Broken Authentication

Web Development and Security (ZEIT3119)

Week 12

Dr. Reza Rafeh

Revision

- At which step SQL Injection is inserted
  1. Connecting to the database;
  2. Sending SQL statements and data to the database;
  3. Fetching the result and display data from the database;
  4. Closing the connection

Answer: Step 2

Revision

- How SQL Injection can harm the database?

Answer:

- Change the content of database (delete/modify the tables)
- Cause denial of service to application
- Retrieve sensitive data from database
- Attacker can get administrative rights

Revision

- What is the most common example where SQL Injection can be inserted easily?

Answer: HTML form

- What happens if attacker injects the following statement in SQL statement

  Select * From users

  Where username =user1’ ‘OR 1=1-- and Password = 123;

Answer:

The query will return all the information of users form table users and bypass the password because of hyphens --

Revision

State true or False

- Web applications are more affected by XSS than SQL Injection

Answer: True

- Reflected XSS is more harmful than Stored XSS

Answer: False

Revision

- How many types of XSS are there?

Answer:

- Reflected XSS
- Stored XSS
- DOM XSS
- What are the mitigation techniques for XSS?

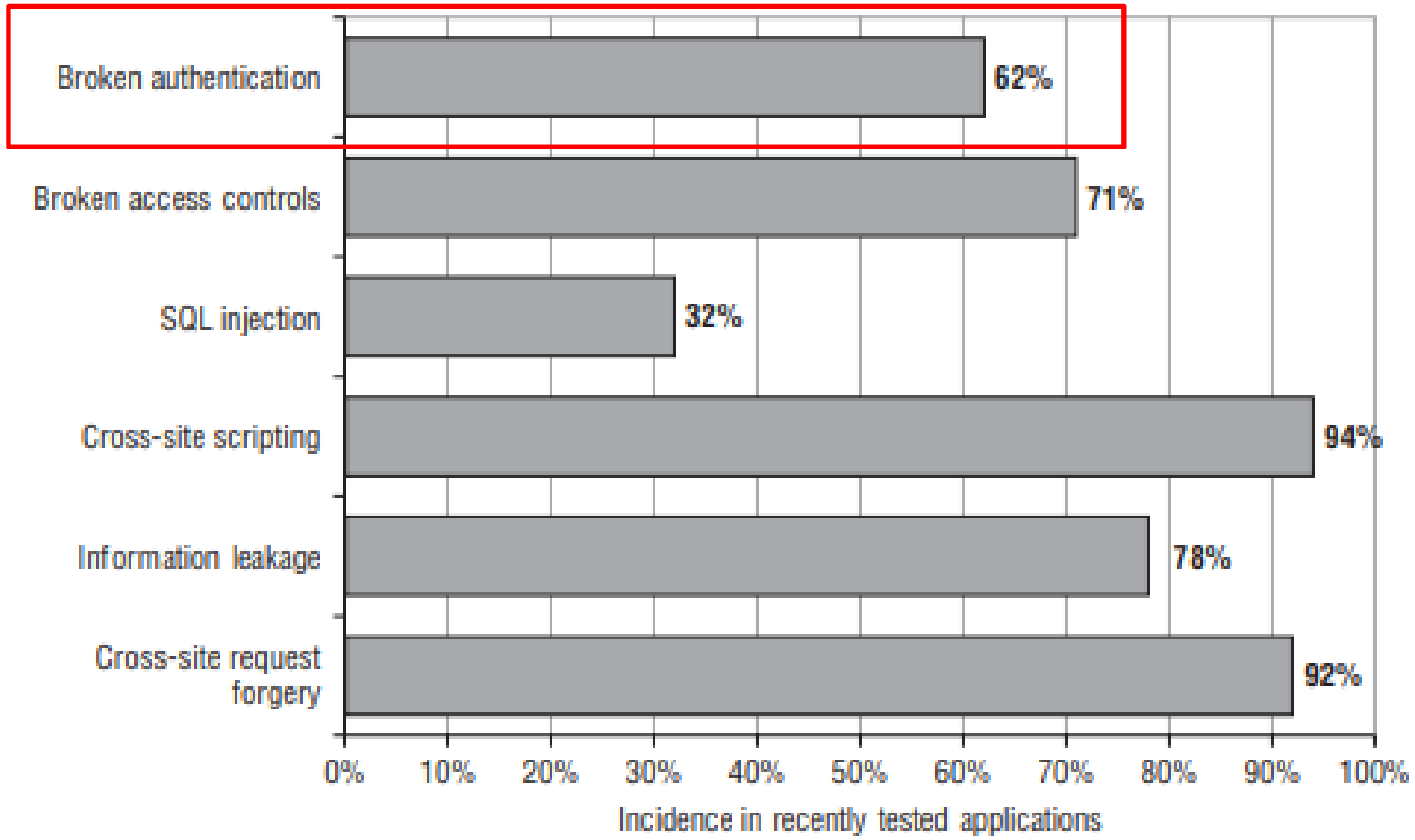
Answer:

- Educating people
- Web Application Firewall

Outline

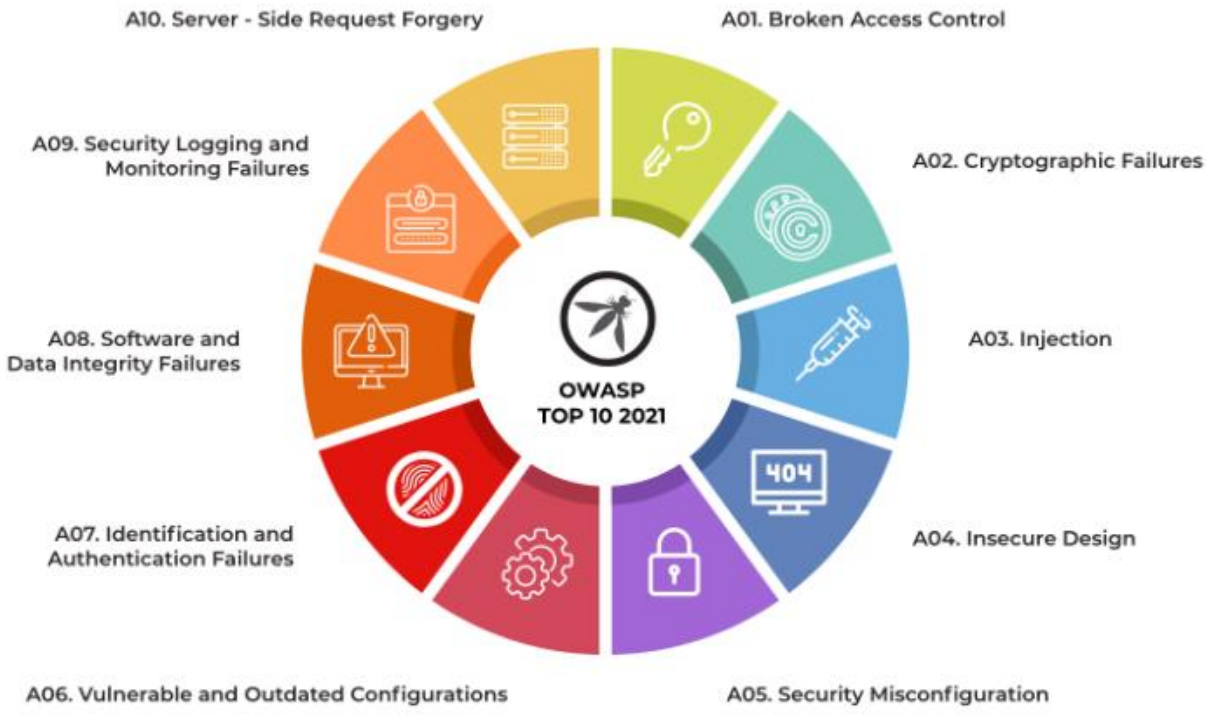
- Authentication and Authorization
- Broken Authentication
- Result of Broken Authentication
- OWASP Testing Guide: Authentication
- Broken Session Management
- OWASP Testing Guide: Session Management
- bWAPP Broken Authentication
- Example
- Final Exam

OWASP Vulnerabilities



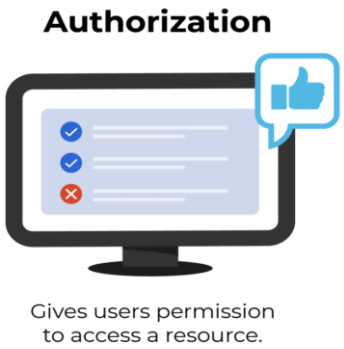
OWASP Vulnerabilities - 2021

Broken Access Control (up from #5 in 2020 to the top spot in 2021)



Authentication and Authorization

- **Authentication:**
  - The Process of verification that an individual, entity or website is who it claims to be examples:
    - Password
    - One-time Pins
    - Authentication App
    - Biometrics
- **Authorization:**
  - Once the user is authenticated, it gives the user permission to access the resource



Authentication and Authorization

- Authentication acts as a key to the door house. Lock on the door grants access to someone with the correct key. Similarly, they username/Password (or any authentication methods) act as key to the system. Correct credentials can give access to the system

- Authorization is the permission. Once the user is Authenticated and use the correct key, the user is authorized to enter the house. Similarly, once the user is authenticated after entering correct credentials it can enter the system and authorized to do things depending on the privilege assigned to the user

- Authorization and Authentication work together.

Authentication and Authorization

	Authentication	Authorization
What does it do?	Verifies credentials	Grants or denies permissions
How does it work?	Through passwords, biometrics, one-time pins, or apps	Through settings maintained by security teams
Is it visible to the user?	Yes	No
It is changeable by the user?	Partially	No
How does data move?	Through ID tokens	Through access tokens

Broken Authentication

Why is it Broken

- Password not hashed.
- Weak Password recovery method.
- Information leaked on failed login
- Unlimited logon attempt
- Exposed Session-Ids’.
- Long session timeout.
- Improper rotation of session-ids’ after logout.
- Sending session-ids’, passwords over unencrypted connections.

