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?

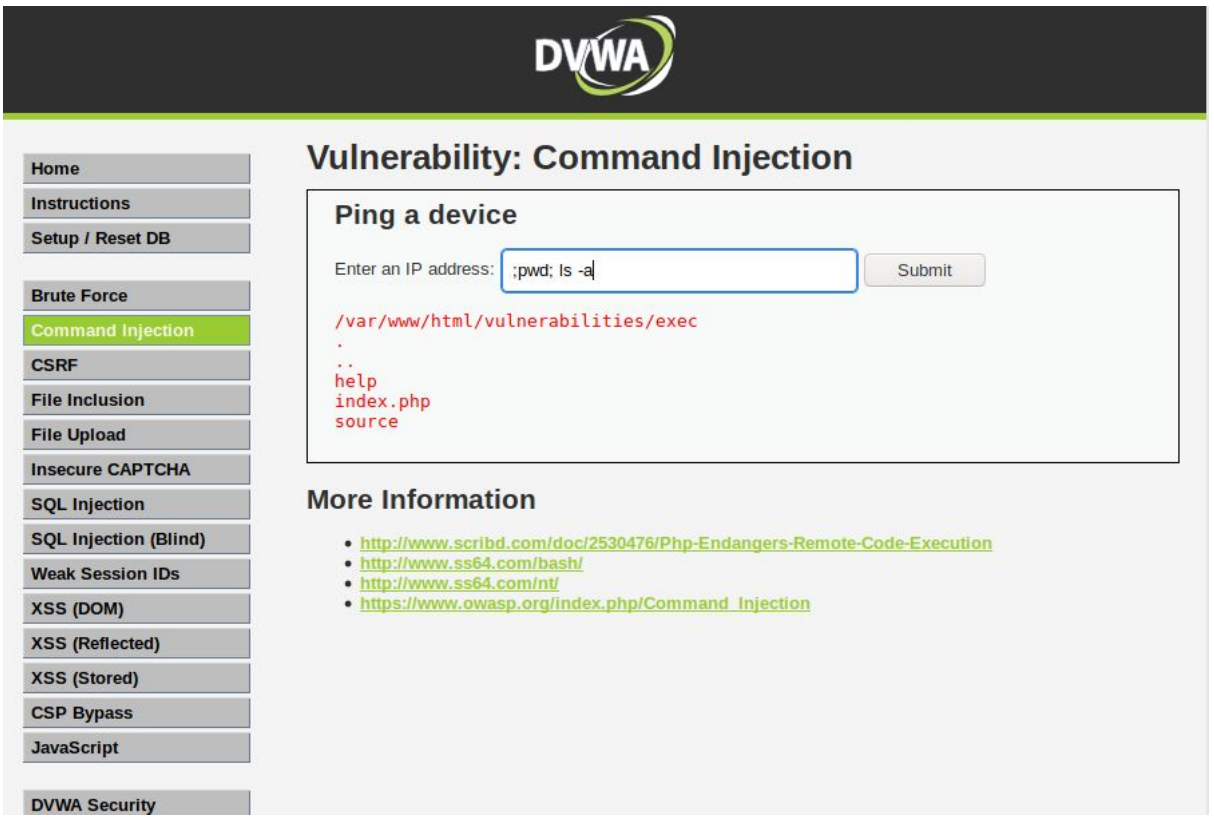
R: Consiste no processo de aquisição de informações, técnicas ou não, do seu alvo a fim de executar um ataque preciso e correto.

9) Command Injection (SO-Injection)

a) O que é Command Injection?

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

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



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 querys em sql. O código quebra a query e tenta adquirir o acesso a página seguinte.

