SQL Injection
Cross-site Scripting

NISRA

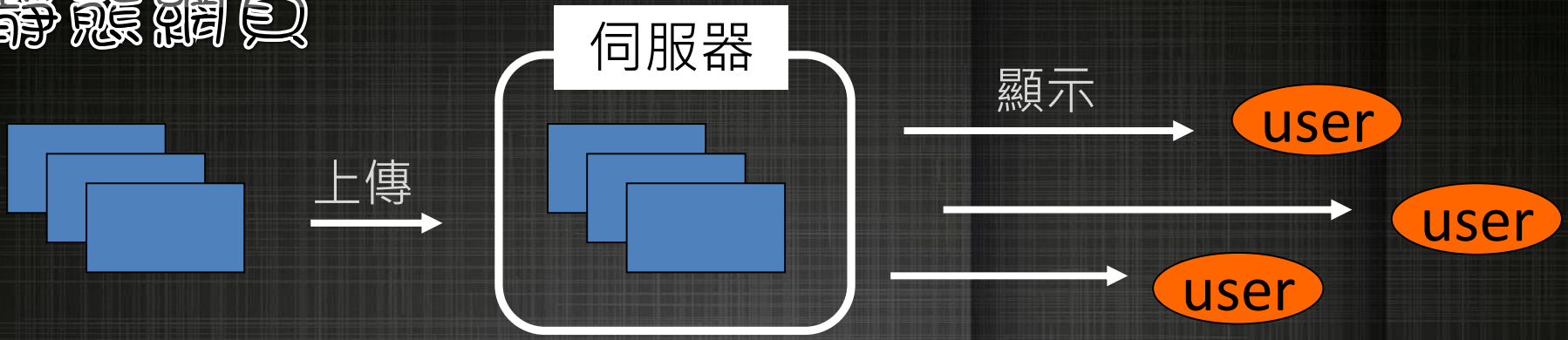
講者介紹

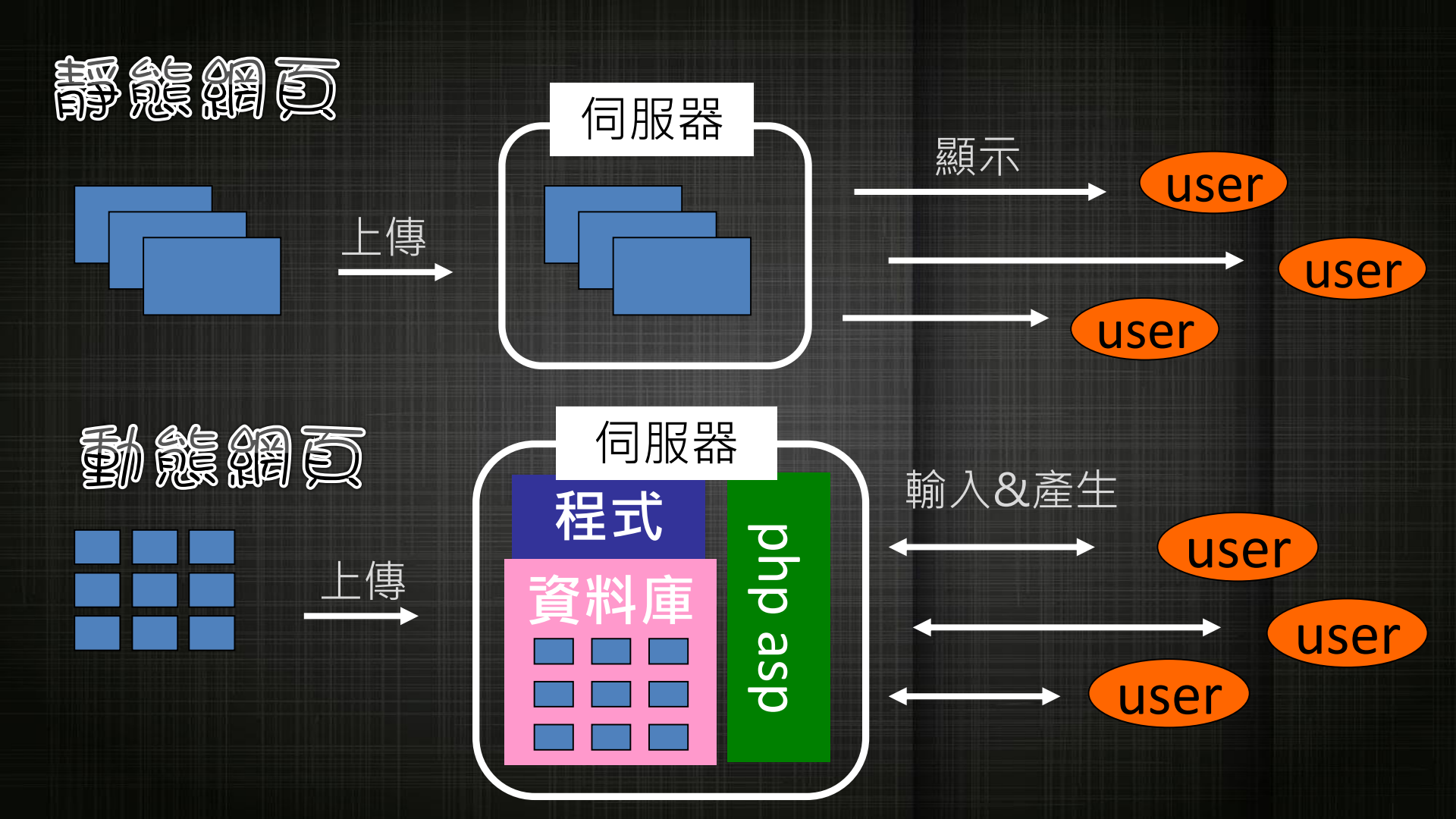
- 王薪嘉
- 現任 NISRA 會長

關於網頁

- 靜態
  - 只能瀏覽資料 (純粹圖文)
  - Html + CSS
- 動態
  - 用戶端與伺服器可以互動
  - Ex: 投票系統、檔案上傳、購物網站, etc
  - PHP, JSP, ASP, etc

靜態網頁




靜態網頁
