Security - Security is freedom from, or resilience against potential harm (or other unwanted coercive change) caused by others", Wikipedia.

We need to understand information
as an **asset**, which can be categorised into three main types

Pure information

For example, a social network dataset. A data scientist within an organisation can use this dataset to better understand relationships between their users. For example, the organization can work out who the 'influencers in a social group are and pay them to advertise their service.

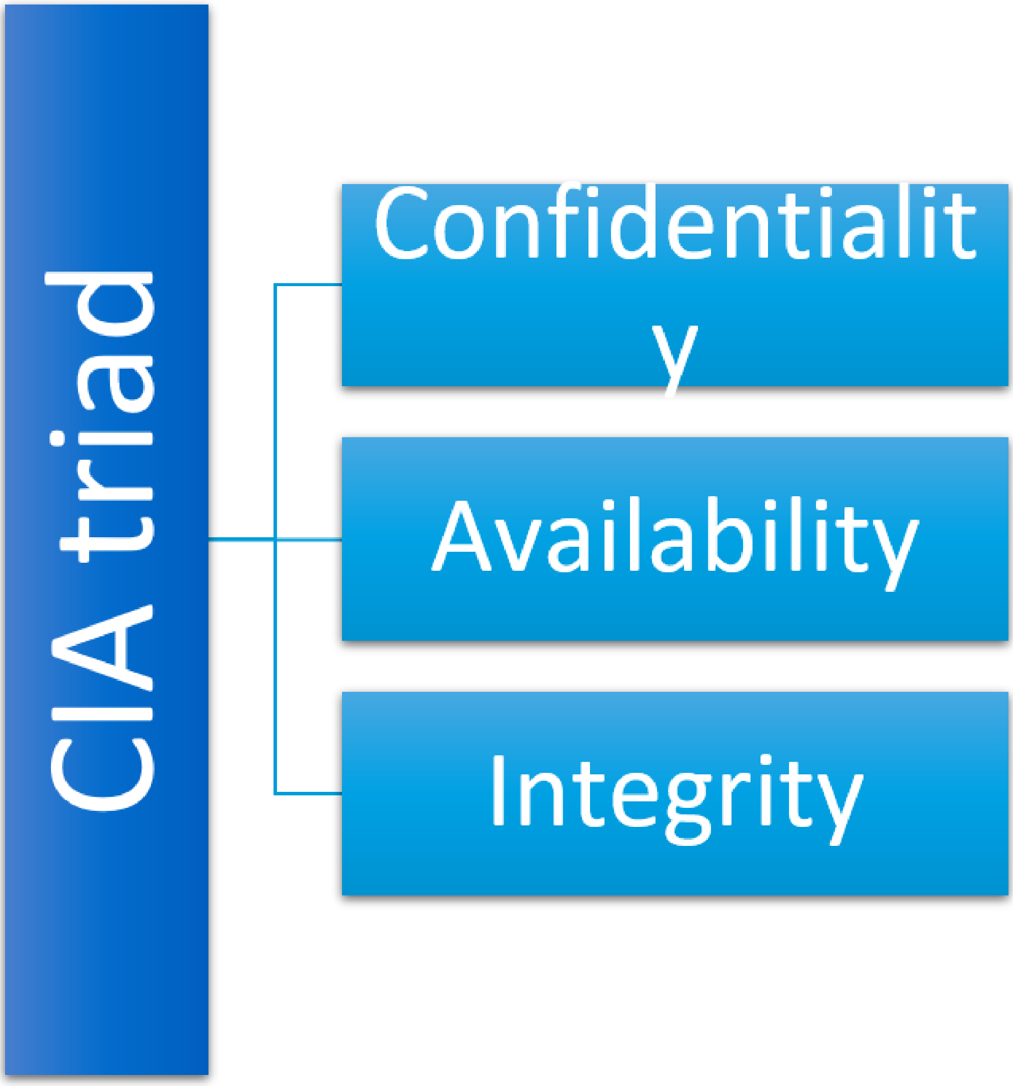
2. Physical assets

For example, computer systems. Hardware is expensive! An IBM mainframe worth $1m in 2003 was still worth $75k more than a decade and a half later! Hardware provides computation resources to both store and process your asset (pure information).

3. Software

This is used to process or manage information. Software is also expensive (for an example of this, look at the current price of Inventor Pro), and is also going to read and process your information!

We need a guarantee of its integrity (ie no backdoors) and that it can keep the information confidential.



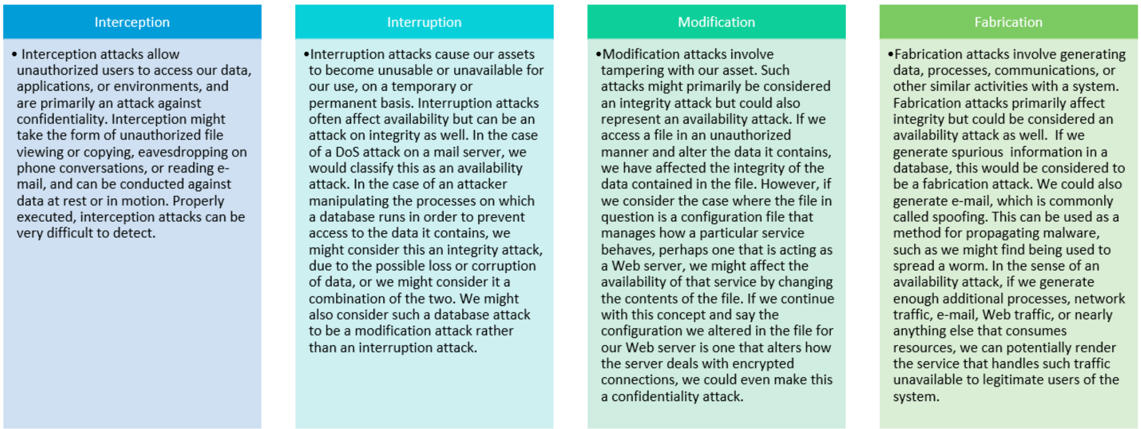
Confidentiality is a concept similar to, but not the same as, privacy. Confidentiality is a necessary component of privacy and refers to our ability to protect our data from those who are not authorized to view it. Confidentiality is a concept that may be implemented at many levels of a process.

