UNIT 3:

Information Security Fundamental

The Importance of the Background

Background information is an important component of an essay, research paper or presentation because it can get the reader's attention and prompts them to keep reading. Depending on the topic, background information might take a particular tone or approach to provide context or address a question.

National & International Scenario:

National & International Scenario of Information Security:

National Scenario:

In recent years, information security has become a top priority for governments and organizations in many countries. As the world becomes more digitally connected, the risk of cyber threats and attacks has increased significantly. Nations are now recognizing the importance of securing their critical infrastructure, sensitive data, and citizens' privacy from various cyber threats.

Governments are establishing national cybersecurity strategies and agencies to tackle the ever-growing cyber challenges. These strategies aim to improve

cybersecurity readiness, enhance incident response capabilities, and foster collaboration between the public and private sectors to share threat intelligence and counter cyber threats effectively.

National cybersecurity agencies work towards identifying and mitigating cyber threats within the country's borders. They develop guidelines and regulations for organizations and businesses to adopt best practices and comply with cybersecurity standards. Furthermore, national governments are investing in cybersecurity research and development to stay ahead of evolving cyber threats.

International Scenario:

Information security is not limited to national boundaries. Cyber threats are global in nature, and cybercriminals often operate across international borders. As a result, international cooperation and collaboration are crucial for addressing global cybersecurity challenges.

Various international organizations, conventions, and agreements play a vital role in fostering cybersecurity cooperation among nations. For instance, the United Nations (UN) recognizes the importance of cybersecurity and advocates for the responsible use of information and communication technologies. The UN's Group of Governmental Experts (GGE) works to develop norms, rules, and principles for responsible state behavior in cyberspace.

The Budapest Convention on Cybercrime, adopted by the Council of Europe, is the first international treaty seeking to address cybercrime and promote international cooperation in investigating and prosecuting cybercriminals. Many countries have ratified this convention to enhance their abilities to combat cybercrime across borders.

Additionally, regional bodies like the European Union Agency for Cybersecurity (ENISA) and the Asia-Pacific Economic Cooperation (APEC) have developed frameworks to facilitate cybersecurity collaboration among member countries in their respective regions.

Furthermore, cybersecurity firms and organizations work across borders to share threat intelligence and develop joint responses to cyber incidents. Public-private partnerships are also emerging, where governments and private sector entities collaborate to strengthen cyber defenses and respond to cyber threats collectively.

The international community is increasingly recognizing the need to address emerging challenges such as cyberwarfare, state-sponsored cyber-espionage, and the protection of critical infrastructure from cyber threats.

Conclusion:

The national and international scenarios of information security highlight the global nature of cyber threats and the need for strong collaboration and cooperation among nations. Governments are increasingly investing in cybersecurity measures to protect their citizens, infrastructure, and economy from cyber threats. International organizations and conventions facilitate global collaboration to tackle cybercrime and address emerging cybersecurity challenges. As technology continues to advance, information security remains an ever-evolving field that requires ongoing efforts from governments, organizations, and individuals to stay ahead of cyber threats and protect the digital ecosystem.

Identification and Authentication:-

In more analog days, hard copies of official government documents, physical identification cards and biometric information like fingertips were all viable ways to make sure a person was who they claimed to be. The digital age with its myriad online transactions, however, requires new ways to identify, authenticate and authorize a person's identity.

Let's take a closer look at identification, authentication and authorization and why each process is increasingly important in the modern world.

The Importance of Identification

In most digital transactions, identification is the step where users prove their identity by providing a name, email address, phone number or username.

Identification is the first step in confirming a person's identity and must happen before authentication and authorization. Users can also provide

more information, like a government-issued photo, ID, or social security number, to further identify themselves.

Identification happens in the initial setup stage of accounts and services. A username and password typically identify a person each time they access that account or service.

However, it can be challenging in digital environments to verify identification simply by receiving personally identifiable information and usernames and passwords. A stolen wallet or hacked email account is also enough for someone to attempt to steal an identity.

Given these risks, identification is simply the first step to establishing the baseline for the authentication process.

Forms of Identification For Authentication

Depending on the transaction's requirements, identification can require one or all of the following:

What you know

Information that only the person in question would easily know, including passwords, personal identification numbers (PINs), maiden names or answers to security questions.

What you have

Possessions that are unique to a specific person, like keys, badges or swipe cards.

What you are

As the most secure form of identification, biometric information is immune to theft or replication and can definitively prove identity. This information includes things like fingerprints or a facial scan.

The Importance of Authentication

Authentication requires users to prove they are still the person they claimed to be during the identification phase.

In 2021, the FTC received more than 2.8 million reports of fraud, resulting in over $5.8 billion in losses. If identification was the only barrier between access to an account or a system, these fraud and identity theft instances would be even more rampant. Authentication provides a layer of protection beyond identification to help users keep their accounts and their identities secure.

Following basic identification, authentication initiates a match between the user's previously provided information. Increasingly, authentication systems ask for a one-time verification code sent to an email address or phone number, even if the user's provided details and stored information match. Authorization requires both identification and authentication.

Methods of Authentication

Password-based authentication

Passwords are the most common method of authentication. Optimal protection requires using many varied passwords using different strings of letters, numbers and characters. However, many people use the same or similar passwords across accounts, which leaves them vulnerable to phishing and password breaches. Malicious entities can easily bypass password protection if they can access the user's email account or a previously used password. In short, passwords alone are not sufficient to provide account protection.

Multi-factor authentication

A safer authentication method involves multi-factor authentication (MFA), which requires using more than one form of authentication, like a Captcha request or a security code sent to your email or phone as an SMS message. MFAs have their own drawbacks, as some users may lose

access to a previous email or phone number, effectively locking them out of their account without intervention.

Certificate-based authentication

Essentially, certificate-based authentication (CBA) uses a digital certificate to identify a user, device or machine before providing access to an application, network or other resource. This form of authentication is more secure because it's based on both what the user has (the digital certificate) and what they know (their password).

Biometric-based authentication

Biometric-based authentication relies on individuals' unique biological characteristics to authenticate their identity. Methods like facial recognition, fingerprint or eye scanning and voice recognition provide a high level of security with minimal disruption. When paired with multi-factor authentication, this method provides an additional layer of security. However, biometric-based authentication does present privacy concerns — and ethical questions for some.

Token-based authentication

Token-based authentication simplifies the authentication process for recognized users. After entering a username and password, a user can access protected systems without providing credentials again.

The Importance of Authorization

Authorization grants users access, rights and privileges to a service, account, or system based on previously secured identification and authentication.

