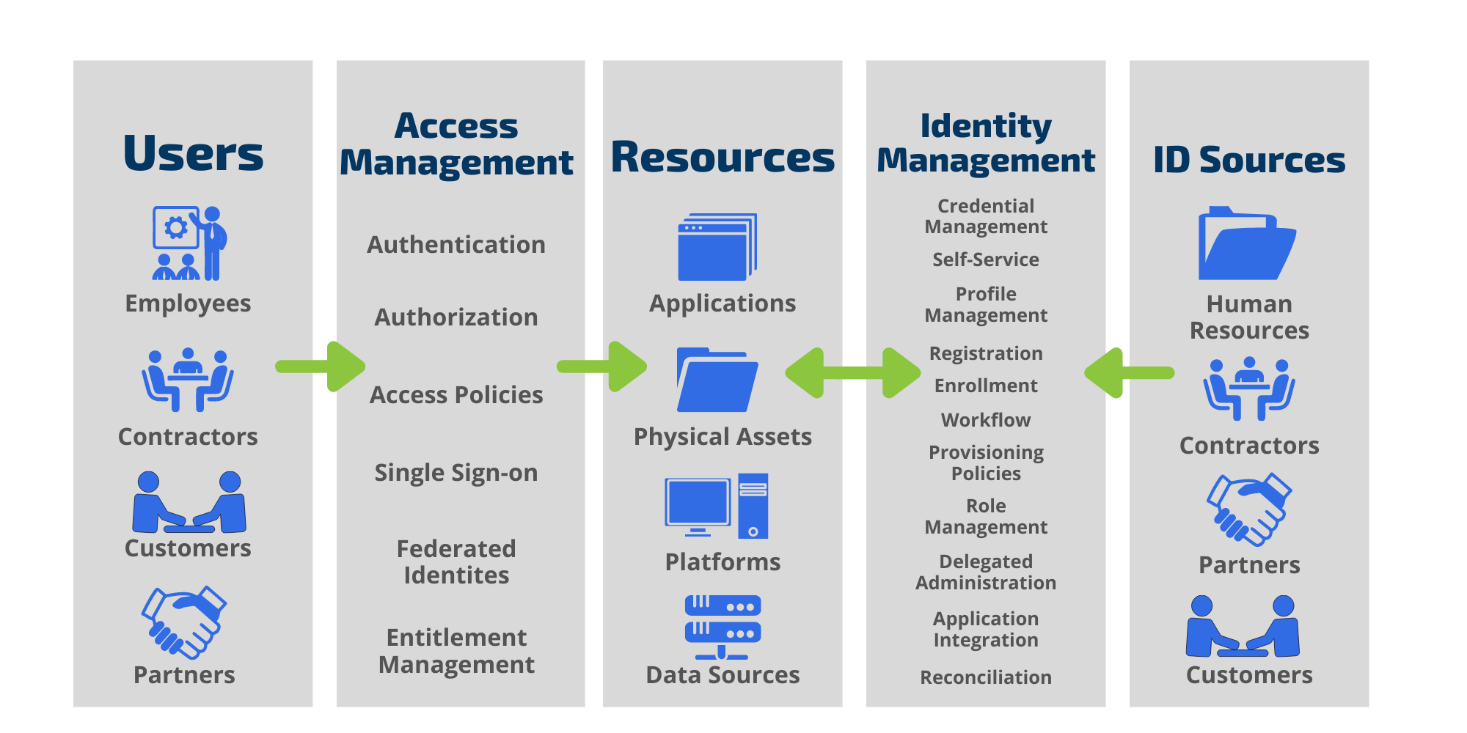
**Module – 4**

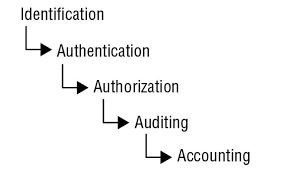
**Identity and Access Management**

Identity and access management (IAM) ensures that the right people and job roles in your organization (identities) can access the tools they need to do their jobs. Identity management and access systems enable your organization to manage employee apps without logging into each app as an administrator.

