Red Team 5
Security Assessment Findings Report Event 2

A online banking system
HTES AAS Factory4.0 project
Webportal for managing climate systems
Desktop chat application

Date: 14 December 2020
Project: Red Team Team 5
Version 2.0
Project members:

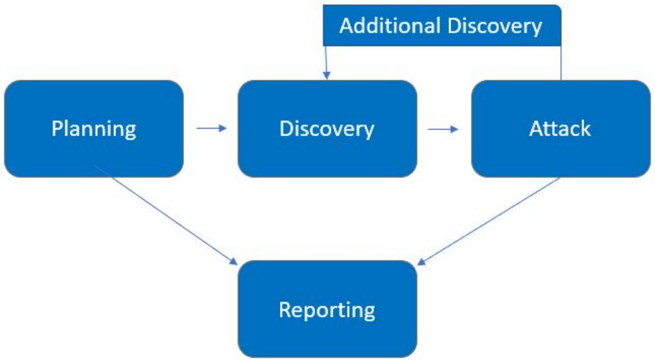
Demian Schouten
Freddy Gomes
Luca Gabriel
Kereem Coipel

Table of Contents

Contact

Name	Title	Contact Information
Freddy Gomes	Red Team specialization	f.gomes@student.fontys.nl
Demian Schouten	Red Team specialization	d.schouten@student.fontys.nl
Luca Gabriel	Red Team specialization	g.luca@student.fontys.nl
Kereem Coipel	Read Team specialization	k.coipel@student.fontys.nl

Assessment Overview & scope



During the pentest the 5 red team groups equally handed out projects to pentest. For our group team 5 we had to research the following 3 projects:

System owner	Application Description	Test goals (as seen by the 'client')	test accounts for all roles in the system(s) (for whitebox testing)	ip-address(es) to test	VLAN
Janssen,Bart B.F.J.W. <bart.janssen@student.fontys.nl>	Desktop Chat application	See in teams Documentation -> Test object clarification Bart Janssen Data encryption, authentication, token steal/intercept, MitM, 2FA, digital licensing, retrieving data from other users	See in 'teems red blue chanel -> Documentation -> Test object clarification Bart Janssen'	Server: 10.10.2.125 Client 1: 10.10.2.126 Client 2: 10.10.2.127	1473_Client2-VLAN

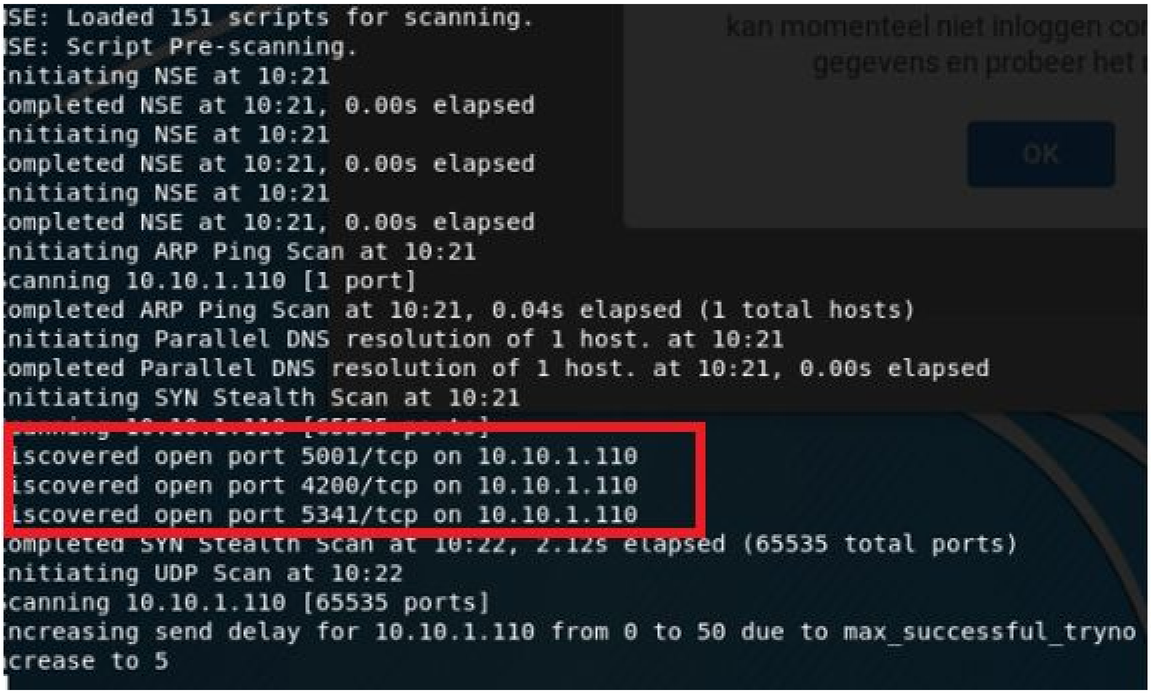
Bommel,Marc M. van <marc.vanbommel@student.fontys.nl>	a online banking system	test the data encryption and the input filtering, jwt authorization, captcha, Password rules,	user account: firstname: Red lastname: Team password: H@ckTh1s admin account: firstname: Red lastname: Admin password: N3wPages!	10.10.1.110:4200	1472_Client1-VLAN
Vliet, Geoffrey G.J. van <g.vanvliet@student.fontys.nl>	HTES AAS Factory4.0 project	See how far you can get ;). First try to hack some of our systems, on 145.220.75.127 (accessible without any VPN (but it is (strongly) recommended to use your own VPN cause of Seclab's outside firewall). If you give up, you could ask me privately questions (after 13.00 UTC+1), so that you can test our inside network devices behind the security.	Ask me private pls.	145.220.75.127	Vliet, Geoffrey G.J. van <g.vanvliet@student.fontys.nl>

