



OWASP  
AppSec Europe  
London 2nd-6th July 2018

# Remediate the Flag

Practical AppSec Training Platform

Andrea Scaduto





# Remediate the Flag

Andrea Scaduto

## Bio

### **Andrea Scaduto**

Pentester & Software Engineer



### *Interests:*

- Web / Mobile Apps Pentesting
- Optimization of costs in addressing security issues
- Training developers in remediation and secure coding



# Remediate the Flag

Andrea Scaduto

## Application Security Training

### Pentesters

- Still finding vulnerabilities that have existed for 10+ years (XSS, SQLi, XXE, RCE, etc)
- Plenty of incorrect security fixes which don't remediate the vulnerability

### Developers

- Focus on creation of functional code, they aren't born knowing how to code securely
- Education and training usually neglects application security and often gives bad advice
- Computer-based training is boring and does not provide practical examples

### Business

- Usually not possible to truly assess competency in secure coding
- Difficult to calculate return on investment for security training



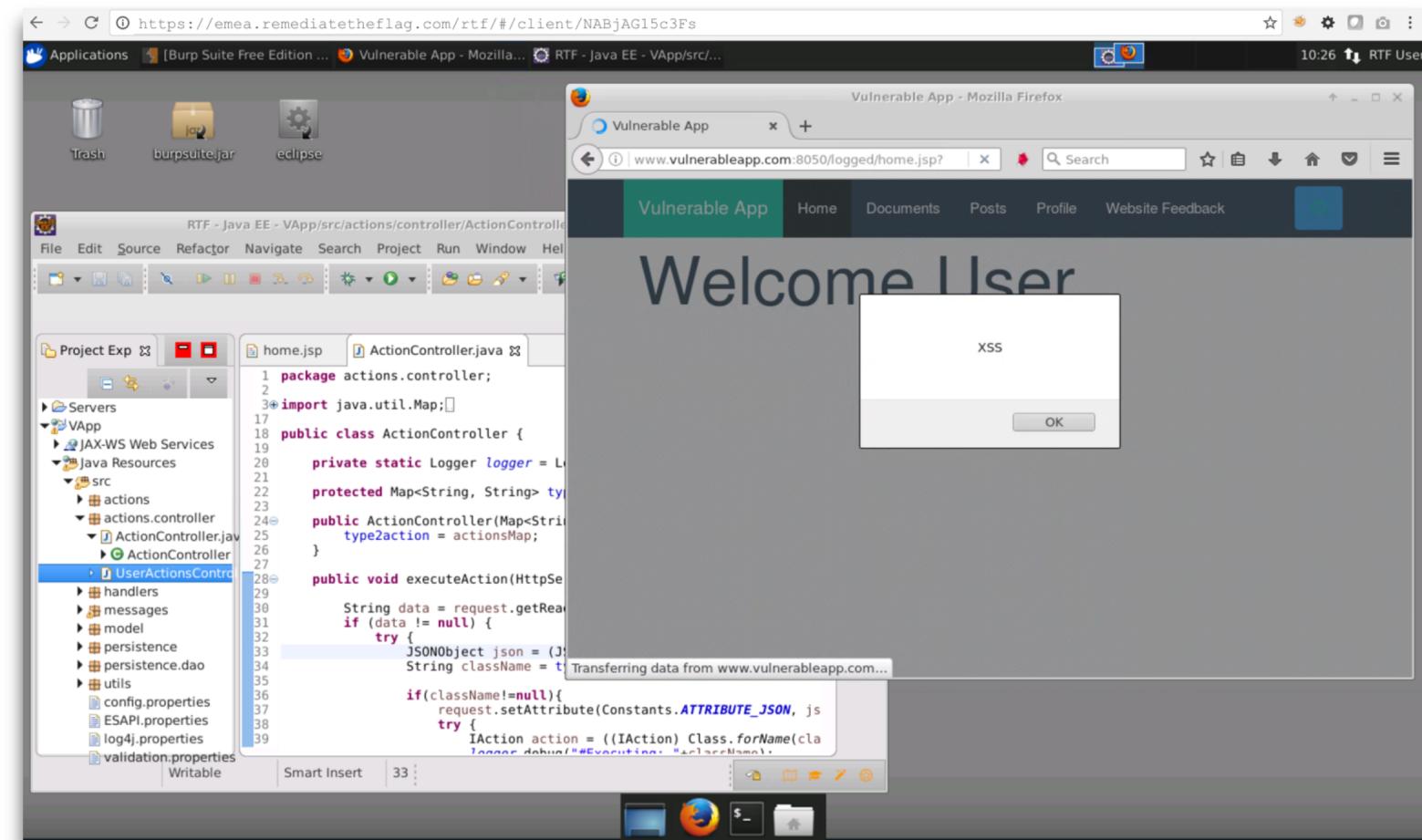
# Remediate the Flag

Andrea Scaduto

## Practical Application Security Training

Open source platform for application security exercises:

- Find security exposures and remediate them, 100% hands-on!
- Exercises are launched *in seconds* and accessed through a web browser
- Candidates exploit and manually remediate vulnerable code





# Remediate the Flag

Andrea Scaduto

## Real Time Feedback & Hints

- Check in real time whether security issues were successfully remediated
- Hints are available (reduces final score)

Java

**Cross-Site Scripting (XSS)**  
Improper Neutralization of Input During Web Page Generation

Exploit and remediate Reflected and Stored Cross-Site Scripting (XSS) exposures. XSS attacks affect web applications that do not neutralize 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

