

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



UNIVERSITÀ
di VERONA

Dipartimento
di INFORMATICA

Obtain case study

`docker pull davidecorradini94/bookstore`



UNIVERSITÀ
di VERONA

Dipartimento
di INFORMATICA

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
Input
parameters

Send
request
button

Response

The screenshot shows the Postman application window. At the top, there's a toolbar with 'File', 'Edit', 'View', 'Help' menus, and various icons for 'New', 'Import', 'Runner', and 'Invite'. Below the toolbar, a header bar displays the URL 'Fundamentals of Software Engineering ...' and a status message 'No Environment'. The main area is titled 'Untitled Request' and shows a 'Method' dropdown set to 'GET' and a 'URL' input field containing 'localhost:8080/books'. To the right of the URL input is a large blue 'Send' button. Below the URL input, there's a table for 'Query Params' with columns for 'KEY', 'VALUE', 'DESCRIPTION', and 'Bulk Edit'. In the bottom section, there's a 'Body' tab showing a JSON response:

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

The response body is displayed in 'Pretty' format. At the bottom of the interface, there are tabs for 'Body', 'Cookies', 'Headers (5)', and 'Test Results'. On the right side, there are status details: 'Status: 200 OK', 'Time: 12 ms', and 'Size: 245 B'. A 'Save Response' button is also present. The bottom navigation bar includes 'Find and Replace', 'Console', 'Bootcamp', 'Build', 'Browse', and a help icon.



List of books

7

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

- or

The screenshot shows the Postman application interface. At the top, there is a search bar with the text "GET read all books" and a "No Environment" dropdown. Below the search bar, the URL "localhost:8080/books" is entered into a "Petstore / read all books" field under the "HTTP" tab. To the right of the URL are "Save", "Edit", and "Copy" buttons. The main area shows a table for "Query Params" with two columns: "Key" and "Value". Below this, tabs for "Body", "Cookies", and "Headers (5)" are visible. The "Body" tab is selected and displays a JSON response:

```
559   "id": 121,
560   "title": "Software Engineering, Vol. 1",
561   "author": "Mariano Ceccato",
562   "price": 16.5
563 },
564 {
565   "id": 122,
```



UNIVERSITÀ
di VERONA

Dipartimento
di INFORMATICA

Add a new book

The screenshot shows the Postman application interface. At the top, it displays a header with 'POST new book' and a status bar indicating 'No Environment'. Below the header, the URL 'localhost:8080/book?title=Requirement engineering&author=Mariano&price=10' is entered in the 'POST' request field. The 'Params' tab is selected, showing three parameters: 'title' (Requirement engineering), 'author' (Mariano), and 'price' (10). The 'Body' tab is selected at the bottom, displaying a JSON response:

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



Read a book

The screenshot shows the Postman application interface. At the top, there is a header bar with the text "GET read book", a red circular icon, a plus sign, and three dots. To the right of the header is a dropdown menu labeled "No Environment". Below the header, the URL "localhost:8080/book/125" is entered into the "Send" field. The "Params" tab is selected, showing a table for "Query Params" with one row containing "Key" and "Value". The "Body" tab is selected in the main content area, displaying a JSON response:

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

At the bottom of the interface, there are tabs for "Body", "Cookies", "Headers (5)", and "Test Results". On the far right, there are status indicators: a globe icon, "200 OK", "9 ms", "240 B", a "Save as Example" button, and three dots.



Update a book

The screenshot shows the Postman application interface for making a **PUT** request to update a book. The request URL is `localhost:8080/book`. The **Body** tab is selected, displaying the following JSON payload:

```
1 {
2   "id": 125,
3   "title": "Requirement engineering",
4   "author": "Mariano",
5   "price": 100.0
6 }
```

The response status is **200 OK** with a **7 ms** latency and **241 B** size. The response body is identical to the request body.



Delete a book

The screenshot shows the Postman application interface. At the top, there is a header bar with tabs for 'DEL delete book' (selected), '+', '...', 'No Environment', and a dropdown arrow. Below the header, the URL 'localhost:8080/book?id=125' is entered into the 'DELETE' request field. To the right of the URL are 'Save' and 'Edit' buttons. Below the request field, there are tabs for 'Params' (selected), 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', 'Settings', and 'Cookies'. Under the 'Params' tab, there is a table titled 'Query Params' with two rows:

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

At the bottom of the interface, there are tabs for 'Body', 'Cookies', 'Headers (4)', and 'Test Results' (selected). Below these tabs, there are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', 'Text' (with a dropdown arrow), and 'Copy' (with a clipboard icon). To the right of the preview area, there is a status indicator showing '200 OK 15 ms 123 B', a 'Save as Example' button, and a '...' button. The preview area itself is currently empty.



