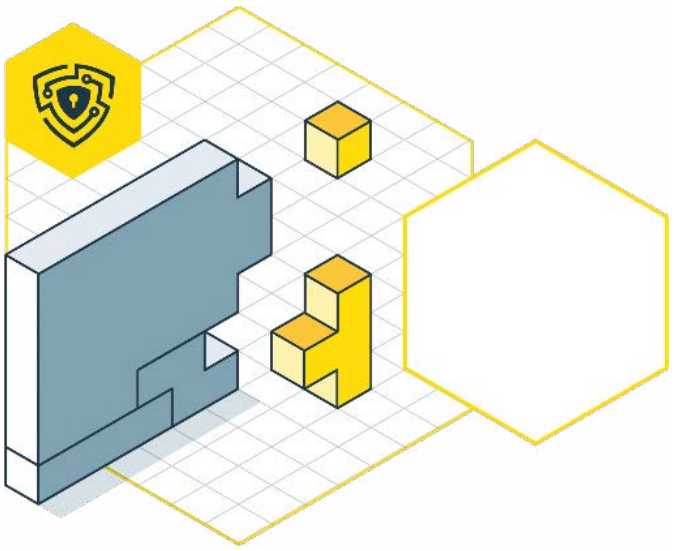
CRASHTEST **SECURITY**

GUIDE TO PREVENT IDENTIFICATION AND AUTHENTICATION FAILURE

WHAT ARE THE STEPS TO KEEP YOUR WEB APP OR API SAFE FROM SUCH VULNERABILITY

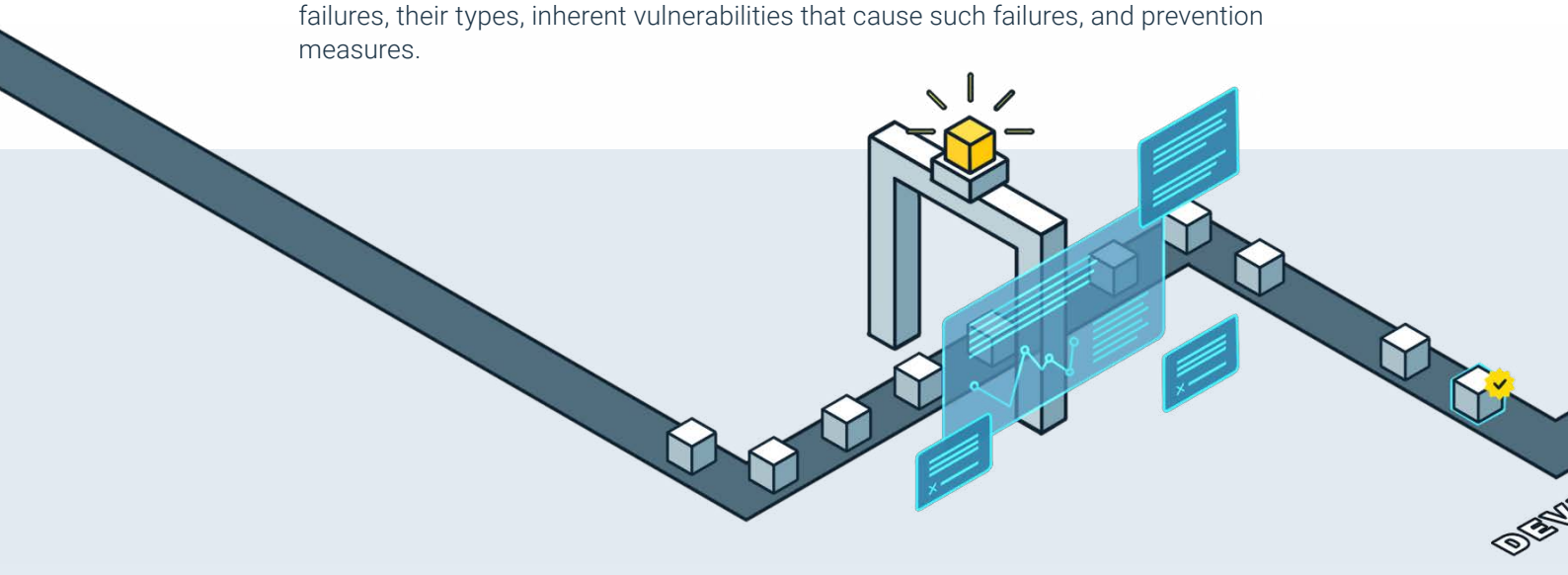
PREVENTIONGUIDE FOR THE IDENTIFICATION AND AUTHENTICATION FAILURE

Table of Contents



INTRODUCTION TO THIS GUIDE

Identification and authentication help secure a digital framework's perimeter as the first line of defense. Identification involves attributing each user's unique identity to use an application's services. On the other hand, an authentication mechanism validates a user session's legitimacy based on assigned identities and access credentials. Identification and authentication failures occur when the application fails to correctly implement functions associated with the user's identity, authenticity, and session management. Such failures often lead to persistent system-level threats exploited by malicious actors to assume a user's identity, data theft, or an entire system compromise. This post discusses identification and authentication failures, their types, inherent vulnerabilities that cause such failures, and prevention measures.



WHAT IS AN IDENTIFICATION AND AUTHENTICATION FAILURE?

Identification and authentication mechanisms of an application layer enforce security by verifying the source of a request before granting access to application resources. In applications containing identification and authentication failure vulnerability, the application logic cannot associate a user access event with the correct user endpoint. When the application lacks secure mechanisms to validate the real identity of a triggered access request, attackers can exploit the vulnerability by claiming the identities of legitimate users to steal sensitive data or orchestrate system-level attacks.

TYPES OF IDENTIFICATION AND AUTHENTICATION FAILURES

Previously known as the Broken Authentication flaw, identification and authentication failures fall under several weaknesses under the Common Weaknesses Enumeration list that often leads to data breaches. These include:

CREDENTIAL MANAGEMENT ERRORS

Applications rely on credentials for controlling identities and access to resources and data. Credential management errors occur when these credentials are exposed, or attackers can abuse credential management systems for malicious authentication attempts. Base-level weaknesses for credential management include:

- **Hard-coded passwords** - Credentials are typically stored in a centralized database and transported to the application during the authentication process. Hard-coded credentials are embedded directly into an executable object or the application's source code. Hard-coded secrets are common targets of password-guessing exploits that allow attackers to bypass authentication mechanisms and retrieve credentials through automated tools or a simple guess.

