REPORT TECNICO

EXPLOIT FILE UPLOAD

Redatto da: *Nicolò Calì Cybersecurity Student*

Data: 12/01/2026

1. Introduzione

1.1 Obiettivo

L'attività d oggi consiste nello sfruttare una vulnerabilità di "File Upload" per ottenere esecuzione di codice remoto tramite una web shell.

1.2 Scopo e Perimetro

- **Target Autorizzato:** 192.168.50.101 – Metasploitable2

2. Ambiente di Lavoro e Strumenti

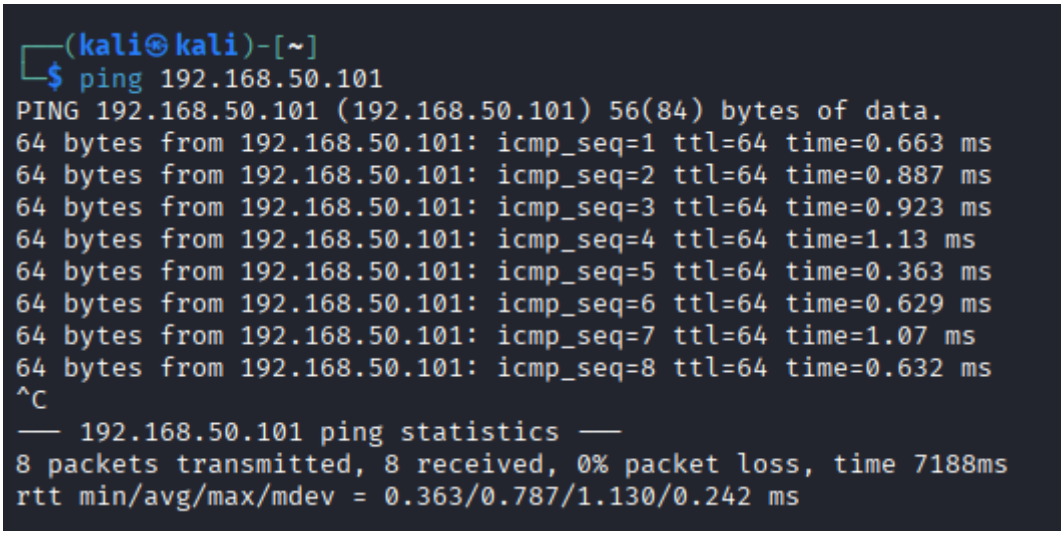
2.1 Configurazione del Laboratorio

Lavoreremo all'interno di un laboratorio virtuale composto da una macchina attaccante ed una macchina target.

- **Macchina Attaccante:** Kali Linux 2025.3 - IP: 192.168.50.100
- **Macchina Vittima:** Metasploitable 2 - IP: 192.168.50.101
- **Rete:** Rete Interna associata ad un interfaccia pfSense

Una volta configurato il nostro laboratorio virtuale eseguiamo il ping su entrambe le macchine per verificare che siano comunicanti tra loro.

Ping da Kali a Metasploitable 2:



Ping da Metasploitable 2 a Kali:

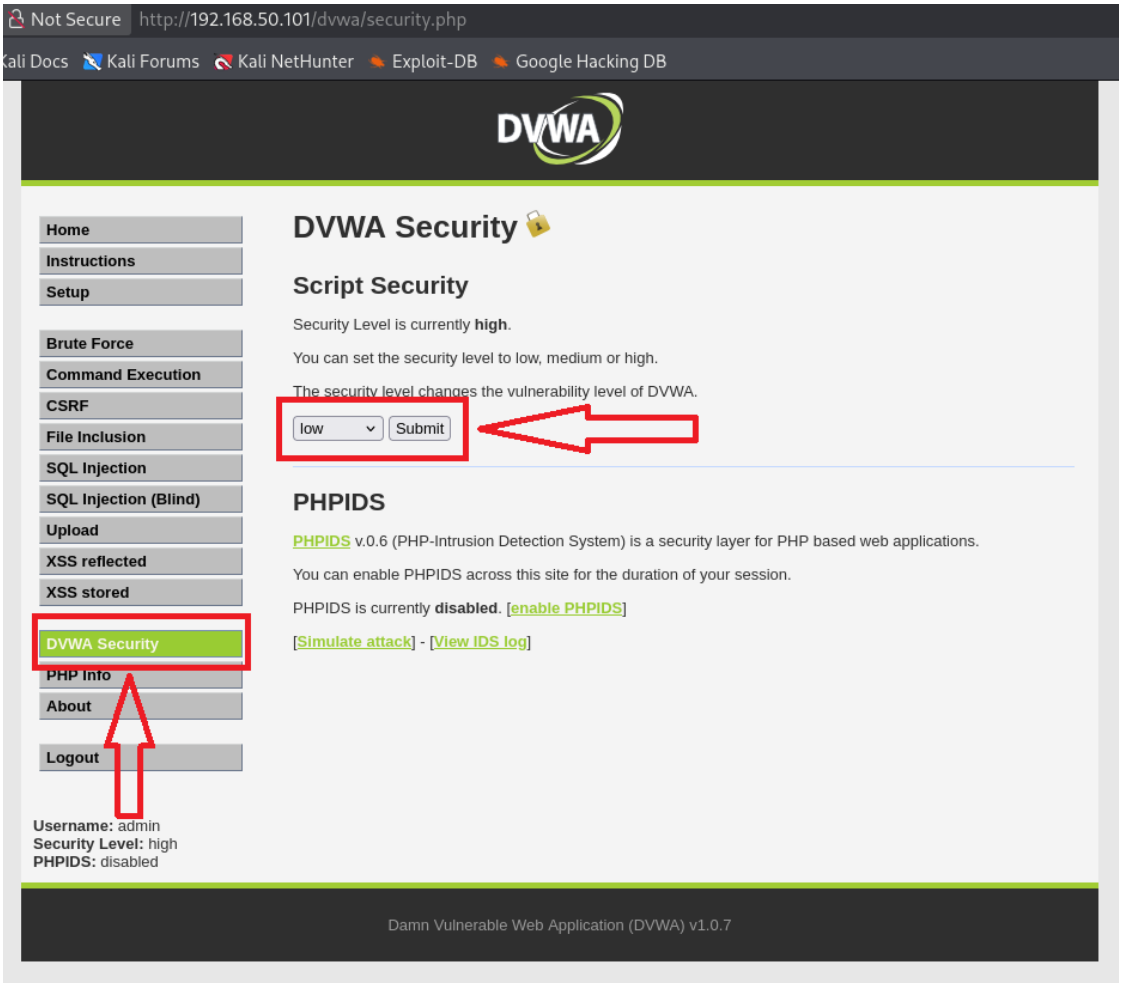
```
msfadmin@metasploitable:~$ ping 192.168.50.100
PING 192.168.50.100 (192.168.50.100) 56(84) bytes of data.
64 bytes from 192.168.50.100: icmp_seq=1 ttl=64 time=9.07 ms
64 bytes from 192.168.50.100: icmp_seq=2 ttl=64 time=0.985 ms
64 bytes from 192.168.50.100: icmp_seq=3 ttl=64 time=0.887 ms
64 bytes from 192.168.50.100: icmp_seq=4 ttl=64 time=0.658 ms
64 bytes from 192.168.50.100: icmp_seq=5 ttl=64 time=0.551 ms

--- 192.168.50.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 0.551/2.431/9.078/3.327 ms
```

Dopo esserci assicurati che le due macchine virtuali comunicano tra loro, accediamo alla Pagina DVWA dal browser della Kali inderendo il seguente URL:

http://192.168.50.101/dvwa

(Ho usato l'indirizzo IP della mia Metasploitable)



In questa fase andiamo ad impostare il livello di sicurezza a “Low” accedendo dal sotto menù “DVWA Security”.

Terminato questo passaggio abbiamo concluso con la fase di configurazione.

2.2 Strumenti Utilizzati

- **Browser**: Usato come interfaccia principale per navigare nell'applicazione web DVWA
- **Burpsuite**: Usato come proxy per intercettare e analizzare il traffico HTTP/HTTPS tra la macchina attaccante e il server vittima.
- **Shell PHP**: Usato come **payload** (codice malevolo). Si tratta dello script che, una volta caricato sul server, ci ha fornito una **backdoor** per inviare comandi di sistema

3. Attività Tecnica e Metodologia

3.1 Fase di *Creazione del payload*

*Il “payload” è il codice che andremo a caricare sul server vittima per fargli fare ciò che vogliamo, in sostanza è il nostro **Malware**.*

Creerò il mio file php direttamente da terminaletramite il seguente comando:

nano *MyMalware.php*

La precedente immagine mostra il file php che sono andato a creare.

La parte evidenziata di rosso è la parte più importante del codice:

- **system()**: Sarebbe una funzione nativa di PHP che consiste nel prendere una stringa di testo ed eseguirla come se fosse un comando scritto direttamente sul terminale del server (Metasploitable2)

- **$_REQUEST[‘cmd’]**: Sarebbe la variabile che ci permette di “parlare” con la funzione **system**. Quando noi scriveremo l’URL nel browser, tutto ciò che metteremo dopo **?cmd=** verrà catturato da questa variabile e passato direttamente al comando di sistema.

