

Security Testing of Mass Assignment Vulnerabilities in RESTful APIs Hands on

Mariano Ceccato

mariano.ceccato@univr.it

Davide Corradini, Michele Pasqua



UNIVERSITÀ
di **VERONA**

Dipartimento
di **INFORMATICA**

1. Nominal & Error testing



Obtain case study

`docker pull davidecorradini94/bookstore`



Run the case study

- Run the bookstore API
`docker run -p 8080:8080 davidecorradini94/bookstore`
- Check the running docker containers
`docker ps`



Open-API specification



Postman <https://www.postman.com/>

Method

URL

Send request button

Input parameters

Response

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 {
2   {
3     "id": 1,
4     "title": "Software Engineering",
5     "author": "Mariano Ceccato",
6     "price": 10.0
7   }
8 }
```



List of books

- Point your browser to
 - <http://localhost:8080/books>
- or

The screenshot shows a REST client interface with the following components:

- Request:** Method: GET, URL: localhost:8080/books. The interface includes tabs for Params, Authorization, Headers (7), Body, Pre-request Script, Tests, Settings, and Cookies.
- Response:** Status: 200 OK, Time: 40 ms, Size: 7.83 KB. The response body is displayed in JSON format.

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

```
639      "id": 121,  
640      "title": "Software Engineering, Vol. 1",  
641      "author": "Mariano Ceccato",  
642      "price": 16.5  
643    },  
644    {  
645      "id": 122,
```



Add a new book

POST new book + ... No Environment

HTTP Petstore / new book Save

POST localhost:8080/book?title=Requirement engineering&author=Mariano&price=10 Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> title	Requirement engineering			
<input checked="" type="checkbox"/> author	Mariano			
<input checked="" type="checkbox"/> price	10			
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 24 ms 240 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 125,
3   "title": "Requirement engineering",
4   "author": "Mariano",
5   "price": 10.0
6 }
```



Read a book

GET read book No Environment

HTTP Petstore / read book Save

GET localhost:8080/book/125 Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 9 ms 240 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 125,
3   "title": "Requirement engineering",
4   "author": "Mariano",
5   "price": 10.0
6 }
```



Update a book

10

The screenshot shows a REST client interface for a PUT request. The request is sent to `localhost:8080/book` with a JSON body. The response is a `200 OK` status with a response time of `7 ms` and a body size of `241 B`. The response body is displayed in a pretty-printed JSON format.

```
PUT localhost:8080/book
```

```
{
  "id": 125,
  "title": "Requirement engineering",
  "author": "Mariano",
  "price": 100.0
}
```

Body: `200 OK 7 ms 241 B`

```
{
  "id": 125,
  "title": "Requirement engineering",
  "author": "Mariano",
  "price": 100.0
}
```



Delete a book

DEL delete book + ... No Environment

HTTP Petstore / delete book Save

DELETE localhost:8080/book?id=125 Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Description	... Bulk Edit
<input checked="" type="checkbox"/>	id	125		
	Key	Value	Description	

Body Cookies Headers (4) Test Results 200 OK 15 ms 123 B Save as Example

Pretty Raw Preview Visualize Text

1



Obtain RestTestGen

- Clone the official public source code repository

```
git clone https://github.com/SeUniVr/RestTestGen  
cd RestTestGen
```

```
docker build -t rtg .
```

With docker

```
sudo chmod +x gradlew  
./gradlew build
```

With gradle on linux/mac

```
./gradlew.bat build
```

With gradle on Windows



Run RestTestGen

- Edit the configuration in file `rtg-config.yml`

```
apiUnderTest: bookstore  
strategyClassName: NominalAndErrorStrategy
```

```
docker run -v ./:/app --network="host" rtg
```



Output

- **CoverageReports:** according to several metrics
- **Report:** interaction sequences
 - NominalFuzzer
 - ErrorFuzzer
- **REST-assured:** estarnally runnable test cases
 - NominalFuzzer
 - ErrorFuzzer



2. Security testing



Obtain case study

`docker pull davidecorradini94/vampi-vuln`



Run the case study

- Run the bookstore API

