



Universidad del
Rosario



MACC
Matemáticas Aplicadas y
Ciencias de la Computación

Hacking Web Applications

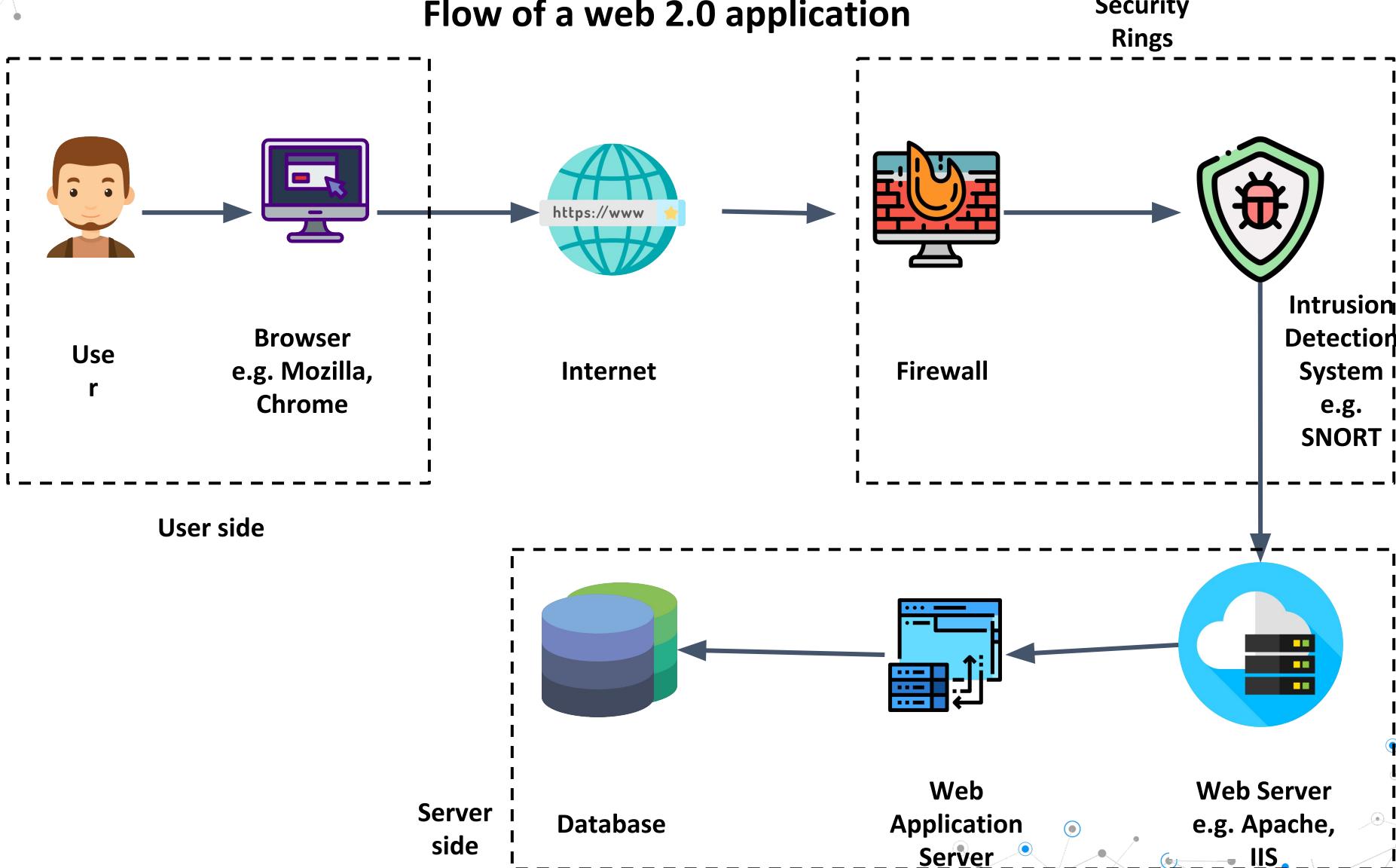
Hacking Ético

Daniel Orlando Díaz López, PhD

Profesor principal
Departamento MACC
Universidad del Rosario
danielo.diaz@urosario.edu.co



Flow of a web 2.0 application



Web 1.0

Sitios web con las siguientes características:

- Unidireccional (el usuario ve el contenido de manera pasiva)
- Mayoritariamente consumidores de contenido
- Páginas estáticas (HTML estático)
- Contenido desplegado desde un sistema de archivos local en lugar de una base de datos
- Uso de SSI o CGI (Common Gateway Interface)

Web 2.0

Sitios web con las siguientes características:

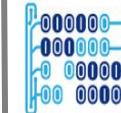
- Participativo
- Contenido generado por el usuario
- Cultura participativa
- Interoperabilidad
- Uso de lenguajes de programación dinámicos (PHP, Perl, Python, Ruby)



Hacking Web Applications



Universidad del
Rosario

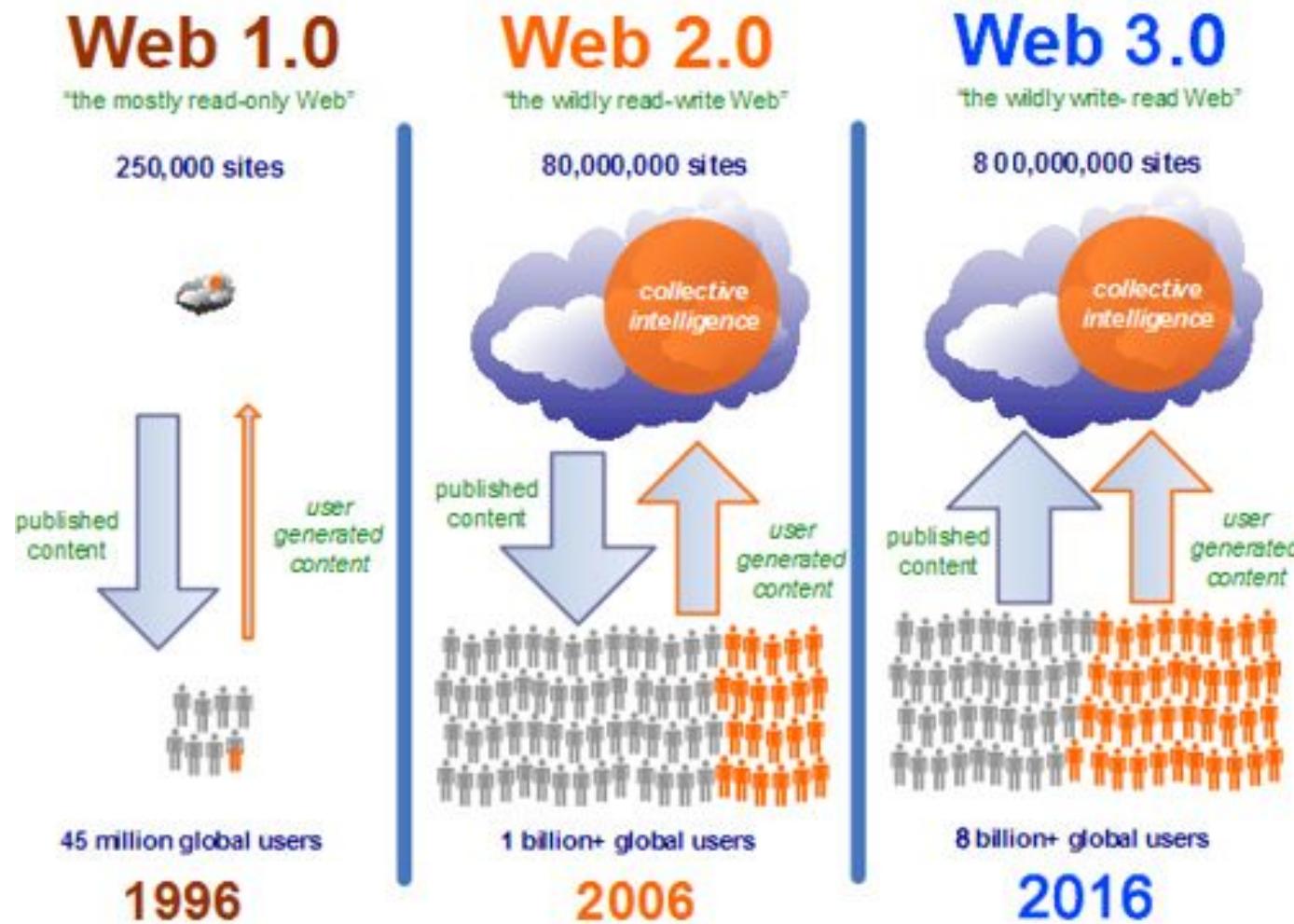
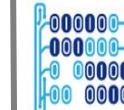


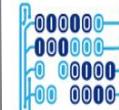
MACC
Matemáticas Aplicadas y
Ciencias de la Computación



Web 1.0	Web 2.0
Banner ads on websites	Automatic text, image, video, and interactive media advertisements, that are targeted to website content and audience
Ofoto, an online digital photography website, on which users could store, share, view and print digital photos	Flickr, an image hosting and video hosting website and web services suite
content delivery networks (CDN)	BitTorrent and eMule, communications protocols of peer-to-peer file sharing (P2P) which is used to distribute data and electronic files over the Internet
mp3.com, a website providing information about digital music and artists, songs, services, community, and technologies and a legal, free music-sharing service	Napster, a pioneering peer-to-peer (P2P) file sharing Internet service that emphasized sharing digital audio files, typically songs, encoded in MP3 format
Britannica Online, written by professionals and experts	Wikipedia, can be written and edited by any person, even amateurs and non-experts
personal websites	blogging
evite	upcoming.org and EVDB
domain name speculation	search engine optimization (SEO)
page views	cost per click
"screen scraping"	web services
publishing of online documents, once approved by gatekeepers and editorial staff	mass user participation, without approval of content by gatekeepers or editorial staff
content management systems	wikis that allow almost any users to contribute
directories (taxonomy)	"tagging" of websites, images and videos (folksonomy)
"stickiness"	syndication







Web 1.0	Web 2.0	Web 3.0
The Web	The Social Web	The Semantic Web
Read-only Web	Read and Write Web	Read, Write and Execute Web
Information sharing	Interaction	Immersion
Connect Information	Connect People	Connect context, people and knowledge
All about static content (one-way interaction)	Two-way communication through social networking, blogging etc.	Visualization
Owning content	Sharing content	Consolidating content
Web Forms	Web Applications	Smart Applications
HTML Portals	XML/RSS	RDF/RDFS/OWL
Banner Advertising	Interactive advertising	Behavioral Advertising
Britannica Online	Wikipedia	Semantic Web



- Open Web Application Security Project (OWASP) is a non-profit community which help organizations to develop and maintain secure application
- Everything is free, open and without commercial influence
- OWASP products:
 - Security tools
 - Documents, standards, books,
 - Local chapters
 - Conferences
 - Mailing lists
- One of the most popular products: **OWASP TOP 10**

<https://www.owasp.org>

https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf



OWASP

The Open Web Application Security Project

Realización de un ataque a una aplicación web a partir de una interceptación de tráfico

