CYBERTECH CLUB @ PSU | CTF WORKSHOP SERIES

MODULE 04

Broken Authentication

Participant Handout • Spring 2026

HOW TO USE THIS HANDOUT

This packet mirrors the live workshop deck so you can study async.

1. Hook

SET THE SCENE

If a login system lets you try unlimited passwords or uses weak checks, anyone can break into accounts just by guessing or reusing credentials.

We are going to do exactly that, on a real PortSwigger lab, using nothing but a browser.

2. Learning Objectives

By the end of this topic, a student can:

- ☐ **Identify** common authentication weaknesses in a login system.
- ☐ **Bypass** authentication using weak or misconfigured login mechanisms.
- ☐ **Exploit** authentication flaws such as missing rate limits or predictable credentials.

3. Core Concept

Authentication, broken

Authentication verifies who a user is before granting access to an application. A vulnerability occurs when this process is weak or improperly enforced, allowing attackers to log in without valid credentials.

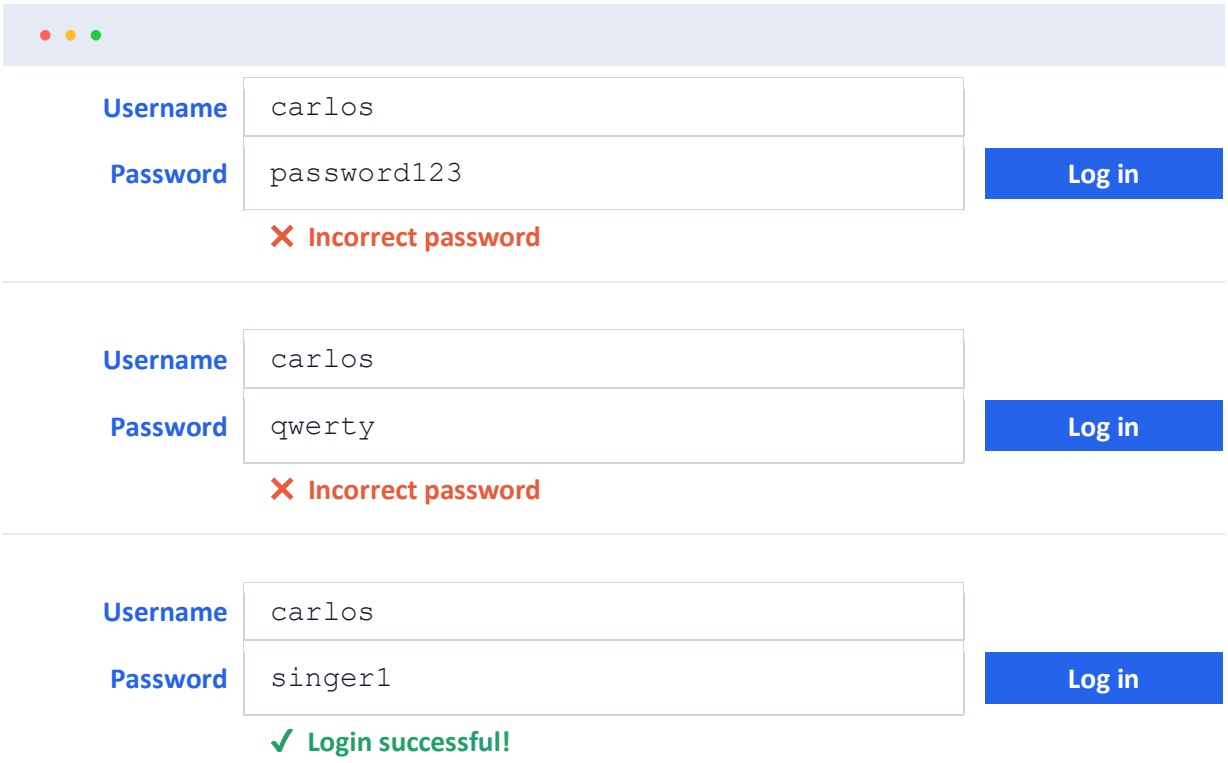
How attackers take advantage

Common issues include weak passwords, unlimited login attempts, predictable credentials, and missing verification checks. Instead of breaking the system, attackers exploit poor controls: they guess repeatedly or try credentials leaked from other sites.

The rule to remember

If authentication is not strictly enforced, anyone can become a legitimate user by imitating one. Rate limits, strong password policies, and multi-factor authentication are the baseline, not a bonus.

FIGURE 1: REPEATED GUESSES EVENTUALLY LAND ON A WEAK PASSWORD



Repeated login attempts eventually succeed because the system does not limit or properly validate authentication attempts. The attacker is not breaking cryptography; they are walking through an open door.

