

CS0424IT — ESERCITAZIONE S3L2 - CONFIGURAZIONE DVWA

Simone La Porta



TRACCIA

Nella lezione pratica di oggi vedremo come configurare una DVWA (Damn Vulnerable Web Application)

SVOLGIMENTO

In questo documento, verranno illustrate le istruzioni per l'installazione e la configurazione della web application DVWA (Damn Vulnerable Web Application). DVWA è un progetto software che include intenzionalmente vulnerabilità di sicurezza, utile per esercitazioni e analisi nel campo della sicurezza informatica. Per iniziare, è necessario installare DVWA.

Dopo aver completato l'installazione, è necessario configurare DVWA modificando il file `config.inc.php`.

Per farlo, eseguire il seguente comando in terminale:

```
sudo nano config.inc.php
```

All'interno del file, aggiornare il nome utente e la password utilizzati per la connessione al database, impostandoli entrambi su `kali`. Il file `config.inc.php` dovrebbe contenere le seguenti righe:

```

└──(kali㉿kali)-[~]
└─$ cd /var/www/html

└──(kali㉿kali)-[/var/www/html]
└─$ sudo git clone https://github.com/digininja/DVWA
Cloning into 'DVWA' ...
remote: Enumerating objects: 4503, done.
remote: Counting objects: 100% (53/53), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 4503 (delta 19), reused 33 (delta 8), pack-reused 4450
Receiving objects: 100% (4503/4503), 2.30 MiB | 5.80 MiB/s, done.
Resolving deltas: 100% (2114/2114), done.

└──(kali㉿kali)-[/var/www/html]
└─$ sudo chmod -R 777 DVWA/
pyright (c) 2000-2019, Oracle
└──(kali㉿kali)-[/var/www/html]
└─$ cd DVWA/config

└──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ sudo cp config.inc.php.dist config.inc.php

└──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ sudo nano config.inc.php

```

```

root@kali:~/var/www/html/DVWA/config
GNU nano 8.0           config.inc.php

<?php

# If you are having problems connecting to the MySQL database and all of the variables below are correct
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to sockets.
#   Thanks to @digininja for the fix.

# Database management system to use
$DBMS = 'MySQL';
#$DBMS = 'PGSQL'; // Currently disabled

# Database variables
#   WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.
#   Please use a database dedicated to DVWA.

#
# If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.
#   See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ] = getenv('DB_SERVER') ?: '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ] = 'kali';
$_DVWA[ 'db_password' ] = 'kali';
$_DVWA[ 'db_port' ] = '3306';

# ReCAPTCHA settings
#   Used for the 'Insecure CAPTCHA' module
#   You'll need to generate your own keys at: https://www.google.com/recaptcha/admin
$_DVWA[ 'recaptcha_public_key' ] = '';
$_DVWA[ 'recaptcha_private_key' ] = '';

# Default security level
#   Default value for the security level with each session.
#   The default is 'impossible'. You may wish to set this to either 'low', 'medium', 'high' or 'impossible'.
$_DVWA[ 'default_security_level' ] = 'impossible';

# Default locale
#   Default locale for the help page shown with each session.

^G Help      ^A Write Out  ^F Where Is  ^K Cut        ^T Execute    ^C Location  M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^M Replace   ^U Paste      ^J Justify    ^V Go To Line M-E Redo      M-6 Copy