Result of Broken Authentication

- By-pass authentication
- Complete control of accounts
- Account theft, sensitive end-user (customer) data could be stolen
- Reputational damage and revenue loss
- Title Placeholder

OWASP Testing Guide: Authentication

1. Testing for Credentials Transported over an Encrypted Channel

Verifying that the user's authentication data are transferred via an encrypted channel to avoid being intercepted by malicious users

Example1 : Sending data with POST method through HTTP

```
POST http://www.example.com/AuthenticationServlet HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; it; rv:1.8.1.14) Gecko/20080404
Accept: text/xml,application/xml,application/xhtml+xml
Accept-Language: it-it,it;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.example.com/index.jsp
Cookie: JSESSIONID=LV-RRQQWg,tkpk7QW:640:bW1yBdqn98CGlkP4jTvVCGdyPkmn3S!
Content-Type: application/x-www-form-urlencoded
Content-length: 64

delegated_service=218&User=test&Pass=test&Submit=SUBMIT
```

Example 2: Sending data with POST method through HTTPs

```
POST https://www.example.com:443/cgi-bin/login.cgi HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; it; rv:1.8.1.14) Gecko/20080404
Accept: text/xml,application/xml,application/xhtml+xml,text/html
Accept-Language: it-it,it;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: https://www.example.com/cgi-bin/login.cgi
Cookie: language=English;
Content-Type: application/x-www-form-urlencoded
Content-length: 50

Command=Login&User=test&Pass=test
```

OWASP Testing Guide: Authentication

1. **Testing for Credentials Transported over an Encrypted Channel**

Example 3: Sending data with POST method via HTTPs on a page reachable via HTTP

```
POST https://www.example.com/homepage.do
Host: www.example.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; it; rv:1.8.1.14) Gecko/20080404
Accept: text/xml,application/xml,application/xhtml+xml,text/html
Accept-Language: it-it,it;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.example.com/homepage.do
Cookie: SERVTIMSESSIONID=s2JyLkvDJ9ZhX3yr5BJ3DFLkdphH0QNSJ3VQB6pLhjkW6F
Content-Type: application/x-www-form-urlencoded
Content-length: 45

User=test&Pass=test&portal=ExamplePortal
```

Example 4: Sending data with GET method via HTTPs

```
GET https://www.example.com/success.html?user=test&pass=test HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; it; rv:1.8.1.14) Gecko/20080404
Accept: text/xml,application/xml,application/xhtml+xml,text/html
Accept-Language: it-it,it;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: https://www.example.com/form.html
If-Modified-Since: Mon, 30 Jun 2008 07:55:11 GMT
If-None-Match: "43a01-5b-4868915f"
```

OWASP Testing Guide: Authentication

2. Testing for default credentials transported over an Encrypted Channel

- **Testing for default credentials of common applications**
  - Try the following usernames - "admin", "administrator", "root", "system", "guest", "operator", or "super".
- **Testing for default password of new accounts**
  - It can also occur that when a new account is created in an application the account is assigned a default password. This password could have some standard characteristics making it predictable.

3. Testing for Weak lock out mechanism

- **Testing for account lock-out policy**
- To evaluate the account lockout mechanism's ability to mitigate brute force password guessing, attempt an invalid log in by using the incorrect password a number of times, before using the correct password to verify that the account was locked out.

bWAPP – Insecure Login Form

bWAPP - Broken Authentication | bWAPP - Broken Authentication | bWAPP - Broken Authentication

localhost/bwapp/ba_insecure_login_1.php

bWAPP

an extremely buggy web app !

Choose your bug:

---------- bWAPP v2.2 ---------- Hack

Set your security level:

low Set Current: **low**

Bugs | Change Password | Create User | Set Security Level | Reset | Credits | Blog | Logout | Welcome Bee

/ Broken Auth. - Insecure Login Forms /

Enter your credentials.

Login:

Password:

Login

Back	Alt+Left Arrow
Forward	Alt+Right Arrow
Reload	Ctrl+R
Save as...	Ctrl+S
Print...	Ctrl+P
Cast...	
Create QR Code for this page	
View page source	Ctrl+U
Inspect	

bWAPP is licensed under © 2014 MME BVBA / Follow @MME_IT on Twitter and ask for our cheat sheet, containing all solutions! / Need an exclusive training?

Type here to search 1°C ENG 11:22 PM 27/05/2023

UNSW CANBERRA

bWAPP – Insecure Login Form

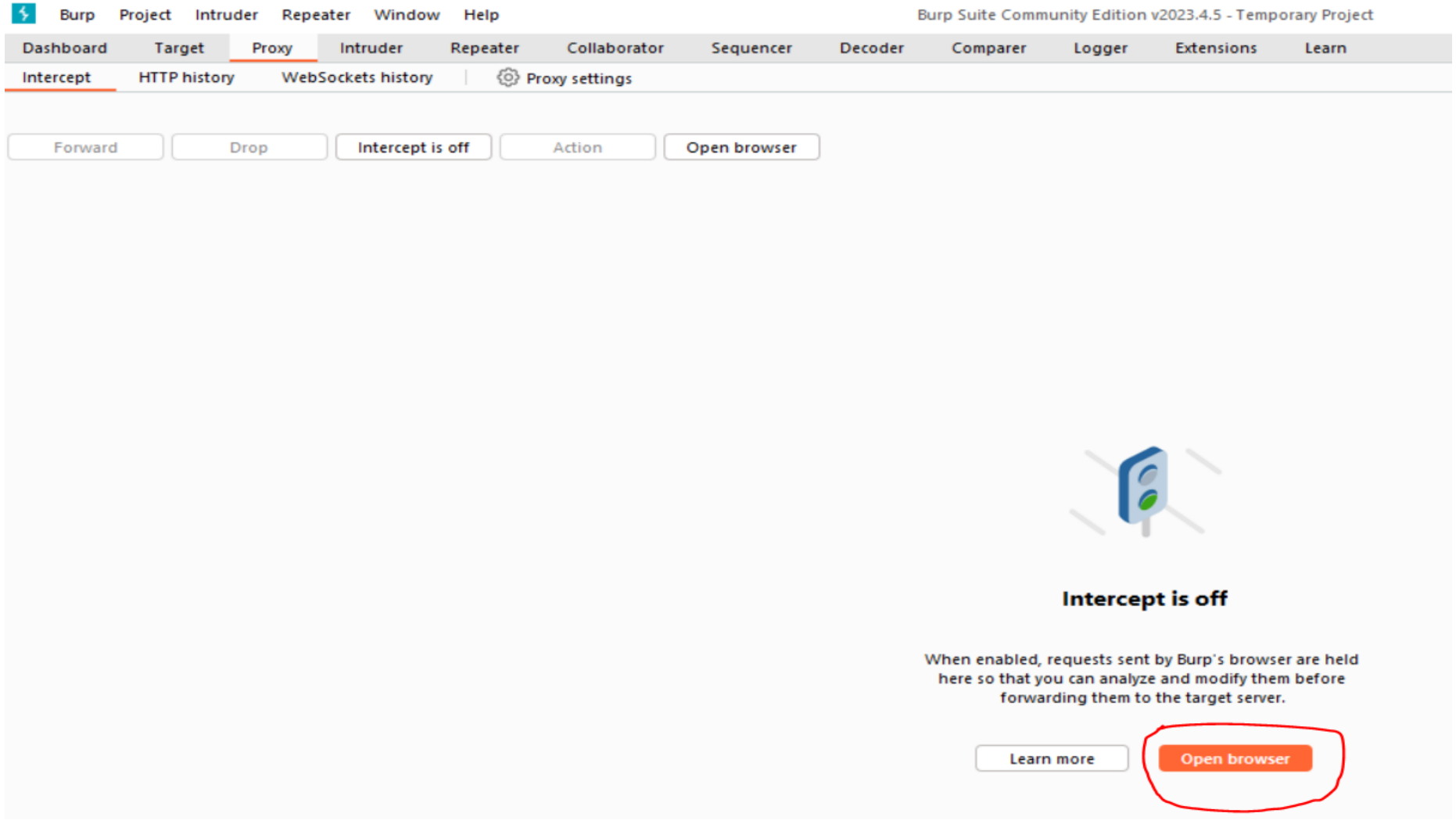
view-source:localhost/bwapp/ba_insecure_login_1.php

```
43            <td><font color="red">Welcome Bee</font></td>
44
45        </tr>
46
47    </table>
48
49 </div>
50
51 <div id="main">
52
53     <h1>Broken Auth. - Insecure Login Forms</h1>
54
55     <p>Enter your credentials.</p>
56
57     <form action="/bwapp/ba_insecure_login_1.php" method="POST">
58
59         <p><label for="login">Login:</label><font color="white">tonystark</font><br />
60         <input type="text" id="login" name="login" size="20" /></p>
61
62         <p><label for="password">Password:</label><font color="white">I am Iron Man</font><br />
63         <input type="password" id="password" name="password" size="20" /></p>
64
65         <button type="submit" name="form" value="submit">Login</button>
66
67     </form>
68
69     </br >
70
71 </div>
72
73 <div id="side">
74
75     <a href="http://twitter.com/MME_IT" target="blank_" class="button"><img src="./images/twitter.png"></a>
76     <a href="http://be.linkedin.com/in/malikmesellem" target="blank_" class="button"><img src="./images/linkedin.png"></a>
77     <a href="http://www.facebook.com/pages/MME-IT-Audits-Security/104153019664877" target="blank_" class="button"><img src="./images/facebook.png"></a>
78     <a href="http://itsecgames.blogspot.com" target="blank_" class="button"><img src="./images/blogger.png"></a>
79
80 </div>
```

