軟體系統安全管理

大綱

1 資安的概念與範圍

2 軟體系統安全設計原則

3 軟體系統安全測試方法

4 資訊系統委外開發RFP資安需求
政府組態基準GCB

1. 資安的概念與範圍

駭客攻擊思維探討、企業常見資安缺口

資安的概念與範圍

■ 近期資安事件-你想知道什麼!

滲透測試

	Top Vulnerabilities
01	Broken Access Control
02	Broken Authentication
03	Cross Site Scripting
04	Security Misconfiguration
05	Sensitive Data Exposure

資料來源：行政院國家資通安全會報技術服務中心

網頁弱點檢測

	Top Vulnerabilities
01	Application error message
02	Directory listing
03	User credentials are sent in clear text
04	Error Message on page
05	SQL Injection

資料來源：2020測試中心

你發現什麼議題???

資安的概念與範圍

出處：IThome

年度網路攻防演練發現重要事項

1. 集中使用相同套件或委外廠商
2. 未即時更新使用之軟體套件
3. 不正確的安全觀念
4. 未落實通行碼強度檢查的機制
5. 不當的轉導設計

OR

■ 防範建議

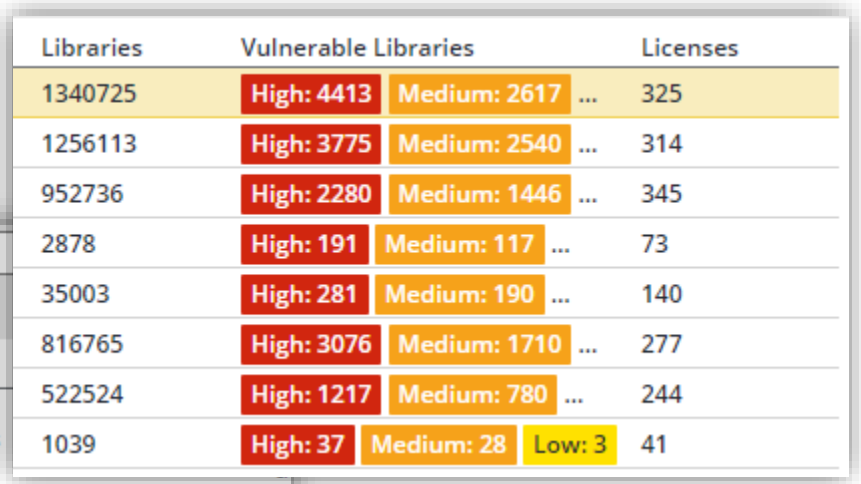
- 必須要建立機制以掌握關連性，共通系統安全維護。最簡單的原則，就是應瞭解內部使用相同套件的系統，納入定期追蹤的範圍，一旦發現該套件弱點，也要建立情資分享管道。
- 權責分離!!!!!!!!!

