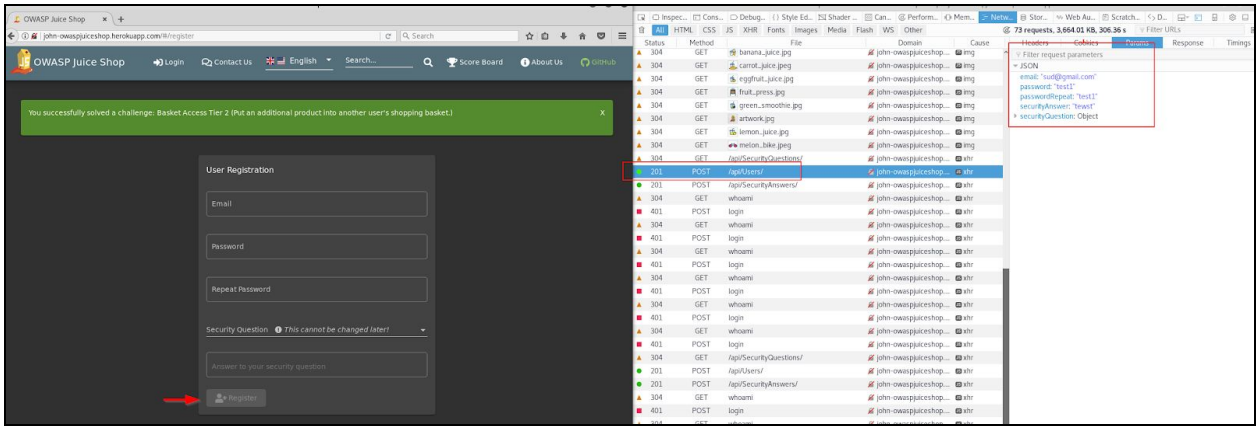
Medium Challenges

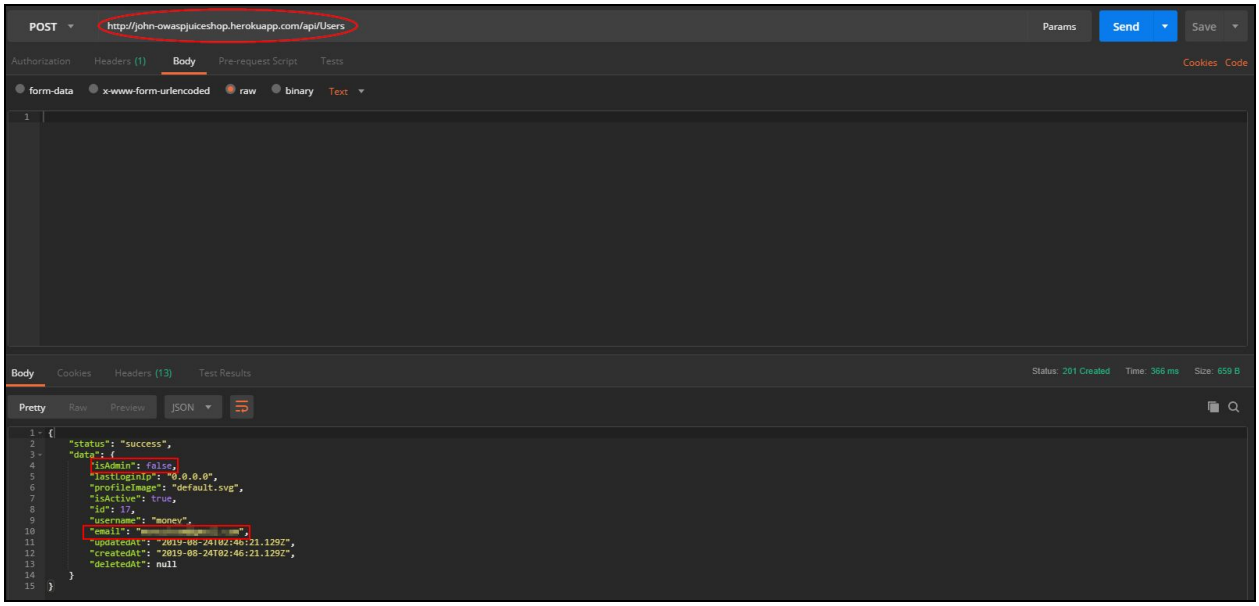
Admin Registration - Get registered as admin user

1. Go to the “User Registration” page and open the Developer Tools (Network tab) at the same time.

2. Register a new user. Remember the API call ⇒ /api/Users/

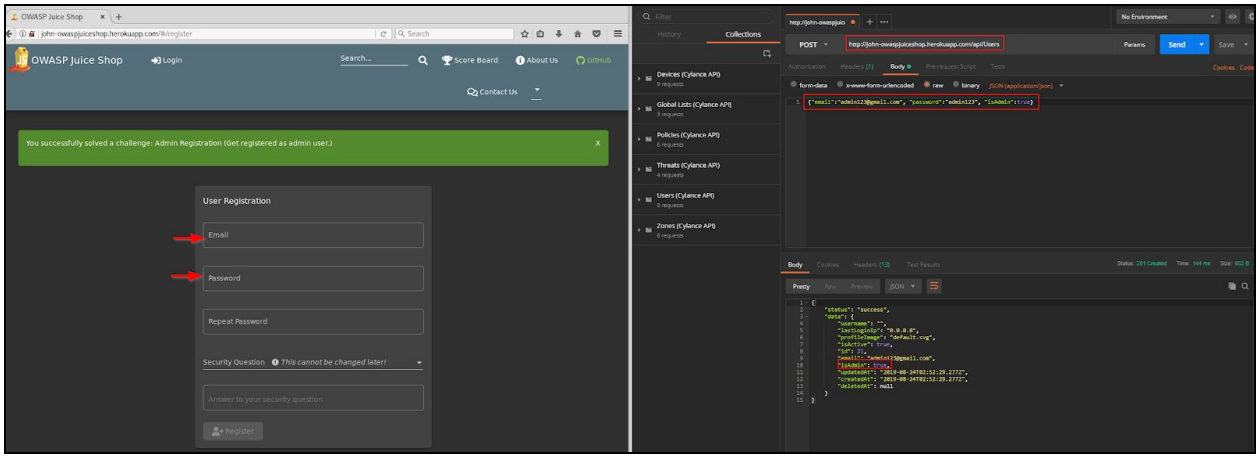


3. Open Postman and perform the same API call (without Body content). Observe the output of the response. There is a parameter called isAdmin that is set to false by default.



Written by Jonathan Harijanto

4. Perform another API call to /api/Users, only this time include some contents in the Body: email, password, and isAdmin parameters. Send the information via POST request

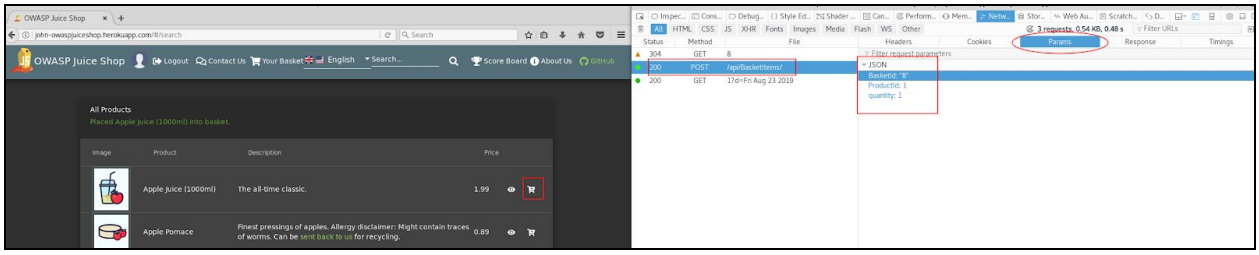


Written by Jonathan Harijanto

Basket Access Tier 2 - Put an additional product into another user's shopping basket

