

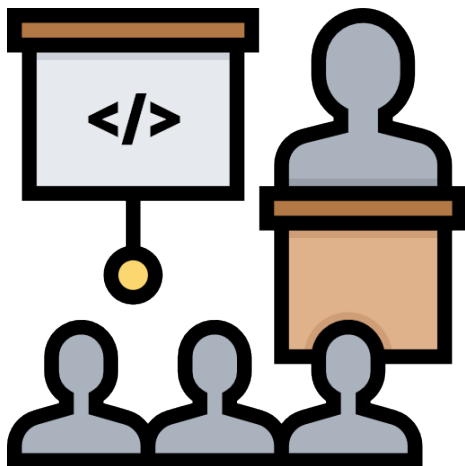
Remediate the Flag

Practical Application Security Training
for Developers

Andrea Scaduto

www.remediatetheflag.com

AppSec Training, today.



In Class Training

- ✓ Provides real-world examples
- ✗ Expensive (Cost / Time)
- ✗ Often a one time event

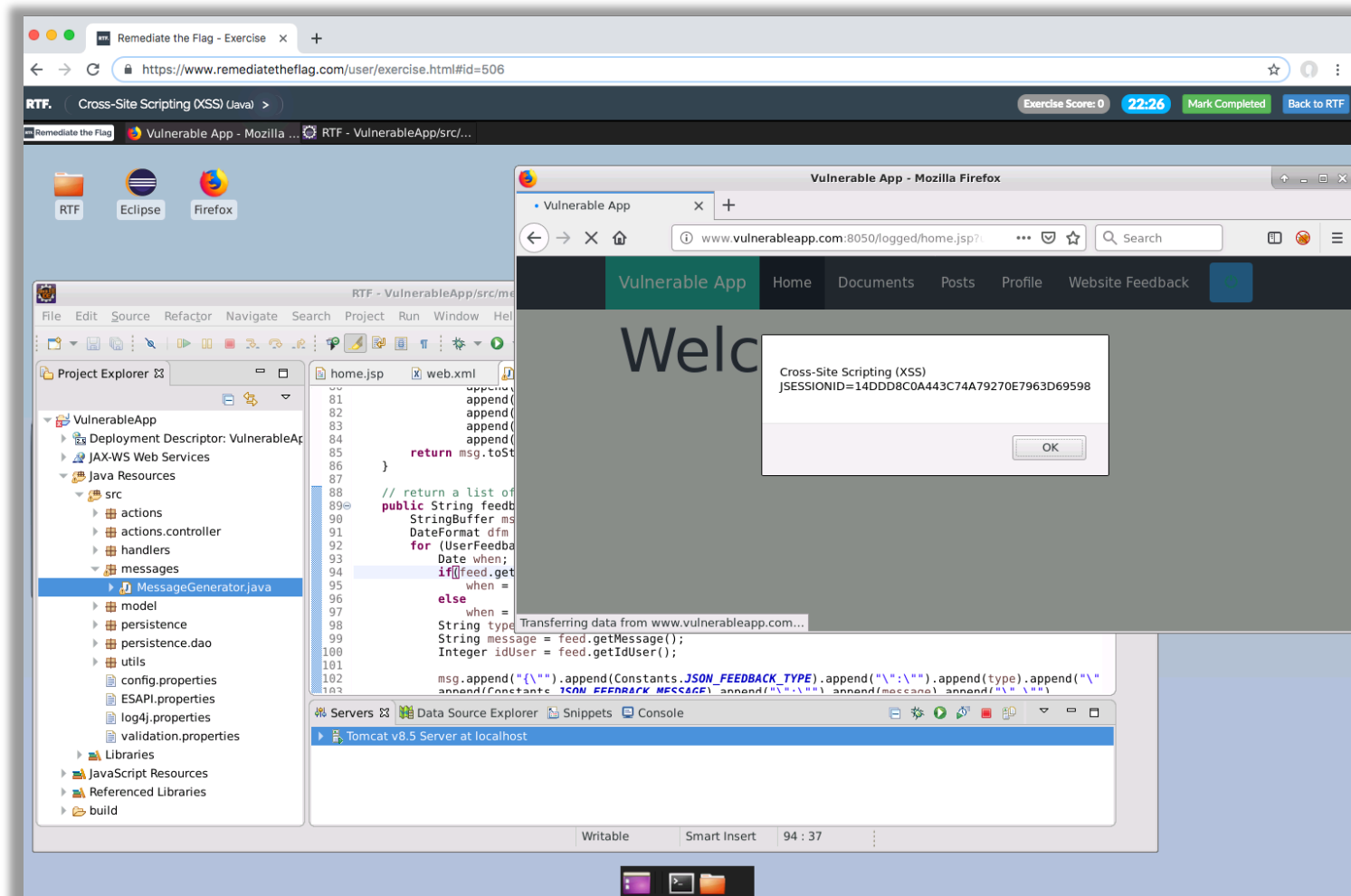


Computer Based Training

- ✗ No hands-on examples
- ✓ Scales well for large companies
- ✗ Difficult to assess competency

AppSec Training, **tomorrow.**

- 100% hands-on training
- Identify, exploit and fix security issues.
- Dedicated DEV environment accessed in seconds through a web browser.
- Learn using the same tools used at the workplace.