資安的概念與範圍

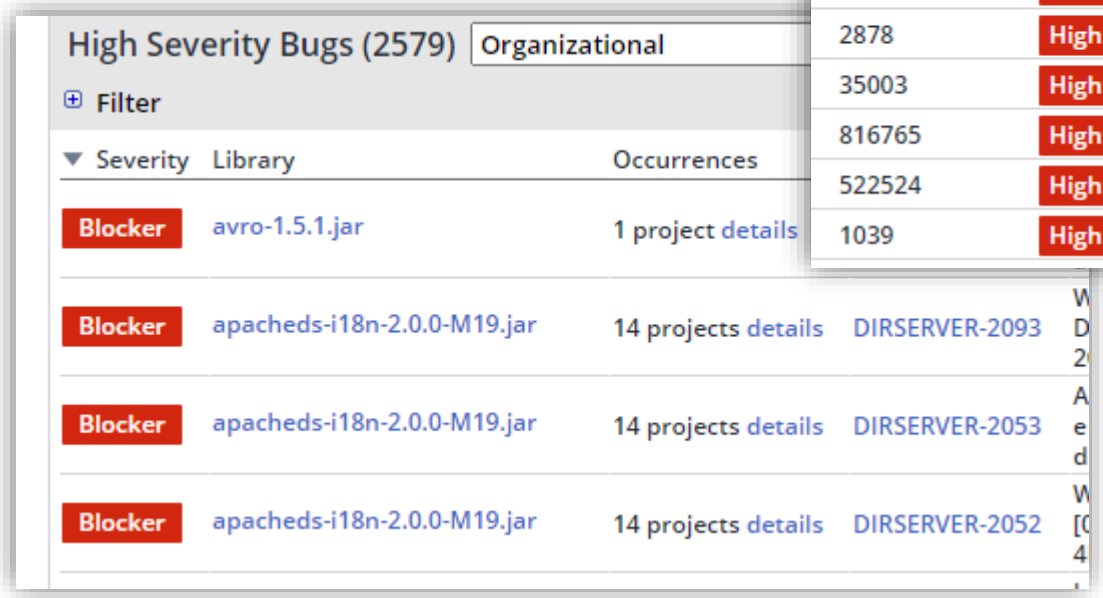
出處：IThome

年度網路攻防演練發現重要事項

1. 集中使用相同套件或委外廠商
2. 未即時更新使用之軟體套件
3. 不正確的安全觀念
4. 未落實通行碼強度檢查的機制
5. 不當的轉導設計



Libraries	Vulnerable Libraries			Licenses
1340725	High: 4413	Medium: 2617	...	325
1256113	High: 3775	Medium: 2540	...	314
952736	High: 2280	Medium: 1446	...	345
2878	High: 191	Medium: 117	...	73
35003	High: 281	Medium: 190	...	140
816765	High: 3076	Medium: 1710	...	277
522524	High: 1217	Medium: 780	...	244
1039	High: 37	Medium: 28	Low: 3	41



High Severity Bugs (2579) Organizational

Filter

Severity	Library	Occurrences	
Blocker	avro-1.5.1.jar	1 project details	
Blocker	apacheds-i18n-2.0.0-M19.jar	14 projects details	DIRSERVER-2093
Blocker	apacheds-i18n-2.0.0-M19.jar	14 projects details	DIRSERVER-2053
Blocker	apacheds-i18n-2.0.0-M19.jar	14 projects details	DIRSERVER-2052

■ 防範建議

- 若開發期間已知使用第三方套件，應將套件名稱與版本，加到該系統的備註資訊，或建立大型第三方套件清單，並做到盤點清冊的關聯，讓弱點揭露時，可以更主動、及時的處理。
- CVE Details

資安的概念與範圍

出處：IThome

年度網路攻防演練發現重要事項

1. 集中使用相同套件或委外廠商
2. 未即時更新使用之軟體套件
3. 不正確的安全觀念
4. 未落實通行碼強度檢查的機制
5. 不當的轉導設計

電腦修補遲未更新(**已知漏洞風險**)

(**線上流通檔案資料**)用 LINE交辦公事

密碼寫在便條紙上(**帳號密碼洩漏**)

(**物理性流通檔案**)買隨身碟備份

申請 Google Drive 來放檔案(**資料脫離公司規範**)

程式弱點未修補(**系統上線風險**)

電腦用自己的 4G 吃到飽上網(**脫離企業網路**)

(**未受公司監測控管**)帶自己家的電腦

裝盜版軟體(**惡意軟體風險**)

