

**PRAKTIKUM KEAMANAN JARINGAN**  
**“BROKEN ACCESS CONTROL”**



**Oleh :**  
**Andre Septian Prayogo**  
**D4 LJ Teknik Informatika B**  
**3122640033**

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**  
**TAHUN AJARAN**  
**2023**

Broken Access Control atau disebut juga sebagai Broken Access Management adalah suatu kerentanan keamanan yang terjadi pada aplikasi web atau sistem yang dapat memungkinkan pengguna tidak sah untuk mendapatkan akses yang tidak seharusnya pada sumber daya yang dilindungi atau fitur yang terbatas.

Secara umum, Access Control adalah proses untuk memastikan bahwa hanya pengguna yang diizinkan atau memiliki otoritas untuk melakukannya yang dapat mengakses sumber daya atau fitur tertentu. Jika mekanisme Access Control tidak dirancang atau diimplementasikan dengan benar, hal ini dapat mengakibatkan celah keamanan yang memungkinkan pengguna tidak sah atau penyerang untuk mengakses sumber daya atau fitur yang dilindungi tersebut.

Berikut adalah cara pent test untuk mengetahui tentang Broken Access management

Dalam pen test ini membutuhkan aplikasi Burpsuite, namun sebelum install aplikasi tersebut cek versi java kalian karena burpsuite hanya akan beroperasi pada versi java 17 ke atas.

1. lakukan pengecekan versi java.

```
(kali㉿kali)-[~]  
$ java -version  
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true  
openjdk version "17.0.6" 2023-01-17  
OpenJDK Runtime Environment (build 17.0.6+10-Debian-1)  
OpenJDK 64-Bit Server VM (build 17.0.6+10-Debian-1, mixed mode, sharing)  
  
(kali㉿kali)-[~]
```

Pada OS saya sudah terinstal java versi 17 maka minimum requirement sudah terpenuhi.

2. lakukan instalasi Burpsuite

```
(kali㉿kali)-[~/home/kali]  
$ sudo apt install burpsuite  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  libarmadillo9 libatk2-0-data libcairo2 libcharls2 libgda3 libgss3-10-2 libproj22 libpython3.10-minimal libpython3.10-stdlib libpython3.9-dev libtbb2 libvulkan1 libxrs-0 mesa-vulkan-drivers python-mpitoolkits.basemap-data  
  python3-pyproj python3-pyshp python3.10 python3.10-minimal python3.9-dev  
Use 'sudo apt autoremove' to remove them.  
The following packages will be upgraded:  
  burpsuite  
1 upgraded, 0 newly installed, 0 to remove and 1058 not upgraded.  
Need to get 0 B/224 MB of archives.  
After this operation, 2,162 kB of additional disk space will be used.  
(Reading database ... 328959 files and directories currently installed.)  
Preparing to unpack .../burpsuite_2023.1.2-0kali1_amd64.deb ...  
Unpacking burpsuite (2023.1.2-0kali1) over (2022.12.5-0kali1) ...  
Setting up burpsuite (2023.1.2-0kali1) ...  
Processing triggers for kali-menu (2021.4.2) ...
```