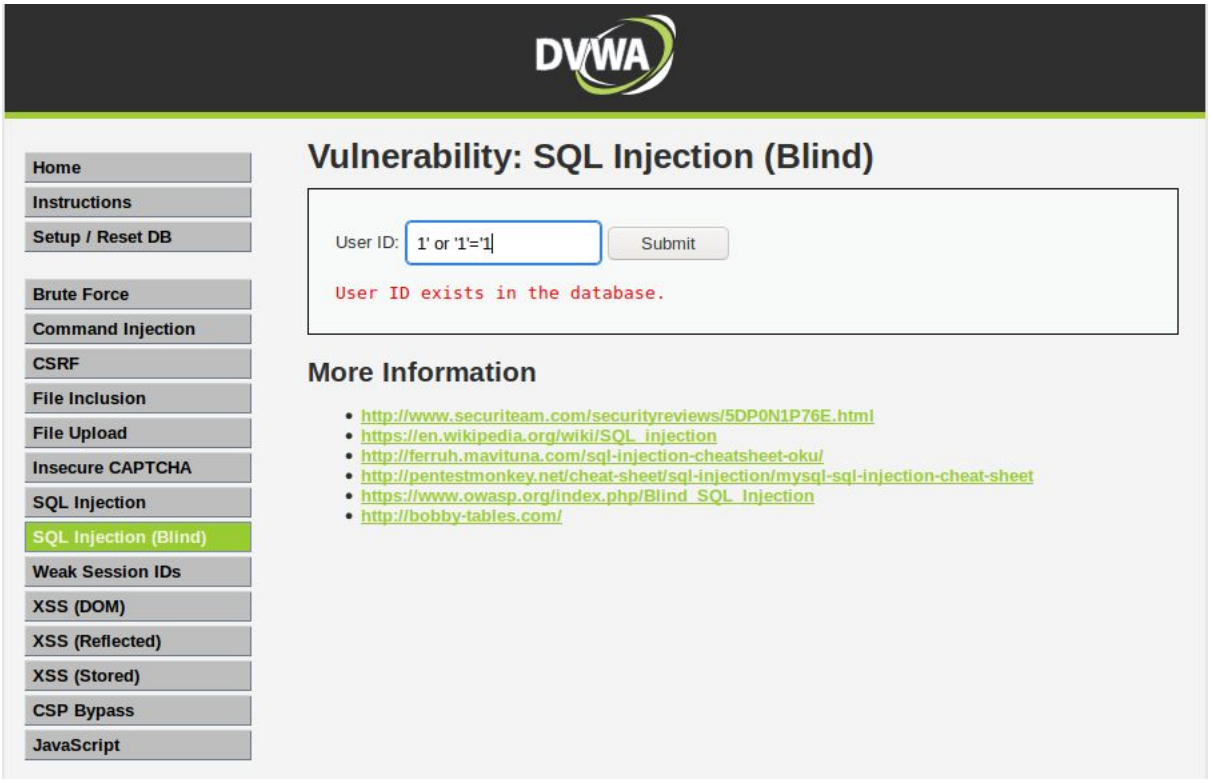
b) O que é Union Based Attack?

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 querys

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



Vemos que, mesmo nao 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:

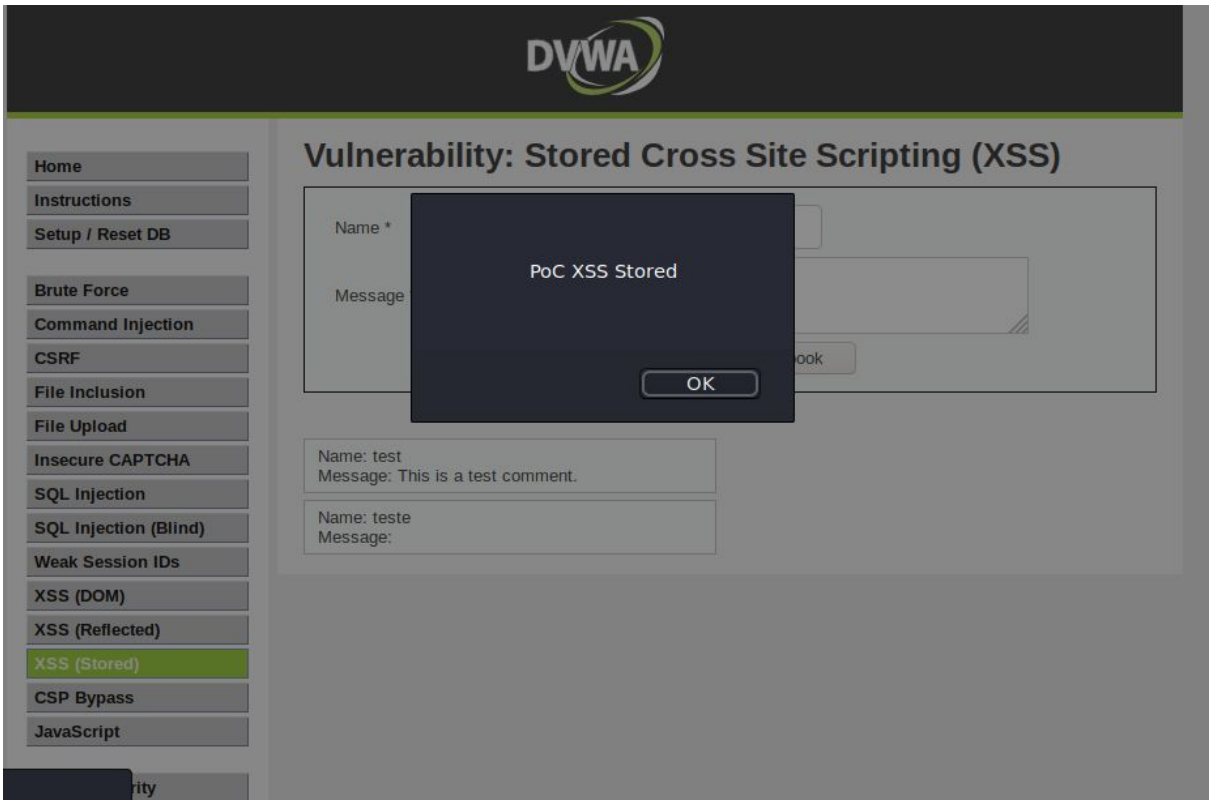
R: Há três tipos Reflected, Stored e Dom.

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

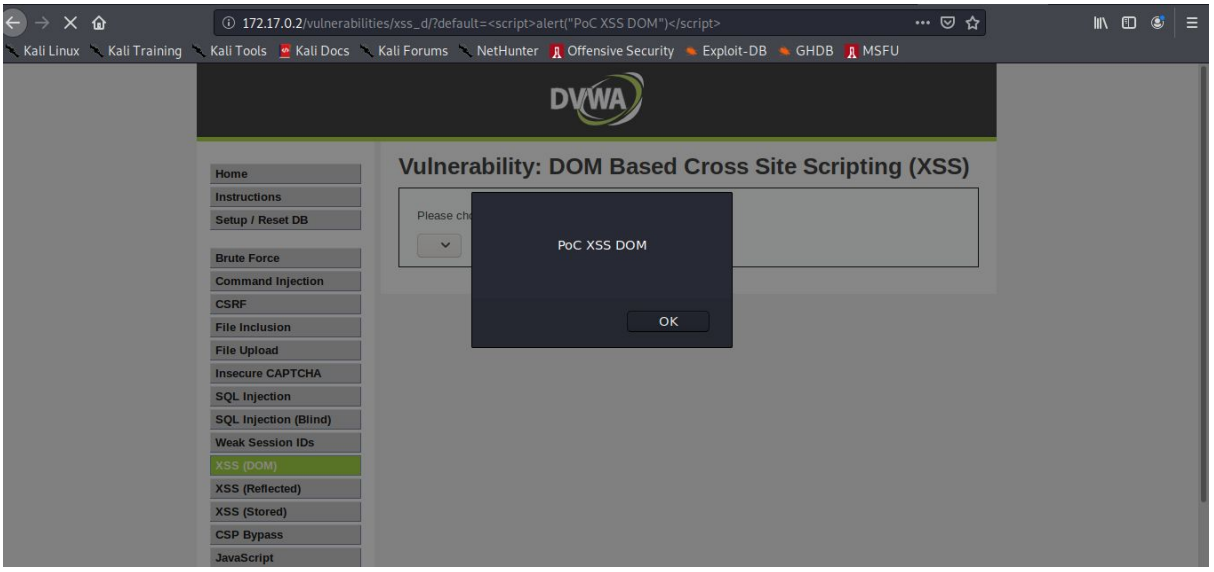
Stored- Usar uma página acessada para rodar o script malicioso

