Aplicaciones Web II

CC5325 - Taller de Hacking Competitivo
Diego Vargas

Contenido

- Inyecciones
  - XSS
  - SQL
  - Otros
- Ejecución de código remoto
- Reverse shell
- Demo

Inyecciones

Cross-Site Scripting

Cross-Site Scripting (XSS)

Objetivo:

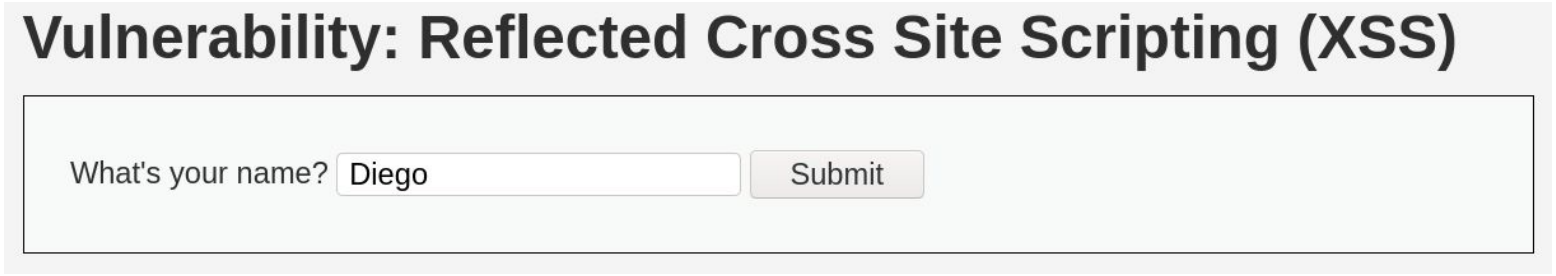
- Ejecutar elementos javascript en el navegador de la víctima.
- Usualmente es necesario bypassear filtros y WAFs.

Tipos:

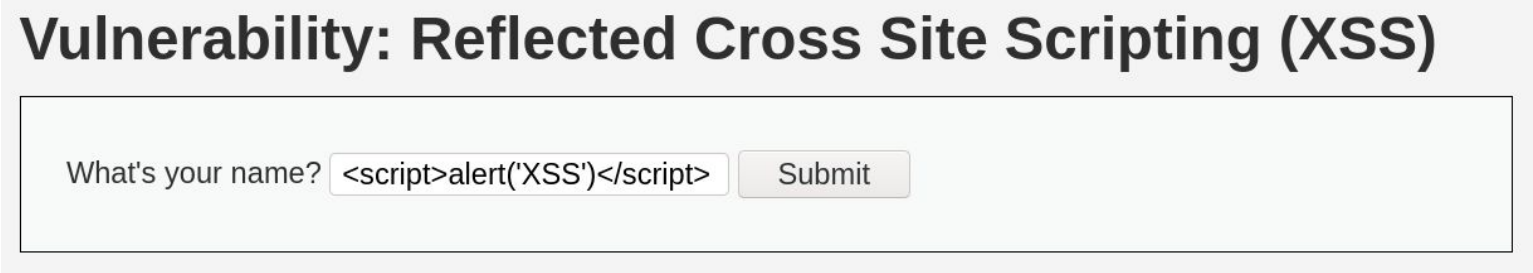
1. Almacenado
2. Reflejado
3. DOM

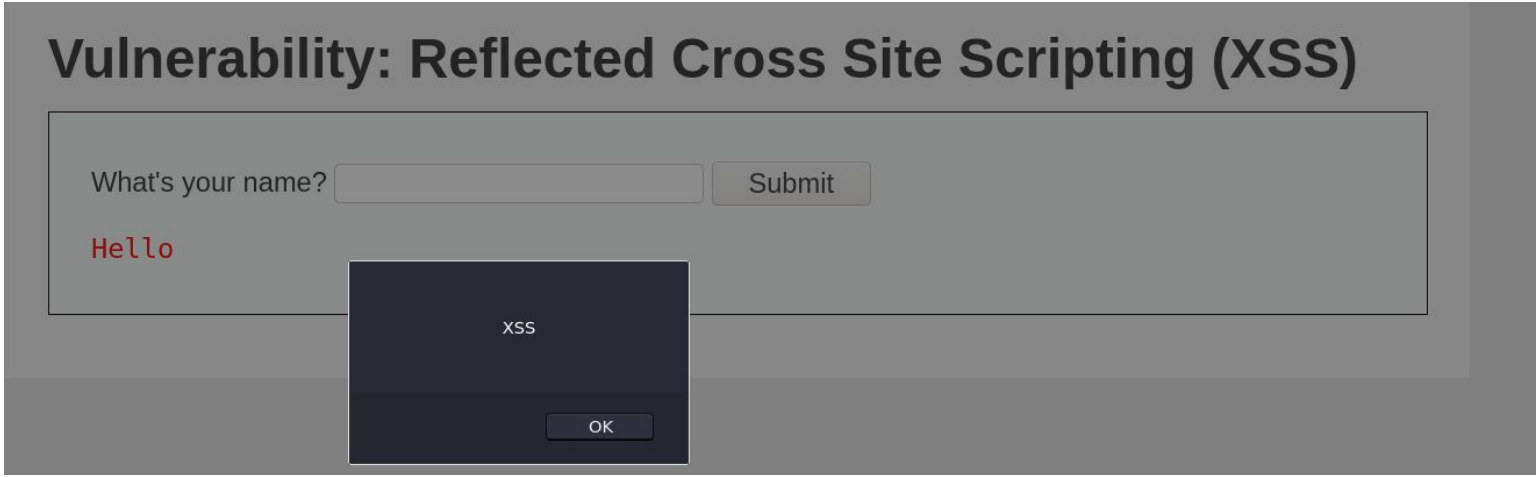
```
<script>
	...code…
</script>

<script src=”https://url.com/code.js”></script>

<button onclick=”...code…”></button>
```

Ejemplo: XSS Reflejado



Ejemplo: XSS Reflejado





Inyecciones
SQL

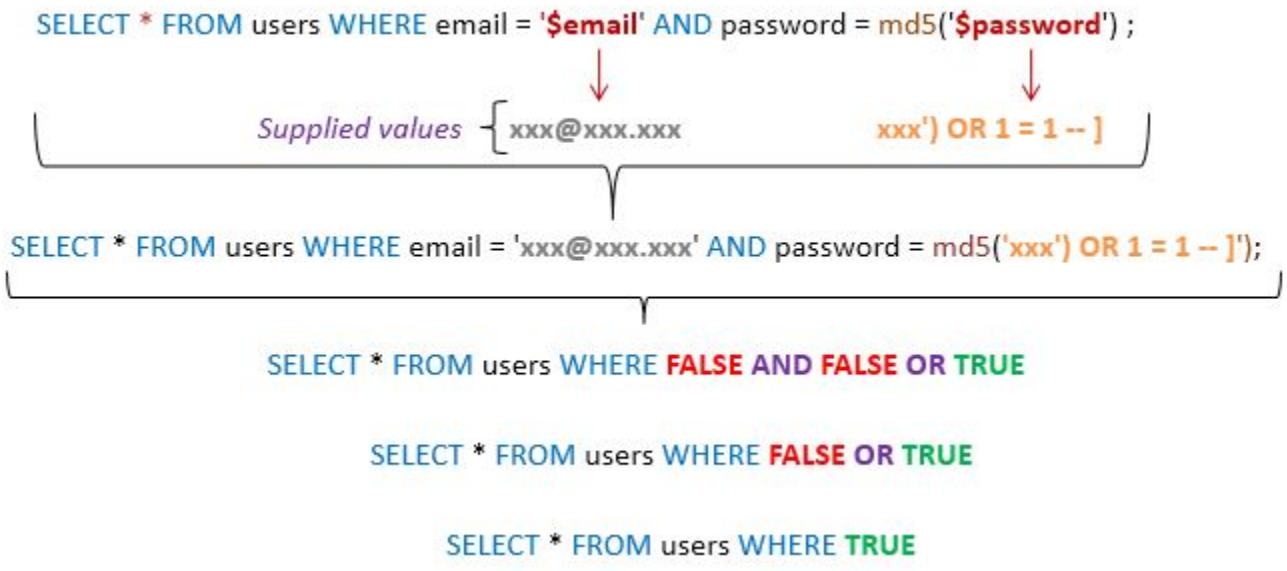
SQL Injection (SQLi)

Objetivo:

- Inyectar comandos SQL.
- Extraer información de la DB.
- Modificar información.
- Ejecutar comandos de sistema.

