Access control

Lab: Unprotected admin functionality

**APPRENTICE |** This lab has an unprotected admin panel.

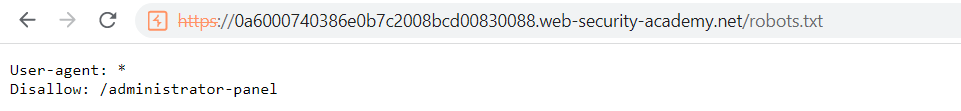
Solve the lab by deleting the user carlos. [**Access the lab**](https://portswigger.net/academy/labs/launch/efb3cd7d563d637400d229d7b4949233047f1b2808b1d46393322d66594a8de2?referrer=%2fweb-security%2faccess-control%2flab-unprotected-admin-functionality)

**Solution**

1. Go to the lab and view robots.txt by appending /robots.txt to the lab URL. Notice that the Disallow line discloses the path to the admin panel.
2. In the URL bar, replace /robots.txt with /administrator-panel to load the admin panel.
3. Delete carlos.

<https://0a6000740386e0b7c2008bcd00830088.web-security-academy.net/>

<https://0a6000740386e0b7c2008bcd00830088.web-security-academy.net/robots.txt>



<https://0a6000740386e0b7c2008bcd00830088.web-security-academy.net/administrator-panel>

delete carlos

Lab: Unprotected admin functionality with unpredictable URL

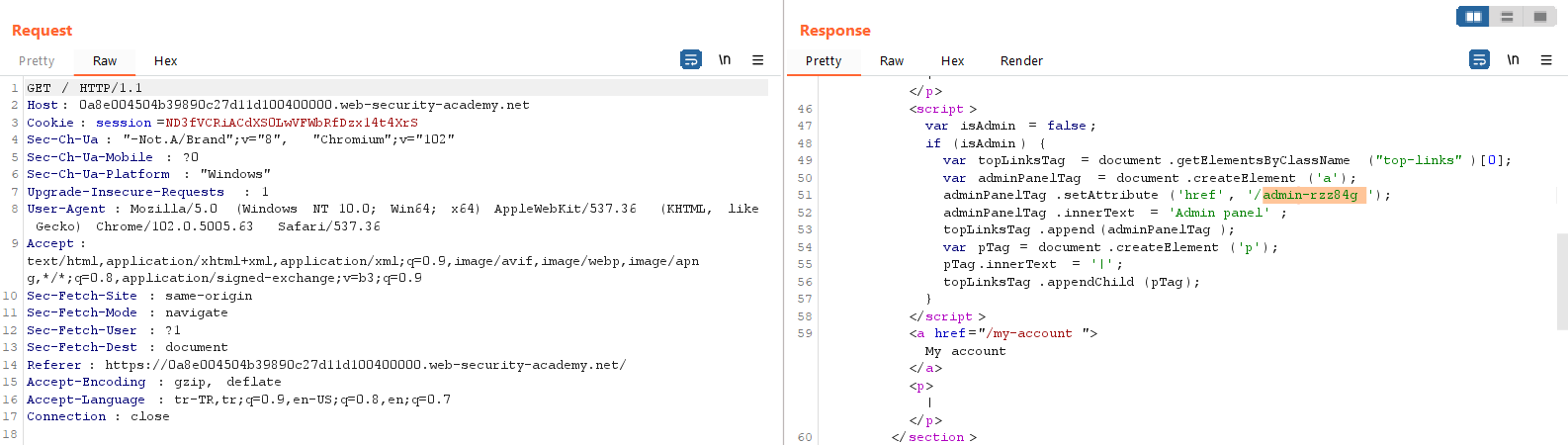
**APPRENTICE**

This lab has an unprotected admin panel. It's located at an unpredictable location, but the location is disclosed somewhere in the application.

Solve the lab by accessing the admin panel, and using it to delete the user carlos. [**Access the lab**](https://portswigger.net/academy/labs/launch/878a6a3a7bf5ca38441716313e1ad46697052f36b6c82c4a6bda1a98e28f446f?referrer=%2fweb-security%2faccess-control%2flab-unprotected-admin-functionality-with-unpredictable-url)

**Solution**

1. Review the lab home page's source using Burp Suite or your web browser's developer tools.
2. Observe that it contains some JavaScript that discloses the URL of the admin panel.
3. Load the admin panel and delete carlos.



Go to <https://0a8e004504b39890c27d11d100400000.web-security-academy.net/admin-rzz84g>

And delete Carlos

Lab: User role controlled by request parameter

**APPRENTICE**

This lab has an admin panel at /admin, which identifies administrators using a forgeable cookie.

Solve the lab by accessing the admin panel and using it to delete the user carlos.

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

[**Access the lab**](https://portswigger.net/academy/labs/launch/f97895e072d86380dd3fdd12e2d6ceb6090708b4e3e0cd0990fe741afb553b2e?referrer=%2fweb-security%2faccess-control%2flab-user-role-controlled-by-request-parameter)

**Solution**

1. Browse to /admin and observe that you can't access the admin panel.
2. Browse to the login page.
3. In Burp Proxy, turn interception on and enable response interception.
4. Complete and submit the login page, and forward the resulting request in Burp.
5. Observe that the response sets the cookie Admin=false. Change it to Admin=true.
6. Load the admin panel and delete Carlos

