Broken Authentication and Session Management

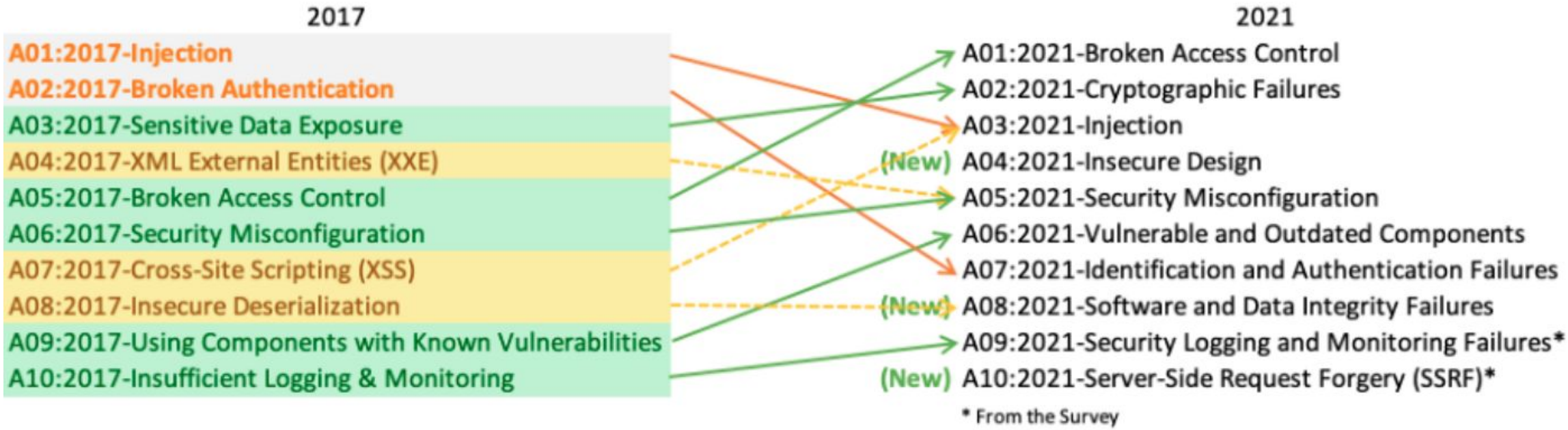
Lecture-9

Department of CSE, IUT

OWASP top 10 Vulnerabilities (for 2017)

- Injection
- Broken Authentication
- Sensitive Data Exposure
- XML External Entities (XXE)
- Broken Access Control
- Security Misconfiguration
- Cross-Site Scripting XSS
- Insecure Deserialization
- Using Components with Known Vulnerabilities
- Insufficient Logging & Monitoring

OWASP Top 10 Vulnerabilities (for 2021)



OWASP Top 10 Vulnerabilities (for 2021)