Score **50** Trophy 

gain up to 50 points Cross-Site Scripting Trophy

**Stored Cross-Site Scripting**

Type	Instructions	Required	Hint	Check result
EXPLOITATION	Exploit the Stored XSS in the "Add Feedback" functionality of the 'Website Feedback' page.	Optional	Show Hint	
REMEDIATION	Remediate the vulnerability by performing Output Encoding in the String feedbackListMessage(List<UserFeedback> feedback) method in the messages.MessageGenerator.java class.	Required	Show Hint	<b>Not Vulnerable</b> Refresh

**Reflected Cross-Site Scripting (User parameter)**

Type	Instructions	Required	Hint	Check result
EXPLOITATION	Exploit the Reflected-XSS in the 'user' query-string parameter of the 'Home' page of the application.	Optional	Show Hint	
REMEDIATION	Remediate the vulnerability by performing the correct Output Encoding for the 'user' parameter in the JSP atWebContent/logged/home.jsp.	Required	Show Hint	<b>Vulnerable</b> Refresh



# Remediate the Flag

Andrea Scaduto

## Exercise Results

- The platform provides automated results including a code diff and logs
- An assessor can review results and provide feedback to the candidate

RTF.

HOME AVAILABLE EXERCISES RUNNING EXERCISES CHALLENGES COMPLETED EXERCISES ACHIEVEMENTS & STATS LEADERBOARD

1

### Results

Flag	Result	Comment
Reflected Cross-Site Scripting	Not Vulnerable	The exposure has been correctly remediated. The candidate performed Output Encoding on both URL parameters using the appropriate encoding context. Well done!
Stored Cross-Site Scripting	Not Vulnerable	The exposure has been correctly remediated. The candidate performed Output Encoding of the user's feedback that is returned as part of the JSON message.
DOM Based Cross-Site Scripting	Not Addressed	No exposure has not been addressed.

Show/Hide Source Code Diff

Show/Hide Instance Logs

Leave Feedback

Download Exercise Solution

#### Source Code Diff:

Files changed (2)

/src/messages/MessageGenerator.java

```
@@ -92,7 +92,7 @@
 92 92      else
 93 93          when = new Date(System.currentTimeMillis());
 94 94      String type = feed.getType();
 95 95      - String message = feed.getMessage();
 95 95      + String message = ESAPI.encoder().encodeForHTML(feed.getMessage());
 96 96      Integer idUser = feed.getIdUser();
 97 97
 98 98      msg.append("{"").append(Constants.JSON_FEEDBACK_TYPE).append(":");append("}").append(type).append(":", "});
```

//WebContent/logged/home.jsp

```
@@ -97,7 +97,7 @@
 97 97      <div class="container">
 98 98          <h1>
 99 99              Welcome
 100 100      -     <%= request.getParameter("user") %>!
 100 100      +     <%= ESAPI.encoder().encodeForHTML(request.getParameter("user")) %>!
 101 101          </h1>
```



# Remediate the Flag

Andrea Scaduto

## Challenges

- Candidates can join time-boxed tournaments or challenge other users
- Choose programming languages, target developer groups or specific vulnerabilities

[Back to Challenges](#)

Total Issues	Total Users	Run Exercises	Completion	Remediation Rate
4	4	12/16	75%	100%
Start Date	End Date	Last Activity	Status	Organization
Apr 9, 2018 10:00:00 AM	May 5, 2018 10:00:00 PM	Apr 23, 2018 08:26:15 PM	In Progress	Acme Corp

**Challenge Table**

User	Score	Run Exercises	Team	Organization
mick	3	4	Team C	Acme Corp
vanessa	2	2	Team A	Acme Corp
keith	2	3	Team C	Acme Corp
john	1	3	Team B	Acme Corp

**Challenge Results**

User	Arbitrary File Upload - Documents	Session Fixation	Session Not Invalidated On Logout	Horizontal Authorization Bypass
vanessa	Not Vulnerable	Not Started	Not Started	Not Vulnerable
keith	Broken Functionality	Not Vulnerable	Not Vulnerable	Not Started
john	Not Started	Vulnerable	Vulnerable	Not Vulnerable
mick	Vulnerable	Not Vulnerable	Not Vulnerable	Not Vulnerable



# Remediate the Flag

Andrea Scaduto

## Exercises & Expansion

- Targeted exercises to address the most prevalent security issues
- Target multiple technology stacks and developer groups
- New vulnerable applications and exercises can be easily integrated

The screenshot shows the RTF platform interface with a navigation bar at the top. The main area displays a grid of six training exercises for Java, each with a title, status, score, trophy icon, and a 'Get Started' button.

Training Exercises		
<b>Java</b> Completed <b>SQL Injection</b> Improper Neutralization of Special Elements used in an SQL ... Score: 50 Trophy: SQL Injection Trophy gain up to 50 points Duration: 30 minutes   Difficulty: Easy Get Started	<b>Java</b> Not Completed <b>OS Command Injection</b> Improper Neutralization of Special Elements used in an OS C... Score: 70 Trophy: Command Injection Trophy gain up to 70 points Duration: 40 minutes   Difficulty: Moderate Get Started	
<b>Java</b> Completed <b>XML Entity Expansion (XXE)</b> Improper Restriction of XML External Entity Reference Score: 50 Trophy: XML Entity Expansion Trophy gain up to 50 points Duration: 30 minutes   Difficulty: Easy Get Started	<b>Java</b> Completed <b>Cross-Site Scripting (XSS)</b> Improper Neutralization of Input During Web Page Generation Score: 50 Trophy: Cross-Site Scripting Trophy gain up to 50 points Duration: 40 minutes   Difficulty: Easy Get Started	<b>Java</b> Completed <b>Broken Session Management</b> Session Fixation, Session Not Invalidated Upon Logout Score: 20 Trophy: Session Management Trophy gain up to 20 points Duration: 40 minutes   Difficulty: Easy Get Started