Saman,Youri Y. <y.saman@student .fontys.nl>	Webportal for managing climate systems	input validation, authentication, authorization, recaptcha at regestration	Accounts can be registered in application, admin accounts are not implemented	10.10.2.140	1473_Client2 -VLAN

Findings

A online banking system

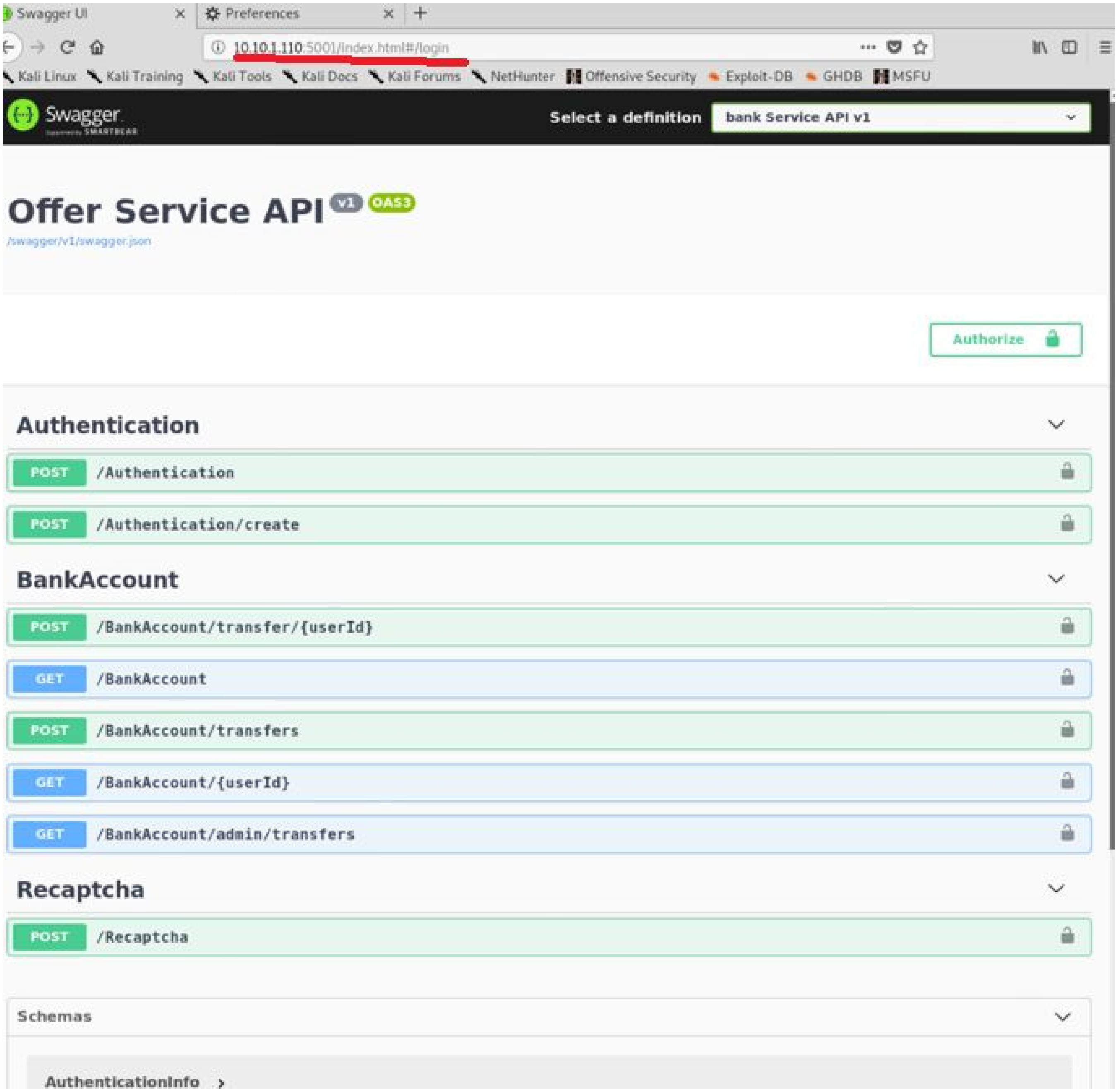
This application had a single fully working IP to connect to and that was 10.10.2.140. Using port 4200 it was obvious that the application ran an Angular project for anyone who had any experience with this platform (we would suggest changing the port and the tab icon).



```
SE: Loaded 151 scripts for scanning.
SE: Script Pre-scanning.
nitiating NSE at 10:21
ompleted NSE at 10:21, 0.00s elapsed
nitiating NSE at 10:21
ompleted NSE at 10:21, 0.00s elapsed
nitiating NSE at 10:21
ompleted NSE at 10:21, 0.00s elapsed
nitiating ARP Ping Scan at 10:21
canning 10.10.1.110 [1 port]
ompleted ARP Ping Scan at 10:21, 0.04s elapsed (1 total hosts)
nitiating Parallel DNS resolution of 1 host. at 10:21
ompleted Parallel DNS resolution of 1 host. at 10:21, 0.00s elapsed
nitiating SYN Stealth Scan at 10:21
canning 10.10.1.110 [65535 ports]
iscovered open port 5001/tcp on 10.10.1.110
iscovered open port 4200/tcp on 10.10.1.110
iscovered open port 5341/tcp on 10.10.1.110
ompleted SYN Stealth Scan at 10:22, 2.12s elapsed (65535 total ports)
nitiating UDP Scan at 10:22
canning 10.10.1.110 [65535 ports]
ncreasing send delay for 10.10.1.110 from 0 to 50 due to max_successful_tryno
crease to 5
```

