

RSA[®]Conference2019

San Francisco | March 4–8 | Moscone Center



BETTER.

SESSION ID: LAB3-W02

Are You a Secure Coding Champion?

Pieter Danhieux

CEO
Secure Code Warrior
pd@scw.io

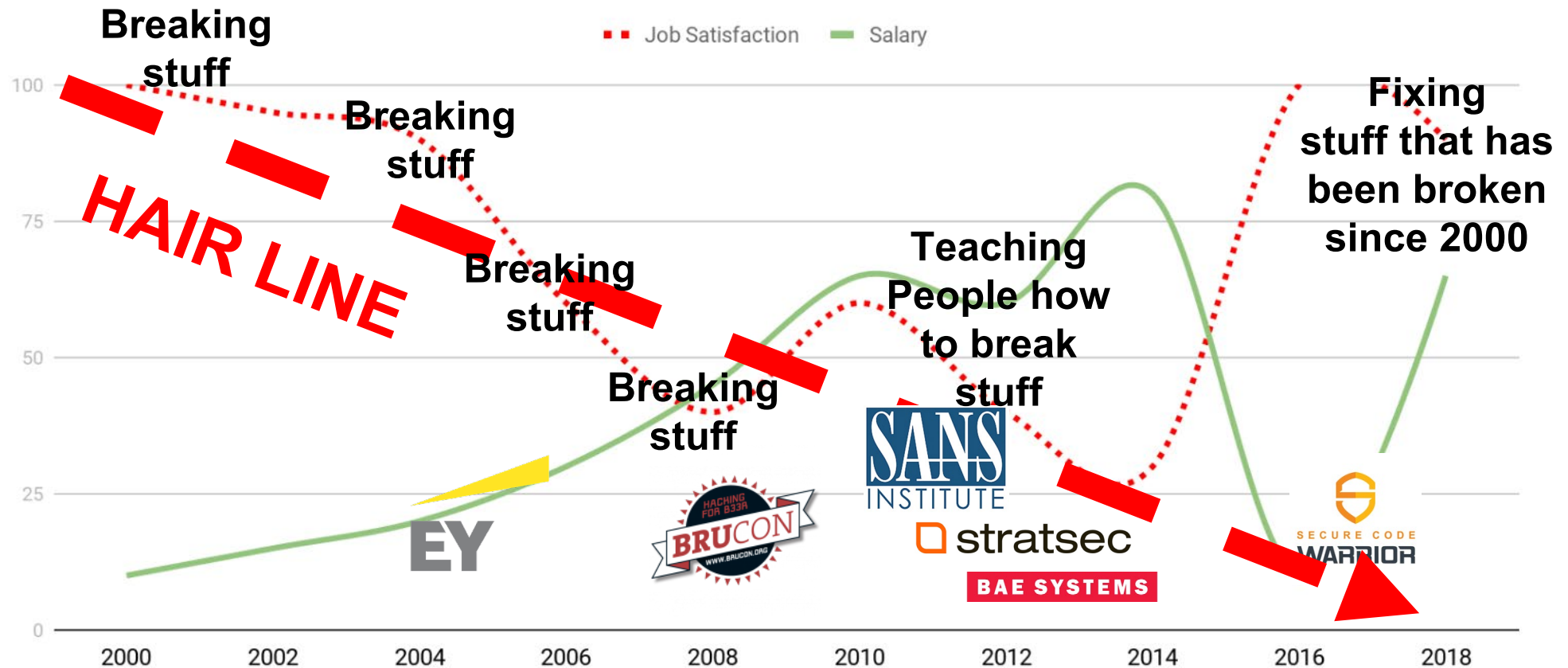
Matias Madou

CPO
Secure Code Warrior
mm@scw.io

#RSAC

Nice to meet you, I'm Pieter

#RSAC



... And I'm Matias Madou, Ph.D.



