Message-Level Security WS 2025/26
Exercise 02-Task

Ruhr University Bochum Faculty of Computer Science
Chair for Network and Data Security

Beginning of Exercise:	Monday 27th October, 2025, 14:00
Processing Time:	14 days
Submission Deadline:	Monday 10th November, 2025, 14:00

Remarks

- The new exercises will be published on **every second Monday** before the lecture.
- The solutions must be submitted via Moodle. Please answer the questions on the first page of the Moodle test.
- You may work together in groups with **two** other persons. Each person in the group must submit the Moodle test **individually** and provide the answers to the first page of the Moodle test. Within each group, only **one** person submits the solutions of the exercise in the Moodle test.
- Please note the announcements and forums via Moodle due to possible postponements and failures of the lecture or exercise.

Exercises

Overview of eHacking REST APIs

We prepared a collection of REST APIs that you need to analyze. In the following tasks, we will guide you to discover and exploit different vulnerabilities. Beforehand, we briefly explain how the APIs work and what is important to know.

All REST APIs are deployed on `https://rest.e-hacking.de/rest-api-sec/`

- Users API: `https://rest.e-hacking.de/rest-api-sec/swagger/index.html?urls.primaryName=Users+API`
- Reports API: `https://rest.e-hacking.de/rest-api-sec/swagger/index.html?urls.primaryName=Reports+API`
- Shops API: `https://rest.e-hacking.de/rest-api-sec/swagger/index.html?urls.primaryName=Shop+API`

eHacking Credentials:

Module	Username	Password
REST: Users API	`natasha_romanoff`	`blackwidow456`
REST: Reports API	`bgreen`	`password5`
	`asmith`	`password2`
REST: Shops API	`user1`	`password1`
	`seller1`	`password3`

DOs and DON'Ts

- Read the description and the permissions for each API, see `https://rest.e-hacking.de/rest-api-sec/`.
- Use the `verifier` that each exercise specifies.
  If it does not specify a `verifier`, set the value to `secure`.
- Solve the tasks by executing the attack in the task description. Using information from previous attacks is not allowed. For instance, a SQLi attack allows you to read the entire database and solve all tasks.
- Use the provided OpenAPI files and consider only the paths described in this file.
- During your API investigations, you may manipulate the database and thus the behavior of the API. By calling the `/reset` endpoint, you can reset the database. This is required when you:
  - ... made a mistake and you need to start from the beginning.
  - ... start investigating a new `verifier`.

Task 1 Manual OpenAPI Analysis ..... Σ 15 P

Analyze the following OpenAPI description file(s).

(a) Analyze the OpenAPI file of *tomtom.com* and answer the following questions. You are allowed to use tools (e.g., Python or `jq`[1][2]) for this analysis.

(1 P) i. How many API paths are defined?

i. _____

(2 P) ii. Which authentication mechanisms are specified? Submit the authentication method's name and its type.

ii. _____

(3 P) iii. Which authentication methods are applied by default on all paths?

iii. _____

(3 P) iv. Is the authentication mechanism compliant to the current best practices? Justify your answer.

iv. _____

(3 P) v. Does the API support that users submit XML as input data?

v. _____

_____

_____

_____

_____

(3 P) vi. How many endpoints allow variables in the path?

vi. _____

Task 2 eHacking REST API: OpenAPI Analysis ..... Σ 15 P

You should start with the analysis of the OpenAPI description of the eHacking REST API and answer the following questions:

(3 P) (a) How many API paths are defined?

(a) _____

(3 P) (b) Does the API support different versions? If yes, name all of them.

(b) _____

[1] `https://earthly.dev/blog/jq-select/`

[2] `https://marketplace.visualstudio.com/items?itemName=petli-full.jq-vscode`

(3 P) (c) Which API path(s) require(s) *HTTP Bearer* authentication?

(3 P) (d) Which API path(s) allow(s) *Basic* authentication?

(d) _____

(3 P) (e) Which API path(s) do not require any authentication mechanism defined in the OpenAPI specification?

Task 3 Warm-Up with eHacking REST APIs ..... Σ 18 P

(a) **Users API:** Navigate to `https://rest.e-hacking.de/rest-api-sec/` and open the `Warm-Up` tab. Authenticate as `natasha_romanoff` and answer the following questions.