Go to <https://0abe0024047a553ac256134200c1000d.web-security-academy.net/admin>

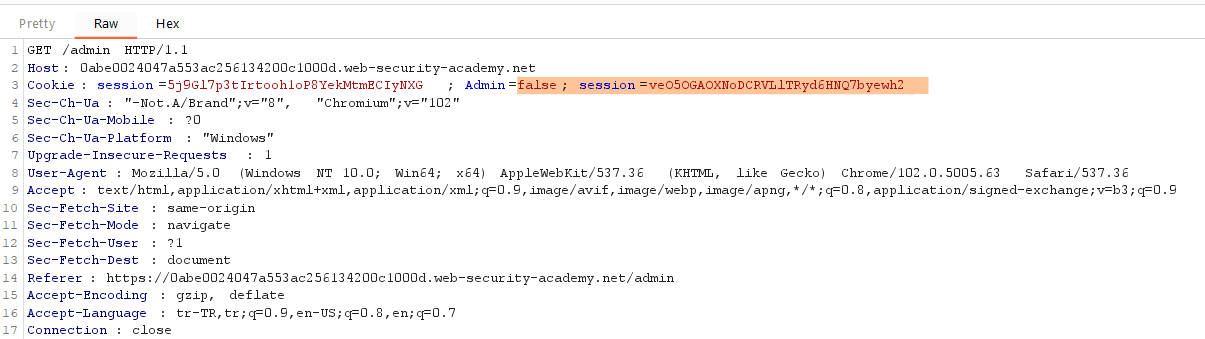


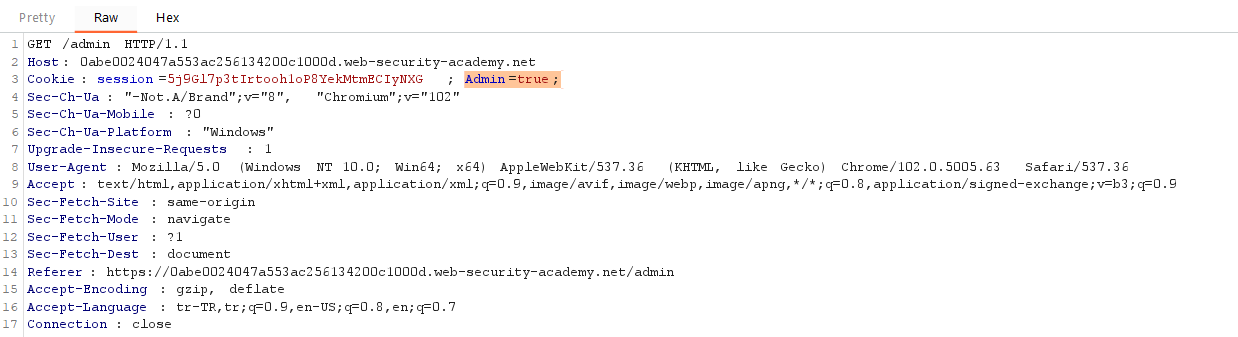
Change selected area to 🡪 Admin=true;

At the same time delete session token

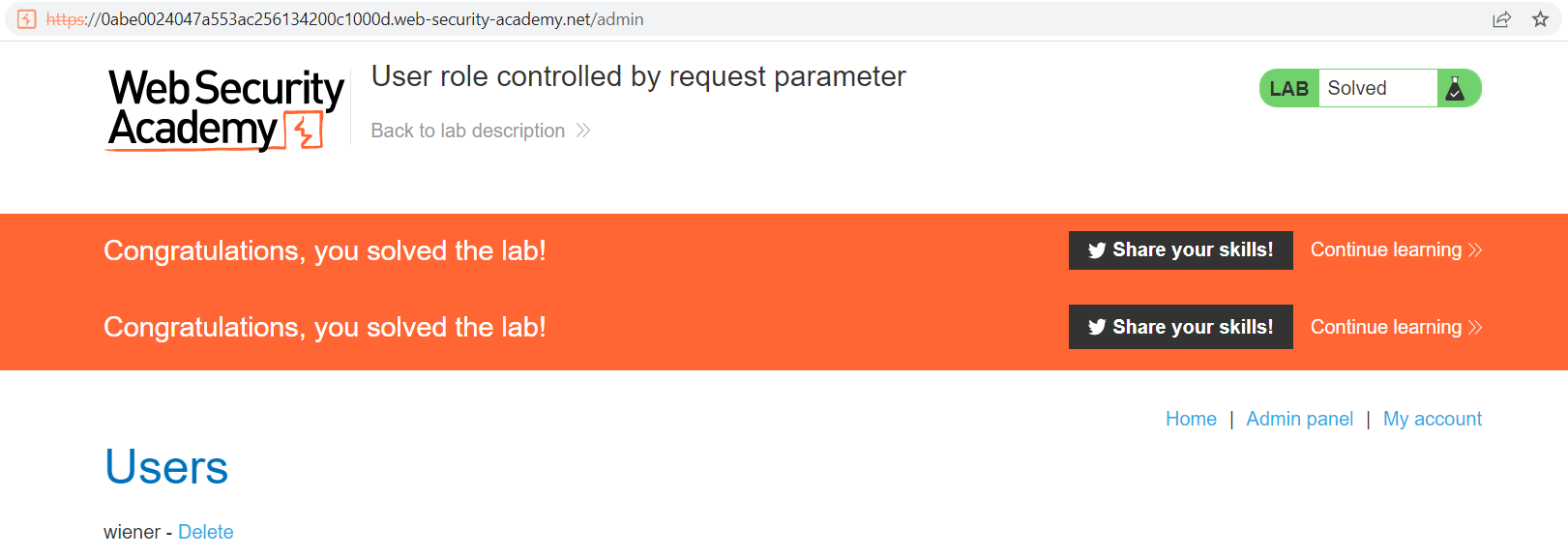


When you go to admin panel change every time selected area that writed way.