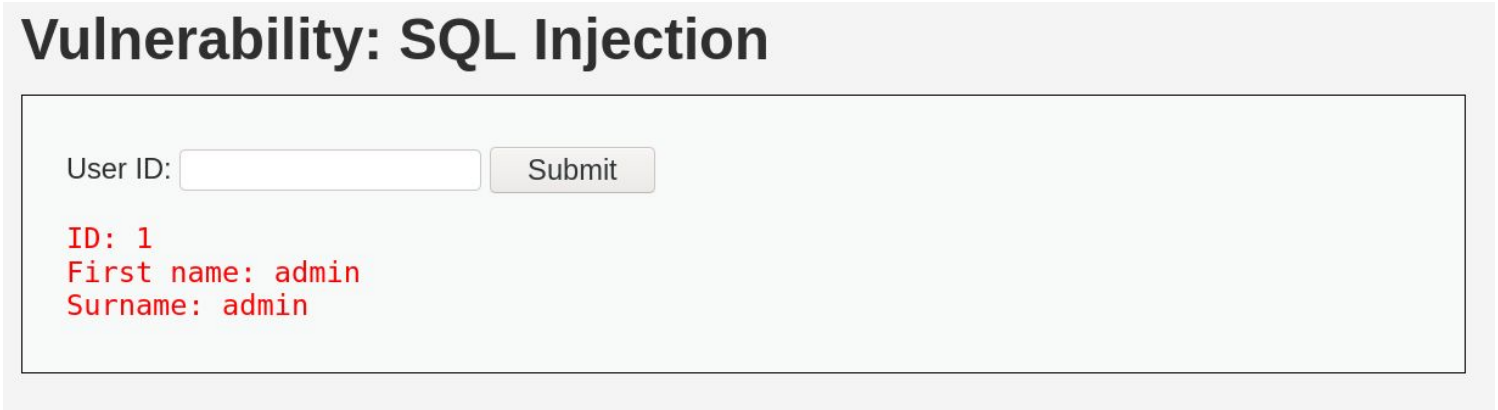
Tipos:

1. Non-Blind
2. Blind (Boolean based, time based)



Descomposición de una SQLi

Ejemplo: SQLi



Ejemplo: SQLi

```
GET /vulnerabilities/sqli/?id='&Submit=Submit HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://localhost/vulnerabilities/sqli/
Cookie: PHPSESSID=a1jdilfph0kna9tmjg3dkpvs76; security=low
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Sun, 14 Feb 2021 13:41:52 GMT
Server: Apache/2.4.25 (Debian)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 162
Connection: close
Content-Type: text/html; charset=UTF-8

<pre>You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right
syntax to use near ''''' at line 1</pre>
```

Ejemplo: SQLi

```
SELECT id, first_name, surname FROM people WHERE id = '$ID';
```


```
SELECT id, first_name, surname FROM people WHERE id = '' or '1'='1';
```

Ejemplo: SQLi

Vulnerability: SQL Injection

User ID: Submit

```
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
```

Inyecciones

Otros

Otros Tipos de Inyecciones

- NoSQLi
- XML
  - Tag Injection
  - XML eXternal Entity (XXE)
  - XML Entity Expansion (XEE)
- LDAP
- XPath

Ejecución de Código Remoto

Remote Code Execution (RCE)

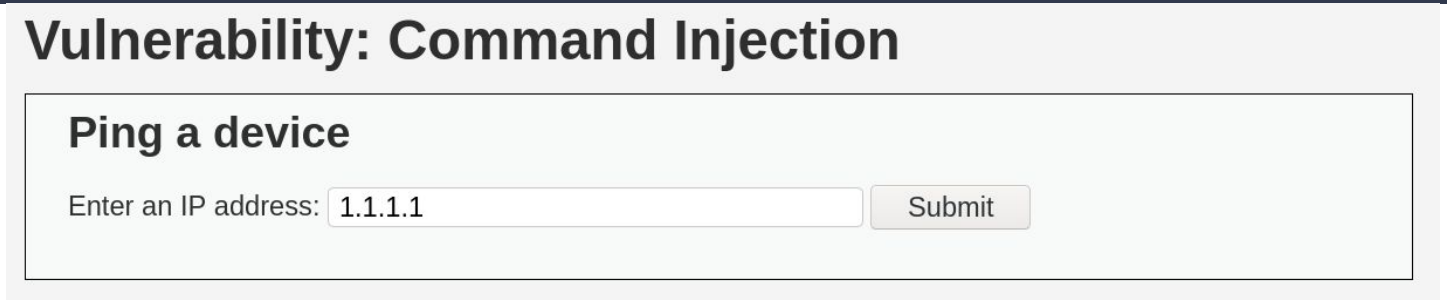
Objetivo:

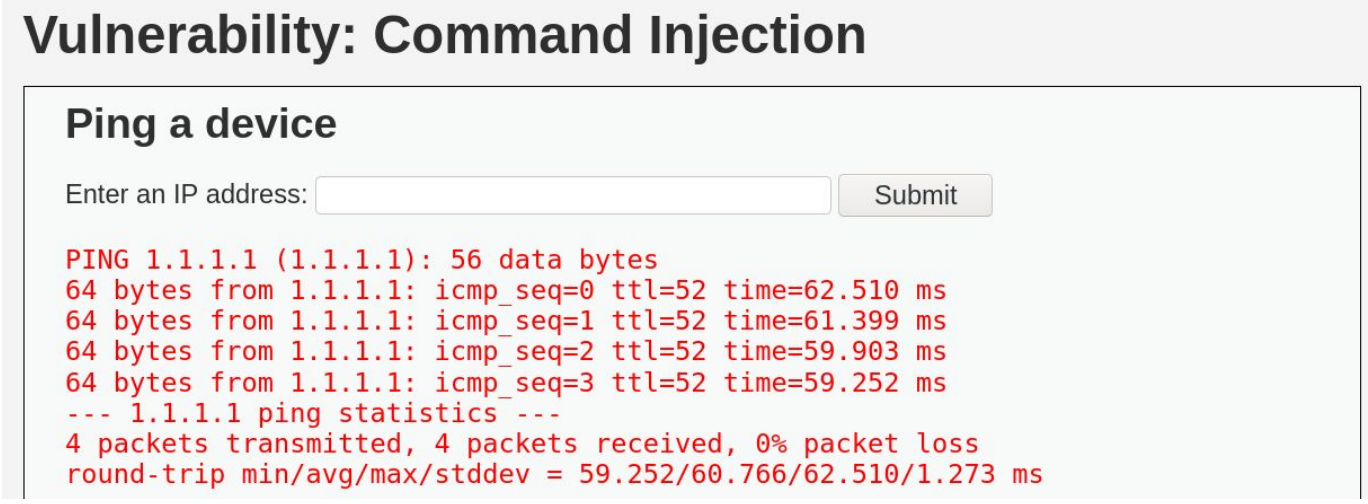
- Forzar la ejecución de comandos de sistema.
- Su alcance depende del lenguaje, framework y OS utilizado en el servidor.
    - Sistema de archivos.
    - Variables de ambiente.
    - Procesos.
    - Red interna.

Métodos de Ejecución

- Subiendo archivos ejecutables (webshell)
- Inyección de comandos
- SQLi
- Buffer Overflows
- Deserialización insegura
- Type Confusion

Ejemplo: Command Injection





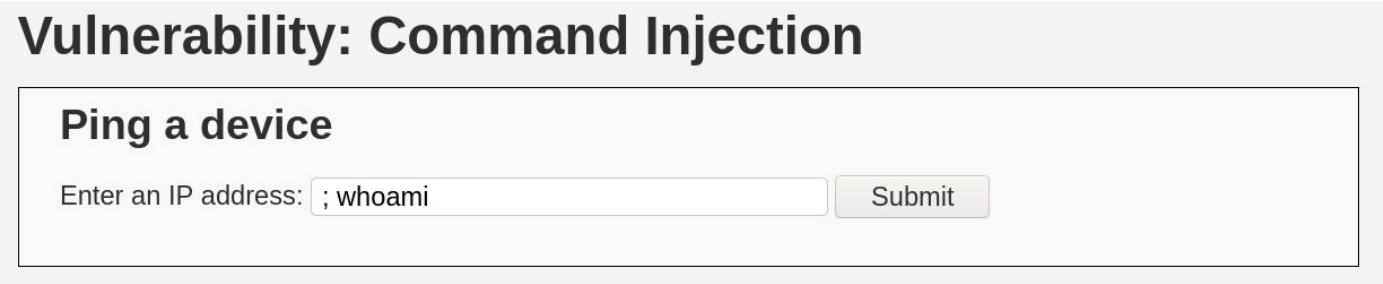
Ejemplo: Command Injection

```
└─$ ping -c 4 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=59 time=14.7 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=59 time=16.3 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=59 time=31.4 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=59 time=15.8 ms

--- 1.1.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 14.689/19.566/31.395/6.855 ms
```