```
docker run -p 5000:5000 davidecorradini94/vampi-vuln
```

- Check the running docker containers

```
docker ps
```



Open-API specification



Read all the users

The screenshot shows a REST client interface with the following details:

- Request:** Method: GET, URL: http://localhost:5000/users/v1/_debug
- Response:** Status: 200 OK, Time: 11 ms, Size: 1.54 KB
- Response Body (JSON):**

```
1 {
2   "users": [
3     {
4       "admin": false,
5       "email": "mail@mail.com",
6       "password": "pass2",
7       "username": "name1"
8     }
9   ]
10 }
```



Create new user

The screenshot displays a REST client interface for a POST request. The request is named "register user" and is sent to the URL "http://localhost:5000/users/v1/register". The request body is a JSON object with the following fields: "email": "mariano@mail.com", "password": "mysecret", and "username": "mariano". The response is a 200 OK status with a response time of 16 ms and a body size of 257 B. The response body is a JSON object with "message": "Successfully registered. Login to receive an auth token." and "status": "success".

POST register user

HTTP vampi / register user

POST http://localhost:5000/users/v1/register

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "email": "mariano@mail.com",
3   "password": "mysecret",
4   "username": "mariano"
5 }
```

Body Cookies Headers (5) Test Results 200 OK 16 ms 257 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Successfully registered. Login to receive an auth token.",
3   "status": "success"
4 }
```



Create user with mass assignment

21

The screenshot displays a REST client interface for a POST request. The request is sent to `http://localhost:5000/users/v1/register` with the following JSON body:

```
1 {
2   "email": "mariano@mail.com",
3   "password": "mysecret",
4   "username": "mariano-under-attack",
5   "admin": true
6 }
```

The response is a 200 OK status with a response time of 34 ms and a body size of 257 B. The response body is shown in the "Body" tab, formatted as JSON:

```
1 {
2   "message": "Successfully registered. Login to receive an auth token.",
3   "status": "success"
4 }
```



Run RestTestGen

- Edit the configuration in file `rtg-config.yml`

```
apiUnderTest: vampi  
strategyClassName: MassAssignmentSecurityTestingStrategy
```

```
docker run -v ./:/app --network="host" rtg
```



Output

- **CRUDgroups**: result of clustering by resource type
- **Report**: interaction sequences
 - NominalFuzzer+MassAssignmentFuzzer+UserInstantiated



New feature: GUI





Welcome in RTG

Select the Api and the strategy in the the top left corner and click Start



Settings

[RTG Settings](#)[API Settings](#)[Strategy Settings](#)

Select the API

Settings

RTG Settings

API Settings

Strategy Settings

Select the API

VAmPI

ART Store

petstore-vuln

Book Store

Google Drive

Spotify

Pet Store

VAmPI

Host

localhost

Reset Command

echo reset

Reset Before Testing

Authentication Commands:

Save

Settings

RTG Settings

API Settings

Strategy Settings

Select the API



VAmPI

ART Store

petstore-vuln

Book Store

Google Drive

Spotify

Pet Store

Add new API



Openapi specification (.json)

Choose File

No file chosen

Name

Host

Reset Command

 Reset Before Testing

Add Authentication Script and command

Choose File

No file chosen

default

