S3/L3 Esercizio

Preparazione Ambiente

Data la traccia:

“Nella lezione pratica di oggi vedremo come configurare una DVWA – ovvero damn vulnerable web application in Kali Linux. La DVWA ci sarà molto utile per i nostri test.

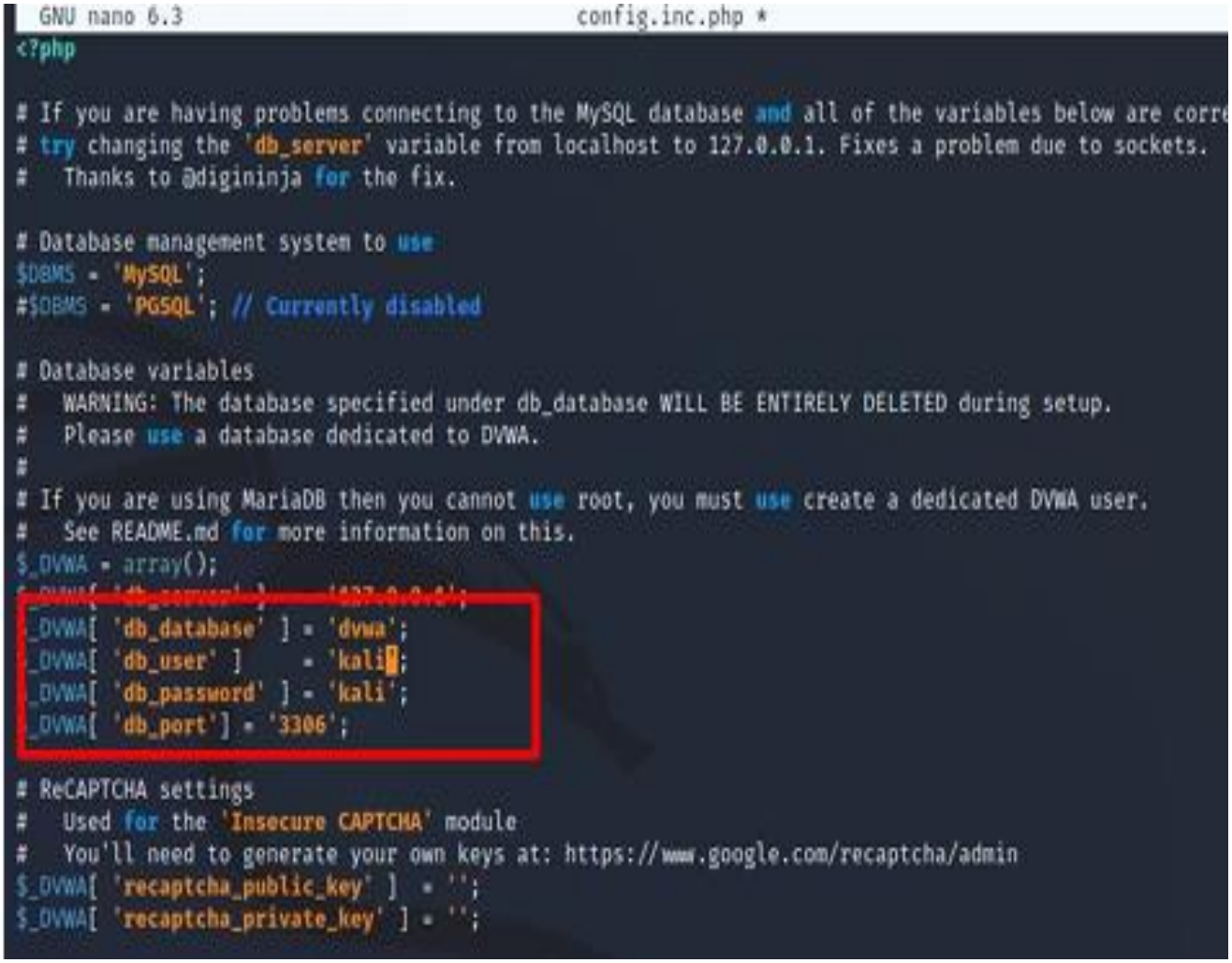
Per installare la DVWA abbiamo bisogno di 3 componenti:

- OS Kali Linux OK.
- Database MySQL (da installare).
- Web Server Apache (da installare).”

