

## Tag de Web Hacking

Candidato: Eduardo Archanjo Cavalcante

Professor: Breno

1) O que é o protocolo HTTP ? E como ele funciona ?

R: É um protocolo de comunicação (na camada de aplicação segundo o Modelo OSI) utilizado para sistemas de informação de hipermídia, distribuídos e colaborativos.

O protocolo HTTP é baseado em requisições e respostas entre clientes e servidores. O cliente — navegador ou dispositivo que fará a requisição; também é conhecido como *user agent* — solicita um determinado recurso (resource), enviando um pacote de informações contendo alguns cabeçalhos (headers) a um URI ou, mais especificamente, URL. O servidor recebe estas informações e envia uma resposta, que pode ser um recurso ou um simplesmente um outro cabeçalho.

2) O que é Response Code ? Cite um exemplo de um programa que você pode fazer com ele ?

R: Toda requisição recebe um código de resposta. Com esse código é possível saber se uma operação foi realizada com sucesso (2\*\*), se ele foi movida e agora existe em outro lugar (3\*\*) ou se não existe mais (4\*\*).

Pode-se criar um programa para poder tirar informações do servidor com base nesses códigos de resposta.

3) O que é um HEADER ? Cite um uso INSEGURO desse cabeçalho.

R: É o cabeçalho da mensagem é utilizado para transmitir informações adicionais entre o cliente e o servidor. Ele é especificado imediatamente após o método, tanto para a requisição do cliente quanto para a resposta do servidor.

- 4) O que é um método HTTP ? Explique o funcionamento do método POST, o funcionamento do método GET. Explique qual é considerado mais seguro e por que.

R:

Quando você vai fazer uma requisição, é preciso que você especifique qual o método será utilizado. Os métodos determinam qual a ação que deve ser executada em um determinado recurso e URL.

Método GET: Método que requisita a implementação do método especificado, onde os dados são enviados na URL o que torna este o método mais inseguro, pois os dados podem ser vistos pelos usuários

Método POST: Envia as informações no corpo da requisição, o que o torna o método mais seguro pois não pode ser guardado no histórico, como também não é visível diretamente pelo usuário.

- 5) O que é o Cache e como ele funciona? Cite os principais HEADERS de Request e Response responsáveis pelo controle de Cache.

R: É um armazenamento temporário no disco de páginas acessadas na web com o intuito de reduzir a largura de banda e aumentar a velocidade de acesso.

HEADERS responsáveis pelo controle de cache são: Cache-Control, Pragma, Expires e Validators.

- 6) O que é um Cookie? Qual é o principal ataque relacionado a ele?

R: São arquivos em textos armazenados pelo navegador no lado do cliente que guarda uma série de dados de navegação do usuário.

O principal ataque relacionado é o Session Hijacking onde um hacker consegue total acesso a conta online do usuário roubando o token de sessão que garante a entrada do usuário ao site.

7) O que é OWASP-TOP-Ten?

R: É o ranking dos ataques mais comuns praticados na rede, que serve como consenso e guia para desenvolvedores web tornarem seus sites mais seguros.

8) O que é Recon e por que ela é importante?

R: Consiste no processo de aquisição de informações, técnicas ou não, do seu alvo a fim de executar um ataque preciso e correto.

9) Command Injection (SO-Injection)

a) O que é Command Injection?

R: É a execução de códigos na máquina alvo através de uma vulnerabilidade conhecida, injetando através de parâmetros não tratados.

b) Mostre um exemplo de Command Injection (PoC da Exploração)

**DVWA**

Home  
Instructions  
Setup / Reset DB  
Brute Force  
**Command Injection**  
CSRF  
File Inclusion  
File Upload  
Insecure CAPTCHA  
SQL Injection  
SQL Injection (Blind)  
Weak Session IDs  
XSS (DOM)  
XSS (Reflected)  
XSS (Stored)  
CSP Bypass  
JavaScript  
DVWA Security

### Vulnerability: Command Injection

**Ping a device**

Enter an IP address:

```
/var/www/html/vulnerabilities/exec
..
help
index.php
source
```

**More Information**

- <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- [https://www.owasp.org/index.php/Command\\_Injection](https://www.owasp.org/index.php/Command_Injection)

Vemos a injeção de comando nos mostrando o diretório e seus arquivos

## 10) SQL INJECTION

a) O que é SQL Injection?

