

Digital Signature

Digital Signature

- A digital signature is a mathematical technique which validates the authenticity and integrity of a message, software or digital documents.
- It allows us to verify the author name, date and time of signatures, and authenticate the message contents
- The digital signature offers far more inherent security and intended to solve the problem of tampering and impersonation (Intentionally copy another person's characteristics) in digital communications.

Application of Digital Signature

- The important reason to implement digital signature to communication is:
 - Authentication
 - Non-repudiation
 - Integrity

Application of Digital Signature

- **Authentication**

- Authentication is a process which verifies the identity of a user who wants to access the system.
- In the digital signature, authentication helps to authenticate the sources of messages.

- **Non-repudiation**

- means assurance of something that cannot be denied.
- It ensures that someone to a contract or communication cannot later deny the authenticity of their signature on a document or in a file or the sending of a message that they originated.

Application of Digital Signature

- **Integrity**
- Integrity ensures that the message is real, accurate and safeguards from unauthorized user modification during the transmission.

Algorithms in Digital Signature

- A digital signature consists of three algorithms:
- **Key generation algorithm**
 - selects private key randomly from a set of possible private keys.
 - provides the private key and its corresponding public key.
- **Signing algorithm**
 - A signing algorithm produces a signature for the document.
- **Signature verifying algorithm**
 - A signature verifying algorithm either accepts or rejects the document's authenticity.

How digital signatures work

- created and verified by using public key cryptography, also known as asymmetric cryptography.
- By the use of a public key algorithm, such as RSA, one can generate two keys that are mathematically linked- one is a private key, and another is a public key.
- The user who is creating the digital signature uses their own private key to encrypt the signature-related document
- There is only one way to decrypt that document is with the use of signer's public key.

How digital signatures work

- This technology requires all the parties to trust that the individual who creates the signature has been able to keep their private key secret.
- If someone has access the signer's private key, there is a possibility that they could create fraudulent signatures in the name of the private key holder.

steps in creating a digital signature are:

1. Select a file to be digitally signed.
2. The hash value of the message or file content is calculated. This message or file content is encrypted by using a private key of a sender to form the digital signature.
3. Now, the original message or file content along with the digital signature is transmitted.
4. The receiver decrypts the digital signature by using a public key of a sender.
5. The receiver now has the message or file content and can compute it.
6. Comparing these computed message or file content with the original computed message. The comparison needs to be the same for ensuring integrity.

Types of Digital Signature

- Different document processing platform supports different types of digital signature.
- They are described below:



Certified Signatures

- signature documents display a unique blue ribbon across the top of the document
- The certified signature contains the name of the document signer and the certificate issuer which indicate the authorship and authenticity of the document.

Approval Signatures

- The approval digital signatures on a document can be used in the organization's business workflow
- They help to optimize the organization's approval procedure.
- The procedure involves capturing approvals made by us and other individuals and embedding them within the PDF document.
- The approval signatures to include details such as an image of our physical signature, location, date, and official seal.

Visible Digital Signature

- The visible digital signature allows a user to sign a single document digitally.
- This signature appears on a document in the same way as signatures are signed on a physical document.

Invisible Digital Signature

- The invisible digital signatures carry a visual indication of a blue ribbon within a document in the taskbar
- use invisible digital signatures when we do not have or do not want to display our signature but need to provide the authenticity of the document, its integrity, and its origin.

Digital Certificate

Digital Certificate

- It is also known as a public key certificate
- It is used to cryptographically link ownership of a public key with the entity that owns it
- Digital certificates are for sharing public keys to be used for encryption and authentication

Digital Certificate

- Digital certificates
 - include the public key being certified,
 - identifying information about the entity that owns the public key,
 - metadata relating to the digital certificate and
 - a digital signature of the public key the certificate issuer created.
- The distribution, authentication and revocation of digital certificates are the primary functions of the PKI, the system that distributes and authenticates public keys.

Digital Certificate

- Public key cryptography depends on key pairs:
 - one private key to be held by the owner and used for signing and decrypting and
 - one public key that can be used for encrypting data sent to the public key owner or authenticating the certificate holder's signed data.
- The digital certificate enables entities to share their public key so it can be authenticated.

Digital Certificate

- A digital certificate contains the following identifiable information:
 - The name of the user
 - The department or company to which the user belongs
 - The internet protocol (IP) address or serial number associated with the device
 - A copy of the public key obtained from a certificate holder
 - The period during which the certificate will be valid
 - The domain that the certificate is authorized to represent

Digital Certificate

- Digital certificates are
 - used in public key cryptography functions most commonly for initializing SSL connections between web browsers and web servers.
 - also used for sharing keys used for public key encryption and authentication of digital signatures
- Digital certificates that are supported by mobile operating environments, laptops, tablet computers, IoT devices, and networking and software applications help protect websites, wireless networks and virtual private networks

How are digital certificates used?

- Digital certificates are used in the following ways:
- **Credit and debit cards**
 - use chip-embedded digital certificates that connect with merchants and banks
 - to ensure that the transactions performed are secure and authentic.
- **Digital payment companies**
 - to authenticate their automated teller machines and point-of-sale equipment in the field with a central server in their data center

How are digital certificates used?

- **Websites** use digital certificates for domain validation to show they are trusted and authentic
- **Secure email**
 - to identify one user to another and may also be used for electronic document signing.
 - The sender digitally signs the email, and the recipient verifies the signature.
- **Computer hardware manufacturers** embed digital certificates into cable modems to help prevent the theft of broadband service through device cloning

Why digital certificates used?

- As cyber threats increase,
 - more companies are considering attaching digital certificates to all of the IoT devices that operate at the edge and within their enterprises
- The goals are to prevent cyber threats and protect intellectual property.

Who can issue a digital certificate?

- An entity
 - can create its own PKI and issue its own digital certificates, creating a self-signed certificate
 - This approach might be reasonable when an organization maintains its own PKI to issue certificates for its own internal use.
- But certificate authorities (CAs)
 - considered trusted third parties in the context of a PKI issue digital certificates.
 - to issue digital certificates enables individuals to extend their trust in the CA to the digital certificates it issues.

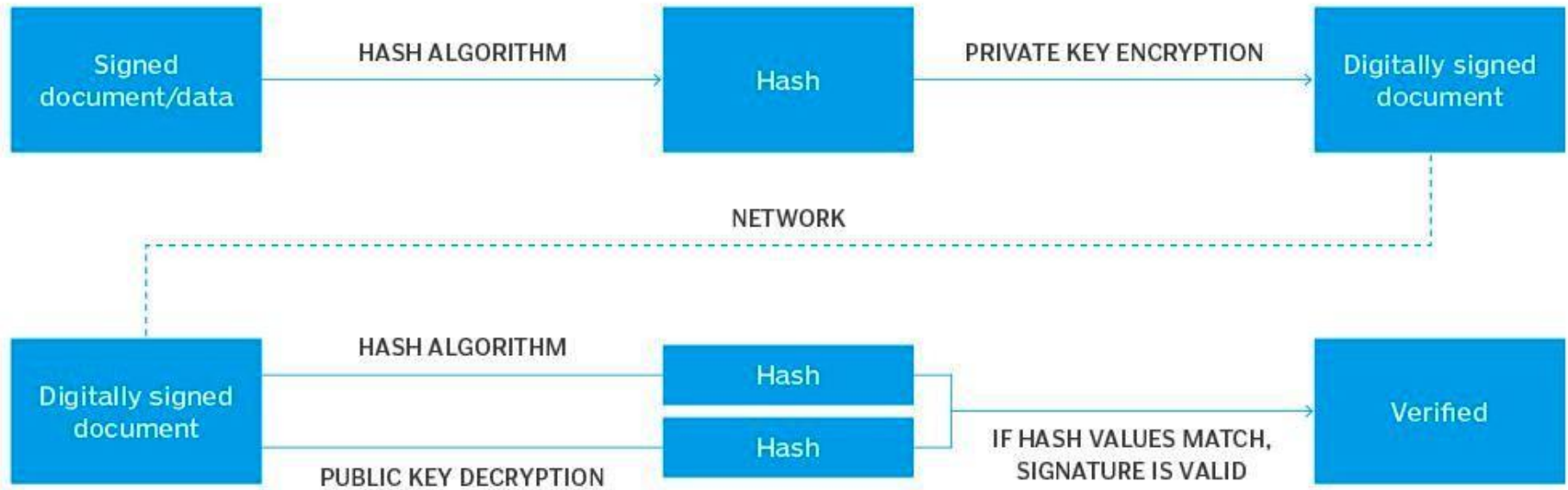
Digital certificates vs. digital signatures

- Public key cryptography supports several different functions, including
 - encryption and authentication, and
 - enables a digital signature.
- Digital signatures are generated using algorithms for signing data so a recipient can irrefutably confirm the data was signed by a particular public key holder.

Digital certificates vs. digital signatures

- Digital signatures
 - are generated by hashing the data to be signed with a one-way cryptographic hash;
 - the result is then encrypted with the signer's private key.
 - The digital signature incorporates this encrypted hash, which can only be authenticated, or verified, by using the sender's public key to decrypt the digital signature and then running the same one-way hashing algorithm on the content that was signed.
 - The two hashes are then compared.
 - If they match, it proves that the data was unchanged from when it was signed and that the sender is the owner of the public key pair used to sign it.

Digital certificates vs. digital signatures



Digital certificates vs. digital signatures

- A digital signature can depend on the distribution of a public key in the form of a digital certificate,
- but it is not mandatory that the public key be transmitted in that form.
- However, digital certificates are signed digitally, and they should not be trusted unless the signature can be verified.

Types of digital certificates

- Web servers and web browsers use digital certificates to authenticate over the internet.
- These digital certificates are
 - used to link a web server for a domain to the individual or organization that owns the domain.
 - usually referred to as SSL certificates even though the Transport Layer
- The three types are the following:
 - Domain-validated (DV) SSL
 - Organization-validated (OV) SSL
 - Extended validation (EV) SSL

Domain-validated (DV) SSL

- **DV SSL certificates**

- offer the least amount of assurance about the holder of the certificate
- Applicants for DV SSL certificates need only demonstrate that they have the right to use the domain name
- While these certificates can ensure the certificate holder is sending and receiving data,
- they provide no guarantees about who that entity is.

Organization-validated (OV) SSL

- **OV SSL certificates**

- provide additional assurances about the certificate holder
- They confirm that the applicant has the right to use the domain
- OV SSL certificate applicants also undergo additional confirmation of their ownership of the domain

Extended validation (EV) SSL

- **EV SSL certificates**

- are issued only after the applicant proves their identity to the CA's satisfaction
- The vetting process verifies the existence of the entity applying for the certificate,
- ensures that identity matches official records and is authorized to use the domain, and
- confirms that the domain owner has authorized issuance of the certificate.

Types of digital certificates

- The exact methods and criteria CAs follow to provide these types of SSL certificates for web domains is evolving as the CA, industry adapts to new conditions and applications.
- There are also other types of digital certificates used for different purposes:
 - Code signing certificates
 - Client certificates

Code signing certificates

- Code signing certificates
 - may be issued to organizations or individuals who publish software.
 - These certificates are used to share public keys that sign software code, including patches and software updates.
 - Code signing certificates certify the authenticity of the signed code.

Client certificates

- Client certificates
 - also called a digital ID,
 - are issued to individuals to bind their identity to the public key in the certificate
 - Individuals can use these certificates to digitally sign messages or other data
 - They can also use their private keys to encrypt data that recipients can decrypt using the public key in the client certificate.

Digital certificate benefits

- **Privacy**

- When you encrypt communications, digital certificates safeguard sensitive data and prevent the information from being seen by those unauthorized to view it.
- This technology protects companies and individuals with large troves of sensitive data.

- **Ease of use**

- The digital certification process is largely automated.

Digital certificate benefits

- **Cost effectiveness**

- Compared to other forms of encryption and certification, digital certificates are cheaper.
- Most digital certificates cost less than \$100 annually.

- **Flexibility**

- Digital certificates do not have to be purchased from a CA.
- For organizations that are interested in creating and maintaining their own internal pool of digital certificates, a do-it-yourself approach to digital certificate creation is feasible.

Digital certificate limitations

- **Security**

- Like any other security deterrent, digital certificates can be hacked.
- The most logical way for a mass hack to occur is if the issuing digital CA is hacked
- This gives bad actors an on-ramp into penetrating the repository of digital certificates the authority hosts.

- **Slow performance**

- It takes time to authenticate digital certificates and to encrypt and decrypt.
- The wait time can be frustrating.

Digital certificate limitations

- **Integration**

- Digital certificates are not standalone technology
- To be effective, they must be properly integrated with systems, data, applications, networks and hardware.
- This is no small task.

- **Management**

- The more digital certificates a company uses, the greater the need to manage them and to track which ones are expiring and need to be renewed.
- Third parties can provide these services, or companies can opt to do the job themselves.
- But it can be expensive.

Information Technology Act, 2000

Information Technology Act, 2000

- Known as an IT Act
- is an act proposed by the Indian Parliament reported on 17th October 2000.
- This Information Technology Act is based on the United Nations Model Law on Electronic Commerce 1996 (UNCITRAL Model) which was suggested by the General Assembly of the United Nations by a resolution dated 30th January 1997.
- It is the most important law in India dealing with Cybercrime and E-Commerce.

Information Technology Act, 2000

- The main objective of this act is
 - to carry out lawful and trustworthy electronic, digital, and online transactions and
 - alleviate or reduce cybercrimes.
- The IT Act has 13 chapters and 94 sections.
- The IT Act, of 2000 has two schedules:
 - **First Schedule:** Deals with documents to which the Act shall not apply.
 - **Second Schedule:** Deals with electronic signature or electronic authentication method.

Features of The IT Act 2000

- The digital signature has been changed to an electronic signature to make it a greater generation-impartial act.
- It elaborates on offenses, penalties, and breaches.
- It outlines the Justice Dispensation Systems for cyber crimes.
- The Information Technology Act defines in a new segment that a cyber cafe is any facility wherein access to the net is offered by any person inside the normal business to the general public.

Features of The IT Act 2000

- It offers the constitution of the Cyber Regulations Advisory Committee.
- The Act is based totally on The Indian Penal Code, of 1860, The Indian Evidence Act, of 1872, The Bankers' Books Evidence Act, of 1891, The Reserve Bank of India Act, of 1934, and many others.
- It adds a provision to Section 81, which states that the provisions of the Act shall have overriding effect. The provision states that nothing contained inside the Act shall limit any person from exercising any right conferred under the Copyright Act, of 1957.

The Offenses and the Punishments in IT Act 2000

- Tampering with the computer source documents.
- Directions of Controller to a subscriber to extend facilities to decrypt information.
- Publishing of information that is obscene in electronic form.
- Penalty for breach of confidentiality and privacy.
- Hacking for malicious purposes.
- Penalty for publishing Digital Signature Certificate false in certain particulars.
- Penalty for misrepresentation.

The Offenses and the Punishments in IT Act 2000

- Confiscation.
- Power to investigate offenses.
- Protected System.
- Penalties for confiscation are not to interfere with other punishments.
- Act to apply for offense or contravention committed outside India.
- Publication for fraud purposes.
- Power of Controller to give directions.

SECTION	PUNISHMENT
Section 43	<p>This section of the IT Act, 2000 states that any act of destroying, altering, or stealing a computer system/network or deleting data with malicious intentions without authorization from the owner of the computer is liable for the payment to be made to the owner as compensation for damages.</p>
Section 43A	<p>This section of the IT Act, 2000 states that any corporate body dealing with sensitive information that fails to implement reasonable security practices causing the loss of another person will also be liable as a convict for compensation to the affected party.</p>

SECTION	PUNISHMENT
Section 66	Hacking a Computer System with malicious intentions like fraud will be punished with 3 years imprisonment or a fine of Rs.5,00,000 or both.
Section 66 B, C, D	Fraud or dishonesty using or transmitting information or identity theft is punishable with 3 years imprisonment or a Rs. 1,00,000 fine or both.
Section 66 E	This Section is for Violation of privacy by transmitting an image of a private area is punishable with 3 years imprisonment or a 2,00,000 fine or both.

SECTION	PUNISHMENT
Section 66 F	This Section is on Cyber Terrorism affecting the unity, integrity, security, and sovereignty of India through digital mediums is liable for life imprisonment.
Section 67	This section states publishing obscene information or pornography or transmission of obscene content in public is liable for imprisonment of up to 5 years or a fine of Rs. 10,00,000 or both.

Malware Reverse Engineering

Types of Malware Analysis

- There are three types of malware analysis that can be conducted:
- Static malware analysis
- Dynamic malware analysis
- Hybrid malware analysis

Static Malware Analysis

- Examines files for signs of malicious intent
- A basic static analysis does not require a malware code that is actually running
- It is useful for revealing malicious infrastructure, packed files, or libraries
- Technical indicators like file names, hashes, strings such as IP addresses, domains, and file header data are identified

Static Malware Analysis

- Various tools like disassemblers and network analyzers have the ability to observe the malware without running it
- These tools can gather information on how the particular malware works
- Since static malware analysis does not run the malware code, there can be malicious runtime behavior in some sophisticated malware, which can go undetected

Static Malware Analysis

- **Example** : a file that generates a string and downloads a malicious file depending on the dynamic string
- The malware could go undetected if a basic static malware analysis is used
- In these cases, dynamic analysis is more helpful in getting a complete understanding of the file behavior

Dynamic Malware Analysis

- A suspected malicious code is run in a safe environment called a sandbox
- This isolated VM is a closed system that allows security experts to observe the malware closely in action without the risk of system or network infection
- This technique provides deeper visibility of the threat and its true nature

Dynamic Malware Analysis

- Automated sandboxing, eliminates the time, which otherwise would have been spent for reverse engineering a file to discover a malicious code
- Dynamic analysis can be a challenge, especially against smart adversaries who know sandboxes will be used eventually
- So, as a form of deception, adversaries hide their code in a way that it remains dormant until specific conditions are met. The code will run only then.

Hybrid Malware Analysis

- We already know now that basic static analysis isn't reliable when the malware has a more sophisticated code, and sophisticated malware are sometimes, able to avoid detection by sandbox technology.
- Combining both types of malware analysis techniques offers the best of both approaches
- Hybrid analysis can detect hidden malicious code, and extract many more IOCs by statically and previously unseen code

Hybrid Malware Analysis

- It is capable of detecting unknown threats, even from the most sophisticated malware
- The hybrid analysis applies static analysis to the data that is generated by behavioral analysis
- Consider a piece of malicious code that runs and causes some changes in memory. The dynamic analysis will be able to detect that and Analysts will immediately know to perform static analysis on that memory dump. This will result in more Indicators of Compromise (IOC)s and exposed zero-day exploits.

Static Vs Dynamic Malware Analysis

- **Analysis**
- Static malware analysis analyzes a malware sample without executing it, eliminating the need for an analyst through each and every phase.
- It observes the behavior of the sample and determines its capability and the extent to which it can exert damage to the system
- Dynamic analysis, performs analysis using the behavior and actions of the malware sample, which means that it works during the execution of the code with proper monitoring.

Static Vs Dynamic Malware Analysis

- **Technique**
- Static analysis involves signature analysis of the malware binary file.
- The binary file has a unique identifier and can be reverse-engineered with the help of a disassembler such as IDA that converts the machine-executable code into assembly language code.
- Some of the techniques used in this type of malware analysis are virus scanning, packer detection, file fingerprinting, debugging, and memory dumping.

Static Vs Dynamic Malware Analysis

- **Technique**
- Dynamic analysis involves a sandbox environment so that analyzing the behavior of malware while running the program won't affect other systems.
- Commercial sandboxes replace manual analysis with automated analysis.

Static Vs Dynamic Malware Analysis

- Approach
- Static analysis has a signature-based approach when it comes to malware detection and analysis.
- The unique identifier in malware is a sequence of bytes.
- The signatures are scanned using different patterns.
- The antimalware programs that are signature-based are effective only against common malware.

Static Vs Dynamic Malware Analysis

- **Approach**
- These are ineffective when it comes to sophisticated and advanced malware. This is where dynamic malware analysis comes into the picture.
- The dynamic analysis doesn't have a signature-based approach. Instead, it uses a behavior-based approach that determines the functionality of the malware.
- It involves studying the actions performed by the malware.

Static Vs Dynamic Malware Analysis

Static vs. Dynamic Malware Analysis: Comparison Chart

Static Malware Analysis	VS	Dynamic Malware Analysis
Static analysis is a process of analyzing a malware binary code without actually running the code		Dynamic analysis requires the malware program to be executed in a closely monitored virtual environment.
It uses a signature-based approach for malware analysis		It uses a behavior-based approach for malware detection and analysis.
It involves file fingerprinting, virus scanning, reverse-engineering the binary, file obfuscation, analyzing memory artifacts, packer detection, and debugging.		Dynamic analysis involves API calls, instruction traces, registry changes, network and system calls, memory writes, and more.
It is ineffective against sophisticated malware program and codes.		It is effective against all types of malware because it analyzes the sample by executing it.

Malware Analysis Use Cases

- **Malware Detection**
- More and more sophisticated techniques are being used by adversaries to evade traditional detection mechanisms.
- Threats can be more effectively detected through deep behavioral analysis by identifying shared code, malicious functionality, or infrastructure.
- Additionally, malware analysis results in the extraction of IOCs.

Malware Analysis Use Cases

- **Malware Detection**
- These IOCs can then be fed into threat intelligence platforms (TIPs), SEIMs, and security orchestration tools for alerting teams to related threats in the future.

Malware Analysis Use Cases

- **Threat Hunting**
- Threat hunters can use the behavior and artifacts that are exposed by malware analysis to find similar activities, like accessing a particular network connection, domain, or port.
- Searching firewall, proxy logs, or SIEM data can help find similar threats.

Malware Analysis Use Cases

- **Threat Alerts and Triage**
- The outputs of malware analysis offer higher-fidelity alerts early in the attack life cycle thus saving time by triaging the results of these alerts.
- **Incident Response (IR)**
- The objective of the IR team is to perform root cause analysis, determine the impact, and successfully offer remediation and recovery solutions.
- Malware analysis helps in the efficacy of this effort.

Malware Analysis Use Cases

- **Malware Research**
- All industry and academic malware researchers apply malware analysis to achieve insights on the latest techniques, tools, and exploits used by adversaries.

Malware Reverse Engineering

What is malware?

- Malware, or malicious software, is any program or file that's intentionally harmful to a computer, network or server.
- These malicious programs
 - Steal and encrypt data
 - Delete sensitive data;
 - Alter or hijack core computing functions, and
 - Monitor end users' computer activity
- Types of malware include computer viruses, worms, Trojan horses, Ransomware and spyware.

What is the intent of malware?

- **Intelligence and intrusion**
- Exfiltrates data such as emails, plans, and especially sensitive information like passwords.
- **Disruption and extortion**
- Locks up networks and PCs, making them unusable
- If it holds your computer hostage for financial gain, it's called ransomware.

What is the intent of malware?

- **Destruction**
 - Destroys computer systems to damage your network infrastructure.
- **Steal computer resources**
 - Uses your computing power to run botnets, cryptomining programs (cryptojacking), or send spam emails.
- **Monetary gain**
 - Sells your organization's intellectual property on the dark web.

What does malware do?

- Malware can infect networks and devices and is designed to harm those devices, networks and their users in some way
- Depending on the type of malware and its goal, this harm might present itself differently to the user or endpoint
- In some cases, the effect of malware is relatively mild and benign, and in others, it can be disastrous.
- Malware can typically perform the following harmful actions:

Data exfiltration

- Data exfiltration is a common objective of malware
- During data exfiltration, once a system is infected with malware, threat actors can steal sensitive information stored on the system, such as emails, passwords, intellectual property, financial information and login credentials
- Data exfiltration can result in monetary or reputational damage to individuals and organizations.