1. Login with any user and open the “Developer Tools” → “Network” tab

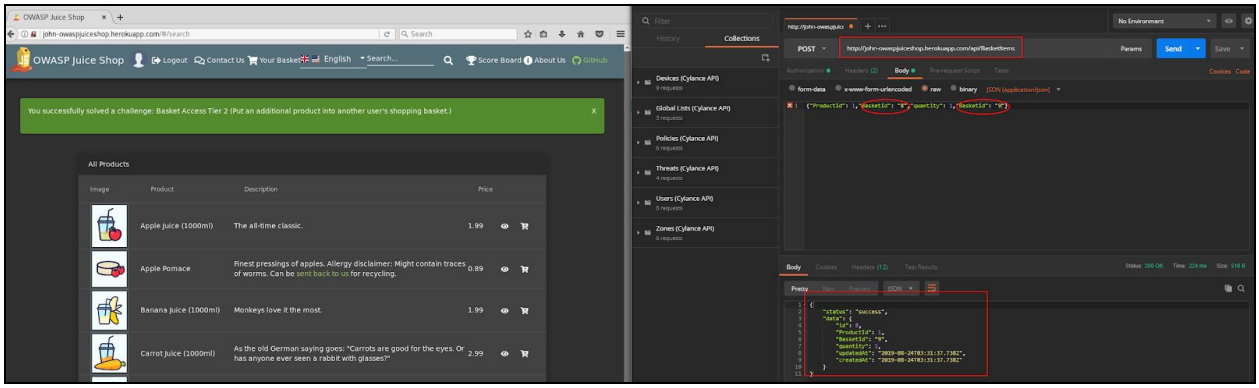
2. Go to the store (homepage) and add a random item into the basket. Observe the API being called (/api/BasketItems) and the parameters sent (BasketId, ProductId, quantity). Also, it doesn’t hurt to memorize the value of the BasketId



3. Let’s perform an HTTP Parameter Pollution attack (https://www.imperva.com/learn/application-security/http-parameter-pollution/)

4. Open Postman and use the /api/BasketItems address in the address bar. In the Body section, provide a “payload” that looks something like this:

{"ProductId": 1,"BasketId": "8","quantity": 1,"BasketId": "9"}

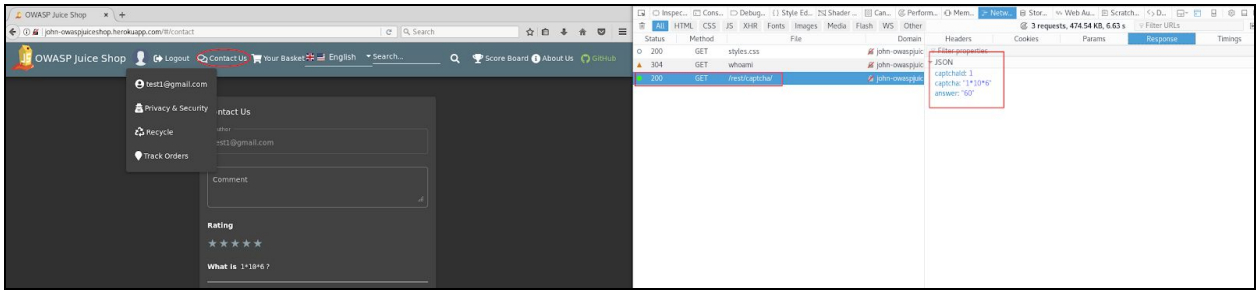


Note: Play around with the value of the BasketIds. I noticed that in order for the HPP attack to work, our BasketId value should be lower than the victim’s BasketId. In this case, mine is ‘8’ and the victim is ‘9’.

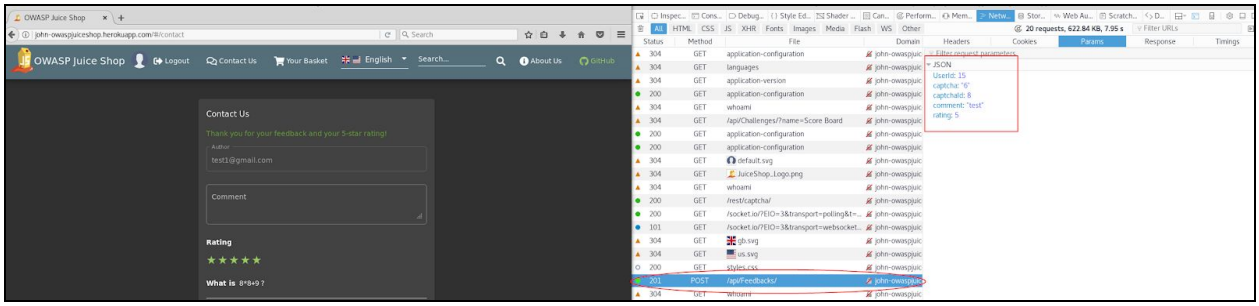
Written by Jonathan Harijanto

CAPTCHA Bypass Tier 1 - Submit 10 or more customer feedbacks within 10 seconds

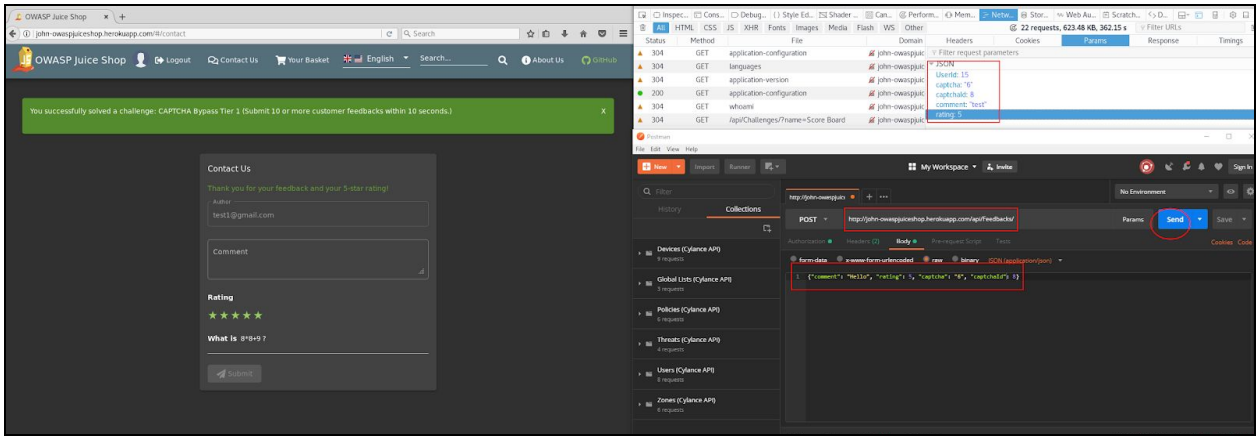
1. Login with any user. Go to the “Contact Us” page and open the “Developer Tools” → Network tab. Look at the API call to “/rest/captcha” and observe the Response (captchaId, captcha, answer)



2. Submit one random feedback to analyze the network activities. Notice that it is doing an API call to “/api/Feedbacks/”



3. Open Postman. Paste the API call to Feedbacks (/api/Feedbacks) in the address bar. Also, copy the information from the “Params” tab and paste it into the Body. Basically, we are going to reuse the old, but verified, captcha to send a feedback 10 times in a row. Hit the Send button 10x.



