CompTIA Security+

ALL IN ONE EXAM GUIDE

9TH EDITION SYO-701

Chapters 1- 5

Summary Compiled by:

Eman Elshimy

Table of Contents

CHAPTER 1: Today's Security Professional

1. CompTIA Security+ Exam Objectives Covered

- Domain 1.0: General Security Concepts: Compare and contrast various types of security controls. Summarize fundamental security concepts. Explain the importance of using appropriate cryptographic solutions
- Domain 3.0: Security Architecture: Compare and contrast concepts and strategies to protect data:
- Domain 5.0: Security Program Management and Oversight: Explain elements of the risk management process

2. Cybersecurity Objectives

- Core Objectives (CIA Triad):
  - Confidentiality: Prevents unauthorized access to sensitive information. Achieved through controls like firewalls, access control lists, and encryption.
  - Integrity: Ensures no unauthorized or accidental modifications to information or systems. Achieved through controls like hashing and integrity monitoring solutions. Threats include malicious alteration or non-malicious corruption.
  - Availability: Ensures legitimate users can access information and systems when needed. Achieved through controls like fault tolerance, clustering, and backups. Threats include malicious disruption or non-malicious incidents (e.g. natural disasters).
- Non-repudiation: Ensures that an individual who performed an action (e.g. sending a message) cannot later deny having performed it.
  - Digital signatures are a common example of a control providing non-repudiation.

3. Data Breach Risks

- Security incidents are breaches of confidentiality, integrity, and/or availability of information or systems.
- These incidents can result from:
  - Malicious activity (e.g. attacker stealing information).
  - Accidental activity (e.g. lost unencrypted laptop).
  - Natural activity (e.g. earthquake destroying a datacenter).
- Security professionals are responsible for understanding these risks and implementing controls to manage them.

3.1. The DAD Triad

- The DAD triad describes three key threats to cybersecurity efforts, directly mapping to the CIA triad:
  - Disclosure: Exposure of sensitive information to unauthorized individuals (data loss). This violates confidentiality.
    - Can be intentional (data exfiltration) or accidental (e.g. misconfigured access controls, lost device).
  - Alteration: Unauthorized modification of information. This violates integrity.
    - Can be malicious (e.g. financial gain), natural (e.g. power surge causing data corruption), or accidental (e.g. typos).
  - Denial: Disruption of an authorized user's legitimate access to information. This violates availability.
    - Can be intentional (e.g. DDoS attack), accidental (e.g. server failure), or natural (e.g. natural disaster impacting communication).
- The CIA and DAD triads are useful tools for cybersecurity planning and risk analysis, providing a starting point for assessing threats and controls.

3.2. Breach Impact Categories

- The impacts of a security incident can be categorized as:
  - Financial Risk: Monetary damage to the organization.
    - Direct: Cost of rebuilding, incident response, forensic analysis.
    - Indirect: Revenue loss due to competitive disadvantage (e.g. stolen product plans).
  - Reputational Risk: Loss of goodwill among customers, employees, suppliers, and stakeholders due to negative publicity. Often difficult to quantify.
    - Identity Theft: A common impact on individuals when Personally Identifiable Information (PII) is exposed (e.g. Social Security numbers, bank accounts, credit card info). Organizations must protect PII.
  - Strategic Risk: Risk that the organization becomes less effective in meeting its major goals and objectives.
    - Can involve inability to bring new products to market, significant development delays, or competitors gaining advantage.
  - Operational Risk: Risk to the organization's ability to carry out its day-to-day functions.
    - Causes inefficiency, delays, or necessitates manual workarounds.
    - Differs from strategic risk by causing inefficiency and delay rather than jeopardizing the organization's existence or core business plans.
  - Compliance Risk: Organization violates legal or regulatory requirements due to a breach.

        - Example: Loss of Protected Health Information (PHI) violating HIPAA, leading to sanctions and fines.
        - Nature of risks depends on jurisdiction, industry, and data types.
- A single data breach can lead to impacts across multiple categories (e.g. reputational damage leading to financial loss, compliance fines, and direct costs).

4. Implementing Security Controls

- Control objectives are statements of a desired security state, defining the level of protection needed for confidentiality, integrity, and availability.
- Security controls are the specific measures implemented to achieve these objectives.

4.1. Gap Analysis

- Cybersecurity professionals conduct gap analyses to evaluate security controls.
- This involves reviewing control objectives and examining the existing controls.
- A gap exists when controls do not meet a control objective.
- Identified gaps should be treated as potential risks and remediated as resources permit.

4.2. Security Control Categories

- Security controls are categorized by their mechanism of action:
    - Technical controls: Enforce CIA in the digital space.
        - Examples: Firewall rules, access control lists, intrusion prevention systems, encryption.
    - Operational controls: Processes for managing technology securely.
        - Examples: User access reviews, log monitoring, vulnerability management.
    - Managerial controls: Procedural mechanisms focusing on risk management.
        - Examples: Periodic risk assessments, security planning, security integration into change management and project management.
    - Physical controls: Impact the physical world.
        - Examples: Fences, perimeter lighting, locks, fire suppression systems, burglar alarms.
- Organizations select controls based on their objectives and external regulations.
- Many objectives require a combination of technical, operational, and managerial controls (e.g. biometric locks (physical), access reviews (operational), risk assessments (managerial) for datacenter access).

4.3. Security Control Types

- Security controls are also categorized by their desired effect:
    - Preventive controls: Stop a security issue before it occurs.

        - Examples: Firewalls, encryption.
    - Deterrent controls: Prevent an attacker from attempting to violate policies.
        - Examples: Guard dogs, barbed wire fences.
    - Detective controls: Identify security events that have already occurred.
        - Example: Intrusion detection systems.
    - Corrective controls: Remediate security issues that have already occurred.
        - Example: Restoring backups after a ransomware attack.
    - Compensating controls: Mitigate risk associated with exceptions to a security policy.
        - Purpose: Provide alternative means to achieve an objective when the original control cannot be met.
        - PCI DSS Example: Using an isolated network for a system with an outdated OS.
        - Often used for temporary exceptions, with remediation plans to achieve full compliance.
    - Directive controls: Inform employees what to do to achieve security objectives.
        - Examples: Policies and procedures.

5. Data Protection

- Security professionals protect the confidentiality, integrity, and availability of sensitive data.
- Data exists in three states:
    - Data at rest: Stored on media (hard drives, tapes, cloud). Vulnerable to theft by insiders or external attackers.
    - Data in transit: In motion over a network. Vulnerable to eavesdropping attacks on untrusted networks.
    - Data in use: Actively being used by a computer system (e.g. in memory). Vulnerable to attackers with system control.
- Different security controls are used to safeguard data in all states.

5.1. Data Encryption

- Encryption uses mathematical algorithms to protect data.
- Renders data unintelligible without the correct decryption key.
- Used to protect data in transit and at rest.

5.2. Data Loss Prevention (DLP)

- DLP systems enforce information handling policies to prevent data loss and theft.
- They search systems for unsecured sensitive information and monitor network traffic for exfiltration attempts.
- Can block transmissions and alert administrators.

- Work in two environments:
  - Agent-based DLP: Software agents installed on systems search for sensitive information (e.g. SSNs, credit card numbers) and can monitor user actions (e.g. blocking USB access).
  - Agentless (network-based) DLP: Dedicated devices on the network monitor outbound network traffic for unencrypted sensitive information and can block or automatically encrypt transmissions (e.g. email).
- Two mechanisms of action:
  - Pattern matching: Watches for characteristic signs of sensitive data (e.g. credit card formats, keywords like "Top Secret").
  - Watermarking: Applies electronic tags to sensitive documents, and DLP systems monitor for unencrypted content with these tags.

5.3. Data Minimization

- Data minimization techniques reduce risk by reducing the amount of sensitive information maintained.
- Best achieved by destroying data when no longer needed.
- If complete removal isn't possible, data can be deidentified (ability to link data back to an individual is removed).
- Alternatively, data obfuscation transforms data so original information cannot be retrieved:
  - Hashing: Transforms a value into a hash value using a hash function. While not directly reversible, vulnerable to rainbow table attacks if possible input values are known.
  - Tokenization: Replaces sensitive values with a unique identifier using a lookup table. The lookup table must be kept secure.
  - Masking: Partially redacts sensitive information by replacing parts with blank characters (e.g. XXXX-XXXX-XXXX-1234 for credit cards).

5.4. Access Restrictions

- Access restrictions limit access to sensitive information or resources.
- Two common types:
  - Geographic restrictions: Limit access based on physical location (e.g. users within a country).
  - Permission restrictions: Limit access based on user's role or authorization level (e.g. only authorized personnel access financial data).

5.5. Segmentation and Isolation

- Segmentation: Places sensitive systems on separate networks with strict communication restrictions to other networks.
- Isolation: Completely cuts a system off from external network access.

CHAPTER 2: Cybersecurity Threat Landscape

1. CompTIA Security+ Exam Objectives Covered

- Domain 2.0: Threats, Vulnerabilities, and Mitigations: Compare and contrast common threat actors and motivations. Explain common threat vectors and attack surfaces. Explain various types of vulnerabilities.
- Domain 4.0: Security Operations: Explain various activities associated with vulnerability management:

2. Exploring Cybersecurity Threats

- Cybersecurity threats are increasingly sophisticated and diverse, involving skilled technologists, organized criminal syndicates, and government-sponsored attackers.
- Understanding the threat environment is crucial for developing appropriate defensive mechanisms to safeguard an organization's confidentiality, integrity, and availability.

2.1. Classifying Cybersecurity Threats

- Understanding the characteristics that differentiate threat actors is crucial for defense. Key attributes include:
  - Internal vs. External: Threats can originate from outside the organization (competitors, criminals) or within (insider threats).
  - Level of sophistication ranges from unskilled attackers using borrowed code to Advanced Persistent Threat (APT) actors exploiting self-discovered vulnerabilities.
  - Resources vary from limitless resources (organized crime, nation-states) to hobbyists with limited means.
  - Motivation can range from the thrill of attack (unskilled) to corporate espionage (competitors), political objectives (nation-states), or direct financial gain (organized crime).

