Sri Lanka Institute of Information Technology

Broken Authentication Vulnerability - Report 05

IT23187214

Web Security - IE2062

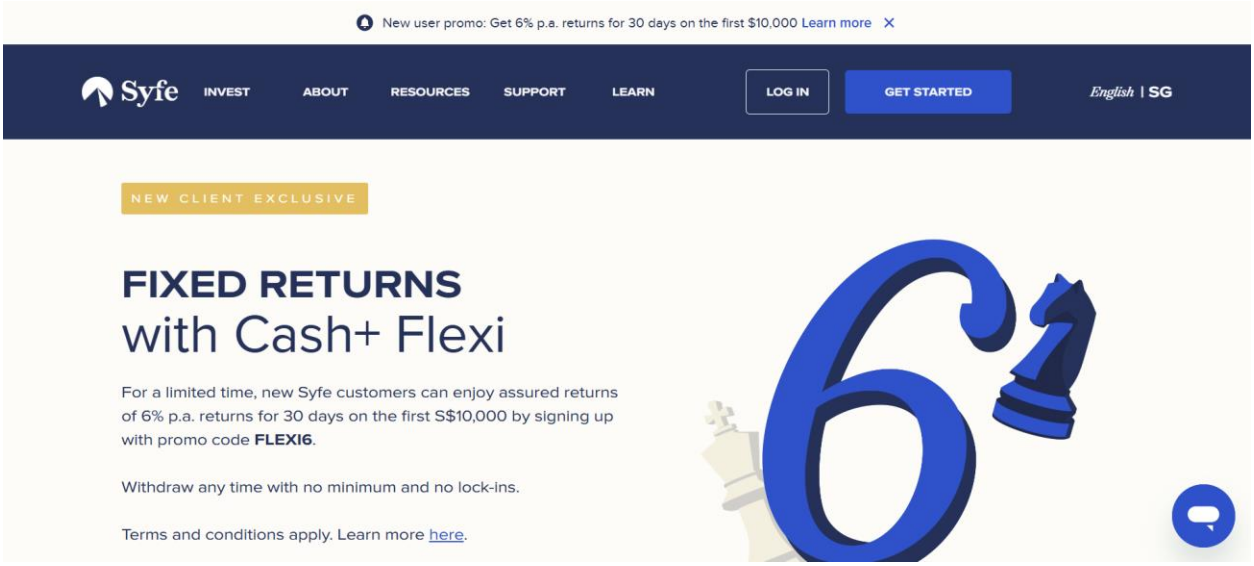
Vulnerability Title:

Broken Authentication Vulnerability

Vulnerability Description:

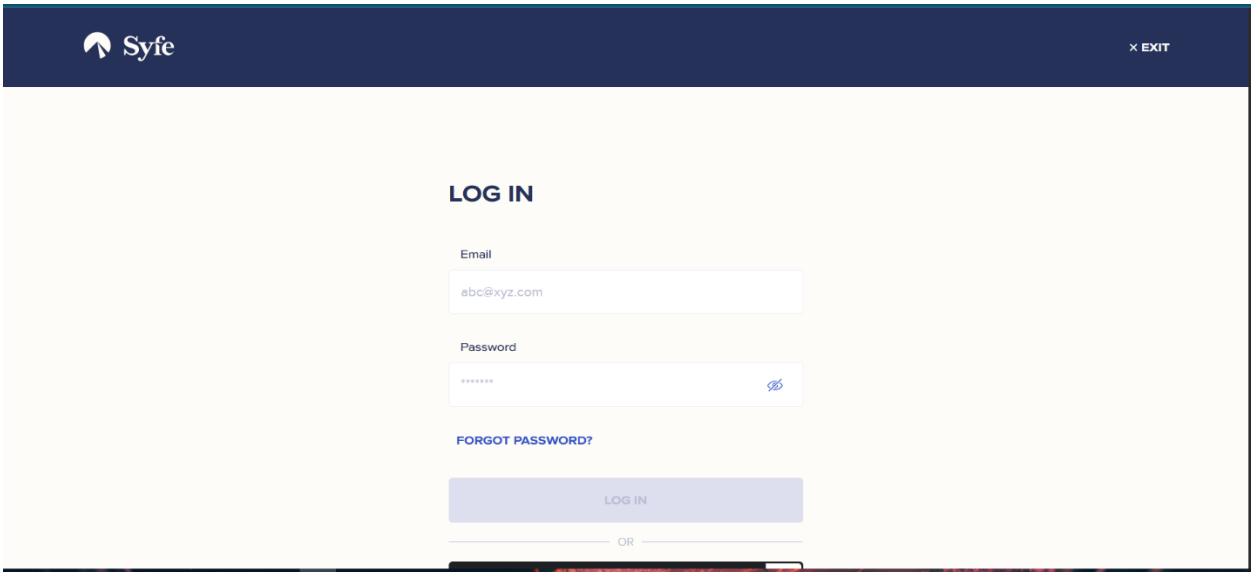
I found this program on the hackerone Bug hunting website. The website hosted at https://www.syfe.com. Broken Authentication occurs when attackers can compromise authentication mechanisms through brute force, credential stuffing, weak session handling, or bypassing login logic. This can result in unauthorized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



