Praktikum Keamanan Jaringan
(Broken Acccess Control)

Dosen Pembimbing :
Dr. Ferry Astika Saputra, ST., M.Sc.

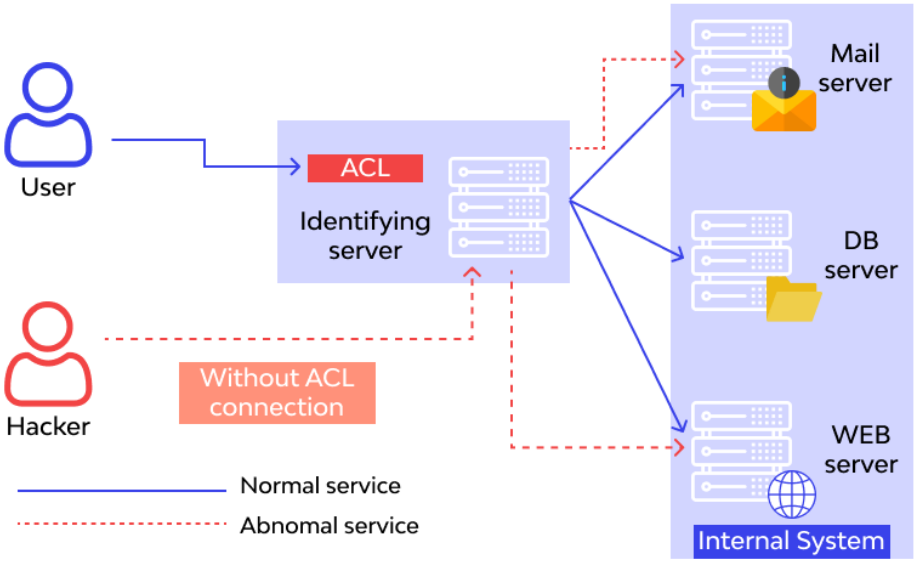
Disusun Oleh :
Fifin Nur Rahmawati (3122640040)

1 D4 – IT B LJ

PROGRAM STUDI TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

2023/2024

Broken Access Control



Broken Access Control

Sumber : https://owasp.org/Top10/id/A01_2021-Broken_Access_Control/

Akses Kontrol menetapkan sebuah peraturan yang dimana user tidak dapat melakukan sebuah aksi diluar permission yang diberikan. Kegagalan atas hal ini dapat mengakibatkan pengeluaran informasi yang tidak diizinkan, modifikasi, atau penghancuran dari semua data atau pemberlakuan sebuah fungsi bisnis di luar limit sebuah user. Kelemahan Akses Kontrol termasuk dari :

- Melewati pengecekan akses kontrol dengan memodifikasi URL, internal application state, atau HTML page, atau menggunakan custom API attack tool.
- Membolehkan primary key untuk dapat diganti ke record user lain, membolehkan penglihatan atau perubahan akun orang lain.
- Penaikan sebuah privilege (Elevation Privilege). Yang dimana sebuah orang dapat dianggap sebagai user tanpa melakukan logged in dan yang dimana sebuah user dapat dianggap sebagai admin tanpa melakukan logged in.
- Manipulasi metadata, seperti memanipulasi dengan JSON Web Token (JWT) akses kontrol token, atau memanipulasi cookie atau hidden field untuk menaikan privilege (elevation privilege) atau menyalahgunakan penggunaan dari JWT invalidation.
- Konfigurasi yang salah pada CORS sehingga menyebabkan API akses yang tidak diizinkan.
- Force browsing untuk mengakses authenticated pages sebagai unauthenticated user atau mengakses privileged pages sebagai user standard. Mengakses API yang tidak memiliki akses kontrol untuk POST, PUT, dan DELETE.

Cara Mencegah

Akses Kontrol hanya efektif pada kode server-side yang dapat dipercaya dan server-less API, yang dimana penyerang tidak dapat memodifikasi pengecek akses kontrol atau meta datanya.

- Menolak semua akses kecuali ke public resource.
- Melakukan implementasi mekanisme akses kontrol sekali dan digunakan kembali pada seluruh aplikasi sehingga meminimalisir penggunaan CORS.
- Agar user tidak dapat melakukan create, read, update, atau mendelete record secara bebas, model akses kontrol seharusnya membatasi hal tersebut dengan menggunakan ownership untuk tiap record.
- Batas yang diperlukan oleh bisnis yang unik pada aplikasi seharusnya dilakukan oleh domain models.
- Nonaktifkan direktori listing web server dan pastikan file metadata (contohnya .git) dan file backup tidak ada di dalam web roots.
- Catat kegagalan akses kontrol dan alert admin jika diperlukan (seperti adanya kegagalan yang terjadi berulang - ulang).
- Ukur batasan dari API dan akses ke kontroler untuk meminimalisir kerusakan dari automated attack tooling.
- JWT tokens harus langsung di hilangkan validasinya pada server setelah logout.

Contoh Skenario Penyerangan

Skenario #1: Aplikasi menggunakan data yang belum diverifikasi pada sebuah pemanggilan SQL yang mengakses informasi akun

```
pstmt.setString(1, request.getParameter("acct"));

ResultSet results = pstmt.executeQuery( );
```

Penyerang hanya perlu untuk memodifikasi parameter ‘acct’ pada browser untuk mengirim nomer akun mana yang diinginkan. Jika parameter tersebut tidak diverifikasi secara benar, maka penyerang dapat mengakses akun user manapun.

https://example.com/app/accountInfo?acct=notmyacct

Skenario #2: Penyerang dapat memaksa untuk melakukan penjelajahan ke target URLs. Halaman Admin memerlukan hak admin untuk dapat diakses.

```
https://example.com/app/getappInfo

https://example.com/app/admin_getappInfo
```

Jika sebuah user yang belum di autentikasi dapat mengakses kedua page tersebut maka itu merupakan suatu kelemahan. Jika user yang non-admin dapat mengakses halaman admin, maka merupakan suatu kelemahan.

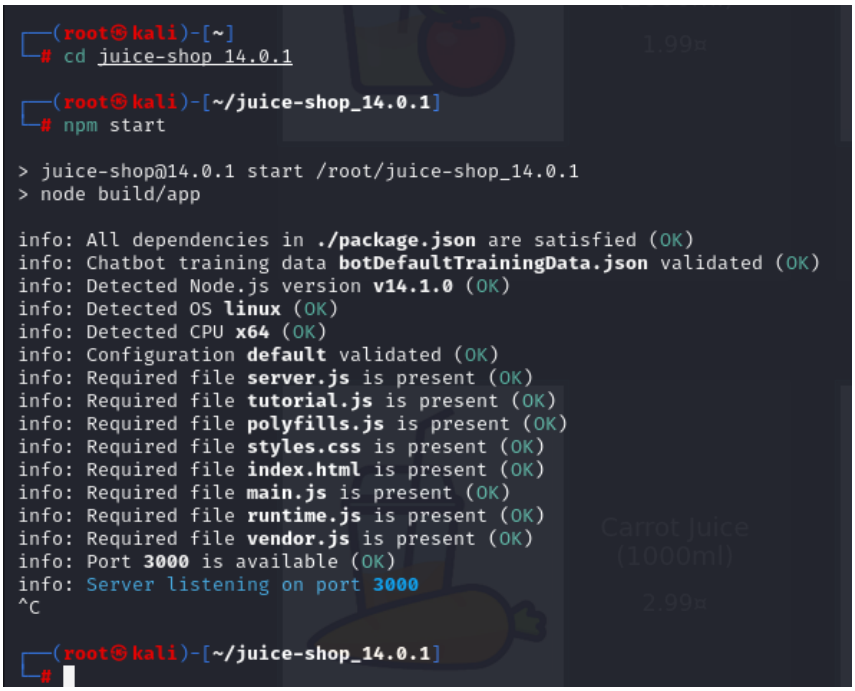
Burpsuite

Burp Suite adalah alat pentesting yang sering digunakan untuk menguji keamanan aplikasi web. Alat ini memiliki berbagai fitur yang dapat membantu dalam mengidentifikasi dan mengeksploitasi kerentanan pada aplikasi web. Burp Suite dapat digunakan untuk memindai aplikasi web dan mengidentifikasi kerentanan, mencoba mengambil alih sesi pengguna, serta melakukan serangan lainnya. Berikut adalah beberapa fungsi utama dari Burp Suite:

1. Intercepting proxy: Burp Suite memiliki fitur intercepting proxy yang memungkinkan pengguna untuk memantau dan memodifikasi data yang dikirimkan antara aplikasi web dan server. Dengan fitur ini, pengguna dapat memodifikasi permintaan dan respon yang dikirimkan antara aplikasi web dan server untuk menguji keamanan aplikasi.
2. Scanner: Burp Suite memiliki fitur scanner yang dapat digunakan untuk melakukan pemindaian (scanning) kerentanan pada aplikasi web. Dengan fitur ini, Burp Suite dapat melakukan pemindaian otomatis pada aplikasi web untuk mengidentifikasi kerentanan seperti SQL injection, cross-site scripting (XSS), dan kerentanan lainnya.
3. Intruder: Burp Suite memiliki fitur Intruder yang dapat digunakan untuk melakukan serangan brute-force atau fuzzing pada aplikasi web. Dengan fitur ini, Burp Suite dapat mengirimkan serangkaian permintaan yang berbeda ke aplikasi web untuk menguji keamanannya.
4. Repeater: Burp Suite memiliki fitur repeater yang memungkinkan pengguna untuk mengirimkan permintaan yang sama berulang kali ke server untuk menguji respons dari server. Dengan fitur ini, pengguna dapat memodifikasi permintaan untuk menguji respons dari server.
5. Collaborator: Burp Suite memiliki fitur Collaborator yang dapat digunakan untuk menguji kerentanan pada aplikasi web yang terhubung dengan sumber eksternal (misalnya, server email, server DNS, dan sebagainya). Dengan fitur ini, pengguna dapat menguji apakah aplikasi web mengirimkan informasi rahasia ke sumber eksternal.
6. Decoder: Burp Suite memiliki fitur decoder yang dapat digunakan untuk memecahkan kode atau enkripsi yang digunakan pada aplikasi web. Dengan fitur ini, pengguna dapat mengidentifikasi jenis enkripsi yang digunakan pada aplikasi web dan melakukan uji coba untuk melihat seberapa mudahnya untuk memecahkan enkripsi tersebut.

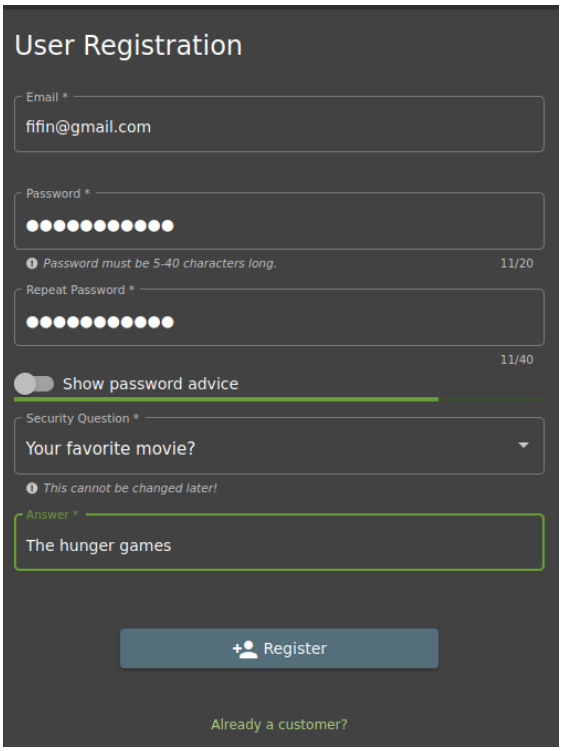
Percobaan 1

Masuk kedalam Website Juice shop dengan menggunakan npm start pada direktori juice shop.



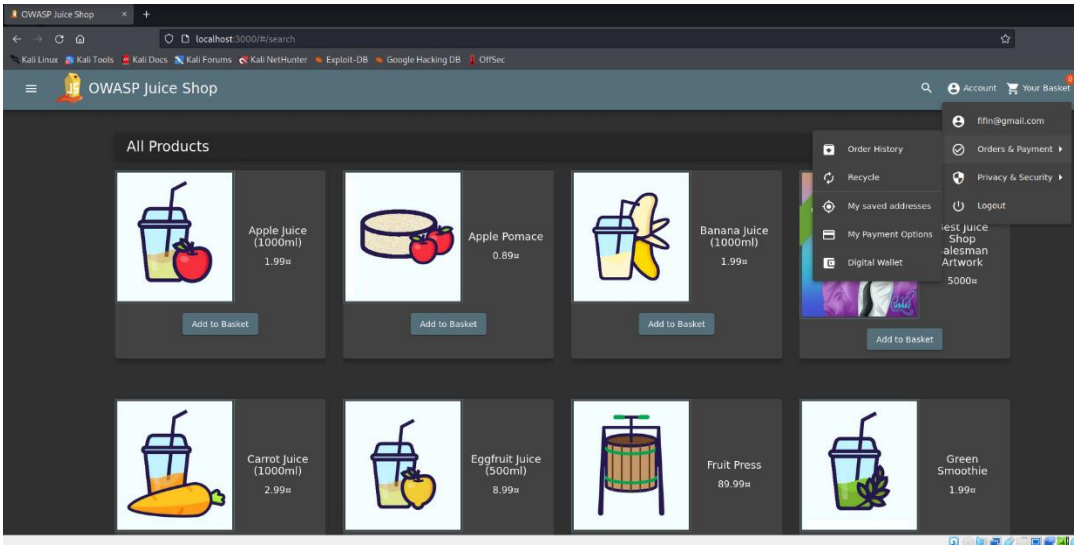
Percobaan 2

Melakukan User Regristration dengan mengisikan email dan password



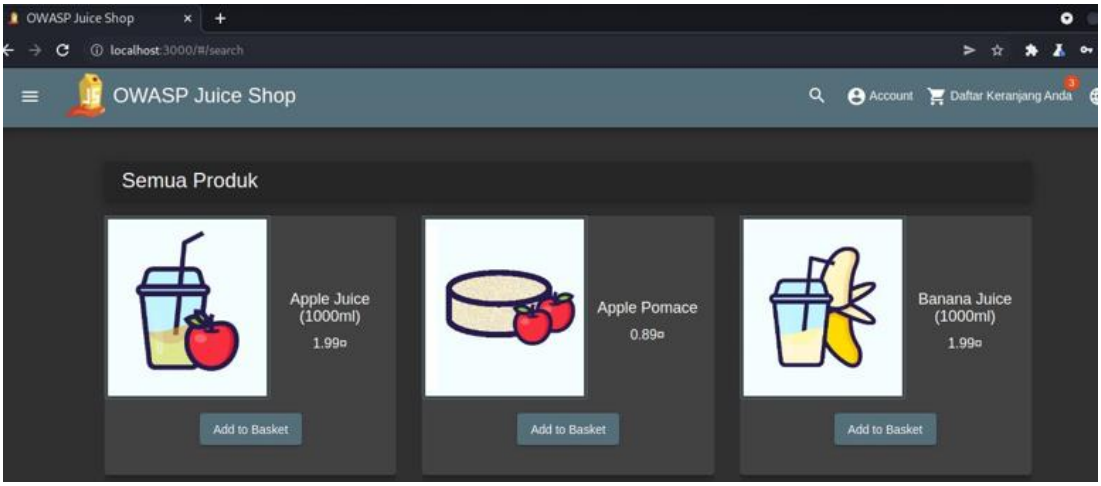
Percobaan 3

Login sebagai email : fifin@gmail.com Setelah melakukan pengaksesan localhost 3000 di terminal dan mempersiapkan aplikasi burpsuite di sini, melakukan login dengan menggunakan akun yang sudah di daftarkan sebelumnya.