```

```

$_DVWA[ 'db_user' ] = 'kali';
$_DVWA[ 'db_password' ] = 'kali';

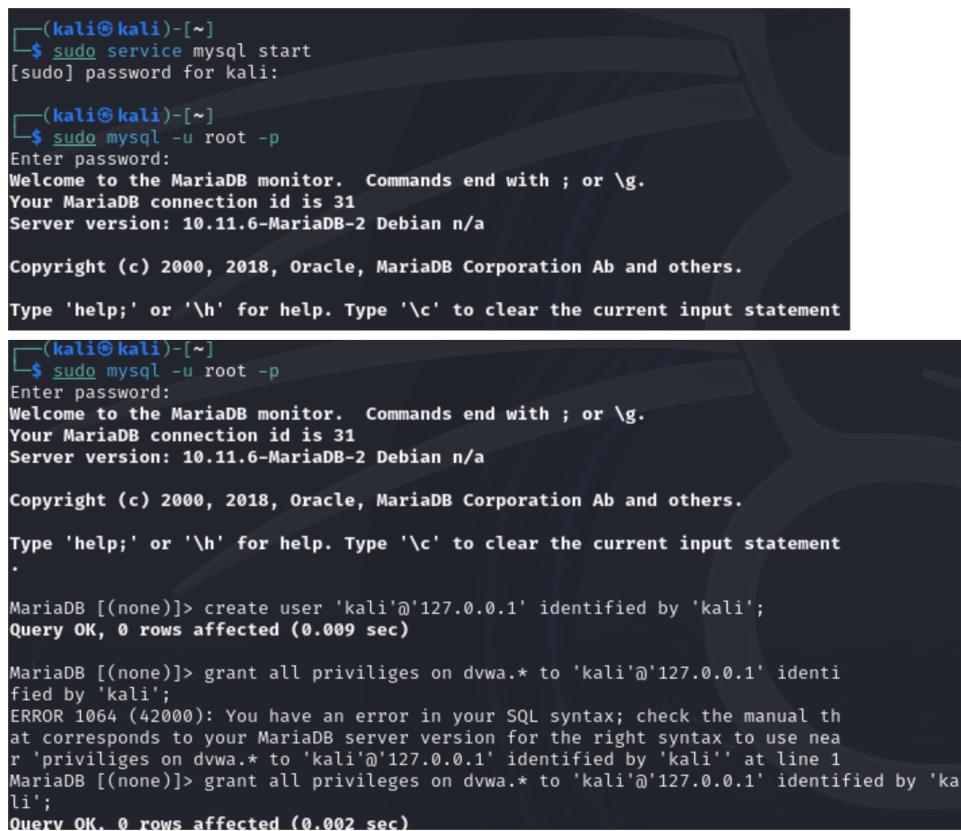
```

Una volta completata la configurazione, salvare le modifiche e chiudere l'editor. DVWA sarà ora pronto per essere utilizzato per le esercitazioni e le analisi delle vulnerabilità di sicurezza.

Dopo aver configurato le credenziali, è necessario avviare il servizio web Apache2 e il servizio database MySQL utilizzando i permessi di amministratore. Per farlo, eseguire i seguenti comandi:

```
sudo service apache2 start  
sudo service mysql start
```

Il servizio MySQL verrà avviato utilizzando il database MariaDB. Successivamente, creare un'utenza kali e assegnarle i privilegi da amministratore.



```
(kali㉿kali)-[~]  
└$ sudo service mysql start  
[sudo] password for kali:  
  
(kali㉿kali)-[~]  
└$ sudo mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 31  
Server version: 10.11.6-MariaDB-2 Debian n/a  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement  
  
(kali㉿kali)-[~]  
└$ sudo mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 31  
Server version: 10.11.6-MariaDB-2 Debian n/a  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement  
  
MariaDB [(none)]> create user 'kali'@'127.0.0.1' identified by 'kali';  
Query OK, 0 rows affected (0.009 sec)  
  
MariaDB [(none)]> grant all privileges on dwva.* to 'kali'@'127.0.0.1' identified by 'kali';  
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'privileges on dwva.* to 'kali'@'127.0.0.1' identified by 'kali'' at line 1  
MariaDB [(none)]> grant all privileges on dwva.* to 'kali'@'127.0.0.1' identified by 'kali';  
Query OK, 0 rows affected (0.002 sec)
```

Dopo aver configurato l'utenza, creare un database su DVWA. Accedere all'applicazione DVWA con username: admin e password: password. Quindi, selezionare il livello di sicurezza della web app impostandolo al minimo (*Low*). Un livello di sicurezza più basso rende meno complicato sfruttare le vulnerabilità.

