

Xss reflected - Sql Injection

Esercizio del Giorno:

Argomento: Sfruttamento delle Vulnerabilità XSS e SQL Injection sulla DVWA

Obiettivi:

Configurare il laboratorio virtuale per sfruttare con successo le vulnerabilità XSS e SQL Injection sulla Damn Vulnerable Web Application DVWA.

Configurazione del Laboratorio:

- Configurare il vostro ambiente virtuale in modo che la macchina DVWA sia raggiungibile dalla macchina Kali Linux (l'attaccante).
- Verificare la comunicazione tra le due macchine utilizzando il comando ping.

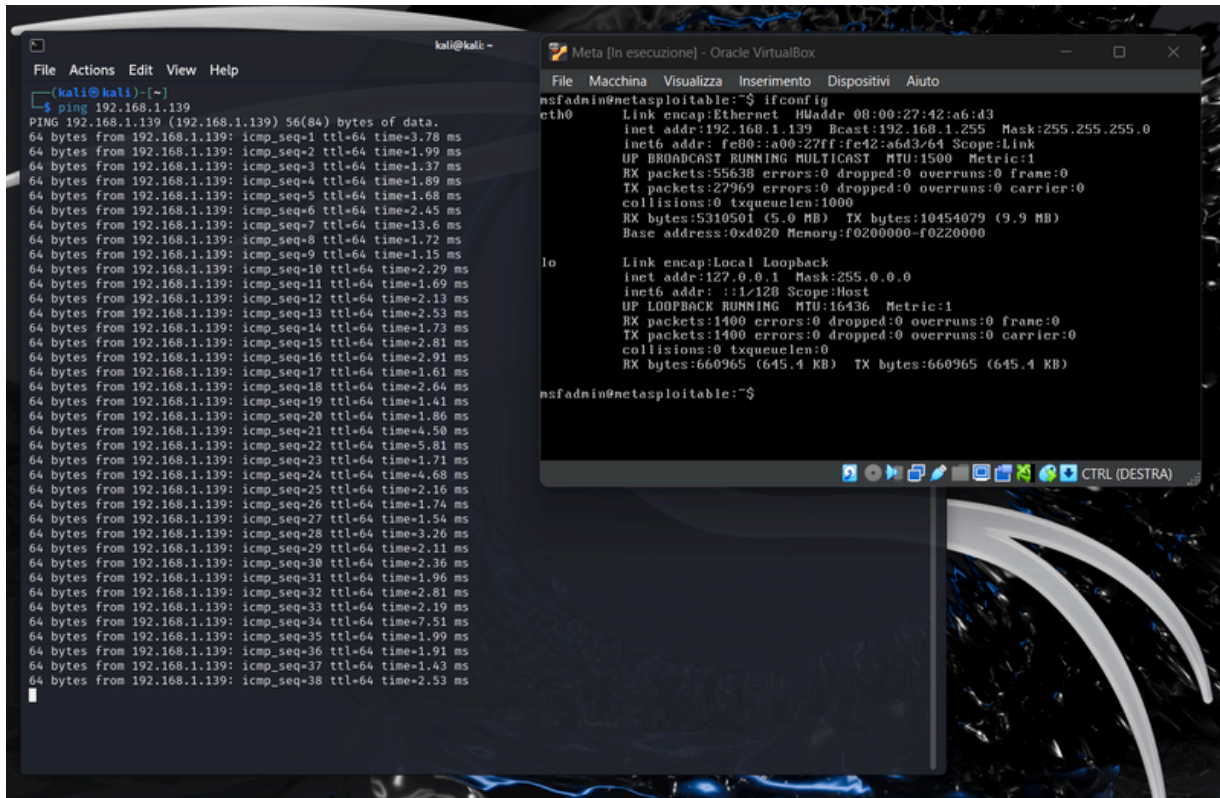
Impostazione della DVWA:

- Accedete alla DVWA dalla macchina Kali Linux tramite il browser.
- Navigate fino alla pagina di configurazione e settate il livello di sicurezza a LOW.

