GOBIERNO DEL ESTADO DE MÉXICO

UNIVERSIDAD TECNOLÓGICA DEL VALLE DE TOLUCA

# UNIVERSIDAD TECNOLÓGICA DEL VALLE DE TOLUCA

## Dirección de Carrera de Tecnologías de la Información y Comunicación

**Ingeniería en Desarrollo y Gestión de Software**

# Report of Practice: SQL Injection in OWASP-ZAP

**Teacher:**

Jorge Luis Galindo Lara

**Name Student:**

Islas Acosta Luis Enrique – 222310478

**Group:**

IDGS – 83

**Four-month period:**

8º

**Place**

Santa María Atarasquillo, Lerma, Mexico.

**Delivery Date:**

16- January -2026

# Contenido



# Imágenes

# Introduction

The object in this practice is demonstrate the use od OWASP-ZAP as a tool of analysis of security for detect vulnerability of Injection SQL in web app. Were used two virtual machines:

- Machine attack: Kali Linux with  OWASP-ZAP.
- Machine objective: Debian with the app vulnerability OWASP Mutillidae II.

## Install the OWASP ZAP of Kali Linux

- First, I entered the codes:
  - sudo apt update
  - sudo apt install zaproxy -y

  These codes are for update the repositories of the System Operative and installation of OWASP-ZAP
  - owas-zap

  This code is for execute OWASP-ZAP

## Verify the connection between machines

- For the verify between machines first enable bridge adapter in settings of the virtual machine in red because the red default is  NAT and in this case they couldn´t connect between this.
- Once it was done this config it is verified if you have a connection whit the code
  - ping (id of the computer) -> ping 192.168.1.11

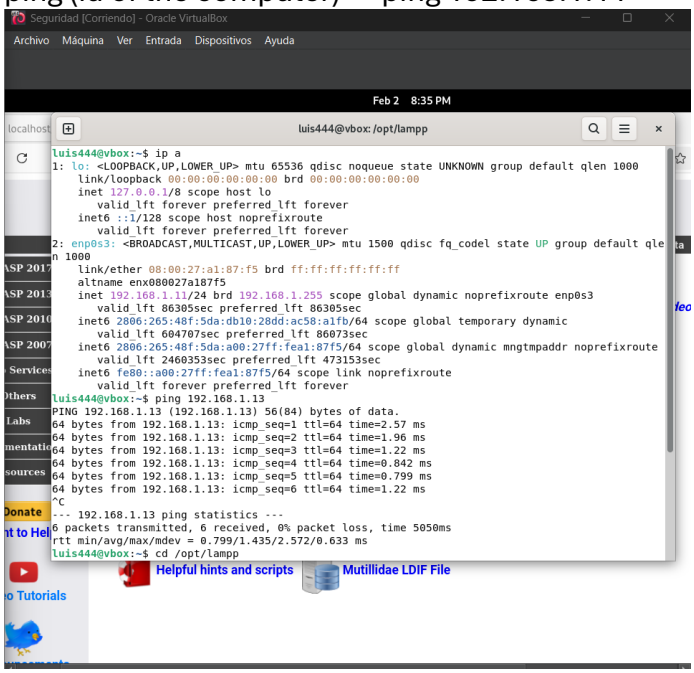


*Figure 1 Confirm Connection successful*

The result was  0% packet loss, confirming  connection successful

## Config of OWASP-ZAP

After of execute owasp-zap in  the quick launch tab, log in the URL: http://192.168.1.11/src/

