Security Threats

The CIA Triad

One of the models often used to describe the relationship between security and its objects is known the CIA triad. CIA stands for Confidentiality, Integrity, and Availability. Each of these is a desirable property of the things we might want to secure, and each of these three properties can be attacked

1. **Confidentiality:** Can actors who should not have access to the system or information access the system or information?
2. **Integrity:** Can the data or the system be modified in some way that is not intended?
3. **Availability:** Are the data or the system accessible when and how they are intended to be?

Risks, Threats, Vulnerabilities, and Exploits

Risk: A simple way to define risk is to consider two axes: the probability that a negative event will occur, and the impact on something we value if such an event happens. As cybersecurity professionals, we should always consider risk by examining the questions How likely is it that a particular attack might happen? and What would be the worst possible outcome if the attack occurs?

Threat: In cybersecurity, a threat is something that poses risk to an asset we care about protecting. Not all threats are human; if our network depends on the local electricity grid, a severe lightning storm could be a threat to ongoing system operations. A person or group of people embodying a threat is known as a **threat actor**, a term signifying agency, motivation, and intelligence.

Vulnerability: For a threat to become an actual risk, the target being threatened must be vulnerable in some manner. A vulnerability is a flaw that allows a threat to cause harm. Not all flaws are vulnerabilities. To take a non-security example, let's imagine a bridge. A bridge can have some aesthetic flaws; maybe some pavers are scratched or it isn't perfectly straight. However, these flaws aren't vulnerabilities because they don't pose any risk of damage to the bridge. Alternatively, if the bridge does have structural flaws in its construction, it may be vulnerable to specific threats such as overloading or too much wind.

Exploits: In computer programs, vulnerabilities occur when someone who interacts with the program can achieve specific objectives that are unintended by the programmer. When these objectives provide the user with access or privileges that they aren't supposed to have, and when they are pursued deliberately and maliciously, the user's actions become an exploit. The word exploit in cybersecurity can be used as both a noun and as a verb. As a noun, an exploit is a procedure for abusing a particular vulnerability. As a verb, to exploit a vulnerability is to perform the procedure that reliably abuses it. An attack surface describes all the points of contact on our system or network that could be vulnerable to exploitation. An attack vector is a specific vulnerability and exploitation combination that can further a threat actor's objectives. Defenders attempt to reduce their attack surfaces as much as possible, while attackers try to probe a given attack surface to locate promising attack vectors.

Cyber Attacks Categories

Here are some popular categories of cyber attacks:

1. **Malware Attacks:**
    - **Viruses:** Malicious software that attaches itself to a legitimate program and spreads when that program is executed.
    - **Worms:** Self-replicating malware that spreads across networks without user interaction.
    - **Trojans:** Malware disguised as legitimate software, which often tricks users into installing it.
2. **Phishing Attacks:**
    - **Phishing:** Deceptive attempts to obtain sensitive information (such as usernames, passwords, or financial details) by posing as a trustworthy entity in electronic communication.
3. **Man-in-the-Middle (MitM) Attacks:**
    - **Eavesdropping:** Unauthorized interception of communication between two parties.
    - **Session Hijacking:** Intercepting and taking over an established session between a user and a system.
4. **Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) Attacks:**
    - **DoS:** Overloading a system, service, or network to disrupt its availability.
    - **DDoS:** Coordinating multiple systems to flood a target with traffic, overwhelming its resources.
5. **Ransomware Attacks:**
    - **Ransomware:** Malware that encrypts a user's data and demands payment (usually in cryptocurrency) for its release.
6. **SQL Injection:**
    - Exploiting vulnerabilities in a web application's database by injecting malicious SQL code, potentially gaining unauthorized access to the database.
7. **Cross-Site Scripting (XSS):**
    - Injecting malicious scripts into web pages viewed by other users, often to steal sensitive information.
8. **Drive-by Downloads:**
    - Automatically downloading malicious software onto a user's device when they visit a compromised or malicious website.
9. **Credential Stuffing:**
    - Using previously stolen usernames and passwords to gain unauthorized access to multiple accounts, exploiting the tendency of users to reuse passwords.
10. **Social Engineering Attacks:**

- Manipulating individuals into divulging confidential information through psychological manipulation.

11. **IoT-Based Attacks:**
    - Exploiting vulnerabilities in Internet of Things (IoT) devices to gain unauthorized access or launch attacks.
12. **Zero-Day Exploits:**
    - Attacks that target undisclosed vulnerabilities in software or hardware before the vendor releases a patch.
13. **Advanced Persistent Threats (APTs):**
    - Long-term targeted attacks in which adversaries gain unauthorized access to a network and remain undetected for an extended period.