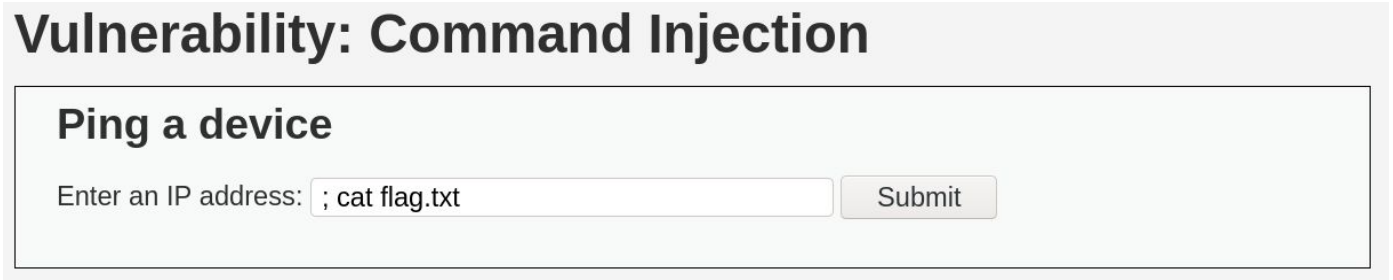
```
system('ping -c 4 $IP');
```

```
system('ping -c 4 ; whoami');
```

Ejemplo: Command Injection

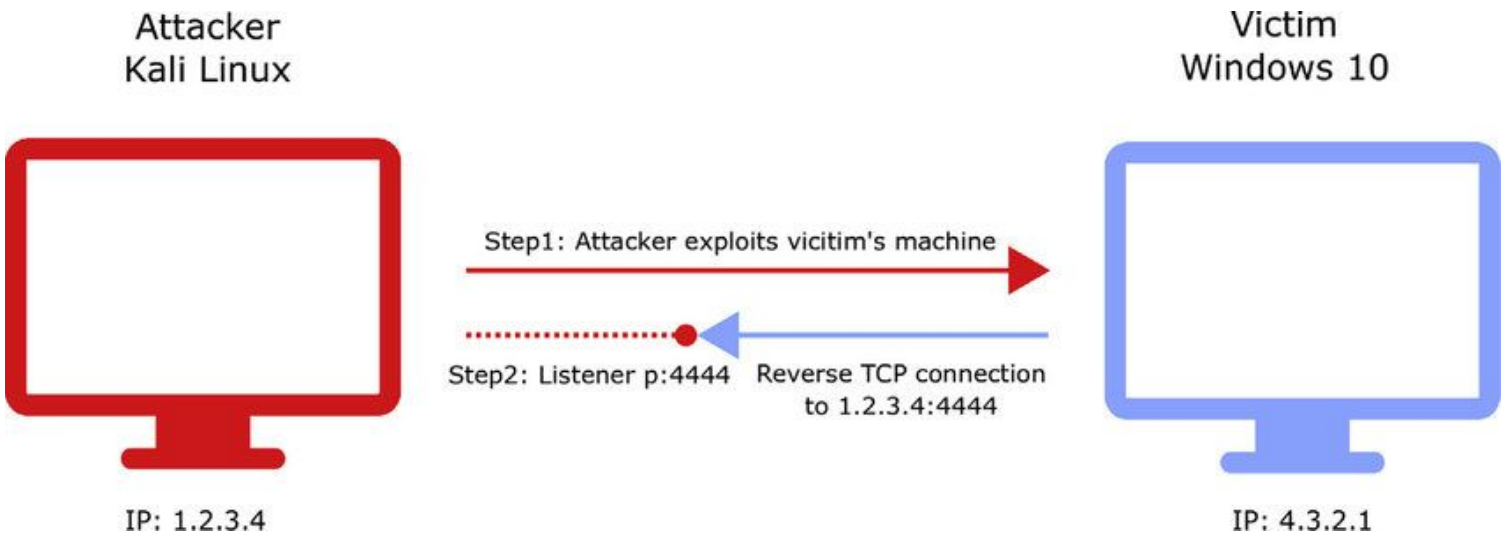


Ejemplo: Command Injection



Reverse Shell

Reverse Shell



Reverse Shell

Objetivo:

- Explotar el RCE para obtener acceso directo a una terminal.
- Más cómodo y rápido que ejecutar cada comando por RCE.

Demo

Herramientas

- Burp
- sqlmap (https://github.com/sqlmapproject/sqlmap)
- DVWA (https://github.com/opsxcq/docker-vulnerable-dvwa)