Access Control Concepts

Access control identifies users by verifying various login credentials, which can include usernames and passwords, PINs, biometric scans, and security tokens. Many access control systems also include multifactor authentication (MFA), a method that requires multiple authentication methods to verify a user's identity.

There are four main types of access control. Organizations typically choose the method that makes the most sense based on their unique security and compliance requirements. The four access control models are:

Discretionary access control (DAC): In this method, the owner or administrator of the protected system, data, or resource sets the policies for who is allowed access.

Mandatory access control (MAC): In this nondiscretionary model, people are granted access based on an information clearance. A central authority regulates access rights based on different security levels. This model is common in government and military environments.

Role-based access control (RBAC): RBAC grants access based on defined business functions rather than the individual user's identity. The goal is to provide users with access only to data that's been deemed necessary for their roles within the organization. This widely used method is based on a complex combination of role assignments, authorizations, and permissions.

Attribute-based access control (ABAC): In this dynamic method, access is based on a set of attributes and environmental conditions, such as time of day and location, assigned to both users and resources.

UNIX & Windows Access Control Summary

UNIX -- Access Control

- Every user has a (typically unique) identifier
- Each user belongs to one or more groups
- Each file belongs to a user and a group
- Each file or directory has a set of permissions associated with it
- Permissions are: write access, read access, execute permission
- A different permission set is specified for the file's user, group, and all other users.
- On directories execute permission implies ability to traverse
- Executable files can be specified to run under the permission of their owner or group (rather than the user executing them).
- A sepcial user, the super user (named *root* overrides all access permissions

Windows NT -- Access Control

Windows NT supports multiple file systems, but the protection issues we will consider are only associated with one: NTFS. In NT there is the notion of an item, which can be a file or a directory. Each item has an owner. An owner is usually the thing that created the item. It can change the access control list, allow other accounts to change the access control list and allow other accounts to become owner. Entries in the ACL are individuals and groups. Note that NT was designed for groups of machines on a network, thus, a distinction is made between local groups (defined on a particular workstation) and global groups (domain wide). A single name can therefore mean multiple things.

NTFS is structured so that a file is a set of properties, the contents of the file being just one of those properties. An ACL is a property of an item. The ACL itself is a list of entries: (user or group, permissions). NTFS permissions are closer to extended permissions in UNIX than to the 9 mode bits. The permission offer a rich set of possibilities:

- R -- read
- W -- write
- X -- execute
- D -- delete
- P -- modify the ACL
- O -- make current account the new owner ("take ownership")

The owner is allowed to change the ACL. A user with permission P can also change the ACL. A user with permission O can take ownership. There is also a packaging of privileges known as permissions sets:

- no access
- read -- RX
- change -- RWXO
- full control -- RWXDPO

Issues in Access Control

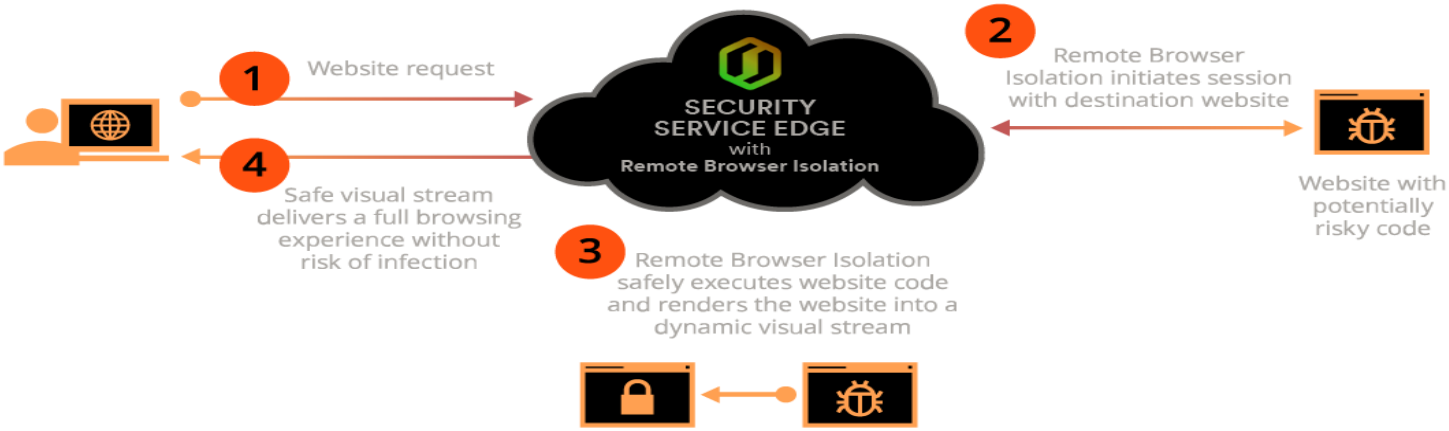
Problems with Access Control Systems

- Outdated equipment.
- Keycards falling into the wrong hands.
- Incorrect setup.
- Lack of integration with other building systems.

Browser Isolation

Browser Isolation (also known as Web Isolation) is a technology that contains web browsing activity inside an isolated environment, like a sandbox or virtual machine, in order to protect computers from any malware the user may encounter. This isolation may occur locally on the computer or remotely on a server. Browser Isolation technology provides malware protection for day-to-day browsing by eliminating the opportunity for malware to access the end user's device.

Browser Isolation essentially secures a computer/network from web-based threats by executing all browsing activity in an isolated virtual environment. Possible threats are contained in this environment and can't infiltrate any part of the user's ecosystem, such as their computer's hard-drive, or other devices on the network. Even though Browser Isolation is gaining traction as an IT security solution, a lot of misinformation regarding Browser Isolation remains.



Web Security Definition: web security refers to the protective measures and protocols that organizations adopt to protect the organization from cyber criminals and threats that use the web channel. Web security is critical to business continuity and to protecting data, users and companies from risk.

Web Security Goal: The objective of Cybersecurity is to protect information from being stolen, compromised or attacked. Cybersecurity can be measured by at least one of three goals-

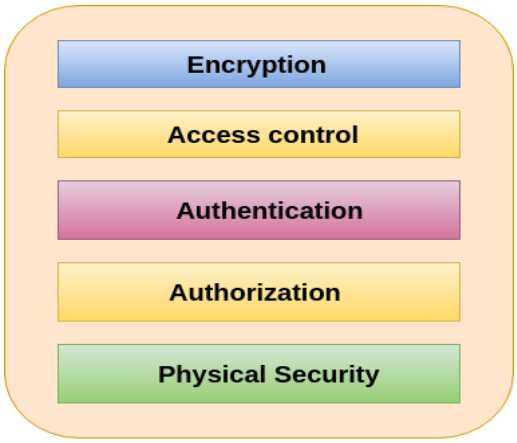
1. Protect the confidentiality of data.
2. Preserve the integrity of data.
3. Promote the availability of data for authorized users.

These goals form the confidentiality, integrity, availability (CIA) triad, the basis of all security programs. The CIA triad is a security model that is designed to guide policies for information security within the premises of an organization or company. This model is also referred to as the AIC (Availability, Integrity, and Confidentiality) triad to avoid the confusion with the Central Intelligence Agency. The elements of the triad are considered the three most crucial components of security.

1. Confidentiality

Confidentiality is roughly equivalent to privacy and avoids the unauthorized disclosure of information. It involves the protection of data, providing access for those who are allowed to see it while disallowing others from learning anything about its content. It prevents essential information from reaching the wrong people while making sure that the right people can get it. Data encryption is a good example to ensure confidentiality.

Tools for Confidentiality



Confidentiality Tools

Encryption

Encryption is a method of transforming information to make it unreadable for unauthorized users by using an algorithm. The transformation of data uses a secret key (an encryption key) so that the transformed data can only be read by using another secret key (decryption key). It protects sensitive data such as credit card numbers by encoding and transforming data into unreadable cipher text. This encrypted data can only be read by decrypting it. Asymmetric-key and symmetric-key are the two primary types of encryption.

Access control

Access control defines rules and policies for limiting access to a system or to physical or virtual resources. It is a process by which users are granted access and certain privileges to systems, resources or information. In access control systems, users need to present credentials before they can be granted access such as a person's name or a computer's serial number. In physical systems, these credentials may come in many forms, but credentials that can't be transferred provide the most security.

Authentication

An authentication is a process that ensures and confirms a user's identity or role that someone has. It can be done in a number of different ways, but it is usually based on a combination of-

- something the person has (like a smart card or a radio key for storing secret keys),
- something the person knows (like a password),
- something the person is (like a human with a fingerprint).