Laboratorio

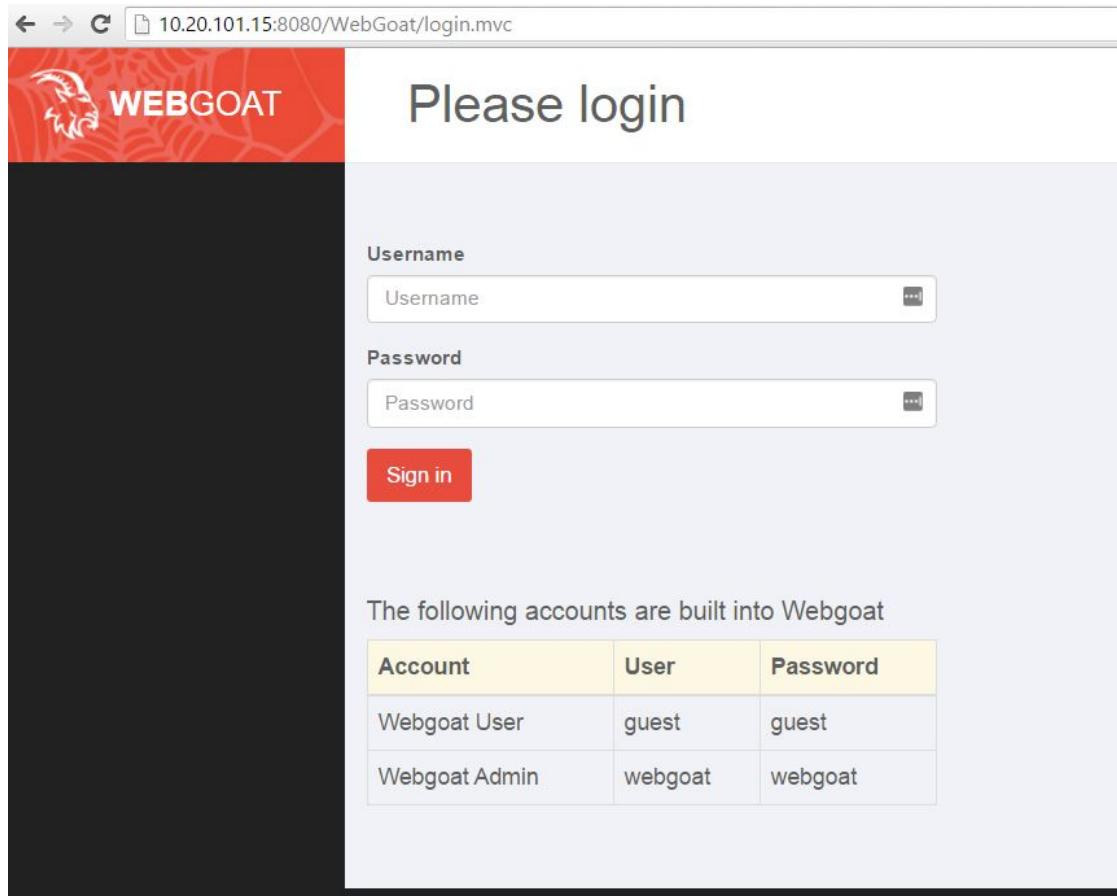
1. Implementar un servidor de aplicaciones web vulnerable (WebGoat)
2. Instalar una herramienta de interceptación de tráfico (ZAP)
3. Realizar un ataque de inyección de código sobre el servidor víctima

Preguntas

1. Explique la forma de mitigar un ataque de inyección de código sobre una aplicación web

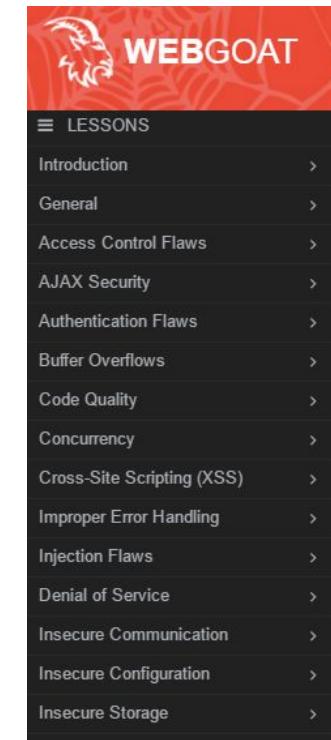
¡Instalemos un servidor web vulnerable!

- **WebGoat:** Deliberately insecure web application for interactive teaching of web application security



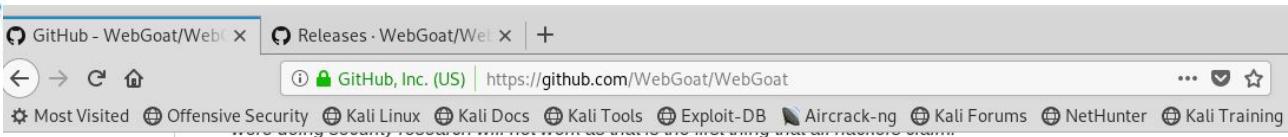
The screenshot shows the WebGoat login interface. At the top, there's a header bar with the WebGoat logo and the text "WEBGOAT". Below it is a "Please login" page. It features two input fields: "Username" and "Password", each with a small icon to its right. A red "Sign in" button is positioned below the password field. Below the login form, a message states: "The following accounts are built into Webgoat". A table lists two accounts:

Account	User	Password
Webgoat User	guest	guest
Webgoat Admin	webgoat	webgoat



<https://github.com/WebGoat/WebGoat>

<https://github.com/WebGoat/WebGoat/releases>



Installation Instructions:

1. Standalone

Download the latest WebGoat release from <https://github.com/WebGoat/WebGoat/releases>

```
java -jar webgoat-server-8.0.0.VERSION.jar [--server.port=8080] [--server.address=localhost]
```

The latest version of WebGoat needs Java 11. By default it runs on port 8080. If you want to run it on a different port. With `server.address` you can bind to a different host.

7.0.1
f825bea

The OWASP WebGoat 7.0.1 Release

dougmorato released this on Feb 1, 2016

WebGoat 7 is the latest in a series of infrastructure improvements to move WebGoat into the modern era. With the new plugin architecture and separation of the server framework from the lessons, lessons now require just a few lines of code. Lessons can now be produced without having to understand the entirety of the WebGoat server.

This release contains both the WebGoat container and 50+ lessons created by the WebGoat team.

Assets 5

 webgoat-container-7.0.1-war-exec.jar	70.7 MB
 webgoat-container-7.0.1.jar	334 KB
 webgoat-container-7.0.1.war	61.7 MB
 Source code (zip)	
 Source code (tar.gz)	

Seleccionar la versión de java a utilizar. En este caso seleccionar la versión Java 1.8.0 de la siguiente forma:

```
root@kali:~/Downloads# update-alternatives --install /usr/bin/java java /usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java 1
root@kali:~/Downloads# update-alternatives --set java /usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java
```

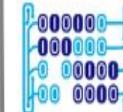
Validar que la versión de Java sea la correcta con el siguiente comando:

```
root@kali:~/Downloads# java -version
openjdk version "1.8.0_171"
OpenJDK Runtime Environment (build 1.8.0_171-8u171-b11-2-b11)
OpenJDK 64-Bit Server VM (build 25.171-b11, mixed mode)
```

Ejecutar Webgoat desde la ruta donde se haya descargado (en mi caso /root/Downloads)

```
root@kali:~/Downloads# java -jar webgoat-container-7.0.1-war-exec.jar
```

```
2019-10-20 22:53:33,826 INFO  - FrameworkServlet 'mvc-dispatcher': initialization completed in 514 ms
2019-10-20 22:53:33,840 INFO  - Initializing main webgoat servlet
2019-10-20 22:53:33,845 INFO  - Browse to http://localhost:8080/WebGoat and happy hacking! ¡No se ha perdido!
Oct 20, 2019 10:53:34 PM org.apache.coyote.http11.Http11Protocol start
INFO: Starting ProtocolHandler ["http-bio-8080"]
Como resetear la contraseña
Como ejecutar Beef en Kali
```

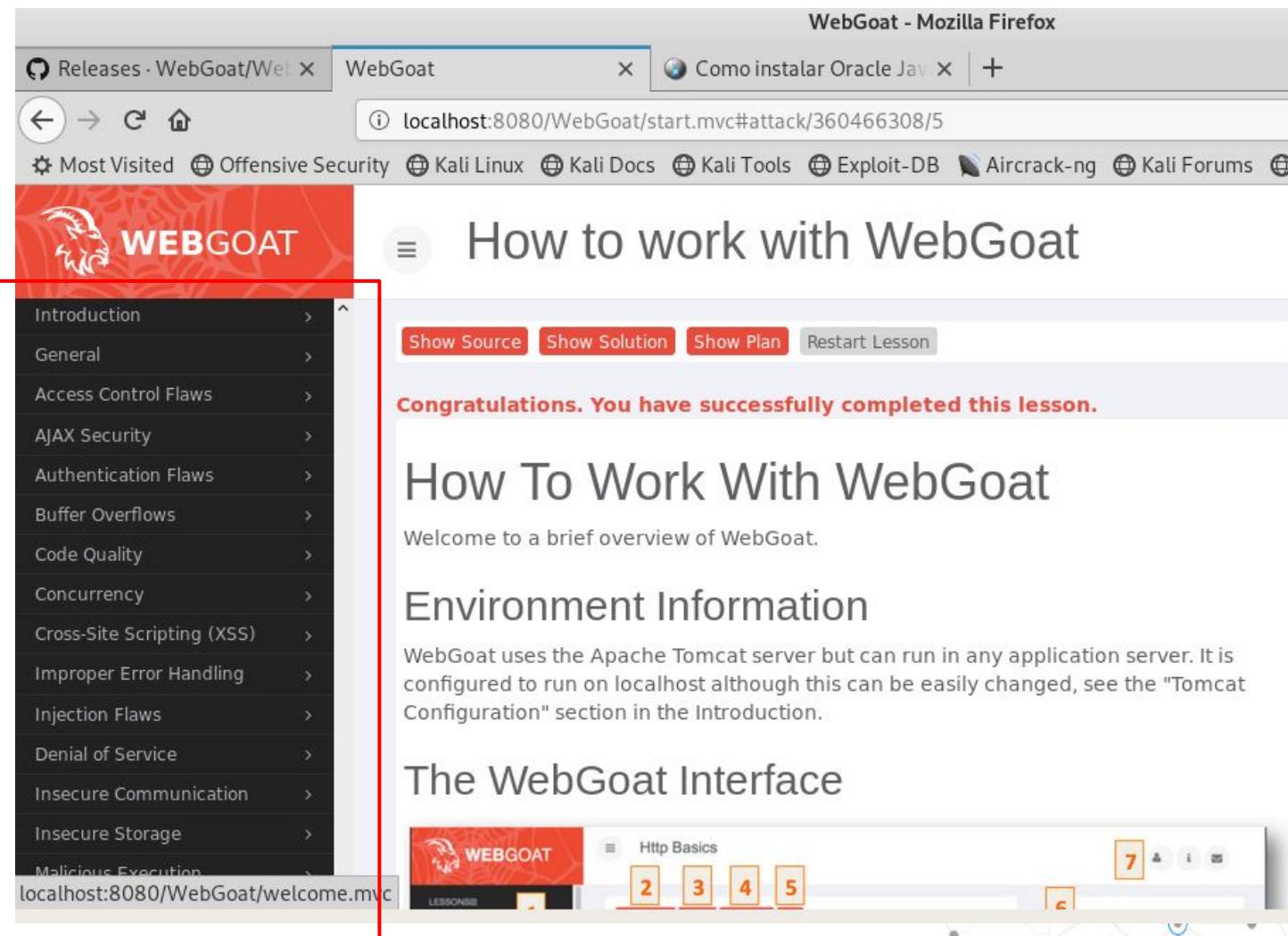


Acceder desde el navegador a la URL <http://localhost:8080/WebGoat>

The screenshot shows a Mozilla Firefox window with the title "Login Page - Mozilla Firefox". The address bar contains the URL "localhost:8080/WebGoat/login.mvc", which is highlighted with a red box. The main content area displays the WebGoat login interface. It features two input fields: "Username" and "Password", both currently empty. Below these fields is a red "Sign in" button. At the bottom of the page, there is a table titled "The following accounts are built into Webgoat". The table has three columns: "Account", "User", and "Password". It lists two accounts: "Webgoat User" with "User" value "guest" and "Password" value "guest", and "Webgoat Admin" with "User" value "webgoat" and "Password" value "webgoat". The "User" and "Password" columns for the "Webgoat User" account are also highlighted with a red box.

Account	User	Password
Webgoat User	guest	guest
Webgoat Admin	webgoat	webgoat

Autenticarse con las credenciales **guest/guest**



The screenshot shows a Mozilla Firefox browser window with three tabs open:

- Releases · WebGoat/WebGoat
- WebGoat (active tab)
- Como instalar Oracle Java

The URL in the address bar is `localhost:8080/WebGoat/start.mvc#attack/360466308/5`.

The main content area displays the "How to work with WebGoat" page. A red box highlights the sidebar menu on the left.