■ 防範建議

- ➢ 公司資訊安全規範
- ➢ 系統開發安全程式碼撰寫指引

Daryl Cheung

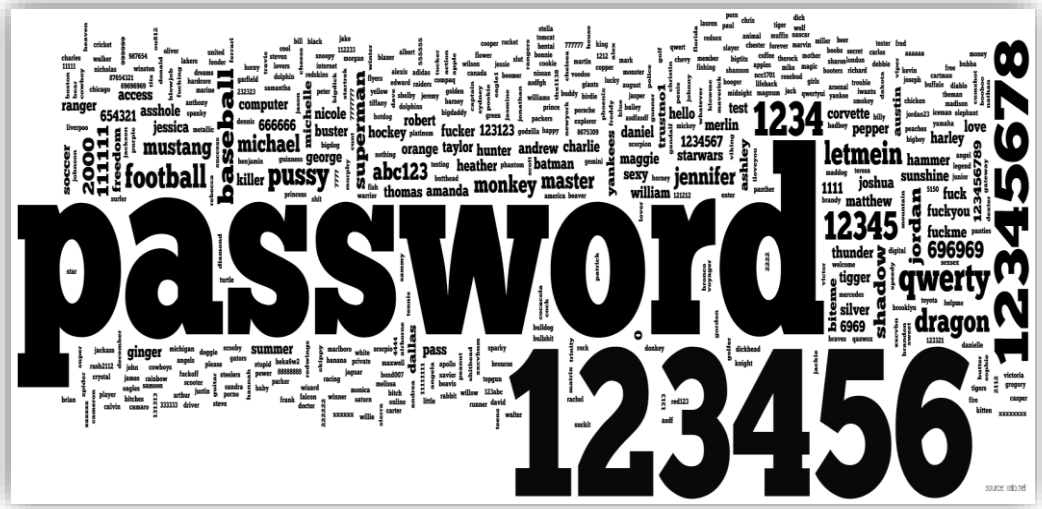
資安的概念與範圍

出處：IThome

年度網路攻防演練發現重要事項

1. 集中使用相同套件或委外廠商
2. 未即時更新使用之軟體套件
3. 不正確的安全觀念
4. 未落實通行碼強度檢查的機制
5. 不當的轉導設計

2018 排名	密碼
1	123456
2	password
3	123456789
4	12345678
5	12345
6	111111
7	1234567
8	sunshine
9	qwerty
10	iloveu



DEMO!

■ 防範建議

- 建議刪除或停用預設帳號，再另行建立管理者帳號；若預設帳號無法更動則至少須修改預設通行碼
- 通行碼建議具備高複雜度，同時禁止使用者設定與帳號相同之通行碼

常見弱點說明及案例

出處：IThome

年度網路攻防演練發現重要事項

1. 集中使用相同套件或委外廠商
2. 未即時更新使用之軟體套件
3. 不正確的安全觀念
4. 未落實通行碼強度檢查的機制
5. 不當的轉導設計

DEMO!

■ 防範建議

- 應對所有功能頁面進行權限控管，避免將未授權之功能頁面併入檢查結果中回傳，以防遭攻擊者竄改進而繞過檢查機制
- 所有檢查應於伺服器端進行，僅回傳必要之檢查結果
- 錯誤頁面正確導轉

資安的概念與範圍

■ Information Security (CIA)

Confidentiality 機密性

資料於網路傳送時被攔截竊取，或公司不小心公佈不該公佈的訊息均是違反資料的機密性

- 1元買義大門票　ibon網頁關閉修正漏洞
- 江蕙演唱會售票系統有漏洞 可篡改售票金額

Integrity 完整性

從網路上下載的檔案，是否真的是來源端原本檔案。
如：所得稅軟體下載

Availability 可用性

系統的高度可用性通常指的是必需確保不能中斷服務

資安的概念與範圍

■ Information Security (AAA + NR)

多因子認證 (Multi-factor authentication)
Something you know：
身分證字號、密碼、出生地及其他所知道的事。
Something you have：
動態密碼鎖、鑰匙、晶片卡及其他所擁有的事物。
Something you are：
性別、指紋、視網膜及其他天生的特徵。

Authentication 認證

辨別資訊使用者的身份，即可以記錄資訊是被誰使用過

Authorization 授權

依照身份給予適當的權限

Accounting 可歸責性

組織內有許多部門與個人，當事件發生時該由誰負責處理必須明確規定

Non-Repudiation 不可否認

防止存心不良的使用者否認其所做過的事，包括送出信件，接收文件，存取資料等

資安的概念與範圍

■ Information Security (STRIDE)

STRIDE is a threat classification model developed by Microsoft for thinking about computer security threats.

資安的概念與範圍

- Information Security (STRIDE)

Spoofing 詐騙

偽造身分：在工作期間假冒另一個使用者的身分

Tampering 竄改

竄改資料：就只是改變資料，竄改資料不見得是資訊揭露，竄改資料只有達到資料的污損。會造成資料被汙染、破壞資料可信度、政府機關形象受損、企業商譽損失、干擾業務正常運作 (補充)

Repudiation 否認

否認：就攻擊者的觀點，否認就是隱匿蹤跡，讓某件行為無法被追蹤，無法歸咎於肇事者。

資安的概念與範圍

■ Information Security (STRIDE)