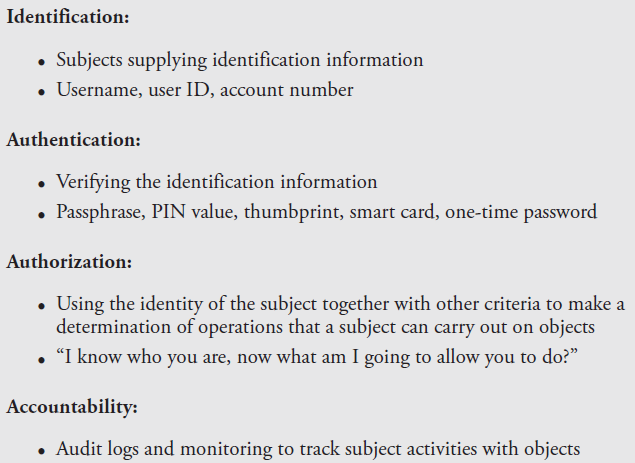




What is Identity & Access Management? - Tools4ever

****





## What are the 4 main access control models?

There are 4 types of access control models that you will commonly see across a variety of properties. Keep in mind that some models are exclusively used for [commercial properties](https://butterflymx.com/blog/commercial-access-control-system/).

**The 4 main access control models are:**

1. [Discretionary access control (DAC)](https://butterflymx.com/blog/access-control-models/#Discretionary)
2. [Mandatory access control (MAC)](https://butterflymx.com/blog/access-control-models/#Mandatory)
3. [Role-based access control (RBAC)](https://butterflymx.com/blog/access-control-models/#Role)
4. [Rule-based access control (RuBAC)](https://butterflymx.com/blog/access-control-models/#Rule)

### 1. Discretionary access control (DAC)

The discretionary access control model is one of least restrictive access control models. It allows for multiple administrators to control access to a property. This is especially convenient for [residential properties](https://butterflymx.com/blog/apartment-access-control/) or businesses with multiple managers.

**Pro:**

* This model is straightforward to use and makes it easy to assign access to users.

**Con:**

* This model can lead to confusion if the multiple administrators don’t communicate properly about who does and doesn’t have access.

2. Mandatory access control (MAC)

Mandatory access control stands as a complete alternative to discretionary access control.

This access control design is best used for businesses that emphasize security and confidentiality. As a result, this model features only one system administrator.

The system administrator cannot be overridden or bypassed, and they determine who is granted access to a property. Government facilities primarily use mandatory access control models.

Pro:

One system administrator in charge can lead to a more organized database of users with access to the property.

Con:

Having one person in charge can lead to a slower approval process when somebody new needs access.

### 3. Role-based access control (RBAC)

The [role-based](https://butterflymx.com/blog/rbac-vs-abac/)model is also known as **non-discretionary access control**. This model assigns every user a specific role that has **unique access permissions**.

**System administrators** have the ability to assign user roles and manage access for each role.

### 4. Rule-based access control (RuBAC)

Rule-based access control features an algorithm that changes a user’s access permissions based on a number of qualifying factors such as the time of day.

An example of rule-based access control is adjusting access permissions for an amenity such as a pool or gym that’s only open during daylight hours.

Another example is an office that’s only accessible to certain users during business hours. In this scenario, a manager with different permissions can still access the office when others can’t.

Another high-security use for this model is the ability to program a role-based access control system to lock down specific areas of a building if there’s a security compromise detected at a main entrance. Of course, the specifics of this feature vary from system to system.

**Pro:**

* A property can comply with **local laws** by restricting access to certain areas after hours (such as a pool or room with heavy machinery).