Written by Jonathan Harijanto

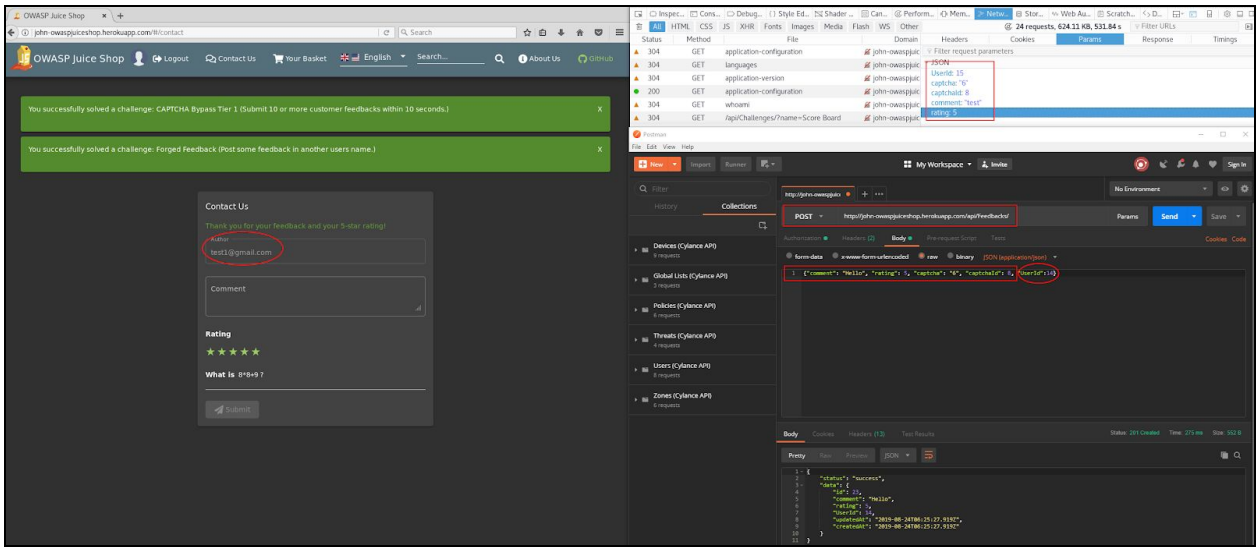
Forged Feedback - Post some feedback in another users name

1. Login with any user. Go to the “Contact Us” page and open the “Developer Tools” → Network tab.

2. Submit one random feedback to analyze the network activities. Notice that it is doing an API call to “/api/Feedbacks/”

3. Analyze the value of UserId. In this scenario, user test1@gmail.com has a UserId equal to 15. This shows that there are other users before test1 with UserId of 14, 13, etc.

4. Open Postman. Paste the API call to Feedbacks (/api/Feedbacks) in the address bar. Also, copy the information from the “Params” tab and paste it into the Body. Make sure to include UserId parameter and replace the value with something below 15.

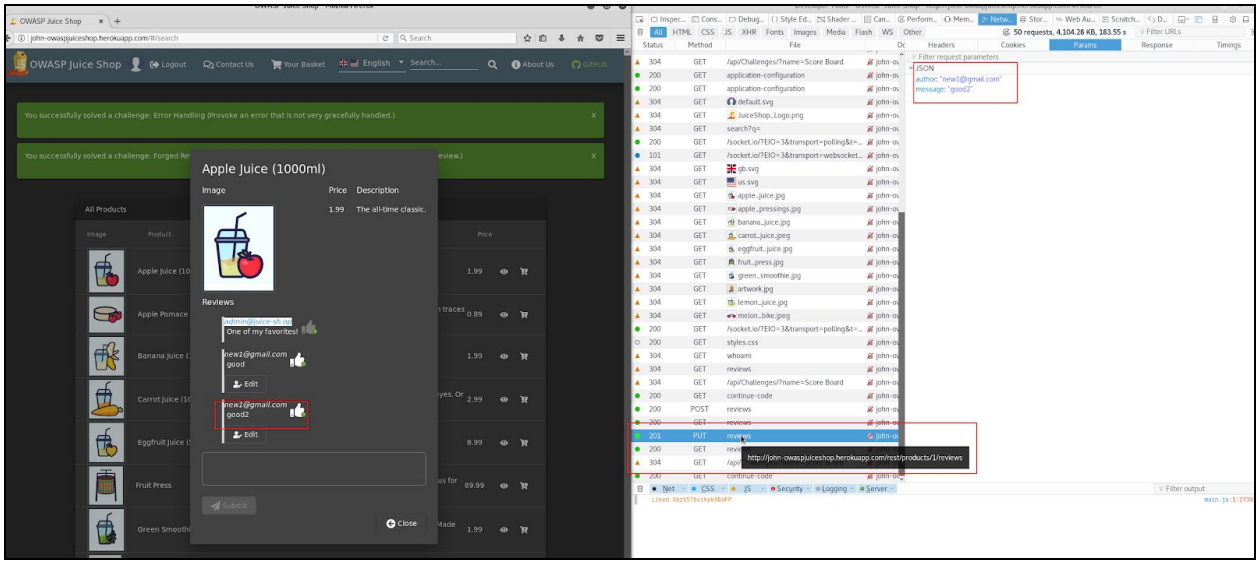


Written by Jonathan Harijanto

Forged Review - Post a product review as another user or edit any user's existing review

1. Login with any user and open the “Developer Tools” → Network tab.

2. Post a single review on any item in the store and observe that the app is doing an API call to /products/1/reviews via PUT request. Also, remember the parameters sent (author & message)

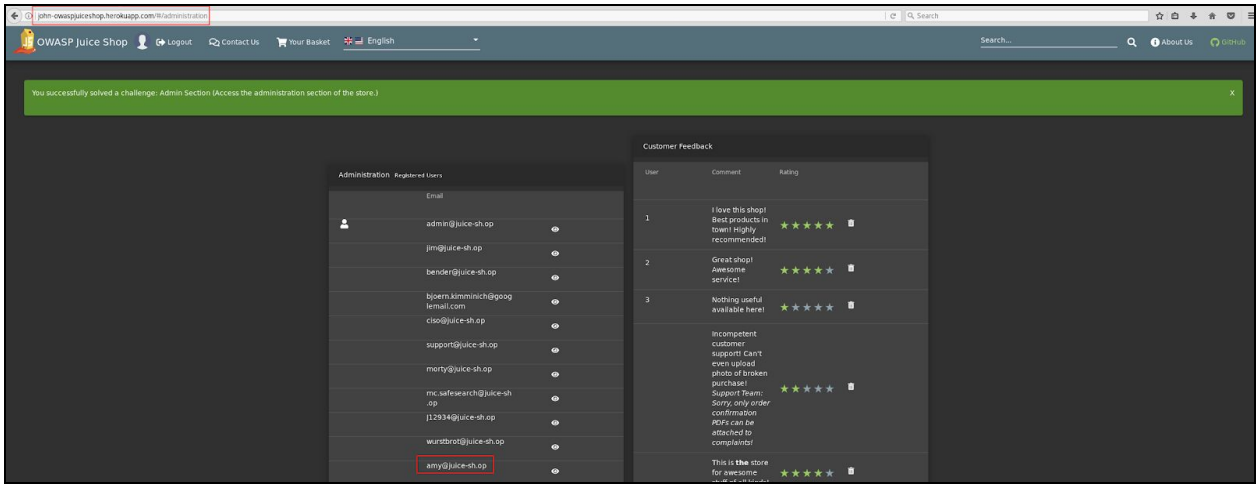


3. Open Postman. Paste the API call to Reviews (/rest/products/1/reviews) in the address bar. Also, copy the information from the “Params” tab and paste it into the Body. This time, change the value of the author from new1@gmail.com to admin@juice-sh.op. Send it via PUT request.

Written by Jonathan Harijanto

Login Amy - Log in with Amy's original user credentials. (This could take 93.83 billion trillion trillion centuries to brute force, but luckily she did not read the "One Important Final Note")

1. Login as Admin and go to the Administration page (/administration). Find Amy's email address ⇒ amy@juice-sh.op



2. The hint says that Amy is from a series called Futurama. It also tells that she has a husband named Kif.

3. Google the sentences "*93.83 billion trillion trillion centuries*" and "*One Important Final Note*". This will lead to a website called https://www.grc.com/haystack.htm?

4. The example provided by the website, "D0g.....", shows that the time required to solve is 93.83 billion trillion trillion centuries. This tells that Amy's password has the same pattern.

