



Fleming College

Course	COMP3: Advanced Pentesting
Lab Assignment	THE MEGA HACKING MEGA GROUP PROJECT OF DEATH
Instructor	Adam Abernethy
Student Name	Manuel Manrique Lopez Ricardo Y. Rubin Uriostegui
Submission Date	10/December/2025

THE MEGA HACKING MEGA GROUP PROJECT OF DEATH

Juice Shop

Network diagram

2 vms

The image shows two terminal windows side-by-side. The left window is titled 'Session Actions Edit View Help' and shows the output of the 'ip a' command on a host named 'ricardor'. It lists two interfaces: 'lo' (loopback) and 'eth0' (ethernet). The 'lo' interface has an IP of 127.0.0.1/8. The 'eth0' interface has an IP of 192.168.194.146/24 and is connected to a bridge 'brd' with an IP of 192.168.194.255/24. The right window is titled 'ricardor@ricardor-virtual-machine:' and shows the output of 'ping' commands between the two VMs. It shows successful pings from 192.168.194.146 to 192.168.194.157 and vice versa.

```
ricardor@ricardor:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
            inet6 ::1/128 scope host noprefixroute
                valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:c9:a0:6e brd ff:ff:ff:ff:ff:ff
        inet 192.168.194.146/24 brd 192.168.194.255 scope global dynamic noprefixroute eth0
            valid_lft 1714sec preferred_lft 1714sec
            inet6 fe80::20c:29ff:fed9:a06e/64 scope link noprefixroute
                valid_lft forever preferred_lft forever
(ricardor@ricardor)-[~]
$ ping 192.168.194.146
PING 192.168.194.146 (192.168.194.146) 56(84) bytes of data.
64 bytes from 192.168.194.146: icmp_seq=1 ttl=64 time=1.28 ms
64 bytes from 192.168.194.146: icmp_seq=2 ttl=64 time=0.934 ms
64 bytes from 192.168.194.146: icmp_seq=3 ttl=64 time=2.74 ms
64 bytes from 192.168.194.146: icmp_seq=4 ttl=64 time=2.43 ms
64 bytes from 192.168.194.146: icmp_seq=5 ttl=64 time=3.38 ms
^C
--- 192.168.194.146 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 0.934/2.152/3.379/0.912 ms
(ricardor@ricardor)-[~]
$ ping 192.168.194.157
PING 192.168.194.157 (192.168.194.157) 56(84) bytes of data.
64 bytes from 192.168.194.157: icmp_seq=1 ttl=64 time=2.39 ms
64 bytes from 192.168.194.157: icmp_seq=2 ttl=64 time=2.16 ms
64 bytes from 192.168.194.157: icmp_seq=3 ttl=64 time=25.2 ms
64 bytes from 192.168.194.157: icmp_seq=4 ttl=64 time=3.26 ms
64 bytes from 192.168.194.157: icmp_seq=5 ttl=64 time=1.42 ms
64 bytes from 192.168.194.157: icmp_seq=6 ttl=64 time=1.18 ms
^C
--- 192.168.194.157 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 1.180/5.927/25.154/8.624 ms
ricardor@ricardor-virtual-machine:~$
```

Connection between both vms

vm segment: 192.168.194.0/24 (NAT)

Role	VM Name	IP Address	Relevant Port	Network Adapter
Attacker	Ubuntu	192.168.194.146	8080	NAT
Victim	Kali Linux	192.168.194.157	3000	NAT