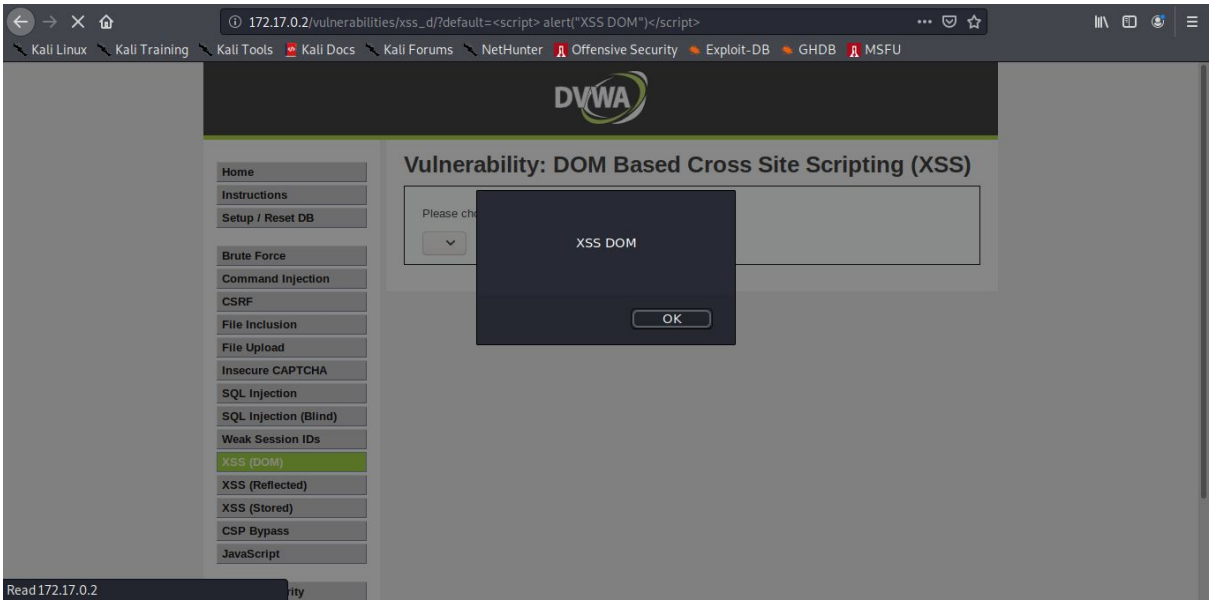
Dom- Todo o código malicioso é executado no lado do cliente

c) Mostre um exemplo de XSS Stored



d) Mostre um exemplo de um DOM-XSS





Vemos que injetamos um script através da url do site, pois o mesmo não trata as url que o usuário envia ao servidor. Assim conseguimos rodar este script

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 de que o atacante pode passar qualquer valor para o parâmetro

da aplicação alvo e a mesma não faz a validação correta do valor informado antes de executar a operação

b) O que é RFI?

R: Semelhante ao LFI, porém Esta falha ocorre, por exemplo, quando uma página recebe como entrada, o caminho para o arquivo que será incluído, e esta entrada não é validada de forma correta pela aplicação web, permitindo assim que uma URL externa seja injetada na aplicação.

