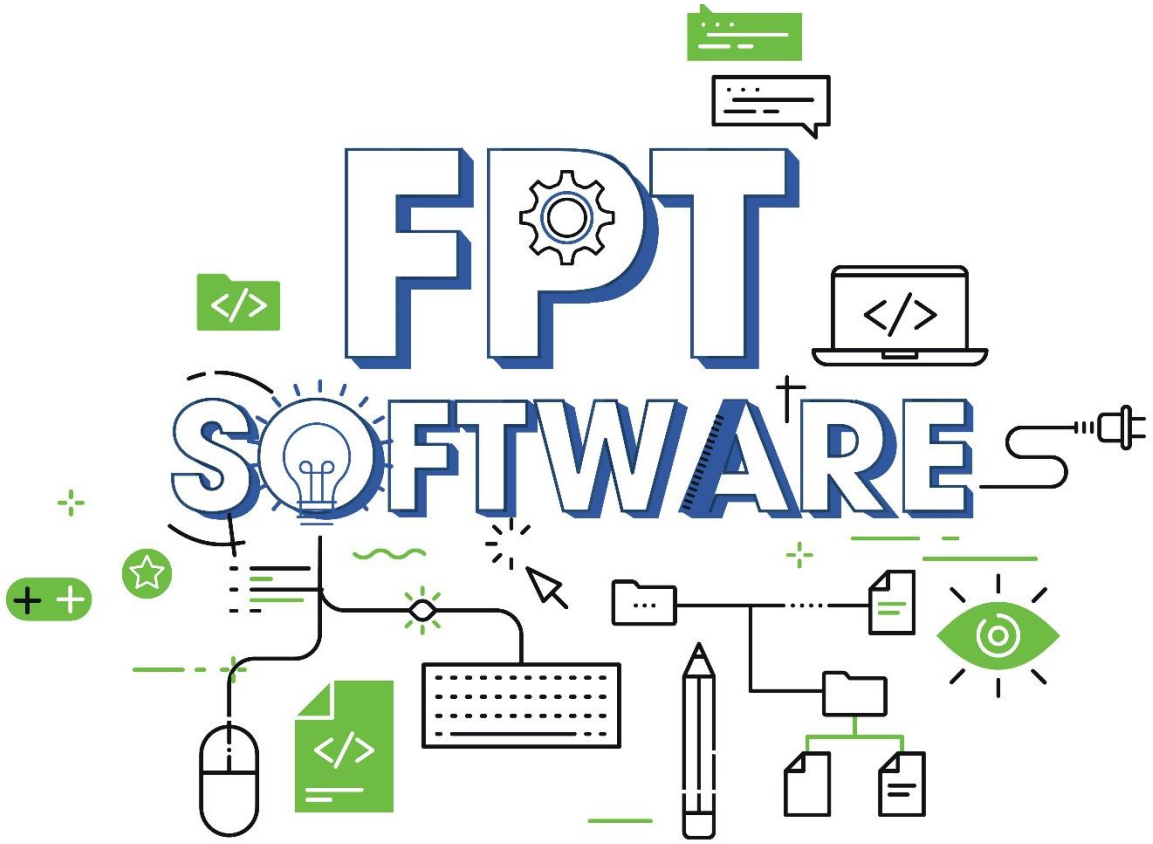
Web Application Security

Broken Authentication



Lesson Objectives

1. Assignment Review
2. Basics of Authentication
3. Authentication vulnerabilities
4. Common Authentication Vulnerabilities
5. Assignment: Web Secure Exercise 1

Section 1

Assignment Review

Assignment Review

Automation Recon

Section 2

Basics of Authentication

Basics of Authentication

Authentication in a web application is the process of verifying the identity of a user who is attempting to access a protected resource, such as a web page, API endpoint, or database.