Identification and authentication validate a person's identity, but authorization ensures the person in question should have access to the system or resource.

Authorization gives users rights and privileges after identifying, authenticating and authorizing them. It secures sensitive resources in a

system and protects individual users from unauthorized access to their accounts or information.

Types of Authorization Methods

API keys

An API key is a secret code that gets you inside a system or resource, essentially acting like an ID card to assign proper permissions and track data usage. In more technical terms, it's a string of characters used to identify and authorize an application or user who requests the service of an API (application programming interface).

Basic auth

Basic auth is akin to providing a key in the form of user credentials to access an online account. Although this process is straightforward, it can leave your credentials and, eventually, your online account vulnerable.

HMAC

HMAC stands for Keyed-Hashing for Message Authentication. Because it uses cryptography keys to enforce integrity and authenticity, HMAC is similar to digital signatures. Secure file transfer protocols like FTPS, SFTP and HTTPS use HMAC to ensure data integrity.

OAuth

OAuth uses authorization tokens vs. a password to connect an app to a user account. It allows users to give other websites or applications access to their information without resupplying passwords.

Notarize Can Help Businesses Identify and Authenticate Online Notarizations

Many applications and processes require consumers to identify, authenticate and authorize their digital identity.

Notarize uses dynamic knowledge-based authentication and database-driven information to confirm a person's digital identity to prevent fraudulent notarizations and secure online notarizations for business.

Confidentiality, Privacy integrity:-

Confidentiality, Privacy, and Integrity are three essential pillars of information security that work together to protect sensitive data and maintain the trust of individuals, organizations, and nations. Let's explore each of these concepts:

1. ****Confidentiality:****

Confidentiality refers to the protection of information from unauthorized access or disclosure. It ensures that sensitive data remains accessible only to authorized individuals or entities. Confidentiality is crucial for safeguarding private information, trade secrets, financial data, personal records, and other sensitive data that could cause harm if accessed by unauthorized parties.

Key aspects of confidentiality include:

- Data Encryption: Converting plaintext data into an unreadable form (ciphertext) using encryption algorithms, which can only be decrypted with the appropriate key.

- Access Controls: Implementing mechanisms such as passwords, access permissions, and role-based access control (RBAC) to restrict data access to authorized users only.

- Data Classification: Categorizing data based on its sensitivity level and applying appropriate security measures based on its classification.

2. **Privacy:**

Privacy is the right of individuals to control their personal information and how it is collected, used, and shared. It involves protecting the personally identifiable information (PII) of individuals and ensuring compliance with privacy laws and regulations. Privacy is essential for maintaining trust between individuals and organizations, especially in an era of extensive data collection and sharing.

Key aspects of privacy include:

- Data Minimization: Collecting and retaining only the minimum amount of personal data necessary to fulfill a specific purpose.

- Consent and Transparency: Obtaining informed consent from individuals before collecting their data and being transparent about how the data will be used.

- Anonymization: Removing or encrypting personally identifiable information from datasets to protect individual identities.

3. **Integrity:**

Integrity ensures that data remains accurate, complete, and unaltered during transmission and storage. It prevents unauthorized modification, deletion, or tampering of data. Data integrity is crucial for maintaining the trustworthiness of information and the reliability of systems.

Key aspects of integrity include:

- Data Hashing: Generating a unique fixed-size hash value based on the content of data to verify its integrity during transmission or storage.

- Digital Signatures: Using cryptographic signatures to ensure the authenticity and integrity of electronic documents or messages.

- Error Checking: Implementing error detection and correction mechanisms to identify and correct data corruption.

Confidentiality, privacy, and integrity are fundamental principles in protecting information assets and maintaining trust in the digital world. Organizations and individuals must implement appropriate security measures and best practices to uphold these principles and protect sensitive data from unauthorized access, disclosure, and tampering. Compliance with relevant privacy regulations and data protection laws is also essential to respect individuals' privacy rights and maintain legal and ethical standards.

Availability. Security & Prevention:-

Availability in information security refers to the ability of authorized users to access and use data and systems as intended. Ensuring availability is an important aspect of information security because it ensures that users can access the resources they need to complete their work and maintain the operations of an organization. Threats to availability include things like network outages, hardware failures, and cyber attacks that disrupt access to systems or data. To ensure availability, organizations often implement measures such as redundant systems, backup and recovery processes, and incident response plans.

Why availability is important

Availability is an important aspect of information security because it ensures that authorized users have access to the data and systems they need to complete their work and maintain the operations of an organization. Without availability, users may be unable to access the resources they need, leading to productivity losses and potentially even financial losses. Ensuring availability is also important for maintaining the trust of customers, stakeholders, and other users, as they rely on being able to access systems and data as needed. In addition, some organizations, such as hospitals and

emergency services, may have critical operations that rely on the availability of information systems, and any disruption to availability could have serious consequences.

Threats of Availability

There are many threats that can compromise the availability of information systems and data. Some examples of these threats include −

- **DDoS attacks** − These are attacks that flood a server or network with traffic, making it difficult or impossible for legitimate users to access the system.
- **Malware** − Malware, or malicious software, can infect systems and disrupt their availability. For example, a ransomware attack could encrypt data on a system and make it unavailable until a ransom is paid.
- **Hardware failures** − Hardware components can fail, leading to system outages.
- **Natural disasters** − Events such as earthquakes, hurricanes, and floods can damage infrastructure and disrupt the availability of systems.
- **Accidental deletion or modification of data** − Users may accidentally delete or modify data, making it unavailable or unusable.
- **Network outages** − Network outages can occur for various reasons, such as equipment failures or cut cables, and can prevent users from accessing systems and data.

To protect against these threats, organizations should implement measures such as redundant systems, backup and recovery processes, incident response plans, and security controls to prevent and mitigate attacks.

Protection for availability threats

There are several measures that organizations can take to protect against threats to the availability of information systems and data −

- **Implement redundant systems** − Redundant systems, such as having multiple servers or backup power sources, can help ensure that systems remain available even if one component fails.
- **Use backup and recovery processes** − Regularly backing up data and having a recovery process in place can help ensure that data is not lost in the event of a failure or attack.

- **Develop an incident response plan** − An incident response plan outlines the steps to take in the event of a disruption to availability, such as a cyber attack or hardware failure.
- **Implement security controls** − Security controls, such as firewalls, intrusion detection and prevention systems, and access controls, can help prevent or mitigate attacks on systems and data.
- **Monitor systems and networks** − Regularly monitoring systems and networks can help organizations identify and respond to potential threats in a timely manner.
- **Train employees** − Educating employees about best practices for information security, such as not clicking on links in suspicious emails, can help prevent accidental or intentional actions that could disrupt availability.

