THE DEVELOPER'S CONFERENCE

# Software Security - secDevLabs

**Daniel Carlier**
Analista de Segurança
Globo.com

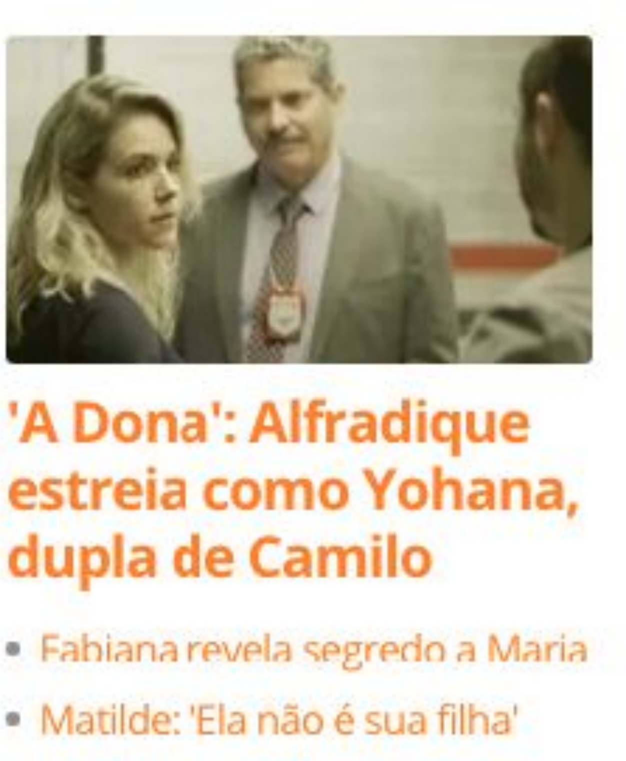
# secDevLabs

https://github.com/globocom/secDevLabs

# Agenda

1. Motivações
2. Planejamento do treinamento
3. Como funciona a dinâmica
4. Resultados dentro da Globo.com

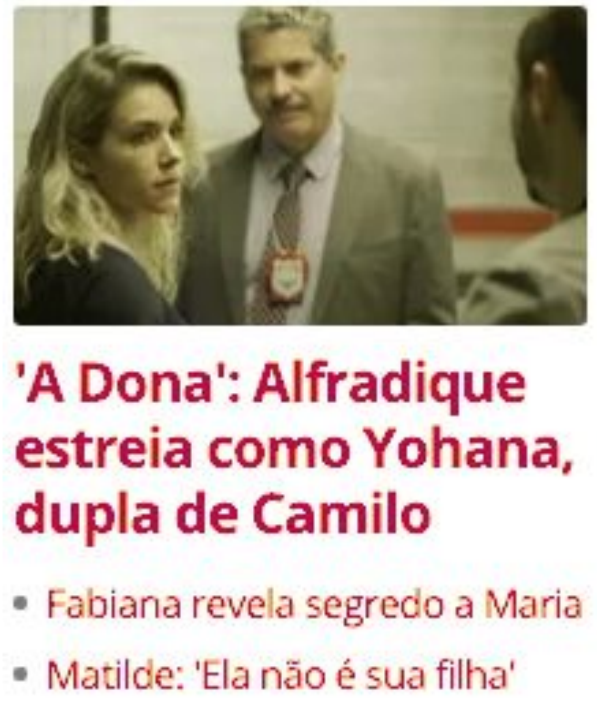
# 1. Um dia na vida de um desenvolvedor



'A Dona': Alfradique estreia como Yohana, dupla de Camilo

- Fabiana revela segredo a Maria
- Matilde: 'Ela não é sua filha'

Feature



'A Dona': Alfradique estreia como Yohana, dupla de Camilo

- Fabiana revela segredo a Maria
- Matilde: 'Ela não é sua filha'

# 1. Um dia na vida de um desenvolvedor

| Backlog | Sprint | Doing | Review | Done |
|---|---|---|---|---|
| Tarefa | | | | |
| Tarefa | | | | |
| Tarefa | | | | |
| Tarefa | | | | |

# 1. Um dia na vida de um desenvolvedor

| Backlog | Sprint | Doing | Review | Done |
|---|---|---|---|---|
| Tarefa | | Tarefa | | |
| Tarefa | | Letra Vinho | | |
| Tarefa | | Tarefa | | |
| Tarefa | | Tarefa | | |

# 1. Um dia na vida de um desenvolvedor

| Backlog | Sprint | Doing | Review | Done |
|---|---|---|---|---|
| Tarefa | | | Tarefa | |
| Tarefa | | Letra Vinho | | |
| Tarefa | | | Tarefa | |
| Tarefa | | Tarefa | | |

# 1. Um dia na vida de um desenvolvedor

| Backlog | Sprint | Doing | Review | Done |
|---|---|---|---|---|
| Tarefa | | | | Tarefa |
| Tarefa | | | Letra Vinho | |
| Tarefa | | | Tarefa | |
| Tarefa | | | Tarefa | |

# 1. Um dia na vida de um desenvolvedor

| Backlog | Sprint | Doing | Review | Done |
|---|---|---|---|---|
| Tarefa | | | | Tarefa |
| Tarefa | | | | Letra Vinho |
| Tarefa | | | | Tarefa |
| Tarefa | | | | Tarefa |

# Vamos fazer um curso de segurança?🐼 🔏

# 1. Um dia na vida de um desenvolvedor

| Backlog | Sprint | Doing | Review | Done |
|---|---|---|---|---|
| Tarefa | | Curso 🐼 | | |
| Tarefa | | Curso 🐼 | | |
| Tarefa | | Curso 🐼 | | |
| Tarefa | | Curso 🐼 | | |