- ***Basic authentication:*** *This involves the use of a username and password, which are transmitted in clear text and Base64-encoded.*
- ***Form-based authentication:*** *This is a type of authentication where users enter their username and password in a login form provided by the web application.*
- ***Token-based authentication:*** *This is a type of authentication that uses tokens, such as JSON Web Tokens (JWT), to verify the identity of a user.*

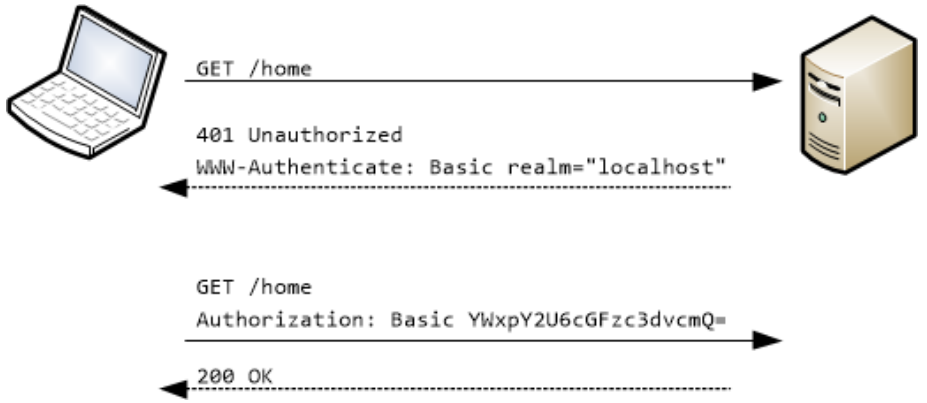
There are also other authentication methods such as ***OpenID***, ***Single sign-on (SSO)***, ***Multi-factor authentication (MFA)***, ***OAuth*** *…*

Basics of Authentication

Basic authentication

Basic authentication *typically involves the following steps:*

1. *User accesses a protected resource*
2. *User enters credentials*
3. *Credentials are sent to the server*
4. *Server verifies the credentials*
5. *User is granted access*