5. Try to replace the keyword "D0g" with "K1f" per hint.

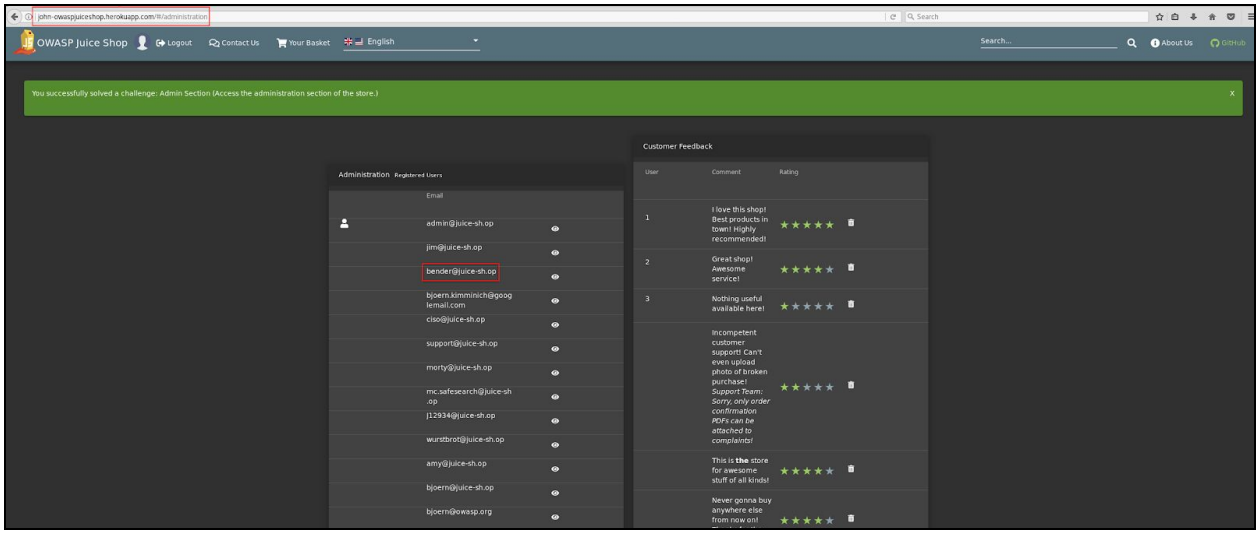
6. Go back to the JuiceShop's Login page and enter the following credential:

Email: amy@juice-sh.op
Password: K1f.....

Written by Jonathan Harijanto

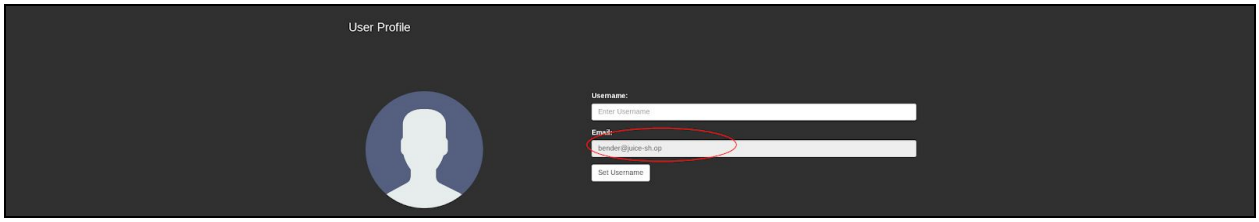
Login Bender - Log in with Bender's user account

1. Login as Admin and go to the Administration page (/administration). Find Bender's email address ⇒ bender@juice-sh.op



2. Go to the Login page and perform an SQL injection attack since there is no extra instruction provided.

Email:	bender@juice-sh.op' --
Password:	test123

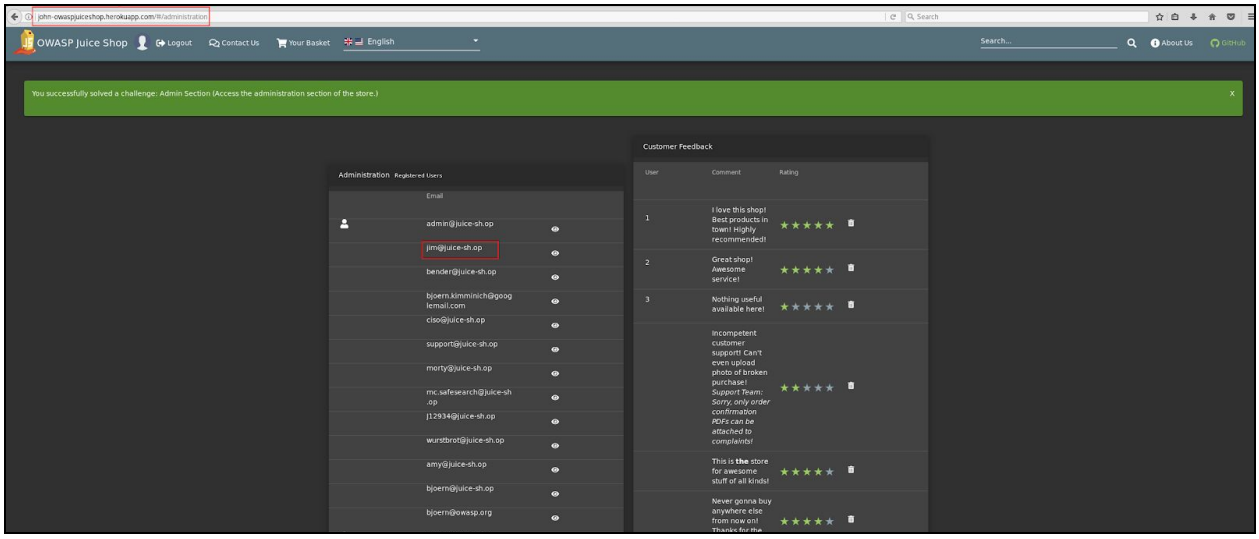


Note: Why this works? The value of bender@juice-sh.op is true because the email exists in the database, and the (' --) tells the SQL query to ignore the rest of the statement, including the password.

Written by Jonathan Harijanto

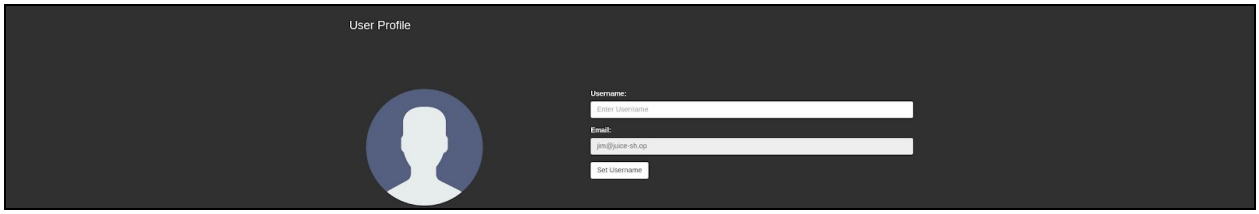
Login Jim - Log in with Jim's user account

1. Login as Admin and go to the Administration page (/administration). Find Jim's email address ⇒ jim@juice-sh.op



2. Go to the Login page and perform an SQL injection attack since there is no extra instruction provided.

Email: jim@juice-sh.op' --
Pass: test123



Note: Why this works? The value of jim@juice-sh.op is true because the email exists in the database, and the (' --) tells the SQL query to ignore the rest of the statement, including the password.

Written by Jonathan Harijanto

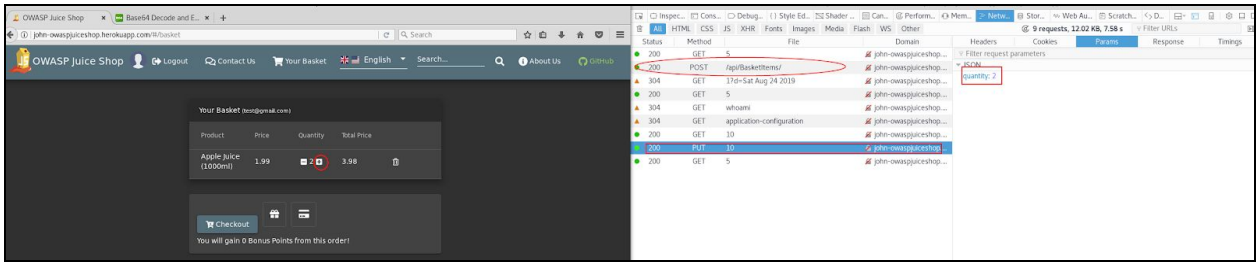
Payback Time - Place an order that makes you rich