Types of Malware

Malware is a broad term that can be associated to any program or script that was intentionally developed to destroy data or cause damage to the normal functionality of a computer or network, or to perform malicious activities such as stealing sensitive information (e.g. login credentials, credit card numbers, financial information, etc.) or gaining unauthorized access to computer systems. It can come in different formats, such as executables, binary shell code, script, or firmware.

The widely used classification is made by malware type, with some being more common than others. The most significant and common malware types are

Virus: It is malicious software that injects its malicious code into other files in a target system, thus spreading within the target system and potentially to other systems as well. Viruses must execute to do their malicious activities, so they target any type of file that could be executed on the system.

Worms: It is like virus, worms are infectious and designed to replicate themselves. However, a worm duplicates itself without targeting and infecting specific files that are already present on the target system. Worms can spread very quickly through the network, relying on security weaknesses and vulnerabilities in the target host to access it, and perform its malicious activities like stealing or deleting data.

Trojan horses: This malicious program pretends to be harmless, in order to deceive the victim into loading and executing it, and therefore perform its malicious tasks. A Trojan payload can be anything but is usually a form of a backdoor that allows attackers unauthorized access to the affected devices. It can also be used to install keyloggers that can easily capture sensitive data such as names and passwords, credit card, financial information, etc.

Rootkits: These are a set of malicious software tools that give attackers privileged access to the victim system. Attackers can then remotely execute files, steal sensitive information, change the system configuration, or alter the functionality of the security mechanism . Unlike virus and worms, rootkits cannot self-propagate or replicate but, it must be installed on the target system.

Adware: This malicious software automatically displays advertisements to users and collect data about their activities without their consent. This type of malware does not usually harm the system, and most of the times the user will never be able to identify its malicious activities; for this reason, adware is also referred to as grayware. Some adware may come with integrated spyware such as keyloggers and other privacy-invasive software

Spyware: This kind of malware installs secretly on the target system for the purpose of monitoring the user's activities without their knowledge. The main goal of spyware is usually to capture sensitive information like bank accounts, passwords, or credit card information. Any software that is downloaded and installed without the user's authorization can be classified as spyware.

Ransomware: This malicious program prevents users from accessing their system, either by disabling the system's functionality or by locking the users' files and displays a message that demands payment (or ransom) to restore its functionality. It can be spread to the victim's devices through vulnerabilities in the system or through downloaded files and links in phishing emails . According to security reports, recent ransomware attacks focused on healthcare, local government, and education sectors, in particular.

Keylogger: It is a malicious piece of software that records the keystrokes on a device to intercept sensitive information typed in through the keyboard. This gives attackers the benefit of access to account numbers and PIN codes, passwords to online shopping websites, email logins, and other confidential information.

Bot/Botnet: Short for "robot network", is a software application or script that is programmed to do certain repetitive tasks automatically. Malicious bots are used by cyber-criminals to remotely take control over compromised devices and use them to launch more attacks, or create botnets, which are networks of infected devices. In this case, infected devices (also referred as zombies) are orchestrated by a command and control (C&C) server that instructs them with specific malicious actions, such as Distributed Denial of Service (DDoS) attacks, Application Programming Interface (API) abuse, phishing attacks, spam emails, ransomware, etc.

Malware programs can span multiple categories. For instance, a worm might include a keylogger that collects login credentials. Malware can also create new vulnerabilities in the victim host or network by disabling their security mechanisms (e.g. removing antivirus), or changing passwords and firewall settings, installing backdoors, and more. For instance, the **Gh0st RAT** (Remote Access Terminal) Trojan, which is one of the top ten alerted malware in February 2020, can create a backdoor into infected devices, and therefore allows the attacker to fully control them.

Cryptography

Cryptography is the technique of taking plain, legible text and implementing an algorithm to it to encrypt it to produce ciphertext, which seems to be gibberish before it is decrypted. Cryptographic functions have a randomization property. If you want to reverse the results of cryptography, like decrypt or verify the cipher, you will need the key. That is why they are called one-way functions. We can apply encryption to maintain two of the three security principle - **confidentiality and integrity**

cryptographic digest

A cryptographic digest, also known as a hash function or hash algorithm, is a mathematical function that takes an input (or "message") and produces a fixed-size string of characters, which is typically a hexadecimal number. The output, commonly called the hash value or hash code, is unique to the input data. Even a small change in the input data should produce a significantly different hash value.

