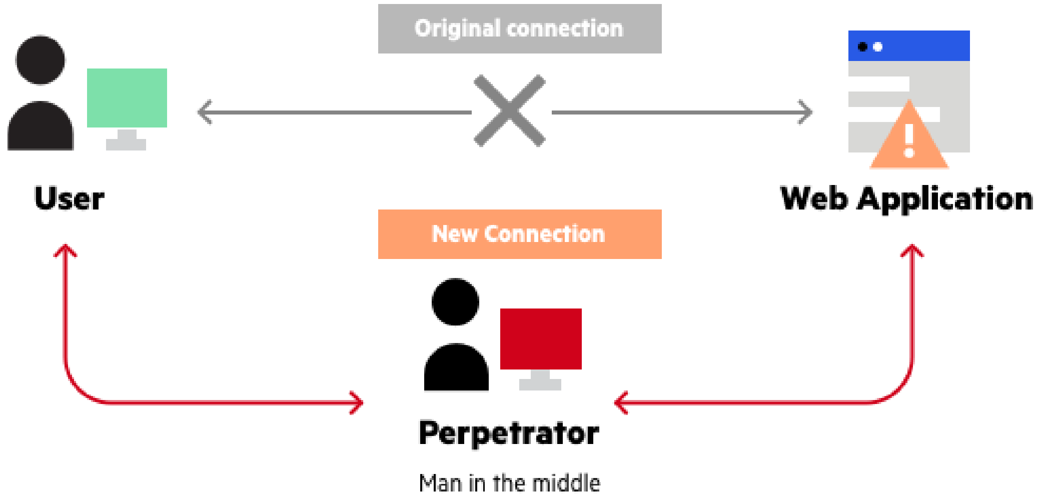
MAN-IN-THE-MIDDLE ATTACK

9th Sep 2022

OVERVIEW

A man-in-the-middle (MITM) attack is a general term for when a perpetrator positions himself in a conversation between a user and an application—either to eavesdrop or to impersonate one of the parties, making it appear as if a normal exchange of information is underway.

Information obtained during an attack could be used for many purposes, including identity theft, unapproved fund transfers or an illicit password change.



INSTALLATION OF BURP SUITE (COMMUNITY EDITION)

Step 1: Download

Download the latest version of Burp Suite Community Edition.

Step 2: Install

Run the installer and launch Burp Suite. When asked to select a project file and configuration, just click **Next** and then **Start Burp** to skip this for now.

USAGES OF BURP SUITE

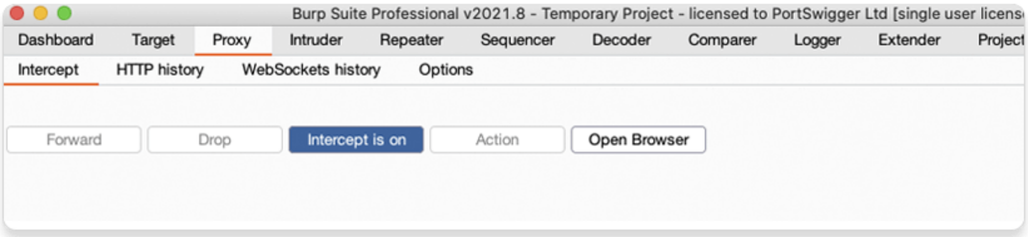
- **Intercepting & Modifying HTTP traffic with Burp Proxy.**
- **Burp Intruder & many more.**

INTERCEPTING & MODIFYING A REQUEST

Burp Proxy lets you intercept HTTP requests and responses sent between Burp's browser and the target server. This enables you to study how the website behaves when you perform different actions.