Integrity refers to the ability to prevent our data from being changed in an unauthorized or undesirable manner. This could mean the unauthorized change or deletion of our data or portions of our data, or it could mean an authorized, but undesirable, change or deletion of our data. To maintain integrity, we not only need to have the means to prevent unauthorized changes to our data but also need the ability to reverse authorized changes that need to be undone.

Availability refers to the ability to access our data when we need it. Loss of availability can refer to a wide variety of breaks anywhere in the chain that allows us access to our data. Such issues can result from power loss, operating system or application problems, network attacks, compromise of a system, or other problems. When such issues are caused by an outside party, such as an attacker, they are commonly referred to as a denial of service (DoS) attack.

- **Attack:** An intentional or unintentional act that can cause damage to or otherwise compromise information and/or the systems that support it.

Types of attack payloads



Interception

- Interception attacks allow unauthorized users to access our data, applications, or environments, and are primarily an attack against confidentiality. Interception might take the form of unauthorized file viewing or copying, eavesdropping on phone conversations, or reading e-mail, and can be conducted against data at rest or in motion. Properly executed, interception attacks can be very difficult to detect.

Interruption

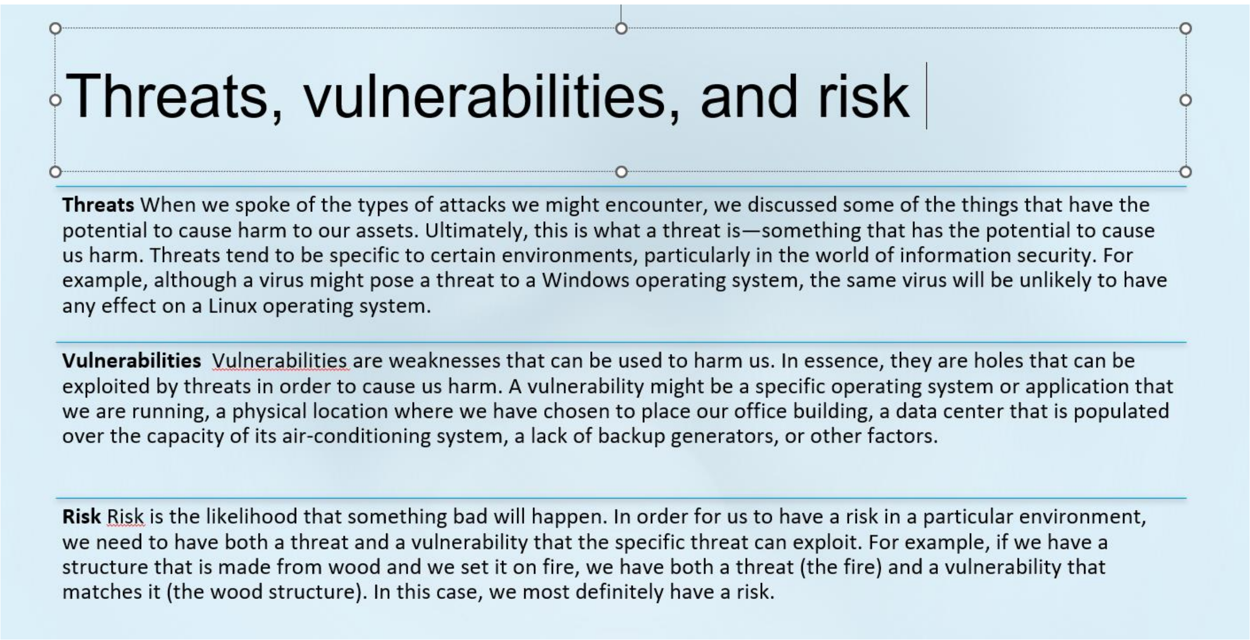
- Interruption attacks cause our assets to become unusable or unavailable for our use, on a temporary or permanent basis. Interruption attacks often affect availability but can be an attack on integrity as well. In the case of a DoS attack on a mail server, we would classify this as an availability attack. In the case of an attacker manipulating the processes on which a database runs in order to prevent access to the data it contains, we might consider this an integrity attack, due to the possible loss or corruption of data, or we might consider it a combination of the two. We might also consider such a database attack to be a modification attack rather than an interruption attack.

Modification

- Modification attacks involve tampering with our asset. Such attacks might primarily be considered an integrity attack but could also represent an availability attack. If we access a file in an unauthorized manner and alter the data it contains, we have affected the integrity of the data contained in the file. However, if we consider the case where the file in question is a configuration file that manages how a particular service behaves, perhaps one that is acting as a Web server, we might affect the availability of that service by changing the contents of the file. If we continue with this concept and say the configuration we altered in the file for our Web server is one that alters how the server deals with encrypted connections, we could even make this a confidentiality attack.