Service disruption

- Malware can disrupt services in several ways
- For example, it can lock up computers and make them unusable or hold them hostage for financial gain by performing a ransomware attack
- Malware can also
 - target critical infrastructure, such as power grids, healthcare facilities or
 - transportation systems to cause service disruptions.

Data espionage

- A type of malware known as spyware performs data espionage by spying on users
- Typically, hackers use
 - keyloggers to record keystrokes,
 - access web cameras and microphones and capture screenshots

Identity theft

- Malware can be used to steal personal data which can be used to impersonate victims, commit fraud or gain access to additional resources
- According to the IBM X-Force Threat Intelligence Index 2024, there was a 71% rise in cyberattacks using stolen identities in 2023 compared to the previous year.

Stealing resources

- Malware can use stolen system resources
 - to send spam emails,
 - operate botnets and
 - run cryptomining software, also known as cryptojacking.

System damage

- Certain types of malware, such as computer worms,
 - damage devices by corrupting the system files,
 - deleting data or changing system settings.
- This damage can lead to an unstable or unusable system.

How do malware infections happen?

- Malware use a variety of physical and virtual means to spread malware that infects devices and networks
- including the following:
 - Removable drives
 - Infected websites
 - Phishing attacks
 - Obfuscation techniques
 - Software from third-party websites

Removable drives

- Malicious programs can be delivered to a system with a USB drive or external hard drive.
- For example, malware can be automatically installed when an infected removable drive connects to a PC

Infected websites

- Malware
 - find its way into a device through popular collaboration tools and drive-by downloads,
 - which automatically download programs from malicious websites to systems without the user's approval or knowledge.

Phishing attacks

- use phishing emails disguised as legitimate messages containing malicious links or attachments to deliver the malware executable file to unsuspecting users
- Sophisticated malware attacks often use a command-and-control server that
 - lets threat actors communicate with the infected systems, exfiltrate sensitive data and
 - even remotely control the compromised device or server

Obfuscation techniques

- Emerging strains of malware include new evasion and obfuscation techniques designed to fool users, security administrators and antimalware products
- Some of these evasion techniques rely on simple tactics, such as using web proxies to hide malicious traffic or source Internet Protocol (IP) addresses

Obfuscation techniques

- More sophisticated cyberthreats include
 - polymorphic malware that can repeatedly change its underlying code to avoid detection from signature-based detection tools;
 - **anti-sandbox techniques** that enable malware to detect when it's being analyzed and to delay execution until after it leaves the sandbox; and
 - fileless malware that resides only in the system's RAM to avoid being discovered

Software from third-party websites

- There are instances where malware can be downloaded and installed on a system concurrently with other programs or apps
- Typically, software from third-party websites or files shared over peer-to-peer networks falls under this category

Software from third-party websites

- For example, a computer running a Microsoft operating system (OS) might end up unknowingly installing software that Microsoft would deem as a potentially unwanted program (PUP)
- However, by checking a box during the installation, users can avoid installing unwanted software.

Types of Malware

Virus

- that attaches to another program and,
- when executed usually inadvertently by the user replicates itself by modifying other computer programs and
- infecting them with its own bits of code.

Worms

- Worms are a type of malware similar to viruses
- Like viruses, worms are self-replicating
- The big difference is that
 - worms can spread across systems on their own, whereas viruses need some sort of action from a user in order to initiate the infection.

Trojan or Trojan horse

- one of the most dangerous malware types
- It usually represents itself as something useful in order to trick you
- Once it's on your system, the attackers behind the Trojan gain unauthorized access to the affected computer
- Trojans can be used to steal financial information or install other forms of malware, often ransomware

Ransomware

- Ransomware is a form of malware that locks you out of your device and/or encrypts your files, then forces you to pay a ransom to regain access.
- called the cybercriminal's weapon of choice because it demands a quick, profitable payment in hard-to-trace cryptocurrency.
- The code behind ransomware is easy to obtain through online criminal marketplaces and defending against it is very difficult

Ransomware

- While ransomware attacks on individual consumers are down at the moment, attacks on businesses are up 365 percent for 2019.
- As an example, the Ryuk ransomware specifically targets high-profile organizations that are more likely to pay out large ransoms.

Rootkit

- is a form of malware that provides the attacker with administrator privileges on the infected system, also known as “root” access.
- Typically, it is also designed to stay hidden from the user, other software on the system, and the operating system itself.

Backdoor virus

- A backdoor virus or remote access Trojan (RAT) secretly creates a backdoor into an infected computer system
- that lets threat actors remotely access it without alerting the user or the system's security programs.

Adware

- Adware is unwanted software designed to throw advertisements up on your screen, most often within a web browser
- Typically, it uses an underhanded method to either disguise itself as legitimate, or piggyback on another program to trick you into installing it on your PC, tablet, or mobile device.

Keylogger

- is malware that records all the user's keystrokes on the keyboard,
- typically storing the gathered information and sending it to the attacker,
- who is seeking sensitive information like usernames, passwords, or credit card details

Logic Bombs

- This type of malicious malware is designed to cause harm and typically gets inserted into a system once specific conditions are met
- Logic bombs stay dormant and are triggered when a certain event or condition is met, such as when a user takes a specific action on a certain date or time.

Exploits

- Exploits are a type of malware that takes advantage of bugs and vulnerabilities in a system in order to give the attacker access to your system
- While there, the attacker might steal your data or drop some form of malware
- A zero-day exploit refers to a software vulnerability for which there is currently no available defense or fix

Exploits

- Exploits are a type of malware that takes advantage of bugs and vulnerabilities in a system in order to give the attacker access to your system
- While there, the attacker might steal your data or drop some form of malware
- A zero-day exploit refers to a software vulnerability for which there is currently no available defense or fix

How to protect against malware

- Pay attention to the domain and be wary if the site isn't a top-level domain, i.e., com, mil, net, org, edu, or biz, to name a few.
- Use strong passwords with multi-factor authentication. A password manager can be a big help here.
- Avoid clicking on pop-up ads while browsing the Internet.
- Avoid opening email attachments from unknown senders.

How to protect against malware

- Do not click on strange, unverified links in emails, texts, and social media messages.
- Don't download software from untrustworthy websites or peer-to-peer file transfer networks.
- Stick to official apps from Google Play and Apple's App Store on Android, OSX, and iOS (and don't jailbreak your phone). PC users should check the ratings and reviews before installing any software.
- Make sure your operating system, browsers, and plugins are patched and up to date.

How to protect against malware

- Delete any programs you don't use anymore.
- Back up your data regularly. If your files become damaged, encrypted, or otherwise inaccessible, you'll be covered.
- Download and install a cybersecurity program that actively scans and blocks threats from getting on your device.

Malwarebytes, for example, offers proactive cybersecurity programs for Windows, Mac, Android, and Chrome

Malware Analysis

- is the process of detecting and reducing potential threats in a website, application, or server.
- It is a crucial process that ensures computer security as well as the safety and security of an organization with regard to sensitive information.
- Malware analysis addresses vulnerabilities before they get out of hand.

Malware Analysis

- If you are looking at it more simply, malware analysis can be considered as the process of understanding the behavior and the intended use of a suspicious file or URL.
- The more you know about the suspicious file, the better it will help to mitigate the threat, if any.

Key Benefits of Malware Analysis

- Identifying the source of the attack
- Determining the damage from a security threat
- Identifying a malware's exploitation level, vulnerability, and appropriate patching preparations
- Triaging the incidents according to the level of severity of the threat in a practical manner
- Uncovering hidden Indicators of Compromise (IOC) that need to be blocked
- Improving the efficacy of IOC, alerts, and notifications
- Enriching context when trying to uncover threats

Malware Reverse Engineering

- Malware researchers require a diverse skill set usually gained over time through experience and self-training.
- Reverse engineering (RE) is an integral part of malware analysis and research but it is also one of the most advanced skills a researcher can have.
- This is one of the reasons why organizations lack reverse engineering manpower.
- Many researchers with a lack of experience struggle to get started in RE.

Malware Reverse Engineering

- Malware RE focuses specifically on understanding malware capabilities and functionalities in order to remediate threats and study different malware families
- RE can be very time-consuming.
- When researching a malware, you will usually not start reversing it right away.

Malware Reverse Engineering

- Instead, you should conduct triage malware analysis by running the malware in a sandbox, extracting strings, and more
- This initial malware analysis phase can provide further context for reverse engineering, if needed.
- For instance, you can search for specific strings in the disassembler or expect to see a certain capability that the malware displays.

Malware Reverse Engineering

- If your goal is to understand a malware's capabilities, analyzing it dynamically via a sandbox will not be enough
- The malware's Command and Control (C2) could go down, the malware could depend on another file for configuration which does not exist on the machine, the malware has sandbox evasion capabilities, or the malware will only run on a certain environment.
- RE, which is part of advanced static malware analysis, is much more effective to achieve this goal.

Security Engineering: Passwords and their limitations

Password

- a secret word or phrase or code that you need to know in order to have access to a place or system
- it is a series of letters or numbers that you must type into a computer or computer system in order to be able to use it
- A password is a real-life implementation of challenge-response authentication (a set of protocols to protect digital assets and data).

Password

- Definition:
- A string of characters i.e letters, numbers, special characters, used to verify the identity of a user during the authentication process is known as password

Password Management

- Since passwords are meant
 - to keep the files and data secret and safe
 - so it is prevented the unauthorized access,
- Password management refers to
 - the practices and
 - set of rules or principles or standards
- that out must follow or at least try to seek help from in order to be a good/strong password and along with its storage and management for the future requirements.

Issues Related to Managing Passwords

- It is not safe to use the same password for multiple sites, therefore having different passwords for different sites and on top of that remembering them is quite difficult
- As per the statistics, more than 65% of people reuse passwords across accounts and the majority do not change them, even after a known breach
- Meanwhile, 25% reset their passwords once a month or more because they forgot them

Password Manager

- To escape from this situation people often tend to use password managers
- A password manager is a computer program that allows users to store, generate, and manage their passwords for local applications and online services
- Password managers to a certain extent reduce the problem by having to remember only one “master password” instead of having to remember multiple passwords

Password Manager

- The only problem with having a master password is that once it is out or known to an attacker, the rest of all the passwords become available
- The main issues related to managing passwords are as follows:
 - Login spoofing
 - Sniffing attack
 - Brute force attack
 - Shoulder surfing attack
 - Data breach

Methods to Manage Password

- There are a lot of good practices that we can follow to generate a strong password and also the ways to manage them
- **Strong and long passwords:**
- A minimum length of 8 to 12 characters long,
- also it should contain at least three different character sets (e.g., uppercase characters, lowercase characters, numbers, or symbols)

Methods to Manage Password

- **Password Encryption:**
- Using irreversible end-to-end encryption is recommended
- In this way, the password remains safe even if it ends up in the hands of cybercriminals.
- **Multi-factor Authentication (MFA):**
- Adding some security questions and
- a phone number that would be used to confirm that it is indeed you who is trying to log in will enhance the security of your password.

Methods to Manage Password

- **Make the password pass the test:**
- Yes, put your password through some testing tools that you might find online in order to ensure that it falls under the strong and safe password category
- **Avoid updating passwords frequently:**
- Though it is advised or even made mandatory to update or change your password as frequently as in 60 or 90 days.

Attacks on Passwords

- Password attacks are one of the most common forms of corporate and personal data breach
- A password attack is simply when a hacker try to steal your password
- 81% of data breaches were due to compromised credentials

Attacks on Passwords

- Because passwords can only contain so many letters and numbers, passwords are becoming less safe
- Hackers know that many passwords are poorly designed, so password attacks will remain a method of attack as long as passwords are being used

Phishing

- Phishing is when a hacker posing as a trustworthy party sends you a fraudulent email,
- hoping you will reveal your personal information voluntarily
- Sometimes they lead you to fake "reset your password" screens; other times, the links install malicious code on your device
- Here are a few examples of phishing

Phishing

- **Regular phishing**
- You get an email from what looks like goodwebsite.com asking you to reset your password,
- but you didn't read closely and it's actually goodwobsite.com
- You "reset your password" and the hacker steals your credentials

Phishing

- **Spear phishing**
- A hacker targets you specifically with an email that appears to be from a friend, colleague, or associate
- It has a brief, generic blurb ("Check out the invoice I attached and let me know if it makes sense.") and hopes you click on the malicious attachment

Phishing

- **Smishing and vishing**
- You receive a text message (SMS phishing, or smishing) or phone call (voice phishing, or vishing) from a hacker who informs you that your account has been frozen or that fraud has been detected
- You enter your account information and the hacker steals it

To avoid phishing attacks

- **Check who sent the email:**
- look at the From: line in every email to ensure that the person they claim to be matches the email address you're expecting
- **Double check with the source:**
- when in doubt, contact the person who the email is from and ensure that they were the sender.
- **Check in with your IT team:**
- your organization's IT department can often tell you if the email you received is legitimate.

Man-in-the-Middle Attack

- Man-in-the middle (MitM) attacks are when a hacker or compromised system sits in between two uncompromised people or systems and deciphers the information they're passing to each other, including passwords
- If Alice and Bob are passing notes in class, but Jeremy has to relay those notes, Jeremy has the opportunity to be the man in the middle.

Man-in-the-Middle Attack

- Similarly, in 2017, Equifax removed its apps from the App Store and Google Play store because they were passing sensitive data over insecure channels where hackers could have stolen customer information

prevent man-in-the-middle attacks

- **Enable encryption on your router**
- If your modem and router can be accessed by anyone off the street, they can use "sniffer" technology to see the information that is passed through it.
- **Use strong credentials and two-factor authentication**
- Many router credentials are never changed from the default username and password.
- If a hacker gets access to your router administration, they can redirect all your traffic to their hacked servers.

prevent man-in-the-middle attacks

- **Use a VPN**
- A secure virtual private network (VPN) will help prevent man-in-the-middle attacks by ensuring that all the servers you send data to are trusted.

Brute Force Attack

- If a password is equivalent to using a key to open a door, a brute force attack is using a battering ram
- A hacker can try 2.18 trillion password/username combinations in 22 seconds, and if your password is simple, your account could be in the crosshairs

prevent brute force attacks

- **Use a complex password**
- The difference between an all-lowercase, all-alphabetic, six-digit password and a mixed case, mixed-character, ten-digit password is enormous
- As your password's complexity increases, the chance of a successful brute force attack decreases

prevent brute force attacks

- **Enable and configure remote access**
- Ask your IT department if your company uses remote access management
- An access management tool like OneLogin will mitigate the risk of a brute-force attack.

prevent brute force attacks

- **Require multi-factor authentication**
- If multi-factor authentication (MFA) is enabled on your account, a potential hacker can only send a request to your second factor for access to your account
- Hackers likely won't have access to your mobile device or thumbprint, which means they'll be locked out of your account

prevent brute force attacks

- **Require multi-factor authentication**
- If multi-factor authentication (MFA) is enabled on your account, a potential hacker can only send a request to your second factor for access to your account
- Hackers likely won't have access to your mobile device or thumbprint, which means they'll be locked out of your account

Dictionary Attack

- A type of brute force attack, dictionary attacks
- rely on our habit of picking "basic" words as our password, the most common of which hackers have collated into "cracking dictionaries"
- More sophisticated dictionary attacks incorporate words that are personally important to you, like a birthplace, child's name, or pet's name

prevent a dictionary attack

- **Never use a dictionary word as a password**
- If you've read it in a book, it should never be part of your password.
- If you must use a password instead of an access management tool, consider using a password management system

prevent a dictionary attack

- **Lock accounts after too many password failures.**
- It can be frustrating to be locked out of your account when you briefly forget a password, but the alternative is often account insecurity.
- Give yourself five or fewer tries before your application tells you to cool down.

prevent a dictionary attack

- **Consider investing in a password manager**
- Password managers automatically generate complex passwords that help prevent dictionary attacks.

Credential Stuffing

- If you've suffered a hack in the past, you know that your old passwords were likely leaked onto a disreputable website
- Credential stuffing takes advantage of accounts that never had their passwords changed after an account break-in
- Hackers will try various combinations of former usernames and passwords, hoping the victim never changed them

Web Application Security

Web Application Security

- Web AppSec is the idea of building websites to function as expected, even when they are under attack.
- The concept involves a collection of security controls engineered into a Web application to protect its assets from potentially malicious agents
- Web applications, like all software, inevitably contain defects

Web Application Security

- Some of these defects constitute actual vulnerabilities that can be exploited, introducing risks to organizations
- Web application security defends against such defects
- It involves
 - leveraging secure development practices and
 - implementing security measures throughout SDLC, ensuring that design-level flaws and implementation-level bugs are addressed

Importance of Web Security Testing

- To find security vulnerabilities in Web applications and their configuration
-
- The primary target is the application layer (i.e., what is running on the HTTP protocol)
- Involves sending different types of input to provoke errors and make the system behave in unexpected ways
- These so called “negative tests” examine whether the system is doing something it isn’t designed to do.

Importance of Web Security Testing

- To understand that Web security testing is not only about testing the security features (e.g., authentication and authorization) that may be implemented in the application
- It is equally important to test that other features are implemented in a secure way (e.g., business logic and the use of proper input validation and output encoding)
- The goal is to ensure that the functions exposed in the Web applications are secure.

Types of Web Security Testing

1. Dynamic Application Security Test (DAST)
2. Static Application Security Test (SAST)
3. Penetration Test
4. Runtime Application Self Protection (RASP)

Dynamic Application Security Test (DAST)

- This automated application security test is best for internally facing, low-risk applications that must comply with regulatory security assessments
- For medium-risk applications and critical applications undergoing minor changes,
- combining DAST with some manual web security testing for common vulnerabilities is the best solution

Static Application Security Test (SAST)

- This application security approach offers automated and manual testing techniques
- It is best for identifying bugs without the need to execute applications in a production environment
-
- It also enables developers
 - to scan source code and
 - systematically find and eliminate software security vulnerabilities.

Penetration Test

- This manual application security test is best for critical applications, especially those undergoing major changes
- The assessment involves business logic and adversary-based testing to discover advanced attack scenarios

Runtime Application Self Protection (RASP)

- This evolving application security approach
 - encompasses a number of technological techniques to instrument an application
 - so that attacks can be monitored as they execute and, ideally, blocked in real time

How does application security testing reduce your organization's risk?

- **Majority of Web Application Attacks**
- SQL Injection
- XSS (Cross Site Scripting)
- Remote Command Execution
- Path Traversal

How does application security testing reduce your organization's risk?

- **Attack Results**
- Access to restricted content
- Compromised user accounts
- Installation of malicious code
- Lost sales revenue
- Loss of trust with customers
- Damaged brand reputation
- And much more

How does application security testing reduce your organization's risk?

- The several of the top attacks used by attackers, which can result in serious damage to an individual application or the overall organization
- Knowing the different attacks that make an application vulnerable,
- in addition to the potential outcomes of an attack, allow your firm to preemptively address the vulnerabilities and accurately test for them.

How does application security testing reduce your organization's risk?

- By identifying the root cause of the vulnerabilities, mitigating controls can be implemented during the early stages of the SDLC to prevent any issues
- Additionally, knowledge of how these attacks work can be leveraged to target known points of interest during a Web application security test.
- Recognizing the impact of an attack is also key to managing your firm's risk, as the effects of a successful attack can be used to gauge the vulnerability's total severity

How does application security testing reduce your organization's risk?

- If issues are identified during a security test, defining their severity allows your firm to efficiently prioritize the remediation efforts.
- Start with critical severity issues and work towards lower impact issues to minimize risk to your firm

What features should be reviewed during a web application security test?

- The following non-exhaustive list of features should be reviewed during Web application security testing
- An inappropriate implementation of each could result in vulnerabilities, creating serious risk for your organization

What features should be reviewed during a web application security test?

- **Application and server configuration**
- Potential defects are related to
 - encryption/cryptographic configurations,
 - Web server configurations, etc.
- **Input validation and error handling**
- SQL injection, cross-site scripting (XSS), and
- other common injection vulnerabilities are the result of poor input and output handling.

What features should be reviewed during a web application security test?

- **Authentication and session management**
 - Vulnerabilities potentially resulting in user impersonation
 - Credential strength and protection should also be considered
- **Authorization**
- Testing the ability of the application to protect against vertical and horizontal privilege escalations.

What features should be reviewed during a web application security test?

- **Business logic**
 - These are important to most applications that provide business functionality
- **Client-side logic**
 - With modern, JavaScript-heavy web pages, in addition to web pages using other types of client-side technologies (e.g., Silverlight, Flash, Javaapplets), this type of feature is becoming more prevalent

Web Application Security Threats

- Each year, attackers develop inventive web application security threats
 - to compromise sensitive data and
 - access their targets' database
- Consequently, security experts build on the exploited vulnerabilities and strengthen their systems through their learning's every year

Injection Attacks

- A web app that is vulnerable to injection attacks accepts untrusted data from an input field without any proper sanitation
- By typing code into an input field, the attacker can trick the server into interpreting it as a system command and thereby act as the attacker intended
- Some common injection attacks include SQL injections, Cross-Site Scripting, Email Header Injection, etc.
- These attacks could lead to unauthorized access to databases and exploitation of admin privileges.

Injection Attacks

- **How to prevent:**
- Keep untrusted inputs away from commands and queries
- Use a safe Application Programming Interface (API) that avoids interpreters or uses parameterized interfaces
- Filter and sanitize all inputs as per a white list.
- This prevents the use of malicious character combinations.

Broken Authentication

- Broken authentication is an umbrella term given to vulnerabilities wherein authentication and session management tokens are inadequately implemented
- This improper implementation allows hackers
 - to make claims over a legitimate user's identity,
 - access their sensitive data, and
 - potentially exploit the designated ID privileges.

Broken Authentication

- **How to prevent:**
- End sessions after a certain period of inactivity.
- Invalidate a session ID as soon as the session ends.
- Place limiters on the simplicity of passwords.
- Implement multi-factor authentication (2FA/MFA).

Cross Site Scripting (XSS)

- It is an injection-based client-side attack
- This attack involves injecting malicious code in a website application to execute them in the victims' browsers
- Any application that doesn't validate untrusted data adequately is vulnerable to such attacks
- Successful implementation results in
 - theft of user session IDs,
 - website defacing, and
 - redirection to malicious sites (thereby allowing phishing attacks).

Cross Site Scripting (XSS)

- **How to prevent:**
- Encode all user-supplied data.
- Use auto-sanitization libraries such as OWASP's AntiSamy.
- White list inputs to disallow certain special character combinations.

Insecure Direct Object References (IDOR)

- Mostly through manipulation of the URL, an attacker gains access to database items belonging to other users
- For instance, the reference to a database object is exposed in the URL.
- The vulnerability exists when someone can edit the URL to access other similar critical information (such as monthly salary slips) without additional authorization.

Insecure Direct Object References (IDOR)

- **How to prevent:**
- Implement proper user authorization checks at relevant stages of users' web app journey.
- Customize error messages so that they don't reveal critical information about the respective user.
- Try not to disclose reference to objects in the URL; use POST based information transmission over GET.

Security Misconfigurations

- According to OWASP top 10, this is the most common web application security threats found across web applications
-
- This vulnerability exists because developers and administrators “forget” to change some default settings such as default passwords, usernames, reference IDs, error messages, etc.
- Given how easy it is to detect and exploit default settings that were initially placed to accommodate a simple user experience, the implications of such a vulnerability can be vast once the website is live: from admin privileges to complete database access.

Security Misconfigurations

- **How to prevent:**
- Frequently maintain and update all web application components: firewalls, operating systems, servers, databases, extensions, etc.
- Make sure to change default configurations.
- Make time for regular penetration tests (though this applies to every vulnerability that a web app could have).

Unvalidated Redirects and Forwards

- Pretty much every website redirects a user to other web pages.
- When the credibility of this redirection is not assessed, the website leaves itself vulnerable to such URL based attacks.
- A malicious actor can redirect users to phishing sites or sites containing malware.
- Phishers search for this vulnerability extensively since it makes it easier for them to gain user trust.