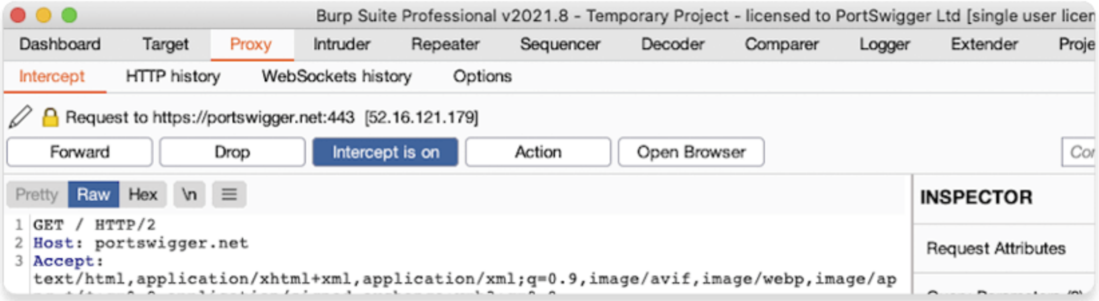
Step 1: Launch Burp's browser

- **Go to the Proxy > Intercept tab.**
- **Click the Intercept is off button, so it toggles to Intercept is on.**



Step 2: Intercept a request

Using Burp's browser, try to visit https://portswigger.net and observe that the site doesn't load. Burp Proxy has intercepted the browser’s HTTP request before it could reach the server. You can see this intercepted request on the Proxy > Intercept tab.



Request to https://portswigger.net:443 [52.16.121.179]

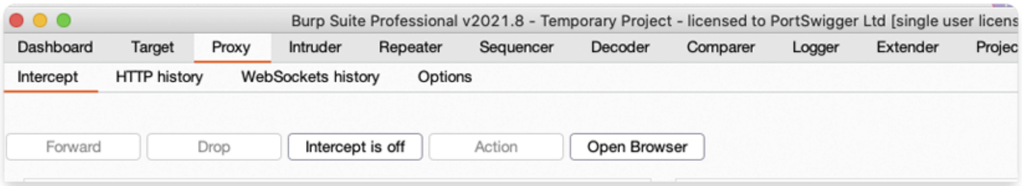
```
GET / HTTP/2
Host: portswigger.net
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
```

Step 3: Forward the request

Click the **Forward** button several times to send the intercepted request, and any subsequent ones, until the page loads in Burp's browser.

Step 4: Switch off the interception

Due to the number of requests browsers typically send, you often won't want to intercept every single one of them. Click the **Intercept is on** the button so that it now says **Intercept is off**.



BURP INTRUDER

Step 1: Access the lab

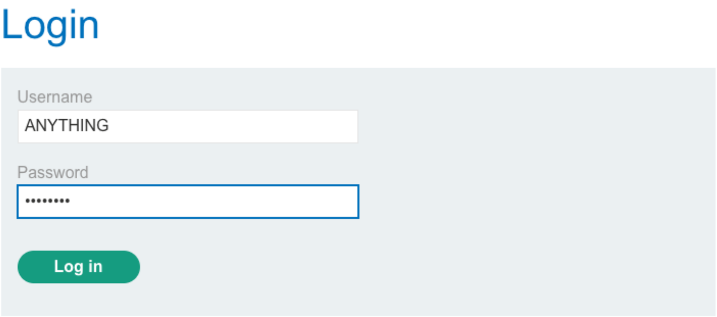
Open Burp's browser, and use it to access the following URL:

https://portswigger.net/web-security/authentication/password-based/lab-username-enumeration-via-different-responses

Click **Access the lab** and log in to your PortSwigger account if prompted. This opens your instance of a deliberately vulnerable blog website.

Step 2: Try to log in

Click **My account**, then try to log in using an invalid username and password. In Burp Suite, go to the Proxy > HTTP history tab. This shows all of the requests you have made in Burp's browser since opening it.