Conclusion

In conclusion, availability is an important aspect of information security that refers to the ability of authorized users to access and use data and systems as intended. Ensuring availability is important because it allows organizations to maintain the operations and productivity of their business. There are various threats to availability, including DDoS attacks, malware, hardware failures, natural disasters, and network outages. To protect against these threats, organizations should implement measures such as redundant systems, backup and recovery processes, incident response plans, and security controls, as well as regularly monitor systems and educate employees about best practices for information security.

Detection & Recovery:-

Deadlock detection and recovery is the process of detecting and resolving deadlocks in an operating system. A deadlock occurs when two or more processes are blocked, waiting for each other to release the resources they need. This can lead to a system-wide stall, where no process can make progress.

There are two main approaches to deadlock detection and recovery:

1. **Prevention**: The operating system takes steps to prevent deadlocks from occurring by ensuring that the system is always in a safe state, where

deadlocks cannot occur. This is achieved through resource allocation algorithms such as the Banker's Algorithm.

2. **Detection and Recovery:** If deadlocks do occur, the operating system must detect and resolve them. Deadlock detection algorithms, such as the Wait-For Graph, are used to identify deadlocks, and recovery algorithms, such as the Rollback and Abort algorithm, are used to resolve them. The recovery algorithm releases the resources held by one or more processes, allowing the system to continue to make progress.

Difference Between Prevention and Detection/Recovery: Prevention aims to avoid deadlocks altogether by carefully managing resource allocation, while detection and recovery aim to identify and resolve deadlocks that have already occurred.

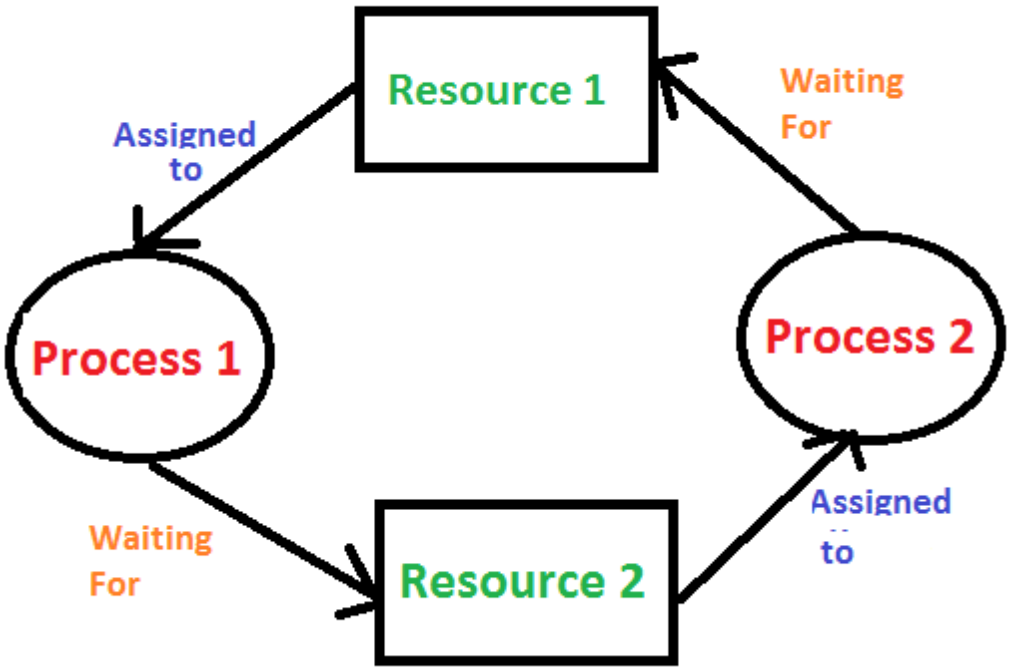
Deadlock detection and recovery is an important aspect of operating system design and management, as it affects the stability and performance of the system. The choice of deadlock detection and recovery approach depends on the specific requirements of the system and the trade-offs between performance, complexity, and risk tolerance. The operating system must balance these factors to ensure that deadlocks are effectively detected and resolved.

In the previous post, we discussed Deadlock Prevention and Avoidance. In this post, the Deadlock Detection and Recovery technique to handle deadlock is discussed.

Deadlock Detection :

1. If resources have a single instance –

In this case for Deadlock detection, we can run an algorithm to check for the cycle in the Resource Allocation Graph. The presence of a cycle in the graph is a sufficient condition for deadlock.



In the above diagram, resource 1 and resource 2 have single instances. There is a cycle R1 → P1 → R2 → P2. So, Deadlock is Confirmed.

2. If there are multiple instances of resources –
Detection of the cycle is necessary but not a sufficient condition for deadlock detection, in this case, the system may or may not be in deadlock varies according to different situations.

3. Wait-For Graph Algorithm –
The Wait-For Graph Algorithm is a deadlock detection algorithm used to detect deadlocks in a system where resources can have multiple instances. The algorithm works by constructing a Wait-For Graph, which is a directed graph that represents the dependencies between processes and resources.

Deadlock Recovery :
A traditional operating system such as Windows doesn't deal with deadlock recovery as it is a time and space-consuming process. Real-time operating systems use Deadlock recovery.

1. **Killing the process –**
   Killing all the processes involved in the deadlock. Killing process one by one. After killing each process check for deadlock again and keep repeating the

process till the system recovers from deadlock. Killing all the processes one by one helps a system to break circular wait conditions.

2. **Resource Preemption –**
   Resources are preempted from the processes involved in the deadlock, and preempted resources are allocated to other processes so that there is a possibility of recovering the system from the deadlock. In this case, the system goes into starvation.
3. **Concurrency Control –** Concurrency control mechanisms are used to prevent data inconsistencies in systems with multiple concurrent processes. These mechanisms ensure that concurrent processes do not access the same data at the same time, which can lead to inconsistencies and errors. Deadlocks can occur in concurrent systems when two or more processes are blocked, waiting for each other to release the resources they need. This can result in a system-wide stall, where no process can make progress. Concurrency control mechanisms can help prevent deadlocks by managing access to shared resources and ensuring that concurrent processes do not interfere with each other.

ADVANTAGES OR DISADVANTAGES:

Advantages of Deadlock Detection and Recovery in Operating Systems:

1. **Improved System Stability:** Deadlocks can cause system-wide stalls, and detecting and resolving deadlocks can help to improve the stability of the system.
2. **Better Resource Utilization:** By detecting and resolving deadlocks, the operating system can ensure that resources are efficiently utilized and that the system remains responsive to user requests.

