

W8D1

```
(kali㉿kali)-[~]  
$ sudo su  
[sudo] password for kali:  
(root㉿kali)-[/home/kali]  
#
```

sudo su" in ambiente Unix/Linux viene utilizzato per ottenere l'accesso alla shell di root o di un altro utente di sistema.

```
(kali㉿kali)-[~]  
$ sudo su  
[sudo] password for kali:  
(root㉿kali)-[/home/kali]  
# git clone https://github.com/digininja/DVWA s  
Cloning into 's' ...  
remote: Enumerating objects: 4494, done.  
remote: Counting objects: 100% (44/44), done.  
remote: Compressing objects: 100% (34/34), done.  
remote: Total 4494 (delta 15), reused 31 (delta 9), pack-reused 4450  
Receiving objects: 100% (4494/4494), 2.29 MiB | 415.00 KiB/s, done.  
Resolving deltas: 100% (2110/2110), done.  
  
(root㉿kali)-[/home/kali]  
#
```

git clone <https://github.com/digininja/DVWA>

è utilizzato per clonare un repository Git dalla piattaforma GitHub. In questo caso, il repository è "DVWA" (Damn Vulnerable Web Application), che è un'applicazione web appositamente progettata per essere vulnerabile, utilizzata per scopi educativi e di testing della sicurezza.

```
(root㉿kali)-[/home/kali]  
# cd /var/www/html  
  
(root㉿kali)-[/var/www/html]  
# chmod -R 777 DVWA/  
  
(root㉿kali)-[/var/www/html]  
#
```

cd /var/www/html

è utilizzato per spostarsi nella directory "/var/www/html". Questo è un percorso comune nelle distribuzioni Linux, dove solitamente vengono ospitate le pagine web e le risorse web.

`chmod -R 777 DVWA/`

è utilizzato per modificare i permessi dei file e delle directory in modo ricorsivo nella directory specificata, nel tuo caso "DVWA".

Ecco cosa fa il comando:

- **chmod:** Stands for "change mode", ed è utilizzato per modificare i permessi dei file.
- **-R:** Sta per "recursive", e indica di applicare le modifiche in modo ricorsivo a tutte le directory e sottodirectory.
- **777:** Questo è il set completo di permessi, che consente la lettura (4), la scrittura (2) e l'esecuzione (1) per il proprietario, il gruppo e gli altri utenti. Quindi, 7 corrisponde a tutti e tre i permessi.

```
(root@kali)-[/var/www/html]
# cd DVWA/config

(root@kali)-[/var/www/html/DVWA/config]
#
```

è utilizzato per spostarsi nella sottodirectory "config" all'interno del percorso corrente. In questo caso, presumibilmente, eseguito il comando precedente per spostarti nella directory principale del progetto DVWA e ora stai entrando nella sottodirectory "config".

```
(root@kali)-[/var/www/html/DVWA/config]
# cp config.inc.php.dist config.inc.php

(root@kali)-[/var/www/html/DVWA/config]
#
```

è utilizzato per copiare un file. In questo caso, stai copiando il file "config.inc.php.dist" in un nuovo file chiamato "config.inc.php".

```
(root@kali)-[/var/www/html/DVWA/config]
# nano config.inc.php

# 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
```

è utilizzato per aprire il file "config.inc.php" con l'editor di testo Nano. Nano è un editor di testo leggero e intuitivo che può essere utilizzato dalla linea di comando.