Affected Components:

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



Impact Assessment:

- **Risk Level:** High

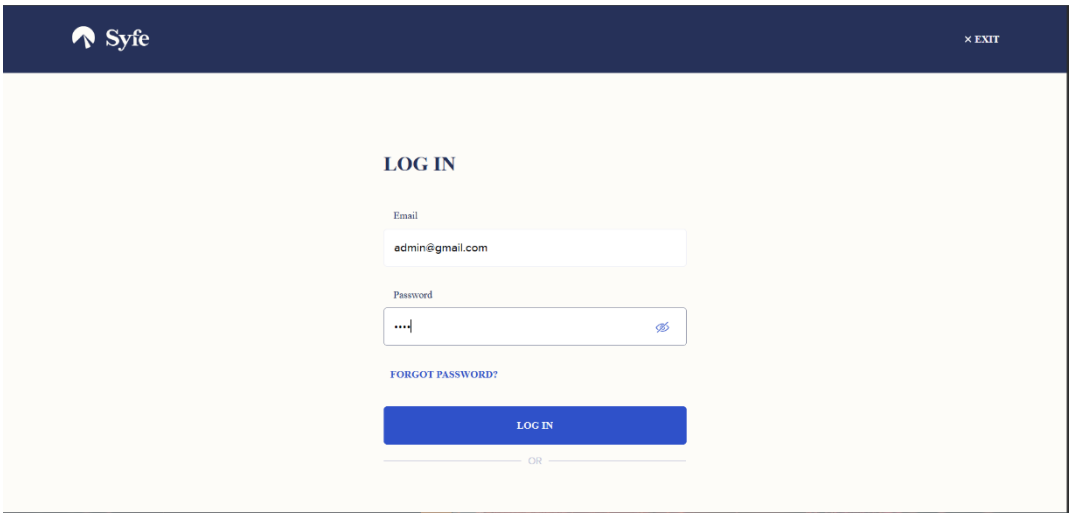
If vulnerable, attackers could:

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

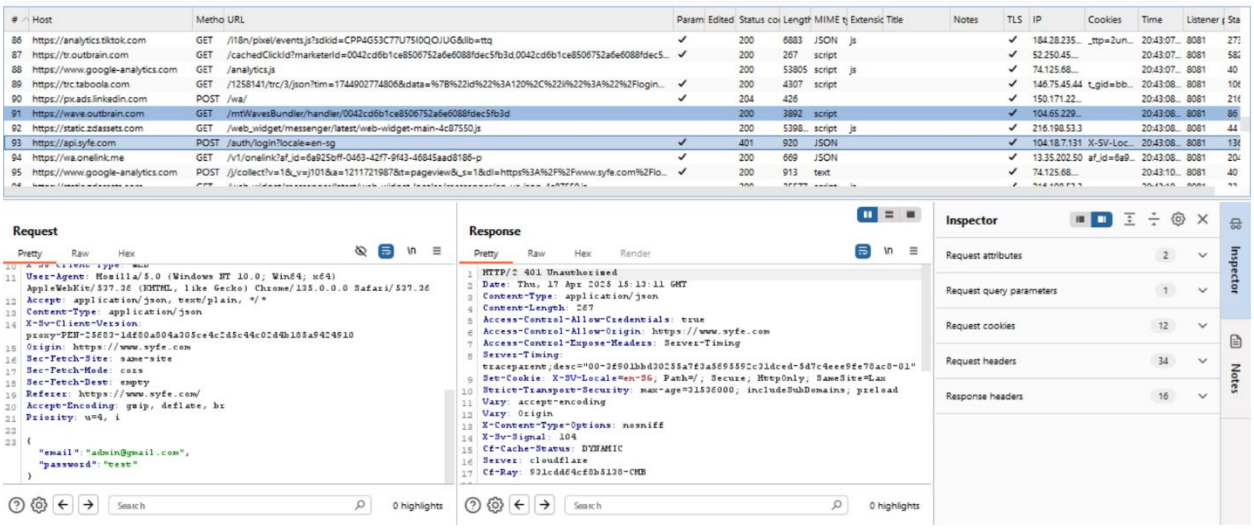
In this case, all login attempts failed, indicating **strong password validation**, consistent response handling, and likely use of **rate limiting or detection systems**.

