

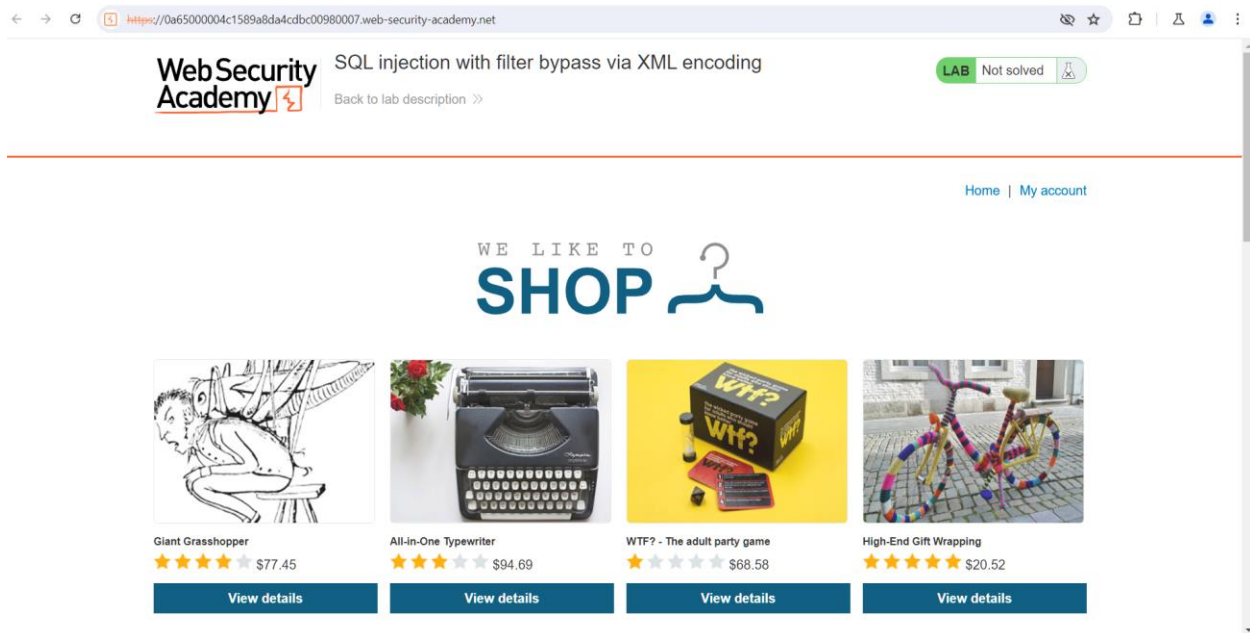
## Lab: SQL injection with filter bypass via XML encoding

### Scenario:

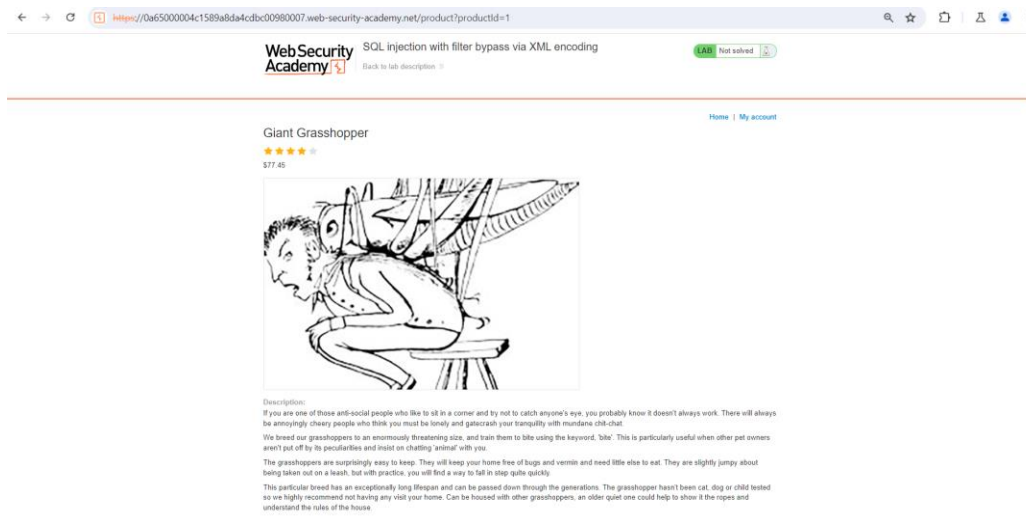
The stock check feature in this lab is vulnerable to SQL injection, allowing for a UNION attack to retrieve data from other tables. To solve the lab, exploit this vulnerability to obtain the admin user's credentials from the users table and log into their account.