Sfruttamento delle Vulnerabilità:

- Scegliete una vulnerabilità XSS reflected e una vulnerabilità SQL Injection (non blind).
- Utilizzate le tecniche viste nella lezione teorica per sfruttare con successo entrambe le vulnerabilità.

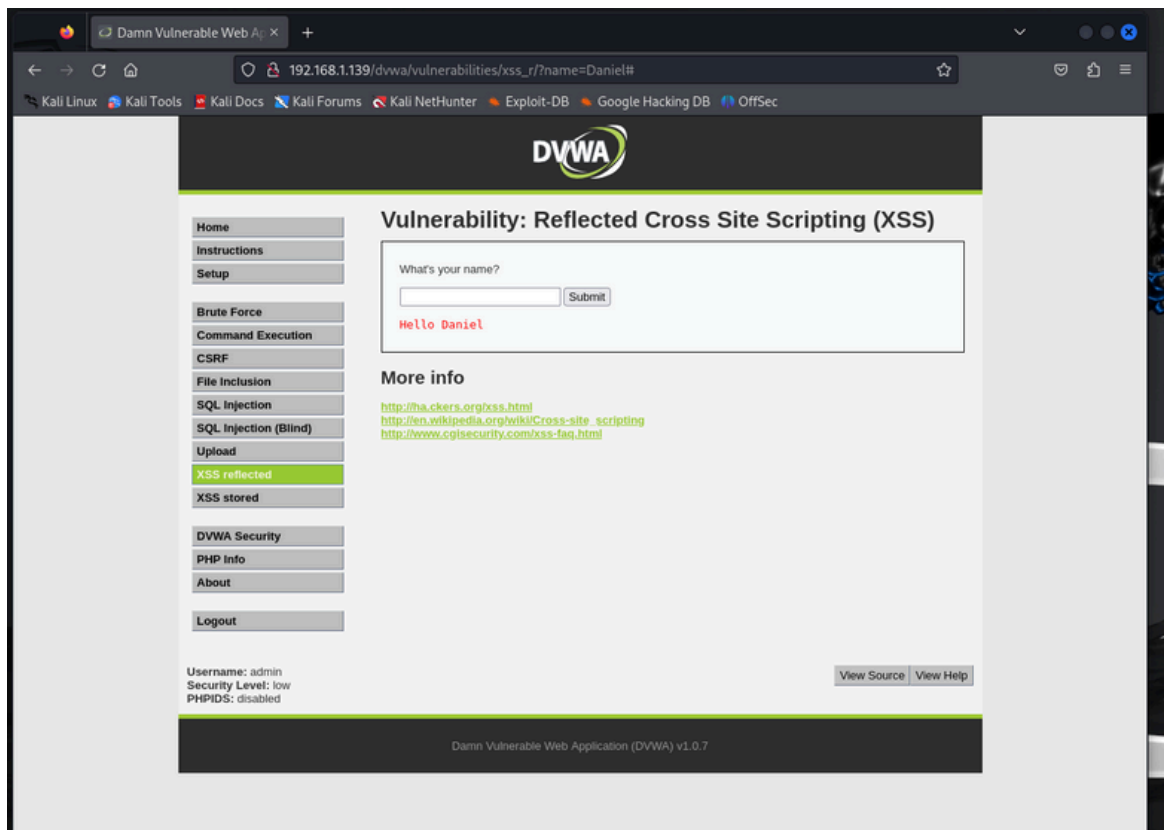
Ping + nome



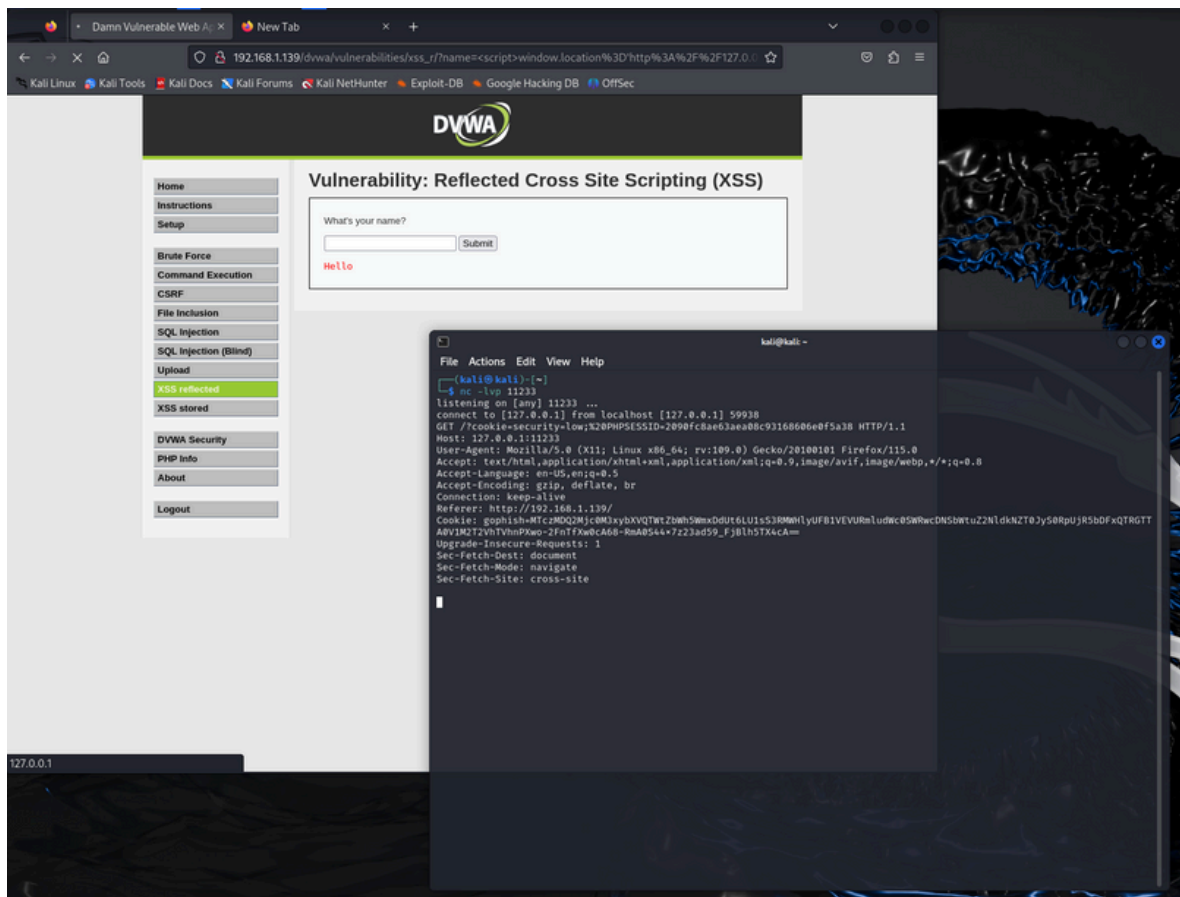
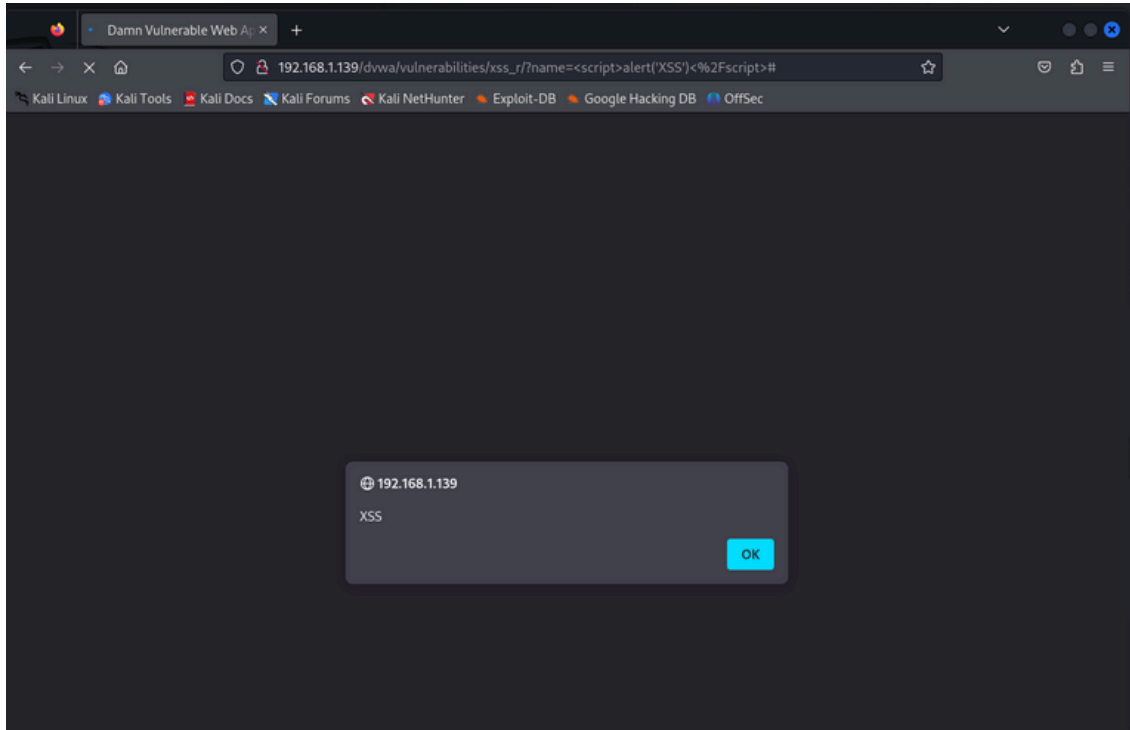
The image shows two overlapping windows from a Kali Linux system. The background window is a terminal running a ping command to 192.168.1.139. The foreground window is a VirtualBox console showing the output of the 'ifconfig' command for the 'eth0' interface.

```
kali@kali: ~  
$ ping 192.168.1.139  
PING 192.168.1.139 (192.168.1.139) 56(84) bytes of data:  
64 bytes from 192.168.1.139: icmp_seq=1 ttl=64 time=3.78 ms  
64 bytes from 192.168.1.139: icmp_seq=2 ttl=64 time=1.99 ms  
64 bytes from 192.168.1.139: icmp_seq=3 ttl=64 time=1.37 ms  
64 bytes from 192.168.1.139: icmp_seq=4 ttl=64 time=1.89 ms  
64 bytes from 192.168.1.139: icmp_seq=5 ttl=64 time=1.68 ms  
64 bytes from 192.168.1.139: icmp_seq=6 ttl=64 time=2.45 ms  
64 bytes from 192.168.1.139: icmp_seq=7 ttl=64 time=13.6 ms  
64 bytes from 192.168.1.139: icmp_seq=8 ttl=64 time=1.72 ms  
64 bytes from 192.168.1.139: icmp_seq=9 ttl=64 time=1.15 ms  
64 bytes from 192.168.1.139: icmp_seq=10 ttl=64 time=2.29 ms  