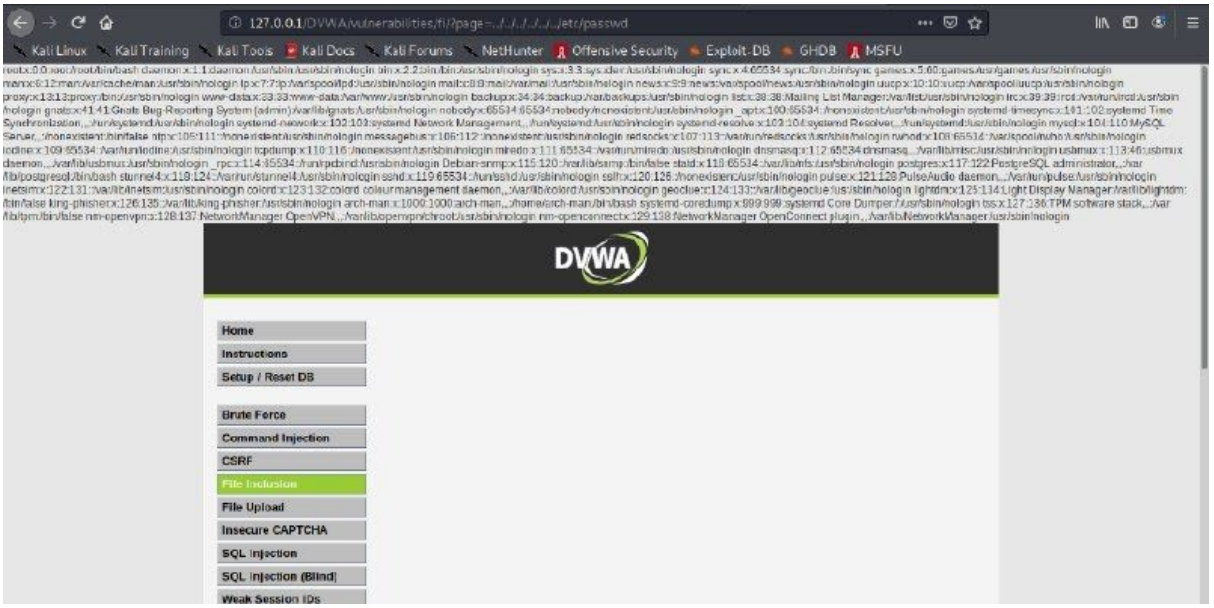
c) O que é Path Traversal?

R: É um ataque utilizado por atacantes para obter acesso não autorizado a arquivos e diretórios

d) Como aliar Path Traversal e LFI?

R: Coma falha de LFI é possível uma página recebe como entrada, o caminho para o arquivo que será incluído, e esta entrada não é validada de forma correta pela aplicação web, possibilitando assim que caracteres de directory traversal(../../) sejam injetados, possibilitando a escalação

e) Mostre um exemplo de LFI utilizando a contaminação de LOGS



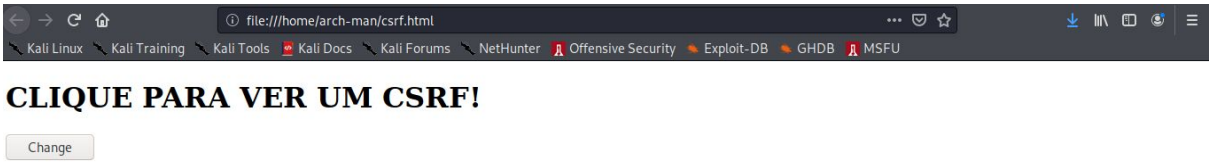
13) CSRF e SSRF

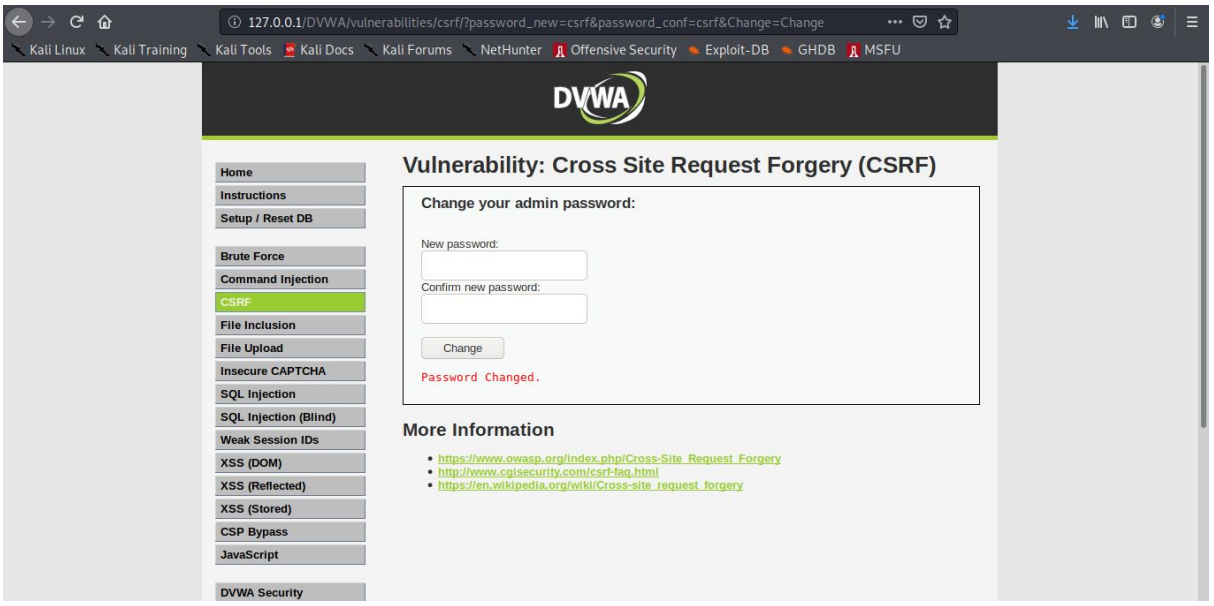
a) o que é CSRF?