[Figure 1.1 - ScanFindings]

On the first tcp scan some of the ports that stood out in addition to port 4200 were 5001 and 5341. We were able to access these 2 in spite of the fact that they contained developer knowledge about the API and a portal that was not listed for testing therefore not intended to the public.



10.10.1.110:5001/index.html#/login

bank Service API v1

Offer Service API v1 OAS3

/swagger/v1/swagger.json

Authentication

POST /Authentication

POST /Authentication/create

BankAccount

POST /BankAccount/transfer/{userId}

GET /BankAccount

POST /BankAccount/transfers

GET /BankAccount/{userId}

GET /BankAccount/admin/transfers

Recaptcha

POST /Recaptcha

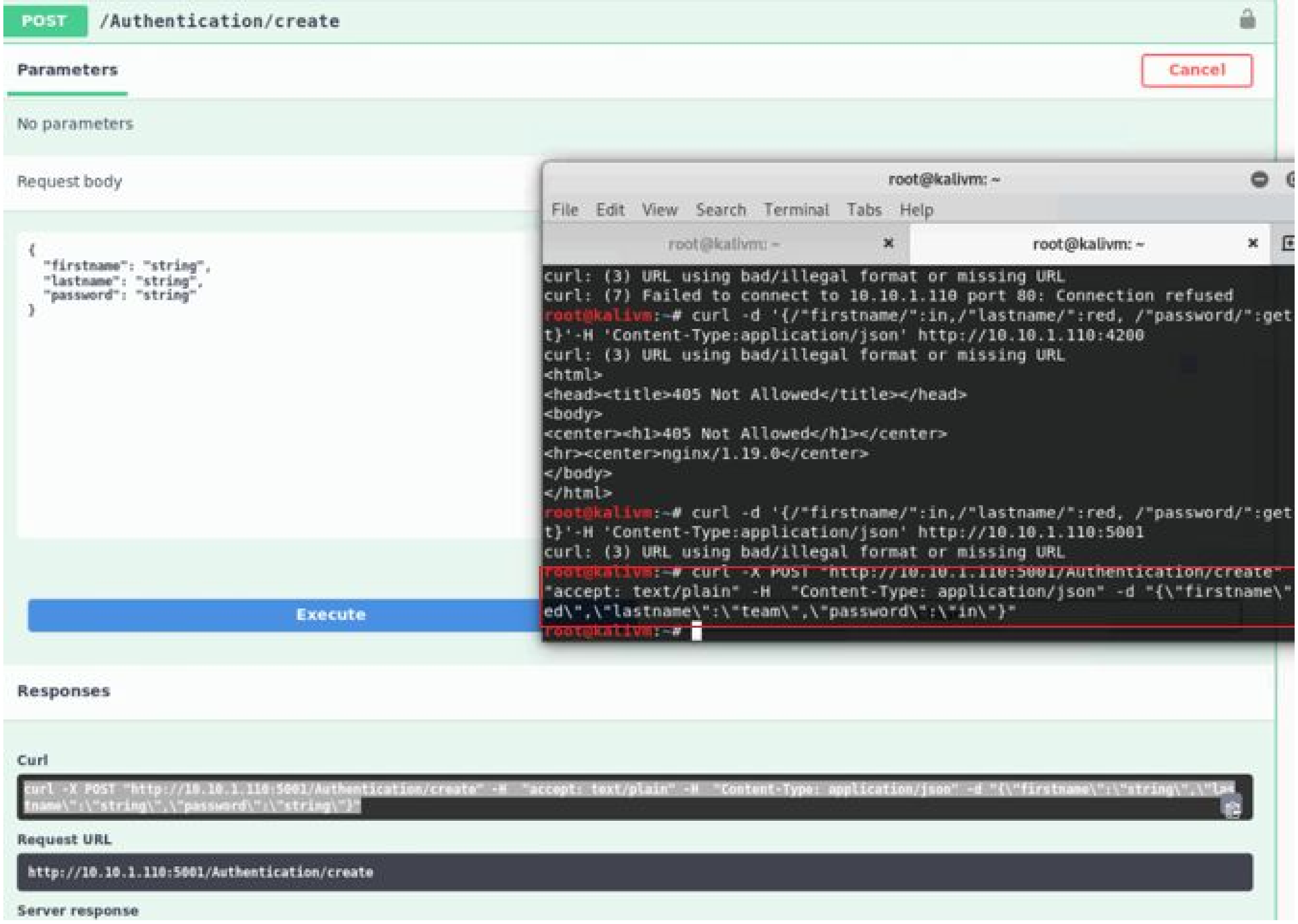
Schemas

AuthenticationInfo

[Figure 1.2 - Port 5001- Swagger API]

If we had the userId we could perform bankAccount transactions or if we wanted we could make ourselves new accounts. This in itself is a high-level vulnerability and we suggest finding a way to hide the ports from scans.

386998@student.fontys.nl



```
{
  "firstname": "string",
  "lastname": "string",
  "password": "string"
}
```

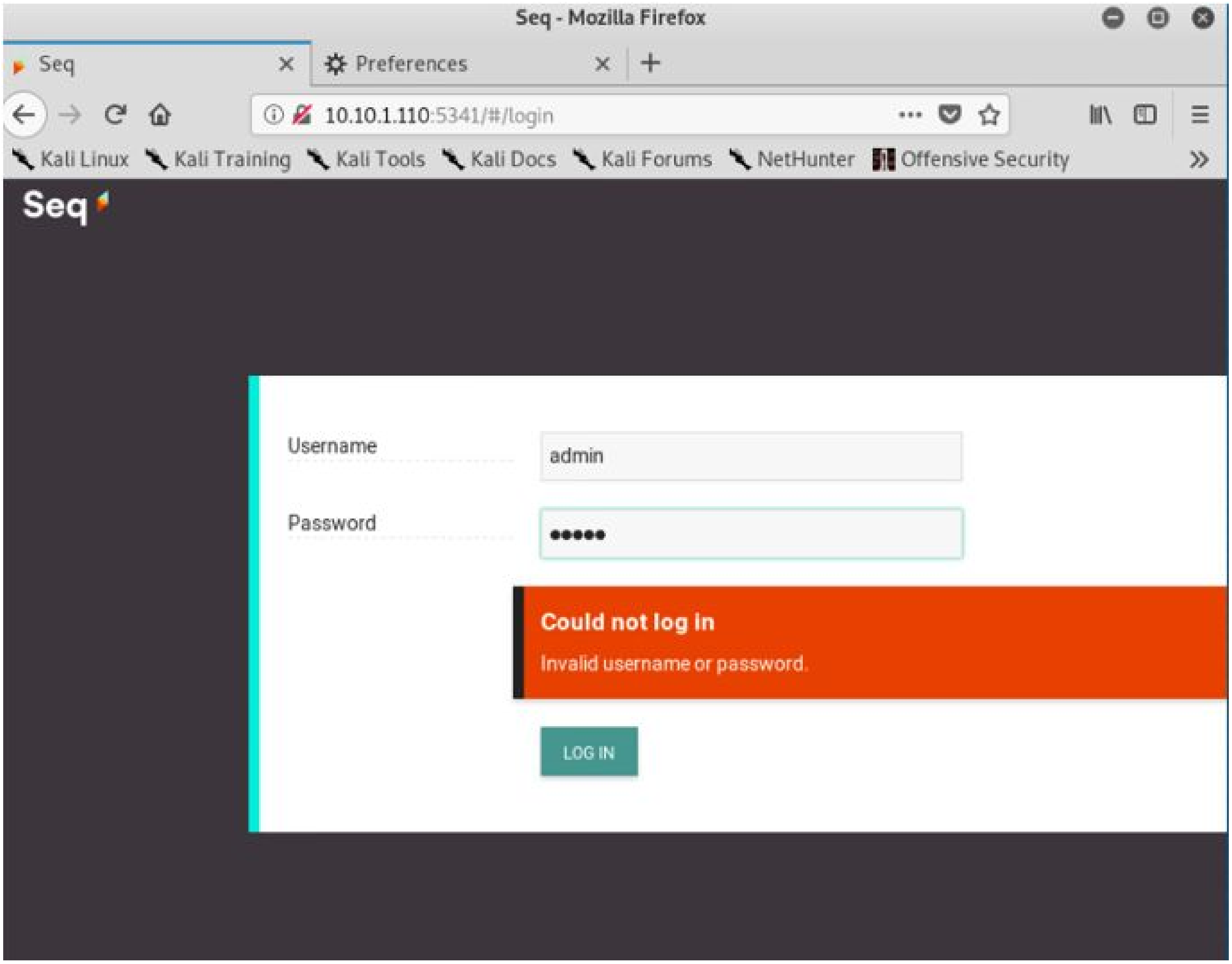