Now you can delete user now



Lab: User role can be modified in user profile

**APPRENTICE**

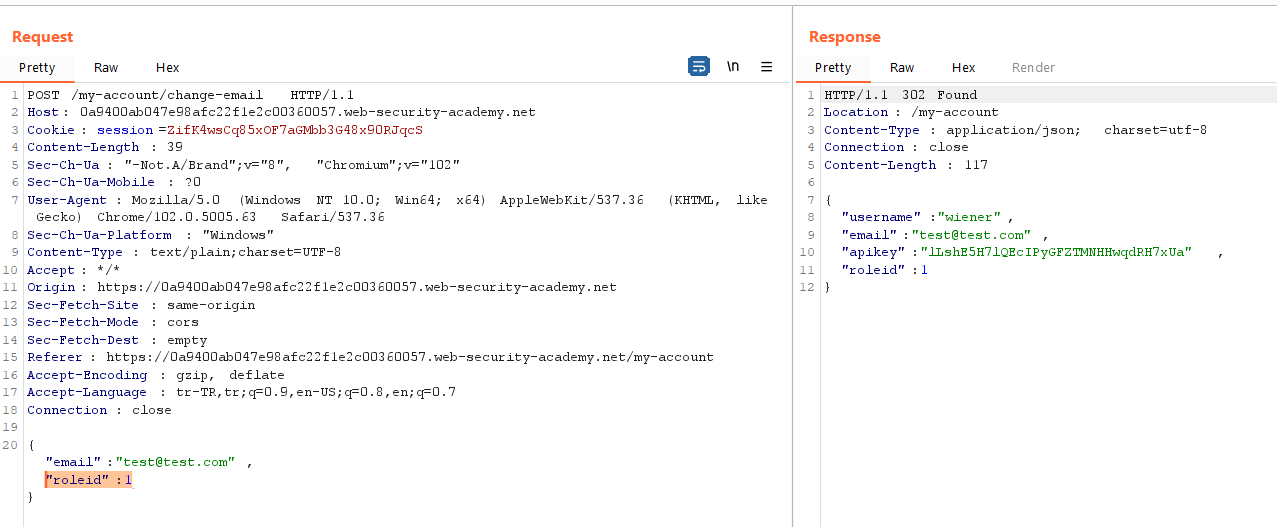
This lab has an admin panel at /admin. It's only accessible to logged-in users with a roleid of 2.

Solve the lab by accessing the admin panel and using it to delete the user carlos.

You can log in to your own account using the following credentials: wiener:peter [**Access the lab**](https://portswigger.net/academy/labs/launch/703ed76cc046c2c9c06b661c1c25f53f7c1cb238e9b7fa9aff664177fd7c7773?referrer=%2fweb-security%2faccess-control%2flab-user-role-can-be-modified-in-user-profile)

**Solution**

1. Log in using the supplied credentials and access your account page.
2. Use the provided feature to update the email address associated with your account.
3. Observe that the response contains your role ID.
4. Send the email submission request to Burp Repeater, add "roleid":2 into the JSON in the request body, and resend it.
5. Observe that the response shows your roleid has changed to 2.
6. Browse to /admin and delete carlos.



When you want to updadate email roleid is equal to 1 but if you change to 2, you will get admin authority, then g oto admin panel and delete Carlos.



Lab: User ID controlled by request parameter **APPRENTICE**

This lab has a horizontal privilege escalation vulnerability on the user account page.

To solve the lab, obtain the API key for the user carlos and submit it as the solution.

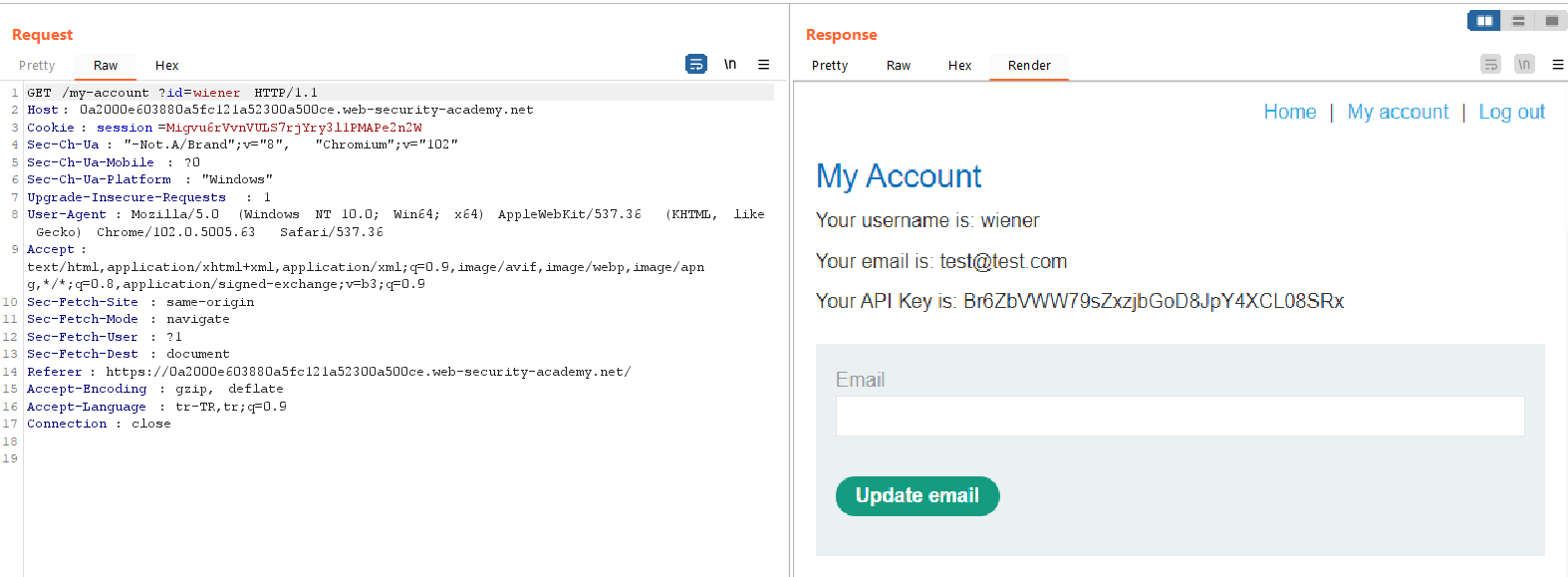
You can log in to your own account using the following credentials: wiener:peter

