

Sri Lanka Institute of Information Technology



Broken Authentication Vulnerability -

Report 05

IT23187214

Web Security - IE2062

Vulnerability Title:


Broken Authentication Vulnerability

Vulnerability Description:

I found this program on the hackerone Bug hunting website. The website hosted at <https://www.syfe.com>. Broken Authentication occurs when attackers can compromise authentication mechanisms through brute force, credential stuffing, weak session handling, or bypassing login logic. This can result in unauthorized access to user accounts or administrative panels.

In this test, I used **Burp Suite** to attempt logging in with multiple usernames and passwords to simulate a brute-force and credential-based attack. The application consistently denied access and responded uniformly, indicating a properly secured authentication system.

New user promo: Get 6% p.a. returns for 30 days on the first \$10,000 [Learn more](#) X

 INVEST ABOUT RESOURCES SUPPORT LEARN [LOG IN](#) [GET STARTED](#) English | SG


NEW CLIENT EXCLUSIVE

FIXED RETURNS with Cash+ Flexi

For a limited time, new Syfe customers can enjoy assured returns of 6% p.a. returns for 30 days on the first \$10,000 by signing up with promo code **FLEXI6**.

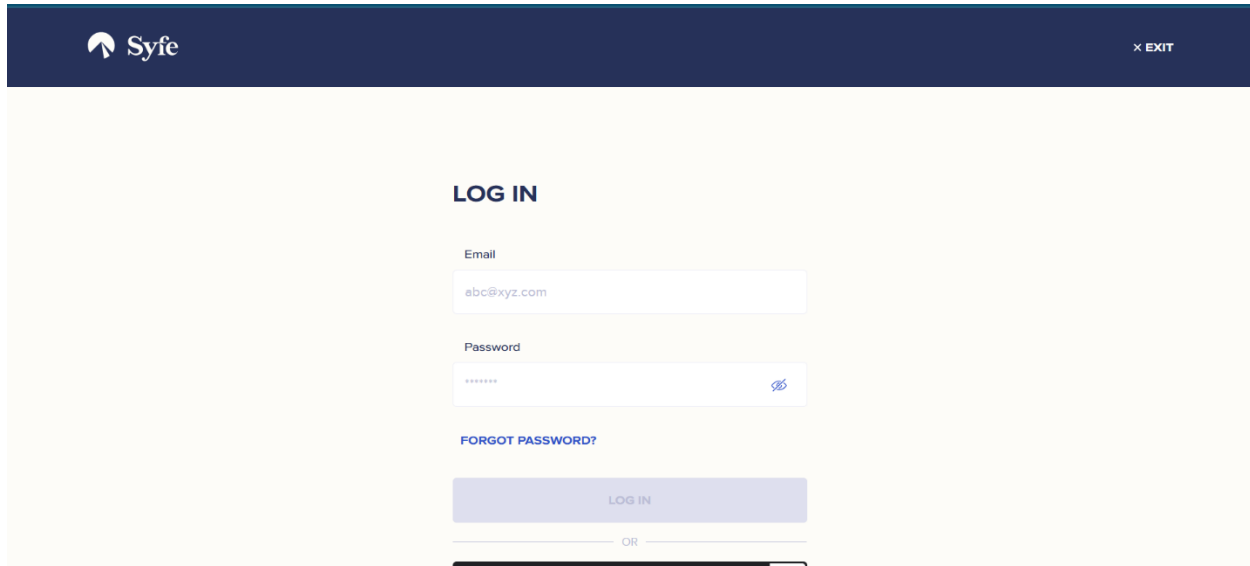
Withdraw any time with no minimum and no lock-ins.

Terms and conditions apply. [Learn more here.](#)



Affected Components:

- Login Endpoint
(e.g., /login, /authenticate, POST /signin)

A screenshot of the Syfe login page. The page has a dark blue header with the Syfe logo on the left and a '× EXIT' link on the right. The main content area is light yellow. In the center, there is a 'LOG IN' section. It includes an 'Email' input field with the placeholder 'abc@xyz.com', a 'Password' input field with a masked password '*****' and a toggle icon, a 'FORGOT PASSWORD?' link, a 'LOG IN' button, and an 'OR' separator. Below the 'OR' separator, there is a partially visible 'Sign up' button.