VM Specifications

VM Ubuntu – Attacker

IP Ubuntu: 192.168.194.146

Memory: 8.9GB

Network Adapter: NAT

BurpSuite proxy: 8080

VM Kali - Victim

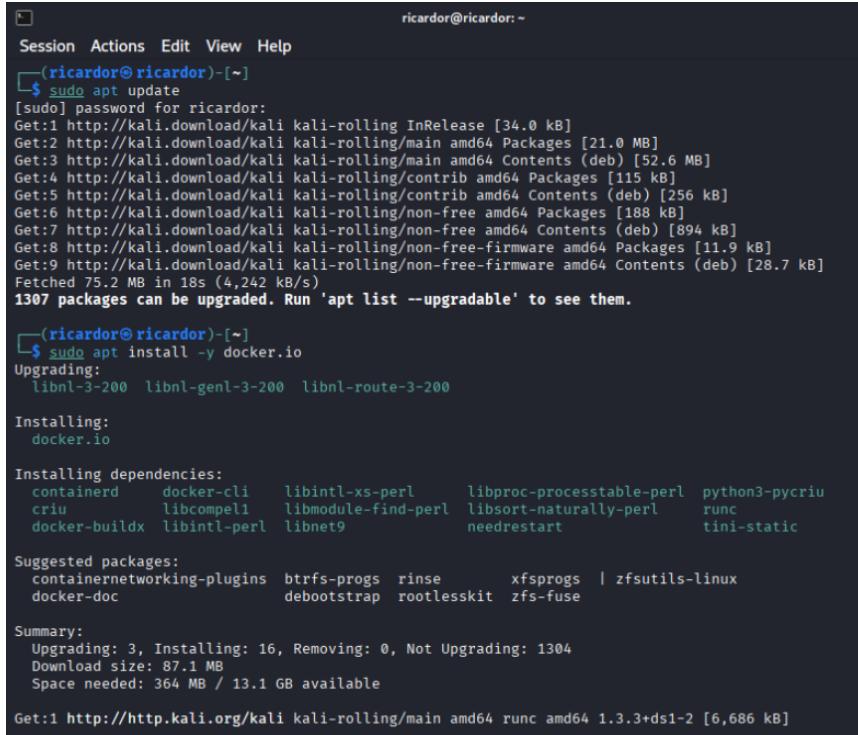
IP Kali: 192.168.194.157

Memory: 8.9GB

Network Adapter: NAT

Juice shop: 3000

Installing docker (old version but works for the lab)



```
ricardor@ricardor: ~
Session Actions Edit View Help
[ricardor@ricardor)-[~]
$ sudo apt update
[sudo] password for ricardor:
Get:1 http://kali.download/kali kali-rolling InRelease [34.0 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [21.0 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [52.6 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [115 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [256 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [188 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [894 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [11.9 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [28.7 kB]
Fetched 75.2 MB in 18s (4,242 kB/s)
1307 packages can be upgraded. Run 'apt list --upgradable' to see them.

[ricardor@ricardor)-[~]
$ sudo apt install -y docker.io
Upgrading:
 libnl-3-200  libnl-genl-3-200  libnl-route-3-200

Installing:
 docker.io

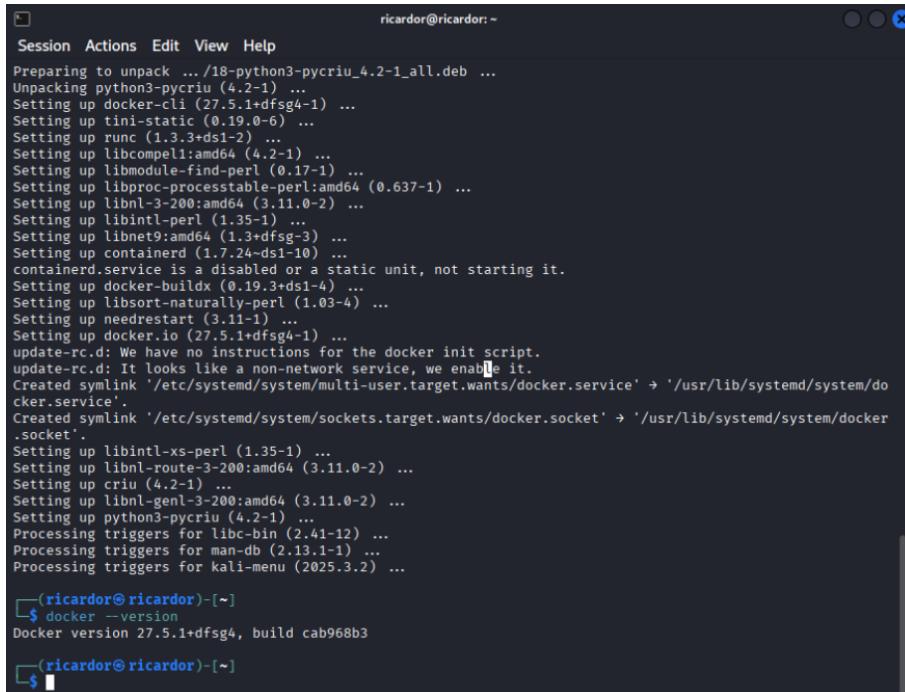
Installing dependencies:
 containerd  docker-cli  libintlx5-perl  libproc-processtable-perl  python3-pycrui
 criu  libcomplibl  libmodule-find-perl  libsort-naturally-perl  runc
 docker-buildx  libintlx5-perl  libnet9  needrestart  tini-static

Suggested packages:
 containernetworking-plugins  btrfs-progs  rinse  xfsprogs | zfsutils-linux
 docker-doc  debootstrap  rootlesskit  zfs-fuse

Summary:
 Upgrading: 3, Installing: 16, Removing: 0, Not Upgrading: 1304
 Download size: 87.1 MB
 Space needed: 364 MB / 13.1 GB available

Get:1 http://http.kali.org/kali kali-rolling/main amd64 runc amd64 1.3.3+ds1-2 [6,686 kB]
```

Figure 1 - Installing docker.



```
ricardor@ricardor: ~
Session Actions Edit View Help
Preparing to unpack .../18-python3-pycrui_4.2-1_all.deb ...
Unpacking python3-pycrui (4.2-1) ...
Setting up docker-cli (27.5.1+dfsg4-1) ...
Setting up tini-static (0.19.0-6) ...
Setting up runc (1.3.3+ds1-2) ...
Setting up libcomplibl:amd64 (4.2-1) ...
Setting up libmodule-find-perl (0.17-1) ...
Setting up libproc-processtable-perl:amd64 (0.637-1) ...
Setting up libnl-3-200:amd64 (3.11.0-2) ...
Setting up libintlx5-perl (1.35-1) ...
Setting up libnet9:amd64 (1.3+dfsg-3) ...
Setting up containerd (1.7.24~ds1-10) ...
containerd.service is a disabled or a static unit, not starting it.
Setting up docker-buildx (0.19.3+ds1-4) ...
Setting up libsort-naturally-perl (1.03-4) ...
Setting up needrestart (3.11-1) ...
Setting up docker.io (27.5.1+dfsg4-1) ...
update-rc.d: We have no instructions for the docker init script.
update-rc.d: It looks like a non-network service, we enable it.
Created symlink '/etc/systemd/system/multi-user.target.wants/docker.service' → '/usr/lib/systemd/system/docker.service'.
Created symlink '/etc/systemd/system/sockets.target.wants/docker.socket' → '/usr/lib/systemd/system/docker.socket'.
Setting up libintlx5-perl (1.35-1) ...
Setting up libnl-route-3-200:amd64 (3.11.0-2) ...
Setting up criu (4.2-1) ...
Setting up libnl-genl-3-200:amd64 (3.11.0-2) ...
Setting up python3-pycrui (4.2-1) ...
Processing triggers for libc-bin (2.41-12) ...
Processing triggers for man-db (2.13.1-1) ...
Processing triggers for kali-menu (2025.3.2) ...

[ricardor@ricardor)-[~]
$ docker --version
Docker version 27.5.1+dfsg4, build cab968b3

[ricardor@ricardor)-[~]
$
```