```
curl: (3) URL using bad/illegal format or missing URL
curl: (7) Failed to connect to 10.10.1.110 port 80: Connection refused
root@kalivm:~# curl -d '{/"firstname/":in,/"lastname/":red, /"password/":get
t}'-H 'Content-Type:application/json' http://10.10.1.110:4200
curl: (3) URL using bad/illegal format or missing URL
<html>
<head><title>405 Not Allowed</title></head>
<body>
<center><h1>405 Not Allowed</h1></center>
<hr><center>nginx/1.19.0</center>
</body>
</html>
root@kalivm:~# curl -d '{/"firstname/":in,/"lastname/":red, /"password/":get
t}'-H 'Content-Type:application/json' http://10.10.1.110:5001
curl: (3) URL using bad/illegal format or missing URL
root@kalivm:~# curl -X POST "http://10.10.1.110:5001/Authentication/create"
"accept: text/plain" -H  "Content-Type: application/json" -d "{\"firstname\"
ed\",\"lastname\":\"team\",\"password\":\"in\"}"
root@kalivm:~# 
```

```
curl -X POST "http://10.10.1.110:5001/Authentication/create" -H  "accept: text/plain" -H  "Content-Type: application/json" -d "{\"firstname\":\"string\",\"las
tname\":\"string\",\"password\":\"string\"}"
```

```
http://10.10.1.110:5001/Authentication/create
```