Fabrication

- Fabrication attacks involve generating data, processes, communications, or other similar activities with a system. Fabrication attacks primarily affect integrity but could be considered an availability attack as well. If we generate spurious information in a database, this would be considered to be a fabrication attack. We could also generate e-mail, which is commonly called spoofing. This can be used as a method for propagating malware, such as we might find being used to spread a worm. In the sense of an availability attack, if we generate enough additional processes, network traffic, e-mail, Web traffic, or nearly anything else that consumes resources, we can potentially render the service that handles such traffic unavailable to legitimate users of the system.



Threats, vulnerabilities, and risk

Threats When we spoke of the types of attacks we might encounter, we discussed some of the things that have the potential to cause harm to our assets. Ultimately, this is what a threat is—something that has the potential to cause us harm. Threats tend to be specific to certain environments, particularly in the world of information security. For example, although a virus might pose a threat to a Windows operating system, the same virus will be unlikely to have any effect on a Linux operating system.

Vulnerabilities Vulnerabilities are weaknesses that can be used to harm us. In essence, they are holes that can be exploited by threats in order to cause us harm. A vulnerability might be a specific operating system or application that we are running, a physical location where we have chosen to place our office building, a data center that is populated over the capacity of its air-conditioning system, a lack of backup generators, or other factors.

Risk Risk is the likelihood that something bad will happen. In order for us to have a risk in a particular environment, we need to have both a threat and a vulnerability that the specific threat can exploit. For example, if we have a structure that is made from wood and we set it on fire, we have both a threat (the fire) and a vulnerability that matches it (the wood structure). In this case, we most definitely have a risk.

CONTENTS

A good general definition to start from would be the following:

“Security Testing is the process to reveal flaws in a system in terms of Confidentiality, Integrity, Availability (CIA)”

As you can imagine, there is no agreement on a universal definition of Security Testing.

Indeed, there are various types of activities that fall under the umbrella of security testing:

- Security Audit
- Vulnerability Assessment
- Penetration Testing
- Red Teaming
- Ethical Hacking

02

Pentesting

A penetration testing (pentest) is an authorized simulated cyberattack on a computer system, performed to evaluate the security of the system.

Important note: Pentest, Red Teaming and Vulnerability Assessment (VA) are different activities.

Goal of the Pentest

- Main objective: The goal of a Pentest is to simulate real-world cyber attacks to identify and mitigate potential security risks.

- Identify vulnerabilities and weaknesses in the system's defenses.
- Improve the overall security posture by addressing discovered vulnerabilities.

Vulnerability

assesment 03

- If we imagine that system that we are testing is the House with safe:

Pentest vs VA ✓ Pentest is activity that is performed to get content of the safe;

✓ VA is activity that performed to find all unsecured windows/doors and find all other ways thief can get into the house and to the safe.

04 RED TEAM

Red Team

The Red Team is a group of security experts who simulate real-world cyber-attacks to test the defenses of a system or organization.

- Objective: Find and exploit vulnerabilities in the system as an external threat.
- Skills required: Advanced knowledge of hacking techniques, creativity, and the ability to think like an attacker.

Work in collaboration with the Blue Team for a comprehensive security approach.

Red Team Engagement

- Red teaming is a term borrowed from the military. In military exercises, a group would take the role of a red team to simulate attack techniques to test the reaction capabilities of a defending team, generally known as blue team, against known adversary strategies. Translated into the world of cybersecurity, red team engagements consist of emulating a real threat actor's Tactics, Techniques and Procedures (TTPs) so that we can measure how well our blue team responds to them and ultimately improve any security controls in place.
- The red team will do everything they can to achieve the goals while remaining undetected and evading any existing security mechanisms like firewalls, antivirus, EDR, IPS and others. Notice how on a red team engagement, not all of the hosts on a network will be checked for vulnerabilities. A real attacker would only need to find a single path to its goal and is not interested in performing noisy scans that the blue team could detect.

Red Team Kill Chain

Pentesting vs Red Teaming

	Pentesting	Red Teaming
Security Assessment	Methodical	Flexible
Scope	• Restrictive Scope • 1-2 weeks engagement • Generally Announced • Identify Vulnerabilities	• No Rules* • 2 weeks - 6 months engagement • No announcement • Test Blue teams on programs, policies, tools, and skills • Useful to estimate organization's Time To Detect (TTD) and Time To Mitigate (TTM)

* Can't be illegal…

Table Source: Peter Kim, "The Hacker Playbook 3"

Blue team   05

Blue team

The Blue Team is responsible for defending against simulated cyber-attacks conducted by the Red Team.

- Objective: Detect, respond, and mitigate attacks to strengthen the overall security posture.
- Skills required: Strong understanding of defensive strategies, incident response, and security technologies.

Work in collaboration with the Red Team for a comprehensive security approach.

Purple Team

- The Purple Team is a collaborative approach that involves both the Red and Blue Teams working together.
- Real-time sharing: Information and feedback are shared to enhance the overall security by improving detection and response capabilities.

How to conduct

the Pentest? 06

How to conduct the Pentest?

- Time to talk about the phase that is usually called ‘Pre-engagement’. There is a bunch of different methodologies, but the idea is the same: agree upon the rules/scope/schedule/etc. of the engagement and record it in some document.
- Rules of engagement is a formalized document that is usually signed by both parties (Customer and Company that perform security testing).

