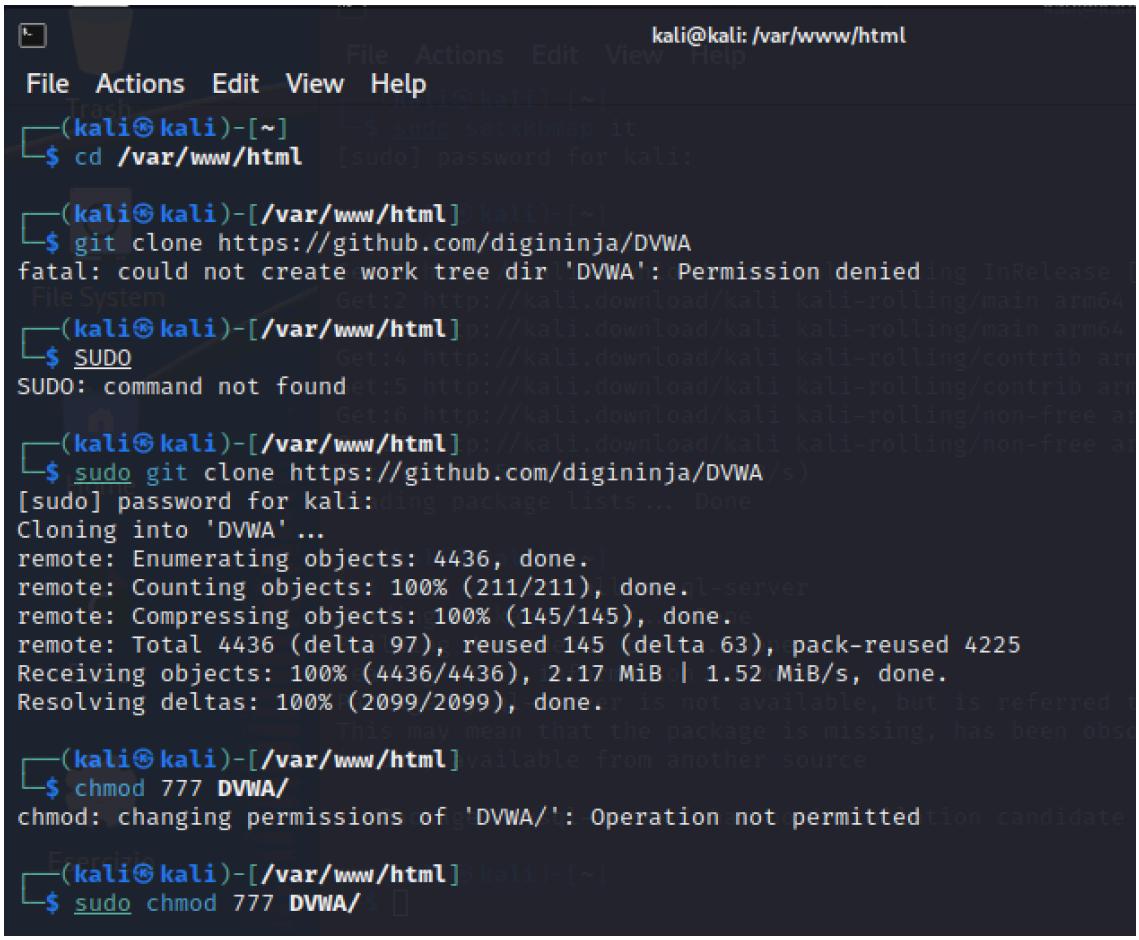
Traccia: Nella lezione pratica di oggi vedremo come configurare una DVWA – ovvero damn vulnerable web application in Kali Linux. La DVWA ci sarà molto utile per i nostri test sia durante la build week 1 che durante lo sviluppo del modulo 2, dove vedremo da vicino le tecniche per sfruttare le vulnerabilità nella fase di exploit.