2.2. The Hats Hackers Wear (Motivations)

- A shorthand lingo in the cybersecurity community to describe attacker motivations:
  - White-hat hackers (Authorized attackers): Act with authorization to discover and correct security vulnerabilities. May be employees or penetration testing contractors.
  - Black-hat hackers (Unauthorized attackers): Have malicious intent, seeking to compromise confidentiality, integrity, or availability for unauthorized purposes.
  - Gray-hat hackers (Semi-authorized attackers): Act without proper authorization but with the intent of informing targets of vulnerabilities.

- Important Note: Despite good intent, gray-hat hacking is not legal or ethical and can be punished as criminal offenses.

3. Threat Actors

3.1. Unskilled Attackers

- Known as script kiddies (a derogatory term).
- Have limited skills and often rely on automated tools downloaded from the Internet.
- Possess little knowledge of how their attacks actually work.
- Seek out convenient targets of opportunity.
- Despite low skill, they pose a real threat due to:
  - Free availability of simplistic hacking tools (e.g. DoS, viruses, Trojans, ransomware-as-a-service). Personal technical skills are no longer a barrier.
  - Their abundance and unfocused nature. They are less discriminating in target selection, often discovering victims unknowingly.
- Motivation is primarily to prove their skill; they attack because a vulnerability exists. Secondary schools and university networks are common targets.
- Limitations: Their large numbers are offset by lack of skill and resources (time and money). They typically work alone and cannot sustain continuous attacks.

3.2. Hacktivists

- Use hacking techniques to accomplish activist goals.
- Motivations stem from philosophical or political beliefs; they believe their activity is for the greater good, even if illegal.
- Measures that deter other attackers are less effective because hacktivists are willing to risk being caught for their cause.
- Skill levels vary widely. From unskilled to highly skilled (some may be cybersecurity professionals in their "day job").
- Resources vary. Many work alone with limited resources; others are part of organized efforts.
  - Anonymous is a well-known hacktivist group, known for collective decision-making and attacks on diverse targets.
  - Large, distributed, and anonymous groups are powerful due to more time/resources and difficulty in identification/prosecution.
- Tend to be external attackers, but some are internal employees (insider threats) who disagree with company policies and may release confidential information (e.g. Edward Snowden).

3.3. Organized Crime

- Motive is illegal financial gain. They avoid drawing attention and are not driven by political causes or showing off skills.
- Activities (per EUROPOL 2021 IOCTA):
  - Cyber-dependent crime: Ransomware, data compromise, DDoS attacks, website defacement, critical infrastructure attacks.
  - Child sexual abuse material.
  - Online fraud: Credit card fraud, business email compromises.
  - Dark web activity: Sale of illegal goods and services.
  - Cross-cutting crime factors: Social engineering, money mules, criminal abuse of cryptocurrencies.
- Skill level ranges from moderately skilled to highly skilled. Unskilled involvement is rare and often quickly caught.
- Tend to have more resources (time and money) than hacktivists or unskilled attackers, willing to invest for significant returns.

3.4. Nation-State Attackers (Advanced Persistent Threats - APTs)

- Describe a series of attacks linked to government-sponsored entities (military, intelligence).
- Characteristics of APTs:
  - Advanced techniques: Use sophisticated methods, not just readily available tools.
  - Persistent: Attacks occur over a significant period, sometimes years, as attackers patiently stalk targets.
- Skill level is characterized by highly skilled attackers.
- Possess significant resources (labor, time, money) to finance ongoing, sophisticated attacks.
- Motives can be political (traditional espionage, gathering defense information) or economic (targeting intellectual property, economic assets).
- Zero-Day Attacks:
  - APT attackers often conduct their own security vulnerability research to find unknown vulnerabilities (zero-days).
  - They do not disclose these but store them for later use.
  - Zero-day attacks exploit these unknown vulnerabilities, making them particularly dangerous as no patches are available.
  - Stuxnet is a well-known example of an APT attack using zero-day vulnerabilities (traced to U.S. and Israeli governments, targeting Iranian nuclear facilities).

3.5. Insider Threat

- Occurs when an employee, contractor, vendor, or authorized individual uses their access to attack the organization.
- Goal is often disclosing confidential information but can also include altering information or disrupting business processes.
- Skill level can be any level, from unskilled to highly technical.
- Motivation is varied, including activist goals, financial gain, or personal grievances.
- Usually working alone with limited financial resources and time.
- Automatic advantage is that they possess existing access and knowledge of the network, which can be significant depending on job role.
- Detection can be behavioural assessments in conjunction with HR can help identify and intervene before escalation.

3.6. The Threat of Shadow IT

- Individuals or groups acquiring and using unapproved technology solutions (e.g. personal cloud storage for work content).
- Intent is often not malicious but driven by a desire for productivity and unmet business needs.
- Risk: Puts sensitive information in the hands of vendors outside the organization's control.
- Mitigation: Cybersecurity teams should be vigilant, recognize that it indicates unmet business needs, and collaborate with users to find secure alternatives.

3.7. Competitors

- Engage in corporate espionage to steal sensitive information for business advantage.
- Targets can be customer information, proprietary software, product development plans, and other beneficial information.
- May use disgruntled insiders or purchase information from the dark web (where confidential corporate data is sold by hackers or insiders).
- Note on Threat Assessments: Organizations should conduct periodic organizational threat assessments to identify the most likely threat actors targeting them and their motivations.

4. Attacker Motivations

- Understanding attacker motivations helps in identifying potential targets and designing defenses.
- Common motivations behind cyberattacks include:
    - Data exfiltration: Desire to obtain sensitive or proprietary information (e.g. customer data, intellectual property).

- Espionage: Organizations (nation-states or corporations) seeking to steal secret information from others.
- Service disruption: Aim to take down or interrupt critical systems or networks (e.g. banking, healthcare).
- Blackmail: Extort money or concessions by threatening to release sensitive information or launch further attacks.
- Financial gain: Desire to make money through theft or fraud (common for organized crime).
- Philosophical/political belief: Motivated by ideological or political reasons (e.g. hacktivists promoting a cause).
- Ethical attacks (white-hat hacking): Desire to expose vulnerabilities and improve security, often with permission.
- Revenge: Desire to get even with an individual or organization, causing embarrassment or retribution.
- Disruption/chaos: Desire to cause chaos and disrupt normal operations.
- War: Military or civilian groups using hacking to disrupt military operations and influence armed conflict outcomes.

Exam Note: It is very likely you will be asked to compare and contrast the various threat actors on the exam. You should know the attributes and motivations behind each.

5. Threat Vectors and Attack Surfaces

- Attack Surface: A system, application, or service with a vulnerability that can be exploited.
- Threat Vector: The means threat actors use to gain access by exploiting a vulnerability.
- Security professionals aim to reduce the size and complexity of the attack surface through effective security and risk mitigation.

5.1. Message-Based Threat Vectors

- Email: Most common vector (e.g. phishing, spam). Easy to execute, target many users, and require only one successful compromise.
- Other Messaging: Includes SMS (text messages) and Instant Messaging (IM).
- Voice Calls: Used for vishing (voice phishing) attacks.
- Social Media: Can be used for direct targeting or to harvest user information for other attacks.

5.2. Wired Networks

- Bold attackers may physically enter facilities to access wired networks.
- Unsecured Network Jacks: Common method; attackers connect laptops to unsecured wall jacks in public areas.

- Attackers gaining physical access may find unsecured computer terminals or network devices.
- Physical access to a component means the device can be compromised.
- Highlights the crucial role of physical security.

5.3. Wireless Networks

- Easier access as attackers can access from outside the facility (e.g. parking lot).
- Bluetooth devices may lack security settings allowing unauthorized connections.
- Unsecured or poorly secured wireless networks pose a significant security risk.

5.4. Systems

- Individual systems can be threat vectors based on configuration and software.
- Vulnerabilities include:
  - Open service ports: Unnecessary or using default credentials.
  - Vulnerable software: Known or undetected flaws in installed applications.
  - Unsupported systems and applications: Legacy software no longer receiving vendor support.
- Any of these can provide a foothold for an attacker.

5.5. Files and Images

- Individual files, including images, can contain embedded malicious code.
- Attackers trick users into opening these files, activating malware infection.
- Distribution is via email, file servers, or any location where users might open them.

5.6. Removable Devices

- USB drives and other removable media are common for spreading malware.
- Attackers distribute them (e.g. in parking lots) hoping users will plug them in, triggering a malware infection and system compromise.

5.7. Cloud

- Attackers scan for:
  - Files with improper access controls.
  - Systems with security flaws.
  - Accidentally published API keys and passwords.
- Organizations must include cloud services in their security program.
- Cloud vulnerabilities are similar to on-premises, but controls often differ.

5.8. Supply Chain

- Sophisticated attackers target an organization's IT supply chain (hardware, software, service providers).
- Provides an indirect attack mechanism.

- Hardware: Tampering with devices during manufacturing or transit to insert backdoors. This is a difficult third-party risk to anticipate.
- Software: Inserting vulnerabilities into software before release or deploying backdoors via official update/patching mechanisms.
- Managed Service Providers (MSPs): Infiltrating an MSP can grant attackers leverage through the MSP's access to its customers' systems.
- Other Issues: Vendor failure to support systems, provide integrations, or secure outsourced code development/data storage.
- Mitigation: Strong vendor management practices help identify and address these risks.

Exam Note: Be ready to identify and explain the common threat vectors and attack surfaces.

6. Threat Data and Intelligence

- Threat intelligence encompasses activities and resources for cybersecurity professionals to learn about the evolving threat environment.
- It's crucial for building appropriate defenses and conducting predictive analysis of risks.
- Threat feed data provides up-to-date details about threats, including:
  - Technical details: IP addresses, hostnames, domains, email addresses, URLs, file hashes, file paths, Common Vulnerabilities and Exposures (CVE) record numbers.
  - Additional context: Why an organization might be a target, descriptions of threat actors, motivations, and methodologies.
- Vulnerability databases are essential, directing defensive efforts and providing insight into new exploits.
- Indicators of Compromise (IoCs): Telltale signs an attack has occurred (e.g. file signatures, log patterns). Found in threat intelligence and file/code repositories.