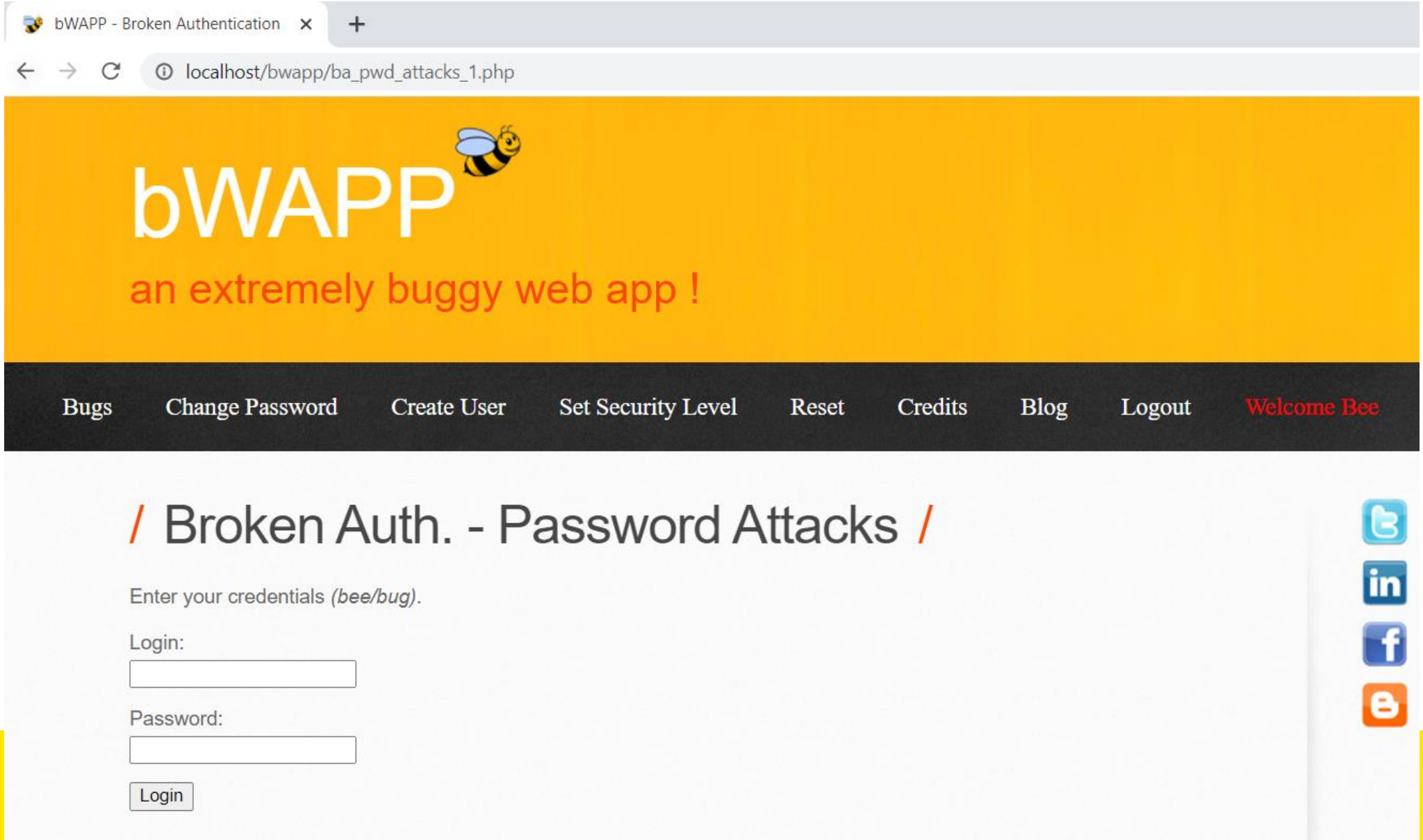
Burp Suite

- Burp Suite is a set of tools used for penetration testing of web applications.
  - Spider
  - Proxy
  - Intruder
  - Repeater
  - Sequencer
  - Decoder
  - Extender
  - Scanner

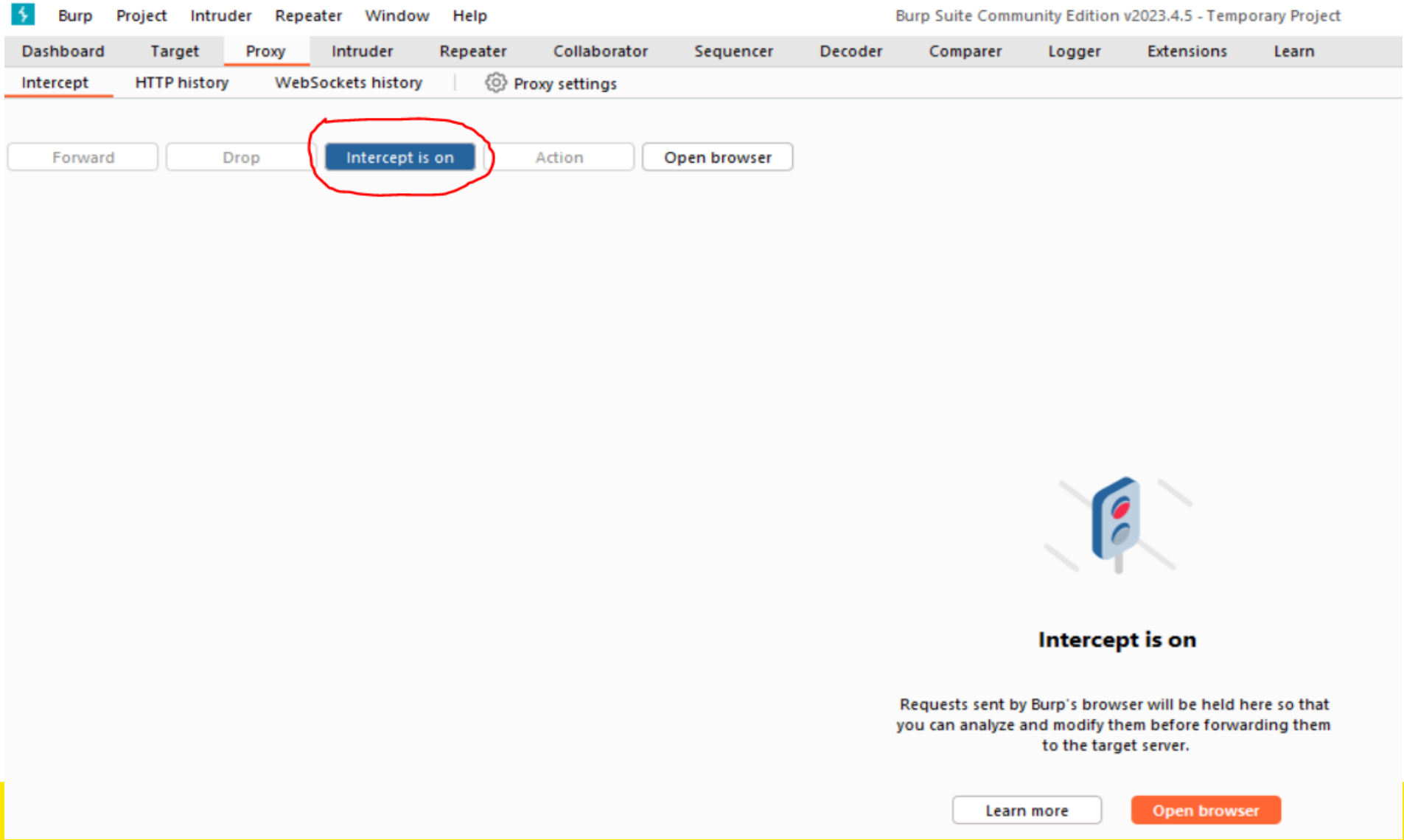
Using Burp Suite for Brute Force a Login Page



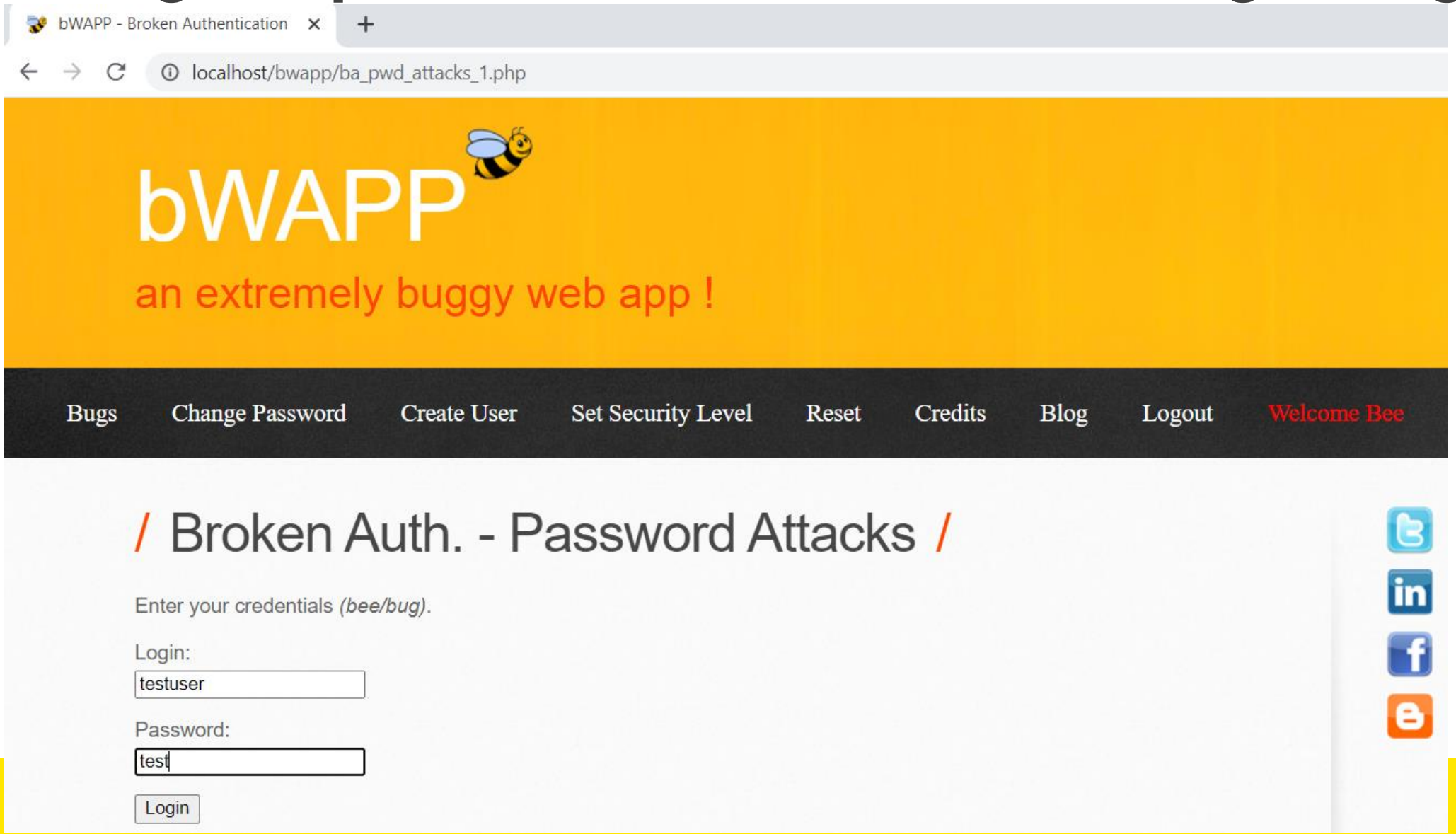
Using Burp Suite for Brute Force a Login Page



Using Burp Suite for Brute Force a Login Page



Using Burp Suite for Brute Force a Login Page



Using Burp Suite for Brute Force a Login Page

Burp Project Intruder Repeater Window Help — Burp Suite Community Edition v2023.4.5 - Temporary Project

Dashboard | Target | Proxy | Intruder | Repeater | Collaborator | Sequencer | Decoder | Comparer | Logger | Extensions | Learn | Settings

Intercept | HTTP history | WebSockets history | Proxy settings

Request to http://localhost:80 [127.0.0.1]

Forward | Drop | Intercept is on | Action | Open browser | Comment this item | HTTP/1