Con il primo comando ci spostiamo in **/var/www/html:** una directory di sistema comune in molte distribuzioni Linux, spesso utilizzata per memorizzare file e dati relativi ai siti web. Nella maggior parte dei casi, i file dei siti web vengono posizionati in questa directory quando si utilizzano server web come Apache.

Il comando **git clone** è utilizzato per clonare un repository Git da una determinata URL remota. Ho cercato di clonare il repository DVWA (Damn Vulnerable Web Application) da GitHub.

https://github.com/digininja/DVWA: è l'URL del repository su GitHub che si desidera clonare. In questo caso, si tratta del repository DVWA di Digininja.

777: è una modalità di impostazione dei permessi in cui tutti gli utenti (proprietario, gruppo e altri) hanno i permessi di lettura, scrittura ed esecuzione sulla directory DVWA.
DVWA/: è la directory su cui sto modificando i permessi.

```
┌──(kali㉿kali)-[/var/www/html]
└─$ cd DVWA/config

┌──(kali㉿kali)-[/var/www/html/DVWA/config]
└─$ cp config.inc.php.dist config.inc.php
cp: cannot create regular file 'config.inc.php': Permission denied

┌──(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
```

Questi comandi si riferiscono alla gestione dei file di configurazione del DVWA (Damn Vulnerable Web Application):
-Cambio la directory corrente alla cartella config all'interno della directory di configurazione del DVWA che ho clonato prima.
-Poi faccio una copia del file di configurazione predefinito rinominandolo e apro il nuovo file creato (**config-inc.php**) per modificarne il contenuto tramite l'editor di testo **nano**.
Questo è un processo comune per configurare applicazioni web come DVWA. Di solito, creando una copia del file di configurazione di default, è possibile personalizzare le impostazioni di configurazione senza modificare direttamente o sovrascrivere il file originale.

Ho poi modificato questo:

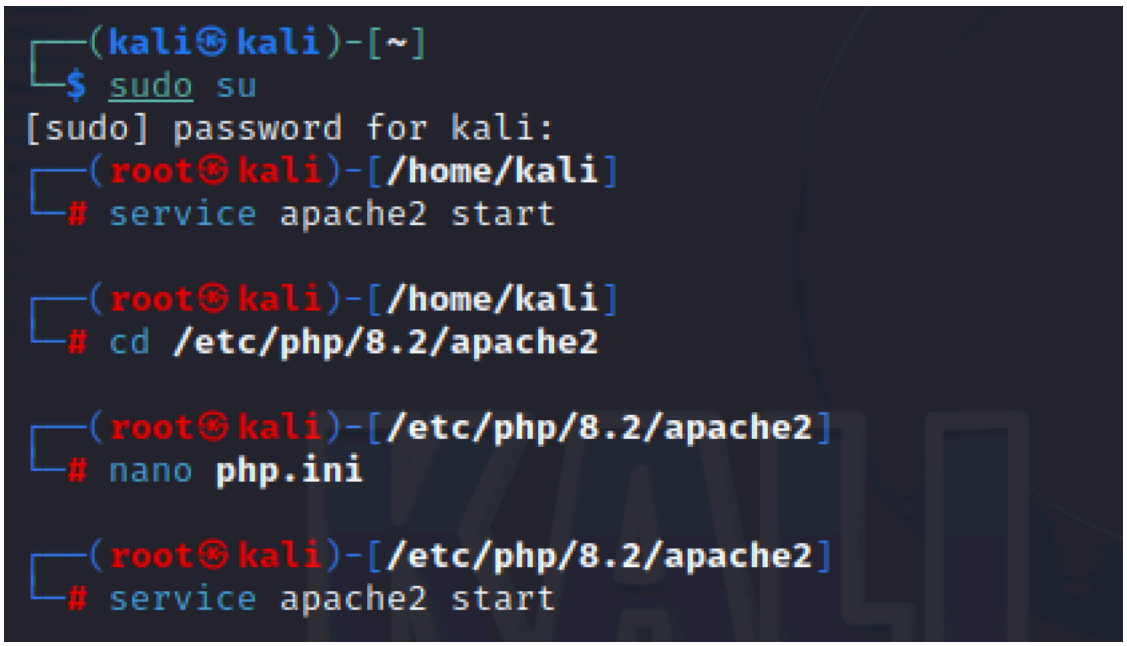
```
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ]     = 'kali';
$_DVWA[ 'db_password' ] = 'kali';
```

```
┌──(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: 10.11.4-MariaDB-1 Debian 12

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 (0.004 sec)

MariaDB [(none)]> grant all privileges on dvwa.* to 'kali'@'127.0.0.1' identified by 'kali' ;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> exit
Bye

┌──(kali㉿kali)-[~]
└─$
```