### Before Testing the Exploit:

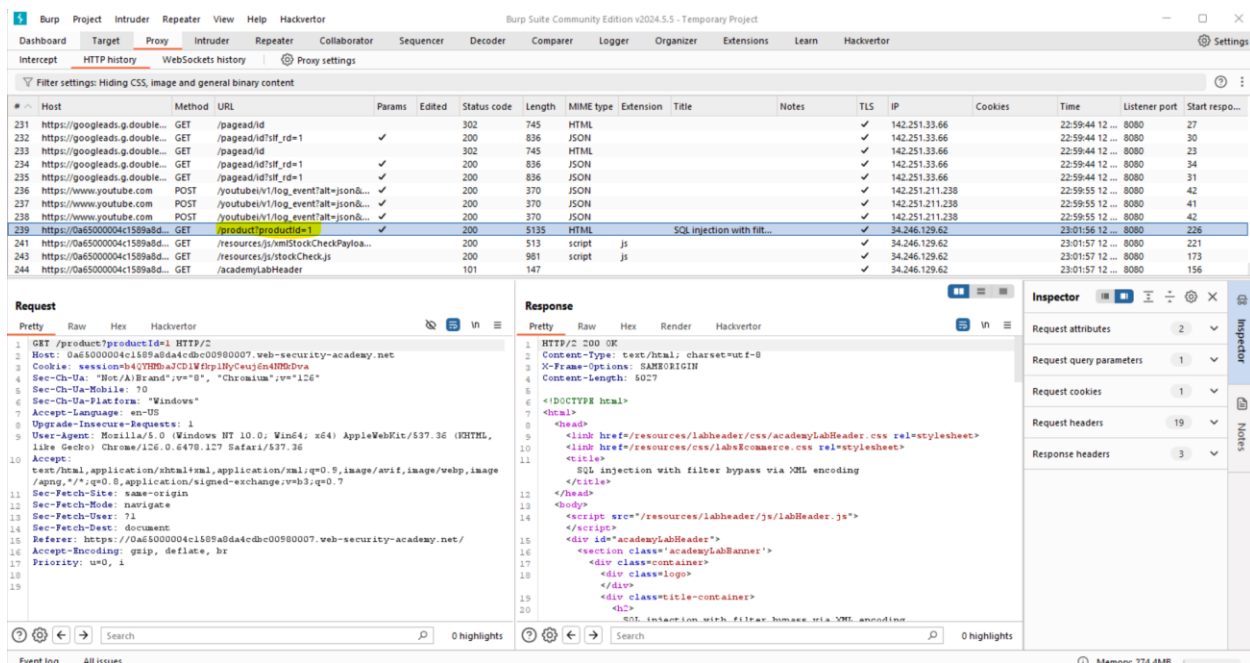
### Web Application Page:



Clicked on the product from the left



Under BurpSuite, the website requests will be passed to the proxy when I click on the product with the ID "1".



To modify the web request, send the selected request to the Repeater.

The screenshot shows the Burp Suite interface with a list of HTTP requests. A context menu is open for a selected request, showing options like 'Send to Repeater', 'Send to Sequencer', and 'Send to Organizer'. The 'Send to Repeater' option is highlighted.