Information Disclosure 資訊洩漏

資訊揭露：允許未授權的人取用敏感資訊，像是信用卡號碼、密碼……等。

Denial of Service 服務阻斷

服務阻斷（DOS）：讓某種資源耗竭，例如網路頻寬、處理器運算能力、磁碟空間。DOS攻擊相當容易匿名發動，所以有時候會難以判斷是否為惡意攻擊。

Example : https://www.youtube.com/watch?v=OWLGUgiz_eE

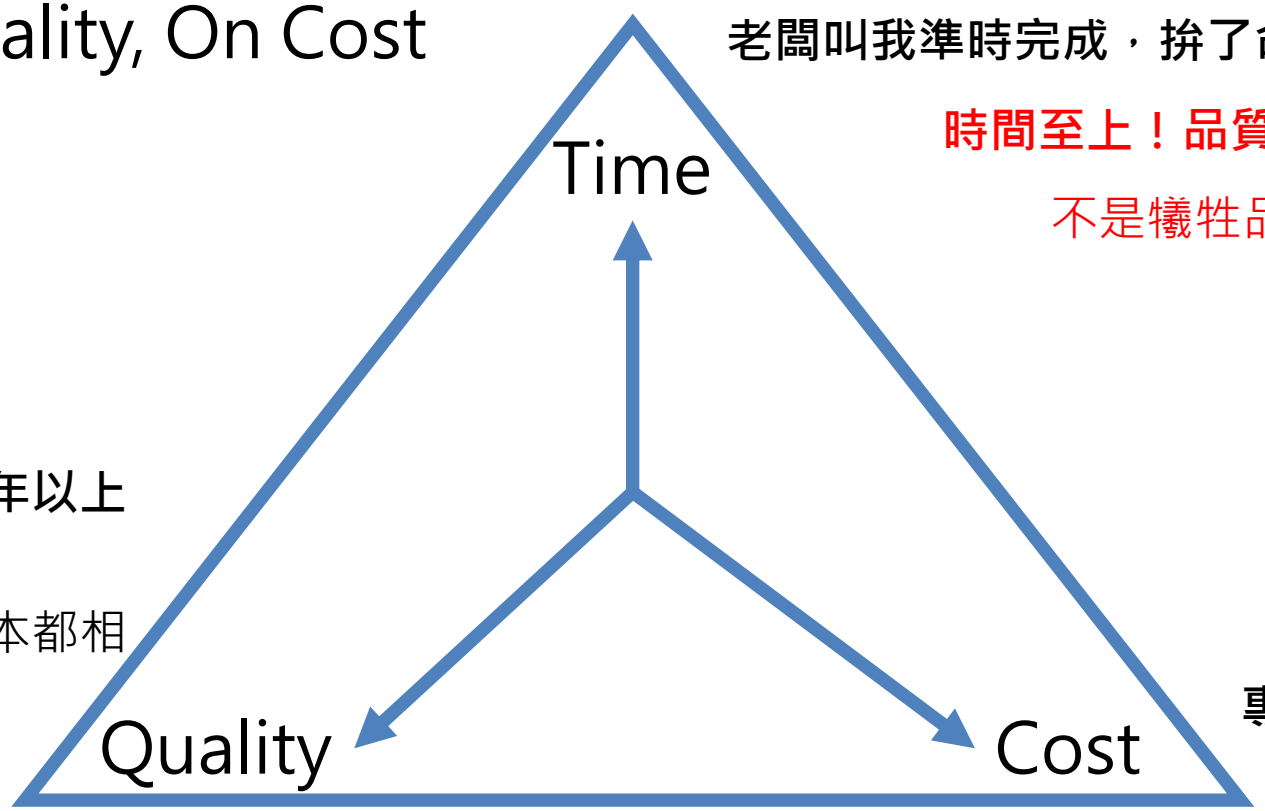
Elevation of Privilege 權限提高

特權提升：不具有特權的使用者或行程，獲得特權的存取，如：偷偷建立一個具有管理者權限的角色。

資安的概念與範圍

■ 專案管理目標(Project Management)

On Time, On Quality, On Cost



老闆叫我準時完成，拚了命先上功能再說

時間至上！品質功能其次

不是犧牲品質就是花了更多資源

一個系統開發至少超過一年以上

品質至上

花費的人力成本與時間成本都相當可觀

專案預算降低!!!