Figure 2 - Confirming dockers was successfully installed.

```
ricardor@ricardor:~  
Session Actions Edit View Help  
└─(ricardor@ricardor)─[~]  
└─$ docker --version  
Docker version 27.5.1+dfsg4, build cab968b3  
└─(ricardor@ricardor)─[~]  
└─$ sudo systemctl enable docker  
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable docker  
└─(ricardor@ricardor)─[~]  
└─$ sudo systemctl status docker  
● docker.service - Docker Application Container Engine  
  Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)  
  Active: active (running) since Sun 2025-12-07 13:47:56 EST; 5min ago  
    Invocation: 104b79420cf142f1b9e8c10a6f13c049  
  TriggeredBy: ● docker.socket  
    Docs: https://docs.docker.com  
    Main PID: 35507 (dockerd)  
      Tasks: 12  
        Memory: 26.5M (peak: 28.6M)  
          CPU: 2.141s  
        CGroup: /system.slice/docker.service  
          └─35507 /usr/sbin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock  
Dec 07 13:47:53 ricardor systemd[1]: Starting docker.service - Docker Application Container Engine ...  
Dec 07 13:47:53 ricardor (dockerd)[35507]: docker.service: Referenced but unset environment variable eval>  
Dec 07 13:47:53 ricardor dockerd[35507]: time="2025-12-07T13:47:53.592741961-05:00" level=info msg="Start">  
Dec 07 13:47:53 ricardor dockerd[35507]: time="2025-12-07T13:47:53.598039915-05:00" level=info msg="OTEL ">  
Dec 07 13:47:53 ricardor dockerd[35507]: time="2025-12-07T13:47:53.921612570-05:00" level=info msg="Loadi">  
Dec 07 13:47:56 ricardor dockerd[35507]: time="2025-12-07T13:47:56.595932935-05:00" level=info msg="Loadi">  
Dec 07 13:47:56 ricardor dockerd[35507]: time="2025-12-07T13:47:56.675191192-05:00" level=info msg="Docke">  
Dec 07 13:47:56 ricardor dockerd[35507]: time="2025-12-07T13:47:56.675413180-05:00" level=info msg="Daemon">  
Dec 07 13:47:56 ricardor dockerd[35507]: time="2025-12-07T13:47:56.792356907-05:00" level=info msg="API l">  
Dec 07 13:47:56 ricardor systemd[1]: Started docker.service - Docker Application Container Engine.  
└─(ricardor@ricardor)─[~]  
└─$
```

Figure 3 - Starting docker and confirming the status is active.

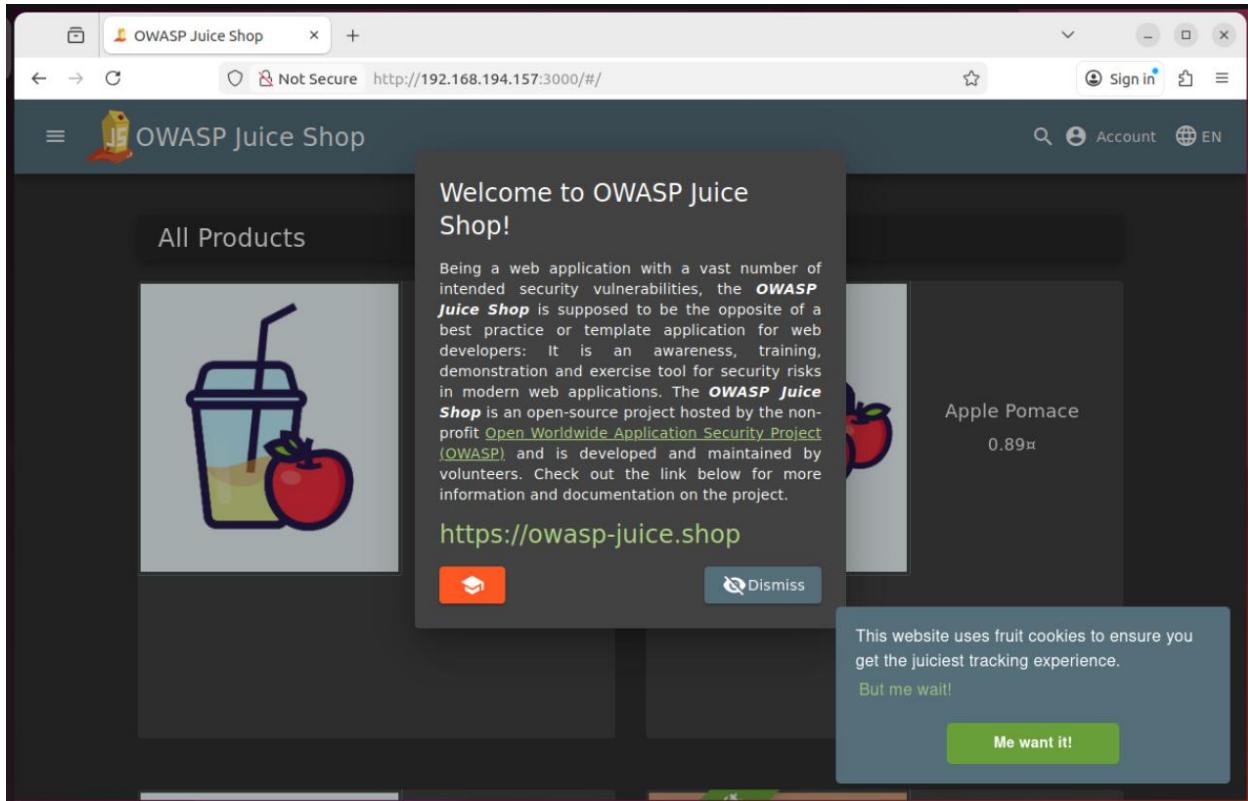
```
ricardor@ricardor:~  
Session Actions Edit View Help  
└─(ricardor@ricardor)─[~]  
└─$ sudo docker pull bkimminich/juice-shop  
Using default tag: latest  
latest: Pulling from bkimminich/juice-shop  
fd4aa3667332: Pull complete  
fbf59b82a9b6: Pull complete  
017886f7e176: Pull complete  
62de241dac5f: Pull complete  
2780920e5dbf: Pull complete  
7c12895b777b: Pull complete  
3214acf345c0: Pull complete  
5664b15f108b: Pull complete  
045fc1c20da8: Pull complete  
4aa0ea1413d3: Pull complete  
da7816fa955e: Pull complete  
ddf74a63f7d8: Pull complete  
e7fa9df358f0: Pull complete  
d8a0d911b13e: Pull complete  
5b14f6c9a813: Pull complete  
33ce0b1d99fc: Pull complete  
f45e0372ce60: Pull complete  
7faf0cfa885c: Pull complete  
9cd2a1476fcc: Pull complete  
7b72e6384ef9: Pull complete  
0168f69dfb16: Pull complete  
Digest: sha256:1c55debeaf4fd5678019b17818a539e1e06ef93d29b268a21f53f0773a9fff5d  
Status: Downloaded newer image for bkimminich/juice-shop:latest  
docker.io/bkimminich/juice-shop:latest
```