伺服器
顯示
上傳
user
user
user
動態網頁
伺服器
程式
資料庫
dsa dup
輸入&產生
上傳
user
user
user

資料隱碼攻擊

SQL Injection

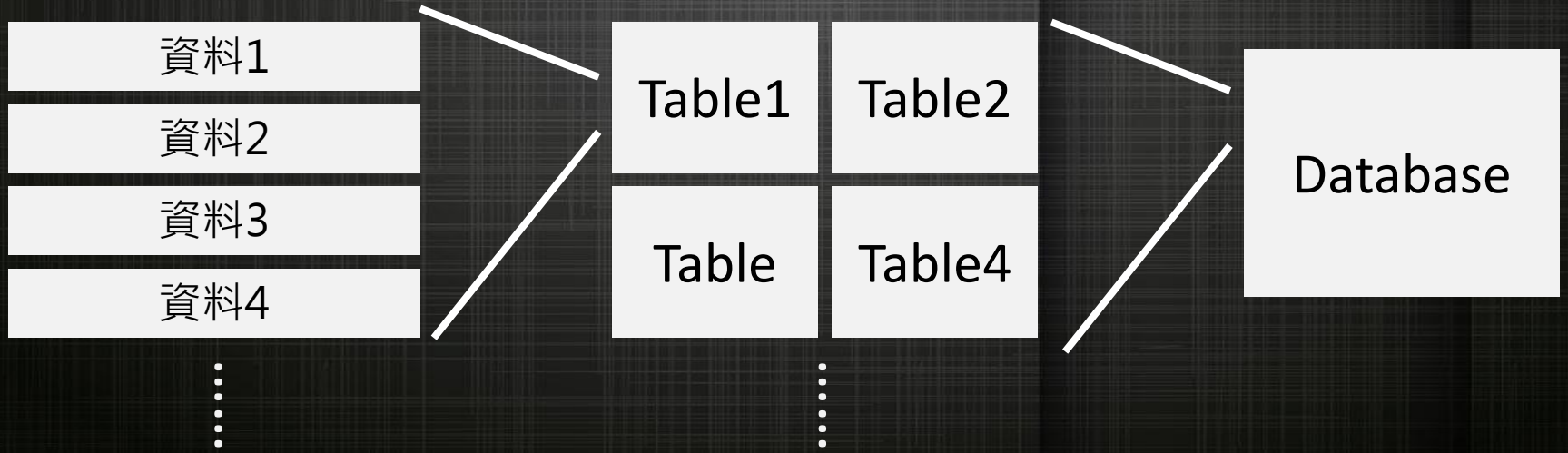
SQL - 結構化查詢語言

- Structured Query Language，一般習慣唸成 "sequel"，不過正確的唸法應該是 "S-Q-L"
- 是一種常見於資料庫的語言，用於資料存取、查詢、更新和管理關聯式資料庫系統
- 同時也是資料庫指令檔的副檔名 (.sql)。