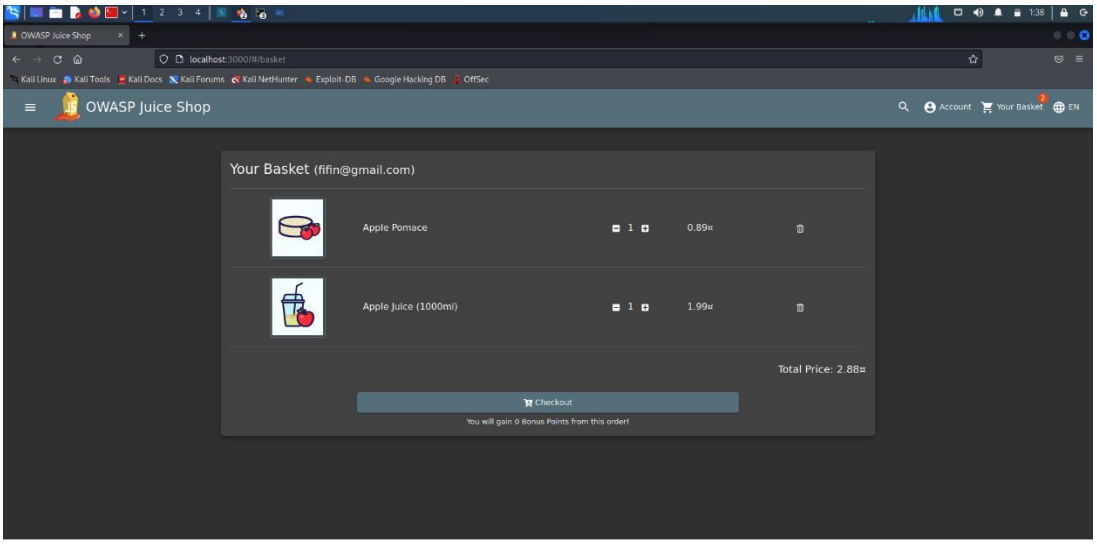
Percobaan 4

Selanjutnya setelah melakukan proses login akan di arahkan ke halaman dashboard untuk dapat memilih beberapa yang tersedia di OWASP Juice Shop ini dan nantinya akan disimpan ke keranjang.



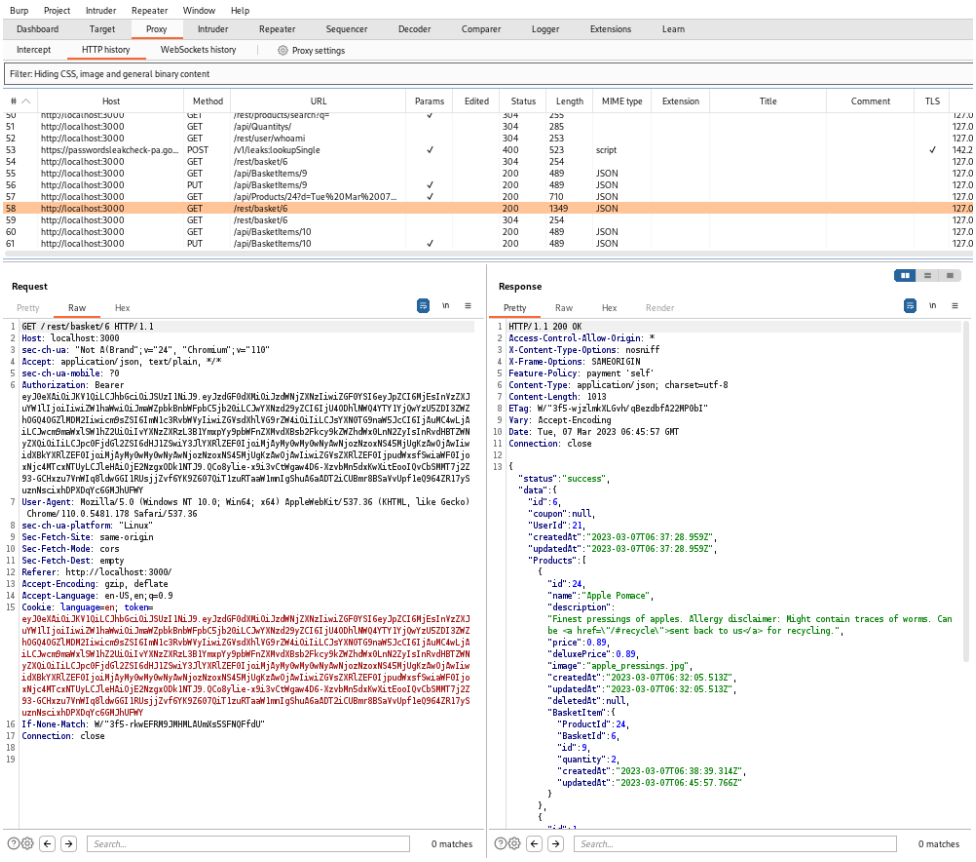
Percobaan 5

Selanjutnya menuju ke halaman keranjang untuk dapat melihat apakah item sudah ditambahkan, jika sudah maka aka nada penambahan aple juice dan apple pomance masing masing 1



Percobaan 6

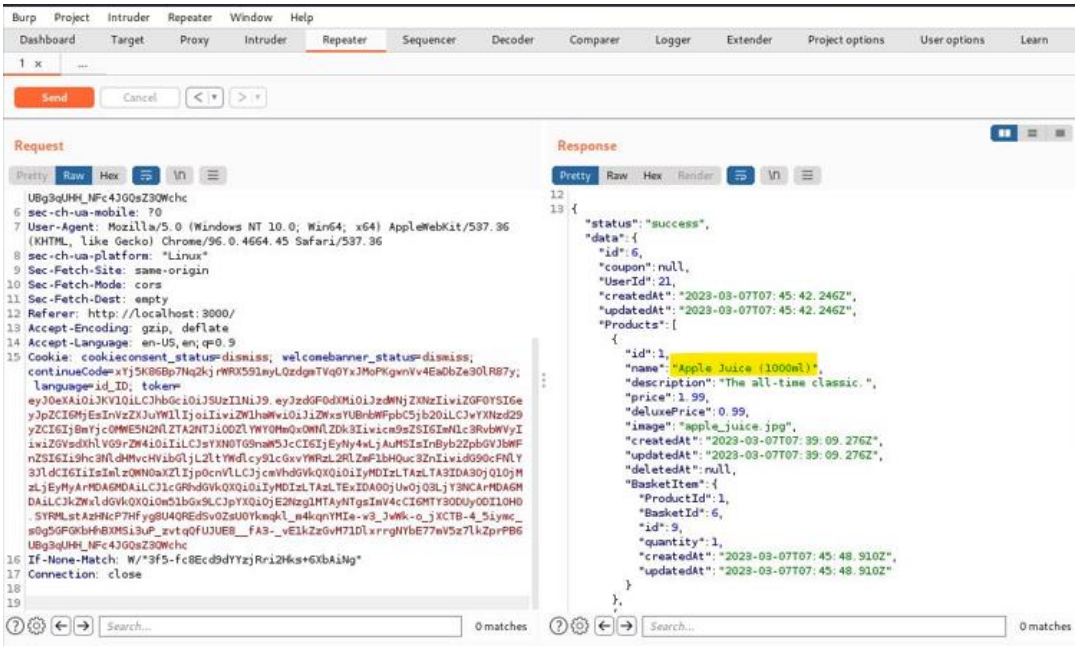
Membuka halaman proxy pada http history yang digunakan untuk melihat track history yang telah dilakukan. Ada beberapa informasi yang tersedia seperti host, method, url, status, length dan informasi pendukung lainnya.