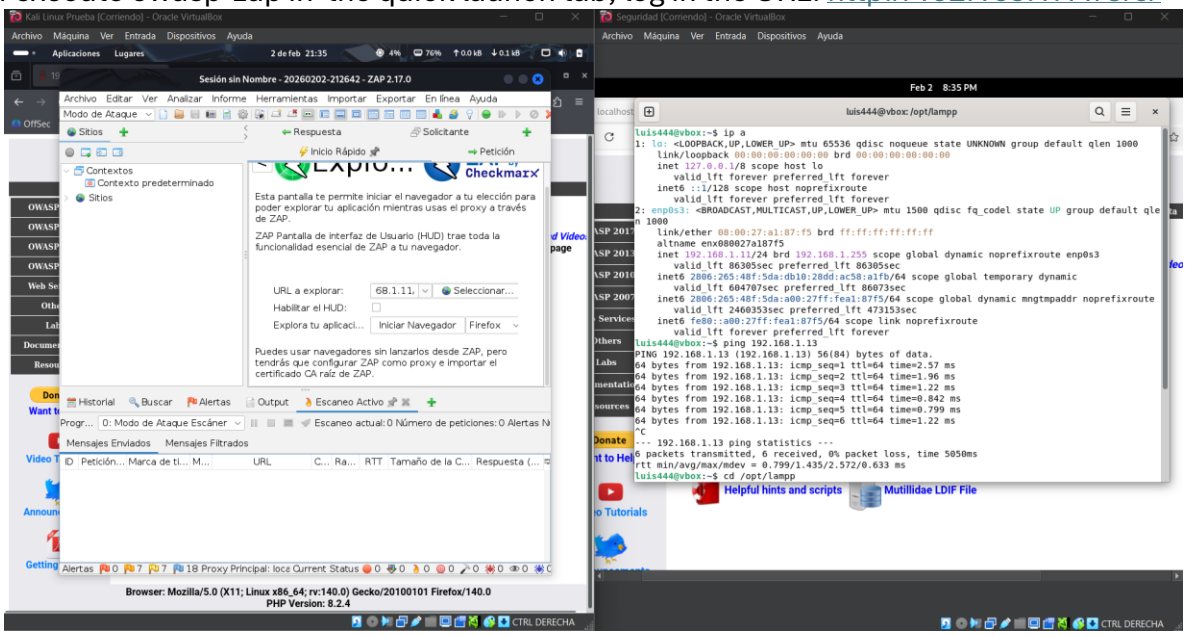

*Figure 2 Quick start*

- After  I did click in attack for start the scan initial

## Execute of Spider

In the left panel I select attack and Spider for track all the pages and formulary of Mutillidae

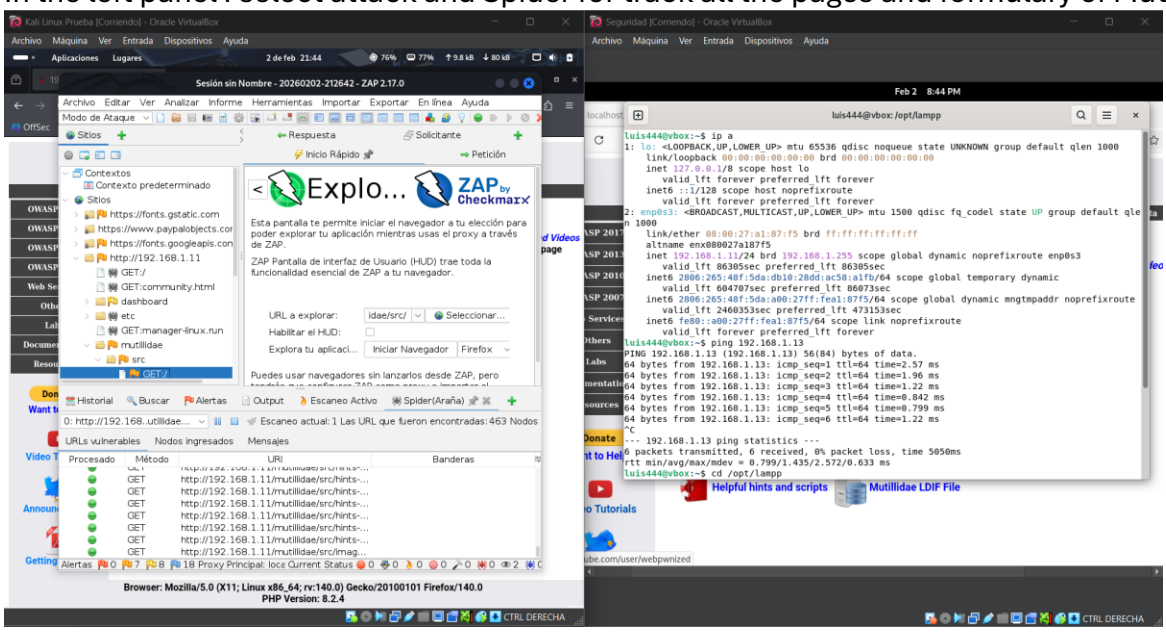

*Figure 3 Execute Spider*

## Scan Active

Subsequently again select the page which I would attack and I did click in attack scan active and ZAP send  payloads of the injection SQL at the parameters discovered