Engaging and Interactive

RTF. Cross-Site Scripting (XSS) (Java) >

Exercise Score: 15 29:17 Mark Completed Back to RTF

Exercise setup click to open

Score 30 Trophy

gain up to 30 points Cross-Site Scripting Trophy

In this Cross-Site Scripting (XSS) (Java) exercise, you will learn how to:

- Exploit Reflected Cross-Site Scripting Optional ?
- Exploit Stored Cross-Site Scripting Optional ?
- Remediate Stored Cross-Site Scripting ?
- Remediate Reflected Cross-Site Scripting

References:

- Stored Cross-Site Scripting in Java
- Reflected Cross-Site Scripting in Java

Remediate Stored Cross-Site Scripting

Status: **Vulnerable** (last refreshed at 10:20) gain up to 15 points

Instructions:

The feedback form functionality of the application is vulnerable to Stored Cross-Site Scripting (XSS). The exposure can be remediated by performing Output Encoding of user-controlled input before this is returned in an HTML response.

Instructions to remediate:

- Switch to Eclipse, browse to the `messages.MessageGenerator.java` class.
- Locate the `String feedbackListMessage(List feedback)` method.
- Observe that user-controlled input is assigned to the `String message` variable.
- Perform Output Encoding for the HTML context of user-controlled input.

Upon saving, the application server (Tomcat) in Eclipse is automatically reloaded with the changes.

```
append(Constants.JSON_RESPONSE).append("\\").append(Constants.JSON_LOCKDOWN).append("\\").append(destination).append("\\");
return msg.toString();
}
// build a JSON message with the user's profile
public String userProfileInfoMessage(UserProfileInfo userProfile, SessionUser user) {
String username = user.getUsername();
if(username!=null
```

- Real-time results & Hints
- Automated scoring
- Gain Points & Trophies

Learning Paths

- Exercises are grouped into logically linked units.
- Learners become expert in a topic in small steps.
- When candidates complete a Learning Path, they receive a RTF certification.
- Certifications expire and they can be renewed by taking refresher exercises.

The screenshot shows the RTF platform interface for the 'Java Secure Backend Developer' learning path. The page is titled 'Details' and includes a navigation bar with links for Dashboard, Exercises, Tournaments, Running Exercises, Completed Exercises, Achievements & Stats, and Team. A progress bar indicates 80% completion. The path consists of five exercises, each with a difficulty level of 'Easy' and a duration of 40 minutes. The exercises are: Cross-Site Scripting (XSS) (30 points, Completed), SQL Injection - Login (40 points, Completed), Horizontal Authorization Bypass (40 points, Completed), Session Fixation (15 points, Completed), and XML Entity Expansion (XXE) (50 points, Not Completed). Each exercise card provides a description of the vulnerability and a 'Run again' button. The XML Entity Expansion (XXE) exercise is currently not completed, and a 'Get Started' button is visible at the bottom of the path.

RTF. DASHBOARD EXERCISES TOURNAMENTS RUNNING EXERCISES COMPLETED EXERCISES ACHIEVEMENTS & STATS TEAM

Java Secure Backend Developer

Details

The foundational Secure Backend Developer path is designed to get you up and running in understanding core secure coding concepts for the Java platform. In this learning path, the developer will gain hands-on experience in exploiting and remediating some of the most common vulnerabilities that affect Java applications.

Difficulty **Completion** **Certification**

Easy 80% Expiration: 12 months

Cross-Site Scripting (XSS) **Completed**

30 gain up to 30 points Cross-Site Scripting (XSS) Trophy

Exploit and remediate Reflected and Stored Cross-Site Scripting (XSS) exposures. XSS attacks affect web applications that do not neutralise user input before it is placed in output as a web page. This could result in the attacker stealing sensitive information or performing actions on behalf of the victim on the vulnerable site.