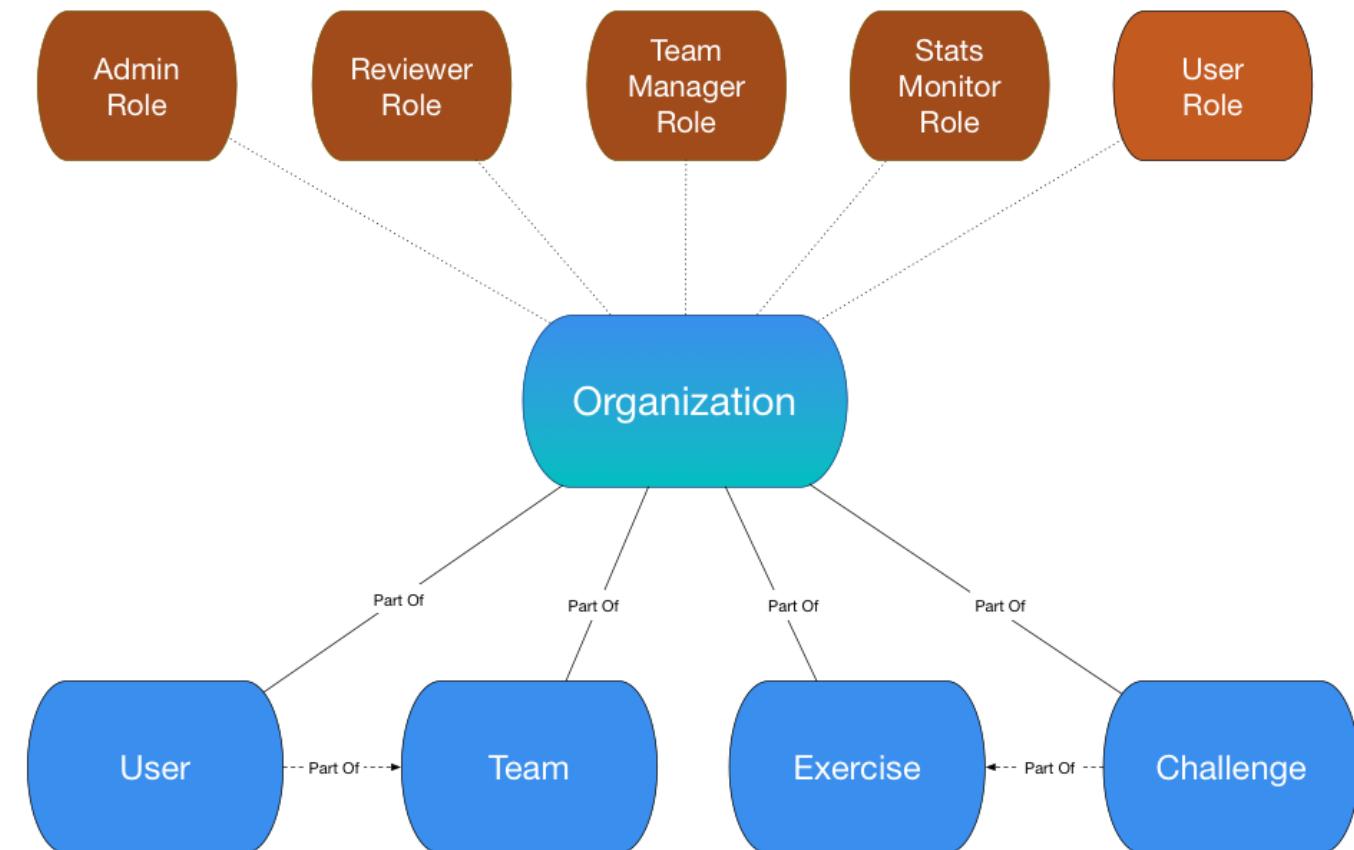


# Remediate the Flag

Andrea Scaduto

## Management Interface

- IAM model based on roles, Teams and Organizations
- Manage Orgs, Users, Teams
- Setup Exercises, Challenges and platform configuration settings
- View metrics and statistics