6.1. Open-Source Intelligence (OSINT)

- Threat intelligence acquired from publicly available sources.
- Organizations increasingly share threat information openly.
- Challenges include selecting reliable, up-to-date sources and leveraging them effectively.
- Examples of Open-Source Threat Intelligence Feeds/Sources:
  - Senki.org: Provides a list at www.senki.org/operators-security-toolkit/open-source-threat-intelligence-feeds
  - AT&T Open Threat Exchange: Global community of security professionals and researchers at cybersecurity.att.com/open-threat-exchange

- MISP Threat Sharing project: Standardized feeds with community collections at www.misp-project.org/feeds
- Threatfeeds.io: Hosts a list with details on maintenance and updates at https://threatfeeds.io

- Government and Public Sources:
  - U.S. Cybersecurity & Infrastructure Security Agency (CISA): www.cisa.gov
  - CISA Automated Indicator Sharing (AIS) program: www.cisa.gov/topics/cyber-threats-and-advisories/information-sharing/automated-indicator-sharing-ais
  - CISA Information Sharing and Analysis Organizations (ISAOs) program: cisa.gov/information-sharing-and-analysis-organizations-isaos
  - U.S. Department of Defense Cyber Crime Center: dc3.mil
  - Australian Signals Directorate's Cyber Security Centre: cyber.gov.au
- Vendor Websites:
  - Microsoft's threat intelligence blog: microsoft.com/en-us/security/blog/topic/threat-intelligence
  - Cisco Security Advisories site: sec.cloudapps.cisco.com/security/center/publicationListing.x (includes expert blog and Cisco Talos reputation lookup tool: https://talosintelligence.com)
- Public Sources:
  - The SANS Internet Storm Center: https://isc.sans.org
  - VirusShare: Details about malware uploaded to VirusTotal: https://virusshare.com
  - The Spamhaus Project: Focuses on blocklists (SBL, XBL, PBL, DBL, DROP, etc.) for spam and compromised systems: www.spamhaus.org

6.2. Exploring the Dark Web

- Dark web: A network using multiple layers of encryption for anonymous communication over standard internet connections.
- Hackers use it to share information and sell stolen credentials and data.
- Threat intelligence teams should monitor dark web marketplaces for their organization's or clients' credentials, as their sudden appearance indicates a potential successful attack.
- Accessible via the Tor browser: www.torproject.org.

6.3. Proprietary and Closed-Source Intelligence

- Developed and used by commercial security vendors, government organizations, and other security-centric entities.
- Involves their own information gathering, research, custom tools, and analysis models.
- Reasons for proprietary nature: keeping data secret, selling/licensing as trade secrets, or preventing threat actors from knowing what data is being collected.

- Can be a compelling resource due to the overwhelming volume of OSINT.
- Offers curated, validated, and readily applicable threat data, saving significant effort in identifying, defining, and applying relevant threats.

6.4. When a Threat Feed Fails

- It is critical to have reliable, up-to-date feeds.
- Delayed or incomplete feeds can lead to system exposure.
- Recommendation: Consider multiple good-quality, reliable feeds to cross-check and ensure timely information.

6.5. Threat Maps

- Provide a geographic view of threat intelligence.
- Many vendors offer real-time maps (e.g. Check Point Cyber Threat Map: hreatmap.checkpoint.com).
- Can offer insight into attack sources, but geographic attribution should be viewed skeptically as attackers often relay attacks through compromised networks, hiding their true location.

6.6. Assessing Threat Intelligence

- Regardless of the source, threat intelligence information needs to be assessed based on several factors:
  - Timeliness: Is the information up to date? Delays can cause missed threats or late reactions.
  - Accuracy: Is the information reliable? Does it rely on single or multiple, often-correct sources?
  - Relevance: Is the information applicable to your organization's platforms, software, or target profile?
- Confidence Scores: A way to summarize threat intelligence assessment, allowing filtering and use based on trust level.
  - Lower confidence information shouldn't be ignored but should not be solely relied upon for important decisions.
- Confidence Level Scales: Many threat feeds include ratings, often with a descriptive scale. An example six-level scale:
  - Confirmed (90–100): Proved by independent sources or direct analysis.
  - Probable (70–89): Relies on logical inference, not direct confirmation.
  - Possible (50–69): Some information agrees, but not confirmed.
  - Doubtful (30–49): Possible but not most likely, or cannot be proven/disproven.
  - Improbable (2–29): Possible but not logical, or refuted by other information.
  - Discredited (1): Confirmed inaccurate or incorrect.
  - Other common scales: 1-5, 1-10, High/Medium/Low.

6.7. Threat Indicator Management and Exchange

- Standardization and tooling are needed for automated processing of threat information.
- Structured markup languages facilitate indicator management:
  - Structured Threat Information eXpression (STIX): An XML language (now maintained by OASIS). Defines 18 STIX Domain Objects (e.g. attack patterns, malware, threat actors) and STIX Relationship Objects (relationship, sighting) to describe threats consistently. Uses defined vocabulary for fields like sophistication and resource level.
  - OpenIOC: Another structured markup language.
- Benefit of Standardization: Allows leveraging multiple threat feeds even if they originally use different formats or finding sources that combine feeds.
- Trusted Automated eXchange of Intelligence Information (TAXII): A companion protocol to STIX, designed to communicate cyber-threat information at the application layer via HTTPS, specifically supporting STIX data exchange. More info: https://oasis-open.github.io/cti-documentation.

6.8. Information Sharing Organizations

- Information Sharing and Analysis Centers (ISACs) (in the U.S.): Help infrastructure owners and operators share threat information and provide tools/assistance to members.
  - Established based on Presidential Decision Directive-63 (PDD-63).
  - Operate on a trust model for in-depth sharing of physical and cyber threats, often 24/7.
  - National Council of ISACs lists sector-based ISACs: www.nationalisacs.org/member-isacs-3.
  - Specific U.S. agencies/department partners for critical infrastructure: www.cisa.gov/topics/critical-infrastructure-security-and-resilience/critical-infrastructure-sectors.
- Many countries have similar government bodies/agencies (e.g. UK National Protective Security Authority: www.npsa.gov.uk).

6.9. Conducting Your Own Research

- Security professionals should continuously conduct their own research into emerging threats.
- Sources for threat research toolkit:
  - Vendor security information websites.
  - Vulnerability and threat feeds (from vendors, government, private organizations).
  - Academic journals and technical publications (e.g. Internet Request for Comments (RFC) documents for detailed protocol specifications).
  - Professional conferences and local industry group meetings.
  - Social media accounts of prominent security professionals.
- Focus on learning adversary tactics, techniques, and procedures (TTPs) to improve your threat intelligence program.

CHAPTER 3: Malicious Code

This chapter covers various types of malware, their characteristics, indicators of compromise (IoCs), and mitigation strategies, essential for the CompTIA Security+ exam.

1. Malware Defined

- Malware is a term for software specifically designed to cause harm to systems, networks, or users.
- It can also covertly gather information, grant unauthorized access, or perform actions without the system owner's consent.
- For the exam, understanding the unique features, identification methods, and countermeasures for each malware type is crucial.

1.1. Ransomware

- Ransomware is a type of malware that takes control of a computer and demands payment for its release.
- Crypto malware is a common form that encrypts files, holding them hostage until a ransom is paid.
- Other tactics include threatening to report users for illicit activities (e.g. pirated software) or exposing sensitive personal data.
- Delivery Methods:
  - Primarily through phishing campaigns, where users unknowingly install malware via malicious emails or links.
  - Direct attacks like exploiting Remote Desktop Protocol (RDP), vulnerable services, or compromised front-facing applications.
- Indicators of Compromise (IoCs):
  - Command and Control (C&C) traffic or connections to known malicious IP addresses.
  - Abnormal use of legitimate tools to maintain system control.
  - Lateral movement attempts to compromise or gather information from other systems within the network.
  - File encryption.
  - On-screen ransom demands or notices about the encryption process.
  - Data exfiltration behaviors, such as large file transfers.
  - *Example:* CISA's advisory on Royal Ransomware (www.cisa.gov/news-events/cybersecurity-advisories/aa23-061a) details specific IoCs.
- Mitigation:
  - Effective backup systems are paramount; backups should be stored separately to prevent encryption by ransomware.

- Organizations need a response plan for ransomware, including whether to pay the ransom (which doesn't guarantee file recovery or prevent further demands).
- Some ransomware can be defeated with preexisting decryption tools provided by antivirus/antimalware vendors and the security community.

1.2. Trojans

- Trojans (Trojan horses) are malware disguised as legitimate software.
- They rely on unsuspecting users to execute them, thereby creating an entry point for attackers.
- *Example:* The Triada Trojan was often distributed as a modified, "feature-enhanced" WhatsApp version.
  - Upon launch, it gathers device information (IDs, hardware address) and registers with a remote server.
  - The Trojan then downloads, decrypts, and executes additional malicious components, leading to activities like displaying ads or signing up for paid subscriptions.
- Indicators of Compromise (IoCs):
  - Signatures for the specific malware application or downloaded files.
  - Command and control (C&C) system hostnames and IP addresses.
  - Unusual folders or files created on target devices.
  - *Further reading on Triada Trojan:* https://securelist.com/triada-trojan-in-whatsapp-mod/103679 and https://securelist.com/malicious-whatsapp-mod-distributed-through-legitimate-apps/107690.
- Remote Access Trojans (RATs):
  - Provide attackers with remote access to systems.
  - Can be challenging to distinguish from legitimate remote support tools, potentially leading to false positives with antimalware.
- Mitigation:
  - Security awareness training is key to discourage users from downloading untrusted software.
  - Controlling software and application installations by users (balanced with user flexibility).
  - Antimalware, Endpoint Detection and Response (EDR), and similar tools to detect and prevent malicious software execution based on file signatures or behavioral analysis.

1.3. Bots, Botnets, and Command and Control

- Command and Control (C&C): Techniques and systems used by attackers to issue commands to infected systems.
- Bots: Individual systems under central command.
- Botnets: Groups of bots controlled by attackers via a C&C system.