Pre-engagement is about asking questions. More questions you ask – less problems you get in the future. You should agree with Customer on following points:

✓ The goal of the security test

✓ Scope of the engagement

✓ Schedule (milestones)

✓ Risks

✓ The allowed techniques

✓ Deliverables

✓ Statement of work

What information should I get?

Scope of the engagement

We call “Scope of the engagement” as a list of activities you will perform (e.g. list of checks or OWASP Top 10).

Also, allowed surface of attack is mentioned here. It might be done with different ways: domain (ask about subdomains), IP, etc.

Understand the surface of attack means resolve all questions concerning environment (ensure environment is available etc.).

Important note: testing out of surface is illegal!

Schedule

✓ Start date of security test

✓ Ensure timetable allowed hours (some project switch off environments for a night)

✓Timetable for possible scans if needed (e.g.: “Please do not switch off environment on some weekend”)

✓ End date

Risks

- Some of activities might cause denial-of-service, loss of the data or slower the work of other people who share the environment. That’s why it is a good idea to conduct security test on a separate environment.
- Also, you need to know who is the person that might help you in critical situation (contact person).

07 Methodologies

Methodologies

Pentest methodologies can be customized based on the specific system, goals, and industry standards.

Common pentesting methodologies:

1. OWASP Testing Guide

2. NIST SP 800-115
3. OSSTMM (Open Source Security Testing Methodology Manual)
4. PTES (Penetration Testing Execution Standard)

OWASP Testing Guide

- The Web Security Testing Guide (WSTG) Project produces the premier cybersecurity testing resource for web application developers and security professionals.
- The WSTG is a comprehensive guide to testing the security of web applications and web services. Created by the collaborative efforts of cybersecurity professionals and dedicated volunteers, the WSTG provides a framework of best practices used by penetration testers and organizations all over the world.

- The OWASP Top 10 is a standard awareness document for

-

  developers and web application security. It represents a broad

  consensus about the most critical security risks to web applications.

  Companies should adopt this document and start the process of

  OWASP Top 10

  ensuring that their web applications minimize these risks.

PTES (Penetration Testing Execution Standard)

Penetration Testing Execution Standard (PTES) defines penetration testing as 7 phases. Particularly, PTES Technical

Guidelines give hands-on suggestions on testing procedures, and recommendation for security testing tools.

1 Pre-engagement Interactions

2 Intelligence Gathering

3 Threat Modeling

4 Vulnerability Analysis

5 Exploitation

6 Post Exploitation

7 Reporting

Summary of the Presentation: Information Security Threats and Vulnerabilities

This presentation provides an overview of various threats, vulnerabilities, and attacks in information security, as well as strategies for mitigating these risks. Below are the key takeaways:

---

1. Types of Threats

- Phishing: A technique used to trick individuals into revealing sensitive information, such as passwords or credit card numbers, often via fake emails or websites.
- Malware: Malicious software designed to disrupt or damage systems, including:
  - Viruses: Require user interaction to spread.
  - Worms: Spread automatically without user interaction.
  - Trojans: Disguised as legitimate programs but contain harmful components.
  - Spyware: Focused on stealing personal details such as logins and passwords.
  - Ransomware: Encrypts files and demands a ransom for decryption.

---

2. Types of Attacks

- Brute-Force Attack: Testing all possible password combinations until the correct one is found.

- Dictionary Attack: A variant of brute-force attacks using a predefined list of common words and passwords.

- Rainbow Table Attack: Precomputed tables of hash values to crack passwords more efficiently.

- Password Cracking Tools: Tools like Hashcat, John the Ripper, and THC Hydra are widely used to execute these attacks.

---

3. Weak Passwords

- Examples of Weak Passwords:
  - Default passwords (e.g., "admin," "password").
  - Simple words with numbers appended (e.g., "password1").
  - Obfuscated words (e.g., "p@ssw0rd").
  - Personal information (e.g., birthdays, pet names).
  - Common patterns (e.g., "123456," "qwerty").
- Consequences: Weak passwords are easy to guess or crack using automated tools.

---

4. Password Security

- Best Practices:
  - Use long passwords with a combination of letters, numbers, and special characters.
  - Avoid personal details, dictionary words, and predictable patterns.
  - Change default passwords immediately.
- Password Managers:
  - Tools like LastPass, BitWarden, and iCloud KeyChain securely store and generate strong passwords.
  - Integration with two-factor authentication (2FA) enhances security.

---

5. Password-Cracking Tools

- Hashcat: Supports cracking multiple password types across multiple devices.

- John the Ripper: Open-source tool for cracking passwords in systems and web applications.

- Brutus: Specializes in online password attacks.

- Wfuzz: Web application password-cracking tool that also identifies vulnerabilities.

- THC Hydra: A versatile tool for online password guessing across multiple protocols.

- Medusa: A modular brute-force tool with parallelized attacks.

---

6. Protecting Against Attacks

- Mitigation Strategies:

  - Implement two-factor authentication (2FA) for added security.

  - Educate users on phishing and social engineering tactics.

  - Regularly update software and systems to patch vulnerabilities.

  - Use strong and unique passwords for each service.

  - Monitor for unauthorized access attempts and respond promptly.

---

7. Importance of Password Security

- Impact of Weak Passwords:

  - Compromised systems, data breaches, and financial losses.

- Advantages of Strong Password Practices:

  - Reduced risk of unauthorized access.

  - Enhanced protection of sensitive data.