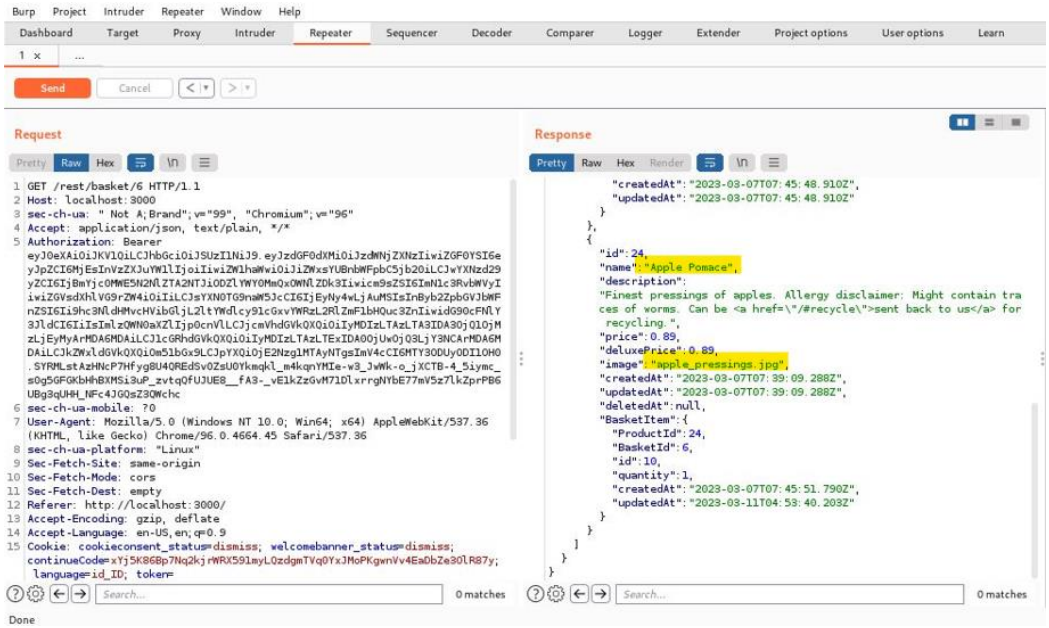


http history→params json dengan get /rest/basket/6. Menghasilkan request dan response yang ada pada halaman response. Response berisikan tentang item yang dipilih pada Juice shop(keranjang yang telah dimasukkan)

Percobaan 7

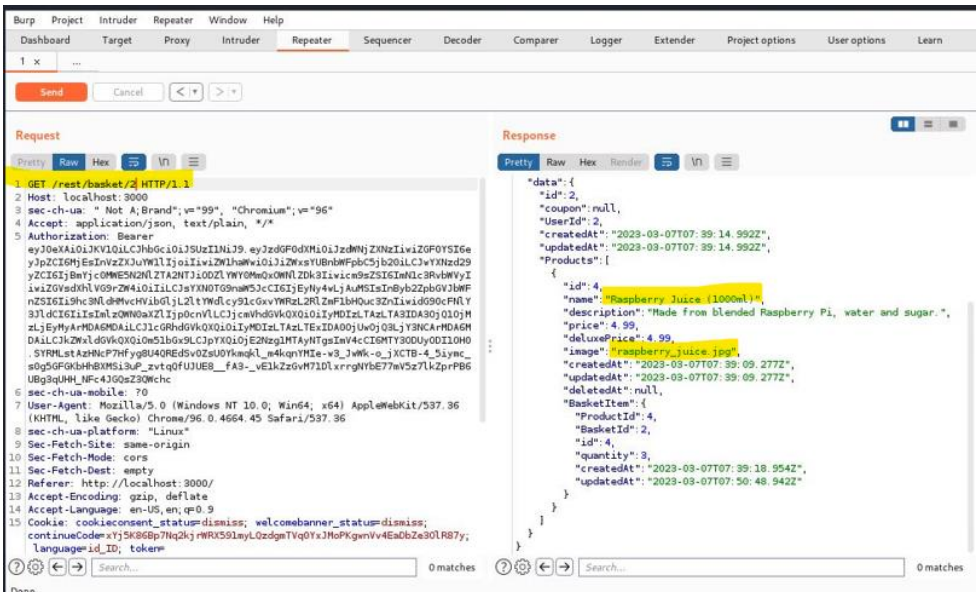
Repeater→Send→ Data Item Keranjang yang telah ditambahkan





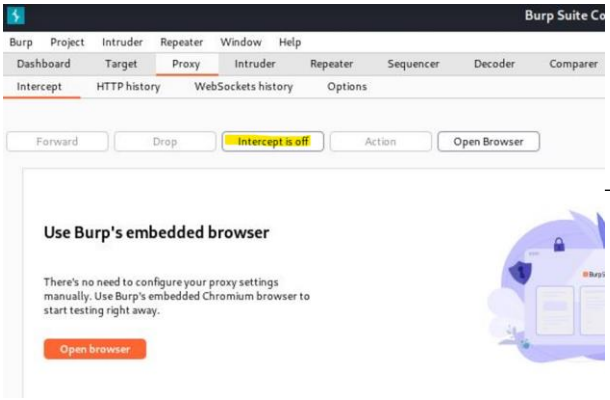
Percobaan 8

Perubahan → (GET /rest/basket/6/HTTP/1.1) menjadi (GET /rest/basket/2/HTTP/1.1)