Duration: 40 minutes | Difficulty: Easy

Run again

SQL Injection - Login **Completed**

40 gain up to 40 points SQL Injection Trophy

Exploit and remediate a SQL Injection exposure. All or part of an SQL command is built using unvalidated user input. This could be used to alter query logic to bypass security checks, read sensitive data, or to insert additional statements that modify the back-end database.

Duration: 30 minutes | Difficulty: Easy

Run again

Horizontal Authorization Bypass **Completed**

40 gain up to 40 points Authorization Trophy

When access control checks are not applied consistently - users are able to access data or perform actions that they should not be allowed to perform. This can lead to a wide range of problems, including exposing sensitive information and fraud.

Duration: 40 minutes | Difficulty: Easy

Run again

Session Fixation **Completed**

15 gain up to 15 points Session Management Trophy

Exploit and remediate session management exposures. A Session Fixation occurs when authenticating users without invalidating any existing session identifier. When terminating users session, if this is performed by setting the Cookie to a new value while the server-side state remains active, the old session could be reused.

Duration: 40 minutes | Difficulty: Easy

Get Started

XML Entity Expansion (XXE) **Not Completed**

50 gain up to 50 points XML Entity Expansion Trophy

RTF.
Browse as User | HOME | ORGS | GATEWAYS & CLUSTERS | USERS | TEAMS | AVAILABLE EXERCISES | RUNNING EXERCISES | CHALLENGES | PENDING REVIEWS | COMPLETED REVIEWS | STATS | [Email] [Settings] [Refresh]

Java Top Vulnerabilities

Back
Edit

Exercises

5

Run Exercises

16

Start Date

2018-09-23 09:00 (+01:00)

Flags

6

Run Flags

17

End Date

2019-03-30 18:00 (+00:00)

Users

5

Total Exercises

25

Last Activity

2019-01-04 22:16 (+00:00)

Completion

56.0%

Total Flags

30

Status

In Progress

Remediation

82.4%

Running Exercises

0

Organization

Stark Industries

Challenge Exercises

- SQL Injection
- XML Entity Expansion (XXE)
- Broken Session Management
- Horizontal Authorization Bypass
- OS Command Injection

Challenge Scoring

Automated Scoring

Challenge Details:

Exploit and remediate a number of Java vulnerabilities. This challenge includes the following exercises: Session Fixation, Ineffective Logout, XML Entity Expansion, Horizontal Authorization Bypass, and OS Command Execution and Arbitrary File Upload. A reference document is available for each exercise.

Challenge Table

User	Country	Score	Run Exercises
michael	United Kingdom	215	4
andrea	Italy	165	5
joanne	United States	85	5
frank	United Kingdom	50	3
john	United States	30	2

	SQL Injection leading to Authentication Bypass	XML Entity Expansion	Session Fixation	Session Not Invalidated On Logout	Horizontal Authorization Bypass	OS Command Injection
andrea	Not Vulnerable	Not Vulnerable	Not Vulnerable	Not Vulnerable	Not Vulnerable	Not Started
joanne	Broken Functionality	Not Started	Not Vulnerable	Not Vulnerable	Vulnerable	Not Started
michael	Not Vulnerable	Not Vulnerable	Not Started	Not Started	Not Started	Not Started
frank	Not Started	Not Vulnerable	Not Vulnerable	Not Vulnerable	Not Started	Not Vulnerable
aarav	Not Vulnerable	Vulnerable	Not Started	Not Started	Not Started	Not Started