- SQL 的語法是由一些簡單的句子構成，簡單易學

資料庫

- Database：按照資料結構來組織、存儲和管理資料的倉庫。使用者可以對檔案中的資料執行新增、更新、刪除、搜尋等操作。

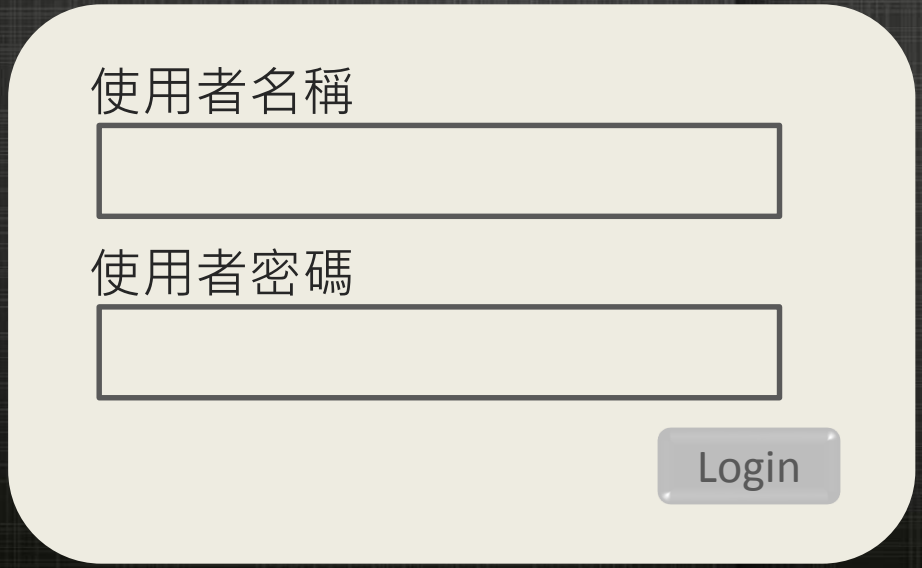


SQL Injection

- 發生於應用程式之資料庫層的安全漏洞。
- 在輸入的字串中夾帶 SQL指令，若程式沒有進行檢查而使這些字串被誤認為是合法的 SQL指令並執行，將會造成：
  - 資料竊取
  - 資料刪除
  - etc

舉例來說 - 無帳密登入

- 登入檢查：判定 User 輸入的帳號、密碼是否正確，來確定登入是否成功。



舉例來說 - 無帳密登入

- 登入檢查：判定 User 輸入的帳號、密碼是否正確，來確定登入是否成功。
- select * from members where account='$name' and password='$password'

舉例來說 - 無帳密登入

- 登入檢查：判定 User 輸入的帳號、密碼是否正確，來確定登入是否成功。
- select * from members where account='$name' and password='$password'
- 此時帳號輸入' or 1=1 /* ，密碼任意輸入

舉例來說 - 無帳密登入

- select * from members
  where account ='' or 1=1 /*' and password=''

舉例來說 – 無帳密登入

- select * from members
  where account ='' or 1=1 /*' and password=''

- /* 在 MySQL 語法中代表註解的意思。
  所以「/*」後面的字串通通沒有執行，而這句判斷式「1=1」永遠成立，就能藉此登入網站成功。

舉例來說 - 無帳密登入

- select * from members
  where account ='' or 1=1 /*' and password=''

- /* 在 MySQL 語法中代表註解的意思。
  所以「/*」後面的字串通通沒有執行，而這句判斷式「1=1」永遠成立，就能藉此登入網站成功。
- MySQL 的註解有三種：「/*」「--」「#」

舉例來說 - 刪除資料

- DROP：SQL 語法中關於刪除的指令
- 假如使用者輸入的地方可以執行 DROP 指令，那也許將會刪除：
  表格 ->　DROP TABLE "表格名稱"
  資料庫 ->　DROP DATABASE "資料庫名稱"

