Out, 2019

Treinando desenvolvedores a programarem de forma segura

Rafael dos Santos 🇧🇷 rafael.santos@corp.globo.com

$ whoami

Rafael dos Santos 🇧🇷 @rafasantos5
github.com/rafaveira3

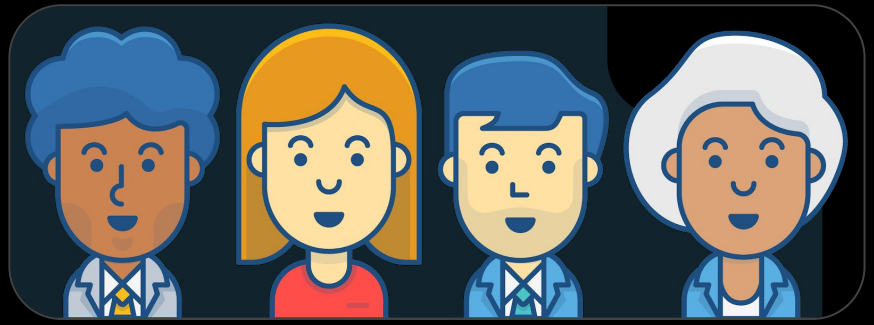
👨‍💻 Ciência da Computação - UFRJ

⚽ Apaixonado por futebol

☕ OSCP + OSCE

🐼 Analista de Segurança @ globo.com

🚀 Sec Tools + Desenvolvimento de Exploit



Um dia na vida de um time de desenvolvimento...

Um dia na vida de um time de desenvolvimento...

Paralamas ensaiam com música de Gil

- Paolla, Grazi e mais famosos

'A Dona do Pedaço': Vivi fica sem comida

- Cássia arma vingança

Nova Feat!

Paralamas ensaiam com música de Gil

- Paolla, Grazi e mais famosos

'A Dona do Pedaço': Vivi fica sem comida

- Cássia arma vingança

Um dia na vida de um time de desenvolvimento...

BACKLOG	SPRINT	DOING	REVIEW	DONE
Tarefa	Tarefa			
Tarefa	Tarefa			
Tarefa	Tarefa			
Tarefa	COR AZUL			

Um dia na vida de um time de desenvolvimento...

BACKLOG	SPRINT	DOING	REVIEW	DONE
Tarefa		Tarefa 		
Tarefa		Tarefa 		
Tarefa		Tarefa 		
Tarefa		COR AZUL 		

Um dia na vida de um time de desenvolvimento...

BACKLOG	SPRINT	DOING	REVIEW	DONE
Tarefa		Tarefa 	Tarefa 	
Tarefa			Tarefa 	
Tarefa				
Tarefa		COR AZUL 		

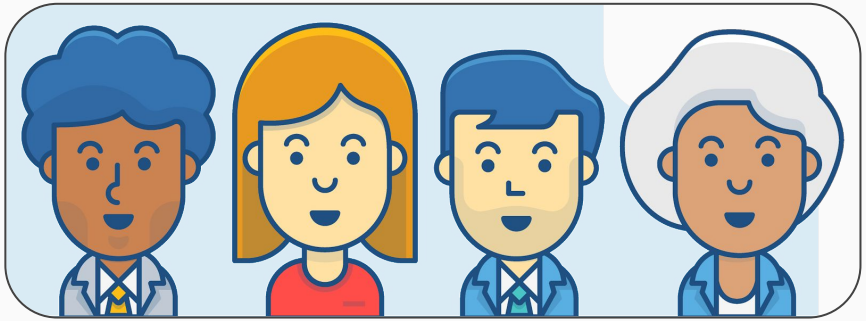
Um dia na vida de um time de desenvolvimento...

BACKLOG	SPRINT	DOING	REVIEW	DONE
Tarefa			COR AZUL 	Tarefa
Tarefa			Tarefa 	Tarefa
Tarefa				
Tarefa				

Um dia na vida de um time de desenvolvimento...

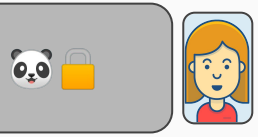
BACKLOG	SPRINT	DOING	REVIEW	DONE
Tarefa				Tarefa
Tarefa				Tarefa
Tarefa				COR AZUL
Tarefa				Tarefa

Um dia na vida de um time de desenvolvimento...



Vamos fazer um treinamento de segurança?

