OWASP

NEW ZEALAND DAY 2024

Find It, Fix It: The most common authentication flaws

Sandro Affentranger

Thank You to Our Sponsors and Hosts!

AppSec NZ appsec.org.nz

OWASP NEW ZEALAND owasp.org.nz

AUT UNIVERSITY TE WĀNANGA ARONUI O TAMAKI MAKAU RAU

BASTION SECURITY GROUP

DATACOM

aws

84.

PentesterLab

plexure

VERACODE

Without them, this Conference couldn’t happen.

whoami

Sandro Affentranger

Senior Penetration Tester at Oneconsult AG

- Working in cyber security for 7 years
- Based in Switzerland, but visiting our New Zealand branch during winter
- Passionate about passwords and password cracking

oneconsult
together against cyberattacks

PLEASE
CLOSE
GATE

The 5 most common authentication flaws

#1 – User Enumeration

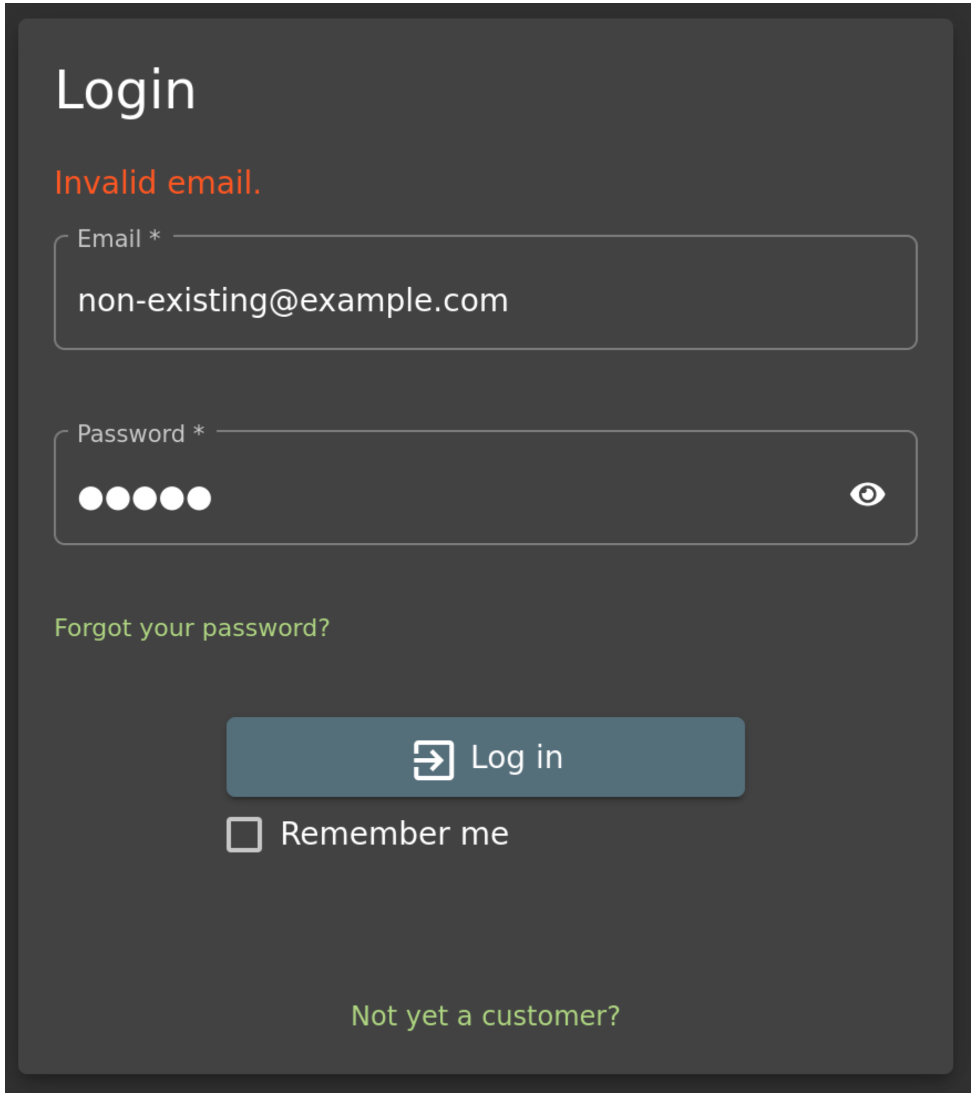
Ability to determine valid usernames or email addresses

- Can be used as starting point for other attacks
  - Credential stuffing, password spraying
  - Phishing attacks

👀 **How to find it:**

- Different error messages for existing vs. non-existing users
  - *Invalid username.*
  - *Invalid password.*
- Timing differences in responses
- Not only for login, but also for registration and password recovery flow