# 1. Um dia na vida de um desenvolvedor

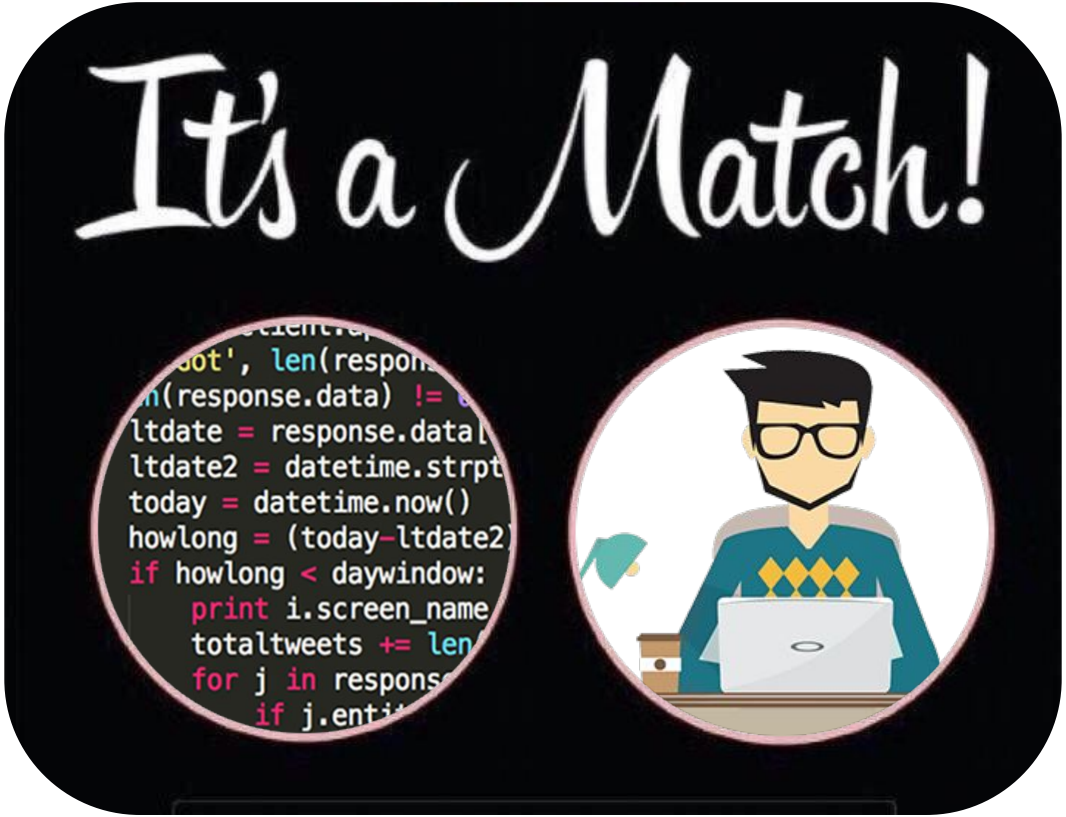
| Backlog | Sprint | Doing | Review | Done |
|---|---|---|---|---|
| Tarefa | | | | Curso<br>🐼 |
| Tarefa | | | | Curso<br>🐼 |
| Tarefa | | | | Curso<br>🐼 |
| Tarefa | | | | Curso<br>🐼 |

# 1. Um dia na vida de um desenvolvedor

| Backlog | Sprint | Doing | Review | Done |
|---|---|---|---|---|
| Tarefa | | | | Curso 🐼 |
| Tarefa | | | | Curso 🐼 |
| Tarefa | | | | Curso 🐼 |
| Tarefa | | | | Curso 🐼 |

# Let's Hack! 🐼

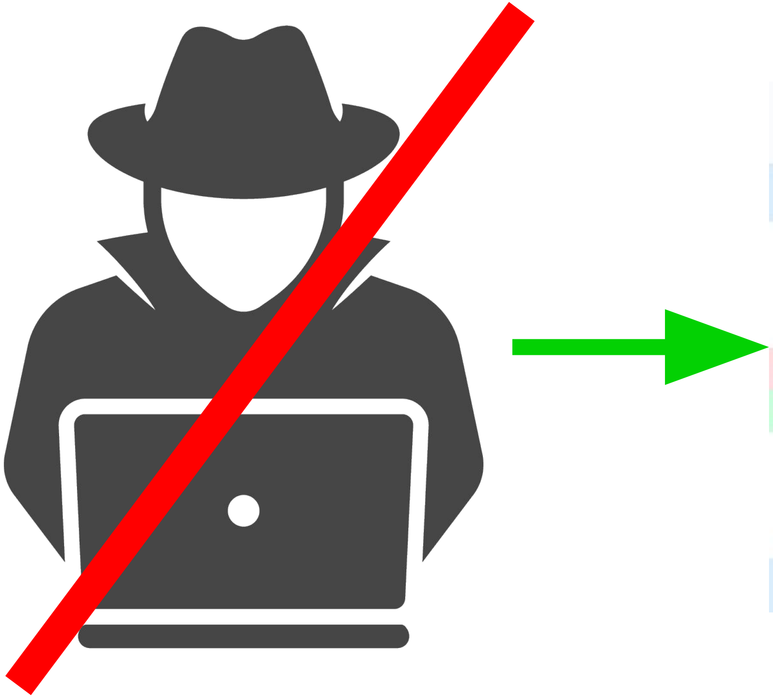
# 2. Planejamento do treinamento

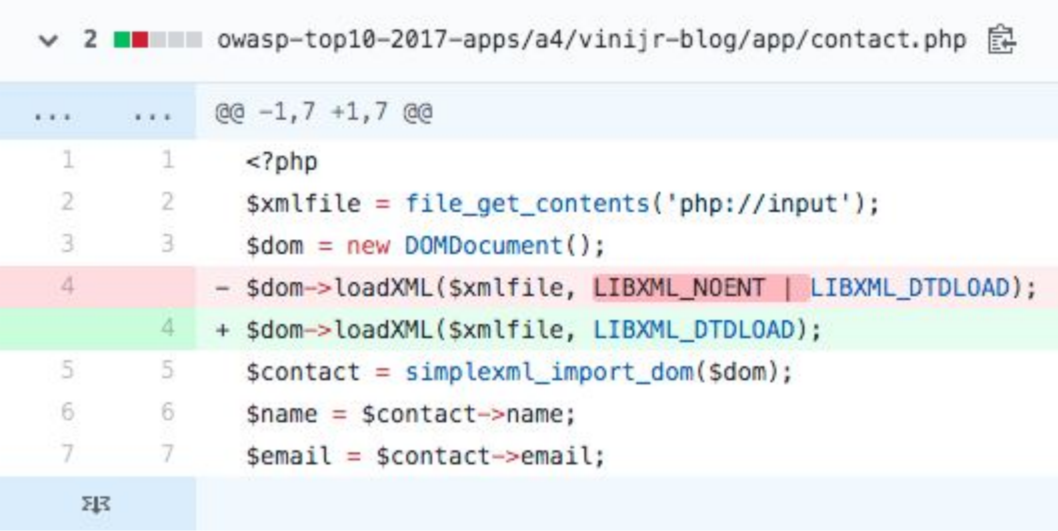


```
client.
ot', len(respon
n(response.data) != 
ltdate = response.data[
ltdate2 = datetime.strpt
today = datetime.now()
howlong = (today-ltdate2)
if howlong < daywindow:
    print i.screen_name
    totaltweets += len
    for j in respons
        if j.enti
```