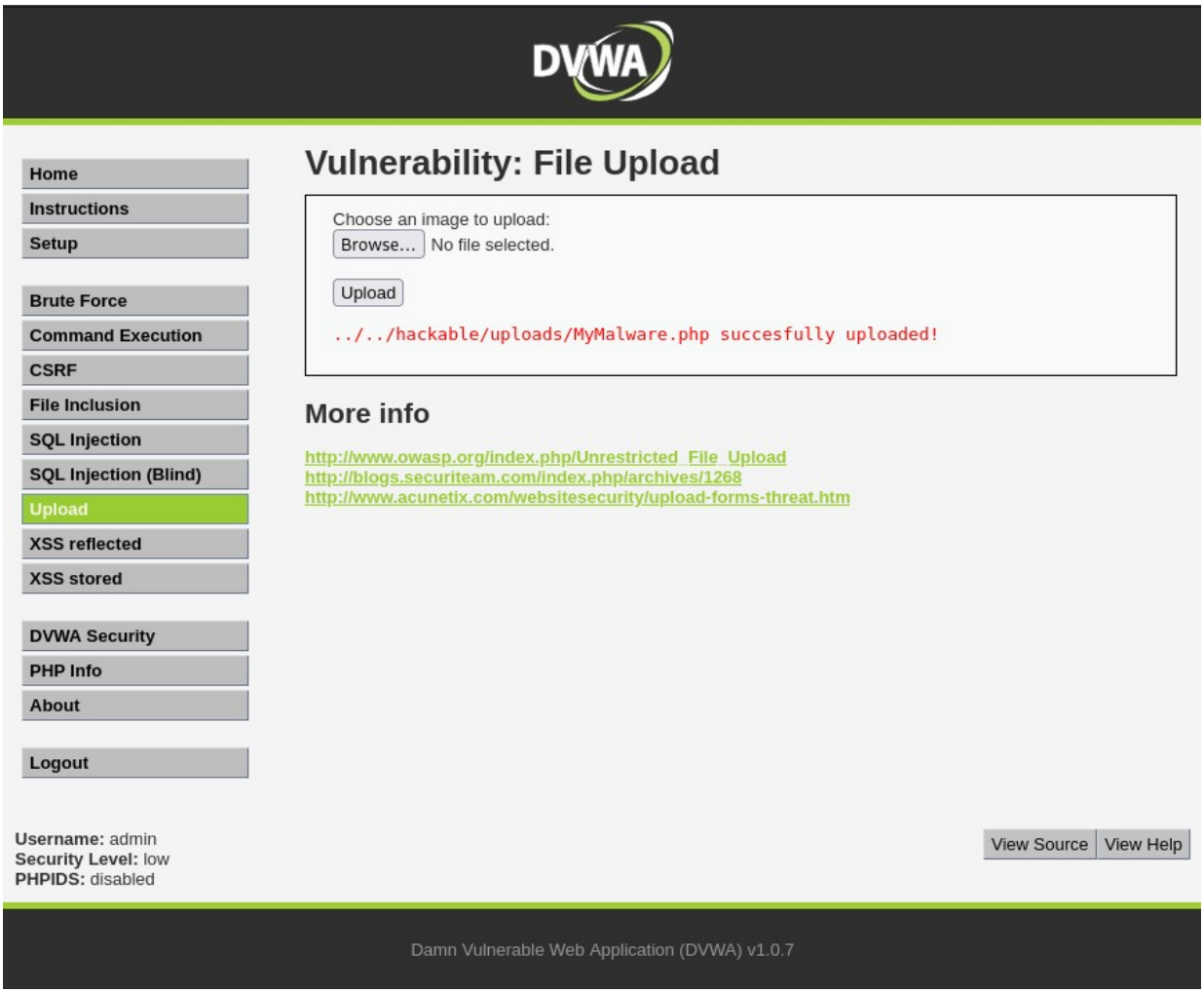
Scritto il file PHP procediamo a salvarlo con **CTRL-X** + **Y** e successivamente **INVIO**.

3.2 *Fase di Attacco*

In questa fase ciò che faremo è andare a spostare il file “Infetto” dalla Kali al server vulnerabile.

Per caricare il file seguiamo questi passaggi:

- **Tornare sul browser** ed assicuriamoci di essere sulla pagina *DVWA → Upload*
- Clicchiamo su **Browse** e selezioniamo il file php creato precedentemente



Come possiamo vedere nel messaggio (../../hackable/uploads/MyMalware.php succesfully uploaded!) il nostro file php è stato caricato con successo sul server.

