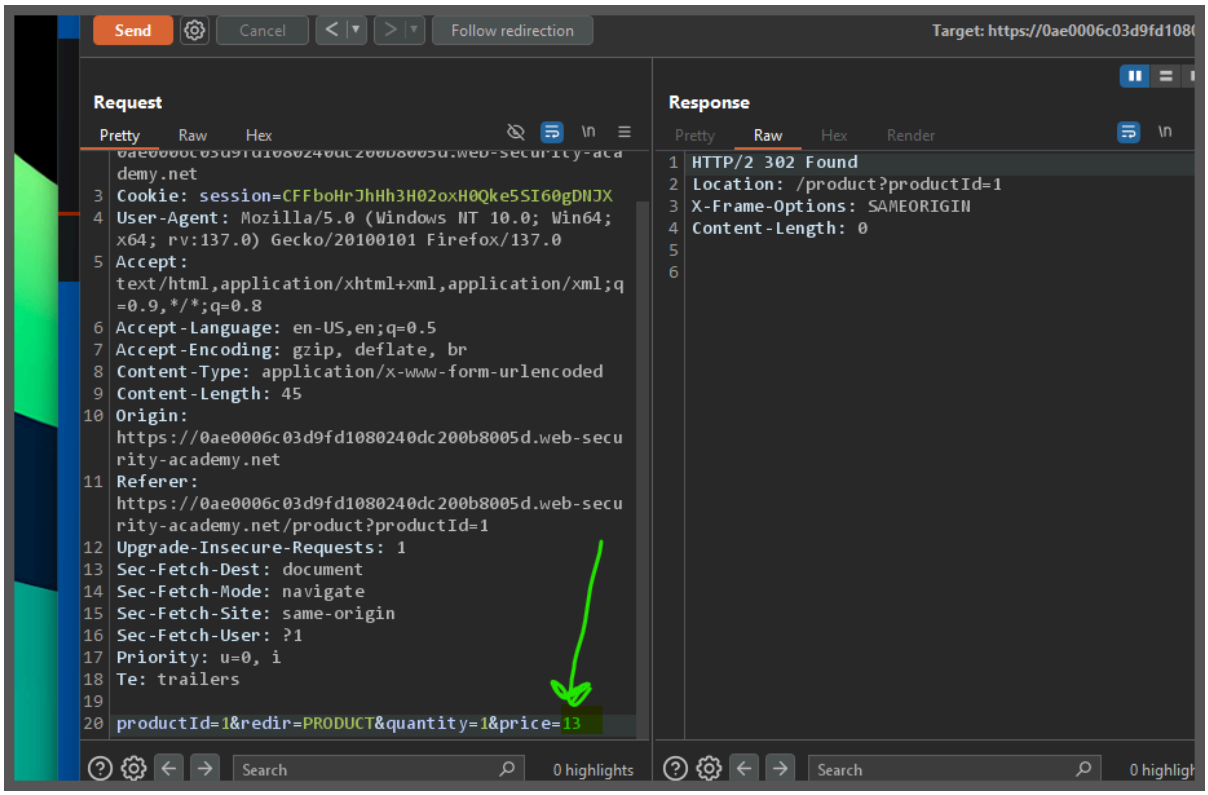
Business Logic flaw

Lab: Excessive trust in client-side controls

This lab doesn't adequately validate user input. You can exploit a logic flaw in its purchasing workflow to buy items for an unintended price. To solve the lab, buy a "Lightweight l33t leather jacket".

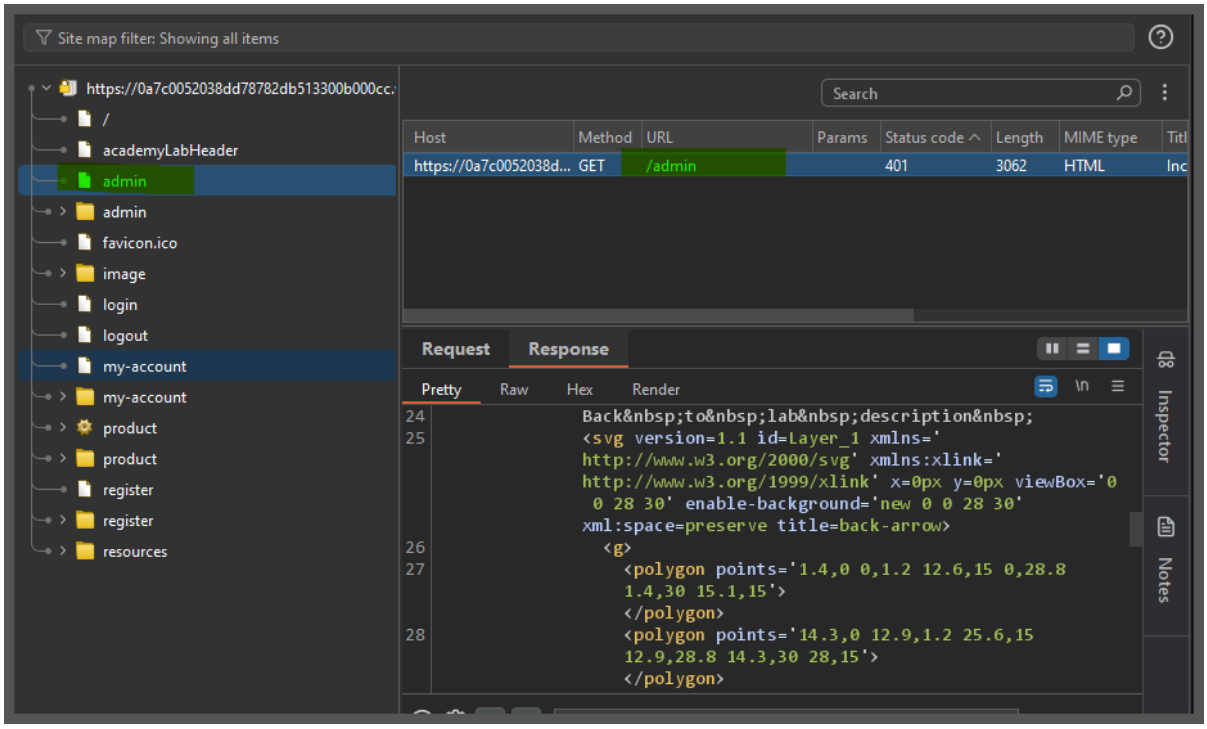
You can log in to your own account using the following credentials: `wiener:peter`

