloud: an option delivered by CSP that build an on-demand semi-isolate environ.

- VPC: exists on a private subnet.

*CASB tools are policy enforcement when cloud resources and services are used.
CASB help with data security, antimalware, service usage and access visibility, risk management.

*CASB requires careful config and continuous maintenance.

Improving Security by Improving Controls: controls can help prevent, detect, counteract, limit impact of security risks.

- Controls are classified based on 2 categories: *How they are implemented*, *when they react relative to security incident or threat*.

Classified controls Categories:** based on implementation type done using the following model:

- **Technical controls:** include firewall, IDS, IPS, network segmentation, authentication and authorization systems, technical capabilities designed to provide security
- **Administrative controls: AKA** (procedural controls): involve processes and procedures found in incident response plans, account creation and management, awareness and training efforts. (*documentation involve human*)
    - Some count "Legal Control" as a type of *administrative control*
- **Physical controls:** locks, fences, controls that limit physical access such as fire extinguisher that can help prevent physical harm.

Control Type:

- **Preventive Control:** intended to stop by taking proactive include firewall, training, and security guard.
- **Detective Control:** detect an incident and capture info about it like alarm or notification.
- **Corrective Control:** remediate an incident or limit how much damage can result from incident.
    - Often use as part of an incident response process, include patching, antimalware software, system restores from backup.
- **Deterrent control:** warn attacker that they shouldn't attack.
- **Directive control:** lead to a desired outcome.
- **Recovery control:** provide ways to respond to a breach.
- **Compensating control:** a mechanism, put in place for requirement of security measure.

Layered Host Security

Layer of security replies on a number of common security controls below:

- Host firewall and host based intrusion prevention software: limit network attack surface of host, help prevent undesired outbound communication.
- Encryption: at file level or full-disk encryption, protect from lsot or stolen.
- File Integrity monitoring (FIM): monitor files and directories for changes, alert admin if changes occur.

Host Security Layer include physical security control to help prevent theft or undesired access.

Permission: access rights or privilege given to user, system, application or devices. (Often granted based on roles).

- Linux permission, r for read, w for write, x for execute or a for no permission.

Type of security devices and controls:

- **Firewall:** block or allow traffic based on rules that use info like source or destination IP, port, protocols. Typically use different trust zones.
- **IPS:** monitor traffic, and apply rules based on behavior and traffic content. Like firewall, IPS, IDS rely on rules, but both *IPS and IDS inspect traffic* as a deeper level *paying attention to content and other detail inside the packets*.

- **DLP:** protecting data from leaving Org or system.
    - Complete DLP system target data in motion, data at rest and in use, and endpoints system where data accessed or stored.
    - DLP relies on identifying data that should be protected and then detecting when leaks occur, can be challenging when encryption is used between system and across networks.
    - DLP installation combines endpoints, software of making network traffic visible to the DLP system.
- Endpoint detection and response (EDR): provide continuous monitoring and response to advanced threats. Typically use endpoints data gathering and monitoring capabilities paired with central processing and analysis to provide high levels of visibility into what occurs on endpoints.
    - Ability to handle multiple threat types, include ransomware, malware, data exfiltration.
- **NAC:** require system to authenticate before connecting to a network. NAC operates in preadmission or postadmission mode, either checking systems before connect or checking user action after system are connected to the network.
    - NAC solution requires agent or software on the connecting system to gather data and handle the NAC connection process, maybe agentless.
    - When not successfully connect to NAC, an issue is detected. Maybe placed into quarantine network where they may have access to remediate or simply prevented from connecting.
- **Sinkhole:** redirects traffic from its original destination to a destination of your choice. This is done via **DNS** to prevent traffic from being sent to malicious site.

Malware Signature: composed of recognizable file pattern or hashes that could be checked to see if a given file or files matches those from known malware.

- **Signature:** can be created by analyzing malware and creating hashes, other comparator cab be checked if the package match.
- **Malware identification:** relies on multiplayers of detecting capabilities, look at more than just file signatures.
- **Behavior-base detection** looks at the action that an executable takes such as *accessing memory or filesystem, changing right, or perform suspicious action*.
    - Look at its behavior help *identify morphic virus, change themselves* so that simply *hashing their binaries will not allow them to be identified*.
    - Now anti-malware software may even roll back action that makware tool while it was being monitored. (this type of monitoring call, "Dynamic Analysis")

Traditional malware signature creation and behavior based detection creation scenario, malware samples are placed into isolated sandbox with instrument that tracks what the malware does.

- Info used to create both signature from files or other materials that malware creates, install, modifies, and actions it taken.
- **Automated malware signature creation** done using this technique so that antimalware providers can provide signatures more quickly.

Policy, Process, and Standards

Admin controls that *involve policies, process, and standards* are necessary layers when *looking at complete layered security designs*.

- Change controls
- Config management
- Monitoring & response policies
- Personnel security controls
- Business continuity & disaster recovery controls
- Human resources like background checks & termination

Most common personnel controls are as follows:

- **Separation of duty:** individual given roles, can abuse the rights that the role provides.
    - Separation of duty *requires more than one individual* to do the job
    - Typical example, payroll, accountings. Person should not able to modify data without being detected.
- **Succession Planning:** ensure continuity for roles. Departing staff, can take expertise and skills with them. This can create single point of failure.
- **Background Check**
- **Termination:** ensure their access all terminated. Ex. laptops, mobile device, data. Account needs disable.
- **Cross Training:** focus on teaching employee that enable them to take on tasks that their coworkers normally do. Prevent single point of failure.

Note: Succession planning and Mandatory vacation made easier if appropriate cross training occurs.

- **Dual control:** process requires 2 people to do an action together.
    - This control is useful when sensitive task are involved, requires both parties to collude for a breach to occur.
    - Dual control can be implemented as an admin control via procedure or technical control.
- **Mandatory vacation:** require staff to members to take vacation, allow to identify if individual exploit the rights or not.

Analyzing Security Architecture:

Analyzing a security infrastructure to identify where defenses are weak, where an attacker able to exploit flaws such as design, process, procedure.

- **Control gap, single point of failure, and improperly implemented controls** are common issues, likely to encounter.

**Security architecture can analyze based on their attributes by reviewing security model and ensuring that it meets specific requirements.

→You will *encounter 3 scenarios* if asked to review security design that used **antimalware software**:

- **Success:** Antimalware software can be successfully prevent unwanted software installation
- **Failure:** will not detect software, is not permitted by Org policy, but not malware.
- **Failure:** may not prevent unknown malware, does not act in ways that are typical or known malware.

→This type of attributes-based testing can be performed based on **Risk Assessment** and **Control Plan** to determine whether security architecture meet control objectives.

Reviewing Architecture: common views be taken when reviewing architecture including:

- **Operational View:** how **function is performed**, what it accomplished.
    - *This view* typically *shows how info flows*, *not capture technical detail*s *about how data is transmitted*.
    - Useful for understanding what is occurring, often influence procedural or admin control
- **Technical View, AKA (service-oriented or system-based view):** *focus on tech, setting, config* used in architecture.
    - Help *identify incorrect config* and *insecure design* decision including details TLS version of a connection, specific setting for password length
- **Logical View**: *describe how system interconnect*.
    - Less technical details than technical view, but convey broader info about how system or service connect or work.

**British Ministry of Defense's Architecture Framework (MODAF) use 7 cate: strategic, operational, service oriented, system, acquisition, technical, "All view point".

**U.S Department of Defense (DoD) sue similar set of view architecture: all, capability, data and info, project, services, standards, and system viewpoints are considered in DoD model.

Common Issues: Security Architecture design concepts and controls.

- *4 most common encountered design issue*: single point of failure, data validation, trust problems, user issues, authentication and authorization security and process problems.

****Single point of failure:** key element to consider when analyzing layered of security architecture.

Solution: Redundancy removes Single point of failure.

****Data Validation:** ability to trust data used for data processing or to make decisions.

The ability to reply on data can be enhanced by:

- *Protect data at rest & in transit* using **Encryption**
- *Validate data integrity using **File Integrity checking*** tools, or perform data integrity validation
- Implement processes to verify data in an automated or manually
- Profile or boundary checking data based on known attributes of the data

→Web Application security, familiar with exploiting weakness data validation scheme.

→Insecure Web Application often trust that data where they receive from a page that they generate is valid, use data to do further tasks.

Note: when analyzing data flows, storage, and usage: remember to look for places where data issues could use high-impact failures.

- **Data Validation** issues in a design far more critical to overall design, making effective control placement extremely important.

****Users:** human errors cause of failure in security architecture, and it is due to user are limited by:

- Automated monitoring & alerting systems to detect human error
- Constraining interface to only allow permitted activities
- Implement procedural checks and balance separation of duty & other personnel controls
- Training and awareness programs

****Authentication and Authorization:** user cred, passwords, user rights are all areas, often creates issues. Common problems includes overly broad user rights, poor cred security or management, embedded and stored passwords and keys, reliance on passwords to protect critical systems.

→Security Design seeks to avoid these issues are:

- MFA
- Centralized accounts & privilege management and monitoring
- Privileged account usage monitoring
- Training and awareness efforts

<When analyzing security design for potential authentication & authorization issues.>

- 1st step, *identify where authentication occurs, how authorization is performed*, and *what rights are needed & provide to users*.
- 2nd step, once understand those, *focus on controls that are implemented* and *where those controls may leave gaps.*

<Remember to consider technical controls, process and human factors.>

Continual Improvement AKA (*CIP or CI processes):* continual improvement processes design to provide incremental improvement over time.

Retirement of Processes: security processes and policy becomes outdated, need to be retired. Often the case for one of a few reasons:

- Process or policy is no longer relevant.
- Has been superseded by a new policy or process.
- *Org no longer wan*ts to use the policy or process.

Retirement of processes should be documented or recorded.

Conclusion:

- Security solution are vital part of infrastructure: Layered security include physical and virtual level goes along with the concept of "system isolation, air gapping" as a part of string defense design.
- Defense-in-depth: Common design use "segmentation to separate different security level".
    - Network architecture take into account of "physical network devices and design, cloud option and SDN". Cloud tool: private cloud (CASB).
- Understanding requirement and identifying potential SPOF**:** include data validation issue, problems cause by trust and requirements for trust versus whether data or system can be trusted, user-related failure, authentication & authorization.
- Maintain layered security design requires "Continual review & validation": scheduled review help ensure the design hasn't become outdated.

Chapter 8: Identity and Access Management Security

Identity, set of claims made about individuals or account holders, made to one party or another such as service providers, system, application, are key part of **authentication, authorization, and accounting.**

- User acc, log in require ability to uniquely identify individuals, other subjects such as service to allow permission, rights, group membership, and attributes.
- **Attributes** associate with identity include subject such as name, address, title, contact info, other details about them.
    - **These Attributes maybe** used as part of authentication processes, maybe used to populate *directory info* or *could be collected.*

- **Traits** are inherent part of subject like hair color, nationality, birthday.
- **Preferences** based on person's choices of their favorite color or band.

Identity used as part of *authentication, authorization* and *accounting (AAA) framework,* used to controls access to computer networks, services. **AAA** authenticate users by requiring credential like username & password, possibly biometric or token-based authenticator.

- Once individual proven who they are, then they're authorize to access or use resources or systems.
- **Authorization** apply policies based on user's identity info & rules or settings, allowing owner of the identity to perform action or gain access to system.
    - **Ongoing management** of these rights is known as "*Privilege management".*
- **Accounting, AAA:** is the logging and monitoring that goes with authentication and authorization.

    - **Accounting** monitor usage and provide info about how & what user are doing

Central Management of identity, *occur in identity and access management (IAM)*

- **IAM,** *built to create, store, and manage identity info as well the permission, groups and other info needed to support the use of identities*.
- **Data** to *create identities* come from systems of *record like Org's ERP, HR system, customer's database*.
- **Data** is *supplied to identify management systems* and *service which provide account creation, provisioning, management, and oversight for Org*.

Identity Systems and Security Design

Identity provide common functions: identity creation and management, authentication and authorization, in some cases, federation of identity info to allow use of identities outside of their home org.

**To enable identity system, use common tech such as *directories, authentication services, identity management platforms, and federated identify tools*.

- **Directory:** directory services used in networks to provide info about systems, users, info about Org. Ex. LDAP common deployed as part of an **IAM**.
- Several LDAP servers such as Open-LDAP, 389 directory server, ApacheDS, OpenDJ..etc. Thus, steps required to implement a secure LDAP server will vary. These includes:
    - 1st: Enable the requiring TLS to keep LDAP queries & authentication secure.
    - 2nd: Setting password storage to use a secure method. **LDAP** passwords are often stored in plain text.

→Using OpenLDAP, the SSHA password storage scheme uses a salted SHA (Secure Hash Algorithm). This is stronger than CRYPT, MD5, SHA, and Simple Authentication and Security Layer (SASL).

- Using passport-based authentication, and requiring TLS: LDAP provides 3 modes of operating:
  Anonymous, authentication, username/password authenticated.
    - When *authenticated sessions are turned on*, unauthorized mode should be disabled to prevent issues with unsecured connection.
- Replication of LDAP servers, help prevent denial-of-service attacks.
- ACL for LDAP, offer the ability to limit access to specific object in directory as well as overall rules for how entries are created, modified, and deleted.

Note: LDAP Injection, which use improperly filtered user input via web applications to send arbitrary LDAP queries.

Authentication Protocols: centralized servers allow clients to authenticate to a central authentication service, then supplies verification of user's identity to the replying system.

→Central authentication services also provide authorization info for user to the relying party, may match identity with their own authorization and rules. Common authentication protocols:

- **TACACS+: Cisco** design extension for **TACACS**, terminal access controller access control systems.
    - Use TCP traffic to provide authentication, authorization and accounting services.
    - TACACS+ suffers from a number of flaws, include lack of integrity checking for data it sends, allow attacker with access to the traffic *it sends to make arbitrary changes or use replay attacks against the TACACS+*.
    - TACACS+ also has encryption flaws, lead to compromise of encryption key.
- **RADIUS,** Remote authentication Dial-in User service: AAA system for network device, wireless network, other services.
    - RADIUS operate via TCP or UDP and operate in a client-server model.

        - RADIUS sends password that is obfuscated by a shared secret and MD5 hash, meaning its password security is not very strong.
        - *RADIUS traffic between RADIUS network access server and RADIUS server* is typically encrypted using IPSec Tunnels.
- **Kerberos** unlike TACACS+ and RADIUS: *designed to operate on untrusted network* and uses encryption to protect its authentication traffic.
    - *Users in Kerberos called principals, are composed of 3 elements*: Primary (often username), the instance (used to differentiate similar primaries), the realm (consist group of principals).
        - **Realm** often separates on trust boundaries and have distinct key distributions center (KDCs).

AD uses Kerberos for authentication, Older version of windows relies on NTLM authentication. **NTLM is out-of-date.

Single Sign-On and Shared Authentication Scheme: Many web application rely on SSO, allows users to authenticate once and then user multiple systems or service without having to use different username or password.