Summary of the Presentation: Cryptography

This presentation provides an in-depth overview of cryptography, covering its principles, methods, and applications. Below are the key highlights:

---

1. What is Cryptography?

- Definition:
  - Cryptography: The science of secret writing.
  - Cryptanalysis: The science of recovering plaintext from ciphertext without a key.
  - Cryptology: The study of both cryptography and cryptanalysis.
  - Steganography: The science of hiding messages within other messages.
- Purpose: To transform insecure communication channels, like the internet, into secure ones by ensuring properties like confidentiality, integrity, authentication, and non-repudiation.

---

2. Key Cryptographic Concepts

- Plaintext: Readable text before encryption.
- Ciphertext: Encrypted text that requires decryption to be understood.
- Encryption: The process of transforming plaintext into ciphertext using a key.
- Decryption: The reverse process of turning ciphertext back into plaintext using a key.

---

3. Types of Ciphers

1. Substitution Ciphers:
   - Replace each letter in the plaintext with another letter based on a key.
   - Examples:
     - Caesar Cipher: Shifts each letter by three places.
     - Monoalphabetic Cipher.
     - Polyalphabetic Cipher.
2. Transposition Ciphers:
   - Rearrange the order of letters without altering them.
   - Examples:
     - Columnar Transposition.

  - Rail Fence Cipher.

---

4. Encoding

- Definition: Converting data from one form to another in a reversible way.
- Examples:
  - URL Encoding: Encodes special characters in a URL.
  - Base64 Encoding: Converts binary data to text for internet transmission.

---

5. Encryption

- Definition: Transforming data to ensure only authorized parties can access it.
- Types:
  - Symmetrical Encryption: Uses one key for both encryption and decryption.
  - Asymmetrical Encryption: Uses a pair of keys—public and private.
- Example Algorithm: Data Encryption Standard (DES).
  - Operates on 64-bit blocks using a 56-bit key.
  - Employs 16 rounds of permutation and substitution.

---

6. Hashing

- Definition: A one-way transformation of data into a fixed-length value.
- Properties of a Secure Hash Function:
  - Deterministic.
  - Quick to compute.
  - Collision resistant.
  - Displays an avalanche effect (small input changes cause significant output changes).
  - Resistant to pre-image attacks.
- Common Algorithms:

  - MD5, SHA-1, SHA-256, and SHA-512.

---

7. Applications of Hashing

- Ensuring data integrity.
- Storing passwords securely.
- Creating digital signatures.
- Data indexing and retrieval.

---

8. Platforms for Cryptography Training

- Cryptohack: Interactive challenges for learning cryptographic concepts.
- CryptoPals: Hands-on exercises covering encryption and decryption.
- OverTheWire - Narnia: Challenges focused on exploiting cryptographic vulnerabilities.
- HackerRank - Cryptography Challenges: Tasks on encryption, decryption, and cryptographic operations.

Summary of the Presentation: Cryptography

This presentation provides an in-depth overview of cryptography, covering its principles, methods, and applications. Below are the key highlights:

---

1. What is Cryptography?

- Definition:
  - Cryptography: The science of secret writing.
  - Cryptanalysis: The science of recovering plaintext from ciphertext without a key.
  - Cryptology: The study of both cryptography and cryptanalysis.
  - Steganography: The science of hiding messages within other messages.
- Purpose: To transform insecure communication channels, like the internet, into secure ones by ensuring properties like confidentiality, integrity, authentication, and non-repudiation.

---

2. Key Cryptographic Concepts

- Plaintext: Readable text before encryption.
- Ciphertext: Encrypted text that requires decryption to be understood.
- Encryption: The process of transforming plaintext into ciphertext using a key.
- Decryption: The reverse process of turning ciphertext back into plaintext using a key.

---

3. Types of Ciphers

1. Substitution Ciphers:
   - Replace each letter in the plaintext with another letter based on a key.
   - Examples:
     - Caesar Cipher: Shifts each letter by three places.
     - Monoalphabetic Cipher.
     - Polyalphabetic Cipher.
2. Transposition Ciphers:
   - Rearrange the order of letters without altering them.
   - Examples:
     - Columnar Transposition.
     - Rail Fence Cipher.

---

4. Encoding

- Definition: Converting data from one form to another in a reversible way.
- Examples:
  - URL Encoding: Encodes special characters in a URL.
  - Base64 Encoding: Converts binary data to text for internet transmission.

---

5. Encryption

- Definition: Transforming data to ensure only authorized parties can access it.
- Types:
  - Symmetrical Encryption: Uses one key for both encryption and decryption.
  - Asymmetrical Encryption: Uses a pair of keys—public and private.
- Example Algorithm: Data Encryption Standard (DES).
  - Operates on 64-bit blocks using a 56-bit key.
  - Employs 16 rounds of permutation and substitution.

---

6. Hashing

- Definition: A one-way transformation of data into a fixed-length value.
- Properties of a Secure Hash Function:
  - Deterministic.
  - Quick to compute.
  - Collision resistant.
  - Displays an avalanche effect (small input changes cause significant output changes).
  - Resistant to pre-image attacks.
- Common Algorithms:
  - MD5, SHA-1, SHA-256, and SHA-512.

---

7. Applications of Hashing

- Ensuring data integrity.
- Storing passwords securely.
- Creating digital signatures.
- Data indexing and retrieval.

---

8. Platforms for Cryptography Training

- Cryptohack: Interactive challenges for learning cryptographic concepts.
- CryptoPals: Hands-on exercises covering encryption and decryption.
- OverTheWire - Narnia: Challenges focused on exploiting cryptographic vulnerabilities.
- HackerRank - Cryptography Challenges: Tasks on encryption, decryption, and cryptographic operations.