Apriamo un terminale su Kali, utilizziamo l’utenza di root, eseguendo il comando «sudo su» e poi eseguiamo i comandi seguenti:

- cd /var/www/html
- git clone https://github.com/digininja/DVWA
- chmod R 777 DVWA/
- cd DVWA/config
- cp config.inc.php.dist config.inc.php
- nano config.inc.php All’interno del file config.inc.php cambiamo utente e password di default come da figura sotto (inserendo, user:kali, password:kali).

Salviamo con la solita sequenza di tasti «ctrl+o» poi «invio» e usciamo facendo «ctrl+x»



Sempre con utenza di root su Kali, facciamo partire il servizio mysql con il comando:

service mysql start

poi connettiamoci al db con utenza di root con il comando seguente:

mysql –u root –p

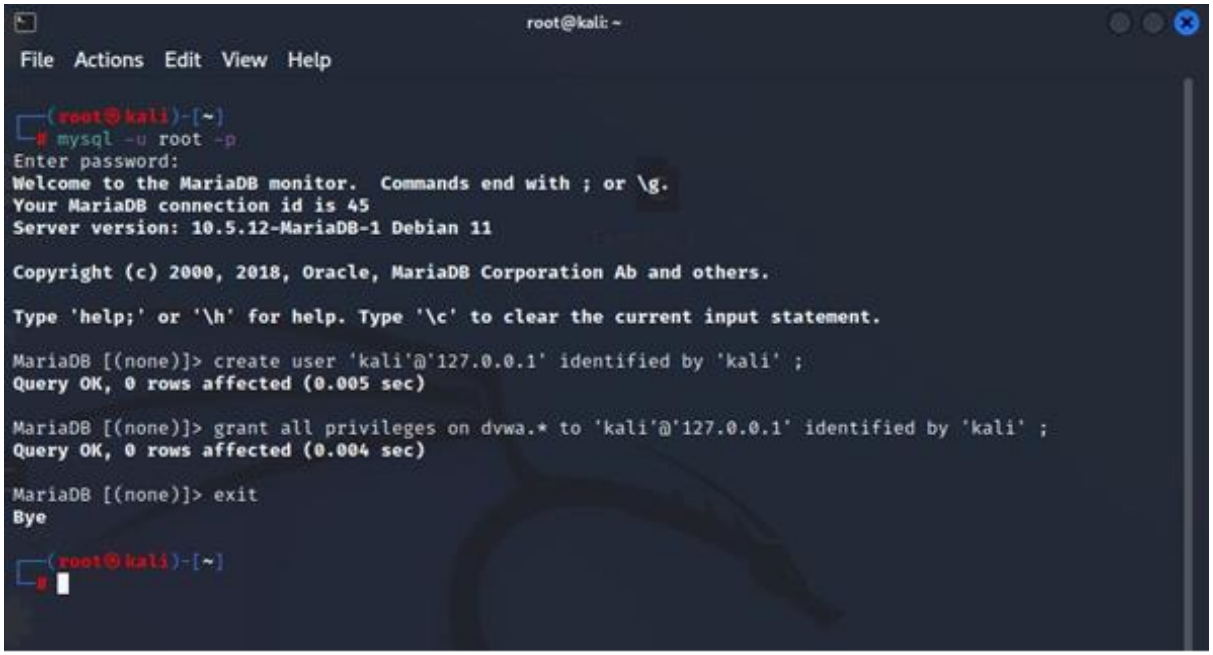
Creiamo un’utenza sul db con il seguente comando:

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 dvwa.* to 'kali'@'127.0.0.1' identified by 'kali’ ;

ed usciamo utilizzando “exit”.