How to work with WebGoat

Introduction
General
Access Control Flaws
AJAX Security
Authentication Flaws
Buffer Overflows
Code Quality
Concurrency
Cross-Site Scripting (XSS)
Improper Error Handling
Injection Flaws
Denial of Service
Insecure Communication
Insecure Storage
Malicious Execution

Show Source Show Solution Show Plan Restart Lesson

Congratulations. You have successfully completed this lesson.

How To Work With WebGoat

Welcome to a brief overview of WebGoat.

Environment Information

WebGoat uses the Apache Tomcat server but can run in any application server. It is configured to run on localhost although this can be easily changed, see the "Tomcat Configuration" section in the Introduction.

The WebGoat Interface

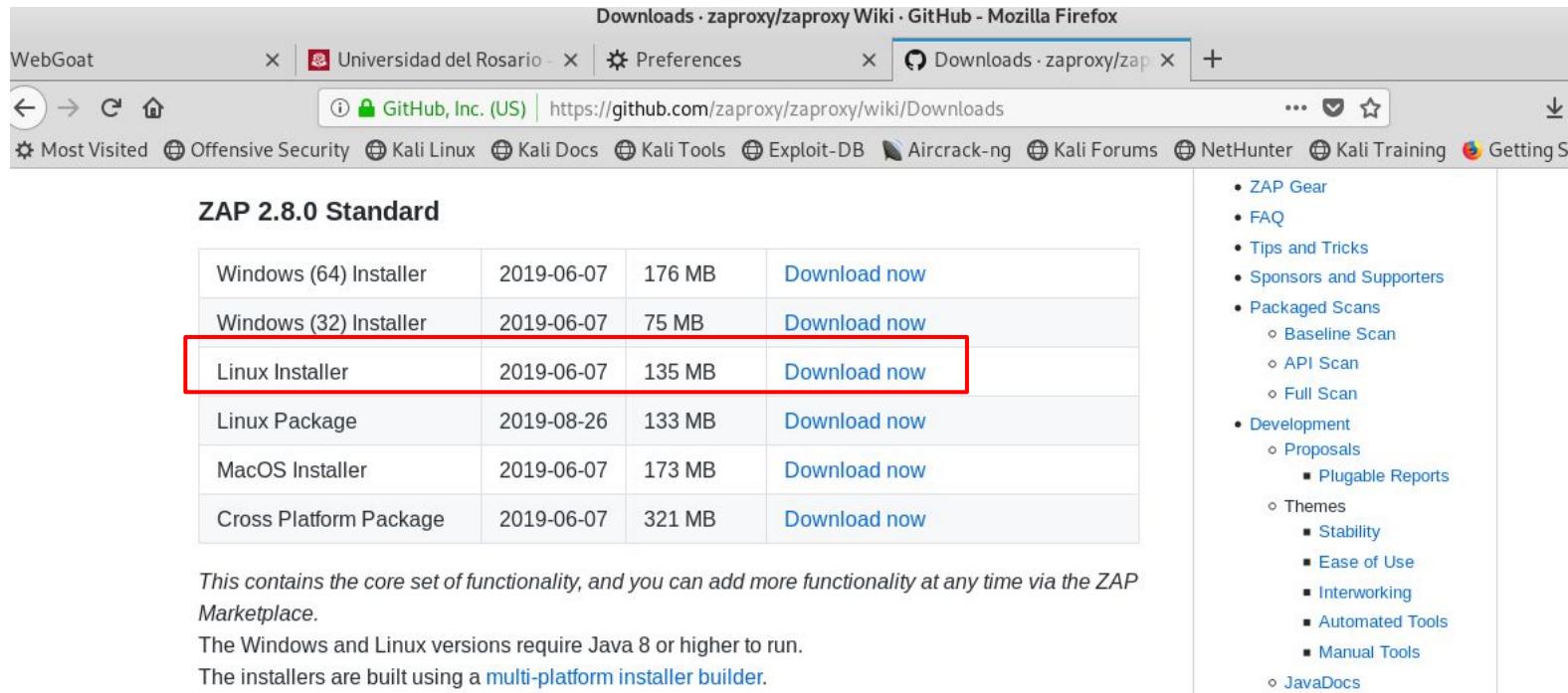
WEBGOAT LESSONSES 2 3 4 5 7

¡Ahora instalaremos nuestro software de
interceptación!

- **Zed Attack Proxy (ZAP): Software de interceptación de comunicaciones**

<https://github.com/zaproxy/zaproxy/wiki/Downloads>

https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project



The screenshot shows a Mozilla Firefox window with several tabs open. The active tab is 'Downloads · zaproxy/zaproxy Wiki · GitHub - Mozilla Firefox'. The address bar shows the URL: 'https://github.com/zaproxy/zaproxy/wiki/Downloads'. The main content area displays a table of ZAP 2.8.0 Standard download options:

	Date	Size	Action
Windows (64) Installer	2019-06-07	176 MB	Download now
Windows (32) Installer	2019-06-07	75 MB	Download now
Linux Installer	2019-06-07	135 MB	Download now
Linux Package	2019-08-26	133 MB	Download now
MacOS Installer	2019-06-07	173 MB	Download now
Cross Platform Package	2019-06-07	321 MB	Download now

This contains the core set of functionality, and you can add more functionality at any time via the ZAP Marketplace.

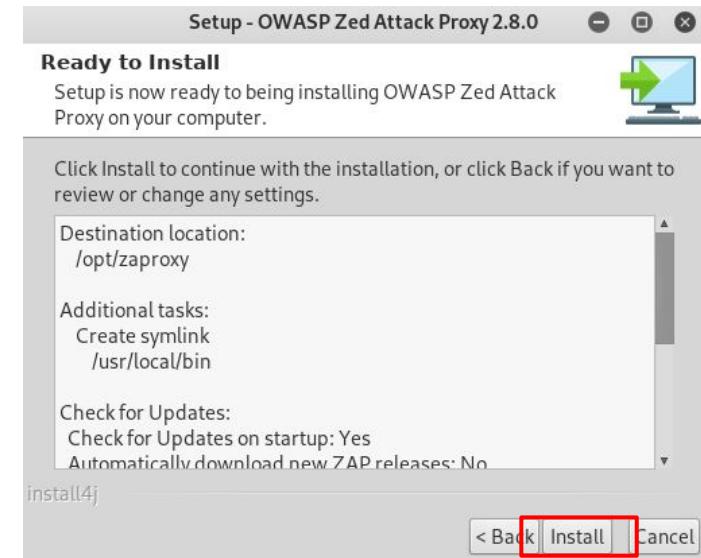
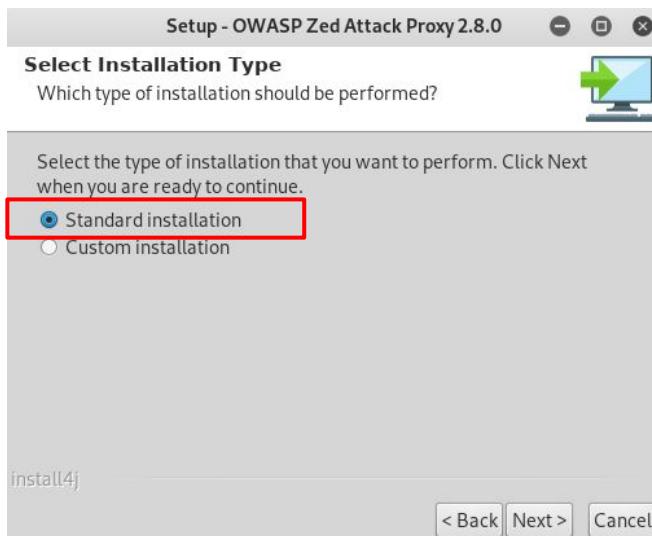
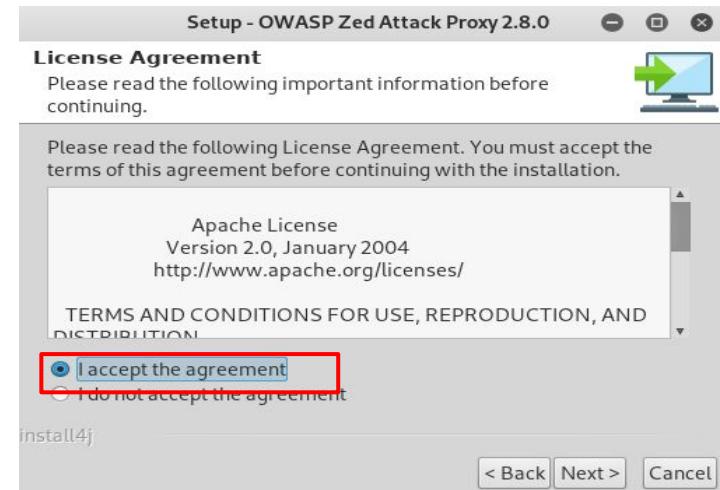
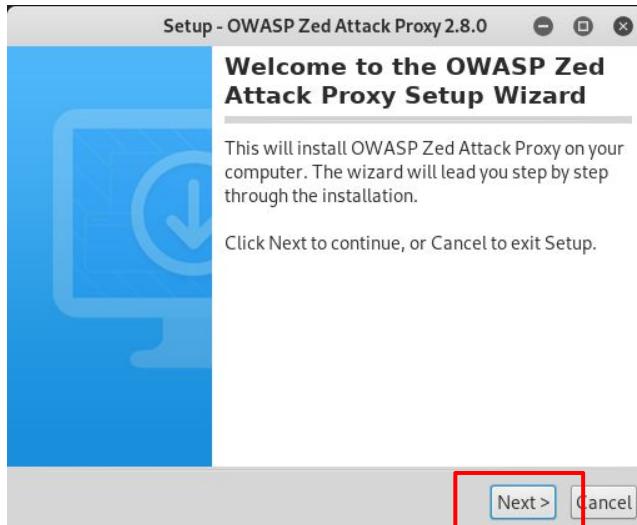
The Windows and Linux versions require Java 8 or higher to run.

The installers are built using a [multi-platform installer builder](#).

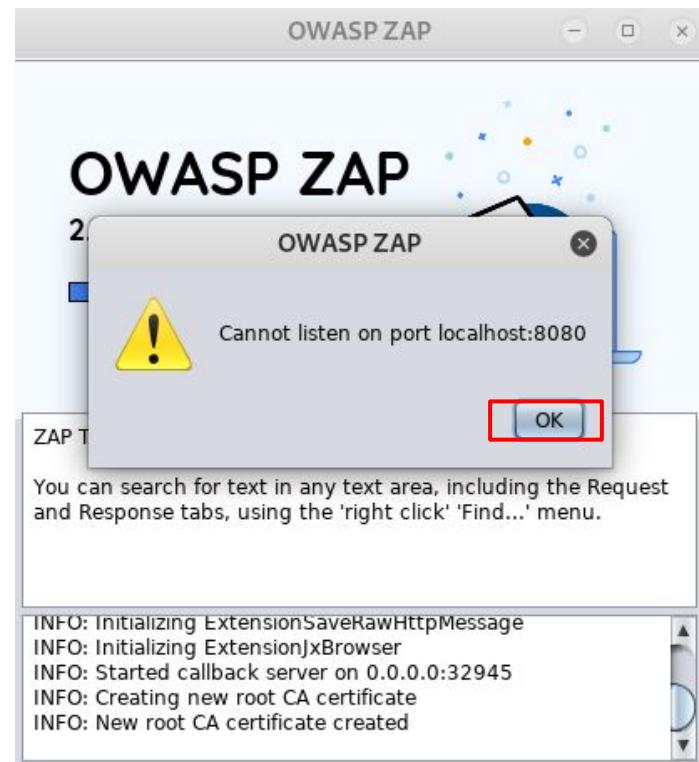
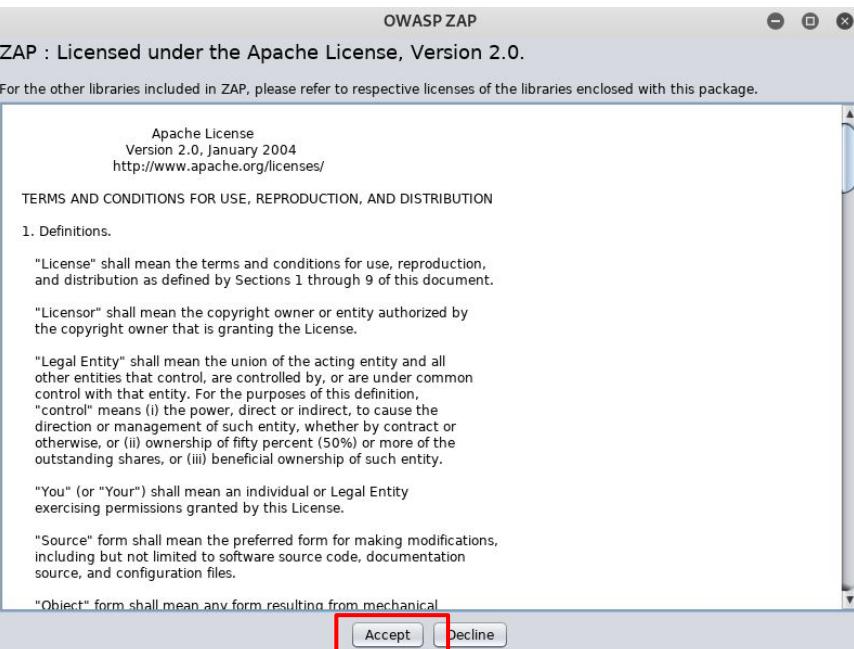
On the right side of the page, there is a sidebar with links to various ZAP resources:

- ZAP Gear
- FAQ
- Tips and Tricks
- Sponsors and Supporters
- Packaged Scans
 - Baseline Scan
 - API Scan
 - Full Scan
- Development
 - Proposals
 - Plugable Reports
 - Themes
 - Stability
 - Ease of Use
 - Interworking
 - Automated Tools
 - Manual Tools
 - JavaDocs

```
root@kali:~/Downloads# ls
UserDatabase.mv.db
webgoat-container-7.1-exec.jar  ZAPGettingStartedGuide-2.8.pdf
webgoat-container-7.0.1-war-exec.jar ZAP_2_8_0_unix.sh
root@kali:~/Downloads# sh ZAP_2_8_0_unix.sh
```



```
root@kali:~/Downloads# owasp-zap
[...]
Available memory: 1996 MB
Setting jvm heap size: -Xmx499m
640 [main] INFO org.zaproxy.zap.GuiBootstrap - OWASP ZAP 2.7.0 started 20/10/19 23:25:58 with home /root/.ZAP/
```



OWASP ZAP - OWASP ZAP 2.7.0

File Edit View Analyse Report Tools Online Help

Standard Mode

Sites + Contexts

Quick Start Request Response +

Welcome to the OWASP Zed Attack

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

OWASP ZAP

Do you want to persist the ZAP Session?

Yes, I want to persist this session with name based on the current timestamp

Yes, I want to persist this session but I want to specify the name and location

No, I do not want to persist this session at this moment in time

Remember my choice and do not ask me again.

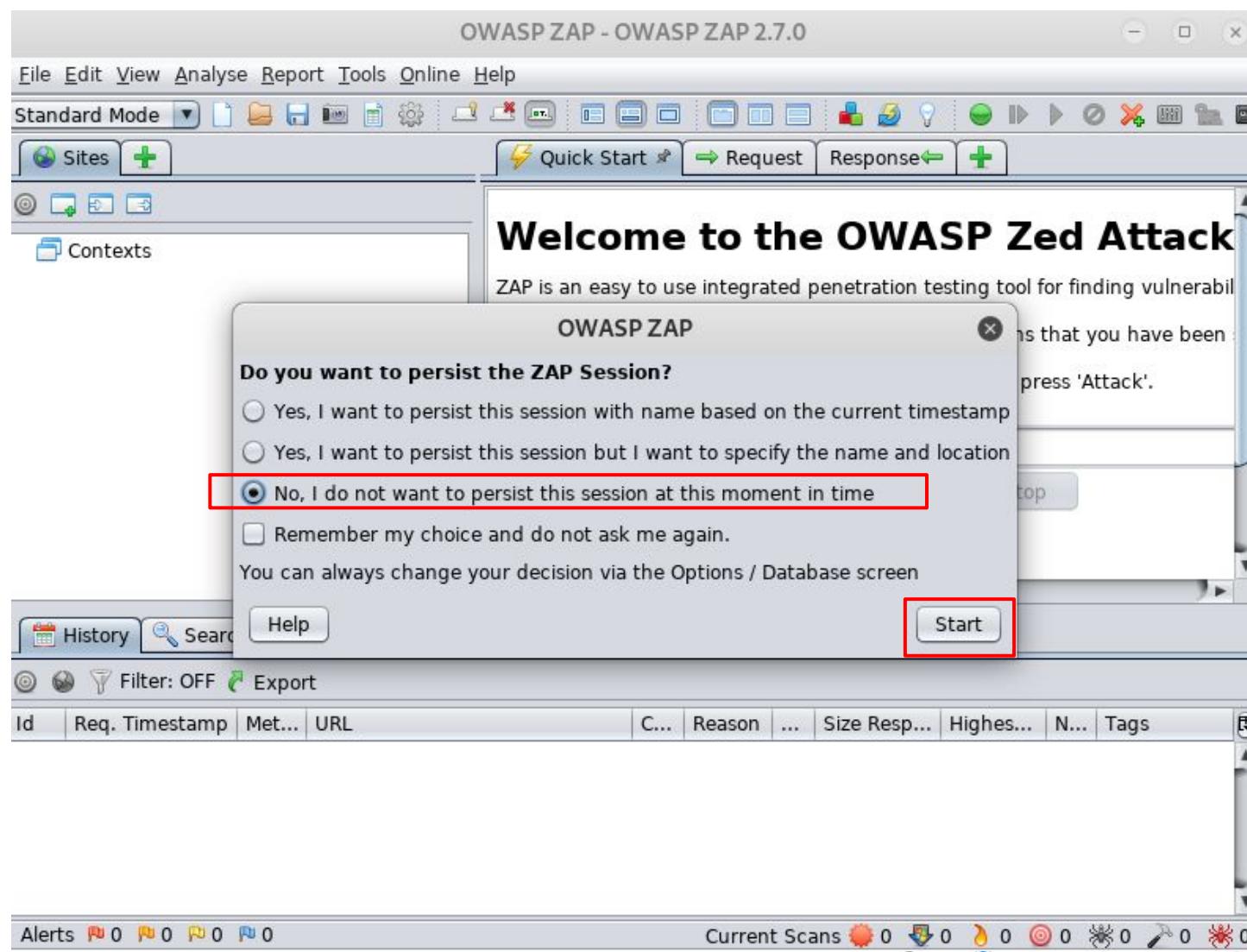
You can always change your decision via the Options / Database screen

Start

History Search Filter: OFF Export

Id	Req. Timestamp	Met...	URL	C...	Reason	...	Size Respon...	Highest...	N...	Tags
----	----------------	--------	-----	------	--------	-----	----------------	------------	------	------

Alerts 0 0 0 0 0 Current Scans 0 0 0 0 0 0 0 0 0 0



Linux-Victima [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Applications ▾ Places ▾ OWASP ZAP ▾ Sun 23:31

Untitled Session - OWASP ZAP 2.7.0

File Edit View Analyse Report Tools Online Help

Standard Mode ▾

Sites +

Quick Start Request Response +

Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http://localhost:8080/WebGoat Select...

Attack Stop

Progress: Attack complete - see the Alerts tab for details of any issues found

For a more in depth test you should explore your application using your browser or automated regression tests while proxying through ZAP.

History Search Alerts Output Spider Active Scan +

Alerts (4)

- X-Frame-Options Header Not Set (3)
- Password Autocomplete in Browser (3)
- Web Browser XSS Protection Not Enabled (4)
- X-Content-Type-Options Header Missing (7)

X-Frame-Options Header Not Set

URL: http://localhost:8080/WebGoat

Risk: Medium

Confidence: Medium

Parameter: X-Frame-Options

Attack:

Evidence:

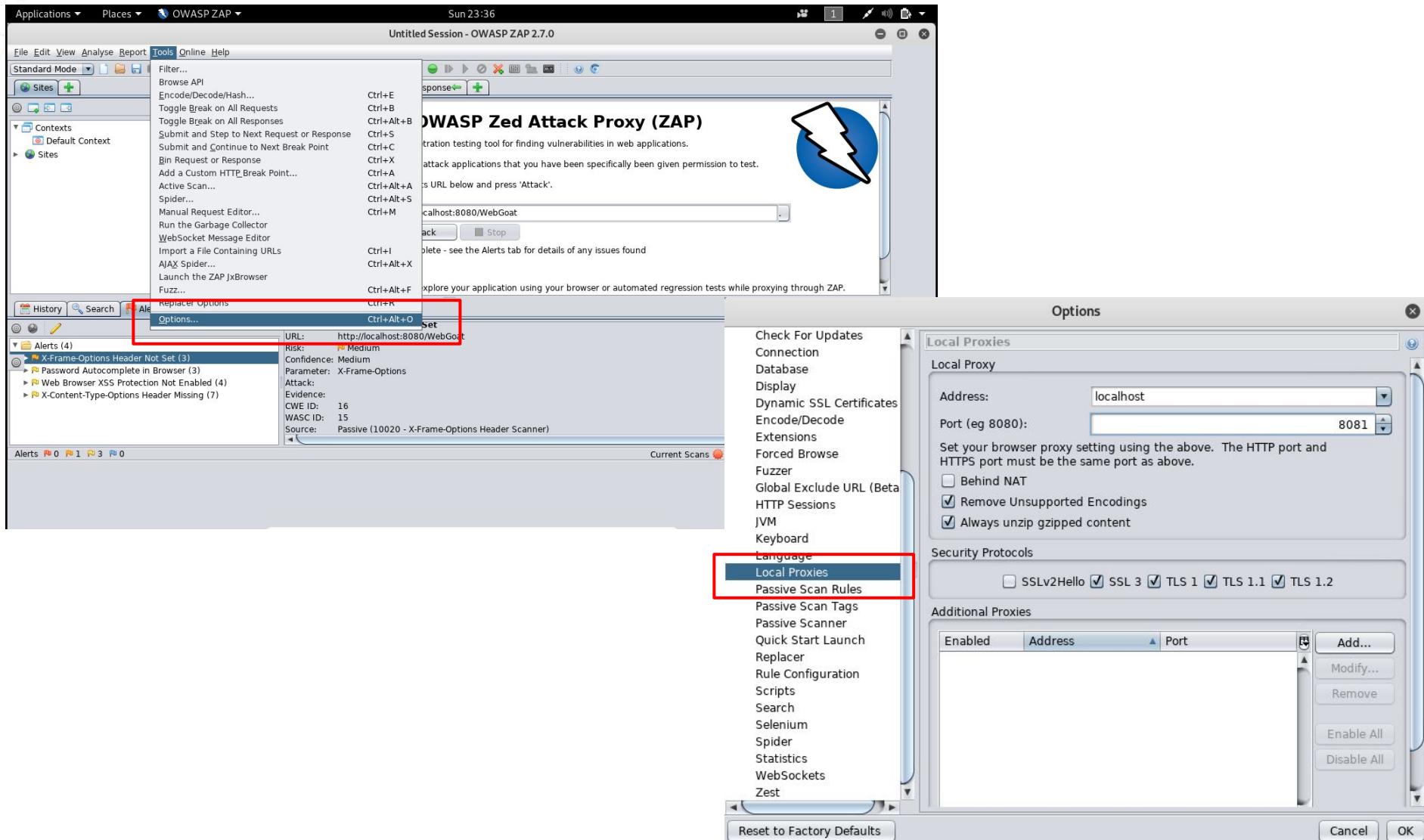
CWE ID: 16

WASC ID: 15

Source: Passive (10020 - X-Frame-Options Header Scanner)