Summary of the Presentation: Cyber Reconnaissance

This presentation provides a comprehensive overview of cyber reconnaissance, an essential phase in cybersecurity testing and attacks. Below are the key highlights:

---

1. What is Cyber Reconnaissance?

- Definition:
  - Reconnaissance is the first phase of any cyberattack or security test.
  - The objective is to gather information about the target, its vulnerabilities, and its defense systems.
- Importance:
  - It sets the foundation for subsequent attack or testing phases by identifying weak points and entry paths.

---

2. Types of Cyber Reconnaissance

1. Passive Reconnaissance:
   - Gathering information without directly interacting with the target.
   - Relies on publicly available information, often referred to as Open-Source Intelligence (OSINT).
   - Common data sources include:
     - Infrastructure details (e.g., network blocks, technology stacks).
     - Public-facing information (e.g., IPs, domains, subdomains, and metadata from documents).
     - Job postings (which can reveal the technology stack used by the organization).

    - WHOIS data for domain registration details.
    - Tools:
      - WHOIS: Provides domain ownership and registration information.
      - nslookup: Queries DNS records to retrieve domain and IP details.
      - Netcraft: Gathers information on web servers, subdomains, and uptime stats.
      - Wappalyzer: Analyzes web technologies and frameworks used by a website.

2. Active Reconnaissance:
   - Direct interaction with the target to extract information.
   - Higher risk due to potential exposure of the attacker or tester.
   - Tools:
     - Nmap:
       - A versatile tool for network discovery and security auditing.
       - Identifies open ports, services, and operating systems.
     - Subdomain enumeration tools (e.g., DNSMap, Burp Suite, **OWASP Zap**).

---

3. Open-Source Intelligence (OSINT)

- Definition: The process of collecting and analyzing publicly available data to gather actionable intelligence.
- Entry Points:
  - Physical (e.g., building layouts from public resources).
  - Electronic (e.g., exposed IPs or domains).
  - Human (e.g., employees whose personal data may be exploited).
- Common Uses:
  - Identifying entry points into an organization.
  - Collecting personal information for social engineering.

---

4. Tools and Techniques

- WHOIS: Identifies domain ownership, creation dates, and contact details.

- nslookup: Retrieves DNS records for a domain or IP.

- Netcraft: Provides information on server details, subdomains, and hosting services.

- Wappalyzer: Detects web technologies and frameworks used by a target site.

- Nmap: A powerful tool for identifying live hosts, open ports, and running services.

---

5. Infrastructure Insights

- Understanding the infrastructure behind web applications helps identify vulnerabilities.

- Example:

  - Web server types (e.g., Apache, Microsoft IIS, nginx) reveal underlying operating systems.

  - Subdomain enumeration widens the attack surface and may expose hidden services like admin panels.

---

6. Subdomain Enumeration

- Identifying all subdomains within a domain expands the attack surface.

- Methods:

  - Search engines with advanced operators (e.g., `site:target.com`).

  - Tools like Netcraft, dnsenum, subbrute, and Harvester.

---

7. Risks and Limitations

- Passive Reconnaissance:

  - Less risky but limited to publicly available data.

- Active Reconnaissance:

  - Provides deeper insights but carries a higher risk of exposure.

Summary of the Presentation: Authentication and Authorization

This presentation provides a detailed overview of authentication and authorization, their processes, associated vulnerabilities, and best practices. Below are the key points:

---

1. What is Authentication and Authorization?

- Authentication: The process of confirming a user's identity (e.g., username and password).
- Authorization: Determines what actions the authenticated user is allowed to perform.
- Example:
  - Authentication verifies "Who you are."
  - Authorization verifies "What you can do."

---

2. Authentication Factors

- Three Types:
  1. Knowledge Factor: Something the user knows (e.g., password, PIN, secret question).
  2. Ownership Factor: Something the user has (e.g., security token, mobile phone).
  3. Inherence Factor: Something the user is or does (e.g., fingerprint, voice, retinal pattern).

---

3. Authentication Methods

- Single-Factor Authentication:
  - Requires one authentication factor, typically a password.
- Multi-Factor Authentication (MFA):
  - Requires two or more authentication factors (e.g., password and OTP).

---

4. Access Control Models

- Discretionary Access Control (DAC):
  - Least restrictive; users manage access to resources (e.g., sharing files on Google Drive).
- Mandatory Access Control (MAC):

  - System administrators manage access based on predefined rules.

- Role-Based Access Control (RBAC):

  - Access is determined by a user's role within an organization.

- Rule-Based Access Control:

  - Access depends on predefined rules, such as group memberships.

---

5. Common Vulnerabilities

- Unencrypted Credentials:

  - Sensitive data transmitted over unencrypted channels can be intercepted (e.g., Man-in-the-Middle attacks). Use HTTPS for protection.

- Weak Password Policies:

  - Simple passwords are vulnerable to brute-force and dictionary attacks.

- Default Credentials:

  - Many systems come with default usernames and passwords (e.g., admin/admin), which attackers can exploit if not changed.

- User Enumeration:

  - Specific error messages (e.g., "Invalid username") can reveal the existence of user accounts.

---

6. Mitigation Techniques

- Strong Password Policies:

  - Passwords should include a mix of uppercase, lowercase, numbers, and special characters.

  - Avoid reusing passwords.

  - Minimum recommended length: 12 characters (per OWASP ASVS).

- Password Storage:

  - Store passwords in a hashed and salted format, never as plain text.