**Cons:**

* RuBAC does not provide access based on a user’s specific role, which makes it difficult for employees at a residential or commercial property to enter restricted areas after hours.
* This model can be difficult to set up and program depending on how many rooms require time-based access.

## Which is the best access control model?

While the most useful access control model depends on the type of property you oversee, a **role-based access control** system is likely your best choice.

Role-based access control systems are some of the most convenient for both property **managers and daily users.**

15. **Access Control Monitoring(IDS/IPS)**

#### **15.1. Intrusion Detection System**

#### 

#### **Basic Concepts**

Intrusion detection is the process of **detecting unauthorized use of**, or attack upon, a computer, network, or telecommunication infrastructure.

IDS are designed to aid in **mitigating the damage** that can be caused by **hacking or breaking into sensitive computer and network systems.**

##### Common Components of an IDS

* Sensors: collect traffic and user activity data and send it to an analyzer.
* Analyzer: detects an activity that it is programmed to deem as fishy and sends an alert to the administrative interface.
* Administrative Interface: Report the alert details.

##### Common Functions of an IDS

* Watch for attacks
* Parse audit logs
* Protect system files
* Alert administrators during attacks
* Expose a hackers technique
* Illustrate which vulnerabilities need to be addressed
* Help track down individual hackers

#### 

#### **IDS Types**

* **Network-Based IDS**: A network-based IDS (NIDS) uses sensors, which are either host computers with the necessary software installed or dedicated appliances— The NIC driver captures all traffic and passes it to an analyzer to look for specific types of patterns.
* **Host-Based IDS:** A host-based IDS (HIDS) can be installed on individual workstations and/or servers and watch for inappropriate or anomalous activity. HIDSs are usually used to make sure users do not delete system files, reconfigure important settings, or put the system at risk in any other way.

#### 

#### **IDS Technologies**

Both HIDS and NIDS can employ the following technologies:

* Knowledge or Signature Based
* Statistical Anomaly Based
* Rule-Based

##### **Knowledge or Signature Based**

* These are knowledge-based systems where some knowledge is accumulated about specific attacks and a model called signatures is developed.
* The main disadvantage of these systems is they cannot detect new attacks and a few signatures need to be written and continuously updated.
* Also known as misuse-detection system
* Attacks
  + Land Attacks ( packets modified to have the same s/c and destination IP)

**Security Humor**: Attacks or viruses that have been discovered in production environments are referred to as being “in

the wild.” Attacks and viruses that exist but have not been released are referred to as being “in the zoo.”

##### **Statistical Anomaly Based**

* These are behavioral-based systems, which do not use any predefined signatures, but rather are put in a learning mode to build a profile by continually sampling the environment's normal activities.
* The longer the IDS is put in a learning mode, in most instances, the more accurate a profile it will build and the better protection it will provide.
* Once a profile is built, a different profile is built based on the same sampling on all the future traffic and the data are compared to identify the abnormalities.
* Also known as profile-based systems
* Advantages
  + Can detect new attacks including 0 day attacks
  + Can also detect low and slow attacks in which an attacker tries to stay beneath the radar by sending a few packets at a time over a longer period of time.
* Disadvantages
  + Developing a correct profile to reduce false positives can be difficult.
  + There is a possibility for an attacker to integrate his/her activities into the behavior pattern of the n/w traffic. This can be controlled by ensuring that there are no attack activities currently underway while the IDS are in learning mode.
  + The success factors for these systems are based on determining the proper threshold in order to reduce/avoid false positives (threshold set to too low) or false negatives (threshold set to too high)
* Attacks
  + Bring the IDS offline by DoS and send the IDS incorrect data in order to distract the n/w and security individuals to make them busy chasing wrong packets, while the real attack takes place.
* Techniques
  + Protocol Anomaly-based:
    - These types of IDS have specific knowledge of each protocol that they will be monitoring.
    - The IDS builds a profile (model) of each protocol’s normal usage and uses it to match with the profile build during the actual operation.
    - Common protocol vulnerabilities
      * At the DLL, the ARP does not have any protection against ARP attacks where bogus data can be inserted into its table.
      * At the n/w layer, the ICMP can be used in a LOKI attack to move data from one place to another, when this protocol was designed to only be used to send status information. This data can be a code which can be made to be executed by the backdoor on a compromised system.
      * IP headers can be easily modified for spoofed attacks ( one acting as other)
      * At the TL, TCP packets can be injected into the connection between the two systems for a session hijack attack.
  + Traffic Anomaly-based:
    - These systems have traffic-anomaly filters, which detect changes in traffic patterns as in DoS attacks or a new service that appears on the network.
    - Once there is a profile that is built that captures the baselines of an environment’s ordinary traffic, all future traffic patterns are compared to that profile.
    - As with all filters, the thresholds are tunable to adjust the sensitivity, to reduce the number of false positives and false negatives.
    - Since this is a type of statistical anomaly-based IDS, it can detect unknown attacks