不能賠錢為原則

影響到品質與投入的人力成本

資安的概念與範圍

- 安全軟體開發生命週期
- Security Software Development Life Cycle (SSDLC)

需求

資訊系統委外開發RFP資安需求範本

資安風險評估

設計

安全軟體設計指引

系統設計

分析攻擊層面

威脅模型確認

實作

應用程式參考指引

現有檢測流程

靜態分析報告檢閱

程式碼檢視流程

測試

安全軟體測試指引

動態分析流程

模糊測試流程

攻擊面審查

驗收

系統安全需求實作與驗證

使用者回饋調整

版本追蹤

事件處理機制

弱點分析

滲透測試

The Microsoft Security Development Lifecycle – Simplified
NCCST.nat.gov

資安的概念與範圍

1. 資訊安全推動的難度：“推動資安會增加工作負擔，影響正常作業”，企業及管理人員存有僥倖心態，以為資安事件不會恰巧發生在自己身上，因此忽略資安的重要性。

2. 資安問題可以一次解決：以為建立完美的資安防禦體系就可以高枕無憂? 其實資訊安全為永無停止的攻防戰，今天安全的東西明天就被破解了。

3. 建置強大的資安產品可杜絕資安事件：只靠防火牆或防毒軟體效果有限，必須結合相關人員資安認知與能力。

2. 軟體系統安全設計原則

軟體系統安全設計原則

■ Security by Design Principles

01 Minimize attack surface area
減少攻擊層面

02 Establish secure defaults
建立安全機制

03 Least privilege
最小權限

04 Defense in depth
縱深防禦

05 Fail securely
安全機制

06 Don’t trust services
不信任的服務

07 Separation of duties
權責分離

08 Avoid security by obscurity
避免安全隱患

09 Keep security simple
保持簡單

10 Fix security issues correctly
正確解決安全議題

軟體系統安全設計原則

■ Minimize attack surface area(減少攻擊層面)

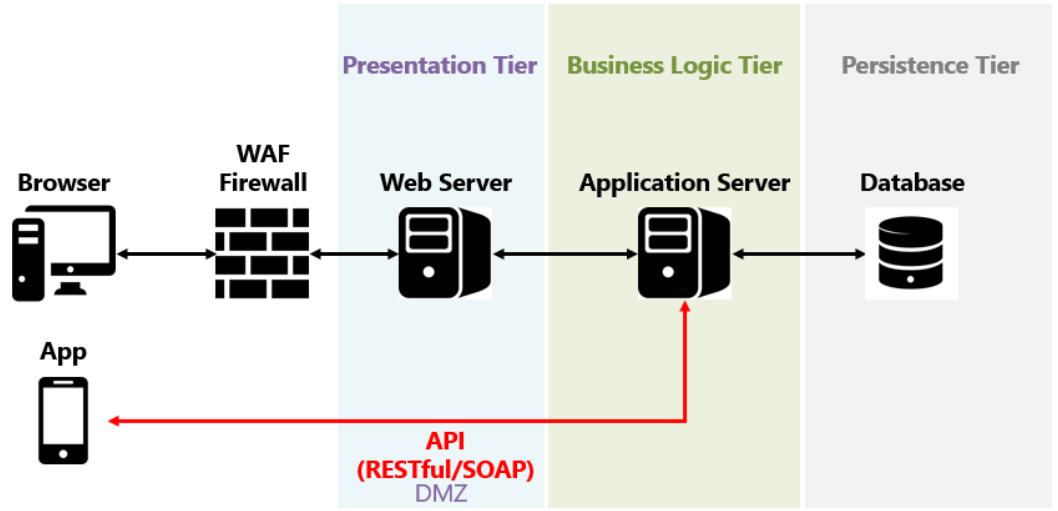
Why?

- Like Keep security simple Security by Design Principles
- Web site need, mobile also want.

Protection

- 新增功能置資訊系統中都會給整個程序帶來一定的風險。主要目的是盡可能減少可能的接觸面積來降低總體風險。
- Example：使用者搜尋功能已有Injection防護，並於後端資料庫限制使用者存取權限。

Example



More points of interaction
=
More difficult to defend

軟體系統安全設計原則

■ Establish secure defaults(建立安全機制)

Why?

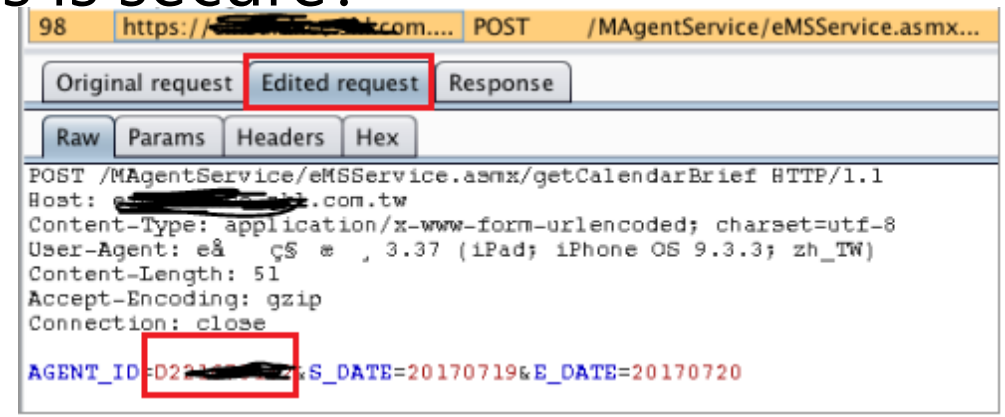
- Always next…next...next...next install. (Use default setting)
- Maybe the system should be secure by default