- Lockout Mechanisms:

  - Introduce delays or CAPTCHAs after failed login attempts.

  - Lock accounts temporarily after multiple failed attempts.

;*7. Session Management**

- Secure Session IDs:
  - Must be random, unpredictable, and time-limited.
- "Remember Me" Functionality:
  - Avoid storing credentials in cookies or browser storage.
  - Use encrypted tokens with strict expiration times.

---

8. Vulnerabilities in Authorization

- Broken Access Control:
  - Users access resources or perform actions they are not authorized for (e.g., modifying URLs to access restricted files).
- Incorrect Redirection:
  - Sensitive data might be exposed due to improper handling of redirects.
- Weak Session Management:
  - Predictable session tokens can allow attackers to impersonate users.

---

9. CAPTCHA and Password Reset

- CAPTCHA:
  - Prevents automated brute-force attacks but is not foolproof.
- Password Reset Features:
  - Use secure mechanisms like sending reset links to verified email addresses.
  - Secret questions should have strong, hard-to-guess answers.

---

10. Bypassing Authentication and Authorization

- Techniques:

  - SQL or command injections to bypass authentication.

  - Path traversal attacks to access unauthorized resources.

- Defense:

  - Validate inputs rigorously and implement proper access control checks.

Summary of the Presentation: Incident Response and Management

1. What is an Incident?

- Definition:

  - A cybersecurity incident is any event, accidental or deliberate, that compromises the confidentiality, integrity, or availability of IT resources.

  - Includes activities like data breaches, policy violations, criminal use of technology (e.g., fraud or theft).

---

2. What is Incident Response?

- Definition:

  - Incident response is the process of managing data breaches or cyberattacks by:

    - Quickly identifying the issue.

    - Minimizing damage.

    - Containing the threat.

    - Preventing future occurrences.

---

3. Approaches to Incident Response

- PICERL Model (SANS):

  - Preparation: Define roles, train responders, establish policies.

  - Identification: Detect and verify incidents.

  - Containment: Stabilize the environment to prevent further damage.

  - Eradication: Remove the cause of the incident (e.g., malware).

  - Recovery: Restore systems and services to normal operations.

- Lessons Learned: Analyze and improve the process.

- SOAR (Security Orchestration, Automation, and Response):
  - Automates incident response tasks using integrated software tools.

- DAIR (Dynamic Approach to Incident Response):
  - Focuses on milestones like preparation, detection, verification, and triage in a cyclical process.

---

4. Handling an Incident

- Preparation Phase:
  - Develop IR policies and procedures.
  - Define roles (team leader, investigator, communication manager).
  - Conduct regular training and risk analysis.

- Detection and Analysis:
  - Monitor logs, network traffic, and IDS alerts.
  - Identify Indicators of Compromise (IoCs), such as unusual CPU usage, unauthorized file changes, or cleared logs.

- Containment:
  - Actions to limit damage:
    - Isolate compromised systems.
    - Disable services.
    - Blacklist suspicious IPs.

- Eradication:
  - Remove malware, patch vulnerabilities, and reset compromised user accounts.

- Recovery:
  - Restore systems to normal operations and improve security controls.

- Lessons Learned:

** - Conduct a post-incident review to identify areas of improvement.**

---

5. Communication in Incident Response

- Create a communication plan:

** - Define how to communicate internally and externally.**

** - Prepare contact lists for stakeholders (e.g., management, IT, legal, regulators).**

** - Coordinate with authorities in cases of cybercrime.**

---

6. Training and Exercises

- Periodic training ensures teams are familiar with:

** - Security procedures.**

** - Communication protocols.**

** - IR tools and techniques.**

- Regular exercises test the effectiveness of response plans and update contact lists.

---

7. Common Incident Indicators

- Unusual network activity.

- High CPU usage or memory consumption.

- Unexpected file changes or deleted logs.

- Suspicious IP connections or login attempts.

---

8. Post-Incident Activities

- Documentation:

- Record every step of the IR process with timestamps.
  - Create a timeline of findings and actions taken.
- Root Cause Analysis:
  - Identify how the incident occurred and implement measures to prevent recurrence.

Summary of the Presentation: Social Engineering and Countermeasures

1. What is Social Engineering?

- Definition:
  - Social engineering is the psychological manipulation of people to reveal confidential or sensitive information or perform specific actions, such as:
    - Opening an infected email attachment.
    - Clicking a malicious link.
    - Providing sensitive information over a phone call or email.
  - It does not require technical knowledge but relies on human psychology and social dynamics.

- Key Features:
  - Exploits human weaknesses.
  - Often uses a deep understanding of the victim's context and background.
  - Famous practitioner: Kevin Mitnick, a former social engineer turned cybersecurity consultant.

---

2. Phases of Social Engineering

1. Reconnaissance:
   - The attacker gathers as much information as possible about the victim or organization using Open Source Intelligence (OSINT) tools.
   - Activities include:
     - Email harvesting.
     - Identifying roles and contact information.
     - Understanding the victim's position in the organization.
     - Building a psychological profile based on personal interests or behaviors.