Treinamento de segurança por uma semana

BACKLOG	SPRINT	DOING	REVIEW	DONE
Tarefa				
Tarefa				
Tarefa				
Tarefa				

Treinamento de segurança por uma semana

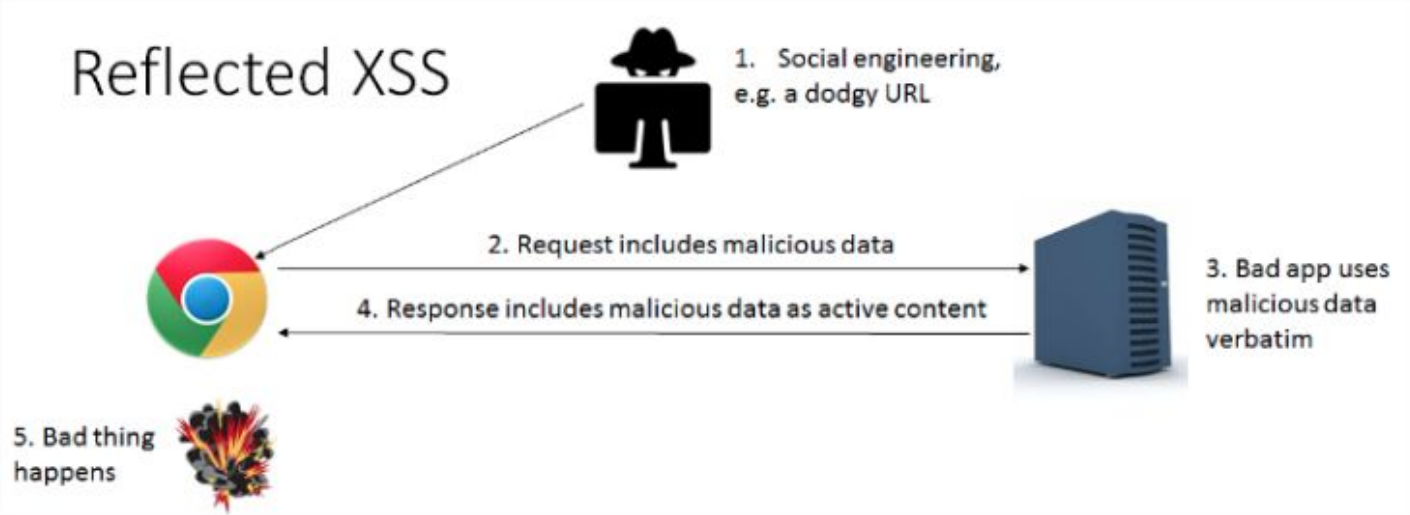
BACKLOG	SPRINT	DOING	REVIEW	DONE
Tarefa				
Tarefa				
Tarefa				
Tarefa				


THIS IS FINE

Treinamento de segurança por uma semana

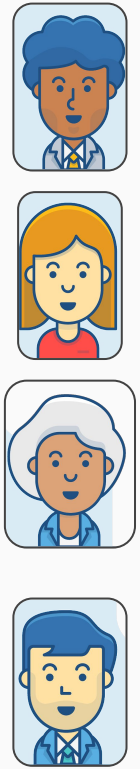
2 - XSS - Refletido

globo.com




```
http://www.example.com/search.asp?q=<script>alert(1);</script>
```