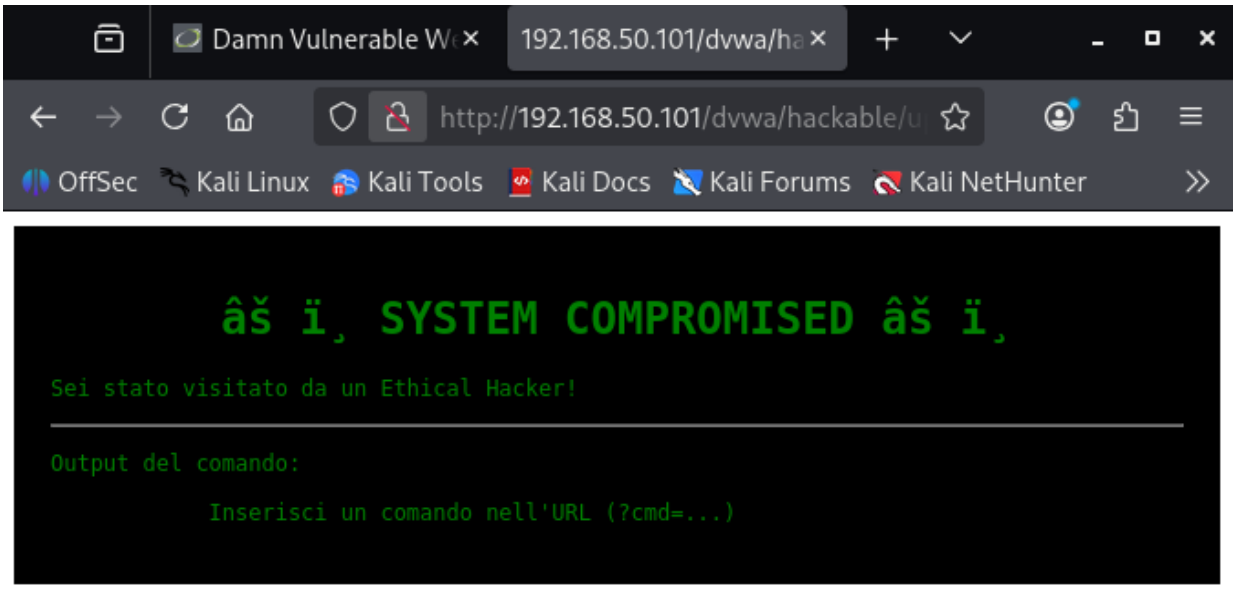
Inoltre ci dice in quale Path è stato inserito.

3.1 Fase di *Esecuzione*

Per eseguire il nostro file php basterà inserire sulla barra di ricerca l'URL in cui è avvenuto l'upload; nel nostro caso l'URL è il seguente:

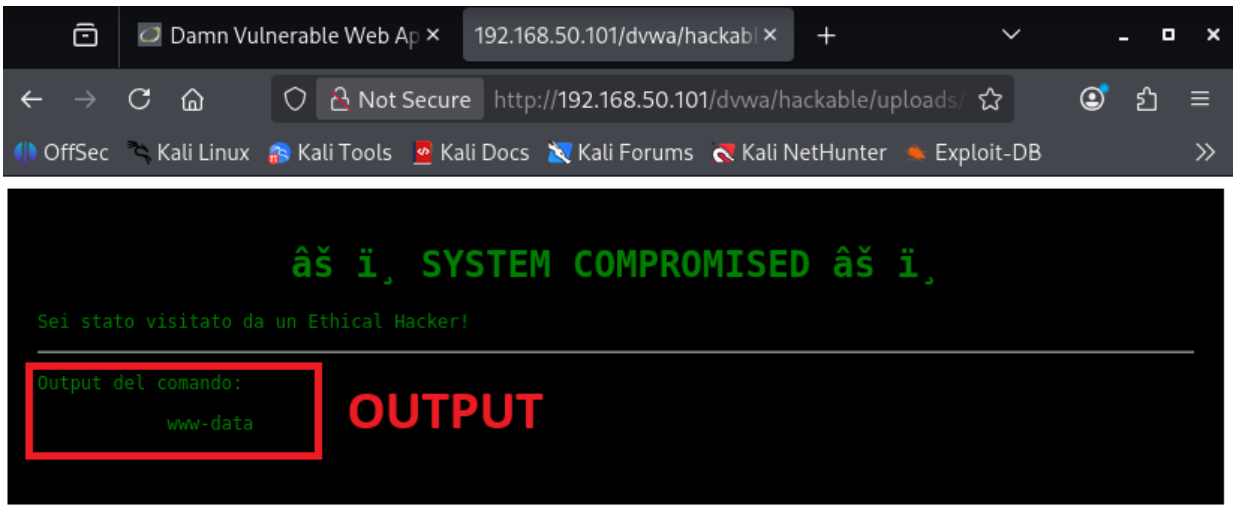
http://192.168.50.101/dvwa/hackable/uploads/MyMalware.php

Procediamo ad eseguirlo e vediamone i risultati.



Ciò che vediamo non è altro che il risultato "Grafico" del nostro file php; possiamo testare la parte "Funzionale" inserendo alla fine dell'URL la stringa **?cmd=whoami** al fine di ottenere come output il nome utente.

In altri termini è come se scrivessimo il comando whoami direttamente sul terminale di metasploitable.