SQL Injection - 預防

- 過濾使用者的輸入：' " /
  ' or 1=1 /* → or 1 = 1

SQL Injection - 預防

- 過濾使用者的輸入：' " /
  ' or 1=1 /* → or 1 = 1
- 加工使用者輸入的字串：把字串中的特殊字元前加上 \ 再回傳

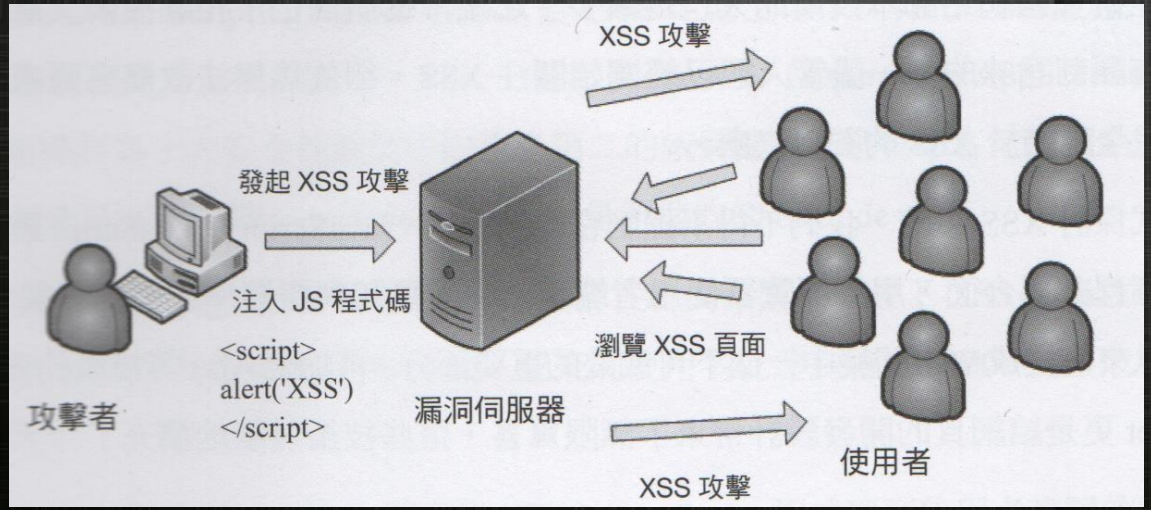
原始字串	過濾後的字串
It's cool	It\'s cool
\n	\\n
"WOW"	\"WOW\"

XSS - 跨網站指令碼

Cross-site Scripting

XSS ?

- 一種常見於 Web 應用程式中的電腦安全性漏洞。
- 在網頁中注入惡意程式，透過使用者在網路上擴散。



普遍的原因

- Web 瀏覽器本身的設計不安全。
- XSS 觸發的門檻低且不受重視。
- Web 2.0 後網站上的交互功能日漸強大，我們將有越來越多的機會可以查看、修改他人的資訊。