Il comando **service mysql start** è utilizzato per avviare il servizio MySQL su un sistema Linux.

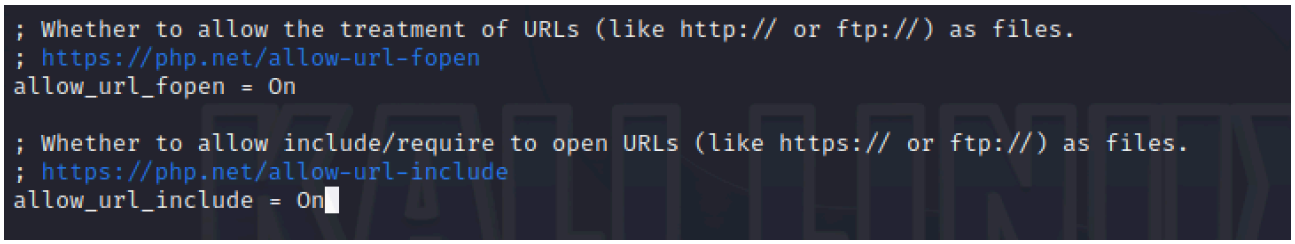
Avvio il servizio MySQL, consentendo al database di essere attivo e pronto per accettare connessioni e gestire le richieste di dati.

Poi accedo al client MySQL come utente **root,** per interagire direttamente con il database e apportare modifiche.