#1 – User Enumeration – Different Error Messages

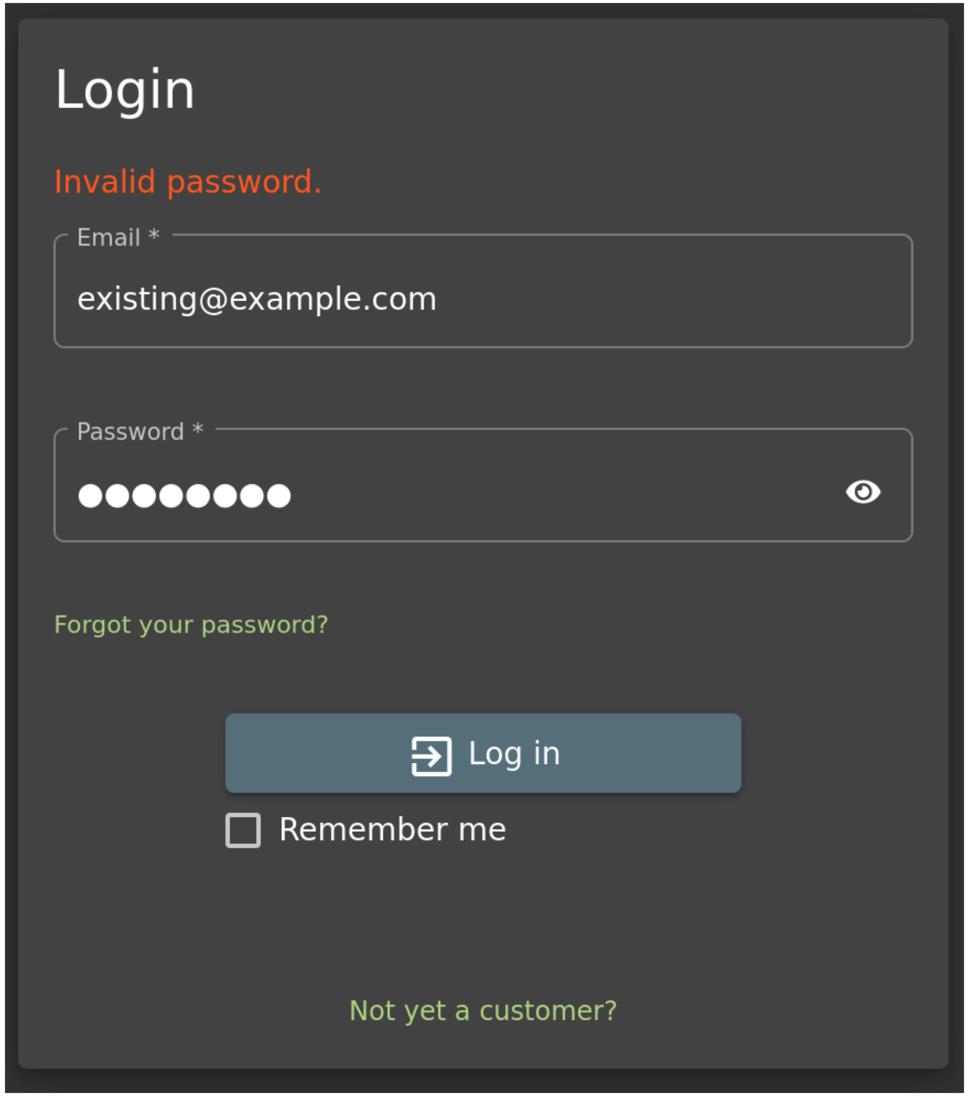


Login

Invalid email.

Email *: non-existing@example.com

Password *: ●●●●●



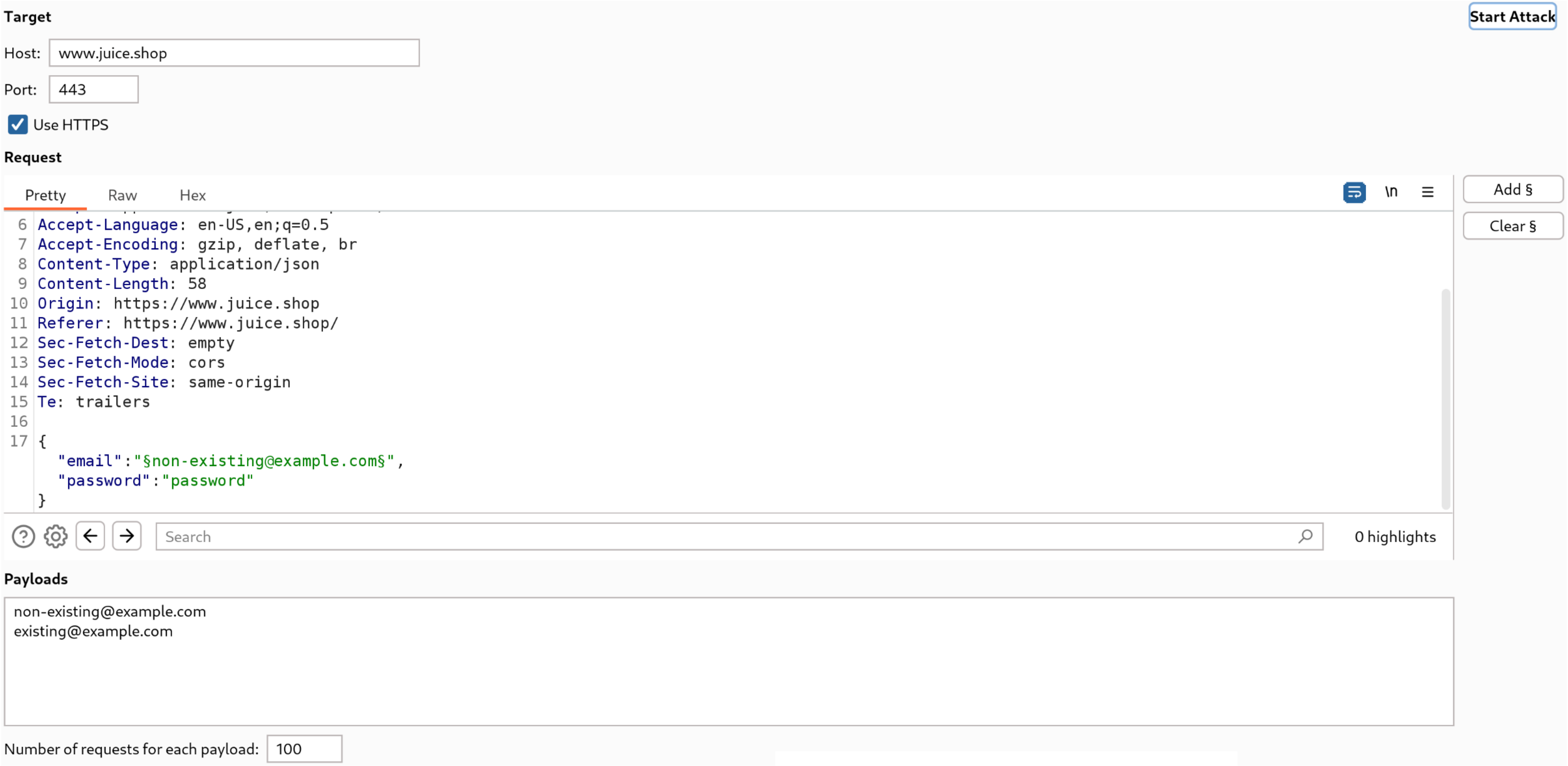
Login

Invalid password.