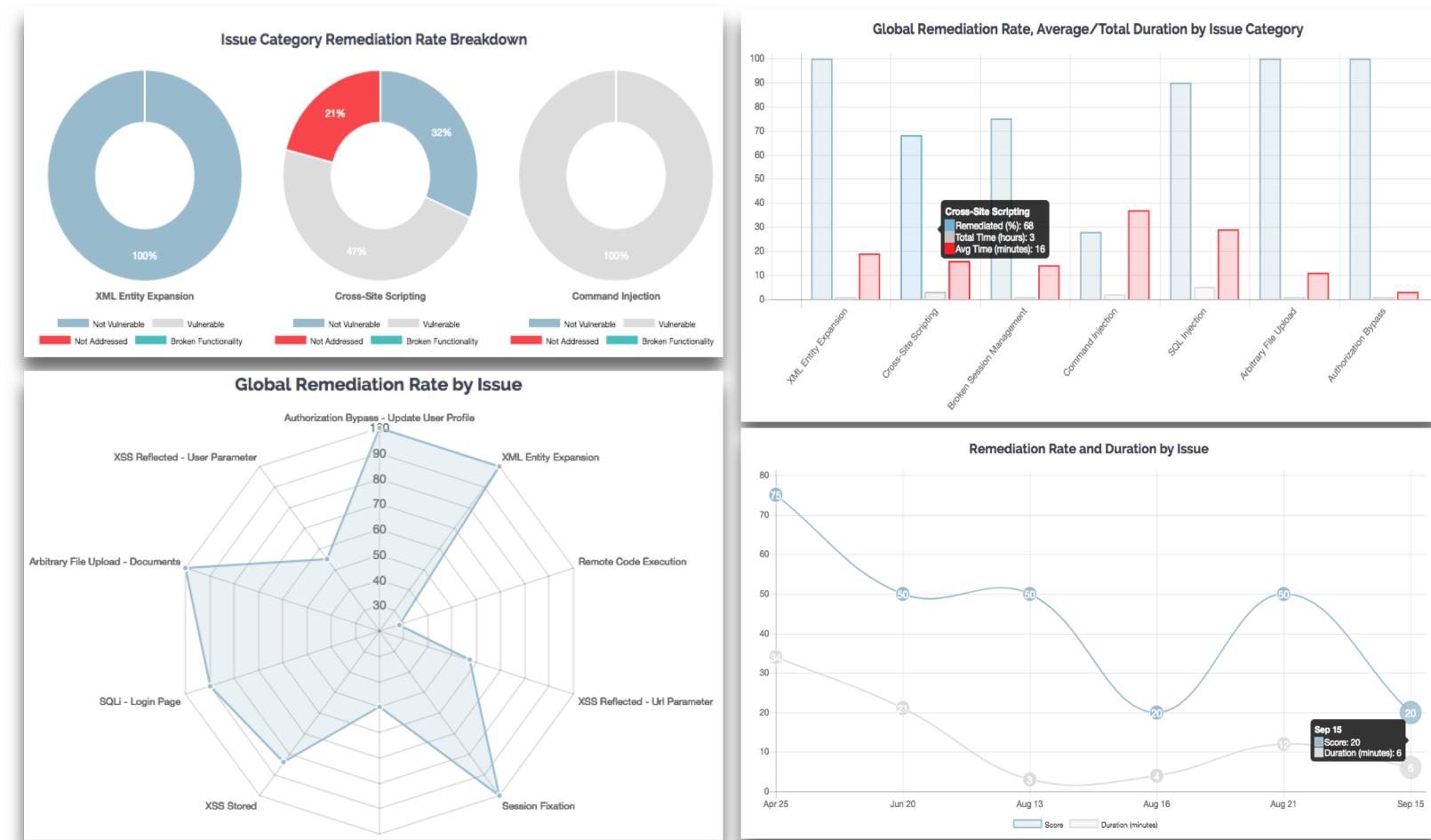


# Remediate the Flag

Andrea Scaduto

## Measure ROI for Training

- Measure *real* competency in secure coding and remediation
- View User, Team and Organization-level metrics to quickly identify and address gaps