Dopo aver eseguito questo comando, si aprirà l'editor Nano e si può visualizzare e modificare il contenuto del file "config.inc.php", come nell'immagine.

```
(root@kali)-[/var/www/html/DVWA/config]
# sudo mysql -u root -p

Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.11.6-MariaDB-1 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)]> grant all privileges on dvwa.* to 'kali'@'127.0.0.1' IDENTIFIED BY 'kali' ;
Query OK, 0 rows affected (0.018 sec)

MariaDB [(none)]> create user 'kali'@'127.0.0.1' identified by 'kali' ;
ERROR 1396 (HY000): Operation CREATE USER failed for 'kali'@'127.0.0.1'
MariaDB [(none)]> create user 'kali'@'127.0.0.1' identified by 'kali' ;
ERROR 1396 (HY000): Operation CREATE USER failed for 'kali'@'127.0.0.1'
MariaDB [(none)]> exit
Bye

(root@kali)-[/var/www/html/DVWA/config]
#
```

è utilizzato per connettersi al server MySQL come utente "root" con privilegi di amministratore. Vediamo cosa fa ciascuna parte del comando:

- **sudo:** Stands for "superuser do" ed è utilizzato per eseguire il comando successivo con privilegi di amministratore o utente root. In questo caso, stai cercando di eseguire il comando MySQL con privilegi di amministratore.
- **mysql:** Questo è il client della riga di comando per MySQL. Consente di interagire direttamente con il server MySQL dalla shell.
- **-u root:** Specifica l'utente "root" come utente MySQL con cui connettersi.
- **-p:** Indica di richiedere la password dopo l'esecuzione del comando. Una volta che premi "Enter" dopo aver inserito il comando, ti verrà chiesto di inserire la password dell'utente "root" di MySQL per autenticarti.

In pratica, questo comando ti consente di accedere all'interfaccia della riga di comando di MySQL con privilegi di amministratore, consentendoti di eseguire

comandi SQL, gestire database, utenti e altre attività amministrative direttamente dal terminale.

creiamo un'utenza sul db con il seguente comando create user 'kali'@'127.0.0.1' identified by 'kali' ; successivamente assegniamo i privilegi all'utente kali con il seguente comando: grant all privileges on dwwa.* to 'kali'@'127.0.0.1' identified by 'kali' ;

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

```
(root@kali)-[/var/www/html/DVWA/config]  
# service apache2 start  
  
(root@kali)-[/var/www/html/DVWA/config]  
# cd /etc/php/8.2/apache2  
  
(root@kali)-[/etc/php/8.2/apache2]  
# service apache2 start  
  
(root@kali)-[/etc/php/8.2/apache2]  
# service mysql start  
  
(root@kali)-[/etc/php/8.2/apache2]  
#
```

Ho completato la configurazione del servizio MySQL e ora sto configurando il servizio Apache (web server):

1. Ho avviato Apache con `service apache2 start`.
2. Mi sono spostato nella directory di configurazione PHP con `cd /etc/php/8.2/apache2` (verificando la versione di PHP se necessario).
3. Nel file `php.ini`, ho modificato le voci `allow_url_fopen` e `allow_url_include`.
4. Infine, ho riavviato Apache con `service apache2 start` per applicare le modifiche al file `php.ini`.

Questi passaggi dovrebbero contribuire a configurare correttamente il servizio Apache dopo la configurazione di MySQL.

[Setup DVWA](#)[Instructions](#)[About](#)

Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: `/var/www/html/DVWA/config/config.inc.php`

If the database already exists, **it will be cleared and the data will be reset.**
You can also use this to reset the administrator credentials ("**admin** // **password**") at any stage.

Setup Check

Web Server SERVER_NAME: **127.0.0.1**

Operating system: ***nix**

PHP version: **8.2.12**
PHP function display_errors: **Disabled**
PHP function display_startup_errors: **Disabled**
PHP function allow_url_include: **Enabled**
PHP function allow_url_fopen: **Enabled**
PHP module gd: **Installed**
PHP module mysql: **Installed**
PHP module pdo_mysql: **Installed**

Backend database: **MySQL/MariaDB**
Database username: **kali**
Database password: *********
Database database: **dvwa**
Database host: **127.0.0.1**
Database port: **3306**

reCAPTCHA key: **Missing**

Writable folder `/var/www/html/DVWA/hackable/uploads/`: **Yes**
Writable folder `/var/www/html/DVWA/config`: **Yes**

Status in red, indicate there will be an issue when trying to complete some modules.