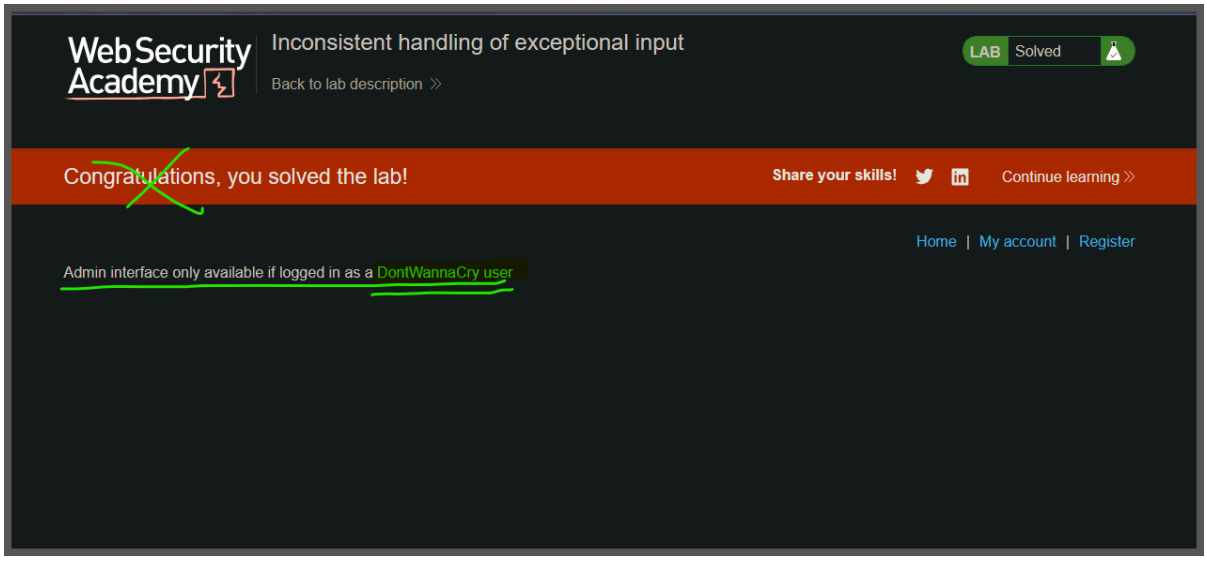


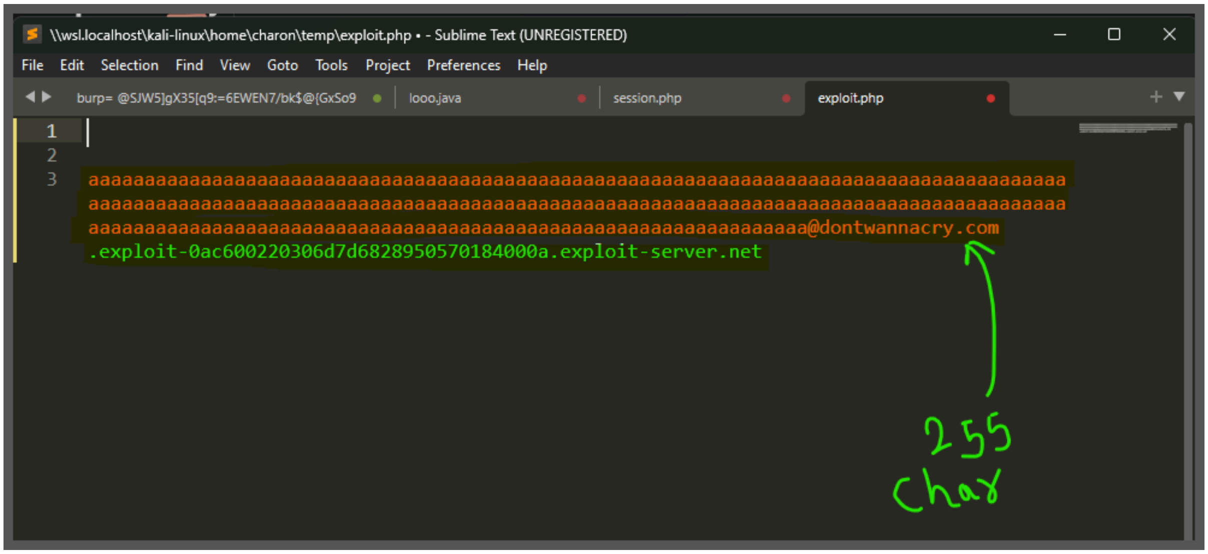
Lab: Inconsistent handling of exceptional input

This lab doesn't adequately validate user input. You can exploit a logic flaw in its account registration process to gain access to administrative functionality. To solve the lab, access the admin panel and delete the user `carlos`.

do directory busting

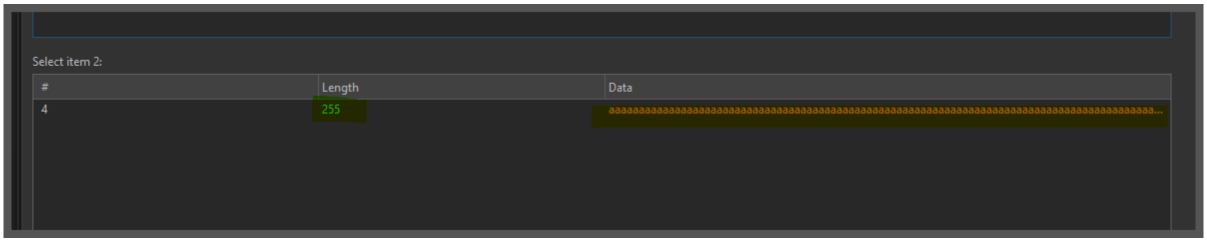






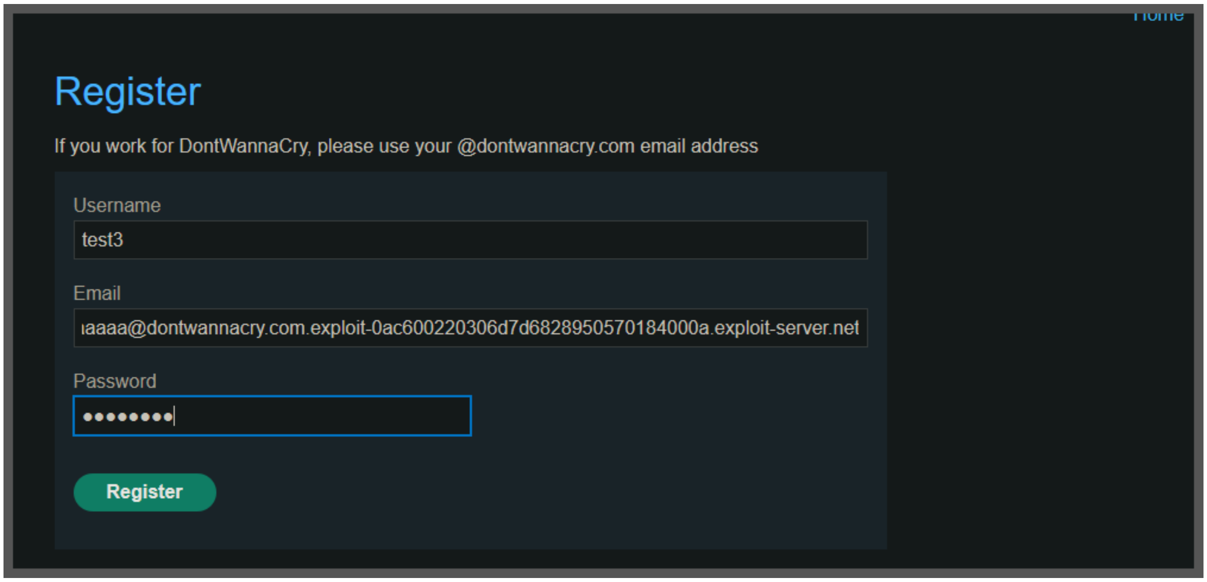
```


aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa@dontwannacry.com
.exploit-0ac600220306d7d6828950570184000a.exploit-server.net
```



Select item 2:

#	Length	Data
4	255	aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa...



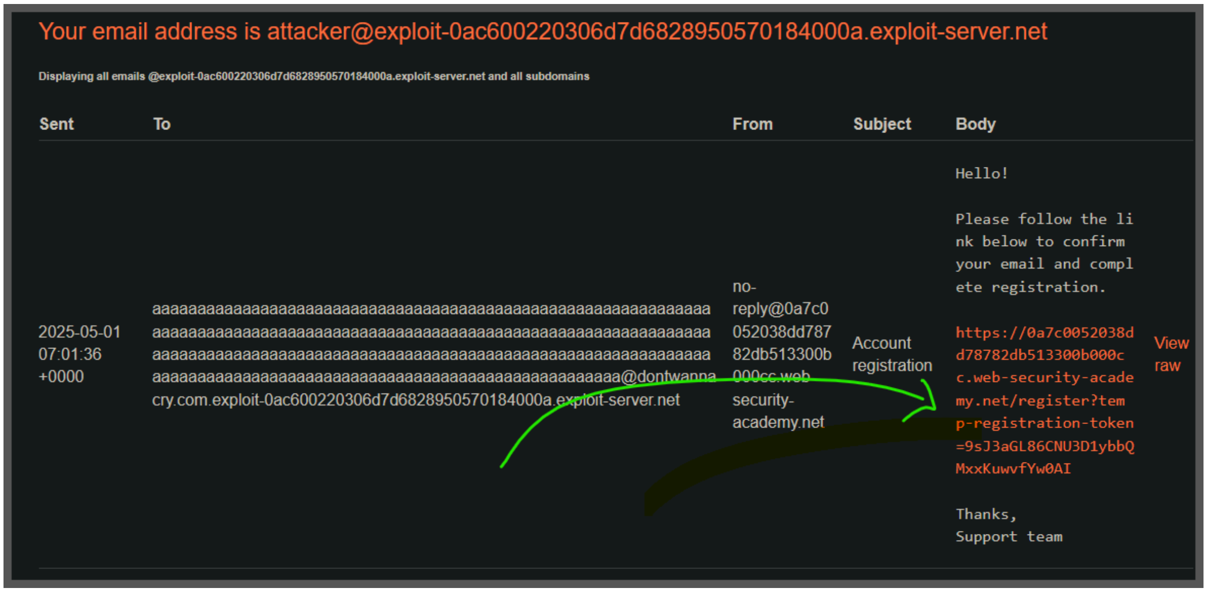
Register

If you work for DontWannaCry, please use your @dontwannacry.com email address

Username
test3

Email
ıaaaa@dontwannacry.com.exploit-0ac600220306d7d6828950570184000a.exploit-server.net

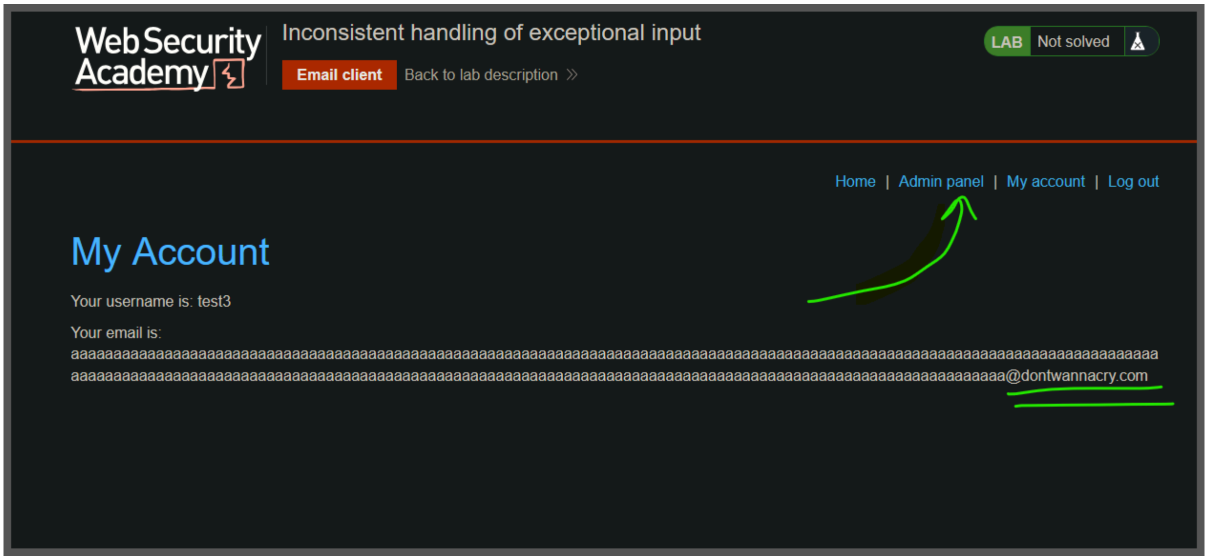
Password
••••••••



Your email address is attacker@exploit-0ac600220306d7d6828950570184000a.exploit-server.net

Displaying all emails @exploit-0ac600220306d7d6828950570184000a.exploit-server.net and all subdomains

Sent	To	From	Subject	Body	
2025-05-01 07:01:36 +0000	aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa@dontwannacry.com.exploit-0ac600220306d7d6828950570184000a.exploit-server.net	no-reply@0a7c0052038dd78782db513300b000cc.web-security-academy.net	Account registration	Hello! Please follow the link below to confirm your email and complete registration. https://0a7c0052038dd78782db513300b000c c.web-security-academy.net/register?temp-registration-token=9sJ3aGL86CNU3D1ybbQMxxKuwvfYw0AI Thanks, Support team	View raw



My Account

Your username is: test3

Your email is: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa@dontwannacry.com

Lab: Inconsistent security controls

This lab's flawed logic allows arbitrary users to access administrative functionality that should only be available to company employees. To solve the lab, access the admin panel and delete the user `carlos`.

Simple:

go normally create an account

get the email on your email clint

login

now you see that update email

do : test@dontwannacry.com

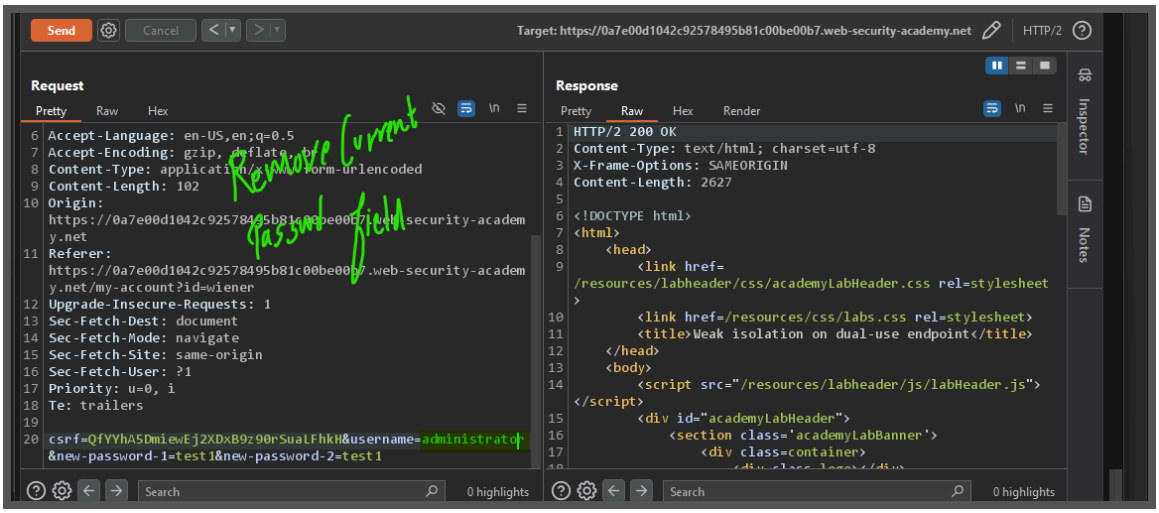
now your admin

delete the carlos

Lab: Weak isolation on dual-use endpoint

This lab makes a flawed assumption about the user's privilege level based on their input. As a result, you can exploit the logic of its account management features to gain access to arbitrary users' accounts. To solve the lab, access the `administrator` account and delete the user `carlos`.

