



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

Introduction	1
Install the OWASP ZAP of Kali Linux	1
Verify the connection between machines	1
Config of OWASP-ZAP	2
Execute of Spider	2
Scan Active	3
Results	3
Generate of report	4
Conclusions	4

Imágenes

Figure 1 Confirm Connection successful.....	1
Figure 2 Quick start	2
Figure 3 Execute Spider.....	2
Figure 4 Scan Active	3
Figure 5 Results Scan Active.....	3
Figure 6 Generate Report	4

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

The screenshot shows a terminal window titled "Seguridad [Corriendo] - Oracle VirtualBox". The window contains the following text:

```
luis444@vbox:~$ ip a
1: lo: <LOOPBACK,NO-SIUP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback brd 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 brd 0.0.0.0 scope host lo
            valid_lft forever preferred_lft forever
            inet6 ::1/128 scope host noprefixroute
                valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:a1:87:f5 brd ff:ff:ff:ff:ff:ff
    altname enx0800027a187f5
    inet 192.168.1.11/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 86305sec preferred_lft 86305sec
    inet6 2806:265:40f:5da:db10:28d3:ac8:a1fb/64 scope global temporary dynamic
        valid_lft 604707sec preferred_lft 60673sec
    inet6 fe80::a0:27ff:fea1:87f5/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
luis444@vbox:~$ ping 192.168.1.13
PING 192.168.1.13 (192.168.1.13) 56(84) bytes of data.
64 bytes from 192.168.1.13: icmp_seq=1 ttl=64 time=2.57 ms
64 bytes from 192.168.1.13: icmp_seq=2 ttl=64 time=1.96 ms
64 bytes from 192.168.1.13: icmp_seq=3 ttl=64 time=1.22 ms
64 bytes from 192.168.1.13: icmp_seq=4 ttl=64 time=0.842 ms
64 bytes from 192.168.1.13: icmp_seq=5 ttl=64 time=0.799 ms
64 bytes from 192.168.1.13: icmp_seq=6 ttl=64 time=1.22 ms
luis444@vbox:~$ cd /opt/lampp
```