Unvalidated Redirects and Forwards

- **How to prevent:**
- Avoid redirection where possible.
- Give the destination parameters a mapping value rather than the actual URL
- Let the server-side code translate the mapping value to the actual URL.

Missing Function Level Access Control

- mostly similar to IDOR
- The core differentiating factor between the two is that IDOR tends to give the attacker access to information in the database
- In contrast, Missing_ Function Level Access Control _allows the attacker access to special functions and features that should not be available to any typical user
- Like, IDOR, access to these functions can be gained through URL manipulation as well

Missing Function Level Access Control

- **How to prevent:**
- Implement adequate authorization measures at relevant stages of user web app use.
- Deny all access to set features and functions unless attempted by a pre-approved (admin) user.
- Allow for a flexible shift in grant and rejection of access to feature privileges in your code. Hence, allowing a practical and secure shift in privilege access when needed.

Client Side Security

Client Side Security

- Today's web applications are complex,
 - often made up of a mix of existing software, open-source and third-party code, and
 - custom JavaScript and HTML all integrated via application program interfaces (APIs).
- While web applications are hosted and maintained on an organization's server, they actually run on an end user's browser.

Client Side Security

- The scripts that run the applications are referred to as 'client-side scripts.'
- These scripts create an incredibly dynamic environment that enable a high level of functionality,
- but also facilitate tremendous risk since the combination of potentially flawed or vulnerable systems, servers, codes, and applications creates the perfect scenario for threat actors to leverage in client-side attacks

client-side attacks

- The Open Web Application Security Project® (OWASP) lists 12 client-side security risks that organizations need to ensure they've mitigated to prevent attacks:

Document Object Model (DOM)-based Cross-site Scripting

- Sometimes also called just 'cross-site scripting' or 'XSS',
- this is a vulnerability that affects websites and enables an attacker to inject their own malicious code onto the HTML pages displayed to users.
- If the malicious code is executed by the victim's browser, the code performs actions, such as stealing credit card information or sensitive credentials

JavaScript Injection

- This type of vulnerability is considered a subtype of XSS
- involving the injection of malicious JavaScript code executed by the end user's browser application
- JavaScript injections can be used
 - to modify the content seen by the end user,
 - to steal the user's session cookies, or
 - to impersonate the user

HTML Injection

- Another type of cross-site scripting attack, an HTML injection involves injecting HTML code via vulnerable sections of the website
- Usually, the purpose of the HTML injection is to change the website's design or information displayed on the website

Client-side URL Redirection

- In this type of attack, an application accepts untrusted input that contains a URL value
- that causes the web application to redirect the user to another, likely malicious page controlled by the attacker.

Cascading Style Sheets (CSS) Injection

- Attackers inject arbitrary CSS code into a website, which is then rendered in the end user's browser.
- Depending on the type of CSS payload,
 - the attack could lead to XSS,
 - user interface (UI) modifications or
 - the exfiltration of sensitive information, like credit card data

Client-side Resource Manipulation

- This type of vulnerability enables the threat actor to control the URL that links to other resources on the web page, thus enabling cross-site scripting attacks.

Cross-origin Resource Sharing (CORS)

- Poorly configured CORS policies can facilitate cross-origin attacks like cross-site request forgery (CSRF)

Cross-site Flashing

- Because Flash applications are often embedded in browsers, flaws or vulnerabilities in the Flash application could enable cross-site scripting attacks

Clickjacking or UI Redress Attack

- This type of attack involves a threat actor using multiple web page frame layers to trick a user into clicking a button or link on a different page from the one intended
- Keystrokes can also be hijacked using this technique.
- By using style sheets, i frames, and text boxes, a threat actor can trick the user into thinking they're entering login credentials or bank account information into a legitimate website, when, in fact, they are actually typing into a frame controlled by the attacker

Web Messaging

- Also called cross-document messaging,
- web messaging enables applications running on different domains to communicate securely
- If the receiving domain is not configured, problems could arise related to redirection or the website leaking sensitive information to unknown or malicious servers

Local Storage

- Sometimes called web storage or offline storage, local storage enables JavaScript sites and apps to store and access the data without any expiration date
-
- Thus, data stored in the browser will be available even after closing the browser window.
- Since the storage can be read using JavaScript, a cross-site scripting attack could extract all the data from the storage.
- Malicious data could also be loaded via JavaScript

Countermeasures:

- Install antivirus software, anti-spyware software, and firewall protection on all workstations, servers, and wireless devices
- Ensure the latest system software patches are applied regularly.
- Maintain a complete backup system of all data on all systems use a separate server or external hard drive or network location to store backups that are no longer needed and keep them off the system they were created on.

Countermeasures:

- If a user is logging in from an unknown location or IP address, consider blocking access from those locations (access control lists).
- Prevent unauthorized access to accounts.
- Use strong passwords and avoid common passwords or patterns that can lead to vulnerabilities
- Limit login attempts (user logout)

Server-side Security

How to secure a server?

- Server security is a major issue for companies
- Indeed, being a central element in the functioning of all the components of an information system (applications, network, infrastructure, employees, etc.), servers are often the prime targets of attacks
- Furthermore, server-side vulnerabilities can have severe consequences

How to secure a server?

- In the event of misconfiguration or lack of control,
 - these flaws can be exploited and lead to the compromise of the data in transit, or
 - even to the server being taken over by malicious persons.

Why is server security important?

- Attacks on servers are a daily occurrence because, very often, too many loopholes exist
- Indeed,
 - applications vulnerable to SQL injections hosted on a server,
 - users unaware of social engineering risks, or
 - simply poor practices in terms of updates and patch management of OS and server services,

Why is server security important?

- easily allow attackers to achieve their goals:
 - data theft,
 - access to sensitive information,
 - paralysis of a company's activity, etc
- Securing a server is therefore vital and necessarily involves implementing best practices in terms of
 - configuration,
 - control,
 - monitoring and
 - security testing

Implement an update and patch management policy for servers

- Implementing an update management policy for operating systems and services is essential to maintain a good level of security
- Indeed, new vulnerabilities are discovered and published regularly
- And if security patches are not applied in time, the risk of attacks and server compromise increases.

Implement an update and patch management policy for servers

- The numerous attacks suffered by companies, via malware
- It is therefore important to define and implement an update and patch management policy for servers and all software and hardware components of the IS
- This involves documenting procedures and continuous monitoring of the various patch releases or new versions

Disable or remove unnecessary services

- All software and components installed on an OS increase the attack surface and therefore the risk of compromise
- However, strengthening server security requires reducing the attack surface
- To do this, it is necessary to disable or even remove (as far as possible) all services, applications, network protocols, third-party components, etc. that are not essential to the operation of your server

Disable or remove unnecessary services

- Removing or disabling unnecessary systems enhances the security of a server in several ways
- Firstly, they cannot be compromised, nor can they be used as attack vectors to alter the services that are essential for the server to operate
- Indeed, it should be kept in mind that each component added to a server increases the risk of compromising it

Disable or remove unnecessary services

- Furthermore, the server can be configured to better meet the requirements of a particular service, while improving performance and the overall level of security
- Finally, reducing services means limiting the number of log entries, which makes it easier to monitor and detect unusual events

Controlling and securing server access: Focus on SSH

- Presentation of SSH: an essential tool to ensure proper management and secure administration of the server
- SSH (Secure Shell) is both a communication protocol and a computer program allowing local and remote administration of a server
- Its most common implementation is OpenSSH, which can be found on many systems, including servers (Unix, Linux, Windows) as well as workstations and network equipment

Controlling and securing server access: Focus on SSH

- Indeed, OpenSSH is a suite of tools offering many features, including:
 - a server (sshd),
 - several clients – remote shell connection (ssh) / file transfer and download (scp and sftp),
 - a key generation tool (ssh-keygen),
 - a keychain service (ssh-agent and ssh-add), etc

Controlling and securing server access: Focus on SSH

- SSH currently exists in two versions: SSHv1 and SSHv2
- SSHv1 contains vulnerabilities that have been fixed in the second version
- Therefore, for enhanced security, only version 2 of the SSH protocol should be authorized
- The most widely used feature of SSH is remote administration, which consists of connecting to a remote machine and launching a shell session following authentication

Public key / certificate authentication

- Not ensuring the authenticity of a server can have several security impacts, including
 - the inability to verify that you are communicating with the correct server,
 - with consequent risks of spoofing and exposure to Man in The Middle attacks.
- SSH relies on asymmetric encryption for authentication, and thus ensures the legitimacy of the server being contacted before access is granted

Public key / certificate authentication

- Moreover, this control is done in several ways with OpenSSH
- Either by
 - ensuring that the public key fingerprint obtained previously with `ssh-keygen` and presented by the server is the correct one, or
 - by verifying the signature of the certificate presented by the server with a certification authority known to the client

Security of authentication and user access control

- Users of a system always have rights, no matter how small
- It is therefore important to protect their access via a secure authentication mechanism and to thwart brute force attacks as much as possible
- Firstly, each user must have his or her own account to ensure better traceability and the allocation of access rights must always follow the principle of least privilege

Security of authentication and user access control

- Furthermore, access to a service should be restricted to users who have a justified need and are explicitly authorized
- Regarding users authorized to configure the operating system of servers, they should be limited to a small number of designated administrators

Security of authentication and user access control

- Furthermore, it is strongly discouraged to use authentication mechanisms in which authentication information is transmitted in the clear over a network, as this information can be intercepted via Man in the Middle attacks and used by an attacker to impersonate an authorized user

Security of authentication and user access control

- Finally, to ensure secure user authentication, the following measures should be implemented:
- **Removing default accounts**
 - because the default configuration of the OS often includes guest accounts, administrator accounts and accounts associated with services.
 - The names and passwords of these accounts are known to attackers.

Security of authentication and user access control

- **Implement a proper password policy**
 - All passwords should be complex and difficult to guess.
 - Above all, they should be long (more than 15 characters).
 - To achieve this, nothing is better than the implementation of a password manager

Logging and monitoring of server events

- Logging is a central aspect of a security strategy
- It is a control mechanism for
 - monitoring the network, systems and servers; and
 - it ensures the traceability of all normal and suspicious events.
- Indeed, log files can be used to track the activities of an attacker and thus detect unusual events or failed or successful intrusion attempts.

Logging and monitoring of server events

- To facilitate the management and exploitation of logs, they should be centralized on a dedicated server
- This is even more important because, in the event of a machine being compromised, it is likely that the logs will be destroyed or altered by the attacker
- Centralizing, regularly backing up and duplicating logs will ensure that a copy is always kept. And given their importance, it is essential to restrict access to logs to authorized users only

Securing APIs, websites and web applications hosted on a server

- APIs, websites and web applications hosted on a server must also be secured
- These can be used as attack vectors to compromise the server if they are vulnerable to SQL injections for example

Application Security: HTTPS and HSTS

Hypertext transfer protocol secure (HTTPS)

What is HTTPS?

Hypertext transfer protocol secure (HTTPS) is the secure version of HTTP, which is the primary protocol used to send data between a web browser and a website.

HTTPS is encrypted in order to increase security of data transfer.

This is particularly important when users transmit sensitive data, such as by logging into a bank account, email service, or health insurance provider.

Any website, especially those that require login credentials, should use HTTPS. In modern web browsers such as Chrome, websites that do not use HTTPS are marked differently than those that are.

Look for a **padlock** in the URL bar to signify the webpage is secure.

Web browsers take HTTPS seriously; Google Chrome and other browsers flag all non-HTTPS websites as not secure.

How does HTTPS work?

HTTPS uses an encryption protocol to encrypt communications.

The protocol is called **Transport Layer Security** (TLS), although formerly it was known as **Secure Sockets Layer** (SSL).

This protocol secures communications by using what's known as an **asymmetric public key infrastructure**.

This type of security system uses two different keys to encrypt communications between two parties:

- The private key - this key is controlled by the owner of a website and it's kept, as the reader may have speculated, private. This key lives on a web server and is used to **decrypt information encrypted by the public key**.
- The public key - this key is available to everyone who wants to interact with the server in a way that's secure. Information that's encrypted by the public key can only be decrypted by the private key.

Why is HTTPS important? What happens if a website doesn't have HTTPS?

HTTPS prevents websites from having their information broadcast in a way that's easily viewed by anyone **snooping** on the network.

When information is sent over regular HTTP, the information is broken into packets of data that can be easily "sniffed" using free software.

This makes communication over the an unsecure medium, such as public Wi-Fi, highly vulnerable to interception.

In fact, all communications that occur over **HTTP occur in plain text**, making them highly accessible to anyone with the correct tools, and vulnerable to on-path attacks.

With HTTPS, traffic is encrypted such that even if the packets are sniffed or otherwise intercepted, they will come across as **nonsensical characters**.

In websites without HTTPS, it is possible for Internet service providers (ISPs) or other intermediaries to **inject content into web pages without the approval of the website owner**.

This commonly takes the form of advertising, where an ISP looking to increase revenue injects paid advertising into the webpages of their customers.

Unsurprisingly, when this occurs, the profits for the advertisements and the quality control of those advertisements are in no way shared with the website owner.

HTTPS eliminates the ability of unmoderated third parties to inject advertising into web content.

What port does HTTPS use?

HTTPS uses port 443. This differentiates HTTPS from HTTP, which uses port 80.

(In networking, a port is a virtual software-based point where network connections start and end. All network-connected computers expose a number of ports to enable them to receive traffic. Each port is associated with a specific process or service, and different protocols use different ports.)

How else is HTTPS different from HTTP?

HTTPS is not a separate protocol from HTTP.

It is simply using TLS/SSL encryption over the HTTP protocol. HTTPS occurs based upon the transmission of TLS/SSL certificates, which verify that a particular provider is who they say they are.

When a user connects to a webpage, the webpage will send over its SSL certificate which contains the public key necessary to start the secure session.

The two computers, the client and the server, then go through a process called an **SSL/TLS handshake, which is a series of back-and-forth communications used to establish a secure connection.**

To take a deeper dive into encryption and the SSL/TLS handshake, read about what happens in a TLS handshake.

Advantages of HTTPS

Following are the advantages or benefits of a Hypertext Transfer Protocol Secure (HTTPS):

- The main advantage of HTTPS is that it provides high security to users.
- Data and information are protected. So, it ensures data protection.
- SSL technology in HTTPS protects the data from third-party or hackers. And this technology builds trust for the users who are using it.
- It helps users by performing banking transactions.

Disadvantages of HTTPS

Following are the disadvantages or limitations of a Hypertext Transfer Protocol Secure (HTTPS):

- The big disadvantage of HTTPS is that users **need to purchase the SSL certificate.**
- The speed of accessing the website is **slow** because there are various complexities in communication.
- Users need to update all their internal links.

Difference between HTTP and HTTPS

HTTPS is an abbreviation of Hypertext Transfer Protocol Secure. It is a secure extension or version of HTTP.

This protocol is mainly used for providing security to the data sent between a website and the web browser.

It is widely used on the internet and used for secure communications. This protocol uses the 443 port number for communicating the data.

This protocol is also called HTTP over SSL because the HTTPS communication protocols are encrypted using the SSL (Secure Socket Layer).

By default, it is supported by various web browsers.

Those websites which need login credentials should use the HTTPS protocol for sending the data.

It allows users to create a secured encrypted connection and helps them to protect their information from being stolen.

HTTP	HTTPS
1. It is an abbreviation of Hypertext Transfer Protocol	1. It is an abbreviation of Hypertext Transfer Protocol Secure.
2. This protocol operates at the application layer.	2. This protocol operates at the transport layer.
3. The data which is transferred in HTTP is plain text.	3. The data which is transferred in HTTPS is encrypted, i.e., ciphertext.
4. By default, this protocol operates on port number 80.	4. By default, this protocol operates on port number 443.
5. The URL (Uniform Resource Locator) of HTTP start with http://	5. The URL (Uniform Resource Locator) of HTTPS start with https://
6. This protocol does not need any certificate.	6. But, this protocol requires an SSL (Secure Socket Layer) certificate.
7. Encryption technique is absent in HTTP.	7. Encryption technique is available or present in HTTPS.
8. The speed of HTTP is fast as compared to HTTPS.	8. The speed of HTTPS is slow as compared to HTTP.
9. It is un-secure.	9. It is highly secure.
10. Examples of HTTP websites are Educational Sites, Internet Forums, etc.	10. Examples of HTTPS websites are shopping websites, banking websites, etc.

HSTS (HTTP Strict Transport Security)

What is HSTS (HTTP Strict Transport Security)?

HTTP Strict Transport Security (HSTS) is a web security policy mechanism that enables web sites to declare themselves accessible only via secure connections.

This helps protect websites and users from protocol downgrade and cookie hijacking attacks.

Why Was HSTS Introduced?

HTTP is used over various transports, typically the Transmission Control Protocol (TCP).

However, TCP does not provide integrity protection, confidentiality or secure host identification.

This led to the development of Secure Sockets Layer (SSL) and its successor Transport Layer Security (TLS).

SSL/TLS provide an encryption layer between application protocols and TCP, commonly known as HTTPS.

In general, user agents (like web browsers) will employ various local security policies to decide how to interact with a host, based on a negotiation between the server, user preferences and their communication method (HTTP or HTTPS).

However, some user agents allow users to choose to continue to interact with a website when they are unable to establish a secure connection.

This could happen when a TLS certificate's trust chain is not validated, when it has expired or when the TLS host's domain name appears incorrectly in the TLS certificate.

This behavior is called **click-through insecurity**.

While giving users the option to continue to use a website despite a lack of HTTPS can keep users happy, it can introduce attack vectors that leave users open to certain types of cyber attacks, particularly man-in-the-middle attacks (MITM attacks), downgrade attacks and session hijacking attacks.

SSL-Stripping

As HSTS allows websites to declare they are only accessible through a secure connection, they can prevent users from connecting to them over any HTTP connection.

This prevents a security vulnerability known as SSL-stripping.

SSL-stripping is a downgrade attack that was introduced by Moxie Marlinspike in his 2009 BlackHat Federal talk New Tricks for Defeating SSL in Practice.

A **downgrade attack** is a form of cryptographic attack on a computer system or in this case, a communications protocol that makes it abandon its encrypted connection (HTTPS) in favor of an older, unencrypted connection (HTTP) that is typically provided for backwards compatibility with older systems.

SSL-stripping is implemented as part of a man-in-the-middle attack where web traffic is intercepted and redirected from the secure HTTPS version of the website to an unencrypted HTTP version.

The primary reason this attack continues to be successful is that many websites continue to not use TLS/SSL certificates.

This makes it impossible to know (without prior knowledge) whether a website's lack of HTTPS is due to an SSL-stripping attack or because they don't have a TLS certificate.

Additionally, there are no warnings to warn the user during the downgrade process, making the attack hard to detect even for the most vigilant user.

With the creation of a tool by Marlinspike to fully automate this type of attack, it represents a real cyber security risk.

Session Hijacking

Session hijacking or cookie hijacking is another vulnerability that is enabled through click-through insecurity.

Session hijacking exploits a valid computer session to gain unauthorized access to information or services.

This is particularly relevant for web developers as cookies are used to maintain a session on many websites.

If a website does not flag their cookies as Secure, telling user agents to only send cookies over HTTPS, they can be easily stolen by an attacker.

As non-Secure cookies are returned to the host regardless of transport security, leaving them open to man-in-the-middle attacks.

Once an attacker has access to the cookies, they can then impersonate the user on a legitimate website.

How Does HSTS Work?

HSTS enables web servers to declare that any interactions by web browsers and other user agents must be conducted over HTTPS connections and not insecure HTTP connections.

A server can implement an HSTS Policy by supplying a response header over an HTTPS connection (HSTS headers sent over HTTP response headers are ignored).

The HSTS header is name "Strict-Transport-Security" and also specifies a period of time during which the user agent should only access the service via HTTPS requests.

This means the first time a site is accessed using HTTPS it returns the Strict-Transport-Security header, the browser records this information, so future attempts to load the site using HTTP automatically use HTTPS.

When the expiration time specified by the Strict-Transport-Security header elapses, the next attempt to load the site via HTTP will proceed as normal instead of automatically using HTTPS.

However, whenever the Strict-Transport-Security header is delivered to the user agent, it will update the expiration time for that site, so sites can refresh this information and prevent the timeout from expiring.

Should it be necessary to disable HSTS, web servers can set the max-age to 0 (over a HTTPS connection) to immediately expire the HSTS header, allowing access via HTTP requests.

For example, a server could send a header that requests that future requests for the next year only use HTTPS via Strict-Transport-Security: max-age=31536000

When a web application issues a HSTS Policy to user agents, conforming user agents behave as follows:

- Any insecure links are automatically turn into secure links (e.g. <http://example.com/> will be modified to <https://example.com> before accessing the server)
- If a secure connection cannot be ensured (e.g. the server does not have a valid certificate), the user agent will terminate the connection and not allow the user to access the website.

The most important thing to understand is that a HSTS Policy prevents click-through insecurity by not allowing the end user to use the insecure connection.

What is an Example Situation Involving HSTS?

Imagine your staff member uses a free WiFi access point at a cafe and starts surfing the web, visiting your organization's payroll system.

Unfortunately, the access point they are using is actually an attacker's laptop and they're intercepting the original HTTP request and redirecting your employee to a clone of your payroll system instead of the real thing, exposing your employees' personally identifiable information (PII).

If your payroll system uses HSTS and your employee has visited it once using HTTPS, then their browser will know to only use HTTPS, preventing this type of man-in-the-middle attack.

What are the Limitations of HSTS?

A key limitation of using HSTS is that a user that cannot connect through HTTPS will be unable to use the site.

Additionally, as the HSTS Policy is communicated in a response header, it requires the user agent to first visit the website to learn that it uses HSTS.

This means the initial request remains unprotected from active attacks if it uses an insecure protocol such as plain HTTP or if the URI for the initial request was obtained over an insecure channel.

This will also apply to the first request after the activity period specified in the HSTS max-age (sites generally set a period of several days or months depending on user activity and behavior).

There is widespread browser support for HSTS including Google Chrome, Mozilla Firefox, Internet Explorer, Microsoft Edge, Opera, and Safari address this limitation by preloading HSTS Policies from the HSTS Preload list.

The HSTS list contains known websites that support HSTS and are distributed with the browser so it uses HTTPS for the initial request for the listed websites.

As this approach can never scale across the entire Web, there have been discussions to be able to declare HSTS in DNS records and be access them securely via DNSSEC, which could ensure validity.

Additionally, HSTS is ineffective against typosquatting domains, DNS-based attacks and man-in-the-middle attacks that serve traffic from an artificial domain that is not on the HSTS Preload list.

And as HSTS relies on TLS itself, it also relies on the security of TLS.

Read RFC 6797 for a deeper discussion of the overall HSTS security considerations.

Web Application Security

What is web application security?

Web application security (also known as Web AppSec) is the idea of building websites to function as expected, even when they are under attack.

The concept involves a collection of security controls engineered into a Web application to protect its assets from potentially malicious agents.

Web applications, like all software, inevitably contain defects. Some of these defects constitute actual vulnerabilities that can be exploited, introducing risks to organizations.

Web application security defends against such defects.

It involves leveraging secure development practices and implementing security measures throughout the software development life cycle (SDLC), ensuring that design-level flaws and implementation-level bugs are addressed.

Why is web security testing important?

Web security testing aims to find security vulnerabilities in Web applications and their configuration.

The primary target is the application layer (i.e., what is running on the HTTP protocol).

Testing the security of a Web application often involves sending different types of input to provoke errors and make the system behave in unexpected ways. These so called “negative tests” examine whether the system is doing something it isn’t designed to do.

It is also important to understand that Web security testing is not only about testing the security features (e.g., authentication and authorization) that may be implemented in the application.