Matias Madou

Phd. Engineer, Built Sensei, CTO & Chief Product Officer

Matias is a researcher and developer with more than 15 years of hands-on software security experience. He has developed solutions for companies such as HP Fortify and his own company Sensei Security. Over his career, Matias has led multiple application security research projects which have led to commercial products and boasts over 10 patents under his belt. When he is away from his desk, Matias has served as an instructor for advanced application security training courses and regularly speaks at global conferences including RSA Conference, Black Hat, DefCon, BSIMM, OWASP AppSec and BruCon. Matias holds a Ph.D. in Computer Engineering from Ghent University, where he studied application security through program obfuscation to hide the inner workings of an application.

When it comes to security, we're not learning from our mistakes.



With the right **skills tools** and **support**, developers can be the **first line of defense** in their organisation.

They have the power to drive and maintain a **positive security culture**, keep best practice **highly visible** and **write secure code** from the beginning.

It's time to ~~SHIFT~~ **START** left.

Upskill and make an **IMPACT** as a security-first developer.

It's time to play a game.

Compete against your peers and challenge yourself to defeat common security vulnerabilities. Ready?



ACCOUNT REGISTRATION

1

WIFI: LAB3W02

PASSWORD: Rs@W3dam!abs

2

GO TO: info.securecodewarrior.com/rsatournament

3

FILL IN YOUR INFORMATION TO REGISTER



Follow us on Twitter for your chance to win even more cool prizes:
[@SecCodeWarrior](#) [#CodersConquerSecurity](#)

CHALLENGES

Web

Challenge Categories

Unvalidated Redirects and Forwards	1
Cross-Site Scripting (XSS)	1
Cross Site Request Forgery	1
Sensitive Data Storage	2
Insufficient Transport Layer Protection	2
Authentication	3
Injection Flaws	20

Tournament Languages *

Java Enterprise Edition (JSP) Kotlin Android SDK C# (.NET) MVC JavaScript Node.js (Express) Java Spring Java Android SDK C# (.NET) Web Forms

Objective-C iOS SDK Swift iOS SDK C++ Basic

Languages cannot be changed once participants have registered for this tournament

Total Number of Challenges *

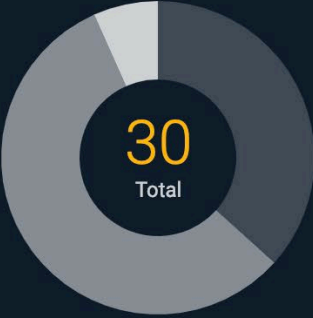
Set the total number of challenges to see approximately how long it would take an average developer to fully complete the tournament. We recommend a minimum of 24 challenges and the maximum will be determined based on the combination of languages selected above.

Approximately **2 hours** for an average developer to complete

30

The number of challenges cannot be changed once participants have registered for this tournament

- ▶ Level 5 (4 challenges)
- ▶ Level 6 (4 challenges)
- ▶ Level 7 (4 challenges)
- ▶ Level 8 (4 challenges)



30
Total

TYPE OF CHALLENGES

All challenges are based on the OWASP Top 10 and require participants to either:

✓ **Identify** a particular vulnerability within a block of code

✓ **Locate** a named vulnerability within a block of code

✓ **Fix** a vulnerable piece of code

+ **Combinations** of any two of the above

REGISTRATION LINK: info.securecodewarrior.com/rsatournament

SCORING & ATTEMPTS

EARNING
POINTS

Base Stage Score

This is the base score of stages for each challenge in the tournament.

Easy

100

Medium

200

Hard

300

Large Codebase Multiplier

Where the challenge code is a large codebase of a more realistic full application, the following multiplier will be applied to the score.

x

2

Attempts Per Stage

The number of times a participant can take a challenge stage before failing it. The percentage of points awarded will typically decrease over multiple attempts and an optional penalty can be applied for each incorrect attempt to discourage guessing.

Maximum Attempts

3

Attempt 1

Correct % Awarded

100

%

Incorrect % Penalty

0

%

Attempt 2

Correct % Awarded

60

%

Incorrect % Penalty

0

%

Attempt 3

Correct % Awarded

30

%

Incorrect % Penalty

0

%

LIVES
LOST

THE “COST” OF USING HINTS – *Your Mileage Varies*

Use these to
learn more
about each
vulnerability
THEY'RE FREE!

