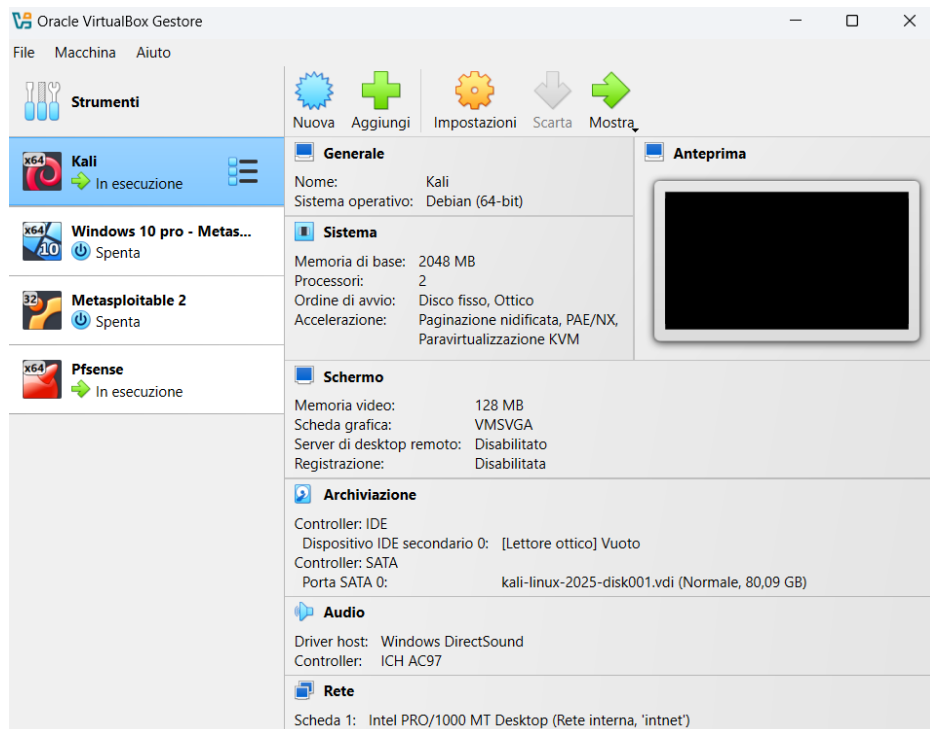
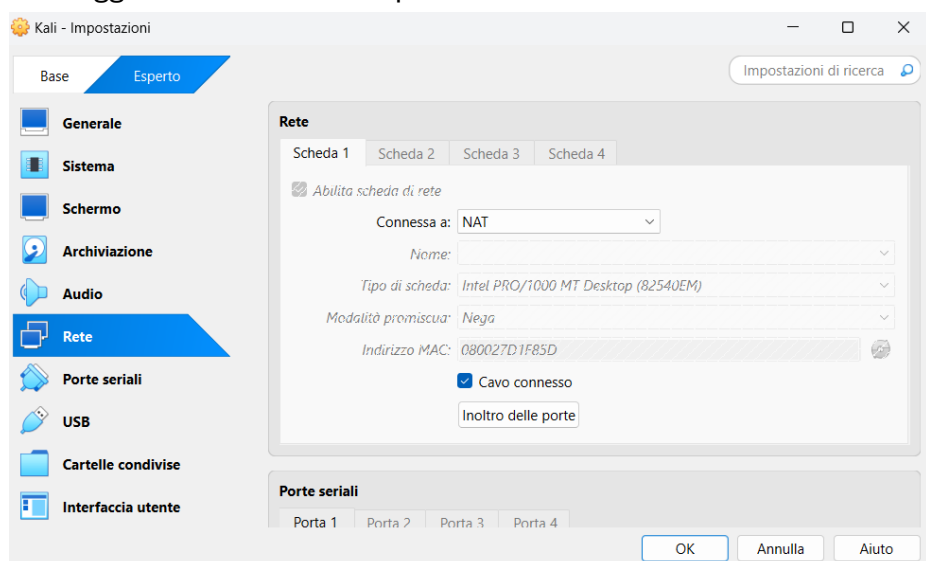


PRATICA S3L3:

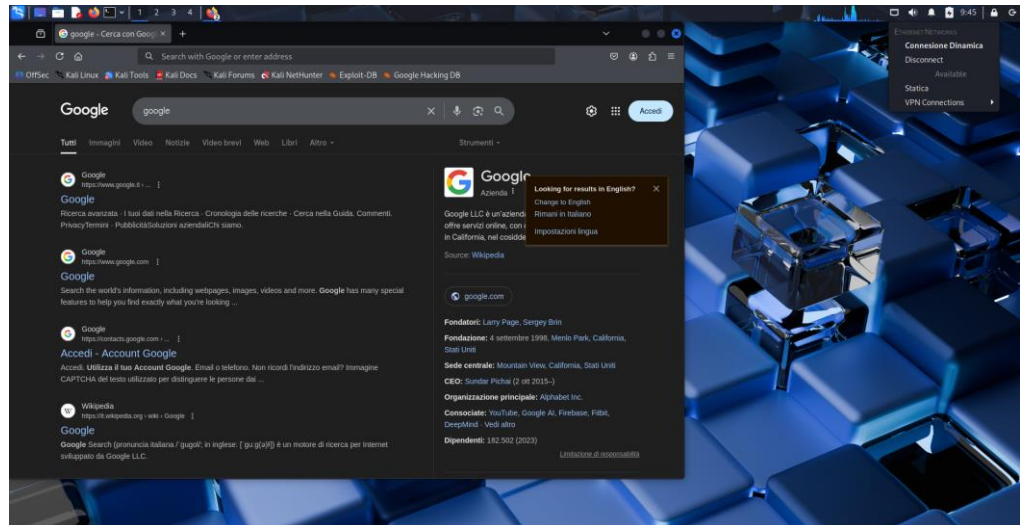
- **Traccia:** Nella lezione pratica di oggi vedremo come configurare una DVWA – ovvero damn vulnerable web application in Kali Linux. La DVWA ci sarà molto utile per i nostri test.
 - **Esecuzione:** Per installare la DVWA abbiamo bisogno di tre componenti:
 - **Kali Linux connesso a Internet:**
 - **Passaggio 1:** avviare kali linux su Oracle VirtualBox



- **Passaggio 2:** modificare l'impostazione di rete su “nat”



- **Passaggio 3:** Inserire la connessione dinamica e testarlo su una pagina web



■ **Database MySQL (da installare) e Web Server Apache (da installare):**

- Aprire un terminale su kali, utilizzare l'utenza di root, eseguendo il comando "sudo su" e poi eseguire i comandi seguenti:

- cd /var/www/html
- git clone <https://github.com/digininja/DVWA>
- chown -R www-data:www-data 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 (inserendo, user:kali, password:kali. Salvare con la solita sequenza di tasti «ctrl+x» poi «y».

```

root@kali: /var/www/html/DVWA/config
File Actions Edit View Help
GNU nano 6.3 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'] = '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 PHPIDS status
# The default is 'disabled'. You can set this to be either 'enabled' or 'disabled'.
$_DVWA['default_phpids_level'] = 'disabled';

# Verbose PHPIDS messages
# Enabling this will show why the WAF blocked the request on the blocked request.
# The default is 'disabled'. You can set this to be either 'true' or 'false'.
$_DVWA['default_phpids_verbose'] = 'false';

```

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



```
root@kali: ~  
File Actions Edit View Help  
(root@kali)~  
# service mysql start  
(root@kali)~  
# mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 44  
Server version: 10.5.12-MariaDB-1 Debian 11  
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)]>
```

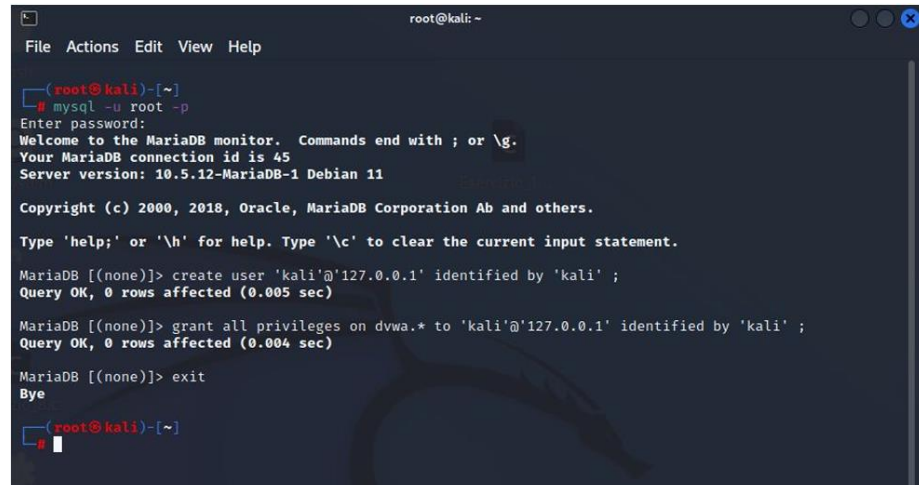
Creiamo un'utenza sul db con il seguente comando:

- 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”.



```
root@kali: ~  
File Actions Edit View Help  
(root@kali)~  
# mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 45  
Server version: 10.5.12-MariaDB-1 Debian 11  
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 (0.005 sec)  
MariaDB [(none)]> grant all privileges on dvwa.* to 'kali'@'127.0.0.1' identified by 'kali' ;  
Query OK, 0 rows affected (0.004 sec)  
MariaDB [(none)]> exit  
Bye  
(root@kali)~  
#
```

Ora che il servizio mysql è configurato, passiamo al servizio apache (il web server), spostatevi nella cartella /etc/php/8.2/apache2 con il comando:

- cd /etc/php/8.4/apache2

Utilizzate editor di testo “**nano**” per modificare il file php.ini all'interno della cartella apache2. Modificare le voci allow_url_fopen e allow_url_include come sotto (dovreste di default trovare la voce allow_url_include configurata ad OFF).

- sudo nano /etc/php/8.4/apache2/php.ini

```
(kali@kali)-[~]  
$ cd /etc/php/8.4/apache2  
  
(kali@kali)-[/etc/php/8.4/apache2]  
$ sudo nano /etc/php/8.4/apache2/php.ini  
[sudo] password for kali:
```

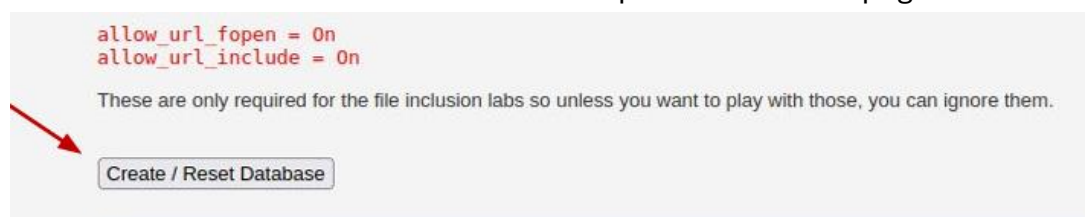
Avviare il sever apache:

- sudo a2enmod rewrite

A questo punto aprite una sessione del vostro browser e scrivete nella barra degli indirizzi:

- 127.0.0.1/DVWA/setup.php

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



Appena creato il database verrete rediretti su una pagina di login, dove potete entrare inserendo le credenziali di admin di default.

- Username: admin
- password: password



Username

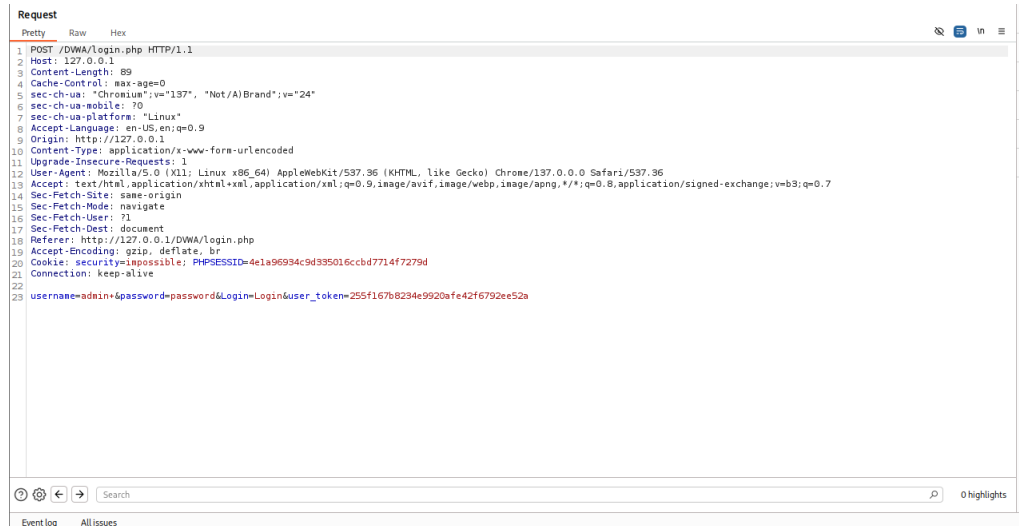
Password

••••••••

Login

▪ BURPSUITE:

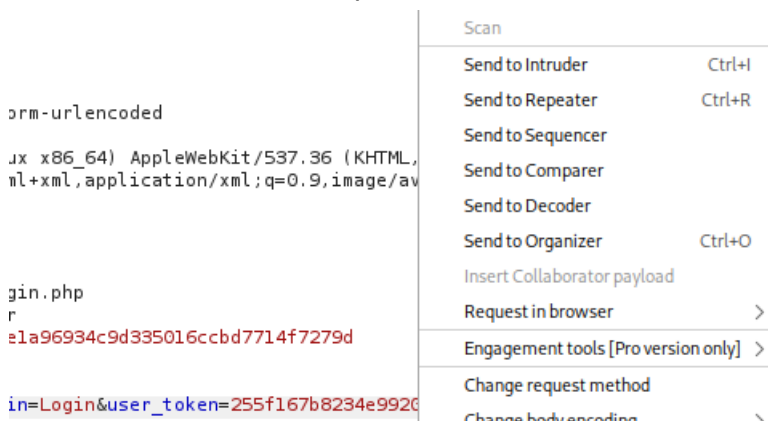
- Lanciamo Burpsuite, scegliamo un progetto temporaneo ed apriamo un browser, inserendo l'indirizzo della nostra DVWA: 127.0.0.1/DVWA e inseriamo nei campi login e password i valori «admin» e «password» rispettivamente. Intercettiamo la richiesta con burp e vediamo come possiamo modificarla.



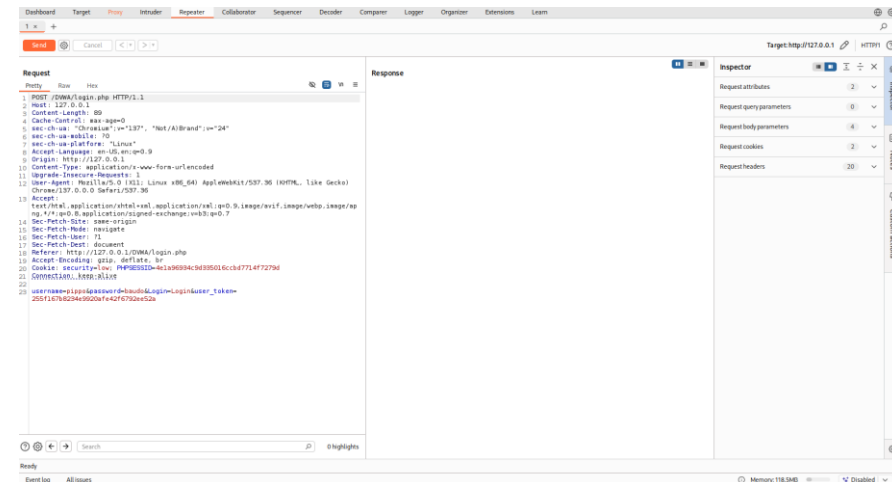
- Per prima cosa, in linea 20, modifichiamo il cookie “security=impossible” con “security=low”.
 Cookie: security=impossible; PHPSESSID=4e1a96934c9d335016ccbd7714f7279d
 Connection: keep-alive
- In linea 23 modifichiamo “username” e “password” con “pippo” e “baudo”.

```
23 username=pippo&password=baudo&Login=Login&user_token=255f167b8234e9920afe42f6792ee52a
```