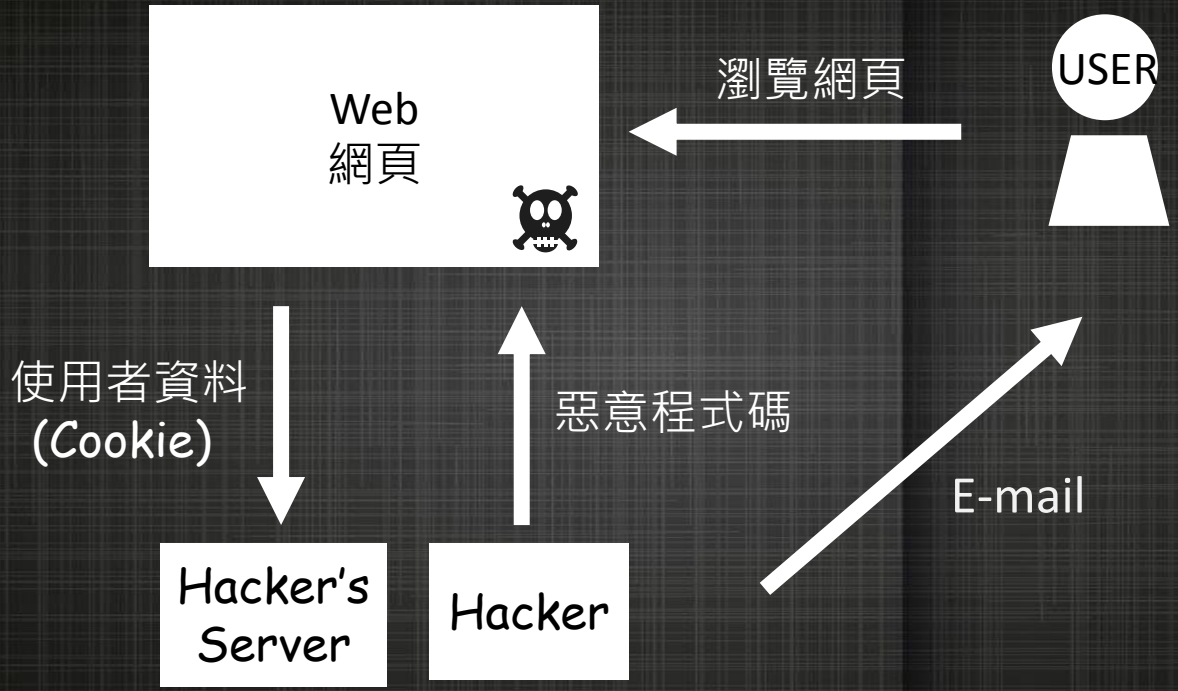
反射型 & 持久型

XSS 分類

反射型 XSS

- 又稱 非持久型、參數型 XSS。
- 在使用者按下時觸發。
- 一般是透過特定手法(如：E-mail)，誘使 User 連結包含惡意程式的 URL。

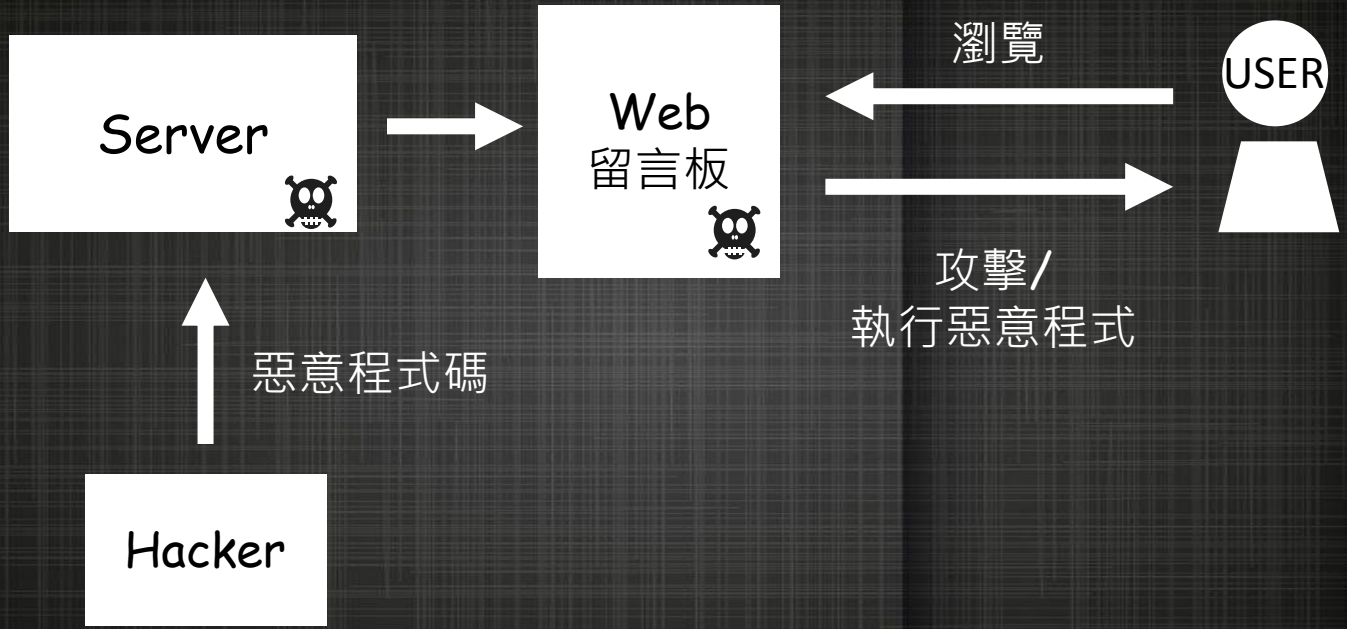
- 常出現於網站的搜尋欄、使用登入介面用來竊取 Cookie 或是釣魚欺騙。


Web
網頁
USER
瀏覽網頁
使用者資料
(Cookie)
惡意程式碼
E-mail
Hacker's
Server
Hacker

持久型 XSS

- 又稱儲存型，可能會影響 Web 伺服器。
- 先將惡意腳本上傳或儲存在有漏洞的伺服器上，只要受害者瀏覽到相關頁面就會執行惡意程式。

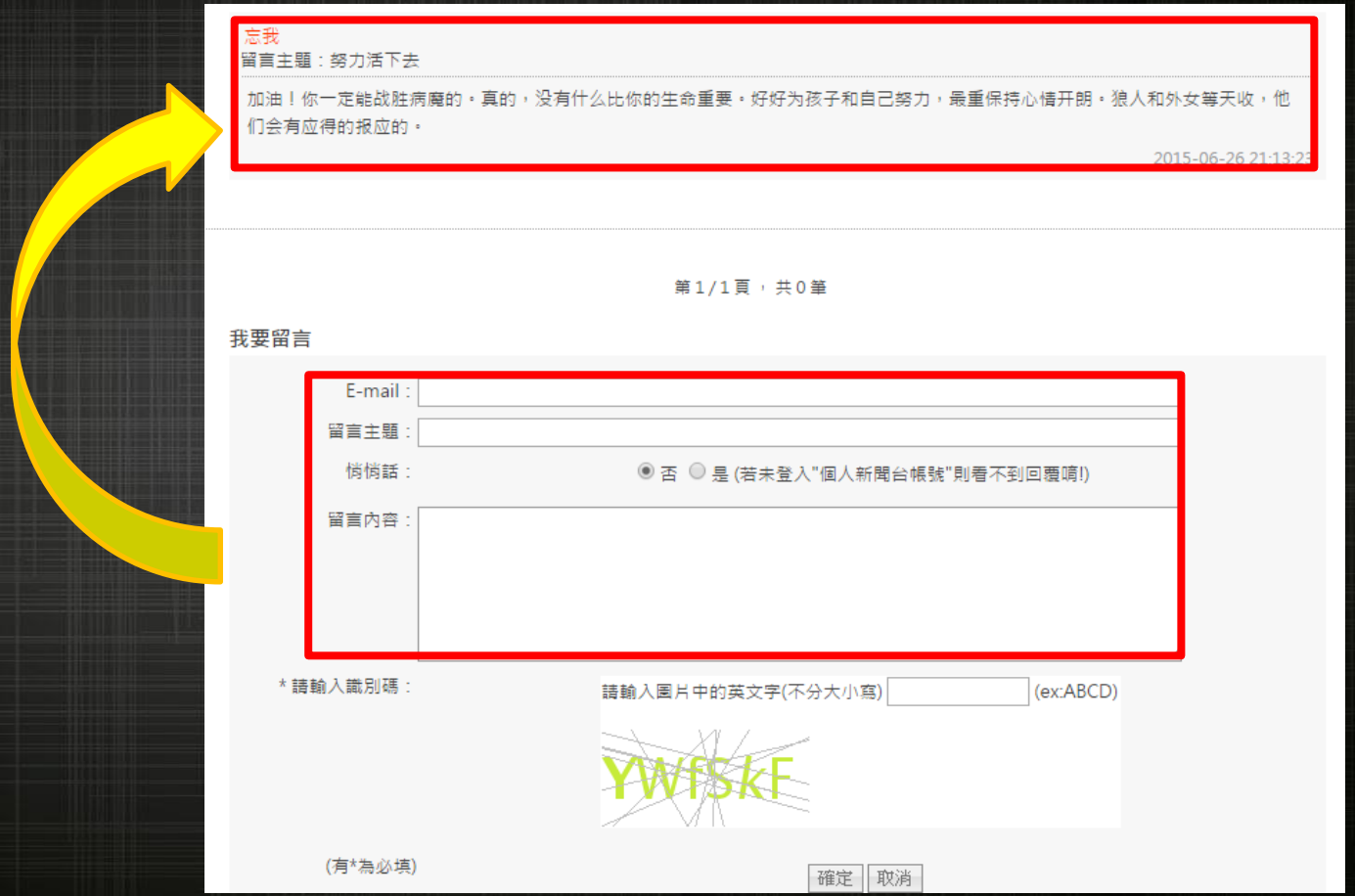