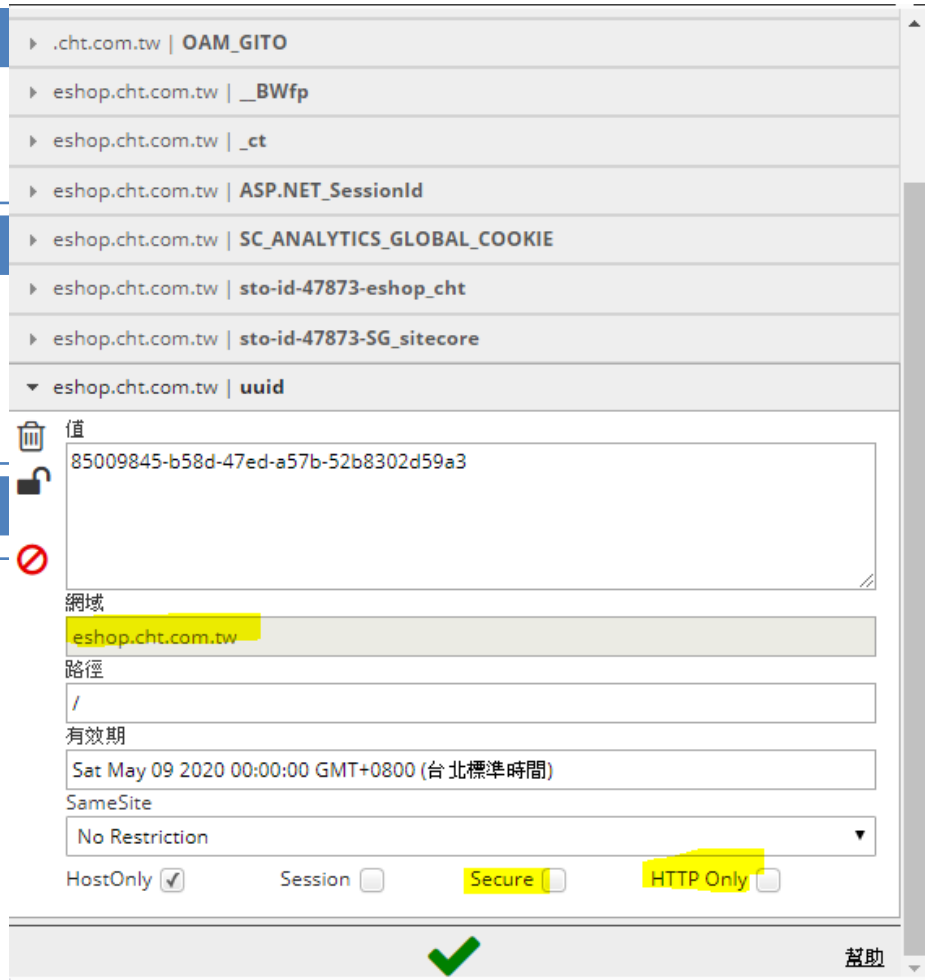
Protection

- 不過度依賴專門配置或啟用基本安全功能的設定
- 具備一定的基本安全性設定(例如SSL、CSP header政策等)

Example

Https is secure?





軟體系統安全設計原則

■ Least privilege(最小權限)

Why?

- Always develop use sudo…
- Open unnecessary service ports..