(1 P) i. What is your user's ID and role?

i. _____

(2 P) ii. Which paths can you use to figure out your Id and role?

(b) Open the `Warm-Up` tab and answer the following questions for the "Vulnerable Reports API":

Authenticate as `bgreen` and answer the following questions.

(1 P) i. What is your user's ID and role?

i. _____

(2 P) ii. Name the ids of your own reports.

ii. _____

(3 P) iii. Can you update the name of your own reports? If yes, which HTTP method and which path do you need to invoke? If no, provide a description of the error message.

iii. _____

(c) Open the `Warm-Up` tab and answer the following questions for the "Vulnerable Shop API":

Authenticate as `user1` and answer the following questions:

(1 P) i. What is your user's ID and role?

i. _____

(2 P) ii. Do you own shops? If yes, name their IDs.

ii. _____

_____

_____

_____

_____

(3 P) iii. Who owns the shop `Adventure Goods`? Name the owner's ID.

iii. _____

(3 P) iv. How much does the `BMX` cost in the shop `Bike World`?

iv. _____

Task 4 OWASP TOP 1: Broken Object Level Authorization Σ 16 P

(a) **Reports API:** Navigate to `https://rest.e-hacking.de/rest-api-sec/` and open the `TOP 1: BOLA` tab. Your task is to find the vulnerabilities in the `Vulnerable Reports API`. There are two Broken Object Level Authorization (BOLA) vulnerabilities. For each vulnerability, you should use the corresponding `verifier=bola-x` for your investigations.Name the paths and HTTP method. Describe the security problem.

Hint: To discover them, you need to authenticate with different roles.

(8 P) i. Solution to bypass Verifier bola-1.

_____

_____

_____

_____

(8 P) ii. Solution to bypass Verifier bola-2.

_____

_____

_____

_____

Task 5 OWASP TOP 2: Authentication Bypass ..... Σ 20 P

(a) **Users API:** Navigate to `https://rest.e-hacking.de/rest-api-sec/` and open the `TOP 2: Auth. Bypasses` tab. For each vulnerability, you should use the given `verifier=auth-x` during your investigations. Name the paths and describe the security problem for each Vulnerability.

(10 P) i. Solution to exploit Vulnerability 2.

(10 P) ii. Solution to exploit Vulnerability 3.

Task 6 OWASP TOP 3: Broken Object Property Level Authorization ..... Σ 40 P

(a) **Shops API:** Navigate to `https://rest.e-hacking.de/rest-api-sec/` and open the `TOP 3: BOPLA` tab. Authenticate as *customer*.

Two paths are vulnerable to *Excessive Data Exposure* by exposing unnecessary sensitive information. Name the paths and describe the problem.

(8 P) i. Vulnerable Path 1:

(8 P) ii. Vulnerable Path 2:

The API is also vulnerable to *Mass assignment*. Discover the two paths allowing you as a customer to overwrite sensitive properties of objects.

(8 P) iii. Vulnerable Path 3:

(8 P) iv. Vulnerable Path 4:

(8 P) v. Now, authenticate as *seller* and analyze the Shops API again. Which further information of other shops do you receive? Name the vulnerable path and describe the problem.
Vulnerable Path:

Task 7 OWASP TOP 4: Broken Function Level Authorization ..... Σ 16 P

(a) **Shops API:** Navigate to `https://rest.e-hacking.de/rest-api-sec/` and open the `TOP 4: BFLA` tab. Authenticate as *seller*. There are two paths vulnerable to Broken Function Level Authorization (BFLA). Describe the problem and the security impact.

(8 P) i. Vulnerable Path 1:

(8 P) ii. Vulnerable Path 2:

Task 8 OWASP TOP 7: Server Side Request Forgery Σ 10 P

(10 P) (a) **Shop API:** Navigate to `https://rest.e-hacking.de/rest-api-sec/` and open the `TOP 7: SSRF` tab. Authenticate as *seller*. Your task is to determine which path is vulnerable to Server Side Request Forgery (SSRF) and show how to exploit the vulnerability.