### 3. Jalankan OWASP Juice Shop

```
File Actions Edit View Help

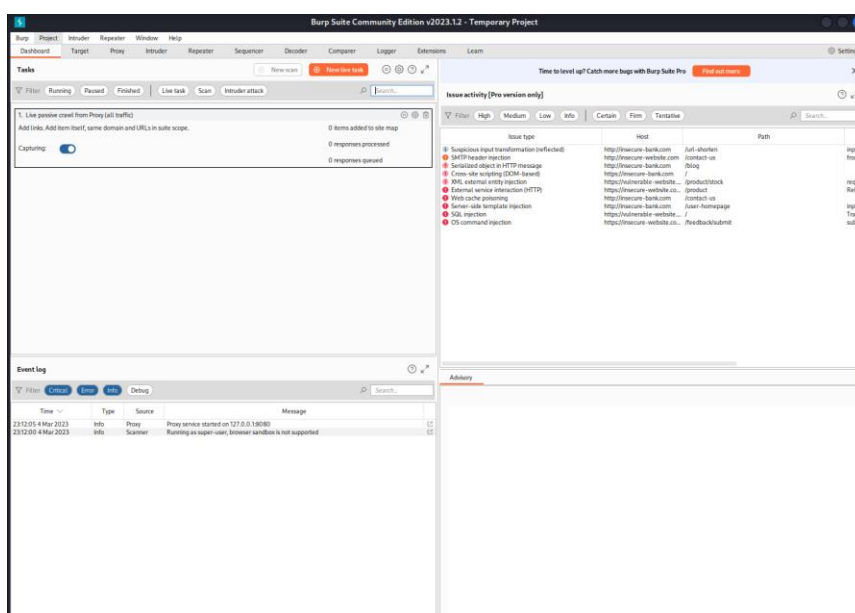
(kali㉿kali)-[~]
$ sudo su
[sudo] password for kali:
(kali㉿kali)-[/home/kali]
# cd /home/kali/juice-shop 14.0.1/

(kali㉿kali)-[/home/kali/juice-shop_14.0.1]
# npm start

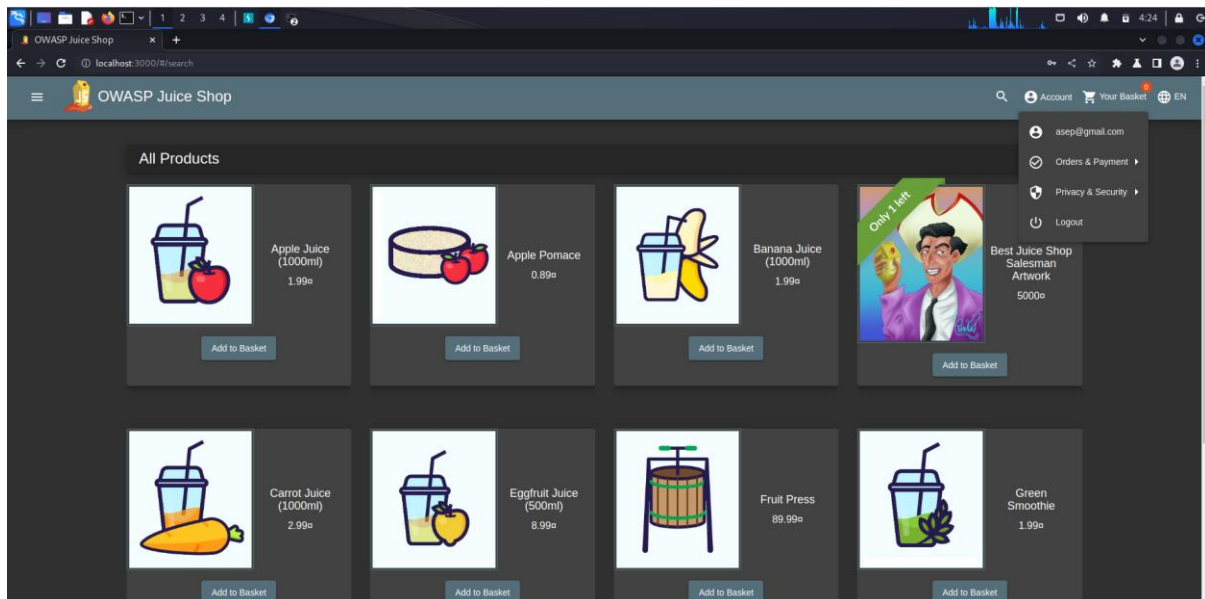
> juice-shop@14.0.1 start /home/kali/juice-shop_14.0.1
> node build/app

info: All dependencies in ./package.json are satisfied (OK)
info: Chatbot training data botDefaultTrainingData.json validated (OK)
info: Detected Node.js version v14.1.0 (OK)
info: Detected OS linux (OK)
info: Detected CPU x64 (OK)
info: Configuration default validated (OK)
info: Required file server.js is present (OK)
info: Required file main.js is present (OK)
info: Required file styles.css is present (OK)
info: Required file tutorial.js is present (OK)
info: Required file index.html is present (OK)
info: Required file polyfills.js is present (OK)
info: Required file vendor.js is present (OK)
info: Required file runtime.js is present (OK)
info: Port 3000 is available (OK)
info: Server listening on port 3000
```

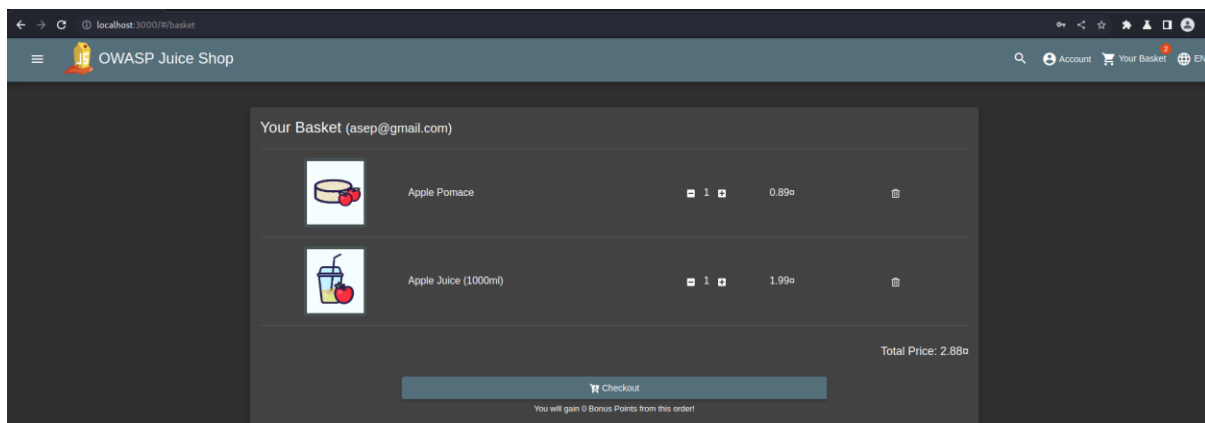
### 4. buka aplikasi burpsuite



Pada pengetesan pertama saya akan mengtrace dari HTTP request. Silahkan memilih proxy dan pilih HTTP History. Setelah itu buka browser dan buka halaman juice shop



Setelah login saya akan menambahkan 1 apple juice dan 1 apple pomace kedalam basket



Terlihat isi basket saya adalah 2 item di atas, sekarang coba buka burp suite dan trace http requestnya.