Pretty | Raw | Hex

```
1  POST /bwapp/ba_pwd_attacks_1.php HTTP/1.1
2  Host: localhost
3  Content-Length: 40
4  Cache-Control: max-age=0
5  sec-ch-ua: "Chromium";v="113", "Not-A.Brand";v="24"
6  sec-ch-ua-mobile: ?0
7  sec-ch-ua-platform: "Windows"
8  Upgrade-Insecure-Requests: 1
9  Origin: http://localhost
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/ 113.0.5672.127 Safari/537.36
12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,imag =0.8,application/signed-exchange;v=b3;q=0.7
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://localhost/bwapp/ba_pwd_attacks_1.php
18 Accept-Encoding: gzip, deflate
19 Accept-Language: en-US,en;q=0.9
20 Cookie: security_level=0; PHPSESSID=v21ln881419imitebekmlq02mr
21 Connection: close
22
23 login=testuser&password=test&form=submit
```

Scan
Send to Intruder Ctrl+I
Send to Repeater Ctrl+R
Send to Sequencer
Send to Comparer
Send to Decoder
Insert Collaborator payload
Request in browser
Engagement tools [Pro version only]
Change request method
Change body encoding
Copy URL
Copy as curl command (bash)
Copy to file
Paste from file
Save item
Don't intercept requests
Do intercept
Convert selection
URL-encode as you type
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V

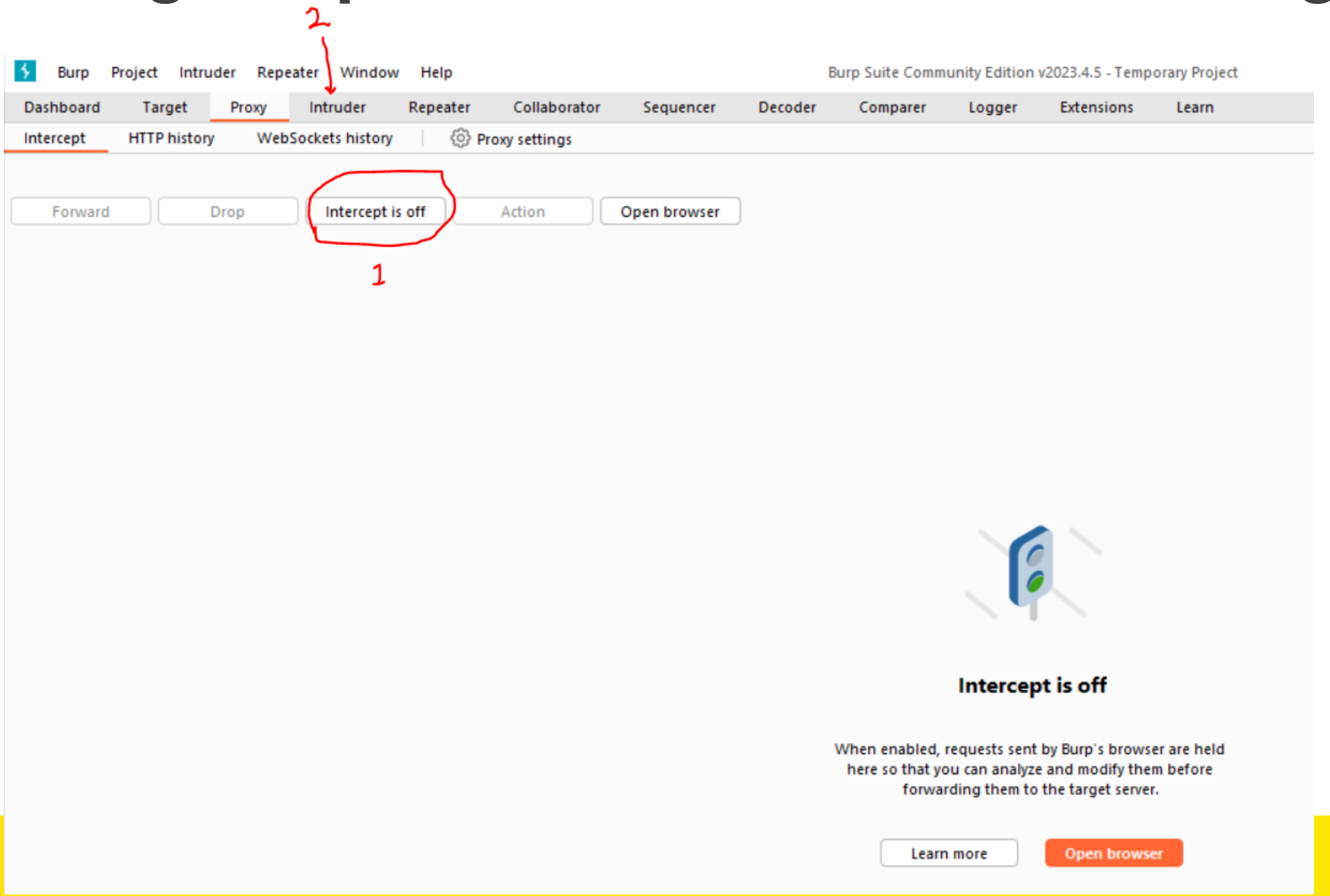
Inspector

Request attributes	2
Request query parameters	0
Request body parameters	3
Request cookies	2
Request headers	20

Search... 0 matches

UNSW CANBERRA

Using Burp Suite for Brute Force a Login Page



Using Burp Suite for Brute Force a Login Page

Burp Project Intruder Repeater Window Help — Burp Suite Community Edition v2023.4.5 - Temporary Project

Dashboard | Target | Proxy | Intruder | Repeater | Collaborator | Sequencer | Decoder | Comparer | Logger | Extensions | Learn | Settings

1 × 2 × 3 × 4 × 5 ×

Positions | Payloads | Resource pool | Settings

Choose an attack type — Start attack

Attack type: Cluster bomb 7

Payload positions

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

Target: http://localhost — Update Host header to match target

Add § 4 — Clear § 2 — Auto § — Refresh 6

```
1  POST /bwapp/ba_pwd_attacks_1.php HTTP/1.1
2  Host: localhost
3  Content-Length: 40
4  Cache-Control: max-age=0
5  sec-ch-ua: "Chromium";v="113", "Not-A.Brand";v="24"
6  sec-ch-ua-mobile: ?0
7  sec-ch-ua-platform: "Windows"
8  Upgrade-Insecure-Requests: 1
9  Origin: http://localhost
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.5672.127 Safari/537.36
12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://localhost/bwapp/ba_pwd_attacks_1.php
18 Accept-Encoding: gzip, deflate
19 Accept-Language: en-US,en;q=0.9
20 Cookie: security_level=0; PHPSESSID=v21ln881419imitebekmlq02mr
21 Connection: close
22
23 login=testuser&password=test&form=submit
```

2 5

Using Burp Suite for Brute Force a Login Page

Prepare two files for default username and password to use in payloads

*usernames.txt - Notepad

File Edit Format View Help

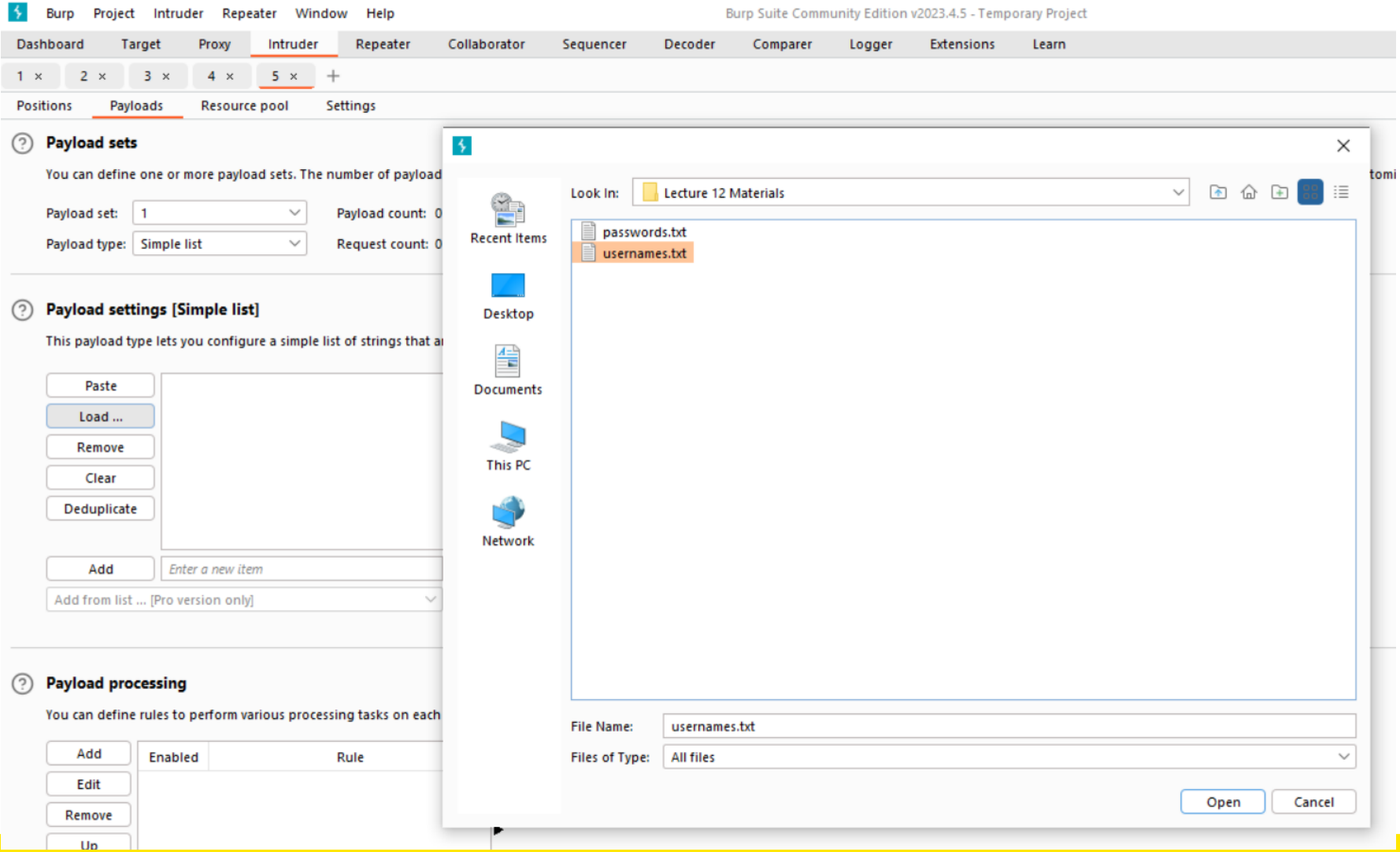
```
root
admin
adminstrator
manager
bee
test
bug
bwapp
```