- C&C Communication:
  - Increasingly uses encrypted HTTP connections to frequently changing remote hosts to evade detection.
  - Older methods like Internet Relay Chat (IRC) via port 6667 are still used.
- Defender's Role: Identify C&C communications and investigate systems reaching out to unknown hosts as potential signs of botnet involvement.

1.4. Worms

- Worms are a type of malware capable of self-replication and automated spreading without user interaction.
- Unlike Trojans, they don't need a user to click on them to activate; they self-install.
- Spread Mechanisms:
  - Exploiting vulnerable services.
  - Email attachments.
  - Network file shares.
  - Compromised IoT (Internet of Things) devices and phones.
- Stuxnet Example (Nation-State Level):
  - A 2010 cyberweapon targeting the Iranian nuclear program.
  - Spread to air-gapped systems (physically isolated) via USB thumb drives.
  - Used advanced techniques:
    - Leveraged a trusted digital certificate.
    - Searched for specific industrial control systems (ICSs).
    - Programmed to damage centrifuges while providing false monitoring data to hide the attack.
  - *Reference:* www.wired.com/2014/11/countdown-to-zero-day-stuxnet and https://spectrum.ieee.org/the-real-story-of-stuxnet.
- Raspberry Robin Example (Modern Worm):
  - A contemporary worm used in pre-ransomware activities.
  - Initially spread via infected USB drives using LNK files.
  - Uses built-in Windows tools (e.g. cmd.exe, msiexec.exe) for persistence and further tasks.
  - *Reference:* Microsoft's detailed write-up: www.microsoft.com/en-us/security/blog/2022/10/27/raspberry-robin-worm-part-of-larger-ecosystem-facilitating-pre-ransomware-activity.
- Indicators of Compromise (IoCs):
  - Detection of known malicious files.

- Downloads of additional components from remote systems.
- Command and Control (C&C) contact to remote servers.
- Malicious behaviors using system commands for injection and other activities (e.g. cmd.exe, msiexec.exe usage).
- Hands-on-keyboard attacker activity.

- Mitigation:
  - Network-level controls are the primary defense: firewalls, Intrusion Prevention Systems (IPS), network segmentation to contain infections.
  - Patching and configuring services to reduce attack surfaces.
  - Post-infection: Antimalware, EDR tools to stop and remove.
  - Severe infections may require reinstallation or resetting to original firmware.

1.5. Spyware

- Spyware is malware designed to secretly gather information about an individual, organization, or system.
- Information Targeted: Browse habits, installed software, sensitive data, or even enabling remote access to webcams.
- Associated with identity theft, fraud, advertising redirection, Digital Rights Management (DRM) monitoring, and stalkerware (monitoring partners).
- Severity can range from relatively innocuous tracking to highly malicious data theft & illicit access.
- IoCs (can be similar to other malware types):
  - Remote-access and remote-control-related indicators.
  - Known software file fingerprints.
  - Malicious processes (often disguised as legitimate system processes).
  - Injection attacks against browsers.
- Defining Spyware: Its primary intent to gather information is the key differentiator, regardless of its propagation method (which might resemble Trojans, worms, or viruses).
- *Example:* NSO Group's Pegasus spyware (Amnesty International write-up: www.amnesty.org/en/latest/research/2021/07/forensic-methodology-report-how-to-catch-nso-groups-pegasus).
- Mitigation:
  - Antimalware tools (often with specific antispyware capabilities).
  - User awareness to prevent installation, especially when bundled with legitimate software installers (acting as a Trojan).
  - Controlling allowed software on devices.

1.6. Bloatware

- Bloatware refers to unwanted applications preinstalled on systems by manufacturers.
- Origin is commercial partnerships, manufacturer-provided programs, or bundled in other installer packages.
- Key Distinction: Unlike other malware, bloatware is not usually intentionally malicious.
- Risks:
  - Poorly written code.
  - "Call home" functionality (reporting system/usage data).
  - Potential vulnerabilities that create additional attack surfaces.
  - Consumes disk space, memory, and CPU cycles unnecessarily.
- Not typically associated with IoCs since it's not inherently malicious.
- Mitigation:
  - Uninstallation of unwanted programs.
  - Using a clean operating system image.
  - Awareness of its presence.

1.7. Exam Note (Spyware vs. Bloatware)

- The CompTIA Security+ exam highlights both spyware and bloatware.
- Spyware's primary intention is to gather information about the user, system usage, and configuration.
- Bloatware is simply unwanted software that may have call-home functions but lacks the core malicious intent of information gathering.

1.8. Viruses

- Viruses are malicious programs that self-copy and self-replicate once activated.
- Unlike worms, they do not spread automatically via vulnerable services; they require an infection mechanism (e.g. copying to a thumb drive or network share).
- They typically have a trigger (conditions for execution) and a payload (the malicious action performed).
- Types of Viruses:
  - Memory-resident viruses: Stay in system memory while the device is running.
  - Non-memory-resident viruses: Execute, spread, and then terminate.
  - Boot sector viruses: Reside in the boot sector of a drive or storage media.
  - Macro viruses: Use macros (embedded code) within applications like word processors to spread.
  - Email viruses: Spread via email attachments or by exploiting flaws in email clients.

- Fileless Virus Attacks:
  - Similar to traditional viruses but do not require local file storage.
  - Spread via methods like spam email and malicious websites, exploiting browser/plugin vulnerabilities.
  - Inject themselves directly into memory and perform malicious activity.
  - Achieve persistence (surviving reboots) through techniques like Registry entries.
  - *Mitigation for Fileless Attacks:*
    - Keep browsers, plugins, and other software updated to patch vulnerabilities.
    - Use antimalware tools that detect unusual behavior from scripting tools like Microsoft PowerShell.
    - Network-level defenses (IPS) and reputation-based protection systems to block access to known malicious sites.
- Indicators of Compromise (IoCs):
  - Often available in threat feeds from organizations like VirusTotal (e.g. crowdsourced YARA rules dashboard: https://support.virustotal.com/hc/en-us/articles/9853517705117-Crowdsourced-YARA-rules-dashboard).
- Mitigation:
  - User awareness to prevent clicking on and activating viruses.
  - Antimalware tools for detection and prevention (on-disk, in-memory, or during execution).
  - Removal challenges: Complex infections are hard to fully remove.
  - Best Practice for Removal: Wiping the infected drive and restoring from a known good backup, or reinstallation/reimaging the system. This ensures complete removal, especially for complex or deeply embedded malware.

1.9. Keyloggers

- Keyloggers are programs that capture keystrokes from a keyboard, and can also capture other inputs (mouse, touchscreen, credit card swipes).
- Mechanism: Capture data from the kernel, APIs, scripts, or directly from memory.
- Goal is to capture user input (especially credentials) for attacker analysis and use.
- Indicators of Compromise (IoCs):
  - File hashes and signatures.
  - Exfiltration activity to command and control (C&C) systems.
  - Suspicious process names.
  - Known reference URLs.

- *Example:* Analysis of a keylogger delivery campaign via PDFs: www.socinvestigation.com/pdf-campaign-delivering-snake-keylogger.

- Mitigation:
  - General security best practices to prevent malware installation: patching, system management, antimalware tools.
  - Multifactor authentication (MFA) can limit the impact of compromised credentials, even if the keylogger itself is not defeated.
  - In untrusted environments, bootable USB drives can be used to bypass a potentially compromised OS.
- Hardware Keyloggers: Physical devices that capture keystrokes, independent of software. These are also a threat, often inexpensive, and can be used to acquire credentials.

1.10. Logic Bombs

- Logic bombs are not independent malicious programs.
- They are functions or code embedded within other programs that activate only when specific pre-set conditions are met.
- While relatively rare, they can have significant impact if triggered.
- Indicators of Compromise (IoCs): Less common, as they require code analysis to discover.
- Mitigation: Primarily focuses on code review during software development and system management.

2. Analyzing Malware (General Techniques)

- Online Analysis Tools: (e.g. VirusTotal) Check against multiple AV engines and identify known tools/behaviors.
- Sandbox Tools: Analyze malware behavior in an isolated, protected environment.
- Manual Code Analysis: Common for scripts (Python, Perl) and interpreted code.
- String Analysis Tools: (e.g. strings command) Look for recoverable artifacts and useful data within executables.

3. Rootkits

- Rootkits are malware designed to provide attackers with persistent backdoor access to a system.
- They actively conceal their presence through various techniques:
  - Hooking filesystem drivers to hide files.
  - Infecting startup code (e.g. Master Boot Record - MBR) to evade full-disk encryption detection.

- Detection Challenges: An infected system cannot be trusted.
  - Best method: Test the suspected system from a trusted external system/device.
  - Rootkit detection tools look for typical behaviors and signatures.
  - Integrity checking and data validation against expected responses can help.
- Removal Challenges: Very difficult to ensure complete removal.
  - Common recommendation: Rebuild the system or restore from a known good backup.
  - Simplified by virtual machines, containers, and imaging.
- Intentional Rootkits: Some are installed legitimately for DRM, anti-cheating, or copy protection defeat (though security professionals primarily focus on malicious ones).
- Indicators of Compromise (IoCs):
  - File hashes and signatures.
  - Command and control (C&C) domains, IP addresses, and systems.
  - Behavior-based identification: creation of services/executables, configuration changes, file access, command invocation.
  - Opening ports or creating reverse proxy tunnels.
  - *Example:* Rootkit used on ATMs: www.socinvestigation.com/unc2891-atm-rootkit-mandiant-advanced-practices-team-tracks-latest-indicators.
- Mitigation:
  - Standard security practices: Patching, secure configurations, proper privilege management.
  - Secure Boot and techniques that validate live systems/files to prevent installation or persistence.
  - Advanced Detection Technique: Remove the drive and connect it to another trusted system, or use system images/snapshots of virtual machines, to bypass the compromised OS's hiding mechanisms.