Figure 4 - Pulling and extracting the juice shock docker image.

```
└─(ricardor@ricardor)─[~]
$ sudo docker run --rm -p 3000:3000 bkimminich/juice-shop
info: Detected Node.js version v22.21.1 (OK)
info: Detected OS linux (OK)
info: Detected CPU x64 (OK)
info: Configuration default validated (OK)
info: Entity models 20 of 20 are initialized (OK)
info: Required file server.js is present (OK)
info: Required file index.html is present (OK)
info: Required file styles.css is present (OK)
info: Required file main.js is present (OK)
info: Required file tutorial.js is present (OK)
info: Required file runtime.js is present (OK)
info: Required file vendor.js is present (OK)
info: Port 3000 is available (OK)
info: Domain https://www.alchemy.com/ is reachable (OK)
info: Chatbot training data botDefaultTrainingData.json validated (OK)
info: Server listening on port 3000
```

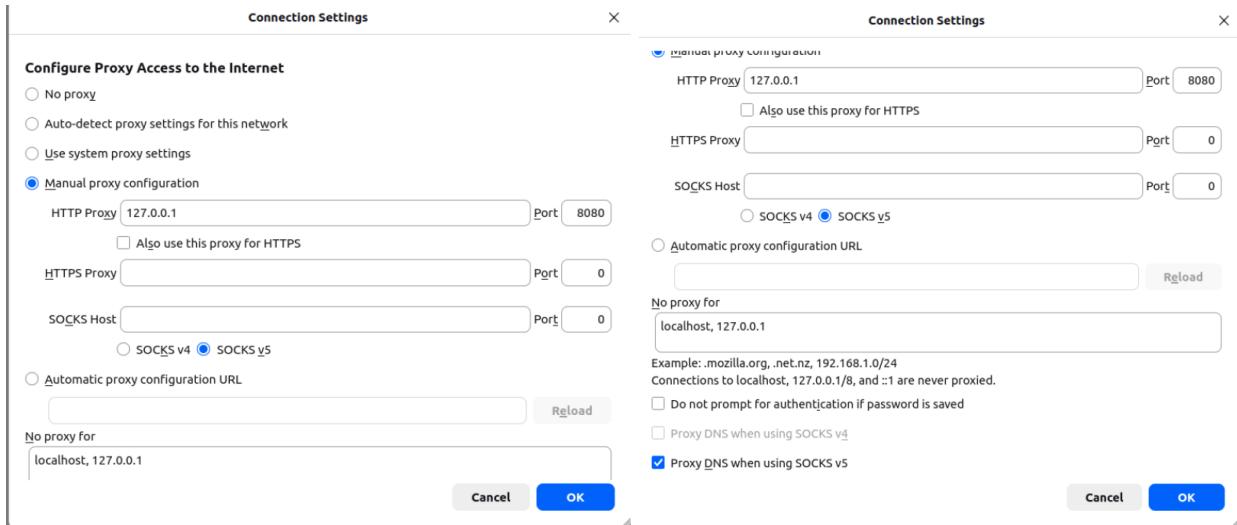
Figure 5 - Running the docker container.

On the attacker machine. I went to <http://192.168.194.157:3000>



On the attacker machine

Firefox -> Settings -> Network Settings



Then click ok

Now in Firefox we typed: http:192.168.194.157:3000

The screenshot shows the Burp Suite interface. At the top, there's a navigation bar with 'Burp', 'Project', 'Intruder', 'Repeater', 'View', 'Help', and 'Burp Suite Community Edition v2025.10.6 - Temporary Project'. Below the navigation bar, there are tabs: 'Dashboard', 'Target', 'Proxy' (which is selected), 'Intruder', 'Repeater', 'Collaborator', 'Sequencer', 'Decoder', 'Comparer', 'Logger', 'Organizer', 'Extensions', and 'Learn'. Under the 'Proxy' tab, there are sub-tabs: 'Intercept' (which is selected), 'HTTP history', 'WebSockets history', 'Match and replace', and 'Proxy settings'. On the left, there are buttons for 'Intercept on' (blue), 'Forward' (orange), 'Drop' (grey), and 'Stop' (grey). In the center, there's a table of captured requests:

Time	Type	Direction	Method	URL	Status code	Length
14:48:24...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:48:29...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:48:33...	HTTP	→ Request	GET	http://detecportal.firefox.com/success.txt?pv4		
14:48:33...	HTTP	→ Request	GET	http://detecportal.firefox.com/success.txt?pv6		
14:48:34...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:48:34...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:48:34...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:48:34...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:48:34...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:48:44...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:48:49...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:49:06...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:49:11...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:49:16...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:49:21...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:49:26...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:49:31...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:49:47...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:49:51...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:49:57...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:50:02...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:50:07...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:50:12...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		
14:50:47...	HTTP	→ Request	GET	http://detecportal.firefox.com/canonical.html		

Request:

```
1 GET / HTTP/1.1
2 Host: 192.168.194.157:3000
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:145.0) Gecko/20100101 Firefox/145.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-CA,en-US;q=0.7,en;q=0.8
6 Accept-Encoding: gzip, deflate, br
7
8 Cookie: language=en; welcomebanner_status=dismiss
9 Upgrade-Insecure-Requests: 1
10 If-Modified-Since: Sun, 07 Dec 2025 19:33:49 GMT
11 If-None-Match: W/"1252f-19afa4e55be"
12 Priority: u=0, i
13
14
```

Inspector:

- Request attributes
- Request query parameters
- Request body parameters
- Request cookies
- Request headers

Figure 6 - And here we confirm the request.