Intercept HTTP history WebSockets history Proxy settings							
Filter: Hiding CSS, image and general binary content							
#	Host	Meth...	URL	Params	Edited	Status	Length
60	http://localhost:3000	GET	/rest/languages			304	255
92	http://localhost:3000	GET	/rest/basket/6			200	488
97	http://localhost:3000	GET	/rest/basket/6			304	253
100	http://localhost:3000	GET	/rest/basket/6			200	977
101	http://localhost:3000	GET	/rest/basket/6			304	254
105	http://localhost:3000	GET	/rest/basket/6			304	254
108	http://localhost:3000	GET	/rest/basket/6			200	1349
109	http://localhost:3000	GET	/rest/basket/6			304	254
13	http://localhost:3000	GET	/rest/admin/application-version			304	253
18	http://localhost:3000	GET	/rest/admin/application-version			304	253
57	http://localhost:3000	GET	/rest/admin/application-version			304	253
63	http://localhost:3000	GET	/rest/admin/application-version			304	253

Saat salah satunya di tekan maka akan keluar informasi sebagai berikut

[illegible]

Jika dilihat dari request ini sebenarnya ini menggunakan API dan angka 6 sepertinya adalah nomer dari user yang sudah terdaftar untuk memanggil isi basket.

Mari kita kirim request di atas ke repeater dan liat apa yang akan terjadi

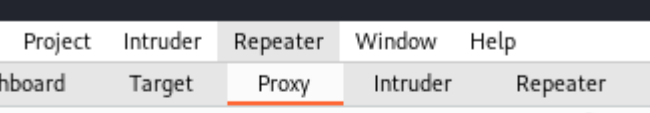
The screenshot shows the Burp Suite Repeater interface. On the left, the 'Request' tab is active, displaying a REST client request for `GET /rest/basket/6 HTTP/1.1`. The request includes headers like `Host: localhost:3000`, `Accept: application/json, text/plain, */*`, and a `Cookie` with `welcomebanner_status=dismiss`. The body is a JSON object representing a basket with two items: an apple and an apple juice. On the right, the 'Response' tab is active, showing the server's response. The status is `200 OK`, and the body is a JSON object with `"status": "success"` and a `"data"` array containing the items from the basket. The items are `"Apple Pomace"` and `"Apple Juice (1000ml)"`.

Disini bisa dilihat terdapat 2 item yaitu apel penance dan apple juice yang mana kedua item tersebut adalah item yang ada di dalam basket kita tadi.

Selanjutnya apa yang akan terjadi jika kita mengubah angka dari API request

This screenshot is similar to the first one, but the request path is `GET /rest/basket/2 HTTP/1.1`. The response body shows a different set of items: `"Rasperry Juice (1000ml)"` and `"Apple Juice (1000ml)"`. This demonstrates that the API is stateless and the response is based on the request parameters.

Dengan mengubah basket number ke angka 2 maka kita akan bisa melihat basket dari user lain dengan basket number 2.



The screenshot shows the Burp Suite application window. The top menu bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'Window', and 'Help'. Below the menu is a toolbar with buttons for 'Dashboard', 'Target', 'Proxy' (which is highlighted with a red underline), 'Intruder', 'Repeater', and 'Sequences'. Below the toolbar is another row of buttons: 'Intercept' (highlighted with a red underline), 'HTTP history', 'WebSockets history', and a gear icon labeled 'Proxy settings'. At the bottom of the interface, there are four buttons: 'Forward', 'Drop', 'Intercept is off', and 'Action'.

**Burp** | Project | Intruder | Repeater | Sequencer | Decoder | Comparer | Logger | Extensions | Learn

---

**Intercept** | HTTP history | WebSockets history | Proxy settings

---

Request to http://localhost:3000 [127.0.0.1]

Forward Drop Intercept is on Action Open browser

---

Pretty Raw Hex

```

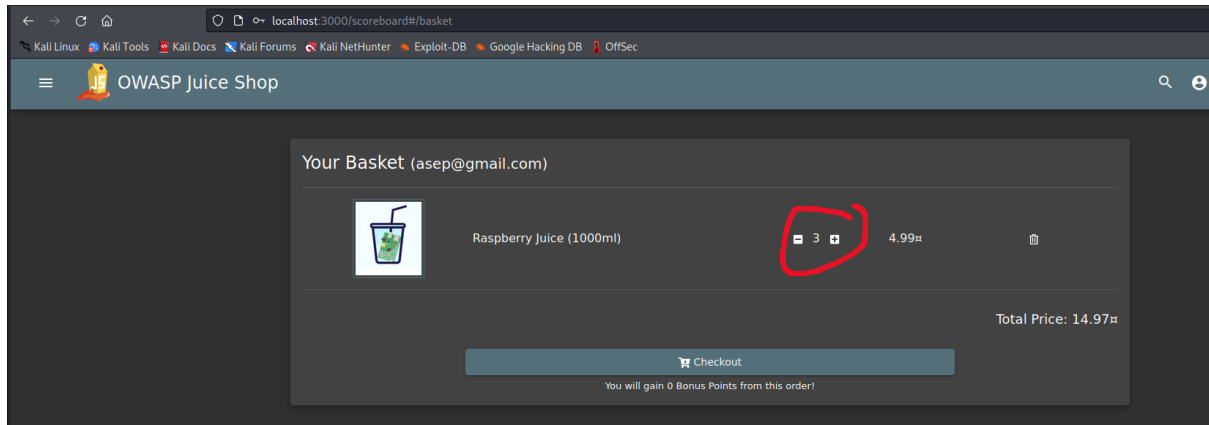
1 GET /rest/basket/6 HTTP/1.1
2 Host: localhost:3000
3 sec-ch-ua: "Not A(Brand";v="24", "Chromium";v="110"
4 Accept: application/json, text/plain, */*
5 X-User-Email: asepg@gmail.com
6 sec-ch-ua-mobile: ?0
7 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiaWF0IjEwMDYyMjZlOTQyZWZlLnRlcjBjb2xlIjoieY3VzdG9tZXIiLCJkZWxleGVub2t1biI6ImxhcjAuMCIsInByb2p6bWFnZSI6Ii9hc3NlLmhmcyhvbGljL2ltYWdlcy91cGxvYWRzLTZRLmZmFlhQuc3ZnIiwidG90cFNLfY3JldCI6IiIsImxlZQWNOdGvkQXQiOiIyMDIzLTAzLTAlIDA1OjElOjM4JmJ4UyNSARMDA6MDAiLCJlcGRhdGVkQXQiOiIyMDIzLTAzLTAlIDA1OjElOjM4JmJ4UyNSARMDA6MDAiLCJ9LCJpYXQiOiJlZDZlNTMzNDZlMTY3ODAxMTMOMX0.v3w7TV-mZmjTw9XnunQgSo-LVfyf7DrOFg4JxUDL6pKWtuEOf9L0JHUAzg0Wp6XMSQ-oQKwu2aCkQcuC08fgRl5uzDGRglRTNHUSTdEnZ4amVrzHQU-EQXEI2ysIxZ4hb_G8YyfMTQFw2kSyoszy20w
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.78 S
9 sec-ch-ua-platform: "Linux"
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: http://localhost:3000/
14 Accept-Encoding: gzip, deflate
    
```