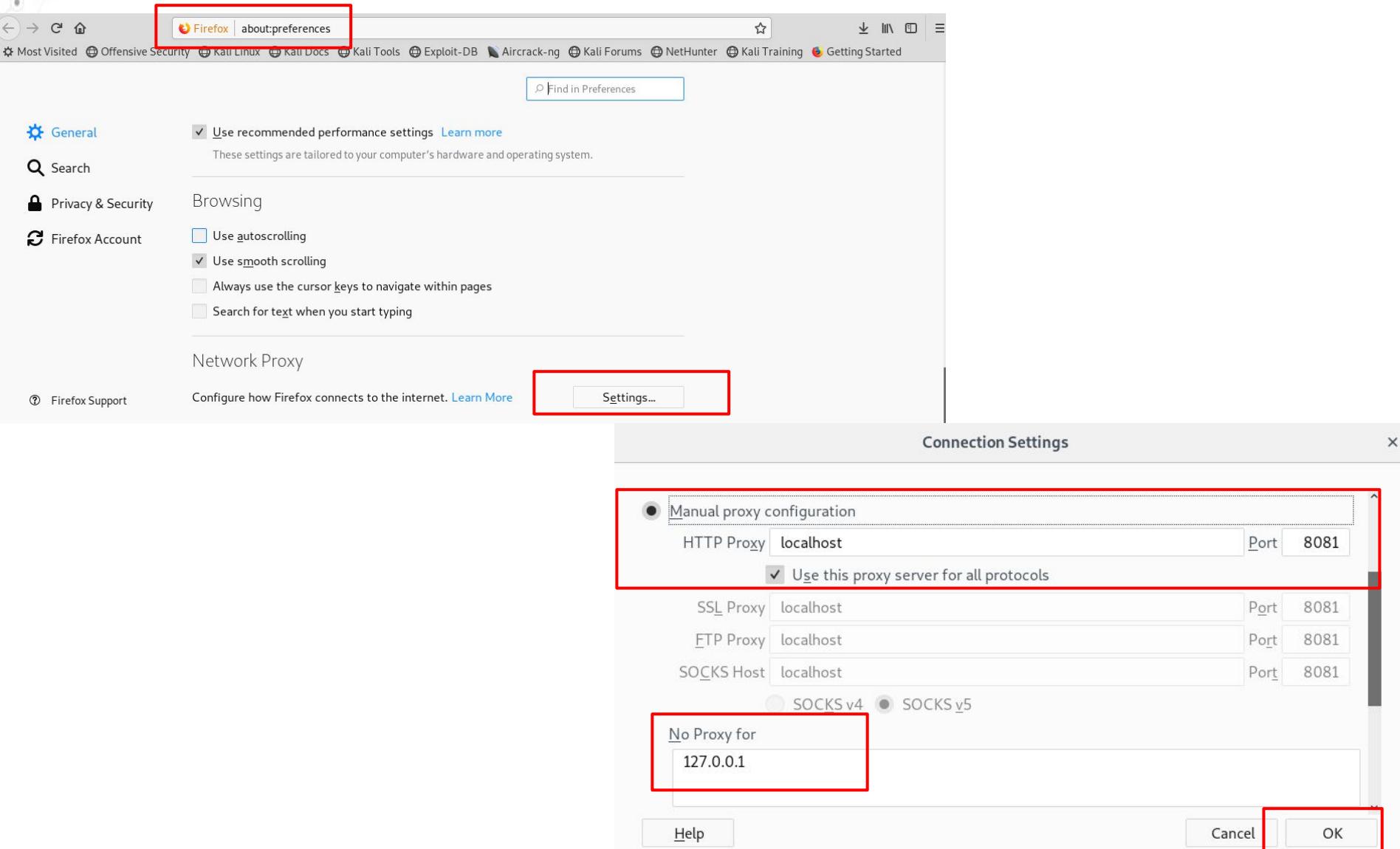
Alerts 0 1 3 0 Current Scans 0 0 0 0 0 0 0 Right Ctrl

Para configurar ZAP como proxy, vamos a Tools -> Options -> Local Proxies



The screenshot shows the OWASP ZAP 2.7.0 interface. The 'Tools' menu is open, with 'Options...' highlighted. A red box highlights the 'Options...' menu item. In the bottom right corner, the 'Options' dialog is open, showing the 'Local Proxies' tab selected. Another red box highlights the 'Local Proxies' tab in the options dialog. The 'Address:' field is set to 'localhost'. The 'Port (eg 8080):' field has '8081' typed into it. There are three checked checkboxes below: 'Behind NAT', 'Remove Unsupported Encodings', and 'Always unzip gzipped content'. The 'Security Protocols' section includes checkboxes for SSLv2Hello, SSL 3, TLS 1, TLS 1.1, and TLS 1.2, with most checked except for SSLv2Hello.

Ahora configuremos que todo el tráfico web vaya primero hacia ZAP

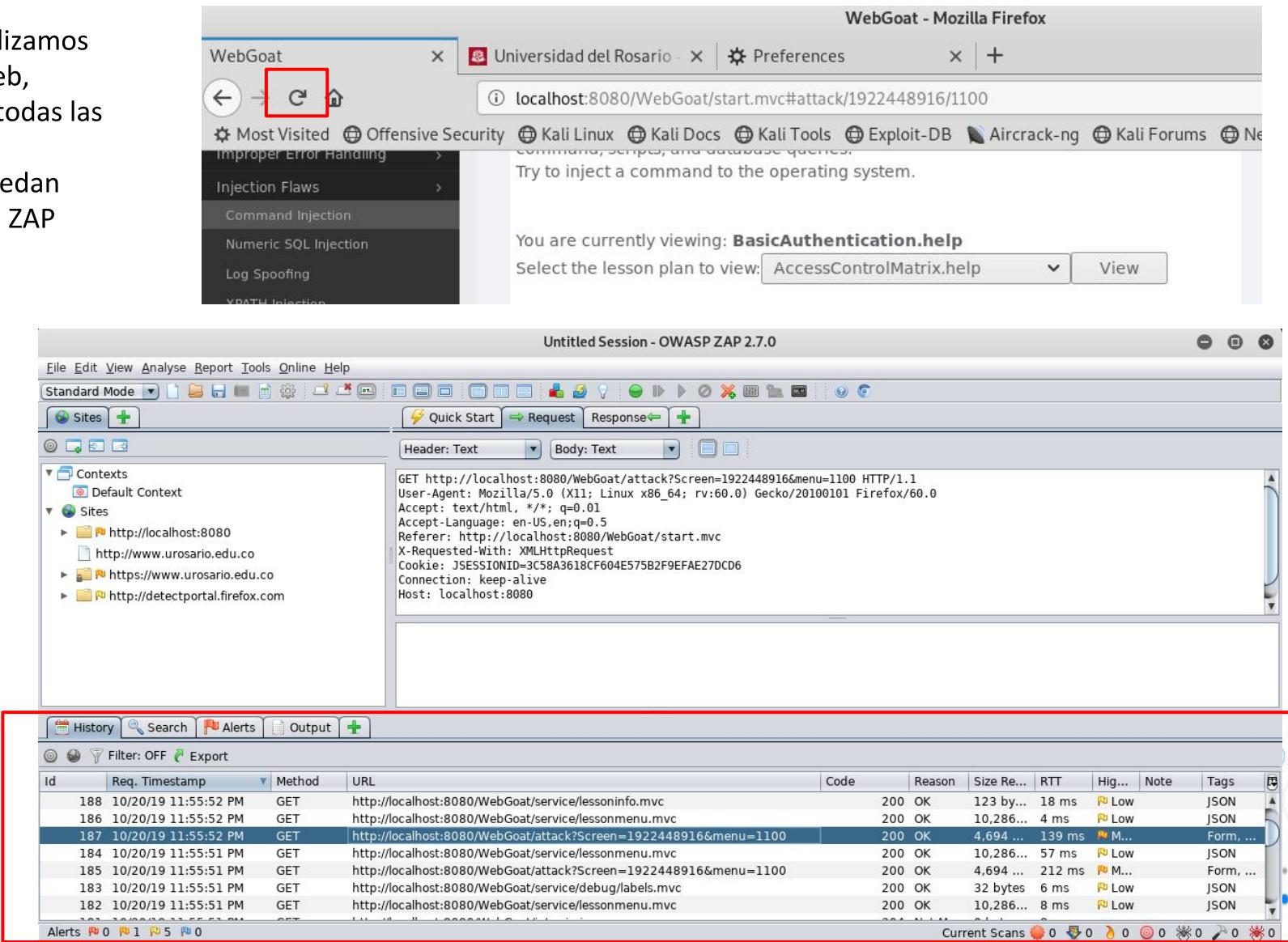


The screenshot shows the Firefox preferences window with several sections highlighted by red boxes:

- General** section: Shows a checked checkbox for "Use recommended performance settings".
Browsing section: Shows checkboxes for "Use autoscrolling" (unchecked), "Use smooth scrolling" (checked), "Always use the cursor keys to navigate within pages" (unchecked), and "Search for text when you start typing" (unchecked).
- Network Proxy** section: Shows a "Settings..." button which is also highlighted by a red box.
- Connection Settings** dialog box:
 - Manual proxy configuration** radio button is selected (highlighted by a red box).
 - HTTP Proxy**: Hostname "localhost" and Port "8081". A checked checkbox "Use this proxy server for all protocols" is also highlighted by a red box.
 - SSL Proxy**: Hostname "localhost" and Port "8081".
 - FTP Proxy**: Hostname "localhost" and Port "8081".
 - SOCKS Host**: Hostname "localhost" and Port "8081".
 - No Proxy for** field contains the IP address "127.0.0.1".
 - Buttons**: "Help", "Cancel", and "OK" buttons at the bottom right.

Probemos las capacidades de interceptación de ZAP

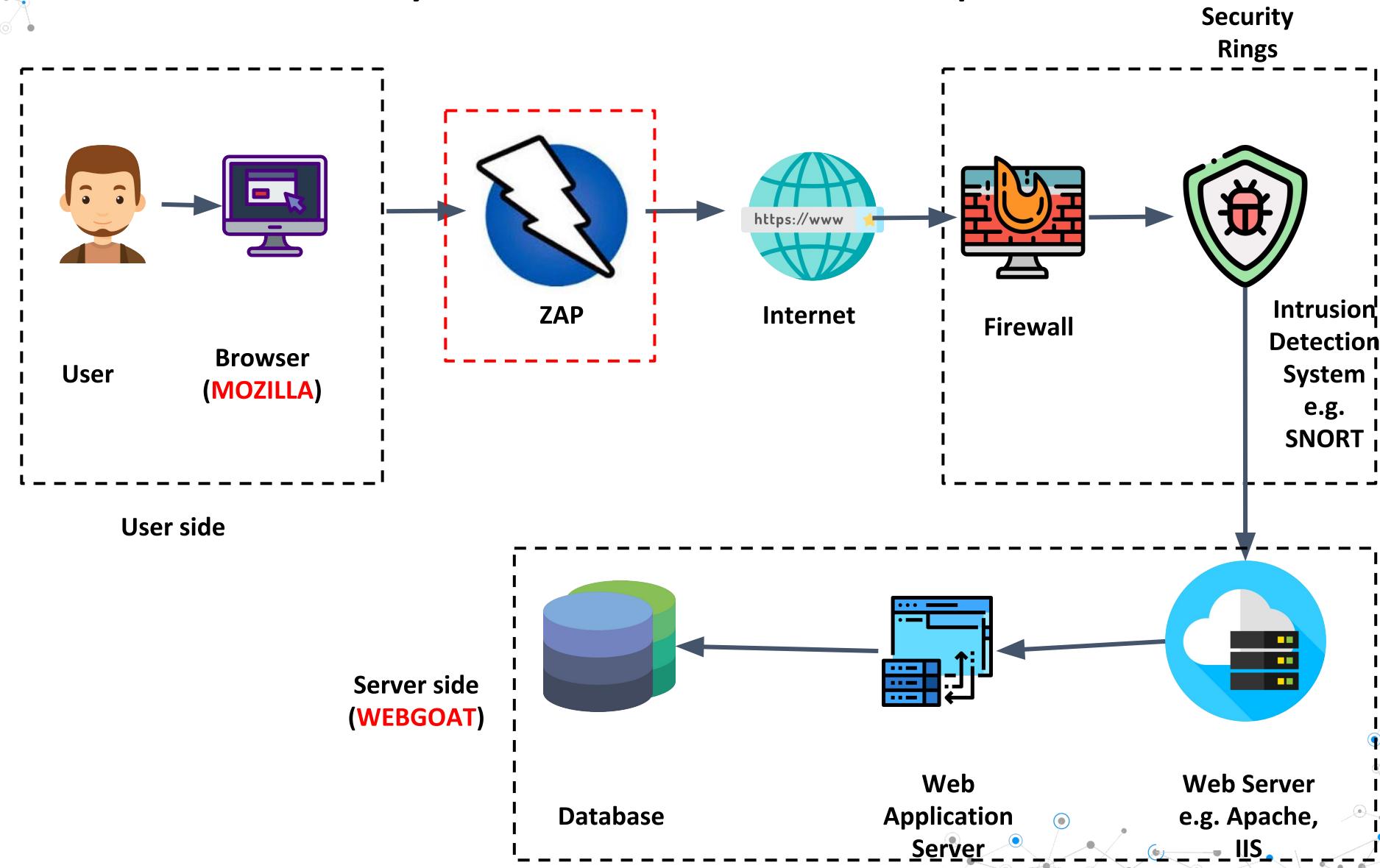
Cuando actualizamos una página web, veremos que todas las peticiones y respuestas quedan registradas en ZAP

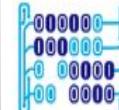


The screenshot shows the OWASP ZAP 2.7.0 interface. At the top, a Firefox browser window displays a WebGoat challenge titled "BasicAuthentication.help". A red box highlights the browser's address bar. Below it, the ZAP interface has a "Standard Mode" toolbar and a sidebar showing contexts and sites. A red box highlights the "Header: Text" and "Body: Text" panes, which display the raw HTTP request and response for the current session. At the bottom, a large red box highlights the "History" tab, where a list of intercepted transactions is shown. The table includes columns for Id, Req. Timestamp, Method, URL, Code, Reason, Size Re..., RTT, Hig..., Note, and Tags. The last transaction in the list is highlighted with a red box.