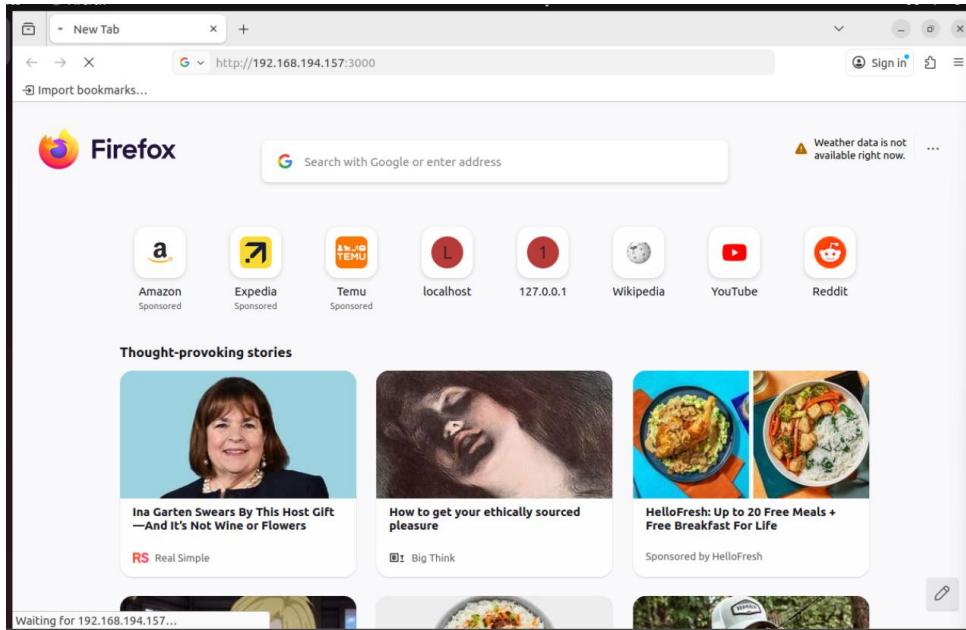


Figure 7 - Website tries to load. But it won't until we forward the request from burp suite.

Time	Type	Direction	Method	URL
14:57:17 ...	HTTP	→ Request	GET	http://detectportal.firefox.com/canonical.html
14:57:53 ...	HTTP	→ Request	GET	http://detectportal.firefox.com/canonical.html
14:57:55 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/assets/08nlen.json
14:57:55 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/socket.io/?EIO=4&transport=polling&t=PhwP6az
14:57:56 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/rest/admin/application-version
14:57:56 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/rest/admin/application-configuration
14:57:56 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/api/Challenges?name=Score%20Board
14:57:56 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/rest/admin/application-version
14:57:56 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/rest/anguag...
14:57:56 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/rest/admin/application-configuration
14:57:56 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/api/Challenges?name=Score%20Board
14:57:56 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/rest/admin/application-configuration
14:57:56 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/api/Quantity
14:57:56 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/rest/products/search?q=
14:57:58 ...	HTTP	→ Request	GET	http://detectportal.firefox.com/canonical.html
14:58:03 ...	HTTP	→ Request	GET	http://detectportal.firefox.com/canonical.html
14:58:09 ...	HTTP	→ Request	GET	http://detectportal.firefox.com/canonical.html
14:58:14 ...	HTTP	→ Request	GET	http://detectportal.firefox.com/canonical.html
14:58:17 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/socket.io/?EIO=4&transport=polling&t=PhwPB0i
14:58:19 ...	HTTP	→ Request	GET	http://detectportal.firefox.com/canonical.html
14:58:25 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/
14:58:40 ...	HTTP	→ Request	GET	http://192.168.194.157:3000/socket.io/?EIO=4&transport=polling&t=PhwPHMT

Figure 8 - After clicking Forward.

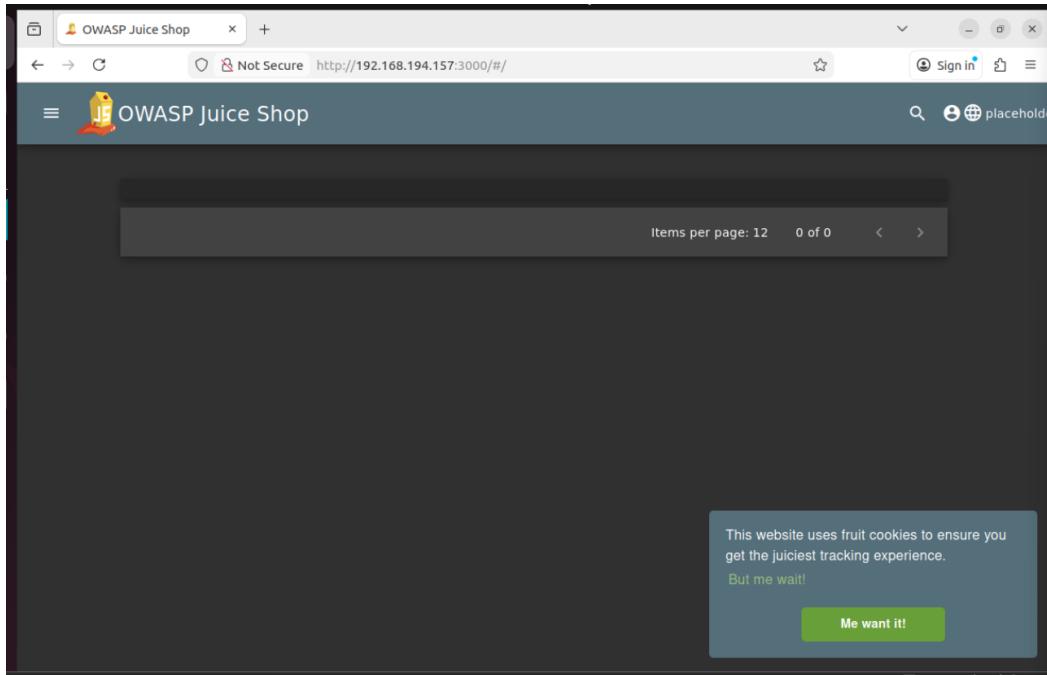


Figure 9 - Now we see the website.