It is equally important to test that other features are implemented in a secure way (e.g., business logic and the use of proper input validation and output encoding).

The goal is to ensure that the functions exposed in the Web application are secure.

What are the different types of security tests?

Dynamic Application Security Test (DAST)

This automated application security test is best for internally facing, low-risk applications that must comply with regulatory security assessments.

For medium-risk applications and critical applications undergoing minor changes, combining DAST with some manual web security testing for common vulnerabilities is the best solution.

Static Application Security Test (SAST)

This application security approach offers automated and manual testing techniques.

It is best for identifying bugs without the need to execute applications in a production environment.

It also enables developers to scan source code and systematically find and eliminate software security vulnerabilities.

Penetration Test

This manual application security test is best for critical applications, especially those undergoing major changes.

The assessment involves business logic and adversary-based testing to discover advanced attack scenarios.

Runtime Application Self Protection (RASP)

This evolving application security approach encompasses a number of technological techniques to instrument an application so that attacks can be monitored as they execute and, ideally, blocked in real time.

How does application security testing reduce your organization's risk?

Majority of Web Application Attacks

- SQL Injection
- XSS (Cross Site Scripting)
- Remote Command Execution
- Path Traversal

Attack Results

- Access to restricted content
- Compromised user accounts
- Installation of malicious code
- Lost sales revenue
- Loss of trust with customers
- Damaged brand reputation
- And much more

A Web application in today's environment can be affected by a wide range of issues.

The diagram above demonstrates several of the top attacks used by attackers, which can result in serious damage to an individual application or the overall organization.

Knowing the different attacks that make an application vulnerable, in addition to the potential outcomes of an attack, allow your firm to preemptively address the vulnerabilities and accurately test for them.

By identifying the root cause of the vulnerabilities, mitigating controls can be implemented during the early stages of the SDLC to prevent any issues.

Additionally, knowledge of how these attacks work can be leveraged to target known points of interest during a Web application security test.

Recognizing the impact of an attack is also key to managing your firm's risk, as the effects of a successful attack can be used to gauge the vulnerability's total severity.

If issues are identified during a security test, defining their severity allows your firm to efficiently prioritize the remediation efforts.

Start with critical severity issues and work towards lower impact issues to minimize risk to your firm.

Prior to an issue being identified, evaluating the potential impact against each application within your firm's application library can facilitate the prioritization of application security testing.

With an established list of high profile applications, web security testing can be scheduled to target your firm's critical applications first with more targeted testing to lower the risk against the business.

What features should be reviewed during a web application security test?

The following non-exhaustive list of features should be reviewed during Web application security testing.

An inappropriate implementation of each could result in vulnerabilities, creating serious risk for your organization.

Application and server configuration

Potential defects are related to encryption/cryptographic configurations, Web server configurations, etc.

Input validation and error handling

SQL injection, cross-site scripting (XSS), and other common injection vulnerabilities are the result of poor input and output handling.

Authentication and session management

Vulnerabilities potentially resulting in user impersonation. Credential strength and protection should also be considered.

Authorization

Testing the ability of the application to protect against vertical and horizontal privilege escalations.

Business logic

These are important to most applications that provide business functionality.

Client-side logic

With modern, JavaScript-heavy web pages, in addition to web pages using other types of client-side technologies (e.g., Silverlight, Flash, Javaapplets), this type of feature is becoming more prevalent.

Web Application Security Threats

Each year, attackers develop inventive web application security threats to compromise sensitive data and access their targets' database.

Consequently, security experts build on the exploited vulnerabilities and strengthen their systems through their learning's every year.

1. Injection Attacks

A web app that is vulnerable to injection attacks **accepts untrusted data from an input field without any proper sanitation.**

By typing code into an input field, the attacker can trick the server into interpreting it as a system command and thereby act as the attacker intended.

Some common injection attacks include SQL injections, Cross-Site Scripting, Email Header Injection, etc.

These attacks could lead to unauthorized access to databases and exploitation of admin privileges.

How to prevent:

- Keep untrusted inputs away from commands and queries.
- Use a safe Application Programming Interface (API) that avoids interpreters or uses parameterized interfaces.
- Filter and sanitize all inputs as per a white list. This prevents the use of malicious character combinations.

2. Broken Authentication

Broken authentication is an umbrella term given to vulnerabilities wherein authentication and session management tokens are inadequately implemented.

This improper implementation allows hackers to make claims over a legitimate user's identity, access their sensitive data, and potentially exploit the designated ID privileges.

How to prevent:

- End sessions after a certain period of inactivity.
- Invalidate a session ID as soon as the session ends.
- Place limiters on the simplicity of passwords.
- Implement multi-factor authentication (2FA/MFA).

3. Cross Site Scripting (XSS)

It is an **injection-based client-side attack**. At its core, this attack involves injecting malicious code in a website application to execute them in the victims' browsers eventually.

Any application that doesn't validate untrusted data adequately is vulnerable to such attacks.

Successful implementation results in theft of user session IDs, website defacing, and redirection to malicious sites (thereby allowing phishing attacks).

How to prevent:

- Encode all user-supplied data.
- Use auto-sanitization libraries such as OWASP's AntiSamy.
- White list inputs to disallow certain special character combinations.

4. Insecure Direct Object References (IDOR)

Mostly through manipulation of the URL, an attacker gains access to database items belonging to other users.

For instance, the reference to a database object is **exposed in the URL**.

The vulnerability exists when someone can edit the URL to access other similar critical information (such as monthly salary slips) without additional authorization.

How to prevent:

- Implement proper user authorization checks at relevant stages of users' web app journey.
- Customize error messages so that they don't reveal critical information about the respective user.
- Try not to disclose reference to objects in the URL; use POST based information transmission over GET.

5. Security Misconfigurations

According to OWASP top 10 2017, this is the most common web application security threats found across web applications.

This vulnerability exists because developers and administrators "forget" to change some default settings such as default passwords, usernames, reference IDs, error messages, etc.

Given how easy it is to detect and exploit default settings that were initially placed to accommodate a simple user experience, the implications of such a vulnerability can be vast once the website is live: from admin privileges to complete database access.

How to prevent:

- Frequently maintain and update all web application components: firewalls, operating systems, servers, databases, extensions, etc.
- Make sure to change default configurations.
- Make time for regular penetration tests (though this applies to every vulnerability that a web app could have).

6. Unvalidated Redirects and Forwards

Pretty much every website redirects a user to other web pages. When the credibility of this redirection is not assessed, the website leaves itself vulnerable to such URL based attacks.

A malicious actor can redirect users to phishing sites or sites containing malware.

Phishers search for this vulnerability extensively since it makes it easier for them to gain user trust.

How to prevent:

- Avoid redirection where possible.
- Give the destination parameters a mapping value rather than the actual URL. Let the server-side code translate the mapping value to the actual URL.

7. Missing Function Level Access Control

The seventh web application security threats in this list is mostly similar to IDOR.

The core differentiating factor between the two is that IDOR tends to give the attacker access to information in the database.

In contrast, Missing_ Function Level Access Control allows the attacker access to special functions and features that should not be available to any typical user.

Like, IDOR, access to these functions can be gained through URL manipulation as well.

How to prevent:

- Implement adequate authorization measures at relevant stages of user web app use.
- Deny all access to set features and functions unless attempted by a pre-approved (admin) user.
- Allow for a flexible shift in grant and rejection of access to feature privileges in your code. Hence, allowing a practical and secure shift in privilege access when needed.

Client Side Security

Today's web applications are complex, often made up of a mix of existing software, open-source and third-party code, and custom JavaScript and HTML all integrated via application program interfaces (APIs).

While web applications are hosted and maintained on an organization's server, they actually run on an end user's browser.

The scripts that run the applications are referred to as 'client-side scripts.' These scripts create an incredibly dynamic environment that enable a high level of functionality, but also facilitate tremendous risk since the combination of potentially flawed or vulnerable systems, servers, codes, and applications creates the perfect scenario for threat actors to leverage in client-side attacks

What are client-side attacks?

Client-side attacks occur when a user unintentionally downloads malicious or vulnerable content from a server, often by doing nothing more than simply clicking on a web page and filling out a form.

That content could take the form of bad JavaScript code or unsafe third-party code that exists as part of the web application.

The term 'client-side' refers to end-user devices, like desktops, laptops, mobile phones, and tablets, which are considered 'clients.'

Conversely, the systems that the devices are connected to are referred to as 'servers.'

Client devices send requests to the server and the server responds to the request.

Servers usually support multiple client devices at the same time, and client devices usually send requests to multiple different servers while operating on the internet.

Because client-side activity happens outside a business's security perimeter, standard security technologies won't protect the end user from malicious activity that is occurring on dynamic web pages accessed from the end user's own device.

What are the most common client-side security risks?

Unmitigated risks present in organizational systems can lead to potentially severe attacks on the client side—that is, an organization's customers or end users. These types of attacks include e-skimming, Magecart-like threats, and form jacking.

The Open Web Application Security Project® (OWASP) lists 12 client-side security risks that organizations need to ensure they've mitigated to prevent attacks:

Document Object Model (DOM)-based Cross-site Scripting—

Sometimes also called just 'cross-site scripting' or 'XSS', this is a vulnerability that affects websites and enables an attacker to inject their own malicious code onto the HTML pages displayed to users.

If the malicious code is executed by the victim's browser, the code performs actions, such as stealing credit card information or sensitive credentials.

JavaScript Injection—

This type of vulnerability is considered a subtype of XSS involving the injection of malicious JavaScript code executed by the end user's browser application.

JavaScript injections can be used to modify the content seen by the end user, to steal the user's session cookies, or to impersonate the user.

Hypertext Markup Language (HTML) Injection—

Another type of cross-site scripting attack, an HTML injection involves injecting HTML code via vulnerable sections of the website.

Usually, the purpose of the HTML injection is to change the website's design or information displayed on the website.

Client-side URL Redirection or Open Redirection—

In this type of attack, an application accepts untrusted input that contains a URL value that causes the web application to redirect the user to another, likely malicious page controlled by the attacker.

Cascading Style Sheets (CSS) Injection—

Attackers inject arbitrary CSS code into a website, which is then rendered in the end user's browser.

Depending on the type of CSS payload, the attack could lead to cross-site scripting, user interface (UI) modifications or the exfiltration of sensitive information, like credit card data.

Client-side Resource Manipulation—

This type of vulnerability enables the threat actor to control the URL that links to other resources on the web page, thus enabling cross-site scripting attacks.

Cross-origin Resource Sharing (CORS)—

Poorly configured CORS policies can facilitate cross-origin attacks like cross-site request forgery (CSRF).

Cross-site Flashing—

Because Flash applications are often embedded in browsers, flaws or vulnerabilities in the Flash application could enable cross-site scripting attacks.

Clickjacking or UI Redress Attack—

This type of attack involves a threat actor using multiple web page frame layers to trick a user into clicking a button or link on a different page from the one intended.

Keystrokes can also be hijacked using this technique. By using style sheets, i frames, and text boxes, a threat actor can trick the user into thinking they're entering login credentials or bank account information into a legitimate website, when, in fact, they are actually typing into a frame controlled by the attacker.

WebSockets—

If servers do not properly verify the origin of an initial HTTP web socket server, a variety of different attack types are possible, including sniffing, cross-site web socket hijacking (CSWH), and cross-site request forgery (CSRF).

Web Messaging—

Also called cross-document messaging, web messaging enables applications running on different domains to communicate securely.

If the receiving domain is not configured, problems could arise related to redirection or the website leaking sensitive information to unknown or malicious servers.

Local Storage—

Sometimes called web storage or offline storage, local storage enables JavaScript sites and apps to store and access the data without any expiration date.

Thus, data stored in the browser will be available even after closing the browser window.

Since the storage can be read using JavaScript, a cross-site scripting attack could extract all the data from the storage.

Malicious data could also be loaded via JavaScript.

Countermeasures:

Steps to take to prevent client-side attacks.

Install antivirus software, anti-spyware software, and firewall protection on all workstations, servers, and wireless devices.

Ensure the latest system software patches are applied regularly.

Maintain a complete backup system of all data on all systems use a separate server or external hard drive or network location to store backups that are no longer needed and keep them off the system they were created on.

If a user is logging in from an unknown location or IP address, consider blocking access from those locations (access control lists).

Prevent unauthorized access to accounts.

Use strong passwords and avoid common passwords or patterns that can lead to vulnerabilities, like "Admin" or "12345".

Limit login attempts (user lockout).

Server-side Security

How to secure a server?

Server security is a major issue for companies. Indeed, being a central element in the functioning of all the components of an information system (applications, network, infrastructure, employees, etc.), servers are often the prime targets of attacks.

Furthermore, server-side vulnerabilities can have severe consequences.

In the event of misconfiguration or lack of control, these flaws can be exploited and lead to the compromise of the data in transit, or even to the server being taken over by malicious persons.

Why is server security important?

Attacks on servers are a daily occurrence because, very often, too many loopholes exist.

Indeed, applications vulnerable to SQL injections hosted on a server, users unaware of social engineering risks, or simply poor practices in terms of updates and patch management of the operating system and server services, easily allow attackers to achieve their goals: data theft, access to sensitive information, paralysis of a company's activity, etc.

Securing a server is therefore vital and necessarily involves implementing best practices in terms of configuration, control, monitoring and security testing.

Update and patch management policy, hardening, access and administration control and security (via SSH), best logging and monitoring practices, etc., we present here the main measures to strengthen the security of your servers.

Implement an update and patch management policy for servers

Implementing an update management policy for operating systems and services is essential to maintain a good level of security.

Indeed, new vulnerabilities are discovered and published regularly.

And if security patches are not applied in time, the risk of attacks and server compromise increases.

The numerous attacks suffered by companies, via malware, following the publication of a patch for a protocol, software, operating system, etc., must push any type of organization to adopt a proactive posture, especially as information on vulnerabilities and exploits is published and therefore accessible to internal and external attackers.

It is also true that caution is recommended when it comes to installing software updates, as testing is always necessary, if not essential.

However, it is generally unwise because it is riskier to delay this process, as a passive attitude very often leads to the compromise of information systems.

It is therefore important to define and implement an update and patch management policy or servers and all software and hardware components of the IS.

This involves documenting procedures and continuous monitoring of the various patch releases or new versions.

Disable or remove unnecessary services

All software and components installed on an operating system increase the attack surface and therefore the risk of compromise.

However, strengthening server security requires reducing the attack surface.

To do this, it is necessary to disable or even remove (as far as possible) all services, applications, network protocols, third-party components, etc. that are not essential to the operation of your server.

Removing or disabling unnecessary systems enhances the security of a server in several ways.

Firstly, they cannot be compromised, nor can they be used as attack vectors to alter the services that are essential for the server to operate.

Indeed, it should be kept in mind that each component added to a server increases the risk of compromising it.

Furthermore, the server can be configured to better meet the requirements of a particular service, while improving performance and the overall level of security.

Finally, reducing services means limiting the number of log entries, which makes it easier to monitor and detect unusual events.

Controlling and securing server access: Focus on SSH

Presentation of SSH: an essential tool to ensure proper management and secure administration of the server

SSH (Secure Shell) is both a communication protocol and a computer program allowing local and remote administration of a server.

Its most common implementation is OpenSSH, which can be found on many systems, including servers (Unix, Linux, Windows) as well as workstations and network equipment.

Indeed, OpenSSH is a suite of tools offering many features, including: a server (sshd), several clients – remote shell connection (ssh) / file transfer and download (scp and sftp), a key generation tool (ssh-keygen), a keychain service (ssh-agent and ssh-add), etc.

SSH currently exists in two versions: SSHv1 and SSHv2. SSHv1 contains vulnerabilities that have been fixed in the second version. Therefore, for enhanced security, only version 2 of the SSH protocol should be authorized.

The most widely used feature of SSH is remote administration, which consists of connecting to a remote machine and launching a shell session following authentication.

In this context, the advantage of SSH is its security. Another commonly used feature is the transfer and download of files, both from a client to a server and from a server to a client.

For this purpose, SSH offers two mechanisms: SCP and SFTP, which should always be preferred to older protocols such as RCP and FTP.

Public key / certificate authentication

Not ensuring the authenticity of a server can have several security impacts, including the inability to verify that you are communicating with the correct server, with consequent risks of spoofing and exposure to Man in The Middle attacks.

SSH relies on asymmetric encryption for authentication, and thus ensures the legitimacy of the server being contacted before access is granted.

Moreover, this control is done in several ways with OpenSSH.

Either by ensuring that the public key fingerprint obtained previously with ssh-keygen and presented by the server is the correct one, or by verifying the signature of the certificate presented by the server with a certification authority known to the client.

Security of authentication and user access control

Users of a system always have rights, no matter how small.

It is therefore important to protect their access via a secure authentication mechanism and to thwart brute force attacks as much as possible.

Firstly, each user must have his or her own account to ensure better traceability and the allocation of access rights must always follow the principle of least privilege.

Furthermore, access to a service should be restricted to users who have a justified need and are explicitly authorized.

Regarding users authorized to configure the operating system of servers, they should be limited to a small number of designated administrators.

Furthermore, it is strongly discouraged to use authentication mechanisms in which authentication information is transmitted in the clear over a network, as this information can be intercepted via Man In the Middle attacks and used by an attacker to impersonate an authorized user.

Finally, to ensure secure user authentication, the following measures should be implemented:

- Removing default accounts because the default configuration of the operating system often includes guest accounts, administrator accounts and accounts associated with services. The names and passwords of these accounts are known to attackers.
- Implement a proper password policy. All passwords should be complex and difficult to guess. Above all, they should be long (more than 15 characters). To achieve this, nothing is better than the implementation of a password manager.

Logging and monitoring of server events

Logging is a central aspect of a security strategy.

It is a control mechanism for monitoring the network, systems and servers; and most importantly, it ensures the traceability of all normal and suspicious events.

Indeed, log files can be used to track the activities of an attacker and thus detect unusual events or failed or successful intrusion attempts.

To facilitate the management and exploitation of logs, they should be centralized on a dedicated server.

This is even more important because, in the event of a machine being compromised, it is likely that the logs will be destroyed or altered by the attacker.

Centralizing, regularly backing up and duplicating logs will ensure that a copy is always kept. And given their importance, it is essential to restrict access to logs to authorized users only.

Securing APIs, websites and web applications hosted on a server

APIs, websites and web applications hosted on a server must also be secured.

These can be used as attack vectors to compromise the server if they are vulnerable to SQL injections for example.

CAPTCHA and OTP

CAPTCHA

CAPTCHA

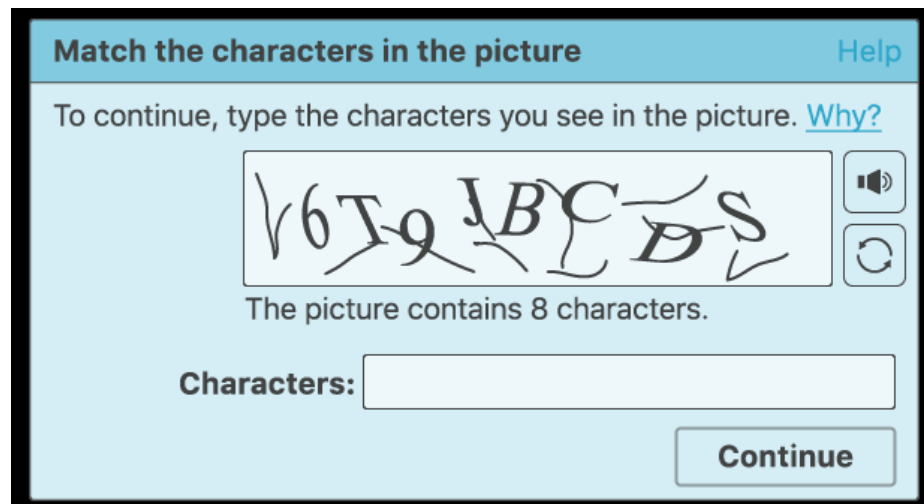
- What is a CAPTCHA?
 - A test is designed to determine if an online user is really a human and not a bot.
 - CAPTCHA is an acronym that stands for "Completely Automated Public Turing test to tell Computers and Humans Apart."
 - Users often encounter CAPTCHA and reCAPTCHA tests on the Internet.
 - Such tests are one way of managing bot activity , although the approach has its drawbacks.

CAPTCHA

- What is a CAPTCHA?
 - Although CAPTCHAs are designed to block automated bots
 - CAPTCHAs are themselves automated
 - They're programmed to pop up in certain places on a website, and they automatically pass or fail users

How does a CAPTCHA work?

- Classic CAPTCHAs, which are still in use on some web properties today, involve asking users to identify letters.
- The letters are distorted so that bots are not likely to be able to identify them.
- To pass the test, users have to interpret the distorted text, type the correct letters into a form field, and submit the form.
- If the letters don't match, users are prompted to try again. Such tests are common in login forms, account signup forms, online polls, and e-commerce checkout pages.



Match the characters in the picture [Help](#)

To continue, type the characters you see in the picture. [Why?](#)

V6T91BCDS

The picture contains 8 characters.

Characters:

Continue

How does a CAPTCHA work?

- The idea is that a computer program such as a bot will be unable to interpret the distorted letters,
- while a human being, who is used to seeing and interpreting letters in all kinds of contexts –different fonts, different handwritings, etc. – will usually be able to identify them.
- The best that many bots will be able to do is input some random letters, making it statistically unlikely that they will pass the test.
- Thus, bots fail the test and are blocked from interacting with the website or application, while humans are able to continue using it like normal.

How does a CAPTCHA work?

- Advanced bots are able to use machine learning to identify these distorted letters, so these kinds of CAPTCHA tests are being replaced with more complex tests.
- Google reCAPTCHA has developed a number of other tests to sort out human users from bots.

What is reCAPTCHA?

- reCAPTCHA is a free service Google offers as a replacement for traditional CAPTCHAs
- reCAPTCHA technology was developed by researchers at Carnegie Mellon University, then acquired by Google in 2009.
- reCAPTCHA is more advanced than the typical CAPTCHA tests.
- Like CAPTCHA, some reCAPTCHAs require users to enter images of text that computers have trouble deciphering.
- Unlike regular CAPTCHAs, reCAPTCHA sources the text from real-world images: pictures of street addresses, text from printed books, text from old newspapers, and so on.

What is reCAPTCHA?

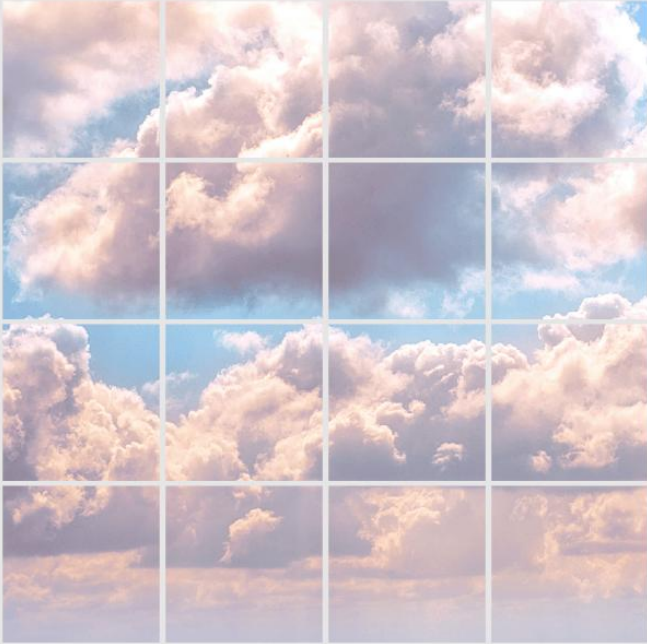
- Over time, Google has expanded the functionality of reCAPTCHA tests so that they no longer have to rely on the old style of identifying blurry or distorted text.
- Other types of reCAPTCHA tests include:
 - Image recognition
 - Checkbox
 - General user behavior assessment (no user interaction at all)






What is reCAPTCHA?