You can log in to your own account using the following credentials: `wiener:peter`

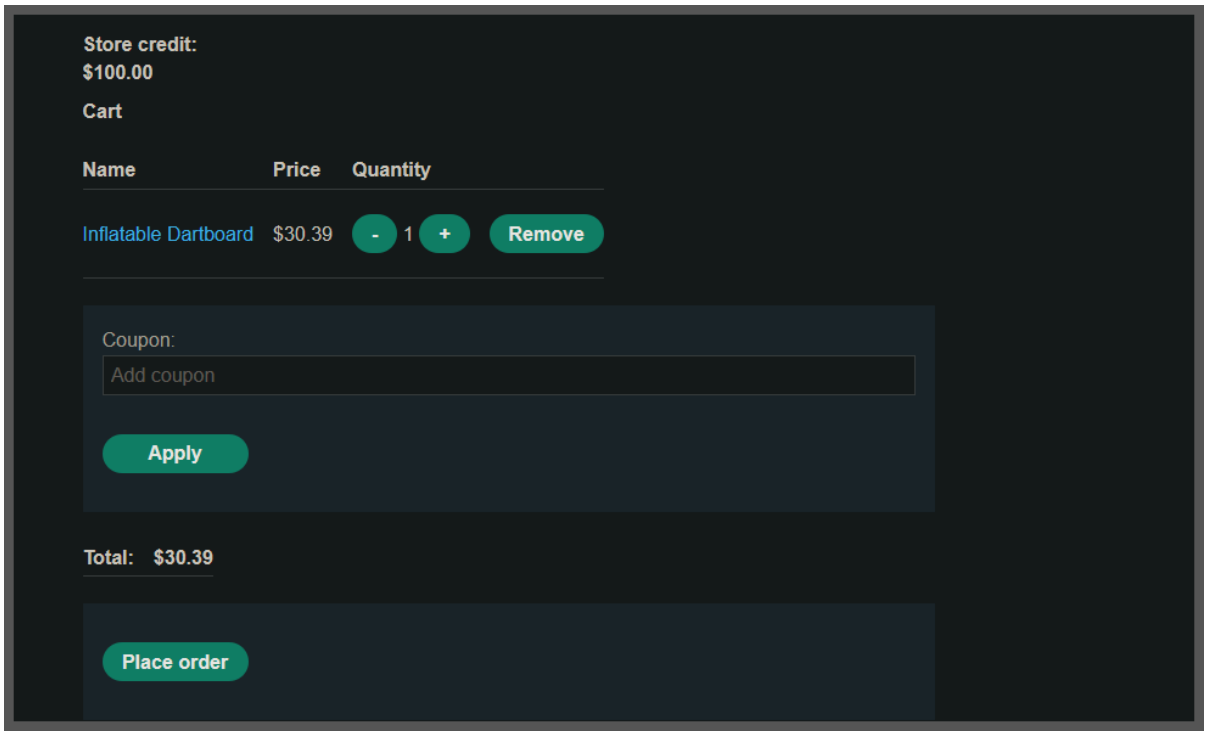


Lab: Insufficient workflow validation

This lab makes flawed assumptions about the sequence of events in the purchasing workflow. To solve the lab, exploit this flaw to buy a "Lightweight l33t leather jacket".

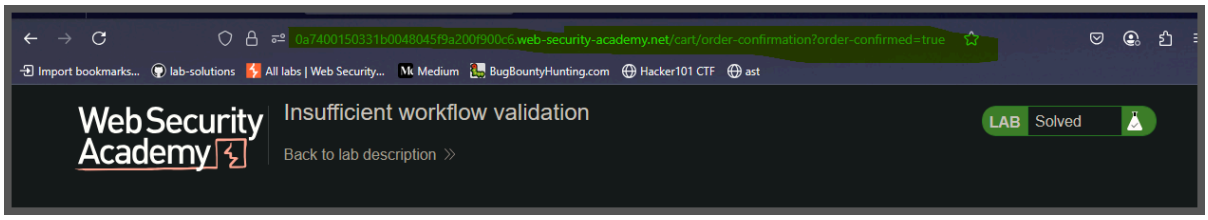
You can log in to your own account using the following credentials: `wiener:peter`