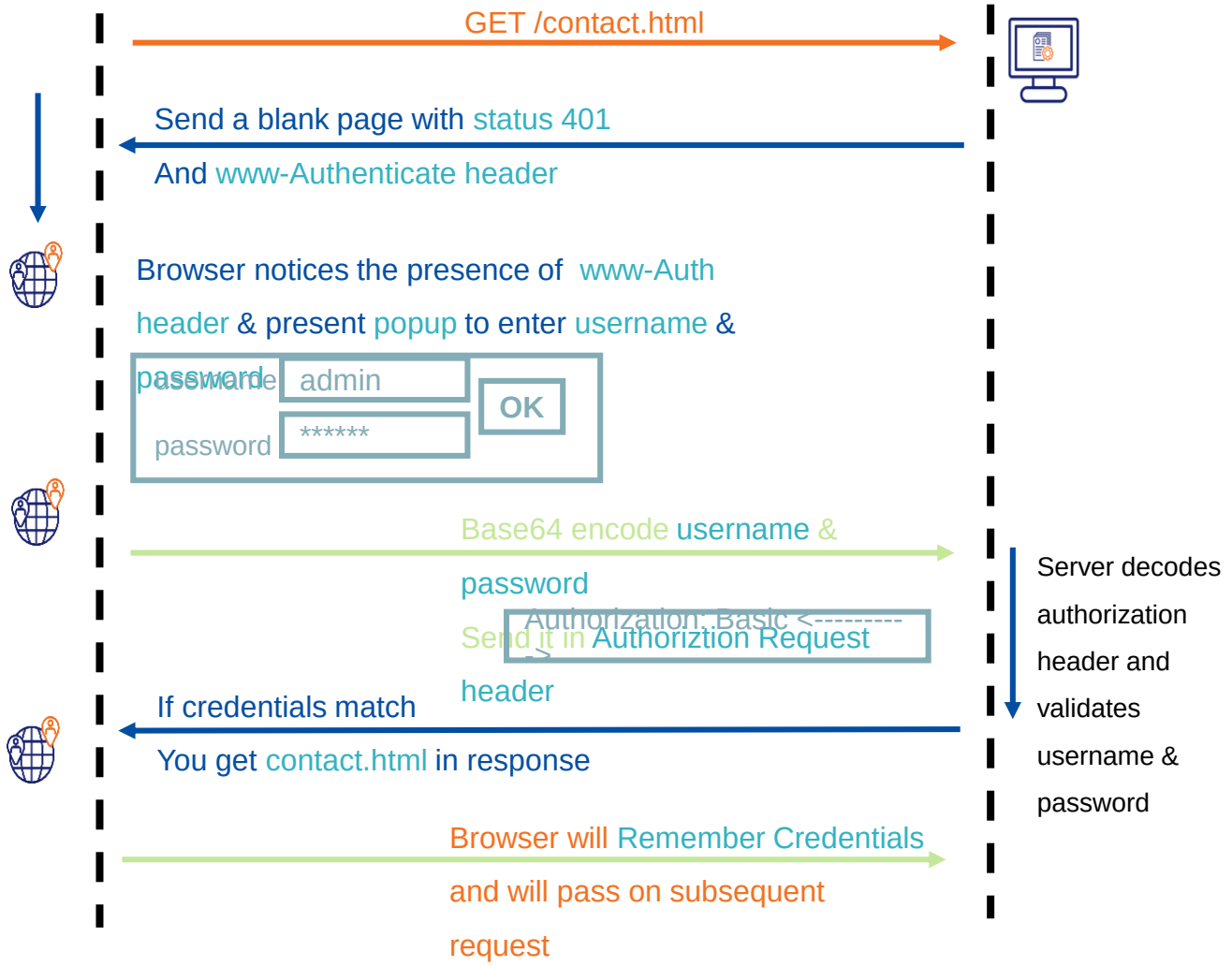


If the user enters invalid credentials, the server will return an HTTP 401 Unauthorized status code, along with a WWW-Authenticate header that specifies the authentication realm and type.

Basics of Authentication

Form-based authentication

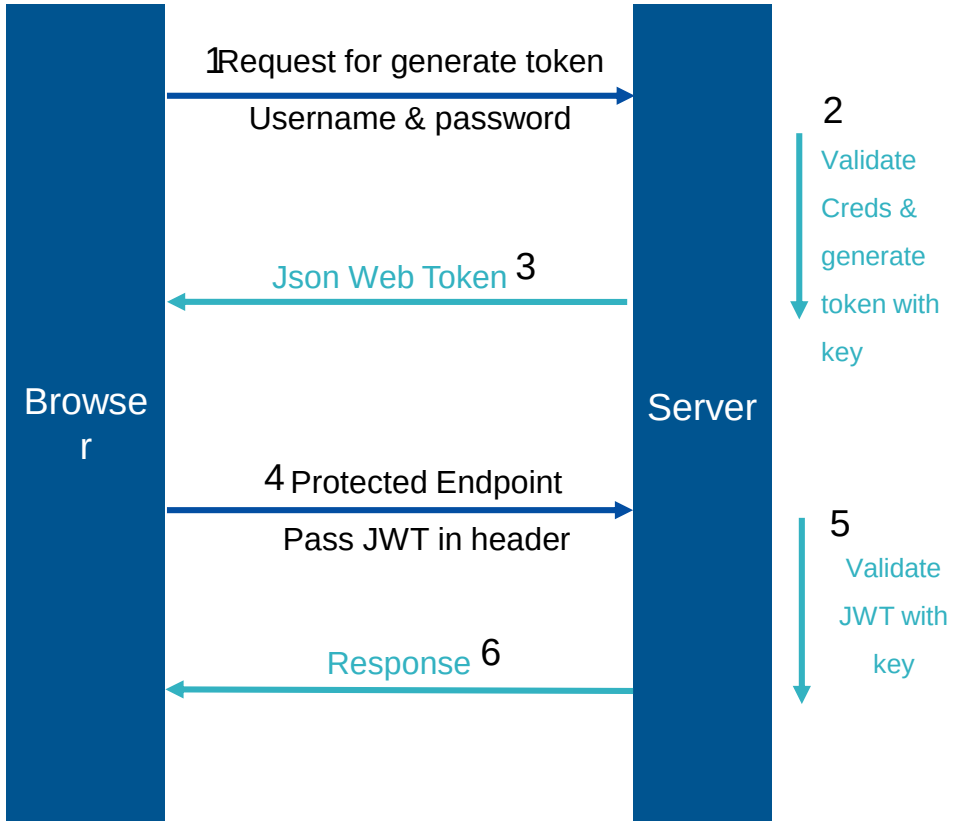
- *Credentials in header are base64 encoded, not encrypted.*
- *Basic Auth with HTTP is not safe as data is sent in plaintext. HTTPS is recommended if Basic Auth is the only option.*
- *Basic Auth Sends the Credential as encoded text in Authorization header. But it is a form of authentication, not authorization.*
- *Client (Browser) will ask for credentials if and only if server sends back www-authenticate Header in response with Unauthorized Status 401.*