- How does an image recognition reCAPTCHA test work?
- For an image recognition reCAPTCHA test, typically users are presented with 9 or 16 square images.
- The images may all be from the same large image, or they may each be different.
- A user has to identify the images that contain certain objects, such as animals, trees, or street signs.
- If their response matches the responses from most other users who have submitted the same test, the answer is considered "correct" and the user passes the test.


Select all squares with clouds.



   [Report a problem](#)

[Verify](#)

☐ I'm not a robot



reCAPTCHA tests with a single checkbox

What is reCAPTCHA?

- **How does an image recognition reCAPTCHA test work?**
- For an image recognition reCAPTCHA test, typically users are presented with 9 or 16 square images.
- The images may all be from the same large image, or they may each be different.
- A user has to identify the images that contain certain objects, such as animals, trees, or street signs.
- If their response matches the responses from most other users who have submitted the same test, the answer is considered "correct" and the user passes the test.

one-time password (OTP)

OTP

- A one-time password (OTP) is a string of numbers and/or characters that is generated and sent to a user to be used for a single login attempt or transaction.

What are the benefits of OTPs?

- OTPs reduce the risk around passwords.
- **Forgotten passwords**
 - One of the most common uses of OTPs is the case where a user has forgotten their password, or had their account breached.
 - An OTP may be issued to the user to access their account before they are prompted to reset their password.

What are the benefits of OTPs?

- **Replay attacks**

- In a replay attack, a user's login credentials, including their password, are intercepted.
- If the password is static, the attacker would now have access to that user's account.
- But when an OTP is used, the password intercepted by the hacker is no longer valid as it was already used once when the user logged into their account and can thus no longer be reused.

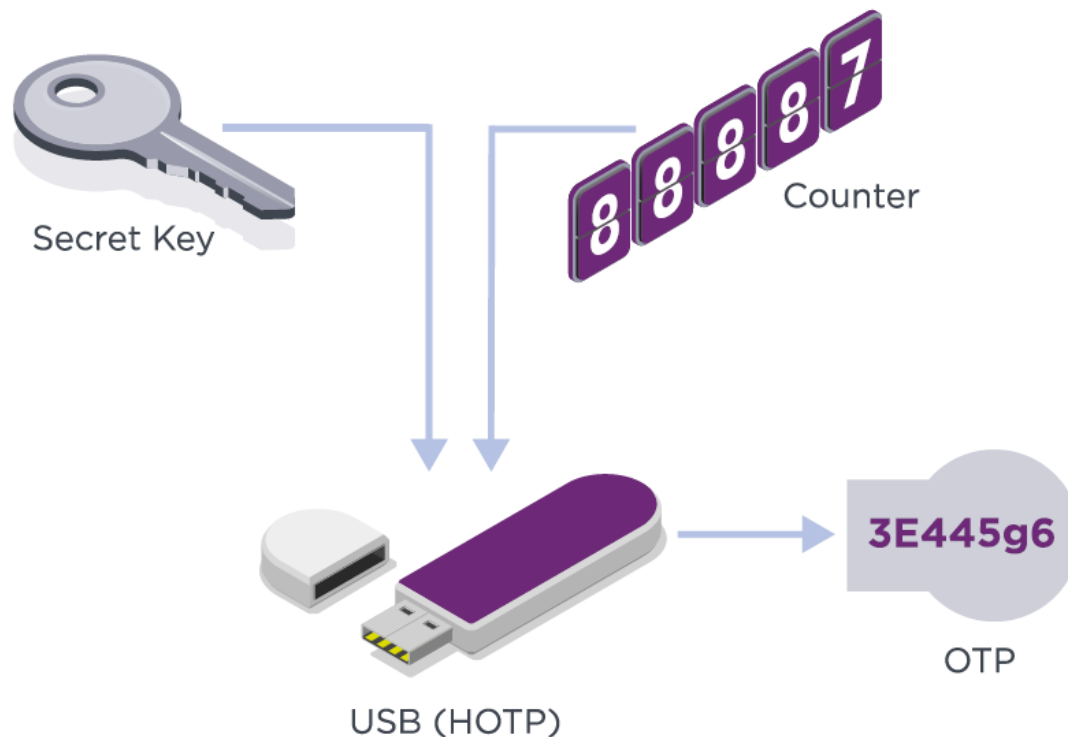
- **Multi-factor authentication**

- OTPs can add an additional layer of authentication.
- Using security tokens, OTPs can be generated for users to provide as an additional form of authentication, which increases security and reduces the risk of a breach.

What are the types of OTPs?

- **Hash-based OTP (HOTP)**

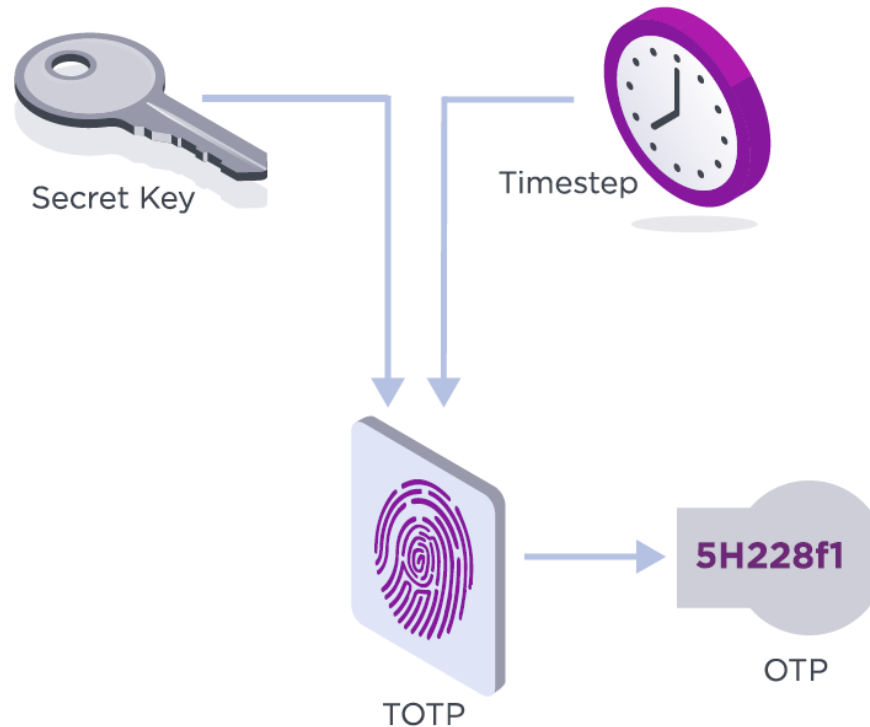
- This type of OTP is generated and sent to a user based on a hash algorithm that syncs the OTP code with counter that changes incrementally.



What are the types of OTPs?

- **Time-based OTP (TOTP)**

- This type of OTP is time-based, in that it provides a window of time within which the OTP code will be valid.
- In general, time steps are 30-60 seconds in length. If the user does not enter the OTP code within the specified time step, they must request a new one.



How are OTPs provided to users securely?

- OTPs are generated and sent to users securely using security tokens.
- **Hard tokens**
 - Smart cards, USB keys, keyless entry systems, mobile phones, and Bluetooth tokens are all capable of generating OTPs.
 - A hard token may be connected, disconnected, or completely contactless.
- **Soft tokens**
 - A push notification to email, via SMS, or an app is the common form of OTP soft tokens.

How are OTPs provided to users securely?

- Autonomous security mechanism where a user is provided an OTP for every login.
- Thus, these terms should not be used synonymously as OTP is just one of many forms of 2FA/MFA and can also stand alone as its own security solution.

Is an OTP more secure than a static password?

- Yes.
- OTPs add an additional layer of security to static passwords. Passwords alone are a vulnerable form of identity verification, responsible for 81% of security breaches.
- Adding another layer of authentication to passwords ensures better security.
- Of course, you could get rid of passwords altogether by going passwordless.

Security Engineering: Passwords and their limitations

A password is a secret word or phrase or code that you need to know in order to have access to a place or system.

In technical terms, it is a series of letters or numbers that you must type into a computer or computer system in order to be able to use it.

A password is a real-life implementation of **challenge-response authentication** (a set of protocols to protect digital assets and data).

Definition: A string of characters i.e letters, numbers, special characters, used to verify the identity of a user during the authentication process is known as password.

Password Management:

Since passwords are meant to keep the files and data secret and safe so it is prevented the unauthorized access, password management refers to the practices and set of rules or principles or standards that one must follow or at least try to seek help from in order to be a good/strong password and along with its storage and management for the future requirements.

Issues Related to Managing Passwords:

The main problem with password management is that it is not safe to use the same password for multiple sites, therefore having different passwords for different sites and on top of that remembering them is quite difficult.

As per the statistics, more than 65% of people reuse passwords across accounts and the majority do not change them, even after a known breach.

Meanwhile, 25% reset their passwords once a month or more because they forgot them.

To escape from this situation people often tend to use password managers (A password manager is a computer program that allows users to store, generate, and manage their passwords for local applications and online services.).

Password managers to a certain extent reduce the problem by having to remember only one "master password" instead of having to remember multiple passwords.

The only problem with having a master password is that once it is out or known to an attacker, the rest of all the passwords become available.

The main issues related to managing passwords are as follows:

- Login spoofing
- Sniffing attack
- Brute force attack
- Shoulder surfing attack
- Data breach

Methods to Manage Password:

There are a lot of good practices that we can follow to generate a strong password and also the ways to manage them.

Strong and long passwords: A minimum length of 8 to 12 characters long, also it should contain at least three different character sets (e.g., uppercase characters, lowercase characters, numbers, or symbols)

Password Encryption: Using irreversible end-to-end encryption is recommended. In this way, the password remains safe even if it ends up in the hands of cybercriminals.

Multi-factor Authentication (MFA): Adding some security questions and a phone number that would be used to confirm that it is indeed you who is trying to log in will enhance the security of your password.

Make the password pass the test: Yes, put your password through some testing tools that you might find online in order to ensure that it falls under the strong and safe password category.

Avoid updating passwords frequently: Though it is advised or even made mandatory to update or change your password as frequently as in 60 or 90 days.

Attacks on Passwords:

Password attacks are one of the most common forms of corporate and personal data breach. A password attack is simply when a hacker try to steal your password.

In 2020, 81% of data breaches were due to compromised credentials. Because passwords can only contain so many letters and numbers, passwords are becoming less safe

Hackers know that many passwords are poorly designed, so password attacks will remain a method of attack as long as passwords are being used.

Protect yourself from password attacks with the information below.

1. Phishing

Phishing is when a hacker posing as a trustworthy party sends you a fraudulent email, hoping you will reveal your personal information voluntarily.

Sometimes they lead you to fake "reset your password" screens; other times, the links install malicious code on your device. We highlight several examples on the One Login blog.

Here are a few examples of phishing: To avoid phishing attacks, follow these steps:

Regular phishing.

You get an email from what looks like goodwebsite.com asking you to reset your password, but you didn't read closely and it's actually goodwobsite.com. You "reset your password" and the hacker steals your credentials.

Spear phishing.

A hacker targets you specifically with an email that appears to be from a friend, colleague, or associate.

It has a brief, generic blurb ("Check out the invoice I attached and let me know if it makes sense.") and hopes you click on the malicious attachment.

Smishing and vishing.

You receive a text message (SMS phishing, or smishing) or phone call (voice phishing, or vishing) from a hacker who informs you that your account has been frozen or that fraud has been detected. You enter your account information and the hacker steals it.

Whaling.

You or your organization receive an email purportedly from a senior figure in your company. You don't do your homework on the email's veracity and send sensitive information to a hacker.

To avoid phishing attacks, follow these steps:

Check who sent the email:

look at the From: line in every email to ensure that the person they claim to be matches the email address you're expecting.

Double check with the source:

when in doubt, contact the person who the email is from and ensure that they were the sender.

Check in with your IT team:

your organization's IT department can often tell you if the email you received is legitimate.

2. Man-in-the-Middle Attack

Man-in-the middle (MitM) attacks are when a hacker or compromised system sits in between two uncompromised people or

systems and deciphers the information they're passing to each other, including passwords.

If Alice and Bob are passing notes in class, but Jeremy has to relay those notes, Jeremy has the opportunity to be the man in the middle.

Similarly, in 2017, Equifax removed its apps from the App Store and Google Play store because they were passing sensitive data over insecure channels where hackers could have stolen customer information.

To help prevent man-in-the-middle attacks:

Enable encryption on your router.

If your modem and router can be accessed by anyone off the street, they can use "sniffer" technology to see the information that is passed through it.

Use strong credentials and two-factor authentication.

Many router credentials are never changed from the default username and password.

If a hacker gets access to your router administration, they can redirect all your traffic to their hacked servers.

Use a VPN.

A secure virtual private network (VPN) will help prevent man-in-the-middle attacks by ensuring that all the servers you send data to are trusted.

3. Brute Force Attack

If a password is equivalent to using a key to open a door, a brute force attack is using a battering ram.

A hacker can try 2.18 trillion password/username combinations in 22 seconds, and if your password is simple, your account could be in the crosshairs.

To help prevent brute force attacks:

- **Use a complex password.**

The difference between an all-lowercase, all-alphabetic, six-digit password and a mixed case, mixed-character, ten-digit password is enormous.

As your password's complexity increases, the chance of a successful brute force attack decreases.

- **Enable and configure remote access.**

Ask your IT department if your company uses remote access management. An access management tool like OneLogin will mitigate the risk of a brute-force attack.

- **Require multi-factor authentication.**

- If multi-factor authentication (MFA) is enabled on your account, a potential hacker can only send a request to your second factor for access to your account.
- Hackers likely won't have access to your mobile device or thumbprint, which means they'll be locked out of your account.

4. Dictionary Attack

A type of brute force attack, dictionary attacks rely on our habit of picking "basic" words as our password, the most common of which hackers have collated into "cracking dictionaries."

More sophisticated dictionary attacks incorporate words that are personally important to you, like a birthplace, child's name, or pet's name.

To help prevent a dictionary attack:

- **Never use a dictionary word as a password.**
 - If you've read it in a book, it should never be part of your password.
 - If you must use a password instead of an access management tool, consider using a password management system.
- **Lock accounts after too many password failures.**
 - It can be frustrating to be locked out of your account when you briefly forget a password, but the alternative is often account insecurity.
 - Give yourself five or fewer tries before your application tells you to cool down.
- **Consider investing in a password manager.**
 - Password managers automatically generate complex passwords that help prevent dictionary attacks.

~~5. Credential Stuffing~~

If you've suffered a hack in the past, you know that your old passwords were likely leaked onto a disreputable website.

Credential stuffing takes advantage of accounts that never had their passwords changed after an account break-in. Hackers will try various combinations of former usernames and passwords, hoping the victim never changed them.

To help prevent credential stuffing:

6. Keyloggers

Keyloggers are a type of malicious software designed to track every keystroke and report it back to a hacker. Typically, a user

will download the software believing it to be legitimate, only for it to install a keylogger without notice.

To protect yourself from keyloggers:

Monitor your accounts. There are paid services that will monitor your online identities, but you can also use free services like [haveIbeenpwned.com](https://haveibeenpwned.com) to check whether your email address is connected to any recent leaks.

Regularly change your passwords. The longer one password goes unchanged, the more likely it is that a hacker will find a way to crack it.

Use a password manager. Like a dictionary attack, many credential stuffing attacks can be avoided by having a strong and secure password. A password manager helps maintain those.

Advanced Security Topics

Secure email systems

Pretty Good Privacy (PGP)

PGP stands for Pretty Good Privacy (PGP) which is invented by Phil Zimmermann.

PGP was designed to provide all four aspects of security, i.e., **privacy, integrity, authentication, and non-repudiation** in the sending of email.

PGP uses a digital signature (a combination of hashing and public key encryption) to provide integrity, authentication, and non-repudiation.

PGP uses a combination of secret key encryption and public key encryption to provide privacy.

Therefore, we can say that the digital signature uses one hash function, one secret key, and two private-public key pairs.

PGP is an open source and freely available software package for email security.

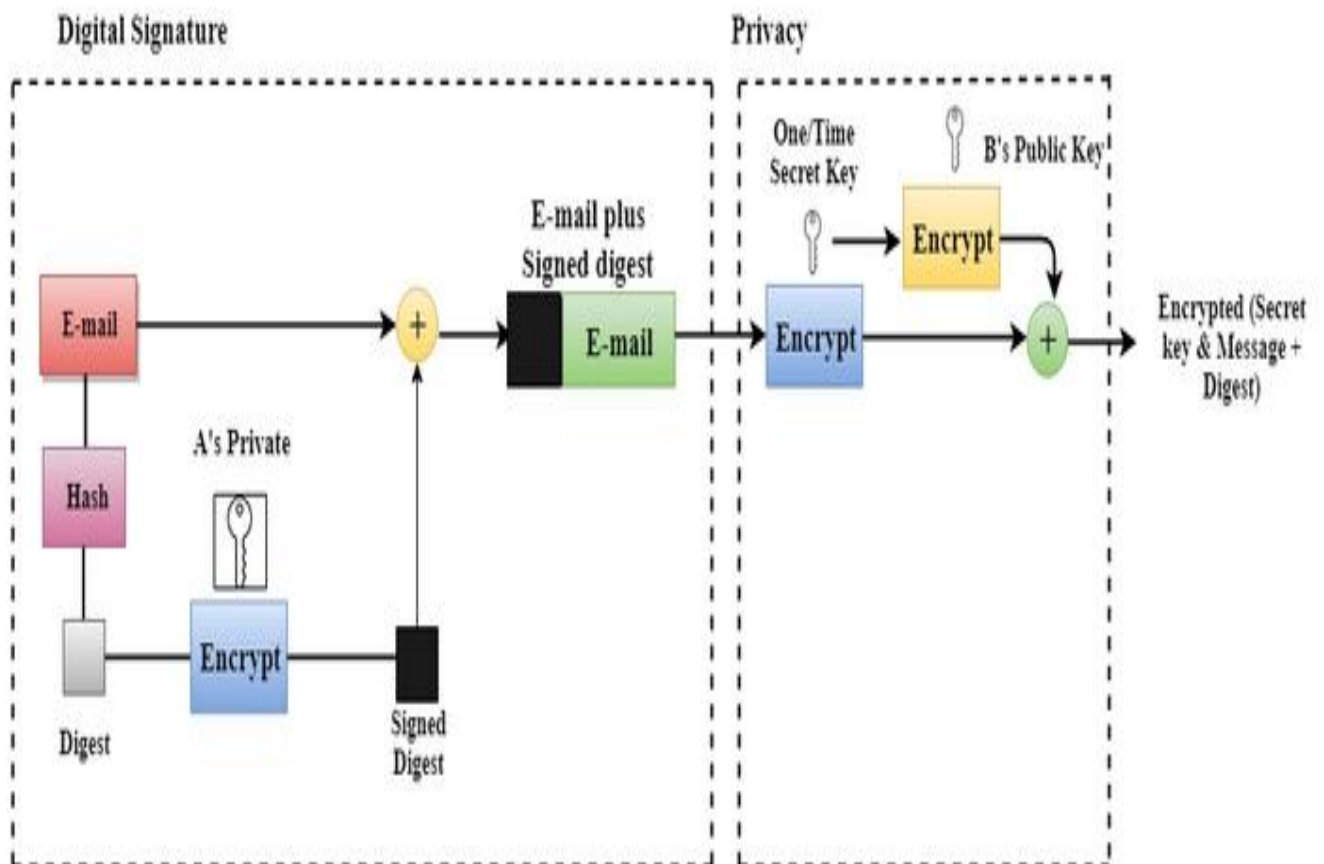
PGP provides authentication through the use of Digital Signature.

It provides confidentiality through the use of symmetric block encryption.

It provides compression by using the ZIP algorithm, and EMAIL compatibility using the radix-64 encoding scheme

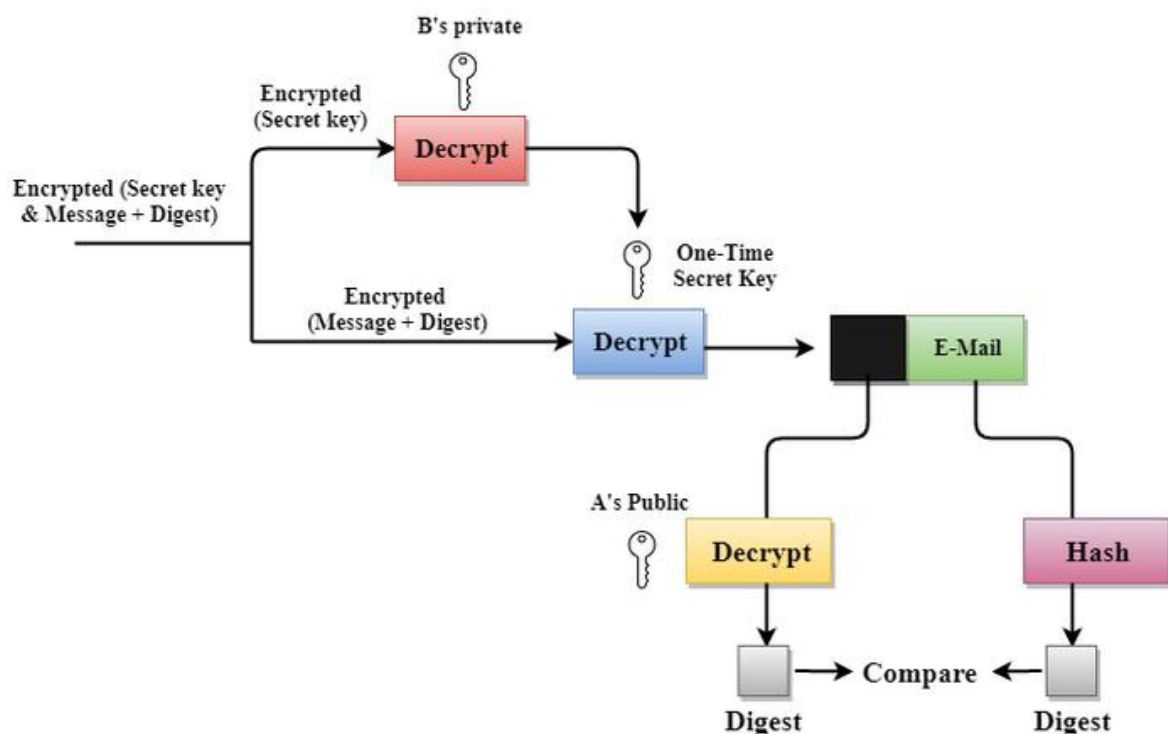
Following are the steps taken by PGP to create secure e-mail at the sender site:

- The e-mail message is hashed by using a hashing function to create a digest.
- The digest is then encrypted to form a signed digest by using the sender's private key, and then signed digest is added to the original email message.
- The original message and signed digest are encrypted by using a one-time secret key created by the sender.
- The secret key is encrypted by using a receiver's public key.
- Both the encrypted secret key and the encrypted combination of message and digest are sent together.



Following are the steps taken to show how PGP uses hashing and a combination of three keys to generate the original message:

- The receiver receives the combination of encrypted secret key and message digest is received.
- The encrypted secret key is decrypted by using the receiver's private key to get the one-time secret key.
- The secret key is then used to decrypt the combination of message and digest.
- The digest is decrypted by using the sender's public key, and the original message is hashed by using a hash function to create a digest.
- Both the digests are compared if both of them are equal means that all the aspects of security are preserved.



Secure/Multipurpose Internet Mail Extensions (S/MIME)

Secure/Multipurpose Internet Mail Extensions (S/MIME) is an end-end encryption protocol for sending digitally signed and encrypted emails that support data confidentiality, authenticity, and integrity.