Impact Assessment:

- Risk Level: **High**

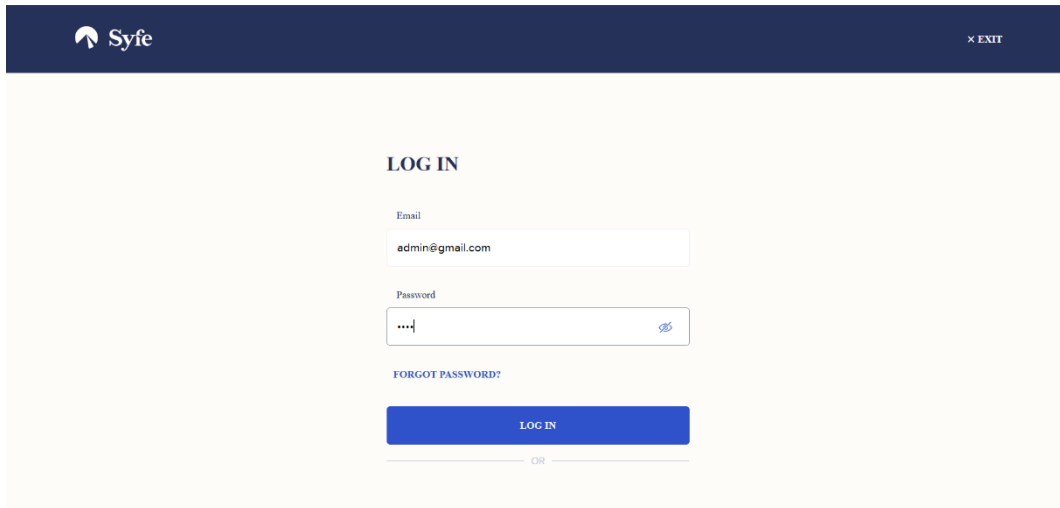
If vulnerable, attackers could:

- Gain unauthorized access to user/admin accounts
- Perform account takeover
- Access sensitive user data
- Escalate privileges in the system

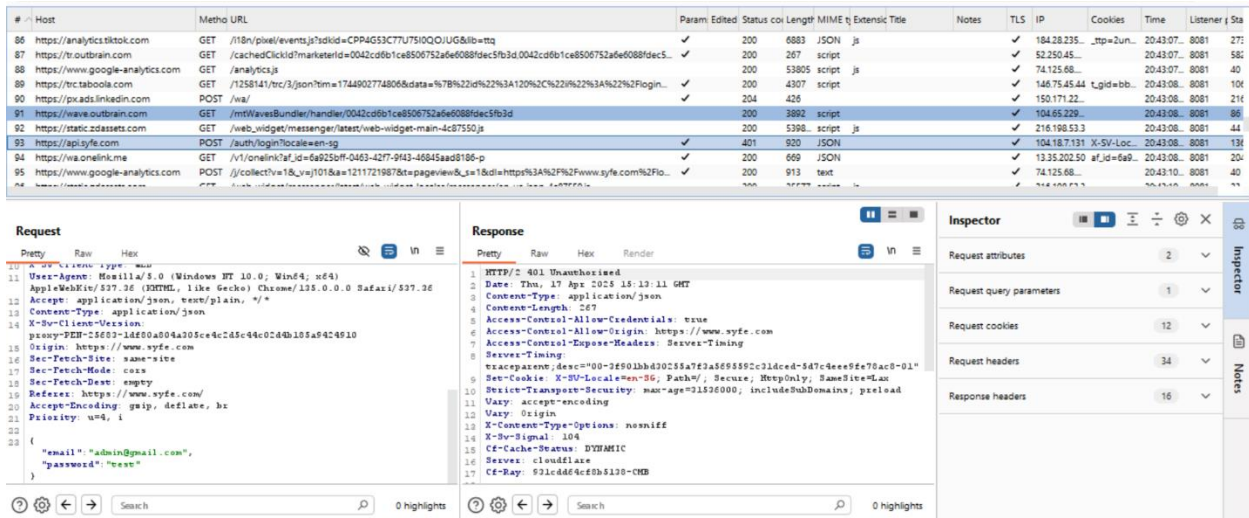
In this case, all login attempts failed, indicating **strong password validation**, consistent response handling, and likely use of **rate limiting or detection systems**.