- **Shared Authentication scheme** similar to SSO and allow an identity to be resued on multiple sites while reply on authentication via single identity provider.
- **Shared Authentication system** require users to enter cred when authenticating to each site, unlike SSO system.

Note: SSO don't need Username/Password, but Shared Authentication system does.

OpenID: you use Gmail Acc to sign into Spotify (so Google send your info to Spotify). *Use Cred from 3rd party for **authentication**.*

Oauth: Never share password. (Click on FB login when log into another web, Web use user info from FB then log from in. *(Use for authorization token).*

Common tech include LDAP and Central Authentication Service (CAS). Shared Auth include following:

- **OpenID:** open source, for decentralize auth. **OpenID** used by Google, Amazon, MS.
    - User create cred with identity provider like Google; site (relying party) use that identity
- **OAuth:** open authorization standard, used by Google. Allows user to share elements of their identity or account info while authenticating via original identity provider.
    - **OAuth** rely on access token, are issued by authorization server and then presented to resources servers like 3rd party web app by client.
- **OpenID Connect:** *auth layer* built using the OAuth protocols.

**One of SSO significant security benefits is reduce password reuse.

- This also reduce likelihood of cred exposure via 3rd site, users use cred sets.
- Shared Authentication allow users to sue their cred without create new account on each site.

Access Control Models: auth and authorization manage who can access or use resources is critical part of IAM system & privilege management.

- Org typically choose access control model based on factors as ***level of control*** they need, ***security*** and ***compliance requirement, and technical & administrative capability to*** implement and maintain them.

1. **Role-base Access Control**: use role associate with job function or other criteria. A subject can have more than 1 role. 3 common constraints implement RBAC:
    - Subject can only use permission if they have a role.
    - Subject active role must be one it's authorized to have.
    - Subject can use permission only if subject's active role authorized to use.

→RBAC can implement both Discretionary access control (owner of info) and mandatory access control (based on sensitivity label).

2. **Attributed-Based Access Control:** gives user right based on policies. Policies use collection of attributes to determine which access rights to grant.
   - **ABAC** tend to be used when a flexible context-sensitive access control mode is required.
   - Combing attributes, describe subject like their role, division or personal attributes)
3. **Mandantory Access Control:** MAC rely on OS to control what subject can access and what actions they can perform.
   - **MAC** usually rely on system admin to implement access control policies.
   - **MAC** has been associated with military system.
4. **Rule-Based Access Control:** use a set of rules implement by admin. **ACL** typically associated with each object, rules are checked against that ACL when access is required.
5. **Discretionary Access Control (DAC):** delegate control to admin or owners of protected resources. Allow delegated control, require trust in the choices that owners and admin make. *Cause issue due to a lack of central access control*.

Threats to Identity and Access:

1st: Threat to underlying authentication and authorization systems seek to exploit in a way that user log in, how their cred are handled or they are authorized.

2nd: Attacker target Acc life cycle by creating cred, preventing them from being removed, causing them to have greater privilege associated with them.

3rd: attacker focus on accounts via phishing or compromising systems where credential may be stored.

Understanding Security Issues with Identities:

Identities including cred, roles, rights, and permissions and related data.

- Personnel-based identity security, include *training and awareness as well insider attackers, phishing, and social engineering*.
- Endpoint and their role in attackers on identity including *capture cred via local exploits, screen capture and keyboard capture app, local admin rights, how password stores, token, other cred store in phone and tablets*.
- Sever-based exploits, can target systems run identity services, which can attack the servers and send identity and authentication data to AAA services.
- Application and services that provides, consume, interact with identity systems.
- Roles, rights and permissions, associated with users or groups,

Attacking AAA Systems and Protocols: identity repository like directory systems, authentication systems and SSO user. Attacks against identity repositories may target the specific software via vulnerability or misconfig.

LDAP Attack: used for authentication and directory info. LDAP server typically focus on:

- Binding connection methods that target unencrypted LDAP traffic, either to *capture the traffic* or to *exploit LDAP as authentication service*.
- Improper LDAP access control allow attacker harvest directory info or to make modification to directory entries.
- **LDAP Injection** which exploit web application that build LDAP queried using user input, allow attacker gather additional info to make changes
- **DOS attack**: disrupt authentication services that rely on LDAP.

Binding: step where LDAP server authenticate the clients, authentication based on client privilege.

++Requiring secure binding methods, setting appropriate access controls, using good web app.

Connectionless LDAP service was found to be potential attack vector, could respond to spoof addresses, resulting in amplification rate up to 55 times higher than the source traffic.

OAuth, OpenID and OpenID Connect: Most *common attack* is the use of open redirects*. When directs and forward are not validated*, *untrusted user input can be sent to the relying web* *application*, ***resulting in users being redirected to untrust sites, allowing phishing scams or permitting attackers***. (similar to SQL Injection except, attacker redirects users to untrusted sites.)

Scenario: if user access website that is **open redirect endpoint**, will allow URL at point A to be any redirect URL, instead of a specific URL associated with that site, and if the site also passes that URL forward at point B, attacker can exploit authorization flow.

→Fortunately, this won't cause the account associated with service provider to be compromised—only cause issue for the site with the open redirect endpoint since redirect can result in phishing scams.

- **OAuth2** can be vulnerable to CSRF which focus on getting a user to click a link that causes that user's browser to perform an action at that user.
- **OpenID Connect**: offer additional protection for encryption and signing.

Kerberos: rely on central key distribution center (KDC). Compromised of KDC allows attacker impersonate any users. Common Kerberos attack include following:

- Admin account attack.
- Kerberos ticket reuse, include ***pass-the-ticket attack***, allow impersonation of legitimate users of lifespan of the ticket, ***Pass-the-key attack***, which reuse a secret key to acquire ticket.
- TGT, focused attack. TGT, valuable and can be created with extended lifespans.
    - *When attacker succeed in acquiring TGT*, the TGT are often called "Golden Ticket" because they *allow create complete access to Kerberos connected systems, include creation of new ticket, account changes, even falsification of acc or services*.

Note: Automated monitoring authentication and authorization help detect anomalous behavior like creation of a golden ticket.

RADIUS: common used for authentication of network devices, include VPNs, network hardware. RADIUS attack focus on following:

- **Session reply of server** response by matching known traffic & replaying previous response or replying server response to authenticate client without valid cred.
- **Targeting RADIUS shared Secret** since RADIUS uses a fixed shared secret that can be compromised at client level.
- **Dos attack** preventing users from authenticating.
- **Credential-based attack** rely on the user of RADIUS shared secret to brute force shared secret given a known password.

Solution: using TLS to protect RADIUS authentication.

Active Directory: AD core identity store and AAA service for Windows centric Org. Common attack:

- **Malware-based attack**, seek to place credential capturing or exploit-based malware.
- **Credential theft** via phishing or other technique.
- **Privilege escalation attack** using known or new Windows exploits.
- **Service Accounts** often forgotten.
- **Domain admin Rights** that exist for more staff than is necessary.
- **Use of down-level version** of protocols, used in Windows domains, NTLM v1 and LAMMAN, NetBIOS, and unsigned LDAP and SMB to capture.

Targeting Account creation, Provisioning, and Deprovisioning: Account requests to creation provisioning of account maintenance, the deprovisioning and deletion or account life cycle.

→**Internal threats** seek to create accounts for their use to avoid detection. Once an account exists, attackers will focus on gaining access.

++Major threats from unused or improperly maintained account include:

- **Unused Acc**

- Account that were not properly deprovisioned and abandoned because they were missed during normal account removal or end-of-life process. Acc that properly deleted.
- Permission, group membership, other privileges often accrue to account. Attacker particularly insider threats, able to leverage rights.

Right and Role: Maintain rights and roles, group membership is another key element in identity management and crucial key feature in identity management system.

- Acc normally managed using *Least Privilege, limit the exposure.*
- Centralized Identity Management suite provide monitoring and privilege management tool design to monitor for privilege creep and can be set to identify Acc that end up with excessive privilege beyond what they need.
- Identity Management system like Centrify, Okta, Sailpoint, Ping identity have a acc life cycle and monitoring features designed up to fight this type of issue.

Prevent Common Exploits of Identity and Authorization:

- ***Impersonation Attack***: attacker take on identity of legitimate user, allow impersonation to occur
    - **Solution**: Preventing impersonation requires "Strong session handling", "Securing Session identifier".
- **MITM:** rely on accessing info between system or services.
    - **Solution**: End-to-end encryption of session.
- **Session Hijacking:** focus on taking over already existing session, by *acquiring session key or cookies, used by remote server to validate the session* or by *causing the session to pass through a system*.
    - MITM attack, securing the data and attacker send to acquire to hijack session.
    - **Solution:** Either via *network encrypting session* or links or on the local system. Limit opportunity for session hijacking.
- **Privilege Escalation:** focus on exploiting flaws to gain elevated permission. Often rely on software vulnerabilities.
    - requiring admin to ensure local application, service, utilities are not vulnerable.
- **Rootkit**: combine multiple malicious software tools to provides continued access to computer while hiding existence.
    - **Solution:** full suite of system security practices, patching, layered of security design to antimalware technique, whitelisting heuristic detection, software detection tools.

Acquiring Credential: *Attack against AAA and identity management acquiring identity and credential*.

- Attack against cred occur in phishing attack, compromise of other services, brute force.
    - Phishing: aimed at cred often use replica of legitimate authentication portal to trick victim enter their cred.
        - More advance version, even reply those entries into legitimate site to prevent their target from noticing that their login did not work.
        - **Solution:** MFA limit impact of phishing, also user education and training.
    - Compromised other services: attack 3rd party services, obtain password that may have been reuse. Attacker, obtain plain text or recoverable passwords can reuse those password on other acc.
    - Brute Force attack: Preventing brute force requires building in the back of algorithms that prevent repeated logins after several failures or similar solution like CAPTCHA to verify the login are not being attempted by a script or bot.
        - Implement account lockout to help with brute force attack.

Identity as a Security Layer: identity is critical part of most defense in depth design. User and service acc are crucial to controlling access to systems and services, also allow detailed monitoring and auditing of usage.

- Rights and permission are assigned to role that acc are associated with or to individual users, identity critical to ensuring that rights management is handled properly.

****Acc Life Cycle:**

- Identity creation: ensure *only valid acc are created*. Avoid duplicate acc creation, ensure that initial authentication are delivered or set securely, acc are added to a central identity management for monitoring and auditing.
- Acc provisioning and rights management: needs to be *consistent, unmanaged system or system that do not integrate can result in unmanaged accounts*.
    - Right management is typically role-based, preventing in individual acc from accruing specialized permission.
- Acc modification and maintenance: *track individuals changing roles and group memberships to prevent privilege creep*.
- Acc termination: need to *ensure that acc are terminated*, are removed from all system that were provisioned to.

****Defense in depth for identity**: should address CIA, ensuring cred and cred stores remain confidential in motion and at rest and that their integrity is monitored to ensure unauthorized do not occur.

Authorization and Rights Management: Matching users with rights and roles and group membership is the next step of identity based security.

→Rights management allow *access control* by matching users with the access they should have.

Building right management security layer relies on:

- Building a set of policies that: describe what rights are allocated to each role or task
- Implement a management system: *ensure that rights are granted to account & groups*, removed from groups & users that do not hold the appropriate roles.
- Monitoring and reporting ensure rights management: *ensure rights management occurs according to policy*.

Note: *Privilege User Management*, the management of admin and super-user right. Privilege user have the ability to override system policy to make changes to login and oversight system.

MFA relies on few common types of authentication factors and methods:

- *Knowledge factor* are sth you know. Ex. Password, passphrases.
- *Possession factor:* sth that you have. Ex. Security token, smartcards or app that provide the code
- *Biometric factor*: sth that you are. Ex. Include fingerprints, retina scans, voiceprints, measuring features of the human body.
- *Location Factors*: less frequently used, rely physical location, determine by where a system or network is located by using GPS or other data to verify that you are in a place that is trusted or allowed to access a system. (**Conditional Access: trusted location)**

Context-Based Authentication: Context-based authentication allows authen decision to be made based on info about user, the system the user is connecting from.

**Common data used for context-based authentication:

- User roles and group membership related to app or service access.
- IP address and IP reputation whether the remote IP is known to be part of a botnet or other IP range with known bad behavior.
- Time of the day, related to a job role or working hours
- Location-based info like their IP address or GPS location.
- Frequently of access, combined with behavior data like keystroke patterns, browsing habits.
- Device-based including about web browser in use and other data, can provide a device fingerprint as its IP address, time zone, screen resolution, cookies or cookies settings, installed fonts, and language.

++A user log in via org VPN where NAC system profile that user's device, identify service-base fingerprint info.

++ User provide their username/password, this example do not match the device. The user has never logged in from it before. Due to this, user is asked to provide a onetime password code from a security token and is then authenticated, having proven that they're who they say they are.

++NAC server records, new device as a valid, trusted device and adds its unique profile to its database, user is connected via VPN.

***Identity as a Service** (IDaaS):* provide authentication services, typically was a cloud-hosted service.

- Identity life cycle management: consist of tech and processes to create provision, manage identity for system, services.
- Directory service: using LDAP, Active directory, another directory tech
- Access Management: with both auth and authorization capabilities
- SSO support: Security Assertion Markup Language (SAML) integration, OAuth, other standards
- Privilege acc management and monitoring
- Reporting, auditing , provides oversight and visibility into the identity life cycle.

Note: IDaaS can create new security concern for Org due to hosting an identity store or an authorization system outside its internal network. Understanding how IDaaS providers handle secure identity info, what their incident response practices and noti policy, performing due diligence all vital parts.

***Implement a cloud hosted identity service* can mean significant change to internal AAA system design:

- Deciding whether Org will centralize their directory services or internal and 3rd party hosted directory will both exist.
- A decision must be made to centralize authentication or to federate multiple auth and authorization.
- Location for Org authoritative cred store maybe local or cloud-base.

Detecting Attack and Security Operation: SIEM use to leverage identity info. Using identity info provide the "Who" when reviewing events and incidents.

- When pair other SIEM data and event logs, provides complete view of what occurred and what user services or account's behavior was.

++configure SIEM or other security monitoring device looks for:

- Privilege Acc Usage
- Privilege change or grant
- Acc Creation and modification
- Employee termination and terminated acc usage
- Acc life cycle
- Separation of duty violation

Centralizing Both IAM and user auth and authorization system ensure accounts and privilege.

Final layer of any identity-base security is active monitoring and administration.

Federate and Single Sign on: federate identity, process of linking an identity and its related attributes between multiple identity management system.

- Each site allows use of their cred as well a set of attributes by 3rd party.

Federated Identity Security Considerations: Federate identity move trust boundaries outside of your own org. This lead to the need look at federate security from 3 point of view:

- IDP: member of federation must provide identity, make assertion about those Identity to relying party.
    - Service provider maybe responsible for providing incident response coordination for federation, communication between federated members.