- **Plaintext storage of password** - Even though user credentials are stored in an external database, they should be encoded and hashed to prevent accidental exposure. Storing passwords in plaintext format within a configuration file, the application's properties or memory is a commonly found misconfiguration. Anyone accessing these resources can obtain passwords, allowing them to mislead an identification process.

- **Unprotected transport of credentials** - This flaw occurs when weak cryptographic techniques are used to secure credentials in transit from an external database. Attackers can eavesdrop on the authentication service to obtain these credentials and assume the identities of registered users.

- **Weak password expiration mechanism** - Password aging is a minimum security-strength enhancement feature to reduce the impact of account breaches. If the application fails to age and rotates passwords regularly, a compromised account becomes a persistent threat since the attacker maintains their hold of the identification process.

AUTHENTICATION FAILURES

The improper configuration of authentication services allows threat actors to exploit application resources through account takeovers. Authentication failures can be categorized into several common weaknesses, including:

- **Missing authentication for critical function** - The application relies on a weak authentication mechanism for actions and services that require privileged access. Such necessary actions include issuing credit cards, health insurance cards, national identity cards, and other administrative functions.

- **Improper restriction of failed authentication attempts** - When the application fails to limit the number of authentication attempts within a small time frame, it is susceptible to brute-force attacks, where attackers trigger a series of authentication attempts to gain access.

- **Lockout mechanism errors** - When the application implements a lenient mechanism to restrict resource access after multiple failed login attempts, hackers can perform an arbitrary number of authentication attempts until they can access the restricted resource. On the other hand, an overly restrictive lockout mechanism degrades the user experience since users are frequently

SESSION MANAGEMENT FLAWS

Web sessions help improve user experience by reducing their time to log in within a specified time frame. In the absence of a secure session management mechanism, an attacker can steal the user's session token to assume the legitimate user's identity. Session management flaws include:

- **Session fixation** - Session fixation errors occur when the application establishes a new user session without terminating an existing session. Attackers leverage session fixation flaws to steal a user's session and exploit it for stealing identities, financial theft, or other malicious purposes.

