1. Hypertext Transfer Protocol, protocolo que pertence a sétima camada, a camada de aplicação. O HTTP utiliza o modelo cliente-servidor. Onde o cliente (programa requisitante) estabelece uma conexão com outro programa (servidor), fazendo-o uma requisição com o envio de um pacote de informações contendo alguns cabeçalhos (headers) a um URI ou URL. Recebendo as informações, o servidor responde com um recurso ou até mesmo outro header.

2. O response code é o código que indica o que aconteceu com a requisição, qual foi a operação realizada. Programa de Bruteforce que envia formulários e verifica se o status HTTP é 200.

3. Cabeçalhos HTTPS são o que permite que o cliente e servidor passem informações com a solicitação ou a resposta HTTP. Cabeçalhos são classificados em 4 tipos diferentes de acordo com seu contexo. Como uso inseguro do Header, podemos citar o HTTP header injection.

4. É o verbo do HTTP, identifica a ação que deve ser executada para o recurso. O GET passa os parâmetros da requisição no cabeçalho, possibilitando ver pela URI. Já o POST envia os parâmetros no corpo da requisisição HTTP, não aparecendo na URI. Além disso, a requisição por GET pode ser armazenada em cache e nos favoritos. O que não é possível para o método POST, logo o método POST é mais seguro exatamente por isso.

5. Cache é uma área de armazenamento de dados para que o próximo acesso seja mais rápido. Diretivas de cache-control padrão de requisições e respostas (Retirados do MDN Cache-Control – HTTP):

Cache-Control: max-age=<segundos>
Cache-Control: max-stale[=<segundos>]
Cache-Control: min-fresh=<segundos>
Cache-Control: no-cache
Cache-Control: no-store
Cache-Control: no-transform
Cache-Control: only-if-cached

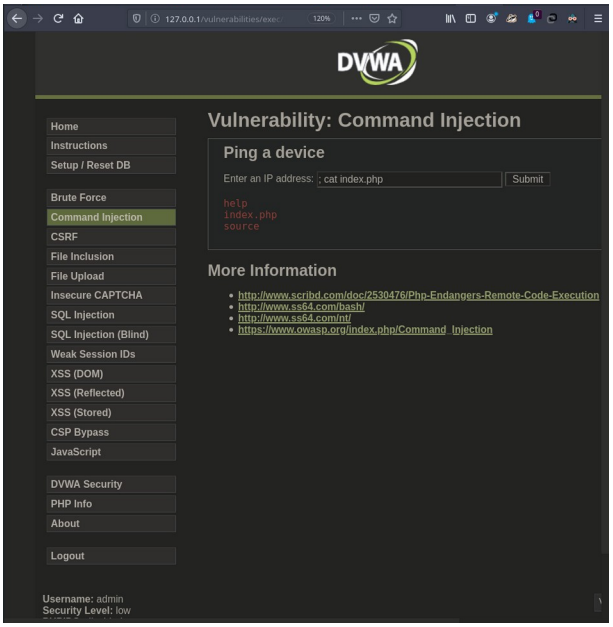
Cache-Control: must-revalidate
Cache-Control: no-cache
Cache-Control: no-store
Cache-Control: no-transform
Cache-Control: public
Cache-Control: private
Cache-Control: proxy-revalidate
Cache-Control: max-age=<segundos>
Cache-Control: s-maxage=<segundos>

6. Os cookies são pequenos arquivos que os sites visitados criam no computador do usuário pelo navegador. O principal ataque de cookies é o session hijacking (sequestro de cookie).

7. Um documento para desenvolvedores e aplicações Webs seguras. Falando quais são os riscos de aplicações webs mais criticos no ano.

8. Recon (reconnaissance) é uma fase de um pentest, onde se coleta e estuda todas as informações sobre sua target.
Sendo importante pois assim saberá com o que está lidando.
Poderá saber mais do sistema que está usando, o que torna mais fácil de pensar nos possíveis ataques.
Se souber dos detalhes do sistemas, pode pesquisar por vunerabilidades já expostas do sistema, economizando tempo.

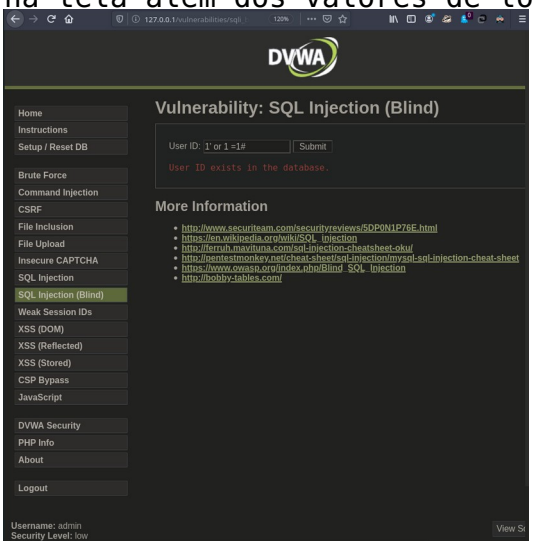
9. a. Command injection é um ataque que tem como como objetivo executar comandos no no sistema host que o servidor está rodando por meio de uma aplicação vunerável.
b.