OWASP
Open Web Application Security Project

Visit

https://www.hacksplaining.com/owasp

https://owasp.org/Top10/

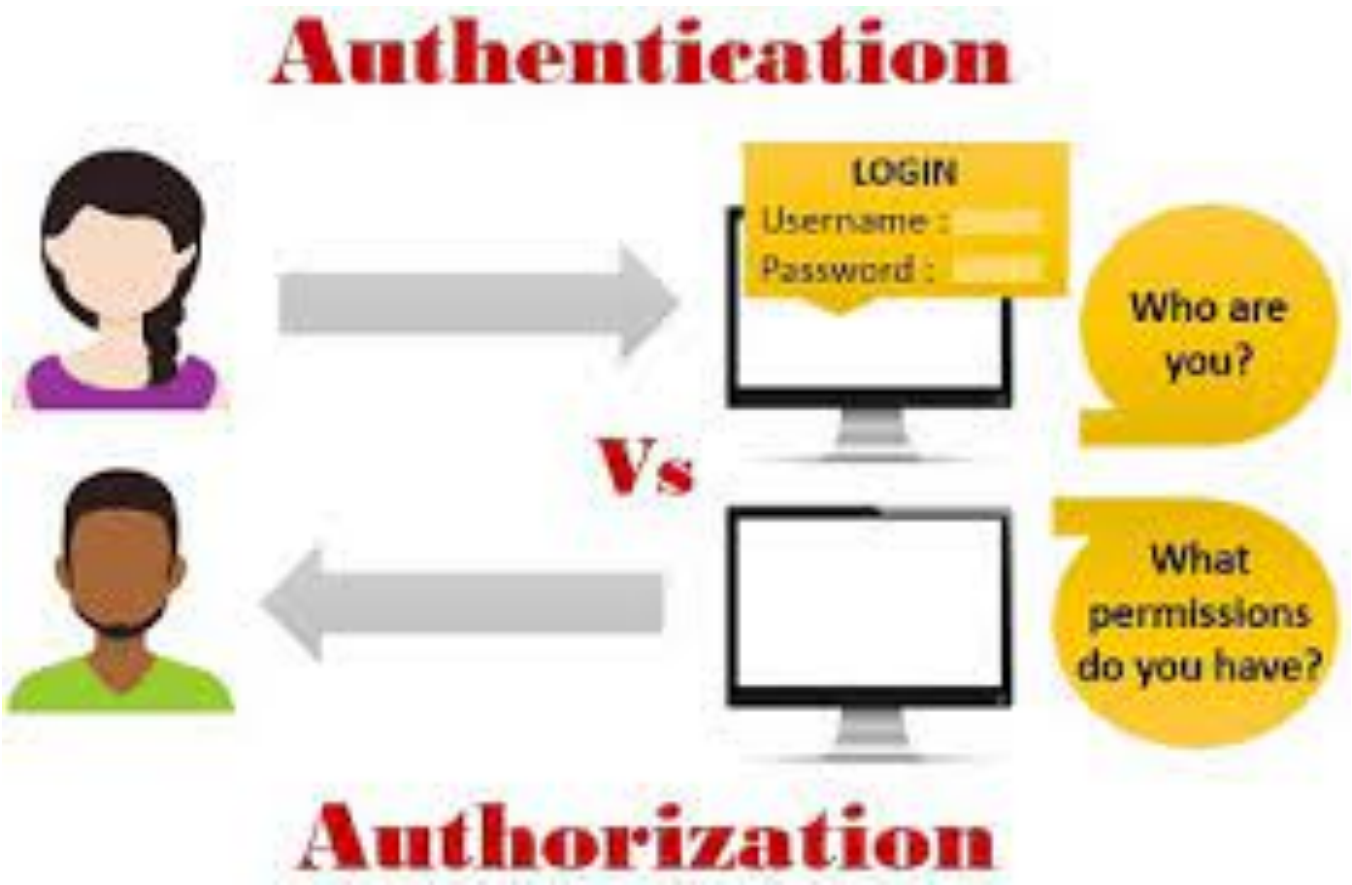
Introduction

Before discussing **Broken Authentication,** Lets discuss the following:

- Authentication vs. Authorization
- Closer look at Authentication Mechanism
- Session Management

Authentication vs. Authorization

Authentication vs. Authorization



Authentication vs. Authorization

Authentication:

- Authentication is the act of validating that users are whom they claim to be. This is the first step in any security process.

Authorization:

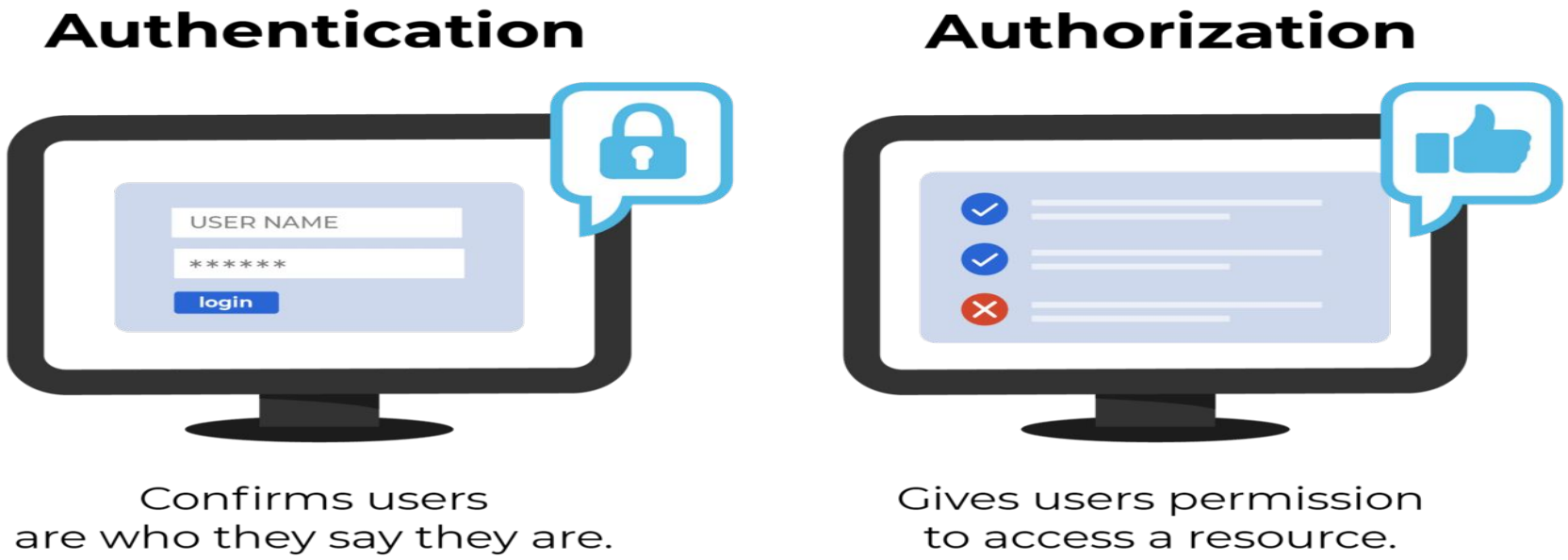
- Authorization in system security is the process of giving the user permission to access a specific resource or function. This term is often used interchangeably with access control or client privilege.

Authentication vs. Authorization

- **Authentication**, in the **form of a key**. The lock on the door only grants access to someone with the correct key in much the same way that a system only grants access to users who have the correct credentials.

- **Authorization,** in the **form of permissions**. Once inside, the person has the authorization to access the kitchen and open the cupboard that holds the pet food. The person may not have permission to go into the bedroom for a quick nap.