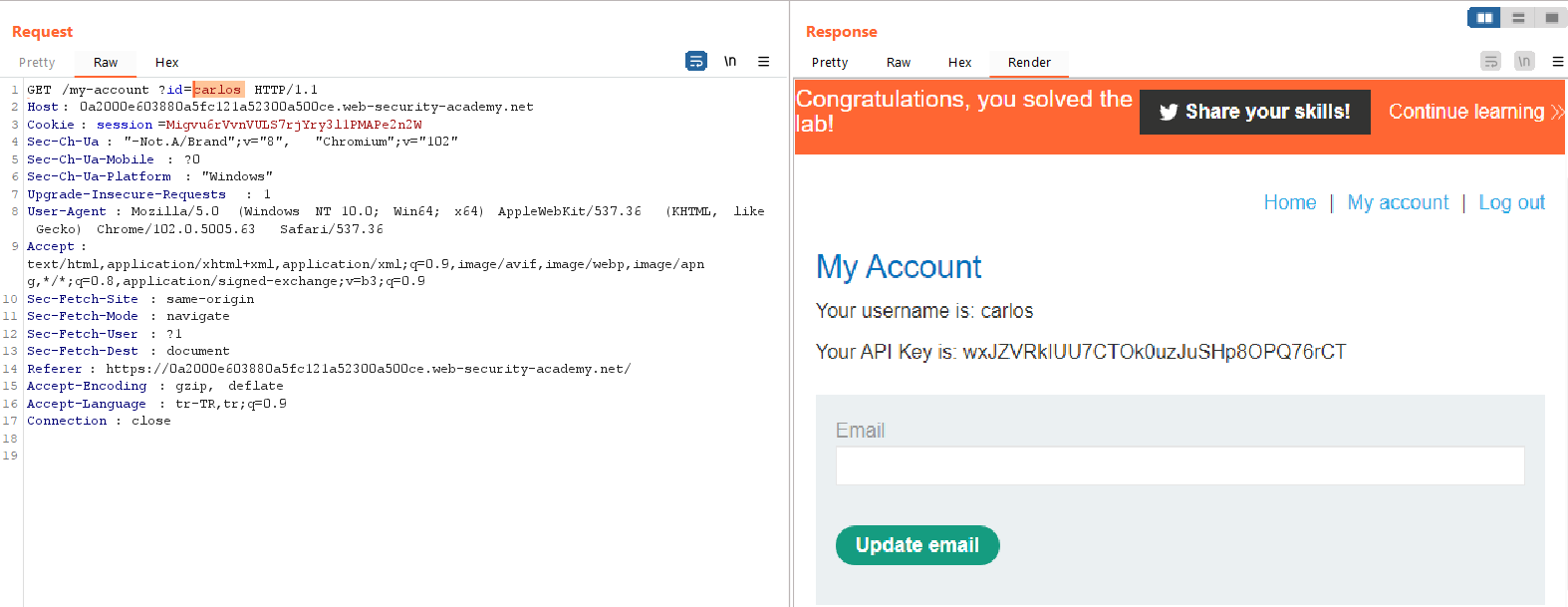
[**Access the lab**](https://portswigger.net/academy/labs/launch/cf2d0a74362ee20a8f6ffb1028e3b44fee9ae125f418ed15832e4a8a2073bff9?referrer=%2fweb-security%2faccess-control%2flab-user-id-controlled-by-request-parameter)

**Solution**

1. Log in using the supplied credentials and go to your account page.
2. Note that the URL contains your username in the "id" parameter.
3. Send the request to Burp Repeater.
4. Change the "id" parameter to carlos.
5. Retrieve and submit the API key for carlos.

Hesaba giriş yaptıktan sonra my account a tıklayıp requesti yakalarsan şöyle oluyor. Wieneri Carlos yaparsan carlosun api keyini getirebiliyorsun.





Lab: User ID controlled by request parameter, with unpredictable user IDs

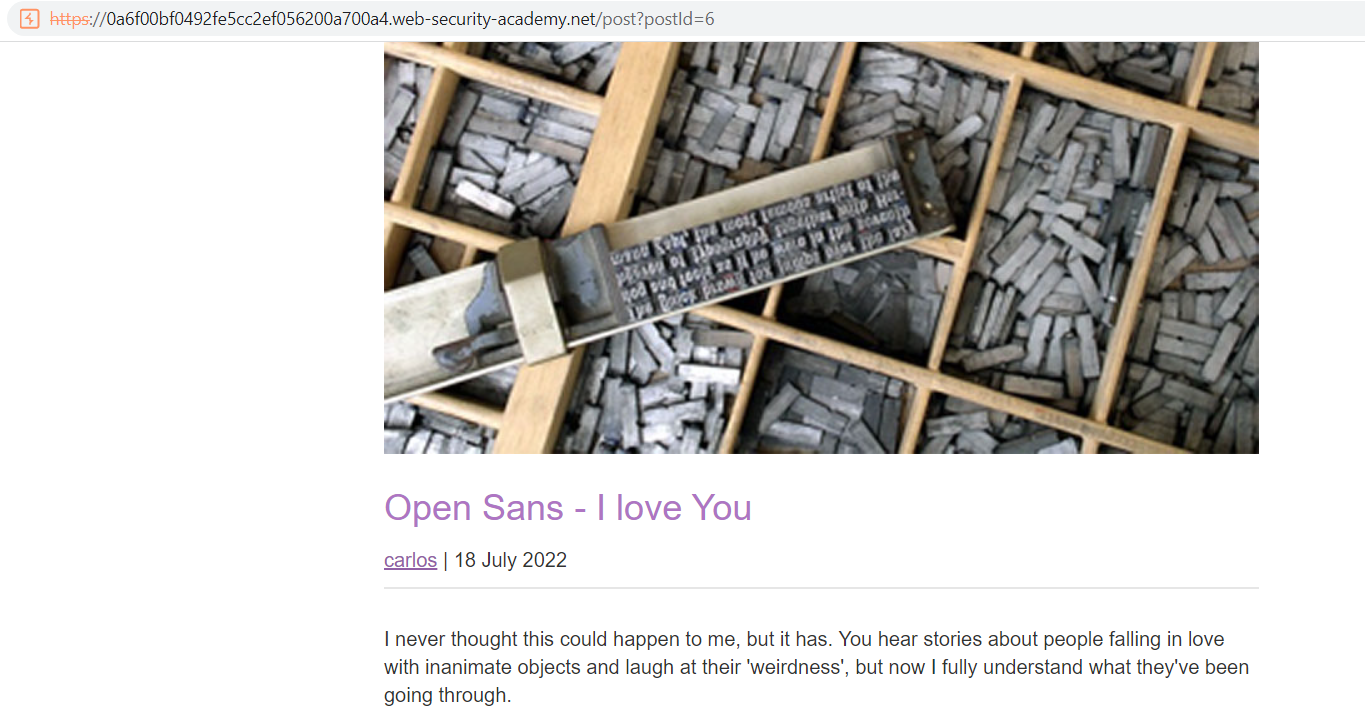
**APPRENTICE** This lab has a horizontal privilege escalation vulnerability on the user account page, but identifies users with GUIDs.To solve the lab, find the GUID for carlos, then submit his API key as the solution.You can log in to your own account using the following credentials: wiener:peter