- Replying Party (RP) or service provider (SP): members of federating must provide services to members of federations, should handle data from both users and identity providers securely.
- Consumer or user of federated service may be asked to make decision about attributes release, provide validation info about their identity claims to IDP.

Federated Identity Design Choices: in federation model rely on verifiable identities, greater level of assurance about users' identity claims is needed.

- Integration with 3rd party federated identity services works best when provisioning occur when user request access with immediate acc provisioning.

Federated Identity Technology: 4 majors of federated tech are SAML, ADFS, OAuth, OpenID Connect.

- SAML: Authorization, authentication and potential risk: DOS, protocol usage & processing risk. Used by enterprises in Linux-centric.
- OpenID: Authentication, Potential risk: CSRF/XSS, message confidentiality, redirect manipulation.
- OAuth2: authorization, some authentication, potential risk: similar to OpenID except XSS/CSRF. Used by API.
- ADFS: authentication and authorization, potential risk: token attack (replay, capture). Used by windows-centric.

SAML: XML-base language, used to send authentication and authorization data between identity provider and service provider.

- Often used to enable SSO for web app and services because SAML allows identity provider to makes assertion about principals to service provider so that they can make decision about user.
- *Allow authentication, attributes, authorization decision to be exchange*.

ADFS: provide authentication and identity info as claims to 3rd party partner sites.

→ADFS use similar process to an OAuth auth process:

1. User attempt to access and ADFS-enable web app hosted by a resource partner
2. ADFS web agent on the partner's web server check for ADFS cookies, if it is there, access is granted. If not there, user is sent to ADFS partner's server.
3. Resource partner's ADFS checks for SAML token from account partner and if not found, ADFS realm discover.
4. Home realm discover identify federate services associated with user then authenticate the user via home realm.
5. Acc partner the provide a security token with identity info in the form of claims, send user back to resource partner ADFS server.
6. Validation then occur, uses it trust policy to map the acc partner claim to claim web support.
7. New SAML token is created that contains resources partner and this cookie is stored on user's computer. User is then redirected to web app, where app can read the cookie and allow access supported by the claims.

ADFS: controlled using ADFS MMC snap-in, adfs.msc. ADFS console allows you to add resource partner and acc partner, map partner claim, manage acc store, config web app and support federation.

OAuth: *OAuth2.0 provide authorization framework, allow 3rd party to access HTTP-base services*. Provide access delegation, allow service provider to perform action for you: OAuth flows recognize 4 parties:

- Clients: the app user want to use.
- Resource owner: end user.
- Resource server: server provided by a service that resource owner want the app to use.
- Authorization server: servers owner by identity provider.

OpenID Connect: often paired with OAuth to provide authentication, *allow authorization server to issue an ID token in addition to authorization toke provided by OAuth*.

- Allow service to know action was authorized and user authenticated with identity provider.

Federated incident Response: building a response plan for federated identity varies based on role your Org holds in the federation:

- IDP responsible for notifying acc owner, relying party. Incident response policy needs to be envision compromise of IDP itself as well as event required password reset for all user entail.
- Service provider need to determine their response would be if the IDP was compromised, a range of smaller incidents include their own authorization system or limited compromised of acc provide by IDP.

Inconclusion:

++AAA is a part of IAM. IAM system manage: user acc life cycle, rights, privilege, provide oversight of identity.

++**AAA system** include LDAP directory server, Kerberos, RADIUS, Active Directory.

++**IAM system**: create, manage, store, and monitor identity and authorization.

++**Key element of IAM is** directories, authentication system, protocols, SSO, shared authen service, federate identity system.

Important part of IAM is "Review rights granted to subjects".

++**OAuth open direct:** is a part of impersonation attack.

++ privilege escalation, impersonation, MITM involves identity and access management technique.

- Session hijacking, cross-site scripting and rootkit, impact controlling of privilege. **Solution:** ensure proper auth and authorization and logging and tracking acc access and privilege acc activity.

Chapter 9: Software and Hardware Development Security

- Software Development Life Cycle (SDLC) are:
- SDLC is useful, provide consistent framework to structure workflow & provide planning for development process.

Software Development Phase: SDLC model:

1. **Feasible Phase:** *initial investigation*, look at alternative solution and *high-level costs each solution* proposed. It results in recommendation with a plan move forward.
2. **Analysis and requirement definition phase:** *customer input* is sought to *determine what the desired functionality is*, what current *system* or app *currently does and doesn't do*, what improvement are desire.
   - Requirement maybe ranked to *determine* which are *most critical to the success of the project*.
3. **Design Phase:** include *design for functionality*, architectures, *integration* points, technique, dataflow, business process, other *element that require design consideration*.
4. **Actual Coding**: *occur during development phase*. Involve testing or parts of the software include, unit testing (testing small components individually to ensure, function properly) and code analysis.
5. **Testing**: *some testing occur in development phase*, *formal testing with customers*, others outside of development team occurs in the testing integration phase.
   - *Individual units or software components integrated and then test* to ensure proper *functionality.*

    - Connection to outside service, data source and other integration may occur during this phase, **User Acceptance Testing** (UAT) occur to *ensure users satisfied* with its functionality.

6. Important task: ensure user are trained on the software, occur in the **Training and transition phase (***this phase AKA acceptance, installation, development phase*)
7. Once project reaches completion, enter **ongoing operation and maintenance,** include patching, updating, minor modification, daily support.
8. **Disposal Phase:** *occur when prod or system reach end-of-life*. **Important Phase**: *shutting down old product can produce cost saving*, replacing existing tools may require specific knowledge, extra effort, data, system may need preserved.

****Test Environment** where software or systems, being tested without impacting prod. Some would use preprod which is staging environment.

Software Development Models:

1.Waterfall (Old legacy): sequential model, each phase follows by next phase.

- Phase 1: Requirement, gathered and documented.
- Phase 2: Analysis: build business rules and models.
- Phase 3: Design and Code.
- Phase 4: Integration of software.
- Phase 5: Testing and debugging
- Phase 6: Operational: supports, maintenance happens daily.

→**Waterfall** has been replaced due to inflexibility. (*Not highly responsive to changes, not account for internal iterative work*).

→Recommend for: *development that involve fix scope, known timeframe for delivery*, well-understood.

2.Spiral: use linear development scope from Waterfall, but add iterative process that revisit 4 phases multiple times. (Identify, design, build, evaluate)

- Put vital emphasis on ***Risk Assessment*** *as part of SDLC*.
- Phase 1: Identification, requirement gathering: gather business and system requirement.
- Phase 2: Design, conceptual, logical, architecture, some physical or final design.
- Phase 3: Build, produces and initial POC& further development release until final prod.
- Phase 4: Evaluation: involve **Risk Analysis.** As development cycle continue, involve customer testing and feedback to ensure customer acceptance.

→***Spiral mode*** *provides greater flexibility* to handle changes, allow software development life cycle to start earlier in the process than **Waterfall** does.

- **Spiral** revisit process, possible result in work or to identify design requirement in process that require a vital design change due to more detailed requirement coming to light.

3.**Agile:** *Iterative and incremental process* rather than linear process.

- **Individual and interaction,** more important than process and tools.
- **Working Software if preferable** to comprehensive doc.
- **Customer collaboration** replace contract negotiation.
- Responding to change is key, rather than following a plan.

→**Agile**: *break up into small units*, allow work to be done more quickly with less up-front planning.

- *Focus on adapting to needs*, rather than predicting, identify early in the process, but *subject to changes* as the project continues to develop.
- Work broken up in to short sessions, called "*sprints"* last days to a few weeks.

Agile based on 12 methods:

1. Ensure customers satisfaction via early and continuous delivery.
2. Welcome changing requirement, even late in development process.
3. Deliver working software frequently (in weeks rather than months).
4. Ensure daily cooperation between developers and business people.

5. Projects, built around motivated individuals who get the support, trust
6. Face-to-Face conversation, most efficient, convey info inside development team.
7. Progress is measured by having working software.
8. Development should be done at a sustainable pace, can be maintain on ongoing basic.
9. Pay continuous attention to technical and good design.
10. Simplicity
11. Best architecture, requirement, design
12. Team should reflect on how to become more effective, implement behavior at regular interval.

Agile development use a number of specialized term:

- **Backlog** are listed of feature or tasks require to complete a project
- **Planning Poker**: tool, estimate and planning used in agile dev.
    - Estimators, given card with value for the amount of work required.
- **Timeboxing**: term describe used of timeboxes (agree to work on time, a person or team use to work on a specific goal).
- **User Stories:** describe higher level of user requirement.
- **Velocity tracking:** conduct by adding up estimate for current sprint's effort, then compare what was completed.

Rapid Application Development (RAD): iterative process, reply on building prototype.
++RAD rely on functional components of the code being develop in parallel, then integrate to produce the finished product. **RAD** involves 5 phase:
1.**Business modeling**: focus on business model, include how it's processed, what business process should involve.
2.**Data modeling:** include gathering & analyzing all datasets and object needed the effort and defining their attributes and relationship.
3.**Process Modeling:** for dataflow on business model as well description fir how data is handled.
4.**Application Generation**: through coding and use of automated tools to convert data and process model into prototype.
5.**Testing and turnover**: focus on dataflow and interface between component since prototype are tested at each iteration for functionality.

Other models:

- The V model: extension of Waterfall model, pairs a testing phase with each development stage.
    - Each stage starts only after the testing is done.
- Big Bang SDLC: rely on no planning or process. Focus on making resources available, simply starting coding based on requirements as they're revealed.
    - Big Bang model doesn't scale, common model individual developers working on their own code.

DevSecOps and DevOps:
****DevOps:** combines software Dev and IT operation with goal of optimizing SDLC. Done using toolchains to improve, coding, building and testing, packaging and release, config and config management, and monitoring elements of SDLC.
****DevSecOps**: describe terms of security in DevOps model.

- Its role includes threat analysis, communication, planning, testing, providing feedback, ongoing improvement, awareness responsibility.
- Require understanding risk tolerance, awareness.

++DevOps and DevSecOps combine continuous integration & continuous deployment methodology where rely on automated security testing & integrated security tooling include scanning, update and config management tools.

Continuous Integration and Continuous Deployment:

- 1st step: Developer commit changes

- 2nd step: Build process is triggered.
- 3rd step: Build report is delivered.
- 4th step: Test run against build.
- 5th step: Test report delivered.
- 6th step: if successful, code is deployed.

Continuous Integration (CI): development practice, *check code into a shared repository on a consistent ongoing basic*.

CI:** rely on automated build process, require automated testing*. Often paired with ***Continuous Deployment (CD) AKA "Continuous Delivery" which roll out tested changes into production automatically.

++Using **CI and CD** require building automated security testing into pipeline testing process.

- Can result in new vulnerability being deployed into prod, allow untrusted or rogue developer to insert flaws into code that is deployed.
- Then remove code as part of a deployment next cycle.
- Logging, reporting, monitoring must designed to fit CI/CD process.

Designing and Coding for Security:

**During testing phase, fully integrated software can be *tested using tool* like *Web Application Security Scanner* or *Pen Testing Technique*. Throughout these steps, help understand the common security, developers face, create, and discover.

++Software Flaws, you may encounter:++

- ***Improper Error Handling:*** result In error message that shouldn't be exposed to public (don't be in detail)
- ***Dereferencing issue:*** *due to null pointer dereferences*. A pointer will valid of NULL (one that isn't set), lead to a crash unless caught by an error handler. (**Race Condition** is a dereference issue)
- ***Insecure Object References:*** when expose info about internal objects, allow attackers to see how object is identified and stored in a backend system.
  - Attacker may leverage info to gain further access.
- ***Race Condition:*** rely on time. Ex. When 2 people, reserve the seat at the same time, and the system wasn't not updating ontime.
- ***Broken Authentication:*** improper implement authentication, wo are not logged in as user with the correct rights, access to resources.
- ***Sensitive Data Exposure:*** occur any of a number of flaws are exploited.
- ***Insecure Components:*** include a broad range of app or services is vulnerable. Thus, introduce vulnerability.
- ***Insufficient Logging and Monitoring:*** what should be logged and monitored.
- ***Weak or default configuration***
- ***Use of Insecure Functions:*** strcpy allow data to be copied without caring whether source is bigger than the destination.
  - Attacker can place arbitrary data in memory location past the original destination, possibly allowing buffer overflow attack.

Security Implications of Target Platform: when looking at software development security, the language that is used, modules, framework, how testing and validation are done, code will run on all important.

**OWASP most recent mobile vulnerability list: insecure communication, insecure authentication and authorization, insufficient, cryptography, code quality, and reverse engineer.

Embedded system such as System on chip (SoC)** which embedded a complete computer in a chip, provide additional security because not as accessible, but often come with less frequent updates or inability to update easily.

➔Both embedded system and SOC may have hardware, firmware, and software vulnerability.

➔Common platform for Application is **Client-server application model.**

- Client: web browser, app, other clients)
- Web application, attack maybe conducted against the client, network, traffic sent between client and server.

****Firmware**, is embedded software used by a computer or hardware device. Firmware flaw is hard to fix, not all device designed to update their firmware.

    - Attacker who target firmware, often seeks copy of firmware directly to device and downloading it.

Secure Best Practices: develop, implement, and design best practices:

- **Have a secure coding policy**: server as foundation for secure development practices & standard
- **Risk Assessment**: understand what risk the app faces to prioritize the remediation.
    - Continuous assessment is recommended, schedule testing tools inform risk assessment process.
- **Input Validation**: prevent cross-site scripting to SQL injection.
- **Output encoding:** translate special characters into equivalent, but safe version before a target app or interpreter read it.
    - Prevent XSS attack by special character being inserted that cause target app perform action
- **Web Application Firewall:** prevent attack against vulnerability app and offer line of defense that don't have available patch or cannot be taken offline.
- **Error Message management:** ensuring error message don't leak info, ensure attacker can't use error message to learn about your app.
- **Database Security:** both app and database help ensure that data leaks don't occur.
- **Using parameterized queries:** *precompile SQL* that take input variable before it executes. Prevent SQL attack.
- **Securing Sensitive Info:** by encrypting mechanism like **password hashes for passwords (Cysa+ called this data protection)**
- **Ensuring Availability:** performing load and stress testing and designing app, may limit impact of DOS attack.
- **Monitoring and Logging:** should be enable, centralized.
- **Authentication:** only authenticate users or system.
- **Multifactor Auth:** limit impact of credential compromise.
- **Use Secure Session Management:** ensure that attacker can't hijack user session.
- **Cookies Management:** important of Web app that rely on cookie-based info.
- **Secure all network traffic:** Encryption of all traffic, prevent network-based attack.

Note: Cysa+: input validation, output coding, session management, authentication, data protection, parameterized queries. (all highlight in yellow color).