R: É a injeção e execução de código SQL através de campos de formulário que trabalham com queries em sql. O código quebra a query e tenta adquirir o acesso a página seguinte.

b) O que é Union Based Attack?

R: Utilizando o operador Union o atacante consegue retirar grande informação do banco de dados

c) O que é Blind-SQL-I?

R: Quando o resultado da injeção de código, ao invés de dar acesso ou passar por um formulário, retorna um código de status http que a partir disso retira-se informação acerca do banco de dados e dos resultados das queries

d) Mostre um exemplo de um Blind-SQL-Injection

**DVWA**

**Vulnerability: SQL Injection (Blind)**

User ID:

User ID exists in the database.

**More Information**

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- [https://www.owasp.org/index.php/Blind\\_SQL\\_Injection](https://www.owasp.org/index.php/Blind_SQL_Injection)
- <http://bobby-tables.com/>

Vemos que, mesmo não conseguindo acesso, vemos conseguimos saber que este db tem falhas de sqli.

## 11) XSS

a) O que é XSS?

R: É a injeção de código javascript nos campos de textos de uma página que será executado.

b) Quais são os tipos de XSS? Explique-os:

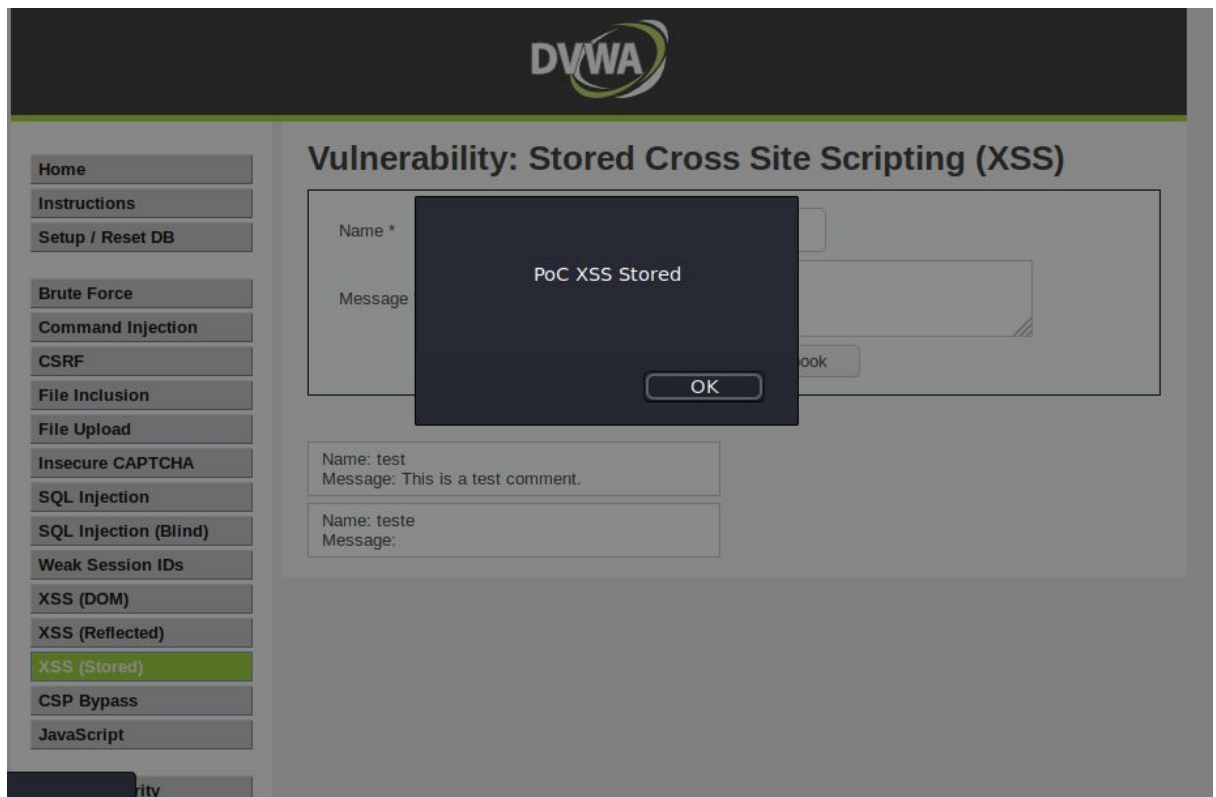
R: Há três tipos Reflected, Stored e Dom.