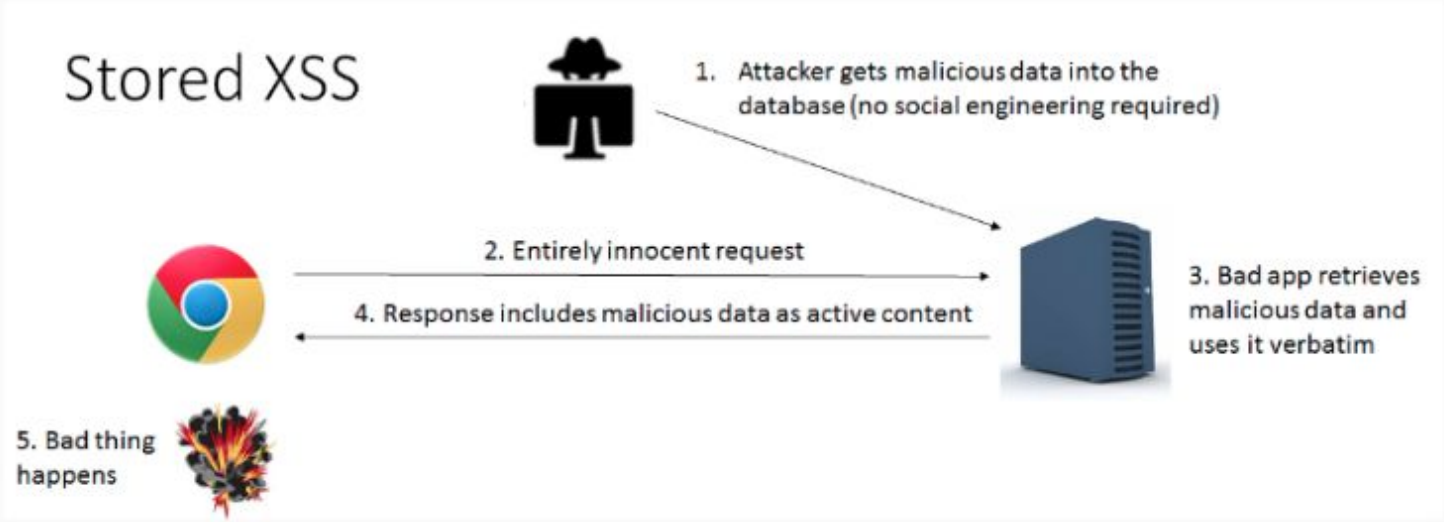
A7



Treinamento de segurança por uma semana

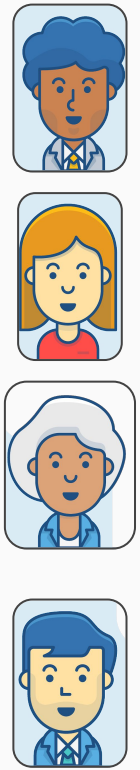
3 - XSS - Persistente

globo.com




```
http://www.example.com/profile.asp
```

A7

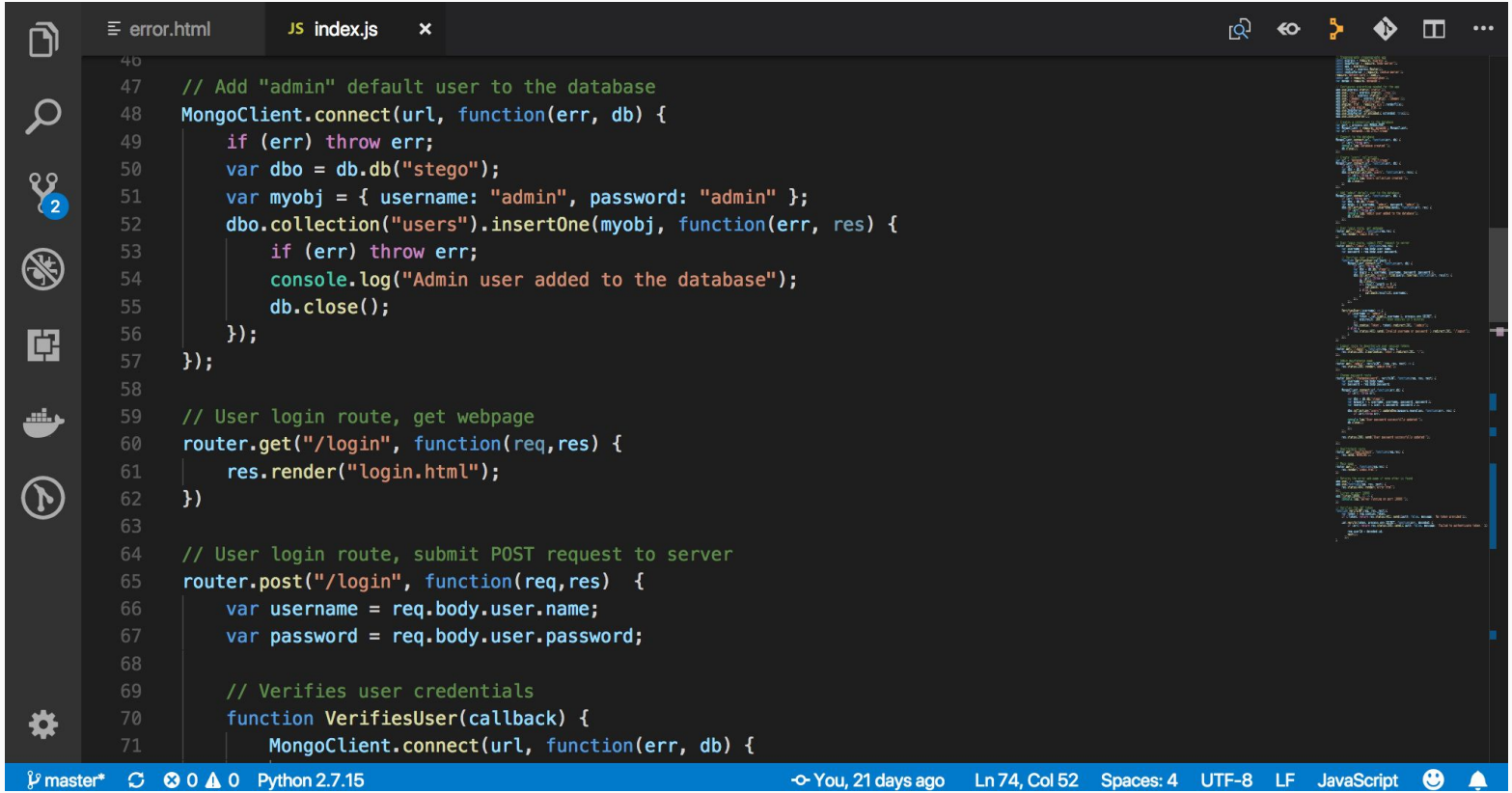


Treinamento de segurança por uma semana

Let's hack! 

Let's hack!

Planejamento: DEVs gostam de... Hands on!