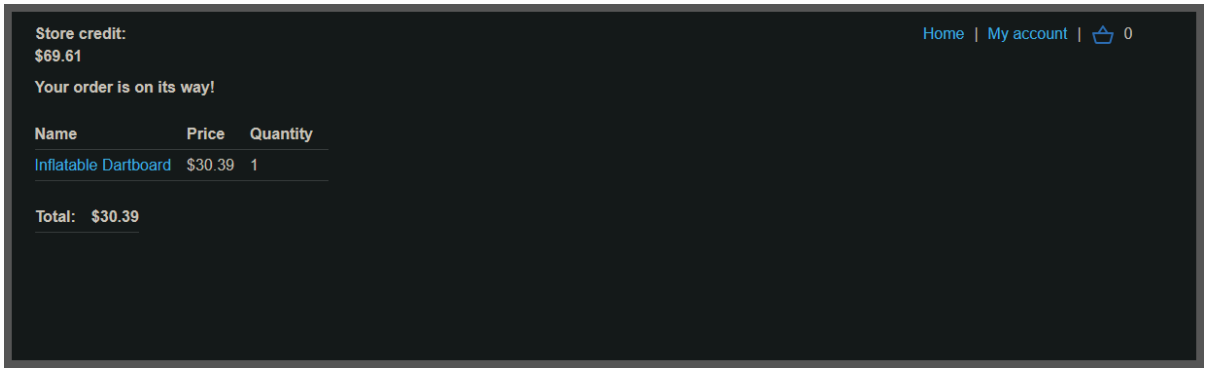
ACCESS THE LAB



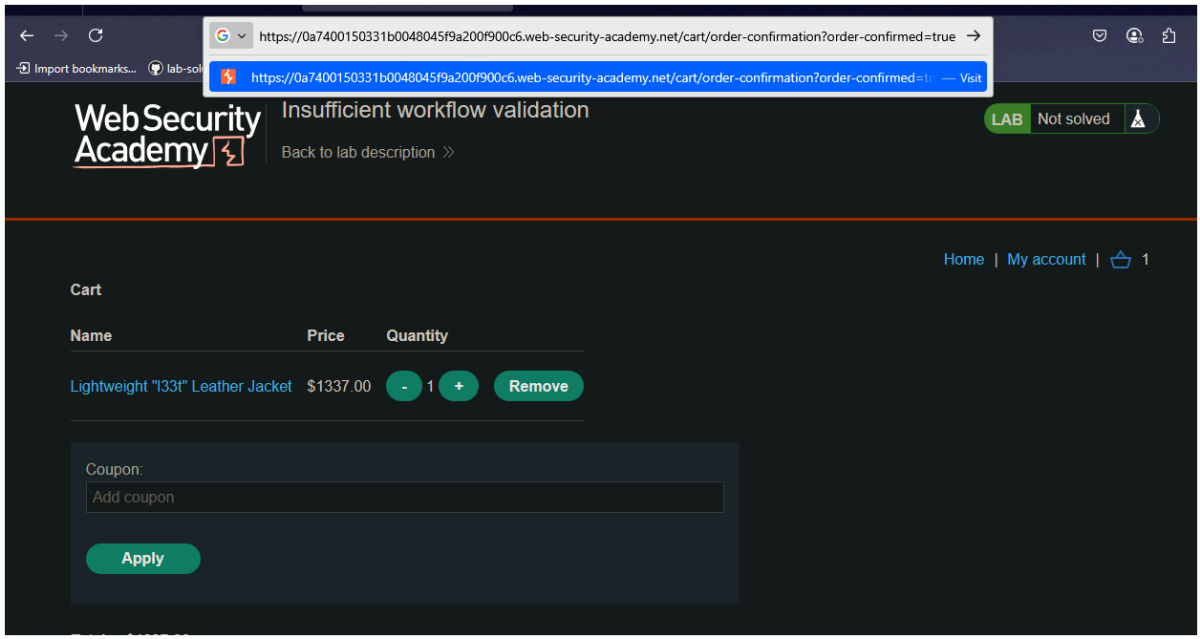
place the order

once we place the order is shifted ot this url





so now ad the leader jacket and put this url

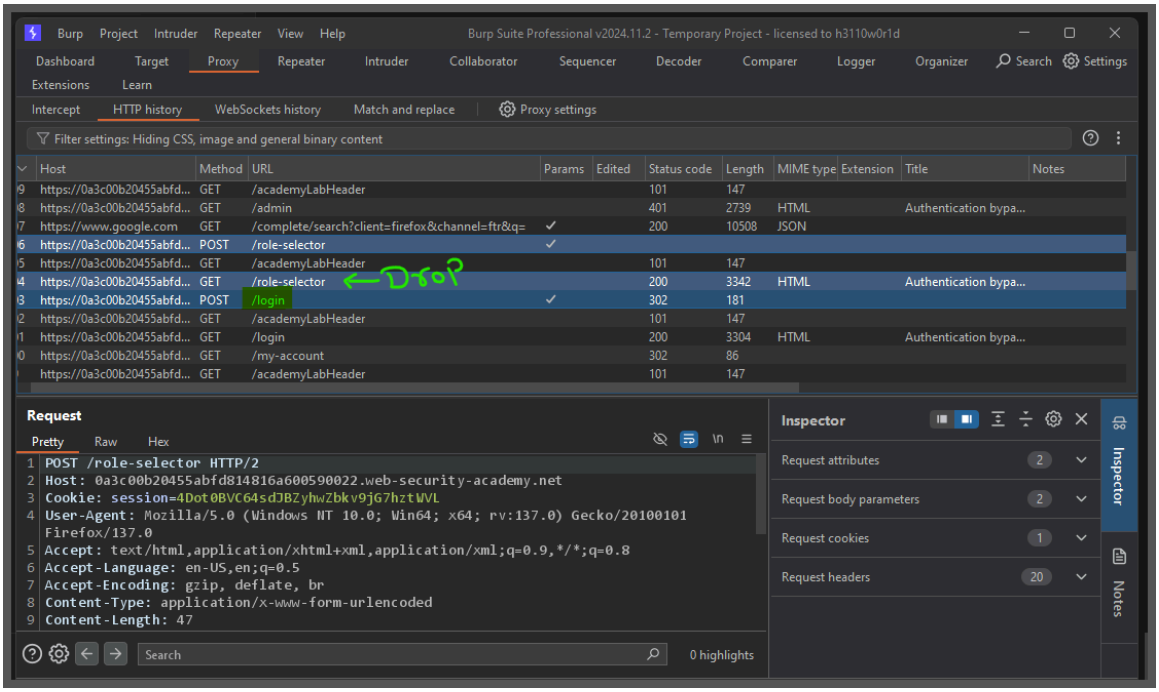


solved the lab

Lab: Authentication bypass via flawed state machine

This lab makes flawed assumptions about the sequence of events in the login process. To solve the lab, exploit this flaw to bypass the lab's authentication, access the admin interface, and delete the user `carlos`.

You can log in to your own account using the following credentials: `wiener:peter`



Turn on intercept

make login

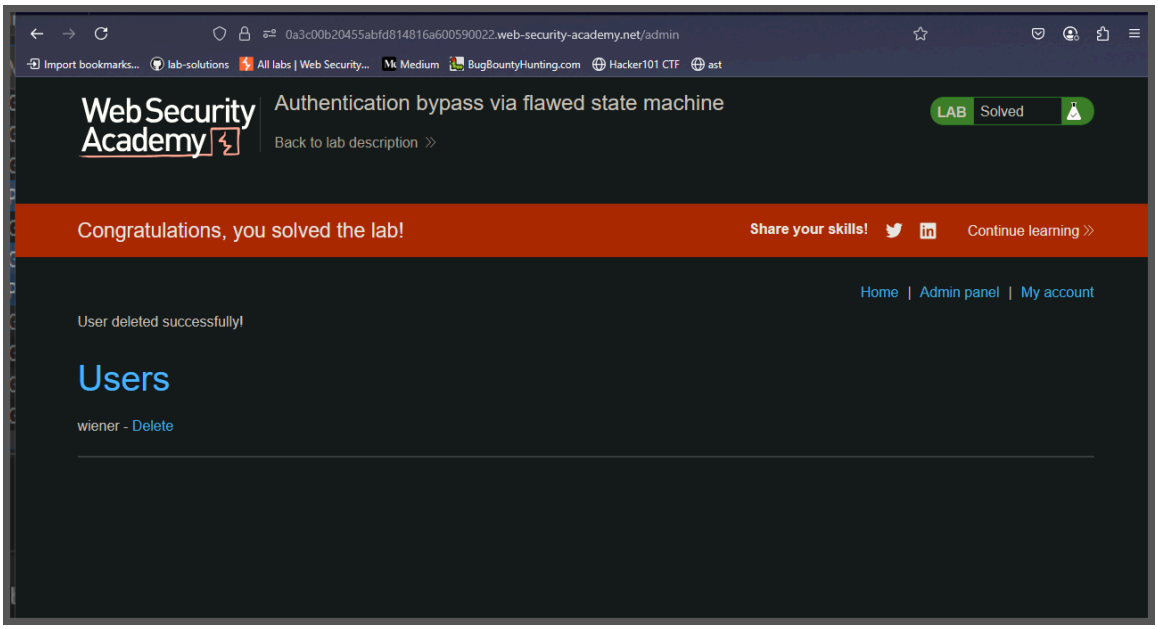
now you see next request is to role selector