1. Login with any user and open the Developer Tools → Network tab

2. Add a random item into the basket and observe the API call to /api/BasketItems/ via POST

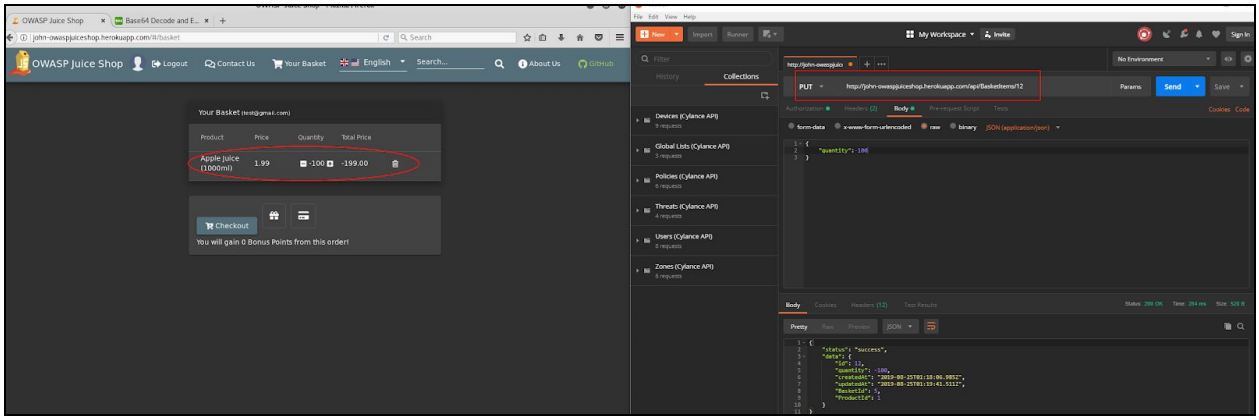
3. Memorize the value of the item's id from the API call to BasketItems. In my case, it's 12.

4. Go to the "Your Basket" page and increase the quantity of the item by one. Notice that there's a PUT request triggered and has "quantity: 2" as the parameter



5. The logic of this challenge is that we need to decrease the quantity of an item so that the store will pay us, instead of the other way around.

6. Open Postman and do a PUT request to /api/BasketItems/12, where 12 is the id of the items added to the basket earlier. In the Body, add a parameter of "quantity" with a value of negative.



7. Hit the send button to perform the call. The UI page should get updated and then click the "Checkout" button.

Written by Jonathan Harijanto

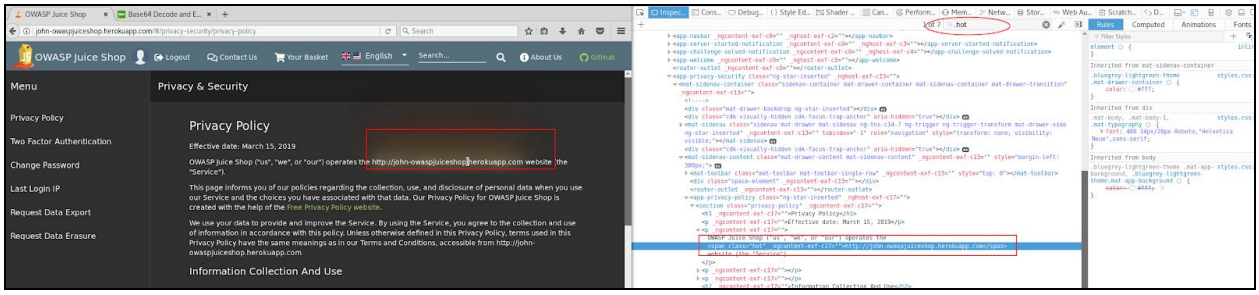
Privacy Policy Tier 2 - Prove that you actually read our privacy policy

1. Login with any user. Go to the “Privacy & Security” page

2. Read the content line by line while hovering your mouse. Eventually, you’ll find some of the words are “hot”

3. Do right-click on these words and select “Inspect”. Notice that these words are under a class called “hot”

4. Search for the other words under the same class “hot” and concatenate those words



5. Here’s the final result:

> http://john-owaspjuiceshop.herokuapp.com We may also instruct you to refuse all reasonably necessary responsibility

6. Convert these words into a URL:

> http://john-owaspjuiceshop.herokuapp.com/we/may/also/instruct/you/to/refuse/all/reasonably/necessary/responsibility

7. The page should’ve displayed an image called thank-you.jpg; however, it seems that the developer forgot to attach it

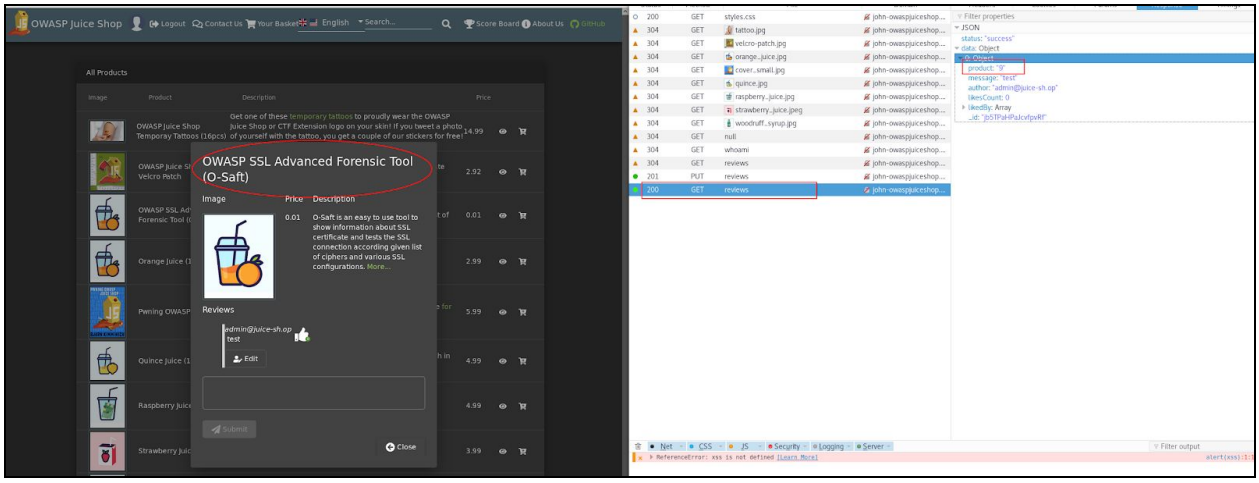
8. Return to JuiceShop homepage to complete the challenge

Written by Jonathan Harijanto

Product Tampering - Change the href of the link within the OWASP SSL Advanced Forensic Tool (O-Saft) product description into https://owasp.slack.com

1. Go to the store page and find the product with the name of O-Saft.

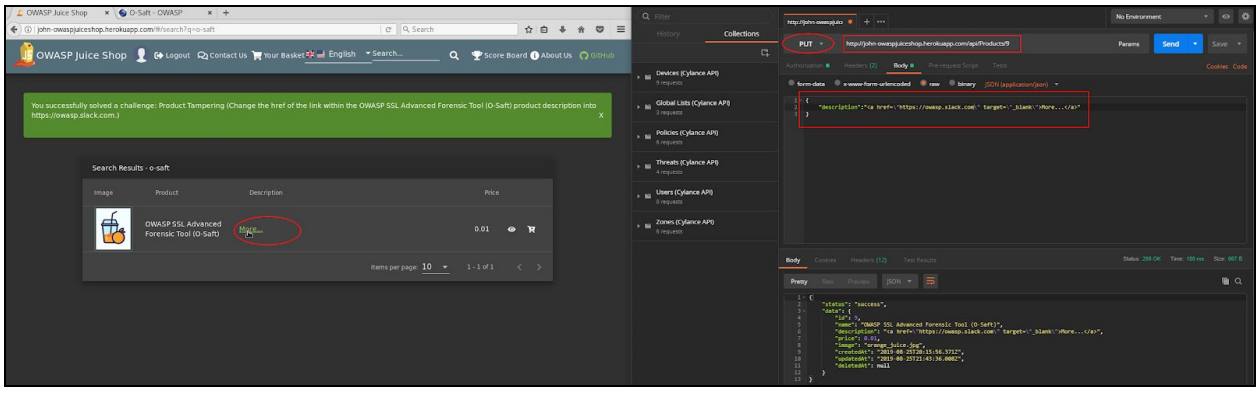
2. Open the Developer Tools → Network tab and post a random review on that item. Remember the productID (9)