Id	Req. Timestamp	Method	URL	Code	Reason	Size Re...	RTT	Hig...	Note	Tags
188	10/20/19 11:55:52 PM	GET	http://localhost:8080/WebGoat/service/lessoninfo.mvc	200	OK	123 by...	18 ms	Low		JSON
186	10/20/19 11:55:52 PM	GET	http://localhost:8080/WebGoat/service/lessonmenu.mvc	200	OK	10,286...	4 ms	Low		JSON
187	10/20/19 11:55:52 PM	GET	http://localhost:8080/WebGoat/attack?Screen=1922448916&menu=1100	200	OK	4,694 ...	139 ms	Medium		Form, ...
184	10/20/19 11:55:51 PM	GET	http://localhost:8080/WebGoat/service/lessonmenu.mvc	200	OK	10,286...	57 ms	Low		JSON
185	10/20/19 11:55:51 PM	GET	http://localhost:8080/WebGoat/attack?Screen=1922448916&menu=1100	200	OK	4,694 ...	212 ms	Medium		Form, ...
183	10/20/19 11:55:51 PM	GET	http://localhost:8080/WebGoat/service/debug/labels.mvc	200	OK	32 bytes	6 ms	Low		JSON
182	10/20/19 11:55:51 PM	GET	http://localhost:8080/WebGoat/service/lessonmenu.mvc	200	OK	10,286...	8 ms	Low		JSON

Ahora ya tenemos nuestro ambiente de prueba





- The ten most critical web application security risks

A1 - Injection – Inyección de código

A2 – Broken authentication and session management - Pérdida de Autenticación y Gestión de Sesiones

A3 - Cross-site scripting (XSS) - Secuencia de Comandos en Sitios Cruzados

A4 – Insecure direct object references - Referencia Directa Insegura a Objetos

A5 – Security misconfiguration - Configuración de seguridad Incorrecta

A6 – Sensitive data exposure - Exposición de Datos Sensibles

A7 – Missing function level Access control - Ausencia de Control de Acceso a Funciones

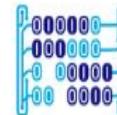
A8 – Cross-Site Request Forgery (CSRF) - Falsificación de Peticiones en Sitios Cruzados

A9 – Components with known vulnerabilities - Uso de Componentes con Vulnerabilidades conocidas

A10- Unvalidated redirects and forwards - Redirecciones y reenvíos no validados



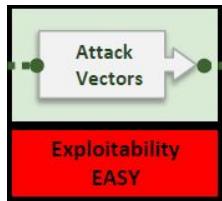
Universidad del
Rosario



MACC
Matemáticas Aplicadas y
Ciencias de la Computación

A1- Inyección de Código

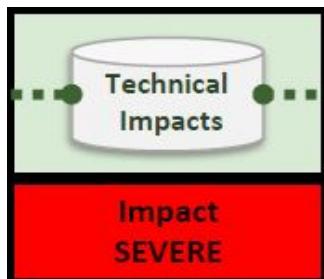
Inyección de Código



- **How?** The intruder sends text-based attacks that exploit the syntax of the application interpreter. Almost any data source (including internal data) can be an attack vector.



- **Why?** It occurs when the data provided by the user are not validated before being processed by the interpreter. Injection can exist in SQL code, LDAP, XPATH, OS Commands, XML parsers, SMTP headers, program arguments, etc.



- **Impact:** Loss of data, modification of data, execution of activities outside of logs, denial of service

OS COMMAND INJECTION

- Same problem: a command sequence is built using **user input** which is NOT sanitized or validated
 - Imagine what would happen if the **OS application account** is not restricted
 - **Two types:**
 1. The software has a field which is the argument for a single command
 2. The software accept an argument which is converted to a command
- Flaw**
- The programmer assumes that the input is always trustworthy
- The programmer assumes that the command will never be used

;/bin/ls -l”



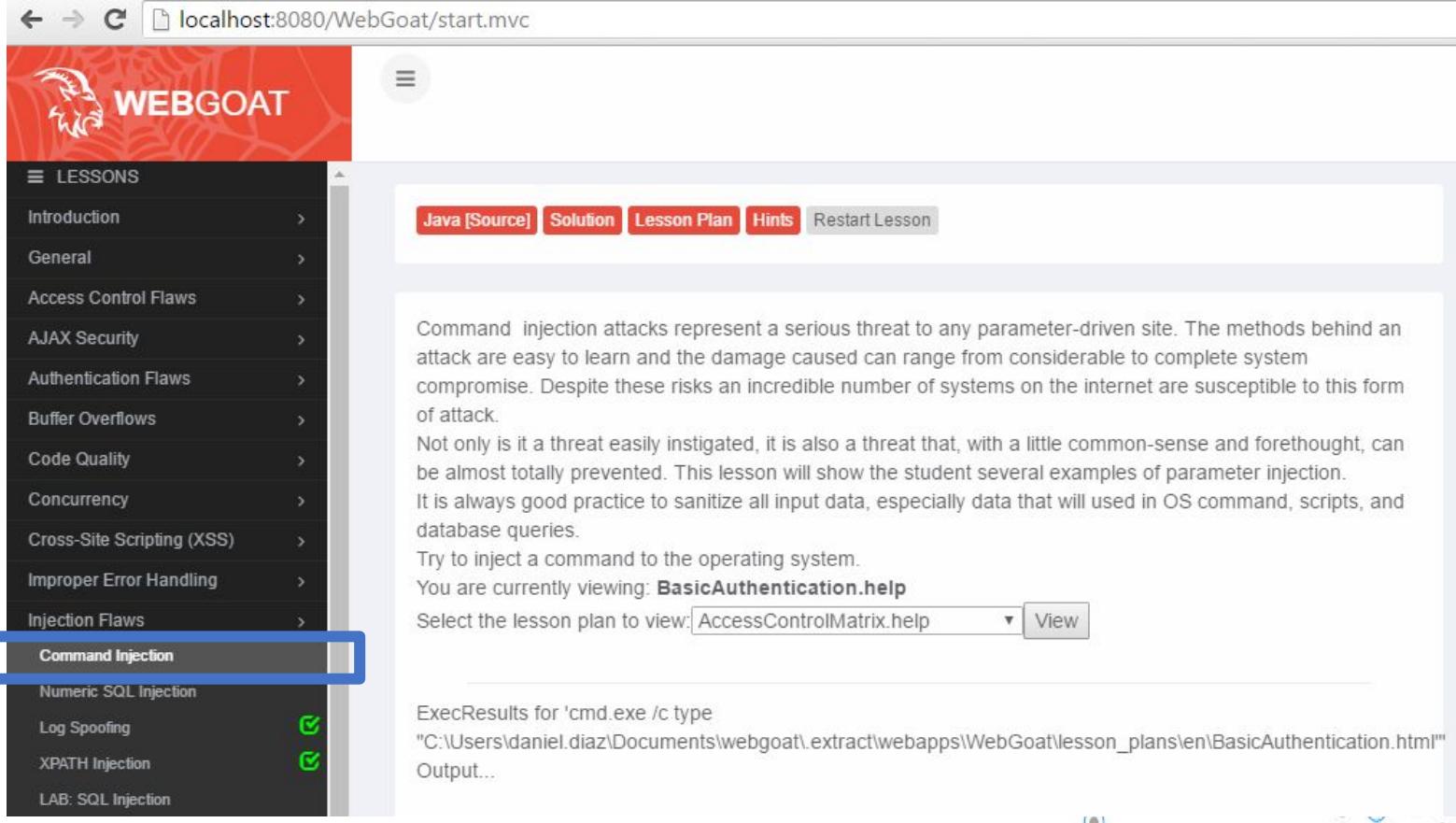
;	%3B
Espace []	%20
“	%22

The command must be codified to URL format

<http://www.mycompany.com/sensitive/cgi-bin/userData.pl?doc=%20%3B%20/bin/ls%20-l%22>

https://www.owasp.org/index.php/Command_Injection

- Example Command injection in WebGoat



The screenshot shows a web browser displaying the WebGoat application at `localhost:8080/WebGoat/start.mvc`. The interface has a red header with the WebGoat logo. On the left, there's a sidebar with a navigation menu under 'LESSONS'. The 'Injection Flaws' section is expanded, and 'Command Injection' is highlighted with a blue box. Other items in this section include 'Numeric SQL Injection', 'Log Spoofing', 'XPATH Injection', and 'LAB: SQL Injection'. To the right of the sidebar, there's a main content area with a toolbar at the top containing 'Java [Source]', 'Solution', 'Lesson Plan', 'Hints', and 'Restart Lesson'. Below the toolbar, the text reads:

Command injection attacks represent a serious threat to any parameter-driven site. The methods behind an attack are easy to learn and the damage caused can range from considerable to complete system compromise. Despite these risks an incredible number of systems on the internet are susceptible to this form of attack.

Not only is it a threat easily instigated, it is also a threat that, with a little common-sense and forethought, can be almost totally prevented. This lesson will show the student several examples of parameter injection. It is always good practice to sanitize all input data, especially data that will be used in OS command, scripts, and database queries.

Try to inject a command to the operating system.

You are currently viewing: **BasicAuthentication.help**

Select the lesson plan to view: View

At the bottom, there's a 'ExecResults' section showing the output of a command injection attempt:

```
ExecResults for 'cmd.exe /c type  
"C:\Users\daniel.diaz\Documents\webgoat\extract\webapps\WebGoat\lesson_plans\en\BasicAuthentication.html"  
Output...'
```

- Example Command injection in WebGoat

You are currently viewing: **BackDoors.help**

Select the lesson plan to view: **BackDoors.help**

This is the normal behavior: I select a file (lesson) and the web server return it

```
ExecResults for '[/bin/sh, -c, cat "/root/Downloads/.extract/webapps/WebGoat/Output...'
```

Lesson Plan Title: How to Create Database Back Door Attacks.