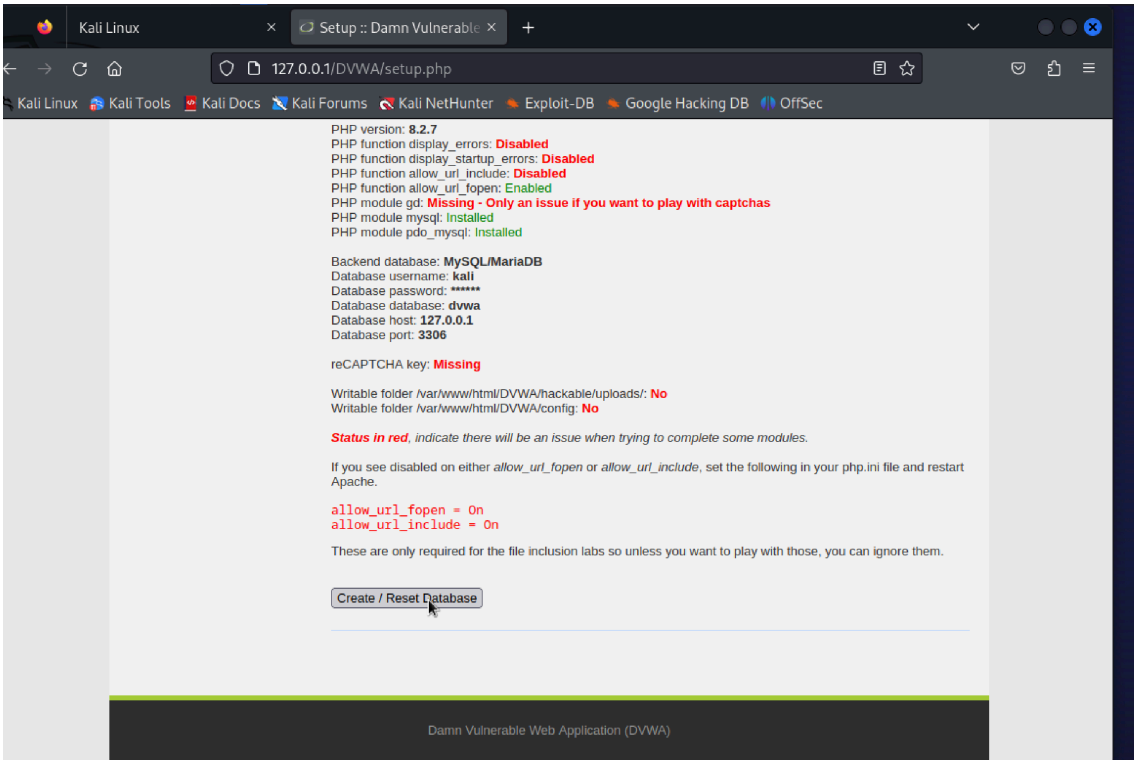


Questi sono i comandi per configurare le impostazioni PHP su un server che utilizza Apache come web server e PHP come linguaggio di scripting:
-Avvio il server Apache (**service apache2 start**),
-accedo alla directory di configurazione di PHP per Apache,
-apro il file **php.ini** per modificarlo e
-riavvio il server Apache (**service apache2 start**) per applicare eventuali modifiche effettuate alla configurazione di PHP.
Questo processo è comune quando si desidera modificare le impostazioni di configurazione di PHP per far funzionare un'applicazione web specifica o per ottimizzare le prestazioni del server.

Ho poi configurato su **on** le voci allow-url-fopen e allow-url-include:

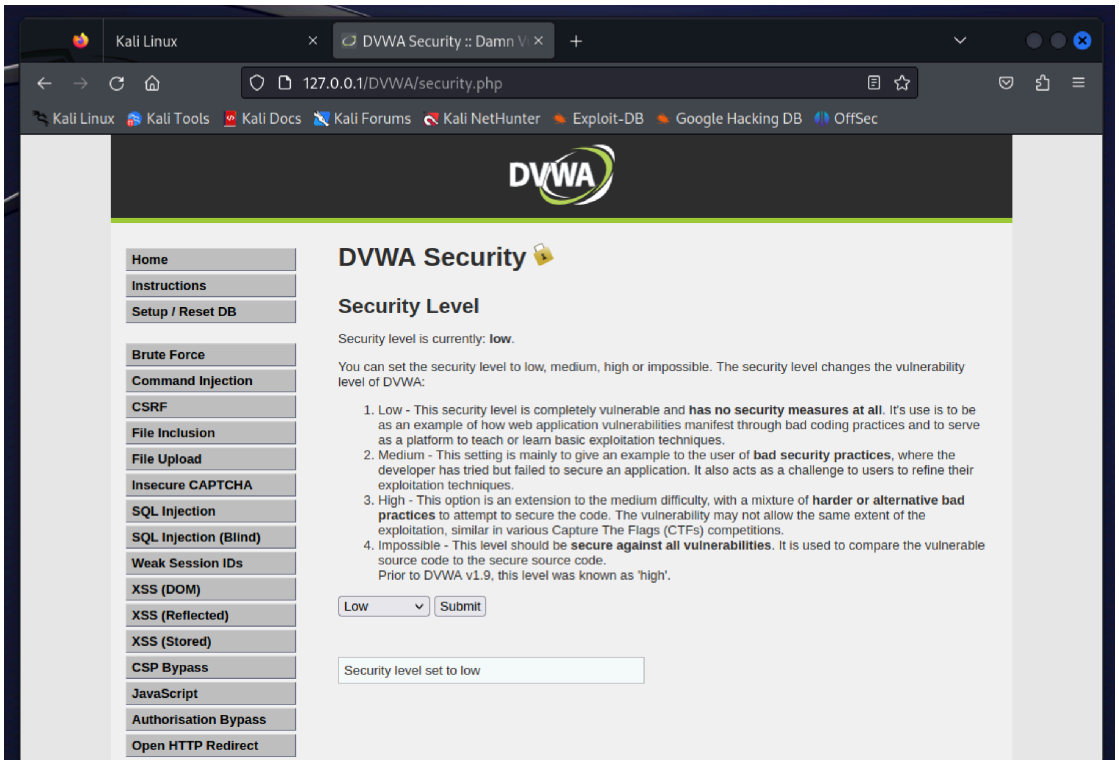


Apro una sessione del browser e scrivo nella barra degli indirizzi: **127.0.0.1/DVWA/setup.php** e si apre questa pagina:

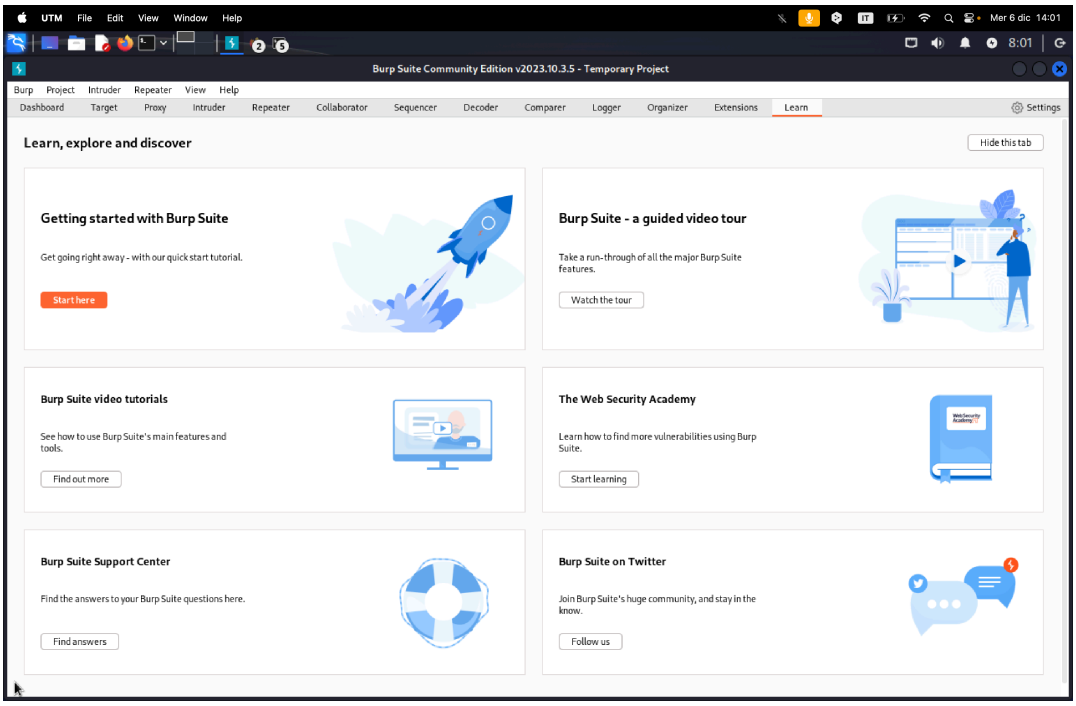


Clicco su Create/Reset Database

Poi scelgo il livello di sicurezza dell'app: più **basso** sarà il livello di sicurezza impostato, meno sarà complicato sfruttare le vulnerabilità.