OWASP top proactive controls:

- **Define Security requirement**: doc security software need, how should be implemented.
- **Leverage Security Framework and Libraries:** use existing security tool.
- **Secure Database Access:** important data.
- **Encode and Escape Data:** ensure attackers can't embed code or special character, that can be executed.
- **Validate all input:** treat user input as untrusted.
- **Implement Digital Identity:** identity is core security layer (use multifactor auth, secure password storage and recovery and session handling.
- **Enforce Access Control:** require all requests go through access control checks, deny by default, apply least privilege.
- **Protect Data Everywhere:** encrypt in transit and at rest.

- ***Implement Security logging and monitoring****:* detect problems, allow investigate after the fact.
- **Handle all error and exception:** error should not provide sensitive data, and app should be tested to ensure it can handle problems carefully.

Top 25 software errors in 3 categories:

- **Insecure Interactive between component**: include issues SQL and OS command injection, file upload path issues, CSRF and XXS.
- **Risky Resource Management**: deal with buffer overflow, path traversal attack.
- **Porous Defense**: not using or misusing defensing technique, hard-core credential, missing authorization and authentication, use of unsalted hashes.

API Security:

API are interfaces between client and servers *over app and OS that define how client should ask for info from the server and how server will respond*.

- This Definition means that program written in any languages can implement API and make request.
- **API** useful for building interfaces between systems, also could be a vulnerability.
- **API** security *rely on authentication, authorization, proper data scoping to ensure that too much data isn't released, rate limiting, input filtering, and appropriate monitoring and logging to remain secure*.

Note: TAXII and **STIX** protocol and language, use "threat intelligence", example of interface might be accessed via API call.

Service-Oriented Architecture (SOA): a software design, provide services via communication protocols on a network.

- Intent of SOA: allow loosely coupled components to communicate in a standardized way, consume and provide data to other components.
- Typical component of a service-oriented architecture include *service provider, service registries or service brokers that provide listings and info about service providers and consumer who access the service*.

Note: SOA: provide service or component via communication protocol on a network. Ex. SOAP, REST, SAML.

SOAP (Simple Object Access Protocol): XML-based messaging protocol, used for web services.

- **SOAP**: defines how message should be formatted and exchanged, how transport of the messages occur, models for processing them.
- **SOAP** extensible, can be customized as needed.

RESTful HTTP (Representational State Transfer): has largely supplanted SOAP, many use case because of its greater flexibility.

- **REST APIs** follow 6 architecture constraints: use uniform interface, they separate client and server, they're stateless (don't use server-side session), they mark whether server response are cacheable, design to allow layering of services between client and server, and include client executable code in their response.
- **Both REST and SOAP:** allow developers to create their own APIs, but ***REST*** *is not a protocol, instead, defines how a RESTful architecture should be design.*

Application Testing: conducted in 1 of 4 ways as a *scanning tool*, via automated vulnerability scan, manual pen testing, via code review.

- **OWASP** code review can tell: common issue that app face: availability, business logic, compliance, privacy, vulnerabilities.
- **Combing code review with penetration**: can provide more insight (AKA 360 review).

Information Security and the SDLC: security need involve each process of SDLC:

1. During **feasibility phase**: security practice being asked to attend initial assessment or cost Evalu*

2. **Analysis and Requirement Definition Phase:** include security requirements and planning for requirement like authentication, data security, technical security needs.
3. **Security Artifacts**: created during **Design Phase** should include Security architecture doc, dataflow diagram.
4. **Development Implementation phase**: involve security testing of code, code review, development-centric security operation.
5. **Testing & Integration Phase:** include vulnerability testing & additional code review of the complete product.
    - Also occur when testing of a complete integrated solution can be conducted to ensure that no security issue shown up once integrated.
6. **Training and Transition Phase**: user training, part of security posture.
7. **Operation & Maintenance Activity**: require ongoing scans, patching, regression testing when upgrades occure.
8. **Disposition** of system and data that the app used when its life is over.

Version Control and Source Code Management

****SDLC** reaches the development phase, code starts to be generated. That means ability to control version of software or components that your team working on combine check-in/check-out functionality and revision histories, a necessary & powerful tool when developing software.
**Strong SDLC require ability to determine code that is being deployed or tested is the correct version, fixes were previously applied haven't been dropped from the release that is under development.

- Popular version: Git, Subversion and CVS (version control system)

Code Review Model: reviewing the code, is written for an app, also help detect issues while enforcing coding best practice & standards by exposing the code to review during its development cycle.

- Common code review process include, both formal & agile process like pair programming, over-the-shoulder, Fagan code review.

Pair Programming: (one Dev write code, other review. Then switch role.)
Agile software development technique, places 2 Dev at one workstation. One write code, another review. **Intend** to provide real-time code review, ensure multiple Dev familiar with code that's written.

- Drawback: add additional cost, require 2 full time Dev.

Over-the-shoulder: (one Dev wrote code to explain other Dev –Cheaper than Pair Programming)
This method rely on 2 Dev. One Dev wrote the code and then explain code to other Dev. **Intend**: allow peer review of code, also assist Dev in understanding how code works, without relative high cost.

Pass-Around Code Review: (One Dev wrote code, then send to others for review doc)
AKA: email pass-around code review. A form of manual peer review done by sending code to other Dev to check code for issues. **Intend:** allow more flexibility than over-the-shoulder review, don't provide same easy opportunity to learn about the code from Dev who wrote it.

Note: pair programming & over-the-shoulder offer making doc more important.

Tool-Assisted Review: (Use software to review code)
Rely on formal or informal software-based tool to conduct code review. Ex. Atlassian's Crusibible, Phabricator.
Formal code review are an in-depth, primary form of formal code review is ***Fagan Inspection.*

Fagan Inspection: form of structured, formal code review to find variety of issues during development process.

- ❖ Specify entry and exit criteria for process, ensure process is not started before appropriate diligence.

Fagan 6 typical phase:

1) Planning, include preparation of material, attendees & location.

2) Prepare the team by reviewing & assigning roles as coder, reader, reviewer, moderator.
3) Preparation which involves ***reviewing code or other item being inspected & doc or questio****
4) Identify defects based on notes from preparation phase.
5) Rework to resolve issues.
6) Follow up by moderate, ensure all issues identified, no new defects were created during resolution.

Space Intended for a pink note from page 332.

<u>Software Assessment: Testing and Analyzing Code</u>

Source code is basic level of every app contains variety bugs & flaws from programming syntax error to problems with business role, error handling, and integration with other services and systems.

- ❖ Testing & analyzing code can be done by 2 methods via static or dynamic code analysis along with testing mode like fuzzing, fault injection, mutation testing, stress testing.

****Static Code Analysis (AKA** source code analysis): conducted by reviewing source code for an app.

- It seen as type of "White-box testing" with full visibility to the testers.
- <u>Static doesn't run program, focus on understanding how program written & what code is intended to do</u>.
- Can be conducted using automated tools or manually review code AKA "code understanding".

****Dynamic Code Analysis:** rely on execution of the code while providing it with input to test the software (Can be automated).

<u>Below are testing methods that can be done with Dynamic and Static code Analysis</u>:

Fuzzing: involve sending invalid or random data to app, test its ability to handle unexpected data.

- Typically automated, useful for detecting input validation, logic issue, memory leaks & error handling.
- Identify only simple problems, not complex logic or business process issues.

Fault Injection: (modify source code, fuzzing, inject fault into running data).

→Directly insert faults into error handling path, error handling mechanism, are rarely used. Otherwise, missed during normal testing. **Fault Injection can be done 3 ways:**

- **Compile-time injection**: *<u>insert faults</u>* by *<u>modifying source code</u>* of app.
- **Protocol Software Fault Injection:** *<u>use fuzzing techniques to send unexpected or protocol</u>* noncompliant data to an app (*<u>use **fuzzing** technique</u>*).
- **Runtime Injection** of data into running program, either by *<u>inserting it into running memory</u>* of the program or by injecting the faults in a way that *<u>causes the program to deal with them</u>*.

Mutation Testing: (made small change to program, by alter version, test then reject if cause failure)

This method relating to **fuzzing and fault injection**, but rather than changing inputs to program or introducing faults to it, **mutation testing make small modification to program, alter version, or mutants, then tested and rejected if they cause failure.**

- Create common error as well replicate the type of errors that Dev might introduce during their normal programming process.

Stress Testing and Load Testing: when app ready to be tested, *stress test application and load testing tools* uses to simulate a full application load.

- **Stress test** can be conducted against individual components of app, they're capable of handling load conditions.
- **During integration and component testing**, *fault injection may also used to ensure* that problems during heavy load are properly handled.

Security Regression Testing (Test to ensure after changes, no new issues)

Regress testing focuses on changes, have been made do not create new issues.

- Security regression testing is performed ensure no new vulnerabilities, misconfigurations or other issues have been introduced.
- Web Application vulnerability scanners & other scanning tools, used as part of an automated or semiautomated regression testing process.

User Acceptance Testing (UAT): vital element in testing cycle.

- Once all functional and security testing are done, user is asked to validate whether it meets business needs.
- UAT should have test plan that involves examples of all common business processes. This should be paired with acceptance criteria that indicate what requirement must be satisfied.

**Manual Testing, testers use tools called "*Intercept Proxy"* allows them capture communication between a browser and web server. (Testers can also modified data that sent & receive).

Cryptographic Hardware (TPM & PUFs):

Hardware root of trust contains cryptographic keys, secure the boot process.

TPM is common in hardware trust of root. TPM chip built into many computers, provide 3 functions:

- Remote attestation, allow hardware & software be verified.
- Binding which encrypts data.
- *Sealing which encrypt data & set requirement for state of TPM before decrypting.*

****TPM** chip cannot be modified, and physically unclonable functions (PUFs). Unique to device, provide **digital fingerprint for device.**

Note: **PUFs**, based on unique feature of a microprocessor.

- Measured boot: security feature, help prevent boot-level malware.
  - Measured boot processes measure each component, starting with firmware, boot start drivers.
  - Data gathered is stored in TPM module, logs can be validated remotely to let security admin know the boot state of system.
- **HSM**: external device or plug-in card used to create, store, mange digital keys for cryptographic functions & authentication.

Firmware Security: UEFI provides ability to secure boot, load only drivers & OS loaders, have been signed using accepted digital signatures.

Bios, computer used to boot from UEFI. (UEFI a replacement for BIOS).

****Trusted Firmware Update** help validation done using methos like checksum validation, cryptographic signing.

- This technique used to validate updates for network devices, motherboards, phones, printers, other hardware.
- *Trusted Firmware, signed by a chip vendor*. Then, use access keys to help control access to hardware.

Hardware Security: Trusted Foundry Program to validate microelectronic supplier throughout supply chain. Program assesses the integrity and processes of companies, staff, distribution chains.

Chain of Custody: process used to *track movement & controls of asset through its life cycle* by doc.

- **Secure Processing:** refer to trusted execution environment. Boot processes include monitoring, privilege execution management. Can leverage or allow access to a trusted execution environment.
- **Processor Security Extension:** exist in CPU including ARM, Intel, AMD CPU. Provide security-related function implemented in CPU. (Memory allocated as secure memory).
- **Atomic Execution**: type of operation during processor both read & write location during same bus operation.
    - Prevent processor accessing or modifying location during operation, ensure integrity of operation.
- **Secure Enclave:** Apple mobile device. Provide cryptographic operation & user authentication, designed to remain secure even OS is compromised.
    - Run their own micro-kernel, their own secure bot process, allow secure processing to separate from rest of OS and CPU.
    - Secure Enclave on Apple Device generate encryption key at boot, pair it with userID to encrypt, validate, use secure enclave's portion of system memory.
    - Also handle FaceID, allow authen to be handled in secure partition.

Solution to Hardware Security Issues:

+SED+: encrypt data as it's written to disk (a hard disk has encrypted circuit built in)
+Trusted Firmware update+: help validation (motherboard, printer, network dvices)
+Measured boot+: measure each component.
+Bus Encryption+: use of encryption program instructed on data bus in computer, include secure crytoprocessor.
+Anti-tamper protection+: Tamper proofing microprocessor often take form encasing electronic.

Conclusion:

- **SLDC:** from planning and requirement gathering to design, coding, testing, training, deployment.
- **SDLC model:** linear Waterfall, Spiral's iterative process-based design, Agile: focus on sprints with timeboxed working sessions and greater flexibility,
- The **V model** with parallel testing cycles for each stage, **Bing Bang model**, without real planning or process.
- **Software coding best practices includes:** risk assessments, validating all user input to app, error handling, secure sessions, traffic and cookies.
- **Security Testing and Code review** improve app security and code quality.
    - Code Review: Par programming, over-the-shoulder, pass-around, tool-assisted.
    - **Fagan inspection:** remain primary, but time-intensive, solution.
    - May involve static or dynamic code analysis, fuzzing, fault injection, mutation testing, stress or load testing, regression testing.
- **Code Best Practice:** understand software to prevent security flaw.
    - **Version control**: help prevent issue that exist older code version from reappearing in new code.

Chapter 10 : Security Operations and Monitoring

Analyzing impact of an event require:

- Knowing if other events correlated with initial events.
- Understand system, users, services, assets were involved.
- Data classification for any data assets, part of the event.
- Other info may influence Org decision.

Vital: Analyst must determine immediate impact of event or incident is Vs total impact.

Logs: Security Analyst need know logs exists by defaults on systems, how to access them, how to find info of those logs, how to interpret that content.

- **Event Logs:** Windows event logs can be viewed, suing the “Event Viewer”.
- **Syslog Linux:** /var/log directory/auth.log

When analyze for **Proxy Logs** look for data:

- Target host IP, and what was request.
- Amount of content requested, may indicate compromised or match a known malicious package.
- HTTP request method, provides details of query string with **Get Request** (Post Request Carry this in body of message, requiring you read full payload).
- Usual user agents and protocols versions, useful for identifying application, malware, others.

IDS and IPS: both of these rely on rules to identify unwanted traffic. When a rule is triggered on IDS or IPS, the logs will contain info about rules was activated and info about traffic was captured and analyzed to trigger the rule.

- IDS and IPS analyze contents of packets & look at traffic across multiple packets or entire convo to perform their function.

Security Orchestration, Automation, and Response (SOAR): rely on stack of security tools to collect data from variety of security sources, then automatically respond. **SOAR** has 3 major components:

- **Threat & Vulnerability Management:** include threat management tools & vulnerability scanners as well, workflow, reporting, collaboration tools.
- **Security Incident Response:** IR suite manage incidents from start to finish.
- **Security Operation Automation:** orchestration and workflow tools as well as reporting, policy, process management tools.

****SOAR** deployment may ingest SIEM alerts and other data, then apply workflow & automation to them

***Analyst* needs to **consider what data may exist on endpoint device**, how to access it, whether that *data should be sent to central logs collection*.

Malware Analysis:

Reverse Engineer Malware: requires tool like disassemblers, debuggers, monitoring tools, unpackers, code and binary analysis tools, to determine what they do, who wrote them, detail of their functionality & construction. Tools that can be used for reverse engineer below:

- **Debugger:** allow u run program in controlled environment, modify variables, including adding stop points & monitoring what it is doing.
- **Disassembler**: used to convert machine code into assembly language whereas **Decompiler** attempts to convert machine code into high-level language like C or Java.
- **Unpacker and packer identifiers**: used to identify what packing & encryption being used to obfuscate program, then undo packing process.
- **System Monitoring Tools**: *impact to system like changes to filesystem*, *registry*, config changes

Heuristic Analysis: analyze suspected malware (can detect unknown malware).