Basics of Authentication

Token-based authentication

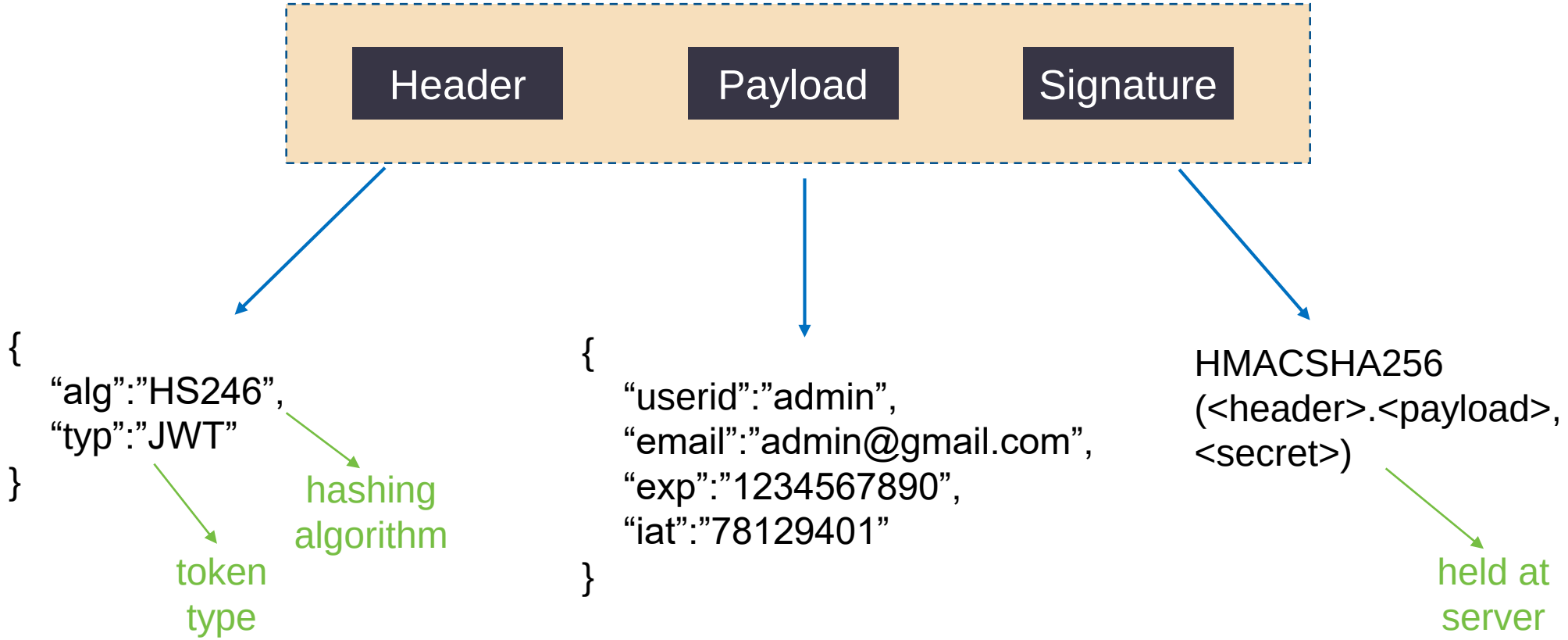
- JWT can be used both for Authentication and Authorization.
- Anyone who has **access to secret key** can do verification.
- If Asymmetric Key Algorithm is used, **Private Key** is used to sign JWT and anyone who have access to **Public Key** can verify.