Steps to Reproduce:

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

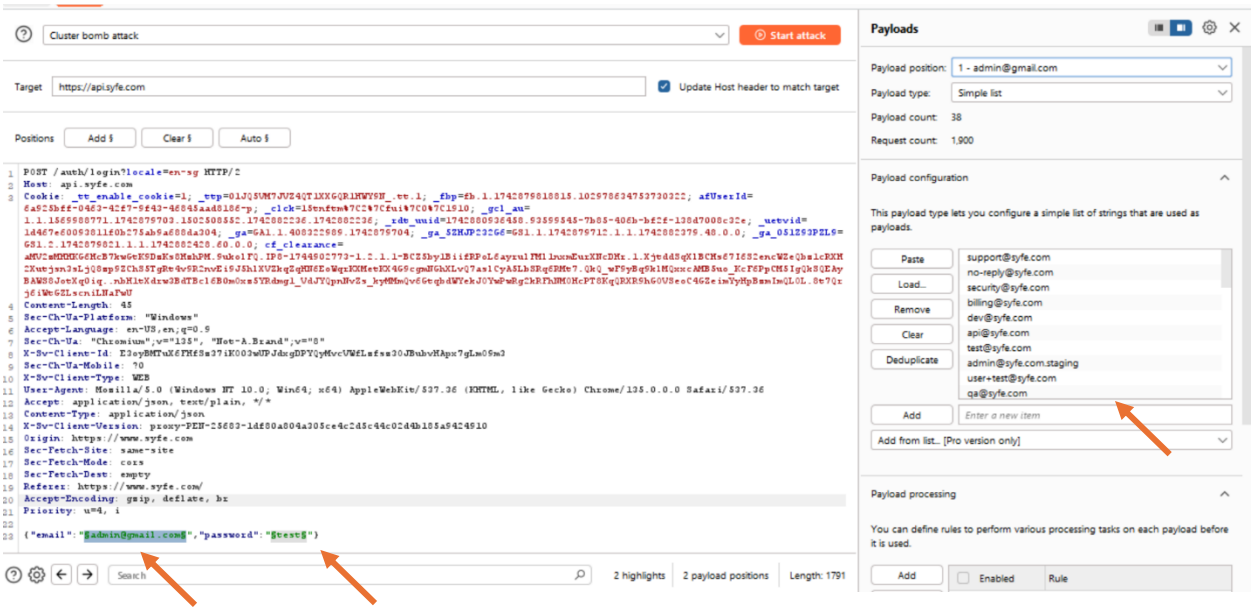


2. Sent the request to **Burp Intruder** and marked the email and password fields for attack.



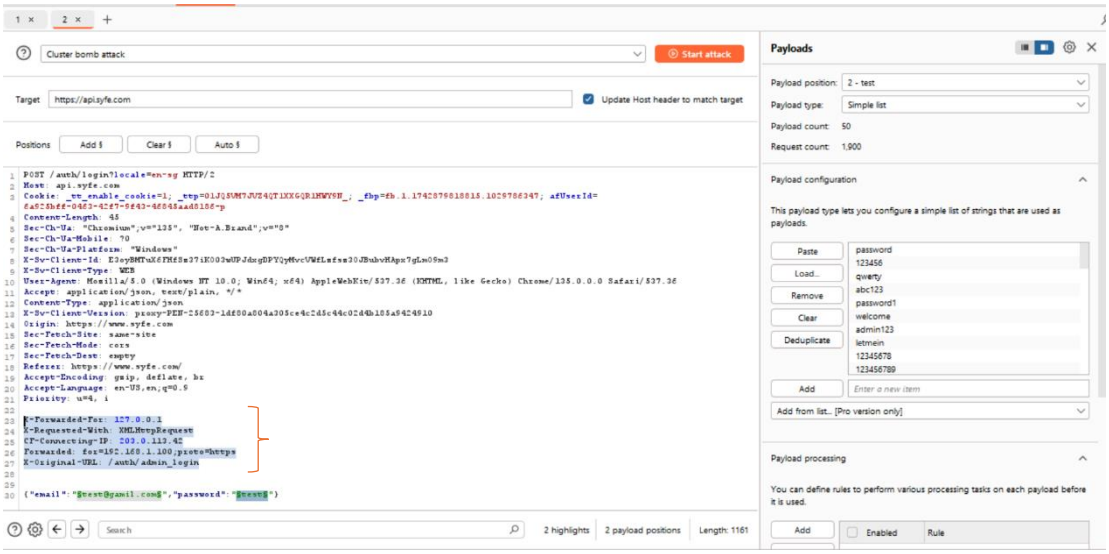
3. Used a list of test usernames and common password payloads:

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



4. **Added the following custom headers** to each request to assess if the application trusts or mishandles them:

   `X-Forwarded-For: 127.0.0.1` – to spoof the client's IP address.
   `X-Requested-With: XMLHttpRequest` – to simulate an AJAX request.
   `CF-Connecting-IP: 203.0.113.42` – to mimic a Cloudflare client's IP.
   `Forwarded: for=192.168.1.100;proto=https` – to test standard proxy forwarding.
   `X-Original-URL: /auth/admin_login` – to attempt URL override.



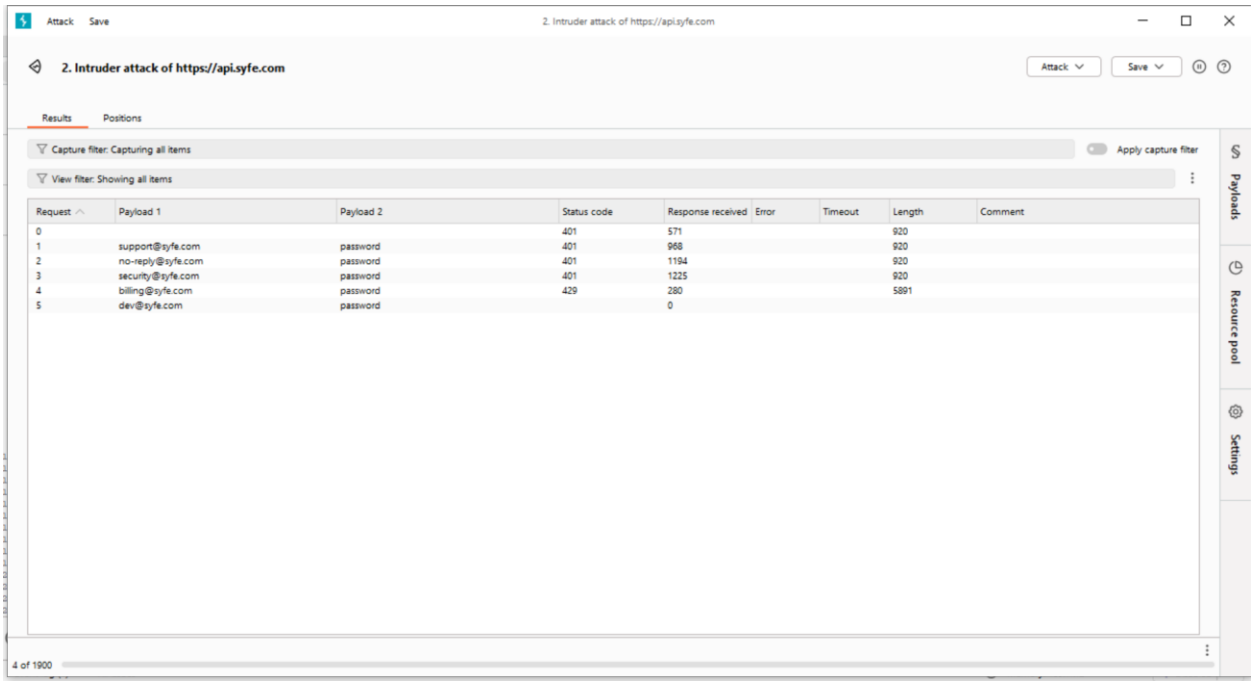
5. Launched the attack and monitored server responses for:
    - HTTP status codes (200, 302, 403)
    - Response length differences
    - Redirects or error messages indicating a successful login

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.



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.