- Heuristic tools starts by *building baseline of info (known-good behavior*) by monitoring normal behaviors over a period of time & then tracking differences from those behaviors.

****Linux ps command**: provide info about processes and their CPU memory utilization.

File Monitoring: Monitor filesystem help detect unauthorized or unexpected changes.

- Tool like Tripwire, OSSEC, and hosted intrusion detecting system (HIDS), used to monitor & report on filesystem changes.

User and Entity Behavior Analytics (UEBA): a type of *tool used to analyze normal user behavior & detect anomalous behavior*.

- Utilize *automated rule creation using machine learning* & statistical analysis techniques.

Cysa+ Note:

- **URL** (Uniform Resources Locators): point web browser.
- **Dynamically Generated Algorithms:** a subset of URL & Domain name analysis, AKA **domain generation algorithms:** used as part of malware packages to generate domain name from a known seeds.
    - Bot control can dynamically generate domain name knowing that bot use the same seed to know where to send their traffic.

DNS Fast Fluxing: *associating multiple IP addresses with a single domain name* and changing out these IP addresses rapidly.

- **Single Flux:** continuous register addresses, part of DNS address A record. When *combine with short time to live for the record*, round-robin DNS that point to different system as record is called.
- **Double Flux:** register & deregister DNS servers for DNS zones, adding another layer of confusion when attempting to pin down malicious system.

****Forward Email message**: will not include original headers. Forward email places the message into a new mail "Envelop", removing the header info that u may need to investigate it.

Sending Policy Framework (SPF): *break where email is forwarded* because forwarding sender will now be the sender and SPF checks may fail at new destination.

Note: Info can be lost when an email is forwarded, automatic email forwarding is a security concern that Org need to address.

- Automatic forwarding: used by attacker who successfully compromised an Acc to send all the emails received by that Acc to a destination of their choosing.

Digital Signature: *rely on digital cert and public key encryption* and can help prove the actual claimed sender was the real sender of the message.

- 1st: when email is digitally signed, its hash is created.
- 2nd: Then hash is encrypted with signer's private key to create digital signature.
- 3rd: The sender's digital cert and signature are attached to email.
- 4th: Recipient can validate the hash against the email they received and can also decrypt the signature using sender's public key.

Email Security Option (DKIM): help identify your domain.

Domain Key Identified Mail includes the sender policy framework (**SPF**), and Domain-based Message Authentication, Reporting and formance (**DMARC**).

- **DKIM:** is an email authentication, allow Org to add content to message to identify them as being from their domain.
    - Sign both of body of message & element of header, helping ensure message is actually from the Org it claims to be from.
    - It adds DKIM-signature header, can be checked against public key that is stored in public DNS entries for DKIM-enable org.
- **SPF:** an email authentication technique, allow Org to publish a list of their authorized email servers.
    - SPF records added to DNS info for your domain, they specify which systems are allowed to send email from that domain.
- **DMARC**: *use SPF and DKIM to determine if an email message is authenticated*.
    - **DMARC** records are published in DNS, unlike DKIM and SPF, DMARC used to determine if you should accept a message from a sender.
    - Using **DMARC**, you can choose to reject or quarantine message that are not sent by a DMARC-supporting sender.

****Trend Analysis:** consistent change period of time.

****DNS Blackhole:** use to spoof DNS server to prevent resolving hostname of specified URL.

**Grep -c: count numbers of occurrence.

**Grep -i: match lower and uppercase.

**Grep -n: show matching line and line numbers.

**Grep -v: show all lines, don’t match string.
**Grep -r: read all files under directories.
**Grep -e: allow multiple search pattern.

Chapter 11: Building an Incident Response Program

Security Incident:

- security event include observable occurrence that relates to security function.
    - User access file stored on a server, an admin changing permission on shared folders, attacker conducting port scan (These are all security incidents).
- Security incident is a violation or imminent threat of violation of computer security policies, acceptable use of policies, standard security practices.
    - Loss pf sensitive information, an intrusion, use of keylogger to steal password, DOS

Computer Security Incident response teams (CSIRT**): responsible for responding to security incidents that occurs to Org.

__Phase of Incident Response:__

Detect an incident→Analyze Data→Conduct a recovery→Close Incident

**Preparation→Detection & Analysis→Containment, Eradication & Recovery→Post-Incident Activities

I. **Preparation: CSIRT** require preparation to ensure that CSIRT has proper policy foundation, operation procedure, training.

→**This phase:** should assemble hardware, software, and info require to conduct incident investigation as below:

- Digital forensic workstation
- Backup devices
- Laptop for data collection, analysis & reporting
- Spare server & networking equipment
- Black removable media
- Portable printers
- Forensic and packet capture software
- Bootable USB media contain trusted copied of forensic tools
- Office supplies & evidence collection

II. **Detection & Analysis: NIST** describe security event indicators:

- ***Alert***: that original from intrusion detection & prevention systems, SIEM, antivirus software, file integrity checking
- ***Log***: generate by OS, services, App, network devices, and network flow.
- ***Public Available info:*** new vulnerability & exploits detected (0 day)
- ***People from inside:*** external resources who report suspicious activities.

****NIST** recommend following to improve effectiveness of incident analysis:

- **Profile network & system to measure characteristic of expected activity:** improve Org ability to identify abnormal activity during detection & analysis process.
- **Understand normal behavior of users, systems, networks, and app:** solid understanding of normal behavior is critical to recognize deviation from those patterns. (**Baseline)**
- **Create logging policy that specifies info must be logged:** policy should specify where those log records should be stored (preferably in SIEM).
- **Perform event correlation to combine info from multiple sources:** perform SIEM**.**
- **Sync clock across servers, workstations, network devices:** is done facilitated correlation pf log entries from different systems. Use **NTP server.**
- **Maintain Org knowledge Base that contain critical info about systems & App:**

- **Capture network traffic as soon as incident is suspected:** responders should immediately begin packet capture during detection & analysis.
- **Filter info to reduce clutter:**incident investigation generate massive info, impossible to interpret all without both inclusion & exclusion filters, CSIRT may wish to create predefined filters.
- **Seek Assistance from external resources**

III. **Containment, Eradication, and Recovery:** incident detection & analysis, *CSIRT engage in primarily passive activities designed to uncover & analyze info about incident*.

➔ Take active measures designed to contain effects, eradicates incident from network, recover normal operations.

➔ ***Containment, eradication, and recovery phase*** is design to achieve these objectives:

- Select containment strategy
- Implement selected containment strategy to limit damage
- Gather additional evidence needed to support response efforts & potential legal action.
- Identify attacker & attacking systems.
- Eradicate effects of incident & recover normal business operation.

Note: contain the damage→Eradicate the effect of incident→recover normal business operation

IV. **Post-incident Activity:** during this phase, teams conduct a lessons learned review and ensure they meet internal and external evidence retention requirements.

****NIST** recommended lesson learned process following:

- Exactly what happened & at what time.
- How well staff & management response to incidents.
- Document procedure were followed? Where they adequate?
- What info needed sooner?
- Any steps pr actions taken have inhibited the recovery?
- What would staff & management do differently next time?
- How could info sharing with other Org been improved?
- What corrective action can prevent similar incident in future?
- What precursor or indicator should be watched for in future to detect similar incident?
- What additional tools or resources needed to detect, analyze & mitigate future incident

****Civil litigation**: *when 2 party have conflicts, seek for money involving solving issues*.

**U.S Federal government must retain all incident-handling records for at least 3 years.

Building Foundation for Incident Response: Major responsibility that Org has during Preparation phase is building solid policy & procedure foundation for program.

Policy: should contain statement that provide authority for incident response, assign responsibility to CSIRT, describe the role of individual users & state Org priority.

- Policy is not a place to describe specific technology, response procedure or evident gathering technique.

NIST recommended incident response policies contain these key elements:

- Statement for management commitment.
- Purpose & objective of policy.
- Scope of policy to whom it applies & under what circumstance.
- Definition of cybersecurity incident & related terms.
- Org structure & definition of roles, responsibilities, level of authority.
- Prioritization or severity rating scheme of incidents.
- Performance measure for CSIRT.
- Reporting & contact forms.

Procedure and Playbooks

****Procedure**: provide detailed, tactical, info that CSIRT need responding to incident.

CSIRT develop playbooks: describe specific procedure, they will follow in event of specific type of cybersecurity incident. ***Playbooks** cover:*

- Breach of PII.
- Web server defacement.
- Phishing attack targeted at customers.
- Loss of laptop.
- General security incident not covered by another playbook.

CSIRT may include representation from following:

- **Technical subject** require during a response, this includes engineers, network administrators, database admin, desktop experts, and app.
- **IT Support:** needed carry out action directed by CSIRT.
- **Legal Counsel:** responsible for ensuring team's actions comply with legal, policy, and regulatory & advise team leaders on compliance.
- **Human Resources**: responsible for investigating potential employee malfeasance.
- **Public relation & marketing:** coordinate communication with media & general public.

Incident Response Providers: Org understand provider's guarantee response time (SLA) and ensure a plan in place to respond to early stages before provider assumes control.

CSIRT Scope of Control: incident response policy outline scope of CSIRT:

- What trigger activation of CSIRT? Who authorize activate CSIRT?
- CSIRT cover entire Org or only certain business units, info categories, or divisions?
- CSIRT authorize to communicate with *law enforcement, regulatory bodies, external parties*?
- Does CSIRT have internal communication and/or escalation responsibilities? If so, what trigger those requirement?

Coordination and Information Sharing: During incident response, CSIRT need to communicate both internal & external partners.

External Communications: **CSIRT**, business leaders, public relations teams & legal counsel bring table requirement may justify sharing limited or detailed info with external entities.

**Type of external communications may include:

- **Law Enforcement**: wish to involve incident appear to be criminal in nature. Org may choose to cooperate or decline participation in investigation.
- **Info Sharing & Analysis Center (ISAC):** provide community warning of cybersecurity risks.
- **Vendor:** maybe able to provide info crucial to the response. Hardware & software used within Org able to provide patches, troubleshoot, guidance crucial to response effort.
- **CSIRT:** may wish to coordinate incident response with other Org.
- **Communication with media** & general public may mandatory under regulatory or legislative reporting requirements, voluntary, forced by media coverage of security incident.

**Incumbent (overload) upon CSIRT leader to control & coordinate external communication in a manner that meets regulatory requirements & best serves response efforts.

***Classifying Incidents (**by threat & severity):* **CSIRT** classify incident by both type of threat & severity of incident according to a standardized incident severity rating system.

Threat Classification:

- **External/Removable Media:** attack executed from removable media. Ex. Malicious code spreading onto a system from infected USB.
- **Attrition:** attack that employs brute-force method to compromise, degrade, destroy system, network, service. Ex. DDOS attack, Cred brute force**.**
- **Web:** attack executed from a website or web-based app. Ex. XXS used to steal cred pr redirect to a site that exploits a browser vulnerability & install malware.
- **Email:** attack executed via email or attachment. Ex. A link to malicious website.
- **Impersonation:** attack involving replacement of Sth benign with Sth malicious. Ex. Spoofing, MITM, rogue wireless access, SQL injection.
- **Improper Usage**: Any incident resulting from violation of an Org acceptable usage policies by authorized user. Ex. User install file sharing software, loss of sensitive data, perform illegal activities on a system**.**
- **Loss or Theft Equipment:** Loss or theft of a computing device or media used. Ex. Laptop, smartphone, auth token.
- **Unknown:** attack of unknown origin**.**
- **Other:** attack of known origin that doesn't fit into any of previous categories.

APT attacker: highly skilled & talented attackers focused on specific objective, often funded by nation-states, organized crime.

  - APT known for taking advantages of Zero day.

Scope of Impact: scope of incident's impact depends on degree of impairment that it causes Org the effort required to recover from incident.

Functional Impact: *impact of incident, vary based on criticality of data, systems or processes affected by incident, Org's ability to continue providing services.*

Put a sticker on page 394

Economic Impact Categories:

- None: Org doesn't experience any financial impact
- Low: Org expect to experience financial impact of $10,000.
- Medium: Org experience financial impact more than $10,000, but less $500,000.
- High: Org experience financial impact of $500,000 or more.

Recovery Effort: measure time that service will be unavailable. Express as an amount of downtime experience or time required to recover form incident.

Datatypes: when security incident affect CIA. Analysts should assign a data impact rating.

NIST Recoverability effort categories:

- Regular: Time to recovery, predictable with existing resources.
- Supplemented: Time to recovery is predictable with additional resources.
- Extended: Time to recover is unpredictable; additional resources & outside help needed.
- Not Recoverable: All data encrypted.

NIST Information Impact Categories:

- None: No info was exfiltrated, changed, deleted, or compromised.
- Privacy breach: PII of taxpayer, beneficiaries, was accessed or exfiltrated.
- Proprietary breach: protected critical infrastructure info (PCII) was accessed or exfiltrated.
- Integrity loss: sensitive or proprietary info was changed or deleted.

Private Org information Impact:

- None: No info was exfiltrated or compromised.
- Regulated info breach: info regulated by external compliance was accessed or exfiltrated. May include PII, PHI under HIPAA, PCI DSS. GDPR, includes sensitive personal info (SPI) as defined under GDPR. **SPI** include info from special categories such as generic data, trade union membership, sexual info.
- Intellectual property breach: sensitive confidential property was accessed or exfiltrated. Include product development plans, formular, sensitive trade secret.
- Confidential info breach: corporate confidential info was accessed or exfiltrated, includes sensitive or classified.

Chapter 12: Analyzing Indicators of Compromise

****Capture Network-Related Events:** bandwidth for the network being used. 3 ways to check this:
Router-based monitoring active monitoring pr passive monitoring.

- ***Router-based monitoring:*** rely on router or switch with routing capabilities to provide info about flow of traffic. Capture traffic passing through devices, often referred as "network flow".
  →Number of tech exist to capture flows and other routers info, include:
    - **Netflow:** like sFlow, J-Flow, are standard for monitoring traffic flows.
- **RMON:** developed to monitor local area networks & operate at layers 1-4. Typically, operate in client/server model & use monitoring devices probes to gather data.
    - **Management info base (MIB):** provide monitoring groups with info about network & focus on flow-based including statistics, history, alarms, and events.
- **SNMP**: used to collect info from routers & other network devices & provide more info about devices instead of network traffic flow info provide by **RMON** or **Netflow**.

****RMON & NetFlow**: monitoring traffic flows.
****SNMP:** *collect info from router or other network devices* (provide info about device themselves).
Active Monitoring: Gather data (availability, routers, packet delay or loss & bandwidth).
Active monitoring reach out to remote systems & device to gather data.
****flows & SNMP** monitoring, where data is gathered by sending info to collectors.