CHAPTER 4: Social Engineering and Password Attacks

1. Social Engineering and Human Vectors

- Social Engineering: The art of manipulating people to perform desired actions they might not otherwise do. It exploits human psychology rather than technical vulnerabilities.
- Key Principles (often combined in attacks):
  - Authority: People obey those perceived to be in charge (e.g. claiming to be a manager or government official).
  - Intimidation: Using fear or bullying to coerce action.
  - Consensus (Social Proof): People tend to follow what others are doing (e.g. "everyone else clicked the link").
  - Scarcity: Making something seem more desirable because it's limited or rare (e.g. "last one available").
  - Familiarity: Relying on the target liking the individual or the organization they claim to represent.
  - Trust: Building a personal connection with the target.
  - Urgency: Creating a feeling that immediate action is required due to a critical situation.
- Exam Note: The Security+ exam doesn't require categorizing attacks by these principles but understanding them helps in comprehending attack success and prevention.

2. Social Engineering Techniques

- Phishing: Fraudulently acquiring information (credentials, sensitive data) often via email.
  - Vishing: Phishing conducted via voice calls or voicemail messages.
    - Relies on urgency, authority, and emotional manipulation.
    - *Examples:* Requests to help a relative abroad (wire fraud), tax scams, threats of law enforcement, urgent tasks for senior executives.
  - Smishing: Phishing conducted via SMS (text) messages.
    - Often prompts users to click a malicious link (leading to fake sites for credential capture, malware download, or MFA code requests).
    - Uses pretexts similar to other phishing, building trust, urgency, or authority.
  - Spear Phishing: Targets specific individuals or groups within an organization.
  - Whaling: A type of spear phishing specifically targeting senior executives (e.g. CEOs, CFOs) – the "big fish."
- Defenses Against Phishing (All Types):
  - Awareness Training: Educating staff to recognize and report phishing (including periodic exercises).

- Technical controls include email filtering (reputation tools, keyword/text pattern matching) to detect likely phishing attempts.

3. Misinformation and Disinformation

- Online influence campaigns are common in cyberwarfare and traditional warfare, using social media, email, etc., to sway public opinion.
- Misinformation: Incorrect information resulting from factual errors or misunderstandings. (Think *mistake*).
- Disinformation: Incorrect, inaccurate, or false information intentionally provided to achieve specific goals. (Think *deception*).
- Malinformation: (Additional term, not explicitly on exam but good to know) Genuine information used with malicious intent.
- CISA's "TRUST" Process to Counter MDM:
  - Tell your story.
  - Ready your team.
  - Understand and assess MDM.
  - Strategize response.
  - Track outcomes.
- Organizations must monitor for misinformation and have a plan to counter it. CISA recommends:
  - Assessing the information environment.
  - Identifying vulnerabilities.
  - Fortifying communication channels.
  - Engaging in proactive communications.
  - Developing an incident response plan.
- *Reference:* CISA guide on MDM: www.cisa.gov/sites/default/files/publications/mdm-incident-response-guide_508.pdf.

4. Impersonation

- Pretending to be someone else to gain access, information, or other desired outcomes.
- Combines the target's willingness to believe the impersonator with social engineering principles (e.g. Authority, Trust).
- Identity Fraud/Theft: The use of someone else's identity, often for financial gain, but also used in penetration testing. Impersonation can be a limited form of identity fraud.

- Types of Impersonation: Can be specific (claiming a particular person's identity) or generic (e.g. pretending to be a delivery driver, service provider employee).

5. Business Email Compromise (BEC)

- Scams that leverage apparently legitimate email addresses to conduct fraud and other attacks.
- Common Examples: Invoice scams, gift card scams, data theft, account compromise.
- Methods for Creating Legitimate-Looking Emails:
  - Using compromised accounts.
  - Sending spoofed emails.
  - Employing fake but similar domain techniques.
  - Using malware or other tools.
- Sometimes called Email Account Compromise (EAC).
- *Reference:* Microsoft's detailed write-up: www.microsoft.com/en-us/security/business/security-101/what-is-business-email-compromise-bec.
- Mitigation:
  - Multifactor Authentication (MFA).
  - Awareness training.
  - Policies supporting appropriate email use and behaviors.

6. Pretexting

- Using a made-up scenario (pretext) to justify approaching an individual and asking for information or action.
- Often used in conjunction with impersonation to enhance believability.
- Defence: An aware target can ask questions, require verification (e.g. a simple verification phone call), which can often defeat these attacks.

7. Watering Hole Attacks

- Attacking targets by compromising websites they frequently visit.
- The compromised site acts as a "watering hole" where victims are guaranteed to visit.
- Attackers identify target's preferred sites, then compromise them (e.g. direct attack, deploying malware via advertising networks).

8. Brand Impersonation (Brand Spoofing)

- A type of phishing attack that uses emails designed to appear to be from a legitimate brand, leveraging name recognition and often using the brand's actual email templates.
- Goals:
  - Trick users into logging into fake accounts (especially for banks, online stores).
  - Request payment.
  - Gather passwords or other sensitive information.
  - Deliver malware (e.g. via malicious attachments).
- Varies from highly convincing to poorly constructed scams.

9. Typosquatting

- Attackers register misspelled or slightly altered URLs that are similar to legitimate websites.
- Relies on users mistyping URLs and landing on the attacker's site.
- Goals: Drive ad traffic, sell fake/similar products, or host malicious content.
- Prevention: Organizations may register common typos of their domains to redirect users to the legitimate site. *Example:* Visiting amason.com redirects to Amazon.com.

10. Pharming (Related to Typosquatting)

- A form of attack that redirects users to a malicious website without them mistyping the URL.
- Mechanism:
  - Modifying a system's hosts file (which prioritizes DNS lookups).
  - Using malware to change a system's DNS servers.
- Outcome: Unsuspecting victims are redirected to lookalike malicious sites.

11. Password Attacks

- Brute-Force Attacks:
  - Iterate through many password guesses until the correct one is found.
  - Can involve simple lists, common passwords, words relevant to the target, and modification rules (e.g. adding numbers/symbols).
  - Essentially, it's a trial-and-error process.
- Password Spraying Attacks:
  - A specialized form of brute-force attack.
  - Attempts to use a single password (or a small set) against many different accounts.

- Effective when attackers suspect common default or easily guessable passwords (e.g. sports team chants for a fan website).

- Dictionary Attacks:
  - Another form of brute-force attack.
  - Uses a pre-compiled list of words (a dictionary) as guesses.
  - Tools like John the Ripper (open source password cracking tool) often include built-in word lists.
  - Penetration testers may create custom dictionaries based on reconnaissance.

12. Online vs. Offline Attacks

- Online Attacks:
  - Occur against a live system (e.g. a login page).
  - Subject to active defenses like account lockout policies, rate limiting, and intrusion detection systems.
- Offline Attacks:
  - Occur against a compromised or captured password store (e.g. a database of hashed passwords).
  - Typically much faster as there are no live system defenses to bypass.

13. Rainbow Tables

- An easily searchable database of precomputed hashes.
- Used in offline attacks when hashed passwords are captured.
- The hashes in the rainbow table are computed using the same hashing methodology as the captured password file (e.g. MD5).
- Allows an attacker to simply "look up" a captured password hash to find its corresponding plaintext password.
- Hashing Concept:
  - A hash is a one-way cryptographic function that produces a unique, fixed-size output from an input.
  - It should ideally be irreversible (cannot derive original input from hash) and collision-resistant (different inputs should not produce the same hash).
  - Rainbow tables don't "break" the hash function; they perform a precomputed brute-force lookup to find the input that generates a given hash.

14. Password Cracking Tools

- John the Ripper: A popular open-source password cracking tool.
- Can perform brute-force and dictionary attacks against various common password storage formats.
- Can also be used as password assessment tools by organizations to identify weak passwords (though MFA and complexity requirements are increasingly replacing this).
- *Tutorials:* https://openwall.info/wiki/john/tutorials.

15. Secure Password Storage

- Avoid plain-text passwords. Never store passwords unencrypted.
- Passwords should never be stored directly. Instead, use a well-constructed password hash for verification at login.
- Hashing Mechanisms: Use strong, modern hashing algorithms.
- Salting: Adding a unique, random value (salt) to each password *before* hashing.
  - Makes rainbow table attacks much harder, as each user's hash is unique even if they have the same password.
- Pepper: An additional secret value (similar to a salt) that is stored separately and added to the password *before* hashing.
  - Provides an extra layer of protection, especially if the hash database is compromised.
- *Further Reading on Secure Password Storage:* OWASP Password Storage Cheat Sheet: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html.

CHAPTER 5: Security Assessment and Testing

1. Introduction to Security Assessment and Testing

- Cybersecurity professionals build, operate, and maintain controls to protect against threats (hackers, malicious code, social engineering).
- Regular security assessment and testing are vital to ensure controls function correctly and to identify exploitable vulnerabilities.
- This chapter covers vulnerability management, penetration testing, and cybersecurity exercises.

2. Vulnerability Management

- Modern technical environments (servers, endpoints, network devices) are complex, contain millions of lines of code, and process intricate configurations, making vulnerabilities inevitable and constantly emerging.
- Vulnerability management programs are crucial for identifying, prioritizing, and remediating vulnerabilities.
- They use vulnerability scanning to detect issues and implement a workflow to address high-priority vulnerabilities. Every organization needs one.

3. Identifying Scan Targets

- Organizations first determine if any regulatory requirements apply (e.g. PCI DSS, FISMA).
- System Identification Criteria:
  - Data Classification: What sensitivity level of data does the system store, process, or transmit?
  - Exposure: Is the system exposed to the Internet or other public/semi-public networks?
  - Services Offered: What functions does the system provide?
  - System Type: Is it a production, test, or development system?
- Automated Techniques for Inventory:
  - Cybersecurity professionals use scanning tools (like Qualys) to discover connected systems (known or unknown).
  - This helps build an asset inventory.
- Additional information about system type and data handled helps determine asset criticality (critical vs. noncritical).
- Asset inventory and criticality guide decisions on:
  - Types of scans to perform.
  - Frequency of scans.

- Priority for vulnerability remediation.

4. Determining Scan Frequency

- Vulnerability scanning tools allow for automated scheduling of scans (e.g. Tenable's Nessus).
- Scans should be configured to provide automated alerts for new vulnerabilities and email reports of scan results.
- Factors Influencing Scan Frequency:
  - Organization's risk appetite
  - Regulatory requirements
  - Technical constraints
  - Business constraints
  - Licensing limitations
- It's often advisable to start small with vulnerability scanning and gradually expand scope and frequency to avoid overwhelming infrastructure or systems.