Basics of Authentication

Token-based authentication

Structure of a JWT:



Section 3

Authentication vulnerabilities

Authentication vulnerabilities

Description

Broken Authentication is an application security risk that can allow malicious actors to compromise keys, passwords, and session tokens, potentially leading to further exploitation of users' identities and in the worst case, complete control over the system.

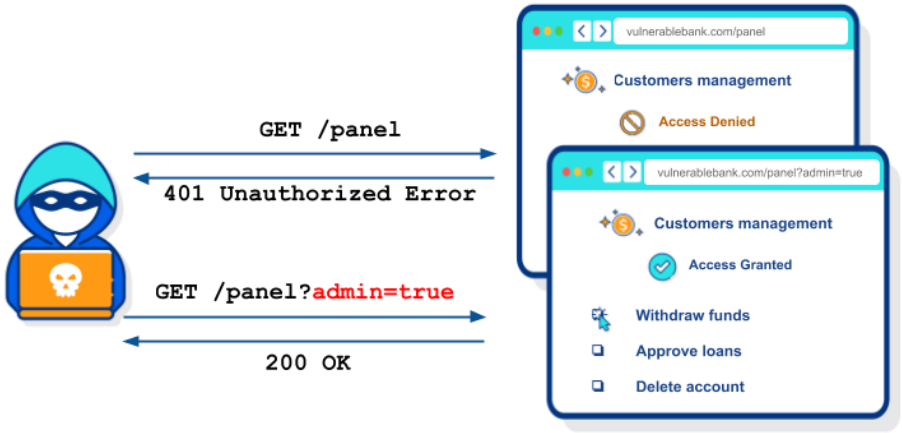
The vulnerability boils down to an attacker being able to bypass the authentication mechanism of the vulnerable application due to poor configuration, logic errors, or bugs in the software.

It has been categorized as a critical risk affecting web applications on the OWASP Top 10 since 2013.