- 一般出現在網站的留言板、評論、部落格日記等。可以用來滲透網站、木馬、釣魚、編寫 XSS 蠕蟲。


Server
Web
留言板
瀏覽
USER
攻擊/
執行惡意程式
惡意程式碼
Hacker

挖掘 XSS 漏洞

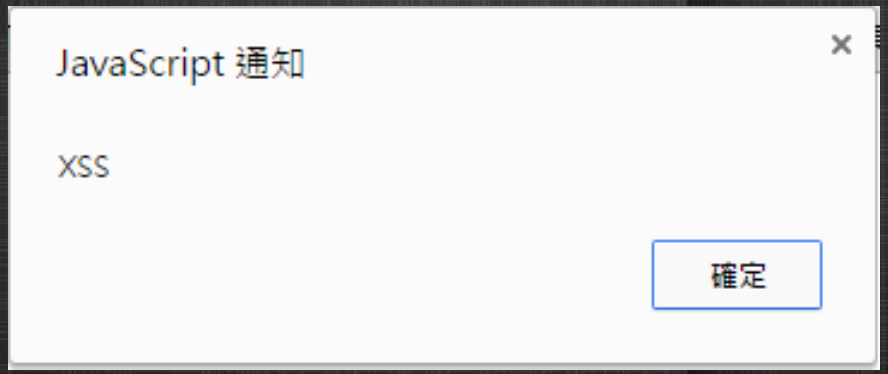
- 尋找 可以顯示使用者輸入文字的地方
- 測試 是否可以執行 腳本語言 (Ex: Javascript)
- 植入 惡意程式

尋找：


忘我
留言主題：努力活下去
加油！你一定能战胜病魔的。真的，没有什么比你的生命重要。好好为孩子和自己努力，最重保持心情开朗。狼人和外女等天收，他们会有应得的报应的。
2015-06-26 21:13:23
第 1 / 1 頁 ， 共 0 筆
我要留言
E-mail :
留言主題 :
悄悄話 :
否
是 (若未登入"個人新聞台帳號"則看不到回覆唷!)
留言內容 :
* 請輸入識別碼 :
請輸入圖片中的英文字(不分大小寫)
(ex:ABCD)
YWfSkF
(有*為必填)
確定
取消