If you see disabled on either `allow_url_fopen` or `allow_url_include`, set the following in your `php.ini` file and restart Apache.

```
allow_url_fopen = On
allow_url_include = On
```

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

Create / Reset Database

DVWA Security

Security Level

Security level is currently: **impossible**.

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'.

Username: admin
Security Level: impossible
Locale: en
SQLi DB: mysql

Damn Vulnerable Web Application (DVWA)

Ho appena completato la configurazione iniziale di DVWA (Damn Vulnerable Web Application) seguendo alcuni passaggi. Ecco cosa ho fatto:

1. **Configurazione di MySQL e Apache:**

- Ho configurato il servizio MySQL e avviato Apache per assicurarmi che DVWA funzionasse correttamente.

2. **Accesso alla Pagina di Setup:**

- Ho aperto il mio browser e ho digitato "127.0.0.1/DVWA/setup.php" nella barra degli indirizzi.

3. **Creazione/Reset del Database:**

- Sono stato reindirizzato a una pagina di setup di DVWA, dove ho cliccato su "Create / Reset Database". Questo passaggio ha creato il database necessario per DVWA.

4. **Accesso alla Pagina di Login:**

- Dopo la creazione del database, sono stato reindirizzato a una pagina di login. Ho usato le credenziali di default:

- **Username:** admin
- **Password:** password

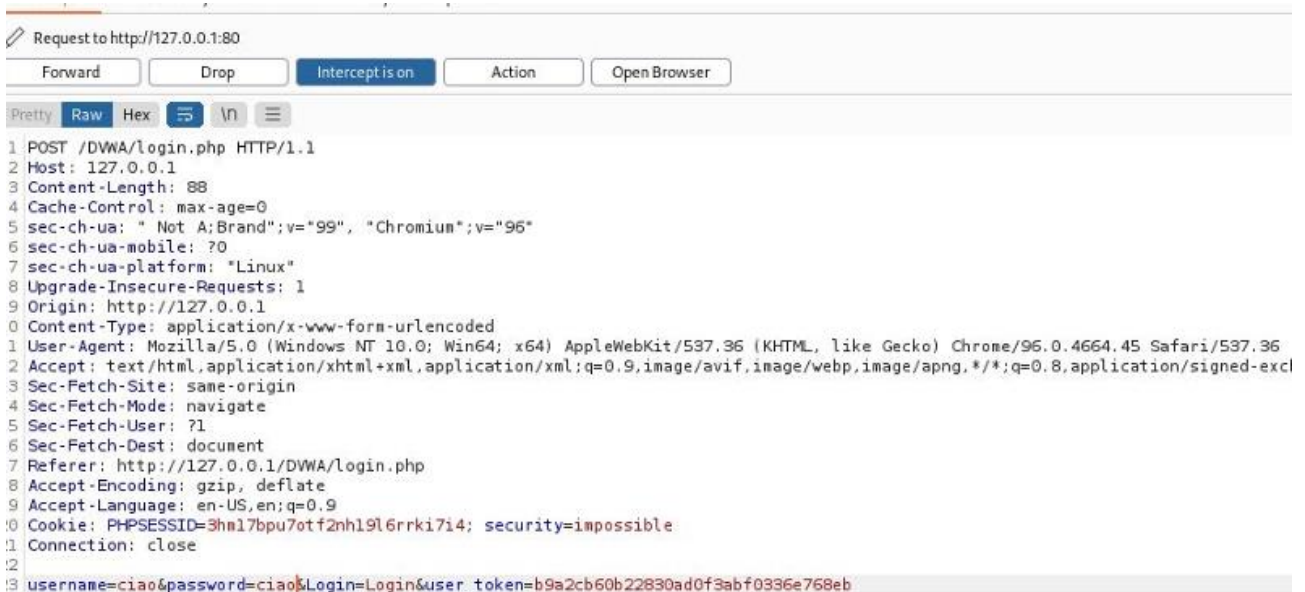
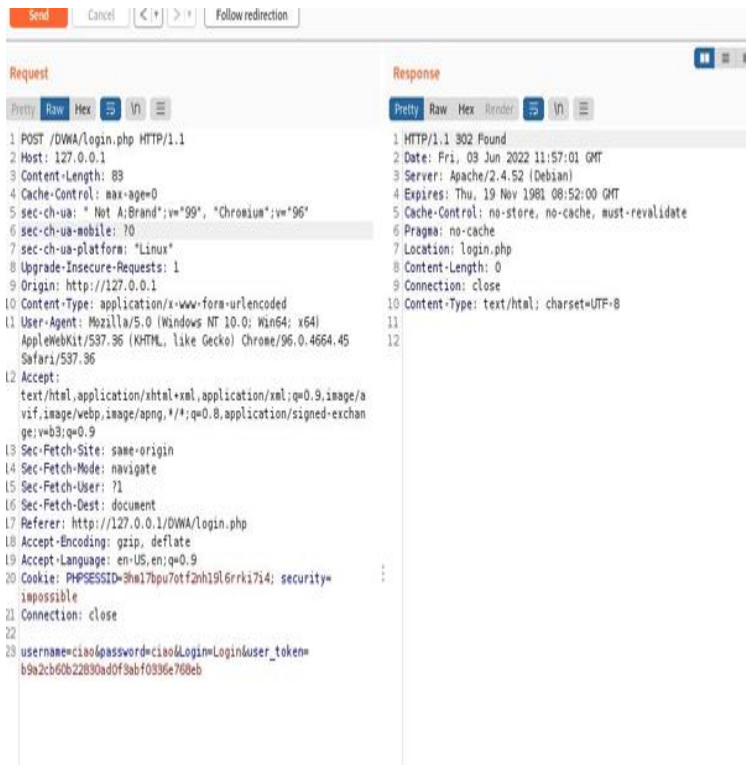
5. **Configurazione della Sicurezza:**