[Figure 1.3 - Swagger API testing the API endpoints]

To see if they are working we tried calling the API endpoint and created a new account / authentication details without needing a administrator account. As mentioned before this is a major vulnerability that is very easy to access.



Username admin

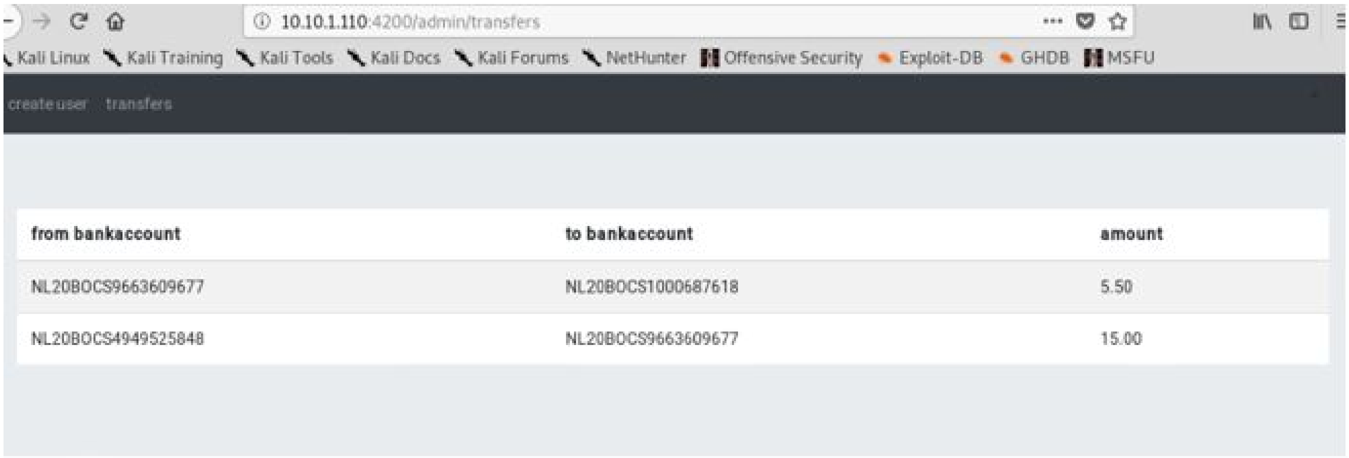
Password •••••

Could not log in

Invalid username or password.