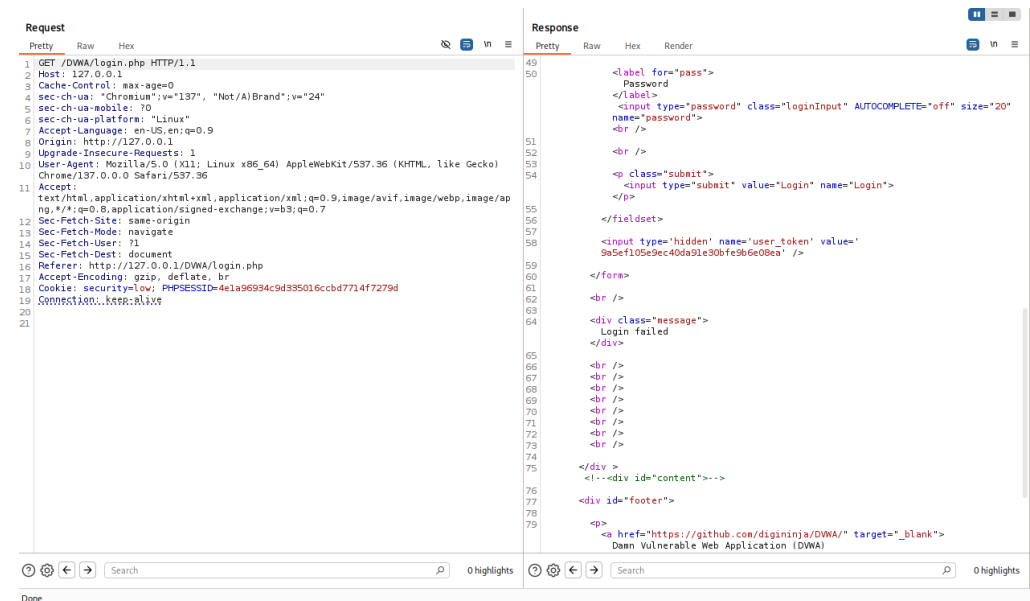
- Fatto ciò, lo inviamo al repeater.



- Clicchiamo su “send” per inviare la richiesta di login e poi su “follow redirection”.



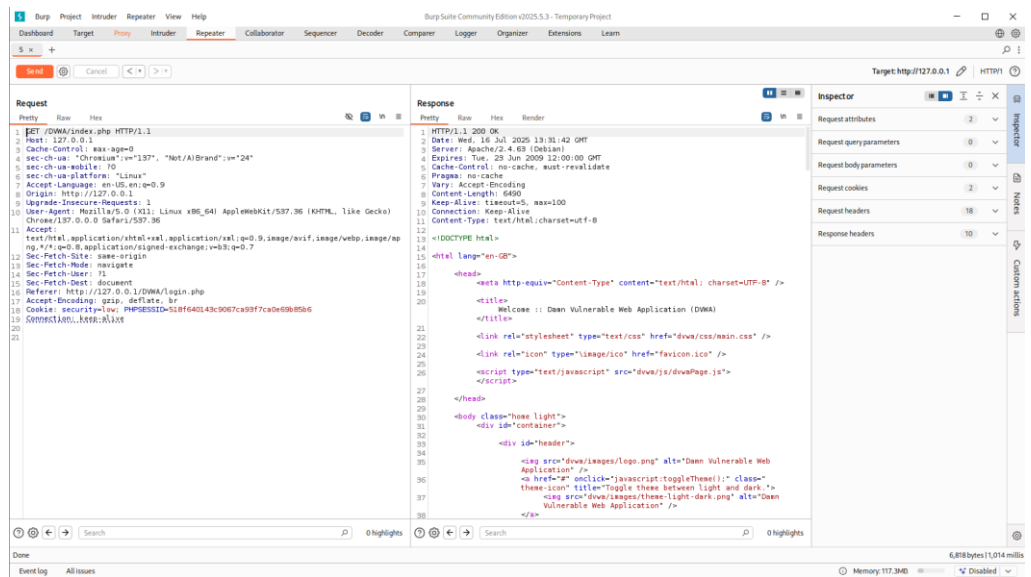
- Con le credenziali errate non riusciamo ad entrare. Ne abbiamo evidenza nel body della http response dove leggiamo “Login failed”.



- Invece, mantenendo security su low e le credenziali di accesso invariate otterremo l’accesso all’area riservata.

Cookie: security=low; PHPSESSID=518f640143c9067ca93f7ca0e69b85b6
Connection: keep-alive

username=admin&password=password&Login=Login&user_token=3b7693973970ec9c7cd5f40b5b2d7926



○ Considerazioni finali:

- Dopo aver configurato DVWA su Kali Linux, insieme al database MySQL e al server web Apache tramite terminale, ho potuto sperimentare direttamente l'installazione e l'avvio di un ambiente di test per la sicurezza. Lavorare con una web application volutamente vulnerabile come DVWA mi ha permesso di comprendere più a fondo le dinamiche degli attacchi informatici. Questa esperienza pratica rappresenta un passo fondamentale nel mio percorso di apprendimento nell'ambito dell'ethical hacking e della cybersecurity in generale