| #   | Host                            | Method | URL                                 | Params                          | Edited | Status code | Length | MIME type | Extension | Title                     | Notes | TLS | IP              | Cookies | Time           | Listener port | Start response... |
|-----|---------------------------------|--------|-------------------------------------|---------------------------------|--------|-------------|--------|-----------|-----------|---------------------------|-------|-----|-----------------|---------|----------------|---------------|-------------------|
| 231 | https://googleads.g.double...   | GET    | /pagead/ld                          |                                 |        | 302         | 745    | HTML      |           |                           |       | ✓   | 142.251.33.66   |         | 22:59:44.12... | 8080          | 27                |
| 232 | https://googleads.g.double...   | GET    | /pagead/ld?if_rd=1                  |                                 | ✓      | 200         | 836    | JSON      |           |                           |       | ✓   | 142.251.33.66   |         | 22:59:44.12... | 8080          | 30                |
| 233 | https://googleads.g.double...   | GET    | /pagead/ld                          |                                 |        | 302         | 745    | HTML      |           |                           |       | ✓   | 142.251.33.66   |         | 22:59:44.12... | 8080          | 23                |
| 234 | https://googleads.g.double...   | GET    | /pagead/ld?if_rd=1                  |                                 |        | 200         | 836    | JSON      |           |                           |       | ✓   | 142.251.33.66   |         | 22:59:44.12... | 8080          | 34                |
| 235 | https://googleads.g.double...   | GET    | /pagead/ld?if_rd=1                  |                                 | ✓      | 200         | 836    | JSON      |           |                           |       | ✓   | 142.251.33.66   |         | 22:59:44.12... | 8080          | 31                |
| 236 | https://www.youtube.com         | POST   | /youtubei/v1/log_event?alt=json&... |                                 |        | 200         | 370    | JSON      |           |                           |       | ✓   | 142.251.211.238 |         | 22:59:55.12... | 8080          | 42                |
| 237 | https://www.youtube.com         | POST   | /youtubei/v1/log_event?alt=json&... |                                 |        | 200         | 370    | JSON      |           |                           |       | ✓   | 142.251.211.238 |         | 22:59:55.12... | 8080          | 41                |
| 238 | https://www.youtube.com         | POST   | /youtubei/v1/log_event?alt=json&... |                                 | ✓      | 200         | 370    | JSON      |           |                           |       | ✓   | 142.251.211.238 |         | 22:59:55.12... | 8080          | 42                |
| 239 | https://0a65000004c1589a8da4... | GET    | /product?productid=1                |                                 | ✓      | 300         | 5134   | HTML      |           | SQL injection with fil... |       | ✓   | 34.246.129.62   |         | 23:01:56.12... | 8080          | 226               |
| 241 | https://0a65000004c1589a8da4... | GET    | /resources/                         | https://0a65000004c1589a8da4... |        |             |        | js        |           |                           |       | ✓   | 34.246.129.62   |         | 23:01:57.12... | 8080          | 221               |
| 243 | https://0a65000004c1589a8da4... | GET    | /resources/                         |                                 |        |             |        | js        |           |                           |       | ✓   | 34.246.129.62   |         | 23:01:57.12... | 8080          | 173               |
| 244 | https://0a65000004c1589a8da4... | GET    | /academy/                           |                                 |        |             |        | js        |           |                           |       | ✓   | 34.246.129.62   |         | 23:01:57.12... | 8080          | 156               |

After that, click on "Check Stock" on the webpage.

The screenshot shows a web browser displaying a page with a drawing of a grasshopper. Below the drawing, there is a description of grasshoppers and a 'Check stock' button. The button is highlighted with a yellow box.

URL: <https://0a65000004c1589a8da4cdbc00980007.web-security-academy.net/product?productid=1>

Description:

If you are one of those anti-social people who like to sit in a corner and try not to catch anyone's eye, you probably know it doesn't always work. There will always be annoyingly cheery people who think you must be lonely and gatecrash your tranquility with mundane chit-chat.

We breed our grasshoppers to an enormously threatening size, and train them to bite using the keyword, 'bite'. This is particularly useful when other pet owners aren't put off by its peculiarities and insist on chatting 'animal' with you.

The grasshoppers are surprisingly easy to keep. They will keep your home free of bugs and vermin and need little else to eat. They are slightly jumpy about being taken out on a leash, but with practice, you will find a way to fall in step quite quickly.

This particular breed has an exceptionally long lifespan and can be passed down through the generations. The grasshopper hasn't been cat, dog or child tested so we highly recommend not having any visit your home. Can be housed with other grasshoppers, an older quiet one could help to show it the ropes and understand the rules of the house.

London

619 units

[Return to list](#)

Now, under BurpSuite, send the "Check Stock" web POST request to the Repeater to modify it.