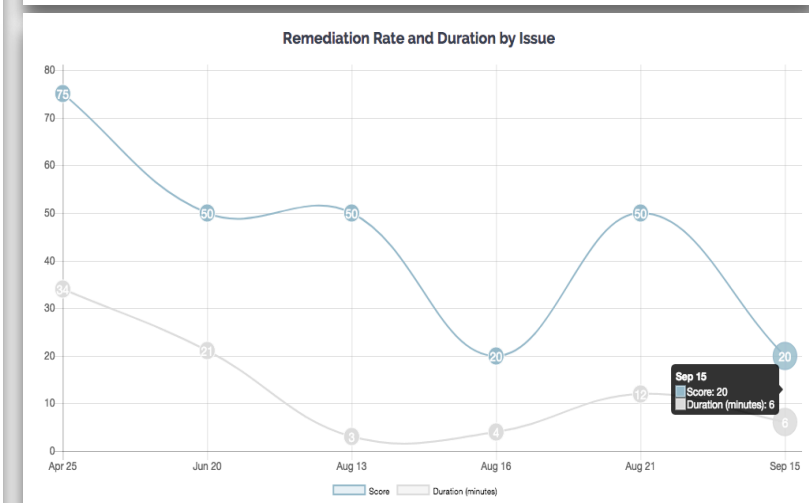
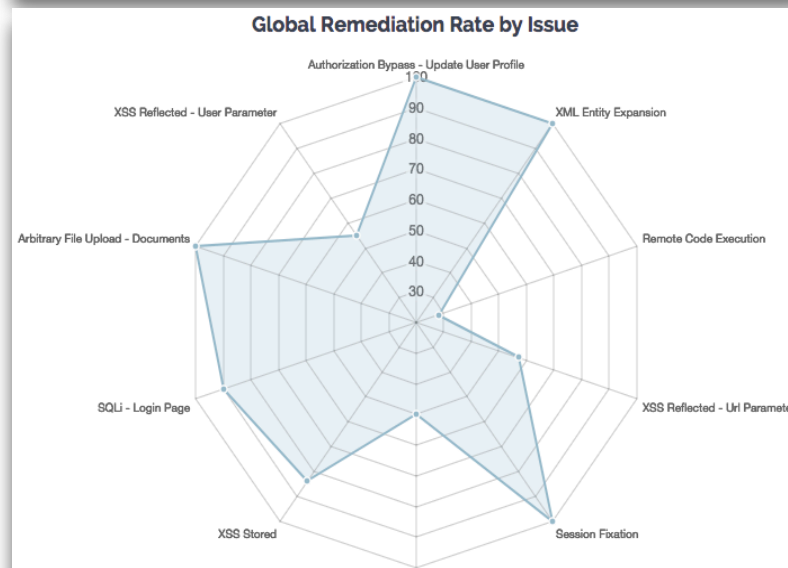
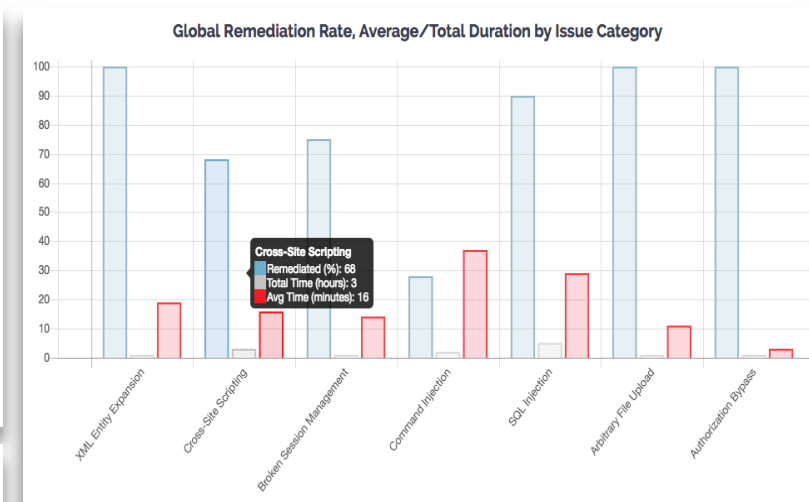
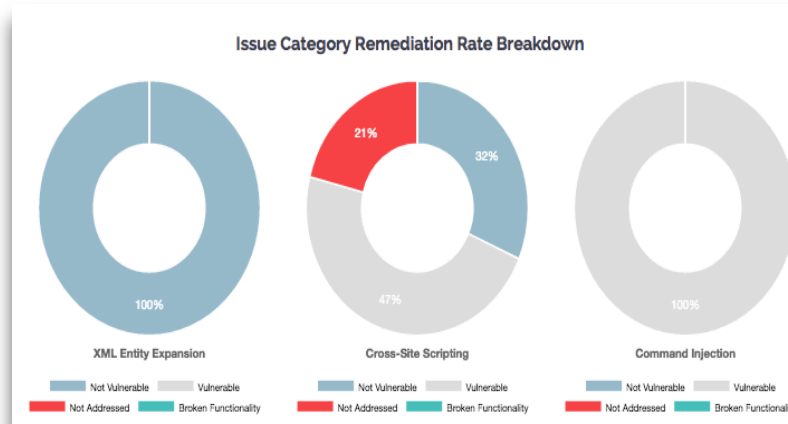
Last refreshed at 12:41 (+00:00) on Feb 10 2019

Tournaments

- Run time-boxed challenges.
- Users of the same Organization compete to remediate security issues.
- Engage the whole developer community.

Measure ROI for Training

- Measure *real* competency in secure coding and remediation
- Metrics allow for rapid discovery and closure of gaps
 - User
 - Team
 - Geographical region
 - Organization



RTF.