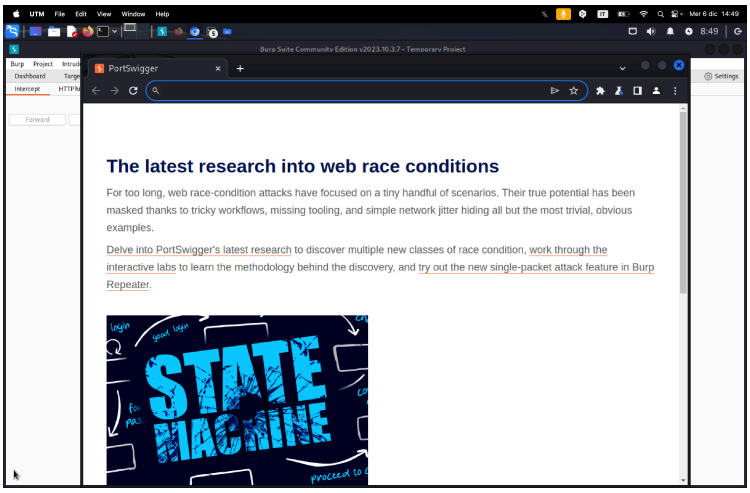


Lancio **Burp Suite**



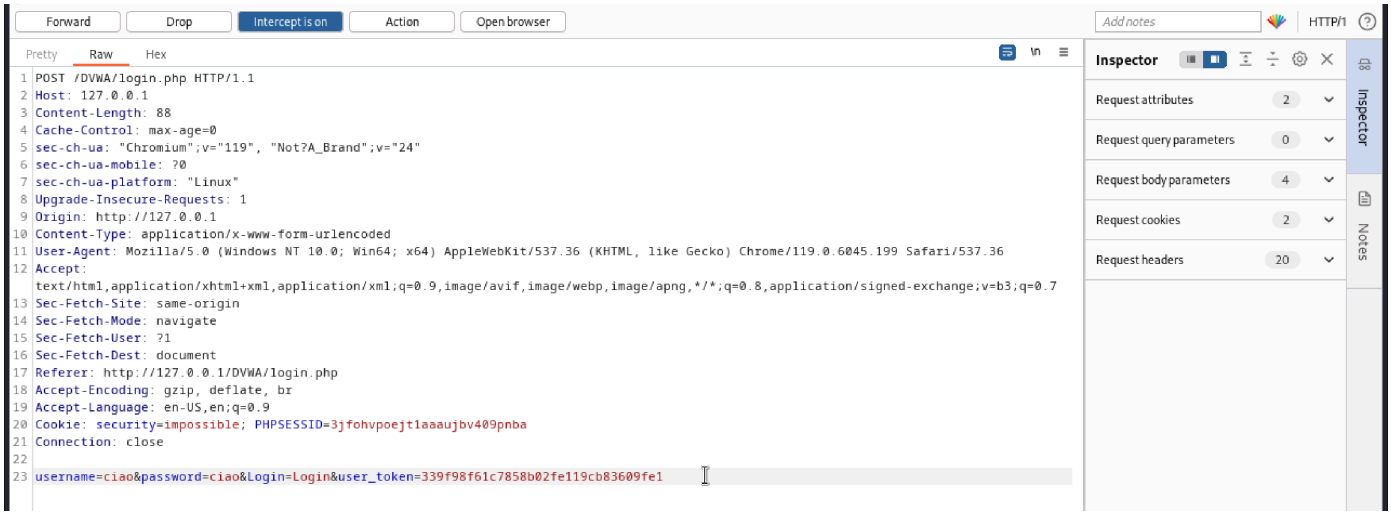
Apro il browser ed inserisco l'indirizzo della mia DVWA: **1270.0.1/DVWA**

Visualizzo questa pagina:



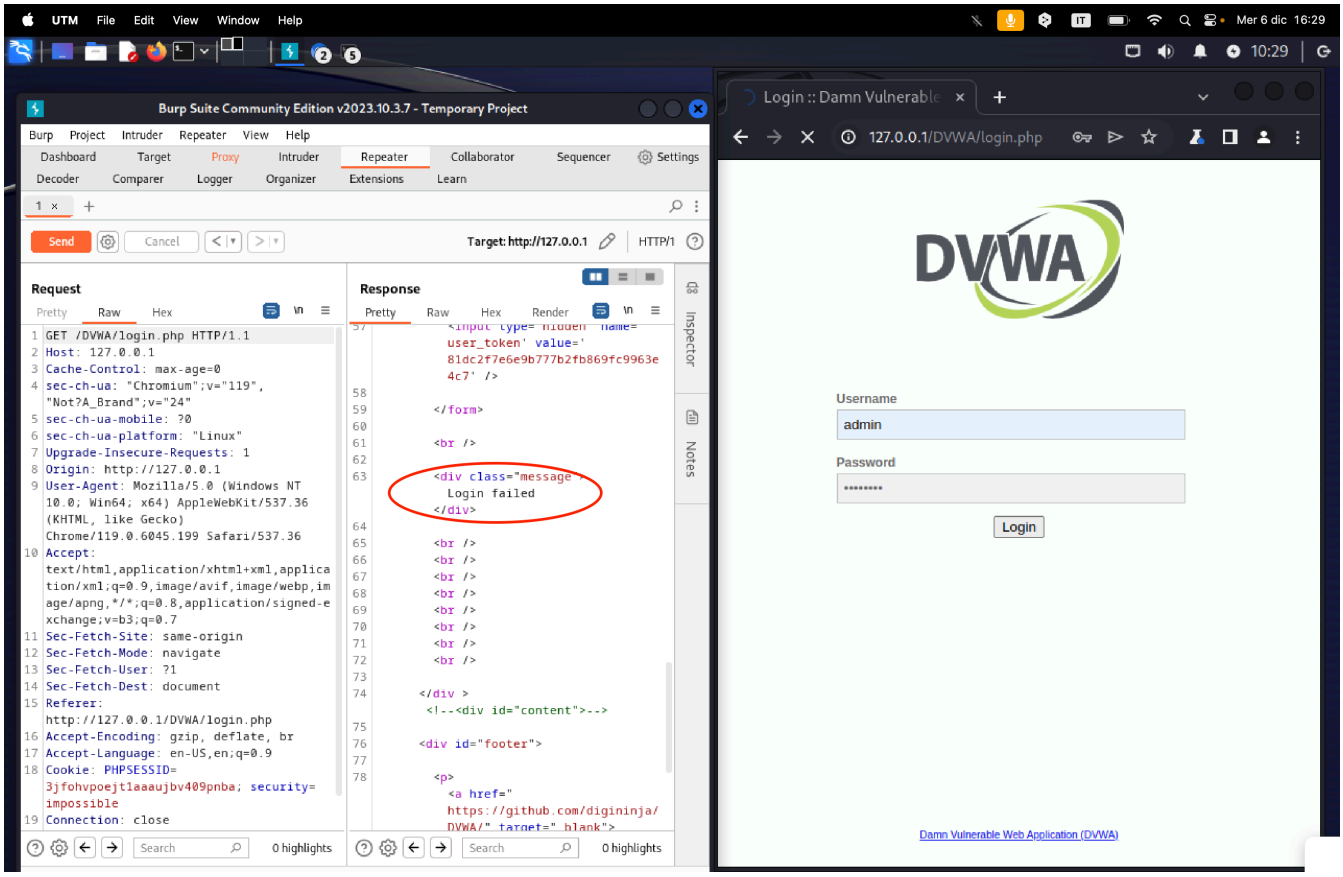
Solo dopo aver cliccato due volte su **Forward** in burp -> la pagina viene sbloccata e riesco ad inserire username e password.

Poi intercettiamo la richiesta con burp e vediamo come possiamo modificarla (mettendo ad esempio ciao come username e password).



Prima di inviare la richiesta all'app, clicco con il tasto destro e seleziono «**send to repeater**». Clicco su **send** per inviare la richiesta di login e poi su **follow redirection**.

Avendo provato ad accedere inserendo credenziali errate (diverse da ciao ciao che ho modificato prima), ottengo questo:



Notiamo quindi che non riesco ad entrare: nel body della http response leggo «**Login failed**».