Reflected- O mais comum dentre os três, onde o script injetado vem da requisição http do servidor, acionado pelo usuário.

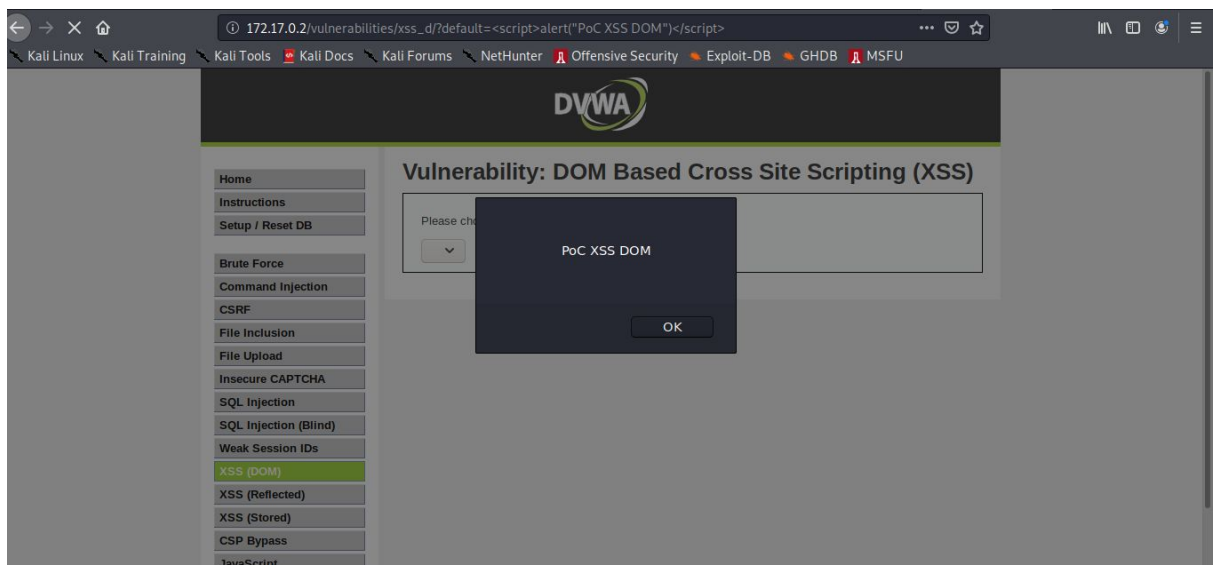
Stored- Usar uma página acessada para rodar o script malicioso

Dom- Todo o código malicioso é executado no lado do cliente

c) Mostre um exemplo de XSS Stored



d) Mostre um exemplo de um DOM-XSS



12) LFI, RFI e Path Traversal

a) O que é LFI?

R: A falha de LFI permite que o atacante inclua um arquivo para explorar o mecanismo de inclusão dinâmica de arquivo implementado na aplicação web. A falha ocorre devido ao fato

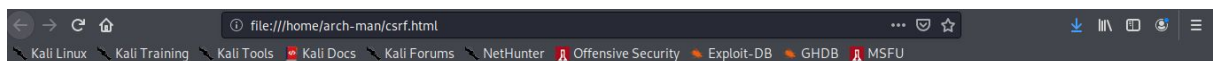
a) o que é CSRF?