Hint Penalties
If hints are enabled above, these are the percentage penalties that will be applied if a participant uses hints for the various stage types.

Locate Vulnerability Stages

Hint 1 %	Hint 2 %	Hint 3 %	Hint 4 %
0 %	-5 %	-35 %	-60 %

Identify Vulnerability Stages

Hint 1 %	Hint 2 %	Hint 3 %
0 %	-50 %	-50 %

Pick Solution Stages

Hint 1 %	Hint 2 %	Hint 3 %
-33.300000000000004 %	-33.300000000000004 %	-33.300000000000004 %

HOUSEKEEPING

You will have **until 10:30AM** to complete

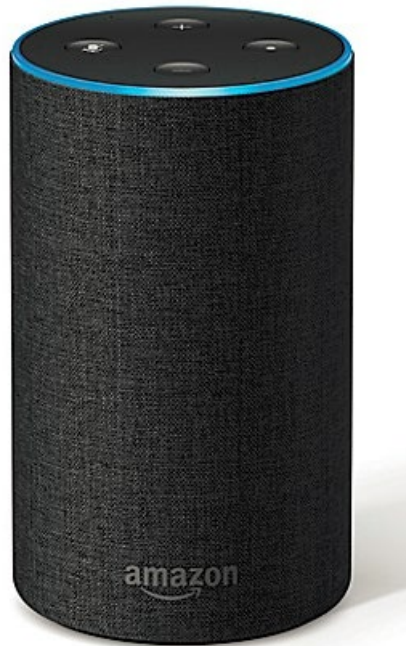
A short survey will be sent out via email upon conclusion, and will take just two minutes to complete. Please fill this out as it helps us create a better tournament experience.

Stick around for the winners announcement and prizes!

Tweet, tweet and keep on tweeting: @SecCodeWarrior

REGISTRATION LINK: info.securecodewarrior.com/rsatournament

PRIZES



REGISTRATION LINK: info.securecodewarrior.com/rsatournament

LET'S DO THIS!

Tournaments ?

Tournaments provide an environment for developers to engage in friendly competition while at the same time improving their secure coding skills and learning about security weaknesses.



2019 RSA Tournament

Welcome to the Secure Code Warrior Tournament! All challenges are based on the OWASP Top 10 and will require you to:

- Identify a particular vulnerability within a code snippet
- Locate a named vulnerability within a code snippet
- Fix a vulnerable piece of code

Earn enough points to climb the top of the leaderboard and be crowned the ultimate 'Secure Code Warrior.'