3. Open Postman and perform an API call to http://john-owaspjuiceshop.herokuapp.com/api/Products/9 via PUT request

4. Change the content of the body into OWASP Slack's URL and hit Send

{"description":"More..."}

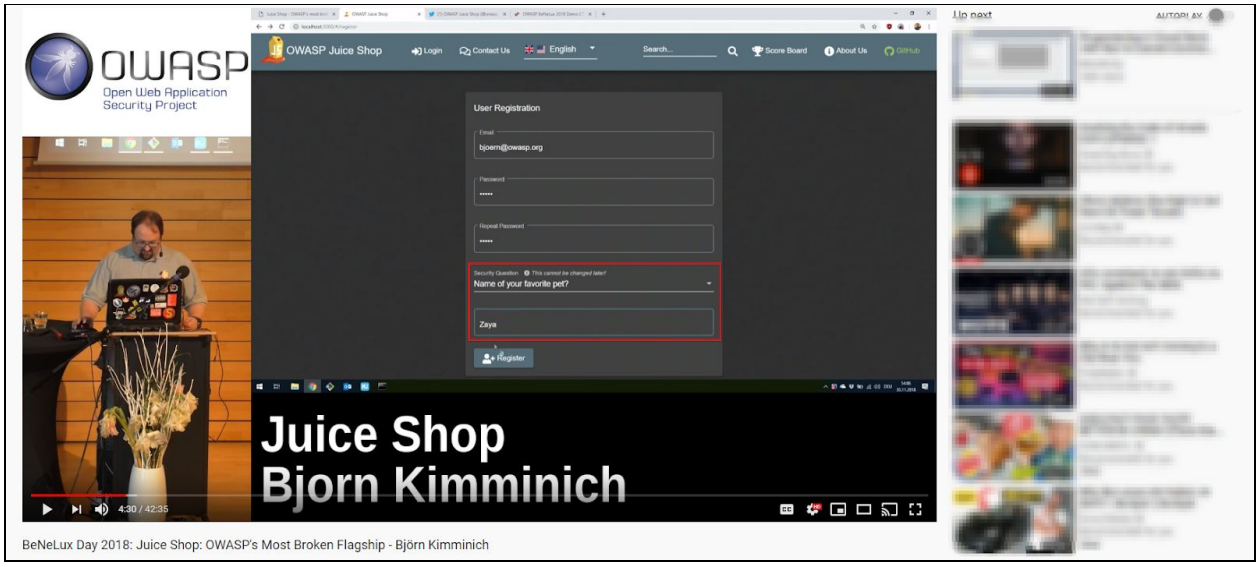


5. Refresh the page, you'll see the description will contain an embedded URL pointing to OWASP slack channel

Written by Jonathan Harijanto

Reset Bjoern's Password Tier 1 - Reset the password of Bjoern's OWASP account via the Forgot Password mechanism with the original answer to his security question

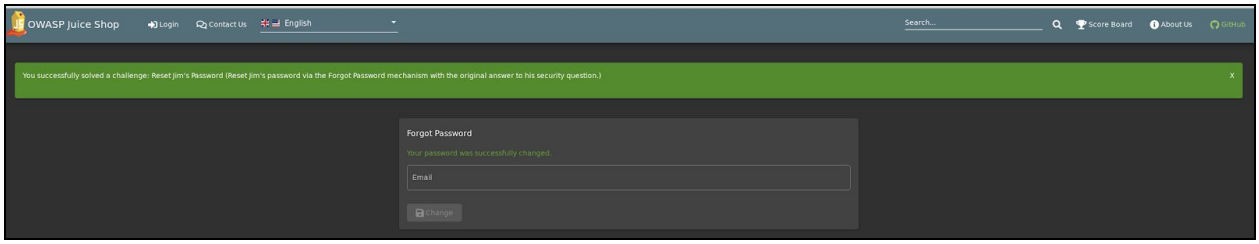
1. Go to Youtube and search for one of Bjorn's talks about JuiceShop. Here's the URL: https://www.youtube.com/watch?v=Lu0-kDdtVf4&feature=youtu.be&t=239



2. From the video above, Bjorn's clearly shows that his email is bjoern@owasp.org and his security's answer is Zaya

3. Go to JuiceShop's Login page and select the "Forgot Your Password" option.

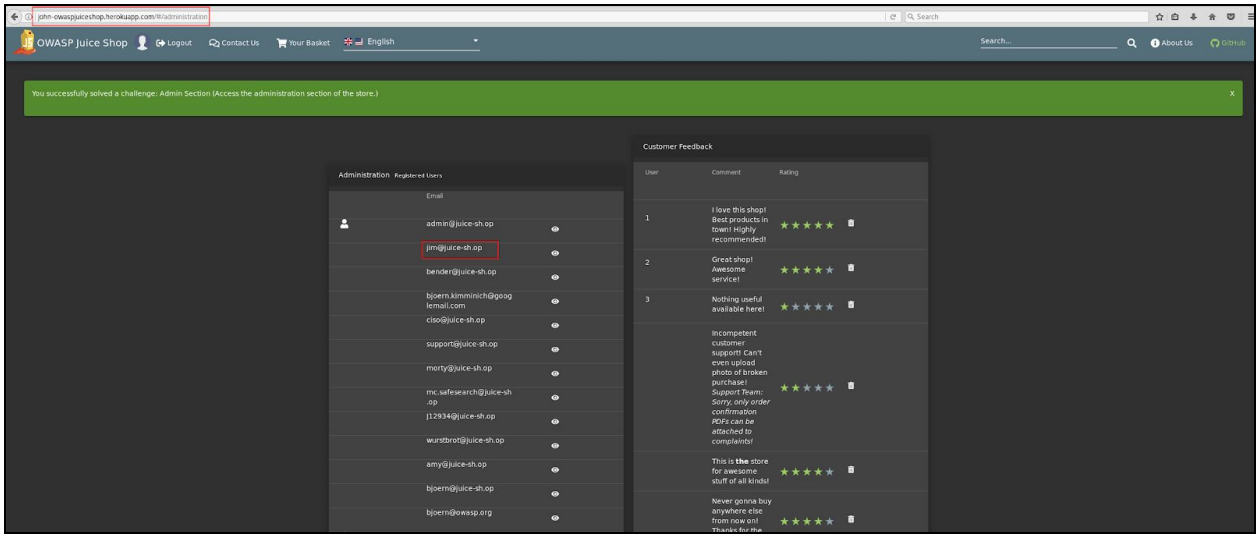
4. Answer the security question and change the password to anything.



Written by Jonathan Harijanto

Reset Jim's Password - Reset Jim's password via the Forgot Password mechanism with the original answer to his security question

1. Go to the Login page and login as an Admin. Look for Jim's email address ⇒ jim@juice-sh.op

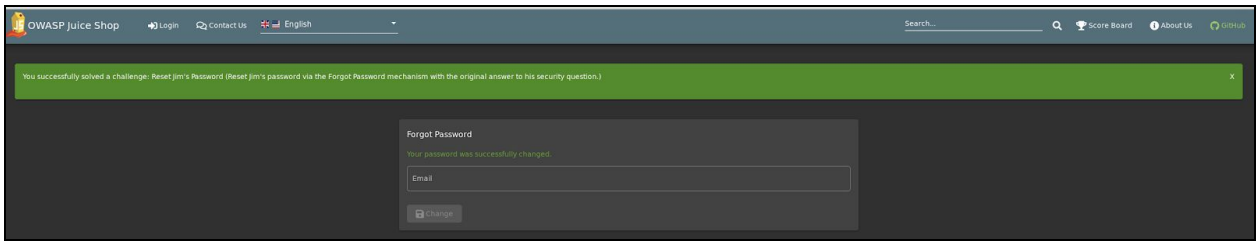


2. Logout and perform a forget password for Jim's email address. The security question for this user is "Your eldest siblings middle name?"