[**Access the lab**](https://portswigger.net/academy/labs/launch/9d95a9e677b81f605d40d00001ffbc8bbb0f0ee73773a3a37727fbde0406b779?referrer=%2fweb-security%2faccess-control%2flab-user-id-controlled-by-request-parameter-with-unpredictable-user-ids)

**Solution**

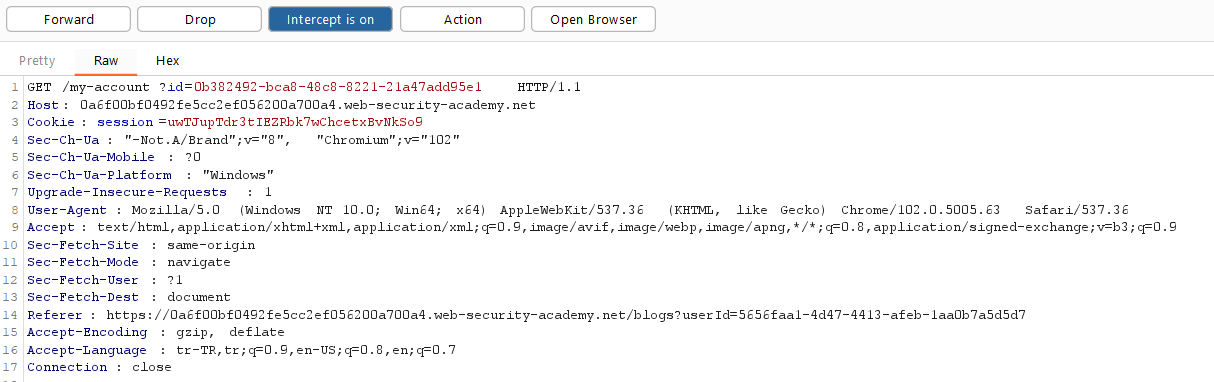
1. Find a blog post by carlos.
2. Click on carlos and observe that the URL contains his user ID. Make a note of this ID.
3. Log in using the supplied credentials and access your account page.
4. Change the "id" parameter to the saved user ID.
5. Retrieve and submit the API key.