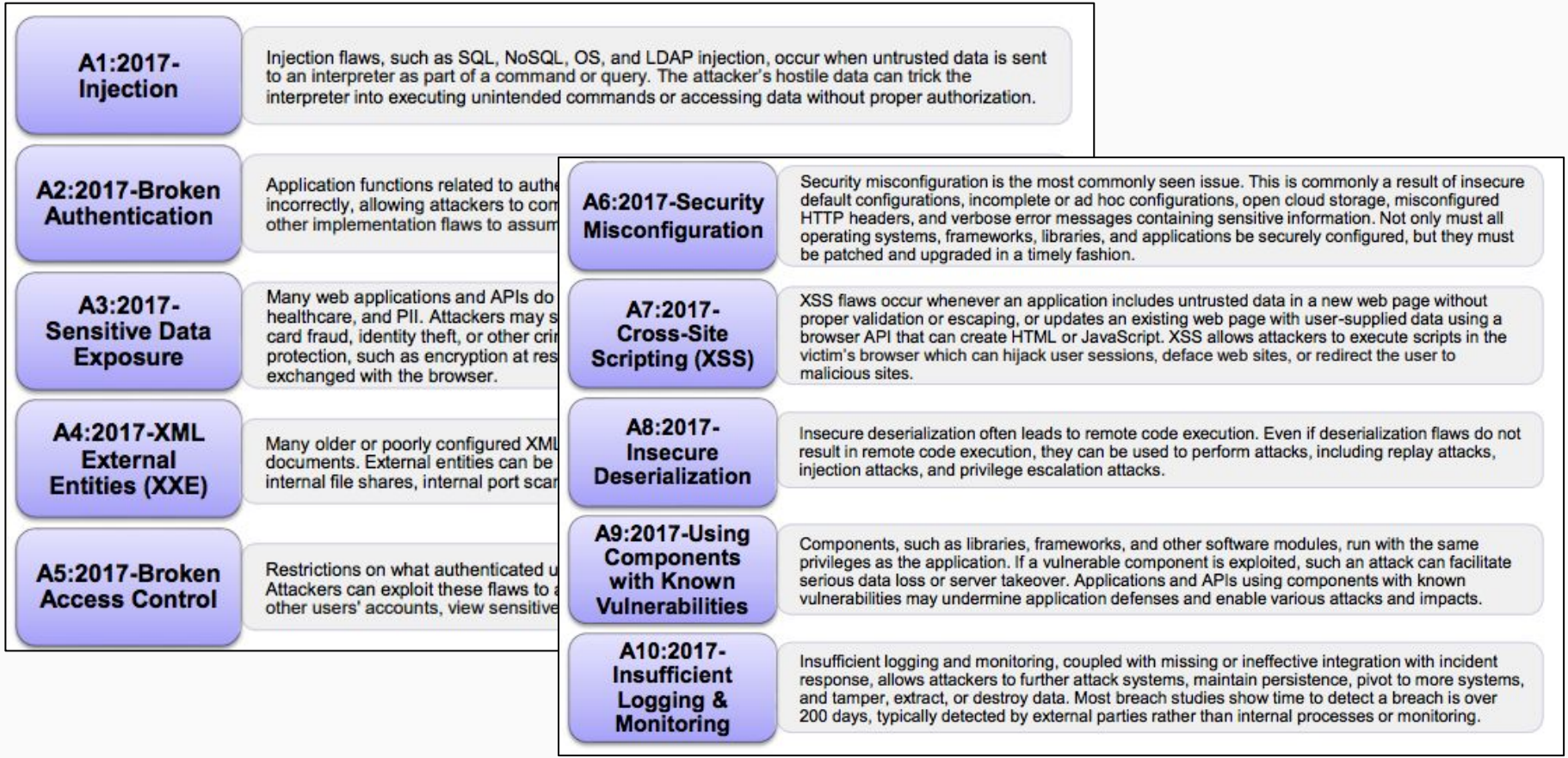
Planejamento: Atacar? Não! Desenvolver 🔒

SECCIM;

```
atabase: webpage
Table: users
[3 entries]
+----+----------+--------------------------------------------------------------+
| id | username | password                                                     |
+----+----------+--------------------------------------------------------------+
| 3  | Rafael   | $2a$14$Lt7QRVZfz9XeN2KWVyzLsuxOBKQ3uciwd440DS0hfRvVKUyEKWMtW |
| 4  | Bianca   | $2a$14$9LkJimyfVsD1WRgqCeX4eeAu0v4fHUNAHN6yj.jA.oLUJ32TZs1cS |
| 5  | C4rl0z   | $2a$14$ObEBiURggHjBDVUUMP6UK.Cnm.t9xTf.Mz9c5Iy4zOuDi7CKSipU6 |
+----+----------+--------------------------------------------------------------+
```

```go
query := fmt.Sprint("select * from Users where username = '" + user + "'")
rows, err := dbConn.Query(query)
stmt, stmtErr := dbConn.Prepare("SELECT * FROM Users WHERE username = ?")
if stmtErr != nil {
        return false, stmtErr
}
```

Planejamento: O que eles precisam aprender?



Planejamento: Não atrapalhar o sprint!

BACKLOG	SPRINT	DOING	REVIEW	DONE
Tarefa		SEC DEV LABS 		
Tarefa		Tarefa 		
Tarefa		Tarefa 		
Tarefa		Tarefa 		

Planejamento: Não atrapalhar o sprint!

BACKLOG	SPRINT	DOING	REVIEW	DONE
Tarefa		Tarefa 		
Tarefa		SEC DEV LABS 		
Tarefa		Tarefa 		
Tarefa		Tarefa 		

Planejamento: Não atrapalhar o sprint!