測試：alert(1)

- JavaScript 讓瀏覽器彈出訊息小框框的內置函數。



測試

- 不一定要放 alert(1)
  - prompt(1)
  - <meta http-equiv="refresh" content="0;">
  - <iframe src=http://www.text.com width=0 height=0></iframe>

XSS Cheat Sheet

- 國外著名安全工程師 Rsnake 研究 XSS 的心得。
- 常見的 XSS 攻擊腳本列表，用來檢測 Web 是否存在 XSS 漏洞。
- ha.ckers.org/xss.html
  現今最完整的 XSS 測試用範例。

- http://www.xenuser.org/xss-cheat-sheet/ 簡易版

原因

- 將腳本語言加到 Web 頁面的過程非常簡單：
  只要加入<script></script> 標籤即可。

- 瀏覽器只負責解釋和執行腳本語言，
  不會判斷程式碼惡意與否。

看起來很雞肋

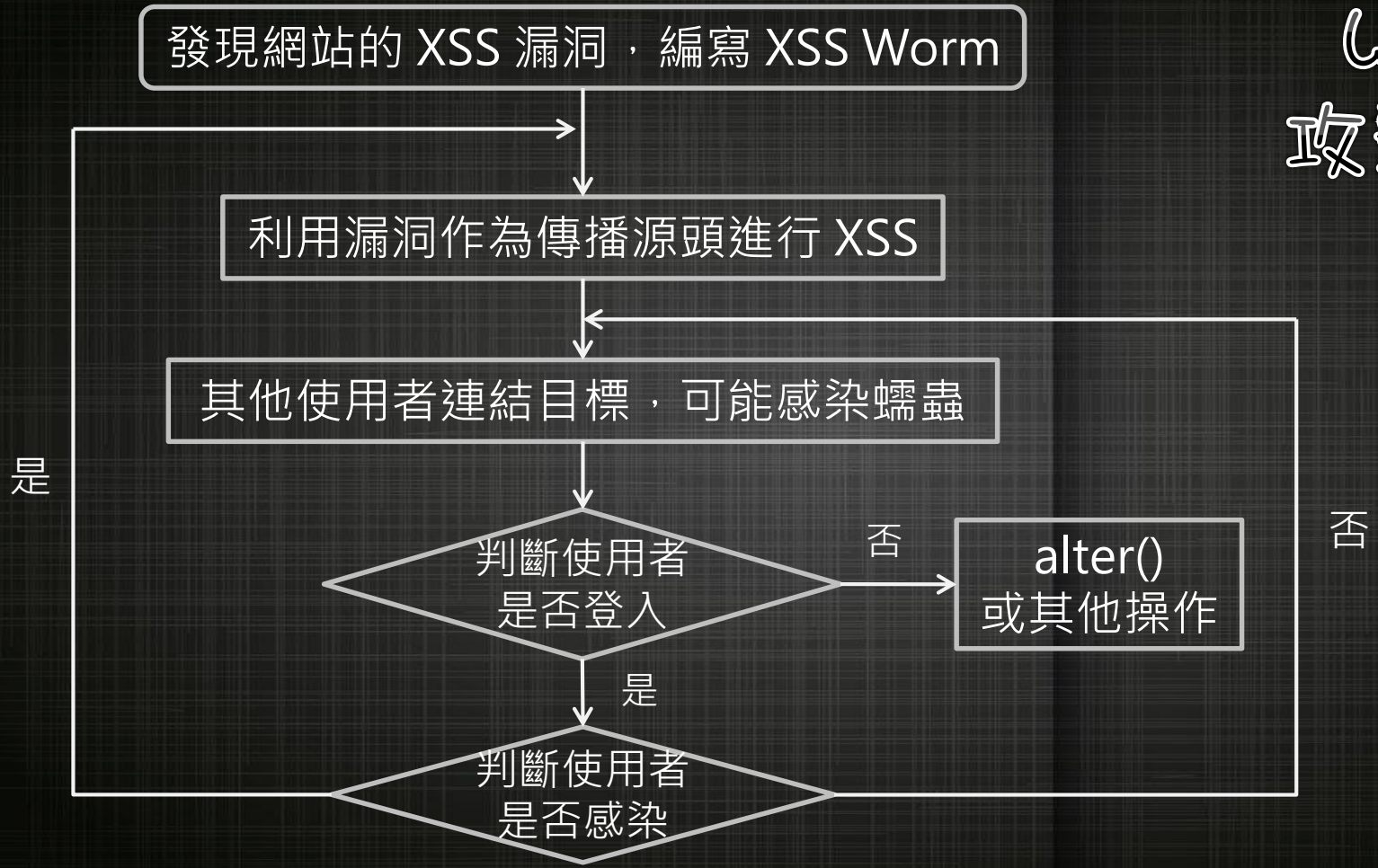
- XSS 不如 SQL Injection、檔案上傳等能夠直接得到較高權限的操作。
- 但是它的運用十分靈活。
- 例如：
  - 2005/10 Myspace跨腳本網站蠕蟲