R: É um tipo de exploit malicioso de um website, no qual comandos não autorizados são injetados a partir de um usuário em quem a aplicação web confia.

b) Mostre um exemplo de CSRF

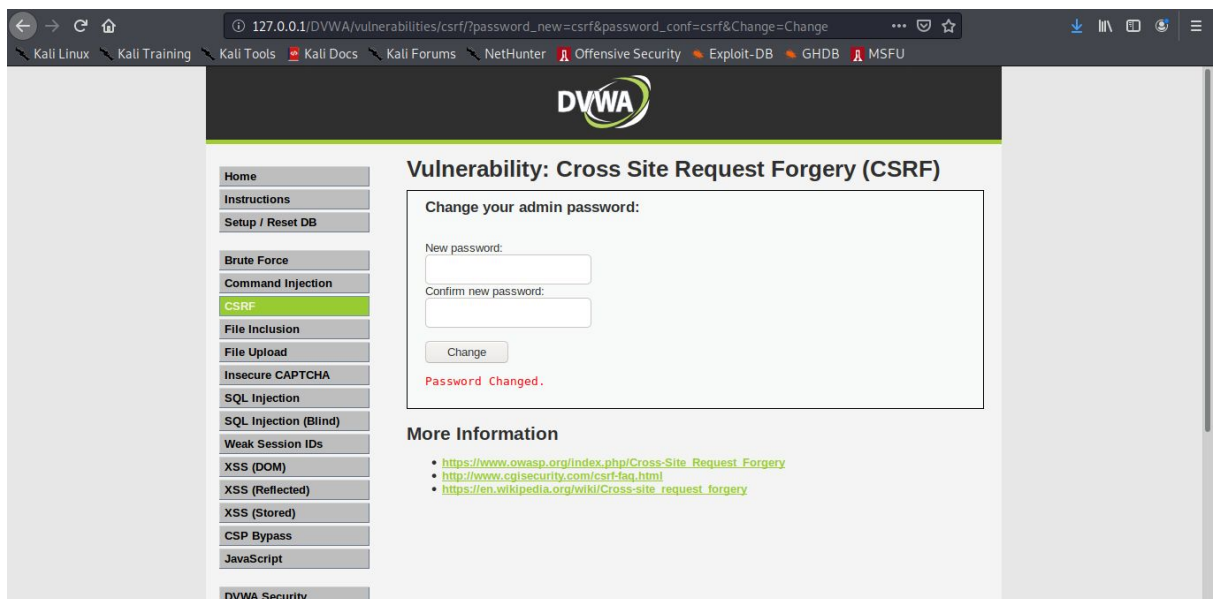
```
<form action="http://127.0.0.1/DVWA/vulnerabilities/csrf/?" method="GET">
  <h1>CLIQUE PARA VER UM CSRF!</h1>
  <input type="hidden" AUTOCOMPLETE="off" name="password_new" value="csrf">
  <input type="hidden" AUTOCOMPLETE="off" name="password_conf" value="csrf">
  <input type="submit" value="Change" name="Change">
</form>
```

Pegamos o trecho de código onde o site recebe a nova senha e introduzimos uma nova senha sem qualquer acesso ao site



## CLIQUE PARA VER UM CSRF!

Change



c) O que é SSRF?

R: É um tipo de exploração em que um invasor faz com que um servidor acesse ou manipule informações no domínio dele próprio que, de outra forma, não seriam diretamente acessíveis ao invasor.



d) Mostre um exemplo de SSRF

R: -----Não consegui-----

e) Como evitar ataques de CSRF?

R: Evitar o uso do "lembrar-me"; usar cookies de páginas para evitar session hijacking; usar extensões que impeçam execução de códigos