2. Victim Approach:

- **The attacker contacts the victim using various channels:**
  - **Phone calls (vishing).**
  - **Emails (phishing or spear phishing).**
  - **Social media.**
  - **In-person interactions (rarely used).**

---

3. Key Principles of Social Engineering

Based on Professor Robert Cialdini's six principles of influence:

1. Reciprocity:

- **People feel obligated to return favors.**
- **Example: An attacker does a small favor to compel the victim to reciprocate.**

2. Commitment and Consistency:

- **People are more likely to honor commitments they've made.**
- **Example: Exploiting a victim's charity activities or professional goals.**

3. Social Proof:

- **People tend to follow the actions of others.**
- **Example: Fake testimonials or claiming that other employees have complied with similar requests.**

4. Authority/Intimidation:

- **Victims are more likely to obey authority figures.**
- **Example: Impersonating a senior executive to intimidate employees.**

5. Liking/Familiarity:

- **People are easily influenced by those they like or share interests with.**
- **Example: Casual conversations about shared hobbies or events.**

6. Scarcity/Urgency:

- Limited-time opportunities create a sense of urgency.

- Example: Claiming a deadline to compel quick action.

---

4. Attack Vectors

1. Phishing and Spear Phishing:

- Generic or targeted emails designed to trick victims into revealing information or clicking malicious links.

2. Vishing:

- Manipulative phone calls pretending to be from a trusted source.

3. Smishing:

- Phishing conducted via SMS messages.

4. Tailgating:

- Gaining unauthorized access by following someone into restricted areas.

5. Watering Hole Attacks:

- Compromising a trusted website to target specific users.

6. Quid Pro Quo:

- Offering something in exchange for the victim's compliance, such as technical help.

---

5. Countermeasures

1. Recognizing Social Engineers:

- Be cautious of:

- Unfamiliar individuals displaying urgency or friendliness.

- Unusual email addresses or communication patterns.

- Requests for sensitive information without proper authorization.

2. Training Employees:

- Conduct regular security awareness programs to educate employees about potential attack strategies.

- Encourage skepticism and verification of suspicious requests.

3. Verification Practices:

- Always verify the requester's identity and reason for the request.

- Use internal communication channels for validation.

4. Reporting Incidents:

- Establish clear processes for reporting suspicious activities to the company's security team.

Summary of the Presentation: Cross-Site Scripting (XSS)

1. What is XSS?

- Definition:

- XSS is a security vulnerability that allows attackers to inject malicious scripts (usually JavaScript) into a web application.

- The scripts execute in the context of the user's browser, potentially stealing sensitive data or compromising user accounts.

- Key Characteristics:

- Targets users, not the web server directly.

- Exploits insufficient input validation or output encoding in web applications.

- Remains one of the OWASP Top 10 web vulnerabilities.

---

2. History and Relevance

- XSS has been a threat since the late 1990s.

- Despite advancements in web security, it remains a critical issue in modern web applications.

---

3. Types of XSS

1. Stored (Persistent) XSS:

- Malicious payload is stored on the server (e.g., in a database or file system).

- Affects all users who access the vulnerable page.

- Most dangerous form as it doesn't require user interaction beyond visiting the page.

2. Reflected XSS:

- Payload is part of the HTTP request (e.g., a URL or form submission).

- Requires the attacker to trick the victim into clicking a malicious link.

- Common in search forms and query parameters.

3. DOM-Based XSS:

- Occurs in the client-side JavaScript code.

- Exploits dynamic modifications to the Document Object Model (DOM) without involving the server.

- Harder to detect as it does not leave server logs.

4. Blind XSS:

- Similar to Stored XSS but targets administrators or backend systems.

- Attacker cannot directly see the results; they occur in parts of the application inaccessible to the attacker.

5. Self-XSS:

- Users inadvertently exploit themselves by entering malicious payloads (e.g., via developer consoles or input fields).

- Usually requires social engineering to convince users to execute the payload.

---

4. Exploitation Techniques

- Attackers inject scripts through:

  - URL parameters (`GET` requests).

  - Form submissions (`POST` requests).

  - Cookies or custom HTTP headers.

  - Input fields like comments, search bars, or feedback forms.

- Common payloads include:

  - <script>alert('XSS');</script> for demonstration.

  - <img src=0 onerror="alert('XSS');"/> to bypass certain input restrictions.

---

5. XSS Impacts

- Cookie theft:

  - Steals session cookies to impersonate the user.

- Keylogging:

  - Tracks user inputs like passwords and personal information.

- Browser takeover:

  - Gains control over browser functionalities using plugins or extensions.

- Defacement:

  - Alters webpage appearance to spread misinformation or malware.

- Phishing:

  - Embeds fake login forms to collect user credentials.

---

6. Mitigation Strategies

- Input Validation:

  - Reject invalid or unexpected input at the server level.

  - Enforce strict rules for acceptable input formats (e.g., only allow numbers in numeric fields).

- Output Encoding:

  - Encode user input when it is rendered in the HTML output.

  - Use context-aware encoding:

- HTML encoding for rendering inside tags.

- URL encoding for attributes in links.

- JavaScript escaping for data used in scripts.

- Use Security Libraries:

  - Leverage built-in sanitization functions in frameworks like Angular, React, or Django.

  - Avoid creating custom sanitization logic to prevent errors.

- Content Security Policy (CSP):

  - Restrict which scripts can execute on the page by defining a whitelist.

- Regular Testing:

  - Use tools like Burp Suite, BeEF (Browser Exploitation Framework), and automated vulnerability scanners to find XSS.

---

7. Case Studies and Advanced Exploits

- Advanced Phishing:

  - Embedding fake login forms within legitimate websites.

  - Exploits user trust by running on the original domain.

- BeEF (Browser Exploitation Framework):

  - Automates XSS exploitation for penetration testing.

  - Can perform attacks like webcam control, phishing, or browser manipulation.