3. **Better System Design**: Deadlock detection and recovery algorithms can provide insight into the behavior of the system and the relationships between processes and resources, helping to inform and improve the design of the system.

Disadvantages of Deadlock Detection and Recovery in Operating Systems:

1. **Performance Overhead:** Deadlock detection and recovery algorithms can introduce a significant overhead in terms of performance, as the system must regularly check for deadlocks and take appropriate action to resolve them.
2. **Complexity:** Deadlock detection and recovery algorithms can be complex to implement, especially if they use advanced techniques such as the Resource Allocation Graph or Timestamping.
3. **False Positives and Negatives:** Deadlock detection algorithms are not perfect and may produce false positives or negatives, indicating the presence of deadlocks when they do not exist or failing to detect deadlocks that do exist.
4. **Risk of Data Loss**: In some cases, recovery algorithms may require rolling back the state of one or more processes, leading to data loss or corruption.

Overall, the choice of deadlock detection and recovery approach depends on the specific requirements of the system, the trade-offs between performance, complexity, and accuracy, and the risk tolerance of the system. The operating system must balance these factors to ensure that deadlocks are effectively detected and resolved.

e-Commerce security :-

e-Commerce security refers to the practices and measures taken to protect the confidentiality, integrity, and availability of online transactions and sensitive information in electronic commerce (e-commerce) environments. As the popularity of online shopping and electronic transactions continues to grow, ensuring robust e-commerce security is crucial for building trust with customers and safeguarding their personal and financial data. Several key aspects contribute to e-commerce security:

1. Secure Sockets Layer (SSL) / Transport Layer Security (TLS):

SSL and its successor, TLS, are cryptographic protocols that establish secure encrypted communication between a web server and a web browser. This encryption ensures that data transmitted during e-commerce transactions, such as credit card information, remains confidential and protected from interception by unauthorized entities.

2. Secure Payment Gateways:

Payment gateways act as intermediaries between e-commerce websites and financial institutions. A secure payment gateway encrypts payment data and ensures that sensitive information is securely transmitted to the payment processor for authorization and processing.

3. PCI DSS Compliance:

The Payment Card Industry Data Security Standard (PCI DSS) is a set of security standards established to protect cardholder data during credit card transactions. Compliance with PCI DSS is mandatory for all organizations that process, store, or transmit credit card information.

4. Multi-Factor Authentication (MFA):

Implementing MFA adds an extra layer of security by requiring customers to provide multiple forms of identification before completing a transaction. This could include a combination of something the customer knows (password), something the customer has (one-time code from a mobile device), or something the customer is (biometric data).

5. Data Encryption:

Encrypting sensitive data stored on e-commerce websites' servers adds an additional layer of protection, even if an unauthorized individual gains access to the database.

6. Regular Security Audits and Vulnerability Assessments:

Conducting regular security audits and vulnerability assessments helps identify and address potential weaknesses in the e-commerce infrastructure before they can be exploited by attackers.

1. Fraud Detection and Prevention:

Employing fraud detection systems and algorithms can help identify and prevent fraudulent transactions, such as account takeovers, identity theft, and credit card fraud.

8. User Education and Awareness:

Educating customers about e-commerce security best practices, such as creating strong passwords, avoiding public Wi-Fi for sensitive transactions, and identifying phishing attempts, helps them become more vigilant against potential threats.

9. Secure Product Delivery and Handling:

Ensuring secure product delivery and handling, especially for digital products or services, helps prevent unauthorized access and piracy.

10. Secure Backup and Disaster Recovery:

Regularly backing up e-commerce data and having robust disaster recovery plans in place help to ensure business continuity and minimize data loss in the event of a security incident or system failure.

By implementing comprehensive e-commerce security measures, online businesses can create a safe and trustworthy environment for their customers, thereby fostering customer loyalty and increasing their competitiveness in the digital marketplace.

Security Threats:-

A cybersecurity threat is the threat of a malicious attack by an individual or organization attempting to gain access to a network, corrupt data, or steal confidential information.

No company is immune from cyber-attacks and data breaches. Some cyberattacks can even destroy computer systems.

As cyber threats become increasingly sophisticated, your business must implement security precautions and analyze cybersecurity risks to keep your data safe.

What are the Top Security Threats?

First you need to understand the difference between infosec and cybersecurity, as well as the types of threats you'll face almost daily – both the information security threats that exist today, as well as the new and emerging threats sure to plague your enterprise tomorrow.

Malware

The most common cyberattack is malicious software, more commonly known as malware. Malware includes spyware, ransomware, backdoors, trojans, viruses, and worms.

- Spyware is software that allows attackers to obtain information about your computer activities by transmitting data covertly from your hard drive.
- Ransomware is designed to encrypt files on a device, rendering any files (and the systems that rely on them) unusable. Usually, malicious actors demand a cash ransom in exchange for decryption.
- A backdoor circumvents routine authentication procedures to access a system. This gives the attacker remote access to resources within an application, such as databases and file servers; and allows malicious actors to issue system commands and update malware remotely.
- Trojans are malware or code that acts as a legitimate application or file to trick you into loading and executing the malware on your device. A trojan's goal is to damage or steal your organization's data or inflict some other harm on your network.
- A computer virus is a malicious piece of computer code designed to spread from device to device. These self-copying threats are usually intended to damage a machine or steal data.
- Worms are malware that spreads copies of themselves from computer to computer without human interaction and do not need to attach themselves to a software program to cause damage.

Malware is usually installed into the system when the user opens a malicious link or email. Once installed, malware can block access to critical components of your network, damage your system, and export confidential information to destinations unknown.

Your organization can prevent malware-based cyber attacks by:

- Using reputable antivirus and anti-malware solutions, email spam filters, and endpoint security solutions.
- Assuring that your cybersecurity updates and patches are all up to date.
- Requiring your employees to undergo regular cybersecurity awareness training to teach them how to avoid suspicious websites and to avoid engaging with suspicious emails.
- Limiting user access and application privileges.

Phishing and Spear Phishing

Phishing is a type of social engineering that attempts to trick users into giving up sensitive data such as usernames and passwords, bank account information, Social Security numbers, and credit card data.

Typically, hackers send out phishing emails that seem to come from trusted senders such as PayPal, eBay, financial institutions, or friends and co-workers. The bogus messages try to get users to click on links in the emails, which will direct the users to fraudulent websites that ask for personal information or install malware on their devices.