Live Demo



1. Start an exercise
2. Exploit vulnerability
3. Remediate code
4. Check results

Installation

Step 1

- Signup to AWS
- Purchase a Domain
- Provision TLS certificate on AWS ACM



Step 2

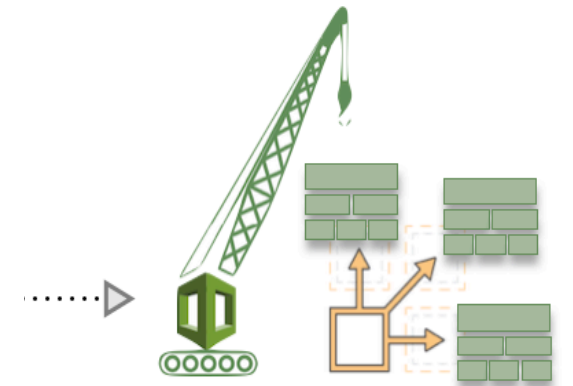
- Import RTF template on AWS CloudFormation
- Tweak configuration (cluster size, passwords, hostnames, SSL certificate, etc.)



image from aws.amazon.com/cloudformation/

Step 3

- Run template
- Wait ~ 14 minutes
- Enjoy

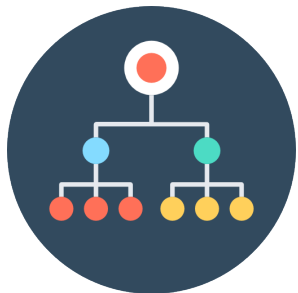


Configuration

Setup your structure

- Create Organizations and Teams based on your structure.
- Onboard/invite users.

Users in the same Team can compare their progress on a leaderboard.



Organizations



Teams



Users

Install Exercises

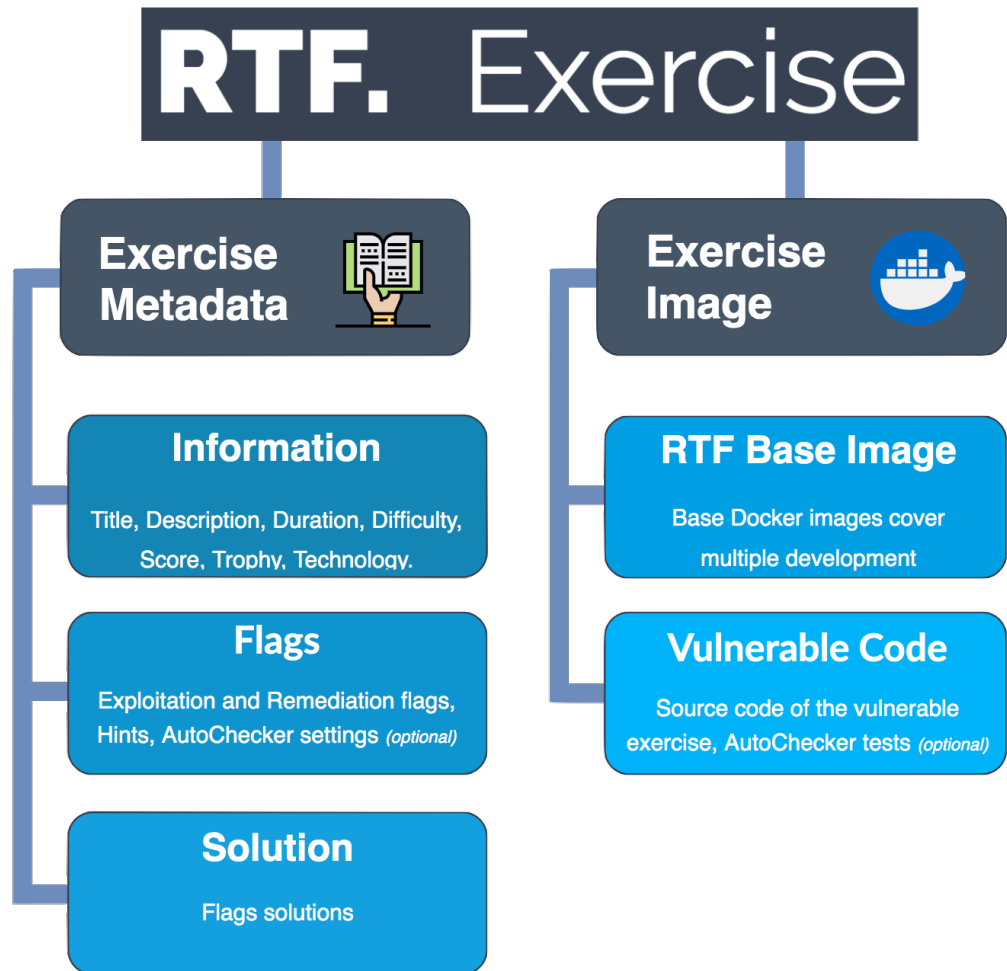
- Install Exercises from RTF Exercise Hub.
- Create your own exercises using the RTF SDK.