# Live Demo

• • •

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



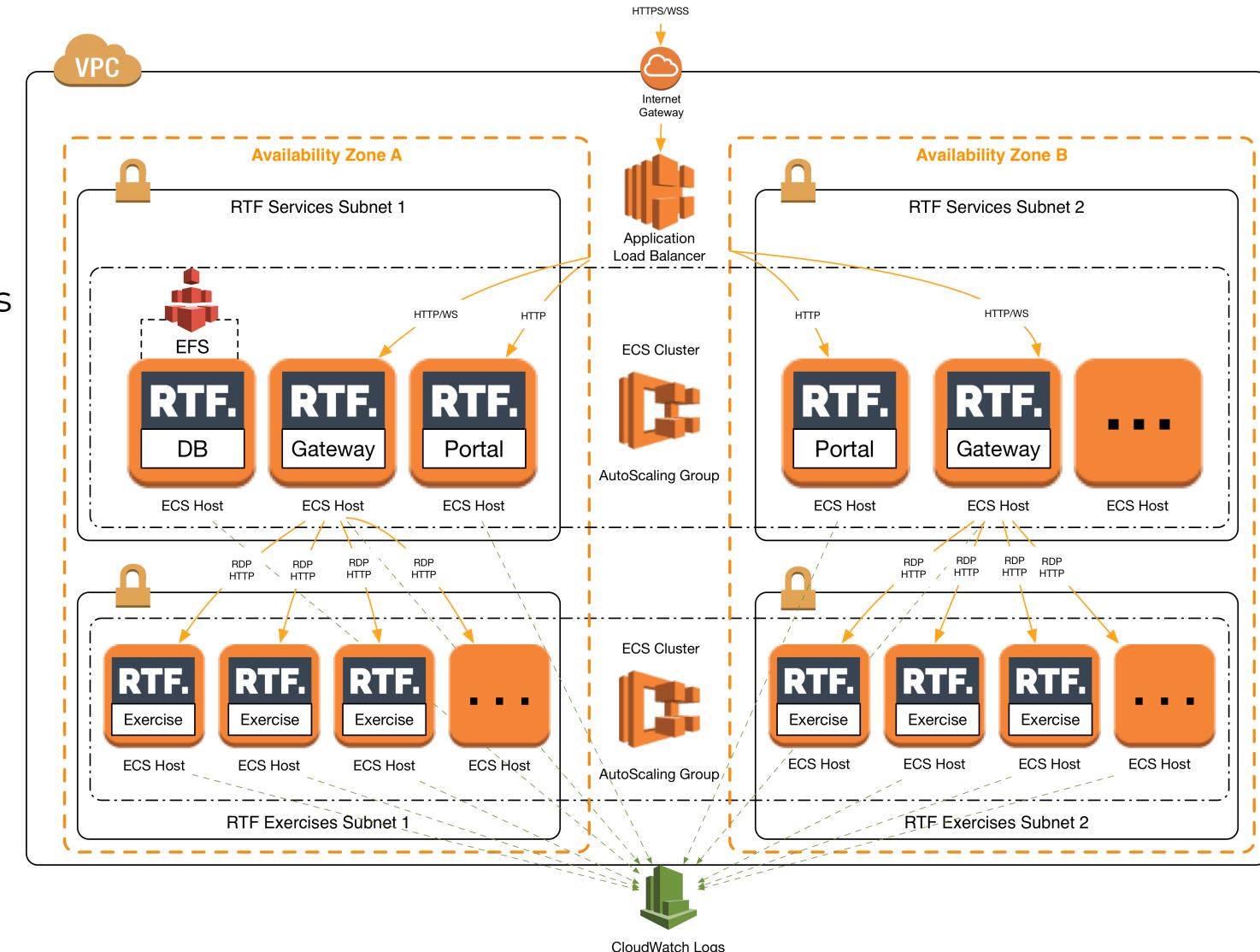
# Remediate the Flag

Andrea Scaduto

## Architecture

Micro-services architecture deployable on AWS through CloudFormation

- **RTF VPC:**
  - 2x Services subnets + 2x Exercises subnets
  - Spread across two availability zones
- **RTF Services (ECS Cluster)**
  - Runs RTF Platform containers
  - Traffic routed through ALB
  - ECS Service/Instances Autoscaling
- **RTF Exercises (ECS Cluster)**
  - Runs RTF Exercise containers
  - Traffic routed through RTF Gateway
  - No outbound connectivity
  - EC2 Instance Autoscaling
- Centralised Logs on AWS CloudWatch



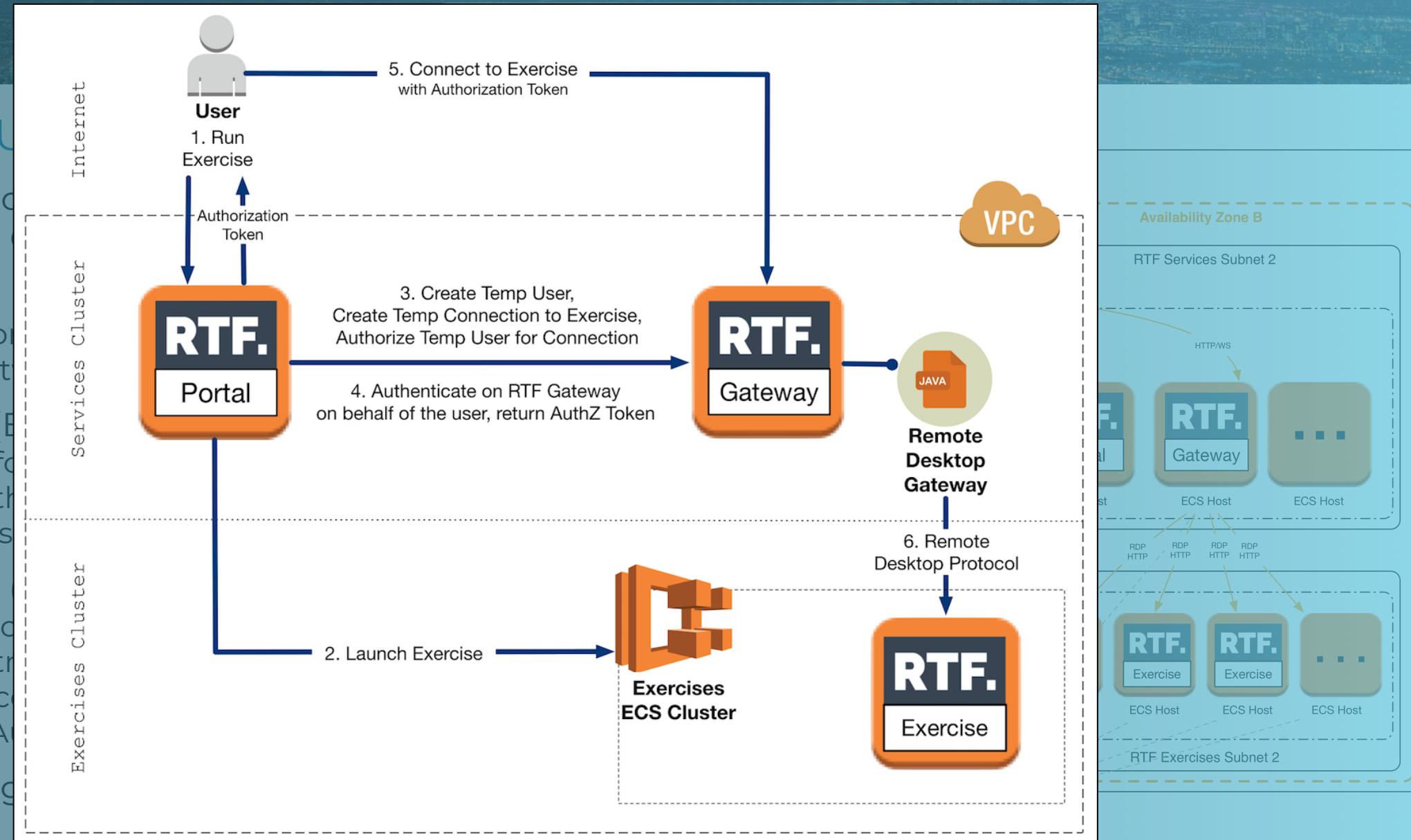


## Architecture

Micro-services architecture on AWS through

- **RTF VPC:**
  - 2x Services subnets
  - Spread across two AZs
- **RTF Services (Backend)**
  - Runs RTF Platform
  - Traffic routed through ECS Service/Instances
- **RTF Exercises**
  - Runs RTF Exercises
  - Traffic routed through ECS Cluster
  - No outbound connection
  - EC2 Instance Automation
- Centralised Logging