3. Go to the Store page and find a product called "OWASP Juice Shop-CTF Velcro Patch" where Jim commented "*Looks so much better on my uniform than the boring Starfleet symbol*".

4. Google the keyword "Starfleet" and you'll see that it refers to Star Trek. So, Jim == James T. Kirk. Next, Google the keywords "James T. Kirk Eldest Siblings". The answer will be George Samuel Kirk. So, the answer to the security question is Samuel.

5. Go to the Login page. Select "Forgot Password" option. Answer the security question and change the password.

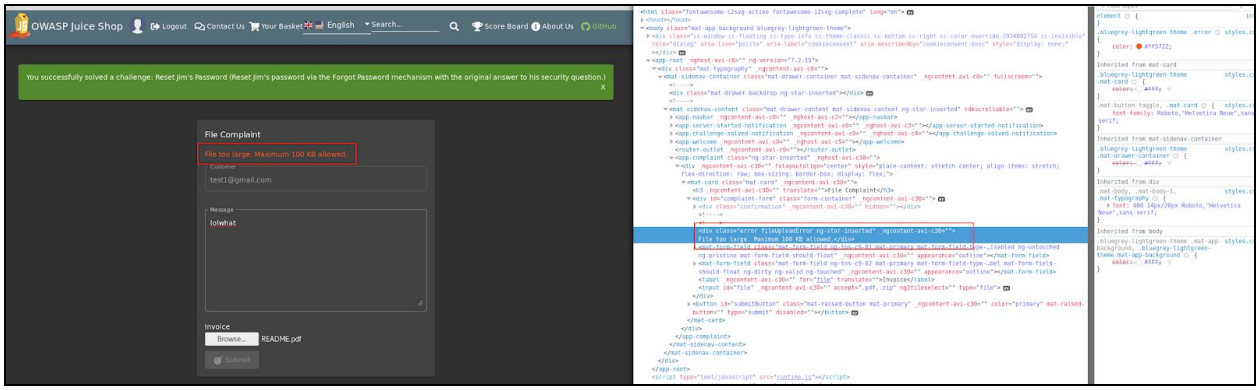


Written by Jonathan Harijanto

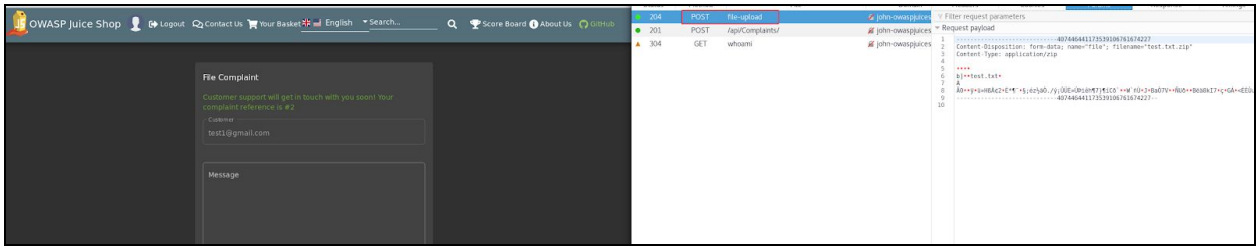
Upload Size - Upload a file larger than 100 kB

1. Login with any user and go to the “File Complaint” page because it’s the only page that allows anyone to upload a file.

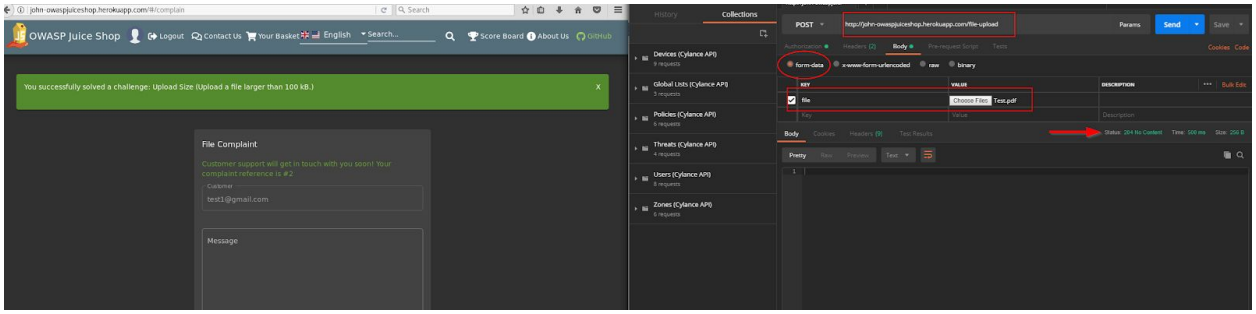
2. Try to upload a file that is larger than 100 KB will immediately get a warning



3. Open the Developer Tools → Network tab and try to upload a file that is less than 100 KB and in a zip format. The application will perform an API call to /file-upload



4. Open Postman and perform an API call to /file-upload via POST request. In Body, go to “form-data”and upload a zip file with a size between 101 KB to 199 KB.

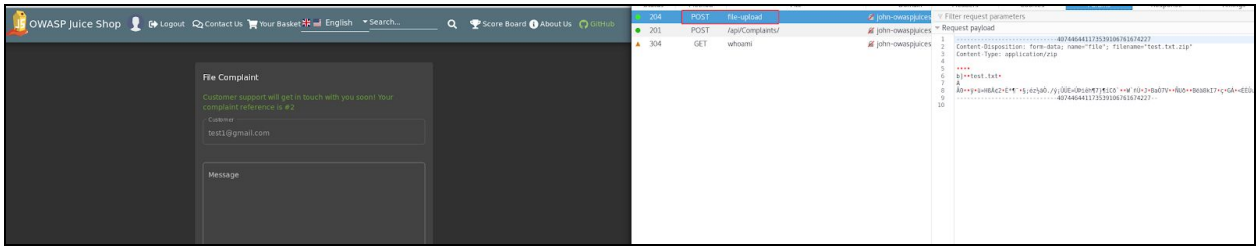


Written by Jonathan Harijanto

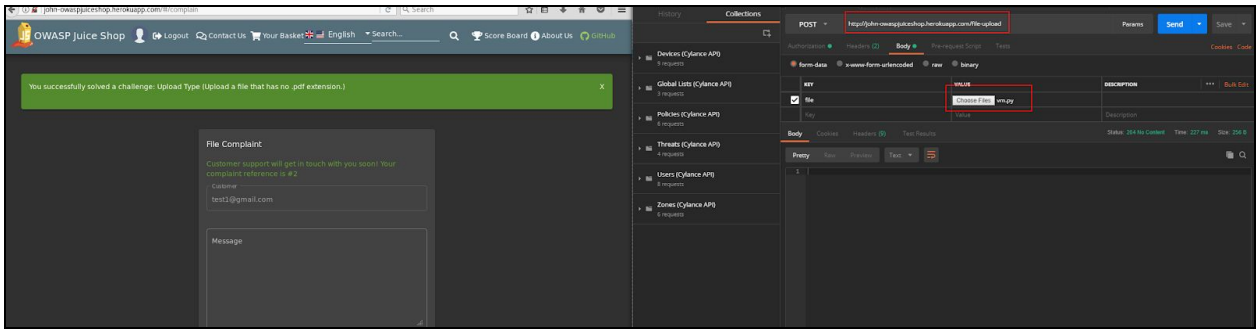
Upload Type - Upload a file that has no .pdf extension

1. Login with any user and go to the “File Complaint” page because it’s the only page that allows anyone to upload a file.

2. Open the Developer Tools → Network tab and try to upload a file that is less than 100 KB and in a zip format. The application will perform an API call to /file-upload



3. Open Postman and perform an API call to /file-upload via POST request. In Body, go to “form-data”and upload a file (not a pdf file) with a size between 101 KB to 199 KB.