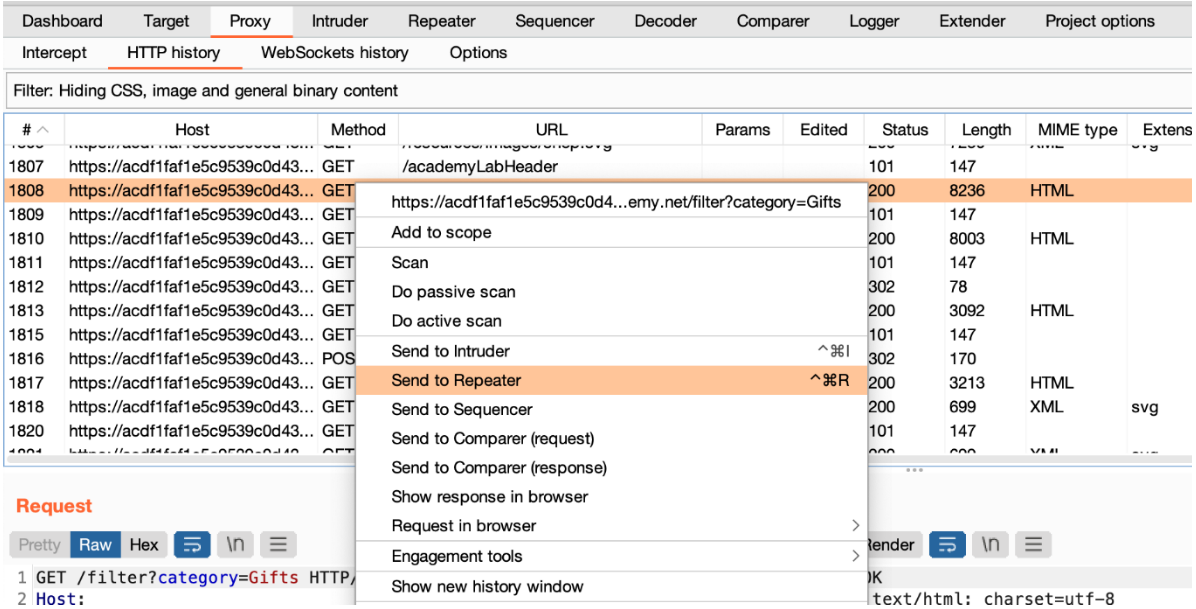


Login

Username: ANYTHING

Password: ••••••••

Find the POST /login request and send it to Burp Intruder.