# Remediate the Flag



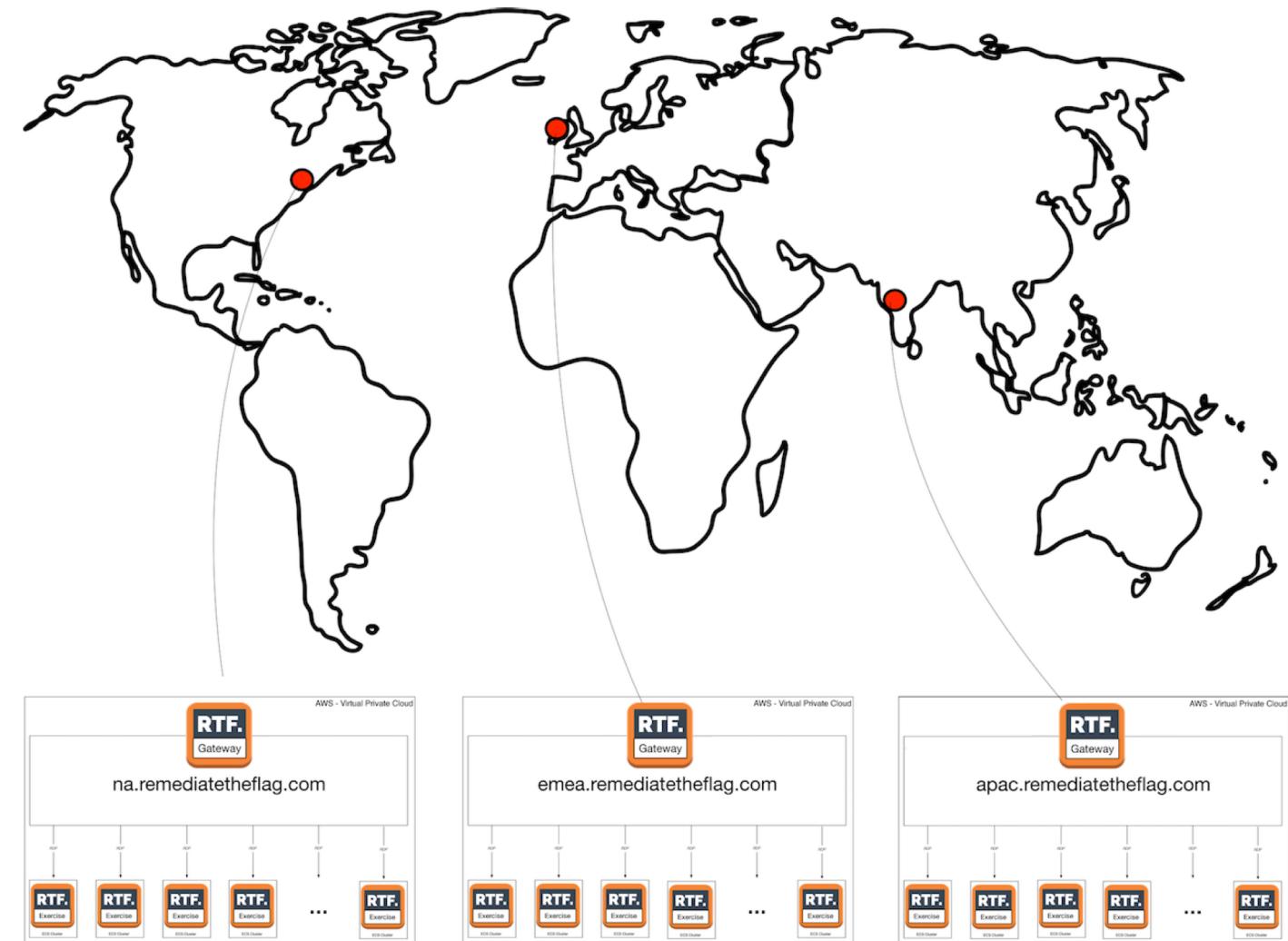


# Remediate the Flag

Andrea Scaduto

## Regional Clusters

- Deploy additional regional exercise clusters in any AWS region:
  - Increase concurrent exercise capacity
  - Reduce latency
- Configure RTF Gateways from the Management Interface
- Enable/Disable exercises for each region





# Remediate the Flag

Andrea Scaduto

## Installation

### Step 1

- Build Docker Images  
(or use pre-built images)
  - RTF Portal
  - RTF Gateway
  - RTF Database



### Step 2

- Signup to AWS
- Provision SSL/TLS certificate on AWS ACM
- Push Docker Images to AWS ECR



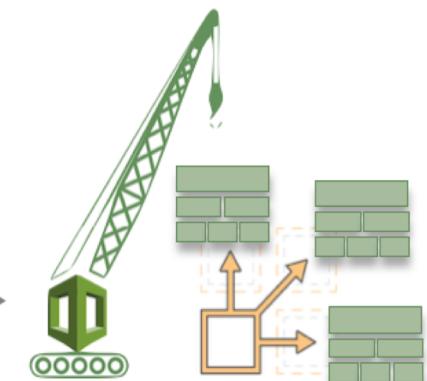
### Step 3

- Import AWS CloudFormation templates from AWS S3
- Tweak configuration (cluster size, password for services, hostname, SSL certificate)