Authentication vulnerabilities

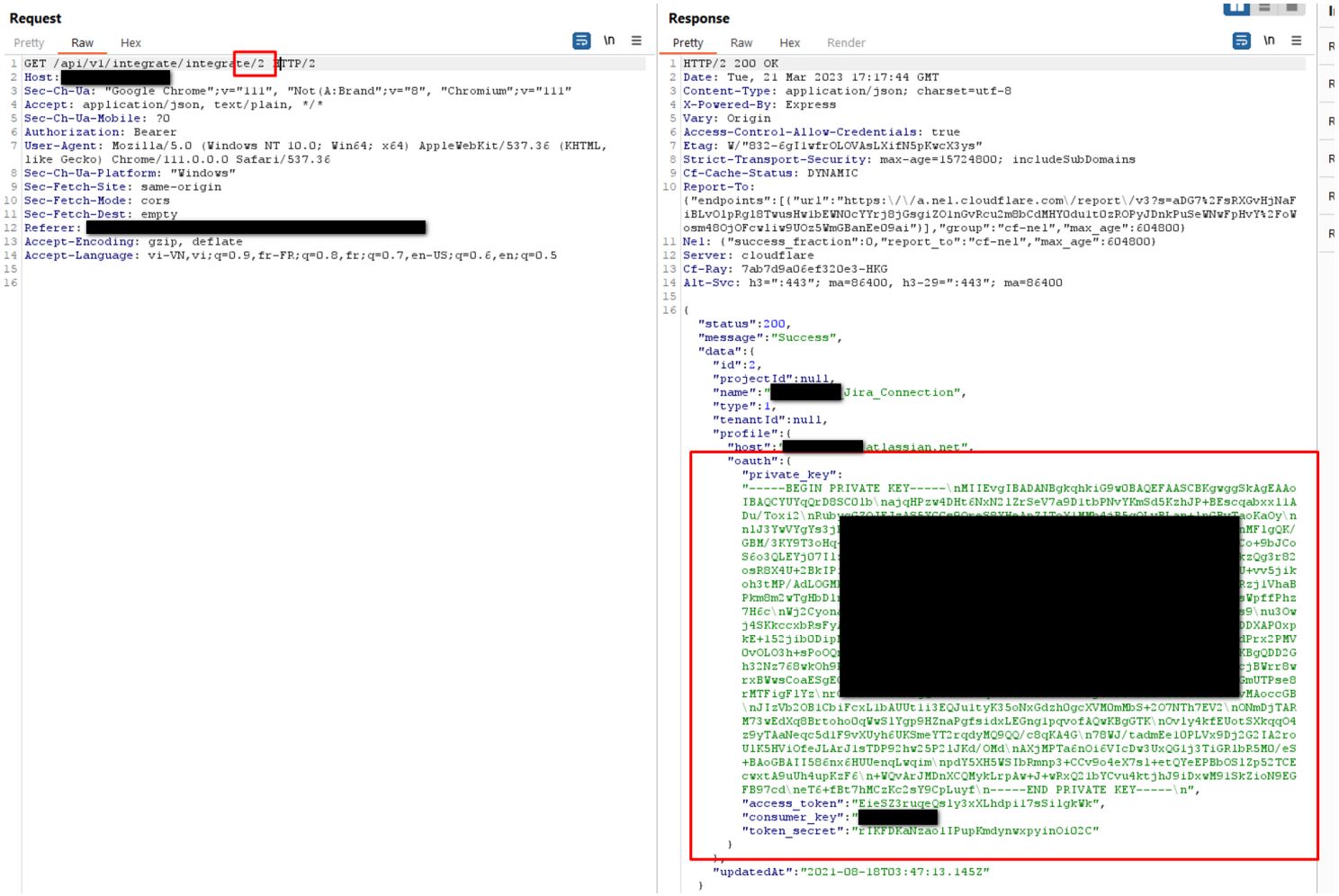
Impact

A successful attack can result in a malicious attacker gaining complete access to all data in the web application, assuming administrator rights, and compromising the confidentiality, integrity, and availability of the application.



Authentication vulnerabilities

Impact



Authentication vulnerabilities

Scenarios:

- *Functionalities requiring authentication lack mechanisms or implement insufficient protections.*
- *Broken object-level protection mechanisms allow unauthenticated users to access private resources.*
- *The use of dictionary-based attacks or credential reuse on applications that permit automated attacks.*
- *The application permits the use of weak passwords, such as "password123" or "123456".*

Authentication vulnerabilities

Prevention

- *Prevent username enumeration.*
- *Implement an effective password policy that disallows the use of weak or overused/common passwords.*
- *Implement robust brute-force protection.*
- *Triple-check your verification logic.*
- *Implement proper multi-factor authentication.*

Authentication vulnerabilities

Prevention

- *Handle Credentials Secretively*
- *Validate Credentials Properly*
- *Prevent Information Leakage*
- *Log, Monitor, and Notify*
- *Prevent Misuse of the Password Change Function*
- *Prevent Misuse of the Account Recovery Function*

Section 4

Common Authentication Vulnerabilities

Common Authentication Vulnerabilities

- Bad Passwords
- Brute-forcible Login
- Verbose Failure Messages
- Vulnerable Transmission of Credentials
- Password Change Function
- Forgot Password Function
- “Remember Me” Function
- Incomplete Validation of Credentials
- Predictable Initial Passwords
- Insecure Distribution of Credentials

Common Authentication Vulnerabilities

Bad Passwords

- *Very short or blank.*
- *Common dictionary words or names.*
- *The same as the username.*
- *Still set to a default value.*