# 2. Planejamento do treinamento

# 2. Planejamento do treinamento



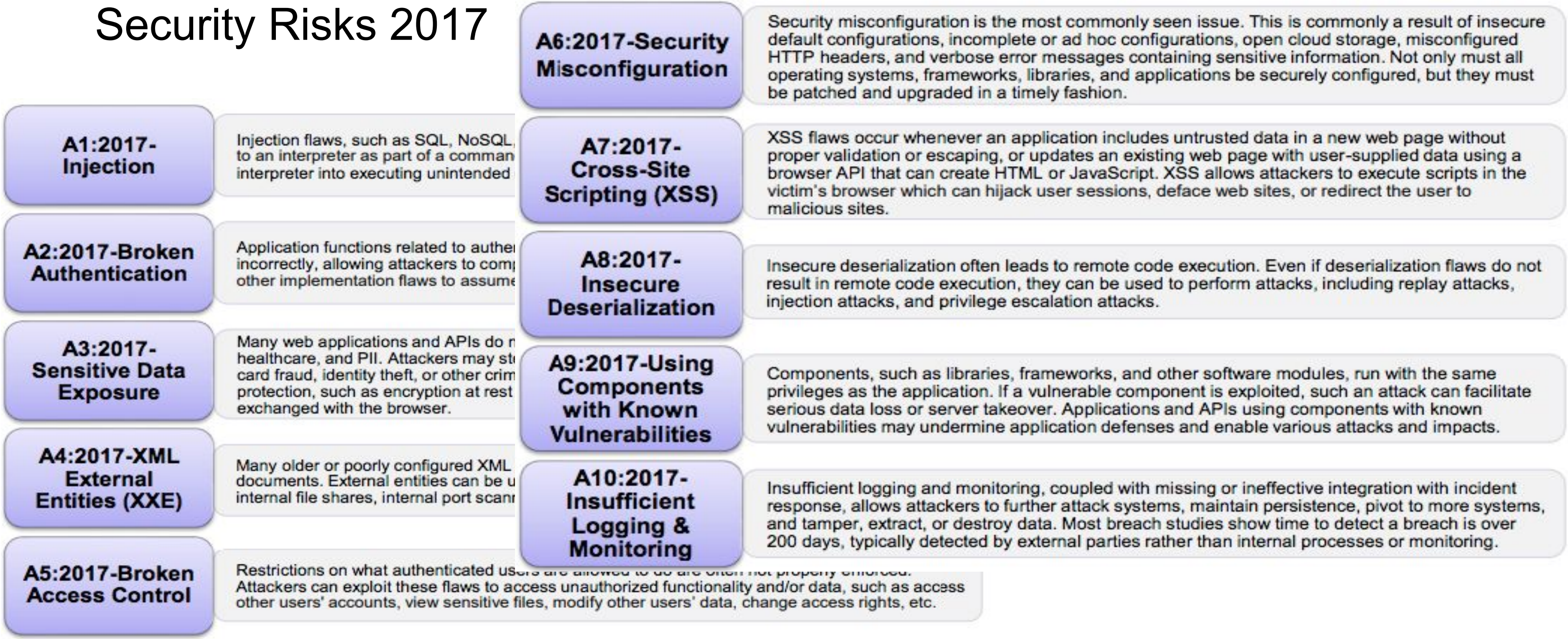


2 owasp-top10-2017-apps/a4/vinijr-blog/app/contact.php

```
@@ -1,7 +1,7 @@
  <?php
  $xmlfile = file_get_contents('php://input');
  $dom = new DOMDocument();
- $dom->loadXML($xmlfile, LIBXML_NOENT | LIBXML_DTDLOAD);
+ $dom->loadXML($xmlfile, LIBXML_DTDLOAD);
  $contact = simplexml_import_dom($dom);
  $name = $contact->name;
  $email = $contact->email;
```

# 2. Planejamento do treinamento

## OWASP Top 10 Application Security Risks 2017



**A1:2017-Injection**

Injection flaws, such as SQL, NoSQL,
to an interpreter as part of a comman
interpreter into executing unintended

**A2:2017-Broken Authentication**

Application functions related to authe
incorrectly, allowing attackers to com
other implementation flaws to assume

**A3:2017-Sensitive Data Exposure**

Many web applications and APIs do n
healthcare, and PII. Attackers may st
card fraud, identity theft, or other crim
protection, such as encryption at rest
exchanged with the browser.

**A4:2017-XML External Entities (XXE)**

Many older or poorly configured XML
documents. External entities can be u
internal file shares, internal port scann

**A5:2017-Broken Access Control**

Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

**A6:2017-Security Misconfiguration**

Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched and upgraded in a timely fashion.

**A7:2017-Cross-Site Scripting (XSS)**

XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

**A8:2017-Insecure Deserialization**

Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

**A9:2017-Using Components with Known Vulnerabilities**

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

**A10:2017-Insufficient Logging & Monitoring**

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

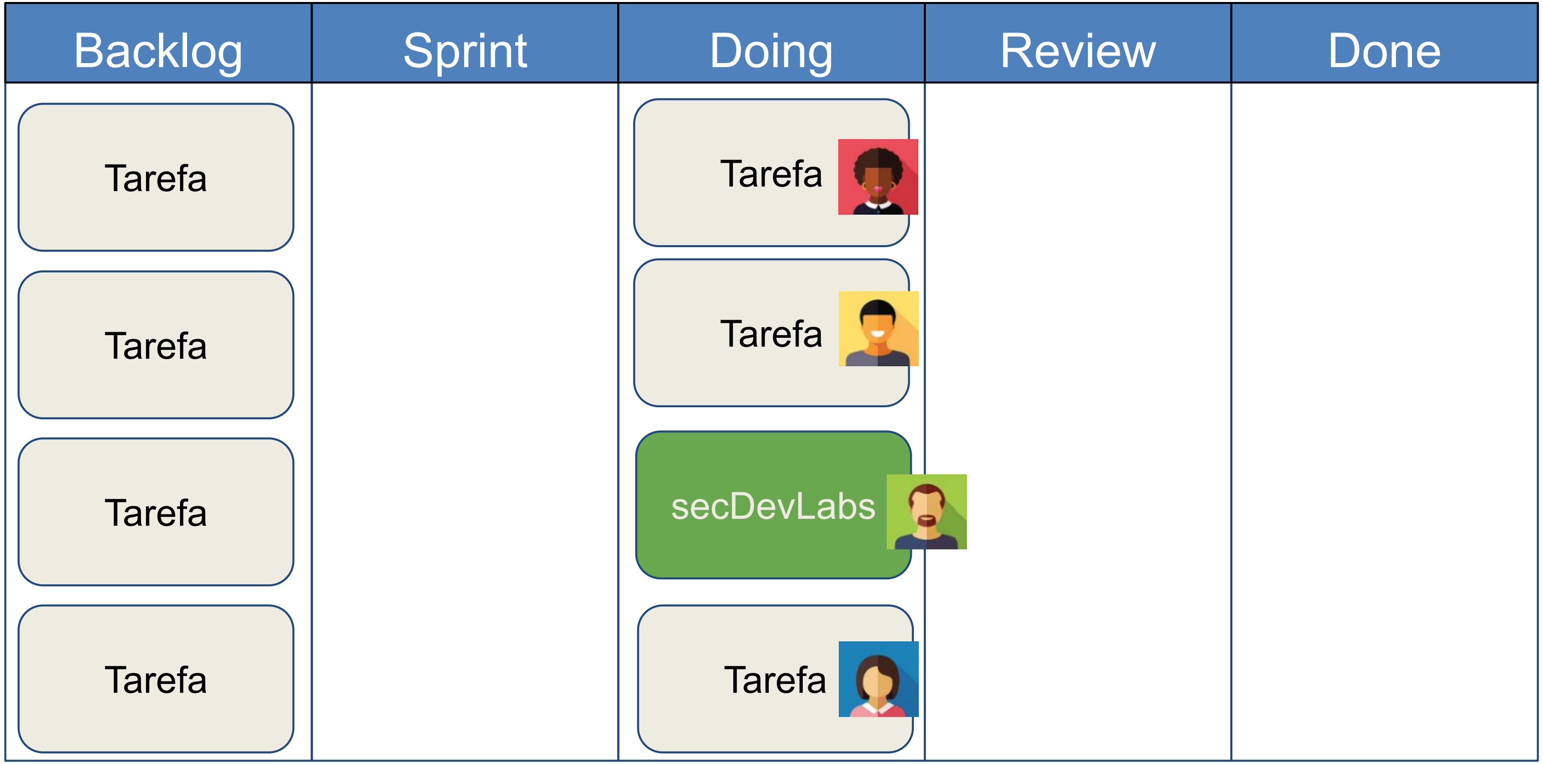
# 2. Planejamento do treinamento

| Backlog | Sprint | Doing | Review | Done |
|---|---|---|---|---|
| Tarefa | | secDevLabs | | |
| Tarefa | | Tarefa | | |
| Tarefa | | Tarefa | | |
| Tarefa | | Tarefa | | |