Authentication is the necessity of every organizations because it enables organizations to keep their networks secure by permitting only authenticated users to access its protected resources. These resources may include computer systems, networks, databases, websites and other network-based applications or services.

Authorization

Authorization is a security mechanism which gives permission to do or have something. It is used to determine a person or system is allowed access to resources, based on an access control policy, including computer programs, files, services, data and application features. It is normally preceded by authentication for user identity verification. System administrators are typically assigned permission levels covering all system and

user resources. During authorization, a system verifies an authenticated user's access rules and either grants or refuses resource access.

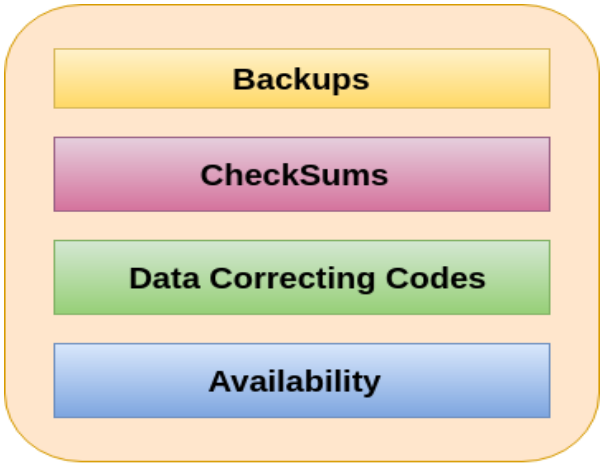
Physical Security

Physical security describes measures designed to deny the unauthorized access of IT assets like facilities, equipment, personnel, resources and other properties from damage. It protects these assets from physical threats including theft, vandalism, fire and natural disasters.

2. Integrity

Integrity refers to the methods for ensuring that data is real, accurate and safeguarded from unauthorized user modification. It is the property that information has not be altered in an unauthorized way, and that source of the information is genuine.

Tools for Integrity



Integrity Tools

Backups

Backup is the periodic archiving of data. It is a process of making copies of data or data files to use in the event when the original data or data files are lost or destroyed. It is also used to make copies for historical purposes, such as for longitudinal studies, statistics or for historical records or to meet the requirements of a data retention policy. Many applications especially in a Windows environment, produce backup files using the .BAK file extension.

Checksums

A checksum is a numerical value used to verify the integrity of a file or a data transfer. In other words, it is the computation of a function that maps the contents of a file to a numerical value. They are typically used to compare two sets of data to make sure that they are the same. A checksum function depends on the entire contents of a file. It is designed in a way that even a small change to the input file (such as flipping a single bit) likely to results in different output value.

Data Correcting Codes

It is a method for storing data in such a way that small changes can be easily detected and automatically corrected.

3. Availability

Availability is the property in which information is accessible and modifiable in a timely fashion by those authorized to do so. It is the guarantee of reliable and constant access to our sensitive data by authorized people.

Tools for Availability

- Physical Protections
- Computational Redundancies

Physical Protections

Physical safeguard means to keep information available even in the event of physical challenges. It ensure sensitive information and critical information technology are housed in secure areas.

Computational redundancies

It is applied as fault tolerant against accidental faults. It protects computers and storage devices that serve as fallbacks in the case of failures.

What is Threat Modeling?

Threat modeling is a method of optimizing network security by locating vulnerabilities, identifying objectives, and developing countermeasures to either prevent or mitigate the effects of cyber-attacks against the system.

The Threat Modeling Process

Threat modeling consists of defining an enterprise's assets, identifying what function each application serves in the grand scheme, and assembling a security profile for each application. The process continues with identifying and prioritizing potential threats, then documenting both the harmful events and what actions to take to resolve them.

Or, in lay terms, threat modeling is the act of assessing your organization's digital and network assets, identifying weak spots, determining what threats exist, and coming up with plans to protect or recover.

Ten Threat Modeling Methodologies

There are as many ways to fight cybercrime as there are types of cyber-attacks. For instance, here are ten popular threat modeling methodologies used today.