- **Active monitor:** typically, data gathering locations. (may forward info to collector).

2 examples of active monitoring are:

- **Ping:** network data, actively using Internet Control Message Protocol (ICMP) to ping.
- **iPerf:** a tool measure maximum bandwidth, an IP handle public iPerf servers allow remote testing of link bandwidth in addition to internal bandwidth testing**.**
    - Help establish a baseline for performance to identify when network will reach useful limit.

Both active and router-based monitoring add traffic to the network. When bandwidth utilization appears, maybe lost or delayed as higher-priority traffic likely to be prioritized over monitoring data.
****Nagios** has plug-ins use both ICMP & TCP pings.
****Network tap**: system monitoring event on a local network.
Passive Monitoring: rely on capturing info about traffic pass a location on a network link.
Network monitor uses network tap to *send a copy all traffic sent between endpoints A and B*.

- Allow monitoring system to capture traffic is sent providing detailed view of *traffic's rate, protocol, content, detail of performance of sending & reviewing packet.*

- Passive monitoring: doesn't add additional traffic to network. Perform after-the-fact analysis, packets must be captured & analyzed rather than being record in real time.

Vital: Active and route-based monitoring: add traffic to network (mean: they may compete with traffic they are monitoring).

- Passive monitoring: not adding additional traffic.
- SNMP trap: are message sent by network device to SNMP management to indicate specific event or error.
  - Detect critical events or errors in real-time.

PRTG (Paessler Router Traffic Grapher): monitoring bandwidth usage. Provide server monitoring, network and bandwidth monitoring. Combine 4 type of monitoring accurate of bandwidth utilization:

- **Packet sniffing**: which monitor only headers of packets to determine what type of traffic is being sent.
- **Flows**: can send info about all connections or sampled dataset.
- **SNMP:** allow network device send info about important events as SNMP traps.
- **WMI**: provide interface, allow script & application access for automation of admin task, as well accessing management data for OS and provide report to tools like **System Center Operation Manager.**

Nagios: Nagios is a network & system log monitoring tool, and able to build & integrate with plug-in using Perl or executable app.

Cacti: (provide graphical view): open-source use SNMP polling to poll network devices for status info and provide graphical view of network & device status include using scripts with data stored in database.

Remember:

- **PRTG**: use flow data, shoe status info indication network bandwidth utilization has peaked.
- **Monitoring tool:** used to check high usage level, send alarm based pm threshold.
- **SNMP data:** monitor high load and other signs of bandwidth utilization at router or network device level.

Beaconing: (AKA heartbeat): activity sent to a C&C system as part of botnet or malware remote control system, typically sent as either HTTP or HTTPs traffic.

- Beaconing can request commands, provide status, download additional malware.
- Beaconing often encrypted & blends with other web traffic, can be difficult to identify, detecting beaconing behavior is critical part of detecting malware infection.
- Detecting beaconing often handled by using IDS or IPS.
- Using flow analysis ensure systems not sending unexpected traffic.
- Inspecting outbound traffic to ensure infecting systems are not resident in network.

Unexpected Traffic: many form: scan, probes, spike in traffic, direct attack traffic.

This can be detected by behavior-based detection capability. This rely on 3 major techniques:

1. **Baseline or anomaly-based detection**: *required knowledge of what normal.* Typically, gather during normal network operation. (Then, can set up threshold.)
2. **Heuristic or behavior-based detection:** using network security devices & defined rules for scans, sweep, attack traffic & other network issues.
3. **Protocol Analysis:** use a protocol analyzer to capture packet & check for problems.
   - Protocol analyzer help find unexpected traffic, like VPN traffic.
   - *Help identify common protocol being sent over an uncommon port*.

Detecting Scans and Probes: Scan, sweeps, and probes not significant threat to infrastructure by themselves, often precursors to more focused attack.

- Network scan easy to detect due to behaviors they include such as testing of service port, *connecting to many IP in network*, repeated requests to services that may not be active.

- Stealthy scan and probes scan can be harder to detect among general noise of network.

Detecting DOS and DDOS:

DOS: attempt overwhelm network, attack on specific service or system vulnerability, attack on intermediary system to prevent traffic from making between locations.

DDOS: come from many system or networks from many places. **Tool** like **low Orbit Ion Cannon (LOIC)**

Detecting DoS & DDOS Attacks: monitoring system could include:

- Performance monitoring using service performance monitoring tools.
- Connection monitoring using local system or app logs.
- Network bandwidth or system bandwidth monitoring.
- IDS or IPS with DoS and DDoS detection rules enabled.

Detecting Other Network Attackers: network-based attacks be detected using same technique:

- Use an IDS or IPS
- Monitor flows, SNMP for suspect behavior.
- Feed logs from firewall, routers, switches, central log analysis.
- Use SIEM and automatically alarm.

Detecting and Finding Rogue Device: Rogue device are devices should not be connected to the network.

Common method for identifying rogue device:

- **Valid MAC Address:** a list of known devices.
- **MAC address Vendor Info:** vendors use prefix for their devices.
- **Network Scanning**: nmap to identify new devices.
- **Traffic Analysis**: Identify irregular or unexpected behavior.
- **Site Surve**y: Physically reviewing devices at a site by manual checking wireless network on-site.

Wired Rogues: NAC are easy target for wired rogues devices:

- An employee or others trusted member connect device without permission.
- An attacker connect device to network.

**Preventing rogue device via port security or MAC whitelisting or NAC.

Wireless Rogue: can't always easily tracked to specific location. Track it down may involve signal strength measure & mapping the area where rogue is attempt to locate it.

**If wireless rogue is plugged into your network, using a port scan with OS that can turned on, often help locate device.

Processor Monitoring: Sudden spike or increased processor consumption in CPU usage indicate new software or a process being install or active.

High level of CPU usage points at DoS attack.

Memory Monitoring: OS level memory focused on memory utilization or memory consumption.

**Visibility into memory usage focus on consumption & process identification.

-->Most Org set memory monitoring levels for alarms & notification based on system memory usage approaching out-of-memory condition. (Monitoring threshole)

Note: Windows buffer overflow indicate insufficient memory allocation.

Memory Leaks: culprit in system crashes & outage, occur when program doesn't release memory after it is no longer needed.

- Overtime app with memory leak will consume more & more memory until app fails or OS runs out of available memory.
- Memory monitoring prevent memory leak from resulting a crash.

Drive Capacity Monitoring: focus on specific capacity levels & is intended to prevent drive or volume from filling up, causing outage.

- Tool: **Centralized monitor drive capacity** consumption are like *System Center Operation Manager (SCOM)* for Windows and *Nagios* for Linux.

Filesystem Changes and Anomalies: Monitoring in real-time for filesystem help catch attackers are occurring.

- Tool: like **Wazuh:** provide file integrity monitoring, open-source.
- Tripwired (commercial) and Open source.
- Advance Intrusion Detection Environment (AIDE)

System Resources Monitoring Tools:

- Resource monitoring (AKA *Resmon) easy* visibility into CPU, memory & disk and network utilization for a system.
- Performance monitoring (AKA *Perfmon): provide* more detail data from energy usage to disk and network activity.

Linux built-in tools check CPU, disk, and memory:

- Ps provide info about CPU & memory utilization, time was start and how long process it runs ,as well as command that started each process.
- Top: provide CPU utilization under CPU statistic
- df: display system's disk usage.
- W: indicate which accounts are logged in.

Detecting malware, malicious process & unauthorized software rely on 4 methods:

- *Central management tools:* Microsoft Intune can manage software installation..etc.
- Anti-virus and antimalware tools: design to detect potentially harmful software & files.
- Software & file blacklisting: use a list of disallowed software & files
- Application whitelisting

Note: Linux Netcat: allow you to create UDP and TCP connection using command *nc -l -p 37337 -e cmd.exe* which open a remote shell on port 37337, which connection to cmd.exe

Unauthorized Use & Detecting Mechanism

****Unauthorized Changes**: check system logs, app logs, monitoring tools.

*Unauthorized Privilege: *Check security event logs, app logs*.

Note: Use Run or Runonce registry key to make a program run when user log on.

Run Key make program run when user log.

****Systeminternal's AcessChk**: help validating access that specific user or group has to object like files, registry keys and services.

Registry Changes or Anomalies: Registry run can be found in:

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
- HKEY_CURRENT_USER\Software\ Microsoft\Windows\CurrentVersion\Run
- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce

Unauthorized Scheduled Tasks: Schedule tasks or cron jobs in Linux maintain persistent access to systems.

- Detect unexpected scheduled task in Linux by checking **cron.** Command: cat /etc/crontab, and /etc/cron
- Use cat /etc/crontab -l: listing cron job.

Application & Service Common Monitoring areas:

- Up/Down –is service running?
- Performance –respond quickly as expect?
- Transaction logging ---function of service is captured what action user take or action performed?
- Application or service logging –logs about function or status of service.

Application Logs: Many Linux logs are in var\log, Windows application logs gathers by Windows logging infrastructure or App specific directory or file.

Application & Service Anomaly Detection: Vary non-security related problem result in:

- App or Service-specific error include authen* error, service dependency or permission issue.
- Don't start on boot, either because of specific error, case of services, because service is disable.
- Service failure often causes by updates, patches or other changes.

Windows Service Status: can check either via service admin tool (service.msc) by using command-line tool like sc, service controller application**.**

- ➔ Linux: service –status-all.
- ➔ Linux system use init.d can be checked by running command like etc/init.d/servicename status.

Note: Linux service restart processes vary depending on distribution.

Application Behavior Analysis:

- Documentation of app's normal behavior (Baseline)
- Logging, to provide view of normal operations.
- Heuristic (behavioral) analysis using antimalware tools & other security-monitoring to flag behavior deviate from the norm.

Application & Service Issue Response & Restoration: app & service issues ranging from cred or right. Bad patches, component version to software flaw & actual attack.

Put a sticker to page 434

Detecting Attack on Application:

- **Anomalous activity**: doesn't match app's typical behavior, indicator of attack or compromise, log analysis, behavior baselines, filesystems integrity checking can help detect unexpected behavior.
- **Introduction of new Acc:** admin right often sign of compromise. App Acc creation not always logged in a central location. Admin controls match change management workflow & approvals to admin Acc creation.
- **Unexpected Output**: improper output or garbled data to errors. This can be challenging to detect using centralized methods for user-level app.
  - **Server-base app** provides file or API-level output, easier to check for errors based on validity checkers.
- **Unexpected Outbound communication:** Like beaconing, outbound file transfers, attacks common type of app exploit indicators.
  - Network monitoring like intrusion detection or prevention system monitoring outbound traffic is critical to detect detection these problems.
- **Severe interruption:** indicate a simple app requires a service or server restart, also indicate a security issue like DoS.
  - Monitoring tools monitor app or service status.
- Memory overflow: result in OS errors and crashes, making crash dump reporting important.
  - Monitoring for memory overflow limitation in memory handling for OS and App.

Conclusion:

- Incident Response requires visibility into network, system, services and app.
- **SNMP** used to gather device info and **RMON** used probes to gather statistical, historical, and event-based data.
- **Active monitoring**: *using ping & performance monitoring tools like iPerf to gather data by sending traffic travels.*

- **Passive monitoring**: doesn't add traffic to network, more useful for analysis than prevention of issue.
- **Network monitoring, PRTG, SolarWind & Cacti** *centralize multiple types of network data & provide both central visibility & analysis capabilities*.
- Service & app detected by monitoring like errors, failures, changes in service behavior.

Exam Essential:

- Flows, SNMP, active & passive monitoring all provide view of network health and usage.
- PRTG, Nagios, Cacti and Solar Wind help make large volume of data from diverse devices accessible & centrally visible.
- Scans and probs difficult to detect but can indicate interest by attackers.
- Wired rogues can be limited by using network admission (NAC), whereas wireless rogue required a monitoring and detection plan.
- Resmon & perform for Windows and ps, top, df, and w for Linux provide current state of system's resources.

Chapter 13: Performing Forensic Analysis and Techniques

Key Toolkit Components: basic set of items allow to perform forensic investigation:

- **Digital forensic workstation**: *designed to allow for data capture and analysis, multicore CPU and RAM.*
- FTK, EnCase, SANs Investigation Forensic Kit (SIFT), provide *ability to capture and analyze forensic images* as well track forensic investigations.
- **Write blockers** which *ensure drives connected to a forensic system or device cannot be written to*. Help ensure integrity of forensic investigation; *having file access time change*, having the system that is *analyzing data modify the content of files*,--can prevent forensic from being useful.
- **Forensic drive duplicators:** design to copy drivers for forensic investigation, then provide validation that original drive and content of new drive match.
- **Wiped drives and wiped removable media** of sufficient capacity to handle any drives system, large SATA, NAS Devices, Large SSDs easier to capture and transport multiple forensic images.
  - Removable media in form of large USB, Blu-ray or DVD media, flash media, valuable for transporting forensic data or for sending it to other Org.
- Camera to Doc system, drive labels. Camera important for forensic capture, speed up data recording and visual record of the stat of system or device.
- Labeling and Doc tools include label maker, label, indelible pens, help chain of custody & forensic process.
- Notebook & preparation doc and checklist to record forensic investigation processes and note.
  - Common type of form include chain-of-custody form that track who was om possession of evidence at any time.
  - Incident response form for tracking a response process, include response plan and incident forms, and escalation lists or call lists of people to contact during a response process.

Mobile Device Forensic Toolkit Component: Mobile devices forensic kit:

- Tools: for accessing SIM card & flash memory cards.
- Mobile device connection cable kit: include connector types.
- Mobile- device-specific forensic software tool.

**Forensic tool capability include: forensic software toolkit are imaging, analysis, hashing and validation, process & memory dump analysis, password cracking & log review.

Vital: File Carving (Endpoint disk & memory): hashing and data acquisition, also include legal holds & common procedure.

Imaging Media & Drives:

- 1st step in forensic investigation is to create copies of media or disks→this's done using imaging utility which can create a forensic image of a complete disk, a disk partition or logical volume.
- Forensic image match original source drive, volume, partition, or device include slack space & unallocated space.
  - Slack space: is the space left when file is written. This *unused space can contain fragment of files previously written to the space or even files that have been intentionally hidden*.
  - Unallocated space: is space hasn't been partition.

Analysis Utilities:u Forensic analysis utilities include:

- Timeline of system changes.
- Validation tools that check known-good versions of files against those found on system.
- Filesystem analysis capabilities can look at filesystem metadata.
- File carving tools: allow recovery of files without filesystem itself available.
- Windows registry analysis.
- Log file parsing and review. (can use SIEM).

Note: Open source like SIFT, CAINE, Autopsy: freely available. Commercial tool: ETK, EnCase.

Carving: scenarios where original filesystem cannot be used; file carving com in handy.

File Carving tool: ***look at data on block-by-block basic***, *looking for info like **file headers** & indicator of file structure*. When they find them, they **attempt to recover complete or even partial files**.