*passwords.txt - Notepad

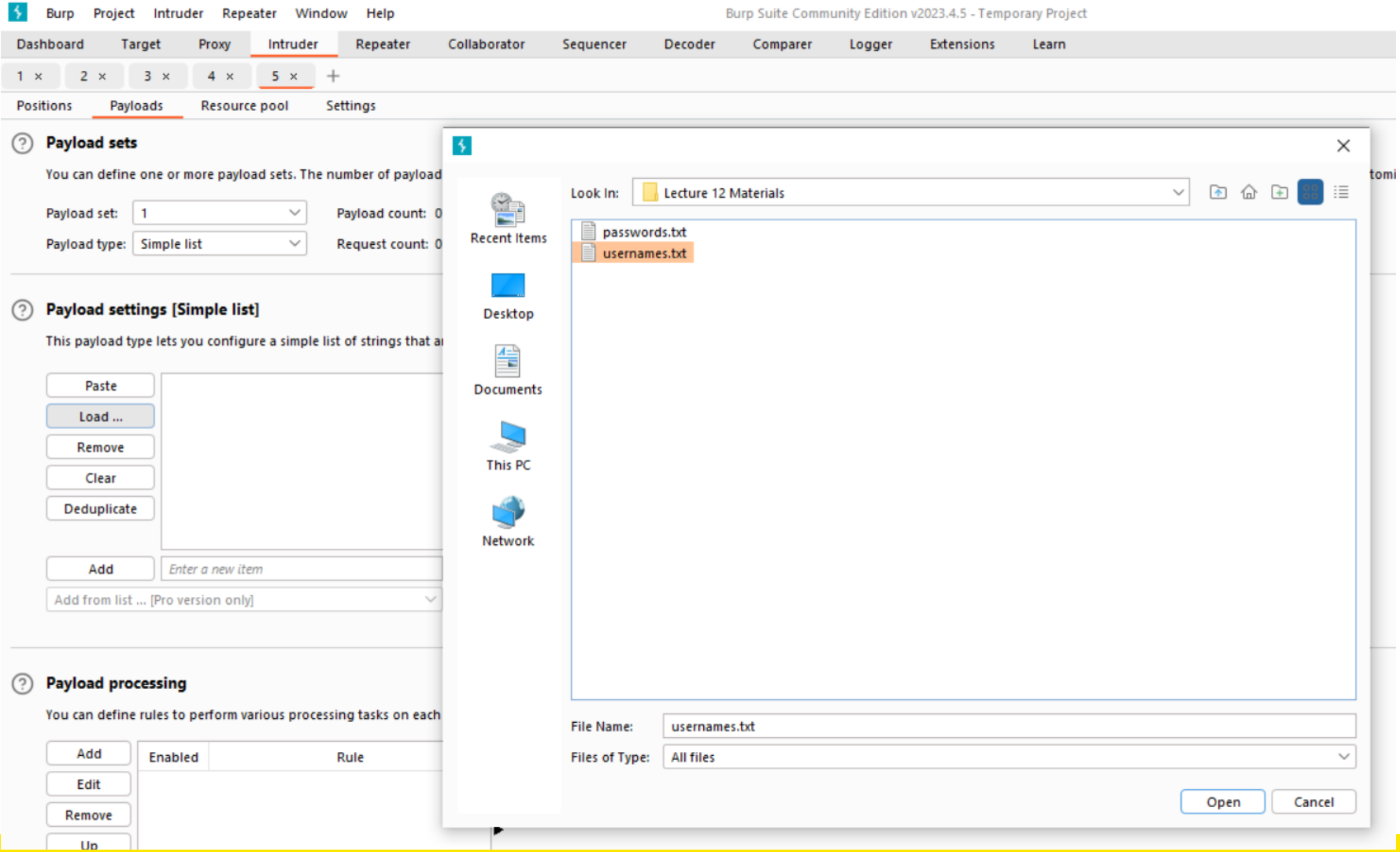
File Edit Format View Help

```
123456
test
test1234
bug
bee
bee1234
demo
```

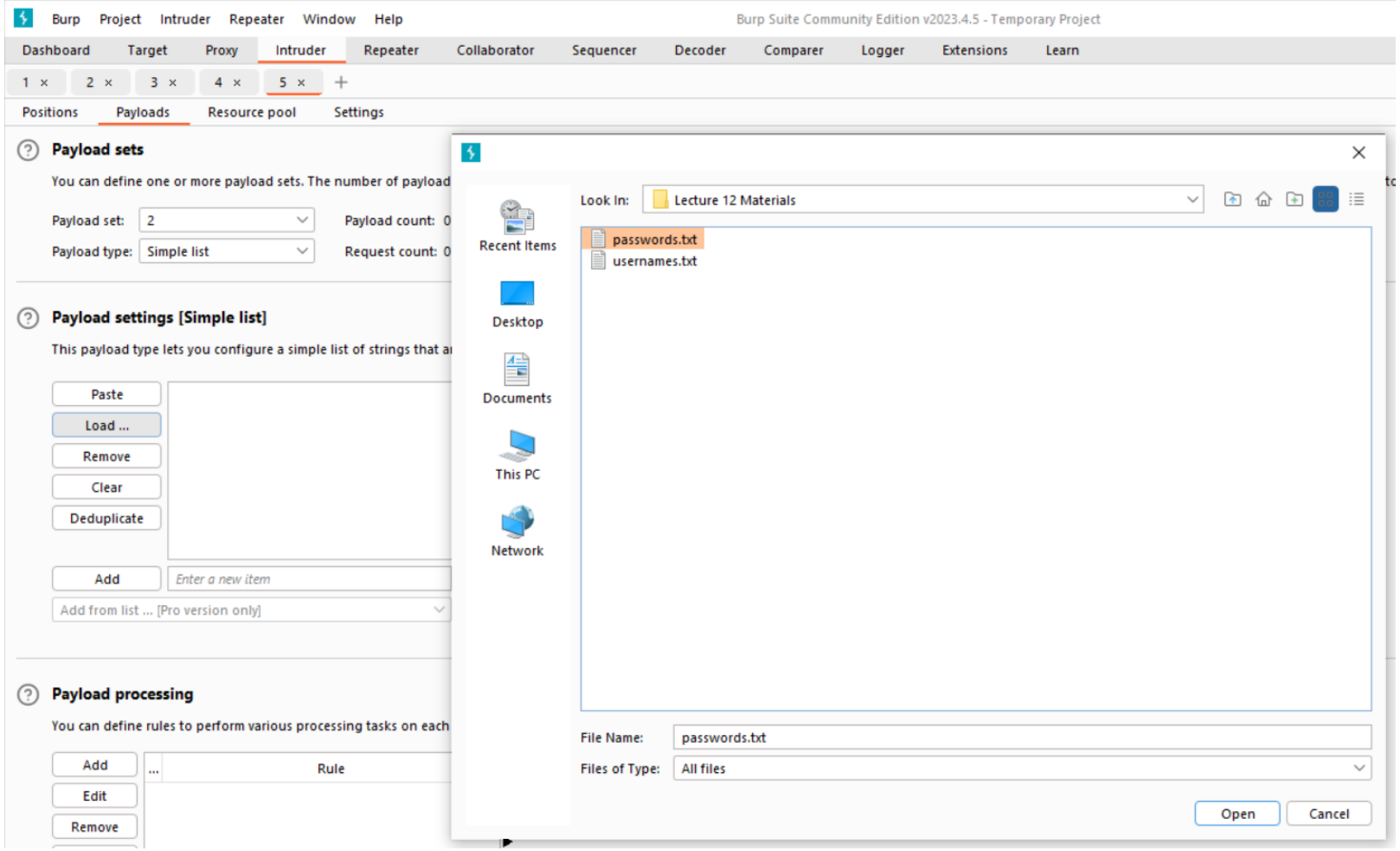
Using Burp Suite for Brute Force a Login Page



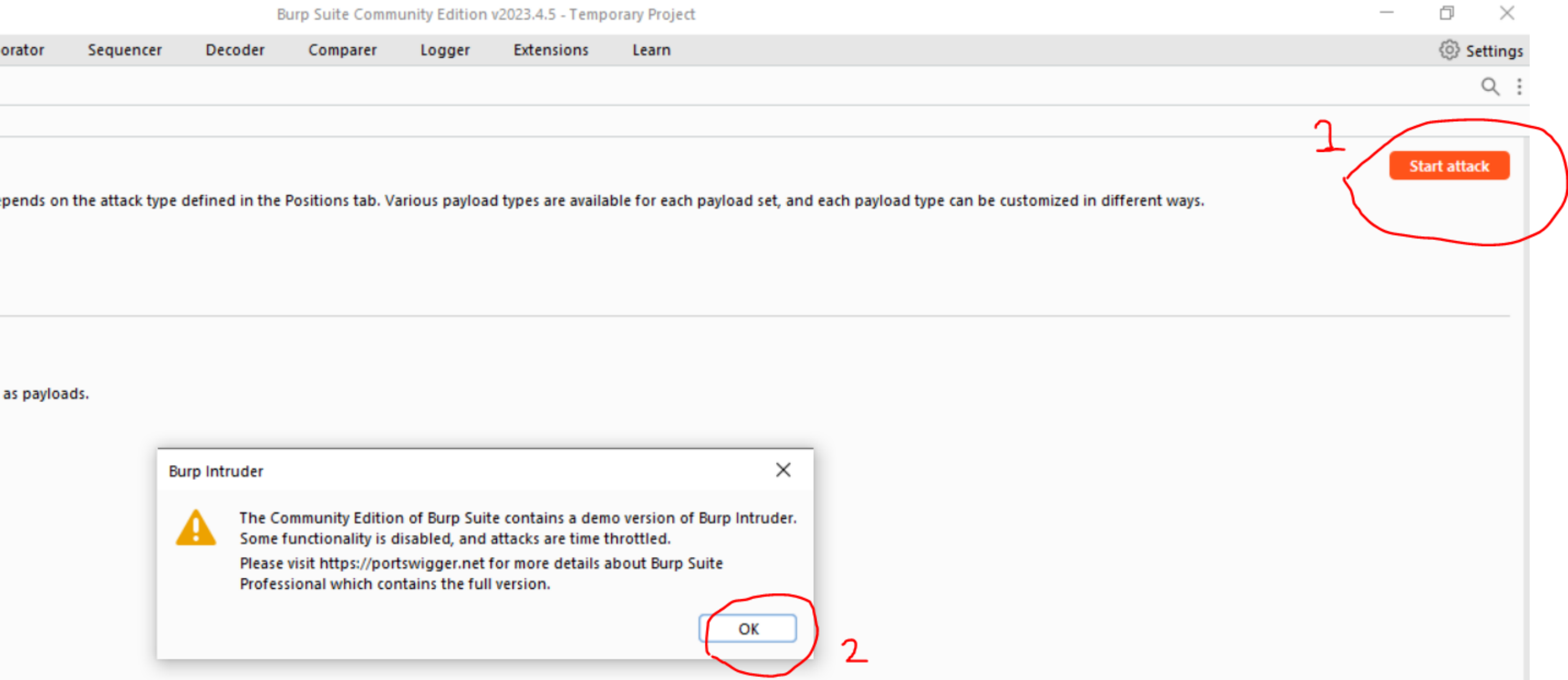
Using Burp Suite for Brute Force a Login Page