A screenshot of a web browser displaying the OWASP Juice Shop application. The browser's address bar shows 'localhost:3000/m/basket'. The page has a dark theme. At the top, there's a navigation bar with a hamburger menu icon, the 'OWASP Juice Shop' logo, and links for 'Account', 'Your Basket', and 'EN'. The main content area is titled 'Your Basket (asep@gmail.com)'. It contains a single item: 'Raspberry Juice (1000ml)' with a price of '4.99'. To the left of the item is an icon of a glass of juice. To the right of the item name are icons for quantity (showing '2') and a trash can icon. At the bottom right of the basket area, it says 'Total Price: 9.98'. Below this, there is a large 'Checkout' button. Underneath the button, a message states: 'You will gain 0 Bonus Points from this order!'.



Maka di browser akan tertampil basket dari user dengan nomor 2. (**BROKEN ACCESS CONTROL VIEW BASKET**)

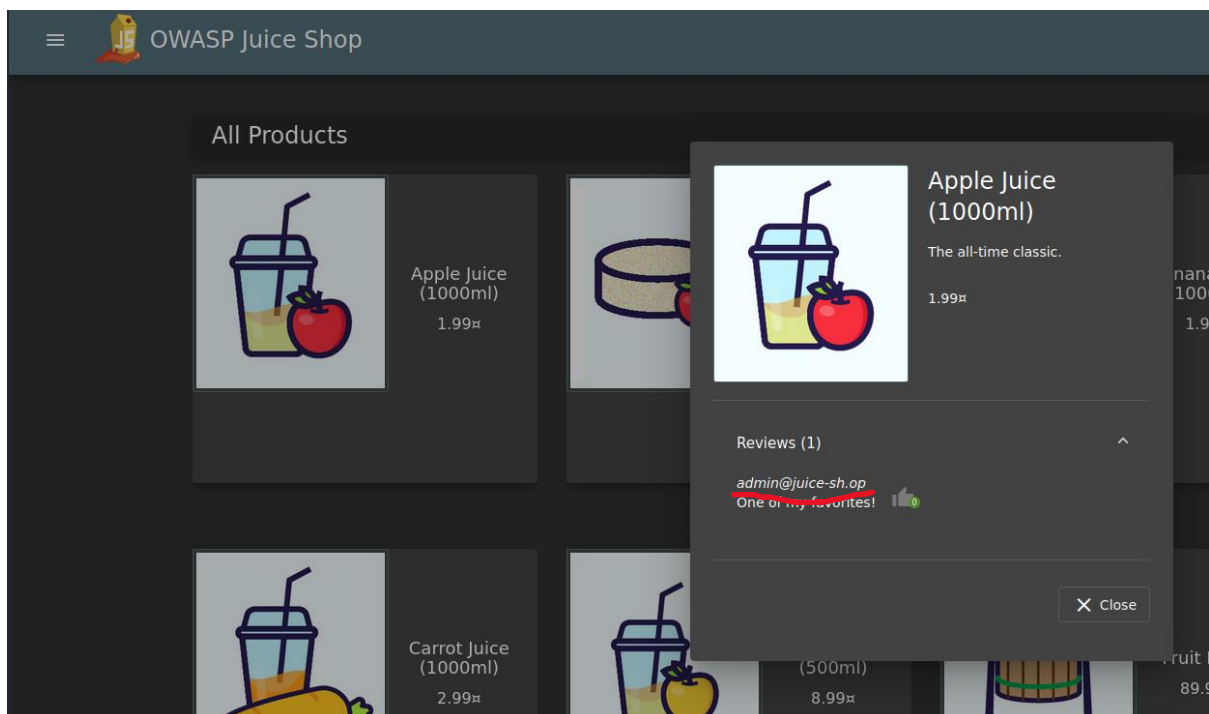
Selanjutnya menambahkan atau mengubah basket dari user lain ( **BROKEN ACCESS CONTROL BASKET MANIPULATE**)



Sekarang user tersebut mempunyai 3 raspberry juice setelah tadinya hanya 2 buah. Untuk menampilkan hasil dari modifikasi di atas dapat mengulangi step dari intercept.