Authentication vs. Authorization

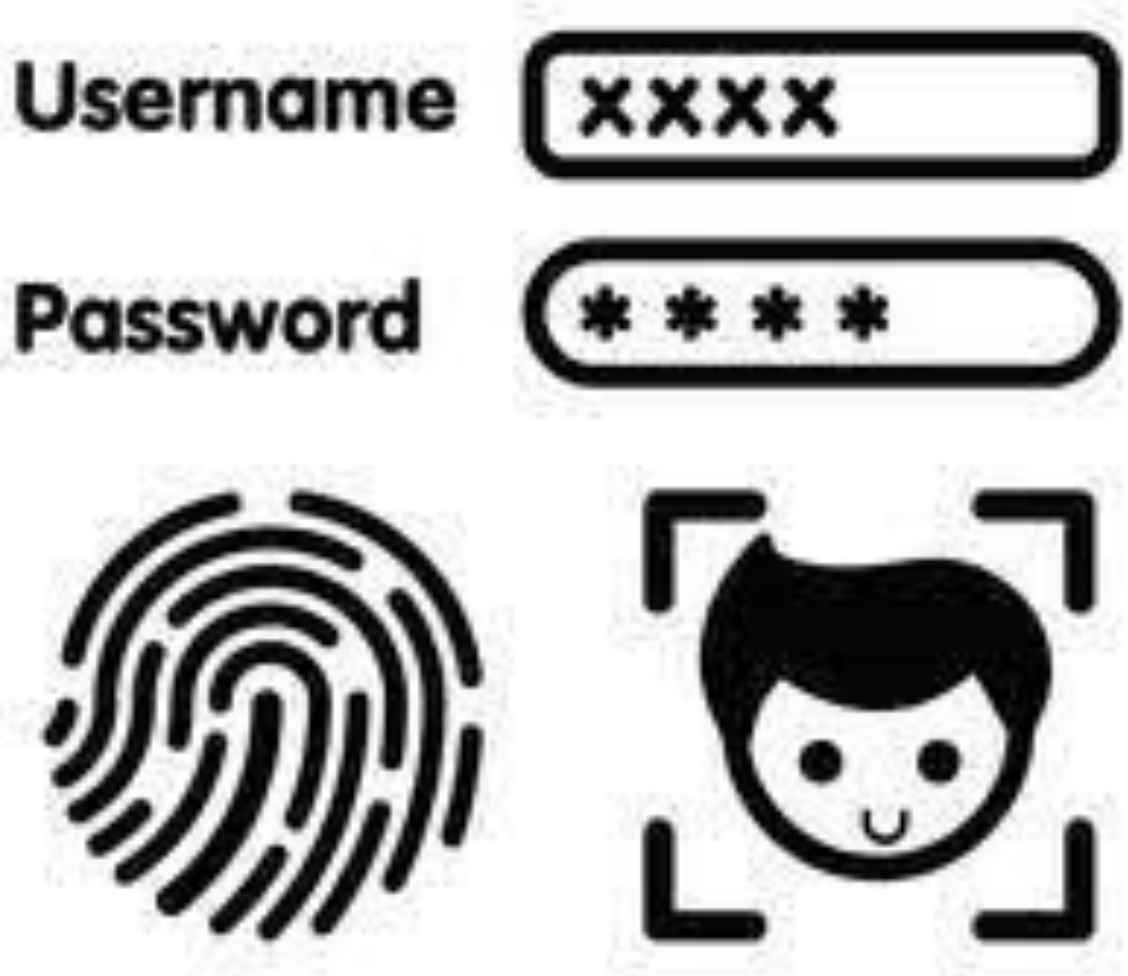


okta

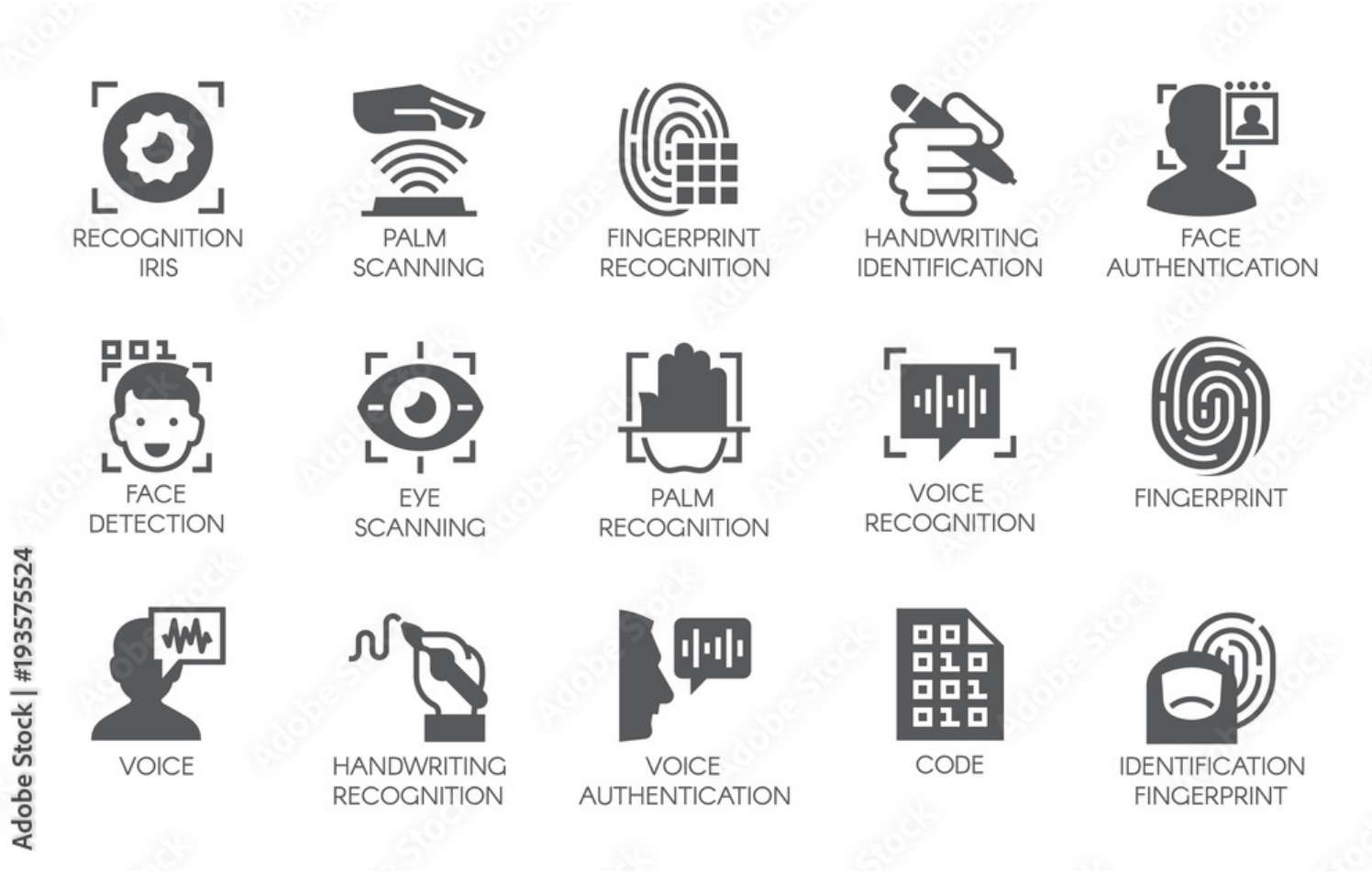
Authentication vs. Authorization

	Authentication	Authorization
What does it do?	Verifies credentials	Grants or denies permissions
How does it work?	Through passwords, biometrics, one-time pins, or apps	Through settings maintained by security teams
Is it visible to the user?	Yes	No
It is changeable by the user?	Partially	No

Closer look at Authentication Mechanism



Closer look at Authentication Mechanism



Closer look at Authentication Mechanism

- There are four general means of authenticating a user's identity, which can be used alone or in combination:
  - **Something the individual knows**: Examples include a password, a personal identification number (PIN), or answers to a prearranged set of questions.
  - **Something the individual possesses**: Examples include cryptographic keys, electronic keycards, smart cards, and physical keys. This type of authenticator is referred to as a token.