```
echo {"name": "apikey", "value": "davide", "in": "query", "duration": 6000}
```

Submit



Settings Reset

APIs



Strategy



Test Sequence

Test Iteration

Coverage

Operations Graph

Test Sequence



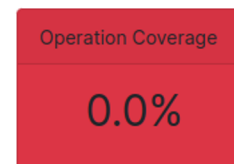
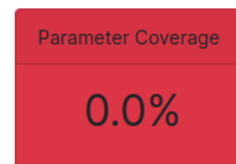
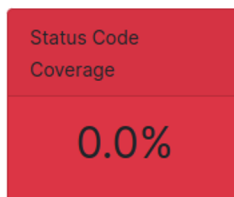
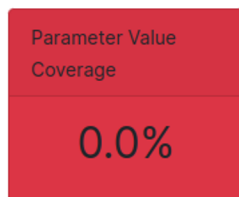
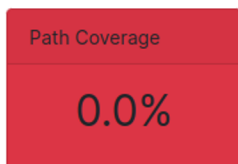
Number	Host	Method	Url	Status	Length	Port	Time
--------	------	--------	-----	--------	--------	------	------

Request Body

Response Body



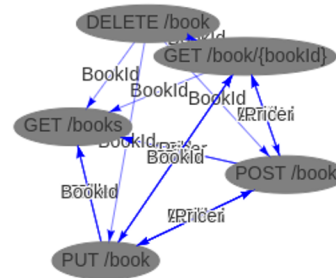
Coverage



Results:

[VAmPI](#)[ART Store](#)[petstore-vuln](#)[Book Store](#)[Google Drive](#)[Spotify](#)[Pet Store](#)

Operations Dependency Graph





Number	Host	Method	Url	Status	Length	Port	Time
3	http://localhost/	GET	/users/v1/_debug	200	1978	5002	17:44:35
10	http://localhost/	POST	/users/v1/register	200	92	5002	17:44:35
15	http://localhost/	GET	/users/v1/_debug	200	2096	5002	17:44:35
20	http://localhost/	PUT	/users/v1/X/email	401	47	5002	17:44:35
25	http://localhost/	PUT	/users/v1/X/email	401	47	5002	17:44:35
30	http://localhost/	PUT	/users/v1/X/email	401	47	5002	17:44:35
35	http://localhost/	PUT	/users/v1/X/email	401	47	5002	17:44:35
40	http://localhost/	PUT	/users/v1/X/email	401	47	5002	17:44:35
45	http://localhost/	PUT	/users/v1/X/email	401	47	5002	17:44:35
50	http://localhost/	PUT	/users/v1/X/email	401	47	5002	17:44:35
55	http://localhost/	PUT	/users/v1/X/email	401	47	5002	17:44:35
60	http://localhost/	PUT	/users/v1/X/email	401	47	5002	17:44:35
65	http://localhost/	PUT	/users/v1/X/email	401	47	5002	17:44:35
70	http://localhost/	PUT	/users/v1/X/email	401	47	5002	17:44:35
75	http://localhost/	PUT	/users/v1/X/email	401	47	5002	17:44:35

Request Body Response Body



Test Sequence

PASS

Sequence GlobalSequenceForDebugPurposes

**PASS**

Sequence back-pedal-Erbe

**PASS**

Sequence ordurous-idiometer

**PASS**

Sequence annihilationistical-golfed

**PASS**

Sequence Jaimie-misconstrual

**PASS**

Sequence unsexually-colonialists

**PASS**

Sequence WELL-purple-black

**PASS**

Sequence sodium-vapor-GGP

**PASS**

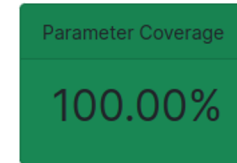
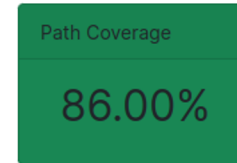
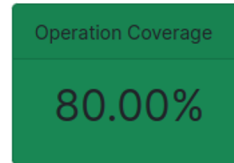
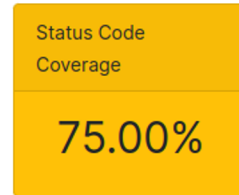
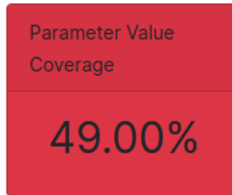
Sequence ungenitured-KKt

**PASS**

Sequence LA-snow-melting



Coverage



Results:

[ART Store](#)[Spotify](#)[Book Store](#)[VAmPI](#)[Pet Store](#)[petstore-vuln](#)[Google Drive](#)

Operations Dependency Graph