Go to Carlos blog and click to Carlos you will see the Carlos id in url



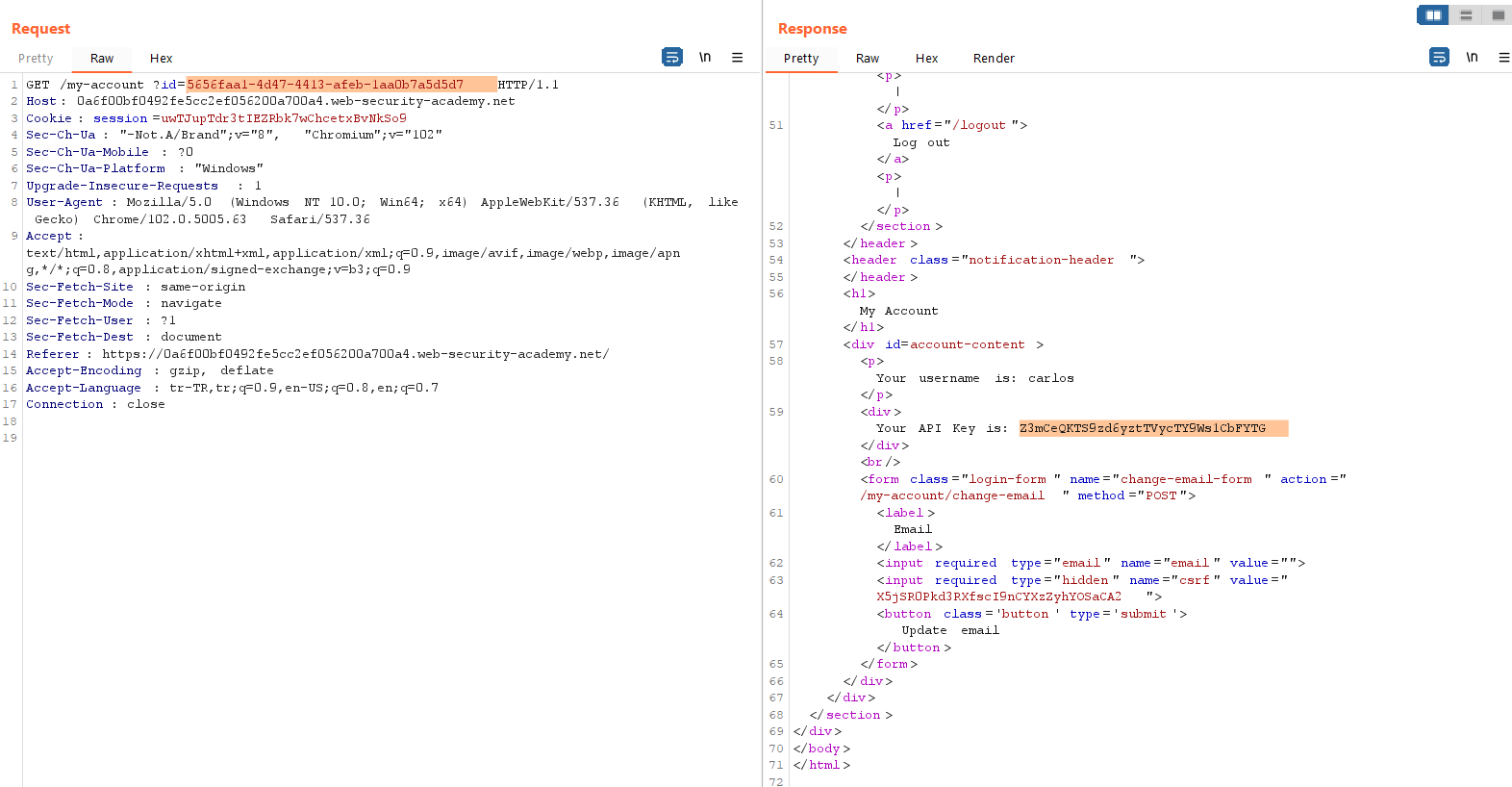


Then g oto my account intercept this request change id to below id which in the url



Change id and send request

Response 🡪



Lab: User ID controlled by request parameter with data leakage in redirect

**APPRENTICE**

This lab contains an [access control](https://portswigger.net/web-security/access-control) vulnerability where sensitive information is leaked in the body of a redirect response.

To solve the lab, obtain the API key for the user carlos and submit it as the solution.

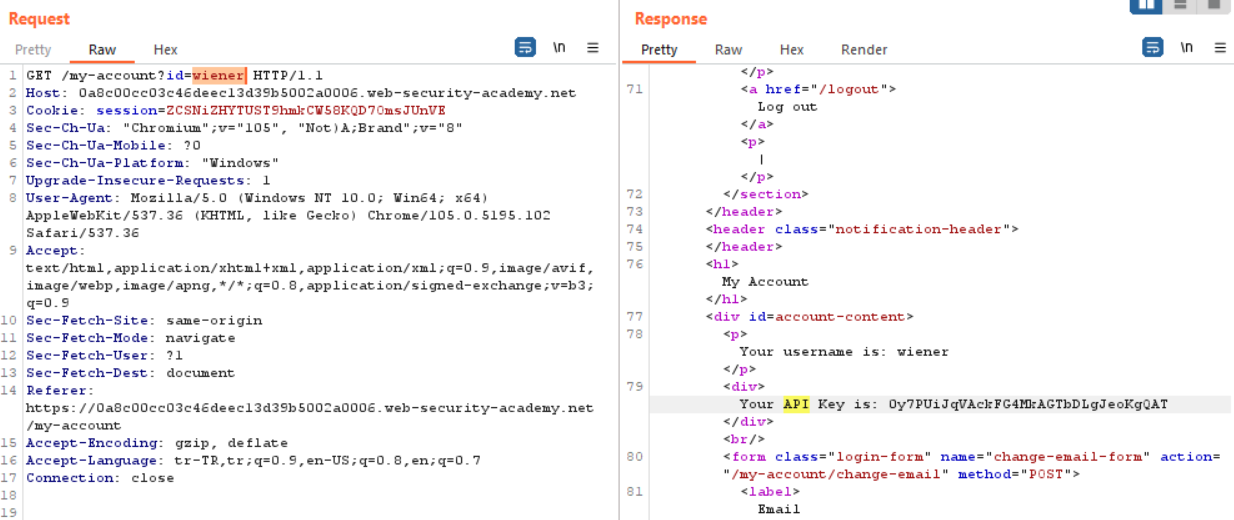
You can log in to your own account using the following credentials: wiener:peter

[**Access the lab**](https://portswigger.net/academy/labs/launch/114de45758e410d7a7458144d568a9c3b78a4be3a14daf61d4e552610739359d?referrer=%2fweb-security%2faccess-control%2flab-user-id-controlled-by-request-parameter-with-data-leakage-in-redirect)

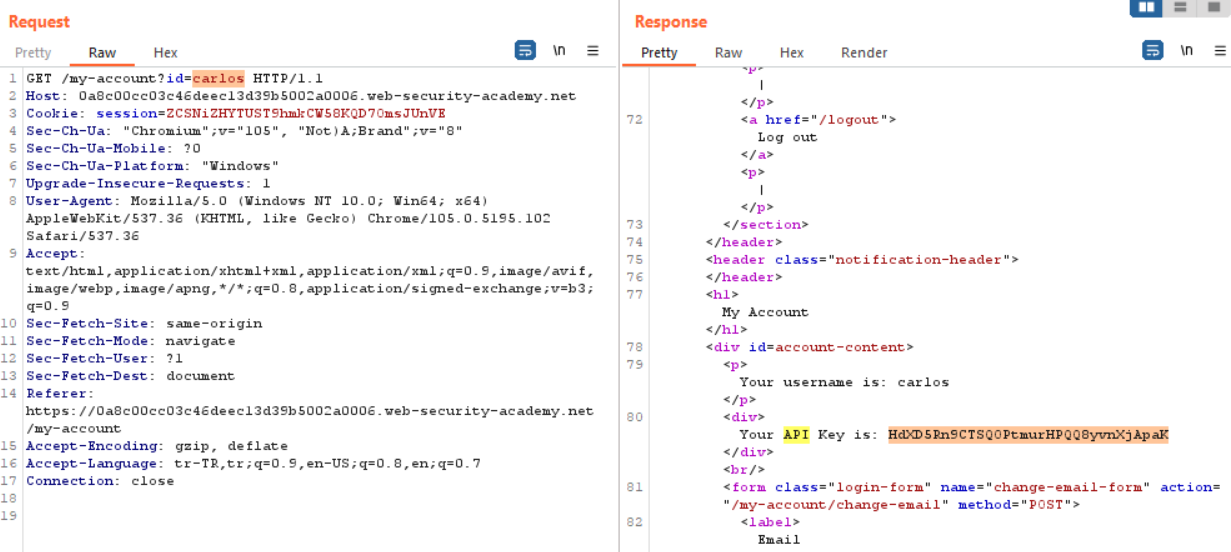
**Solution**

1. Log in using the supplied credentials and access your account page.
2. Send the request to Burp Repeater.
3. Change the "id" parameter to carlos.
4. Observe that although the response is now redirecting you to the home page, it has a body containing the API key belonging to carlos.
5. Submit the API key.