A questo punto, utilizzare l'applicazione Burp Suite e scegliere un progetto temporaneo. Attivare l'intercettazione delle richieste di login su Burp Suite. Accedere tramite browser all'indirizzo <http://127.0.0.1/DVWA> e tentare di intercettare la propria richiesta di login.

```

└──(kali㉿kali)-[~]
└─$ service apache2 start

└──(kali㉿kali)-[~]
└─$ cd /etc/php

└──(kali㉿kali)-[/etc/php]
└─$ ls -l
total 4
drwxr-xr-x 5 root root 4096 Feb 25 10:47 8.2

└──(kali㉿kali)-[/etc/php]
└─$ ls
8.2

└──(kali㉿kali)-[/etc/php]
└─$ cd /etc/php/8.2/apache2

└──(kali㉿kali)-[/etc/php/8.2/apache2]
└─$ ll

└──(kali㉿kali)-[/etc/php/8.2/apache2]
└─$ sudo nano php.ini
[sudo] password for kali:

; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; https://php.net/allow-url-fopen
allow_url_fopen = On

; Whether to allow include/require to open URLs (like https:// or ftp://) as files.
; https://php.net/allow-url-include
allow_url_include = On

```

DVWA Security

Security Level

Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
Prior to DVWA v1.9, this level was known as 'high'.

Security level set to low

Per verificare il funzionamento dell'intercettazione, cambiare username e password iniziali con credenziali errate, proprio per verificare che il login fallisca.

Come previsto, con credenziali errate non sarà possibile effettuare il login. Questo sarà confermato dal corpo della risposta HTTP (*HTTP response*) dove si potrà leggere "Login

failed".

The screenshot shows the Burp Suite Community Edition interface. On the left, the "Proxy" tab is selected, displaying a POST request to `http://127.0.0.1:80/DVWA/login.php`. The request details show various headers and a body containing a login attempt. On the right, a browser window titled "Login :: Damn Vulnerable Web Application" is open at `http://127.0.0.1/DVWA/`. It displays the DVWA logo and a login form with fields for "Username" (set to "admin") and "Password" (set to "password"). Below the form is a "Login" button and a message stating "You have logged out". At the bottom of the browser window, a link reads "Damn Vulnerable Web Application (DVWA)".

```
Pretty Raw Hex
Request to http://127.0.0.1:80
POST /DVWA/login.php HTTP/1.1
Host: 127.0.0.1
Content-Length: 88
Cache-Control: max-age=0
sec-ch-us: "Chromium";v="125", "Not.A/Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
Origin: http://127.0.0.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.112
Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-Dest: document
Referer: http://127.0.0.1/DVWA/login.php
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=mg4u41rjd205a55dc00c8ng6; security=low
Connection: keep-alive
username=admin&password=password&Login=Login&user_token=6887d66ba3cf5b497c01a7dc18f983d
```

Burp Suite Community Edition v2024.4.5 - Temporary Project

Target: http://127.0.0.1 / HTTP/1.1

Request

```

1 GET /DVWA/Login.php HTTP/1.1
2 Host: 127.0.0.1
3 Cache-Control: max-age=0
4 sec-ch-ua: "Chromium";v="125", "Not A/Brand";v="24"
5 sec-ch-ua-mobile: ?0
6 sec-ch-ua-platform: "Linux"
7 Upgrade-Insecure-Requests: 1
8 Origin: http://127.0.0.1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
  like Gecko) Chrome/125.0.6422.112 Safari/537.36
10 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/
  apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-User: ?1
14 Sec-Fetch-Dest: document
15 Referer: http://127.0.0.1/DVWA/Login.php
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: en-US,en;q=0.9
18 Cookie: PHPSESSID=ng4u41rdj2l05a55dc00c8ng6; security=lw
19 Connection: keep-alive
20
21

```

Response

```

36 <br />
37 </div>
38 <!--div id="header">-->
39 <div id="content">
40   <form action="login.php" method="post">
41     <fieldset>
42       <label for="username">
43         Username
44       </label>
45       <input type="text" class="loginInput" size="20" name="username">
46       <br />
47       <label for="password">
48         Password
49       </label>
50       <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">
51       <br />
52       <br />
53       <p class="submit">
54         <input type="submit" value="Login" name="Login">
55       </p>
56     </fieldset>
57     <input type="hidden" name="user_token" value="eb85fddad595ae6588d5ffde51192ae39" />
58   </form>
59   <br />
60   <div class="message">
61     Login failed
62   </div>
63   <br />
64   <br />
65   <br />
66   <br />
67   <br />
68   <br />
69   <br />
70   <br />
71   <br />
72   <br />

```

0 highlights

1,709 bytes | 1,004 millis

Memory: 106.6MB



Username

Password

Login

Login failed