  - **Something the individual is (static biometrics)**: Examples include recognition by fingerprint, retina, and face.
  - **Something the individual does (dynamic biometrics)**: Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm. [1]

Closer look at Authentication Mechanism

Authentication: The process of validating that the identity accessing an asset is the one it claims to be.

- Most common way of authentication is username and password.
- Example: When you log in to your email account, you provide a username and password.

Authentication Features

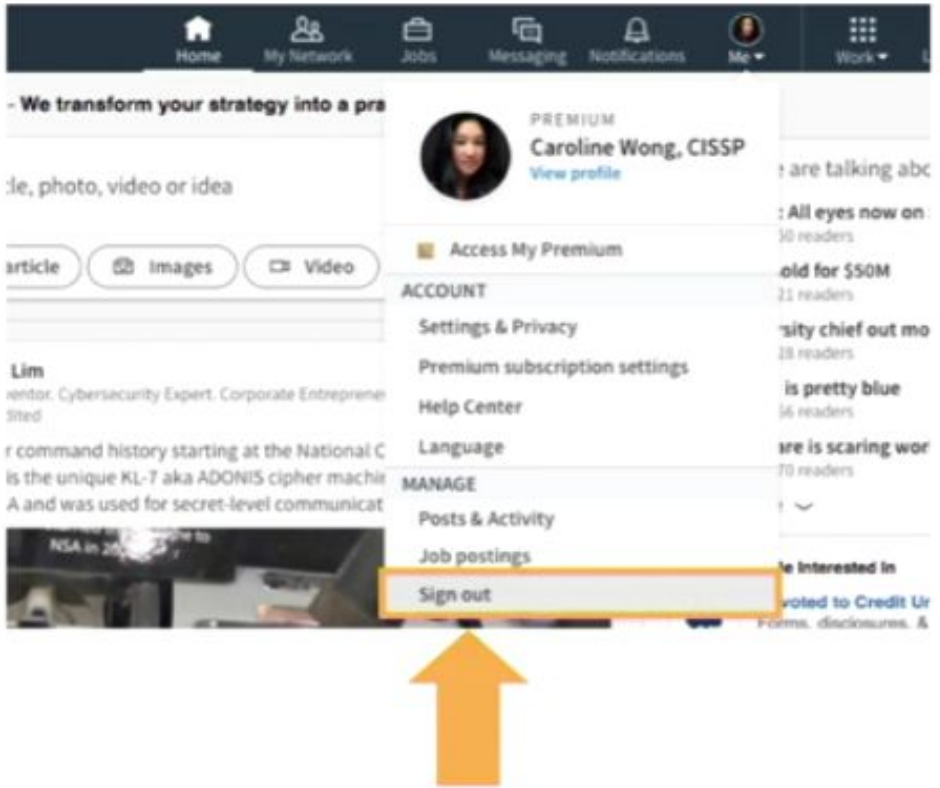
Authentication requires set of features like-

- Sign up: Allows users to register their accounts on the system.
- Login: Allows them to access it. We generally refer this part as authentication.
- Password Reset: Helps them recover their credentials.
- Multi-Factor Authentication (MFA): Increases Authentication security by adding other factors such as a token, an SMS, physical biometrics such as fingerprints, etc.