drop that request

turn off intercept

and remove the roleselector from url

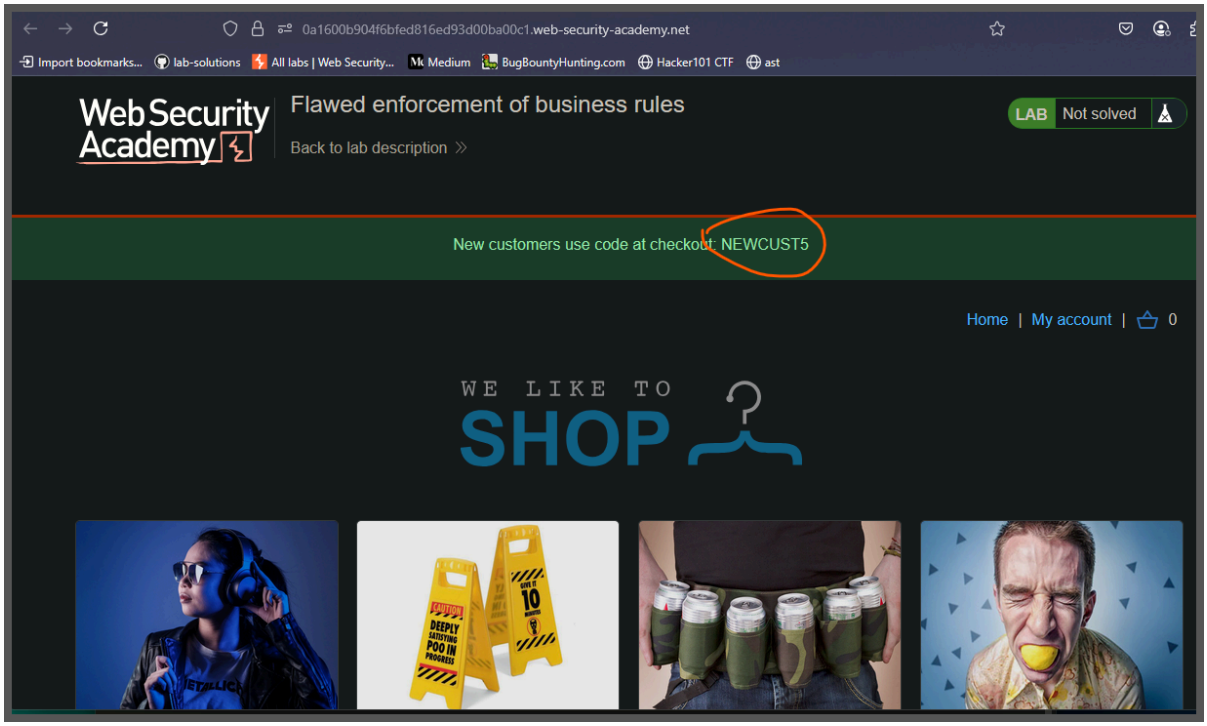
now go your admin



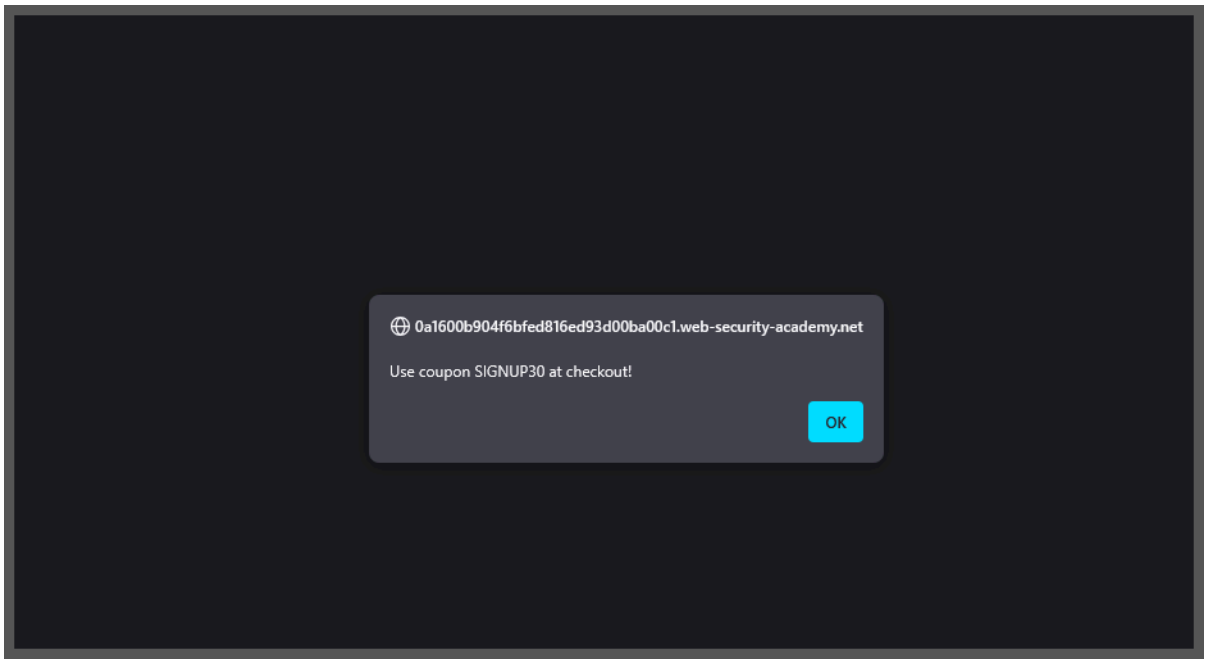
Lab: Flawed enforcement of business rules

This lab has a logic flaw in its purchasing workflow. To solve the lab, exploit this flaw to buy a "Lightweight l33t leather jacket".

You can log in to your own account using the following credentials: `wiener:peter`



scroll down you will see sighup of newlatter make it



till now we got two cupons

Lab: Infinite money logic flaw

This lab has a logic flaw in its purchasing workflow. To solve the lab, exploit this flaw to buy a "Lightweight l33t leather jacket".

You can log in to your own account using the following credentials: `wiener:peter`