1. STRIDE

A methodology developed by Microsoft for threat modeling, it offers a mnemonic for identifying security threats in six categories:

- Spoofing: An intruder posing as another user, component, or other system feature that contains an identity in the modeled system.
- Tampering: The altering of data within a system to achieve a malicious goal.
- Repudiation: The ability of an intruder to deny that they performed some malicious activity, due to the absence of enough proof.
- Information Disclosure: Exposing protected data to a user that isn't authorized to see it.
- Denial of Service: An adversary uses illegitimate means to exhaust services needed to provide service to users.
- Elevation of Privilege: Allowing an intruder to execute commands and functions that they aren't allowed to.

2. DREAD

Proposed for threat modeling, but Microsoft dropped it in 2008 due to inconsistent ratings. OpenStack and many other organizations currently use DREAD. It's essentially a way to rank and assess security risks in five categories:

- Damage Potential: Ranks the extent of damage resulting from an exploited weakness.
- Reproducibility: Ranks the ease of reproducing an attack
- Exploitability: Assigns a numerical rating to the effort needed to launch the attack.
- Affected Users: A value representing how many users get impacted if an exploit becomes widely available.
- Discoverability: Measures how easy it is to discover the threat.

3. P.A.S.T.A

This stands for Process for Attack Simulation and Threat Analysis, a seven-step, risk-centric methodology. It offers a dynamic threat identification, enumeration, and scoring process. Once experts create a detailed analysis of identified threats, developers can develop an asset-centric mitigation strategy by analyzing the application through an attacker-centric view.

4. Trike

Trike focuses on using threat models as a risk management tool. Threat models, based on requirement models, establish the stakeholder-defined "acceptable" level of risk assigned to each asset class. Requirements model analysis yields a threat model where threats are identified and given risk values. The completed threat model is then used to build a risk model, factoring in actions, assets, roles, and calculated risk exposure.

5. VAST

Standing for Visual, Agile, and Simple Threat modeling, it provides actionable outputs for the specific needs of various stakeholders such as application architects and developers, cybersecurity personnel, etc. VAST offers a unique application and infrastructure visualization plan so that the creation and use of threat models don't require any specialized expertise in security subject matters.

6. Attack Tree

The tree is a conceptual diagram showing how an asset, or target, could be attacked, consisting of a root node, with leaves and children nodes added in. Child nodes are conditions that must be met to make the direct parent node true. Each node is satisfied only by its direct child nodes. It also has "AND" and "OR" options, which represent alternative steps taken to achieve these goals.

7. Common Vulnerability Scoring System (CVSS)

This method provides a way to capture a vulnerability's principal characteristics and assigning a numerical score (ranging from 0-10, with 10 being the worst) showing its severity. The score is then translated into a qualitative representation (e.g., Low, Medium, High, and Critical). This representation helps organizations effectively assess and prioritize their unique vulnerability management processes.

8. T-MAP

T-MAP is an approach commonly used in Commercial Off the Shelf (COTS) systems to calculate attack path weights. The model incorporates UML class diagrams, including access class, vulnerability, target assets, and affected value.

9. OCTAVE

The Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) process is a risk-based strategic assessment and planning method. OCTAVE focuses on assessing organizational risks only and does not address technological risks. OCTAVE has three phases:

- Building asset-based threat profiles. (Organizational evaluation)
- Identifying infrastructure vulnerabilities. (Information infrastructure evaluation)

- Developing and planning a security strategy. (Evaluation of risks to the company's critical assets and decision making.)

10. Quantitative Threat Modeling Method

This hybrid method combines attack trees, STRIDE, and CVSS methods. It addresses several pressing issues with threat modeling for cyber-physical systems that contain complex interdependencies in their components. The first step is building components attack trees for the STRIDE categories. These trees illustrate the dependencies in the attack categories and low-level component attributes. Then the CVSS method is applied, calculating the scores for all the tree's components.

HTTP Content Rendering

What is rendering?

Rendering is a process used in web development that turns website code into the interactive pages users see when they visit a website. The term generally refers to the use of HTML, CSS, and JavaScript codes. The process is completed by a rendering engine, the software used by a web browser to render a web page. Because of its close association with web browsers, rendering engines are commonly referred to as browser engines.