Session

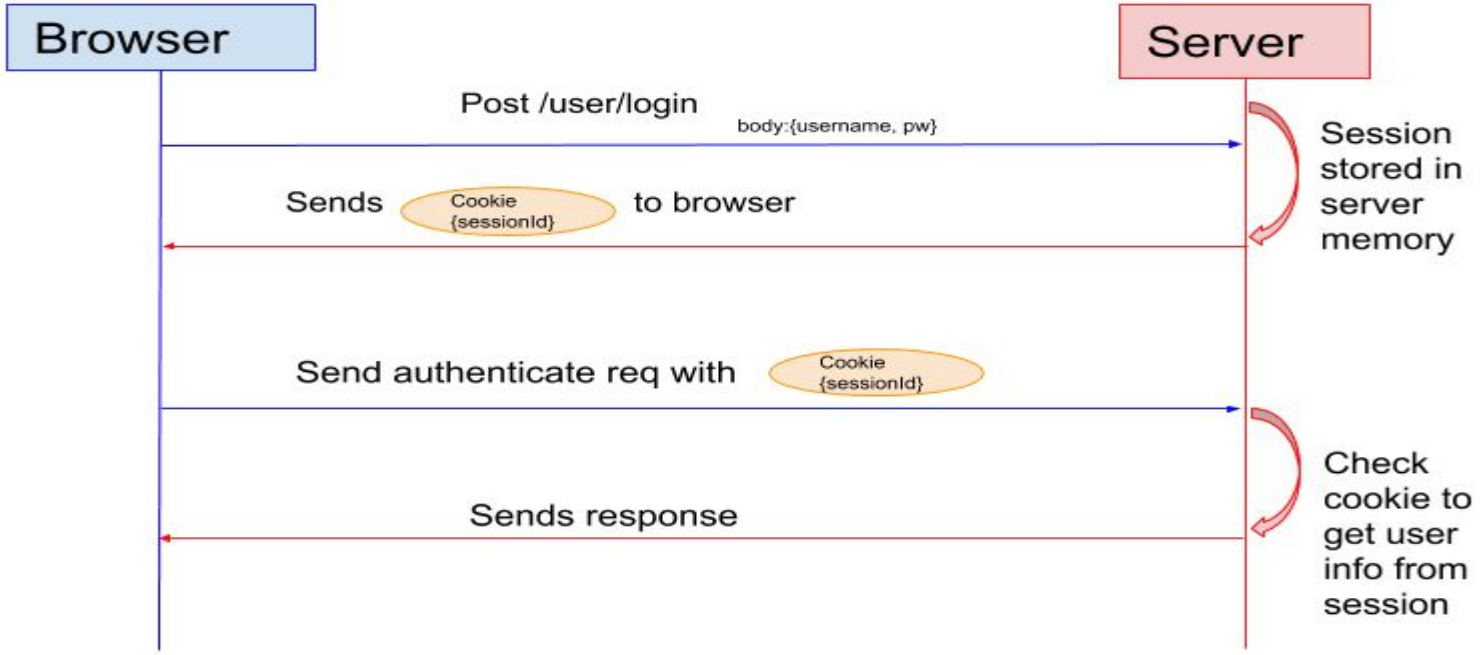
Session

- Active account
- In use by account owner



Session Management

- Prompting authentication info subsequently for every HTTP request is impractical.
- After first authentication, identity receives a session for further requests. It typically lasts as long as the identity is connected to the asset.
- Once the identity logs out, the server destroys the session



Broken Authentication

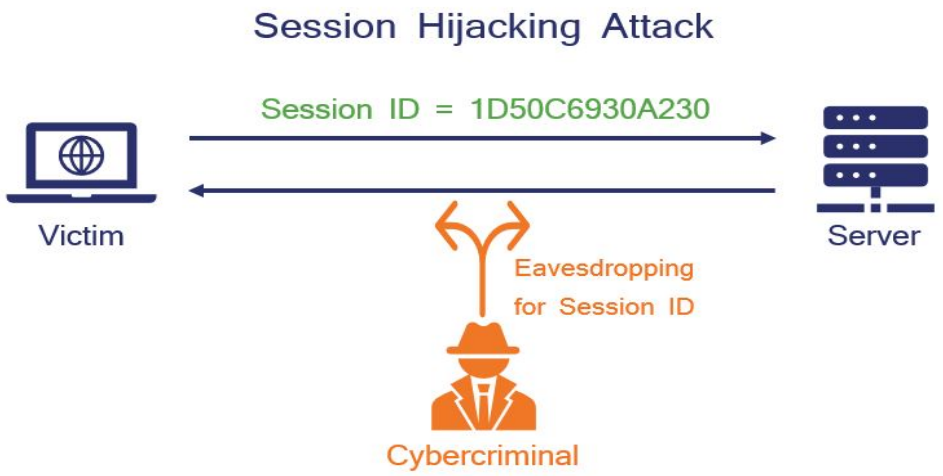
Broken Authentication

Broken Authentication

- An umbrella term for several vulnerabilities
- Attackers exploit it to impersonate legitimate users online.
- Refers to 2 weaknesses areas:
  1. session management
  2. credential management
- attackers can use either avenue to masquerade as a user:
  1. hijacked session IDs
  2. stolen login credentials

Broken Authentication and Session Management

- Someone **log in to an account they're not supposed to have access to.**
- Application functions related to authentication and session management are often implemented incorrectly,
  - allowing attackers to compromise passwords, keys, or session tokens



Example Attack Scenarios

- **Scenario #1**:
- Credential stuffing, the use of lists of known passwords, is a common attack.
- If an application does not implement automated threat or credential stuffing protections,
- the application can be used as a password oracle to determine if the credentials are valid.

