


## TAG Web Hacking

- 1) HTTP é um protocolo de comunicação web request-response, ou seja, em uma situação de cliente e servidor o cliente enviaria uma mensagem request para o servidor, que irá enviar uma mensagem response de volta de acordo com o que foi solicitado.
- 2) Response Code é uma mensagem de resposta a uma request do cliente, que possuem cinco diferentes categorias de respostas dependendo do processo. Um exemplo de programa seria um de força bruta que ficaria adivinhando a senha caso desse erro 401 como retorno.
- 3) Header é um campo que contém informações sobre as mensagens de response e request. Um uso inseguro é o uso direto do comando nos headers, podendo ser vulnerável a HTTP injection.
- 4) Métodos são as ações de consulta ou modificação que podem ser feitos numa requisição. O método POST envia dados para criar ou atualizar algum recurso e seus dados podem ser postos no corpo da requisição. O método GET é usado para retornar dados e é enviado na URL, tendo um limite de informação dado pela URL. O método POST é mais seguro que o GET porque o último guarda informações na URL, podendo ser informações sensíveis, e fica visível e salvo no histórico.

- 5) Cache é uma técnica usada para salvar temporariamente documentos web com o fim de otimizar um serviço. Alguns recursos de um site podem ser salvos temporariamente em um disco rígido local para que o tempo de carregamento da página seja mais rápido. Headers usados para o controle de cache são Cache-Control, Pragma e ETag.
- 6) Cookies são dados baixados de serviços web para um computador que servem para guardar dados como informações de sessão (como carrinho de compras e login), preferências de cor e tamanho da fonte, entre outros dados. Um ataque envolvendo cookies é o sequestro deles, em que o atacante pode entrar na sessão da vítima utilizando os cookies roubados.
- 7) É um documento para desenvolvimento web que fala sobre os 10 riscos de segurança web considerados mais críticos.
- 8) Recon é a fase que tem como objetivo localizar, identificar, coletar e gravar informações sobre o alvo. Essa fase é importante para saber de possíveis falhas a serem exploradas e dados para serem pegos.
- 9)
  - a) É um ataque que se aproveita de uma vulnerabilidade que permite a injeção de comandos no shell do sistema.
  - b) Para fazer um command injection, foi inserido o && que serve para concatenar comandos no CMD do Windows, inserindo um echo logo em

seguida.



[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: Command Injection

### Ping a device

Enter an IP address:

### 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)

Username: admin View Source View Help



[Home](#)  
[Instructions](#)  
[Setup / Reset DB](#)  
  
[Brute Force](#)  
**[Command Injection](#)**  
[CSRF](#)  
[File Inclusion](#)  
[File Upload](#)  
[Insecure CAPTCHA](#)  
[SQL Injection](#)

## Vulnerability: Command Injection

### Ping a device

Enter an IP address:

### 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)

**DVWA**

**Vulnerability: Command Injection**

**Ping a device**


Enter an IP address:

Disparando 127.0.0.1 com 32 bytes de dados:  
 Resposta de 127.0.0.1: bytes=32 tempo<1ms TTL=128  
 Resposta de 127.0.0.1: bytes=32 tempo<1ms TTL=128  
 Resposta de 127.0.0.1: bytes=32 tempo<1ms TTL=128  
 Resposta de 127.0.0.1: bytes=32 tempo<1ms TTL=128

Estatísticas do Ping para 127.0.0.1:  
 Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),  
 Aproximar um número redondo de vezes em milissegundos:  
 Mínimo = 0ms, Máximo = 0ms, Média = 0ms  
 ola, tudo bem?

10)

- a) É a inserção de requisições SQL via input de dados do cliente.
- b) Union Based Attack consiste quando é usado o comando UNION na injeção de comandos SQL, o que possibilita colocar mais SELECT e selecionar outras informações do sistema.
- c) É a inserção de comandos SQL que retornam valores de verdadeiro ou falso, sendo normalmente enviada uma mensagem de erro quando falso e nada quando verdadeiro. A partir disso, pode-se saber mais sobre o sistema que quer invadir.
- d) Foi verificado que o ID “1” existia no banco. Posteriormente, vimos que o 1’ não estava contido, então foi colocado uma comparação para ver comportamento do sistema com a entrada, gerando um resultado verdadeiro.