Steps to Reproduce:

1. Intercepted a valid login request using **Burp Suite**.



2. Sent the request to **Burp Intruder** and marked the email and password fields for attack.



The screenshot displays the Burp Suite interface. The top panel shows a list of HTTP requests. The selected request is a POST to `https://www.syfe.com/api/login`. The 'Request' tab is active, showing the raw HTTP request. The 'Response' tab is also visible, showing the raw HTTP response. The 'Inspector' panel on the right shows the request attributes, query parameters, cookies, headers, and response headers.

Request

```
POST /api/login HTTP/1.1
Host: www.syfe.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Content-Type: application/json
X-Syfe-Client-Version: proxy-25f03-144f0a04a305ce4c2d5c44c02d4b105a94c4910
Origin: https://www.syfe.com
Sec-Fetch-Site: same-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://www.syfe.com/
Accept-Encoding: gzip, deflate, br
Priority: u=4, i

{"email": "admin@gmail.com", "password": "test"}
```

Response

```
HTTP/1.1 401 Unauthorized
Date: Thu, 17 Apr 2025 15:10:11 GMT
Content-Type: application/json
Content-Length: 267
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: https://www.syfe.com
Access-Control-Expose-Headers: Server-Timing
Server-Timing: racepaxent,desc="00-26901bbd30255a7f3a5658592c31dced-5d7c4ee9fe70ac8-01"
Set-Cookie: X-Syfe-Local=en-SG; Path=/; Secure; HttpOnly; SameSite=Lax
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
Vary: accept-encoding
Vary: Origin
X-Content-Type-Options: nosniff
X-Syfe-Signal: 104
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Server: cloudflare
CF-Ray: 631cd4664c69b5130-CHB
```

Inspector

- Request attributes: 2
- Request query parameters: 1
- Request cookies: 12
- Request headers: 34
- Response headers: 16

3. Used a list of test usernames and common password payloads:

- admin@syfe.com, user@ syfe.com, test@ syfe.com
- 123456, password, admin123, qwerty, letmein

5. Launched the attack and monitored server responses for:

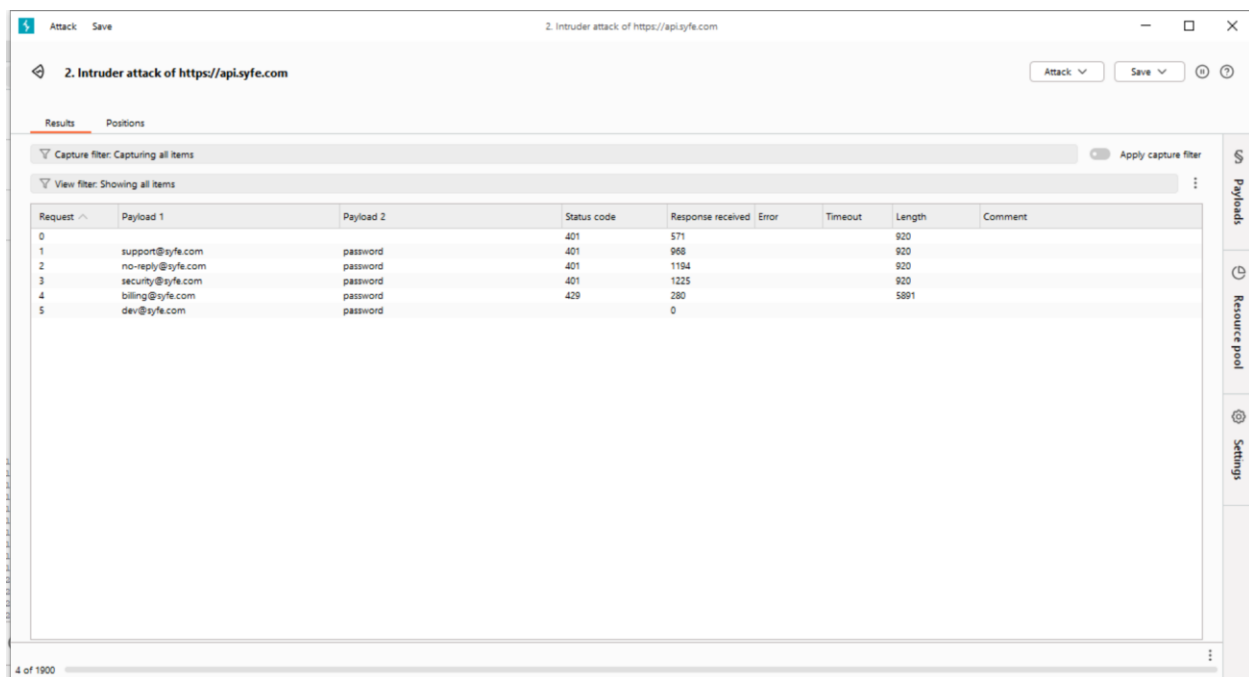
- HTTP status codes (200, 302, 403)
- Response length differences
- Redirects or error messages indicating a successful login