- **Insufficient session expiration** - These flaws occur when a user's session persists beyond their interaction with the website. An attacker can use the existing session's credentials for successful authentication, granting them access to the user's account.

AUTHENTICATION BYPASS VULNERABILITIES

Even with the proper configuration of authentication factors, threat actors can develop workarounds to bypass the authentication service. Common authentication service bypass vulnerabilities include:

- **Authentication bypass with a secondary channel** - The software requires authentication for its primary medium but includes an alternate path that does not require authentication. Attackers can exploit such setups to connect to the application through the secondary channel, allowing them to access data and resources without undergoing authentication.

- **Authentication bypass by spoofing** - This weakness allows the attacker to access resources by spoofing (pretending to be a recognized user) the authentication service

- **Authentication bypass by capture-replay** - These flaws allow the hacker to eavesdrop on network traffic and replay it to the identification process (with minor changes), which further may result in an authentication bypass

- **Authentication bypass using assumed immutable data** - The identification process often uses key data elements that are considered to be immutable. In instances where attackers obtain access to this data, they can orchestrate successful authentication attempts even without access to user credentials.

IDENTIFICATION AND AUTHENTICATION FAILURE - SEVERITY LEVEL

The identification and authentication failure vulnerability is ranked **number 7** on the OWASP 2021 Top 10 list, 5 positions down from the 2017 list (Broken Authentication A02:2017). The exposure is mapped to 22 Common Weaknesses and Enumerations, with an average **incidence rate of 2.55%**, an average **weighted exploit of 7.4**, and an **average weighted impact of 79.51%**.

Common exploits over improper authentication vulnerabilities include:

- **Credential stuffing attacks** - Malicious actors rely on an automated hit & trial method to pass user credentials for unauthorized access

- **Brute-force attacks** - The attacker submits an arbitrary number of username-password combinations with the hope that they'll eventually obtain access to legitimate accounts.

- **Data breaches** - Attackers leveraging identification and authentication failures to log into secure devices and access restricted content, such as customer details and user credentials.

- **System compromise** - After gaining initial access to crucial systems such as centralized databases, payment interfaces, etc., attackers can further perform unwanted actions that lead to partial or full system compromise.

HOW TO IDENTIFY IDENTIFICATION AND AUTHENTICATION FAILURES WITH CRASHTEST

Crashtest Security Suite automates penetration and vulnerability testing methods for detecting numerous vulnerabilities, including identification and authentication failures. The platform is a collective suite of various vulnerability scanners that help detect identification and authentication failures, including:

- **OWASP Scanner** - Tests the web application against all vulnerabilities listed on the OWASP Top 10, including losses that lead to broken authentication attacks
- **Privilege Escalation Scanner** - Scans privilege management algorithms to inspect whether malicious insiders can gain elevated horizontal or vertical permissions
- **CSRF Scanner** - Tests for flaws within the session management process that can allow an attacker to perform session fixation attacks
- **DAST Scanner** - Performs simulated tests to identify runtime flaws within the authentication service

Crashtest Security implements automated penetration testing and ethical hacking workflow to help simulate how attackers can leverage identification and authentication failures in exploits. The automated black-box penetration testing tool provides a rapid security assessment by benchmarking the application's security posture against OWASP's Top 10 recommendations.

Try Crashtest Security today and discover how its automated detection techniques can help eliminate security blind spots within your application framework.

IDENTIFICATION AND AUTHENTICATION FAILURE PREVENTION TECHNIQUES

Techniques to secure devices and applications against broken authentication and identification attacks include:

CRYPTOGRAPHIC TECHNIQUES

When storing credentials in a central database, developers should emphasize using cryptographic solid hash functions to prevent the misuse of sensitive secrets. To ensure sufficient entropy for each database entry and reduce the likelihood of an attack, thorough research of the most effective algorithms for each use case is often the first starting point. To enhance the randomness of the hash function, administrators can also use salts and store them separately from other customer details to prevent advanced credential-stuffing attacks.

STRONG IDENTIFICATION SOLUTIONS

Adopting strong identification solutions, such as biometric authentication factors, into the identification process introduces a new dimension in identification mechanisms since these use physical features that are unique to each individual.