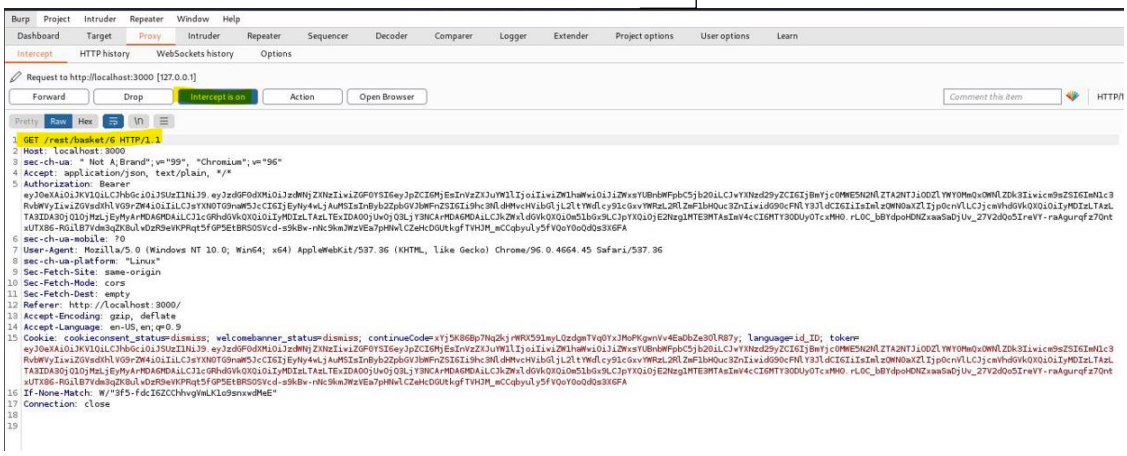
Percobaan 9

Merubah status intercept is off→intercept is on



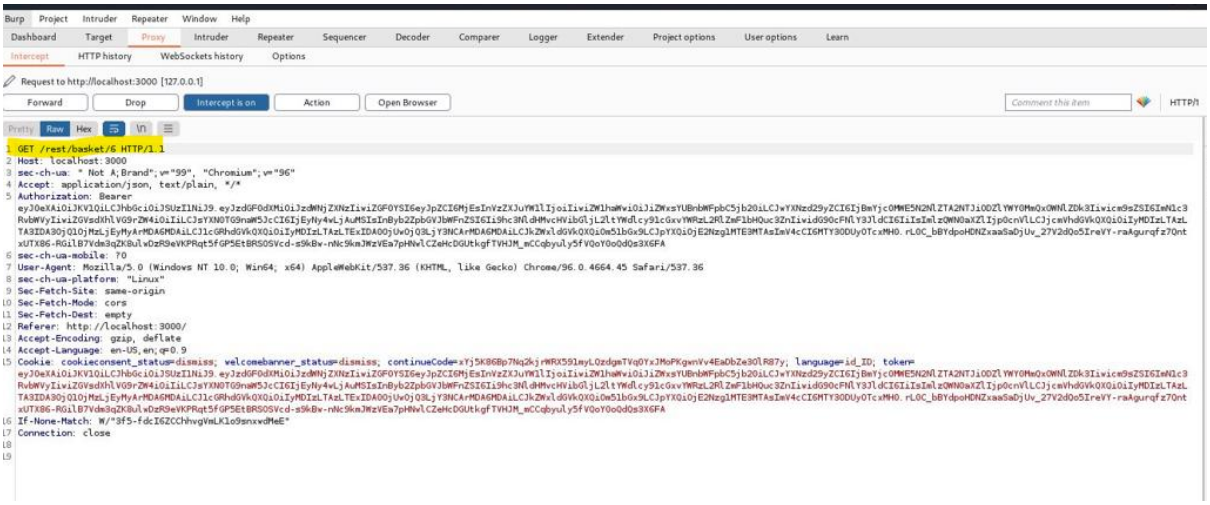
Keadaan Ketika intercept is off

Keadaan Ketika intercept is on



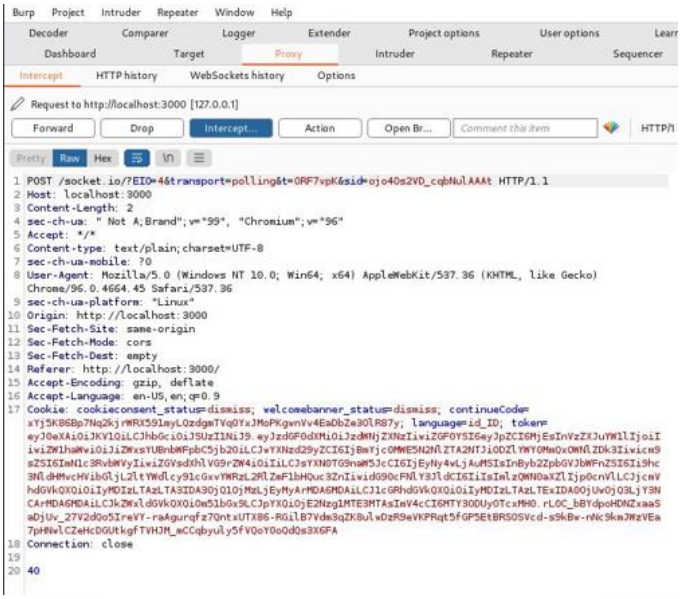
Percobaan 10

data GET /rest/basket/6 forward dengan pergantian id lain



Percobaan 11

Pengecheckan website basket dengan pergantian id 2



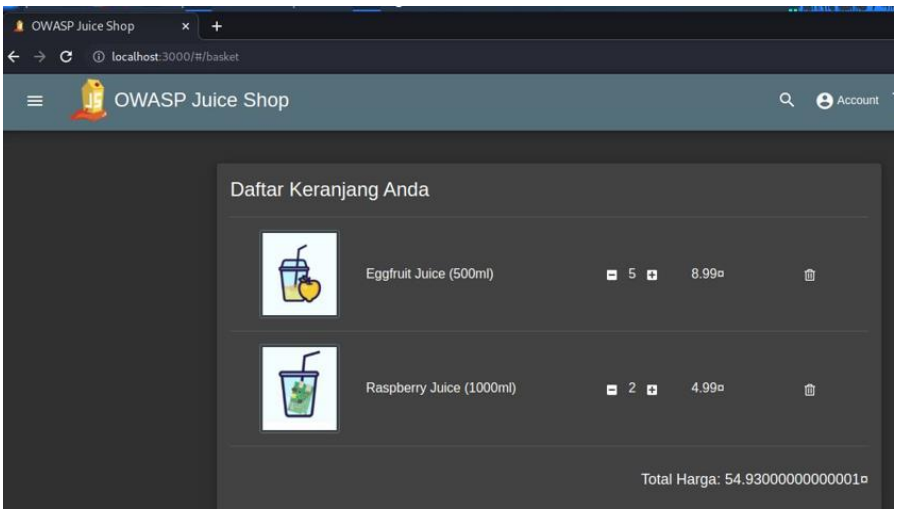
Hasil website juice shop

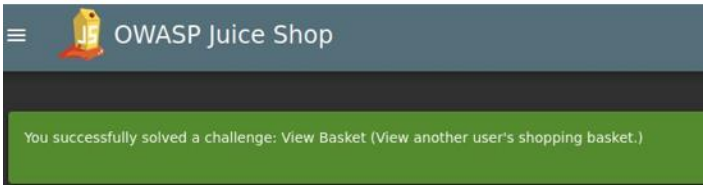
Percobaan 12

Keranjang pengguna dengan id 4

Percobaan 13

Keranjang pengguna dengan id 5 meng