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/>

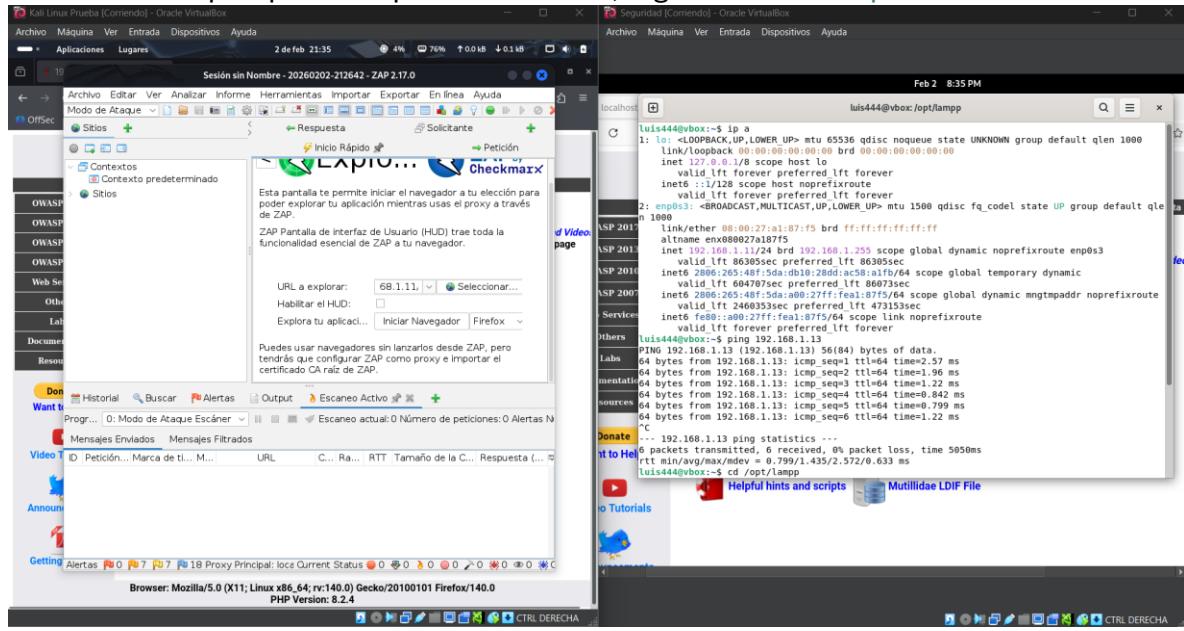


Figure 2 Quick start

- 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

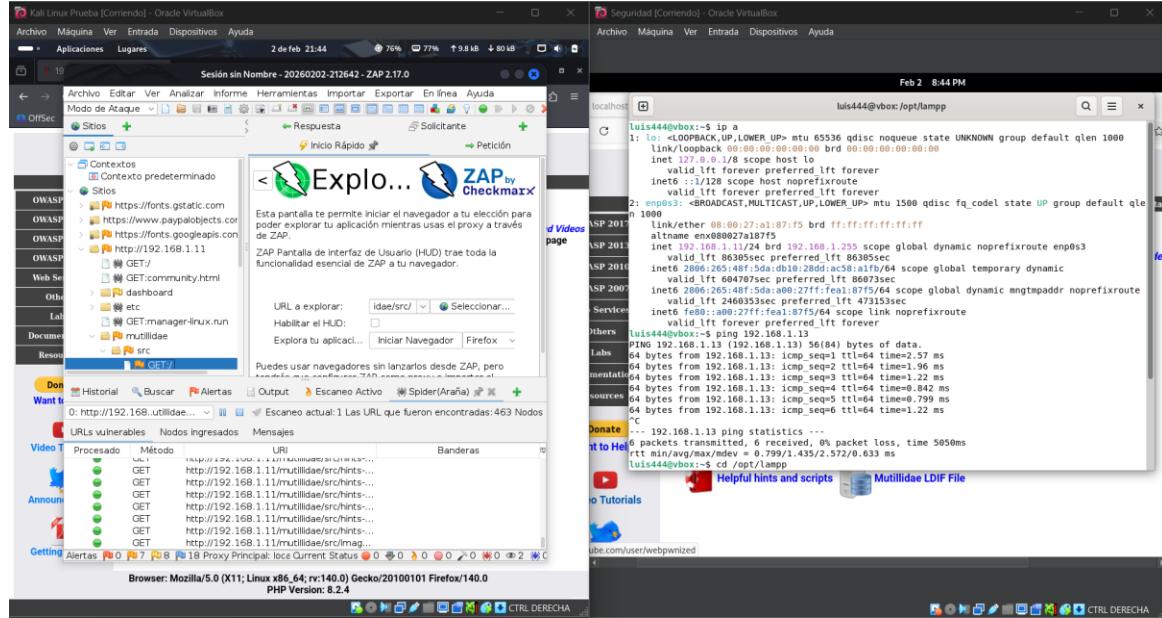


Figure 3 Execute Spider

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

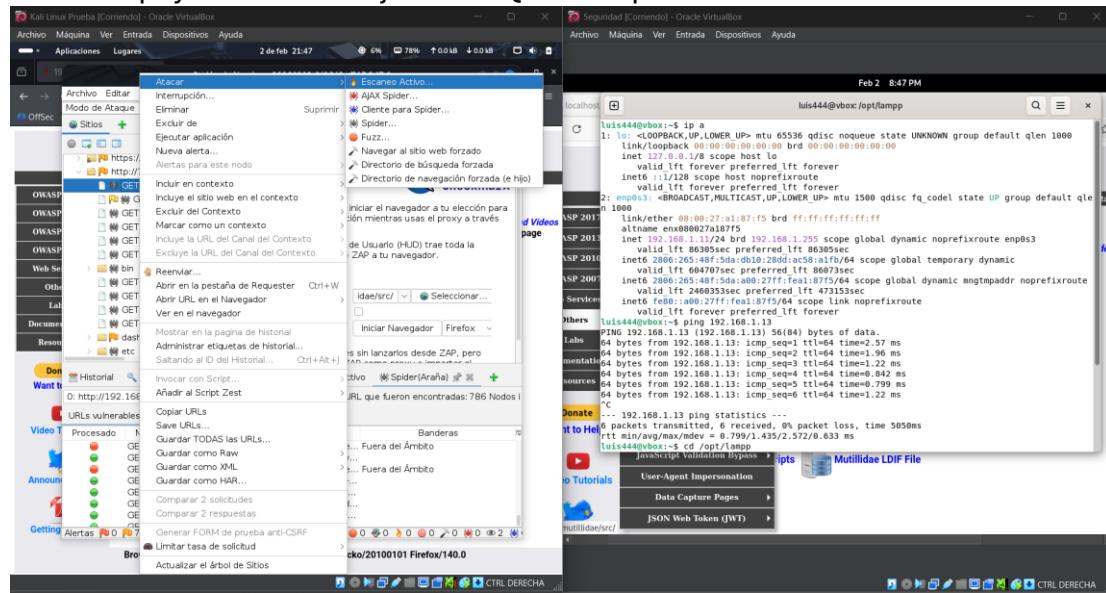


Figure 4 Scan Active

Results

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

- Example:

Parameters vulnerability: id in GET /mutillidae/src/sqlip.php?id=1

Payload used: 'OR '1'='1

Evidence: answer of the service showing dates not authorized

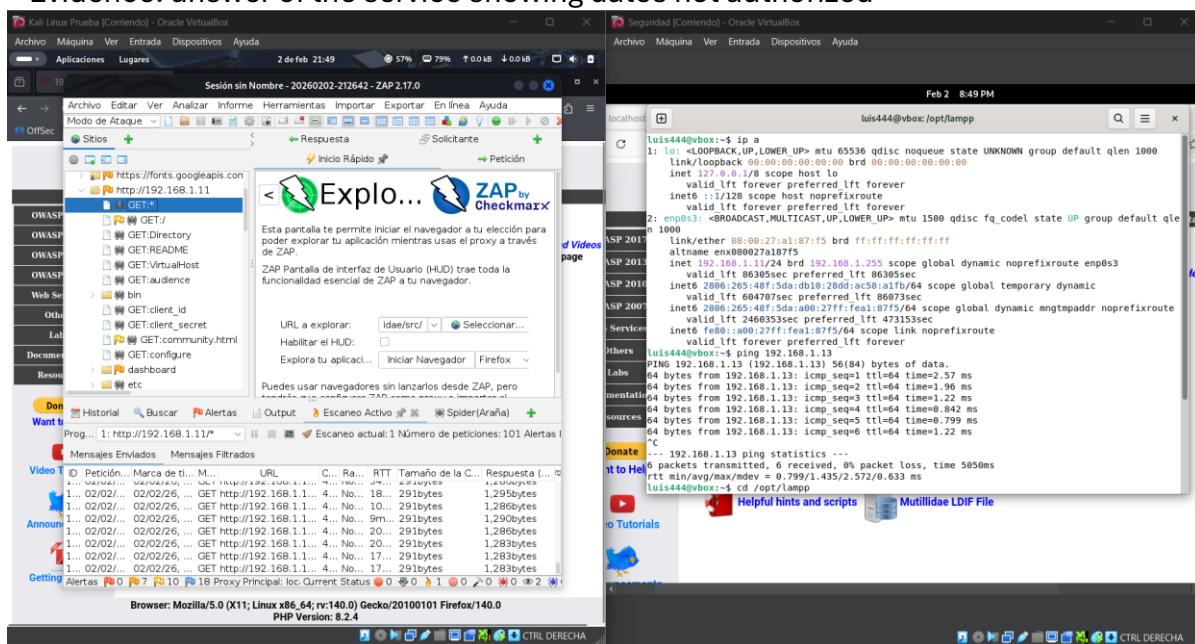


Figure 5 Results Scan Active

Generate of report

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

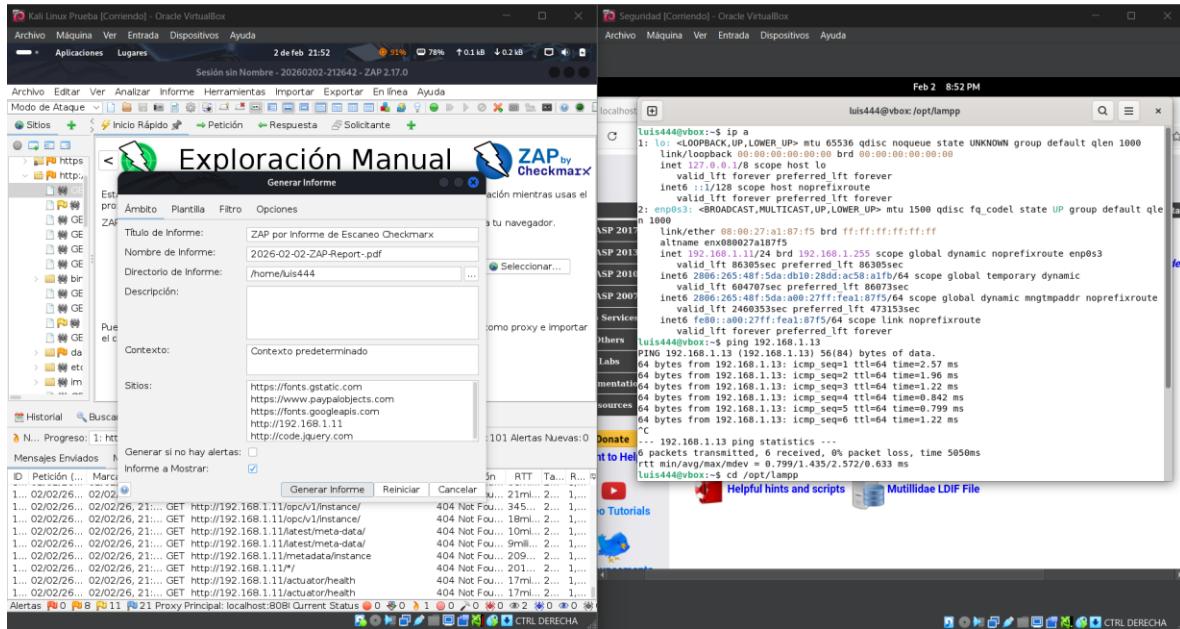


Figure 6 Generate Report

Conclusions

OWASP ZAP permitted identify vulnerabilities in SQL Injection in Mutillidae II. The practice demonstrates the importance of 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.