- 3 common type of file carving methods:
- Header-and footer-based carving: focus on header found in JPEG, JPEG can be found by looking at ***\xFF\xD8*** and ***\xFF\xD9*** in the footer.
- Content-based carving: look for info *content of a file such as character count & text recognition*.
- File Structure-based carving: require use of info about *structure of files*.

Chain-of-Custody Tracking: logging manner. **Maintaining chain-of-custody** ensure drive image & other data properly validated & available for review, thus reducing potential legal challenges based on poor custodial practices.

Legal Holds (Litigation): conducted when info be retained for a legal case.

- Forensic and backup tool leverage to ensure that a current copy of target system, drive, network storage is preserved & maintained as required by hold.

Hashing and Validation: forensic image format like EnCase's E01 format provide built-in hashing as part of file.

- To validate an image, **hash is generated for both original and copy**.
- **Hashing often used to validate binaries** and other app related files to detect changes to be binaries.

Operation System, Process, Memory Dump Analysis: often data kept encrypted is accessible in memory to process.

- OS analysis can provide key data about *what was occurring on system during timeframe targeted by investigation.*
- Memory dump analysis particularly valuable *when recovering decryption keys for full-disk encryption to products like BitLocker*.
- Hibernation files and crash dump can both *contain data needed to decrypt drive which makes accessing an unlocked machine*.

Memory Forensic: conducting memory forensic require either running live forensic analysis on a running machine or making a copy of live memory to point in time forensic memory analysis.

- **Tools: like Volatility**, open source memory forensic, can capture and analyze memory.
- **Volatility** has a wide range of plug-in command, include *ability to detect API hooks, read keyboard buffer, grab window clipboard, look for live TCP connection, scan for drive objects*.
- If there is data accessible in live memory in an unencrypted form, assume it can be recovered. If it's encrypted, the encrypted version can be accessed & potentially decrypted if key is available.
- **System Crash Dump**: can contain a copy of live memory.

Note: ElcomSoft's advance Office Password Recovery.

Using a GPU can result in massive speed increase over traditional CPU-based cracking if you ever need to perform a brute-force password.

Cryptographic Tools: common both protect forensic data & protect data.

Many malware package use tools called "**packers"** intended to protect them reverse engineering.

- **Packer** intended to make direct analysis of code difficult or impossible.

Log Viewer: log files provides info about system state, actions token on system, error or problems.

Forensic suites typically build in ***Log Viewer*** *can match log entries.*

TCP Dump and Wireshark: both are network capturing traffic.

Wireshark Network Forensic*:* Wireshark, open source, network protocol analyzer (AKA Packer Sniffer, or Sniffer): run on modern OS, allow users to capture & view network in GUI.

- Captures can be saved, analyzed and output in numbers of formats.

TCPDump Network Forensics (combine with Grep for Liux and Unix)

- Tcpdump command-line packet capture found on Linux and Unix. Powerful when use with "grep" to sort & analyze same packet data, you could capture with Wireshark.

Performing Cloud Service Forensic:

- Determine your contract says about investigation.
- Determine legal recourse you have with vendor.
- Identify data whether it's available via methods you or yours Org controls.
- Work with vendor to identify course of action if you don't control data.

Important: Virtualized system can be copied and moved to a safe environment for analysis.

Container Forensic:

Most containers are designed to be disposable if sth goes wrong, Org will have processes in place to shut down, destroy, & rebuild the container.

- Container may be destroyed or rescheduled to a difficult node, meaning forensic artifacts may be lost.
- Containerization tech create other challenges: internal lost & filesystem are ephemeral they communicate over software-defined network (SDN) that change frequently as container bought online, take offline or move.

Forensic Procedures: All investigation, you will take these step:

1. Determine what you are trying to find out. This form problem statement help define what forensic activities you will take.
2. Outline location & type of data would help answer questions from Step 1.
   - Data may exist in many forms, app and system can determine format & accessibility for the data
   - Knowing where and how u need collect data will influence forensic process.
   - Need to know the rest of data & system, u need to capture data from.

3. Doc & review your plan,
4. Acquire & preserve evidence. Acquisition process may require cloning media seizing system or devices or making live memory images to ensure that info not lost when system is powered off.
5. Perform initial analysis, carefully tracking your actions, system and data you work with , your finding, and any question you need to answer.
6. Use initial analysis to guide further work include deeper investigation & review where initial analysis pointed to additional data or where info missing that needed to answer you original asked.
7. Report on findings of investigation

→**Acquisition process** need take into account of volatility which measure how easily data is to lose.

→Data store in memory or cache is considered highly volatile since it'll be lost if the system is turned off whereas data stored in printed form or as a backup is considered much less volatile.

Forensic App of Windows System Artifact:

- **Windows registry**: info about files & services, location of deleted file, evidence of app being run
- **Autorun keys:** program set to run at startup (often associated with malware or compromise).
- **Master File Table** (MFT): detail of inactive/removed records.
- **Event Logs:** logins, service start/stop, evidence, app being run.
- **INDX Files & changes logs:** evidence of deleted file, MAC timestamps.
- **Volume Shadow Copy**: *point-in-time info from prior action*.
- ***Hibernation Files & memory dump****: memory artifact of command run*.
- **Removable drive**: system logs may indicate drives were plugged in data.

Acquiring & Validating Drive Image: Imaging utility such as write blocker are employed to prevent possibility of modifying source drive & multiple copies are made so that original drive can be retained for evidence.

→Forensic Copies: need to use an imaging tool to create forensic images rather than using the copy command.

Important of bit-by-bit copies: one reason that copy isn't done using command is to ensure slack space & unallocated space are both copied as part of image.

- Captured deleted files that haven't yet been overwritten, fragments of older files in space that wasn't written by new files, data that was stored on a drive before it was partitioned.
- Slack & unallocated space can provide rich detail about history of system, simply copying new files will not provide that visibility.

Imaging with dd: **Linux** dd utility used to clone drive in RAW format, bit-by-bit format.

- Block size set using **bs flag** & defined in bytes. Ex. Set bs flag to 64k.
- Operator "if" set input file. Ex. If = /dev/disk/sda1
- Operator "of" sets the output file. Ex. Of = /mnt/usb/

Note: critical verify input and output location for dd command. List drive, use *fdisk -l or lsblk.*

- dd: create file in RAW, bit-by-bit format.
- EN01, EnCase file format.
- OVF: virtual file format.
- Core file & hibernation file both contain image of live memory of system, allow encrypt key to be retrieved from store file.
- Lime and fmem: Linux forensic tool.

Note: Windows Crash dump: store in %system%\memory.DMP

Complete Chain of custody: doc what is collected, why collected or analyzed the data, when each action occurred, when device & other evidence were transferred, handled, accessed & securely stored. (You may need a 3rd party to validate this process).

User Write Blockers: forensic investigation image acquisition. (write blocker not modify image of drives).

- **Hardware write blocker:** prevent writes from occurring while drive is connected through them. (can be certified to NIST).
- **Software write blocker: t**ypically, less popular than hardware write blocker.

Verifying Images: when investigation use **dd** or other manual imaging tools, md5sum or sha1sum hashing utility are frequently used to validate images.

- Each time, u generate an image, u should record hash or verification info for both original & cloned copy, that info should be recorded in your forensic logbook or chain of custody.

EnCase and FTK (Memory Capture).

Dd (Linux, used to clone drive).

Imaging Live Systems: Live imaging may not obtain some desirable data:

- Live imaging can leave remnants (small quantity of sth left) *due to utility being mounted from removable drive or installed.*
- Content of a drive or memory may change during imaging process.
- Malware or other software maybe able detect imaging tool & could take action to avoid or disable it.
- Live image not include unallocated space.

Acquiring other data: beyond drive image include log data, USB device histories, app data, browser cache & history, email, user-generated files.

Acquiring and Reviewing Log Data: Log data often stored remotely & may not accurate in case of compromised machine.

**Investigation may involve actions, are logged centrally or on network devices, but not on single local system or device that you're likely to create forensic image of, preserving logs is important:

- Determine where logs reside & what format are stored in.
- *Determine time period, need to preserve. Remember to obtain logs from longer period incase u find out that an issue or compromise started before u initially suspected*.
- Obtain a copy of logs & doc how logs were obtained. Checksums or other validation is often appropriate.
- Identify items of interest, this might include UserID, EventID, timeframes, other elements identified in your scope.
- Use log analysis tools like Spunk, Event logs analyzed, text editor to search and review logs.

Viewing USB Device History: Windows tracks history of USB devices. Provide system name, device name, its serial number, time of use, vendor ID, type of device.

Capturing Memory-Resident Data: shutting down system result loss of data stored in memory. **Info in browser cache or program states will be lost. **Tool to capture memory:**

- **Fmem & LiME:** Linux Kernel, allow access to physical memory. Design to be used with **dd**. **LiME** directly copied data to a designated path & file.
- **DumpIt,** windows memory capture too, simple copied a system's physical memory to folder where DumpIt program is. Easy capture to a USB thumb drive.
- Both **EnCase & FTK**, built-in memory capture & analysis capability. (For Windows).

Using Core Dumps & Hibernation Files: Memory image, core dump and crush dump files. Since they contain contents of live memory include memory-resident encryption keys, malware that runs only in memory, other items not typically stored to the disk.

→**Windows crash dump file**: can be found by checking setting found under control Panel> System & Security > System > Advanced System Setting > Startup & Recovery > Settings.

- Typically, crash dump files located in system root directory: %systemroot%\memory.DMP
- Windows Memory dump: can be analyzed using **WinDbg.**

Acquisition from Mobile Devices: Mobile device forensic starts with disabling device's network (1st step), then ensure access to device is possible by disabling passcode (2nd step). Physical acquisition of SIM card, media cards, device backup occurs (3rd step). Device is imaged (4th step).

- ➔ 4 primary mode of data acquisition from mobile device:
  - Physical: acquisition of SIM card, memory card, backups.
  - Logical: create image of logical storage volume.
  - Reviewing content of live, unlocked phone & taking pictures & note about what is found. (for chain of custody).
  - Filesystem, *provide details of deleted files as well existing files & directories.*

Forensic Investigation: An example: require for forensic image & will perform analysis include:

- Import data into FTK include indexing & case management.
- Evidence of data leakage.
- Email communication with 3rd parties about files.
- Evidence of app installs.
- Evidence of filesystem changes, renaming files.

Importing a Forensic Image: Forensic Image made a copy to use in your investigation, import it into your forensic tools.

→Once your image being imported to a case & properly logged, image then indexed & analyzed, include identifying file types, searching slack & unallocated space, building an index of file timestamps.

Analyzing the Image: since this is data leakage case, internet browser history & emails are likely to be of particular interest.

****FTK** also index & display deleted files, allowing u see that CCleaner, a cleanup program that removed browser history & cache & wipe other info useful for forensic.

****Eraser**, a file wiping utility, appear to have been partially deleted but left a remnant directory in Program Files Folder.

Reporting: Final stage of forensic investigation is preparing & presenting a report include 3 major component: *goals & scope, the target of target of forensic activities including all system, devices, media, complete list of finding and result.*

Goal of Investigation: *include initial scope statement, person, or Org that asked for investigation*. Example, John smith, Directory of HR, requested that we review Ally Matt's workstation, email, system she administer to ensure data was recently leaked to a competitor wasn't sent from her email acc or workstation.

Targets: report you create should include: list of all devices, systems & media was captured & analyzed.

- Target should be all listed in tracking notes & chain of custody form.

Finding and Analysis: Finding should list what was discovered, how it was discovered, why it is important.

Exam Essential:

- Hashing & validation prove forensic images match the original.
- Forensic tool include: analysis utility, can provide timelines, file validation, filesystem analysis for changes, deletion, other log files viewing.
- Key data acquisition ability include: dead or offline system, cloning & validating via hashing, ability to identify changes to binaries & other files, file system carving, chain of custody and activity Loggins, live system imaging.

- Identify locations of relevant data, planning, acquisition planning, acquisition, analysis and reporting.
- Target include system info: file medication, access & change details, user artifacts and stored data like memory dumps, shadow copies, Recycle bin contents.

Chapter 14: Containment, Eradication, and Recovery

****Containment** designed to isolate incident & prevent from spreading further before eradicating the damage or recovering data.

****Segmentation:** often use network segmentation as a proactive strategy to prevent spread of future security incidents.

****Isolation**: 2 primary isolation techniques: isolating affected system or isolating the attacker.

- **Isolating Affected System:** affected systems complete disconnected from remainder of the network. Objective is to continue allowing attacker to access isolated systems, but restrict their ability to other systems & cause further damages.

****Removal**: removal differs from segmentation and isolation in affected system are completely disconnected from other network, although may still allow to communicate with other compromised systems within quarantine VLAN.

Evidence Gathering and Handling: primary objective during containment is to limit damage to the Org & its resources.

- Gathering evidence during containment can be crucial in the continuing analysis of incident for internal purposes, it can be used during legal proceeding against attackers.

NIST recommended maintain detailed evidence log:

- Identifying info (location, serial number, model number, hostname, MAC address, IP).
- Name, title, phone number of each individual who collect or handled evidence during inve*
- Time & date (time zone) of each occurrence of evidence handling.
- Location where evidence was stored.

Incident Eradication and Recovery:

Once successfully contain an incident, time to move on to ***eradicate*** phase of response. Primary purposes of eradication is to remove any artifacts of incident may remain on Org network.

- This could include removal of malicious code from network, sanitization of compromise media, securing of compromised user accounts.

Recovery phase focus on restoring normal capabilities & services include reconstituting resources & correcting security control deficiencies.

- could include rebuilding & patching systems, reconfiguring firewall, updating malware signatures.
- The goal of recovery is not just rebuilding Org network, also reduce likelihood of successful future attack.

→During eradication & recovery effort should develop clear understanding of incident's root cause.

- Root cause assessment, critical component of incident recover.
- Root cause analysis help Org identify other systems they operate might share same vulnerability

Sanitization and Secure Disposal: During recovery effort need to dispose of or repurpose media from systems that were compromising during incident.

→Secure disposition of media: clear, purge and destroy.

NIST defines 3 activities in NIST SP 800-88 for *Media Sanitization:*

- 1st: **Clear**: applied logical technique to sanitize data in all user-addressable storage location for protecting against simple noninvasive data recovery techniques.
    - Typically, applied through Read & Write command to storage device by rewriting with new value.

- 2nd: **Purge:** applied physical or logical technique that render target data recovery using state-of-the-art laboratory technique.
    - Overwriting, block erase, and cryptography erase activities.
    - **Degaussing:** form of purging that use strong magnetic fields to disrupt data stored on a device.
- 3rd: **Destroy:** render target data recovery subsequent inability to use media for storage of data.
    - **Destruction techniques:** include disintegration, pulverization, melting, incinerating.

Vital: Physical incinerating a hard drive, remove any possibility that data will be recovered, but require use of an incinerator and renders the drive unusable for future purposes.