[Figure 1.4 - Port 5341 - Seq]

Additionally, the other discovered port had a login page for what was believed to be a monitoring and alerting system of the application. The team tested it with bruteforcing, common passwords and sql injections and it did not open therefore it is well protected. But with a more extensive test period the login credentials could be obtained. We suggest hiding this port



from bankaccount	to bankaccount	amount
NL20BOCS9663609677	NL20BOCS1000687618	5.50
NL20BOCS4949525848	NL20BOCS9663609677	15.00

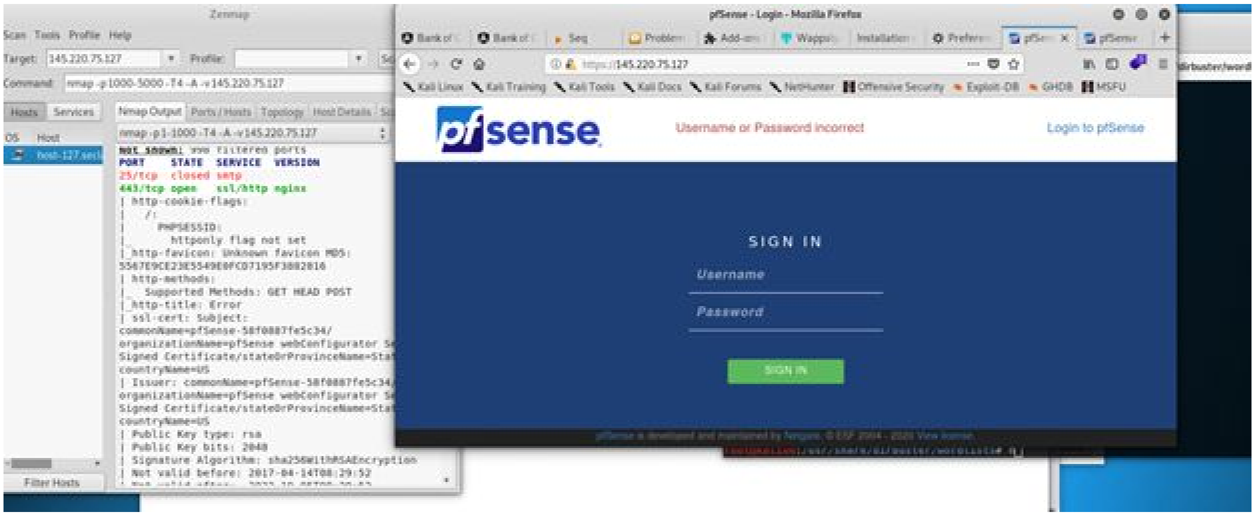
[Figure 1.5- Admin page]

While testing the admin page we realised that the webpage has no logout button, while this is not a vulnerability the team suggests the application owner invests in this functionality for better tests in the future.

In addition to this the create user functionality did not work or was on visible in the main/home page. Leaving us unsure if the functionality actually worked or not. Adding some kind of list of users in the admin page would help with this.

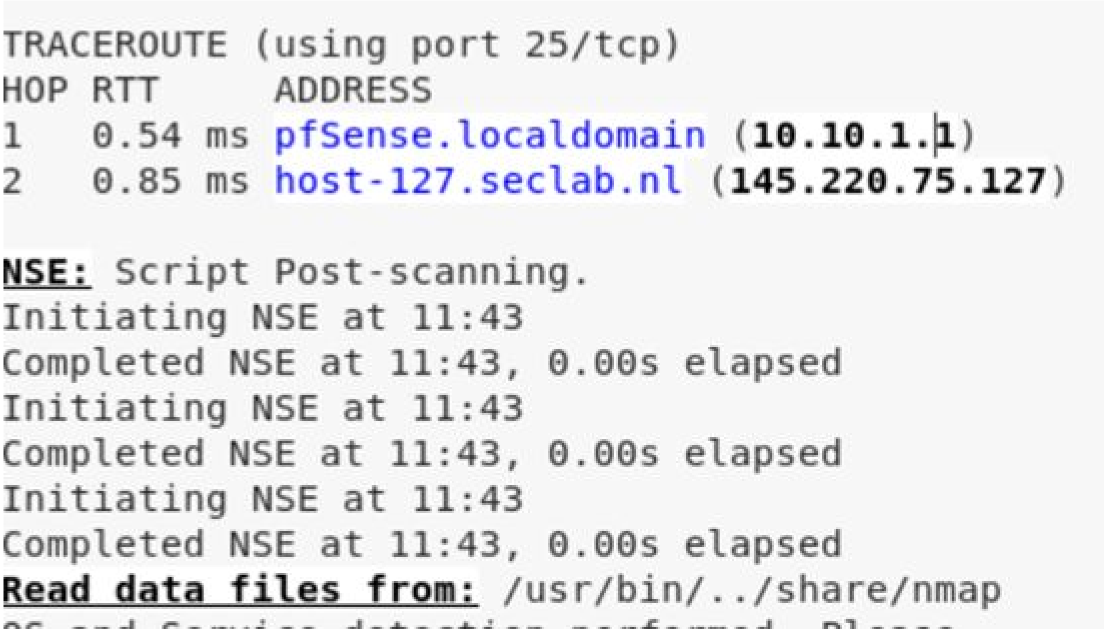
Also clicking or interacting with the recaptcha was not possible so we could not test it.