Some strong identification solutions based on biometrics include:

- **Facial recognition systems** - A biometric authentication technology that identifies an individual using a facial image built with a set of physical features stored in a biometric database. Individuals are only allowed access if their facial features can be reconstructed to match the correct image stored in the authentication database.
- **Fingerprint recognition systems** - A biometric authentication technology that authenticates users using unique digital fingerprints as digital signatures. Fingerprint matching algorithms have found a wide application in mobile devices, banking, and automotive industries due to their simplicity and effectiveness.

Besides biometric devices, some strong identification solutions also go beyond physical features. These include:

- **DNA-based recognition** - An advanced biometric data protection method that uses a person's genetic information for identity and access management. While it is an expensive strong identification solution, DNA-based recognition provides robust guarantees as a person's genetic makeup stays constant throughout their lifetime.
- **Physical control** - This technique involves the placement of physical barriers with personnel checks to ensure that only authorized individuals can access secure devices. This is typically the first line of defense against identification and authentication attacks by securing the location of physical infrastructure.

MULTI-FACTOR AUTHENTICATION

Multi-factor authentication is a security control that requires the user to provide two or more authentication factors before they are granted access to the requested resource. The authentication mechanism is widely used as a critical feature of a strong identification solution, requiring users to identify themselves using more than just a username-password combination.

Multi-factor authenticators rely on three types of information:

something the user knows (represents knowledge, e.g., PIN)
something the user has (represents possession, e.g., single-factor OTP device, mobile device, smart card)
something the user is (represents inherence, e.g., biometric authentication technology).

Some typical multi-factor authentication schemes include:

- **Credentials with One-time password (OTP)** - Once the user has supplied their username-password combination, the authentication service sends a 4-8 digit code to a personal security component, multi-factor OTP device, the user's email address, or SMS number. A new code is sent to the multi-factor OTP device every time the user submits an authentication request, giving access only when they can present the OTP for enhanced security.

- **Credentials with biometric device** - Once the user supplies their username-password combination, the authentication process requires additional biometric data such as facial images, fingerprints, or biometric payment cards.

BEST PRACTICES IN PREVENTING IDENTIFICATION AND AUTHENTICATION FAILURES

Some foundational practices that prevent the occurrence of identification and authentication failures include:

REPLACE DEFAULT CREDENTIALS

As default credentials are often easy to guess or can be cracked through automated tools, developers should enforce the restriction on using default credentials, especially for administrative accounts and decentralized devices.

ENFORCE MINIMUM SECURITY STRENGTH FOR STRONG PASSWORDS

Developers should align the complexity, length, and aging policies with robust standards recommended by the International Standards Organization, National Institute of Science and Technology (NIST), or other consumer protection benchmarks. To ensure all passwords in use conform to the adopted standards, it is also crucial to perform routine weak password tests (WPT), which essentially inspect all currently used passwords within the centralized database against the top 10,000 worst password list.

ENFORCE SERVER-SIDE SESSION MANAGEMENT

Utilize an inbuilt, server-side session manager for the random generation of a unique session ID with each authentication attempt. The session ID should be stored in a secure device, not exposed via a URL, and should be terminated after a specified set of interactions with the application.

IMPLEMENT A REASONABLE LOCKOUT MECHANISM

Enforce a lockout mechanism to prevent login attempts after several failed authentication attempts. The mechanism should also log all authentication failures while including alerts for brute-force attacks, credential stuffing, or other exploit patterns that leverage identification and authentication failures.

ELIMINATE IDENTIFICATION AND AUTHENTICATION FAILURES WITH CRASHTEST SECURITY

Crashtest Security integrates seamlessly with most major development frameworks to get started with automated penetration testing and vulnerability mitigation in minutes. Through various vulnerability scanners, the security suite helps detect and eliminate identification and authentication failures before attackers can spot them. The tools also offer comprehensive security reports with hardening and remediation advice for enhancing an application's security posture.

To know more about how Crashtest Security's automated methods for detection and remediation of authentication failures can help your organization, try a free 14-day trial here.

Start 2-Week Trial for Free

CRASHTEST **SECURITY**

WWW.CRASHTEST-SECURITY.COM