# 2. Planejamento do treinamento

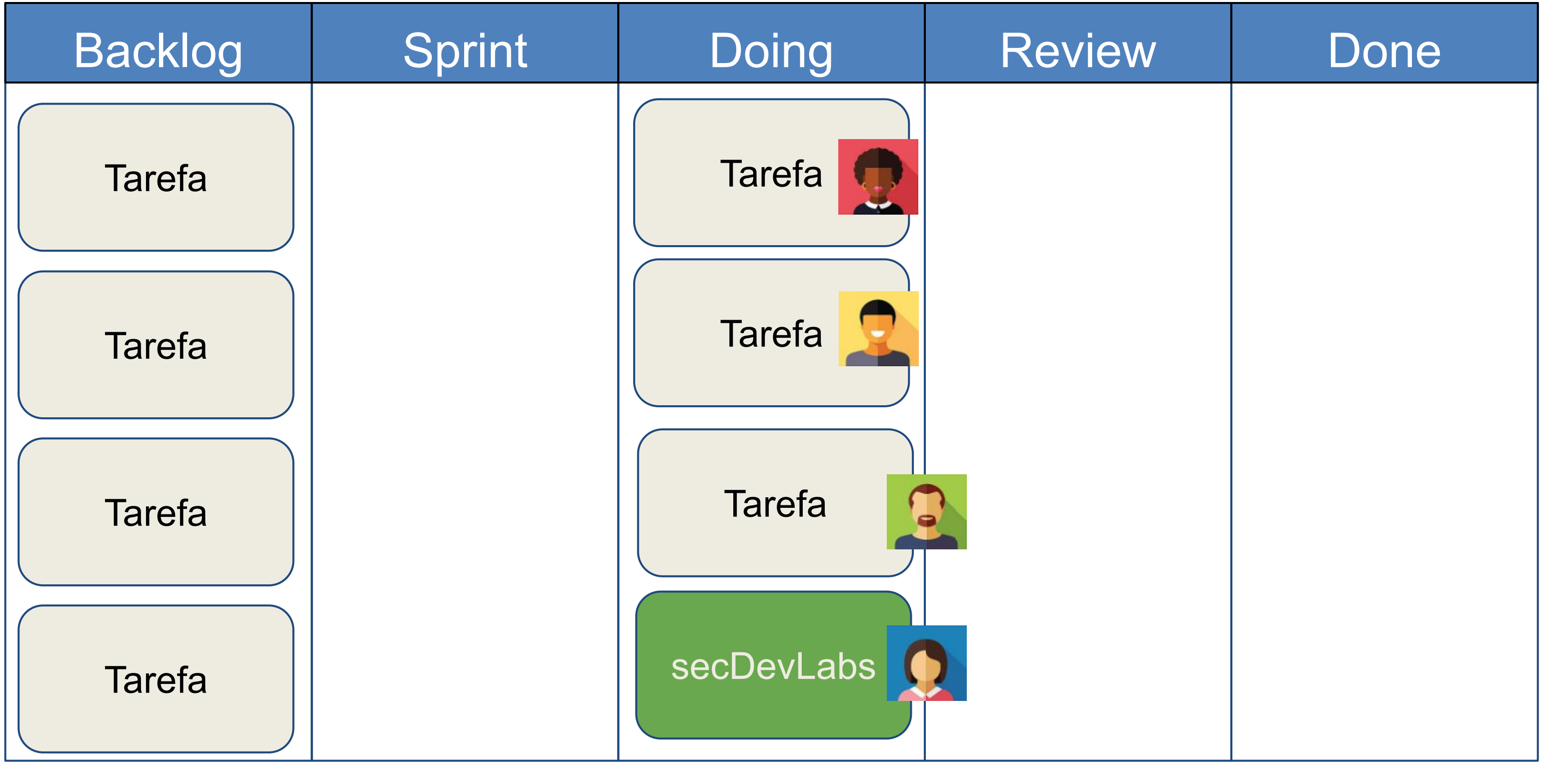
| Backlog | Sprint | Doing | Review | Done |
|---|---|---|---|---|
| Tarefa |  | Tarefa |  |  |
| Tarefa |  | secDevLabs |  |  |
| Tarefa |  | Tarefa |  |  |
| Tarefa |  | Tarefa |  |  |

# 2. Planejamento do treinamento



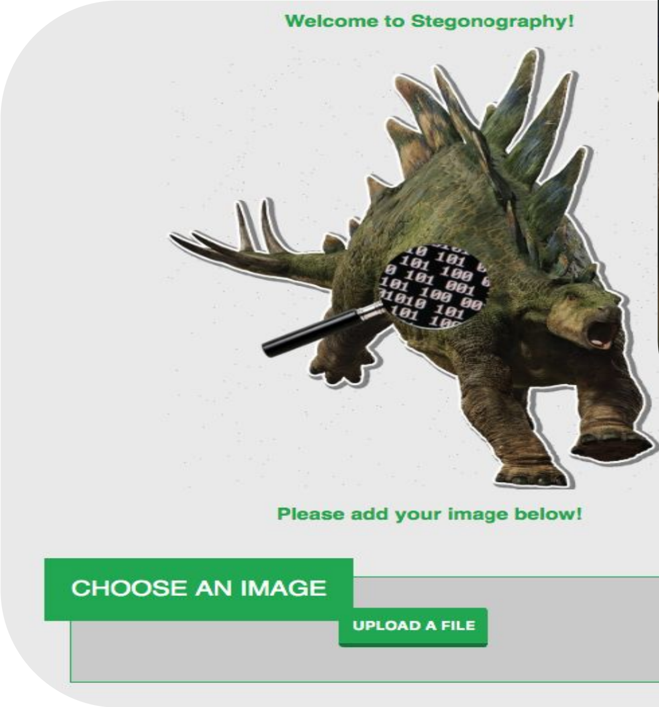
| Backlog | Sprint | Doing | Review | Done |
|---|---|---|---|---|
| Tarefa | | Tarefa | | |
| Tarefa | | Tarefa | | |
| Tarefa | | secDevLabs | | |
| Tarefa | | Tarefa | | |