Selanjutnya menggunakan metode Burte Force untuk melakukan login Admin (**BROKEN ACCESS CONTROL ADMIN SECTION**)

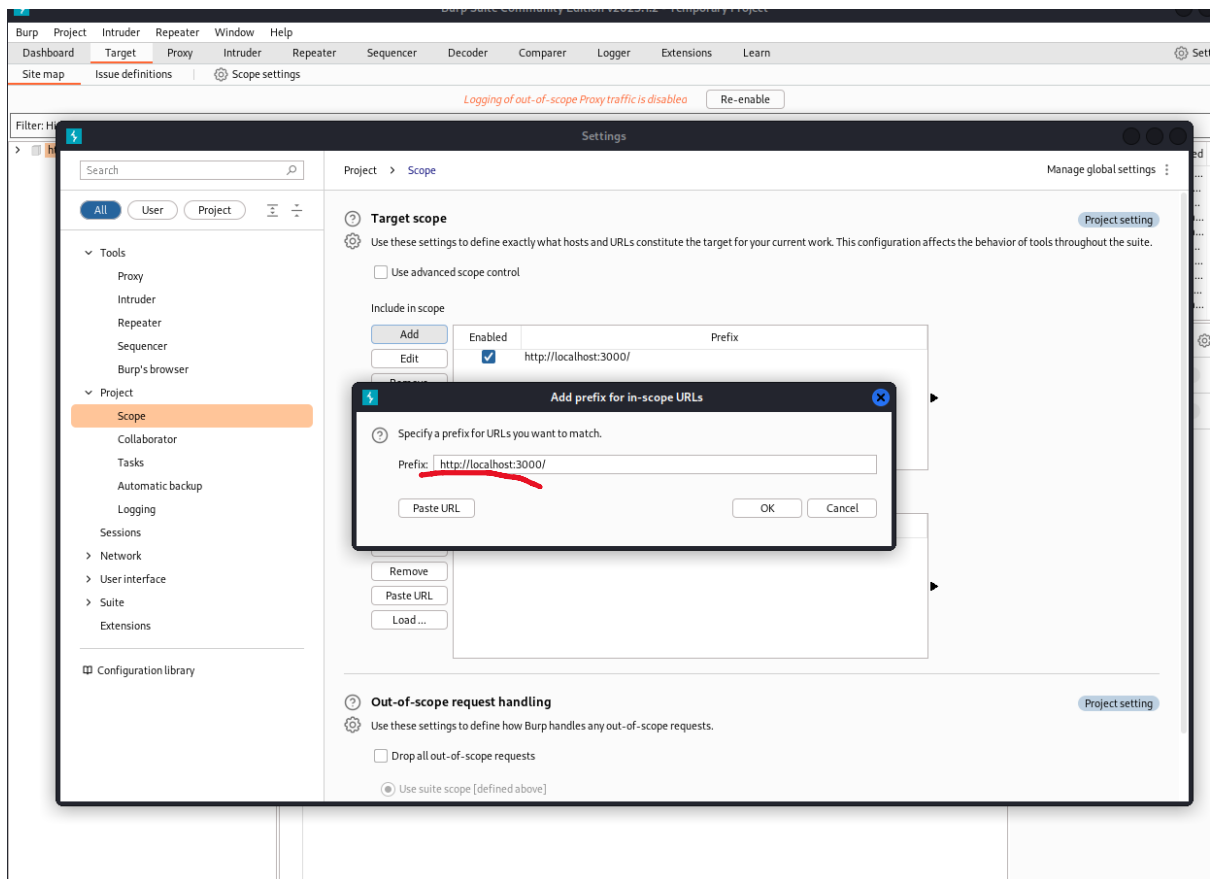
Hal pertama yang harus dilakukan adalah mengetahui email dari admin terlebih dahulu



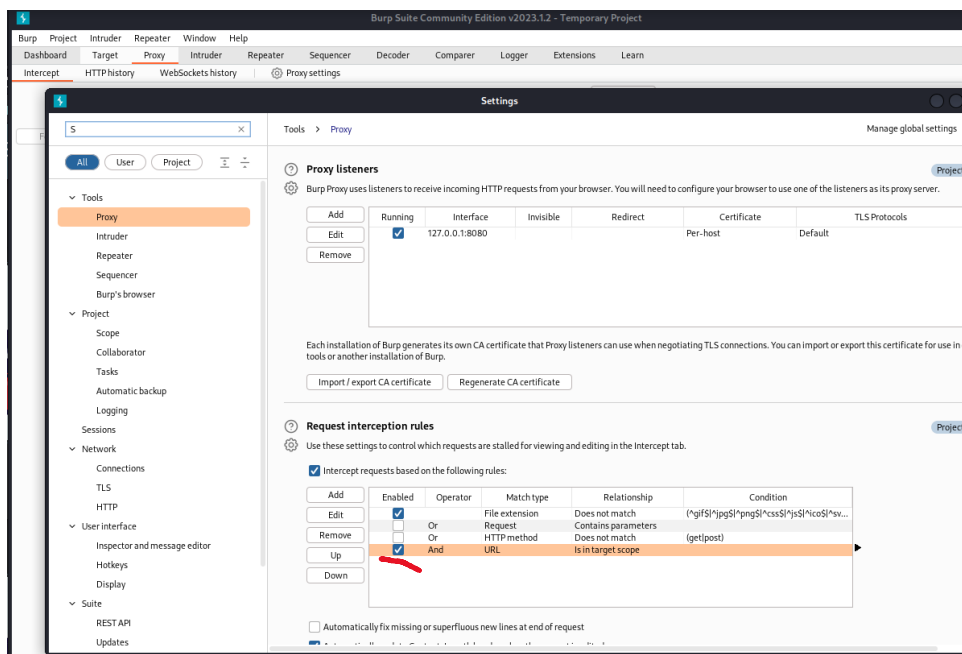
Dengan membuka informasi dari item maka kita akan menemukan informasi email dari admin