[Home](#)  
[Instructions](#)  
[Setup / Reset DB](#)  
  
[Brute Force](#)  
[Command Injection](#)  
[CSRF](#)  
[File Inclusion](#)  
[File Upload](#)  
[Insecure CAPTCHA](#)  
[SQL Injection](#)  
**[SQL Injection \(Blind\)](#)**  
[Weak Session IDs](#)

## Vulnerability: SQL Injection (Blind)

User ID:

### 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/>



[Home](#)  
[Instructions](#)  
[Setup / Reset DB](#)  
  
[Brute Force](#)  
[Command Injection](#)  
[CSRF](#)  
[File Inclusion](#)  
[File Upload](#)  
[Insecure CAPTCHA](#)  
[SQL Injection](#)  
**[SQL Injection \(Blind\)](#)**

## 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/>




## Vulnerability: SQL Injection (Blind)

[Home](#)  
[Instructions](#)  
[Setup / Reset DB](#)  
  
[Brute Force](#)  
[Command Injection](#)  
[CSRF](#)  
[File Inclusion](#)  
[File Upload](#)  
[Insecure CAPTCHA](#)  
[SQL Injection](#)  
[SQL Injection \(Blind\)](#)

User ID:    
 User ID is MISSING from 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/>



## Vulnerability: SQL Injection (Blind)

[Home](#)  
[Instructions](#)  
[Setup / Reset DB](#)  
  
[Brute Force](#)  
[Command Injection](#)  
[CSRF](#)  
[File Inclusion](#)  
[File Upload](#)  
[Insecure CAPTCHA](#)  
[SQL Injection](#)  
[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/>

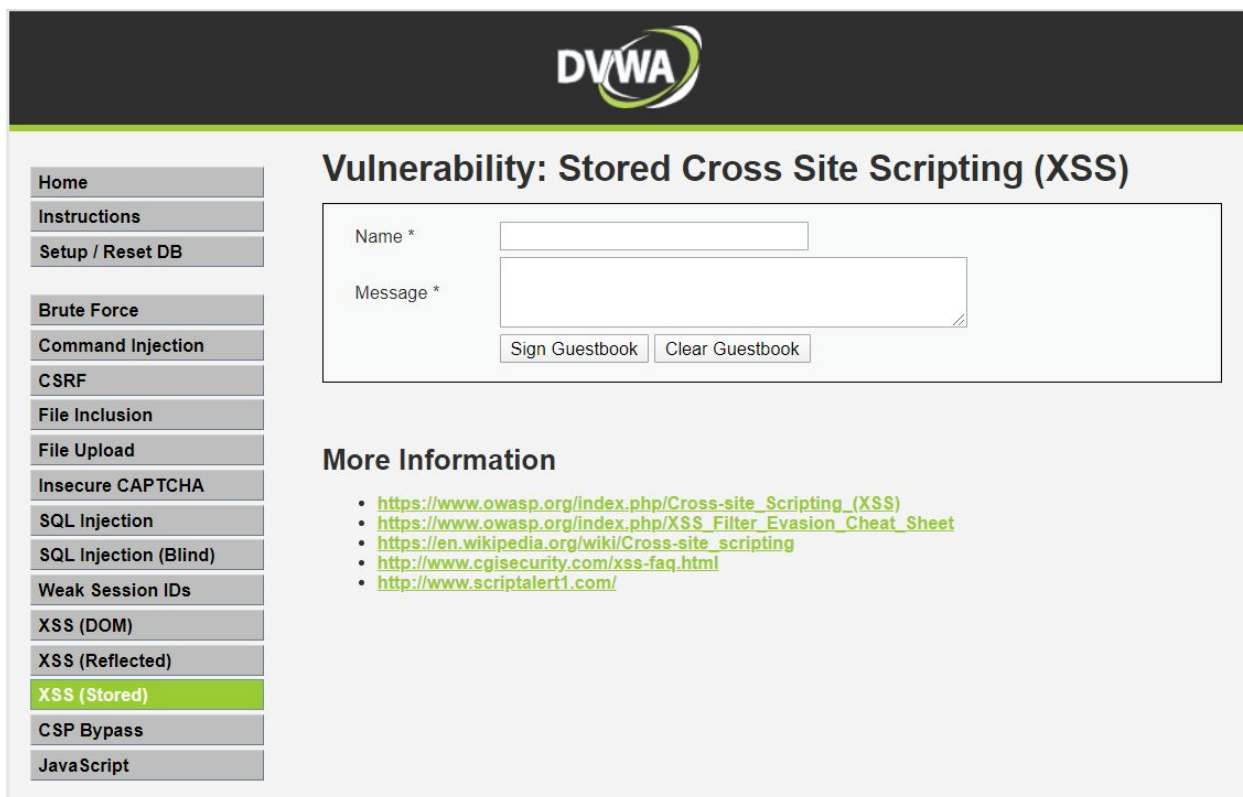
11)