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

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



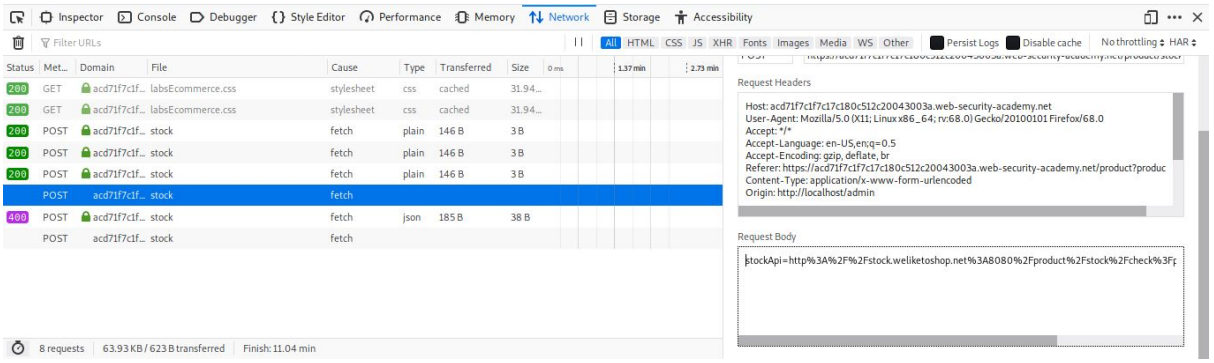


c) O que é SSRF?

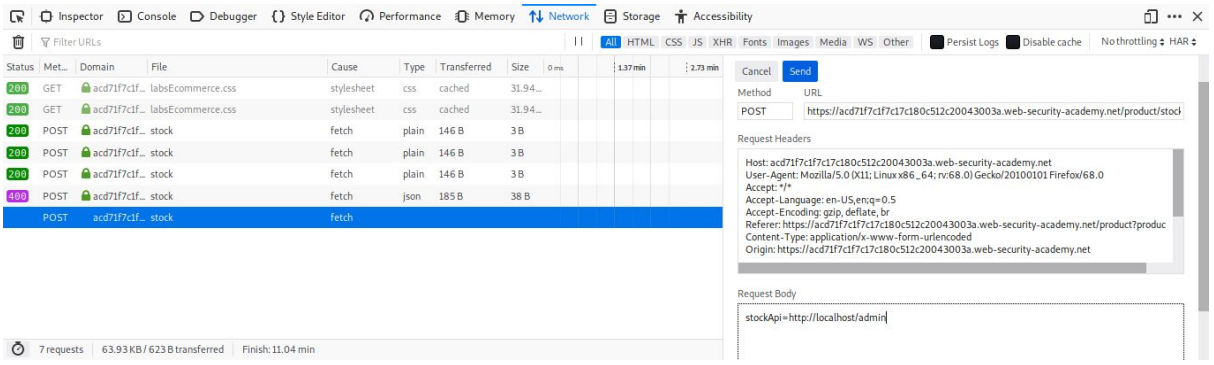
R: É um tipo de exploração em que um invasor faz com que um servidor acesse ou manipule informações no domínio dele próprio que, de outra forma, não seriam diretamente acessíveis ao invasor.

d) Mostre um exemplo de SSRF

R:

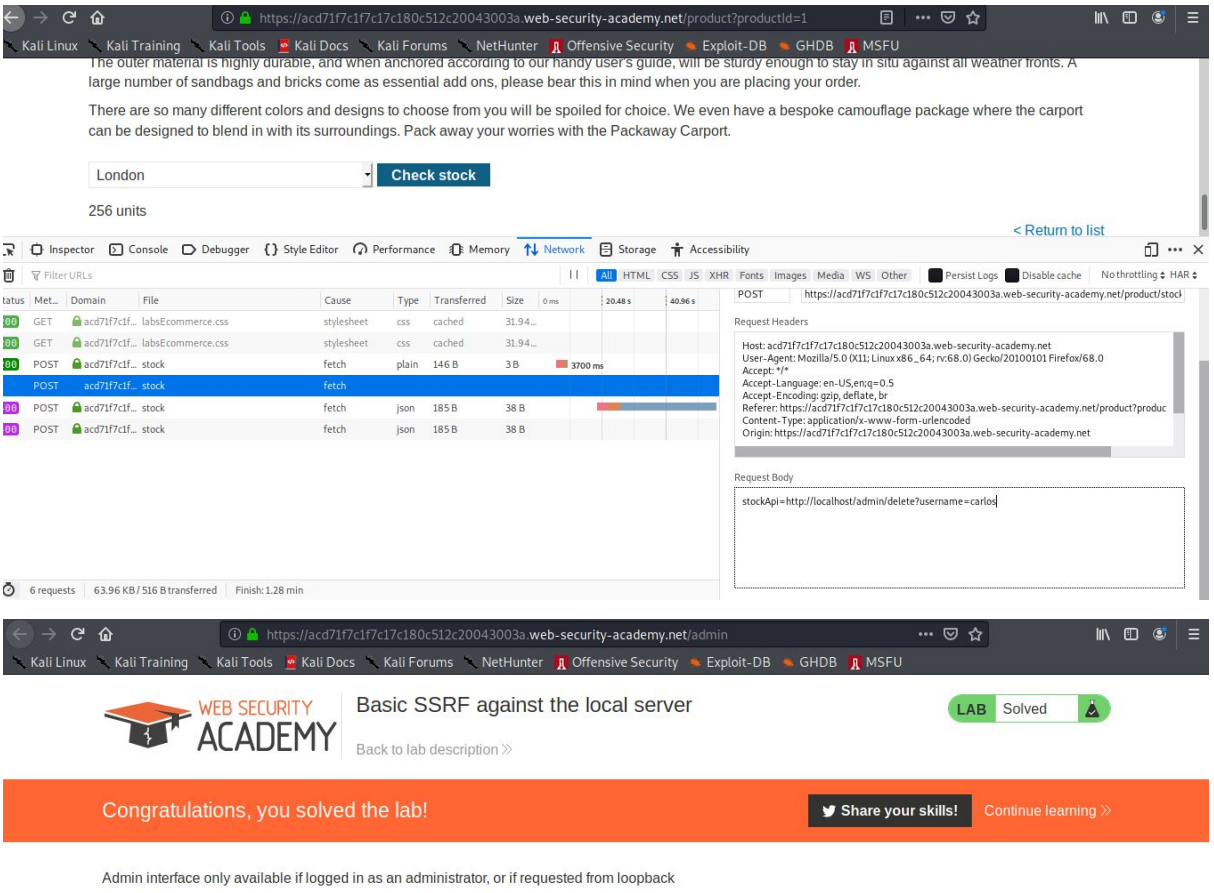


Vemos ali a requisição do site através da API para obter o número de unidades do produto em estoque. O que fazemos é alterar este endereço de requisição para um endereço crítico do servidor no qual não deveríamos ter acesso.



Depois de alterada, enviamos nossa falsa requisição:

Voila! Conseguimos acesso página de gerenciamento de usuário. Podemos até mesmo deletar um dos usuários do site, basta fazermos o mesmo processo.
Dessa vez obtemos a url que corresponde ao action de deletar o usuário e inserimos na api da requisição passada.



Pronto! Conseguimos deletar o usuário Carlos sem mesmo ter acesso a conta de administrador.

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