Email *: existing@example.com

Password *: ●●●●●●●●

#1 – User Enumeration – Timing Differences
with Burp Plugin: Timeinator



Target

Host: www.juice.shop

Port: 443

☑ Use HTTPS

Request

```
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 58
Origin: https://www.juice.shop
Referer: https://www.juice.shop/
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers

{
  "email":"§non-existing@example.com§",
  "password":"password"
}
```

Search

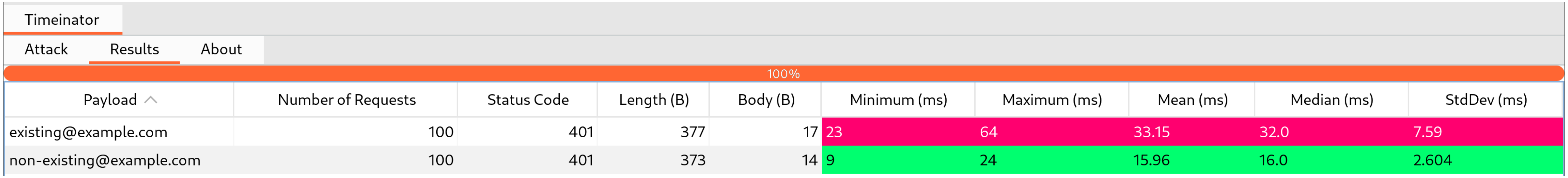
0 highlights

Payloads

```
non-existing@example.com
existing@example.com
```

Number of requests for each payload: 100

#1 – User Enumeration – Timing Differences
with Burp Plugin: Timeinator



Payload	Number of Requests	Status Code	Length (B)	Body (B)	Minimum (ms)	Maximum (ms)	Mean (ms)	Median (ms)	StdDev (ms)
existing@example.com	100	401	377	17	23	64	33.15	32.0	7.59
non-existing@example.com	100	401	373	14	9	24	15.96	16.0	2.604

→ Significant difference in response times between existing and non-existing user

#1 – User Enumeration

🔧 How to fix it:

- Use generic error messages:
  - *Invalid username or password.*
- Same for registration and password recovery:
  - *A link to activate your account has been emailed to the address provided.*
  - *If that email address is in our database, we will send you an email to reset your password.*

OWASP Cheat Sheet:

Authentication → Authentication and Error Messages

#2 – Weak Password Policy

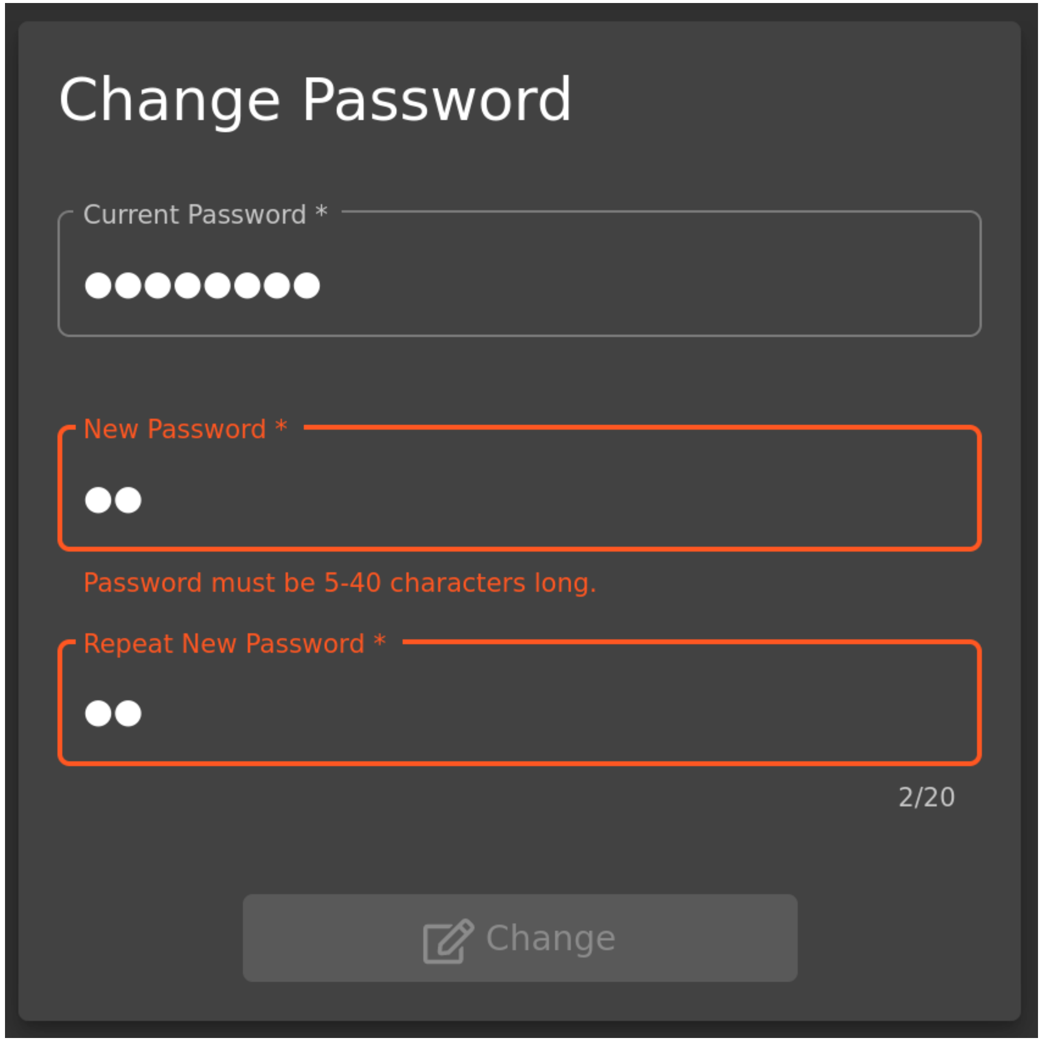
Allowing users to choose easily guessable or commonly used passwords

- Most users are bad at choosing good passwords
- Increases the risk of passwords being guessed

👀 How to find it:

- Attempt to change password to weak ones
- Check whether the same policy is enforced everywhere
  - Front-end vs. back-end
  - During registration, password change and password recovery