Filter: Hiding CSS, image and general binary content

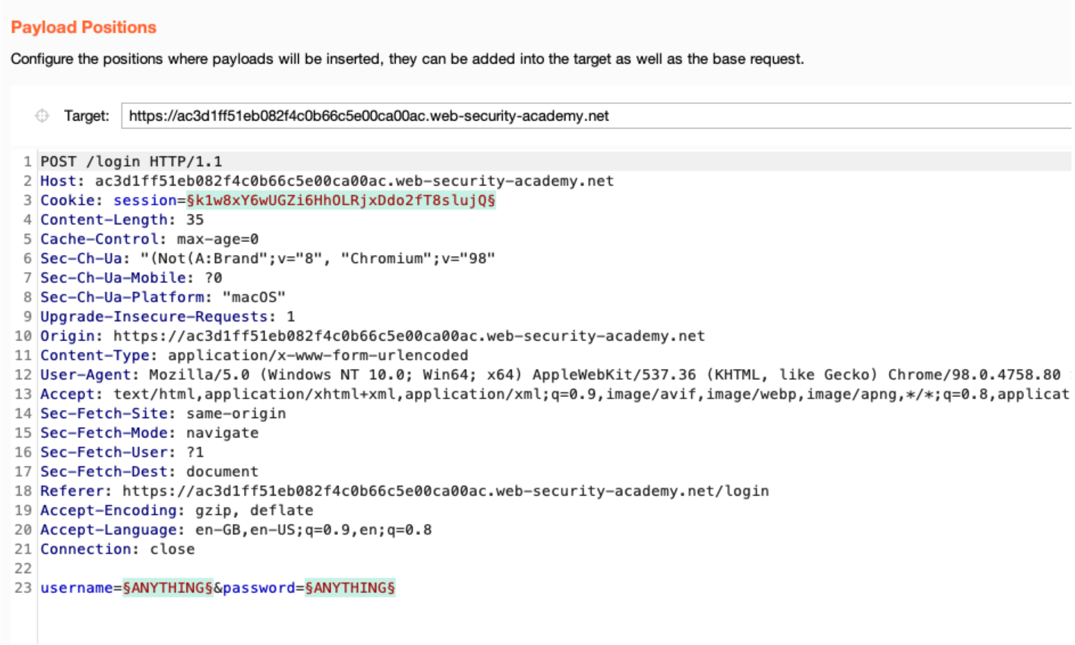
#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extens
1807	https://acdf1faf1e5c9539c0d43...	GET	/academyLabHeader			101	147		
1808	https://acdf1faf1e5c9539c0d43...	GET				200	8236	HTML	
1809	https://acdf1faf1e5c9539c0d43...	GET				101	147		
1810	https://acdf1faf1e5c9539c0d43...	GET				200	8003	HTML	
1811	https://acdf1faf1e5c9539c0d43...	GET				101	147		
1812	https://acdf1faf1e5c9539c0d43...	GET				302	78		
1813	https://acdf1faf1e5c9539c0d43...	GET				200	3092	HTML	
1815	https://acdf1faf1e5c9539c0d43...	GET				101	147		
1816	https://acdf1faf1e5c9539c0d43...	POS				302	170		
1817	https://acdf1faf1e5c9539c0d43...	GET				200	3213	HTML	
1818	https://acdf1faf1e5c9539c0d43...	GET				200	699	XML	svg
1820	https://acdf1faf1e5c9539c0d43...	GET				101	147		

```
GET /filter?category=Gifts HTTP/
Host:
```

```
K
text/html: charset=utf-8
```

Step 3: Set the payload positions

Notice that Burp Intruder has automatically inserted § characters in various positions throughout the request. These mark the beginning and end of a payload position, where Burp Intruder will attempt to insert payloads during the attack.



Payload Positions

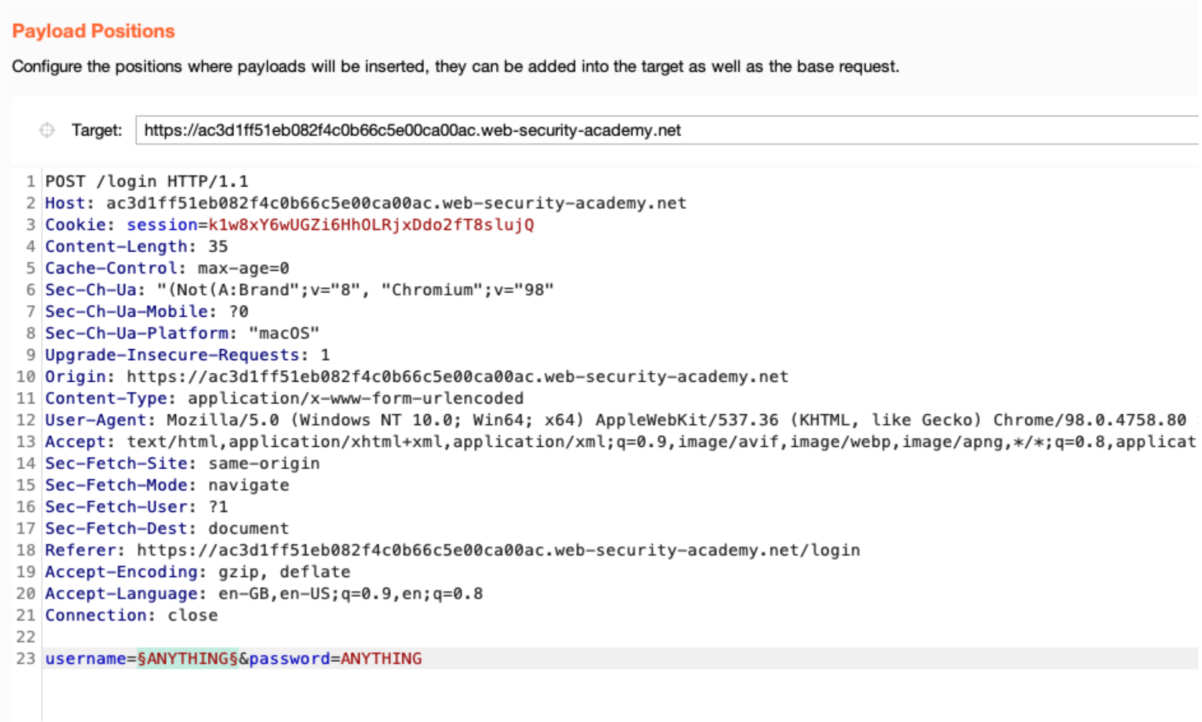
Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net

```
POST /login HTTP/1.1
Host: ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net
Cookie: session=§k1w8xY6wUGZi6HhOLRjxDdo2fT8slujQ§
Content-Length: 35
Cache-Control: max-age=0
Sec-Ch-Ua: "(Not(A:Brand";v="8", "Chromium";v="98"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
Origin: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.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/98.0.4758.80
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicat
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net/login
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close

username=§ANYTHING§&password=§ANYTHING§
```