lab is easy but there are few more steps

ther is and logic vulnerability

using cupon get discount

buy the product and hence due to discont you get few money back

repeating this you will create infinate money

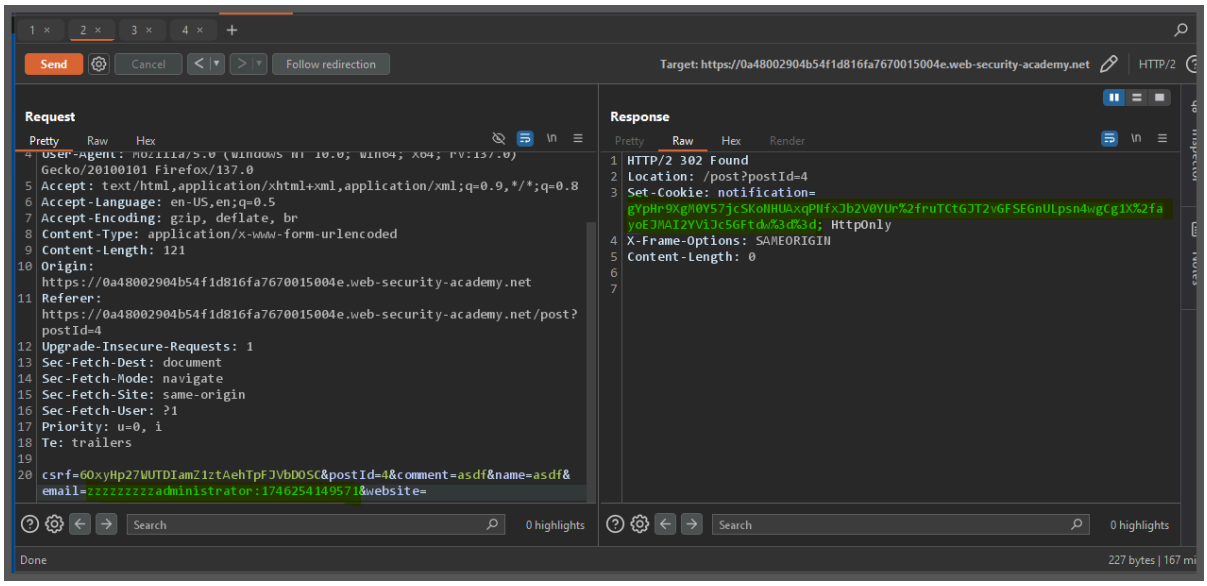
once you got more money

buy leather jacket

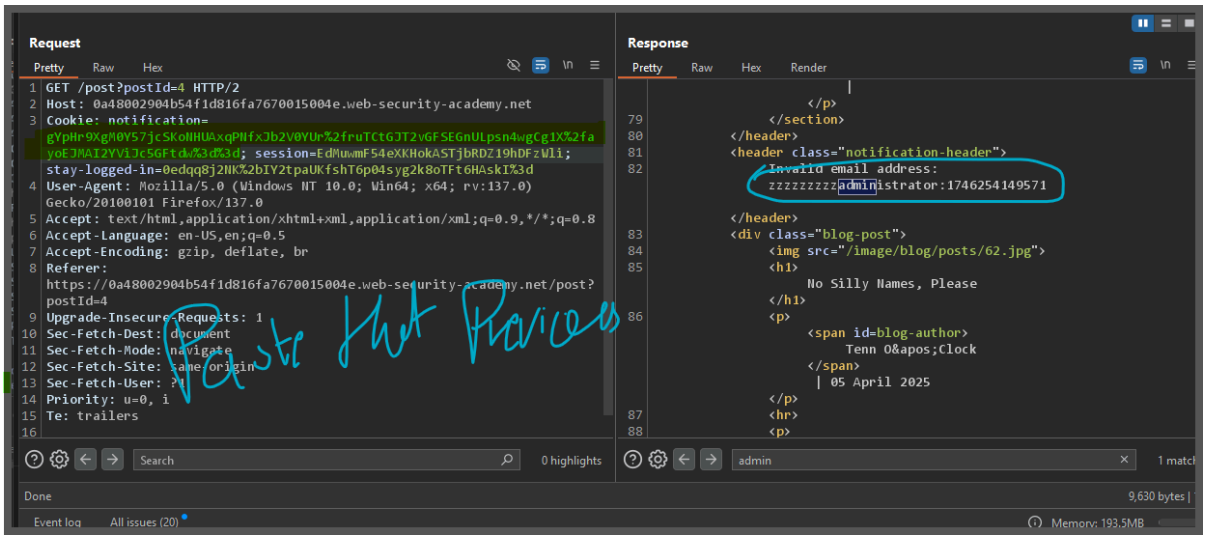
Lab: Authentication bypass via encryption oracle

This lab contains a logic flaw that exposes an encryption oracle to users. To solve the lab, exploit this flaw to gain access to the admin panel and delete the user `carlos`.

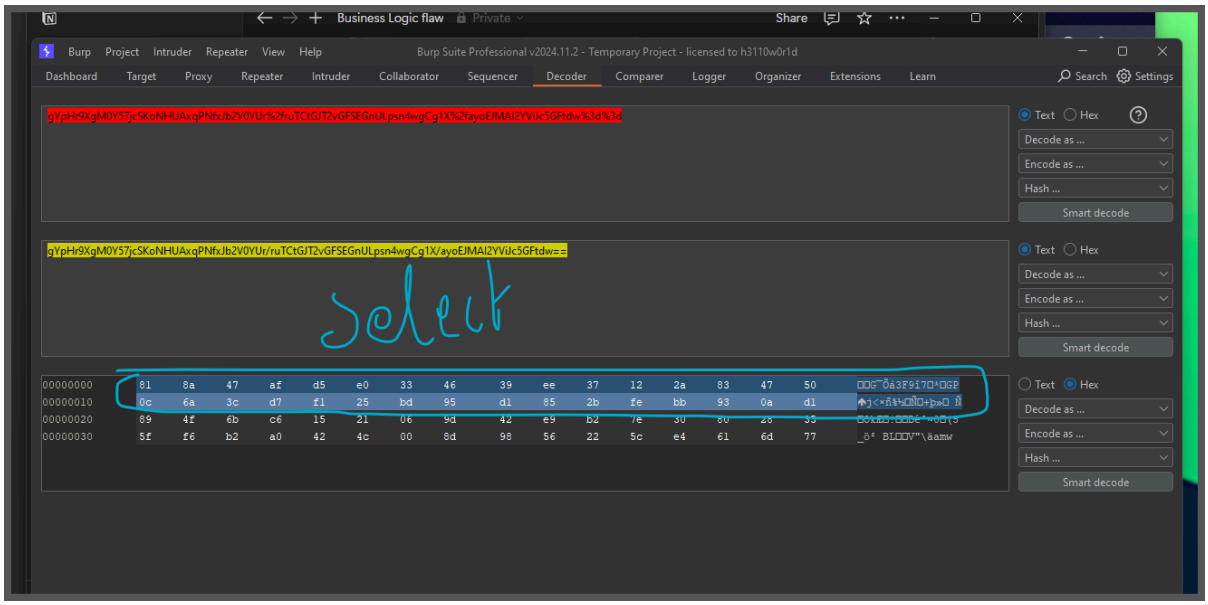
You can log in to your own account using the following credentials: `wiener:peter`

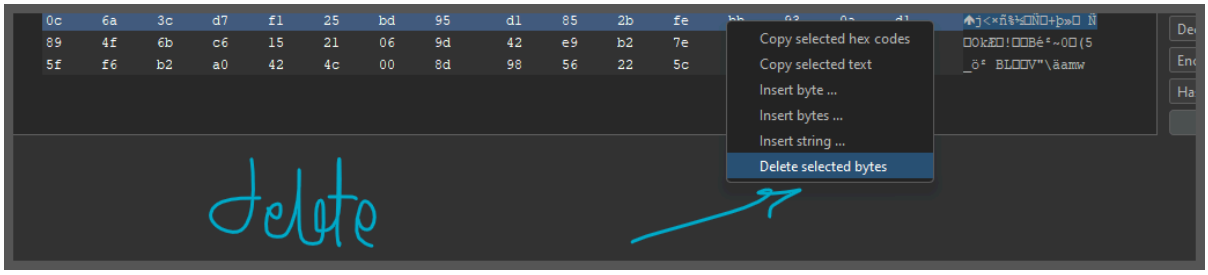


zzzzzzzzzadministrator:1746254149571



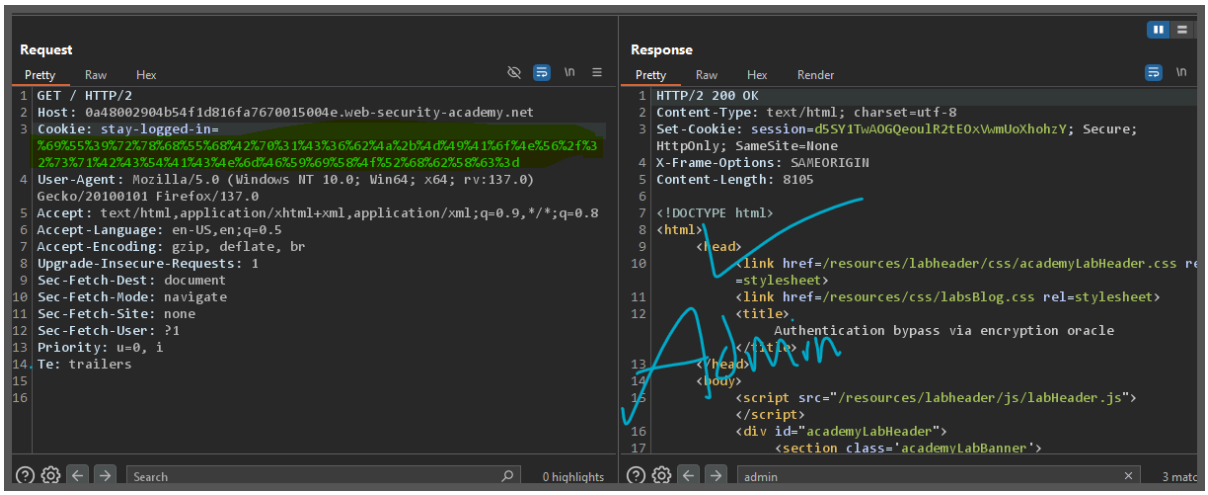
in first screenshot see in response you have got some notification value paste it in second screenshot request