#2 – Weak Password Policy



Change Password

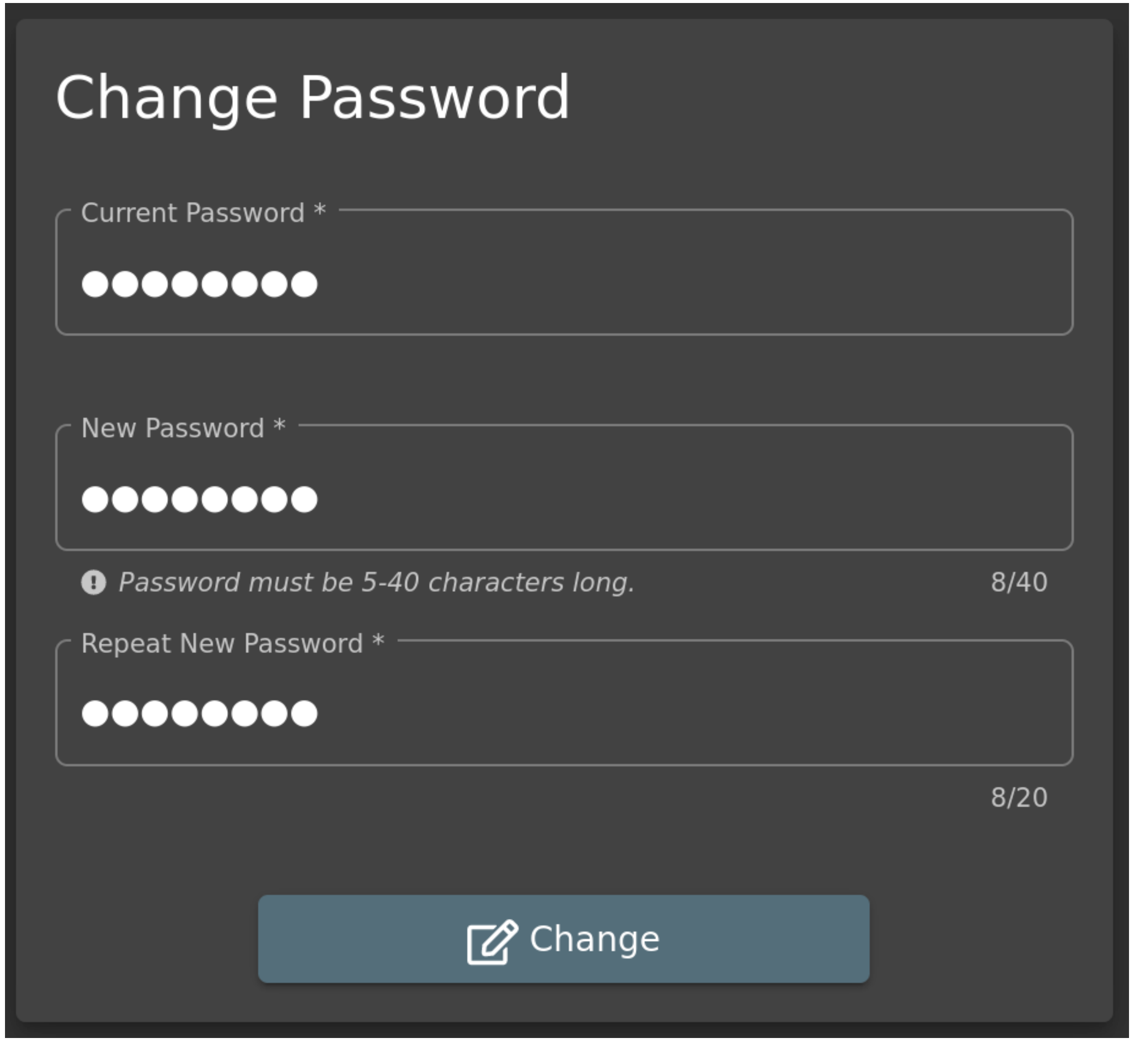
Current Password *
●●●●●●●●

New Password *
●●

Password must be 5-40 characters long.

Repeat New Password *
●●

2/20



Change Password

Current Password *
●●●●●●●●

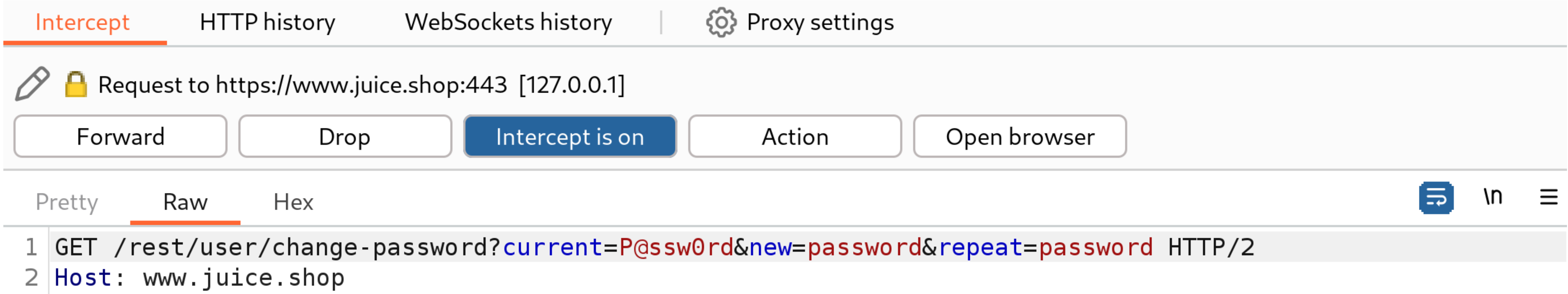
New Password *
●●●●●●●●

Password must be 5-40 characters long. 8/40

Repeat New Password *
●●●●●●●●

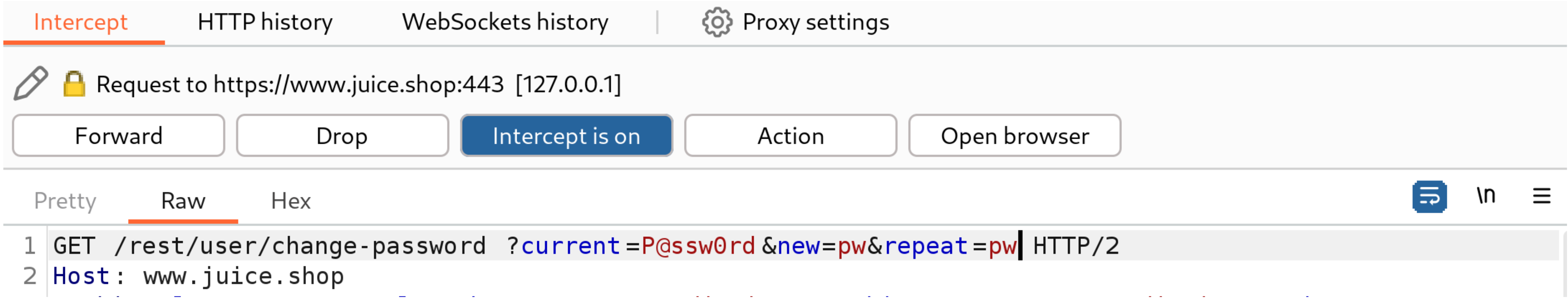
8/20

#2 – Weak Password Policy



Request to https://www.juice.shop:443 [127.0.0.1]