To understand how S/MIME works we need to understand the following first:

- Digital signatures and signature verification
- Message encryption and decryption
- Public key
- Digital certificates

Digital signatures and verification

With digital signature, S/MIME verifies the identity of the sender of the email.

This verification ensures the following:

- Message in the email is the exact message sent by the sender.
- Message is received from the right sender and not someone pretending to be the sender.

Message encryption and decryption

S/MIME uses encryption to protect the content of the email, which ensures that only the receiver can decrypt the content.

Encryption creates coded information so that it cannot be read or understood until it is decoded and readable.

Message encryption helps with the two key security factors of confidentiality and data integrity.

Public key

S/MIME uses key pairs and asymmetric cryptography.

A private key in a key pair belongs only to the sender.

If the private key has been used, the owner of that key has used it.

Public key cryptography ensures secure communication between the sender and the receiver.

Both have a key-pair, with one being private and the other public .

Public keys are shared between the sender and the receiver.

A public key is paired to only one private key.

The corresponding public key is used to identify its paired private key and only its paired private key.

A public key can be used by multiple recipients.

A key pair can be used to

- Sign and verify a signature
- Encrypt and decrypt the content of an email

S/MIME digital signatures and encryption require each sender and recipient to have it enabled.

They also need to send or exchange public keys through digital certificates to identify each other.

Digital certificates

Digital certificates help in delivering the public key in the key pair.

A digital certificate is a digital credential that provides information about the identity, validity, and any other required information.

Digital certificates are issued by a **certification authority (CA)** and are valid for only a specific period of time.

How does S/MIME work?

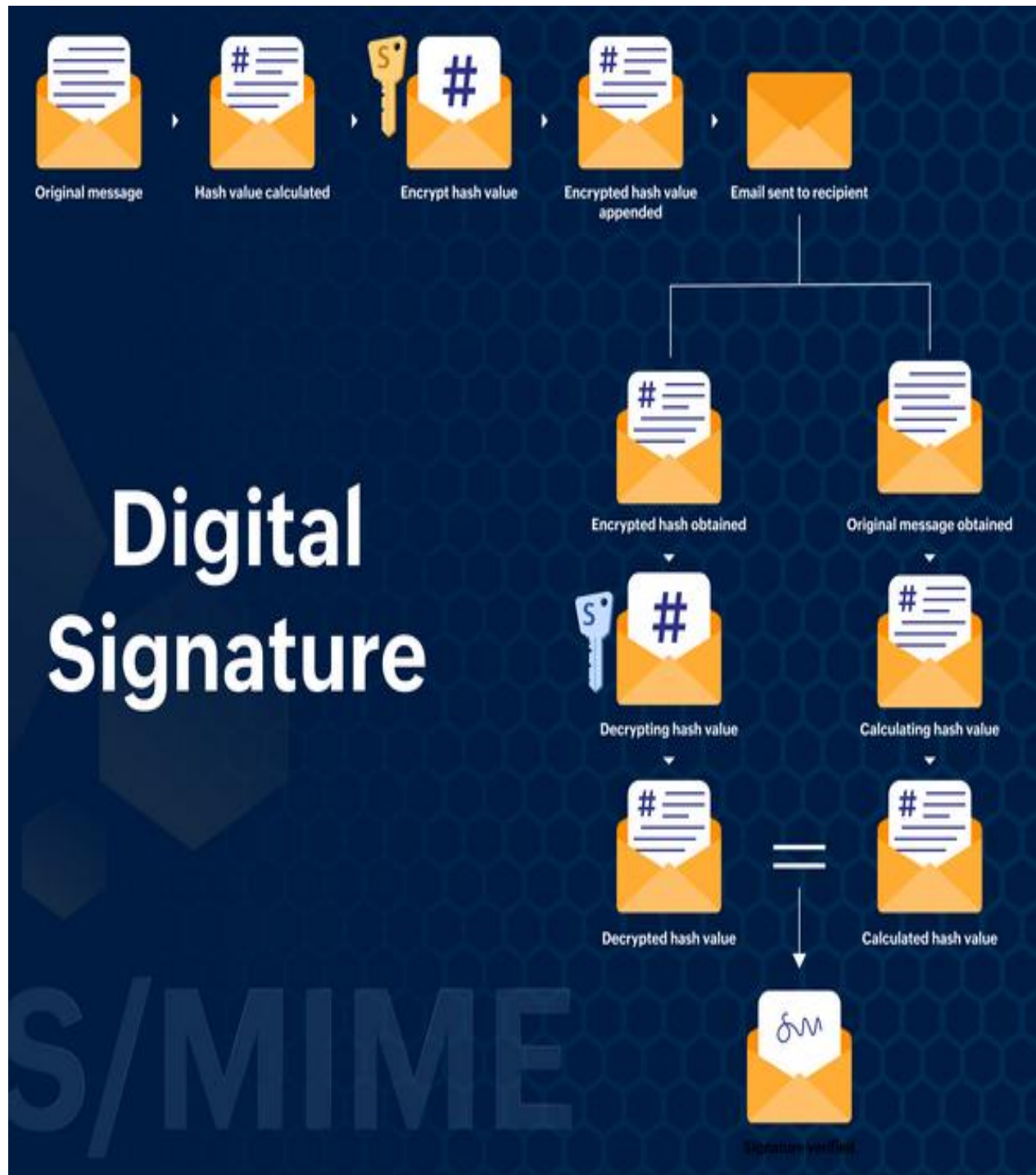
S/MIME works based on asymmetric encryption.

This means that this protocol uses **a two-key system (Public and Private)** that is mathematically related but different, to encrypt and decrypt an email.

The sender and receiver have their own pair of private and public keys in which the public keys are known to the other party.

A S/MIME certificate needs to be installed on the email clients of both the recipient and the sender to ensure email encryption at both ends.

When an email is sent, the sender encrypts the email using the recipient's public key and the recipient decrypts the email using the private key.



Benefits of S/MIME

The encryption and digital signing of an email ensure that the data transmitted through email is confidential, and true to its sender.

S/MIME protects an email in the following methods:

Email Encryption

The email content is encrypted using the recipient's public key, the moment the sender hits the Send button.

Even if the email gets intercepted by anyone, they cannot view the content of the email unless they have access to the private key of the recipient.

Data Confidentiality

The encryption of the email content ensures the confidentiality of the data and attachments sent through the email.

Any attempt to view the content of the email is made void as the data can be decrypted only with the help of a private key unique to the recipient.

Digital Signature

The email will be digitally signed along with encryption on installing the S/MIME certificate.

The email is signed using the private key of the sender and authenticated by the public key of the recipient.

An unaltered digital signature shows that the email content has not been compromised and tampered with.

Signature Authentication

When the sender digitally signs the email using their private key, the recipient validates and authenticates the signature using their public key to ensure that the email is received from a reliable source.

Non-repudiation by the Sender

The digital signature of each sender is unique and is assigned to the user and the domain when the S/MIME certificate is purchased and installed.

This voluntarily provides the non-repudiation of the signature by the sender in case of any legal proceedings.

Content Integrity of the Email

When the recipient of a digitally signed email is validated using the public key of the recipient, they're assured of the absence of any alterations in the content of the email and is intact as and when it was sent.

Difference between PGP and S/MIME :

S.NO	PGP	S/MIME
1.	It is designed for processing the plain texts	It is designed to process email as well as many multimedia files.
2.	PGP is less costly as compared to S/MIME.	S/MIME is comparatively expensive.
3.	PGP is good for personal as well as office use.	It is good for industrial use.
4.	PGP is less efficient than S/MIME.	It is more efficient than PGP.
5.	It depends on user key exchange.	It relies on a hierarchically valid certificate for key exchange.
6.	PGP is comparatively less convenient.	It is more convenient than PGP due to the secure transformation of all the applications.
7.	PGP contains 4096 public keys.	It contains only 1024 public keys.
8.	PGP is the standard for strong encryption.	It is also the standard for strong encryption but has some drawbacks.
9.	PGP is also be used in VPNs.	It is not used in VPNs, it is only used in email services.
10.	PGP uses Diffie hellman digital signature .	It uses Elgamal digital signature .
11.	In PGP Trust is established using Web of Trust.	In S/MIME Trust is established using Public Key Infrastructure.
12.	PGP doesn't provides authentication.	S/MIME provides authentication.
13.	PGP is used for Securing text messages only.	S/MIME is used for Securing Messages and attachments.
14.	Their is less use of PGP in industry .	S/MIME is widely used in industry.
15.	Convenience of PGP is low.	Convenience of S/MIME is High.
16.	Administrative overhead of PGP is high.	Administrative overhead of S/MIME is low.

Domain Keys Identified Mail (DKIM)

Domain Keys Identified Mail (DKIM) is a digital signature added to every email sent from a given email address.

Why use DKIM?

Imagine the following scenario.

You're sending a quick follow-up message to a potential investor after a meeting, "Yvonne, let me know if you would like to proceed with what we discussed earlier."

Some time goes by, and you never got a reply from Yvonne but you bump into her in another meeting and discreetly mention that email.

Puzzled, Yvonne says, "Mark, I never heard from you back."

There are many potential reasons for poor deliverability, but, as it turned out, Mark forgot to set up DKIM authentication for his email account.

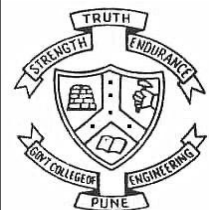
As a result, Yvonne's server wasn't quite sure if it was really Mark emailing her and discarded the message.

The main purpose of DKIM is to prevent spoofing.

Email spoofing is changing the original message's content and sending it from an alternative sender that looks like a trusted source.

This type of cyber attack is widely used for fraud — for example, someone sending payment request messages from an email address that looks like yours (mark@whatevercompany.io vs. mark@whatever-company.io).

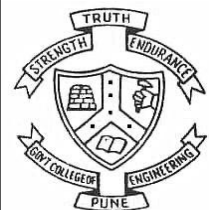
Introduction to Hacking



Department of Computer Engineering and Information Technology
College of Engineering Pune (COEP)
Forerunners in Technical Education

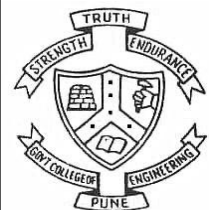
Introduction

- The first known event of hacking had taken place in 1960 at MIT and at the same time, the term "Hacker" was originated.
- Hacking is the act of finding the possible entry points that exist in a computer system or network and finally entering into them.
- Hacking is usually done to gain unauthorized access to a computer system or network, either to harm the systems or to steal sensitive information available on the computer.



Introduction

- Hacking is usually legal as long as it is being done to find weaknesses in a computer or network system for testing purpose. This sort of hacking is what we call Ethical Hacking.
- A computer expert who does the act of hacking is called a "Hacker".
- Hackers are those who seek knowledge, to understand how systems operate, how they are designed, and then attempt to play with these systems.



Types of Hacking

- **Website Hacking**

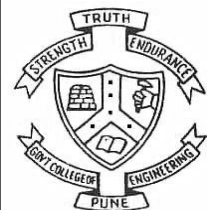
Taking unauthorized control over a web server and its associated software such as databases and other interfaces.

- **Network Hacking**

Gathering information about a network by using tools like Telnet, NS lookup, Ping, Tracert, Netstat, etc. with the intent to harm the network system and hamper its operation.

- **Email Hacking**

It includes getting unauthorized access on an Email account and using it without taking the consent of its owner.



Types of Hacking

- **Ethical Hacking**

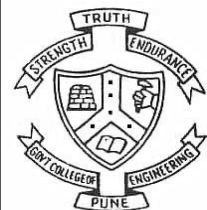
Involves finding weaknesses in a computer or network for testing purpose and finally getting them fixed.

- **Password Hacking**

The process of recovering secret passwords from data that has been stored in or transmitted by a computer system.

- **Computer Hacking**

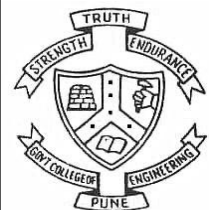
The process of stealing computer ID and password by applying hacking methods and getting unauthorized access to a computer system.



Advantages of Hacking

Hacking is quite useful in the following scenarios –

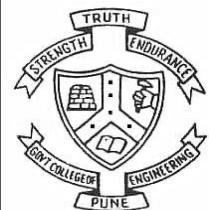
- To recover lost information, especially in case you lost your password.
- To perform penetration testing to strengthen computer and network security.
- To put adequate preventative measures in place to prevent security breaches.
- To have a computer system that prevents malicious hackers from gaining access.



Disadvantages of Hacking

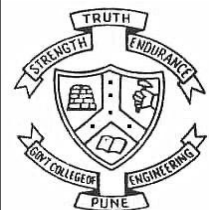
Hacking is quite dangerous if it is done with harmful intent. It can cause

- Massive security breach.
- Unauthorized system access on private information.
- Privacy violation.
- Hampering system operation.
- Denial of service attacks.
- Malicious attack on the system.

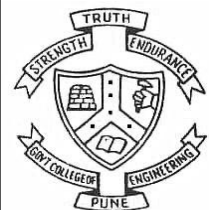


Purpose of Hacking

- Just for fun
- Show-off
- Steal important information
- Damaging the system
- Hampering privacy
- Money extortion
- System security testing
- To break policy compliance



Ethical Hacking - Process



Ethical Hacking - Process

Reconnaissance

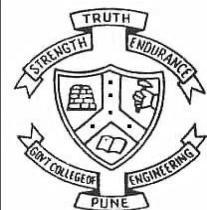
- The attacker **gathers information** about a target using active or passive means.
- The tools that are widely used in this process are NMAP, Hping, Maltego, and Google Dorks.

Scanning

- The attacker begins to actively probe a target machine or network for **vulnerabilities that can be exploited.**
- The tools used in this process are Nessus, Nexpose, and NMAP.

Gaining Access

- The vulnerability is located and you attempt to exploit it in order to enter into the system.
- The primary tool that is used in this process is **Metasploit.**



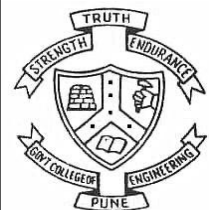
Ethical Hacking - Process

Maintaining Access

- The hacker has already gained access into a system.
- After gaining access, the hacker **installs some backdoors** in order to enter into the system when he needs access in this owned system in future.
- Metasploit is the preferred tool in this process.

Clearing Tracks

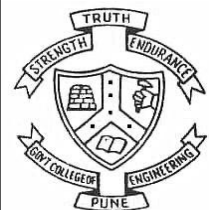
- This process is actually an unethical activity.
- It has to do with the **deletion of logs** of all the activities that take place during the hacking process.



Ethical Hacking - Process

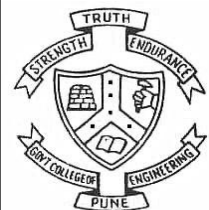
Reporting

- **Reporting** is the last step of finishing the ethical hacking process.
- Here the Ethical Hacker compiles a report with his findings and the job that was done such as the tools used, the success rate, vulnerabilities found, and the exploit processes.



Ethical Hacking - Reconnaissance

- Reconnaissance is a set of processes and techniques (Footprinting, Scanning & Enumeration) used to covertly discover and collect information about a target system.
- During reconnaissance, an ethical hacker attempts to gather as much information about a target system as possible, following the seven steps listed below
 - Gather initial information
 - Determine the network range
 - Identify active machines
 - Discover open ports and access points
 - Fingerprint the operating system
 - Uncover services on ports
 - Map the network

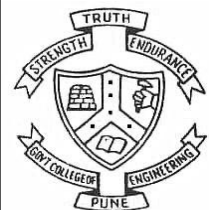


Ethical Hacking - Reconnaissance

- Reconnaissance takes place in two parts
 - Active Reconnaissance and
 - Passive Reconnaissance.

Active Reconnaissance

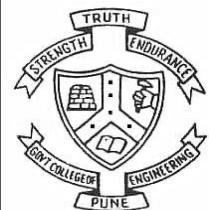
- In this process, you will directly interact with the computer system to gain information.
- This information can be relevant and accurate.
- But there is a risk of getting detected if you are planning active reconnaissance without permission.
- If you are detected, then system admin can take severe action against you and trail your subsequent activities.



Ethical Hacking - Reconnaissance

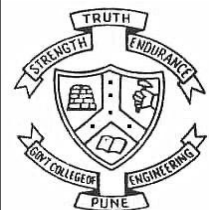
Passive Reconnaissance

- In this process, you will not be directly connected to a computer system.
- This process is used to gather essential information without ever interacting with the target systems.



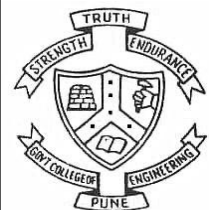
Ethical Hacking - Footprinting

- Footprinting is a part of reconnaissance process which is used for gathering possible information about a target computer system or network.
- Footprinting could be both passive and active.
- Reviewing a company's website is an example of passive footprinting.
- Whereas attempting to gain access to sensitive information through social engineering is an example of active information gathering.
- Footprinting is basically the first step where hacker gathers as much information as possible to find ways to intrude into a target system or at least decide what type of attacks will be more suitable for the target.



Ethical Hacking - Footprinting

- During this phase, a hacker can collect the following information
 - Domain name
 - IP Addresses
 - Namespaces
 - Employee information
 - Phone numbers
 - E-mails
 - Job Information

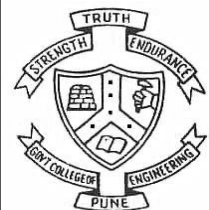


Domain Name Information

- You can use

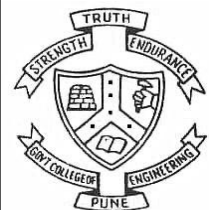
<http://www.whois.com/whois>

- The website to get detailed information about
- domain name information including its owner, its registrar, date of registration, expiry, name server, owner's contact information, etc.



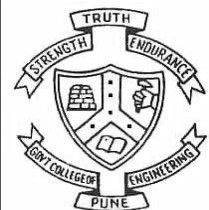
Domain Name Information

- It's always recommended to keep your domain name profile a private one which should hide the above-mentioned information from potential hackers.



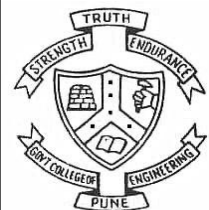
Finding IP Address

- You can use ping command at your prompt.
- This command is available on Windows as well as on Linux OS.
- Following is the example to find out the IP address of coep.org.in
- \$ping coep.org.in



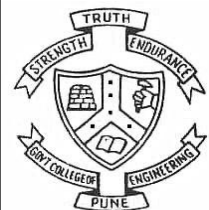
Finding Hosting Company

- Once you have the website address, you can get further detail by using ip2location.com website.
- Here the ISP row gives you the detail about the hosting company because IP addresses are usually provided by hosting companies only.



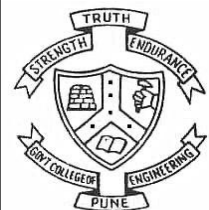
Finding Hosting Company

- If a computer system or network is linked with the Internet directly, then you cannot hide the IP address and the related information such as the hosting company, its location, ISP, etc.
- If you have a server containing very sensitive data, then it is recommended to keep it behind a secure proxy so that hackers cannot get the exact details of your actual server.
- This way, it will be difficult for any potential hacker to reach your server directly.
- Another effective way of hiding your system IP and ultimately all the associated information is to go through a Virtual Private Network (VPN).



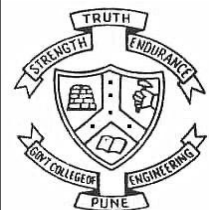
Finding Hosting Company

- If you configure a VPN, then the whole traffic routes through the VPN network, so your true IP address assigned by your ISP is always hidden.



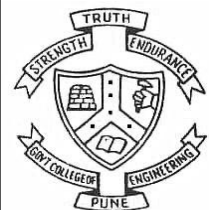
IP Address Ranges

- Small sites may have a single IP address associated with them, but larger websites usually have multiple IP addresses serving different domains and sub-domains.
- You can obtain a range of IP addresses assigned to a particular company using American Registry for Internet Numbers (ARIN).
- <https://www.arin.net/>
- You can enter company name in the highlighted search box to find out a list of all the assigned IP addresses to that company.



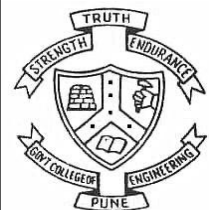
History of the Website

- It is very easy to get a complete history of any website using www.archive.org
- You can enter a domain name in the search box to find out how the website was looking at a given point of time and what were the pages available on the website on different dates.
- Though there are some advantages of keeping your website in an archive database, but if you do not like anybody to see how your website progressed through different stages, then you can request archive.org to delete the history of your website.



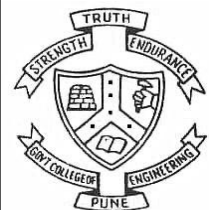
Ethical Hacking - Fingerprinting

- The term OS fingerprinting in Ethical Hacking refers to any method used to determine what operating system is running on a remote computer.
- This could be
 - Active Fingerprinting
 - Passive Fingerprinting
- **Active Fingerprinting**
- It is accomplished by sending specially crafted packets to a target machine and then noting down its response and analyzing the gathered information to determine the target OS.



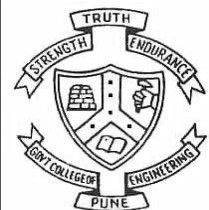
Ethical Hacking - Fingerprinting

- Example to explain how you can use NMAP tool to detect the OS of a target domain.
- **Passive Fingerprinting**
- It is based on sniffer traces from the remote system.
- Based on the sniffer traces (such as Wireshark) of the packets, you can determine the operating system of the remote host.



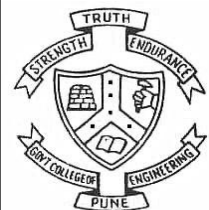
Ethical Hacking - Fingerprinting

- We have the following four important elements that we will look at to determine the OS
 - **TTL** : What the operating system sets the Time-To-Live on the outbound packet.
 - **Window Size** : What the operating system sets the Window Size at.
 - **DF** : Does the operating system set the Don't Fragment bit.
 - **TOS** : Does the operating system set the Type of Service, and if so, at what.



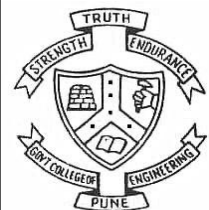
Ethical Hacking - Fingerprinting

- By analyzing these factors of a packet, you may be able to determine the remote operating system.
- This system is not 100% accurate, and works better for some operating systems than others.



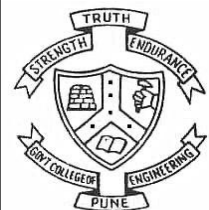
Basic Steps

- Before attacking a system, it is required that you know what operating system is hosting a website.
- Once a target OS is known, then it becomes easy to determine which vulnerabilities might be present to exploit the target system.
- A simple nmap command which can be used to identify the operating system serving a website and all the opened ports associated with the domain name, i.e., the IP address.
- `$nmap -O -v coep.org.in`



Basic Steps

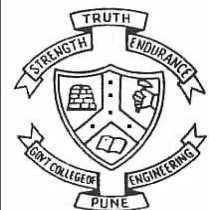
- If you do not have nmap command installed on your Linux system, then you can install it using the following yum command
- `$yum install nmap`
- You can go through nmap command in detail to check and understand the different features associated with a system and secure it against malicious attacks.
- You can hide your main system behind a secure proxy server or a VPN so that your complete identity is safe and ultimately your main system remains safe.