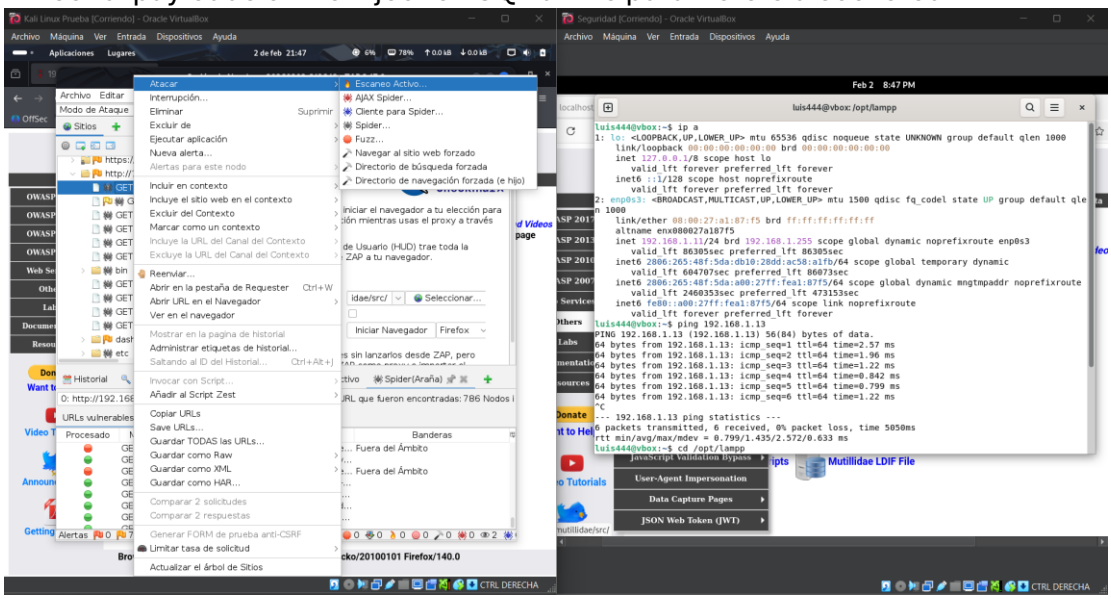

*Figure 4 Scan Active*

## Results

In the pages Alerts, ZAP report vulnerabilities of the SQL Injection

- Example:
  Parameters vulnerability: id in GET /mutillidae/src/sqli.php?id=1
  Payload used: ‘OR ‘1’=’1
  Evidence: answer of the service showing dates not authorized