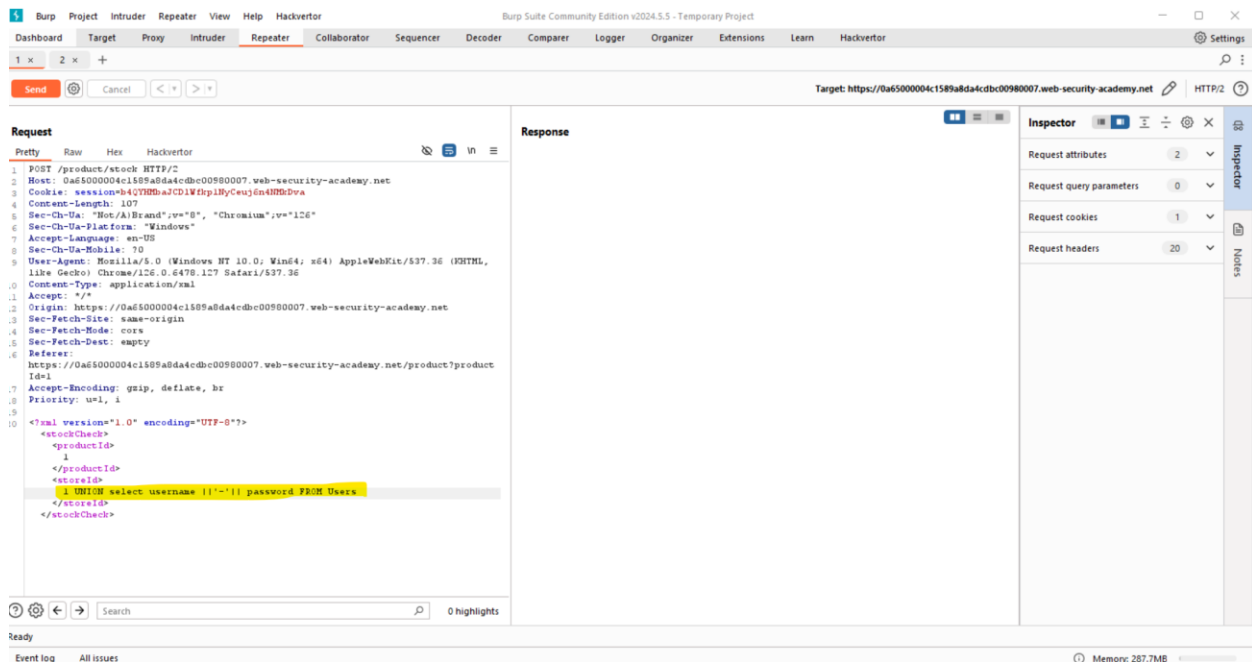
The screenshot displays the Burp Suite interface. The top menu bar includes Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, Learn, and Hackvector. The main window is divided into three panes: a list of HTTP history on the left, a detailed view of the selected request in the center, and an Inspector pane on the right.

The HTTP history list shows several requests to `https://googleads.g.doubleclick.net` and `https://www.youtube.com`. The selected request is a POST to `https://0a65000004c1589a8d4...-ity-academy.net/product/stock`. The request details pane shows the raw HTTP request, including headers and body. The body contains an XML payload with a `<stockCheck>` element.

The Inspector pane on the right shows request attributes, cookies, headers, and response headers. The memory usage at the bottom right is 274.4MB.