Port Scanning

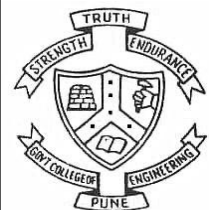
- We have just seen information given by nmap command. This command lists down all the open ports on a given server

PORT	STATE	SERVICE
22/tcp	open	ssh
80/tcp	open	http
443/tcp	open	https
3306/tcp	open	mysql



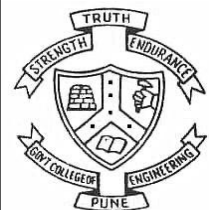
Port Scanning

- You can also check if a particular port is opened or not using the following command
- `$nmap -sT -p 443 coep.org.in`
- It will produce the following result –
Starting Nmap 5.51 (<http://coep.org.in>) at 2021-10-01 10:19 CDT
Nmap scan report for coep.org.in (IP)
Host is up (0.000067s latency).
PORT STATE SERVICE
443/tcp open https
Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds



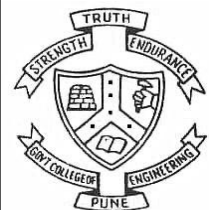
Port Scanning

- Once a hacker knows about open ports, then he can plan different attack techniques through the open ports.
- It is always recommended to check and close all the unwanted ports to safeguard the system from malicious attacks.



Ping Sweep

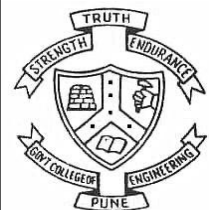
- A ping sweep is a network scanning technique that you can use to determine which IP address from a range of IP addresses map to live hosts.
- Ping Sweep is also known as ICMP sweep.
- You can use fping command for ping sweep.
- This command is a ping-like program which uses the ICMP echo request to determine if a host is up.
- fping is different from ping in that you can specify any number of hosts on the command line, or specify a file containing the lists of hosts to ping.
- If a host does not respond within a certain time limit and/or retry limit, it will be considered unreachable.



Ping Sweep

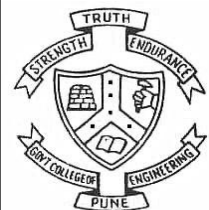
- To disable ping sweeps on a network, you can block ICMP ECHO requests from outside sources.
- This can be done using the following command which will create a firewall rule in iptable.

```
$iptables -A OUTPUT -p icmp --icmp-type echo-request -j DROP
```



DNS Enumeration

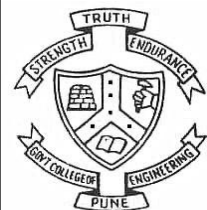
- Domain Name Server (DNS) is like a map or an address book.
- In fact, it is like a distributed database which is used to translate an IP address 192.111.1.120 to a name www.example.com and vice versa.
- DNS enumeration is the process of locating all the DNS servers and their corresponding records for an organization.
- The idea is to gather as much interesting details as possible about your target before initiating an attack.
- You can use **nslookup command** available on Linux to get DNS and host-related information.



DNS Enumeration

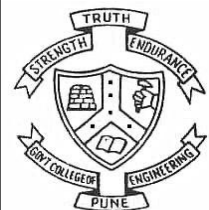
- In addition, you can use the following DNSenum script to get detailed information about a domain

DNSenum.pl



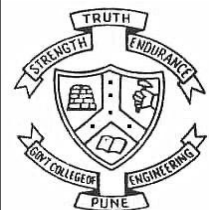
DNS Enumeration

- DNSenum script can perform the following important operations –
 - Get the host's addresses
 - Get the nameservers
 - Get the MX record
 - Perform axfr queries on nameservers
 - Get extra names and subdomains via Google scraping
 - Brute force subdomains from file can also perform recursion on subdomain that has NS records
 - Calculate C class domain network ranges and perform whois queries on them
 - Perform reverse lookups on netranches



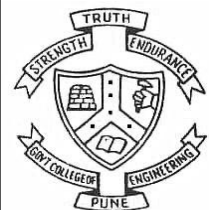
DNS Enumeration

- DNS Enumeration does not have a quick fix and it is really beyond the scope.
- Preventing DNS Enumeration is a big challenge.
- If your DNS is not configured in a secure way, it is possible that lots of sensitive information about the network and organization can go outside and an untrusted Internet user can perform a DNS zone transfer.



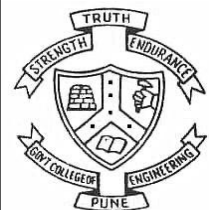
Ethical Hacking - Sniffing

- Sniffing is the process of monitoring and capturing all the packets passing through a given network using sniffing tools.
- It is a form of “tapping phone wires” and get to know about the conversation.
- It is also called wiretapping applied to the computer networks.
- There is so much possibility that if a set of enterprise switch ports is open, then one of their employees can sniff the whole traffic of the network.
- Anyone in the same physical location can plug into the network using Ethernet cable or connect wirelessly to that network and sniff the total traffic.



Ethical Hacking - Sniffing

- In other words, Sniffing allows you to see all sorts of traffic, both protected and unprotected.
- In the right conditions and with the right protocols in place, an attacking party may be able to gather information that can be used for further attacks or to cause other issues for the network or system owner.

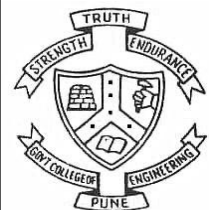


Ethical Hacking - Sniffing

- What can be sniffed?

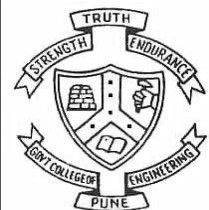
One can sniff the following sensitive information from a network –

- Email traffic
- FTP passwords
- Web traffics
- Telnet passwords
- Router configuration
- Chat sessions
- DNS traffic

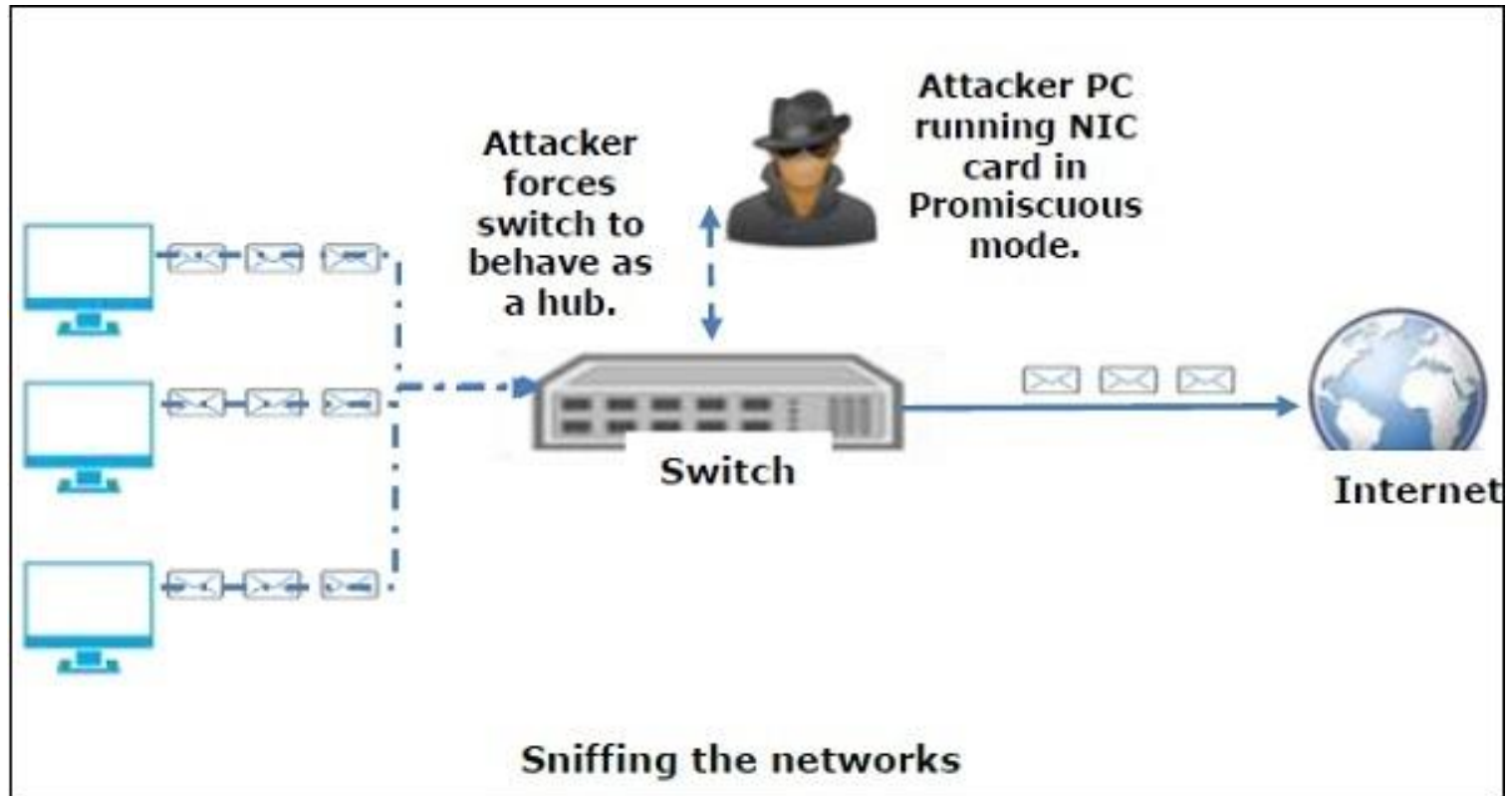


How it works

- A sniffer normally turns the NIC of the system to the promiscuous mode so that it listens to all the data transmitted on its segment.
- Promiscuous mode refers to the unique way of Ethernet hardware, in particular, network interface cards (NICs), that allows an NIC to receive all traffic on the network, even if it is not addressed to this NIC.
- By default, a NIC ignores all traffic that is not addressed to it, which is done by comparing the destination address of the Ethernet packet with the hardware address (a.k.a. MAC) of the device.
- While this makes perfect sense for networking, non-promiscuous mode makes it difficult to use network monitoring and analysis software for diagnosing connectivity issues or traffic accounting.



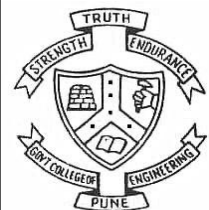
How it works



A sniffer can continuously monitor all the traffic to a computer through the NIC by decoding the information encapsulated in the data packets

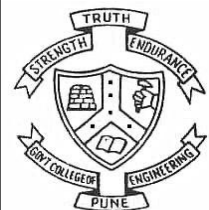
Types of Sniffing

- Sniffing can be either Active or Passive in nature.
- **Passive Sniffing**
- The traffic is locked but it is not altered in any way. It allows listening only.
- It works with **Hub devices**. On a hub device, the traffic is sent to all the ports. In a network that uses hubs to connect systems, all hosts on the network can see the traffic. Therefore, an attacker can easily capture traffic going through.
- The good news is that hubs are almost obsolete nowadays. Most modern networks use switches. Hence, passive sniffing is no more effective.



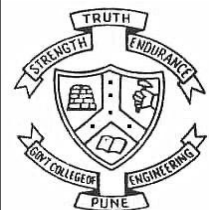
Types of Sniffing

- **Active Sniffing**
- The traffic is not only locked and monitored, but it may also be altered in some way as determined by the attack.
- Active sniffing is used to sniff a switch-based network.
- It involves injecting address resolution packets (ARP) into a target network to flood on the switch content addressable memory (CAM) table.
- CAM keeps track of which host is connected to which port.



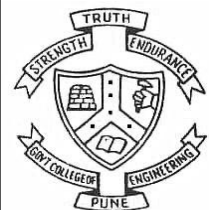
Types of Sniffing

- Following are the Active Sniffing Techniques
 - MAC Flooding
 - DHCP Attacks
 - DNS Poisoning
 - Spoofing Attacks
 - ARP Poisoning



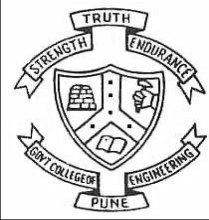
Protocols which are affected

- Protocols such as the tried and true TCP/IP were never designed with security in mind and therefore do not offer much resistance to potential intruders. Several rules lend themselves to easy sniffing
- **HTTP** : It is used to send information in the clear text without any encryption and thus a real target.
- **SMTP (Simple Mail Transfer Protocol)** : SMTP is basically utilized in the transfer of emails. This protocol is efficient, but it does not include any protection against sniffing.
- **NNTP (Network News Transfer Protocol)** : It is used for all types of communications, but its main drawback is that data and even passwords are sent over the network as clear text.



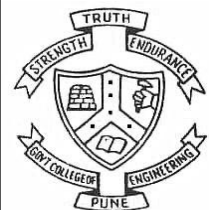
Protocols which are affected

- **POP (Post Office Protocol)** : POP is strictly used to receive emails from the servers. This protocol does not include protection against sniffing because it can be trapped.
- **FTP (File Transfer Protocol)** : FTP is used to send and receive files, but it does not offer any security features. All the data is sent as clear text that can be easily sniffed.
- **IMAP (Internet Message Access Protocol)** : IMAP is same as SMTP in its functions, but it is highly vulnerable to sniffing.
- **Telnet** : Telnet sends everything (usernames, passwords, keystrokes) over the network as clear text and hence, it can be easily sniffed.



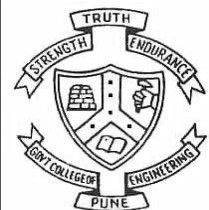
Ethical Hacking - Sniffing Tools

- There are so many tools available to perform sniffing over a network, and they all have their own features to help a hacker analyze traffic and dissect the information. Sniffing tools are extremely common applications.
- **BetterCAP** – BetterCAP is a powerful, flexible and portable tool created to perform various types of MITM attacks against a network, manipulate HTTP, HTTPS and TCP traffic in real-time, sniff for credentials, and much more.
- **Ettercap** – Ettercap is a comprehensive suite for man-in-the-middle attacks. It features sniffing of live connections, content filtering on the fly and many other interesting tricks. It supports active and passive dissection of many protocols and includes many features for network and host analysis.



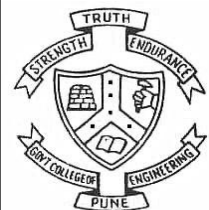
Ethical Hacking - Sniffing Tools

- **Wireshark** – It is one of the most widely known and used packet sniffers. It offers a tremendous number of features designed to assist in the dissection and analysis of traffic.
- **Tcpdump** – It is a well-known command-line packet analyzer. It provides the ability to intercept and observe TCP/IP and other packets during transmission over the network. Available at www.tcpdump.org.
- **WinDump** – A Windows port of the popular Linux packet sniffer tcpdump, which is a command-line tool that is perfect for displaying header information.
- **OmniPeek** – Manufactured by WildPackets, OmniPeek is a commercial product that is the evolution of the product EtherPeek.



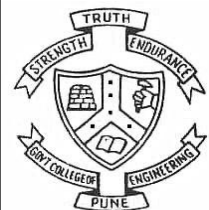
Ethical Hacking - Sniffing Tools

- **Dsniff** – A suite of tools designed to perform sniffing with different protocols with the intent of intercepting and revealing passwords. Dsniff is designed for Unix and Linux platforms and does not have a full equivalent on the Windows platform.
- **EtherApe** – It is a Linux/Unix tool designed to display graphically a system's incoming and outgoing connections.
- **MSN Sniffer** – It is a sniffing utility specifically designed for sniffing traffic generated by the MSN Messenger application.



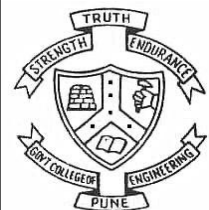
Ethical Hacking - Sniffing Tools

- **NetWitness NextGen** – It includes a hardware-based sniffer, along with other features, designed to monitor and analyze all traffic on a network. This tool is used by the FBI and other law enforcement agencies.
- A potential hacker can use any of these sniffing tools to analyze traffic on a network and dissect information.



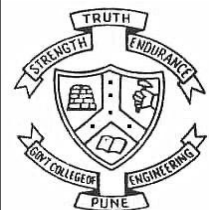
Types of Hackers

- However, if you look at the media's definition of a hacker in the 1990s, you would find a few common characteristics, such as creativity, the ability to solve complex problems, and new ways of compromising targets.
- Therefore, the term has been broken down into three types:
 - White hat hacker
 - Black hat hacker
 - Gray hat hacker



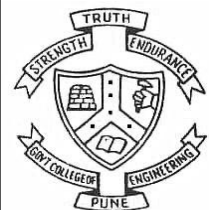
Types of Hackers

- **White hat hacker**
 - This kind of hacker is often referred to as a security professional or security researcher.
 - Such hackers are employed by an organization and are permitted to attack an organization to find vulnerabilities that an attacker might be able to exploit.
- **Black hat hacker**
 - Also known as a cracker, this kind of hacker is referred to as a bad guy, who uses his or her knowledge for negative purposes.
 - They are often referred to by the media as hackers.



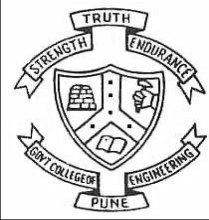
Types of Hackers

- **Gray hat hacker**
 - This kind of hacker is an intermediate between a white hat and a black hat hacker.
 - For instance, a gray hat hacker would work as a security professional for an organization and responsibly disclose everything to them; however, he or she might leave a backdoor to access it later and might also sell the confidential information, obtained after the compromise of a company's target server, to competitors.



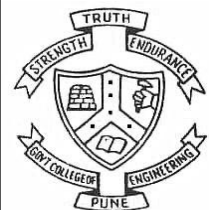
Types of Hackers

- Similarly, we have categories of hackers about whom you might hear oftentimes. Some of them are as follows:
- **Script kiddie**
 - Also known as skid, this kind of hacker is someone who lacks knowledge on how an exploit works and relies upon using exploits that someone else created.
 - A script kiddie may be able to compromise a target but certainly cannot debug or modify an exploit in case it does not work.
- **Elite hacker**
 - An elite hacker is someone who has deep knowledge on how an exploit works; he or she is able to create exploits, but also modify codes that someone else wrote.
 - He or she is someone with elite skills of hacking.



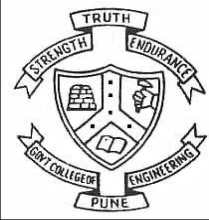
Types of Hackers

- **Hacktivist**
- Hacktivists are defined as group of hackers that hack into computer systems for a cause or purpose.
- The purpose may be political gain, freedom of speech, human rights, and so on.
- **Ethical hacker**
- An ethical hacker is as a person who is hired and permitted by an organization to attack its systems for the purpose of identifying vulnerabilities, which an attacker might take advantage of.
- The sole difference between the terms “hacking” and “ethical hacking” is the permission.



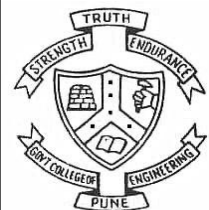
Important Terminologies

- Let's now briefly discuss some of the important terminologies that I will be using throughout
- **Asset**
 - An asset is any data, device, or other component of the environment that supports information related activities that should be protected from anyone besides the people that are allowed to view or manipulate the data/information.
- **Vulnerability**
 - Vulnerability is defined as a flaw or a weakness inside the asset that could be used to gain unauthorized access to it.
 - The successful compromise of a vulnerability may result in data manipulation, privilege elevation, etc



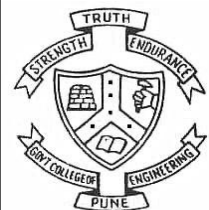
Important Terminologies

- **Threat**
 - A threat represents a possible danger to the computer system.
 - It represents something that an organization doesn't want to happen.
 - A successful exploitation of vulnerability is a threat.
 - A threat may be a malicious hacker who is trying to gain unauthorized access to an asset.
- **Exploit**
 - An exploit is something that takes advantage of vulnerability in an asset to cause unintended or unanticipated behavior in a target system, which would allow an attacker to gain access to data or information.



Important Terminologies

- **Risk**
- A risk is defined as the impact (damage) resulting from the successful compromise of an asset.
- For example, an organization running a vulnerable apache tomcat server poses a threat to an organization and the damage/loss that is caused to the asset is defined as a risk.
- Normally, a risk can be calculated by using the following equation:
- **$\text{Risk} = \text{Threat} * \text{vulnerabilities} * \text{impact}$**



Cloud Security

What is cloud security?

- Cloud security refers
 - to the cybersecurity policies,
 - best practices, controls, and technologies used to secure applications, data, and infrastructure in cloud environments.
- In particular, cloud security works
 - to provide storage and network protection against internal and external threats,
 - access management,
 - data governance and compliance, and
 - disaster recovery.

- Cloud computing has
 - become the technology of choice for companies looking to gain the agility and flexibility needed to accelerate innovation and meet the expectations of today's modern consumers.
- But migrating to more dynamic cloud environments requires
 - new approaches to security to ensure that data remains secure across online infrastructure, applications, and platforms.

How does cloud security work?

- Cloud security mainly focuses on
 - how to implement policies, processes, and technologies together so they ensure data protection, support regulatory compliance, and
 - provide control over privacy, access, and authentication for users and devices.

How does cloud security work?

- Cloud service providers (CSPs) typically follow
 - a shared responsibility model, which means implementing cloud computing security is both the responsibility of the cloud provider and you—the customer.
 - Think of it as a responsibility framework that defines which security tasks belong to the cloud provider and which are the duty of the customer.
 - Understanding where your provider's security responsibilities end and yours begin is critical for building a resilient cloud security strategy


How does cloud security work?

- Broadly speaking,
 - the CSP is always responsible for the cloud and its core infrastructure,
 - while the customer is expected to secure anything that runs “in” the cloud,
 - such as network controls, identity and access management, data, and applications.
 - **Shared responsibility models** vary depending on the service provider and the cloud computing service model you use—the more the provider manages, the more they can protect.

Guide to the shared responsibility model

■ USER'S RESPONSIBILITY ■ SERVICE PROVIDER'S RESPONSIBILITY



 EXAMPLES		APPLICATIONS	MIDDLEWARE	VIRTUALIZATION	DATA	O/S	NETWORKING	RUNTIME	SERVERS	STORAGE
SaaS	Dropbox, Salesforce CRM, Zoom, Microsoft 365, Google Workspace	■	■	■	■	■	■	■	■	■
PaaS	Microsoft Azure App Service, AWS Elastic Beanstalk, Google Kubernetes Engine, Red Hat OpenShift	■	■	■	■	■	■	■	■	■
IaaS	Microsoft Azure, Amazon Web Services (AWS), Google Compute Engine (GCE)	■	■	■	■	■	■	■	■	■

Here's a look at how this typically works:

Cloud computing service model	Your responsibility	CSP responsibility
Infrastructure as a service (IaaS)	You secure your data, applications, virtual network controls, operating system, and user access.	The cloud provider secures compute, storage, and physical network, including all patching and configuration.
Platform as a service (PaaS)	You secure your data, user access, and applications.	The cloud provider secures compute, storage, physical network, virtual network controls, and operating system.
Software as a service (SaaS)	You are responsible for securing your data and user access.	The cloud provider secures compute, storage, physical network, virtual network controls, operating system, applications, and middleware.

Cloud security risks and challenges

- Cloud suffers from similar security risks that you might encounter in traditional environments, such as insider threats, data breaches and data loss, phishing, malware, DDoS attacks, and vulnerable APIs.
- However, most organizations will likely face specific cloud security challenges, including:
 - Lack of visibility
 - Misconfigurations
 - Access management
 - Dynamic workloads
 - Compliance

Lack of visibility

- Cloud-based resources run on infrastructure that is located outside your corporate network and owned by a third party.
- As a result, traditional network visibility tools are not suitable for cloud environments, making it difficult for you to gain oversight into all your cloud assets, how they are being accessed, and who has access to them.

Misconfigurations

- Misconfigured cloud security settings are one of the leading causes of data breaches in cloud environments.
- Cloud-based services are made to enable easy access and data sharing, but many organizations may not have a full understanding of how to secure cloud infrastructure.
- This can lead to misconfigurations, such as leaving default passwords in place, failing to activate data encryption, or mismanaging permission controls.