Nell'immagine precedente la parte evidenziata in rosso mostra l'output del comando **whoami** che rivelerà il nome utente *(nel nostro caso "www.data").*

4. Analisi del Traffico (Burp Suite)

In questa sezione andremo ad intercettare ed analizzare ogni richiesta in modo da poter toccare con mano cosa succede "Dietro le quinte".

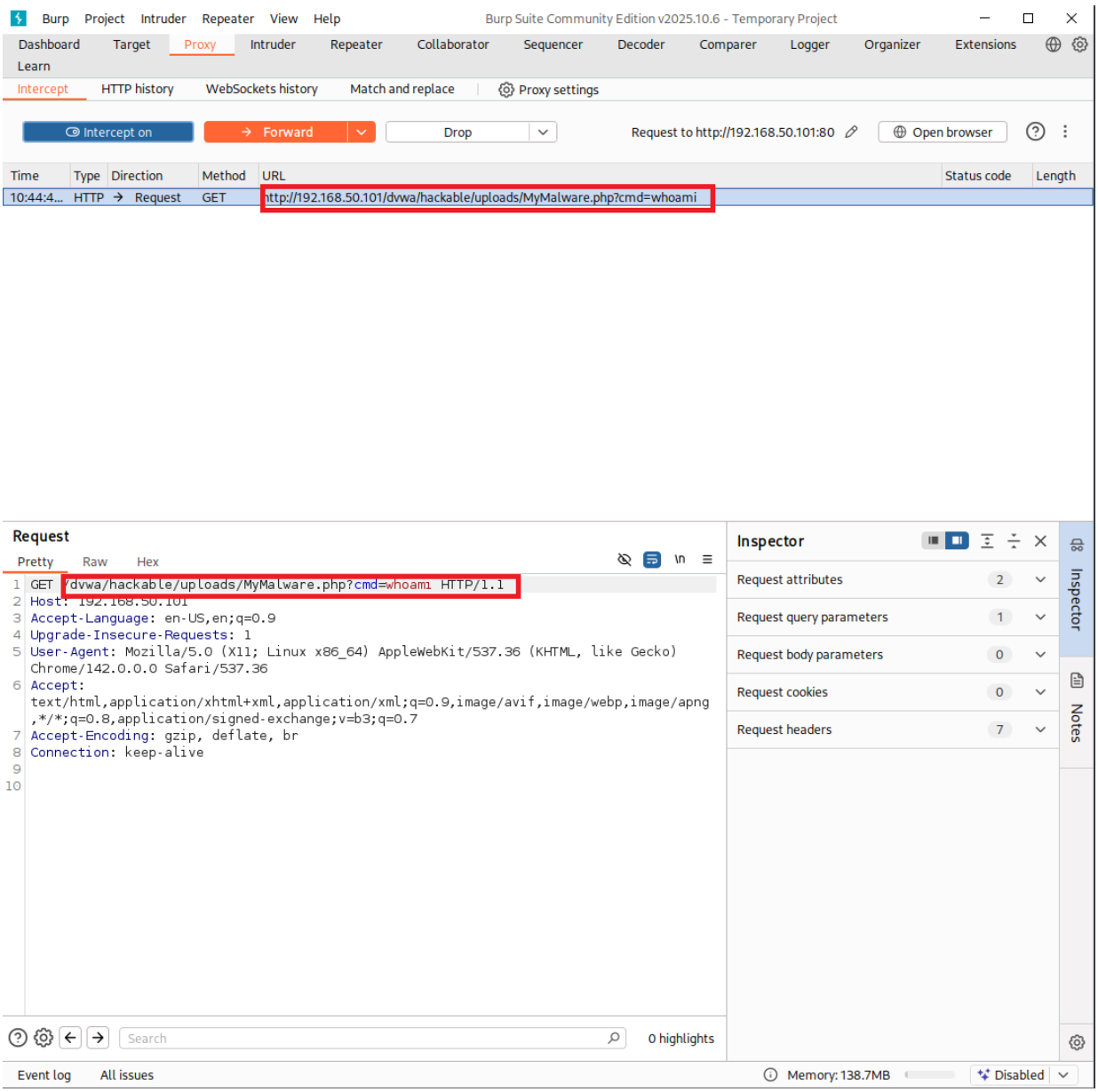
4.1 *Avvio intercettazione*

Per cominciare avviamo Burbsuite ed assicuriamoci di aver la "Modalita Intercettazione" seguendo questo percorso: **Proxy** -> **Intercept** -> **Intercept is on**