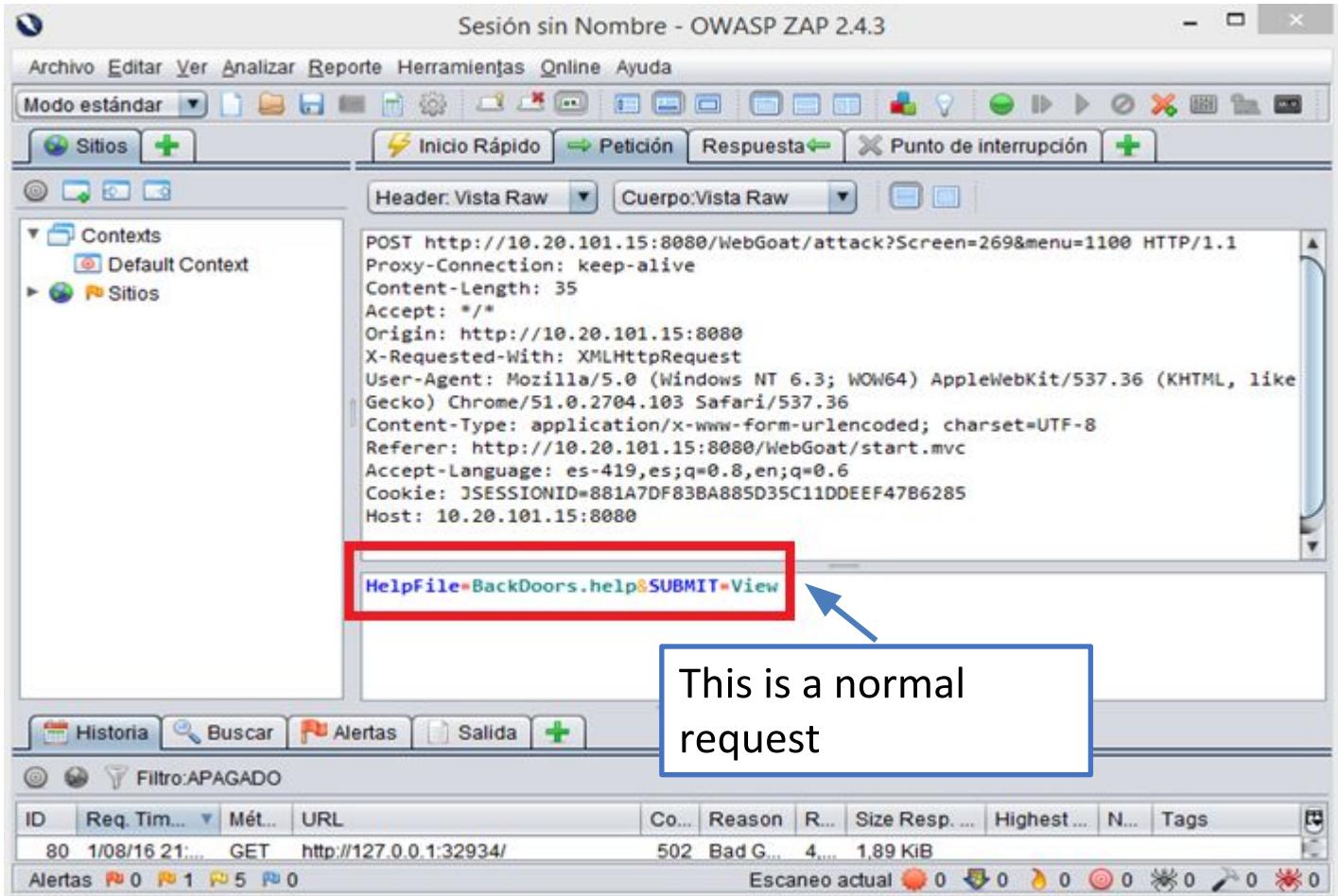
Concept / Topic To Teach:

How to Create Database Back Door Attacks.

How the attacks works:

Databases are used usually as a backend for web applications. Also it is used be used as a place to store a malicious activity such as a trigger. A trigger system upon the execution of another database operation like insert, select, can create a trigger that would set his email address instead of every new user.

- Example Command injection in WebGoat



The screenshot shows the OWASP ZAP 2.4.3 interface. In the Request pane, a POST request is being constructed to the URL `http://10.20.101.15:8080/WebGoat/attack?Screen=269&menu=1100`. The Headers tab shows standard HTTP headers. The Cuerpo tab contains the following raw payload:

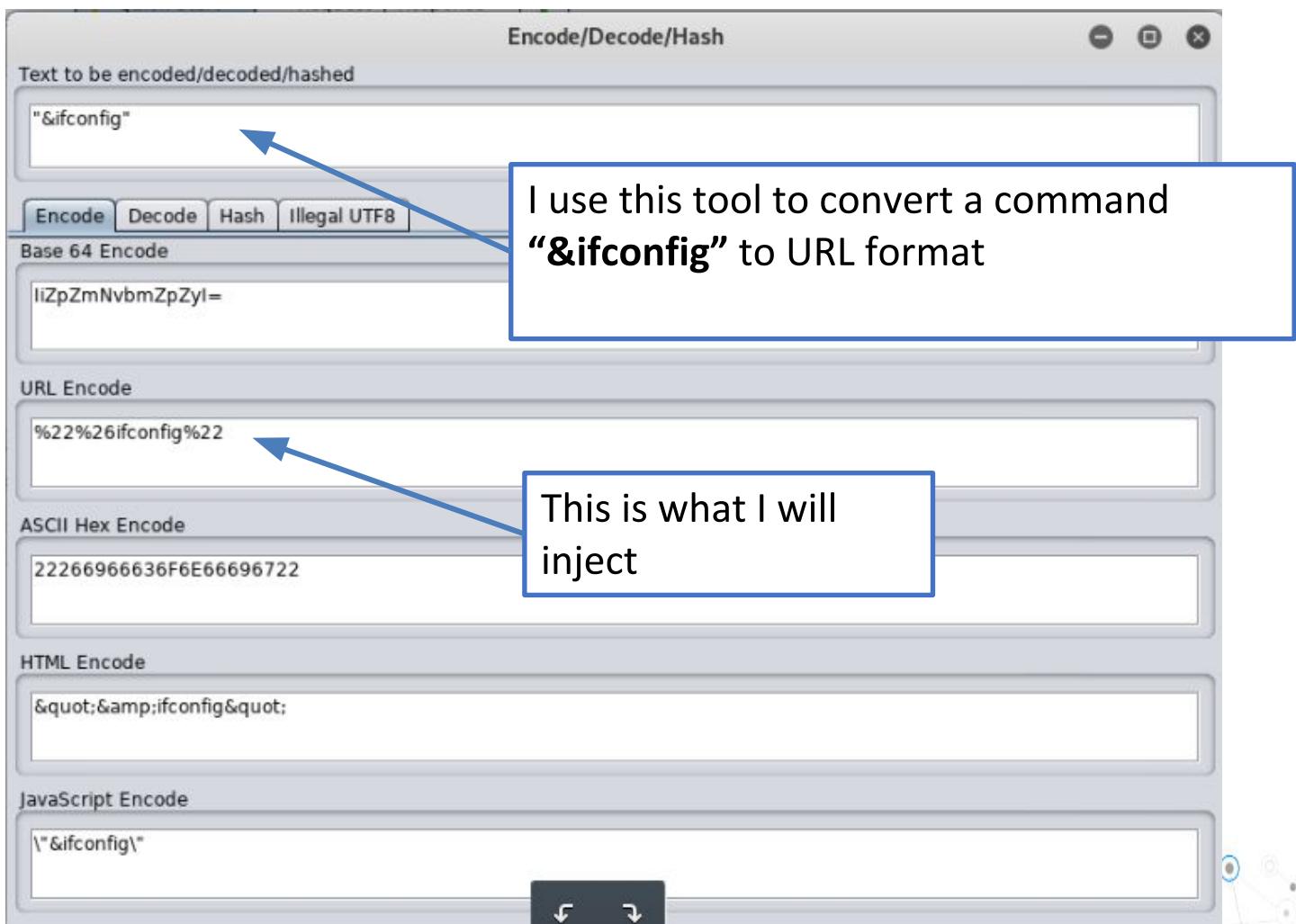
```
POST http://10.20.101.15:8080/WebGoat/attack?Screen=269&menu=1100 HTTP/1.1
Proxy-Connection: keep-alive
Content-Length: 35
Accept: /*
Origin: http://10.20.101.15:8080
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Referer: http://10.20.101.15:8080/WebGoat/start.mvc
Accept-Language: es-419,es;q=0.8,en;q=0.6
Cookie: JSESSIONID=881A7DF83BA885D35C11DDEEF47B6285
Host: 10.20.101.15:8080

HelpFile=BackDoors.help&SUBMIT=View
```

A red box highlights the payload `HelpFile=BackDoors.help&SUBMIT=View`, which includes a command injection payload (`&SUBMIT=View`). A blue callout box points to this payload with the text "This is a normal request".

Inyección de Código

- Example Command injection in WebGoat



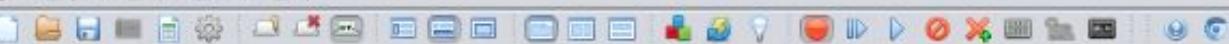
The screenshot shows a window titled "Encode/Decode/Hash". In the top input field, the text "&ifconfig" is entered. Below it, under "Base 64 Encode", the output is "iZpZmNvbmZpZyl=". A blue arrow points from this output to a callout box containing the text: "I use this tool to convert a command "&ifconfig" to URL format". In the "URL Encode" section, the output is "%22%26ifconfig%22", with another blue arrow pointing from this to a callout box containing the text: "This is what I will inject". The other sections (ASCII Hex Encode, HTML Encode, JavaScript Encode) show their respective outputs for the input "&ifconfig".

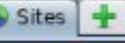
Inyección de Código

- Example Command injection in WebGoat

Untitled Session - OWASP ZAP 2.7.0

File Edit View Analyse Report Tools Online Help

Standard Mode 

Sites  Quick Start Request Response Break 

Method Header: Text Body: Text

Contexts Default Context
Sites http://localhost:8080

POST http://localhost:8080/WebGoat/attack?Screen=1922448916&menu=1100 HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5