For this attack, we only need a single payload position in the username parameter. Click **Clear §** to clear the default positions. Highlight the value of the username parameter, then click **Add §**.



Payload Positions

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

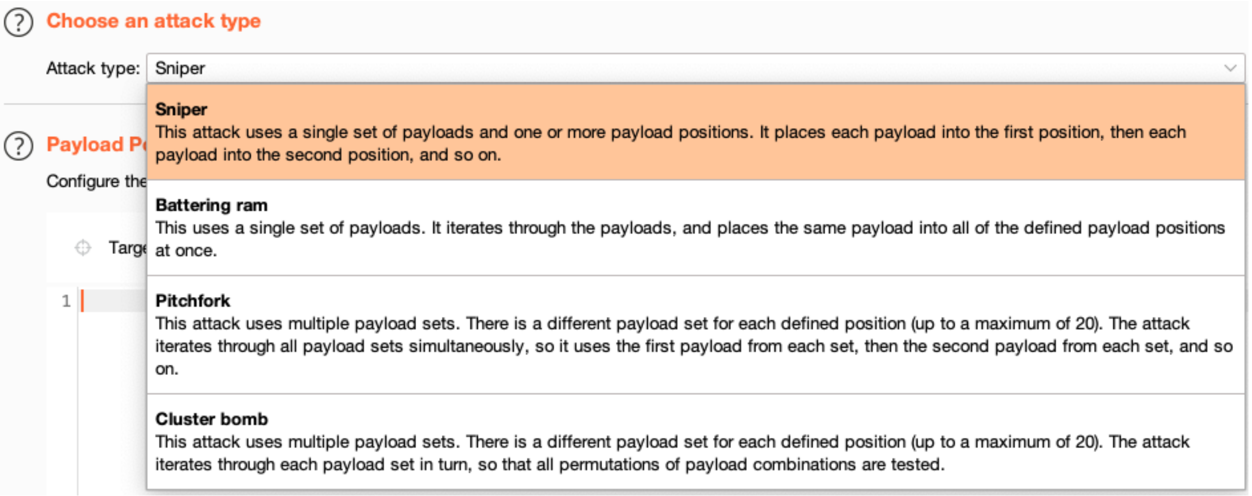
Target: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net

```
POST /login HTTP/1.1
Host: ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net
Cookie: session=k1w8xY6wUGZi6HhOLRjxDdo2fT8slujQ
Content-Length: 35
Cache-Control: max-age=0
Sec-Ch-Ua: "(Not(A:Brand";v="8", "Chromium";v="98"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
Origin: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.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/98.0.4758.80
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicat
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net/login
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close

username=§ANYTHING§&password=ANYTHING
```

Step 4: Select an attack type

At the top of the screen, you can select different attack types. For now, just make sure this is set to **Sniper**.