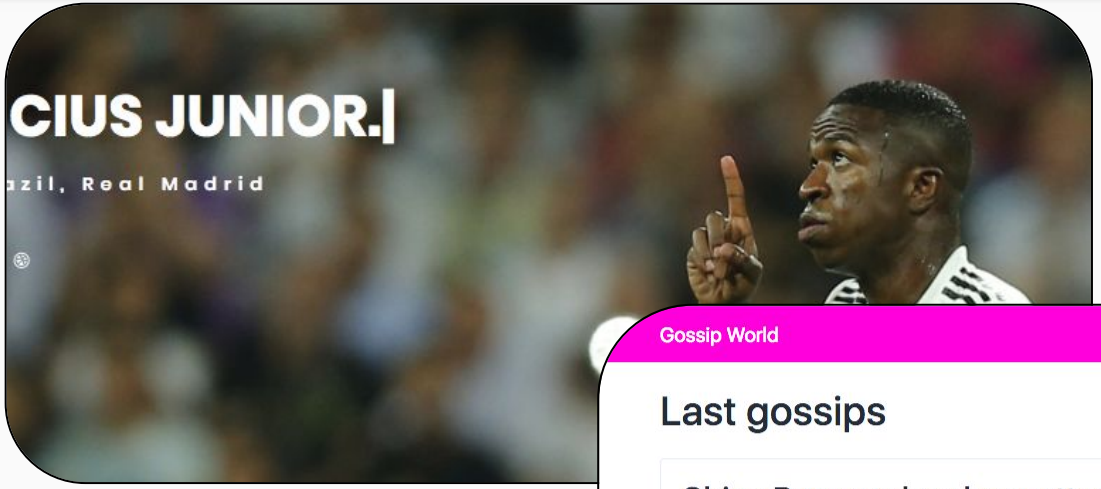
BACKLOG	SPRINT	DOING	REVIEW	DONE
Tarefa		Tarefa 		
Tarefa		Tarefa 		
Tarefa		SEC DEV LABS 		
Tarefa		Tarefa 		

Planejamento: Não atrapalhar o sprint!

BACKLOG	SPRINT	DOING	REVIEW	DONE
Tarefa		Tarefa 		
Tarefa		Tarefa 		
Tarefa		Tarefa 		
Tarefa		SEC DEV LABS 		

secDevLabs

secDevLabs: Apps vulneráveis!



Gossip World

Home New gossip Logout

Last gossips

Chico Buarque buy baguettes for snack

The singer and composer was dressed in shorts, T-shirts and slippers along the streets of Leblon, South Zone of Rio

Read more →

Posted on 2019-01-21 by vitoria

← Older Newer →

Search

<script>alert(1);</script> Go!

Copyright © Gossip World 2018

secDevLabs: Apps vulneráveis!

OWASP Top 10 (2017) apps:

Disclaimer: You are about to install vulnerable apps in your machine! 🔥

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
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	Gameslrados.com

secDevLabs: Laboratório local

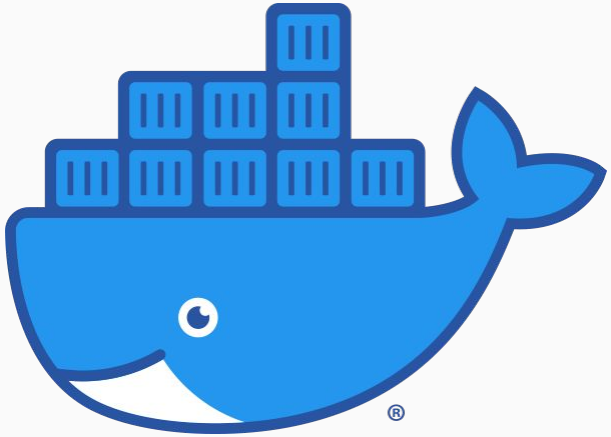
docker-compose.yml

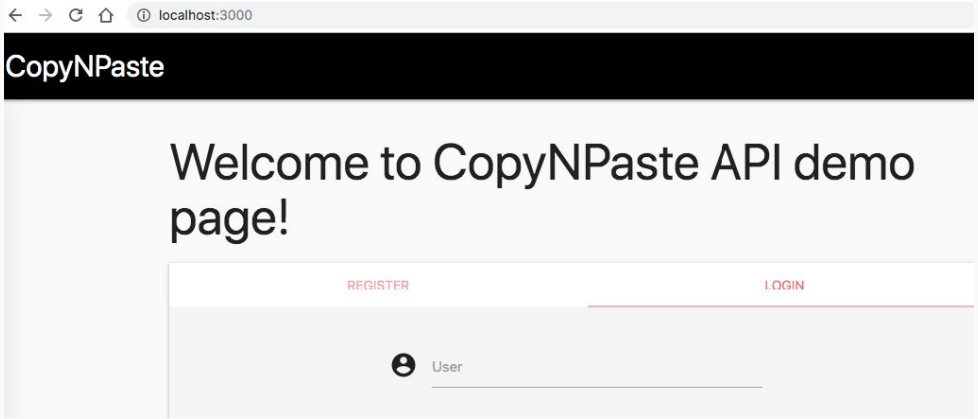
```yaml
version: '3'

services:

    api:
        container_name: a1_api
        build:
            context: ../
            dockerfile: deployments/a1inj.Dockerfile
        ports:
            - "3000:3000"
        networks:
            - a1net
        command: "go run app/server.go"
        environment:
            MYSQL_ROOT_PASSWORD: root
            MYSQL_USER: user
            MYSQL_PASSWORD: pass
            MYSQL_DATABASE: a1db
        depends_on:
            - mysqldb
        external_links:
            - mysqldb:mysqldb
        restart: unless-stopped

    mysqldb:
        container_name: mysqldb
        image: mysql:5.7
        ports:
          - "3307:3307"
        volumes:
            - db_data:/var/lib/mysql
        environment:
            MYSQL_ROOT_PASSWORD: root
```