### Step 4

- Run template
- Wait ~11 minutes
- Enjoy



Picture from [aws.amazon.com/cloudformation/](http://aws.amazon.com/cloudformation/)



# Remediate the Flag

Andrea Scaduto

## Platform Setup

### Populate user base

- Create Organizations, Teams and Users

### Onboard Gateways

- Onboard RTF Gateways for deployed regional Exercise Clusters

Add Gateway

Name	Region	FQDN
Gateway EU	EU (London)	emea.remediatestheflag.net
Status	Enabled	
<input type="button" value="Close"/> <input type="button" value="Add"/>		

### Add Exercises

- Add exercise metadata
- Register exercise on RTF Gateway
- Enable exercise for organization

Add Exercise

Description	Resources	Info	Flags	Score & Trophy
Title	Cross-Site Scripting Exercise 1			
Topics	Reflected XSS, Stored XSS, and DOM-based XSS			
Description	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 (minutes)	Difficulty	Technology	Exercise Status	
40	Easy	Java	Available	
<input type="button" value="Import"/>	<input type="button" value="Close"/>	<input type="button" value="Add Exercise"/>	Start Date: 2018-07-02 10:00:00 UTC End Date: 2018-07-02 11:00:00 UTC	

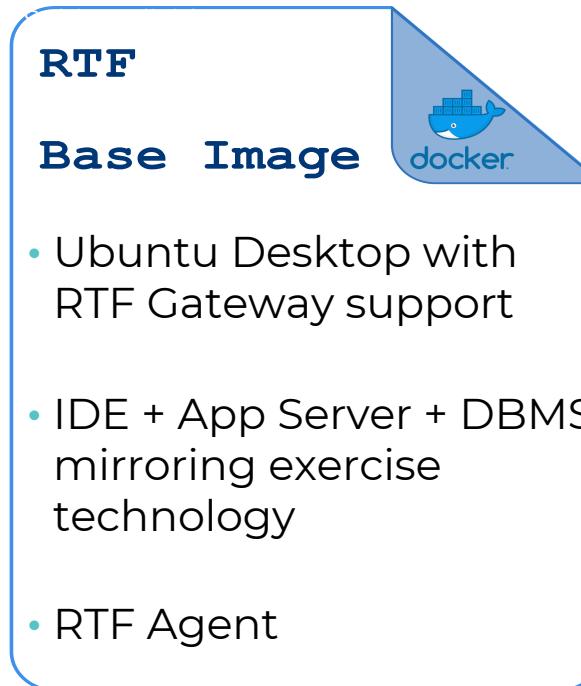


# Remediate the Flag

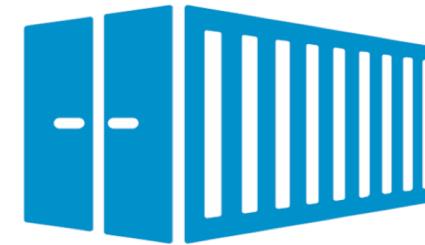
Andrea Scaduto

## Create new exercise

### 1. Customise public base image



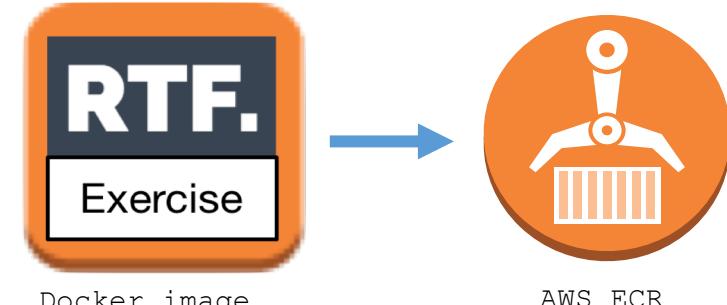
### 2. Run container, integrate and test



Docker container

- Run and test the container locally, connect via RDP
- Integrate your application in the IDE and customize user settings and appearance.

### 3. Export home folder and re-build



Docker image

AWS ECR

- Export the user's home folder
- Add the folder to Dockerfile
- Build a new image
- Push image to AWS ECR