64 bytes from 192.168.1.139: icmp_seq=11 ttl=64 time=1.69 ms  
64 bytes from 192.168.1.139: icmp_seq=12 ttl=64 time=2.13 ms  
64 bytes from 192.168.1.139: icmp_seq=13 ttl=64 time=2.53 ms  
64 bytes from 192.168.1.139: icmp_seq=14 ttl=64 time=1.73 ms  
64 bytes from 192.168.1.139: icmp_seq=15 ttl=64 time=2.81 ms  
64 bytes from 192.168.1.139: icmp_seq=16 ttl=64 time=2.91 ms  
64 bytes from 192.168.1.139: icmp_seq=17 ttl=64 time=1.61 ms  
64 bytes from 192.168.1.139: icmp_seq=18 ttl=64 time=2.64 ms  
64 bytes from 192.168.1.139: icmp_seq=19 ttl=64 time=1.41 ms  
64 bytes from 192.168.1.139: icmp_seq=20 ttl=64 time=1.86 ms  
64 bytes from 192.168.1.139: icmp_seq=21 ttl=64 time=4.50 ms  
64 bytes from 192.168.1.139: icmp_seq=22 ttl=64 time=5.81 ms  
64 bytes from 192.168.1.139: icmp_seq=23 ttl=64 time=1.71 ms  
64 bytes from 192.168.1.139: icmp_seq=24 ttl=64 time=4.68 ms  
64 bytes from 192.168.1.139: icmp_seq=25 ttl=64 time=2.16 ms  
64 bytes from 192.168.1.139: icmp_seq=26 ttl=64 time=1.74 ms  
64 bytes from 192.168.1.139: icmp_seq=27 ttl=64 time=1.54 ms  
64 bytes from 192.168.1.139: icmp_seq=28 ttl=64 time=3.26 ms  
64 bytes from 192.168.1.139: icmp_seq=29 ttl=64 time=2.11 ms  
64 bytes from 192.168.1.139: icmp_seq=30 ttl=64 time=2.36 ms  
64 bytes from 192.168.1.139: icmp_seq=31 ttl=64 time=1.96 ms  