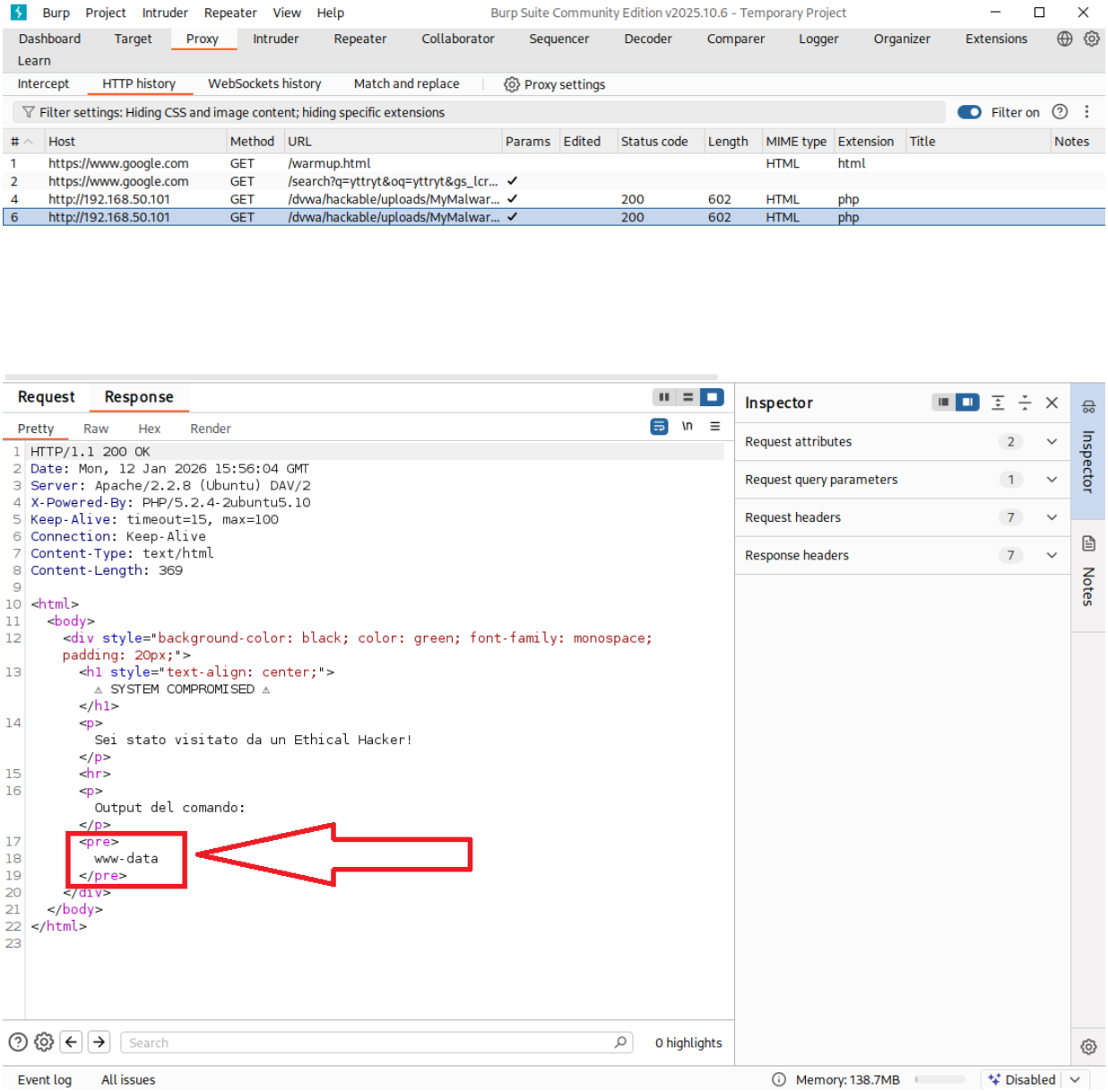
Fatto ciò possiamo avviare il browser ed andare nell'URL che abbiamo usato poco fa:

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

Burpsuite intercetta la prima richiesta di GET come possiamo vedere nella figura:



Procediamo cliccanto sul pulsante arancione Forward per proseguire con le chiamate HTTP e vedere cosa ci risponde il server alla nostra richieste.



Quello che vediamo evidenziato in rosso sarà il nome utente (il risultato del malware che abbiamo eseguito).