# 2. Planejamento do treinamento



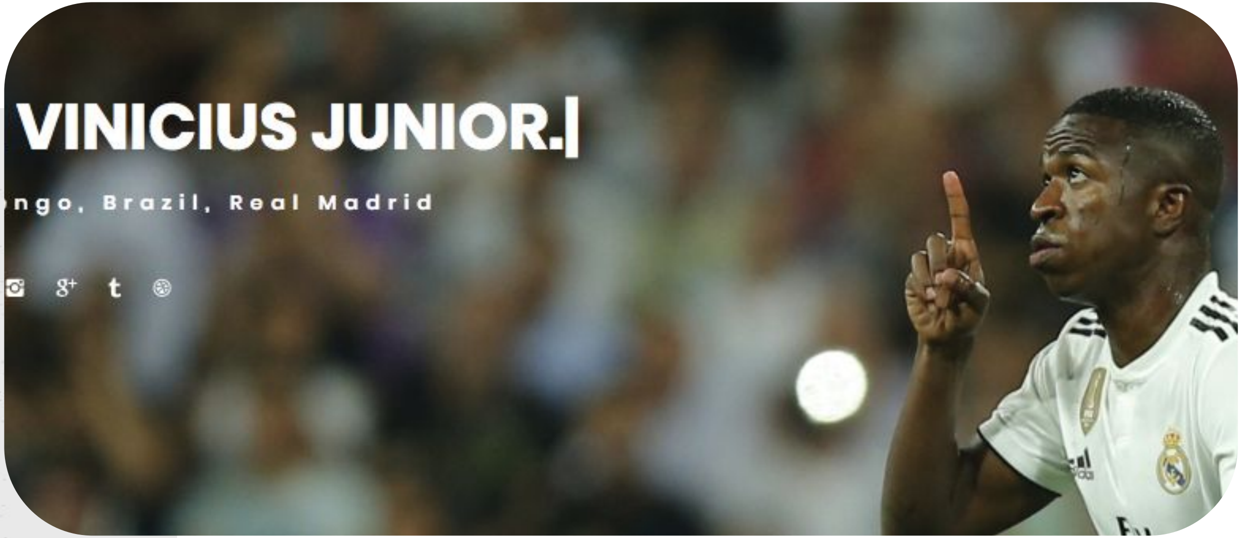
| Backlog | Sprint | Doing | Review | Done |
|---|---|---|---|---|
| Tarefa | | Tarefa | | |
| Tarefa | | Tarefa | | |
| Tarefa | | Tarefa | | |
| Tarefa | | secDevLabs | | |

# 3. Como funciona a dinâmica



Welcome to Stegonography!

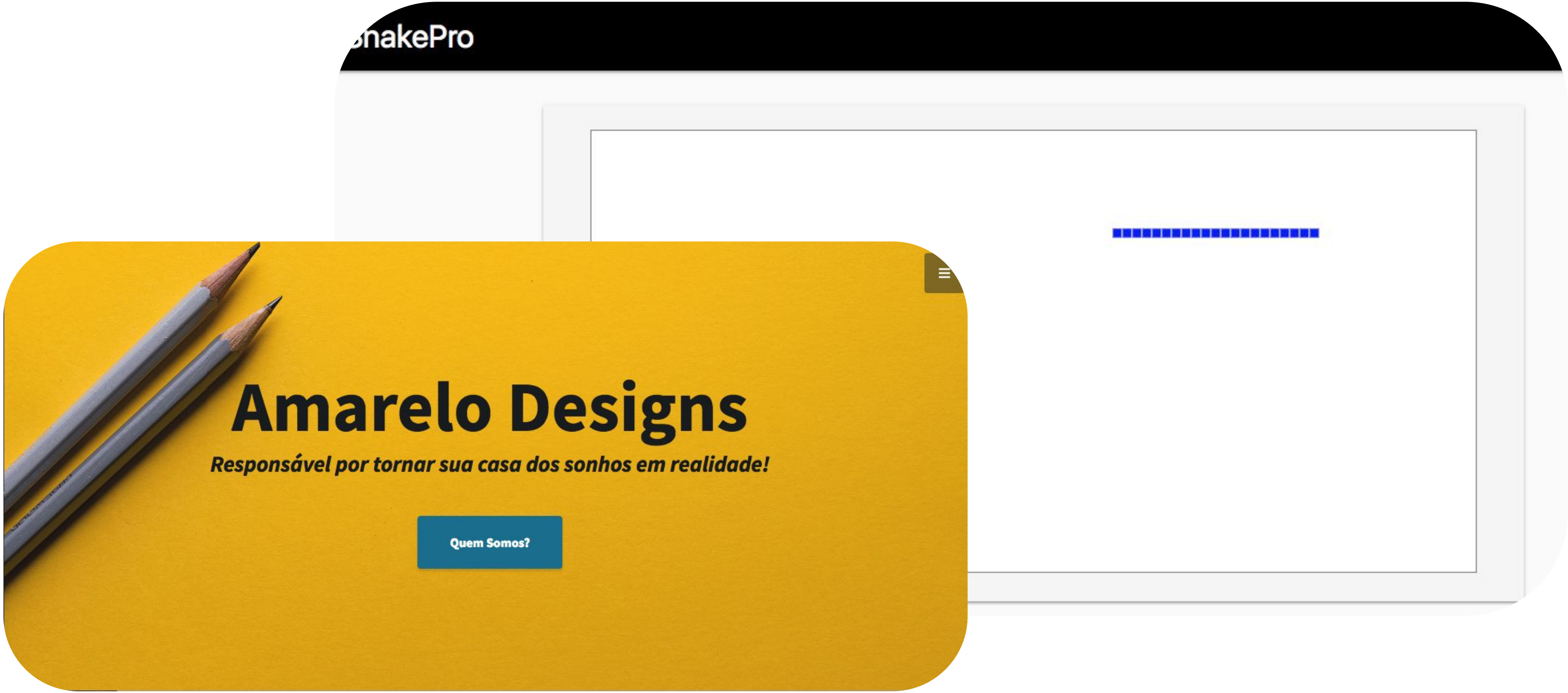
Please add your image below!