Opening attachments sent via phishing emails can also install malware or allow hackers to control your devices remotely.

Spear phishing is a more sophisticated form of phishing attack, where cybercriminals target only privileged users such as system administrators and C-suite executives. The attackers might use details from a person's social media accounts to seem even more legitimate to the target.

Other types of phishing can include smishing, vishing, clone phishing, domain spoofing, URL phishing, watering hole phishing, and evil twin phishing. All can be very costly.

Organizations can do several things to reduce the chances of phishing:

- Implement cybersecurity awareness training for every employee.
- Emphasize the importance of phishing reporting.
- Run random phishing simulations.
- Push HTTPS on your website to create secure, encrypted connections.
- Institute access management policies and procedures.
- Use reliable email and spam filters.
- Require two-factor authentication.
- Use email encryption and email signing certificates.

Man-in-the-Middle (MITM) Attacks

These attacks occur when malicious actors insert themselves into the middle of a two-party communication. Once the attacker intercepts the incoming message,

he or she filters and steals sensitive information and then returns different responses to the original user.

Sometimes malicious actors set up fake wi-fi networks or install malware on users' computers or networks. Also called eavesdropping attacks, MITM attacks aim to gain access to your business or customer data.

Distributed Denial of Service (DDoS)

A DDoS attack aims to take down a company's website by overwhelming its servers with requests. It's analogous to calling a company's phone number constantly, so that legitimate callers only get a busy signal and never get through.

In this attack, requests come from hundreds or thousands of IP addresses that have probably also been compromised and tricked into continuously requesting a company's website.

A DDoS attack can overload your servers, slowing them down significantly or temporarily taking them offline. These shutdowns prevent customers from accessing your website and completing orders.

Structured Query Language (SQL) injection

SQL injection attacks occur when cybercriminals attempt to access databases by uploading malicious SQL scripts. Once successful, the malicious actor can view, change, or delete data stored in the SQL database.

Domain Name System (DNS) attack

A DNS attack is a cyberattack where cybercriminals exploit vulnerabilities in the DNS. The attackers leverage the DNS vulnerabilities to divert site visitors to malicious pages (DNS hijacking) and exfiltrate data from compromised systems (DNS tunneling).

See also

How to Upgrade Your Cyber Risk Management Program with NIST

What are the Cybersecurity Risks and Familiar Sources of Cyber Threats?

Understanding threat actors and their tactics, techniques, and procedures is essential to respond effectively to any cyberattack. Attackers can include:

- **Nation-states.** Cyber attacks by a nation can disrupt communications, military activities, and everyday life.
- **Organized crime.** Criminal groups aim to infiltrate systems or networks for financial gain. These groups use phishing, spam, spyware, and malware to conduct identity theft, online fraud, and system extortion.
- **Hackers.** Hackers explore various cyber techniques to breach defenses and exploit vulnerabilities in a computer system or network. They are usually motivated by personal gain, revenge, stalking, financial gain, or political activism. Hackers may

develop new threats for the hacker community's thrill of challenge or bragging rights.

- **Terrorist groups.** Terrorists conduct cyberattacks to destroy, infiltrate, or exploit critical infrastructure to threaten national security, compromise military equipment, disrupt the economy, and cause mass casualties.
- **Insiders with malicious intent.** Insiders can be workers, contractors, third-party suppliers, or other business partners who have lawful access to company resources but abuse it to steal or destroy data for their own or others' financial or personal advantage.

Emerging Cyber Threats

The coronavirus pandemic created a huge challenge for businesses and IT organizations in 2020 and 2021. During the pandemic, cyber threats and data breaches proliferated and grew more sophisticated. Alas, that wave of innovation in cyber attacks will not recede in 2022 or beyond. So your organization should pay close attention to emerging threats, as well.

Pandemic-Related Attacks

Cybercriminals will probably continue to use COVID-19-related topics as themes for phishing and social engineering campaigns.

Over the past several years, these attacks have often coincided with significant events, such as a sudden surge in coronavirus cases or the announcement of a new vaccine. Threat actors lure users into clicking a malicious link or attachment disguised as pandemic-related.

Cloud Breaches

More and more companies are migrating to the cloud for remote working and to assure business continuity. Unfortunately, cybercriminals follow the same trend and frequently target the cloud.

Cloud-based security risks, including cloud misconfigurations, incomplete data deletion, and vulnerable cloud apps, will be the most common sources of cyberattacks.

IoT (Internet of Things) Attacks

Global organizations increasingly use “Internet of Things” (IoT) devices – really, sensors and other physical devices connected to the Internet – to accelerate operations, capture more data, manage infrastructure remotely, improve customer service, and more.

Examples of IoT technologies in the workplace include everything from smart thermostats and videoconferencing technologies to warehouse stock monitors and even “smart” vending machines that can order refills.

Many IoT devices, however, lack security features, putting them at risk of cyber attacks. Cybercriminals can exploit IoT vulnerabilities to gain control of devices for use in botnets and to penetrate your network.

What makes IoT technology so convenient is also what makes it so vulnerable: enhanced connectivity and convenience come with more security risks.

How Can Businesses Manage Cybersecurity Risks?

For organizations of all sizes, cybersecurity threats are growing increasingly severe. To guard against cyberattacks effectively, you'll need to implement a risk management program.

Cybersecurity risk management is the process of detecting, assessing, and managing an organization's IT security risks. In addition, IT workers must create a robust cybersecurity architecture that complies with pertinent regulations, standards, and best practices.

Developing a cybersecurity risk management strategy, and distinguishing between strategic versus operational risk, makes your entire firm more aware of cyber threats. Implementing a preventative approach can:

- Reduce the impact of cyberattacks and the harm brought on by cyber hazards
- Boost operational efficiency
- Safeguard company resources and earnings
- Improve your compliance with legal or regulatory obligations
- Boost the standing of the company with customers and other stakeholders

Creating a Framework for the Management of Cybersecurity Risks

This risk management program checklist will enhance your understanding of cybersecurity risks and your capacity to stop harmful assaults involving malware, phishing, and ransomware.

- Recognize the security environment
- Find any gaps
- Establish a team and delegate responsibilities
- Increase the importance of risk management education and awareness campaigns
- Put in place a risk management framework based on industry standards
- Create a program for assessing the risk to cyber security
- Make a business continuity and incident response plan

Take Control of Cybersecurity Risks with the ROAR Platform.

The ROAR Platform from Reciprocity works with governance, risk management, and changing compliance demands to keep you up-to-date and safe.