Validating the Recovery Effort: 4 activities always include these validating efforts:

- *Validate only authorized user Acc exist on every system & app in Org.*
- *Verify proper restoration of permission assigned to each Acc:* Acc review, verify that Acc don't have excessive permission that violate principal of least privilege.
- *Verify that all system is login properly:* each system & app be configured to log security-related is consistent in Org logging policy.
- *Conduct vulnerability scan on all system: Vulnerable scans plays vital role, verify system, are safeguard against future attack. Run scan against system & initiate remediation workflow.*

Conducting Lesson Learnt: team should clearly identify any new IOC & make recommendation for updating the Org security monitoring program to include these IOC.

Developing a Final Report: formal written report creates an institutional memory of incident, useful for training new security team member.

- It may server record of legal action.
- Creating final report, help identify previously undetected deficiencies that may feed back through the lesson learned process.

→CSIRT should cover post incident report include:

- *Chronology events of incident & response efforts.*
- *Root cause of incident.*
- *Location & description evidence collected during incident response.*
- *Specific action taken by responder to contain, eradicate, and recover form incident.*
- *Estimate of impact.*
- *Result of post recovery validation efforts.*
- *Doc of issue identified during lesson learned.*

**Org should have defined retention period for incident reports & destroy old reports when exceed that period.

Summary:

- CSIRT: identify incident, then move to containment, eradication, and recovery phase. 1st: Priority contain the damage caused by severity incident to lower the impact of Org. Once incident is contained, responder take action eradicate effects of incident & recovery normal operation.
    - Move to post incident phase, lesson learn session & written report.
- Purpose of containment activity: Identify, contain the damage. Also include network segmentation, isolation, removal of affected systems.
- Important of collecting evidence: Much of evidence is volatile, may not be available later if not collected during incident response.
    - CSIRT must determine priority evidence collection, properly handle, used in legal proceeding.

- Purpose of eradication & recovery: after containing damage, responder should move on to eradicate & recovery that seek to remove all trace of incident and restore normal operations.
    - Should include validation efforts that verify security controls are properly implemented before closing the incident.

Chapter 15: Risk Management

Analyzing Risk: enterprise risk management (ERM): approach to risk analysis that begin with identifying risks, continue determining severity of each risk then result in adopting one or more *risk management strategies* to address each risk.

- **Threat:** possible events, have adverse impact on CIA.
- **Vulnerability:** weakness could be exploited by a threat.
- **Risk**: likelihood that a particular threat will exploit a particular vulnerability, resulting in harm or damage.

Note: A threat without a corresponding vulnerability doesn't pose a risk, nor does a vulnerability without a corresponding theat.

→The greater the threat and vulnerability, the greater the risk.

Risk Identification: *risk identification process require identifying threats & vulnerabilities that exists in you environment*.

Risk Calculation: evaluate any risk, using 2 different factors:

- Probability or likelihood risk will occur.
- Magnitude or impact that risk will have on Org.

Note: 2 factors contribute to degree of a risk: probability & magnitude.

Risk severity = Probability * Magnitude

→combining magnitude & impact determine severity of risk.

Business Impact Analysis: (BIA): approach risk prioritization:

1st: *Quantitative Assessment*: user numeric data in analysis, allow straightforward prioritization of risks.

2nd: *Qualitative Assessment*: substitute subjective judgements & categories for strict numerical analysis, allowing assessment of risks are difficult to quantify.

Quantitative Risk Assessment:

1. **Determine asset value** (AV) that affected by risk: This AV is expressed in $ and determined using the cost to acquire the asset.
2. **Determine likelihood the risk will occur**. Risk will occur in a given year, expressed in number of times the risk is expected each year. *Annualized rate of occurrence* **(ARO).**
3. Determine amount of damage: occur to the asset, known **as** *exposure factor* (**EF**), express as percentage.
4. **Calculate** *single Loss of Expectancy* (**SLE)**: it is financial damage expected each time as risk materialized. **SLE = AV x EF.**
5. Calculate annualized loss expectancy **(ALE):** is amount of damage expected from a risk each year. **ALE = SLE x ARO.**

Note: Quantitative technique: works well for evaluating financial risks, expressed in numeric terms.

- Qualitative risk assessment technique: seek to overcome limitation of quantitative technique by substituting subjective judgement for objective data.
- Qualitative technique use probability & magnitude to evaluate severity of risk, but do so using subjective categories.

Managing Risks: Risk management process, addressing risks facing Org. Risk Assessment servers 2 vital roles in risk management.

- Risk assessment provides guidance in prioritization risks with highest probability & magnitude are addressed first.

- Quantitative risk assessment determines whether potential impact of a risk justify cost of incurred by adopting risk management approach.

Vital: 4 strategies to choose: risk mitigation, risk avoidance, risk transfer, risk acceptance.

Risk Mitigation: process of applying security control to reduce probability and/or magnitude or a risk.

- Mitigate risk through design, implementation & management of security controls.

Risk Avoidance: where change your business practices completely eliminate potential that a risk will materialize.

Risk Transfer: shift risks from Org to another entity.

Risk Acceptance: deliberately choosing to take no other risk management strategy & simply continue operations as normal in face of the risk.

Data Classifications:

1st: **Top Secret:** require highest protection. Unauthorized disclosure of **Top Secret** info grave damage to national security.

2nd: **Secret**: info requires substantial degree of protection. Unauthorized disclosure of **Secret** info cause serious damage to national security.

3rd: **Confidential**: requires some protection. Unauthorized disclosure of **confidential** info expected to cause identifiable damage to national security.

4th: **Unclassified:** does not meet classification under other categories.

****Business term:** highly sensitive, Sensitive, Internal, Public.

Data Sovereignty: both EU & U.S laws would apply and that could cause serious issue for company.

- Before deploying new services, determine where data will be stored.
- Ask cloud provider to specify locations where data will be stored in writing.
- Use encryption, If a foreign country government demand cloud provider give them access to your data, they won't be able to read it if you hold the decryption key.

Nondisclosure Agreements (NDA); prohibit from sharing info with unauthorized individuals.

Training & Exercise:

- **Red Team**: attackers attempt to gain access to system.
- **Blue Team**: defender secure system & network. Monitor environment during exercise, conducting active defense technique**.**
- **White team:** observe & judge. Serve as referee oversee the rules & watch the exercise to douc lessons learned from the tests.

Tabletop Exercise: gather participants in same room to walk through their response to a fictious exercise scenario.

Data Loss Prevention: works in 2 different environments:

- Host-based DLP (Trend Micro at LHCC)
- Network-based DLP: monitor outbound, watching for unencrypted transmission.
  - Use software agents installed on systems that search presence of sensitive info.
  - Monitor system config & user actions, blocking undesirable actions.
  - They are dedicated devices that sit on the network & monitor outbound network traffic watching for any transmission that contains unencrypted sensitive info.
  - Can block those transmissions, preventing unsecured loss of sensitive info.

Data Loss Prevention: block traffic that violate Org policy, may automatically apply encryption to the content. This automation encryption often focus on email:

- **Pattern machine:** they watch sign of sensitive info. Ex. Credit card or SSN, Top Secret, Business confidential.
- **Watermarking:** where system or admin tags to sensitive doc then DLP can monitor systems & network for unencrypted content containing those tags.

    - Watermarking commonly used in *digital rights management (DRM) that enforce copyright & data ownership restriction.*

Data Minimization: deidentification process removes ability to link data back to an individual, reducing its sensitivity.

- An alternative to deidentifying data is transforming it into a format where original info can't be retrieved (AKA "data obfuscation") and can use tools to assist:
    - **Hashing**: apply strong hash to a data element, replace value in our file with hash value.
    - **Tokenization**: replace sensitive value with a unique identifier using lookup table. (need to keep lookup table secure).
    - **Masking:** redacts sensitive info by replacing some or all of sensitive fields with blank characters. Ex. Might replace all, but last 4 digits or a credit card # with X's or *'s to render the card number unreadable.

Chapter 16: Policy and Compliance:

Framework generally include 4 types of doc:

- Policies
- Standards
- Procedure
- Guidelines

Polices: high level statement or management intent. Compliance with policies mandatory. (Require CEO to create a policy).

- Statement the important of cybersecurity to the Org.
- Requirement that all staff take measures to protect CIA.
- Statement on ownership of info created and/or possessed by Org.
- Designation of chief info security officers (CISO)
- Delegation of authority granting CISO the ability to create standards, procedures & guidelines that implement policy.

*****Org commonly include doc in their info security policy library*****: Important: Must remember.

- **Info security policy**: provide high-level authority & guidance for security program.
- ***Acceptance Use Policy:*** provide network & system users with clear direction on permissible users of info resources.
- ***Data Ownership policy***: clearly states ownership of info created or sued by Org.
- ***Data Classification policy:*** describe classification structure used by Org and process used to properly assign classification to data.
- ***Data Retention Policy:*** outline what info Org will maintain & length of time, will be retained prior to destruction.
- ***Account Management Policy:*** describe account life cycle from provisioning through active use and decommissioning.
- ***Password Policy:*** sets 4 requirement for password length, complexity, reuse, & similar issue
- ***Continuous Monitoring policy:*** describe Org to monitoring & inform employee that their activities is subject to monitor in workspace.
- ***Code of conduct/ethics:*** describe expected behavior of employees & serves as a backstop.

Note: Standard is approved at lower level than policy.

Standard: provide mandatory requirement describing how Org carry out its info security policy, include specific config setting used for a common OS, control that must be out in place for highly sensitive info, other security objectives.

- *Standard is lower than policy. Thus, may change more regularly*.

Policy: sets out high level objective of security program & require compliance with standard which include detail of required security controls.
Guideline: provide advice to Org seeking to comply with policy & standard.

→Failure to follow industry best practices may be seen as negligence & can cause legal liability for Org.

Procedure: detailed, step-by-step processes that individuals & Org must follow in specific circumstance.

- Procedure ensures persistent process for achieving a security objective.
- Org may create procedures for building new systems, releasing code to production, responding to security incidents.

****Org commonly include following procedures in their policy frameworks****:

- ***Monitoring procedure***: describe how Org perform security monitoring, include possible use of continuous monitoring technology.
- ***Evidence production procedure***: describe how Org respond to subpoenas, count order, legitimate requests to produce digital evidence**.**
- ***Patching Procedure***: describe frequency & process of applying patches to app & system under Org care.

Guidelines: provides best practices & recommendations related to given concept, technology or task.

- Compliance with guidelines not mandatory & guidelines are offered in spirit of providing helpful advice.

Exceptions & Compensating Controls: adopting new security policies, standard, and procedure, Org should provide for exception to those rules.

Exception processes require use of ***compensating control* to mitigate risk associated with exception to security standards.

3 criteria must meet for compensating control to be satisfactory:

1. Control must meet intent, rigor original doc.
2. Control must provide similar level of defense as original requirement, such compensating control sufficiently offsets the risk original PCI DSS requirement was designed to defend against.
3. Control must be "above & beyond" other PCI DSS requirements.

*****Compensating control*** find alternative means to achieve an objective when Org cannot meet original control requirements.

- This control balance the fact that it's simply isn't possible to implement every requirement security control with desire to manage risk to greatest feasible degree.

Complying with Laws and Regulations:

Some major info security regulation facing U.S Org:

- Health Insurance Portability & accountability Act (HIPPA): affect healthcare provider, health insurer.
- Payment Card Industry Data Security Standards (PCI DSS): detailed rule about storage, proceeding & transmission of credit & debit card info.
- Gramm-Leach-Bliley Act (GLBA): cover financial institutions, broadly defined it required those institutions have formal security program & designate individual overall responsibility for that program.
- Sarbanes-Oxley (SOX) Act: applies to financial records of public traded companies & requires those companies have strong degree of insurance around IT systems that store & process those records.
- Family Educational Rights & Privacy Act (FERPA): requires that educational institutions implement security & privacy for student educational records.

- Data breach notification laws: describe requirement that individual states place on Org that suffer data breaches regarding notification of individuals affected by breach.

NIST Cybersecurity Framework: National Institute Standards & Technology responsible for developing cybersecurity standards across the U.S federal government.

**NIST release cybersecurity Framework designed to assist Org to meet one or more 5 objectives:

- Describe that current cybersecurity posture.
- Describe their target state cybersecurity.
- Identify & prioritize opportunities for improvement within context of continuous & repeatable process.
- Assess progress tpoward targeted state.
- Communicate among internal & external stakeholders about cybersecurity risk.

NIST framework include 3 components:

- Framework core: set of 5 security functions: *identify, protect, respond, detect and recover*.
- Framework implementation: *maturity model that describe current & desired positioning of Org along continuum of progress*.
- *Separate profile to describe its desired future state*.

ISO 27001: International Org for Standardization (ISO) published ISO 27001, standard document title.

Space for 14 category.

Control Objective for information & Related Technologies (COBIT): a set of best practices for IT governance developed by Info System Audit & Control Association (ISACA):

- Plan and Organize
- Acquire & Implement
- Deliver & Support
- Monitor & Evaluate

**COBIT address each domain, provide 5 COBIT frameworks:

1. COBIT Framework
2. Process Description
3. Control Objectives
4. Management Guidelines
5. Maturity Models

Information Technology Infrastructure Library (ITIL): offer comprehensive approach to IT service management ITSM cover 5 activities:

- Service Strategy
- Service Design
- Service Transition

- Service Operation
- Continual Service Improvement

Implementing Policy-Based Controls: *control objectives* are statement of desired security state, but they do not carry out security activities.

**Security control are specific measures that fulfill security objectives of Org.

Security Control Strategy: 3 different security controls:

- *Technical Control*: enforce CIA. Ex. Include firewall rules, access control list, IPS, encryption.
- Operational Control: processes we put in place to manage technology in secure manner. Include user access review, log monitoring, and vulnerability management.
- *Managerial control:* procedural mechanism focus on mechanic of risk management process. Include periodic risk assessment, security planning exercises, security changing management, service acquisition, project management practice.

Security Control Types: type of security controls:

- *Preventive control:* stop security issue before it occurs. Ex. Firewall, encryption.
- Detective Control: identify events have already occurred. Ex. IDS.
- *Corrective Control:* remediate security issue have already occurred. Ex. Restoring backup after ransomware attack.
- *Deterrent control:* prevent attacker attempting violate security policy. Ex. Vicious guard dog, barbed wire fences.
- *Physical Control:* security controls that impact physical world. Ex. Fences, perimeter lighting, locks, fire suppression.
- *Compensating control:* designed to mitigate risk associated with exception made to as security policy.

Security Control Verification & Quality Control:

- Quality control procedure: *verify Org has sufficient security control in place*.

**Every security program include procedure for conducting regular internal tests.

There are 2 evaluations: audits and assessments.

- Audits: formal review of Org security program or specific compliance conducted on behalf of 3rd party.
  - Require rigorous, formal testing of controls & result in formal statement from auditor regarding entity's compliance.
- Assessments: less formal review of security controls, typically requested by security Org in effort to engage in process improvements.
  - Typically, gather info by interviewing employees.