VINICIUS JUNIOR.|

ngo, Brazil, Real Madrid

# 3. Como funciona a dinâmica

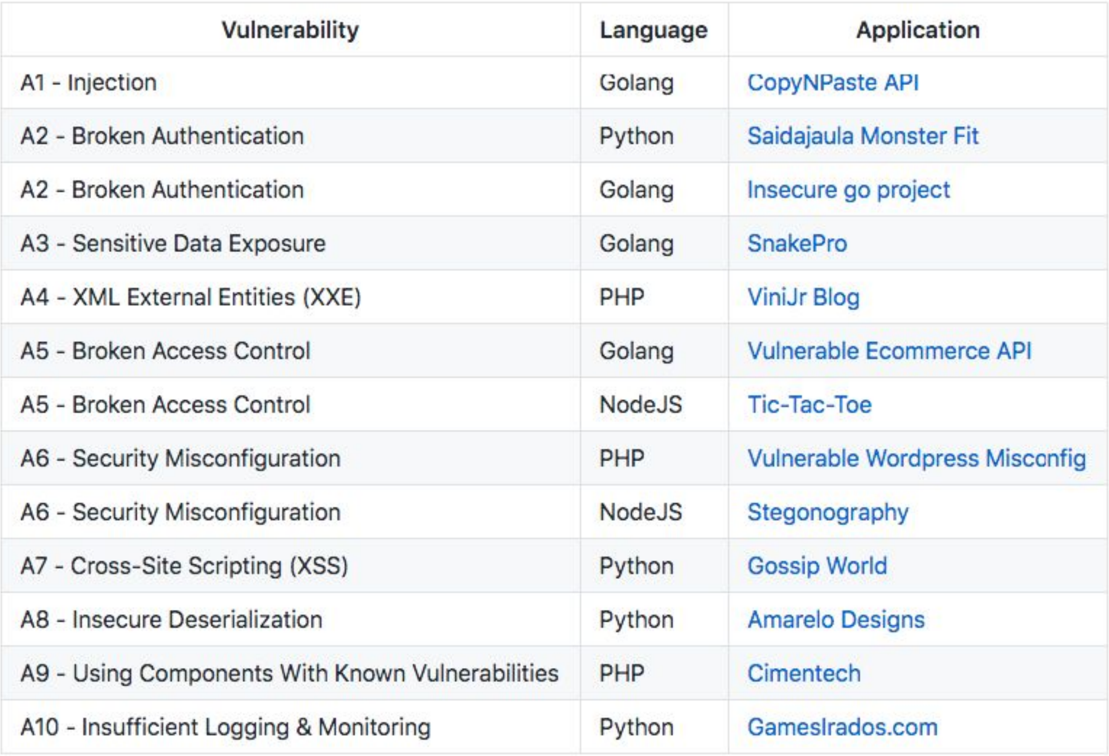


SnakePro

# Amarelo Designs

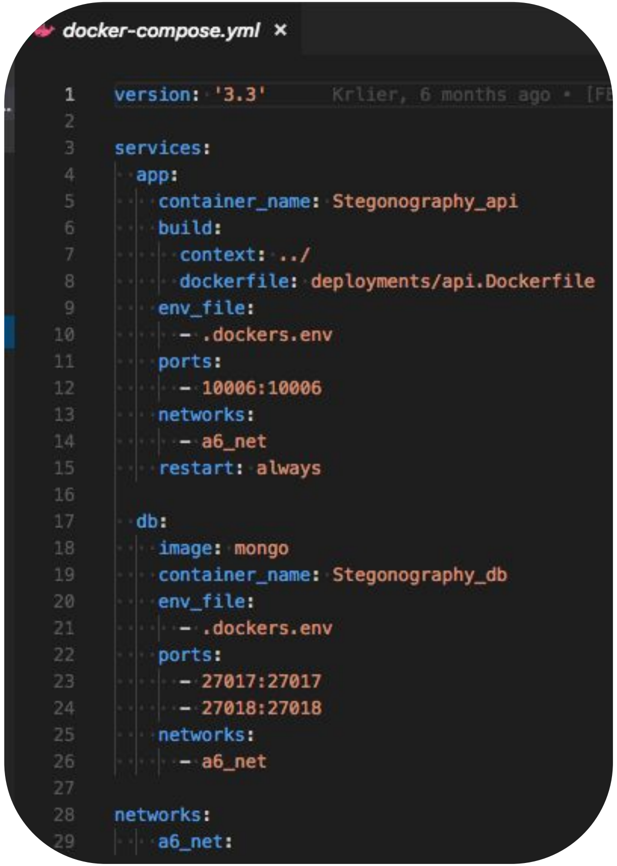
*Responsável por tornar sua casa dos sonhos em realidade!*

# 3. Como funciona a dinâmica



| Vulnerability | Language | Application |
|---|---|---|
| A1 - Injection | Golang | CopyNPaste API |
| A2 - Broken Authentication | Python | Saidajaula Monster Fit |
| A2 - Broken Authentication | Golang | Insecure go project |
| A3 - Sensitive Data Exposure | Golang | SnakePro |
| A4 - XML External Entities (XXE) | PHP | ViniJr Blog |
| A5 - Broken Access Control | Golang | Vulnerable Ecommerce API |
| A5 - Broken Access Control | NodeJS | Tic-Tac-Toe |
| A6 - Security Misconfiguration | PHP | Vulnerable Wordpress Misconfig |
| A6 - Security Misconfiguration | NodeJS | Stegonography |
| A7 - Cross-Site Scripting (XSS) | Python | Gossip World |
| A8 - Insecure Deserialization | Python | Amarelo Designs |
| A9 - Using Components With Known Vulnerabilities | PHP | Cimentech |
| A10 - Insufficient Logging & Monitoring | Python | GamesIrados.com |

# 3. Como funciona a dinâmica



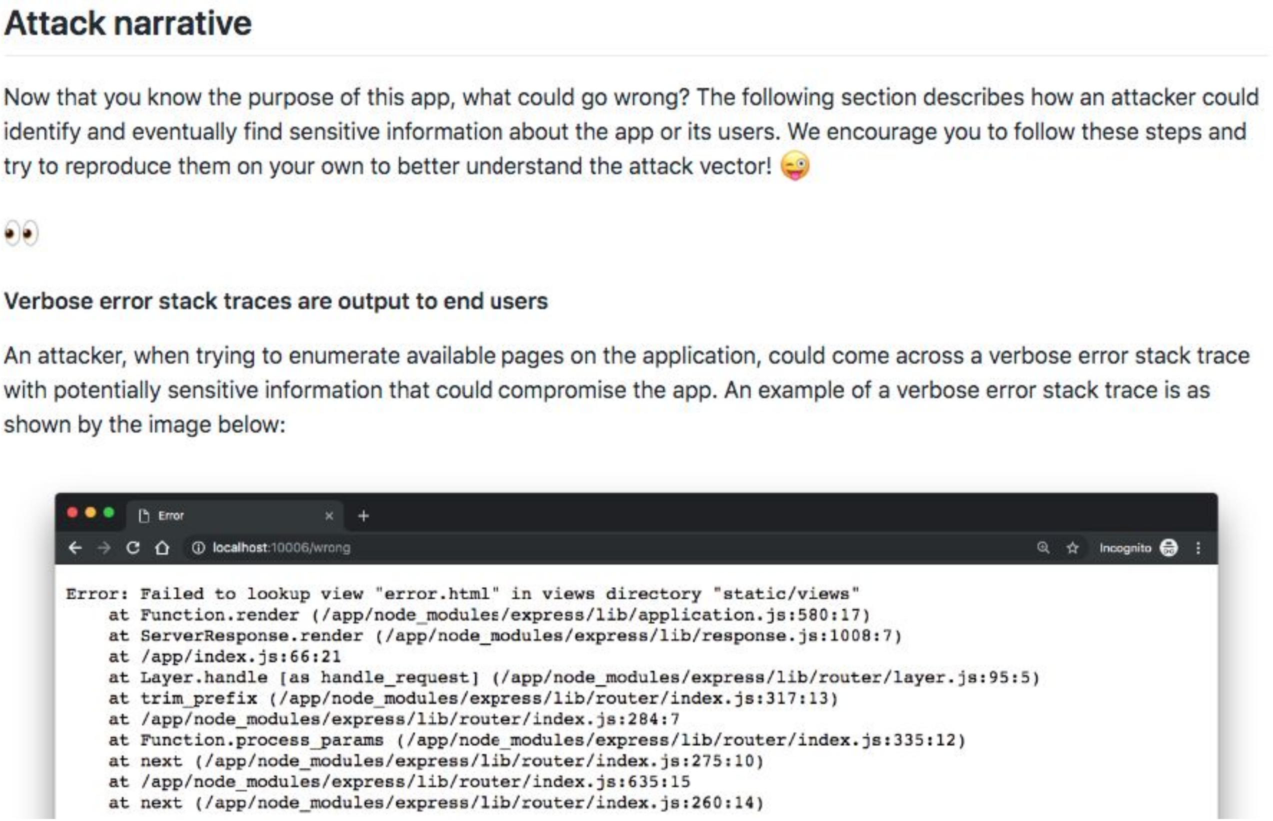
```yaml
version: '3.3'      Krlier, 6 months ago • [FE

services:
  app:
    container_name: Stegonography_api
    build:
      context: ../
      dockerfile: deployments/api.Dockerfile
    env_file:
      - .dockers.env
    ports:
      - 10006:10006
    networks:
      - a6_net
    restart: always

  db:
    image: mongo
    container_name: Stegonography_db
    env_file:
      - .dockers.env
    ports:
      - 27017:27017
      - 27018:27018
    networks:
      - a6_net

networks:
  a6_net:
```