Protection

- 開發過程中應要使用最小的權限來執行，讓大家能夠完成任務即可，不要拿過多的權限。
- 關閉系統中預設管理者權限，包含Admin, Root等預設帳號。
- 一般使用者存取資料庫時，只允許存取對應的資料表權限。

Example

卡巴斯基實驗室指出，IoT裝置的安全威脅不再是概念性的，而已非常真實，全球的IoT裝置數量將從目前的數十億成長到2020年的200至500億台，更顯現其安全問題的急迫性。

在製造商還未採取行動的狀況下，卡巴斯基實驗室建議使用者不要讓裝置曝露在公開網路上，關閉裝置未使用的所有網路服務，變更裝置的預設憑證，以及定期更新韌體，只要遵行這幾項建議，即可躲過大多數的IoT惡意程式。

iThome 電腦報 Weekly
按讚追蹤 iThome 最新報導

勒索病毒肆虐 桃園市政府關閉漏洞服務埠

自由時報 Liberty Times Net

2017-05-15 23:17

〔記者邱奕統／桃園報導〕桃園市府研考會今天到市議會工作報告，多位市議員針對「WannaCry」電腦勒索病毒，關心市府是否做好因應措施，研考會表示，今天上班前即已關閉開45PORT與139PORT等有問題的服務埠，並遵照中央資安辦公室指示，隨時保持作業系統更新、修補，並做好資料備份。

多位市議員今天關切市府勒索病毒防治情形。（記者邱奕統攝）

研考會代主委邱俊銘表示，市府包含13區公所在內，共有2萬多個使用者，配合行政院政策，本來就持續辦理相關資訊安全作業，作業系統也都有定期更新、修補，並做好資料備份，至於Windows XP、Windows Server 2003也因一年多前通知後續不再進行支援，陸續汰換。

他說，除關閉有問題的服務埠，也通知所有同仁今天上班開電腦時，先不要連接網路，務必先確認漏洞有修補、作業系統有更新，也提醒不要點來源不明網址，目前各局處公務電腦都沒有受害情形。

軟體系統安全設計原則

■ Defense in depth(縱深防禦)

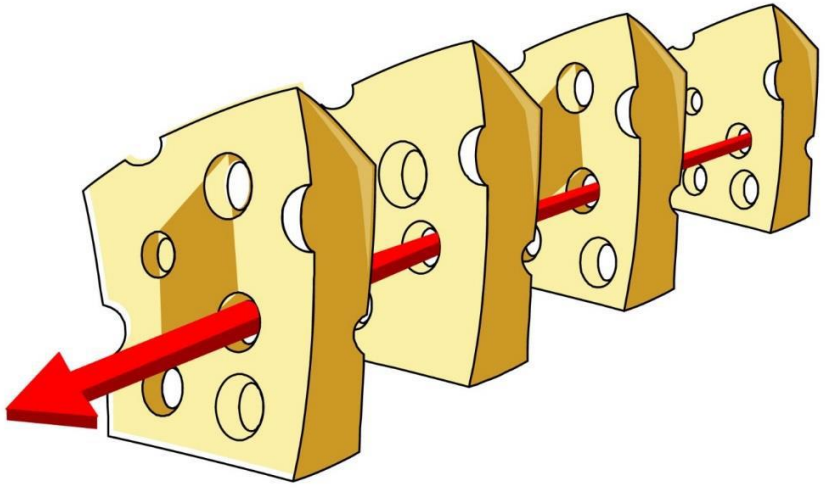
Why?

- On a single security method to protect everything.
- Restart or shutdown is the best method?

Protection

- 就像乳酪理論有很多層防禦，系統的防禦也應該從 UI, API, Database 都作防禦。
- 避免被駭客單點突破後，整個系統就被予取予求。

Example



軟體系統安全設計原則

■ Fail securely(安全機制)

Why?

- 當系統遇到異常時，必須有一套異常處理機制

Protection

- Example：IPS或防火牆設備遇到停電，應所有流量都不准過
- Example：門禁系統遇到停電，應該要讓所有人都能自由進出

Example

https://www.asd-usa.com/blog/fail-safe-or-fail-secure-for-the-access-control-safety-and-security-of-your-workplace-you-need-to-know-the-difference/

```
isAdmin = true;
try {
  codeWhichMayFail();
  isAdmin = isUserInRole( "Administrator" );
}
catch (Exception ex) {
  log.write(ex.toString());
}
```

OWASP Security by Design Principles

If this problem. You can?

I don't know...人心難測...

軟體系統安全設計原則

■ Don’t trust services(不信任的服務)

Why?