Using Burp Suite for Brute Force a Login Page



Using Burp Suite for Brute Force a Login Page



Using Burp Suite for Brute Force a Login Page

Attack Save Columns

5. Intruder attack of http://localhost - Temporary attack - Not saved to project file

Results Positions Payloads Resource pool Settings

Filter: Showing all items

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
28	root	bug	200			13844	
29	admin	bug	200			13844	
30	adminstrator	bug	200			13844	
31	manager	bug	200			13844	
32	bee	bug	200			13813	
33	test	bug	200			13844	

Request Response

Pretty Raw Hex Render

```
76          <button type="submit" name="form" value="submit">
              Login
            </button>

77
78        </form>
79
80        </br >
81        <font color="green">
            Successful login!
          </font>

82      </div>
83
84      <div id="side">

85
86        <a href="http://twitter.com/MME_IT" target="blank_" class="button">
            <img src="./images/twitter.png">
          </a>
87        <a href="http://be.linkedin.com/in/malikmesellem" target="blank_" class="button">
            <img src="./images/linkedin.png">
          </a>
88        <a href="http://www.facebook.com/pages/MME-IT-Audits-Security/104153019664877" target="blank_" class="button">
            <img src="./images/facebook.png">
          </a>
89        <a href="http://itsecgames.blogspot.com" target="blank_" class="button">
            <img src="./images/blogger.png">
          </a>
90
91      </div>

92
93      <div id="disclaimer">
94
```

Search...

Finished

UNSW CANBERRA

Using Burp Suite for Brute Force a Login Page

Attack Save Columns — 5. Intruder attack of http://localhost - Temporary attack - Not saved to project file

Results | Positions | Payloads | Resource pool | Settings

Filter: Showing all items

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
28	root	bug	200			13844	
29	admin	bug	200			13844	
30	adminstrator	bug	200			13844	
31	manager	bug	200			13844	
32	bee	bug	200			13813	
33	test	bug	200			13844	

Request | Response

Pretty | Raw | Hex | Render

```
76        <button type="submit" name="form" value="submit">
            Login
          </button>

77
78      </form>
79
80      </br >
81      <font color="green">
          Successful login!
        </font>

82    </div>
83
84    <div id="side">

85
86      <a href="http://twitter.com/MME_IT" target="blank_" c
          <img src="./images/twitter.png">
        </a>
87      <a href="http://be.linkedin.com/in/malikmesellem" tar
          <img src="./images/linkedin.png">
        </a>
88      <a href="http://www.facebook.com/pages/MME-IT-Audits-                    "blank_" class="button">
          <img src="./images/facebook.png">
        </a>
89      <a href="http://itsecgames.blogspot.com" target="blan
          <img src="./images/blogger.png">
        </a>
90
91    </div>

92
93    <div id="disclaimer">
94
```

- Scan
- Send to Intruder Ctrl+I
- Send to Repeater Ctrl+R
- Send to Sequencer
- Send to Comparer
- Send to Decoder
- Show response in browser
- Request in browser
- Engagement tools [Pro version only]
- Copy Ctrl+C
- Copy URL
- Copy as curl command (bash)
- Copy to file
- Save item
- Convert selection
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Message editor documentation
- Intruder results documentation

Show response in browser

To show this response in your browser, copy the URL below and paste into a browser that is configured to use Burp as its proxy.

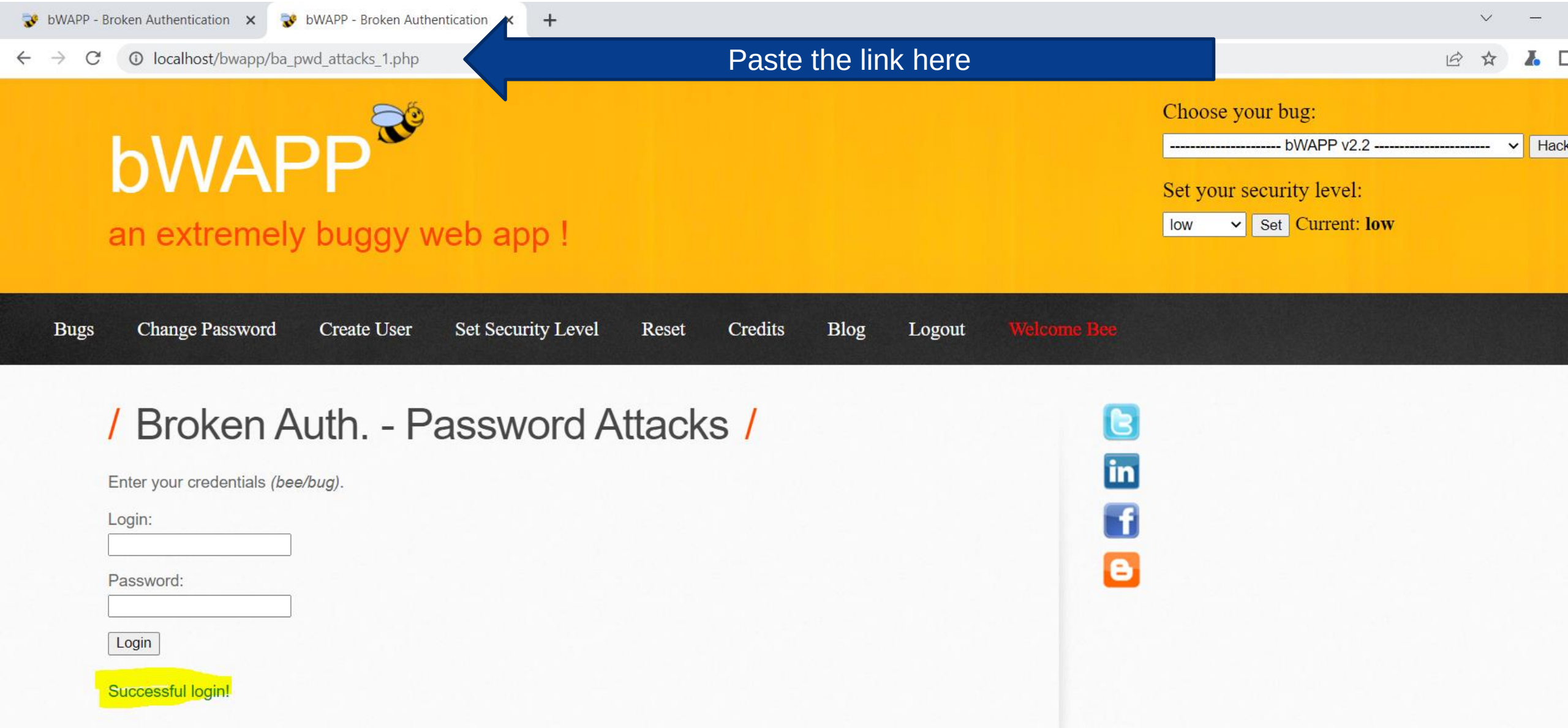
http://burpsuite/show/4/2xvy4cqqda6kjxcrj0um4q71yl6hr09d [Copy]

In future, just copy the URL and don't show this dialog [Close]