##### Rule-Based

* Rule-based intrusion detection is commonly associated with the use of an expert system.
* An expert system is made up of a knowledge base, inference engine, and rule-based programming.
  + Knowledge is represented as rules, and the data that is to be analyzed is referred to as facts.
  + The knowledge of the system is written in rule-based programming (IF situation THEN action). These rules are applied to the facts, the data that comes in from a sensor, or a system that is being monitored.
* Example: Consider the Rule-*IF a root user creates File1 AND creates File2 SUCH THAT they are in the same directory THEN there is a call to AdministrativeTool1 TRIGGER send alert.* This rule has been defined such that if a root user creates two files in the same directory and then makes a call to a specific administrative tool, an alert should be sent.
* The more complex the rules, the more demands on software and hardware processing requirements
* Cannot detect new attacks
* Techniques
  + State-Based IDS
    - A state transition takes place when a variable’s value changes, which usually happens continuously within every system.
    - In a state-based IDS, the initial state is the state prior to the execution of an attack, and the compromised state is the state after successful penetration.
    - The IDS has rules that outline what state transition sequences should sound an alarm. The activity that takes place between the initial and compromised state is what the state-based IDS looks for, and it sends an alert if any of the state-transition sequences match its preconfigured rules.
    - This type of IDS scans for attack signatures in the context of a stream of activity instead of just looking at individual packets. It can only identify known attacks and requires frequent updates of its signatures.
  + Model-Based IDS
    - In a model-based IDS, the product has several scenario models that represent how specific attacks and intrusions take place. The models outline how the system would behave if it were under attack, the different steps that would be carried out by the attacker, and the evidence that would be available for analysis if specific intrusions took place.
    - The IDS takes in the audit log data and compares it to the different models that have been developed, to see if the data meets any of the models’ specifications. If the IDS finds data in an audit log that matches the characteristics in a specific model, it sends an alert.

#### 

#### **IDS Sensors**

* Network-based IDSs use sensors for monitoring purposes. A sensor, which works as an analysis engine, is placed on the network segment the IDS is responsible for monitoring.
* The sensor receives raw data from an event generator and compares it to a signature database, profile, or model, depending upon the type of IDS.
* If there is some type of a match, which indicates suspicious activity, the sensor works with the response module to determine what type of activity needs to take place (alerting through instant messaging, page, e-mail, or carry out firewall reconfiguration, and so on).
* The sensor’s role is to filter received data, discard irrelevant information, and detect suspicious activity.
* A monitoring console can be used to monitor all sensors and supplies the network staff with an overview of the activities of all the sensors in the network, but the difficulty arises in a switched environment, where traffic is forwarded through a VPN and is not rebroadcast to all the ports. This can be overcome using Spanning Ports by mirroring the traffic from all the ports to one monitored port.
* Sensor Placement
  + Sensors can be placed outside of the firewall to detect attacks
  + Inside the firewall (in the perimeter network) to detect actual intrusions.
  + At highly sensitive areas, DMZs, and on extranets
* Multiple Sensors can be used in high traffic environments to ensure all packets are investigated. Also If necessary to optimize network bandwidth and speed, different sensors can be set up to analyze each packet for different signatures. That way, the analysis load can be broken up over different points.