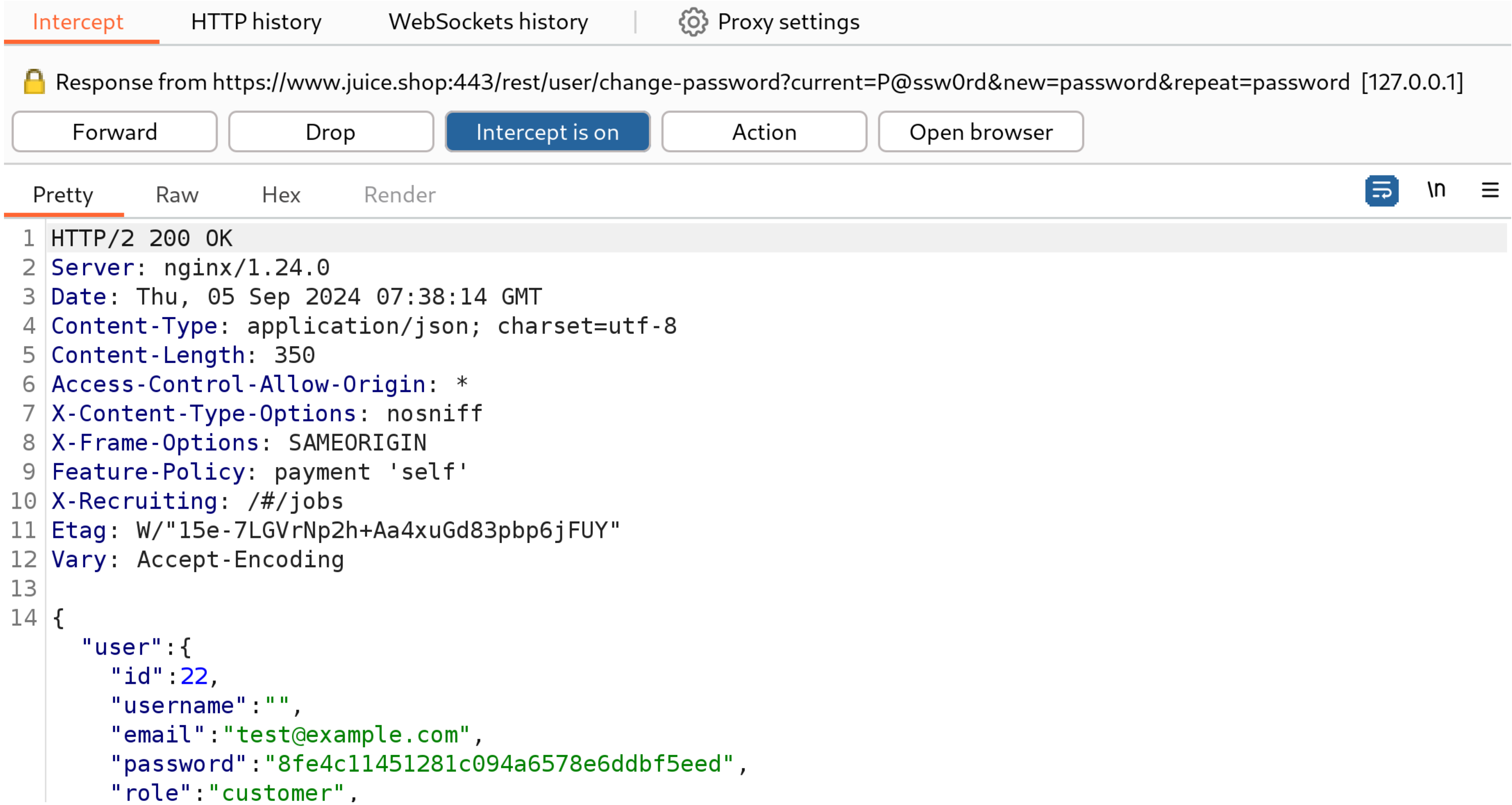
```
GET /rest/user/change-password?current=P@ssw0rd&new=password&repeat=password HTTP/2
Host: www.juice.shop
```



Request to https://www.juice.shop:443 [127.0.0.1]

```
GET /rest/user/change-password?current=P@ssw0rd&new=pw&repeat=pw HTTP/2
Host: www.juice.shop
```

#2 – Weak Password Policy



Response from https://www.juice.shop:443/rest/user/change-password?current=P@ssw0rd&new=password&repeat=password [127.0.0.1]

```
HTTP/2 200 OK
Server: nginx/1.24.0
Date: Thu, 05 Sep 2024 07:38:14 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 350
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Etag: W/"15e-7LGVrNp2h+Aa4xuGd83pbp6jFUY"
Vary: Accept-Encoding

{
  "user":{
    "id":22,
    "username":"",
    "email":"test@example.com",
    "password":"8fe4c11451281c094a6578e6ddbf5eed",
    "role":"customer",
```

#2 – Weak Password Policy

🔧 How to fix it:

- Implement strong password requirements (follow recommendations in ASVS / OWASP Cheat Sheet):
  - Require a minimum length of 12 characters
  - Don’t enforce periodic password rotation
  - Check passwords against known breached passwords
    - Use HashMob or HaveIBeenPwned API, or do a local check
- Encourage use of password managers
  - Allow passwords up to 128 characters, allow pasting

OWASP Cheat Sheet:

Authentication → Implement Proper Password Strength Controls

#3 – No Multi-Factor-Authentication

Relying solely on single-factor (typically password-only) authentication

- Increases risk of account compromise
- Most users are bad at choosing good passwords

👀 How to find it:

- Check if any additional authentication methods are offered
  - What methods are offered?
  - Are they required?