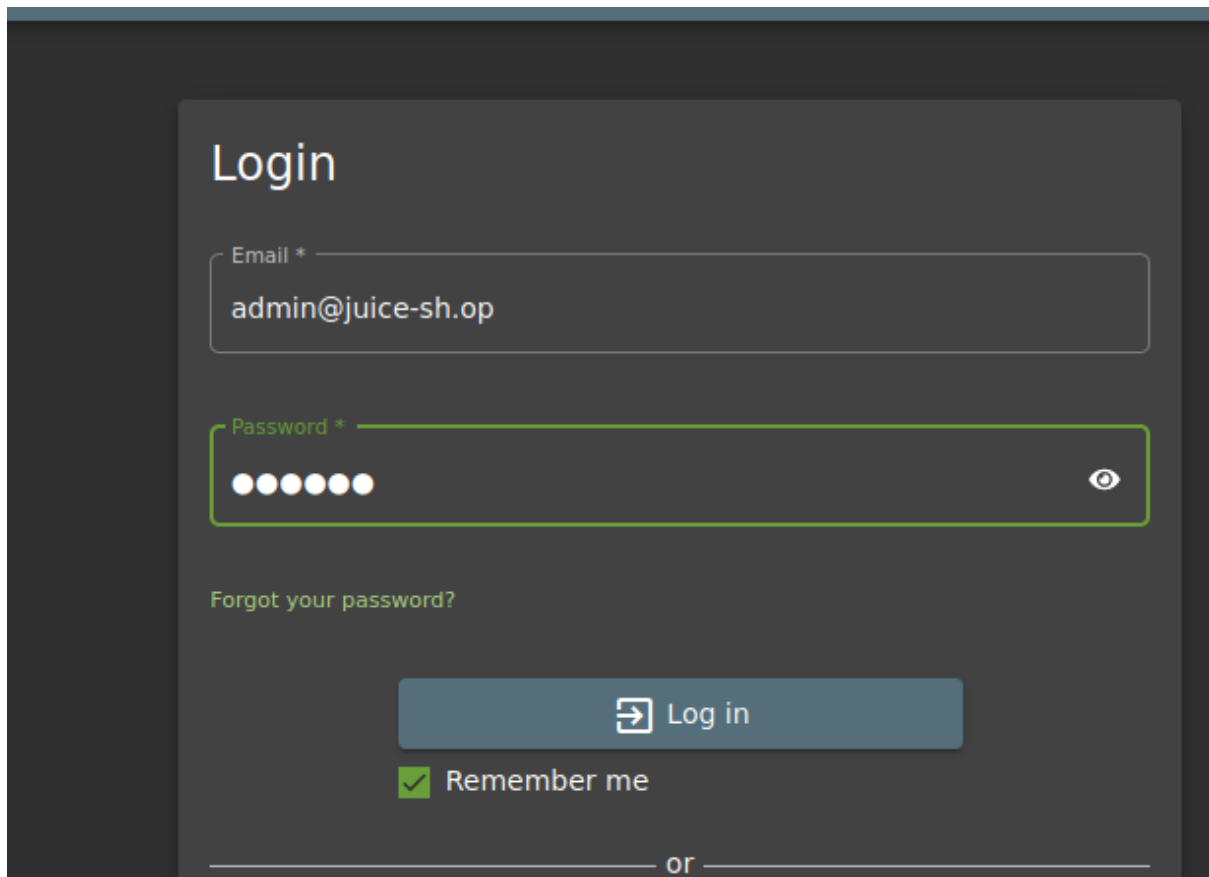
Selanjutnya tambahkan URL dari juice shop ke target site



Pada proxy seting centang “Is In Target Scope”