With ROAR Platform, a team of cybersecurity professionals is always looking out for your organization and its assets to assure that you get the best protection against security breaches and cyberattacks.

ROAR Platform's compliance, risk, and workflow management software is an intuitive, easy-to-understand platform that keeps track of your workflow and lets you find areas of high risk before that risk becomes a real threat.

Schedule a demo today for more information on how the ROAR Platform can help your organization anticipate cybersecurity threats. Worry-free compliance management is the Zen way.

Weaknesses:-

Weaknesses in security refer to vulnerabilities or flaws in an organization's or system's defenses that could be exploited by attackers to compromise the confidentiality, integrity, or availability of sensitive data or resources. Identifying and addressing these weaknesses is essential to enhance the overall security posture and reduce the risk of security breaches. Some common weaknesses in security include:

1. Outdated Software and Firmware:

Running outdated software, operating systems, or firmware can leave systems vulnerable to known security flaws that have been patched in newer versions.

2. Weak Authentication Mechanisms:

Using weak passwords, lack of multi-factor authentication, or allowing default credentials to remain unchanged can make it easier for attackers to gain unauthorized access.

3. Unpatched Vulnerabilities:

Failure to apply security patches and updates in a timely manner can leave systems exposed to known vulnerabilities that attackers may exploit.

4. Insufficient Access Controls:

Inadequate access controls may allow unauthorized users to gain access to sensitive data or perform actions they are not authorized to do.

5. Phishing and Social Engineering:

Human error remains a significant weakness in security, and attackers often use phishing emails and social engineering techniques to trick users into revealing sensitive information or granting access.

6. Lack of Security Awareness Training:

Employees who are not adequately trained in security best practices may inadvertently engage in risky behavior or fall victim to social engineering attacks.

7. Insecure Network Configurations:

Misconfigurations in network devices or firewalls can expose sensitive services or resources to the internet, making them potential targets for attackers.

8. Missing Encryption:

Failure to encrypt sensitive data, both in transit and at rest, can result in data exposure if attackers gain access to the data.

9. Weak Physical Security:

Inadequate physical security measures, such as unsecured server rooms or lack of access controls for sensitive areas, can lead to unauthorized physical access to critical systems.

10. Lack of Incident Response Plans:

Not having well-defined incident response plans and procedures can result in delayed or ineffective responses to security incidents.

Addressing these weaknesses requires a proactive approach to security, which includes regular security assessments, penetration testing, patch management, security awareness training for employees, and the implementation of security best practices and standards. Organizations

must continuously monitor their systems, networks, and processes to identify and remediate potential vulnerabilities to ensure a strong and resilient security posture.

Buffer overflow:-

What Is Buffer Overflow?

Buffer overflow is a software coding error or vulnerability that can be exploited by hackers to gain unauthorized access to corporate systems. It is one of the best-known software security vulnerabilities yet remains fairly common. This is partly because buffer overflows can occur in various ways and the techniques used to prevent them are often error-prone.

The software error focuses on buffers, which are sequential sections of computing memory that hold data temporarily as it is transferred between locations. Also known as a buffer overrun, buffer overflow occurs when the amount of data in the buffer exceeds its storage capacity. That extra data overflows into adjacent memory locations and corrupts or overwrites the data in those locations.

What Is a Buffer Overflow Attack?

A buffer overflow attack takes place when an attacker manipulates the coding error to carry out malicious actions and compromise the affected system. The attacker alters the application's execution path and overwrites elements of its memory, which amends the program's execution path to damage existing files or expose data.

A buffer overflow attack typically involves violating programming languages and overwriting the bounds of the buffers they exist on. Most buffer overflows are caused by the combination of manipulating memory and mistaken assumptions around the composition or size of data.

A buffer overflow vulnerability will typically occur when code:

1. Is reliant on external data to control its behavior
2. Is dependent on data properties that are enforced beyond its immediate scope
3. Is so complex that programmers are not able to predict its behavior accurately

Buffer Overflow Exploits

The buffer overflow exploit techniques a hacker uses depends on the architecture and operating system being used by their target. However, the extra data they issue to a program will likely contain malicious code that enables the attacker to trigger additional actions and send new instructions to the application.

For example, introducing additional code into a program could send it new instructions that give the attacker access to the organization's IT systems. In the event that an attacker knows a program's memory layout, they may

be able to intentionally input data that cannot be stored by the buffer. This will enable them to overwrite memory locations that store executable code and replace it with malicious code that allows them to take control of the program.

Attackers use a buffer overflow to corrupt a web application's execution stack, execute arbitrary code, and take over a machine. Flaws in buffer overflows can exist in both application servers and web servers, especially web applications that use libraries like graphics libraries. Buffer overflows can also exist in custom web application codes. This is more likely because they are given less scrutiny by security teams but are less likely to be discovered by hackers and more difficult to exploit.

Buffer Overflow Consequences

Common consequences of a buffer overflow attack include the following:

1. System crashes: A buffer overflow attack will typically lead to the system crashing. It may also result in a lack of availability and programs being put into an infinite loop.
2. Access control loss: A buffer overflow attack will often involve the use of arbitrary code, which is often outside the scope of programs' security policies.
3. Further security issues: When a buffer overflow attack results in arbitrary code execution, the attacker may use it to exploit other vulnerabilities and subvert other security services.

Types of Buffer Overflow Attacks

There are several types of buffer overflow attacks that attackers use to exploit organizations' systems. The most common are:

1. Stack-based buffer overflows: This is the most common form of buffer overflow attack. The stack-based approach occurs when an attacker sends data containing malicious code to an application, which stores the data in a stack buffer. This overwrites the data on the stack, including its return pointer, which hands control of transfers to the attacker.
2. Heap-based buffer overflows: A heap-based attack is more difficult to carry out than the stack-based approach. It involves the attack flooding a program's memory space beyond the memory it uses for current runtime operations.
3. Format string attack: A format string exploit takes place when an application processes input data as a command or does not validate input data effectively. This enables the attacker to execute code, read data in the stack, or cause segmentation faults in the application. This could trigger new actions that threaten the security and stability of the system.

Which Programming Languages Are More Vulnerable?

Nearly all applications, web servers, and web application environments are vulnerable to buffer overflows. Environments that are written in interpreted languages, such as Java and Python, are immune to the attacks, with the exception of overflows in their interpreter.

Buffer overflow attacks are typically caused by coding errors and mistakes in application development. This results in buffer overflow as the application does not allocate appropriately sized buffers and fails to check for overflow issues. These issues are particularly problematic in the

programming language C/C++ as it does not have buffer overflow protection built in.