(Comando usado: ls, após isso iria utilizar o cat index.php para pegar o index)

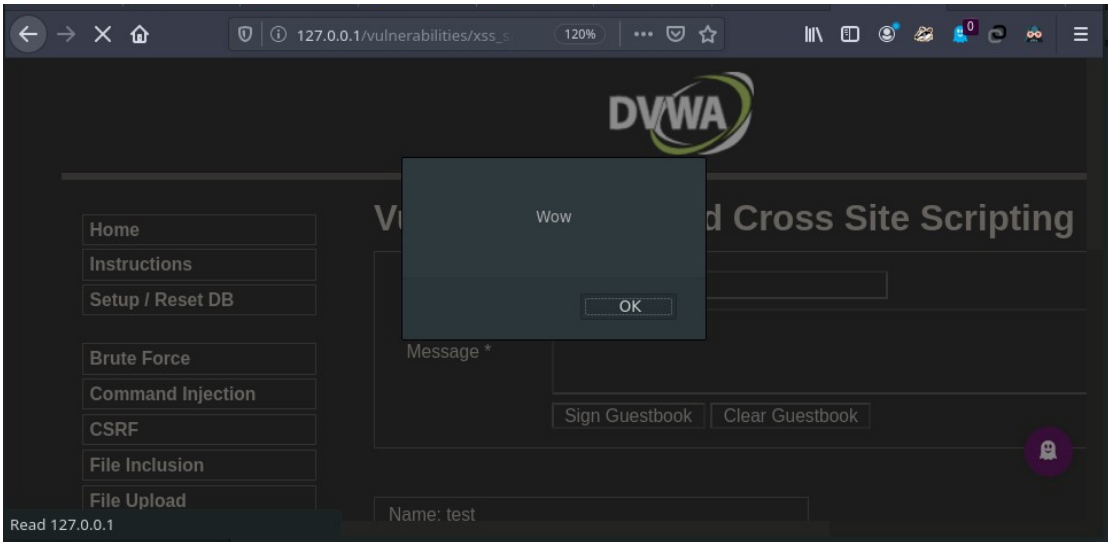
10. a. SQL injection é um tipo de ataque baseado em manipulação de código SQL (linguagem de banco de dados), inserindo ou manipulando consultas criadas pela aplicação que são enviadas para obanco de dados relacional.
b.Um Union Based Injection é um tipo de ataque SQL o qual perimtie que o atacante extraia informações do banco de dados ao aumentar os resultados (acrescentando no pedidO) da query original.

c.Blind SQL injection se difere pois faz perguntas de verdadeiro ou falso e determina a resposta baseando-se nas respostas, e seu tempo, da aplicação, sem retornar nada na tela além dos valores de lógica booleana.
d.