2005/10 Samy

- 世界上第一隻網路蠕蟲
- 網路社群 MySpace
- 20 小時內感染 “一百多萬個” 使用者，最後 MySpace 伺服器崩潰。

- 19歲的 Samy 和女友打賭他可以在 Myspace 上擁有眾多粉絲。
- 當然……辦不到！
- 研究 Myspace → 發現 個人簡介 處存在 XSS 漏洞。
- 注入一段 JS 蠕蟲，每個查看他簡介的人在不知不覺中自動執行這段程式碼。
- 蠕蟲打開受害者的個人簡介，自我複製在受害者的個人簡介。
- 瘋狂散播直到伺服器崩潰。

Worm 攻擊原理

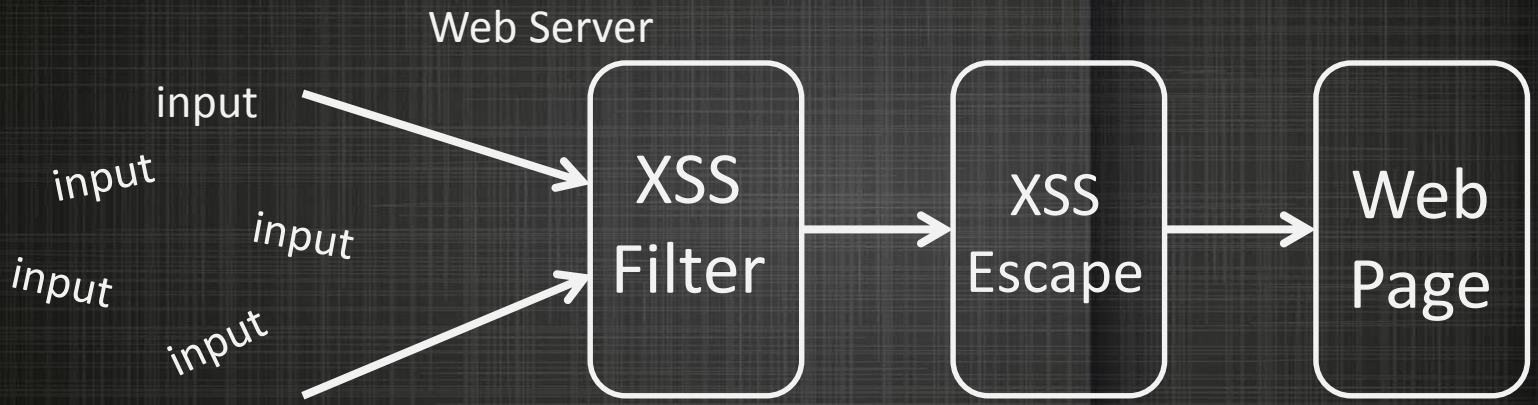

發現網站的 XSS 漏洞，編寫 XSS Worm
利用漏洞作為傳播源頭進行 XSS
其他使用者連結目標，可能感染蠕蟲
判斷使用者
是否登入
否
alter()
或其他操作
是
判斷使用者
是否感染
是
否

XSS 防禦

- XSS 和 SQL Injection 一樣，都是利用 Web 編寫不完善來攻擊。
- 因此每一個漏洞該利用和針對的弱點都不盡相同。

- 這給 XSS 防禦帶來了困難：
  不可能以單一特徵來概括所有 XSS 攻擊。

一切輸入都是有害的


Web Server
input
input
input
input
input
XSS Filter
XSS Escape
Web Page

XSS Filter

- 把要處理的資料分做黑、白名單兩大列表：
  白名單存放可信賴的、無威脅的資料；
  黑名單則相反。
- 其實就是一段精心編寫的過濾函式。

- 還是很容易被繞過 OTZ

XSS Escape

- 驗證：設定格式、數字範圍、字數限制等
- 數據消毒：過濾一些敏感字元：< > ' " & #
- Javascript Expression：

顯示	實體名字	實體編號
<	&lt;	&#60:
>	&gt;	&#62:
&	&amp;	&#38;
"	&quot;	&#34;

XSS 防禦 - 使用者

- 別亂點陌生人給的連結
- 禁用 JavaScript
  - 網頁會變得難用

THE END