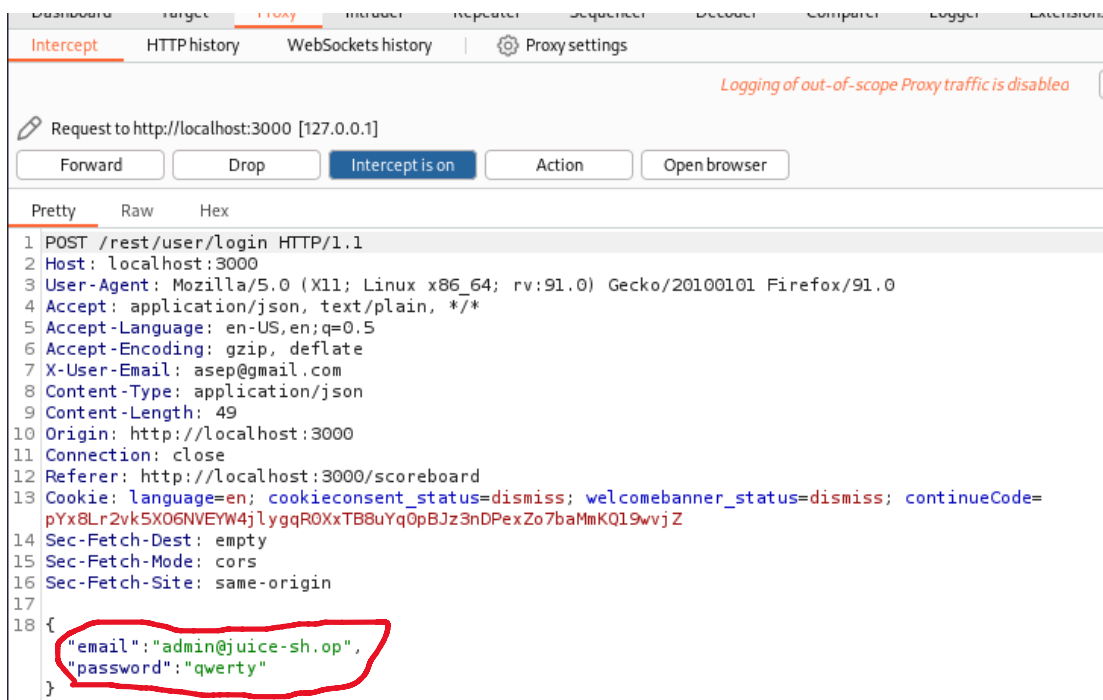


Kembali ke halaman login, masukkan email dari admin dengan password random



Sebelum itu siapkan intercept di burpsuite, aktifkan dan tekan login

Pada intercept klik forward sampai muncul username admin dan password yang kita inputkan tadi



Kemudian kirim ke intruder

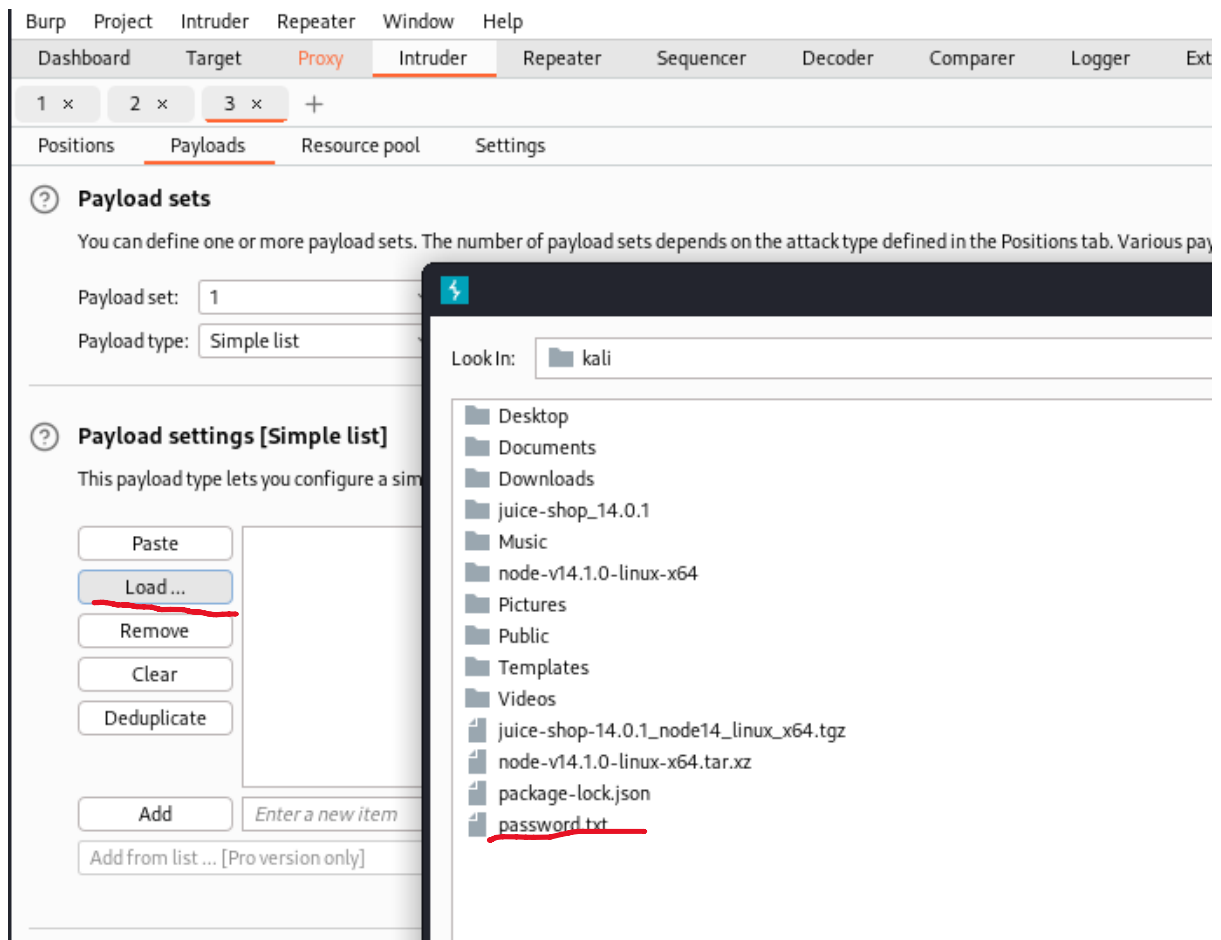
The screenshot shows the Burp Suite interface. At the top, there are tabs for 'Positions', 'Payloads', 'Resource pool', and 'Settings'. The 'Positions' tab is active. Below the tabs, there's a section 'Choose an attack type' with a dropdown menu set to 'Sniper' and a 'Start attack' button. Below that, the 'Payload positions' section is visible. It has a 'Target' field set to 'http://localhost:3000' and a checkbox 'Update Host header to match target' which is checked. On the right side of this section, there are buttons: 'Add \$', 'Clear \$', 'Auto \$', and 'Refresh'. The main area displays an HTTP request log. The request is a POST to '/rest/user/login' with various headers and a JSON body. The body contains 'email': 'Sadmin@juice-sh.op\$' and 'password': '\$qwerty\$'. The 'Clear \$' button is highlighted with a red line.