# 3. Como funciona a dinâmica



## Attack narrative

Now that you know the purpose of this app, what could go wrong? The following section describes how an attacker could identify and eventually find sensitive information about the app or its users. We encourage you to follow these steps and try to reproduce them on your own to better understand the attack vector! 😜

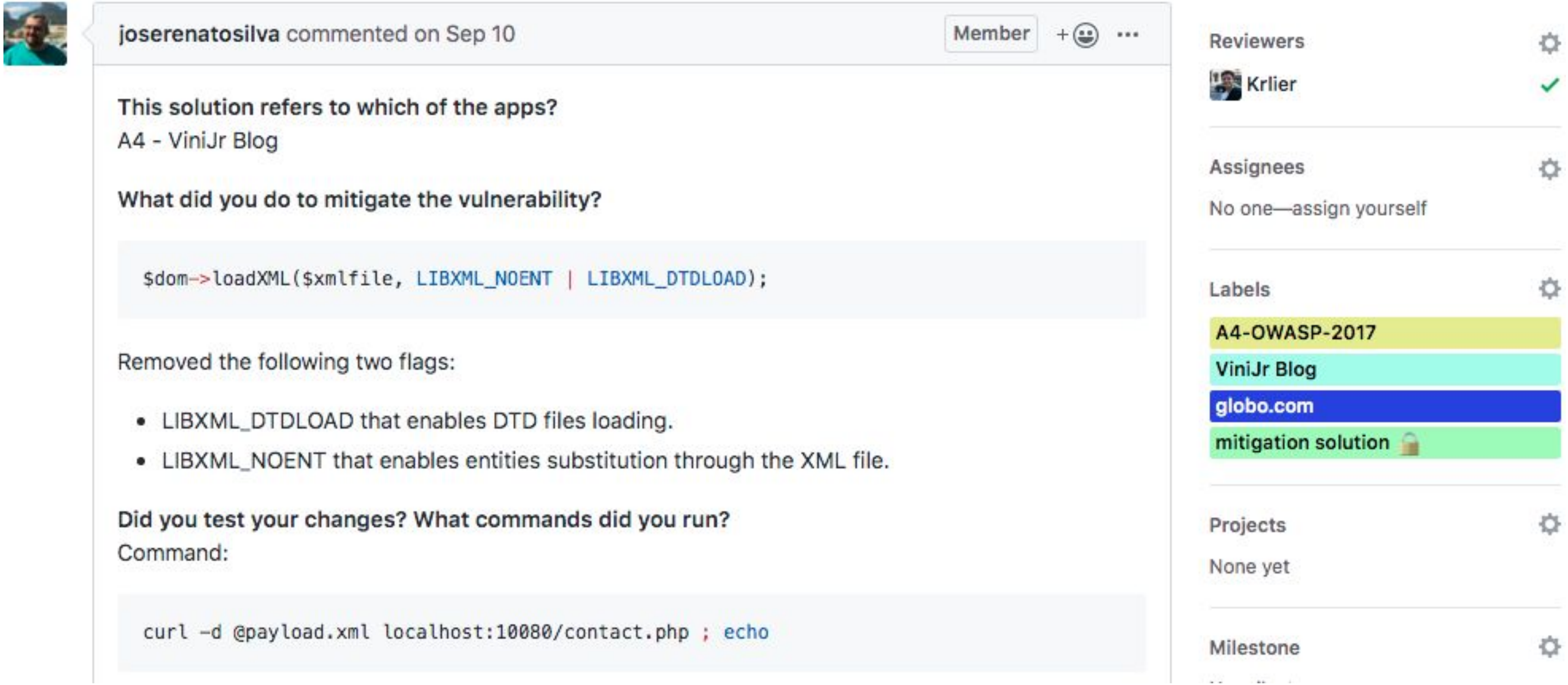
👀

### Verbose error stack traces are output to end users

An attacker, when trying to enumerate available pages on the application, could come across a verbose error stack trace with potentially sensitive information that could compromise the app. An example of a verbose error stack trace is as shown by the image below:

```
Error: Failed to lookup view "error.html" in views directory "static/views"
    at Function.render (/app/node_modules/express/lib/application.js:580:17)
    at ServerResponse.render (/app/node_modules/express/lib/response.js:1008:7)
    at /app/index.js:66:21
    at Layer.handle [as handle_request] (/app/node_modules/express/lib/router/layer.js:95:5)
    at trim_prefix (/app/node_modules/express/lib/router/index.js:317:13)
    at /app/node_modules/express/lib/router/index.js:284:7
    at Function.process_params (/app/node_modules/express/lib/router/index.js:335:12)
    at next (/app/node_modules/express/lib/router/index.js:275:10)
    at /app/node_modules/express/lib/router/index.js:635:15
    at next (/app/node_modules/express/lib/router/index.js:260:14)
```

# 3. Como funciona a dinâmica



joserenatosilva commented on Sep 10

**This solution refers to which of the apps?**
A4 - ViniJr Blog

**What did you do to mitigate the vulnerability?**

```
$dom->loadXML($xmlfile, LIBXML_NOENT | LIBXML_DTDLOAD);
```

Removed the following two flags:

- LIBXML_DTDLOAD that enables DTD files loading.
- LIBXML_NOENT that enables entities substitution through the XML file.