4. How It Works

Four steps that match how a tester probes a login flow for broken authentication.

STEP 01

Submit Login Credentials

Start with a normal request.

- User enters a username and a password into the login form.
- The application sends a request to the server for verification.

STEP 02

Validate Credentials Weakly

Find out how strict the server is.

- Server checks credentials against stored values, often without defense in depth.
- Weak controls (no rate limit, no lockout, simple password policy) allow repeated attempts.

STEP 03

Attempt Multiple Logins

Exploit the absence of a limit.

- Attacker guesses common passwords: password, 123456, qwerty, company name.
- No protections (CAPTCHA, delay, lockout, notification) stop the attempts.

STEP 04

Gain Unauthorized Access

Confirm the login lands and stays.

- Correct credentials are eventually guessed and the session is established.
- Application grants access without detecting the abuse pattern.

5. Demo Walkthrough

Run this end to end on a PortSwigger authentication lab. The live demo follows the same script.

Prerequisites

LAB	TOOLS	STATE
Authentication (weak password)	**Web browser**	**Lab launched, login page visible**

Lab URL: https://portswigger.net/web-security/authentication

Step 1: Attempt login with a common credential pair

Action. On the lab login page, submit username carlos with password password123.

Expected. Login fails with an incorrect password message. The form is still available for another try.

Step 2: Try another weak or common password

Action. Keep the same username and try qwerty, 123456, and password in sequence.

Expected. Login eventually succeeds because the site has no rate limit and carlos is using a weak password.

Step 3: Confirm access

Action. Navigate to the account dashboard or profile page.

Expected. You are logged in as carlos. The lab marks the challenge as solved.

What the requests look like

```
# Attempt 1 (fails)
POST /login HTTP/1.1
username=carlos &password=password123

# Attempt N (succeeds, no rate limit)
POST /login HTTP/1.1
username=carlos &password=123456
```

6. Common Pitfalls

When something does not work, check these first before asking for help.

SYMPTOM

Login never succeeds

Cause: Using the wrong username or skipping part of the lab instructions.

Fix: Use the exact username provided in the lab (usually carlos) and stick to the listed login form.

SYMPTOM

Students stop after one failed attempt

Cause: Not realizing multiple attempts are allowed here.

Fix: Encourage repeated guessing: real sites without rate limits let you try hundreds of passwords.

SYMPTOM

Students think the attack is unrealistic

Cause: Underestimating how often weak or reused passwords are used in production.

Fix: Emphasize real breaches driven by leaked credential dumps and password reuse.

SYMPTOM

Account locks after a few tries

Cause: The lab has a lockout policy, or the attacker is hitting a CAPTCHA.

Fix: Slow down, switch usernames, or check whether the lab requires a different attack path entirely.

7. Your Challenge

5 – 10 MIN **BEGINNER**

THE TASK

Use the provided lab to log in to an account by guessing weak or common passwords.

SUCCESS CRITERION

Submit a screenshot showing a successful login to the target account (dashboard or profile visible).

Hints

1. Try common passwords that users often reuse.
2. Notice there is no visible limit on login attempts.
3. Use simple passwords like 123456, password, or qwerty.

8. Cheatsheet

Keep this page next to you while you practice. Payloads and patterns only, no prose.

COMMAND / PAYLOAD	WHAT IT DOES	WHEN TO USE
`carlos:password`	Common credential pair.	Test for weak or default passwords.
`carlos:123456`	Common weak password.	Brute-force simple logins.
`carlos:qwerty`	Keyboard pattern password.	Try predictable passwords.
`admin:admin`	Default admin credentials.	Check for default accounts.
`username=*&password=*`	Wildcard attempt (basic bypass cases).	Test weak validation logic.
`Multiple login attempts`	Repeated credential guessing.	When no rate limiting is present.

Basic login request

```
# Basic login request
POST /login HTTP/1.1
Host: <lab-id>.web-security-academy.net
Content-Type: application/x-www-form-urlencoded

username=carlos &password=123456
```

Default credential test

```
# Default credential test
POST /login HTTP/1.1
Host: <lab-id>.web-security-academy.net
Content-Type: application/x-www-form-urlencoded

username=admin &password=admin
```

9. Further Reading

Go deeper on your own time with these three resources.

01

PortSwigger: Authentication

Hands-on labs demonstrating real authentication flaws and attack techniques.

https://portswigger.net/web-security/authentication

02

OWASP Top 10: Identification and Authentication Failures

Overview of common authentication weaknesses and their impact in modern systems.

https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/

03

OWASP Authentication Cheat Sheet

Practical guidelines for implementing secure authentication mechanisms.

https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html