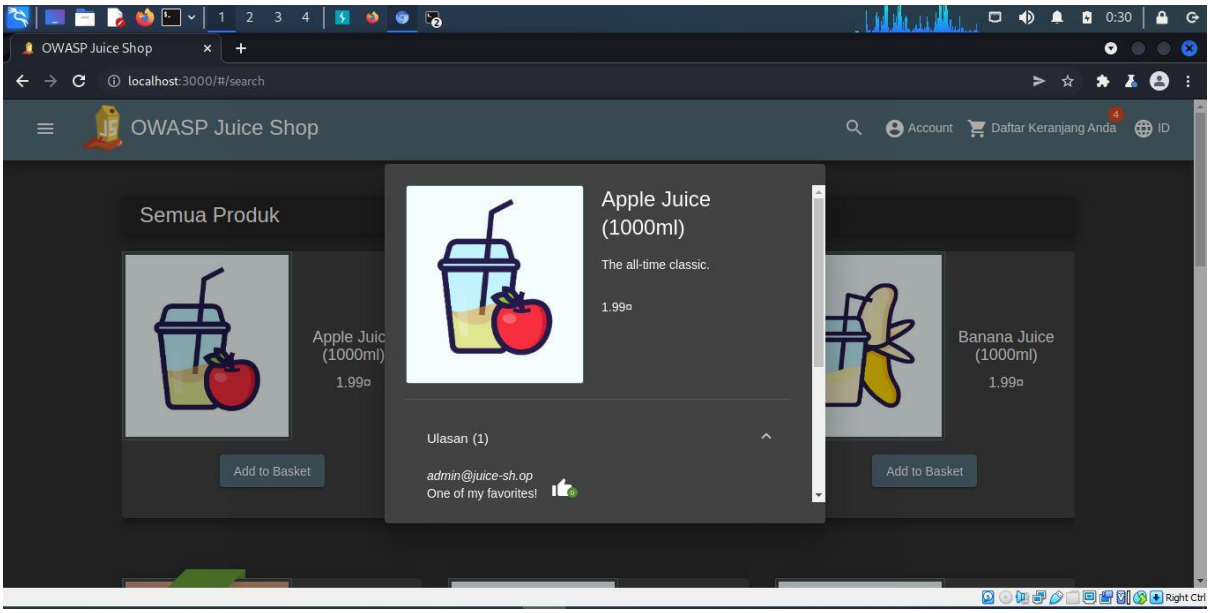




Percobaan : Admin Section

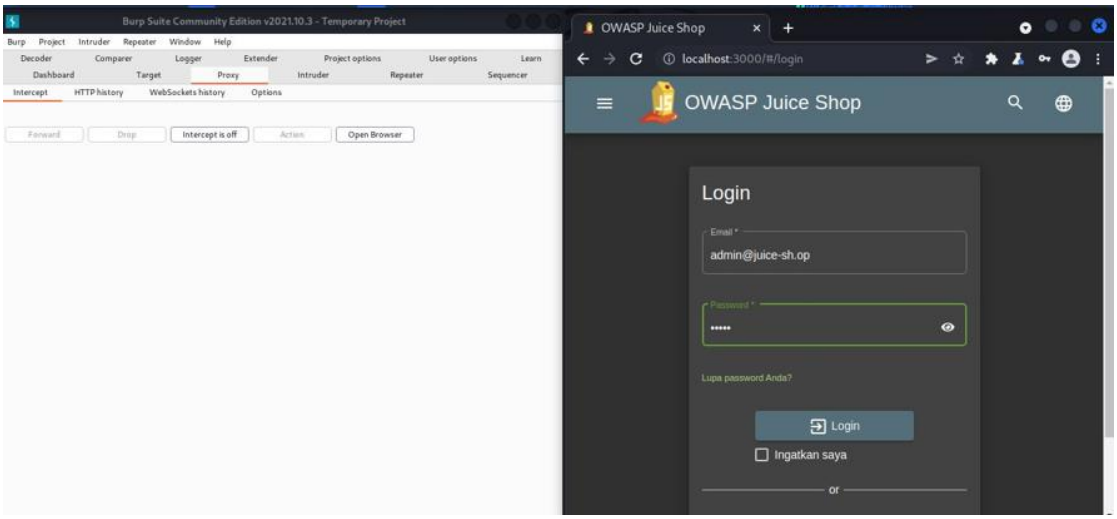
Percobaan 1

Percobaan login dengan menggunakan admin@juice-sh.op



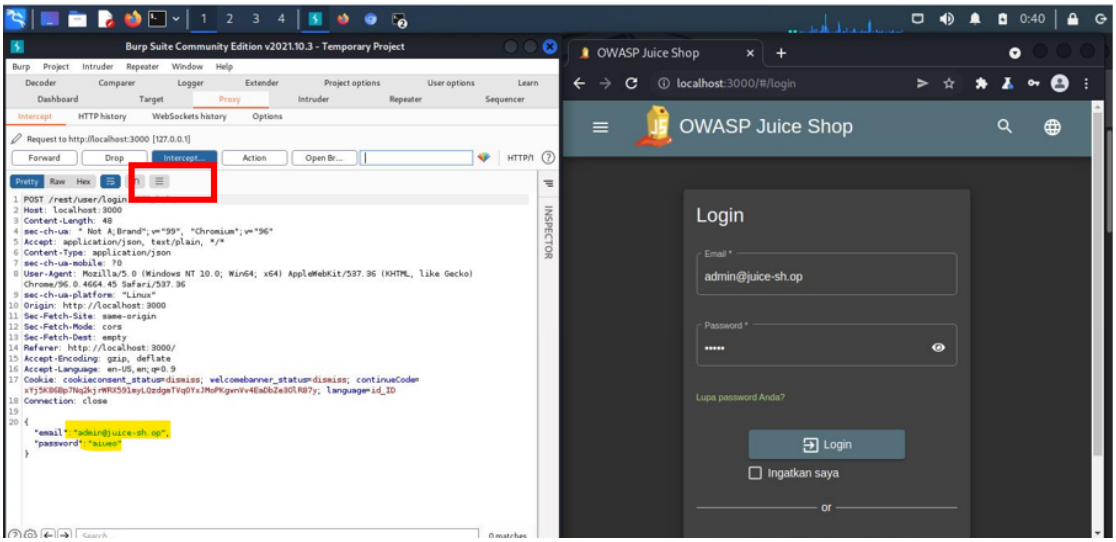
Percobaan 2

Login dengan admin@juice-sh.op dengan password yang telah dibuat random dan mengaktifkan intercept is on



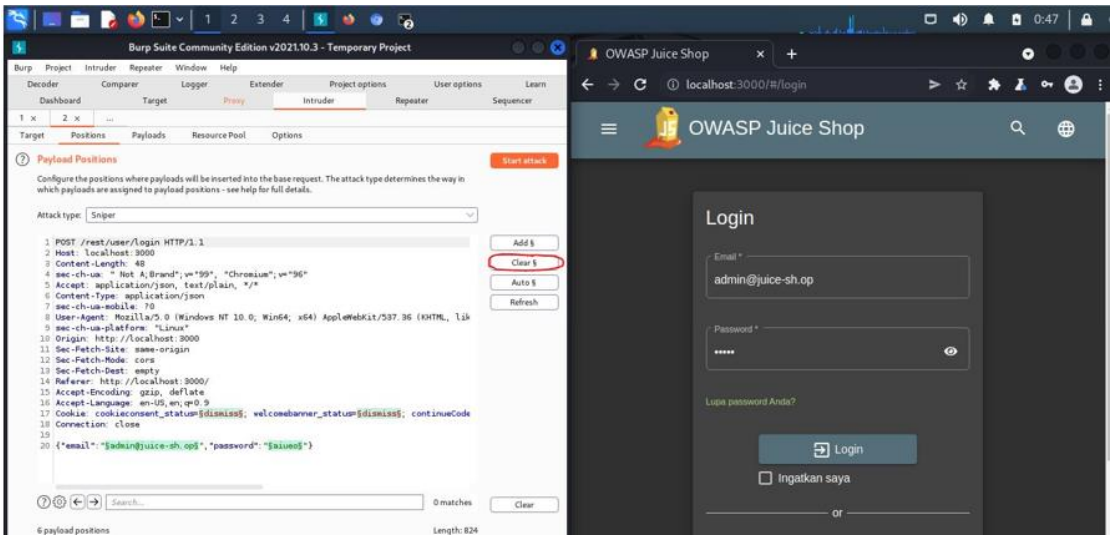
Percobaan 3

Melihat informasi pada proxy-intercept untuk melakukan pengiriman→ intruder,positions



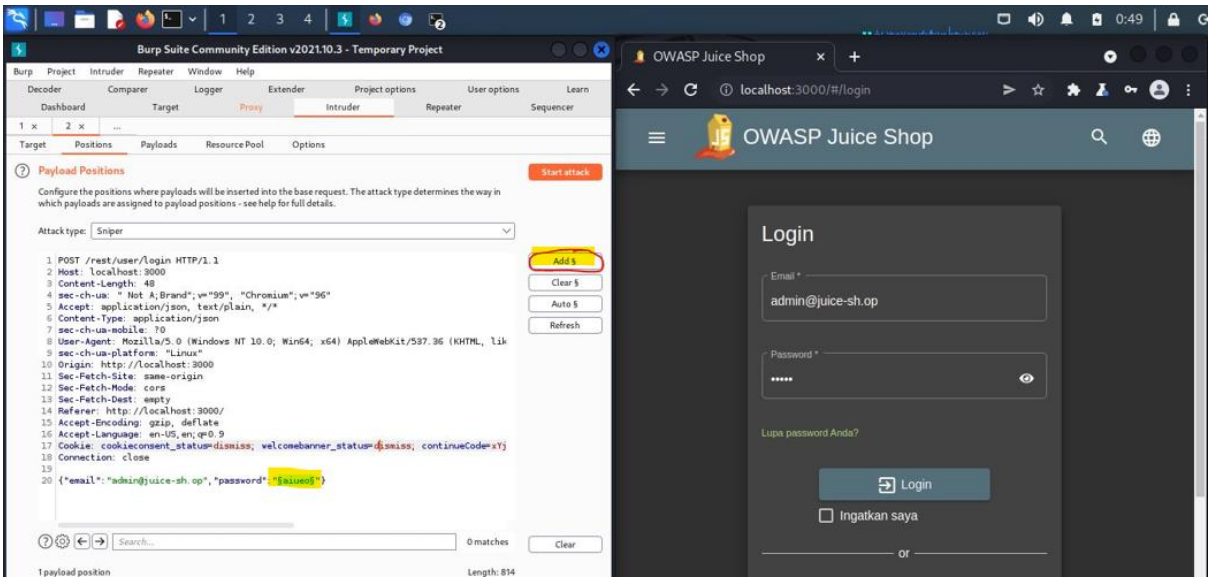
Percobaan 4

Membuka halaman intruder, positions yang digunakan untuk melihat informasi yang ada→klik clear