Example Attack Scenarios

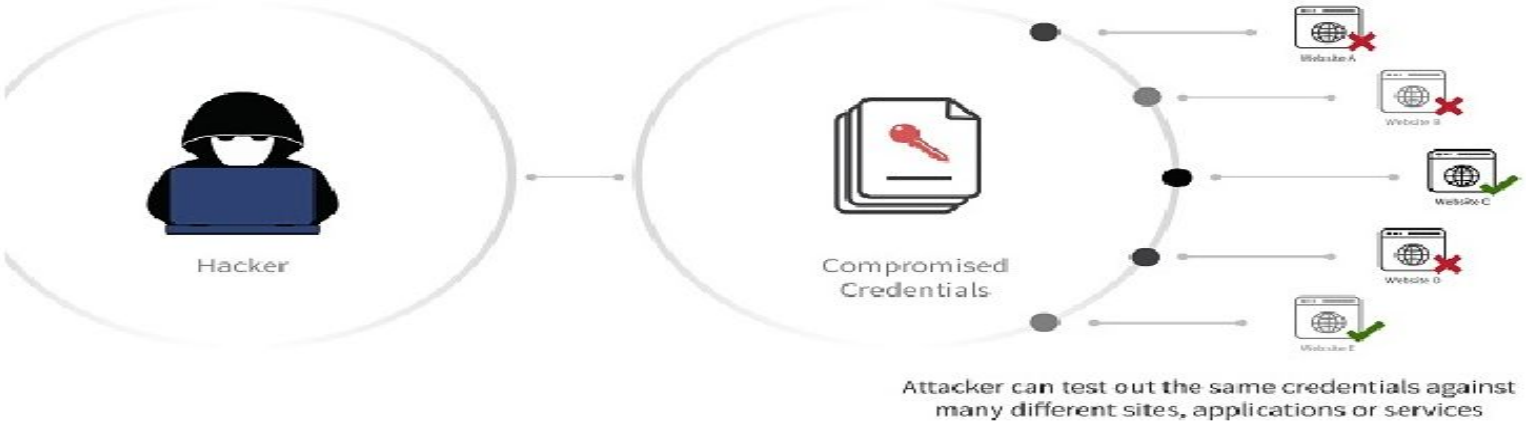
- **Scenario #2**:
- Application session timeouts aren’t set properly.
- A user uses a public computer to access an application.
- Instead of selecting “logout” the user simply closes the browser tab and walks away.
- An attacker uses the same browser an hour later, and the user is still authenticated.

Types of Broken Authentication Attack

- **Session Hijacking**
  - Hackers can use stolen session IDs to gain unauthorized access to a system and impersonate the victim within.
  - These kinds of attacks can occur through attack vectors such as phishing attacks, malware.

Types of Broken Authentication Attack

- **Credential Stuffing**
  - When databases of unencrypted login credentials are leaked, hackers can use automated tools or botnet to test credentials stolen from one site on different accounts.
  - This works because, people frequently use the same password across applications.



Types of Broken Authentication Attack

- **Password Spraying**
  - Instead of DB, attackers use lists of the most common and weak passwords for **brute forcing to break** into a user's account.
  - A 2019 survey by the UK's National Cyber Security Centre (NCSC) found that 23.2 million accounts used "123456" as their password.[4]

Types of Broken Authentication Attack

- **Phishing Attacks/ Social Engineering Attacks**
  - Attackers typically phish by sending users an email pretending to be from a trusted source and then tricking users into sharing their credentials.
  - According to Verizon's 2020 Data Breach Investigations Report, attackers used phishing in 22% of the investigated breaches.[4]

Types of Broken Authentication Attack

1. Social Engineering

- How to trick someone to give you their username and password

Broken Authentication Vulnerabilities

- **Weak or default passwords**
  - If the system doesn't enforce a strong password policy, there is a high chance that users will use weak passwords
  - The software ships with default admin credentials, which are publicly available. So, any user can log in to the system as an administrator.

Broken Authentication Vulnerabilities

- **Broken Password Reset/ Broken Application Logic**
  - Password reset can be vulnerable depending on how the password is restored.
  - Popular implementation is a password reset link based on user-controlled data, like the HTTP Host Header.
  - An attacker can hijack the password reset token using a malicious Host Header pointing to his website