a) XSS é um ataque de injeção no qual é inserido código em sites e eles são executados pelo navegador.

b) XSS Stored, DOM e Reflected. Stored é quando o script é armazenado no servidor alvo, seja inserindo em um banco de dados, em mensagens, dentre outros. DOM executa o código a partir do navegador da vítima. Reflected é

usado em um local do site que não armazena dados, podendo ser injetado na URL.

c) Para explorar a vulnerabilidade, foi injetado um script no campo de mensagem. Como o campo não foi devidamente tratado, o comando foi executado ao dar sign.



**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

### Vulnerability: Stored Cross Site Scripting (XSS)


Name \*

Message \*

#### More Information

- [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/XSS\\_Filter\\_Evasion\\_Cheat\\_Sheet](https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet)
- [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- <http://www.cgisecurity.com/xss-faq.html>
- <http://www.scriptalert1.com/>





- 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

## Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

Message \*

### More Information

- [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/XSS\\_Filter\\_Evasion\\_Cheat\\_Sheet](https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet)
- [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- <http://www.cgisecurity.com/xss-faq.html>
- <http://www.scriptalert1.com/>

- 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

localhost diz

hasta la vista

## Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

Message \*

Name: baby

Message:


### More Information

- [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/XSS\\_Filter\\_Evasion\\_Cheat\\_Sheet](https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet)
- [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- <http://www.cgisecurity.com/xss-faq.html>
- <http://www.scriptalert1.com/>

d) Um script foi inserido na URL do site e executado após pressionar Enter.



localhost/vulnerabilities/xss\_d/



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)**

## Vulnerability: DOM Based Cross Site Scripting (XSS)


Please choose a language:

English

### More Information

- [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/Testing\\_for\\_DOM-based\\_Cross\\_site\\_scripting\\_\(OTG-CLIENT-001\)](https://www.owasp.org/index.php/Testing_for_DOM-based_Cross_site_scripting_(OTG-CLIENT-001))
- <https://www.acunetix.com/blog/articles/dom-xss-explained/>

localhost/vulnerabilities/xss\_d/?default=<script>alert("ola")</script>



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)**

## Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

English

### More Information

- [https://www.owasp.org/index.php/Cross-site\\_Scripting\\_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- [https://www.owasp.org/index.php/Testing\\_for\\_DOM-based\\_Cross\\_site\\_scripting\\_\(OTG-CLIENT-001\)](https://www.owasp.org/index.php/Testing_for_DOM-based_Cross_site_scripting_(OTG-CLIENT-001))
- <https://www.acunetix.com/blog/articles/dom-xss-explained/>

localhost/vulnerabilities/xss\_d/?default=<script>alert("ola")</script>

localhost diz

ola

OK

12)

a) Local File Inclusion é a inserção de arquivos que já estão no servidor local a partir da injeção do endereço do arquivo na entrada.

- b) Remote File Inclusion é a inserção de arquivos remotos a partir do recebimento de uma URL que contém esses arquivos.
- c) Path Transversal é a vulnerabilidade em que o atacante consegue navegar nos diretórios da vítima.
- d) Ambos podem se aliados ao explorar os diretórios com a falha de Path Transversal e posteriormente incluir o arquivo local com a falha LFI.
- e)

13)

- a) CSRF é uma vulnerabilidade que faz com que a vítima receba uma requisição pelo atacante, podendo mudar a senha e/ou e-mail de uma conta dentre outras ações que podem ser feitas através de requisições.
- b)
- c) SSRF é uma vulnerabilidade que permite com que o atacante direcione requisições HTTP de um site para outro domínio, podendo ser para o próprio atacante.
- d)
- e) Pode-se usar um token de validação CSRF.