Percobaan 5

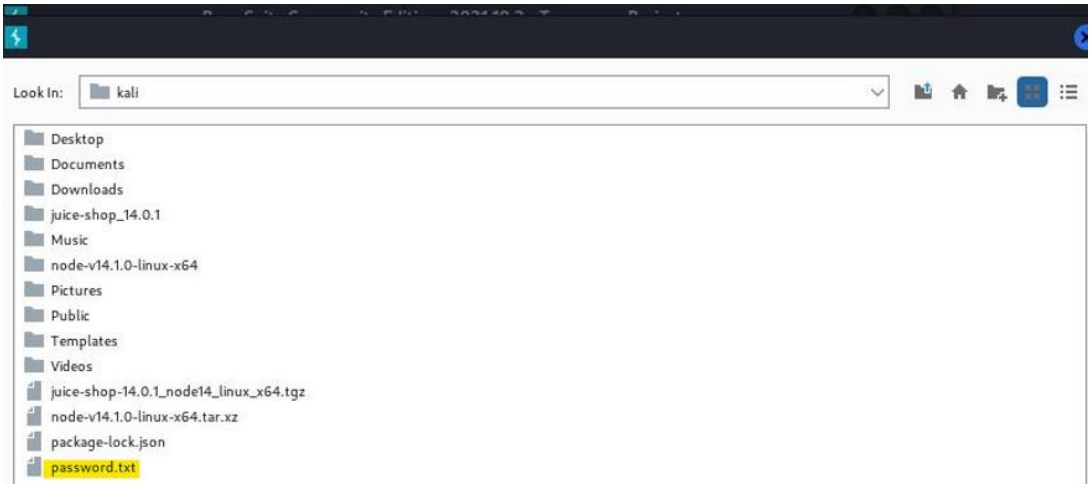
Klik add, maka akan ada tambahan sign dengan bertambahkan dollar dibagian depan. Hal ini membuat adanya perubahan yang ada



Percobaan 6

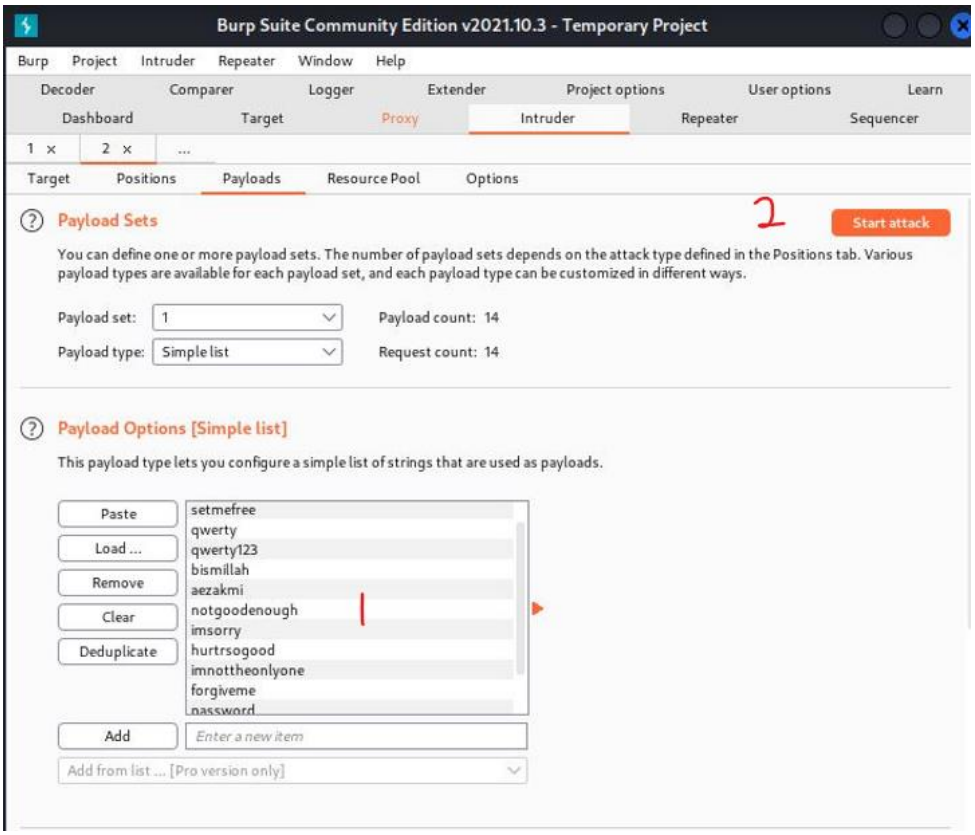
Membuka payloads dan melakukan load data file password.text yang telah kita buat

-



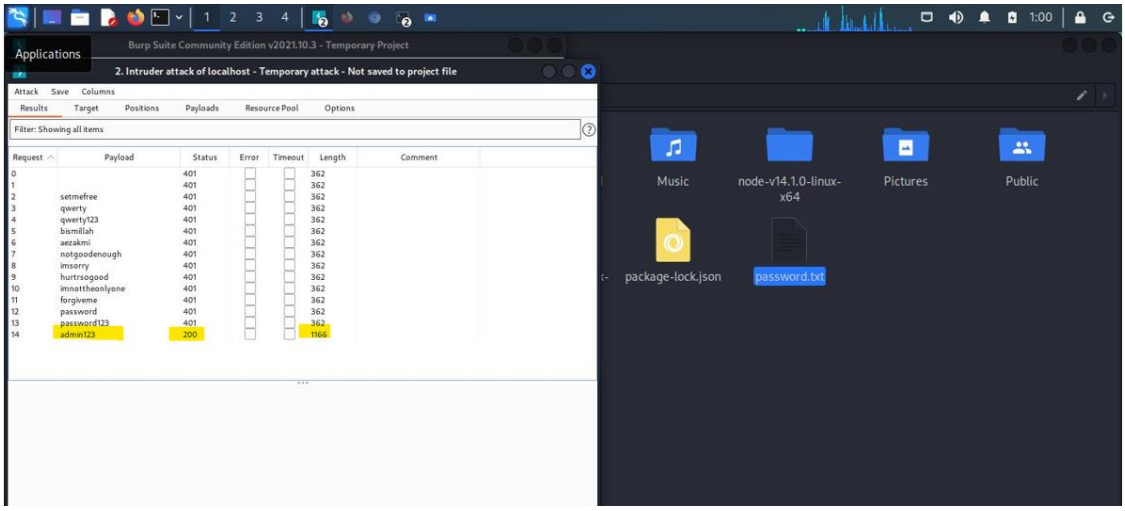
Percobaan 7

Data password.text yang telah kita inputkan kedalam payloads maka data password.text akan muncul pada bagian kotak putih yang berisi list password random yang telah kita buat kemudian clik → **START ATTACK**