Ora che il servizio mysql è configurato, passiamo al servizio apache (il web server).

Facciamo partire il servizio con il comando:

service apache2 start

Poi, ci spostiamo nella cartella /etc/php/8.1/apache2 con il comando:

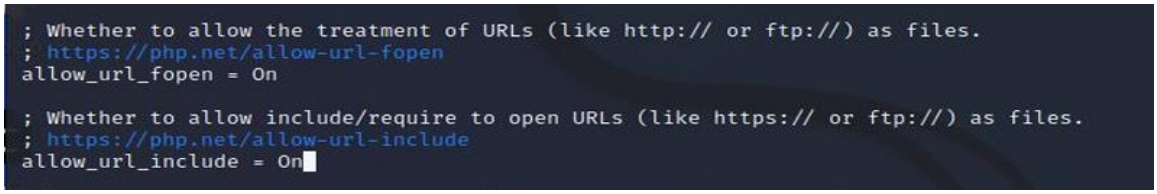
cd /etc/php/8.1/apache2

Successivamente, andremo ad utilizzare l'editor "nano" per modificare il file php.ini all'interno della cartella apache2.

Modifichiamo le voci allow-url-fopen e allow-url-include come sotto.

Eseguiamo nuovamente il comando:

service apache2 start



A questo punto apriamo una sessione del nostro browser e scriviamo nella barra degli indirizzi: 127.0.0.1/DVWA/setup.php

Clicchiamo su «Create / Reset Database» nella parte bassa della pagina.

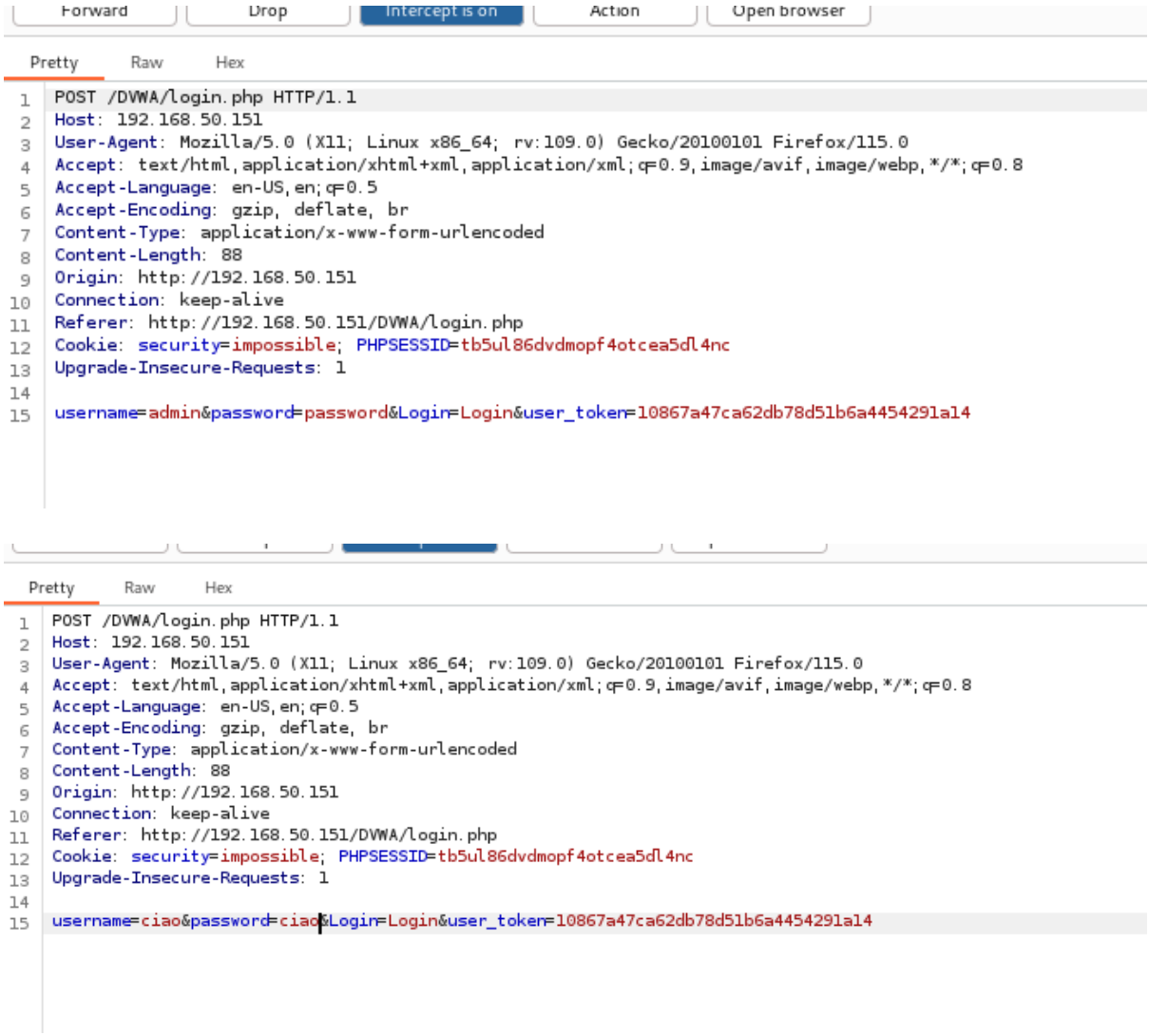
Ora che la nostra app è attiva, facciamo un po' di pratica con Burpsuite con la traccia seguente:

"Esercizio Web App Lanciamo Burpsuite, scegliamo un progetto temporaneo ed apriamo un browser, inserendo l'indirizzo della nostra DVWA 127.0.0.1/DVWA e inseriamo nei campi login e password i valori «admin» e «password» rispettivamente. Intercettiamo la richiesta con burp e vediamo come possiamo modificarla. Guardate i parametri di login, possiamo modificarli a nostro piacimento prima di inviare la richiesta all'app."

Accediamo a Burpsuite, una volta entrati selezioniamo "proxy". Attiviamo l'intercettazione.

Una volta nella pagina di login, aperta dal nostro browser proxy (in questo caso abbiamo cambiato le impostazioni su "Firefox"), raggiunta tramite l'inserimento nell'URL del nostro IP ossia "192.168.50.151"+/DVWA" inserendo "User" e "Password" la pagina rimarrà in caricamento fino a quando non gli diamo l'input "forward".

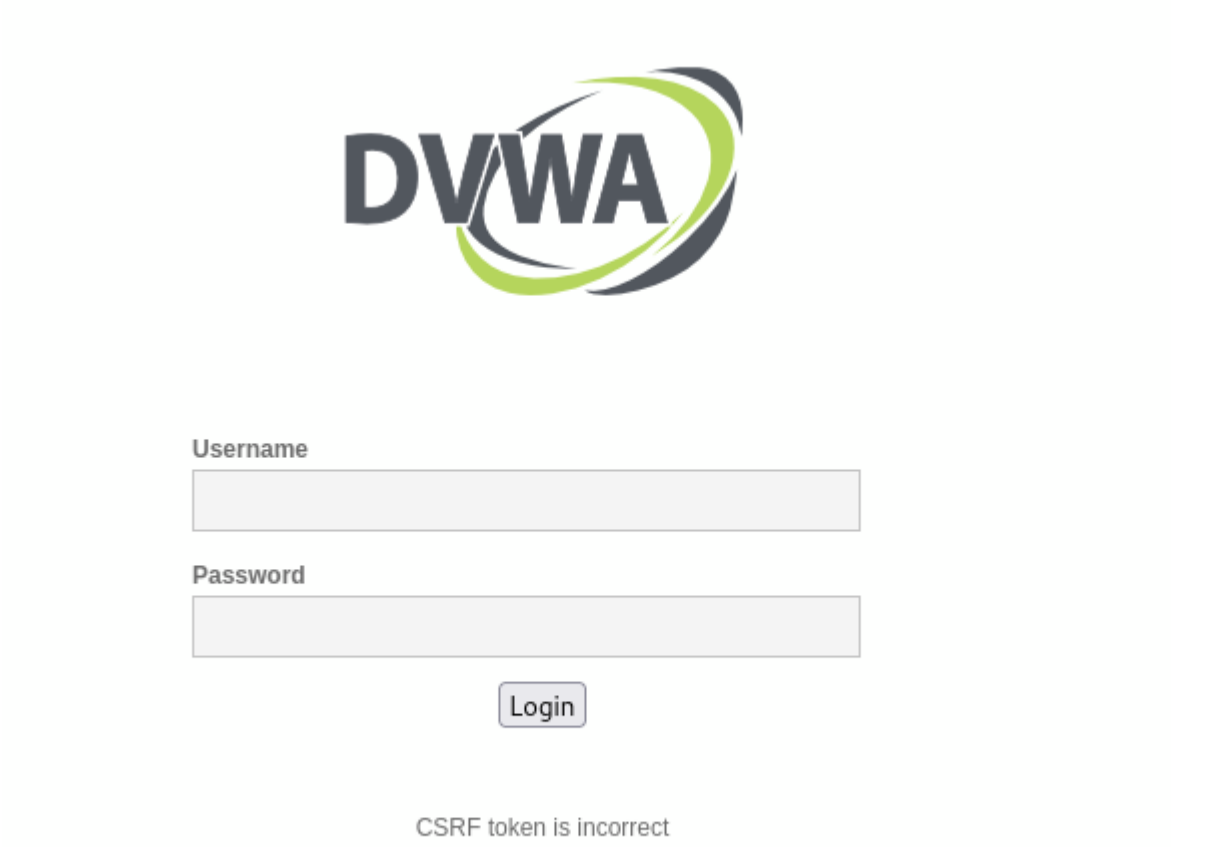
Nell’immagine che riporterò di seguito constateremo che possiamo fare delle modifiche all’ “User” e alla “Password” che abbiamo inserito, modificando anche il livello di sicurezza da “impossible” a “low”:



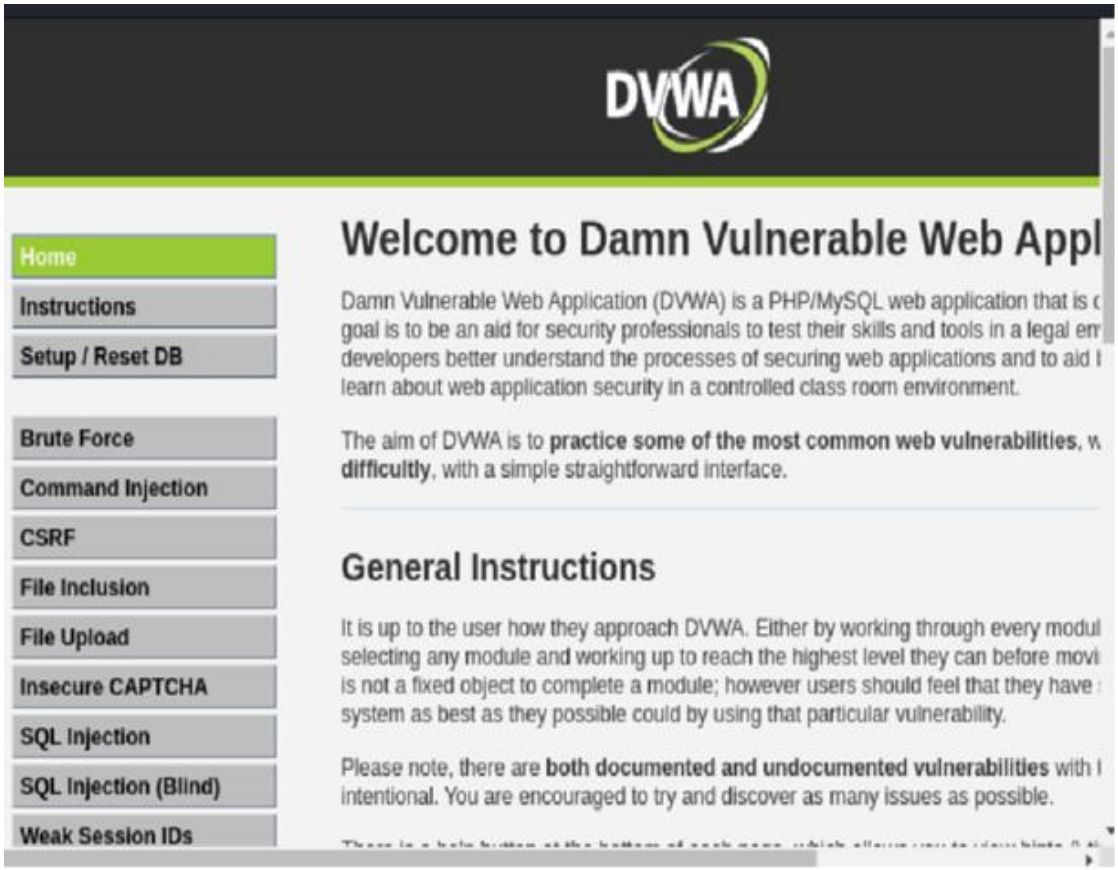
Dopo aver effettuato le modifiche e aver seguito i passaggi dati dalla consegna dell’esercizio, (quindi prima di inviare la richiesta, clicchiamo con il tasto destro e selezioniamo «send to repeater» Clicchiamo su send per inviare la richiesta di login ed e poi su follow redirection) il risultato sarà il seguente:

```
    </p>

    <br />

  </div>
   <!--<div id="header">-->

  <div id="content">

    <form action="login.php" method="post">

      <fieldset>

        <label for="user">
          Username
        </label>
         <input type="text" class="loginInput" size="20" name="username">
        <br />


        <label for="pass">
          Password
        </label>
         <input type="password" class="loginInput" AUTOCOMPLETE="off" size="20" name="password">
        <br />

        <br />

        <p class="submit">
          <input type="submit" value="Login" name="Login">
        </p>

      </fieldset>

      <input type='hidden' name='user_token' value='f180f20f74c0d3a0ad41267b51869f7b' />

    </form>

    <br />

    <div class="message">
      Login failed
    </div>

    <br />
    <br />
```

Chiaramente sul nostro browser proxy avremo questo risultato:



Se invece mettessimo le credenziali corrette, accederemmo correttamente alla pagina e questo è quello che visualizzeremmo:


DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
Welcome to Damn Vulnerable Web Appl
Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is d
goal is to be an aid for security professionals to test their skills and tools in a legal en
developers better understand the processes of securing web applications and to aid t
learn about web application security in a controlled class room environment.
The aim of DVWA is to practice some of the most common web vulnerabilities, w
difficultly, with a simple straightforward interface.
General Instructions
It is up to the user how they approach DVWA. Either by working through every modul
selecting any module and working up to reach the highest level they can before movi
is not a fixed object to complete a module; however users should feel that they have
system as best as they possible could by using that particular vulnerability.
Please note, there are both documented and undocumented vulnerabilities with t
intentional. You are encouraged to try and discover as many issues as possible.