Choose an attack type

Attack type: Sniper

Sniper
This attack uses a single set of payloads and one or more payload positions. It places each payload into the first position, then each payload into the second position, and so on.

Battering ram
This uses a single set of payloads. It iterates through the payloads, and places the same payload into all of the defined payload positions at once.

Pitchfork
This attack uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through all payload sets simultaneously, so it uses the first payload from each set, then the second payload from each set, and so on.

Cluster bomb
This attack uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through each payload set in turn, so that all permutations of payload combinations are tested.

Step 5: Add the payloads

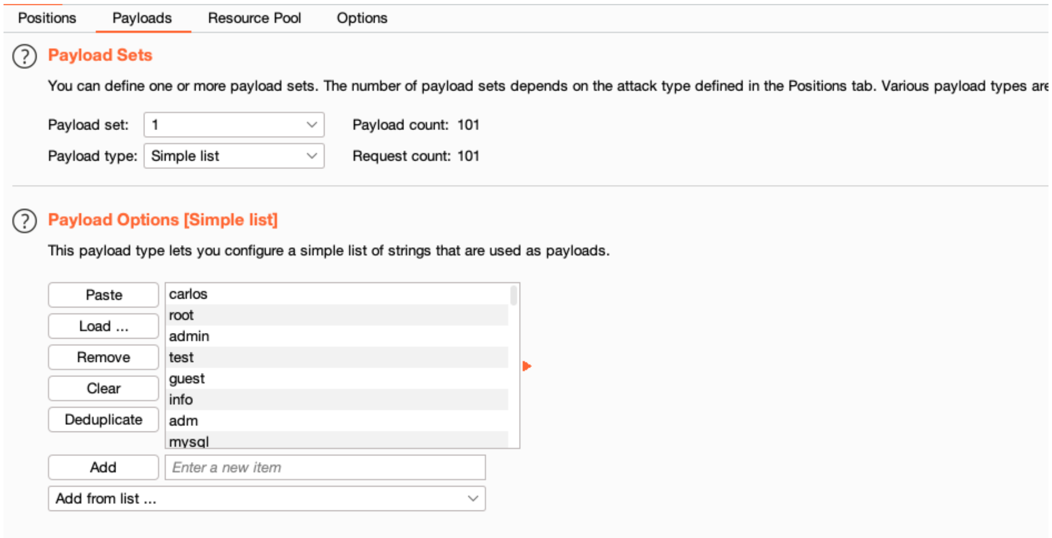
You now just need to configure the list of payloads that you want to use. For this demonstration, we'll try sending the request with different usernames to test how the login mechanism behaves.

Go to the **Payloads** sub-tab.

Leave the **Payload type** set to **Simple list**.

In the **Payload options** section, click **Paste** to add the copied usernames to the list.

In the **Payload sets** section, you can see how many payloads you have added, and how many requests this attack will send. For this attack, you should see **Payload count: 101 / Request count: 101**.



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

Payload set: 1
Payload type: Simple list
Payload count: 101
Request count: 101

Payload Options [Simple list]