5. Configuring Vulnerability Scans

- Vulnerability management solutions offer extensive parameters for scan configuration.
- Administrators can customize:
  - Scheduling automated scans.
  - Report generation.
  - Types of checks performed.
  - Credentials for target access.
  - Installation of scanning agents.
  - Network perspectives for scans.
- Regular configuration reviews of scanners are essential to ensure settings align with current requirements.

5.1. Scan Sensitivity Levels

- Careful attention to scan sensitivity settings is crucial. These settings dictate the checks performed.
- Goal is to customize to meet scan objectives while minimizing disruption to the target environment.
- Templates:
  - Administrators often start with vendor-provided templates (e.g. Nessus scan templates).
  - Custom organizational templates can also be developed and saved for efficient reuse, reducing errors and saving time.

- Optimizing Scan Efficiency (Plug-ins):
  - Plug-ins: Each performs a check for a specific vulnerability; grouped by OS, application, or device family.
  - Disabling unnecessary plug-ins (e.g. checks for unused operating systems like Amazon Linux) improves scan speed and reduces false positives.
- Intrusive Plug-ins:
  - Some plug-ins can disrupt or damage production systems.
  - Dilemma: They can identify critical, exploitable issues, but also pose risk.
  - Mitigation:
    - Limit production scans to non-intrusive plug-ins.
    - Maintain a test environment (copies of production systems) to run intrusive scans first. Fix issues in both test and production based on test results before scanning production.

5.2. Supplementing Network Scans

- Basic Network Scans (Remote/Non-credentialed):
  - Probe a system from a distance, simulating an external attacker's view.
  - Limitations: Results can be skewed by firewalls, IPS, and other security controls on the network path, leading to an inaccurate view of the *server's* inherent security.
  - Can also produce false positives because they may detect potential vulnerabilities but cannot confidently confirm them.
- Credentialed Scanning:
  - Mechanism: Administrators provide the scanner with credentials to connect directly to the target server.
  - Benefit: Allows the scanner to retrieve configuration information directly (e.g. patch levels for OS updates) to confirm vulnerabilities more accurately.
  - Credentialed scans *retrieve information* and typically do not make changes.
  - Enforce principle of least privilege by using a read-only account for the scanner to minimize risk.
  - Exam Focus: Understand differences between credentialed and non-credentialed scanning.
- Agent-Based Scanning:
  - Mechanism: Small software agents are installed on each target server.
  - Agents perform an "inside-out" vulnerability scan of the server's configuration and report findings to the central vulnerability management platform.
  - System administrators may be wary of agent installation due to potential performance or stability issues.

- Implement conservatively with a small pilot deployment to build confidence before widespread rollout.

5.3. Scan Perspective

- Comprehensive programs scan from various network locations to gain different views of vulnerabilities.
- Types of Perspectives:
  - External Scan: Run from the Internet; shows what an outside attacker would see.
  - Internal Scan: Run from the corporate network; shows what a malicious insider might see.
  - Datacenter Scanners/Agents: Offer the most accurate view of server's true state by bypassing network controls.
- Controls Affecting Scan Results:
  - Firewall settings
  - Network segmentation
  - Intrusion Detection Systems (IDSs)
  - Intrusion Prevention Systems (IPSs)
- Example (PCI DSS): Requires both internal (organization's own) and external (by an Approved Scanning Vendor - ASV) scans, illustrating different perspectives.
- Vulnerability management platforms can manage multiple scanners and consolidate results from different sources.

6. Scanner Maintenance

- Like all technology, vulnerability management solutions require regular maintenance to ensure effectiveness.
- Key Maintenance Tasks:
  - Keeping scanning software up-to-date.
  - Ensuring vulnerability feeds are current.
- Automated Updates:
  - Scanning systems usually have automatic updating capabilities for both the scanner software and vulnerability plug-in feeds (e.g. Nessus Automatic Updates).
  - Organizations *should* utilize these features, but manual verification periodically is a good practice to confirm proper updating.

6.1. Scanner Software

- Vulnerabilities in Scanners: Even vulnerability scanners themselves can have security flaws (e.g. Nessus vulnerability in NIST National Vulnerability Database).

- Patching: Regular patching of scanner software is critical for:
  - Protecting against scanner-specific vulnerabilities.
  - Providing important bug fixes.
  - Delivering feature enhancements to improve scan quality.

6.2. Vulnerability Plug-in Feeds

- Security researchers constantly discover new vulnerabilities. Scanners rely on frequent updates to their plug-ins to detect these new threats.
- Administrators should configure scanners to retrieve new plug-ins regularly, ideally daily.
- This update process is typically easily automated.

7. Security Content Automation Protocol (SCAP)

- A NIST-led initiative to standardize how security-related information is communicated, crucial for automating interactions between security components.
- Key SCAP Standards:
  - Common Configuration Enumeration (CCE): Standardizes nomenclature for system configuration issues.
  - Common Platform Enumeration (CPE): Standardizes naming for product names and versions.
  - Common Vulnerabilities and Exposures (CVE): Provides a standard nomenclature for describing security-related software flaws.
    - Exam Note: CompTIA refers to CVE as "Common Vulnerability Enumeration," but "Common Vulnerabilities and Exposures" is the industry-standard term. The purpose (standard naming for flaws) is what matters.
  - Common Vulnerability Scoring System (CVSS): Standardized approach for measuring and describing the severity of security flaws.
  - Extensible Configuration Checklist Description Format (XCCDF): Language for specifying checklists and reporting their results.
  - Open Vulnerability and Assessment Language (OVAL): Language for specifying low-level testing procedures used by checklists.
- *References:* NIST SP 800-126 Rev 3, SCAP website (https://csrc.nist.gov/projects/security-content-automation-protocol).

8. Vulnerability Scanning Tools

- Essential for preventive scanning, testing, and are used by both penetration testers (to identify exploitable systems) and attackers.
- Cybersecurity professionals should have a network vulnerability scanner, an application scanner, and a web application scanner.

8.1. Infrastructure Vulnerability Scanning (Network Scanners)

- Probe a wide range of network-connected devices for known vulnerabilities. They identify device types/configurations and run targeted tests.
- Examples:
  - Tenable Nessus: Well-known, widely respected, and one of the earliest products.
  - Qualys: Commercial SaaS-based scanner using appliances (on-premises and cloud).
  - Rapid7 Nexpose: Another commercial system similar to Nessus and Qualys.
  - OpenVAS: Free, open-source alternative.
- Defense-in-Depth: Many mature organizations deploy two different vulnerability scanning products for enhanced coverage.

8.2. Application Testing

- Analyzes custom-developed software for security vulnerabilities, integrated into the software development process.
- Techniques:
  - Static Testing: Analyzes code without executing it; provides direct vulnerability locations and remediation suggestions.
  - Dynamic Testing: Executes code and tests interfaces with various inputs to find vulnerabilities.
  - Interactive Testing: Combines static and dynamic methods, analyzing source code while interacting with the application.
- Integration: Often a requirement in software release processes, demanding "clean" tests before production deployment.

8.3. Web Application Scanning

- Specialized tools to examine the security of web applications.
- Tests for web-specific vulnerabilities (e.g. SQL injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF)).
- Combines traditional network scans of web servers with detailed probing of web applications using malicious input and fuzzing.

- Examples:
  - Nikto: Popular open-source command-line tool.
  - Arachni: Another open-source packaged scanner for Windows, macOS, Linux.
  - Commercial Products: Acunetix is a dedicated scanner; many use web application capabilities built into traditional network scanners like Nessus, Qualys, and Nexpose.

9. Reviewing and Interpreting Scan Reports

- Vulnerability scan reports provide detailed information for analyzing identified vulnerabilities.
- Key Sections of a Report:
  - Vulnerability Name & Severity: Descriptive title and overall severity (Low, Medium, High, Critical). *Example: "SSL Medium Strength Cipher Suites Supported," High severity.*
  - Description: Detailed explanation of the vulnerability and its flaws. *Example: Flaws in SSL protocol, no longer acceptable for use.*
  - Solution: Instructions on how to correct the vulnerability, often providing specific steps for administrators. *Example: Disable SSL 2.0/3.0; use TLS 1.2 or higher.*
  - "See Also" References: Links to external resources (blogs, documentation, IETF documents) for more details.
  - Output: Verbatim information returned by the remote system during the scan. Extremely valuable for analysts to understand *why* a vulnerability was reported, pinpoint its location, and identify false positives. *Example: Specific insecure ciphers being used.*
  - Port/Hosts: Details the affected server(s) (IP addresses often obscured) and specific services/ports where the vulnerability exists. *Example: Insecure SSL on ports 443 and 4433.*
  - Vulnerability Information: Miscellaneous relevant information, such as whether the vulnerability has appeared in news reports.
  - Risk Information: Includes overall risk factor (consistent with severity) and Common Vulnerability Scoring System (CVSS) scores.
    - *Example:* CVSS base score of 7.5 for the SSL vulnerability.
  - Vulnerability Scanner Plug-in Details: Information about the specific plug-in that detected the issue (ID, publication/update dates).

9.1. Understanding CVSS (Common Vulnerability Scoring System)

- An industry standard for assessing the severity of security vulnerabilities.
- Provides a numerical score (0-10) to prioritize response actions.