Exercises run in isolated DLP-friendly environments.

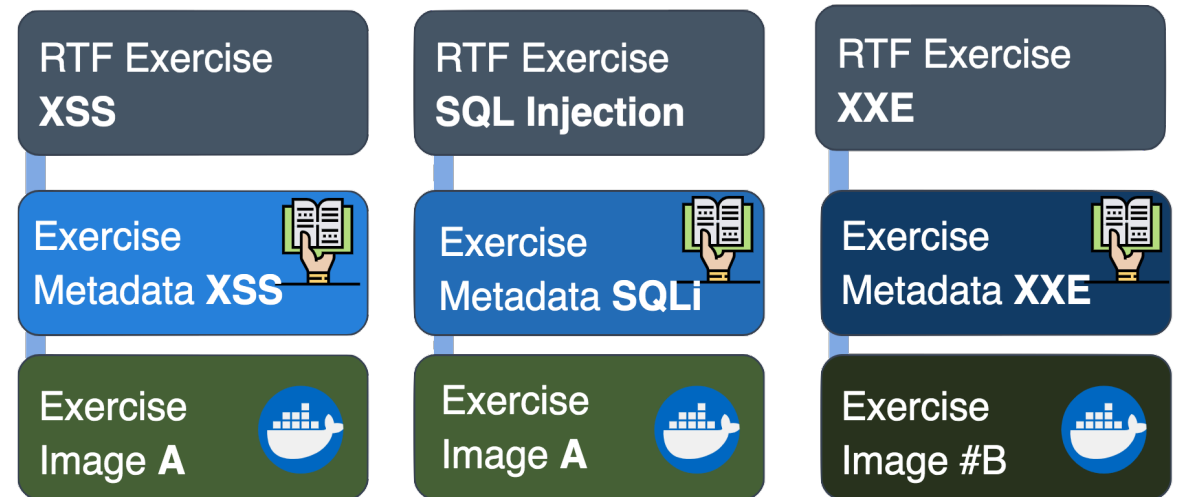
RTF. Exercise Hub

RTF. SDK

Exercise structure



It is possible to create multiple RTF Exercises reusing the same Exercise Image (e.g. *one* Vulnerable App with *many* vulnerabilities).