#3 – No Multi-Factor-Authentication

🔧 **How to fix it:**

- Implement MFA options
  - Use only strong second factors
    - authenticator apps, push notifications, hardware token, …
    - not OTP via SMS or email
  - Ideally, phishing-resistant options
- Encourage the use of MFA
- And make it a requirement for privileged users

OWASP Cheat Sheet:

Multifactor Authentication

#4 – No Brute-Force Protection

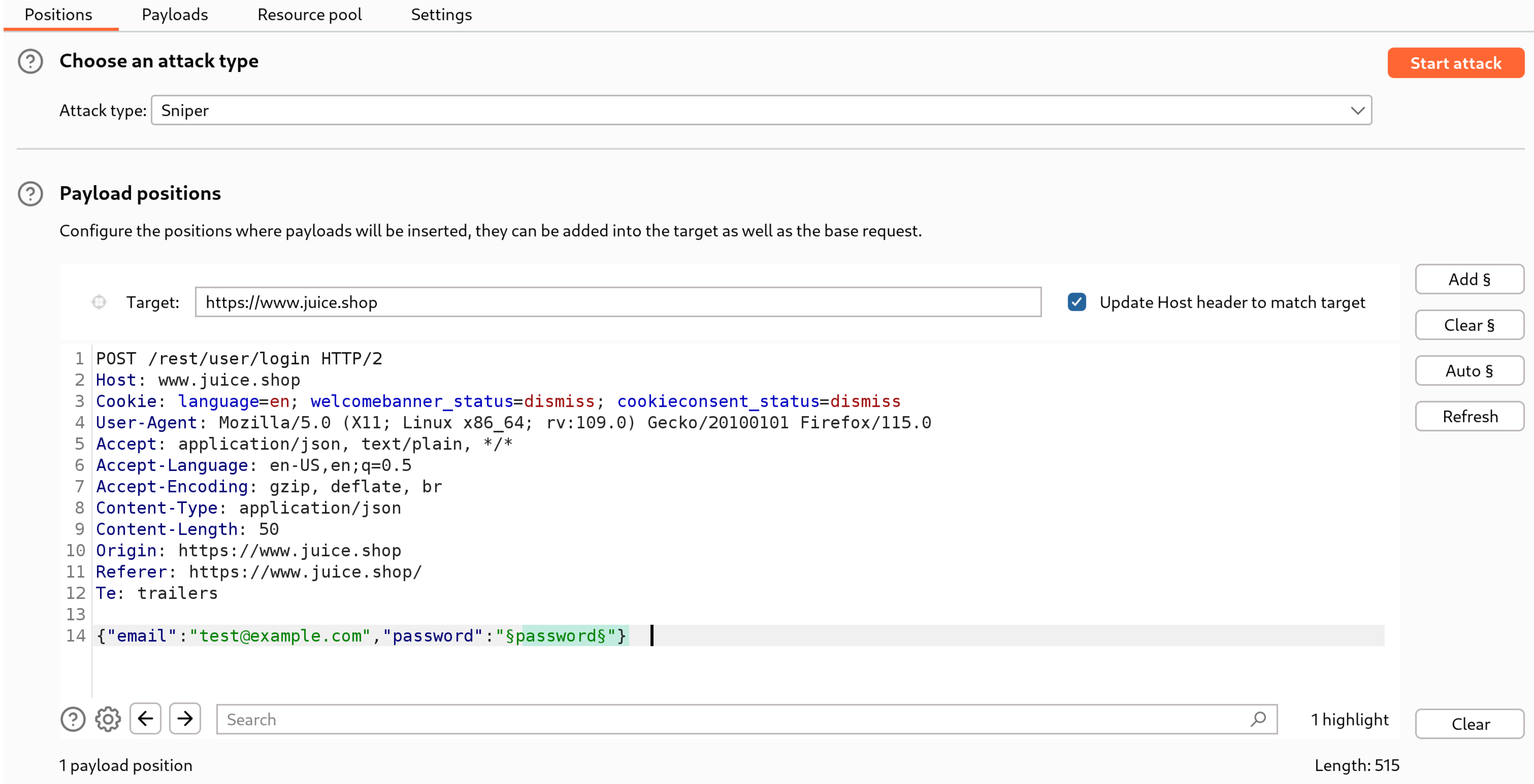
Allowing unlimited login attempts without any restrictions

- Easier to enumerate usernames
- Easier to guess passwords

👀 How to find it:

- Attempt multiple (like 100-200) logins with a wrong password, followed by a login with the correct password
  - Blocked or slowed down after too many attempts?
  - Or was the login successful?
- The same for the second factor, if a code has to be entered

#4 – No Brute-Force Protection with Burp Intruder



Choose an attack type

Attack type: Sniper

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: https://www.juice.shop

☑ Update Host header to match target

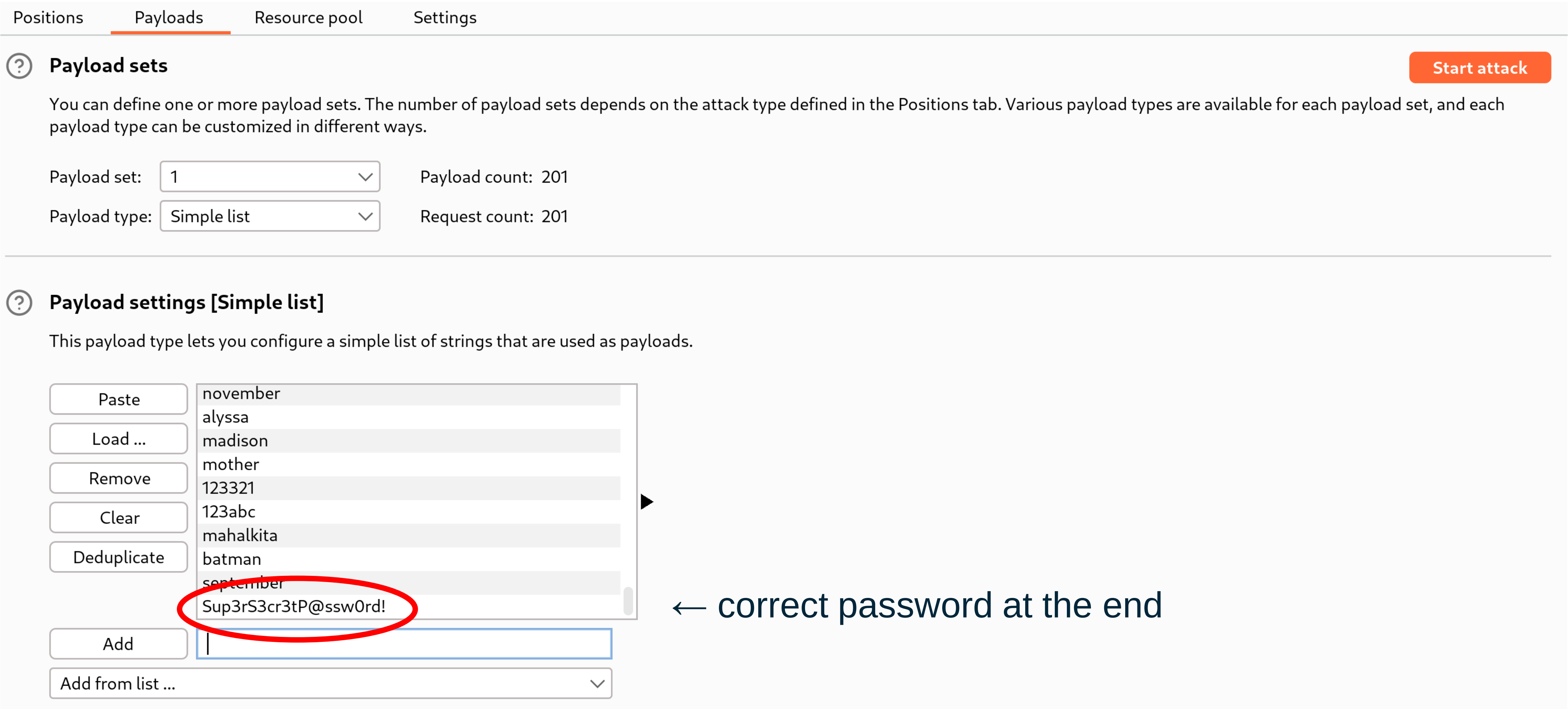
```
POST /rest/user/login HTTP/2
Host: www.juice.shop
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 50
Origin: https://www.juice.shop
Referer: https://www.juice.shop/
Te: trailers

{"email":"test@example.com","password":"§password§"}
```