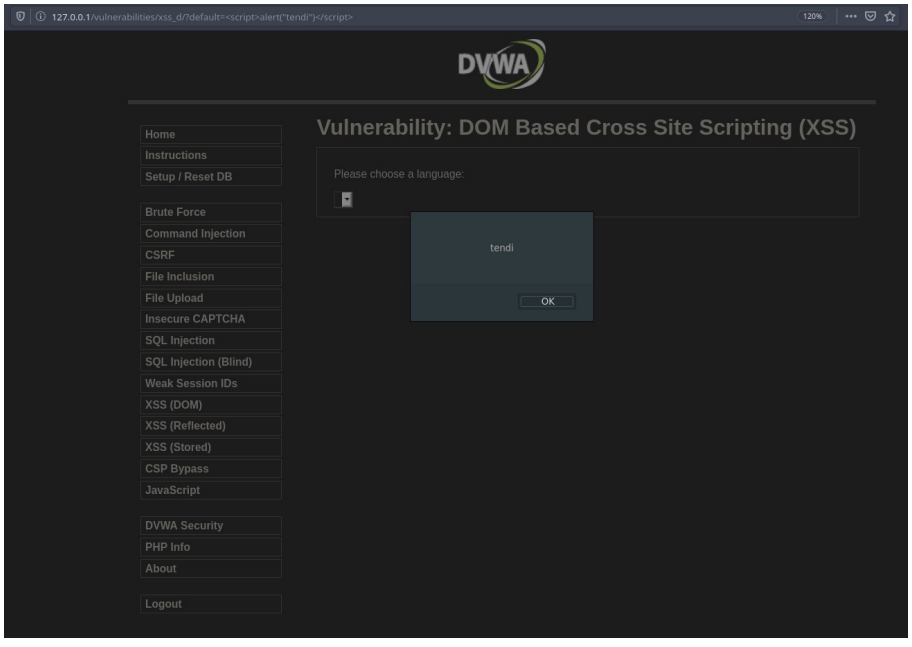


11. a. É um tipo de vunerabilidade de sites por onde um atacante é capaz de inserir scripts maliciosos e usá-los para instalar malwares nos navegadores dos usuários.
b. XSS (DOM): Vunerabilidade XSS onde o código source e de resposta vão ser os mesmos, podendo apenas encontrar o script da vunerabilidade na DOM da página ou observando o runtime.
XSS (Reflected): Vunerabilidade onde o servidor deliberadamente exibe os dados passados para aplicação, sem validação ou higienização.
XSS (Stored): Ocorre quando um servidor salva uma entrada malisiosa no banco de dados, fazendo o servidor não validar a saída e a retornar de maneira explicita bem como encontra-se no banco de dados.

c.



d.



12. a. Local File Include (LFI): Ataque que através da estrutura do site e por uma falha de programação,onde o atacante pega arquivos do servidor por meio de códigos na barra de endereço.

b. Remote File Inclusion (RFI): Ataque que é feito apartir da falta de validação do “input” do usuário, deixando passar scripts para uma aplicação WEB, executando-a no servidor de aplicação. Quando executado, o atacante pode roubar arquvios temporários, manipular arquivos, etc.

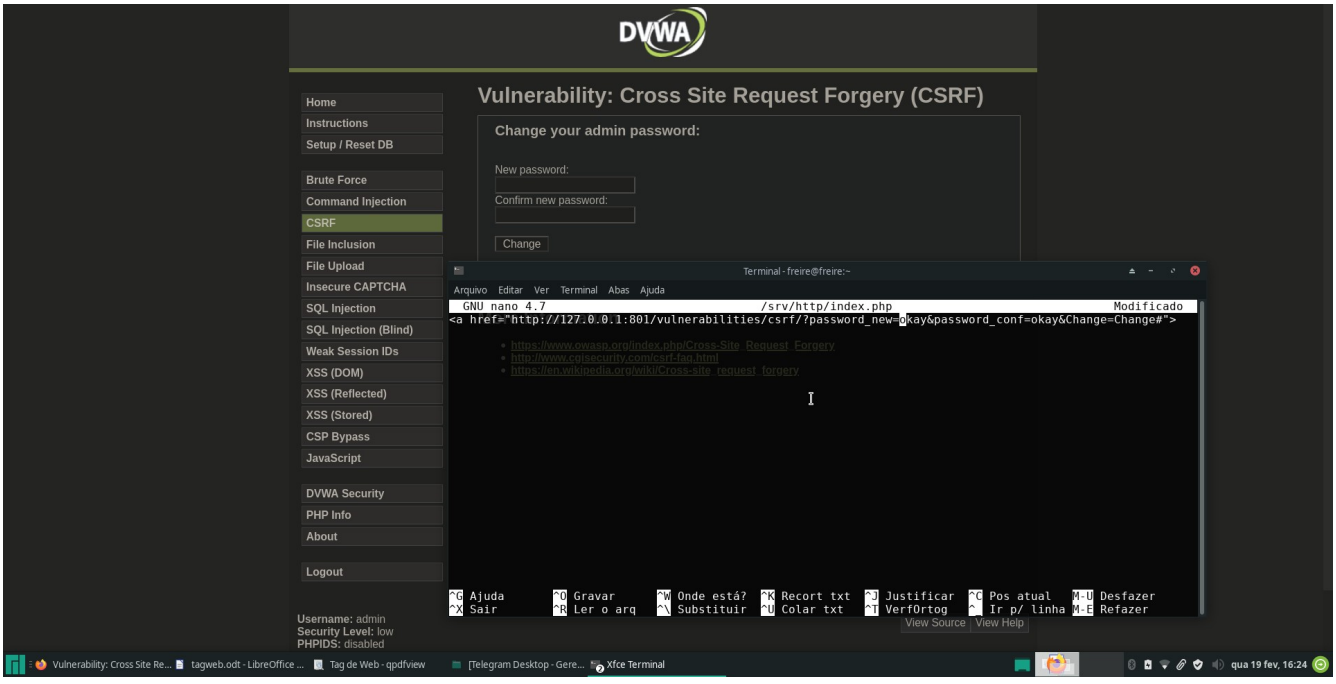
c. Ataque usado para obter acesso não autorizado a arquivos e diretórios, e, através disso, podendo com comprometer completamente o servidor.