Proof of Concept (PoC):

To test for Broken Authentication vulnerabilities, I used Burp Suite's Intruder tool to simulate brute-force login attempts on the endpoint <https://api.syfe.com>. I captured a valid login request and configured Burp Intruder to inject payloads into both the email and password fields. A list of common email addresses and the password was used as part of a Cluster Bomb attack.

During the test, I closely monitored the HTTP response status codes, response times, and content lengths. All attempted logins returned a status code of 401 Unauthorized, indicating failed login attempts. At one point, the server responded with a 429 Too Many Requests status, suggesting that rate limiting or brute force protection was in place. No differences in responses were observed that would indicate valid usernames or credentials, and no session tokens or sensitive error messages were leaked.

This indicates the application effectively prevents user enumeration and brute-force attacks and implements proper mechanisms to handle repeated failed login attempts.



The screenshot shows the Burp Suite Intruder tool interface. The title bar indicates the attack is on <https://api.syfe.com>. The main window displays a table of attack results. The table has columns for Request, Payload 1, Payload 2, Status code, Response received, Error, Timeout, Length, and Comment. The results show 5 requests, all with a status code of 401 or 429, indicating failed login attempts. The response lengths are 571, 968, 1194, 1225, and 280 bytes respectively. The last request (index 5) returned a 429 status code, which is a 'Too Many Requests' error, suggesting rate limiting or brute force protection is in place.

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
0			401	571			920	
1	support@syfe.com	password	401	968			920	
2	no-reply@syfe.com	password	401	1194			920	
3	security@syfe.com	password	401	1225			920	
4	billing@syfe.com	password	429	280			5891	
5	dev@syfe.com	password		0				

Proposed Mitigation or Fix:

Although the authentication system appears secure, the following security best practices should continue to be applied and monitored:

- Use strong, hashed password storage (e.g., bcrypt or Argon2) to protect stored credentials.
- Enforce strong password policies, including minimum length, complexity, and no reuse of previous passwords.
- Implement rate limiting or progressive delays to slow down repeated failed login attempts.
- Add CAPTCHA after multiple failed attempts to stop automated brute-force tools.
- Avoid detailed error messages – always return a generic message like “Invalid email or password” to prevent username enumeration.
- Monitor and alert suspicious login behavior, such as failed attempts from the same IP or unusual geographic access.
- Enable account lockouts or temporary suspensions after a threshold of failed logins to reduce brute-force risk.
- Implement Multi-Factor Authentication (MFA) to provide an additional layer of protection for users and admins.
- Use secure cookies with HttpOnly, Secure, and SameSite=Strict flags to protect session data after login.

Conclusion:

Login functionality was tested with multiple username and password combinations using Burp Suite. All attempts failed, and the application returned consistent error messages without exposing any sensitive data. This confirms that the authentication system is well secured against brute force and login bypass attacks.