1 highlight

1 payload position

Length: 515

#4 – No Brute-Force Protection
with Burp Intruder



Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1

Payload count: 201

Payload type: Simple list

Request count: 201

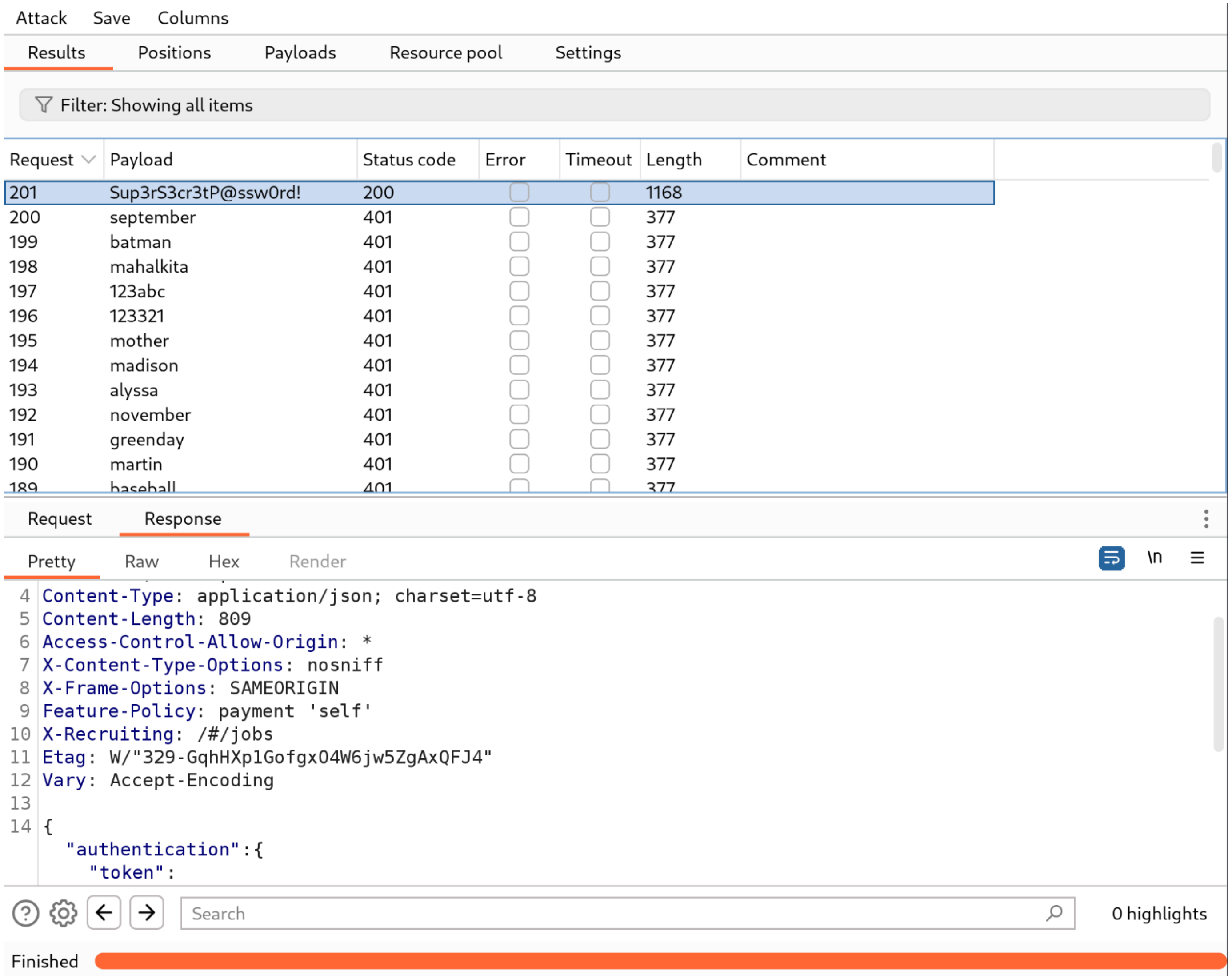
Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

```
november
alyssa
madison
mother
123321
123abc
mahalkita
batman
september
Sup3rS3cr3tP@ssw0rd!
```

← correct password at the end

#4 – No Brute-Force Protection with Burp Intruder



Request	Payload	Status code	Error	Timeout	Length	Comment
201	Sup3rS3cr3tP@ssw0rd!	200	☐	☐	1168	
200	september	401	☐	☐	377	
199	batman	401	☐	☐	377	
198	mahalkita	401	☐	☐	377	
197	123abc	401	☐	☐	377	
196	123321	401	☐	☐	377	
195	mother	401	☐	☐	377	
194	madison	401	☐	☐	377	
193	alyssa	401	☐	☐	377	
192	november	401	☐	☐	377	
191	greenday	401	☐	☐	377	
190	martin	401	☐	☐	377	
189	baseball	401	☐	☐	377	

```
Content-Type: application/json; charset=utf-8
Content-Length: 809
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Etag: W/"329-GqhHXp1Gofgx04W6jw5ZgAxQFJ4"
Vary: Accept-Encoding

{
  "authentication":{
    "token":
```

#4 – No Brute-Force Protection

🔧 How to fix it:

- Add anti-automation controls
  - Implement rate-limiting
  - Implement increasing delays between login attempts
  - Use CAPTCHAs or similar challenges after a few failed attempts
- Ensure that accounts are not locked after too many failed attempts

OWASP Cheat Sheet:

Authentication → Protect Against Automated Attacks

#5 – No Notification on Critical Data Change

Failing to alert users when important account changes occur

- Attackers might make unauthorized changes without user noticing and being able to react

👀 How to find it:

- Attempt to change critical data
  - Reset password
  - Change email
  - Change second factor
- Login from a new location (using a VPN)
- Check if any notifications received?