HelpFile=BackDoors.help%22%26ifconfig%22&SUBMIT=View

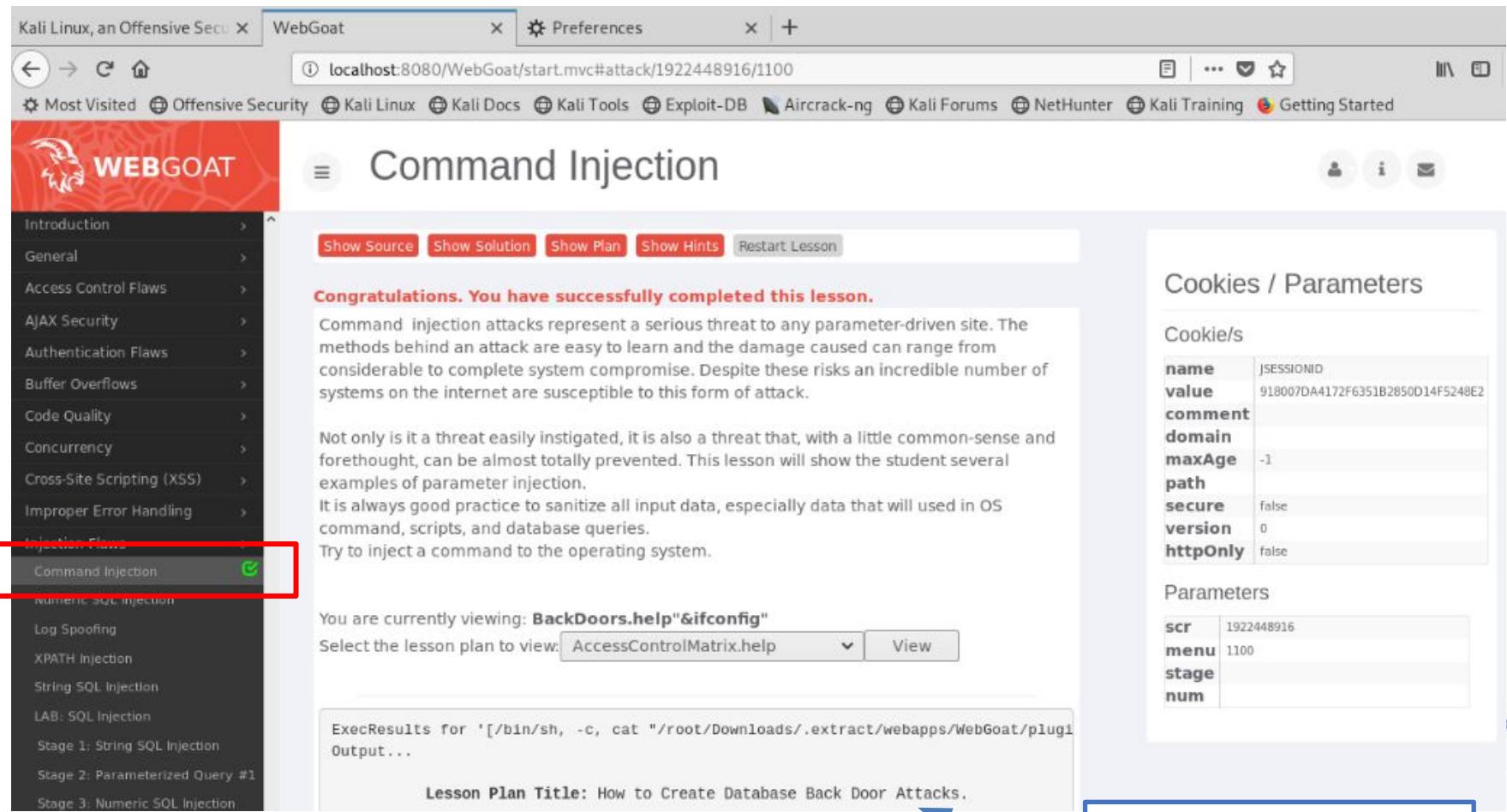
History Search Alerts Output 

Filter: OFF Export

Here I am injecting the command

ID	Req. Timestamp	Method	URL	Code	Reason	RTT	Size	Resp. Body
1	10/26/19 11:48:46 AM	POST	http://localhost:8080/WebGoat/attack?Screen=... HelpFile=BackDoors.help%22%26ifconfig%22&SUBMIT=View	200	OK	397 ms	4,792 bytes	
4	10/26/19 11:48:46 AM	GET	http://localhost:8080/WebGoat/service/lessoninf...	200	OK	53 ms	123 bytes	
6	10/26/19 11:48:46 AM	GET	http://localhost:8080/WebGoat/service/cookie.m...	200	OK	9 ms	163 bytes	
7	10/26/19 11:48:46 AM	GET	http://localhost:8080/WebGoat/service/hint.mvc	200	OK	50 ms	540 bytes	
8	10/26/19 11:48:46 AM	GET	http://localhost:8080/WebGoat/service/lessonpl...	200	OK	96 ms	1,037 bytes	
9	10/26/19 11:48:47 AM	GET	http://localhost:8080/WebGoat/service/lessonpr...	200	OK	7 ms	106 bytes	
10	10/26/19 11:48:46 AM	GET	http://localhost:8080/WebGoat/service/solution....	200	OK	69 ms	34,343 bytes	
11	10/26/19 11:48:46 AM	GET	http://localhost:8080/WebGoat/service/source....	200	OK	72 ms	11,175 bytes	
12	10/26/19 11:48:47 AM	GET	http://localhost:8080/WebGoat/service/lessonm...	200	OK	8 ms	10,934 bytes	

- Example Command injection in WebGoat



Kali Linux, an Offensive Secu × WebGoat × Preferences × | +

localhost:8080/WebGoat/start.mvc#attack/1922448916/1100

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training Getting Started

WEBGOAT

Introduction General Access Control Flaws AJAX Security Authentication Flaws Buffer Overflows Code Quality Concurrency Cross-Site Scripting (XSS) Improper Error Handling Injection Flaws **Command Injection** Numeric SQL Injection Log Spoofing XPATH Injection String SQL Injection LAB: SQL Injection Stage 1: String SQL Injection Stage 2: Parameterized Query #1 Stage 3: Numeric SQL Injection

Command Injection

Show Source Show Solution Show Plan Show Hints Restart Lesson

Congratulations. You have successfully completed this lesson.

Command injection attacks represent a serious threat to any parameter-driven site. The methods behind an attack are easy to learn and the damage caused can range from considerable to complete system compromise. Despite these risks an incredible number of systems on the internet are susceptible to this form of attack.

Not only is it a threat easily instigated, it is also a threat that, with a little common-sense and forethought, can be almost totally prevented. This lesson will show the student several examples of parameter injection.

It is always good practice to sanitize all input data, especially data that will be used in OS command, scripts, and database queries.

Try to inject a command to the operating system.

You are currently viewing: **BackDoors.help"&ifconfig"**
Select the lesson plan to view: AccessControlMatrix.help View

ExecResults for '/bin/sh -c cat "/root/Downloads/.extract/webapps/WebGoat/plug1 Output...'

Lesson Plan Title: How to Create Database Back Door Attacks.

Cookies / Parameters

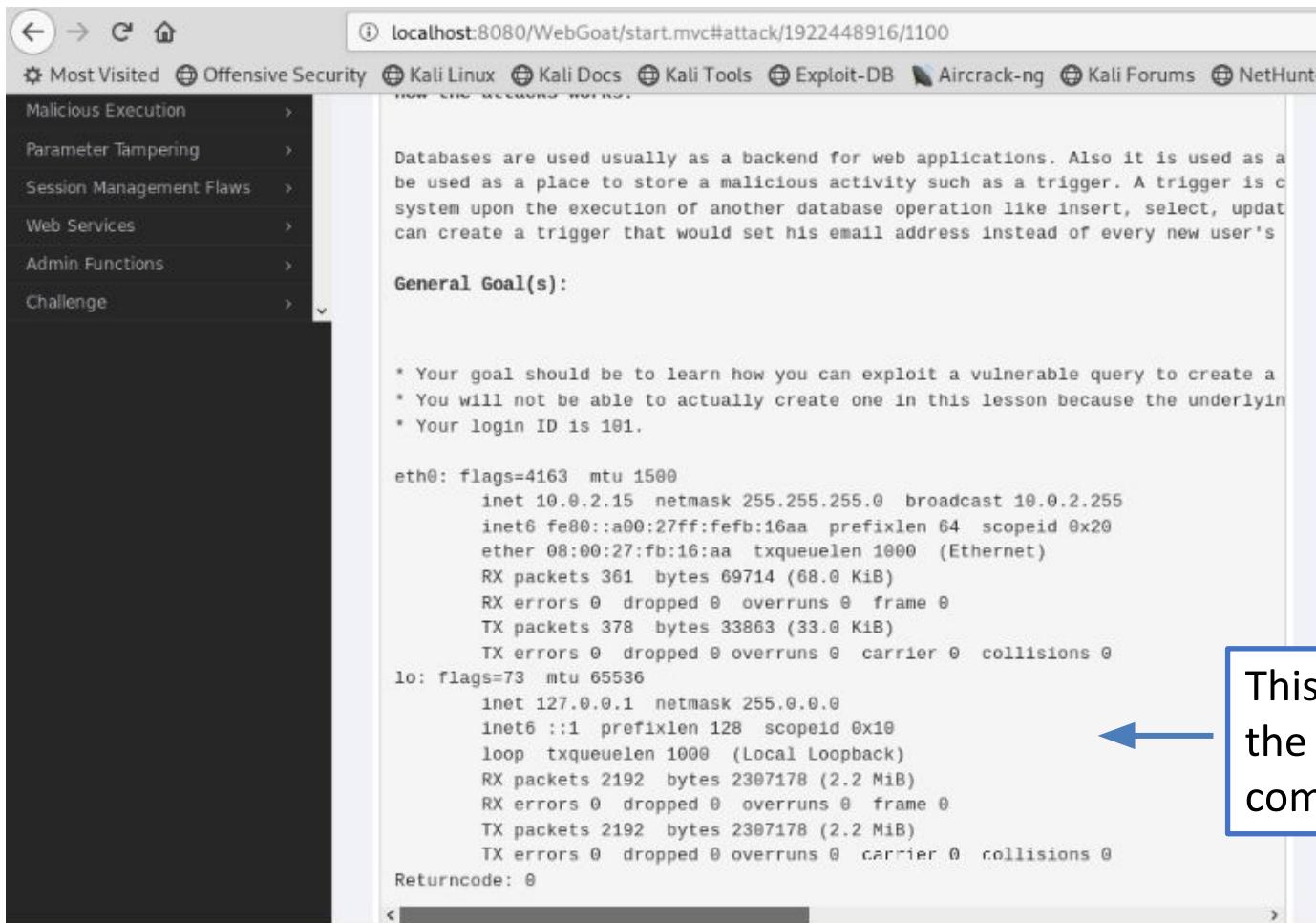
Cookie/s	
name	JSSESSIONID
value	918007DA4172F6351B2850D14F5248E2
comment	
domain	-1
maxAge	-1
path	false
secure	false
version	0
httpOnly	false

Parameters

scr	1922448916
menu	1100
stage	
num	

This is the expected answer

- Example Command injection in WebGoat



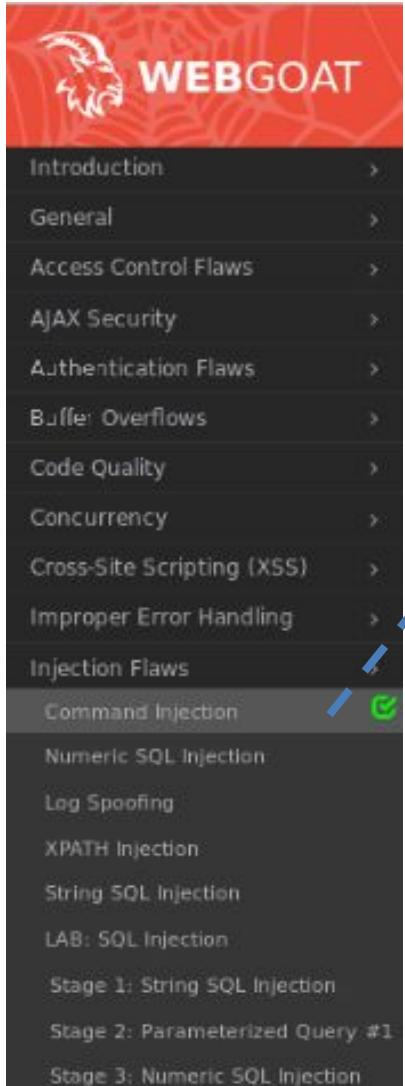
The screenshot shows a web browser window with the URL `localhost:8080/WebGoat/start.mvc#attack/1922448916/1100`. The page content discusses database triggers and general goals. It then displays a shell command output:

```
* Your goal should be to learn how you can exploit a vulnerable query to create a
* You will not be able to actually create one in this lesson because the underlying
* Your login ID is 101.

eth0: flags=4163 mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:feb:16aa prefixlen 64 scopeid 0x20
        ether 08:00:27:fb:16:aa txqueuelen 1000 (Ethernet)
        RX packets 361 bytes 69714 (68.0 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 378 bytes 33863 (33.0 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73 mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10
        loop txqueuelen 1000 (Local Loopback)
        RX packets 2192 bytes 2307178 (2.2 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 2192 bytes 2307178 (2.2 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
Returncode: 0
```

A blue callout box with an arrow points from the text "This is the result of the injected command" to the command output.

Execution on live!



The image shows a screenshot of the WEBGOAT navigation menu. The menu has a red header with the WEBGOAT logo. Below the header, there is a list of categories: Introduction, General, Access Control Flaws, AJAX Security, Authentication Flaws, Buffer Overflows, Code Quality, Concurrency, Cross-Site Scripting (XSS), Improper Error Handling, Injection Flaws, Command Injection, Numeric SQL Injection, Log Spoofing, XPATH Injection, String SQL Injection, LAB: SQL Injection, Stage 1: String SQL Injection, Stage 2: Parameterized Query #1, and Stage 3: Numeric SQL Injection. The "Command Injection" category is highlighted with a green checkmark icon.

<https://www.youtube.com/watch?v=M2wUCSleeqY&list=PL9HjVcGKtXM305JN2FjKyHNAGIZxZNzNn&index=1>

Bibliography

Main resources:

https://www.owasp.org/index.php/Top_10-2017_Top_10

https://www.owasp.org/index.php/Top_10_2013-Top_10

Multimedia resources:

<https://www.youtube.com/playlist?list=PL9HjVcGKtXM305JN2FjKyHNAGIZxZNzNn>

<https://www.youtube.com/playlist?list=PL9HjVcGKtXM2pi1nY2g6SWM9yiyetEB5A>



Universidad del
Rosario



MACC



HINNT

¡Gracias!