Html Content Rendering Engine or browser rendering engine

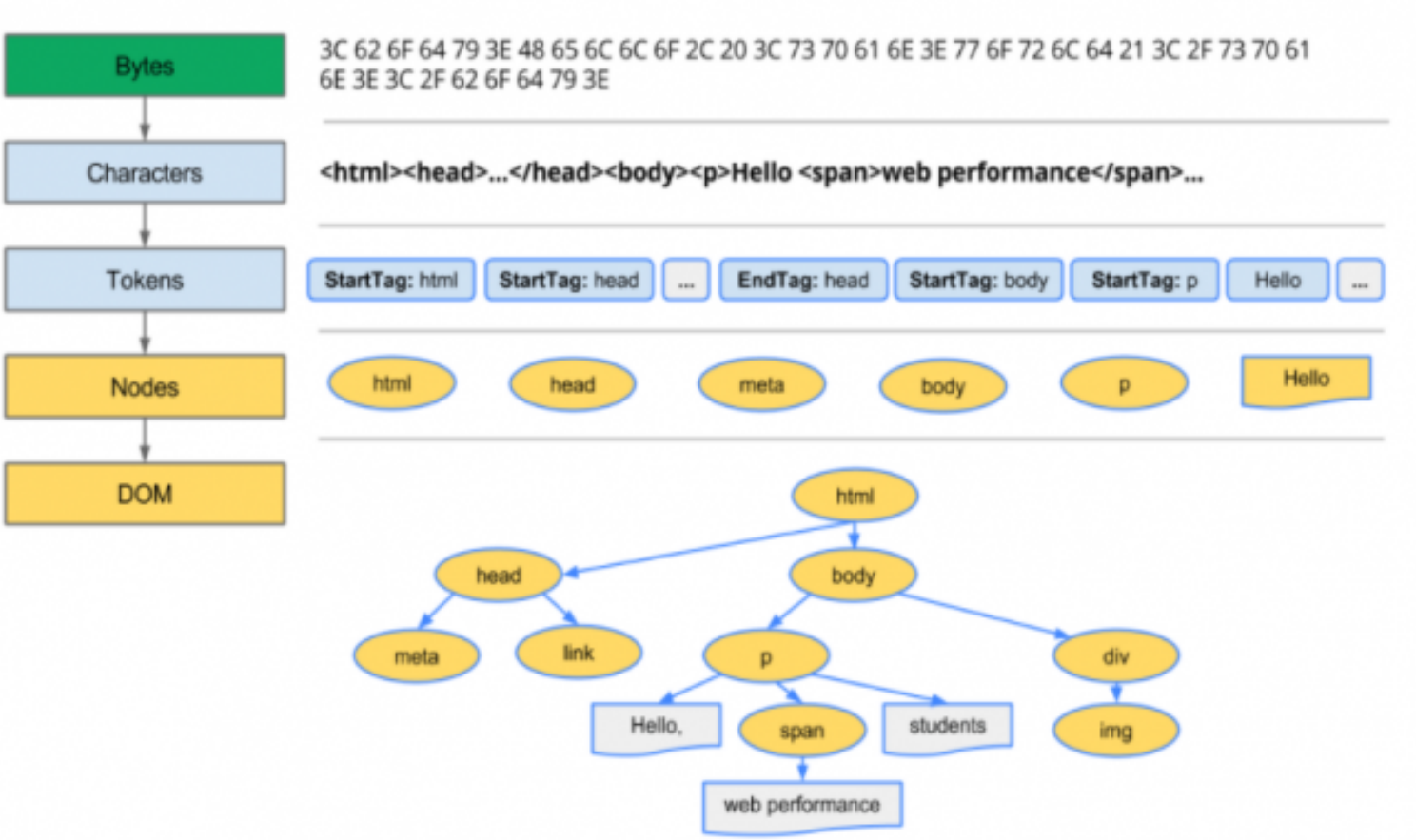
Bowser rendering engine, software that renders HTML pages (Web pages). It turns the HTML layout tags in the page into the appropriate commands for the operating system, which causes the formation of the text characters and images for screen and printer.

How browsers render web pages

Web browsers render web pages in the following sequence:

Constructs DOM and CSSOM from raw code

- While loading a web page, a web server sends a folder of files containing HTML, CSS, and JavaScript code to a user’s web browser.
- The browser engine converts this data (bytes) into characters (the HTML code).
- It parses the characters into tokens, which are further parsed into nodes.
- The browser engine links the nodes into a tree-like structure known as a Document Object Model (DOM). The DOM is the JavaScript representation of the HTML.
- Simultaneously, the browser converts the CSS code to a CSS Object Model (CSSOM) through a similar process.