Bersihkan target untuk semua burte force dengan menekan clear

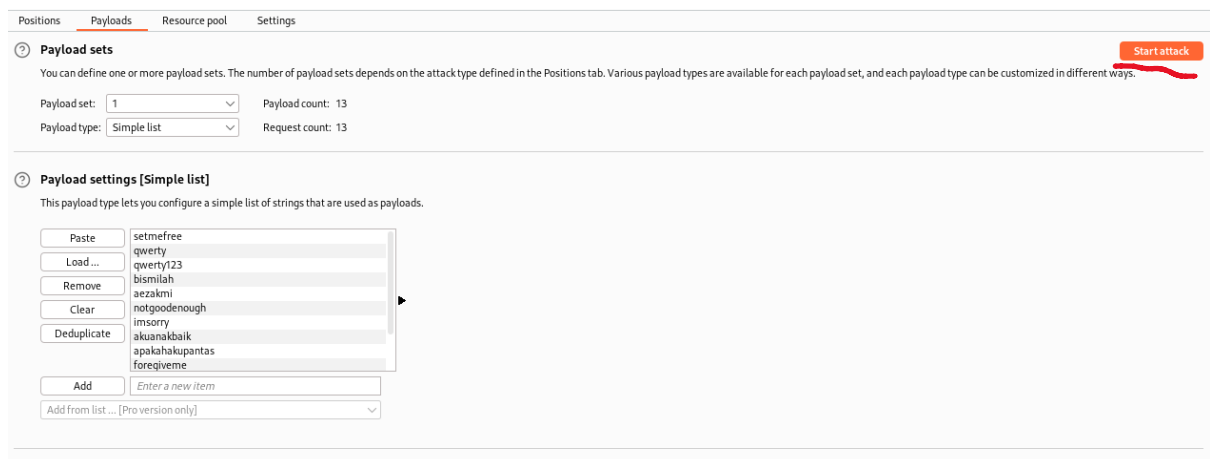
Setelah itu pilih target burteforce yang mana ini adalah password dan tekan add

This screenshot is similar to the previous one, showing the 'Payload positions' section. The 'Target' field is still 'http://localhost:3000'. The 'Update Host header to match target' checkbox is checked. The 'Add \$' button is now highlighted with a red circle. The HTTP request log shows the same request, but the body is now '{"email": "admin@juice-sh.op", "password": "\$qwerty\$"}'. The 'Clear \$' button is still highlighted with a red line.

Pada menu payload, muat file berformat .txt sebagai password bank yang nantinya akan di gunakan untuk burte force



Disini saya sudah menyiapkan file tersebut



Setelah password sudah di load tekan start attack

5. Intruder attack of http://localhost:3000 - Temporary attack - Not saved to project file							
Attack Save Columns							
Results Positions Payloads Resource pool Settings							
Filter: Showing all items							
Request ^	Payload	Status	Error	Timeout	Length	Comment	
0		401	<input type="checkbox"/>	<input type="checkbox"/>	362		
1	setmefree	401	<input type="checkbox"/>	<input type="checkbox"/>	362		
2	qwerty	401	<input type="checkbox"/>	<input type="checkbox"/>	362		
3	qwerty123	401	<input type="checkbox"/>	<input type="checkbox"/>	362		
4	bismilah	401	<input type="checkbox"/>	<input type="checkbox"/>	362		
5	aezakmi	401	<input type="checkbox"/>	<input type="checkbox"/>	362		
6	notgoodenough	401	<input type="checkbox"/>	<input type="checkbox"/>	362		
7	imsorry	401	<input type="checkbox"/>	<input type="checkbox"/>	362		
8	hurtsogood	401	<input type="checkbox"/>	<input type="checkbox"/>	362		
9	imnottheonlyone	401	<input type="checkbox"/>	<input type="checkbox"/>	362		
10	foregiveme	401	<input type="checkbox"/>	<input type="checkbox"/>	362		
11	password	401	<input type="checkbox"/>	<input type="checkbox"/>	362		
12	password123	401	<input type="checkbox"/>	<input type="checkbox"/>	362		
13	<u>admin123</u>	<u>200</u>	<input type="checkbox"/>	<input type="checkbox"/>	1169		

Pada data set password di atas terdapat 2 jenis HTTP status yaitu 401 dan 200, 401 sendiri terjadi karena saat permintaan browser ke server tidak memiliki kredensial autentik yang valid, untuk HTTP request 200 server berhasil menerima request dari browser yang kita gunakan dapat diartikan OK.

Login dengan password dari hasil brute force

## Login

Email \*

Password \*

Forgot your password?

☒ Remember me

or

Not yet a customer?