5. Esercizio Bonus: Bypass Security Level "Medium"

5.1 Analisi della Protezione

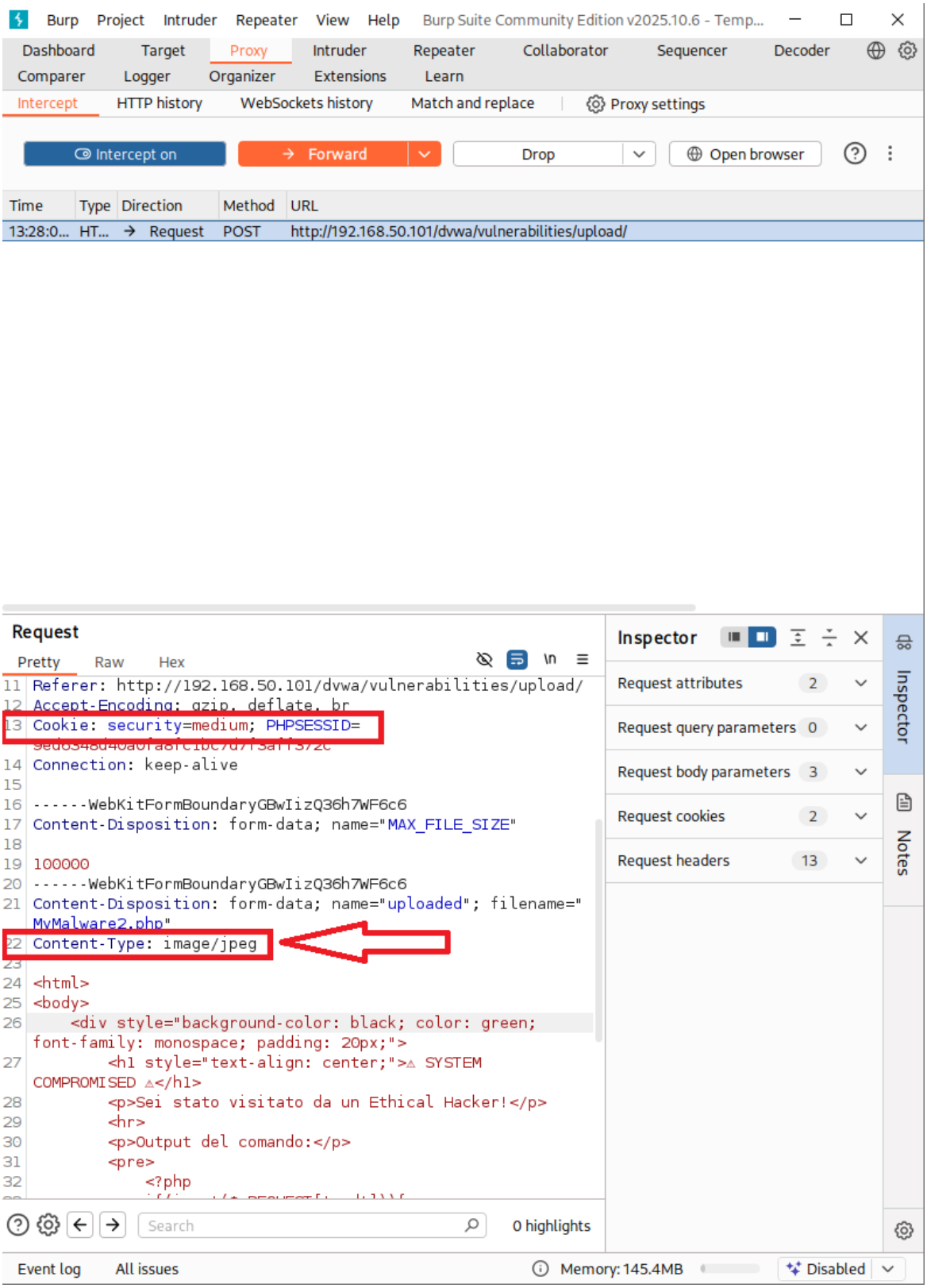
Dopo aver completato l'esercizio al livello "Low", abbiamo alzato l'asticella impostando il livello di sicurezza della DVWA su ***"Medium"****.*

In questa configurazione, il server introduce un meccanismo di difesa più sofisticato: non accetta più ciecamente qualsiasi file, ma verifica il ***MIME Type*** *(il tipo di contenuto dichiarato dal browser). Tentando di caricare il nostro file* `MyMalware.php` *normalmente, il server blocca l'operazione restituendo l'errore "Your image was not uploaded", poiché rileva che si tratta di uno script PHP (*`application/x-php`*) e non di un'immagine.*