Creating new exercises

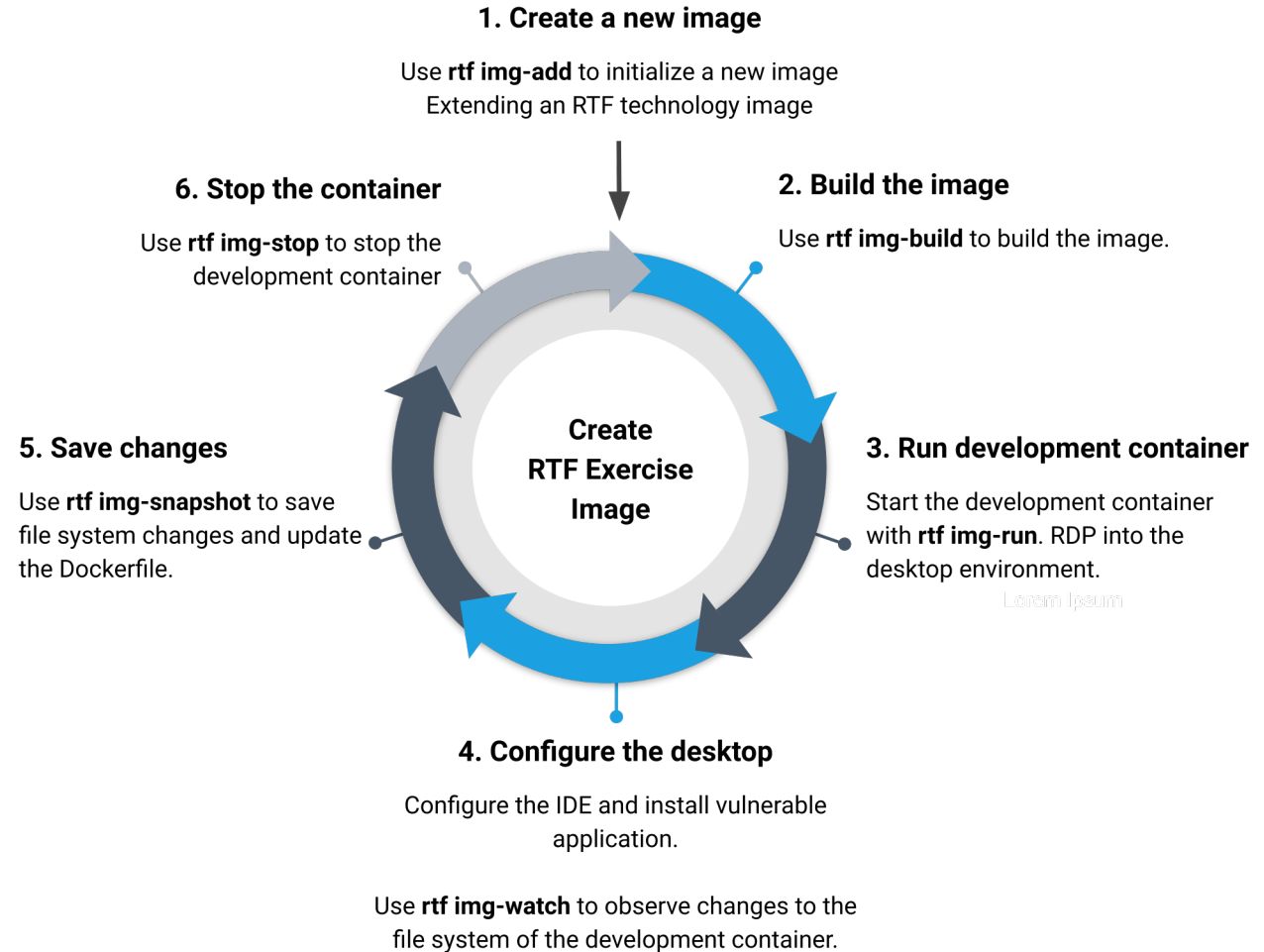


Create Exercise Image

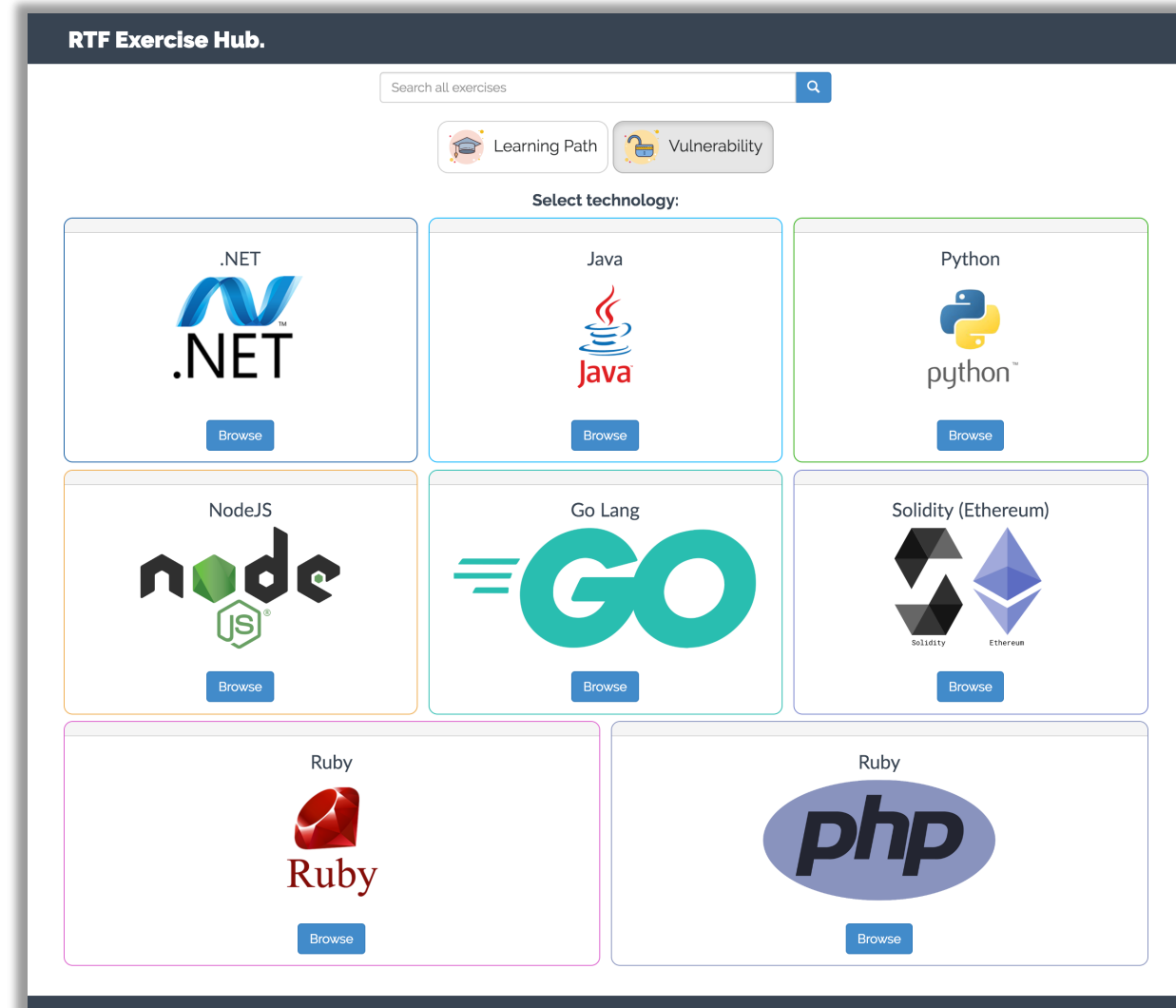
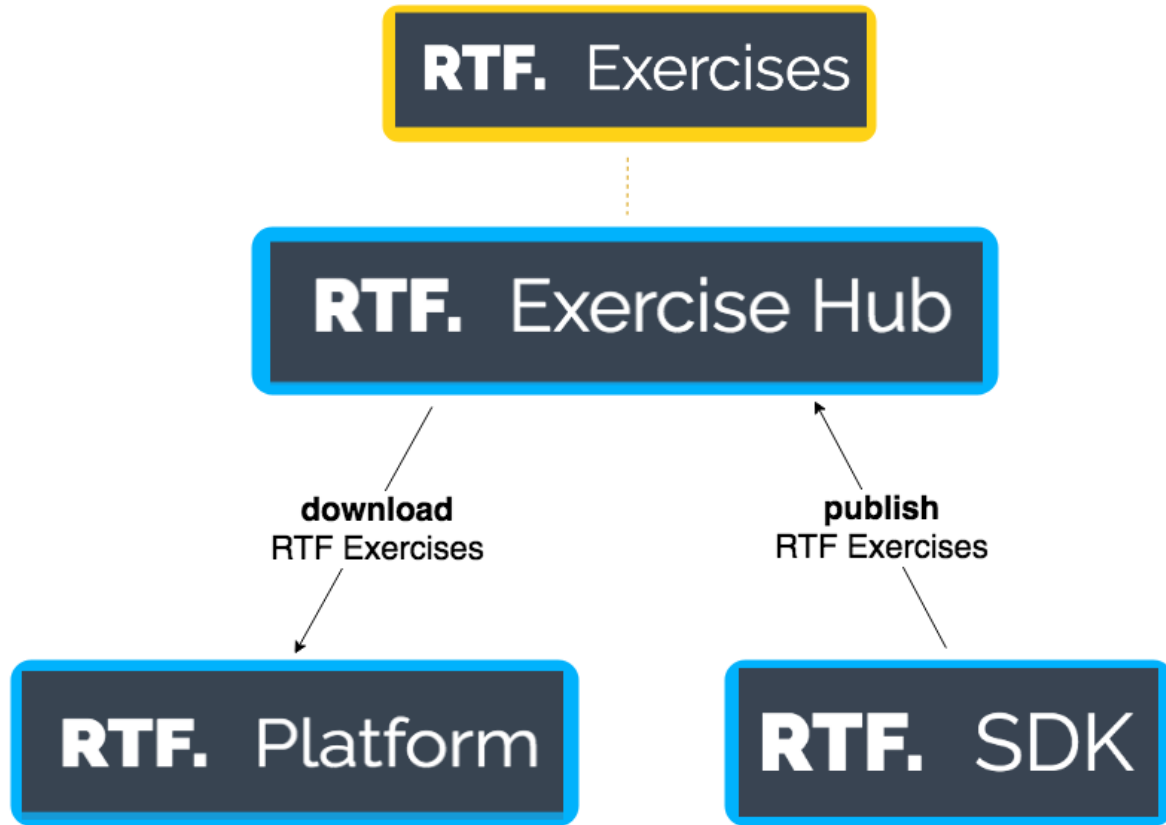
Create Exercise Metadata

Publish

- 1 Create a RTF Exercise docker image that runs a vulnerable application
- 2 Describe how to exploit and remediate the vulnerabilities in the application
- 3 Publish your exercise to the RTF Exercise Hub



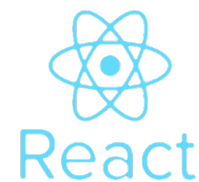
RTF Exercise Hub



Supported Exercise Technologies



On the Roadmap



Q&A

info@remediatetheflag.com

www.remediatetheflag.com