Start: Mar 6, 2019 11:30:00 AM

Finish: Mar 6, 2019 1:30:00 PM

0 developers registered

rSa_19

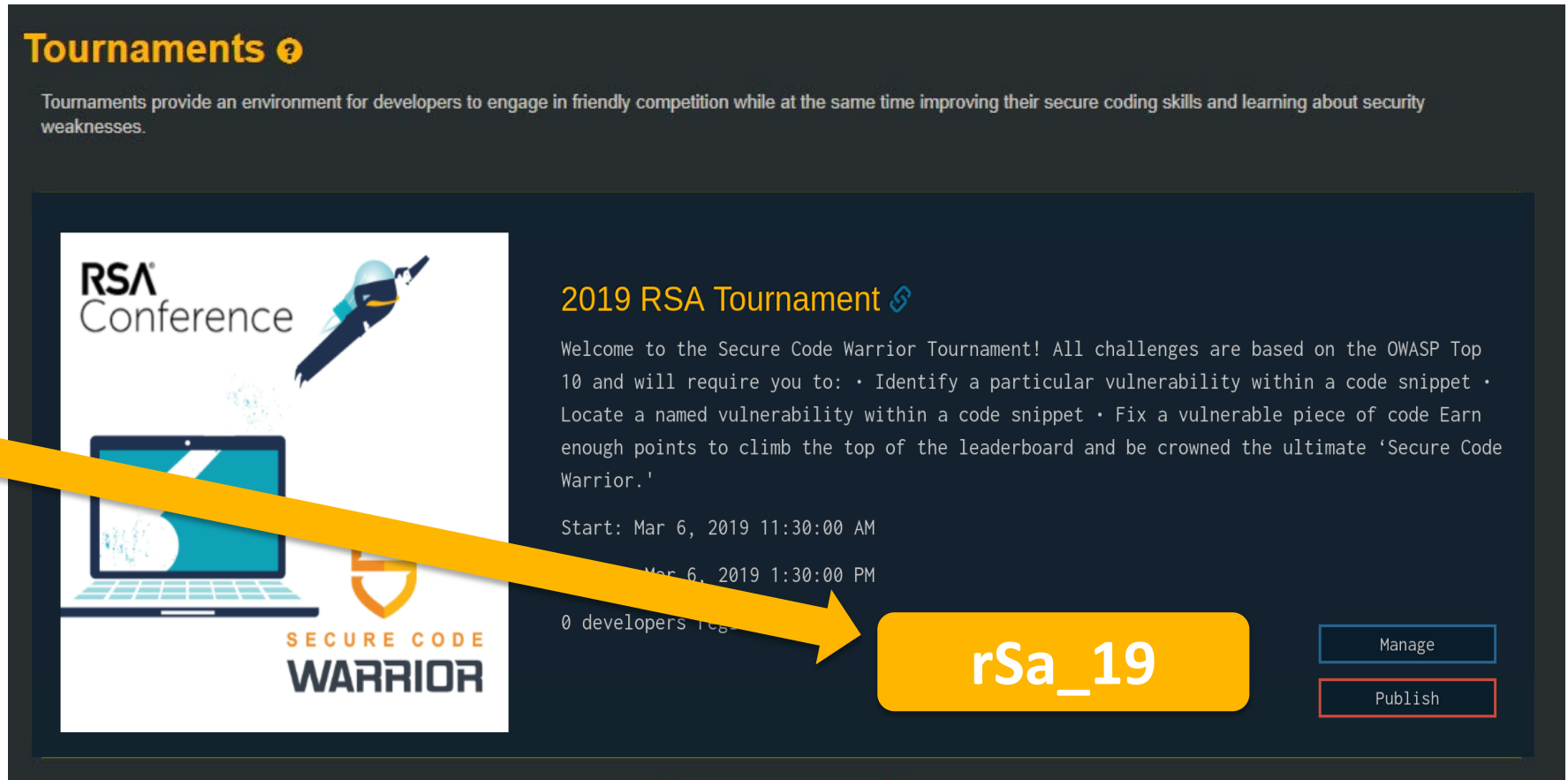
Manage

Publish

REGISTRATION LINK: info.securecodewarrior.com/rsatournament


PS: DON'T FORGET TO HAVE SOME FUN!

You need this
code to
Participate



Tournaments ⓘ

Tournaments provide an environment for developers to engage in friendly competition while at the same time improving their secure coding skills and learning about security weaknesses.



2019 RSA Tournament ⓘ

Welcome to the Secure Code Warrior Tournament! All challenges are based on the OWASP Top 10 and will require you to:

- Identify a particular vulnerability within a code snippet
- Locate a named vulnerability within a code snippet
- Fix a vulnerable piece of code

Earn enough points to climb the top of the leaderboard and be crowned the ultimate 'Secure Code Warrior.'

Start: Mar 6, 2019 11:30:00 AM

End: Mar 6, 2019 1:30:00 PM

0 developers registered

rSa_19

Manage

Publish

REGISTRATION LINK: info.securecodewarrior.com/rsatournament

FORGETTING SOMETHING?