Access management

- Cloud deployments can be accessed directly using the public internet, which enables convenient access from any location or device.
- At the same time, it also means that attackers can more easily gain authorized resources with compromised credentials or improper access control.

Dynamic workloads

- Cloud resources can be provisioned and dynamically scaled up or down based on your workload needs.
- However, many legacy security tools are unable to enforce policies in flexible environments with constantly changing and ephemeral workloads that can be added or removed in a matter of seconds.

Compliance

- The cloud adds another layer of regulatory and internal compliance requirements that you can violate even if you don't experience a security breach.
- Managing compliance in the cloud is an overwhelming and continuous process.
- Unlike an on-premises data center
 - where you have complete control over your data and
 - how it is accessed, it is much harder for companies to consistently identify all cloud assets and controls,
 - map them to relevant requirements, and properly document everything.

Types of cloud security solutions

- Cloud security is constantly evolving and adapting as new security threats emerge.
- As a result, many different types of cloud security solutions are available on the market today, and the list below is by no means exhaustive.
 - Identity and access management (IAM)
 - Data loss prevention (DLP)
 - Security information and event management (SIEM)
 - Public key infrastructure (PKI)

Identity and access management (IAM)

- IAM services and tools allow administrators
 - to centrally manage and control who has access to specific cloud-based and on-premises resources.
- IAM can enable you
 - to actively monitor and restrict how users interact with services,
 - allowing you to enforce your policies across your entire organization

Data loss prevention (DLP)

- DLP can help you
 - gain visibility into the data you store and
 - process by providing capabilities to automatically discover, classify, and de-identify regulated cloud data

Security information and event management (SIEM)

- SIEM solutions
 - combine security information and security event management to offer automated monitoring, detection, and incident response to threats in your cloud environments.
 - Using AI and ML technologies,
 - SIEM tools allow you to examine and analyze log data generated across your applications and network devices—
 - and act quickly if a potential threat is detected.

Public key infrastructure (PKI)

- PKI is the framework
 - used to manage secure, encrypted information exchange using digital certificates.
- PKI solutions typically provide
 - authentication services for applications and
 - verify that data remains uncompromised and confidential through transport.
- Cloud-based PKI services allow organizations
 - to manage and deploy digital certificates used for user, device, and service authentication.

Ecommerce security

A good ecommerce security strategy is vital to the success of any online business.

Threats can come from many different sources, and 88% of professional hackers can infiltrate an organization in just 12 hours, according to a Data Prot study.

With the risk of unauthorized access to your company's data looming around the corner, you need protect yourself from potential reputational damage and disruptions to your business.

While many ecommerce companies use some baseline security measures, staying vigilant is difficult when hackers constantly change their methods of attack. Merely hoping that you have the right solutions in place isn't good enough.

Understanding and implementing the methods your ecommerce company can use to secure your online store is key to safeguarding your data.

What is ecommerce security?

Ecommerce security is a set of guidelines that ensures safe online transactions.

Just like physical stores invest in security guards or cameras to prevent theft, online stores need to defend against cyberattacks.

According to the 2020 Trustwave Global Security Report, the retail industry was the most-targeted sector for cyberattacks.

In order to adequately protect your company from attack, you first need to know four key terms that are essential to understanding ecommerce security protocols.

Privacy

In the context of ecommerce security, privacy involves preventing unauthorized internal and external threats from accessing customer data.

Disrupting customer privacy is considered a breach of confidentiality and could have devastating consequences for your customers' privacy and your reputation as a retailer.

Privacy measures include antivirus software, firewalls, encryption, and other data protection measures.

Integrity

Integrity refers to how accurate a company's customer data is. Maintaining a clean, curated customer dataset is critical to running a successful ecommerce business.

Using incorrect customer's data — such as their phone number, address, or purchase history — causes people to lose confidence in your ability to protect their data and in your company as a whole.

Authentication

Authentication proves that your business does what it claims and that customers are who they say they are.

Your site should have at least some proof that it sells what it says it does and delivers those goods according to expectations.

Using customer quotes throughout your website and publishing case studies are two strategies for adding to your business's credibility.

Customers should also be required to verify their identities before processing their online transactions.

Requiring two-factor authentication or using magic links to log customers into their accounts are examples of customer authentication.

Non-repudiation

Non-repudiation means neither a company nor a customer can deny transactions they've participated in.

Non-repudiation is somewhat implicit in physical stores but pertains to online purchases as well.

Non-repudiation measures like digital signatures ensure that neither party can deny a purchase after it has been made.

Common ecommerce security threats

There are a multitude of different cyberattacks that could threaten your online business.

It's crucial to know what these threats are and how to prevent them.

The best way to get started is to make sure you understand the basic types of ecommerce security threats.

Phishing

Phishing is a method of cyberattack that tricks victims into providing confidential personal information — like passwords or social security numbers — via email, text, or phone.

Phishing messages convey urgency and come from addresses or phone numbers similar to those their targets interact with frequently.

Hackers will take other measures to make it seem like they represent a trusted company, like including links to pages that mimic sites the victim would recognize.

But phishing only works if customers provide the information attackers are requesting.

Informing customers that you will never email or text to ask for personal information will help them stay vigilant.

Malware and ransomware

Malware — short for “malicious software” — is specifically designed to disrupt, damage, or gain unauthorized access to a computer system.

For example, ransomware is a type of malware that encrypts a victim's files until they pay the attacker to release them.

Malware has the potential to cause major inconveniences for you, your employees, and your customers.

Attacks can slow your business to a halt and lock you out of critical systems. And removing malware is expensive.

Preventive measures like installing antivirus and antispyware software, keeping your systems updated, and using secure authentication can thwart these malware attacks.

SQL injection

While storing data in a structured query language (SQL) server database is a fairly normal practice, it's not automatically secure. SQL servers store data in a series of tables that can be retrieved by applications using requests or "queries."

If these servers are unprotected, attackers can write and inject their own queries, giving them access to view or change any information in an SQL database.

Providing security training to your developers, treating any edits to data tables as untrusted, and adopting modern web development technologies are good first steps to SQL injection (SQLi) prevention.

Cross-site scripting (XSS)

Cross-site scripting (XSS) occurs when an attacker inserts a piece of malicious code into a web page. Although XSS doesn't impact the entire site, it exposes customers on that page to cyberattacks like phishing and malware.

Scanning regularly for vulnerabilities in your website's code or API integrations and patching them quickly can help hinder XSS attacks.

Brute force attacks

Brute force attacks are an attempt to gain access to your site by targeting an online store's administrator console and trying to figure out its password by "brute force."

Once an attacker establishes a connection to your site, they'll run automated programs called scripts to guess every possible combination of letters, numbers, and characters that could make up your password.

You can protect your ecommerce site by choosing a complex, strong password for your admin panel and changing it regularly.

Ask your customers to do the same for any loyalty accounts they create.

E-skimming

E-skimming is a method of stealing credit card information and personal data from payment processors on ecommerce sites.

In this attack, hackers gain access to checkout pages and capture payment information as customers type it in real time.

E-skimming can result from XSS, phishing, or brute force attacks.

To help prevent e-skimming you need to regularly push patches to your web server, vet your ad server code, and keep third-party APIs updated.

If your site has already been impacted by e-skimming, see if your cyber insurance covers any losses and shut down your shopping cart page to investigate and eliminate the source.

Spam

Spam is an irrelevant message that prompts users to click on malicious links.

Spammers often use email to spread these links, but they may also leave infected links in comments on a blog, social media post, or contact forms.

Spam impacts a website's security and slows browsing speeds.

Deleting unwanted comments and enabling reCAPTCHA on forms can help thwart spam attacks.

ReCAPTCHAs require users to enter a slightly distorted series of numbers and letters, which spam bots can't read.

Deleting any spam comments that do get through and performing a root cause analysis to see where they came from cannot only keep your form response reports clean, they can also help you determine a solution.

Bots

Bots are designed to scrape websites for pricing and inventory.

Hackers then gain access to the site and use this information to hike prices or add the most popular inventory to their shopping carts.

When customers can't buy what they want or need, sales decline and stores may experience negative reviews or bad press.

Putting reCAPTCHA tools on your site, checking your API connections, and blocking old browser versions are good ways to combat bots.

You can also set up alerts for unusually high web traffic, failed gift card validations, and failed login attempts, as these can be signs of bots trying to gain access to your site.

Trojan horses

Trojan horses are a type of malware disguised as useful programs.

Because Trojan horses seem benign, team members or customers may download them onto their computers, at which point malware code is activated and attackers can steal personal information.

Robust antivirus software and firewalls offer some protection, but you also need to remind staff members and customers to be wary of email attachments and to avoid unapproved third-party downloads.

Best practices for ecommerce security

Hackers are always inventing new strategies for stealing data.

In addition to protecting against known threats, there are some general best practices for ecommerce security.

1. Use multilayer security

Multilayer security is the practice of adding secondary or tertiary layers of security controls throughout a technology system.

If one layer is compromised, attackers have to penetrate at least one other layer to get the information they are seeking.

Multiple security layers adds more obstacles attackers have to break through to infiltrate your site.

One important layer is a **content delivery network (CDN)**.

The best CDNs use machine learning to block threats and infectious traffic.

Another layer could be multifactor authentication for employees logging in to company systems and for customers logging in to their loyalty accounts.

When they enter their information, they'll need to enter another code sent to them via text, email, or authenticator app.

2. Secure your website with SSL certificates

Secure sockets layer (SSL) certificates verify a website's identity and serve as an encrypted connection.

SSL certificates protect credit card details and other potentially sensitive transactions that occur on your ecommerce website and prevent hackers from using your site as part of a phishing attack.

3. Use firewalls

Firewall software and plugins allow trusted traffic but keep untrusted connections off of an ecommerce site. Regulating traffic flow makes detecting any anomalies easier and stops them before they enter your network.

This makes firewalls especially useful for protecting against cyber threats like XSS, spam, and malicious SQL injections.

4. Install antivirus and antimalware software

Attackers often use stolen credit card information to place orders, which puts your store at risk of enabling fraudulent activity.

Antivirus and antimalware software uses sophisticated algorithms to flag malicious transactions and provide fraud risk scores to determine whether transactions are legitimate.

Regularly scanning your site can greatly reduce malware attacks.

5. Train your staff

All employees should be aware of regulations that protect customer information.

Enforcing password updates, limiting access to sensitive information, and requiring employee cybersecurity and privacy training are all steps you can take to decrease your liability.

And remember to revoke access to all systems when employees leave, so they can't sell data to cyberattackers or commit cybercrimes themselves.

6. Educate your clients

Some lapses in security happen as a result of customer behavior.

Customers have logins to many sites and sometimes reuse the same password over and over.

Requiring long, complex passwords and reminding customers about the risks of phishing attacks decreases the potential for cyberattacks.

IoT security | IoT device security

IoT security is the practice of protecting Internet of Things (IoT) devices from attack.

What is IoT security? | IoT device security

Internet of Things (IoT) devices are computerized Internet-connected objects, such as networked security cameras, smart refrigerators, and WiFi-capable automobiles.

IoT security is the process of securing these devices and ensuring they do not introduce threats into a network.

Anything connected to the Internet is likely to face attack at some point. Attackers can try to remotely compromise IoT devices using a variety of methods, from credential theft to vulnerability exploits.

Once they control an IoT device, they can use it to steal data, conduct distributed denial-of-service (DDoS) attacks , or attempt to compromise the rest of the connected network.

IoT security can be particularly challenging because many IoT devices are not built with strong security in place — typically, the manufacturer's focus is on features and usability, rather than security, so that the devices can get to market quickly.

IoT devices are increasingly part of everyday life, and both consumers and businesses may face IoT security challenges.

What attacks are IoT devices most susceptible to?

Firmware vulnerability exploits

All computerized devices have firmware, which is the software that operates the hardware.

In computers and smart phones, operating systems run on top of the firmware; for the majority of IoT devices, the firmware is essentially the operating system.

Most IoT firmware does not have as many security protections in place as the sophisticated operating systems running on computers.

And often this firmware is rife with known vulnerabilities that in some cases cannot be patched.

This leaves IoT devices open to attacks that target these vulnerabilities.

Credential-based attacks

Many IoT devices come with default administrator usernames and passwords. These usernames and passwords are often not very secure — for instance, "password" as the password — and worse, sometimes all IoT devices of a given model share these same credentials.

In some cases, these credentials cannot be reset.

Attackers are well aware of these default usernames and passwords, and many successful IoT device attacks occur simply because an attacker guesses the right credentials.

On-path attacks

On-path attackers position themselves between two parties that trust each other — for example, an IoT security camera and the camera's cloud server — and intercept communications between the two.

IoT devices are particularly vulnerable to such attacks because many of them do not encrypt their communications by default (encryption scrambles data so that it cannot be interpreted by unauthorized parties).

Physical hardware-based attacks

Many IoT devices, like IoT security cameras, stoplights, and fire alarms, are placed in more or less permanent positions in public areas.

If an attacker has physical access to an IoT device's hardware, they can steal its data or take over the device.

This approach would affect only one device at a time, but a physical attack could have a larger effect if the attacker gains information that enables them to compromise additional devices on the network.

How are IoT devices used in DDoS attacks?

Malicious parties often use unsecured IoT devices to generate network traffic in a DDoS attack.

DDoS attacks are more powerful when the attacking parties can send traffic to their target from a wide range of devices.

Such attacks are harder to block because there are so many IP addresses involved (each device has its own IP address).

One of the biggest DDoS botnets on record, the Mirai botnet , is largely made up of IoT devices.

What are some of the main aspects of IoT device security?

Software and firmware updates

IoT devices need to be updated whenever the manufacturer issues a vulnerability patch or software update.

These updates eliminate vulnerabilities that attackers could exploit.

Not having the latest software can make a device more vulnerable to attack, even if it is outdated by only a few days.

In many cases IoT firmware updates are controlled by the manufacturer, not the device owner, and it is the manufacturer's responsibility to ensure vulnerabilities are patched.

Credential security

IoT device admin credentials should be updated if possible.

It is best to avoid reusing credentials across multiple devices and applications — each device should have a unique password.

This helps prevent credential-based attacks.

Device authentication

IoT devices connect to each other, to servers, and to various other networked devices.

Every connected device needs to be authenticated to ensure they do not accept inputs or requests from unauthorized parties.

For example, an attacker could pretend to be an IoT device and request confidential data from a server, but if the server first requires them to present an authentic TLS certificate, then this attack will not be successful.

For the most part, this type of authentication needs to be configured by the device manufacturer.

Encryption

IoT device data exchanges are vulnerable to external parties and on-path attackers as they pass over the network — unless encryption is used to protect the data.

Think of encryption as being like an envelope that protects a letter's contents as it travels through the postal service.

Encryption must be combined with authentication to fully prevent on-path attacks.

Otherwise, the attacker could set up separate encrypted connections between one IoT device and another, and neither would be aware that their communications are being intercepted.

Turning off unneeded features

Most IoT devices come with multiple features, some of which may go unused by the owner.

But even when features are not used, they may keep additional ports open on the device in case of use.

The more ports an Internet-connected device leaves open, the greater the attack surface —often attackers simply ping different ports on a device, looking for an opening.

Turning off unnecessary device features will close these extra ports.

DNS filtering

DNS filtering is the process of using the Domain Name System to block malicious websites.

Adding DNS filtering as a security measure to a network with IoT devices prevents those devices from reaching out to places on the Internet they should not (i.e. an attacker's domain).

What is mutual TLS (mTLS)?

Mutual Transport Layer Security (mTLS) is a type of mutual authentication, which is when both sides of a network connection authenticate each other.

TLS is a protocol for verifying the server in a client-server connection; mTLS verifies both connected devices, instead of just one.

mTLS is important for IoT security because it ensures only legitimate devices and servers can send commands or request data.

It also encrypts all communications over the network so that attackers cannot intercept them.

mTLS requires issuing TLS certificates to all authenticated devices and servers.

A TLS certificate contains the device's public key and information about who issued the certificate.

Showing a TLS certificate to initiate a network connection can be compared to a person showing their ID card to prove their identity.

Operating System Security

Every computer system and software design must handle all security risks and implement the necessary measures to enforce security policies.

At the same time, it's critical to strike a balance because strong security measures might increase costs while also limiting the system's usability, utility, and smooth operation.

As a result, system designers must assure efficient performance without compromising security.

What is Operating System Security?

The process of ensuring OS availability, confidentiality, integrity is known as operating system security.

OS security refers to the processes or measures taken to protect the operating system from dangers, including viruses, worms, malware, and remote hacker intrusions.

Operating system security comprises all preventive-control procedures that protect any system assets that could be stolen, modified, or deleted if OS security is breached.

Security refers to providing safety for computer system resources like software, CPU, memory, disks, etc. It can protect against all threats, including viruses and unauthorized access. It can be enforced by assuring the operating system's integrity, confidentiality, and availability.

If an illegal user runs a computer application, the computer or data stored may be seriously damaged.

System security may be threatened through two violations, and these are as follows:

1. Threat

A program that has the potential to harm the system seriously.

2. Attack

A breach of security that allows unauthorized access to a resource.

There are two types of security breaches that can harm the system: malicious and accidental.

Malicious threats are a type of destructive computer code or web script that is designed to cause system vulnerabilities that lead to back doors and security breaches.

On the other hand, Accidental Threats are comparatively easier to protect against.

Security may be compromised through the breaches. Some of the breaches are as follows:

1. Breach of integrity

This violation has unauthorized data modification.

2. Theft of service

It involves the unauthorized use of resources.

3. Breach of confidentiality

It involves the unauthorized reading of data.

4. Breach of availability

It involves the unauthorized destruction of data.

5. Denial of service

It includes preventing legitimate use of the system. Some attacks may be accidental.

The goal of Security System

There are several goals of system security. Some of them are as follows:

1. Integrity

Unauthorized users must not be allowed to access the system's objects, and users with insufficient rights should not modify the system's critical files and resources.

2. Secrecy

The system's objects must only be available to a small number of authorized users.

The system files should not be accessible to everyone.

3. Availability

All system resources must be accessible to all authorized users, i.e., no single user/process should be able to consume all system resources.

If such a situation arises, service denial may occur. In this case, malware may restrict system resources and preventing legitimate processes from accessing them.

Types of Threats

There are mainly two types of threats that occur. These are as follows:

Program threats

The operating system's processes and kernel carry out the specified task as directed.

Program Threats occur when a user program causes these processes to do malicious operations.

The common example of a program threat is that when a program is installed on a computer, it could store and transfer user credentials to a hacker.

There are various program threats. Some of them are as follows:

1.Virus

A virus may replicate itself on the system.

Viruses are extremely dangerous and can modify/delete user files as well as crash computers.

A virus is a little piece of code that is implemented on the system program.

As the user interacts with the program, the virus becomes embedded in other files and programs, potentially rendering the system inoperable.

2. Trojan Horse

This type of application captures user login credentials.

It stores them to transfer them to a malicious user who can then log in to the computer and access system resources.

3. Logic Bomb

A logic bomb is a situation in which software only misbehaves when particular criteria are met; otherwise, it functions normally.

4. Trap Door

A trap door is when a program that is supposed to work as expected has a security weakness in its code that allows it to do illegal actions without the user's knowledge.

System Threats

System threats are described as the misuse of system services and network connections to cause user problems.

These threats may be used to trigger the program threats over an entire network, known as program attacks.

System threats make an environment in which OS resources and user files may be misused.

There are various system threats. Some of them are as follows:

1. Port Scanning

It is a method by which the cracker determines the system's vulnerabilities for an attack.

It is a fully automated process that includes connecting to a specific port via TCP/IP.

To protect the attacker's identity, port scanning attacks are launched through Zombie Systems, which previously independent systems now serve their owners while being utilized for such terrible purposes.

2. Worm

The worm is a process that can choke a system's performance by exhausting all system resources.

A Worm process makes several clones, each consuming system resources and preventing all other processes from getting essential resources. Worm processes can even bring a network to a halt.

3. Denial of Service

Denial of service attacks usually prevents users from legitimately using the system.

For example, if a denial-of-service attack is executed against the browser's content settings, a user may be unable to access the internet.

Threats to Operating System

There are various threats to the operating system.

Some of them are as follows:

Malware

It contains viruses, worms, trojan horses, and other dangerous software.

These are generally short code snippets that may corrupt files, delete the data, replicate to propagate further, and even crash a system.

The malware frequently goes unnoticed by the victim user while criminals silently extract important data.

Network Intrusion

Network intruders are classified as masqueraders, misfeasors, and unauthorized users.

A **masquerader** is an unauthorized person who gains access to a system and uses an authorized person's account.

A **misfeisor** is a legitimate user who gains unauthorized access to and misuses programs, data, or resources.

A **rogue** user takes supervisory authority and tries to evade access constraints and audit collection.

Buffer Overflow

It is also known as buffer overrun.

It is the most common and dangerous security issue of the operating system.

It is defined as a condition at an interface under which more input may be placed into a buffer and a data holding area than the allotted capacity, and it may overwrite other information.

Attackers use such a situation to crash a system or insert specially created malware that allows them to take control of the system.

How to ensure Operating System Security?

There are various ways to ensure operating system security. These are as follows:

Authentication

The process of identifying every system user and associating the programs executing with those users is known as authentication.

The operating system is responsible for implementing a security system that ensures the authenticity of a user who is executing a specific program.

In general, operating systems identify and authenticate users in three ways.

1. Username/Password

Every user contains a unique username and password that should be input correctly before accessing a system.

2. User Attribution

These techniques usually include biometric verification, such as fingerprints, retina scans, etc. This authentication is based on user uniqueness and is compared to database samples already in the system. Users can only allow access if there is a match.

3. User card and Key

To login into the system, the user must punch a card into a card slot or enter a key produced by a key generator into an option provided by the operating system.

One Time passwords

Along with standard authentication, one-time passwords give an extra layer of security.

Every time a user attempts to log into the One-Time Password system, a unique password is needed.

Once a one-time password has been used, it cannot be reused. One-time passwords may be implemented in several ways.

1. Secret Key

The user is given a hardware device that can generate a secret id that is linked to the user's id. The system prompts for such a secret id, which must be generated each time you log in.

2. Random numbers

Users are given cards that have alphabets and numbers printed on them. The system requests numbers that correspond to a few alphabets chosen at random.

3. Network password

Some commercial applications issue one-time passwords to registered mobile/email addresses, which must be input before logging in.

Firewalls

Firewalls are essential for monitoring all incoming and outgoing traffic.

It imposes local security, defining the traffic that may travel through it.

Firewalls are an efficient way of protecting network systems or local systems from any network-based security threat.

Physical Security

The most important method of maintaining operating system security is physical security.

An attacker with physical access to a system may edit, remove, or steal important files since operating system code and configuration files are stored on the hard drive.

Operating System Security Policies and Procedures

Various operating system security policies may be implemented based on the organization that you are working in.

In general, an OS security policy is a document that specifies the procedures for ensuring that the operating system maintains a specific level of integrity, confidentiality, and availability.

OS Security protects systems and data from worms, malware, threats, ransomware, back door intrusions, viruses, etc.

Security policies handle all preventative activities and procedures to ensure an operating system's protection, including steal, edited, and deleted data.

As OS security policies and procedures cover a large area, there are various techniques to addressing them.

Some of them are as follows:

1. Installing and updating anti-virus software
2. Ensure the systems are patched or updated regularly
3. Implementing user management policies to protect user accounts and privileges.
4. Installing a firewall and ensuring that it is properly set to monitor all incoming and outgoing traffic.

OS security policies and procedures are developed and implemented to ensure that you must first determine which assets, systems, hardware, and data are the most vital to your organization.

Once that is completed, a policy can be developed to secure and safeguard them properly