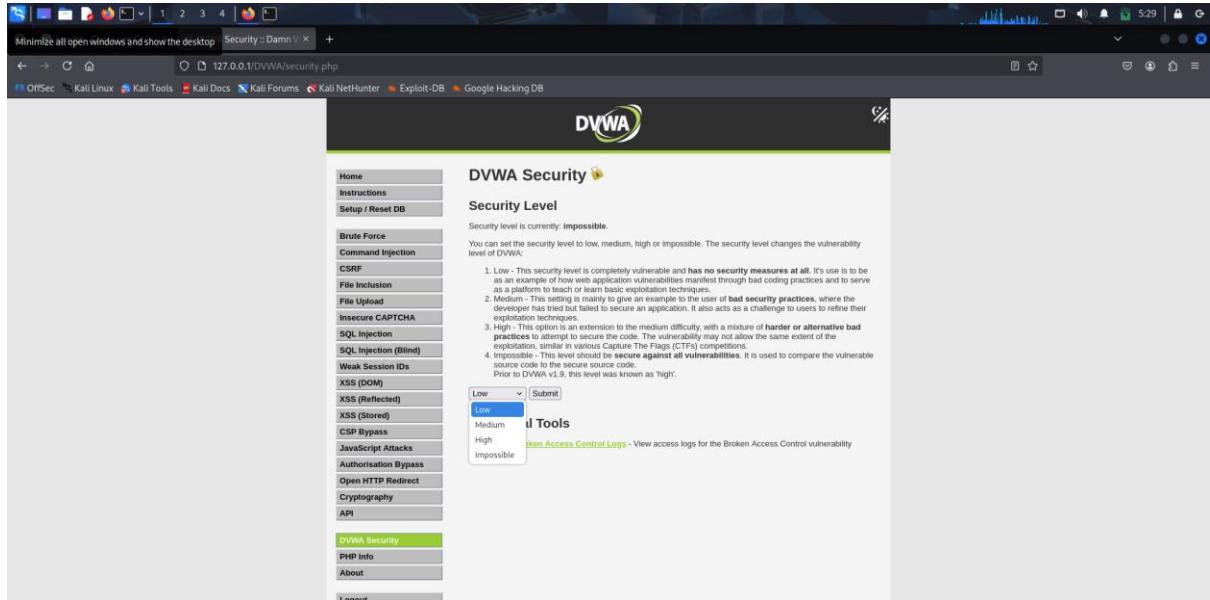
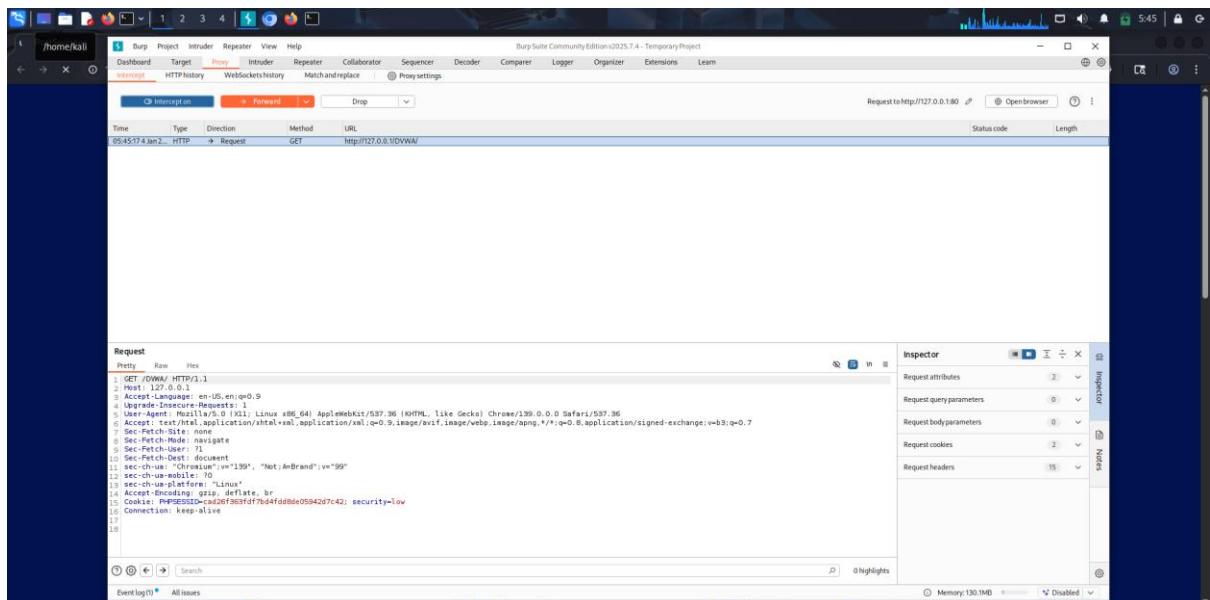


W13D1_ESERCIZIO DVWA- EXPLOIT FILE UPLOAD

Si utilizza la DVWA impostata su un livello di difficoltà basso (estremamente vulnerabile) per andare a vedere le tecniche che consentono di sfruttare le vulnerabilità, in questo caso si sfrutta la vulnerabilità UPLOAD.