5.2 Esecuzione dell'Exploit (MIME Type Bypass)

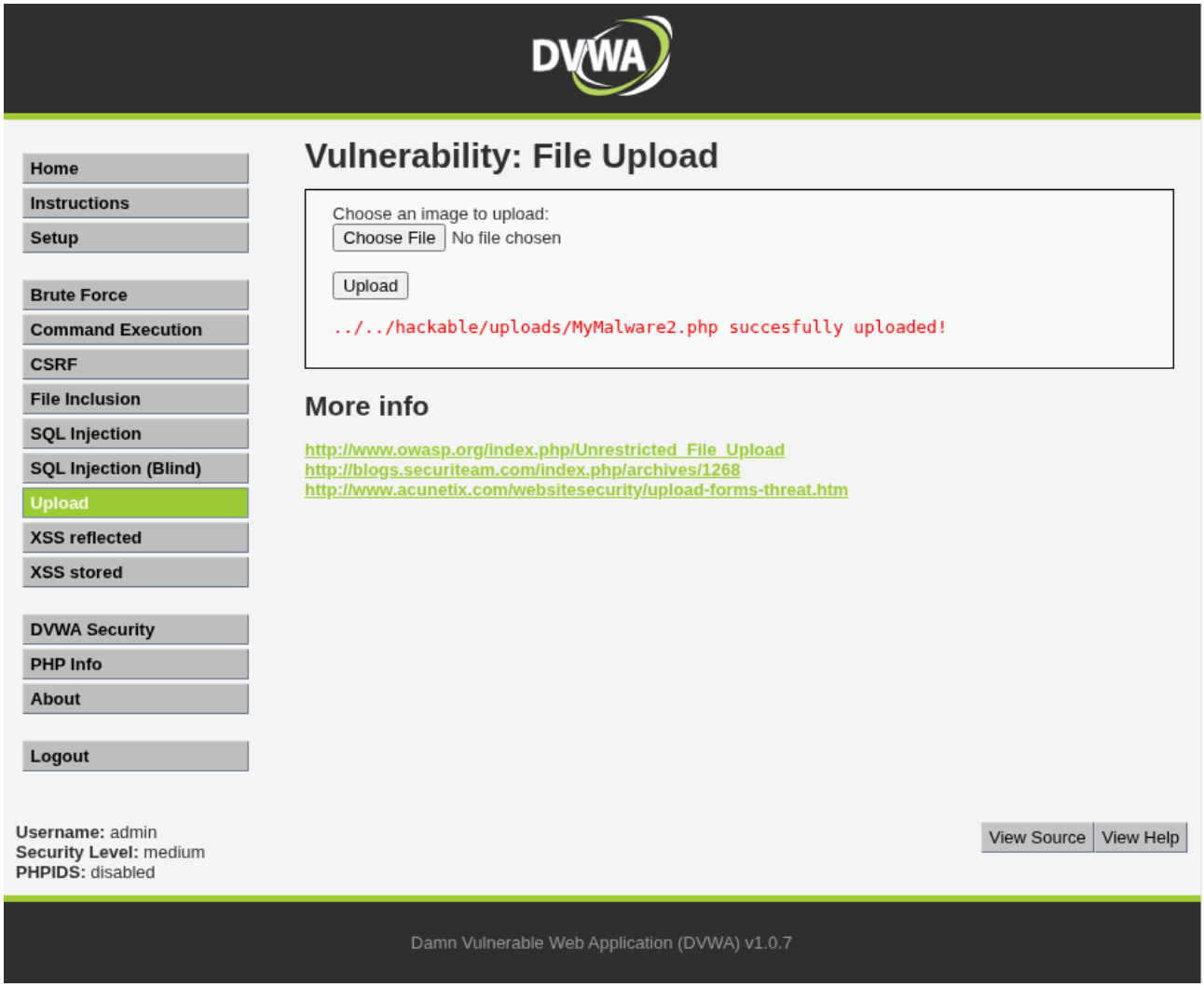
Per aggirare questa protezione, abbiamo utilizzato **Burp Suite** per manipolare la richiesta HTTP "al volo", ingannando il server. Ecco i passaggi eseguiti:

1. **Intercettazione:** Abbiamo avviato l'upload del file `MyMalware.php` con Burp in modalità Intercept On.
2. **Manipolazione:** Nella richiesta intercettata, abbiamo individuato l'header `Content-Type: application/x-php`. Lo abbiamo modificato manualmente in `Content-Type: image/jpeg`.



3. **Inoltro:** Abbiamo premuto Forward per inviare la richiesta modificata al server.

Il server, fidandosi del **Content-Type falsificato**, ha accettato il file credendolo un'immagine **JPEG**. Poiché l'estensione del file è rimasta `.php` (il filtro controllava solo il tipo, non l'estensione), il server web lo ha interpretato come codice eseguibile. Visitando l'URL del file caricato, abbiamo ottenuto nuovamente la nostra shell e l'esecuzione remota dei comandi (**RCE**), bypassando con successo il filtro di sicurezza.



Il file PHP è stato caricato all'interno del server eludendo i suoi controlli grazie alle modifiche che abbiamo apportato all'header **Content-Type**.

6. Conclusioni

6.1 Riepilogo

L'obiettivo dell'attività è stato pienamente raggiunto. Siamo riusciti a sfruttare con successo la vulnerabilità di ***File Upload*** *presente nella web application DVWA (configurata a livello "Low"). Attraverso il caricamento del payload* `MyMalware.php` *, abbiamo ottenuto l'****esecuzione di codice remoto (RCE)*** *sul server target Metasploitable2. La compromissione del sistema è stata confermata sia visivamente sia funzionalmente: l'intercettazione con Burp Suite e l'esecuzione del comando* `whoami` *hanno restituito l'utente* `www-data`*, dimostrando che abbiamo acquisito la capacità di eseguire comandi sul server ospite.*