Security Interface: The Security Interface framework is a set of Objective-C classes that provide user interface elements for programs that implement security features such as authorization, access to digital certificates, and access to items in key chains.

Cookies Frames and frame busting

What are cookies?

Cookies are text files that a website places on your computer, phone or any other device, with information about your navigation on that site. Cookies are necessary to facilitate navigation and improve your browsing experience. For example, to remember your preferences (language, country, etc.) during navigation and in future visits.

Types of Cookies

There are three different types of cookies −

- **Session Cookies** − These are mainly used by online shops and allows you to keep items in your basket when shopping online. These cookies expire after a specific time or when the browser is closed.
- **Permanent Cookies** − These remain in operation, even when you have closed the browser. They remember your login details and password so you don't have to type them in every time you use the site. It is recommended that you delete these type of cookies after a specific time.
- **Third-Party Cookies** − These are installed by third parties for collecting certain information. For example: Google Maps.

Frame Bursting:

It is a study of Clickjacking vulnerabilities at popular sites. Web framing attacks such as Clickjacking use iframes to hijack a user's web session. The most common defense, called frame busting, **prevents a site from functioning when loaded inside a frame**. Clickjacking allows an attacker to trick your users into clicking parts of your interface without their consent. A simple way to describe this is, an attacker will embed your application in their site as an iframe. On top of the iframe they can show a completely different interface. You are thinking you are clicking buttons on your own interface, while in fact you are hitting the "**Delete my account"** button in for example GMail.

Web Server Threats

Websites are hosted on web servers. Web servers are themselves computers running an operating system; connected to the back-end database, running various applications. Any vulnerability in the applications, Database, Operating system or in the network will lead to an attack on the web server.

Web Server Attacks types:

DOS attack:

An attacker may cause a denial of service attack by sending numerous service request packets overwhelming the servicing capability of the web server, or he may try to exploit a programming error in the application causing a DOS attack. E.g. buffer overflow attack, SYN flooding, HTTP get Request Flooding, Ping of death.

Website Defacement:

SQL injection attacks are used to deface the website. When an attacker finds out that input fields are not sanitized properly, he can add SQL strings to maliciously craft a query which is executed by the web browser. He may store malicious/unrelated data in the database; when the website is requested, it will show irrelevant data on the website, thus displaying a defaced website.

Directory Traversal:

This is vulnerability where an attacker is able to access beyond the web root directory from the application. If he is able to access beyond web root directory, he might execute OS commands and get sensitive information or access restricted directories.

Misconfiguration attacks:

If unnecessary services are enabled or default configuration files are used, verbose/error information is not masked; an attacker can compromise the web server through various attacks like password cracking, Error-based SQL injection, Command Injection, etc.

Phishing Attack:

An attacker may redirect the victim to malicious websites by sending him/her a malicious link by email which looks authentic, but redirects him/her to malicious web page thereby stealing their data.

Methodology:

Information Gathering:

Information related to the target server is collected from various sources like

From websites

WHOIS information

Netcraft information

Banner grabbing

Port scanning with Nmap.

Mirroring a website using Htttrack.

Vulnerability Scanning:
There are automated tools for scanning a web server and applications running on it. The results may show various threats and vulnerabilities on the target web server; these vulnerabilities may later be exploited using tools or manually. E.g. Acunetix, Nikto, Vega etc

Password Attacks:

Guessing/Default passwords

Brute Forcing

Dictionary Attacks

Countermeasures:

Update and patch web servers regularly.
Do not use the default configuration.
Store configuration files securely.
Scan the applications running on the web server for all vulnerabilities.
Use IDS and firewall with updated signatures.
Block all unnecessary protocols and services.
Use secure protocols.
Disable default accounts, follow strict access control policy.
Install Anti-virus, and update it regularly.
All OS and software used should be latest and updated.

Cross-site request forgery (CSRF)

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing. If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application.

How does CSRF work?

For a CSRF attack to be possible, three key conditions must be in place:

- **A relevant action.** There is an action within the application that the attacker has a reason to induce. This might be a privileged action (such as modifying permissions for other users) or any action on user-specific data (such as changing the user's own password).
- **Cookie-based session handling.** Performing the action involves issuing one or more HTTP requests, and the application relies solely on session cookies to identify the user who has made the requests. There is no other mechanism in place for tracking sessions or validating user requests.
- **No unpredictable request parameters.** The requests that perform the action do not contain any parameters whose values the attacker cannot determine or guess. For example, when causing a user to change their password, the function is not vulnerable if an attacker needs to know the value of the existing password.