- Exam Note: Remember CVSS is a component of SCAP. It's a publicly available framework. Also, be familiar with CVE (Common Vulnerabilities and Exposures), which provides a list of publicly known vulnerabilities with IDs, descriptions, and references.
- Scoring Metrics (8 Measures, providing both descriptive rating and numeric score):
    - Exploitability Metrics (first four):
        - Attack Vector (AV): How an attacker exploits it.
            - Physical (P): 0.20 (physical access required)
            - Local (L): 0.55 (physical/logical access to system)
            - Adjacent (A): 0.62 (access to local network)
            - Network (N): 0.85 (remotely over network)
        - Attack Complexity (AC): Difficulty of exploitation.
            - High (H): 0.44 (specialized conditions needed)
            - Low (L): 0.77 (no specialized conditions)
        - Privileges Required (PR): Account access level needed by attacker.
            - High (H): 0.27 (admin privileges) or 0.50 if Scope is Changed
            - Low (L): 0.62 (basic user privileges) or 0.68 if Scope is Changed
            - None (N): 0.85 (no authentication needed)
        - User Interaction (UI): Whether user action is needed.
            - None (N): 0.85 (no user action beyond attacker)
            - Required (R): 0.62 (user action needed)
    - Impact Metrics (next three):
        - Confidentiality (C): Type of information disclosure.
            - None (N): 0.00
            - Low (L): 0.22 (some info, no attacker control)
            - High (H): 0.56 (all info compromised)
        - Integrity (I): Type of information alteration.
            - None (N): 0.00
            - Low (L): 0.22 (some info, no attacker control)
            - High (H): 0.56 (system integrity totally compromised)
        - Availability (A): Type of disruption.
            - None (N): 0.00
            - Low (L): 0.22 (system performance degraded)
            - High (H): 0.56 (system completely shut down)
    - Scope Metric (last one):

        - Scope (S): Whether vulnerability affects components beyond its immediate scope. (No direct score, impacts PR score).
            - Unchanged (U): Affects resources managed by same security authority.
            - Changed (C): Affects resources beyond the immediate security authority.
- CVSS Version: Current is 3.1 (minor update from 3.0); functionally equivalent for exam purposes. Older CVSSv2 uses different metrics/ratings.

9.2. Interpreting the CVSS Vector

- Format: Single-line string conveying ratings for all 8 metrics.
- *Example:* CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
    - CVSS:3.0: Version 3.0 used.
    - AV:N: Attack Vector: Network
    - AC:L: Attack Complexity: Low
    - PR:N: Privileges Required: None
    - UI:N: User Interaction: None
    - S:U: Scope: Unchanged
    - C:H: Confidentiality: High
    - I:N: Integrity: None
    - A:N: Availability: None

9.3. Calculating CVSS Base Score (for conceptual understanding, actual calculation not required for exam usually)

- Impact Sub-Score (ISS): Combines C, I, A metrics.
    - Formula: 1 - [(1 - C) * (1 - I) * (1 - A)]
    - *Example:* For SSL vulnerability (C:H, I:N, A:N): 1 - [(1 - 0.56) * (1 - 0.00) * (1 - 0.00)] = 0.56
    - Then, adjusted ISS calculation for actual Impact Score based on Scope.
- Exploitability Score: Combines AV, AC, PR, UI.
    - Formula: 8.22 * AV_score * AC_score * PR_score * UI_score
    - *Example:* For SSL vulnerability: 8.22 * 0.85 * 0.77 * 0.85 * 0.85 ≈ 3.9
- Base Score: Calculated using Impact and Exploitability scores, considering Scope.
    - If Impact is 0, Base Score is 0.
    - If Scope Unchanged: Impact Score + Exploitability Score
    - If Scope Changed: (Impact Score + Exploitability Score) * 1.08
    - Max score is 10.
- CVSS Calculator: NIST provides an online calculator (www.first.org/cvss/calculator/3.1); manual calculation is not expected for the exam.

9.4. Categorizing CVSS Base Scores (Qualitative Severity Rating Scale)

- Exam Note: Be familiar with this scale for the exam!
- 0.0: None
- 0.1–3.9: Low
- 4.0–6.9: Medium
- 7.0–8.9: High
- 9.0–10.0: Critical
- *Example:* SSL vulnerability with CVSS 7.5 falls into the High risk category.

10. Confirmation of Scan Results

- Cybersecurity analysts should always confirm vulnerabilities reported by a scanner.
- Methods:
  - Simple verification (e.g. checking if a patch is truly missing).
  - Complex manual processes that simulate exploits (e.g. attempting a SQL injection to confirm).
- Analysts should leverage their own expertise and the subject matter expertise of others (DBAs, system engineers, network techs, developers) to evaluate potential false positives.
- False Positives vs. False Negatives (Exam Focus!):
  - Positive Report: Scanner reports a vulnerability.
    - True Positive: The vulnerability genuinely exists (accurate).
    - False Positive: The vulnerability does *not* exist (inaccurate; scanner mistake due to insufficient access or plug-in error).
  - Negative Report: Scanner reports no vulnerability.
    - True Negative: No vulnerability exists (accurate).
    - False Negative: A vulnerability *does* exist but was not detected (inaccurate; a critical blind spot).
  - Exam Note: Understand these error types and be able to identify them in scenarios.

10.1. Reconciling Scan Results with Other Data Sources

- Vulnerability scans should not be analyzed in isolation.
- Valuable Information Sources for Reconciliation:
  - Log reviews: From servers, applications, network devices, etc., to find evidence of exploitation attempts.
  - Security Information and Event Management (SIEM) systems: Correlate logs from multiple sources to provide actionable intelligence.
  - Configuration management systems: Provide details on installed OS and applications.

- These sources help analysts reconcile scan reports with the actual computing environment.

11. Vulnerability Classification (Commonly Detected Vulnerabilities)

11.1. Patch Management

- Issue: Systems running outdated operating systems or applications lacking security patches.
- Solution: Implement a robust patch management program that routinely applies security updates at the OS, application, and firmware levels.

11.2. Legacy Platforms (End-of-Life/Unsupported Software)

- Issue: Continuing to run software for which the vendor has discontinued support.
- Risk: The vendor will no longer investigate or correct security flaws. Organizations are "on their own."
  - *Famous Example:* Microsoft Windows Server 2003 end of support in July 2015.
- Description in Report: Explicitly states "Lack of support implies that no new security patches...will be released."
- Solution:
  - Upgrade to a currently supported version.
  - If upgrade is impossible (e.g. due to legacy applications):
    - Isolate the system as much as possible (ideally no network connection).
    - Apply compensating security controls (e.g. increased monitoring, strict firewall rules).
- Exam Note: Good vulnerability response/remediation practices include patching, insurance, segmentation, compensating controls, exceptions, and exemptions.

11.3. Weak Configurations

- Issue: Systems configured with settings that introduce security risks.
- Examples:
  - Default settings (e.g. administrative setup pages left enabled in production).
  - Default credentials or unsecured accounts (lacking strong authentication, using default passwords, unsecured root accounts).
  - Open service ports not necessary for normal operations (systems should follow principle of least privilege for services).
  - Open permissions violating the principle of least privilege (allowing excessive user access).

11.4. Error Messages (Debug Mode)

- Issue: Application development platforms supporting debug modes that expose sensitive internal information (database structure, authentication mechanisms).

- Risk: Inadvertently assists attackers in gaining information.
- Scan Alert: Vulnerability scanners alert on the presence of debug mode.
- Solution: Disable debug modes on publicly exposed systems. Development should occur in isolated, private environments.

11.5. Insecure Protocols

- Issue: Use of older network protocols designed without security in mind, often lacking encryption for credentials and data.
- Risk: Eavesdropping attacks.
- Examples:
  - Telnet (cleartext command-line access).
  - File Transfer Protocol (FTP) (cleartext file transfer and authentication).
- Solution: Switch to secure alternatives:
  - SSH (Secure Shell) for Telnet.
  - SFTP (Secure File Transfer Protocol) or FTPS (FTP-Secure) for FTP.

11.6. Weak Encryption

- Issue: Using weak encryption algorithms or easily guessable encryption keys.
- Risk: Encryption can be easily defeated by attackers.
- Example: Support for insecure ciphers like RC4.
- Solution: Update systems to support strong, modern ciphers like AES (Advanced Encryption Standard).
- Further Details: Chapter 7 covers secure encryption implementation.

12. Penetration Testing Overview

- Authorized, legal attempts to defeat an organization's security controls and perform unauthorized activities.
- Bridge the gap between automated tools and real-world attacker capabilities.
- Benefits:
  - Provides a complete picture of security vulnerability that other methods cannot.
  - Offers unique visibility into the organization's security posture.
  - Complements and builds upon other cybersecurity activities.
  - Answers the question: "Can an attacker with similar skills successfully penetrate our defenses?"
  - Provides a blueprint for remediation by tracing attacker actions.
  - Offers focused information on specific attack targets (e.g. a new system deployment).

- Challenges: Time-consuming and requires highly skilled staff.

12.1. Adopting the Hacker Mindset

- Defender Mindset: Focus on protecting confidentiality, integrity, and availability against all possible threats.
- Hacker Mindset (Penetration Tester Mindset):
  - Focus on finding just one single vulnerability to exploit to achieve goals.
  - Think like the adversary.
  - *Analogy:* Instead of protecting every aspect of a store, find the single weakest link (e.g. unsecured window when the alarm is only on doors).
- Crucial Corollary: Cybersecurity defenders must "win every time," while attackers (and penetration testers) "need to win only once" for an attack to succeed.

12.2. Threat Hunting (Related Concept)

- Closely related to penetration testing but distinct. Threat hunters use the attacker mindset to search for artifacts of a *successful* attack already present in the infrastructure.
- Presumption of Compromise: Assumes attackers have *already* breached the organization.
- Process: Identify potential compromises, then initiate incident handling (contain, eradicate, recover), followed by postmortem analysis for remediation.
- Threat hunters combine information from threat feeds, security advisories, and other sources to trace attacker paths.

12.3. Penetration Test Types (Categories)

- Exam Note: Be able to differentiate and explain these four categories.
  - Physical Penetration Testing: Focuses on vulnerabilities in an organization's physical security controls (e.g. breaking into buildings, bypassing access controls, compromising surveillance).
  - Offensive Penetration Testing: Proactive approach where professionals act as attackers to identify and exploit vulnerabilities in networks, systems, applications. Goal: Simulate real attacks, assess detection/response/recovery.
  - Defensive Penetration Testing: Evaluates an organization's ability to defend against attacks. Focuses on assessing effectiveness of security policies, procedures, and technologies in detection and mitigation.
  - Integrated Penetration Testing: Combines aspects of both offensive and defensive testing for a comprehensive assessment. Involves close collaboration between teams.

12.4. Penetration Test Types (Knowledge Classifications)

- Exam Note: Be able to differentiate and explain these three classifications and identify them in scenarios.