The primary purposes of cryptographic digests are:

1. **Data Integrity**: Hash functions are used to ensure the integrity of data. If the data changes in any way, the hash value will change as well. By comparing the hash values of the original and received data, one can verify if the data has been altered.
2. **Digital Signatures**: Cryptographic digests are a fundamental component of digital signatures. When someone signs a message or a document, they create a hash of the content and encrypt it with their private key. The recipient can use the sender's public key to verify the signature by checking that the decrypted hash matches the hash of the received data.
3. **Password Storage**: Hash functions are commonly used to store passwords securely. Instead of storing the actual passwords, systems store the hash values of passwords. When a user attempts to log in, the system hashes the entered password and compares it to the stored hash.
4. **Checksums**: Hash functions are used to create checksums for data transmitted over a network. The sender computes the hash of the data and sends both the data and the hash to the receiver. The receiver recalculates the hash upon receiving the data and compares it with the received hash to verify data integrity.

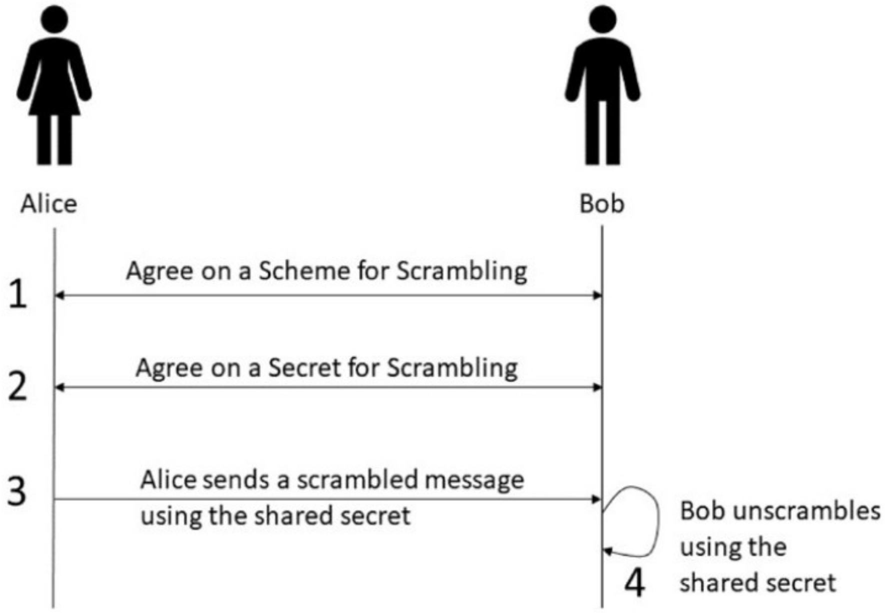
Common cryptographic hash functions include SHA-256 (Secure Hash Algorithm 256-bit), SHA-3, and MD5 (Message Digest Algorithm 5). However, MD5 is considered insecure for cryptographic purposes due to vulnerabilities, and it is recommended to use stronger hash functions like SHA-256 for security-critical applications.

Symmetric cryptography

Symmetric cryptography, also known as secret-key or private-key cryptography, is a form of encryption where the same key is used for both the encryption of the plaintext and the decryption of the ciphertext. In other words, the communicating parties must share a secret key and keep it private. This shared key is used both to transform the original message (plaintext) into an unreadable format (ciphertext) and to reverse the process, turning the ciphertext back into its original form.

Here's how symmetric cryptography works:

1. **Key Generation**: Two parties who wish to communicate securely must first agree on a secret key. This key is then used for both encryption and decryption.
2. **Encryption**: The sender uses the shared secret key to encrypt the plaintext message, turning it into ciphertext. The encryption algorithm and the key are kept secret.
3. **Transmission**: The ciphertext is sent over the communication channel.
4. **Decryption**: The recipient uses the same secret key to decrypt the received ciphertext, transforming it back into the original plaintext.