Broken Authentication Vulnerabilities

- **Broken Password Reset**

1	Get data.
2	Verify questions.
3	Lock out.
4	Verify code and allow password reset.

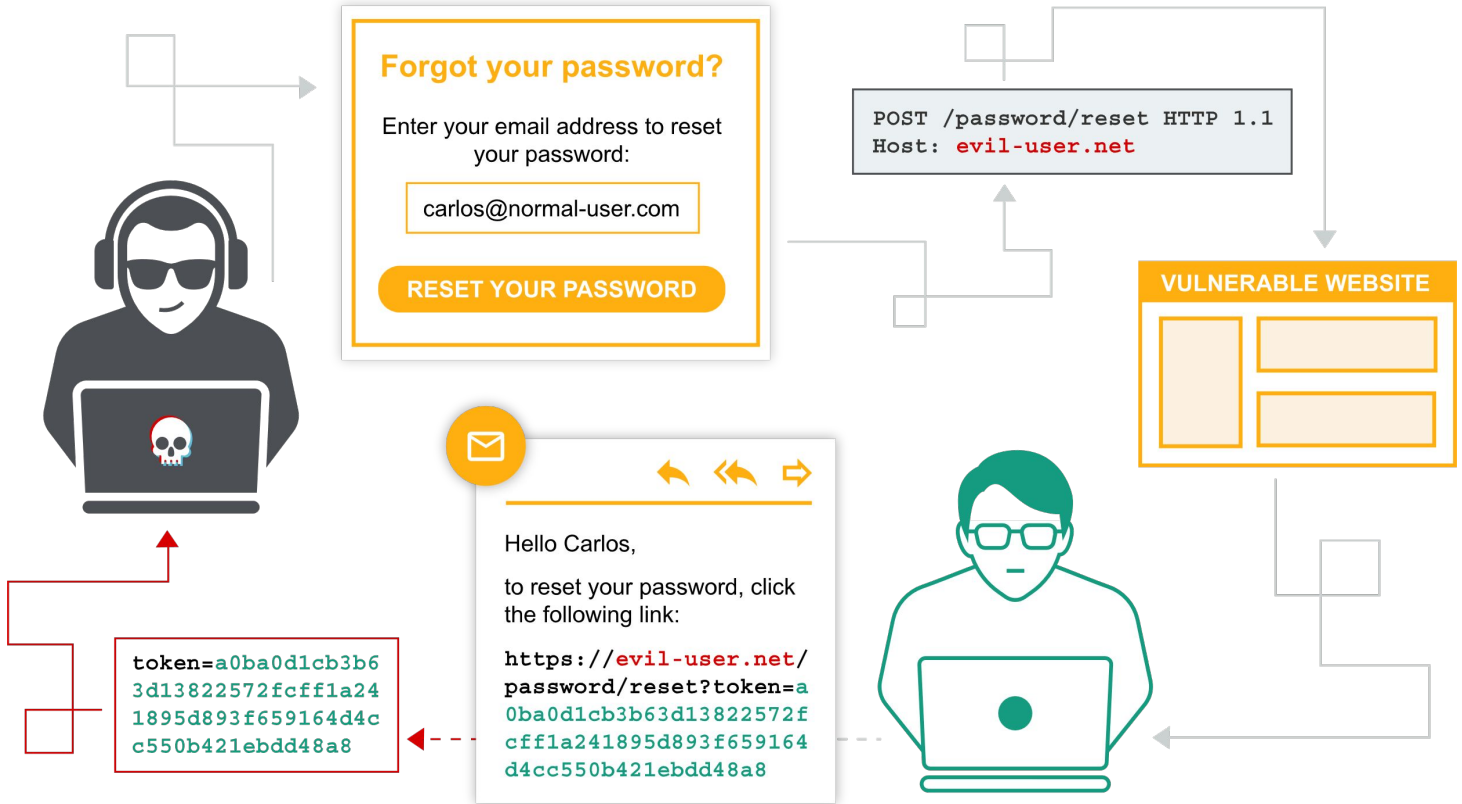
What could go wrong?

Forgot your password?

Broken Authentication Vulnerabilities

- How does a password reset work?

Visit
https://portswigger.net/web-security/host-header/exploiting/password-reset-poisoning



Impact of **Broken Authentication**

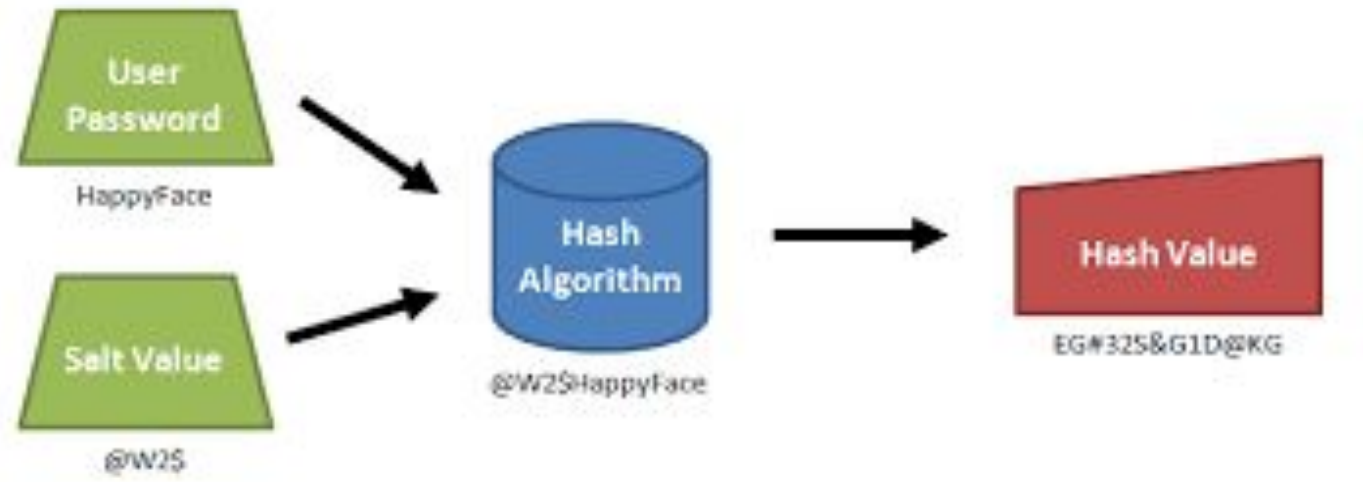
- Target a specific or group of account holders.
- If the attacker is successful, they get full access to the account and can harm the victim in many ways.
- The attacker can cause reputational and financial loss. They can act as an impostor to malign the personal relationships of the victim, too.
- Selling the compromised credentials to the other party is another possibility.
- Example: In 2018, cybercriminals made 30 billion login attempts, highlighting the credential-stuffing attacks. The attacks were automated and performed through miscreant leverage bots [4].