# Remediate the Flag

Andrea Scaduto

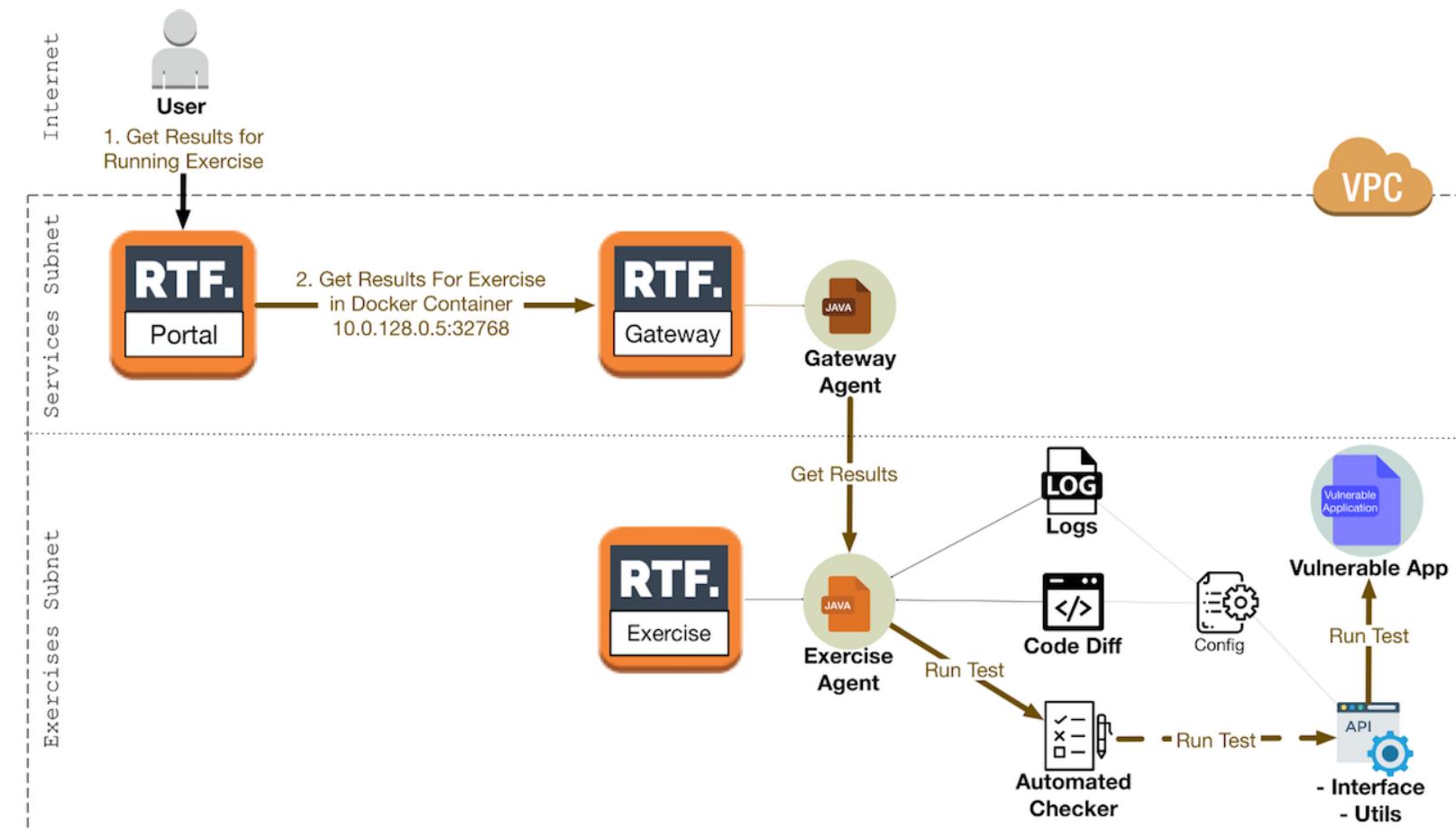
## RTF Agent

### Agent Functionalities:

- Retrieve Exercise Logs
- Source Code Diff
- Automated Checker

### Automated Checker Strategies:

- Blackbox
  - *HTTP requests/responses with condition checking*
  - *UI Automation scripts*
- Whitebox
  - *Unit Tests*
  - *Static Analysis (late 2018)*





## Testing for Reflected XSS (using Java)

```
public Boolean xssTest(String cookie) {  
    return sendGet(host, "/logged/home.jsp?user=%3C/test()%3E", "</test()%3E", cookie);  
}
```

## Testing for DOM-Based XSS (using NightmareJS / Chai / Mocha)

```
nightmare  
  .goto(host)  
  .type('#username', username)  
  .type('#password', password)  
  .click('#loginButton')  
  .wait(2000)  
  .goto(host+"/#lang=document.getElementById('error').innerHTML='pwned'")  
  .evaluate(() => document.querySelector('#error').innerHTML)  
  .end()  
  .then(string => {  
    expect(string).to.equal('pwned')  
    done()  
  })
```



Vulnerable App

Run Test

- Interface  
- Utils



# Remediate the Flag

Andrea Scaduto

## Benefits

### Developers

- 100% hands-on training, learn in an engaging way and challenge other users
- Get familiar with the most prevalent vulnerabilities and recognise insecure coding patterns

### Business

- Measure real competency in secure coding and remediation
- Provide targeted training to fill gaps and reduce new security issues introduced in development

### Community

- Open source platform, simple deployment on AWS through CloudFormation
- Easily extendable with new exercises and technologies



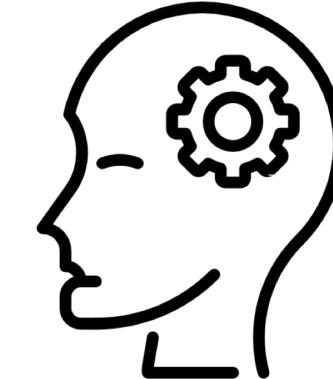
# Remediate the Flag

Andrea Scaduto

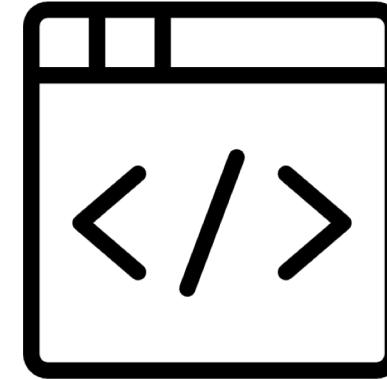
## Next Steps



Provide Feedback



Create New Exercises



Contribute to Development

[www.remediatetheflag.com](http://www.remediatetheflag.com)  
[github.com/sk4ddy/remediate-the-flag](https://github.com/sk4ddy/remediate-the-flag)