WIFI: **LAB3W02** PASSWORD: **Rs@W3dam!abs**

GO TO: **info.securecodewarrior.com/rsatournament**

FILL IN YOUR INFORMATION

CLICK ON THE TOURNAMENT TAB AND ENTER JOIN CODE **rSa_19**

THE TOURNAMENT WILL GO LIVE AT **NOW** and WILL END **@ 10:30AM**

Follow us on Twitter for your chance to win even more cool prizes:
@SecCodeWarrior #CodersConquerSecurity

> Need a quick kick starter on Secure Coding?



MANICODE
SECURE CODING EDUCATION

A little background dirt...

jim@manicode.com



@manicode

- Former OWASP Global Board Member
- Project manager of the OWASP Cheat Sheet Series and several other OWASP projects
- 20+ years of software development experience
- Author of "Iron-Clad Java, Building Secure Web Applications" from McGraw-Hill/Oracle-Press
- Kauai, Hawaii Resident



ALL DATA ENTERING YOUR SOFTWARE MUST BE VALIDATED

Input that is not directly entered by the user is typically **less prone to validation**.

Attacks discussed in this section apply to external input from any client-side source

- Standard form input control
- Read-only HTML form controls (drop down lists, radio buttons, hidden fields, etc.)
- HTTP Cookie Values
- HTTP Headers
- Embedded URL parameters (e.g., in the GET request)