Common Authentication Vulnerabilities

Bad Passwords / Hack steps

Attempt to discover any rules regarding password quality:

1. *Review the website for any description of the rules.*
2. *If self-registration is possible, attempt to register several accounts with different kinds of weak passwords to discover what rules are in place.*
3. *If you control a single account and password change is possible, attempt to change your password to various weak values.*

Common Authentication Vulnerabilities

Brute-forcible Login

- *If application has no anti brute-force mechanisms, attacker can have password of any user if he has unlimited compute power.*
- *Simple but very powerful method to compromise application.*
- *Admin password may be simpler than normal account.*

Common Authentication Vulnerabilities

Brute-forcible Login / Hack steps

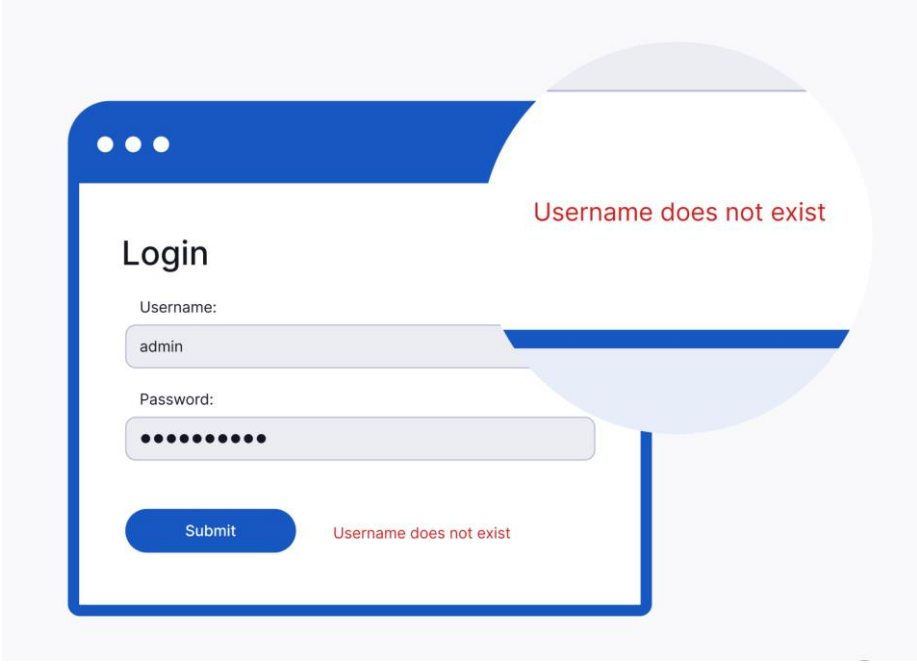
1. *Submit bad login attempts for an account you control.*
2. *Attempt to log in after 10 times submit wrong credentials.*
3. *If cannot log in:*
   - *Repeat with each cookie per login try. Check if server only locks session.*
   - *Compare app's behavior between right password with wrong password.*

Common Authentication Vulnerabilities

Verbose Failure Messages

Application may have different error message whether username or password is incorrect.

Enumerate user:



Common Authentication Vulnerabilities

Verbose Failure Messages / Hack steps

1. *Try login with valid username and incorrect password, compare with random username (ensure it's not existing).*
2. *Record and compare every detail of server's responses.*
3. *Try other location of application where can submit username.*
4. *Write script to enumerate username if possible.*

Common Authentication Vulnerabilities

Vulnerable Transmission of Credentials

1. *If no HTTPS: Attacker can eavesdrop traffic – **MITM.***
2. *If HTTPS is present:*
   - *Credentials are transmitted as query string parameters.*
   - *Credentials in URL through 302 redirects.*
   - *Credentials stored in cookies.*