secDevLabs: Narrativa de ataque

As no validation is being used to avoid ENTITIES being sent to the PHP file, an attacker could create the following `evilxml.xml` to perform a XXE:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE root [
<!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
<contact>
<name>&xxe;</name>
<email>RAFAEL@EXAMPLE.com</email>
<subject>YOU ROCK</subject>
<message>I LOVE WATCHING YOUR SKILLS, MAN</message>
</contact>
```

And, as the following picture shows, it is possible to realize that the attack succeeds and sensitive information is retrieved from the server that is hosting the vulnerable app:

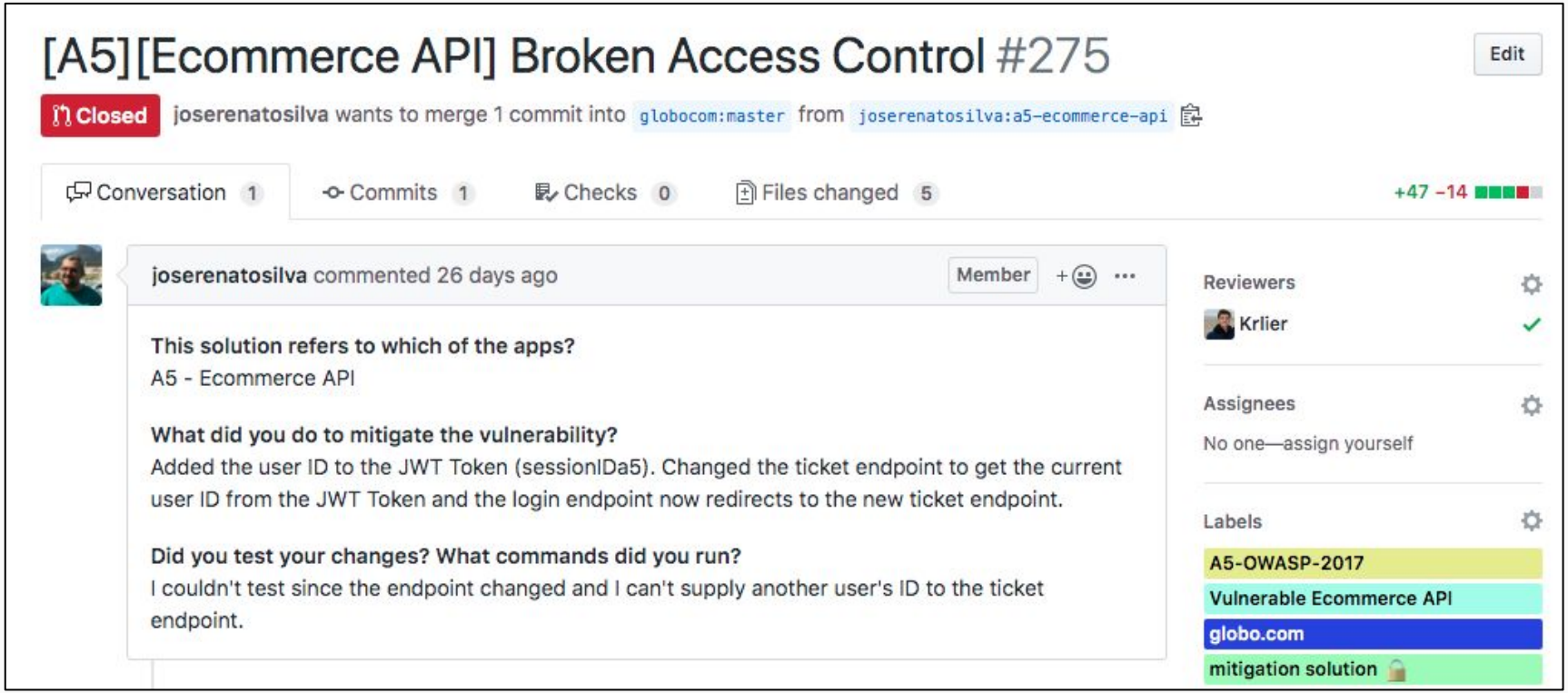
```
curl -d @evilxml.xml localhost:10080/contact.php ; echo
```

```
✔ [14:06] rafael.santos@labs:~
$ curl -d @evilxml.txt localhost:10080/contact.php ; echo
Thanks for the message, root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
```

secDevLabs: Mitigação via Pull Requests