The overall suggestions would be to hide the swagger API page and the SEQ login page ports, change the port of the main application, add logout functionality and user lists in the admin page for testing purposes.



```
Not shown: 998 filtered ports
PORT    STATE  SERVICE  VERSION
25/tcp  closed smtp
443/tcp open   ssl/http nginx
| http-cookie-flags:
|   /:
|     PHPSESSID:
|       httponly flag not set
|_http-favicon: Unknown favicon MD5:
5567E9CE23E5549E0FCD7195F3882816
| http-methods:
|_  Supported Methods: GET HEAD POST
|_http-title: Error
| ssl-cert: Subject:
commonName=pfSense-58f0887fe5c34/
organizationName=pfSense webConfigurator Se
Signed Certificate/stateOrProvinceName=Sta
countryName=US
| Issuer: commonName=pfSense-58f0887fe5c34/
organizationName=pfSense webConfigurator Se
Signed Certificate/stateOrProvinceName=Sta
countryName=US
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2017-04-14T08:29:52
```

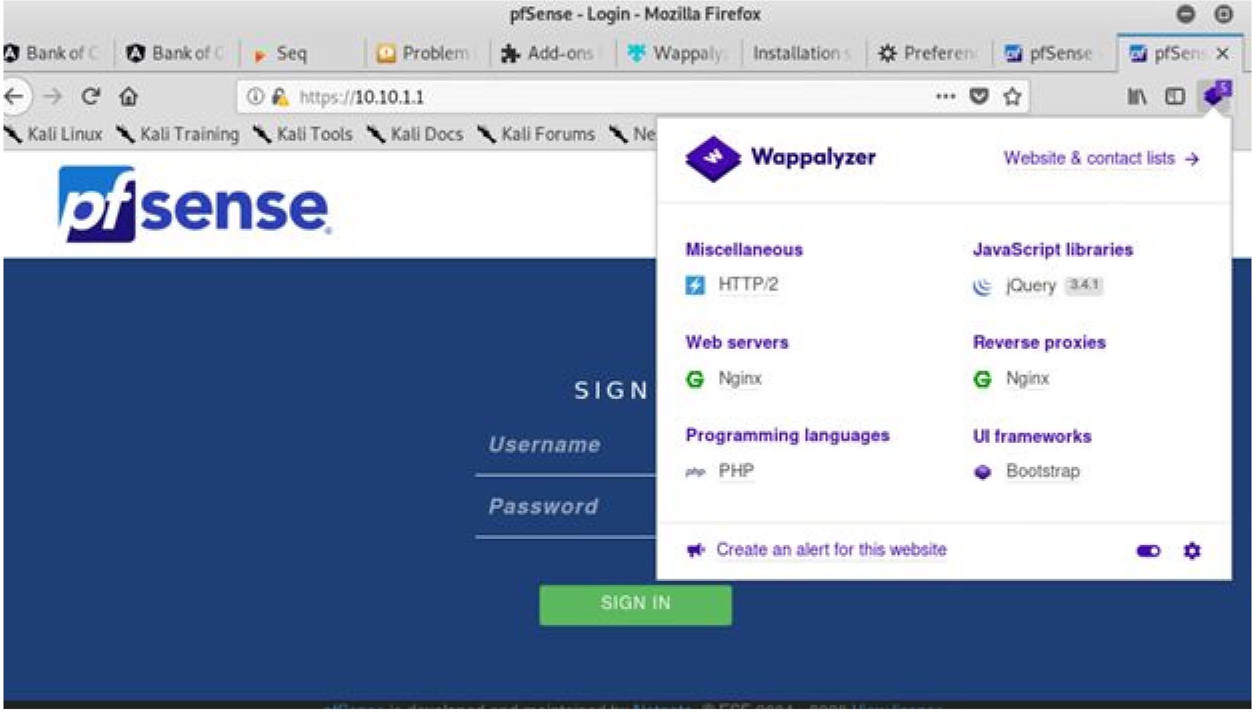
[Figure 2.1 - Testing the given IP]
The first thing found was a login page for pfSense, we assume this is the firewall used. getting inside was not possible with default credentials.



```
TRACEROUTE (using port 25/tcp)
HOP RTT     ADDRESS
1   0.54 ms pfSense.localdomain (10.10.1.1)
2   0.85 ms host-127.seclab.nl (145.220.75.127)

NSE: Script Post-scanning.
Initiating NSE at 11:43
Completed NSE at 11:43, 0.00s elapsed
Initiating NSE at 11:43
Completed NSE at 11:43, 0.00s elapsed
Initiating NSE at 11:43
Completed NSE at 11:43, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please
```

[Figure 2.2 - Sanning for additional ports or informaiton]
After scanning the given IP we were able to discover that the firewall was placed in the test environment in seclab. This is not a vulnerability but we did investigate further.