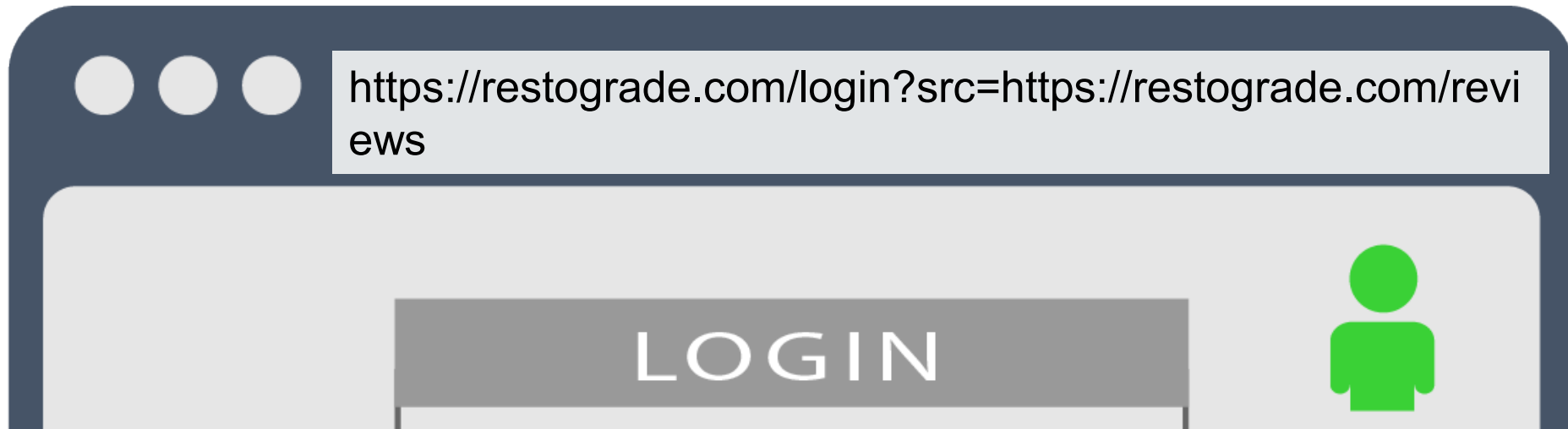


RSA®Conference2019

Unvalidated Redirects



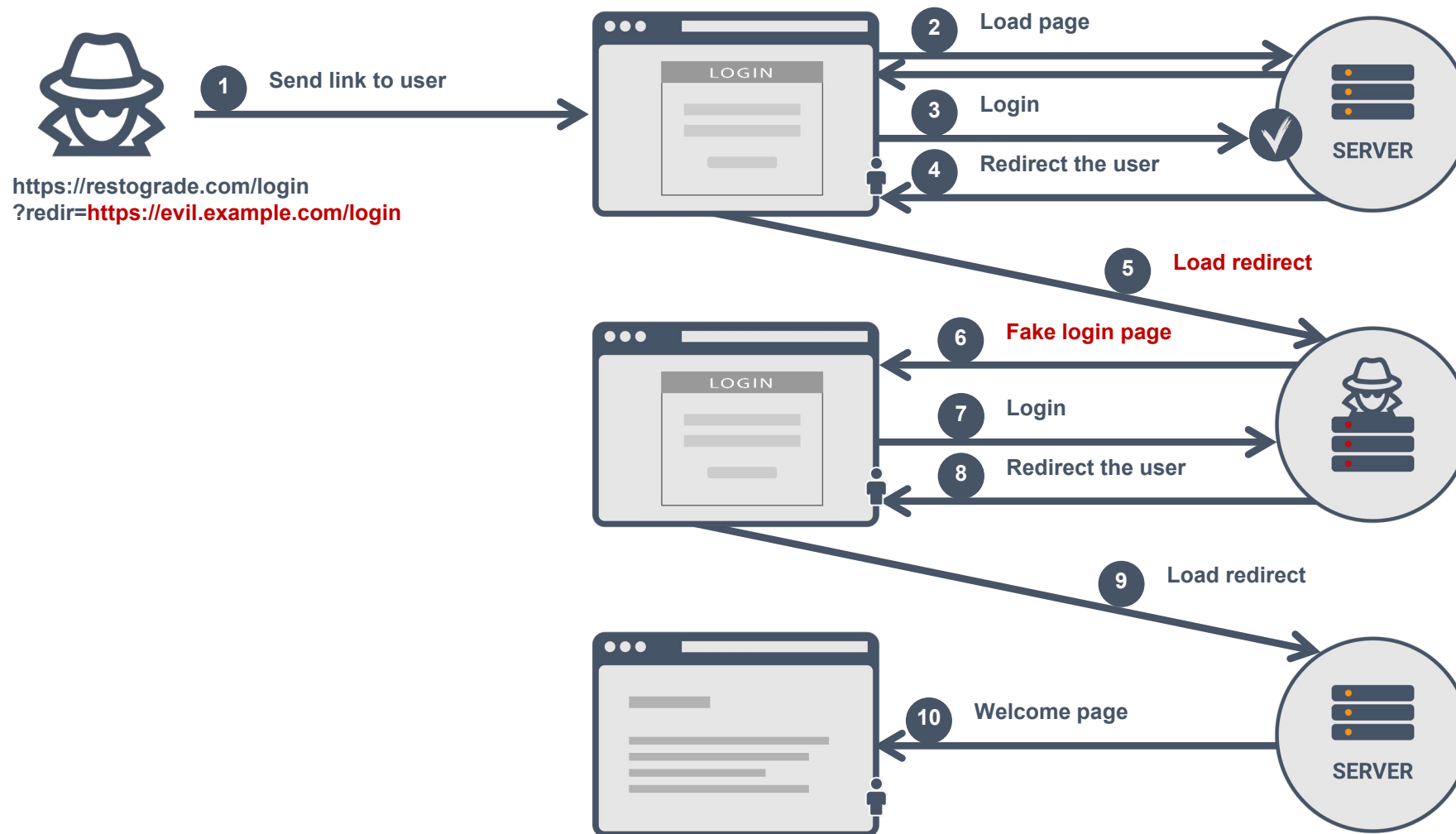
Can you spot the **problem** here?