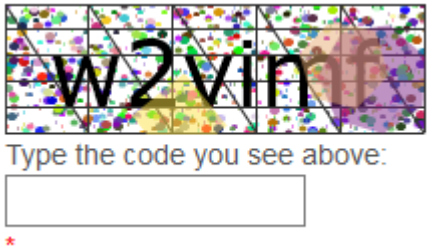
Search... 0 matches

Finished

Using Burp Suite for Brute Force a Login Page

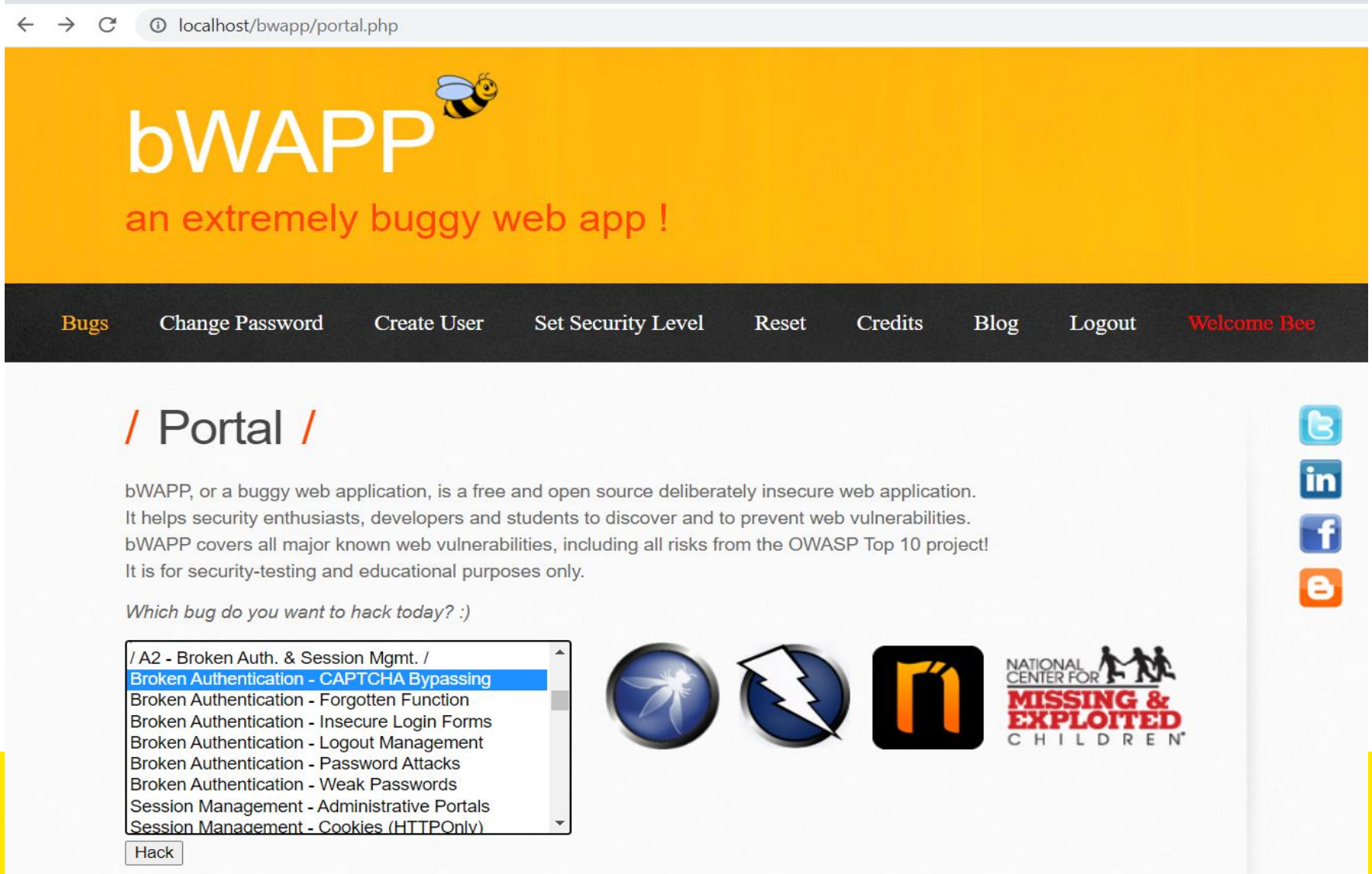


Captcha



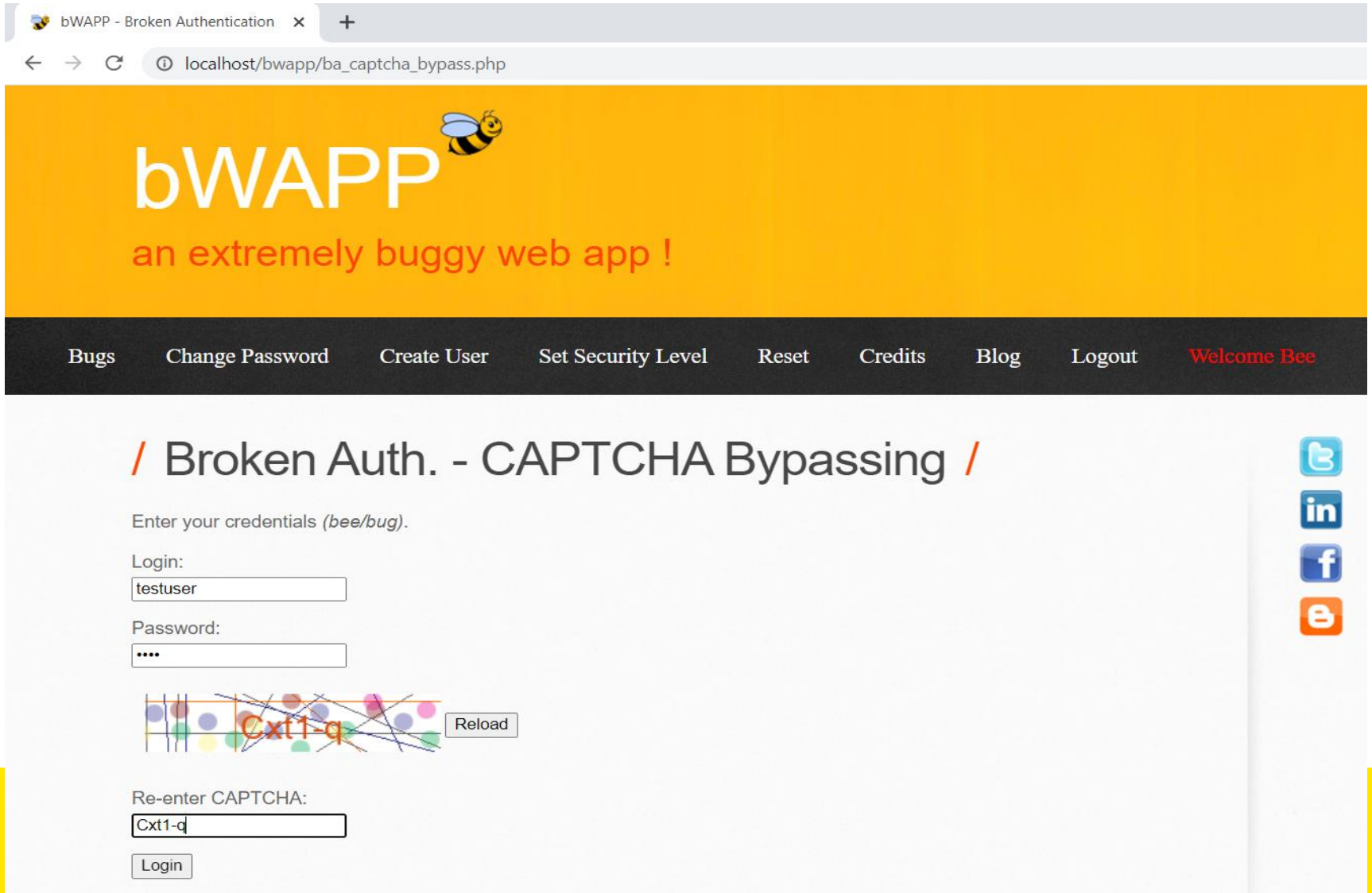
- CAPTCHA was first invented in 1997
- It is used to prevent the bot, malware and attacks such as brute force.
- Application of Captcha:
  - Form Authentication: For login and sign up it can be used to ensure that the end user is human.
  - Preventing Fake Registrations: With the captcha we can prevent bots from creating an account on a system.
  - Preventing Fake comments: This way bot would not be able to do Comment on a system.
  - NetBanking and financial institutes: To ensure that Authentication is only done by humans and this way manipulation of transactions can be prevented.
- CAPTCHA bypass Is too easy with modern bots
- Alternatives:
  - Using biometrics
  - Multi-Factor Authentication
  - Ad Fraud Solutions

bWAPP – Bypass Captcha



UNSW CANBERRA

bWAPP – Bypass Captcha



bWAPP – Bypass Captcha

Burp Project Intruder Repeater Window Help — Burp Suite Community Edition v2023.4.5 - Temporary Project

Dashboard | Target | Proxy | Intruder | Repeater | Collaborator | Sequencer | Decoder | Comparer | Logger | Extensions | Learn | Settings

Intercept | HTTP history | WebSockets history | Proxy settings

Request to http://localhost:80 [127.0.0.1]

Forward | Drop | Intercept is on | Action | Open browser | Comment this item | HTTP/1

Pretty | Raw | Hex

```
1  POST /bwapp/ba_captcha_bypass.php HTTP/1.1
2  Host: localhost
3  Content-Length: 60
4  Cache-Control: max-age=0
5  sec-ch-ua: "Chromium";v="113", "Not-A.Brand";v="24"
6  sec-ch-ua-mobile: ?0
7  sec-ch-ua-platform: "Windows"
8  Upgrade-Insecure-Requests: 1
9  Origin: http://localhost
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.  (KHTML, like Gecko) Chrome/113.0.5672.127 Safari/537.36
12 Accept: text/html,application/xhtml+xml ,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://localhost/bwapp/ba_capt
18 Accept-Encoding: gzip, deflate
19 Accept-Language: en-US,en;q=0.9
20 Cookie: security_level=0; PHPSESSID=c66
21 Connection: close
22
23 login=testuser&password=test&captcha_us
```

- Scan
- Send to Intruder Ctrl+I
- Send to Repeater Ctrl+R
- Send to Sequencer
- Send to Comparer
- Send to Decoder
- Insert Collaborator payload
- Request in browser
- Engagement tools [Pro version only]
- Change request method
- Change body encoding
- Copy URL
- Copy as curl command (bash)
- Copy to file
- Paste from file
- Save item
- Don't intercept requests
- Do intercept
- Convert selection
- URL-encode as you type
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V

Inspector

Request attributes	2
Request query parameters	0
Request body parameters	4
Request cookies	2
Request headers	20

Search... 0 matches

bWAPP – Bypass Captcha

Burp Project Intruder Repeater Window Help — Burp Suite Community Edition v2023.4.5 - Temporary Project

Dashboard | Target | Proxy | Intruder | Repeater | Collaborator | Sequencer | Decoder | Comparer | Logger | Extensions | Learn | Settings

1 × 2 × 3 × 4 × 5 ×

Positions | Payloads | Resource pool | Settings

Choose an attack type

7 Start attack

Attack type: Cluster bomb 6

Payload positions

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

5

Target: http://localhost

Update Host header to match target

3 Add §

1 Clear §

Auto §

Refresh

```
1  POST /bwapp/ba_captcha_bypass.php HTTP/1.1
2  Host: localhost
3  Content-Length: 60
4  Cache-Control: max-age=0
5  sec-ch-ua: "Chromium";v="113", "Not-A.Brand";v="24"
6  sec-ch-ua-mobile: ?0
7  sec-ch-ua-platform: "Windows"
8  Upgrade-Insecure-Requests: 1
9  Origin: http://localhost
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.5672.127 Safari/537.36
12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
13 Sec-Fetch-Site: same-origin 4
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://localhost/bwapp/ba_captcha_bypass.php
18 Accept-Encoding: gzip, deflate
19 Accept-Language: en-US,en;q=0.9
20 Cookie: security_level=0; PHPSESSID=c66qnujiv4cgllmg3b46mhd9bd
21 Connection: close
22
23 login=§testuser§&password=§test§&captcha_user=Cxtl-q&form=submit
```