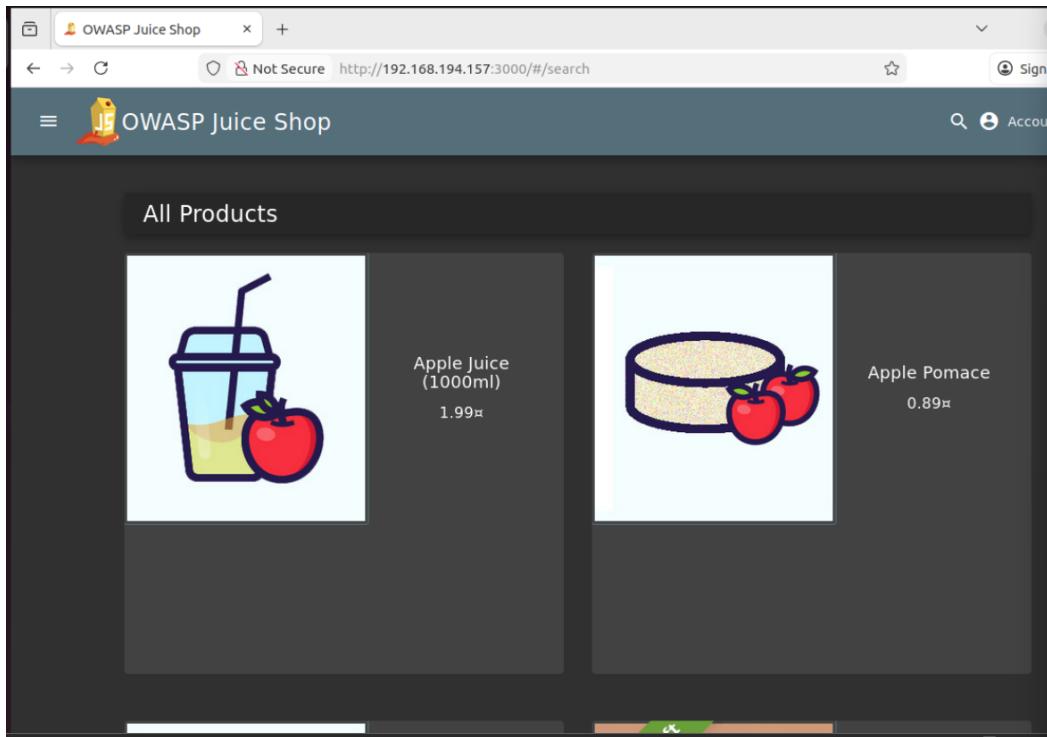


Figure 10 - In order to see the pictures, we turned off the intercept and the refresh the website.

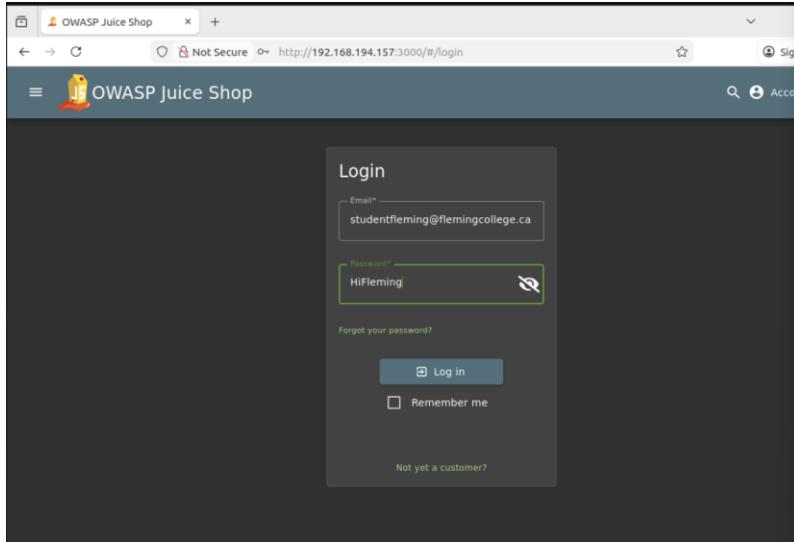


Figure 11 - False credentials used.

And in Burpsuite we can see the false credentials being intercepted.

A screenshot of the Burp Suite Community Edition v2025.10.6 interface. The 'Proxy' tab is selected. The 'Intercept' button is highlighted in blue. The 'Request' tab is selected under the Intercepted section. The 'Inspector' tab is also visible. The main pane shows a list of network requests. The 15th request in the list is highlighted with a blue selection bar. This request is a POST to http://192.168.194.157:3000/rest/user/login. The Request pane displays the raw POST data:

```
POST /rest/user/login HTTP/1.1
Host: 192.168.194.157:3000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:145.0) Gecko/20100101 Firefox/145.0
Accept: application/json, text/plain, */*
Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 67
Origin: http://192.168.194.157:3000
Connection: keep-alive
Referer: http://192.168.194.157:3000/
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
Priority: u=0
{
  "email": "studentfleming@flemingcollege.ca",
  "password": "HiFleming"
}
```

The URL column of the table lists various URLs such as canonical.html, socket.io, and rest/user/login.

Figure 12 - Credentials intercepted.

The Attack

And now we will perform the attack for the vulnerability:

Username enumeration via different response (Sopyan, 2025).

The goal is to identify valid usernames through differential error responses.

How? Using BurpSuite -> Intruder to send multiple logins attempts with the emails we are going to add to search for the existing usernames.