Original request



Changed request.



Lab: User ID controlled by request parameter with password disclosure

**APPRENTICE**

This lab has user account page that contains the current user's existing password, prefilled in a masked input.

To solve the lab, retrieve the administrator's password, then use it to delete carlos.

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

[**Access the lab**](https://portswigger.net/academy/labs/launch/9038df39e72961a071967a83c2472bf2edb0ef9595e6f13bd05ce1b2dce81234?referrer=%2fweb-security%2faccess-control%2flab-user-id-controlled-by-request-parameter-with-password-disclosure)

**Solution**

1. Log in using the supplied credentials and access the user account page.
2. Change the "id" parameter in the URL to administrator.
3. View the response in Burp and observe that it contains the administrator's password.
4. Log in to the administrator account and delete carlos.

Solution

GET /my-account?id=carlos HTTP/1.1

GET /my-account?id=administrator HTTP/1.1

Lab: Insecure direct object references

**APPRENTICE**

This lab stores user chat logs directly on the server's file system, and retrieves them using static URLs.

Solve the lab by finding the password for the user carlos, and logging into their account.

[**Access the lab**](https://portswigger.net/academy/labs/launch/bf6b2f755e63ba10eefdb6633048fc5cf8e4c1d5f386a756fd8e4b6864d70857?referrer=%2fweb-security%2faccess-control%2flab-insecure-direct-object-references)

**Solution**

1. Select the **Live chat** tab.
2. Send a message and then select **View transcript**.
3. Review the URL and observe that the transcripts are text files assigned a filename containing an incrementing number.
4. Change the filename to 1.txt and review the text. Notice a password within the chat transcript.
5. Return to the main lab page and log in using the stolen credentials.

Lab: URL-based access control can be circumvented

**PRACTITIONER**

This website has an unauthenticated admin panel at /admin, but a front-end system has been configured to block external access to that path. However, the back-end application is built on a framework that supports the X-Original-URL header.

To solve the lab, access the admin panel and delete the user carlos.[**Access the lab**](https://portswigger.net/academy/labs/launch/9955f5350904edabe30089828d2c08ace886493264a0b469354503102379a3e9?referrer=%2fweb-security%2faccess-control%2flab-url-based-access-control-can-be-circumvented)

**Solution**

1. Try to load /admin and observe that you get blocked. Notice that the response is very plain, suggesting it may originate from a front-end system.
2. Send the request to Burp Repeater. Change the URL in the request line to / and add the HTTP header X-Original-URL: /invalid. Observe that the application returns a "not found" response. This indicates that the back-end system is processing the URL from the X-Original-URL header.
3. Change the value of the X-Original-URL header to /admin. Observe that you can now access the admin page.
4. To delete the user carlos, add ?username=carlos to the real query string, and change the X-Original-URL path to /admin/delete.

STEPS:

<https://0a2100130410e5f3c0d93def000400a0.web-security-academy.net/>

<https://0a2100130410e5f3c0d93def000400a0.web-security-academy.net/admin>

GET / HTTP/1.1

add the HTTP header X-Original-URL: /invalid

GET /?username=carlos HTTP/1.1

.

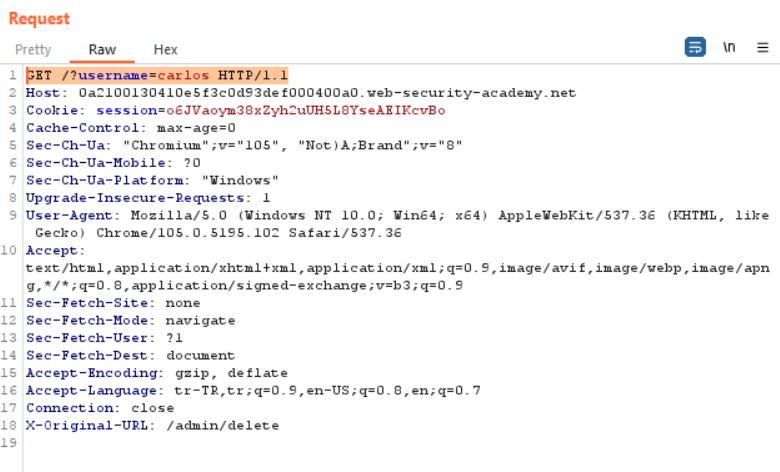
.

X-Original-URL: /admin/delete

Original request:



Changed request:



Lab: Method-based access control can be circumvented

**PRACTITIONER**

This lab implements [access controls](https://portswigger.net/web-security/access-control) based partly on the HTTP method of requests. You can familiarize yourself with the admin panel by logging in using the credentials administrator:admin.