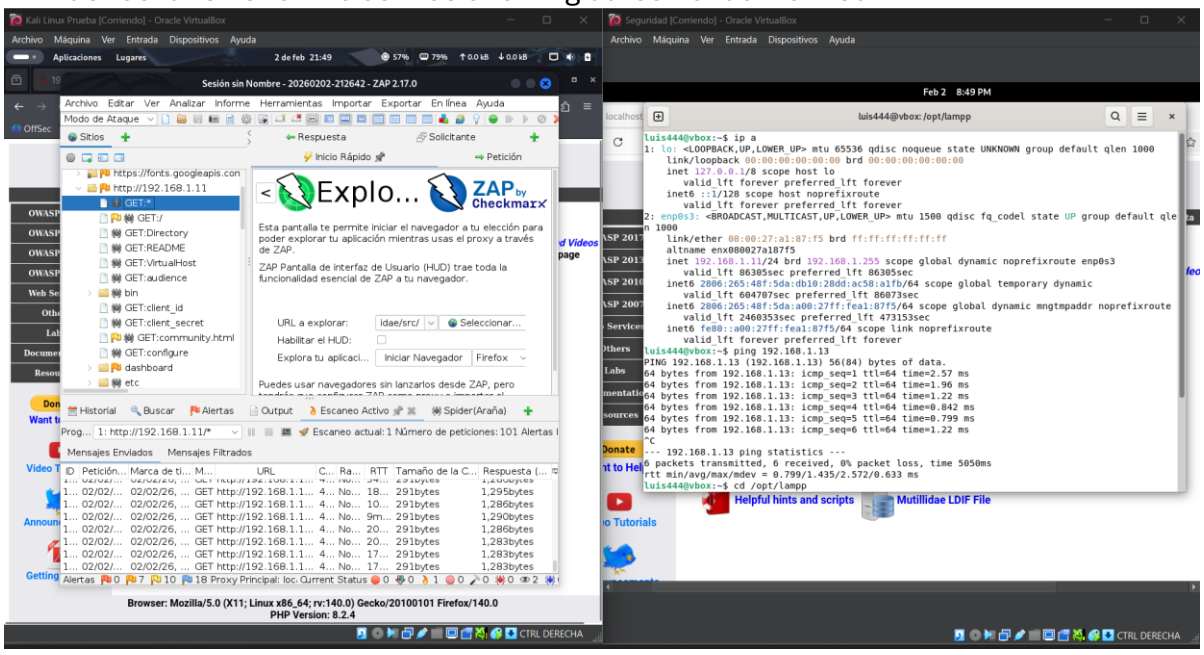

*Figure 5 Results Scan Active*

## Generate of report

In menu Inform I did click in Generate inform  and I select format HTML and keep archive with find and evidences

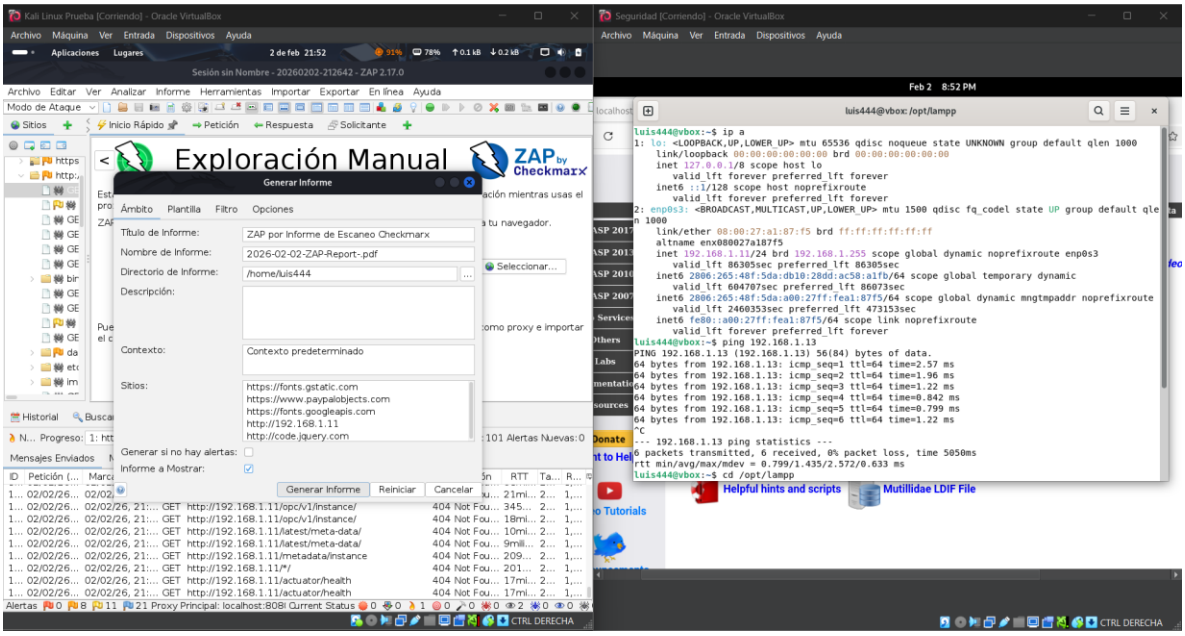


*Figure 6 Generate Report*

## Conclusions

OWASP ZAP permitted identify vulnerabilities in SQL Injection in Mutillidae II. The practice demonstrates the importance od validate entries and use consults parameterized and the implement the prepared statements, applied controllers of security in the web app and used the tokens and validate of dates correct in the database about of the user who wants login in the app and register all the credentials attempting to log in.