The 401 and 500 http error will help us to identify which emails exists. And the impact that this can cause is that it can allows an attacker to confirm valid usernames or accounts and then he can use other techniques to figure it out the passwords.

Once we had the intercepted credentials. Then we are going to modified it. To know if an email is already registered.

Click Intruder. And we modified the last line. We select
studentfleming@flemingcollege.ca and then click on “Add 5”

The screenshot shows the Burp Suite Community Edition interface. In the top navigation bar, the 'Sniper attack' tab is selected. The 'Target' field contains the URL <http://192.168.194.157:3000>. A button labeled 'Start attack' is visible. On the right, the 'Payloads' section is open, showing a simple list payload type with one item: 'studentfleming@flemingcollege.ca'. Below the payloads, the 'Payload processing' section is shown with a table for defining rules.

```
POST /rest/user/login HTTP/1.1
Host: 192.168.194.157:3000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:145.0) Gecko/20100101 Firefox/145.0
Accept: application/json, text/plain, */*
Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 67
Origin: http://192.168.194.157:3000
Connection: keep-alive
Referer: http://192.168.194.157:3000/
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
Priority: u=0
{"email": "studentfleming@flemingcollege.ca", "password": "HiFleming"}
```

Figure 13 – Line we change.

Then we added emails to the list. On the right side we see payloads.

The screenshot shows a software interface for managing payloads. On the left, there's a sidebar with tabs for 'Payloads' (selected), 'Resource pool', and 'Settings'. The main area has sections for 'Payload configuration' and 'Payload processing'. In 'Payload configuration', the 'Payload type' is set to 'Simple list'. A list of emails is displayed, with a red bracket highlighting the list items: admin@flemingcollege.ca, administrator@flemingcollege.ca, johnwilson@flemingcollege.ca, terryrozier@flemingcollege.ca, steveadams@flemingcollege.ca, jackmcmahon@flemingcollege.ca, and studentfleming@fleming.ca. Below this list are buttons for Paste, Load..., Remove, Clear, Deduplicate, Add, and Add from list... [Pro version only]. In 'Payload processing', there's a table with columns for 'Enabled' and 'Rule', and buttons for Add, Edit, Remove, Up, and Down.

Payloads

Payload position: All payload positions

Payload type: Simple list

Payload count: 8

Request count: 8

Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

Paste
Load...
Remove
Clear
Duplicate

admin@flemingcollege.ca
administrator@flemingcollege.ca
johnwilson@flemingcollege.ca
terryrozier@flemingcollege.ca
steveadams@flemingcollege.ca
jackmcmahon@flemingcollege.ca
studentfleming@fleming.ca

Add
Add from list... [Pro version only]

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Enabled	Rule

Add
Edit
Remove
Up
Down

Resource pool

Settings

Figure 14 - Adding emails.

After adding the emails. We Clicked on “Start Attack”. And it opens the following window:

The screenshot shows the OWASP ZAP Intruder attack interface. At the top, there are buttons for 'Attack' and 'Save'. The title bar says '2. Intruder attack of http://192.168.194.157:3000'. Below the title, there are tabs for 'Results' and 'Positions', with 'Results' selected. A 'Capture filter' dropdown is open, showing 'Capturing all items'. A 'View filter' dropdown is also open, showing 'Showing all items'. On the right, there is a 'Apply capture filter' button. The main area displays a table of captured requests:

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		401	103			413	
1		500	30			1621	
2	admin@flemingcollege.ca	500	15			1619	
3	administrator@flemingcollege.ca	500	12			1619	
4	johnwilson@flemingcollege.ca	500	11			1619	
5	terryrozier@flemingcollege.ca	500	12			1627	
6	steveadams@flemingcollege.ca	500	11			1619	
7	jackcmahon@flemingcollege.ca	500	17			1619	
8	studentfleming@fleming.ca	500	12			1619	

Below the table, there are tabs for 'Request', 'Response', 'Pretty', 'Raw', and 'Hex'. The 'Pretty' tab is selected. The request details are as follows:

```
1 POST /rest/user/login HTTP/1.1
2 Host: 192.168.194.157:3000
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:145.0) Gecko/20100101 Firefox/145.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/json
8 Content-Length: 67
9 Origin: http://192.168.194.157:3000
10 Connection: keep-alive
11 Referer: http://192.168.194.157:3000/
12 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
13 Priority: u=0
14
15 {
  "email": "studentfleming@flemingcollege.ca",
  "password": "HiFleming"
}
```

Figure 15 - Intruder attack window.

Here we can see how the emails we added are giving us the Status code as 500. Which means the internal server error, and it can be because the email doesn't exist and they are trying to validate a user that doesn't exit. (MDN n.d.)

On the other hand, we can see that the first line has the status code as 401. Which means that the user exists, but the password is invalid. (MDN n.d.)

Attack Save X

2. Intruder attack of http://192.168.194.157:3000

2. Intruder attack of http://192.168.194.157:3000

Attack Save

Results Positions

Capture filter: Capturing all items

View filter: Showing all items

Apply capture filter

Payloads Resource pool Settings

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		401	103			413	
1		500	30			1621	
2	admin@flemingcollege.ca	500	15			1619	
3	administrator@flemingcollege.ca	500	12			1619	
4	johnwilson@flemingcollege.ca	500	11			1619	
5	terryrozier@flemingcollege.ca	500	12			1627	
6	steveadams@flemingcollege.ca	500	11			1619	
7	jackcmahon@flemingcollege.ca	500	17			1619	
8	studentfleming@fleming.ca	500	12			1619	

Request Response

Pretty Raw Hex

```

1 POST /rest/user/login HTTP/1.1
2 Host: 192.168.194.157:3000
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:145.0) Gecko/20100101 Firefox/145.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/json
8 Content-Length: 67
9 Origin: http://192.168.194.157:3000
10 Connection: keep-alive
11 Referer: http://192.168.194.157:3000/
12 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
13 Priority: u=0
14
15 {
    "email": "studentfleming@flemingcollege.ca",
    "password": "HiFleming"
}

```

Therefore, we can confirm that the email address studentfleming@flemingcollege.ca is a valid email address registered.