Guidelines to Prevent Broken Authentication

- Tighten Password Policies
- Update Session Management
- Guard Against Attacks

Tighten Password Policies

- **Don’t Store Passwords in Clear text**
  - A strong password storage strategy is critical to mitigate data breaches.
  - Use of hashing and salting to convert password into an arbitrary output which is not reversible.



Tighten Password Policies

- **Don’t Permit Weak Passwords**
  - Weak and common passwords are vulnerable to brute force attack.
  - Enforce a minimum password length of 8 characters and include lower case, uppercase, digit and symbol characters.
  - Regular Scanning to avoid well known passwords (common passwords)

Tighten Password Policies

- **Implement Multi-Factor Authentication (MFA)**
  - OWASP's number one tip for fixing broken authentication is to "implement multi-factor authentication to prevent automated, credential stuffing, brute force, and stolen credential reuse attacks."
  - MFA provides an extra level of security by demanding an additional credential that's harder for attackers to fake, such as a biometric scan or one-time code.

Tighten Password Policies

- **Use Breached Password Protection**
  - Use an identity and access management (IAM) service with breached password protection. When the service discovers a compromised credentials cache, it will notify users. Those users will be locked out until they change their passwords.

- **Limit Failed Login**
  - **lock a user out** if they entered the wrong password more than the specified time.

Tighten Password Policies

- Do not ship or deploy with any default credentials, particularly for admin users.

- Implement weak-password checks, such as testing new or changed passwords against a list of the top 10000 worst passwords.

- Align password length, complexity and rotation policies with NIST 800-63 B's guidelines in section 5.1.1 for Memorized Secrets or other modern, evidence based password policies.

Update Session Management

- **Random session ID with high entropy**
  - Use a server-side, secure, built-in session manager that generates a new random session ID with high entropy after login.
- **Control Session Length**
  - Every web application automatically ends sessions at some point, either after logout, a period of no activity, or a certain length of time
  - A streaming-video service sessions can be weeks long, whereas a banking app should automatically log users out after a few minutes since the risk of a hijacked session is much higher.

Update Session Management

- **Rotate and Invalidate Session IDs**
  - Issue a user with a new session ID after every login.
  - Similarly, sessions and authentication tokens must be immediately invalidated after a session ends, so attackers can't reuse them.
- **Don't Put Session IDs in URLs**
  - Session IDs should not be in the URL, be securely stored and invalidated after logout, idle, and absolute timeouts.
  - There are so many ways that URL rewriting can end up exposing session IDs
  - Use cookies generated by a secure session manager.

Guard Against Attacks

- **Generic Authentication response**
  - The user ID or password was incorrect.
  - The account does not exist.

- **Conduct Workplace Phishing Training**
  - Teaching your workforce how to spot malicious emails. As phishing attacks get more sophisticated, they become harder to spot, and the only way to combat this threat is to keep your staff informed.

Guard Against Attacks

- **Protect against brute force attacks**
  - Implement rate-limiting against automated login attempts to prevent password guessing attacks, so bots can’t flood system.
  - Use captchas, account lockouts or Multi-Factor authentication.
  - Following OWASP Broken Authentication Cheat Sheet for an exhaustive list of security measures which you can apply according to your needs.

Guard Against Attacks

- **Enable logging and monitoring of authentication functions**
  - Identify anomaly and ensure that all account lockouts are logged and reviewed
  - Log all failures and alert administrators when credential stuffing, brute force, or other attacks are detected.

Guard Against Attacks

- **Avoid injection issues**
  - Always validate user input before processing the login request.
  - If there is an injection vulnerability in one of your authentication features, your primary line of defense will be compromised.

Is the Application Vulnerable?

There may be authentication weaknesses if the application:

- Permits automated attacks such as credential stuffing
- Permits brute force or other automated attacks.
- Permits default, weak, or well-known passwords (common passwords), such as “Password1” or “admin/admin“.
- Uses weak or ineffective credential recovery and forgot-password processes, such as “knowledge-based answers”, which cannot be made safe.

Is the Application Vulnerable?

There may be authentication weaknesses if the application:

- Uses plain text, encrypted, or weakly hashed passwords (see A3:2017-Sensitive Data Exposure).
- Has missing or ineffective multi-factor authentication.
- Exposes Session IDs in the URL (e.g., URL rewriting).
- Does not rotate Session IDs after successful login.
- Does not properly invalidate Session IDs. User sessions or authentication tokens (particularly single sign-on (SSO) tokens) aren't properly invalidated during logout or a period of inactivity.

References

- *https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication*
- https://auth0.com/blog/what-is-broken-authentication/
- https://thehackerish.com/broken-authentication-and-session-management-explained/
- https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html
- https://www.thesslstore.com/blog/everything-you-need-to-know-about-broken-authentication/