Accountability: Event-Monitoring and log reviews. Log Protection

Event monitoring in IT is the process of collecting, analyzing, and signaling event occurrences to operating system processes, active database rules, and human operators.

**What is log event monitoring?**

The Monitor Event Log activity invokes runbooks when new events that match a filter that you specify appear in the Windows Event Log.

Log analysis is the process of reviewing computer-generated event logs to proactively identify bugs, security threats or other risks. Log analysis can also be used more broadly to ensure compliance with regulations or review user behavior.

## What Are Authentication Vulnerabilities?

Authentication vulnerabilities are issues that affect authentication processes and make websites and applications susceptible to security attacks in which an attacker can masquerade as a legitimate user.

Authentication is a vital part of any website or application since it is simply the process of recognizing user identities.

Several common categories of authentication issues can result from vulnerabilities in login credentials, site protection, or even underlying code.

## How Do Authentication Vulnerabilities Emerge?

There are several ways through which authentication vulnerabilities can arise, the most common of which are neglecting to address areas of risk, errors in code or logic, and poor user choices which can combine with the previous two. If an application has poor brute-force protection, attackers can take advantage of this to gain access to even well-protected accounts or cheap bulk access to poorly protected ones. Likewise, if there are logic or coding errors, malicious users may be able to bypass some or all of the authentication process.

## How Could Authentication-Based Vulnerabilities Impact You?

Authentication vulnerabilities have serious repercussions — whether it’s because of weak passwords or poor authentication design and implementation.

Malicious users can use these vulnerabilities to get access into systems and user accounts to:

* Steal sensitive information
* Masquerade as a legitimate user
* Gain control of the application
* Destroy the system completely

If attackers can illegally pass through the authentication process and gain access to a user account, they can steal critical data like names, credit card details, social security numbers (SSNs), medical records, and tax IDs. They can also take actions on a user’s behalf, such as initiating financial transactions, deleting data, or transferring ownership of resources.

Worse still, attackers can fully control your system or completely shut it down if they find their way into top-level accounts such as admin profiles. Once present, these authentication vulnerabilities can substantially affect your company’s viability and credibility.

Moreover, you may face legal consequences with [GDPR](https://gdpr-info.eu/), [CCPA](https://privacyrights.org/resources/california-privacy-rights-act-overview), financial services regulations, or other local laws depending on what type of data is breached.

## 11 Most Common Authentication Vulnerabilities

Authentication vulnerabilities, if not properly controlled, can damage not just a company’s security but its reputation as well.

Here are 11 of the most common authentication-based vulnerabilities to watch out for:

### 1. Flawed Brute-Force Protection

A [brute-force attack](https://www.strongdm.com/blog/brute-force-attack), such as a dictionary attack, is an attempt to gain illegal access to a system or user’s account by entering large numbers of randomly generated or pregenerated combinations of usernames and passwords until they find one that works.

If there’s a flawed brute-force protection system such as a flaw in the authentication logic, firewall, or secure shell (SSH) protocol, attackers can hijack login credentials and processes, compromising the security of user credentials.

### 2. Weak Login Credentials

When users register for an account on a site or application that uses password-based logins, they’re prompted to create a username and password.

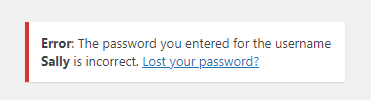
However, if the password is predictable, this can lead to vulnerabilities in the authentication process. Predictable usernames can make it easier for attackers to target specific users.