64 bytes from 192.168.1.139: icmp_seq=32 ttl=64 time=2.81 ms  
64 bytes from 192.168.1.139: icmp_seq=33 ttl=64 time=2.19 ms  
64 bytes from 192.168.1.139: icmp_seq=34 ttl=64 time=7.51 ms  
64 bytes from 192.168.1.139: icmp_seq=35 ttl=64 time=1.99 ms  
64 bytes from 192.168.1.139: icmp_seq=36 ttl=64 time=1.91 ms  
64 bytes from 192.168.1.139: icmp_seq=37 ttl=64 time=1.43 ms  
64 bytes from 192.168.1.139: icmp_seq=38 ttl=64 time=2.53 ms
```

```
msfadmin@metasploit> ifconfig  
eth0      Link encap:Ethernet  HWaddr 08:00:27:42:a6:d3  
          inet addr:192.168.1.139  Bcast:192.168.1.255  Mask:255.255.255.0  
          inet6 addr: fe80::a00:27ff:fe42:a6d3/64  Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:55638  errors:0  dropped:0  overruns:0  frame:0  
          TX packets:27969  errors:0  dropped:0  overruns:0  carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:5310501 (5.0 MB)  TX bytes:10454079 (9.9 MB)  
          Base address:0xd020  Memory:f0200000-f0220000  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128  Scope:Host  
          UP LOOPBACK RUNNING  MTU:16384  Metric:1  
          RX packets:1400  errors:0  dropped:0  overruns:0  frame:0  
          TX packets:1400  errors:0  dropped:0  overruns:0  carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:660965 (645.4 KB)  TX bytes:660965 (645.4 KB)
```