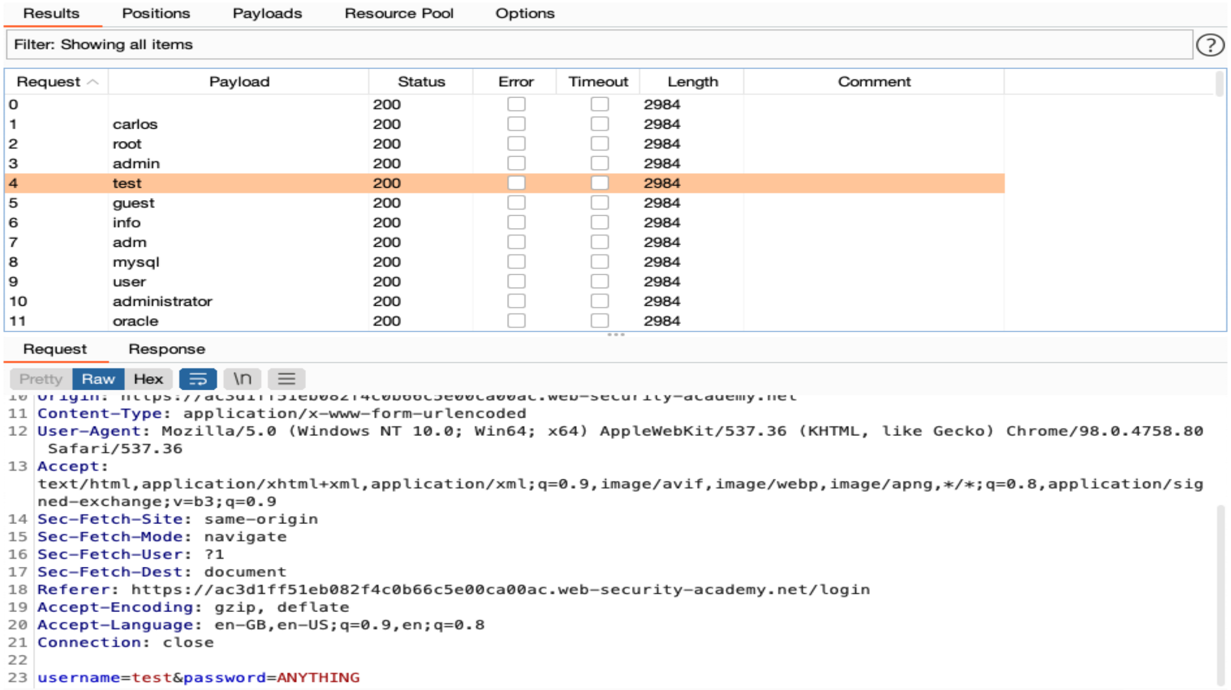
This payload type lets you configure a simple list of strings that are used as payloads.

```
carlos
root
admin
test
guest
info
adm
mysql
```

Step 6: Start the attack

In the upper-right corner, click **Start attack**. This opens a new attack window in which you can see each of the requests that Burp Intruder is making.

If you select one of the entries in the table, you can view the request and response in the message editor. Notice that the username parameter contains a different value from our payload list in each request.



Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	□	□	2984	
1	carlos	200	□	□	2984	
2	root	200	□	□	2984	
3	admin	200	□	□	2984	
4	test	200	□	□	2984	
5	guest	200	□	□	2984	
6	info	200	□	□	2984	
7	adm	200	□	□	2984	
8	mysql	200	□	□	2984	
9	user	200	□	□	2984	
10	administrator	200	□	□	2984	
11	oracle	200	□	□	2984	

