

Rapport dating site

Red vs Blue team event

Fontys ICT

28-10-2020

## **Chapter 1**

**Management Summary**

No interesting issues were found, this application is completely secured by default due to the choices made in the technologies en functionalities. The only issues were input issues that can lead to confusing outcomes but nothing that would break the site or lead to privilege escalation.

**Table of Contents**

[**Chapter 1**](#_p5zugf2e9v6t) **2**

[**Chapter 2**](#_9ebguro5h6st) **4**

[**2.1 Research question**](#_l5zynsppyp6s) **4**

[**2.2 Scope**](#_gm2gnfqavhfj) **4**

[2.2.1 Web Applications](#_u2jmujnsg316) 4

[**2.3 Extra conditions**](#_2kws0jxvesmh) **4**

[**Chapter 3**](#_o48glnspgqg6) **5**

[**3.1 Approach**](#_ubioixhhvus6) **5**

[3.1.1 General](#_m75qw8rytc2y) 5

[3.1.2 Web](#_kd6p7povjkss) 5

[**3.2 Analysis**](#_vf3fa0h49e3n) **5**

[3.2.1 Web](#_n2sqeu3dfvvb) 5

[Chapter 4](#_frgwujh712ou) 6

[**4.1 Conclusion**](#_cfnbf0p1jr4l) **6**

[**4.2 Advice**](#_b282lo8radkt) **6**

[**Appendix A**](#_x9pd8qhgsqbk) **7**

## **Chapter 2**

**Your request**

Test the data encryption, regular expressions/input filtering, Jwt token authorization and roles management/authentication.

# **2.1 Research question**

We’re going to test the application and any functionality that are available to us; we’re going to test for any cross-site scripting or injection, test the image upload for vulnerabilities, and test if the server will take any values that aren’t logical.

# **2.2 Scope**

The entire application was in the scope. The only thing not in the scope was a DOS attack and heavy dictionary attacks, since that could crash seclab.

## **2.2.1 Web Applications**

The entire web application(10.10.2.147) including the backend, hosted on seclab.

# **2.3 Extra conditions**

No.

## **Chapter 3**

Our Findings

A description of how every part of the scope is scanned, then a description of the results

# **3.1 Approach**

First, we scanned the application with nmap. We scanned for any ports open.

Then we tried out several things in the application; we tested any text boxes and input fields for cross site scripting and sql injection. We tested if you could enter non-numerical or negative values, or set dates that were in the future when they’re supposed to be in the past.

## **3.1.1 General**

In general, we looked at the website and tried out its functions. We made an account, uploaded images, and chatted with other members.

## **3.1.2 Web**

The tools used were mainly nmap and Burp Suite. We used this to check for malformed input, path traversal and bad authorization.

# **3.2 Analysis**

What were the results of the test, what are the problems and what was good? Again per defined chapter in the scope.

## **3.2.1 Web**

Nothing. All our attempts to upload a modified image failed and input filtering was applied to every field. There were no cookies or JWT tokens to modify.

The only noteworthy thing is that you were able to insert negative values into inputs that were supposed to have positive values in them. During account creation, you are also able to choose a birthdate in the future, which makes you -1 years old. You could also chat with a user that doesn’t exist by changing the user ID in the link. Lastly, you can sign up for a new account while you are logged in.

## 

## **Chapter 4**

**Conclusion and advice**

We weren’t able to find anything truly noteworthy. The application used a separate service to upload images and uses frameworks that makes hacking the website hard. There isn’t enough functionality in there to hack, which isn’t necessarily a bad thing.

# **4.1 Conclusion**

We only really found small input errors that can lead to confusing outcomes, but nothing that would break the site or give privilege escalation.

# **4.2 Advice**

They can fix these input errors by implementing more checks and making unit tests to ensure it doesn’t happen again.