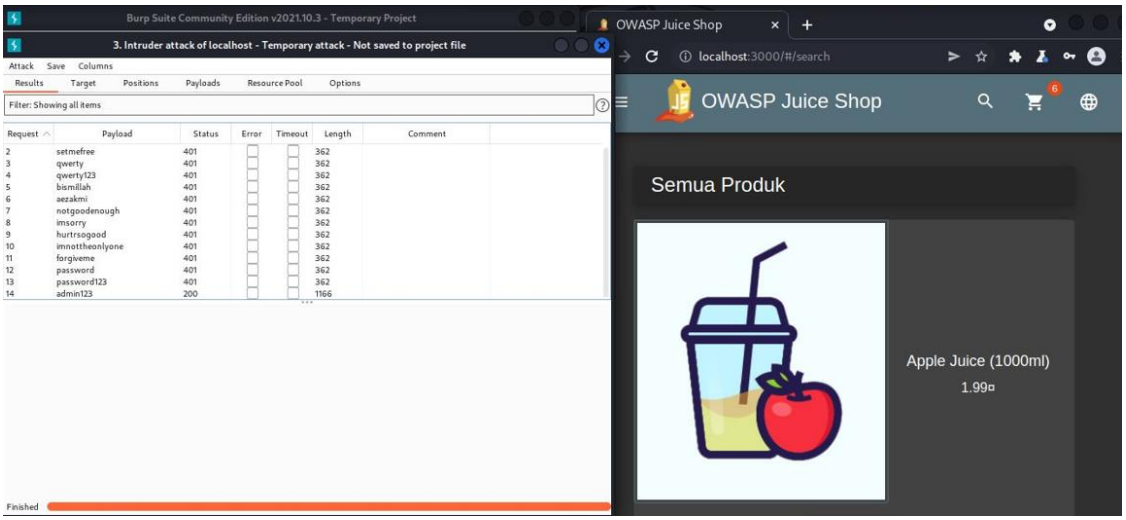
Percobaan 8

Menampilkan list password dan memuat status yang berisi 401 (tidak memiliki autorisasi) pengecualian untuk “admin123” yang memiliki status 200 (memiliki autorisasi)



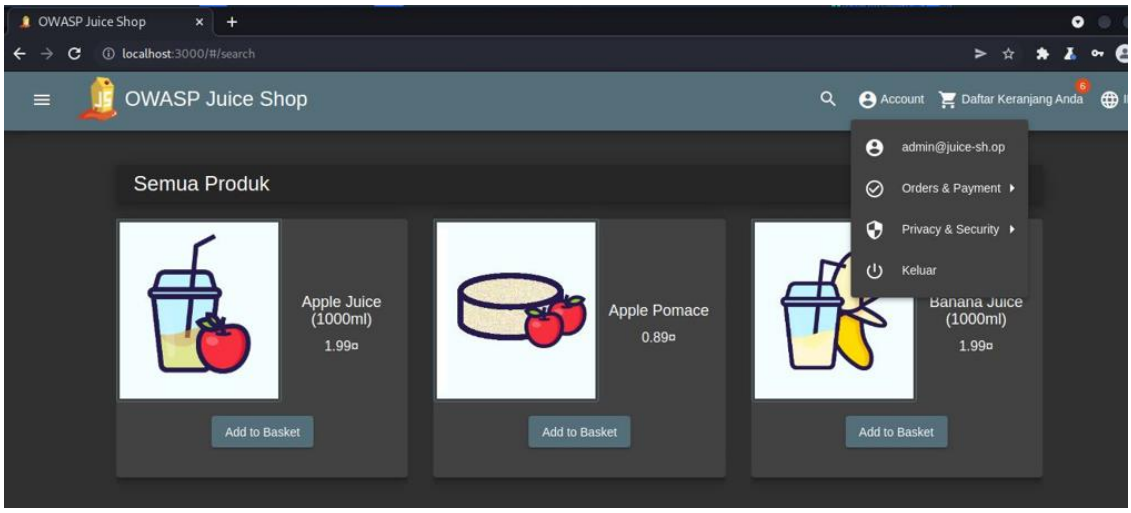
Percobaan 9

Melakukan login menggunakan “admin123” dan success login sebagai admin (dapat mengakses basket keranjang



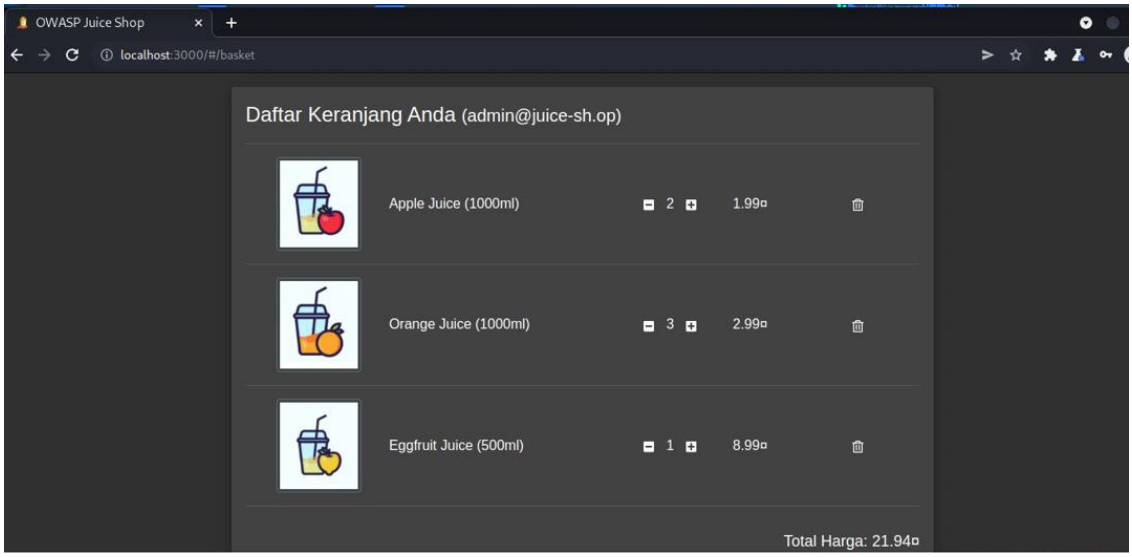
Percobaan 10

Login dengan username admin@juice-sh.op dan berhasil



Percobaan 11

Menampilkan daftar keranjang yang ada dengan email admin@juice-sh.op



Kesimpulan :

Tujuan dilakukannya percobaan Broken Access Control pada daftar kerentanan OWASP Top 10 di aplikasi Juice Shop adalah untuk mengidentifikasi setiap kerentanan yang terkait dengan kontrol akses yang mungkin ada di aplikasi. mengacu pada situasi di mana aplikasi tidak menerapkan pembatasan dengan benar pada apa yang boleh dilakukan atau diakses oleh pengguna yang diautentikasi dalam sistem. Penyerang dapat

mengeksploitasi kerentanan ini untuk mendapatkan akses tidak sah ke informasi sensitif atau melakukan tindakan yang harus dibatasi hanya untuk pengguna yang berwenang.

Dengan melakukan eksperimen pada aplikasi Juice Shop, penguji keamanan dapat mengidentifikasi potensi kerentanan dan menentukan apakah aplikasi menerapkan Control Access dengan benar atau tidak. Eksperimen mungkin melibatkan upaya untuk mengakses sumber daya atau melakukan tindakan yang harus dibatasi untuk pengguna atau peran tertentu, dan mengevaluasi respons aplikasi.

Setelah kerentanan teridentifikasi, tim pengembangan dapat mengambil langkah untuk mengatasinya, seperti menerapkan Control Access yang lebih baik atau memperkuat yang sudah ada. Ini dapat membantu meningkatkan keamanan aplikasi secara keseluruhan dan mengurangi risiko akses tidak sah atau pelanggaran data.