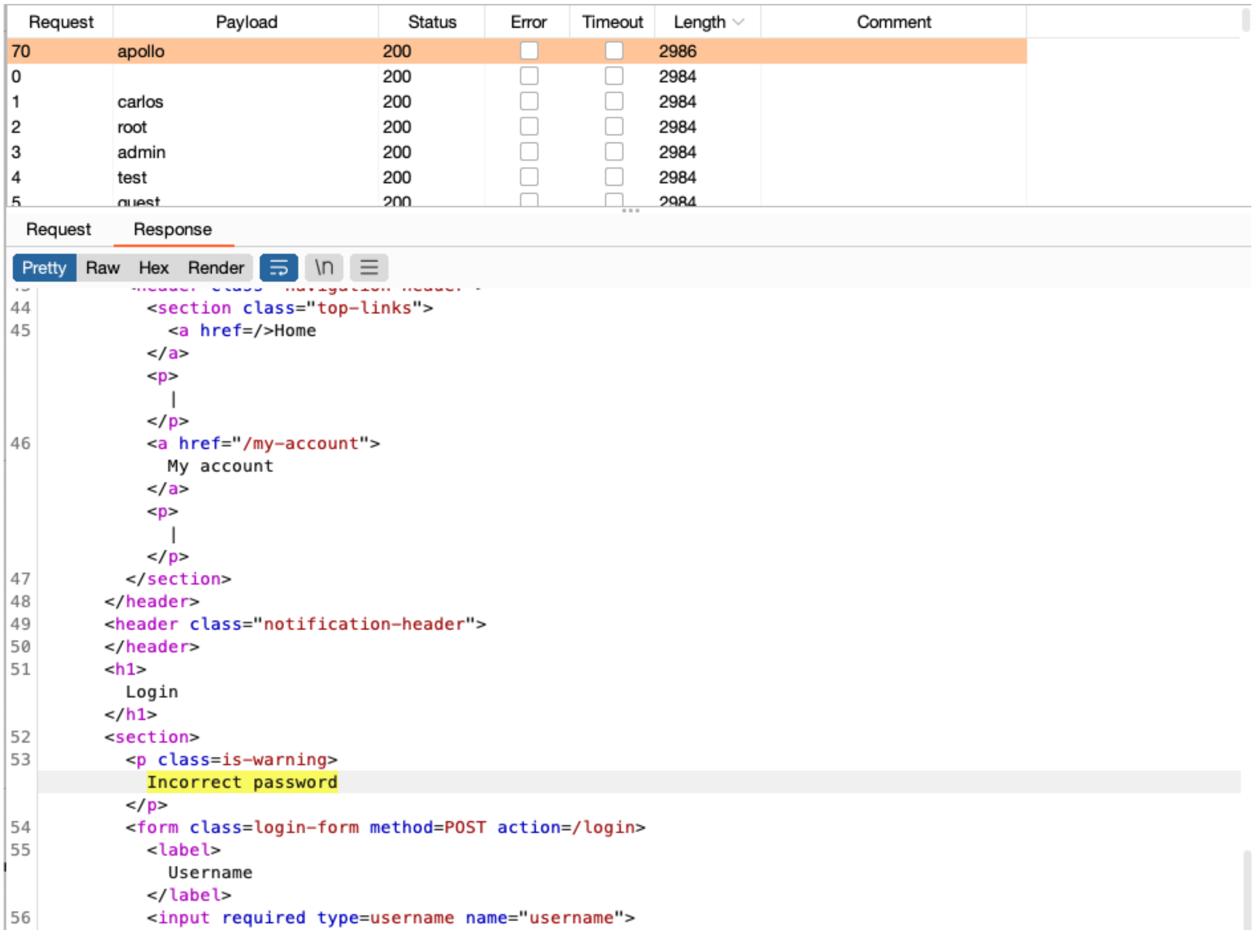
```
Origin: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.80
 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/sig
ned-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net/login
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close

username=test&password=ANYTHING
```

Step 8: Study the response

Select any request from the list to display it in the message editor.

Studying the responses, notice that most contain an Invalid username error message, but the one with the different length response has an Incorrect password error message. This different response strongly suggests that this username might be valid in this case.



Request	Payload	Status	Error	Timeout	Length	Comment
70	apollo	200	□	□	2986	
0		200	□	□	2984	
1	carlos	200	□	□	2984	
2	root	200	□	□	2984	
3	admin	200	□	□	2984	
4	test	200	□	□	2984	
5	guest	200	□	□	2984	

```
    <header class="navigation-header">
    <section class="top-links">
      <a href=/>Home
      </a>
      <p>
        |
      </p>
      <a href="/my-account">
        My account
      </a>
      <p>
        |
      </p>
    </section>
  </header>
  <header class="notification-header">
  </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">
```