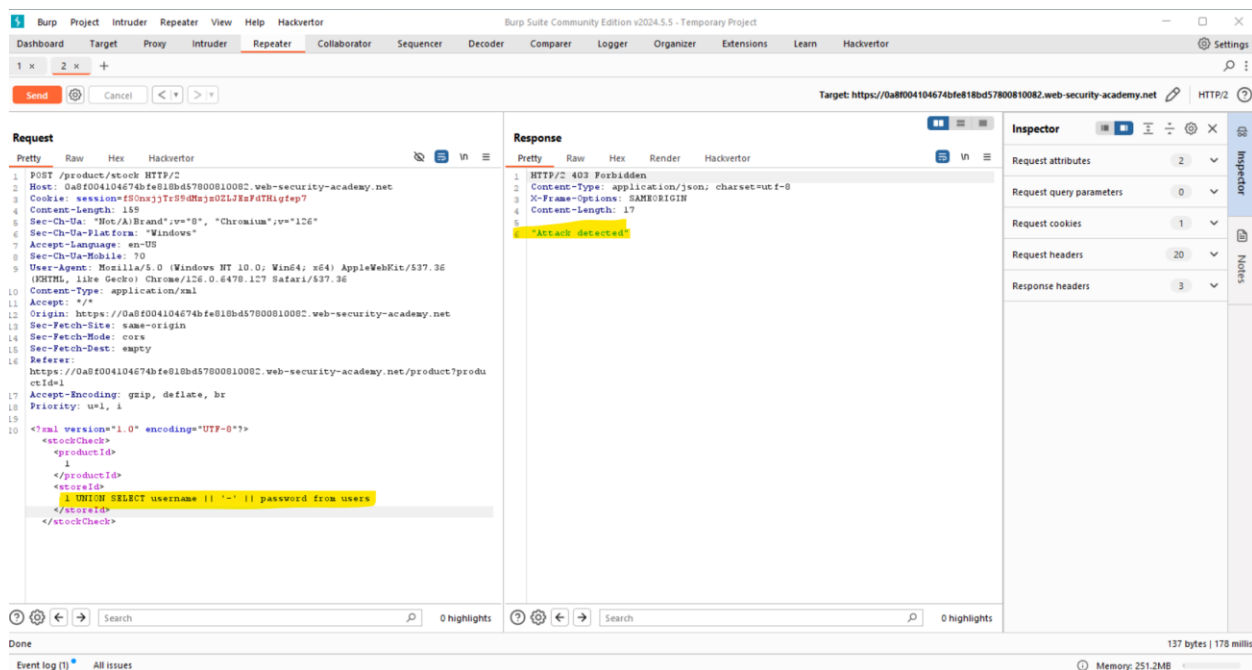
There is a vulnerability in the "Check Stock" feature, so I modified the POST request with an SQL query to fetch user credentials:

SQL query: `SELECT 'some_column' UNION SELECT username || '~' || password FROM users;`



After encoding the query payload using the Hackvector extension, we can fetch the user credentials of the application, as a basic SQL query payload would be blocked by the Web Application Firewall.

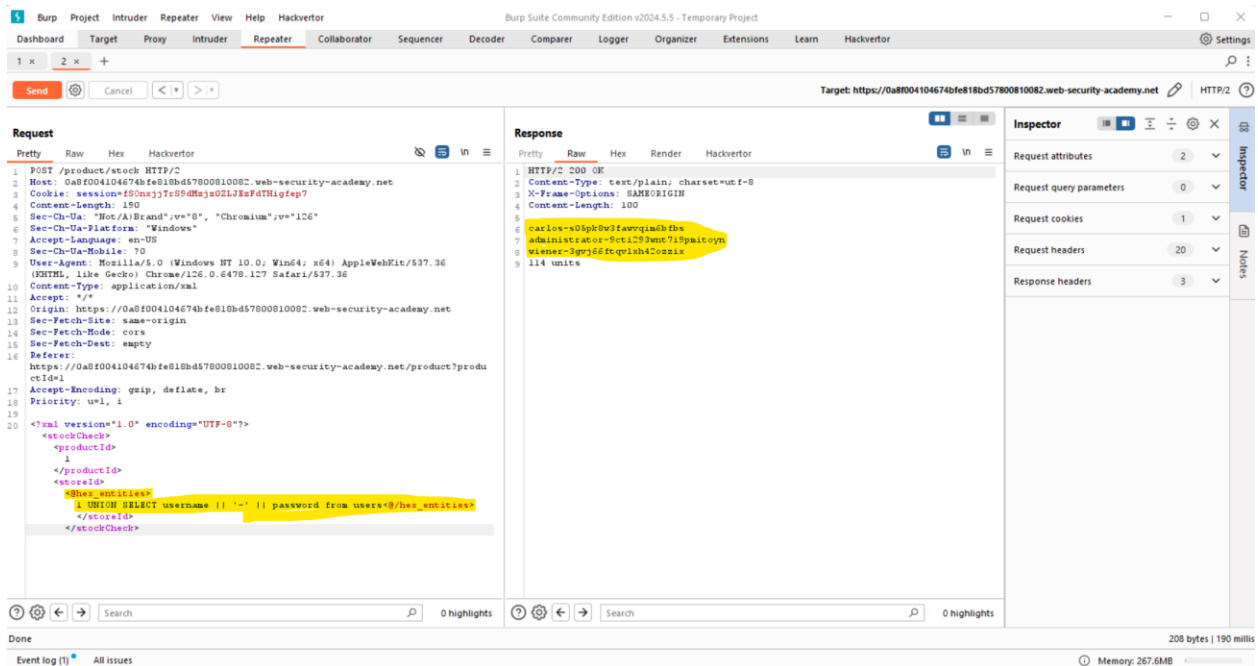
## Basic SQL query payload:



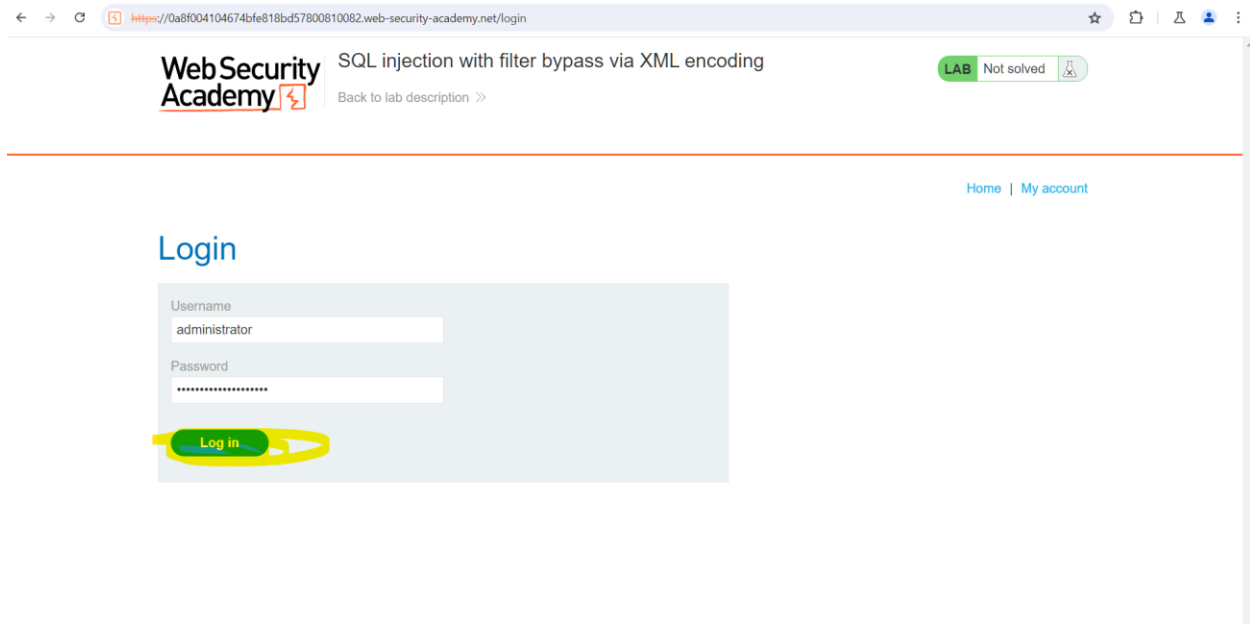
## SQL payload after encoding

The screenshot displays the Burp Suite interface with the 'Repeater' tab selected. The 'Request' pane on the left shows an HTTP POST request to a target URL. The request body contains an SQL payload: `1 UNION SELECT username || '-' ||`. The 'Extensions' menu is open, and the 'Encode' option is highlighted. The 'Encode' submenu is also visible, showing various encoding options like 'base32', 'base58', 'base64', 'base64url', 'burp\_urlencode', 'css\_escapes', 'css\_escapes6', 'dec\_entities', 'hex', 'hex\_entities', 'hex\_escapes', 'html5\_entities', 'html\_entities', 'js\_string', 'jwt', 'octal\_escapes', 'php\_chr', 'php\_non\_alpha', 'powershell', 'quoted\_printable', 'sami', 'sql\_hex', 'unicode\_alternatives', 'unicode\_escapes', 'urlencode', 'urlencode\_all', and 'urlencode\_not\_gpus'. The 'Inspector' pane on the right shows the request details, including the target URL and the request body.

After encoding the payload , you can see the user credentials to access the web application



Finally performed SQL injection by logging with admin credentials  
(username :administrator, password: 9ct1293wnt7i9pmitoyn)





Congratulations, you solved the lab!

Share your skills! [🐦](#) [in](#) [Continue learning](#) >>

[Home](#) | [My account](#) | [Log out](#)

## My Account

Your username is: administrator

Email

[Update email](#)