This programming language is not the only one vulnerable to buffer overflow attacks. A buffer overflow program in Assembly, C, C++ or Fortran is also particularly vulnerable and more likely to enable attackers to compromise a system. However, applications written in JavaScript or Perl are typically less vulnerable to buffer overflow attacks.

How to Prevent Buffer Overflows

Application developers can prevent buffer overflows by building security measures into their development code, using programming languages that include built-in protection, and regularly testing code to detect and fix errors.

One of the most common methods for preventing buffer overflows is avoiding standard library functions that have not been bounds-checked, which includes gets, scanf, and strcpy. Another common method is to prevent buffer overruns by using bounds-checking that is enforced at runtime. This automatically checks that the data written to a buffer is within the appropriate boundaries.

Modern operating systems now deploy runtime protection that enables additional security against buffer overflows. This includes common protection like:

1. Address space layout randomization (ASLR): Buffer overflow attacks typically need to know where executable code is located. ASLR moves at random around locations of data regions to randomize address spaces, which makes overflow attacks almost impossible.

2. Data execution prevention: This method prevents an attack from being able to run code in non-executable regions by flagging areas of memory as executable or non-executable.
3. Structured exception handling overwrite protection (SEHOP): Attackers may look to overwrite the structured exception handling (SEH), which is a built-in system that manages hardware and software exceptions. They do this through a stack-based overflow attack to overwrite the exception registration record, which is stored on the program's stack. SEHOP prevents attackers' malicious code from being able to attack the SEH and use its overwrite exploitation technique.

Implementing security measures around development code and operating systems is not enough to protect organizations' systems. When a buffer overflow vulnerability is discovered, it is crucial to quickly patch the software and ensure it is made available to all users.

Buffer Overflow Attack Examples

A common buffer overflow example is when an attacker injects their malicious code into corrupted memory. Or they may simply take advantage of the buffer overflow and the adjacent memory corruption.

For example, a simple buffer overflow can be caused when code that relies on external data receives a 'gets()' function to read data in a stack buffer. The system cannot limit the data that is read by the function, which makes code safety reliant on users entering fewer than 'BUFSIZE' characters. This code could look like this:

```
"...
char buf[BUFSIZE];
gets(buf);
…"
```

Other buffer overflow attacks rely on user input to control behavior then add indirection through the memory function ‘memcpy()’. This accepts the destination buffer, source buffer, and amount of bytes to copy, fills the input buffer with the ‘read()’ command, and specifies how many bites for ‘memcpy()’ to copy.

“...
```
char buf[64], in[MAX_SIZE];
printf("Enter buffer contents:\n");
read(0, in, MAX_SIZE-1);
printf("Bytes to copy:\n");
scanf("%d", &bytes);
memcpy(buf, in, bytes);
```
…”

Another scenario for buffer overflow is when data properties are not verified locally. The function ‘lccopy()’ takes a string and returns a heap-allocated copy with uppercase letters changed to lowercase. The function does not perform bounds-checking as it expects ‘str’ to be smaller than ‘BUFSIZE’. An attacker can bypass the code or change the assumption of the size to overflow the buffer. An example of this code is:

“
```
char *lccopy(const char *str) {
  char buf[BUFSIZE];  char *p;


  strcpy(buf, str);
for (p = buf; *p; p++) {
    if (isupper(*p)) {
      *p = tolower(*p);
    }
  }
return strdup(buf);
}
```
”

Another example of buffer overflow is when code is too complex to predict its behavior. The below example is from the libPNG image decoder, which

is used by browsers like Mozilla and Internet Explorer. The code appears safe as it checks the variable-length size but performs a ‘png_ptr->mode’ check that makes it more complicated. This can result in blind length checks in the ‘png_crc_read()’ call, which shows the importance of minimizing the complexity of code in memory operations.

```
“if (!(png_ptr->mode & PNG_HAVE_PLTE)) {
  /* Should be an error, but we can cope with it */
  png_warning(png_ptr, "Missing PLTE before tRNS");}
else if (length > (png_uint_32)png_ptr->num_palette) {
  png_warning(png_ptr, "Incorrect tRNS chunk length");
  png_crc_finish(png_ptr, length);
  return;
}
...
png_crc_read(png_ptr, readbuf, (png_size_t)length);”
```

Brute force, Protocol:-

A **Brute force attack** is a well known breaking technique, by certain records, brute force attacks represented five percent of affirmed security ruptures. A brute force attack includes ‘speculating’ username and passwords to increase unapproved access to a framework. Brute force is a straightforward attack strategy and has a high achievement rate.
A few attackers use applications and contents as brute force devices. These instruments evaluate various secret word mixes to sidestep confirmation forms. In different cases, attackers attempt to get to web applications via scanning for the correct session ID. Attacker inspiration may incorporate taking data, contaminating destinations with malware, or disturbing help.

While a few attackers still perform brute force attacks physically, today practically all brute force attacks are performed by bots. Attackers have arrangements of usually utilized accreditations, or genuine client qualifications, got through security breaks or the dull web. Bots

deliberately attack sites and attempt these arrangements of accreditations, and advise the attacker when they obtain entrance.

Types of Brute Force Attacks:

1. **Dictionary attacks –** surmises usernames or passwords utilizing a dictionary of potential strings or phrases.
2. **Rainbow table attacks –** a rainbow table is a precomputed table for turning around cryptographic hash capacities. It very well may be utilized to figure a capacity up to a specific length comprising of a constrained arrangement of characters.
3. **Reverse brute force attack –** utilizes a typical password or assortment of passwords against numerous conceivable usernames. Focuses on a network of clients for which the attackers have recently acquired information.
4. **Hybrid brute force attacks –** begins from outer rationale to figure out which password variety might be destined to succeed, and afterward proceeds with the simple way to deal with attempt numerous potential varieties.
5. **Simple brute force attack –** utilizes an efficient way to deal with 'surmise' that doesn't depend on outside rationale.
6. **Credential stuffing –** utilizes beforehand known password-username sets, attempting them against numerous sites. Adventures the way that numerous clients have the equivalent username and password across various frameworks.

How to Prevent Brute Force Password Hacking ?

To protect your organization from brute force password hacking, enforce the use of strong passwords.

Passwords should:

- Never use information that can be found online (like names of family members).
- Have as many characters as possible.
- Combine letters, numbers, and symbols.
- Avoid common patterns.
- Be different for each user account.
- Change your password periodically
- Use strong and long password
- Use multifactor authentication

Cross site:-

Cross-Site Scripting (XSS) is a type of security vulnerability commonly found in web applications. It occurs when an attacker injects malicious scripts into web pages that are viewed by other users. XSS attacks exploit the trust that a web application has for user-provided data, allowing attackers to execute arbitrary scripts in the context of legitimate users' browsers. This can lead to various security issues, such as data theft, session hijacking, and malware distribution.