To solve the lab, log in using the credentials wiener:peter and exploit the flawed access controls to promote yourself to become an administrator. [**Access the lab**](https://portswigger.net/academy/labs/launch/e115e4b43d59771bb787573f305861a795fb30f2447cef9b53618f4232d1c83b?referrer=%2fweb-security%2faccess-control%2flab-method-based-access-control-can-be-circumvented)

**Solution**

1. Log in using the admin credentials.
2. Browse to the admin panel, promote carlos, and send the HTTP request to Burp Repeater.
3. Open a private/incognito browser window, and log in with the non-admin credentials.
4. Attempt to re-promote carlos with the non-admin user by copying that user's session cookie into the existing Burp Repeater request, and observe that the response says "Unauthorized".
5. Change the method from POST to POSTX and observe that the response changes to "missing parameter".
6. Convert the request to use the GET method by right-clicking and selecting "Change request method".
7. Change the username parameter to your username and resend the request.

Login with credentials administrator: admin

Promote carlos user

Open new session with wiener: peter credentials

Lab: Method-based access control can be circumvented

**PRACTITIONER**

This lab implements [access controls](https://portswigger.net/web-security/access-control) based partly on the HTTP method of requests. You can familiarize yourself with the admin panel by logging in using the credentials administrator:admin.

To solve the lab, log in using the credentials wiener:peter and exploit the flawed access controls to promote yourself to become an administrator. [**Access the lab**](https://portswigger.net/academy/labs/launch/e115e4b43d59771bb787573f305861a795fb30f2447cef9b53618f4232d1c83b?referrer=%2fweb-security%2faccess-control%2flab-method-based-access-control-can-be-circumvented)

**Solution**

1. Log in using the admin credentials.
2. Browse to the admin panel, promote carlos, and send the HTTP request to Burp Repeater.
3. Open a private/incognito browser window, and log in with the non-admin credentials.
4. Attempt to re-promote carlos with the non-admin user by copying that user's session cookie into the existing Burp Repeater request, and observe that the response says "Unauthorized".
5. Change the method from POST to POSTX and observe that the response changes to "missing parameter".
6. Convert the request to use the GET method by right-clicking and selecting "Change request method".
7. Change the username parameter to your username and resend the request.

Olayın özeti:

Admin bilgisi ile oturum açılıyor.

Ayrı browserdan (gizli sekmeden de oturum açabilirsin, bu sessionların karışmaması için, yani 2 ayrı oturum açacaz özet olarak) normal user olarak oturum açılıyor.

Admin hesabında carlos upgrade ediliyor.

Normal userın session id si alınarak adminde carlos upgrade edilirken session bilgisi normal user ile değiştiriliyor. Ve upgrade de downgrade olarak değiştiriyor. POST ->> POSTX ile değiştiriliyor.

Sonuç unauthorized.

Ama metod çeşidisi change request metot ile get e çekip username kısmını wiener ile değiştirdiğimizde kendimizi upgrade etmiş oluyor.

Sonuç:

Admin kullanıcısının hesabından upgrade user fonksiyonu kullanılarak session id si normal kullanıcı id si ile değiştirilerek ve metot tipi ilk önce postx yapılıp sonra metot change request metot ile gete çekilerek yetkisiz biçimde kendimiz admin yapabiliyoruz.

Olayın saçmalığı admin account takeover yapmadan admin sessionı üzerinden kendimizi admin yapmak. Session id change kurgulanmış ama bu şekilde olmamalı bence.

Lab: Multi-step process with no access control on one step

**PRACTITIONER**

This lab has an admin panel with a flawed multi-step process for changing a user's role. You can familiarize yourself with the admin panel by logging in using the credentials administrator:admin.

To solve the lab, log in using the credentials wiener:peter and exploit the flawed [access controls](https://portswigger.net/web-security/access-control) to promote yourself to become an administrator.

[**Access the lab**](https://portswigger.net/academy/labs/launch/48ea5eff7577d0d5e50c64b5d91f3e2fcc175ebb055c95f30bf8e909f6ba9d2c?referrer=%2fweb-security%2faccess-control%2flab-multi-step-process-with-no-access-control-on-one-step)

**Solution**

1. Log in using the admin credentials.
2. Browse to the admin panel, promote carlos, and send the confirmation HTTP request to Burp Repeater.
3. Open a private/incognito browser window, and log in with the non-admin credentials.
4. Copy the non-admin user's session cookie into the existing Repeater request, change the username to yours, and replay it.

Burda olan olay kısaca şu

Admin bilgileri ile giriş yapılıyor.

Admin hesabından carlos kullanıcısının yetkisi yükseltiliyor.

Normal kullanıcı ile giriş yapılıyor.

Admin kullanıcısında carlos hesabının yetkisi yükseltilirken kullanılan requestte

Adminin session id sis normal kullanıcıcının session id si ile değiştiriiliyor.

Ve username bilgisi de wiener olarak değiştiriliyor.

Ve normal kullanıcısının session bilgisi kullanılarak admin yetkisi elde ediliyor.

Burda sıkıntı admin hesabında başkasının yetkisi yükseltilirken son adımda işlemi gerçekleştiren kişinin admin olup olmadığının kontrol edilmemesi yani session id si son adımda kontrol edilmiyor.

Lab: Referer-based access control

**PRACTITIONER**

This lab controls access to certain admin functionality based on the Referer header. You can familiarize yourself with the admin panel by logging in using the credentials administrator:admin.

To solve the lab, log in using the credentials wiener:peter and exploit the flawed [access controls](https://portswigger.net/web-security/access-control) to promote yourself to become an administrator.

[**Access the lab**](https://portswigger.net/academy/labs/launch/16e507638191c125691e9f9d752a1d42bf2e8fa811002b872ca607bd2cd8c334?referrer=%2fweb-security%2faccess-control%2flab-referer-based-access-control)

**Solution**

1. Log in using the admin credentials.
2. Browse to the admin panel, promote carlos, and send the HTTP request to Burp Repeater.
3. Open a private/incognito browser window, and log in with the non-admin credentials.
4. Browse to /admin-roles?username=carlos&action=upgrade and observe that the request is treated as unauthorized due to the absent Referer header.
5. Copy the non-admin user's session cookie into the existing Burp Repeater request, change the username to yours, and replay it.

Bu lab yukarda anlatılan işlemlerin tamamen aynısı. Sadece endpointte isim değişmiş.