Unvalidated / open redirects

- Redirects are often used to establish a user-friendly flow
 - Common example is redirecting back to the original page after login
 - The redirect URL is added by the server as a request parameter, and passed around
- The redirect URL should be considered untrusted data
 - It is generated by the server, but becomes untrusted when sent to the client
- Attackers can use unvalidated redirects to trick other users
 - Ideal launching platform for social engineering attacks
 - The first step is a legitimate application URL, so the attack is difficult to spot





- I've got this!

```
url.contains("restograde.com")
```



- I've got this!

```
url.contains("restograde.com")
```

```
url.startsWith("https://restograde.com")
```

```
https://restograde.com.example.com/yougothacked
```

```
https://hackrestograde.com/
```

```
https://example.com/restograde.com/yougothacked
```



Securing redirects

- Applications should only redirect to valid destinations
 - Match the given redirect URL against a whitelist of valid URLs
 - When doing partial matching, at least check the full origin, including the path separator

```
url.equals("https://restograde.com") ||  
url.startsWith("https://restograde.com/")
```

- A better, more secure option is to keep the URL on the server side
 - When the server generates the URL, the value is still considered safe
 - Unless it is extracted from client-side data, such as the *Referer* header
 - The server can store it in a server-side (unauthenticated) session
 - This value cannot be manipulated by an attacker, so remains safe to use



Validating Untrusted URLs in Java (.NET)

```
public static String validateURL(String rawURI)
throws ValidationException {

    // throws URISyntaxException if invalid URI
    URI uri = new URI(rawURI);

    if (!uri.isAbsolute()) throw new ValidationException ("not an absolute uri");

    // don't allows javascript urls, etc...
    if (!"http".equals(uri.getScheme()) && !"https".equals(uri.getScheme())) throw
new ValidationException("we only support http(s) urls");

    // who legitimately uses user-infos in their urls?!?
    if (uri.getUserInfo() != null) throw new ValidationException("this can only be
trouble");

    // check: uri.getHost() against whitelist/blacklist?
    // check: uri.getPort() for shenanigans?
    return uri.toASCIIString();
}
```



Big Picture

Libraries/Vulns	CR-LF Injection			URL Parsing		
	Path	Host	SNI	Port Injection	Host Injection	Path Injection
Python httplib	☠	☠	☠			
Python urllib		☠	☠		☠	
Python urllib2		☠	☠			
Ruby Net::HTTP	☠	☠	☠			
Java net.URL		☠			☠	
Perl LWP			☠	☠		
NodeJS http	☠					☠
PHP http_wrapper				☠	☠	
Wget		☠	☠			
cURL				☠	☠	



Orange Tsai



	cURL / libcurl
PHP parse_url	☠
Perl URI	☠
Ruby uri	
Ruby addressable	☠
NodeJS url	☠
Java net.URL	
Python urlparse	
Go net/url	☠



Orange Tsai



Mitigations

- Application layer

Use the only IP and hostname, do not reuse the input URL

- Network layer

Using Firewall or NetWork Policy to block Intranet traffics

- Projects

SafeCurl by @fin1te

Advocate by @JordanMilne



Orange Tsai





It's been a pleasure.

jim@manico.net

JIM MANICO OWASP Board Member, Secure Coding Instructor

www.manicode.com

RSA®Conference2019

PowerUp Your Developers with Security

Weak Authentication

Cross-Site Scripting

SQL Injection

Sensitive Data Exposure

Secure Code Warrior

```
function generate_license_key( $l
    $optio
    $price_id' : false;
    $edd_sl_download_id', $
    $edd_sl_download_pr
    $edd_sl_payment_id', $pa
    $edd_sl_key', $license_k
    $edd_sl_user_id', $user_
    $edd_sl_status', 'inacti
    $edd_sl_site_count', 0 );
    $this->get_license_length( $license
    $edd_sl_key
```

TN-491

How are you going to apply what you've learned today?

- You **have the power** to create better outcomes from the star
- Distribute knowledge to scale AppSec and build a positive security culture.
- Define good patterns and re-use them.
- **Put some fun into everything**





SECURE CODE
WARRIOR



SECURECODEWARRIOR.COM



BLOG.SECURECODEWARRIOR.COM



[@SECCodeWarrior](https://twitter.com/SECCodeWarrior)



LINKEDIN.COM/COMPANY/SECURE-CODE-WARRIOR



FACEBOOK.COM/SECURECODEWARRIOR/