Written by Jonathan Harijanto

XSS Tier 2 - Perform a persistent XSS attack with <iframe src="javascript:alert(`xss`)"> bypassing a client-side security mechanism

The definition of persistent XSS is: “it occurs when the data provided by the attacker is saved by the server, and then permanently displayed on "normal" pages returned to other users in the course of regular browsing, without proper HTML escaping”

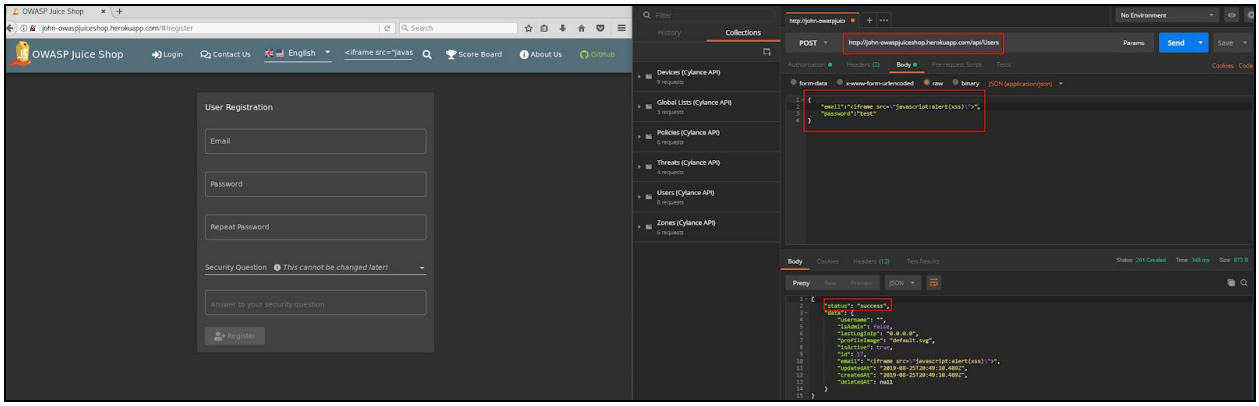
1. Go to the “Signup” page and open the Developer Tools → “Network” tab.

2. Try to register for a new user. Notice that the web application performs a POST request to /api/Users. The parameters used in that API call are: email, password, passwordRepeat, securityAnswer, securityQuestion

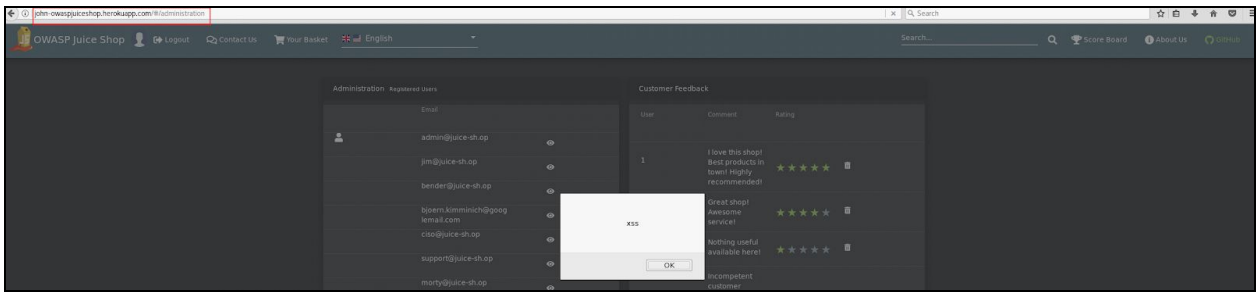
The hint says “Bypassing client-side security can typically be done by ignoring it completely and interacting with the backend instead”.

3. Open Postman and perform a call to /api/Users via POST request. Paste the following information in the Body section and hit Send:

“Email”==“<iframe src=\"javascript:alert(`xss`)\">" & "password"=="test123"



4. Go to the “Login” page. Login as an Administrator and go to the Administration page. The script will get triggered when the web application is trying to load all the users.



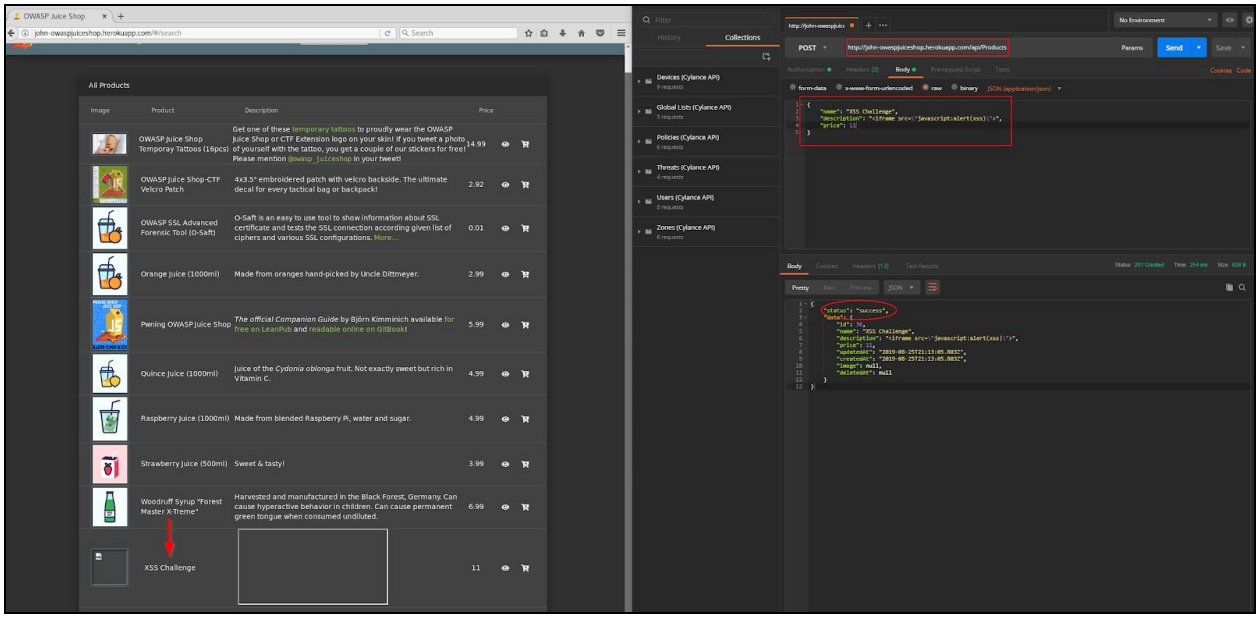
Written by Jonathan Harijanto

XSS Tier 3 - Perform a persistent XSS attack with <iframe src="javascript:alert(`xss`)"> without using the frontend application at all

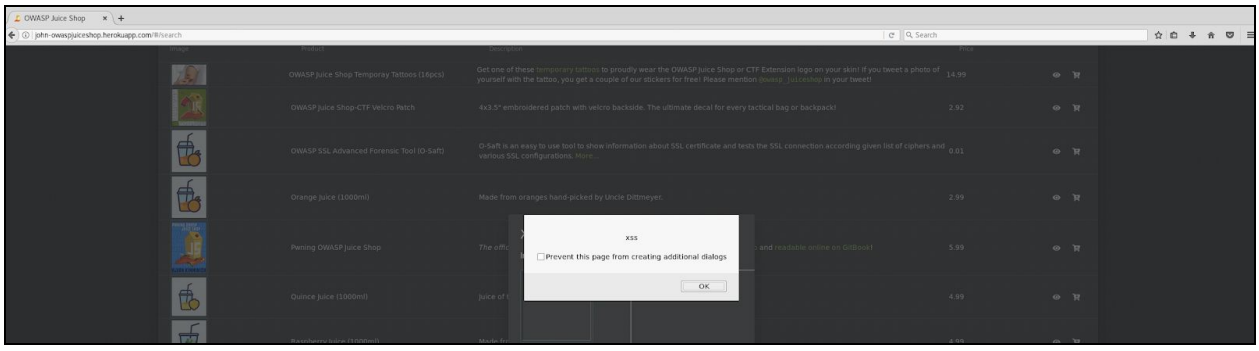
1. Open Postman and perform an API call to /api/Products (the API to list the products on the front page) via POST request.

2. Enter the following information as the parameters in the Body section:

    {"name": "XSS Challenge", "description": "<iframe src=\"javascript:alert(`xss`)\">", "price": 11}



3. We have successfully listed a new product on the store page. Find the product and click on the eye icon to trigger the script.



Written by Jonathan Harijanto