- All'interno dell'applicazione, ho cliccato sulla scheda "DVWA Security" e ho impostato il livello di sicurezza dell'app. Ho notato che un livello di sicurezza più basso semplifica la scoperta delle vulnerabilità.

6. **Esplorazione delle Vulnerabilità:**

- Sulla colonna a sinistra, ho visto un elenco di tutte le vulnerabilità presenti. Queste saranno il mio focus durante la fase successiva di exploit.

Questo setup mi ha fornito l'accesso a DVWA, un'applicazione che contiene intenzionalmente vulnerabilità. Ora sono pronto a esplorare e praticare le tecniche di sicurezza informatica.



Durante la sessione pratica con Burp Suite, ho eseguito diverse attività di test sulla nostra app, DVWA. Dopo aver avviato Burp Suite, ho selezionato un progetto temporaneo e aperto il browser per accedere a DVWA all'indirizzo 127.0.0.1/DVWA. Successivamente, ho inserito le credenziali di accesso (admin/password) e ho intercettato la richiesta con Burp.

Ho esaminato i parametri di login e ho sperimentato la modifica di tali parametri prima di inviare la richiesta all'applicazione. Ho anche tentato di inserire

credenziali errate e successivamente ho inviato la richiesta al repeater di Burp per ulteriori analisi.

Dalla risposta HTTP, ho confermato che l'accesso con credenziali errate è stato negato, come indicato nel corpo della risposta con il messaggio "Login failed". Questa attività ha sottolineato l'importanza della pratica e della conoscenza degli strumenti, come Burp Suite, per esplorare e comprendere il comportamento dell'applicazione in scenari di test.

Infine, ho evidenziato che la pratica continua è essenziale e che, nelle future lezioni sugli exploit delle web app, esamineremo approfonditamente i metodi per ottenere accesso a aree riservate sfruttando errori comuni di configurazione e programmazione.