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**: externally 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

18



UNIVERSITÀ
di VERONA

Dipartimento
di INFORMATICA

Read all the users

The screenshot shows the Postman application interface. At the top, there is a header bar with the title "GET all users all data" and a "No Environment" dropdown. Below the header, the URL "vampi / all users all data" is displayed under the "HTTP" method. A "Save" button and a "Send" button are visible. The main workspace contains a "GET" request with the URL "http://localhost:5000/users/v1/_debug". Below the request, tabs for "Params", "Authorization", "Headers (7)", "Body", "Pre-request Script", "Tests", "Settings", and "Cookies" are present, with "Params" being the active tab. Under "Query Params", there is a table with columns: Key, Value, Description, and Bulk Edit. The table has one row with the key "Key" and value "Value". In the bottom section, tabs for "Body", "Cookies", "Headers (5)", and "Test Results" are shown, with "Body" being the active tab. The "Headers (5)" tab shows the following headers:

- Content-Type: application/json
- Accept: application/json
- User-Agent: PostmanRuntime/7.29.0
- Host: localhost:5000
- Connection: keep-alive

The "Test Results" tab displays a successful response with status code 200 OK, 11 ms duration, and 1.54 KB size. The response body is shown in JSON format:

```
1 {  
2   "users": [  
3     {  
4       "admin": false,  
5       "email": "mail1@mail.com",  
6       "password": "pass2",  
7       "username": "name1"  
8     }  
9   ]  
10 }  
11 }
```



Create new user

The screenshot shows the Postman application interface. At the top, there is a header bar with a search field containing "POST register user", a "+" button, and three dots. To the right of the search field is a dropdown menu set to "No Environment". Below the header, the URL "HTTP vampi / register user" is displayed, along with a "Save" button and a trash bin icon.

The main workspace shows a POST request to "http://localhost:5000/users/v1/register". The "Body" tab is selected, showing a JSON payload:

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

Below the body, the response status is shown as 200 OK with 16 ms and 257 B. The "Pretty" tab is selected in the results panel, displaying the response message:

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



UNIVERSITÀ
di VERONA

Dipartimento
di INFORMATICA

Create user with mass assignment

The screenshot shows a Postman collection named "register user injection" with one item. The item has the method set to "POST" and the URL "http://localhost:5000/users/v1/register". The "Body" tab is selected, showing a JSON payload:

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

The response tab shows a 200 OK status with the message: "Successfully registered. Login to receive an auth token.", "status": "success".



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



UNIVERSITÀ
di VERONA

Dipartimento
di INFORMATICA



Settings Reset

APIs



Strategy



Test Sequence Test Iteration Coverage Operations Graph

Welcome in RTG

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



UNIVERSITY
VERONA



Settings Reset

APIs



Strategy



Test Sequence Test Iteration Coverage Operations Graph

Settings

RTG Settings

API Settings

Strategy Settings

Select the API 

VAmPI

ART Store

petstore-vuln

Book Store

Google Drive

Spotify

Pet Store



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

 RTG Settings Reset

APIs ▾

Strategy ▾



Test Sequence

Test Iteration

Coverage

Operations Graph

Test Sequence





Settings

Reset

APIs

Strategy



Test Sequence

Test Iteration

Coverage

Operations Graph

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

Request Body

Response Body



 RTG Settings Reset

APIs

Strategy



Test Sequence Test Iteration Coverage Operations Graph

Coverage

Path Coverage

0.0%

Parameter Value
Coverage

0.0%

Status Code
Coverage

0.0%

Parameter Coverage

0.0%

Operation Coverage

0.0%

Results:

VAmPI

ART Store

petstore-vuln

Book Store

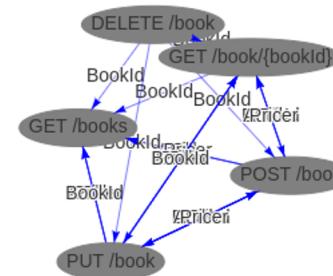
Google Drive

Spotify

Pet Store



Operations Dependency Graph



RTG		Settings	Reset	APIs	Strategy	Test Sequence	Test Iteration	Coverage	Operations Graph
Number	Host	Method	Url						
3	http://localhost/	GET	/users/v1/_debug					200	1978
10	http://localhost/	POST	/users/v1/register					200	92
15	http://localhost/	GET	/users/v1/_debug					200	2096
20	http://localhost/	PUT	/users/v1/X/email					401	47
25	http://localhost/	PUT	/users/v1/X/email					401	47
30	http://localhost/	PUT	/users/v1/X/email					401	47
35	http://localhost/	PUT	/users/v1/X/email					401	47
40	http://localhost/	PUT	/users/v1/X/email					401	47
45	http://localhost/	PUT	/users/v1/X/email					401	47
50	http://localhost/	PUT	/users/v1/X/email					401	47
55	http://localhost/	PUT	/users/v1/X/email					401	47
60	http://localhost/	PUT	/users/v1/X/email					401	47
65	http://localhost/	PUT	/users/v1/X/email					401	47
70	http://localhost/	PUT	/users/v1/X/email					401	47
75	http://localhost/	PUT	/users/v1/X/email					401	47

Request Body

Response Body





Settings Reset

APIs

Strategy



Test Sequence Test Iteration Coverage Operations Graph

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 ungentured-KKt



PASS

Sequence LA-snow-melting



 RTG Settings Reset

APIs

Strategy



Test Sequence Test Iteration Coverage Operations Graph

Coverage

Parameter Value
Coverage

49.00%

Status Code
Coverage

75.00%

Operation Coverage

80.00%

Path Coverage

86.00%

Parameter Coverage

100.00%

Results:

ART Store

Spotify

Book Store

VAmPI

Pet Store

petstore-vuln

Google Drive

Operations Dependency Graph