#5 – No Notification on Critical Data Change

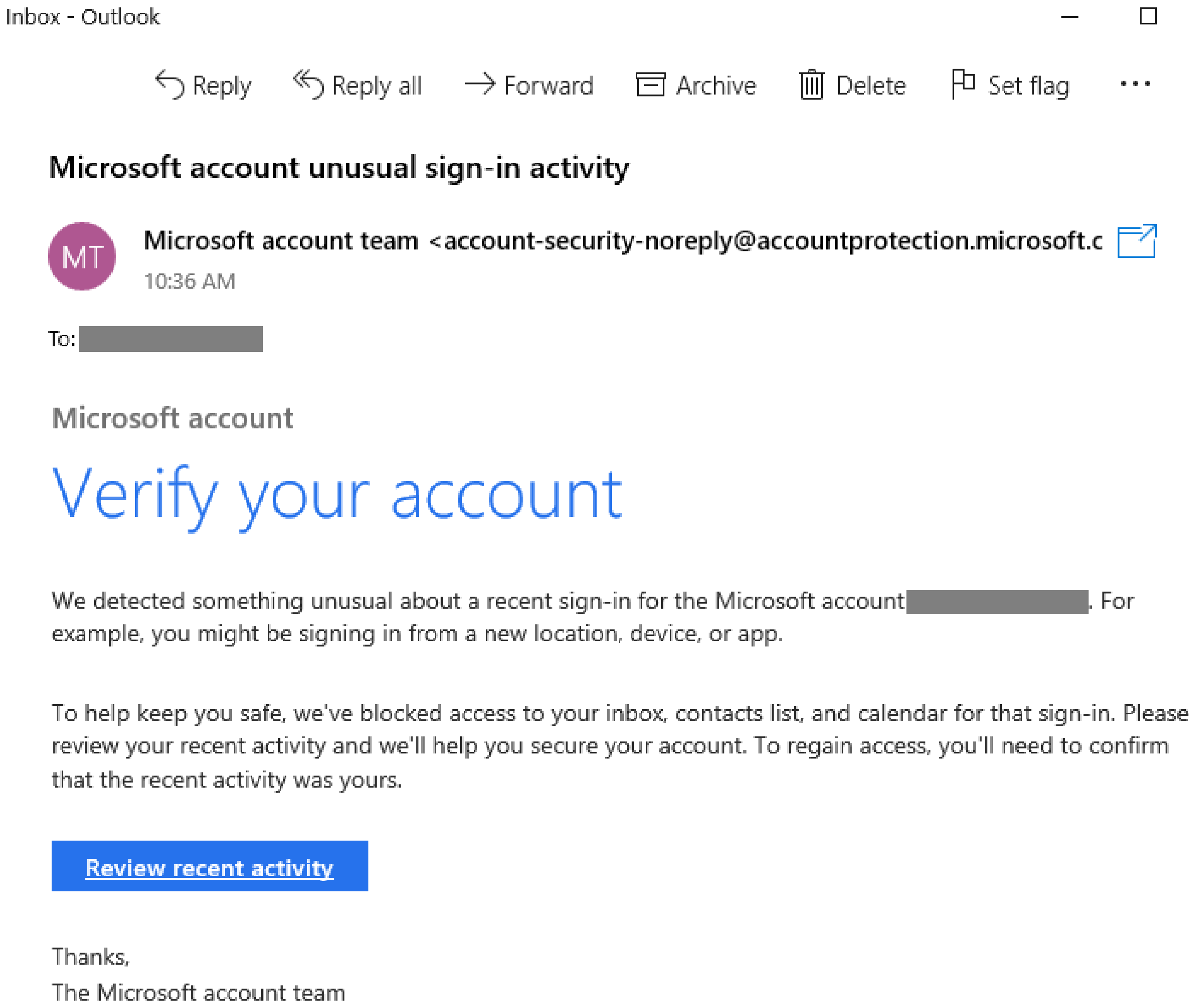
🔧 How to fix it:

- Implement immediate notifications for critical changes (via email, SMS, or in-app)
  - Important: If the email address is changed, send the notification to the old address

OWASP Cheat Sheet:

Authentication → Changing A User's Registered Email Address

#5 – No Notification on Critical Data Change



Microsoft account unusual sign-in activity

Microsoft account team <account-security-noreply@accountprotection.microsoft.c

10:36 AM

To: [redacted]

Microsoft account

Verify your account

We detected something unusual about a recent sign-in for the Microsoft account [redacted]. For example, you might be signing in from a new location, device, or app.

To help keep you safe, we've blocked access to your inbox, contacts list, and calendar for that sign-in. Please review your recent activity and we'll help you secure your account. To regain access, you'll need to confirm that the recent activity was yours.

Review recent activity

Thanks,
The Microsoft account team

Takeaways

- **5 most common authentication flaws:**
  - User enumeration
  - Weak password policy
  - No MFA
  - No brute-force protection
  - No notifications on critical data change
- Are not hard to find yourself without any expensive tools
- Use available resources on how to fix them (like OWASP Cheat Sheet Series) and what needs to be taken into consideration
- Want to do more? Check out the OWASP Application Security Verification Standard for more requirements

Thank you!

Find me online

✉ sandro.affentranger@oneconsult.com

in sandro-affentranger

afsa@infosec.exchange

𝕏 0xAF5A

Some helpful links:

- https://owasp.org/www-project-application-security-verification-standard/
- https://owasp.org/www-project-web-security-testing-guide/
- https://cheatsheetseries.owasp.org/