Alice
Bob
1
Agree on a Scheme for Scrambling
2
Agree on a Secret for Scrambling
3
Alice sends a scrambled message using the shared secret
Bob unscrambles using the shared secret
4

Symmetric Cryptography Scheme

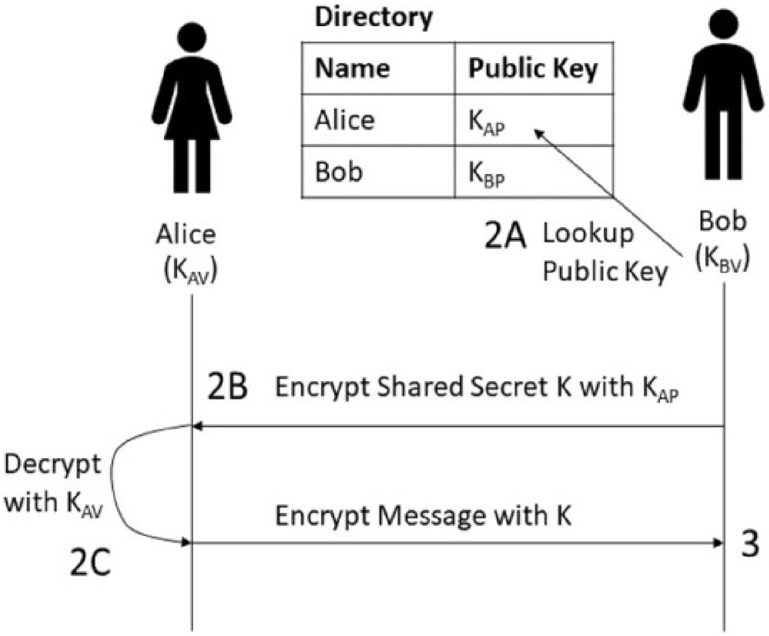
The main advantage of symmetric cryptography is its efficiency. The algorithms used for symmetric encryption are generally fast and require less computational resources compared to their asymmetric counterparts. However, a significant challenge with symmetric key cryptography lies in securely distributing and managing the secret keys, especially when there are many communicating parties.

One common use of symmetric cryptography is in securing the confidentiality of data in transit. For example, when you access **a secure website (using HTTPS)**, symmetric cryptography is often used to encrypt the data exchanged between your browser and the web server.

Common symmetric key algorithms include Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Triple DES. While symmetric cryptography is efficient for confidentiality, it doesn't provide a mechanism for key exchange or digital signatures, which are addressed by asymmetric (public-key) cryptography. Often, a combination of both symmetric and asymmetric cryptography is used to achieve a balance of efficiency and security in various cryptographic applications.

Asymmetric cryptography

Asymmetric cryptography, also known as public-key cryptography, is a cryptographic system that uses pairs of keys: public keys, which may be disseminated widely, and private keys, which are known only to the owner. The idea is that what one key encrypts, only the other corresponding key can decrypt. This is in contrast to symmetric cryptography, where the same key is used for both encryption and decryption.



Key exchange using asymmetric cryptography

Here's how asymmetric cryptography works:

1. **Key Pair Generation**: Each participant generates a pair of keys – a public key and a private key. The public key is shared openly, while the private key is kept secret.
2. **Encryption**: If someone wants to send an encrypted message to a recipient, they use the recipient's public key to encrypt the message.
3. **Transmission**: The encrypted message (ciphertext) is sent to the recipient.
4. **Decryption**: The recipient uses their private key to decrypt the received ciphertext and recover the original message (plaintext).

The use of asymmetric cryptography brings several advantages:

- **Key Distribution**: Unlike symmetric cryptography, where sharing secret keys securely can be a challenge, in asymmetric cryptography, only the public keys need to be shared openly. Private keys are kept secret, and there's no need to transmit them.
- **Digital Signatures**: Asymmetric cryptography is often used for digital signatures. The sender can use their private key to create a digital signature, and the recipient can verify the

signature using the sender's public key. This ensures the authenticity and integrity of the message.

- **Key Exchange**: Asymmetric cryptography can facilitate secure key exchange for symmetric cryptography. Two parties can use asymmetric encryption to exchange a shared secret key that is then used for secure communication using symmetric encryption.

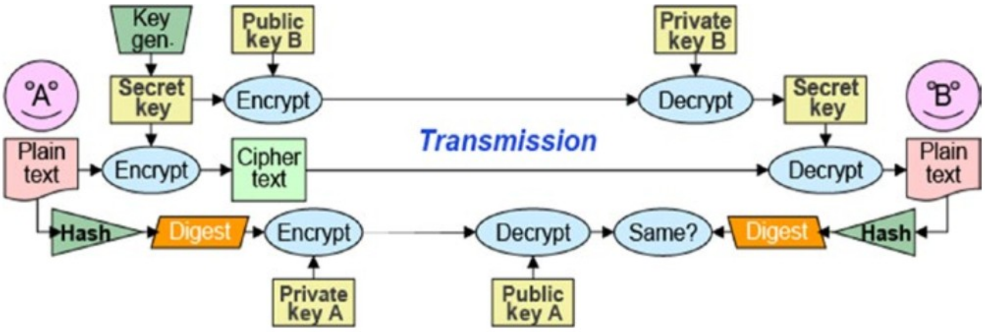
Common asymmetric key algorithms include **RSA (Rivest-Shamir-Adleman)**, **DSA (Digital Signature Algorithm)**, and **ECC (Elliptic Curve Cryptography)**. These algorithms are computationally more intensive than symmetric algorithms, so they are often used for tasks where efficiency is less critical, such as key exchange, digital signatures, and securing the initial connection in secure communication protocols like HTTPS. In many cryptographic systems, a combination of symmetric and asymmetric cryptography is used to leverage the strengths of both approaches.