Common Authentication Vulnerabilities

Vulnerable Transmission of Credentials / Hack Steps

1. *Monitor all traffic in both directions through authentication process.*
2. *If no cases of actual credentials being transmitted insecurely are identified, pay close attention to any data that appears to be encoded or obfuscated. If this includes sensitive data, it may be possible to reverse engineer the obfuscation algorithm.*

Common Authentication Vulnerabilities

Password Change Function

- *Provide a verbose error message indicating whether the requested username is valid.*
- *Allow unrestricted guesses of the “existing password” field.*
- *Check whether the “new password” and “confirm new password” fields have the same value only after validating the existing password.*

Common Authentication Vulnerabilities

Password Change Function / Hack steps

1. *Identify any password change functionality within the application.*
2. *Make various requests to the password change function using invalid usernames, invalid existing passwords, and mismatched "new password" and "confirm new password" values.*
3. *Try to identify any behavior that can be used for username enumeration or brute-force attacks.*

Common Authentication Vulnerabilities

Forgot Password Functionality

Security Question/Challenge:

- *Disclose the existing, forgotten password to the user after successful completion of a challenge.*
- *Immediately drop the user into an authenticated session after successful completion of a challenge.*
- *Send reset password URL to an e-mail address specified by the user at the time the challenge is completed.*
- *Reset password's value directly after successful completion of a challenge and do not send any e-mail notification to the user.*

Common Authentication Vulnerabilities

Forgot Password Functionality / Hack steps

1. *Identify and understand forgot password function by using account you control.*
2. *For security question challenges, try guess answer.*
3. *Username enumeration.*
4. *Guess reset password pattern.*

Common Authentication Vulnerabilities

"Remember Me" Functionality

- *Implemented using a simple persistent cookie, such as*
  - *RememberUser=daf – Username*
  - *RememberUser=1328 – Session identifier*
- *Can be captured through a bug such as XSS*

Common Authentication Vulnerabilities

"Remember Me" Functionality / Hack steps

1. *Activate "remember me" function, check how long it can remember user session.*
2. *Check cookies and other storage to find out how "remember me" work, what identifier is used.*
3. *Predict how identifier is generated.*
4. *Guess and try modify identifier.*

Common Authentication Vulnerabilities

Incomplete Validation of Credentials

- *Truncate passwords and validate only first n characters.*
- *Case-insensitive check password.*
- *Strip unusual characters.*

Common Authentication Vulnerabilities

Incomplete Validation of Credentials / Hack steps

Attempt to login with account you control, modify password:

- *Remove last character*
- *Changing the case of character*
- *Remove special typographical characters*

Common Authentication Vulnerabilities

Predictable Initial Passwords

- *Usually happened with internal applications.*
- *Users are created all at once or in sizeable batches and are automatically assigned initial passwords.*
- *Generated passwords may be similar or contain sequences that could be identified or guessed easily.*

Common Authentication Vulnerabilities

Predictable Initial Passwords / Hack steps

- *If application allow registration and return password that looks like default password, attempt to register multiple accounts to verify .*
- *If password looks like correlated with username, try login using known or guessed usernames with corresponding inferred passwords.*

Common Authentication Vulnerabilities

Insecure Distribution of Credentials

Many applications send newly created account or activation URL to users via out-of-band channels (email, SMS …).

- *OOB channels is out of control.*
- *Activation URL is not expired.*
- *Activation URL is predictable.*

Common Authentication Vulnerabilities

Insecure Distribution of Credentials / Hack steps

1. *Register new account, check if application distributes credentials by OOB channels,*
2. *If activation URL is used, try register multiple accounts and identify pattern of activation URL,*
3. *Reuse activation URL multiple times, before and after log in, log out, change password,*

Section 5

Assignment: Web Secure Exercise 1

Lesson Summary

Authentication functions are perhaps the most prominent target in a typical application's attack surface.

- *The front line of defense against unauthorized access.*
- *If broken, they grant access to protected functionality and sensitive data.*
- *At the other end of the spectrum, defects may be very hard to uncover.*

References

- *The web application hackers handbook*
- *SecureFlag Knowledge Base*
- *Web Security Academy*

THANK YOU!