Cross Site Scripting (XSS)

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.

How does XSS work?

Cross-site scripting works by manipulating a vulnerable web site so that it returns malicious JavaScript to users. When the malicious code executes inside a victim's browser, the attacker can fully compromise their interaction with the application.

What are the types of XSS attacks?

There are three main types of XSS attacks. These are:

- Reflected XSS, where the malicious script comes from the current HTTP request.
- Stored XSS, where the malicious script comes from the website's database.
- DOM-based XSS, where the vulnerability exists in client-side code rather than server-side code.

Reflected cross-site scripting

Reflected XSS is the simplest variety of cross-site scripting. It arises when an application receives data in an HTTP request and includes that data within the immediate response in an unsafe way.

Stored cross-site scripting

Stored XSS (also known as persistent or second-order XSS) arises when an application receives data from an untrusted source and includes that data within its later HTTP responses in an unsafe way.

DOM-based cross-site scripting

DOM-based XSS (also known as DOM XSS) arises when an application contains some client-side JavaScript that processes data from an untrusted source in an unsafe way, usually by writing the data back to the DOM.

What can XSS be used for?

An attacker who exploits a cross-site scripting vulnerability is typically able to:

- Impersonate or masquerade as the victim user.
- Carry out any action that the user is able to perform.
- Read any data that the user is able to access.
- Capture the user's login credentials.
- Perform virtual defacement of the web site.
- Inject trojan functionality into the web site.

How to find and test for XSS vulnerabilities

Manually testing for reflected and stored XSS normally involves submitting some simple unique input (such as a short alphanumeric string) into every entry point in the application, identifying every location where the submitted input is returned in HTTP responses, and testing each location individually to determine whether suitably crafted input can be used to execute arbitrary JavaScript. In this way, you can determine the context in which the XSS occurs and select a suitable payload to exploit it.

How to prevent XSS attacks

In general, effectively preventing XSS vulnerabilities is likely to involve a combination of the following measures:

- **Filter input on arrival.** At the point where user input is received, filter as strictly as possible based on what is expected or valid input.
- **Encode data on output.** At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding.
- **Use appropriate response headers.** To prevent XSS in HTTP responses that aren't intended to contain any HTML or JavaScript, you can use the Content-Type and X-Content-Type-Options headers to ensure that browsers interpret the responses in the way you intend.
- **Content Security Policy.** As a last line of defense, you can use Content Security Policy (CSP) to reduce the severity of any XSS vulnerabilities that still occur.

Ways to find a vulnerability

Free vulnerability scanning: An application security scanner is a tool configured to query specific interfaces to detect security and performance gaps. These tools rely on documented tools and scripts to check for known weaknesses. Crashtest Security Suite is a highly popular and effective scanner that simplifies vulnerability scanning by helping organizations establish an end-to-end continuous testing process. Besides detecting and alerting system weaknesses, the online scanner helps developers establish a reliable, repeatable remediation process.

Penetration Testing: penetration testing is how cyber security professionals check for security gaps so they can be closed before a malicious attack occurs. The general steps of a penetration test usually involve:

1. Getting a "white hat" hacker to run the pen test at a set date/time.
2. Auditing existing systems to check for assets with known vulnerabilities.
3. The "hackers" running simulated attacks on the network that attempt to exploit potential weaknesses or uncover new ones.
4. The organization running its incident response plan (IRP) to try and contain the "attacks" simulated during penetration testing.

Creating a Threat Intelligence Framework: Once the penetration test report has been tabled, it is important to create a central repository for detecting, alerting, and managing security threats. A threat intelligence framework outlines a repeatable, scalable security incident management plan for all stakeholders involved in securing the website. A robust threat intelligence mechanism helps organizations lower expenses by speeding up the response to data breaches. The shared repository includes crucial information that can be used as a collaborative knowledge base for organization-wide security compliance.

Secure Development: 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.

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 webpages, in addition to webpages using other types of client-side technologies (e.g., Silverlight, Flash, Java applets), this type of feature is becoming more prevalent.