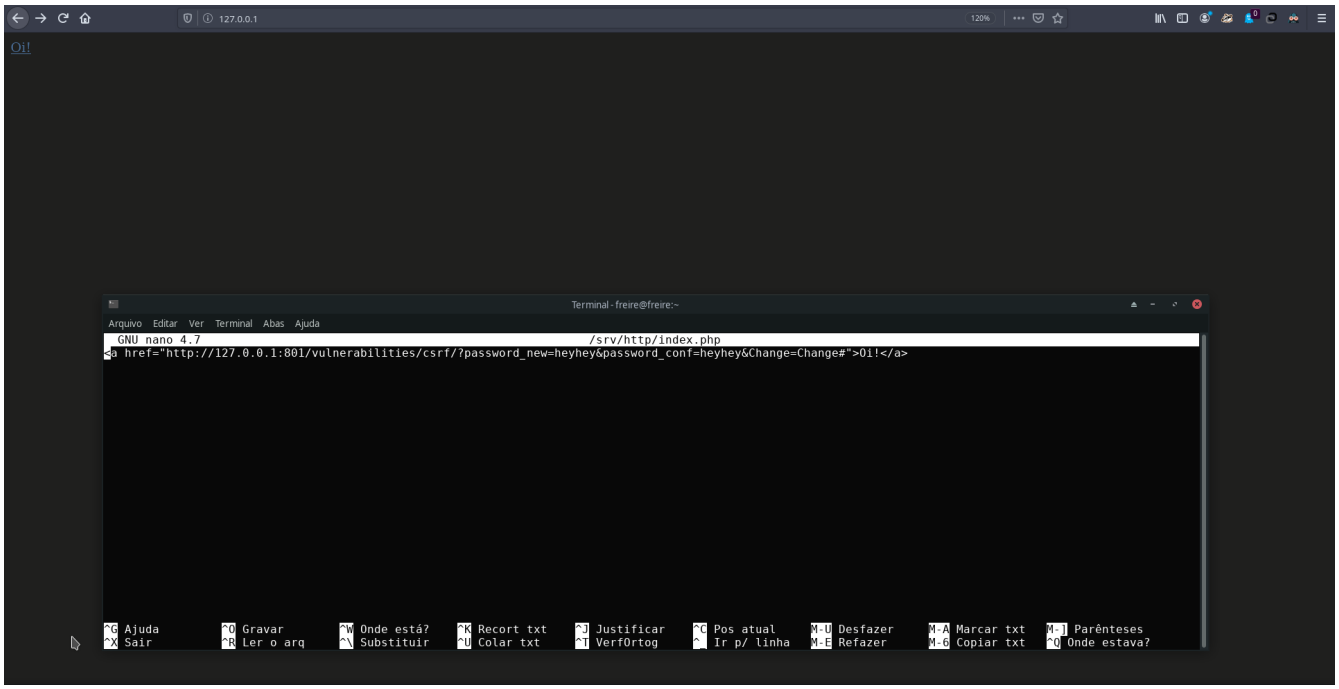
d. Usando o path traversal para acessar arquivos fora da raiz da pasta do site – por exemplo, ..;/..;/..;/..;/..;/etc/passwd – e pegando e vizualizando esses arquivos com o LFI.

e.

13. a. Ocorre quando a requisição web é feita entre sites, na tentativa de se passar por um usuário legítimo.

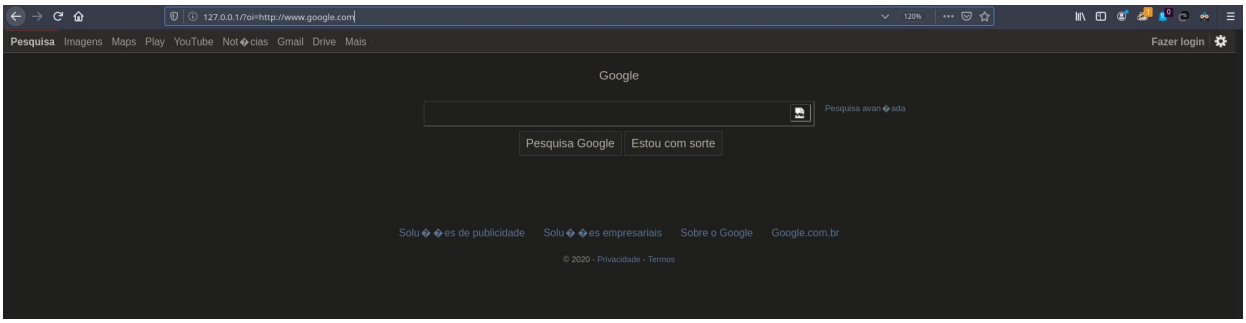
b.





c. Ataque o atacante pode abusar funcionalmente do servidor para ler e atualizar recursos internos, em suma, forja um request do próprio servidor.

d.



e. Usando CSRF token, pois ele mantém a integridade dos requests, verificando se ele foi realemnte feito pelo servidor.