Rather than using a full brute-force attack, the attackers will look for accounts with easy-to-guess passwords, which are used far too often. . They’ll try common credentials like "admin," "admin1," and "password1." With no restrictions on weak passwords, even sites protected against brute-force attacks can find themselves compromised.

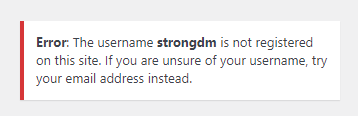
### 3. Username Enumeration

Username enumeration is not exactly an authentication vulnerability. But, it can make an attacker’s life easier by lowering the cost for other attacks, such as brute-force attacks or weak credential checks.

This process of username enumeration confirms whether or not a username is valid. For example:



In this case, the username is correct but the password isn’t.



Here, the username is invalid.

The problem with username enumeration is that attackers can tell what usernames are valid. Then, they can try to hack valid user accounts using brute-force techniques without wasting their time and money testing a multitude of invalid account names.

### 4. HTTP Basic Authentication

This classic web authentication protocol is easy to implement, however, it is not without its risks.

HTTP basic authentication is simple, sending a username and password with each request. However, if appropriate security protocols such as TLS session encryption are not used for all communication, the username and password information can be sent in the clear, making it easy for attackers to steal the credentials.

Because the included credentials contain so little context, they can easily be misused in attacks such as cross-site request forgeries (CSRF). Also, because they are included with every single request, modern browsers normally cache this information indefinitely, with minimal ability to "log out" and prevent a local attacker from reusing the credential at some future point.

### 5. Poor Session Management

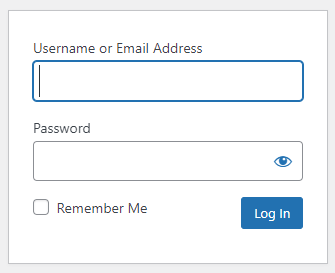
A vulnerability in the management of session identifiers leads to hijacking of valid authenticated sessions. This is one of the common web vulnerabilities to bypass passwords.

There are several session mismanagement vulnerabilities such as no session timeouts, exposure of session IDs in URLs, session cookies without the Http-Only flag set, and poor session invalidation.

If attackers can seize control of an existing session, they easily get into a system by assuming the identity of an already-authenticated user, bypassing the authentication process entirely.

### 6. Staying Logged In

A "Remember me" or "Keep me logged in" checkbox beneath a login form makes it super easy to stay logged in after closing a session. It generates a cookie that lets you skip the process of logging in.



However, this can lead to a cookie-based authentication vulnerability if an attacker can predict a cookie or deduce its generation pattern. They can use malicious techniques like brute-force attacks to predict cookies, and cross-site scripting (XSS) to hack user accounts by allowing a malicious server to make use of a legitimate cookie.

If a cookie is poorly designed or protected, attackers may be able to obtain passwords or other sensitive (and legally protected) data such as user addresses or account information from a stored cookie.

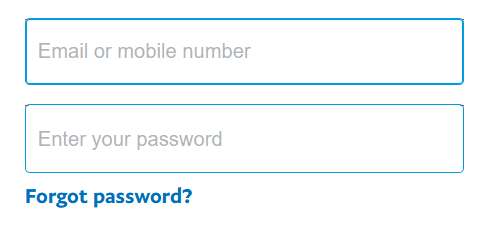
### 7. SQL Injection

SQL injection is an [attack vector](https://www.strongdm.com/blog/attack-vector) that uses malicious SQL code input in an unexpected way to manipulate and access a database.

SQL injections can enable attacks on authentication mechanisms by stealing relevant data (such as poorly protected password hashes) from an unprotected database. They can also bypass authentication mechanisms if the injected SQL code is executed by an internal (and already authorized) tool that failed to sufficiently validate external input.

### 8. Unsecure Password Change and Recovery

Sometimes, users forget or just want to change their passwords and click the "Forgot password" or "Lost your password" links.