```
                +         comment = escape(request.form.get('comment'))
                +         user = escape(session.get('username'))
168             168       date = datetime.datetime.now()
169             169       if comment == '':
170             170           flash('All fields are required', 'danger')
                ⇵ ⇵
                          @@ -198,10 +200,10 @@ def gossip(id):
198             200     @login_required
199             201     def newgossip():
200             202         if request.method == 'POST':
201             -           text = request.form.get('text')
202             -           subtitle = request.form.get('subtitle')
203             -           title = request.form.get('title')
204             -           author = session.get('username')
                203     +         text = escape(request.form.get('text'))
                204     +         subtitle = escape(request.form.get('subtitle'))
                205     +         title = escape(request.form.get('title'))
                206     +         author = escape(session.get('username'))
205             207       date = datetime.datetime.now()
206             208       if author is None or text is None or subtitle is None or title is None:
207             209           error('gossip', 'Invalid parameters', session.get('username'))
```

secDevLabs: Mitigação via Pull Requests



secDevLabs: Time de segurança aprende junto!

Krlier commented on Aug 28 — Member

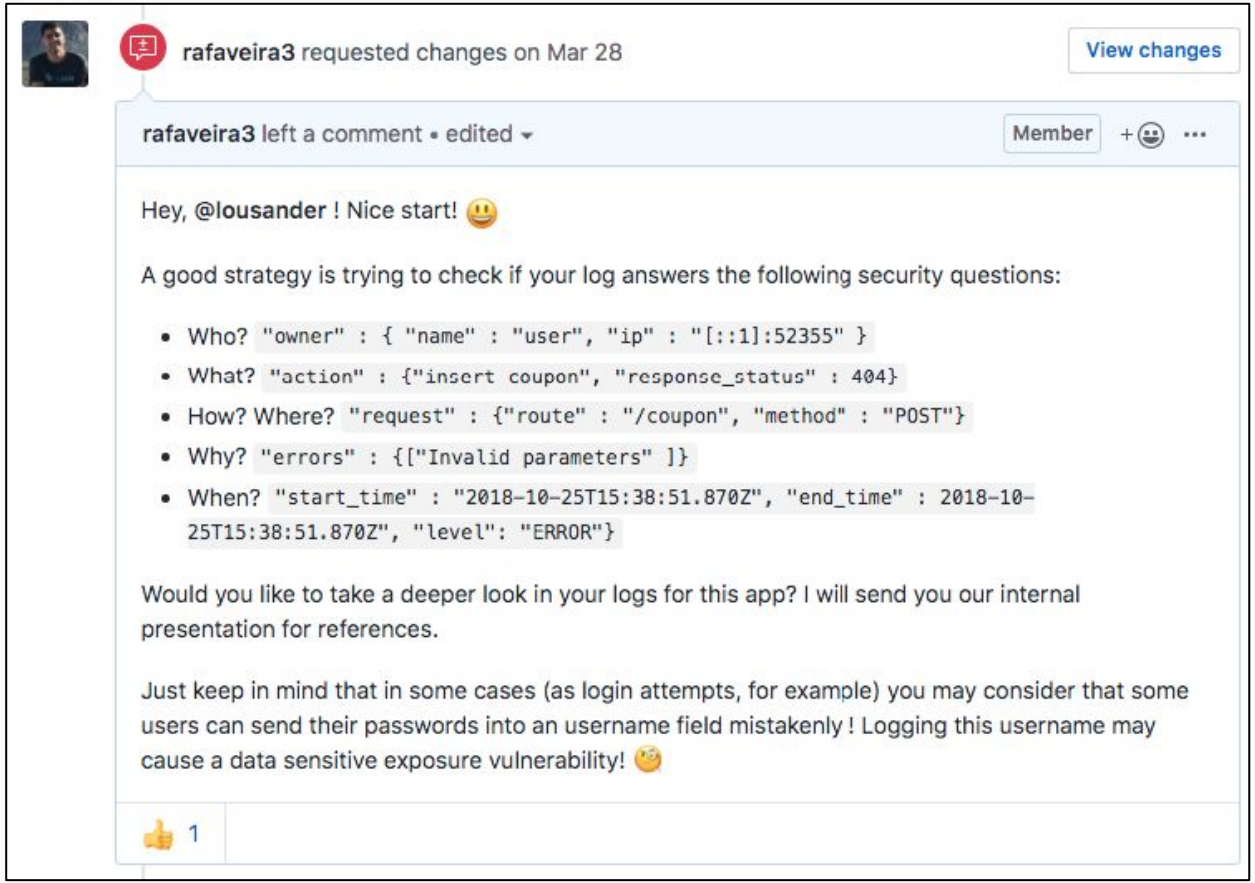
Very nice, **@joserenatosilva**! What you proposed mitigates this vulnerability! 🎊🎉

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?

👍 1

secDevLabs: Time de segurança aprende junto!



rafaveira3 requested changes on Mar 28 — View changes

rafaveira3 left a comment • edited — Member

Hey, **@lousander** ! Nice start! 😃

A good strategy is trying to check if your log answers the following security questions:

- Who? `"owner" : { "name" : "user", "ip" : "[::1]:52355" }`
- What? `"action" : {"insert coupon", "response_status" : 404}`
- How? Where? `"request" : {"route" : "/coupon", "method" : "POST"}`
- Why? `"errors" : {["Invalid parameters" ]}`
- When? `"start_time" : "2018-10-25T15:38:51.870Z", "end_time" : 2018-10-25T15:38:51.870Z", "level": "ERROR"}`

Would you like to take a deeper look in your logs for this app? I will send you our internal presentation for references.

Just keep in mind that in some cases (as login attempts, for example) you may consider that some users can send their passwords into an username field mistakenly ! Logging this username may cause a data sensitive exposure vulnerability! 😉

👍 1

secDevLabs: Time de segurança aprende junto!

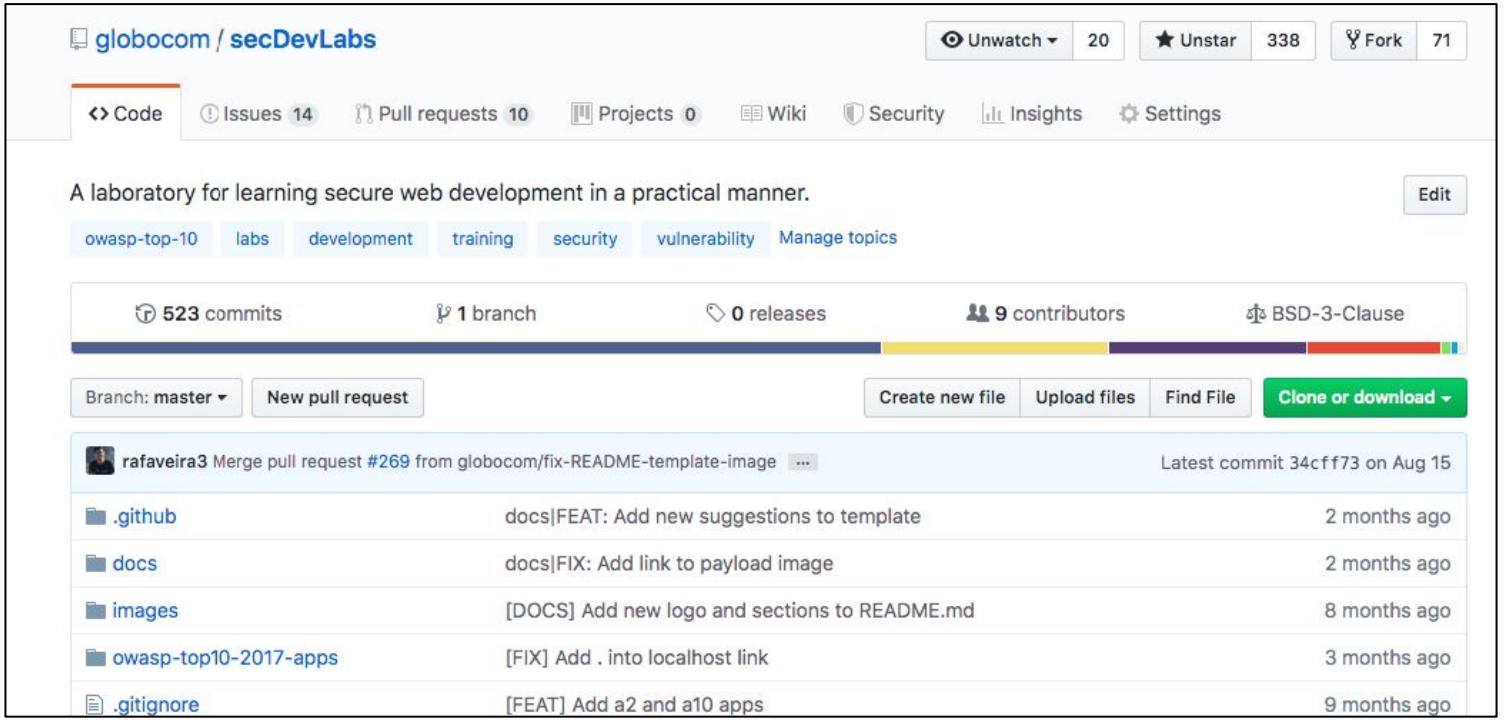
vitoriario commented on Feb 5 • edited ▾ Contributor +😀 ⋯

Hi **@pedrokiefer**, your solution works well! But I noticed that you are using bleach in password fields too. This is not a good idea because you are changing the user password, for example:

If I have the following password: **<script>**
bleach changes it to **<script&gt**

Remember that passwords should not be shown to users so, in this case, we don't have to escape special characters.

secDevLabs: Projeto Open Source

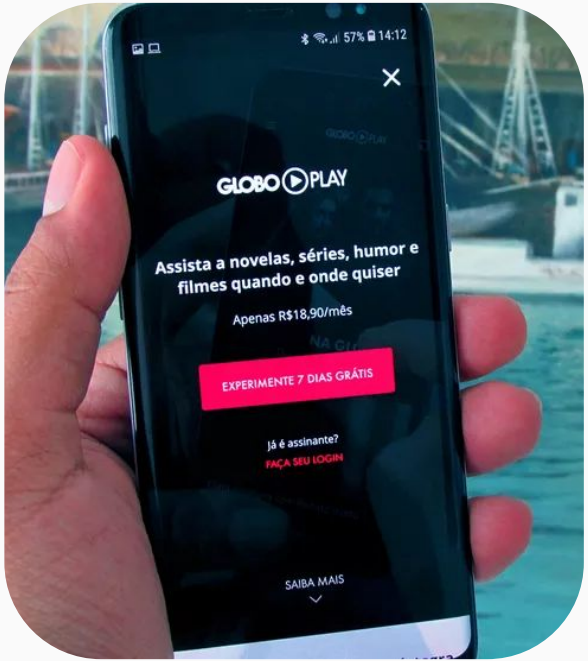


https://github.com/globocom/secDevLabs

Demo 🔥

Próximos passos ⭐

Próximos passos: Apps em Mobile



Próximos passos: Contemplar mais linguagens








Extra ☕

Extra: Premiações secDevLabs dentro da Gcom!

Extra: Hacktoberfest

https://opensource.globo.com

Extra: Trabalhe conosco!

https://talentos.globo.com

Out, 2019

https://github.com/globocom/secDevLabs

Perguntas?

🤔

Rafael dos Santos 🇧🇷 rafael.santos@corp.globo.com

Out, 2019

https://github.com/globocom/secDevLabs

Obrigado!

🙃

Rafael dos Santos 🇧🇷 rafael.santos@corp.globo.com