Mitigation

During the attack, we used Burp intruder to test the /user/login endpoint. The server gave us different http status code when the email existed vs when it did not. This difference allows us to enumerate valid accounts by looking at the status code.

The screenshot shows the Burp Suite interface during an 'Intruder attack' on the endpoint `http://192.168.194.157:3000`. The 'Results' tab is selected, displaying a table of requests. The first row (index 0) shows a 401 status code, while subsequent rows (indices 1 through 8) show 500 status codes for various email addresses. The 'Payload' column lists the email addresses being tested. The 'Comment' column is empty for all entries. On the left, the 'Request' and 'Response' tabs are visible, with the 'Pretty' tab selected. The response body shows a POST request to the /rest/user/login endpoint with a JSON payload containing an email and password. The 'Payload' column also displays the JSON payload for each request. The right side of the interface includes tabs for 'Payloads', 'Resource pool', and 'Settings'.

Request ^	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		401	103			413	
1	admin@flemingcollege.ca	500	30			1621	
2	administrator@flemingcollege.ca	500	15			1619	
3	johnwilson@flemingcollege.ca	500	12			1619	
4	terryrozier@flemingcollege.ca	500	11			1619	
5	steveadams@flemingcollege.ca	500	12			1627	
6	jackmcmahon@flemingcollege.ca	500	11			1619	
7	studentfleming@fleming.ca	500	17			1619	
8	studentfleming@fleming.ca	500	12			1619	

Now the goal is to remove all observable differences between valid and invalid user accounts so an attacker can no longer identify existing users during a brute-force enumeration attack.

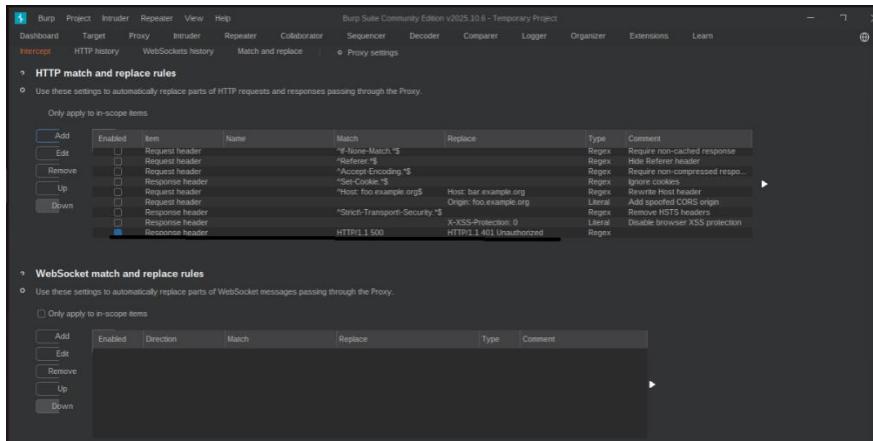
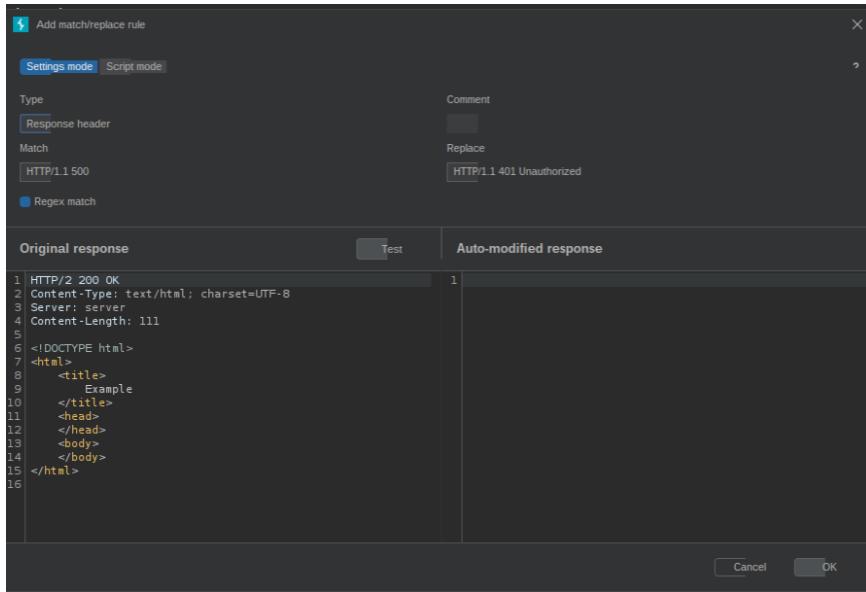
In order to achieve this, the login endpoint must always return the same type of response for every failed login, regardless of whether the email exists.

Security control

To fix this problem, we can apply a security control that makes the login endpoint respond the exact same way for every failed attempt. The problem happened because the server returned different HTTP status codes depending on whether the email existed or not. And because the attacker can see this information it is easy to identify which accounts were valid.

We forced every 500 status response to look like a 401 Unauthorized:

We would use “Response first line” on the match and replace rules to change the http status code. However, this option is only available in the burpsuite pro version. So, because we are using the community edition, we applied the closest possible control by using the “Response header rule” that forces every 500 responses to look like a 401 Unauthorized. So, this can remove the observable difference attackers were using to enumerate accounts.



References

Kimminich, B., & Hollenbach, J. (n.d.). *Juice-shop/juice-shop: Owasp Juice Shop: Probably the most modern and sophisticated insecure web application*. GitHub.
<https://github.com/juice-shop/juice-shop#docker-container>

MDN. (n.d.). *HTTP response status codes*. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference>Status>

Sopyan, A. (2025, June 4). *Authentication part-1 : Username enumeration via different responses*. Medium. <https://medium.com/@AhmadSopyan/authentication-part-1-username-enumeration-via-different-responses-8ffc76acf511>