- Known Environment Tests (White Box Testing):
  - Full knowledge of underlying technology, configurations, and settings (e.g. network diagrams, system lists, credentials).
  - Pros: Efficient testing, complete coverage of in-scope systems/services.
  - Cons: May not accurately reflect an external attacker's view; may bypass controls that would stop most attackers.
- Unknown Environment Tests (Black Box Testing):
  - No prior access or information about the environment. Testers must gather info like a real attacker.
  - Pros: Most accurate replication of a real-world external attack against unknown systems.
  - Cons: Time-consuming; heavily relies on the skill of the testing team to realistically simulate a threat actor.
- Partially Known Environment Tests (Gray Box Testing):
  - A blend; some information provided but not full access/details.
  - Pros: Can focus tester efforts (more efficient than black box) while still providing a more realistic attacker view than white box.

12.5. Rules of Engagement (RoE)

- A formal document defining the scope, expectations, and limitations of a penetration test. Crucial for legal and operational clarity.
- Key Elements:
  - Timeline: When testing occurs (e.g. non-critical hours or business hours to test reactions).
  - Scope: What targets are included/excluded (locations, systems, applications, third-party providers).
  - Technical Constraints: Any special technical limitations.
  - Data Handling: Requirements for sensitive information gathered (confidentiality, encryption, disposal).
  - Expected Behaviors: How the target's defenses might react (e.g. shunning, denylisting) and its impact on the test.
  - Resource Commitment: Time from administrators, developers, etc., needed for the test.
  - Legal Concerns: Review of applicable laws (organizational, remote locations, service providers).
  - Communication Plan: How and when communication will occur (updates, incident discovery).
- Always secure documented permission (signed agreement, memo from senior management) before any penetration testing to avoid legal issues ("get out of jail free card").

- Limitations: RoE should also define test limitations: valid only at the time conducted, scope/methodology impact comprehensiveness.
- Clear communication, notification, and escalation paths for potential outages or issues caused by the test. Document responsibilities and limitations of liability.

12.6. Reconnaissance (Phase 1 of Penetration Test)

- Gather as much information as possible about the target.
- Passive Reconnaissance:
  - Gathering information without directly engaging the target.
  - Examples: DNS/WHOIS lookups, web searches, reviewing public websites (OSINT).
- Active Reconnaissance:
  - Directly engaging the target in intelligence gathering.
  - Examples: Port scanning (identifying open ports), footprinting (identifying OS/applications), vulnerability scanning (identifying exploitable vulnerabilities).
- Exam Note: Know the difference between passive and active reconnaissance.
- War Driving/War Flying:
  - War Driving: Driving by facilities with antennas to eavesdrop on or connect to wireless networks.
  - War Flying: Using drones/UAVs for similar wireless network reconnaissance.

12.7. Running the Test (Key Phases - similar to Cyber Kill Chain)

- Initial Access: Exploiting a vulnerability to gain first entry into the network.
- Privilege Escalation: Shifting from initial access to higher privileges (e.g. root access on a system).
- Pivoting (Lateral Movement): Using a compromised system to gain access to other systems on the network.
- Persistence: Installing backdoors or mechanisms to regain access even if the initial vulnerability is patched.
- Tools: Testers use the same tools as attackers, including exploitation frameworks like Metasploit (simplifies configuring and deploying exploits).

12.8. Cleaning Up

- Post-Test Activities:
  - Present results to management.
  - Remove all installed tools and persistence mechanisms.
  - Provide a close-out report detailing vulnerabilities and advice for improving cybersecurity posture.

13. Audits and Assessments Overview

- Core Maintenance Activity: A comprehensive security assessment and testing program is vital for an information security team.
- Purpose is to regularly verify that an organization has:
  - Adequate security controls.
  - Properly functioning security controls.
  - Effectively safeguarding information assets.
- Three Major Components:
  1. Security Tests
  2. Security Assessments
  3. Security Audits

13.1. Security Tests

- Verify that a specific control is functioning properly.
- Include automated scans, tool-assisted penetration tests, and manual attempts to undermine security.
- Scheduling Factors for Security Managers:
  - Availability of testing resources.
  - Criticality of systems/applications.
  - Sensitivity of information.
  - Likelihood of technical failure.
  - Likelihood of misconfiguration.
  - Risk of system attack.
  - Rate of control configuration change.
  - Other environmental changes affecting control performance.
  - Difficulty and time required for the test.
  - Impact on normal business operations.
- Often involves frequent automated tests supplemented by infrequent manual tests (e.g. nightly automated vulnerability scans for credit card system + annual manual penetration test by external consultant).
- Security testing programs should be carefully designed and risk-prioritized, not haphazard.
- Reviewing Test Results:
  - Manual Review: Reading output and verifying success (some tests require human interpretation).

- Automated Review: Tools verify successful completion, log results, and alert/trigger actions (email, text, trouble ticket) for significant findings.

13.2. Responsible Disclosure Programs

- Allow security researchers to securely share vulnerability information with vendors.
- Goal is to foster collaboration between organizations and the security community for timely identification, reporting, and remediation.
- Benefits: Organizations benefit from external cybersecurity talent.
- Bug Bounty Programs:
  - A type of responsible disclosure program.
  - Incentivizes submissions by offering financial rewards ("bounties") for discovered vulnerabilities.
  - Outsiders will probe security regardless; formal programs provide incentive for disclosure rather than malicious exploitation.

13.3. Security Assessments

- A comprehensive review of the security of a system, application, or environment.
- Trained security professionals perform a risk assessment to identify vulnerabilities that could lead to compromise and recommend remediation.
- Goes beyond automated scanning and manual penetration tests. Includes thoughtful review of:
  - Threat environment.
  - Current and future risks.
  - Value of the targeted environment.
- Main Product: An assessment report for management, using non-technical language, with specific recommendations for security improvement.
- Conducted by: Internal teams or outsourced to third-party assessment teams.

13.4. Security Audits

- Formal examinations to demonstrate the effectiveness of controls to a third party.
- Key difference from assessments: Must be performed by independent auditors to ensure impartiality and unbiased view.
- Reports are for external audiences: board of directors, government regulators, other third parties.
- Primary Outcome: Attestation – a formal statement by auditors that controls are adequate and functioning properly.

13.5. Types of Audits

- Internal Audits:
  - Performed by: Organization's internal audit staff.

- Audience: Typically internal.
- Independence: Internal audit staff reports independently (e.g. to President, CEO, or directly to the Board/Audit Committee).
- Reasons: Management reassurance on compliance, leading self-assessments before external audits.

- External Audits:
  - Performed by: Outside auditing firm (independent third party).
  - Credibility: High degree of external validity (e.g. "Big Four" firms: Ernst & Young, Deloitte, PwC, KPMG).
  - Request Origin: Request comes from the organization or its governing body.
- Independent Third-Party Audits:
  - Subcategory of External Audits.
  - Performed by/on behalf of: Another organization (e.g. a regulatory body, a customer).
  - Request Origin: Request comes from a regulator, customer, or other outside entity.
  - Challenge for Service Providers: Can be burdensome if many clients require separate audits.
  - SSAE 18 (Statement on Standards for Attestation Engagements document 18):
    - Released by AICPA to alleviate the burden for service organizations.
    - Provides a common standard for auditors assessing service organizations.
    - Allows a single external assessment report to be shared with multiple clients.
    - Commonly referred to as Service Organization Controls (SOC) audits.

13.6. Auditing Standards

- Provide a clear description of control objectives that should be met, guiding the audit/assessment.
- Common Frameworks:
  - COBIT (Control Objectives for Information and related Technologies):
    - Describes common requirements for information systems controls.
    - Maintained by ISACA (creators of CISA, CISM certifications).
  - ISO 27001:
    - International standard for setting up an information security management system (ISMS).
    - Organizations can become officially certified as compliant.
  - ISO 27002:
    - Provides more detail on specific information security controls.

14. The Vulnerability Life Cycle

The vulnerability life cycle outlines the systematic process for managing security weaknesses within an environment.

1. Vulnerability Identification:
    - Discovering that a vulnerability exists within the organization's environment.
    - Sources of Identification:
        - Vulnerability scans (run internally or by external assessors).
        - Penetration tests (simulated attacks).
        - Reports from responsible disclosure or bug bounty programs.
        - Results of system and process audits.
2. Vulnerability Analysis:
    - Understanding the identified vulnerability and its potential impact.
    - Key Tasks:
        - Verifying the vulnerability is real and not a false positive.
        - Prioritization/Categorization: Using standardized tools for external assessment:
            - CVSS (Common Vulnerability Scoring System): Scores severity (0-10) for prioritization.
            - CVE (Common Vulnerabilities and Exposures): Standardized naming for known flaws.
        - Organizational Context: Supplementing external analysis with internal details like:
            - Exposure factor (how exposed the organization is).
            - Environmental variables (specific system details).
            - Industry and organizational impact.
            - Organization's risk tolerance.
3. Vulnerability Response and Remediation:
    - Addressing identified and prioritized vulnerabilities.
    - Response Options:
        - Patching: Applying corrective measures (e.g. software updates) to directly fix the vulnerability.
        - Network Segmentation: Isolating the affected system to reduce exploit probability.
        - Compensating Controls: Implementing other security measures (e.g. application firewalls, IPS) to reduce exploit likelihood if a patch isn't feasible immediately.
        - Insurance: Transferring the financial risk to a provider.

        - Exception/Exemption (Risk Acceptance): Formally accepting the risk associated with the vulnerability, typically after a formal review and approval process.

4. Validation of Remediation:
    - Confirming that the vulnerability has been successfully resolved.
    - Methods:
        - Rescanning: Re-running vulnerability scans on the affected system to verify the vulnerability no longer appears.
        - Audits: For serious vulnerabilities, internal or external auditors may validate the fix to provide independent assurance.
5. Reporting:
    - Communicating findings, actions, and lessons learned to relevant stakeholders.
    - Content:
        - Summary of identified, analyzed, and remediated vulnerabilities (initial severity, impact).
        - Details on remediation actions (patches, controls, risk acceptance).
        - Trends or patterns (recurring issues, susceptible systems).
        - Recommendations for improving the vulnerability management process, security policies, or training.
    - Benefit: Demonstrates commitment to cybersecurity and enables continuous improvement.