**Did you test your changes? What commands did you run?**
Command:

```
curl -d @payload.xml localhost:10080/contact.php ; echo
```

Reviewers

Krlier ✓

Assignees

No one—assign yourself

Labels

A4-OWASP-2017

ViniJr Blog

globo.com

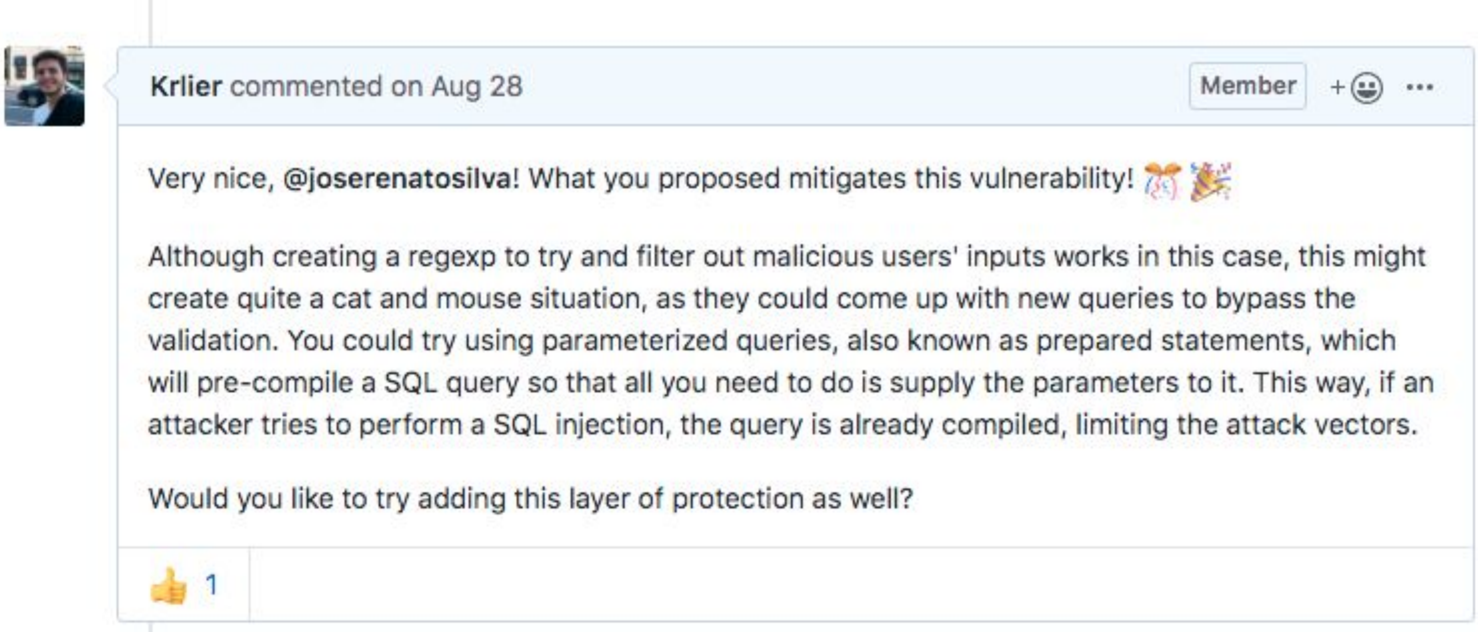
mitigation solution 🔒

Projects

None yet

Milestone

# 3. Como funciona a dinâmica



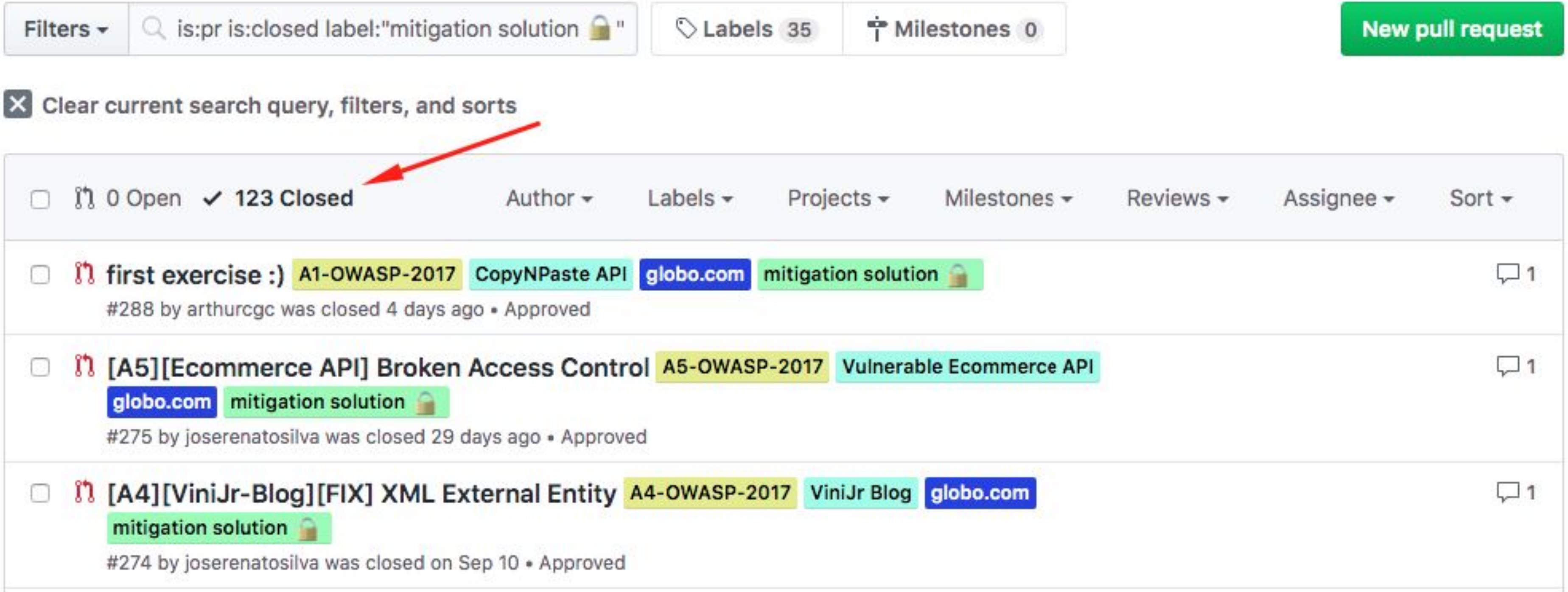
**Krlier** commented on Aug 28

Very nice, **@joserenatosilva**! What you proposed mitigates this vulnerability! 🎊 🎉

Although creating a regexp to try and filter out malicious users' inputs works in this case, this might create quite a cat and mouse situation, as they could come up with new queries to bypass the validation. You could try using parameterized queries, also known as prepared statements, which will pre-compile a SQL query so that all you need to do is supply the parameters to it. This way, if an attacker tries to perform a SQL injection, the query is already compiled, limiting the attack vectors.

Would you like to try adding this layer of protection as well?

# 4. Resultados dentro da Globo.com



□ first exercise :) A1-OWASP-2017 CopyNPaste API globo.com mitigation solution 🔒 — 1
#288 by arthurcgc was closed 4 days ago • Approved

□ [A5][Ecommerce API] Broken Access Control A5-OWASP-2017 Vulnerable Ecommerce API globo.com mitigation solution 🔒 — 1
#275 by joserenatosilva was closed 29 days ago • Approved

□ [A4][ViniJr-Blog][FIX] XML External Entity A4-OWASP-2017 ViniJr Blog globo.com mitigation solution 🔒 — 1
#274 by joserenatosilva was closed on Sep 10 • Approved

# 4. Resultados dentro da Globo.com

# 4. Resultados dentro da Globo.com

Hacktoberfest

**1 a 31 de outubro**
na Globo.com

Contribua e ganhe uma camiseta exclusiva.

https://opensource.globo.com

THE DEVELOPER'S CONFERENCE

secDevLabs

https://github.com/globocom/secDevLabs

**Daniel Carlier**
Analista de Segurança
Globo.com