now copy that url encoded and paste it in the

and remove other session cookie only keep stay logged in cookie



Lab: Bypassing access controls using email address parsing discrepancies

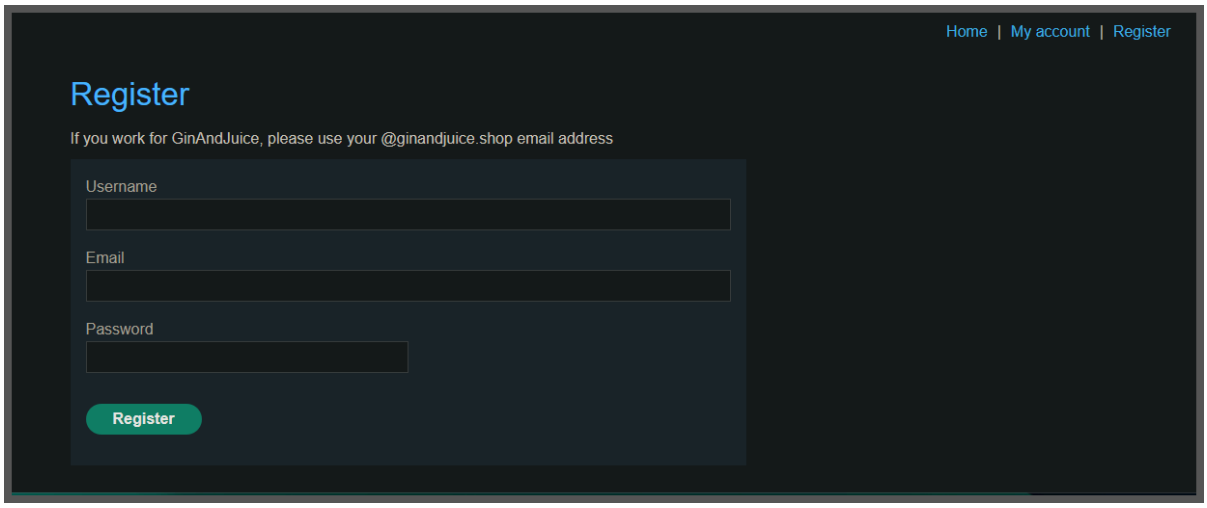
This lab validates email addresses to prevent attackers from registering addresses from unauthorized domains. There is a parser discrepancy in the validation logic and library used to parse email addresses.

To solve the lab, exploit this flaw to register an account and delete `carlos`.

solving part is easy

but concept is little bit hard to accept

read this ;https://portswigger.net/research/splitting-the-email-atom



```
=?utf-7?q?attacker&AEA-exploit-0a8b00fa04d93dfc80d7612c011e004d.exploit-server.net&ACA-?=@ginandjuice.shop
```

put this in email

and fill other input username and password

you will get and email infact your email will be consider as ownerdomain email but inside that is another

🔍 Step 1: Identify the format

This string starts with:

```
ruby
CopyEdit
=?utf-7?q?
```

That means it's using **MIME encoding**:

- `=?...?...?...?=` is the format used in **email headers** (like in "From", "To", or "Subject") to include special characters.
- `utf-7` means the **character encoding** used is UTF-7.
- `q` means it's using **"quoted-printable" encoding**, often used in email headers.
- So it's a **MIME-encoded email string**.

🧠 Step 2: Strip the wrapper

You can extract this part from the MIME wrapper:

```
pgsql
CopyEdit
attacker&AEA-exploit-0a8b00fa04d93dfc80d7612c011e004d.exploit-serve
```

```
r.net&ACA-
```

Then this part comes after:

```
graphql
CopyEdit
=@ginandjuice.shop
```

Now combine them:

```
css
CopyEdit
attacker&AEA-exploit-0a8b00fa04d93dfc80d7612c011e004d.exploit-serve
r.net&ACA-@ginandjuice.shop
```

✨ Step 3: Decode special tokens

In **quoted-printable encoding**:

- `&AEA-` = `@`
- `&ACA-` = `.`

So let's **replace** them:

```
css
CopyEdit
attacker@exploit-0a8b00fa04d93dfc80d7612c011e004d.exploit-server.ne
t.@ginandjuice.shop
```

Hmm... this looks weird because there's a **dot just before the @ginandjuice.shop**. It's likely trying to mess with **email validation logic**.

🧩 Final Decoded Email (Intended):

Most likely intended to be:

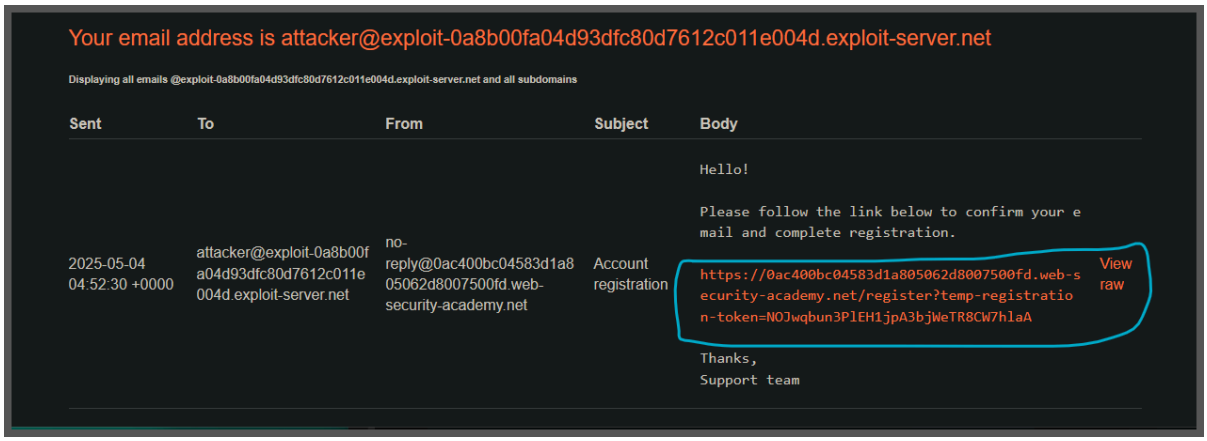
```
css
CopyEdit
attacker@exploit-0a8b00fa04d93dfc80d7612c011e004d.exploit-server.net
```

But wrapped in a **MIME-encoded way** and **split across the domain** `@ginandjuice.shop`, possibly to bypass filters or exploit bugs in email parsing.

🔥 What's the goal of this?

The attacker is:

- **Registering with a weird-looking email address**.
- Trying to **confuse the backend logic** (like how it verifies or validates emails).
- Possibly trying to:
  - Bypass email filters.
  - Hijack flows where emails are auto-sent.
  - Exploit SSRF or misrouted logic in how emails are handled.



you your admin