Digital signature

A digital signature is a cryptographic technique used to ensure the authenticity and integrity of a digital message, document, or transaction. It provides a way for the sender of a message to demonstrate that the message was indeed created by them (authentication) and that it has not been altered in transit (integrity).

Here's how digital signatures typically work:

1. **Key Pair Generation**: Similar to asymmetric cryptography, a user generates a pair of cryptographic keys - a private key and a corresponding public key.
2. **Signing**: To digitally sign a message or document, the sender uses their private key to create a unique digital signature for the content. This process involves applying a mathematical algorithm to the message or a hash of the message.
3. **Distribution**: The signed message, along with the digital signature, is sent to the recipient. The sender's public key is usually made publicly available or provided to the recipient through a trusted channel.
4. **Verification**: The recipient uses the sender's public key to verify the digital signature. If the verification is successful, it confirms that the message was indeed signed by the possessor of the private key and that the message has not been altered since it was signed.



Hashing functions and asymmetric cryptography used to create digital signatures

Digital signatures provide the following benefits:

- **Authentication**: The recipient can be confident that the message was sent by the claimed sender, as only the possessor of the private key could have produced the digital signature.
- **Integrity**: The recipient can be sure that the message has not been tampered with in transit since any alteration to the message would result in an invalid digital signature.
- **Non-repudiation**: The sender cannot later deny their involvement in creating the message because the digital signature is uniquely tied to their private key.

Digital signatures are widely used in various applications, including secure email communication, online transactions, software distribution, and legal documents. Common digital signature algorithms include RSA (Rivest-Shamir-Adleman), DSA (Digital Signature Algorithm), and ECDSA (Elliptic Curve Digital Signature Algorithm). The use of digital signatures is a crucial aspect of secure and trusted communication in the digital age.

Digital certificate

A digital certificate is a cryptographic credential that is used to verify the identity of an entity, such as a person, device, or organization, in the digital realm. Digital certificates are a crucial component of public-key cryptography and are often used in various online security protocols to establish trust and enable secure communication.

Here are key components and concepts related to digital certificates:

1. **Public Key Infrastructure (PKI):** Digital certificates are a fundamental component of a PKI, a framework that manages digital keys and certificates. PKI provides a way to secure communication over networks like the internet.
2. **Certificate Authority (CA):** A certificate authority is a trusted third party that issues digital certificates. The CA's role is to verify the identity of the entity requesting the certificate (through a process known as authentication) and then digitally sign the certificate. Popular CAs include Verisign, DigiCert, and Let's Encrypt.
3. **Certificate Contents:**
    - **Public Key:** The digital certificate includes the public key of the entity to which the certificate is issued.
    - **Subject:** Information about the entity (such as its name and other identifying information) is included in the certificate.
    - **Issuer:** The entity (usually a CA) that issues and signs the certificate.
    - **Validity Period:** The time during which the certificate is considered valid.
    - **Digital Signature:** The CA's digital signature, which verifies the authenticity of the certificate.
4. **Digital Signature Verification:** When a digital certificate is presented, the recipient can verify its authenticity by checking the digital signature. This involves using the public key of the CA (which is typically widely available and trusted) to verify that the signature was indeed generated by the CA.

5. **SSL/TLS Certificates:** One common use of digital certificates is in securing web communication through the use of SSL/TLS (Secure Sockets Layer/Transport Layer Security) protocols. Websites use SSL/TLS certificates to encrypt data in transit and to authenticate the server to the client.
6. **Code Signing Certificates:** Developers and software publishers use digital certificates to sign their software. This provides a way for users to verify the origin and integrity of the software.

Digital certificates play a critical role in establishing a secure and trusted digital environment, allowing users to confidently engage in online transactions, access secure websites, and communicate over the internet without compromising the confidentiality and integrity of their data.

Security Technology