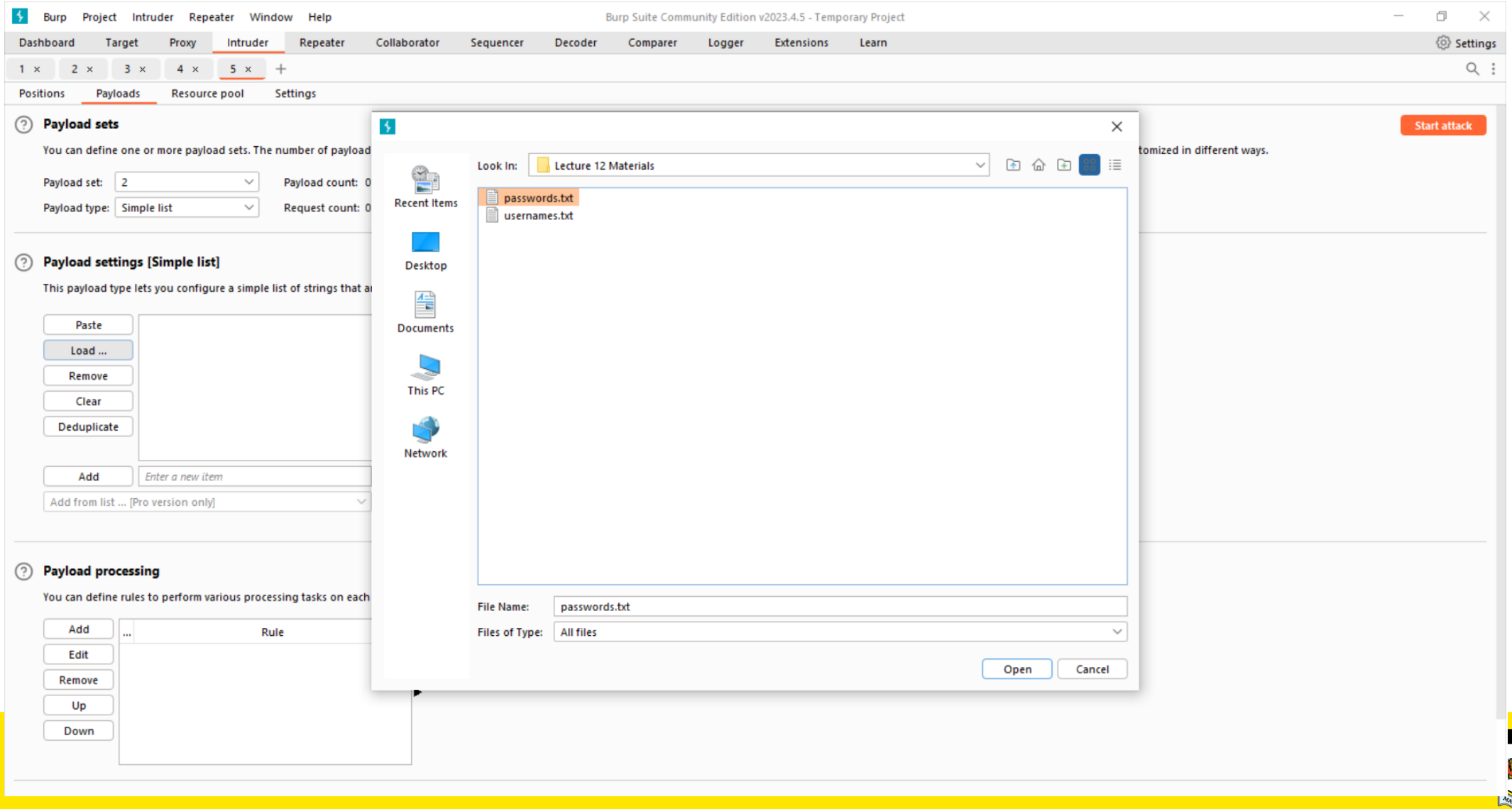
2

Search... 0 matches Clear

2 payload positions Length: 963

UNSW CANBERRA

bWAPP – Bypass Captcha



bWAPP – Bypass Captcha

Attack Save Columns

7. Intruder attack of http://localhost - Temporary attack - Not saved to project file

Results Positions Payloads Resource pool Settings

Filter: Showing all items

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
29	admin	bug	200			13887	
30	adminstrator	bug	200			13887	
31	manager	bug	200			13887	
32	bee	bug	200			13856	

Request Response

Pretty Raw Hex Render

```
          <br />
74        <input type="password" id="password" name="password" size="20" autocomplete="off" />
        </p>
75
76      <p>
          <iframe src="captcha_box.php" scrolling="no" frameborder="0" height="70" width="350">
          </iframe>
        </p>
77
78      <p>
          <label for="captcha_user">
            Re-enter CAPTCHA:
          </label>
          <br />
79        <input type="text" id="captcha_user" name="captcha_user" value="" autocomplete="off" />
        </p>
80
81      <button type="submit" name="form" value="submit">
          Login
        </button>
82
83         <font color="green">
          Successful login!
        </font>
84
85    </form>
86
87  </div>
88
89  <div id="side">
90
91    <a href="http://twitter.com/MME_IT" target="blank_" class="button">
        <img src="./images/twitter.png">
      </a>
92    <a href="http://be.linkedin.com/in/malikmesellem" target="blank_" class="button">
```


UNSW CANBERRA

OWASP Testing Guide: Authentication

4. Testing for Bypassing Authentication Schema

- Bypass by simply skipping the log in page
- Directly calling an internal page
- Parameter modification
- Session ID prediction

5. Testing for Vulnerable Remember Password

- The "remember my password" mechanism can be implemented with one of the following methods:
- Set autocomplete="off" for the username and password field including captcha field
- The password must be hashed/encrypted and not sent in the clear.

6. Testing for Browser cache weakness

- **Browse History**
  - Entering sensitive information into the application and logging out. Then the tester clicks the "Back" button of the browser to check whether previously displayed sensitive information can be accessed whilst unauthenticated.

OWASP Testing Guide: Authentication

7. Testing for Weak password policy

- Password complexity
- Password history + password changing period
- Password expires
- Different between last password and next password
- Prevent user to use username or other account in formation as a password

8. Weak password change or reset functionalities

- if users, other than administrators, can change or reset passwords for accounts other than their own.

Broken Session Management

- **Session Management:**
  - It is a process by which server maintains the state of an entity interacting with it
  - This is required for a server to remember how to react to subsequent requests throughout a transaction.
- **Broken Session Management**
  - Inadequate Session Management policies
  - Sending session cookie over an insecure channel
  - Insecure session generation
  - Session fixation vulnerability
  - No protection of session cookie

Broken Session Management

- **Result of Broken Session Management**
  - By-pass authentication
  - Complete control of accounts
  - Account theft, sensitive end-user (customer) data could be stolen
  - Reputational damage and revenue loss.

OWASP Testing Guide: Session Management

1. **Testing for Bypassing Session Management Schema**

- **Cookie Analysis**
  - How many cookies are used in the application
  - Which parts of the application generate and/or modify the cookie?
  - Which parts of the application require this cookie in order to be accessed and utilized?

- **Session ID Predictability and Randomness**
  - Session Time-out
  - What elements of the Session IDs are time-linked?

OWASP Testing Guide: Session Management

2. Testing for cookies attributes

- **Secure Attribute**
  - ";secure“ (Cookie will only be sent over SSL/TLS)
- **HttpOnly**
  - ";HttpOnly“ (JS cannot access cookie. Prevent client side script attack)
- **Domain Attribute**
  - "; domain=app.mysite.com" and NOT "; domain=.mysite.com"
- **Path Attribute**
  - "; path=/myapp/" and NOT "; path=/".
- **Expires Attribute**
  - "; expires=Sun, 31-Jul-2016 13:45:29 GMT"

OWASP Testing Guide: Session Management

3. Testing for Session Fixation

4. Testing for Exposed Session Variables

- How are Session IDs transferred? e.g., GET, POST, Form Field (including hidden fields)
- Are GET requests incorporating the Session ID used?
- If POST is used, can it be interchanged with GET?

5. Testing for logout functionality

- A secure session termination requires at least the following components:
  - Availability of user interface controls that allow the user to manually log out.
  - Session termination after a given amount of time without activity (session timeout).

6. Test Session Timeout

- The log out function effectively destroys all session token
- The server performs proper checks on the session state, disallowing an attacker to replay previously destroyed session identifiers

Example

- Gibson Security detailed vulnerabilities in the snapchat service, which was dismissed as a purely theoretical attack. A week later, brute force enumeration had revealed 4.6 million usernames and phone numbers.

- Laxman Muthiyah found that it was possible for a malicious user to use a request to assign admin permissions to himself for a particular Facebook page. A sample request can be found here:

https://www.horangi.com/blog/real-life-examples-of-web-vulnerabilities

- In 2012, a foreign hacker was reported to have stolen 387,000 credit card numbers and 3.6 million Social Security numbers from the South Carolina Department of Revenue.

- In 2013 over 34 million Americans reported some form of identity theft.

- Three quarters through 2014 there is already a reported 568 data breaches with over 75 million records compromised and hundreds of millions of users affected. This is up from the 439 breaches in 2013.

- Identity theft isn't a possibility, it's a reality that is happening all the time and identity theft is at the core of the 2nd of OWASP's top 10 most critical web security risks of 2013; Broken Authentication and Session Management.

Final Exam: Conditions

- **Exam Duration:** 180 minutes
- **Exam Condition:** Open book, invigilated, not allowed to use ChatGPT or any communication software (like email, WhatsApp, Teams, etc).
- **Bring your own Laptop**
- **Ensure the following software packages are installed on your device:**
  - PHP
  - MySQL
  - Node.js
  - Express.js
  - Laravel
  - Postman Desktop
  - Laragon or XAMPP
- **Make sure that port TCP 3000, TCP 3306 and TCP 8000 are open on your firewall.**
  - Check with your lab demonstrator during the lab if your laptop is ready for the exam. You should be able to run the codes for Lab 8 and Lab 9.

Final Exam: Format

- **Exam will be on Moodle:** The exam will consist of four tasks that involve coding as well as answering questions.
- **Delivery:**
  - For each finished task in the exam, you must:
    - Create a folder in which you store the task related code and documents (code, data files, images, readme, or any other files that are needed to run the codes).
    - Assign the folder the name of the task. For instance, for Task1 (no space), you name the folder Task 1.
  - At the end of the exam, compress all tasks folders into a zip file
  - Name the file **Exam-***StudentID***-***StudentName***.zip**.
  - Submit the zip file to the exam submission box.
  - More instructions will be provided on the day of the exam.

Final Exam: Preparation

- The best source for preparation would be labs and lectures material.
- Review all techniques and sample codes covered in lectures and labs solutions.
- All exam tasks are practical.
- Use the techniques, coding languages, and tools specified in the task description.
  - For example, if you are asked to create a front end code using HTML, CSS, or JS, you should not use PHP.
  - If it is mentioned to use CSS (or Bootstrap) you are allowed to use Bootstrap, otherwise, you need to stick with CSS for styling.
- If you are not able to finish one task, try to complete some of its subtasks to achieve partial marks. The marker should be able to verify the subtask is complete.

Final Note

- **Deadline for Project 2 is Sunday 4th of June, 23h59.**
  - Submission boxes are now open.
  - One submission box for submission of the group part(one submission per group suffice).
  - One submission for individual part.
- **Due to compensation day, Quiz 3 will take place during week 13's labs**
  - Quiz will cover material discussed in weeks 7 – 12
- **Lab 12 would be a help session for Project 2. If you need an informal feedback, please come to the lab and talk to your lab demonstrator.**

QUESTIONS?