Authentication Vulnerabilities

# Definition of Authentication

**Authentication** is the process of verifying the identity of a user or client. Websites are potentially exposed to anyone who is connected to the internet. This makes robust authentication mechanisms integral to effective web security.

There are three main types of authentication:

* Something you **know**, such as a password or the answer to a security question. These are sometimes called "knowledge factors".
* Something you **have**, This is a physical object such as a mobile phone or security token. These are sometimes called "possession factors".
* Something you **are** or do. For example, your biometrics or patterns of behavior. These are sometimes called "inherence factors".

# What is Authentication Vulnerability

An **authentication vulnerability** is a security flaw that allows an attacker to bypass or weaken the authentication process, enabling unauthorized access to a system or user account. This is some of the most common vulnerabilities in authentication:

* Vulnerabilities in password-based login
* Vulnerabilities in multi-factor authentication
* Vulnerabilities in other authentication mechanisms

# Write-up labs

## Username enumeration via response timing

* Payload: username=?&password=’a’\*100 , if a username in database, response time will longer than others.

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, Sơ đồ

Nội dung do AI tạo ra có thể không chính xác.

* Server implemented rate-limit to incorrect login attempts, so I tried to use header X-Forwarded-For to bypass

Ảnh có chứa văn bản, số, hàng, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

* With username= ftp, we brute-force password and get password=jennifer

## Broken brute-force protection, IP block

* After 3 or more fail attempts, server will block your IP. But after a true login, your IP will be enabled
* While brute-forcing password, after a trial attempt we send a true attempt

## Username enumeration via account block

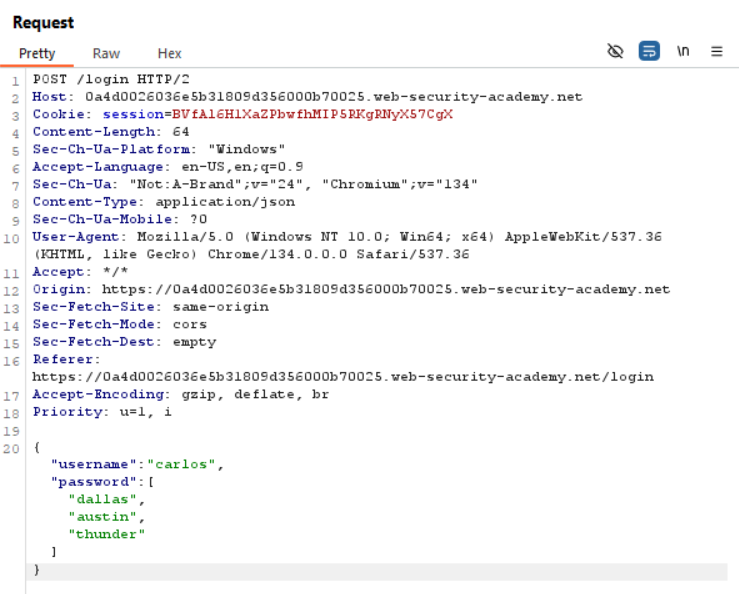
* An account will be blocked after few fail requests. With each username we send with a random password, then base on error message in response we can know if that username exists

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Nội dung do AI tạo ra có thể không chính xác.

## Broken brute-force protection, multiple credentials per request

* We see that format of sent data in json format. So I try to set parameter password as a list of passwords



* Applying binary search algorithm, we can find the correct password.

## 2FA simple bypass

* After provide valid username and password, we can access other routes successfully without enter correct code sent to that user’s email address

## Offline password cracking

* Feature comment in this lab is vulnerable to XSS at paramter responseBody

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

* After I got hash password of carlos, I went to this website to decode md5 hash: <https://md5decrypt.net/en/> .

Ảnh có chứa văn bản, Phông chữ, Hình chữ nhật, hàng

Nội dung do AI tạo ra có thể không chính xác.

## Password reset broken logic

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

## Password reset poisoning via middleware

* The main difference between above lab and this is token is generated dynamically or each token is to reset only 1 account. When I request to /forgot-password , an email with a url (contain a reset-code) sent to me. With that token, we can reset any account exist in system. After searching for documentation of this attack, I found header X-Forwarded-Host (identify the original host requested by a client when a request passes through a proxy or load balancer). If we set value to our exploit domain, server will trust and send an url with our domain

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Nội dung do AI tạo ra có thể không chính xác.

* If client click this url, it will send a request contain reset-password-toke to our server

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Nội dung do AI tạo ra có thể không chính xác.

## Password brute-force via password change

* In request to change password, we can modify username to account we want to change password

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Nội dung do AI tạo ra có thể không chính xác.

* With login session, we can reset password of any account => brute-force password of carlos

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Nội dung do AI tạo ra có thể không chính xác.

## 2FA Broken Logic

* After entering true username and password, I was redirected to /login2 page to enter mfa-code. I noticed the value of header Cookie: verify=wiener; session=2xFrR4lMgS8t5G0XIflFBRyoEV2AWSYx  . If I enter the true mfa-code without login first, I still can access my account.
* So I change the value of verify to carlos then brute-force code. If success status code is 302 and I will get a new session with carlos account.

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.  
Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Nội dung do AI tạo ra có thể không chính xác.

## 2FA bypass using a brute-force attack

* In this lab, we are given username and password and we must brute-force mfa-code to login. After 2 failed mfa-code, our session will be logged out. Futhermore, in a post request, we must contain csrf\_token, both in /login and /login2
* The process to attack here is:
* Make a get request to /login to get csrf\_token
* Then make a post request to /login with above username, password, csrf\_token
* Then we will be redirected to /login2 to get another csrf\_token
* Now we can brute-force mfa-code. If our account is logged out, repeat from step 1
* After serveral trials, we can login carlos account.

## Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, số Nội dung do AI tạo ra có thể không chính xác. Ảnh có chứa văn bản, ảnh chụp màn hình, số, phần mềm Nội dung do AI tạo ra có thể không chính xác. 3.12 Brute-forcing stay-logged-in cookie

* After entering true password and choose option “stay login”, server send back with a cookie:

Ảnh có chứa văn bản, phần mềm, Trang web, số

Nội dung do AI tạo ra có thể không chính xác.

* This cookie is base64 encoded. I decoded and know the structure : wiener:51dc30ddc473d43a6011e9ebba6ca770 . The second part is md5hash of something, then I decoded it and got that hash equals to md5(peter)   . Then we can brute-force to get password of carlos