

CONSEGNA W8D2

Nel corso dell'esercizio abbiamo allestito su Kali Linux una piattaforma di test chiamata DVWA (Damn Vulnerable Web Application), pensata per sperimentare tecniche di penetration testing. L'obiettivo è quello di iniziare l'analisi e l'exploit delle vulnerabilità web imparando a configurare i servizi e a utilizzare strumenti di proxying come Burpsuite.

1. Iniziamo con l'installazione di DVWA e la configurazione del database.

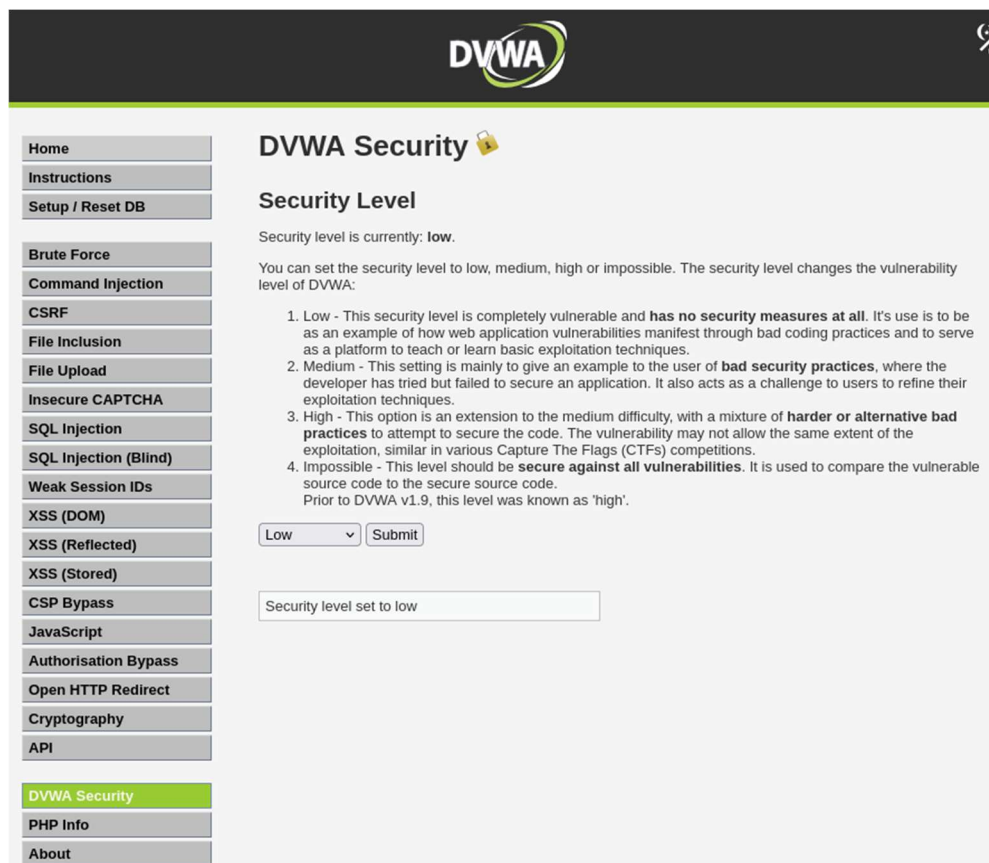
Dal terminale in root siamo entrati in `/var/www/html`, clonato il repository DVWA e regolati i permessi di scrittura. In seguito, abbiamo avviato MySQL, creato il database DVWA e l'utente kali con tutti i privilegi sullo schema dedicato. Questa separazione garantisce che eventuali reset o crash interessino solo l'applicazione vulnerabile, senza compromettere altri dati.

2. Successivamente procediamo la configurazione di Apache.

Avviato Apache2, abbiamo adattato il file `"php.ini"` attivando `"allow_url_fopen"` e `"allow_url_include"`, parametri essenziali per i laboratori di file inclusion. Il riavvio di Apache ha reso effettive le modifiche, assicurando che DVWA potesse eseguire include remoti e testare le vulnerabilità di inclusione di file.

3. Setup iniziale di DVWA.

Iniziamo con l'accedere a `"http://127.0.0.1/DVWA/setup.php"`, creando e resettando in questo modo il database con un click. Dopo la procedura di setup, con il login `"admin/password"` conferma il collegamento tra web server, PHP e database MySQL, successivamente andiamo ad entrare nell'interfaccia e scegliere il livello di sicurezza più adeguato, nel nostro caso andiamo ad impostare `"low"`.



The screenshot shows the DVWA Security Level configuration page. On the left is a sidebar with a menu of security challenges: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, Authorisation Bypass, Open HTTP Redirect, Cryptography, API, DVWA Security (highlighted), PHP Info, and About. The main content area is titled 'DVWA Security' with a lock icon. Below the title, it states 'Security level is currently: low.' and explains that the security level can be set to low, medium, high, or impossible. A list of four levels is provided: 1. Low (completely vulnerable), 2. Medium (bad security practices), 3. High (extension to medium difficulty), and 4. Impossible (secure against all vulnerabilities). At the bottom, there is a dropdown menu set to 'Low' and a 'Submit' button. Below the button, a message states 'Security level set to low'.

4. Analisi del Login con Burpsuite

Con Burpsuite in ascolto (Proxy -> Intercept on) abbiamo intercettato la richiesta “post” verso “/DVWA/login.php”, per poi esaminare l’header e body, e inoltrare la request al Repeater. Questa fase ha sottolineato l’importanza del token CSRF e dei cookie di sessione, evidenziando come modifiche ai valori di login portino subito a un “Login failed”. Il server ha ricevuto le credenziali non riconoscendole come valide. DVWA ci riporta alla pagina di login per riprovare.

The screenshot displays the Burp Suite interface with the 'Intercept' tab selected. A request to `http://127.0.0.1/DVWA/login.php` is shown in the 'Request' pane. The request is an HTTP POST with the following details:

- Method:** POST
- URL:** `http://127.0.0.1/DVWA/login.php`
- Headers:**
 - `Content-Type: application/x-www-form-urlencoded`
 - `Upgrade-Insecure-Requests: 1`
 - `User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36`
 - `Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7`
 - `Sec-Fetch-Site: same-origin`
 - `Sec-Fetch-Mode: navigate`
 - `Sec-Fetch-User: ?1`
 - `Sec-Fetch-Dest: document`
 - `Referer: http://127.0.0.1/DVWA/login.php`
 - `Accept-Encoding: gzip, deflate, br`
 - `Cookie: security=impossible; PHPSESSID=c7789f070c4317c6aa04e201e01bdd4e`
 - `Connection: keep-alive`
- Body:**

```
username=admin&password=password&Login=Login&user_token=681814f1c1397ba557f630b382571c92d
```

The 'Inspector' pane shows the selected text `username=admin&password=password` decoded from URL encoding. The 'Event log' pane shows the request and response details. The response is an HTTP 200 OK with the following details:

- Method:** GET
- URL:** `http://127.0.0.1/DVWA/login.php`
- Headers:**
 - `Content-Type: text/html; charset=utf-8`
 - `Server: Apache/2.4.18 (Ubuntu)`
 - `Set-Cookie: PHPSESSID=...; security=impossible`
- Body:**

```
<div id="content">
  <form action="/login.php" method="post">
    <input type="text" class="logininput" size="20" name="username">
    <input type="password" class="logininput" size="20" name="password">
    <input type="submit" value="Login" name="Login">
  </form>
  <div class="message">Login failed</div>
</div>
```