Pop-up + netcat



Sql

The screenshot shows a web browser window with the URL `192.168.1.139/dvwa/vulnerabilities/sqli/?id='+OR+'1'%3D'1&Submit=Submit#`. The page title is "Vulnerability: SQL Injection". On the left is a navigation menu with options like Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (highlighted), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area shows a "User ID:" label with an input field and a "Submit" button. Below this, a list of user records is displayed in red text:

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

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

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

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

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

Below the list, there is a "More info" section with three links:

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

At the bottom left, the status is shown: "Username: admin", "Security Level: low", and "PHPIDS: disabled". At the bottom right, there are "View Source" and "View Help" buttons. The footer of the application says "Damn Vulnerable Web Application (DVWA) v1.0.7".

Ho effettuato il ping su Kali per verificare la comunicazione con Meta. Successivamente, sulla DVWA (Damn Vulnerable Web Application), ho impostato la sicurezza su low. Poi, nella sezione XSS Reflected, ho inserito il nome, ottenendo come output "Hello Daniel". Per testare la vulnerabilità, ho inserito il comando `<script>alert('XSS');</script>`, generato da ChatGPT, per verificare la presenza di un attacco XSS Reflected.

A richiesta del professore, ho utilizzato Netcat. Ho chiesto a ChatGPT di generare uno script da inserire nella DVWA e successivamente il comando per mettermi in ascolto sulla porta da me indicata.

Infine, per SQL Injection, ho inserito il comando `' OR '1'='1'`, per bypassare il sistema di autenticazione o visualizzare informazioni aggiuntive. Come mostrato nello screenshot, ciò ha funzionato correttamente.

Daniel_Gabriel_Costeanu