The password reset process poses an authentication vulnerability if an application uses a weak password recovery mechanism such as easy security questions, no CAPTCHAs, or password reset e-mails with overly long or no timeouts.

If the password recovery functionality is flawed, attackers can potentially use brute-force techniques or access to other compromised accounts to gain access to user accounts and credentials that are well-protected under normal circumstances.

### 9. Flawed Two-Factor Authentication

While two-factor authentication (2FA) is effective for secure authentication, it can cause critical security issues if not well-implemented.

Attackers can figure out the four- and six-digit 2FA verification codes through [SIM swap attacks](https://www.verizon.com/about/account-security/sim-swapping) if they are sent through SMS. Some two-factor authentication is also not truly two-factor; if a user is attempting to access sensitive information on a stolen phone using cached credentials, a "second factor" that sends a message to that same phone adds no additional security.

Two-factor authentication vulnerabilities can also occur if there’s no brute-force protection to lockout an account after a specific number of attempted logins.

### 10. Vulnerable Authentication Logic

Logic vulnerabilities are common in software applications. This occurs as a result of poor coding or design that affects authentication and authorization access, and application functionality.

Flawed application logic can be due to the abuse of functionality, weak security measures, or a skipped step in the verification procedure.

For example, an application can prompt a user to answer a security question the logic deems as something "only the correct person would know." But questions like the user’s birthday or mother’s maiden name are often easy to discover. This vulnerability makes it easy for cyber attackers to bypass authentication and gain illegal access to such accounts.

### 11. Human Negligence

According to [Shred-it 2020 report](https://www.shredit.com/content/dam/shred-it/global/documents/Shred-it_2020-Data-Protection-Report_US.pdf.coredownload.inline.pdf), up to 31% of C-suite executives reported employee negligence to be the second major cause of their data breaches.

Human error can result in serious authentication vulnerabilities that are far easier to take advantage of than brute-force attacks, SQL injections, and authentication bypasses. This negligence includes actions such as:

* Leaving a computer on and unlocked in a public place
* Losing devices to theft
* Leaking sensitive information to strangers
* Writing bad code

## How to Prevent Authentication Vulnerabilities?

While authentication vulnerabilities are easy to identify, they greatly impact cybersecurity. But, you can prevent them from happening.

Here are eight best practices to prevent authentication-based vulnerabilities and keep critical information safe.

1. **Implement a reliable brute-force protection system:** Brute-force attacks can be prevented by enforcing account lockouts, rate limiting, IP-based monitoring, application firewalls, and CAPTCHAs.
2. **Enforce a secure password policy:** Do this by creating a password checker that tells users how strong their passwords are in real-time. You can also implement [passwordless authentication](https://www.strongdm.com/blog/passwordless-authentication) using standards like [FIDO2](https://www.strongdm.com/blog/fido2) to mitigate the risk and stress of managing passwords.
3. **Apply HTTP strict transport security (HSTS):**This forces web sessions to use TLS encryption, preventing sensitive information from being accessed in transit.
4. **Consider disabling username enumeration:**By generating the same error for a login failure whether the username was valid or invalid, you force an attacker to brute-force not just the set of possible passwords, but also the set of likely usernames, rather than sticking to the ones they know are valid.
5. **Modify cookie headers:** Modifying cookie headers protects them against malicious attacks. Using the HttpOnly and SameSite tags when setting cookie headers prevents them from XSS and CSRF attacks, respectively.
6. **Scrutinize your coding on verifications:** This is important for detecting any vulnerabilities in your code.  
     
   Overall, periodically audit your code to discover logic flaws and authentication bypass and strengthen your security posture.
7. **Use parameterized statements:** You can prevent SQL Injection attacks through input validation and parameterized queries. They are safer to avoid directly putting user-provided input directly into SQL statements.
8. **Implement proper multi-factor authentication:** Using multi-factor authentication is more secure than password-based mechanisms. However, you need solid code and secured generation of verification codes to effectively implement this form of authentication.