In contemporanea, con l'avvio di Burpsuite, si intercettano le richieste che possono essere modificate grazie all'avvio di una pagina web ed inserendo l'indirizzo della DVWA (damn vulnerable web application).



Si iniziano a vedere le prime informazioni raccolte che sono:

-GET

-LIVELLO SICUREZZA

-SISTEMA OPERATIVO

-COOKIE

Site map filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host Method URL Params Status code Length Mimetype Title Notes Time requested

http://127.0.0.1 GET /DVWA/index.php 200 91 script

http://127.0.0.1 GET /DVWA/deeplinks/deeplink_event_listen... 200 1560 script

http://127.0.0.1 GET /DVWA/index.php 200 6857 HTML Welcome : Damn Vulnerable Web App...

http://127.0.0.1 GET /DVWA/login.php 200 1669 HTML Login: Damn Vulnerable Web App...

http://127.0.0.1 GET /DVWA/security.php 200 5589 HTML DVWA Security : Damn Vulnerable...

http://127.0.0.1 GET /DVWA/ 302 678

http://127.0.0.1 POST /DVWA/login.php 302 482

http://127.0.0.1 POST /DVWA/security.php 302 498

Request Response Inspector

Pretty Raw Hex Render Request attributes

Pretty Raw Hex Render Request headers

Pretty Raw Hex Render Response headers

Vulnerability: File Upload

Metasploitable2 - Linux

Warning: Never expose this VM to an untrusted network!

Contact: msfdev@metasploit.com

Login with msfadmin/msfadmin to get started

- TWiki
- phpMyAdmin
- Muttillidate
- DVWA
- WebDAV

Si esegue tramite burpsuite la connessione a metasploitable e si lascia attiva l'attività di cattura.

The screenshot shows the Burp Suite interface. In the top navigation bar, 'Applications' is selected. Below it, the 'Proxy' tab is active, showing a list of intercepting requests. A single request from 'http://192.168.50.106/dvwa/login.php' is selected. The 'Request' tab is open, displaying the raw HTTP request:

```

1 | GET /dvwa/login.php HTTP/1.1
2 | Host: 192.168.50.106
3 | Accept-Language: en-US,en;q=0.9
4 | Upgrade-Insecure-Requests: 1
5 | User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.5364.114 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
9 | 
10 |

```

The 'Inspector' panel on the right shows the request attributes, query parameters, body parameters, and cookies, all currently empty.

Si prosegue caricando sulla DVWA una shell.php andando a sfruttare la vulnerabilità upload come già detto.

The screenshot shows the DVWA 'File Upload' page. The URL is `127.0.0.1/DVWA/vulnerabilities/upload/`. The left sidebar menu is expanded, showing various attack types, with 'File Upload' currently selected. The main content area displays the following message:

Vulnerability: File Upload

The PHP module GD is not installed.

Choose an image to upload:

No file selected.

.../.../hackable/uploads/shell.php successfully uploaded!

More Information

- https://owasp.org/www-community/vulnerabilities/Insecure_File_Upload
- <https://www.acunetix.com/references/vulnerabilities/upload-forms-thru/>

The screenshot shows a Mozilla Firefox browser window with the title "DVWA Vulnerable Web App (DVWA) v1.0.7 - Source - Mozilla Firefox". The URL in the address bar is "192.168.50.106/dvwa/vulnerabilities/view_source.php?id=upload&security=low". The main content area displays the source code of a PHP script titled "File Upload Source". The code handles file uploads and prints a success message if the file is successfully moved to the target path.

```
<?php
if (isset($_POST['Upload'])) {
    $target_path = DVWA_WEB_PAGE_TO_ROOT . 'hackable/uploads/';
    $target_path = $target_path . basename($_FILES['uploaded']['name']);
    if(move_uploaded_file($_FILES['uploaded']['tmp_name'], $target_path)) {
        echo '<pre>';
        echo 'Your image was not uploaded.';
        echo '</pre>';
    } else {
        echo '<pre>';
        echo $target_path . ' successfully uploaded!';
        echo '</pre>';
    }
}
?>
```

La shell.php è stata correttamente caricata come restituisce anche il risultato del caricamento.