Miscellaneous
HTTP/2

JavaScript libraries
jQuery 3.4.1

Web servers
Nginx

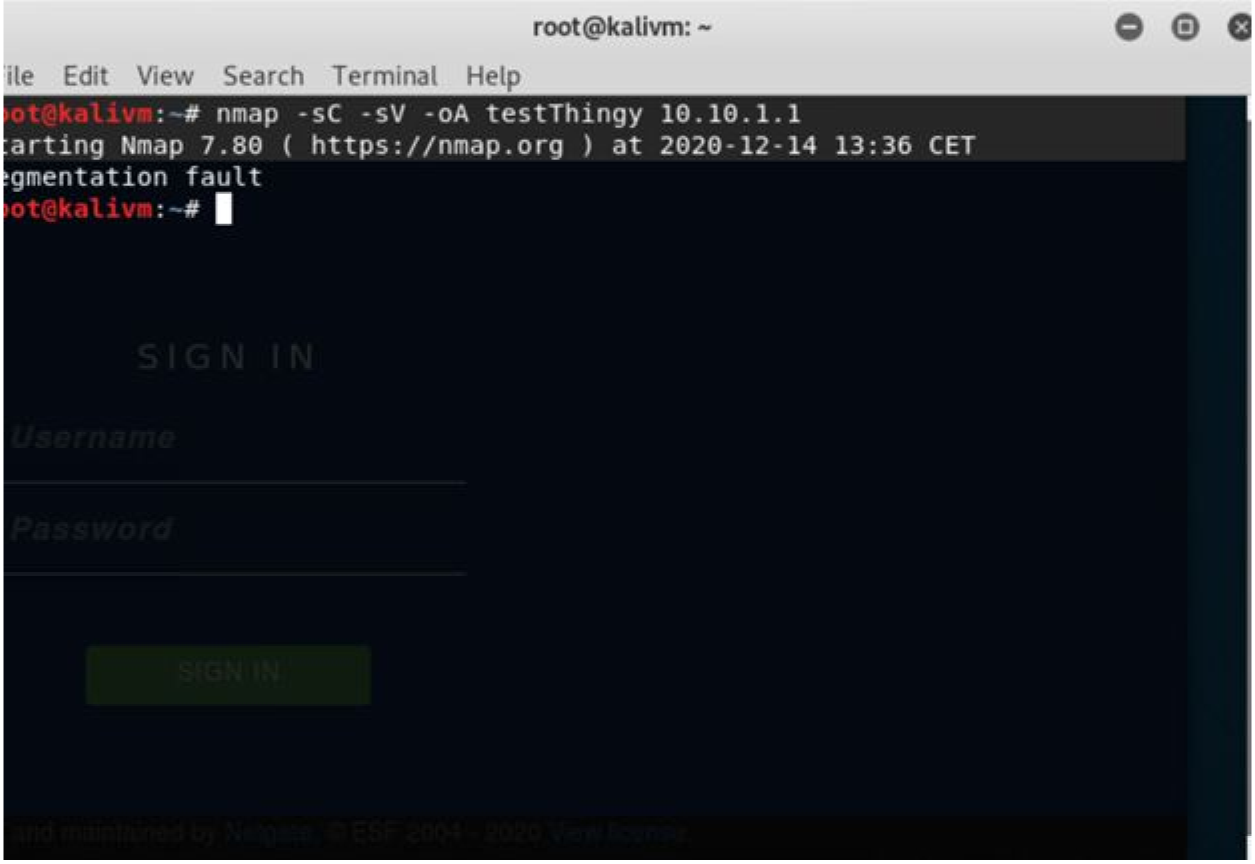
Reverse proxies
Nginx

Programming languages
PHP

UI frameworks
Bootstrap

[Figure 2.3 - Inspecting the seclab ip]

Upon inspection some general information was found from inspecting the seclab ip nothing major but we could see the server type and programming language of the firewall.



```
ot@kalivm:~# nmap -sC -sV -oA testThingy 10.10.1.1
arting Nmap 7.80 ( https://nmap.org ) at 2020-12-14 13:36 CET
gmentation fault
ot@kalivm:~# 
```

[Figure 4.0 - Scanning the internal firewall]

Security Strengths

- Use well known supported frameworks such as Angular and SpringBoot with standard security measures built in (In case of Angular anti XSS protection).

Security Weaknesses

- Missing some multi-factor authentication, that can make hackers life’s harder while trying to perform brute force attacks and man-in-the-middle attacks.

- Keep software up-to-date to prevent malicious users from using exploits targeting already known vulnerabilities.

Conclusion

During this day we had trouble setting up the pentesting environments since the pentesting machines provided by Fontys were outdated and we had to solve connectivity issues because of the load on the seclab.

Overall there is not much functionality to test which is unfortunate for us. In general it allowed us to practice our skills acquired during our studies once more. We hope that for a next event like this, we can come even more prepared and that there are no problems with the systems to be tested.