There are three main types of XSS attacks:

1. Stored XSS (Persistent XSS):
   In a stored XSS attack, the malicious script is permanently stored on the target server, typically in a database or other data storage. When a user requests the infected web page, the server includes the malicious script in the response, which is then executed by the user's browser.

2. Reflected XSS (Non-Persistent XSS):
   In a reflected XSS attack, the malicious script is embedded in a URL or form input and sent to the server as part of the request. The server then reflects the script back in the response, and the browser executes it. The payload is not permanently stored on the server.

3. DOM-based XSS:
   DOM-based XSS occurs when the malicious script manipulates the Document Object Model (DOM) of a web page directly, rather than relying on server-side vulnerabilities. The attack takes place entirely in the client-side code, and the payload is usually in the URL or other client-side data.

Preventing XSS Attacks:
To prevent XSS attacks, web developers and application security professionals can implement the following best practices:

1. Input Validation:

Validate and sanitize all user-supplied data before displaying it on web pages. Input validation ensures that only expected and safe data is accepted, reducing the risk of script injection.

2. Output Encoding:
Encode all user-generated content and dynamic data before rendering it in web pages. Encoding converts special characters into their respective HTML entities, preventing them from being interpreted as executable scripts.

3. HTTP Security Headers:
Utilize security headers like Content Security Policy (CSP), which helps prevent malicious scripts from running by specifying which sources are allowed to execute scripts on a web page.

4. Session Management:
Implement secure session management practices to prevent session hijacking and unauthorized access.

5. Regular Security Audits:
Conduct periodic security audits and vulnerability assessments to identify and address potential XSS vulnerabilities in the application.

6. Security Awareness Training:
Educate developers and application users about the risks of XSS attacks and security best practices to minimize the likelihood of successful attacks.

By following these preventive measures, web application developers can significantly reduce the risk of XSS vulnerabilities and create more secure and resilient web applications.

Spoofing:-

Spoofing is a cybersecurity attack technique where an attacker impersonates another person, device, or system to deceive or manipulate a target. The goal of spoofing is to gain unauthorized access to information, bypass security controls, or

trick users into providing sensitive data or performing certain actions. Spoofing attacks can occur in various forms, targeting different aspects of communication and trust within a network or system. Some common types of spoofing attacks include:

1. IP Spoofing:

IP spoofing involves forging the source IP address in a network packet to make it appear as if the packet originated from a trusted source. This technique is often used in Denial of Service (DoS) attacks, where attackers send packets with fake source IP addresses to overwhelm and disrupt a target system.

2. MAC Address Spoofing:

In MAC address spoofing, attackers change the Media Access Control (MAC) address of their network interface to impersonate another device on the network. This can be used to bypass MAC-based access controls or launch man-in-the-middle attacks.

3.Email Spoofing:

Email spoofing involves forging the "From" field in an email to make it appear as if the message was sent from a different sender. Attackers may use email spoofing for phishing attacks, social engineering, or to distribute malware.

4. DNS Spoofing (DNS Cache Poisoning):

DNS spoofing manipulates DNS records to redirect users to malicious websites or fraudulent pages. This can be used to conduct phishing attacks or redirect users to fake login pages.

5. ARP Spoofing (ARP Poisoning):

Address Resolution Protocol (ARP) spoofing involves sending fake ARP messages to associate an attacker's MAC address with the IP address of a legitimate device on the local network. This allows the attacker to intercept and manipulate network traffic.

6. Caller ID Spoofing:

Caller ID spoofing is commonly used in phone scams, where attackers manipulate the caller ID to display a different number or a familiar one to deceive the recipient into answering the call.

Spoofing attacks exploit the trust and assumptions that systems and users make about the authenticity of information and the identity of the sender. To mitigate spoofing attacks, it is essential to implement strong authentication mechanisms, use encryption for sensitive communications, and deploy security protocols and mechanisms that verify the integrity and authenticity of data and entities. Additionally, security awareness training for users can help them recognize and respond to potential spoofing attempts and reduce the risk of falling victim to such attacks.

Denial of Service attacks:-

A **Denial-of-Service (DoS) attack** is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash. In both instances, the DoS attack deprives

legitimate users (i.e. employees, members, or account holders) of the service or resource they expected.

Victims of DoS attacks often target web servers of high-profile organizations such as banking, commerce, and media companies, or government and trade organizations. Though DoS attacks do not typically result in the theft or loss of significant information or other assets, they can cost the victim a great deal of time and money to handle.

There are two general methods of DoS attacks: flooding services or crashing services. Flood attacks occur when the system receives too much traffic for the server to buffer, causing them to slow down and eventually stop. Popular flood attacks include:

- **Buffer overflow attacks** – the most common DoS attack. The concept is to send more traffic to a network address than the programmers have built the system to handle. It includes the attacks listed below, in addition to others that are designed to exploit bugs specific to certain applications or networks
- **ICMP flood** – leverages misconfigured network devices by sending spoofed packets that ping every computer on the targeted network, instead of just one specific machine. The network is then triggered to amplify the traffic. This attack is also known as the smurf attack or ping of death.
- **SYN flood** – sends a request to connect to a server, but never completes the handshake. Continues until all open ports are saturated with requests and none are available for legitimate users to connect to.

Other DoS attacks simply exploit vulnerabilities that cause the target system or service to crash. In these attacks, input is sent that takes advantage of bugs in the target that subsequently crash or severely destabilize the system, so that it can't be accessed or used.

An additional type of DoS attack is the Distributed Denial of Service (DDoS) attack. A DDoS attack occurs when multiple systems orchestrate a synchronized DoS attack to a single target. The essential difference is that instead of being attacked from one location, the target is attacked from many locations at once. The distribution of hosts that defines a DDoS provide the attacker multiple advantages:

- He can leverage the greater volume of machine to execute a seriously disruptive attack
- The location of the attack is difficult to detect due to the random distribution of attacking systems (often worldwide)
- It is more difficult to shut down multiple machines than one
- The true attacking party is very difficult to identify, as they are disguised behind many (mostly compromised) systems

Modern security technologies have developed mechanisms to defend against most forms of DoS attacks, but due to the unique characteristics of DDoS, it is still regarded as an elevated threat and is of higher concern to organizations that fear being targeted by such an attack.