- Store sensitive information in external services. (like SQLite, cookie, file etc.)
- Very cheap, click here. (Social Engineering)

Protection

- 加解密技術(Encrypt / Decrypt)、雜湊技術(Hash)、簽章(Sign)。
- 白名單機制(Whitelist) 、憑證綁定(Certificate pinning)。

Example

趨勢

勒索軟體威脅全台十餘家醫院，密碼管理成漏洞

2019/09/02 ‧ 中央社 ‧ WannaCry、密碼、醫療、資安、勒索病毒、醫院

全台約有 16 至 18 家醫療院所有類似情況，但因被鎖住的系統都非核心系統，不影響門急診、資料也沒遺失，「事情並不難處理，也沒有醫院付贖」

軟體系統安全設計原則

■ Separation of duties(權責分離)

Why?

- Because my boss…….tell me do everything by myself.

Protection

- 不管是在商業邏輯或是系統層級，都應該做到權責分離。例如：付款模組跟訂單模組應該要分開，前端介面跟後端資料庫要分開。
- 開發人員、測試人員、上版人員均有各自的工作責任，避免錯誤發生。

Example

From:【root】<rootmail@cht.com.tw>
Sent: Wednesday, July 29, 2020 5:07 PM
Subject:【TL訊息】:個資與風險評鑑(ISRMS)系統服務暫停通知

各位主管及同仁 鈞安：

總公司來信，因內雲ECC基礎服務異常，導致個資與風險評鑑(ISRMS)系統服務暫停，待系統修復後，會再發信通知，若造成您的不便，敬請見諒!

軟體系統安全設計原則

■ Avoid security by obscurity(避免安全隱患)

Why?

- Using third party library and not update.
- Key and password save in the code.

Protection

- 定期檢測第三方套件(Open Source)，確保無已知安全性風險。
- 避免將敏感性資訊Hardcode於程式碼或文檔中。

Example

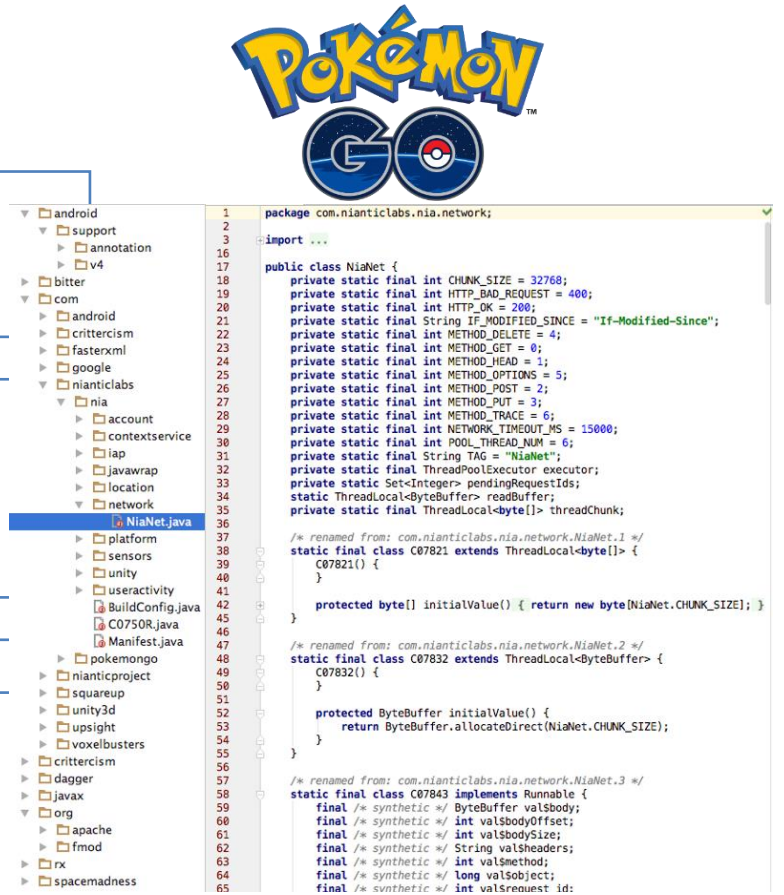
```
public void saveCredentials(String userName, String password) {

    SharedPreferences credentials = this.getSharedPreferences(
            "credentials", MODE_WORLD_READABLE);
    SharedPreferences.Editor editor = credentials.edit();
    editor.putString("username", userName);
    editor.putString("password", password);
    editor.putBoolean("remember", true);
    editor.commit();
}
```

— Very Bad

— Convenient!

Hardcoded Password !

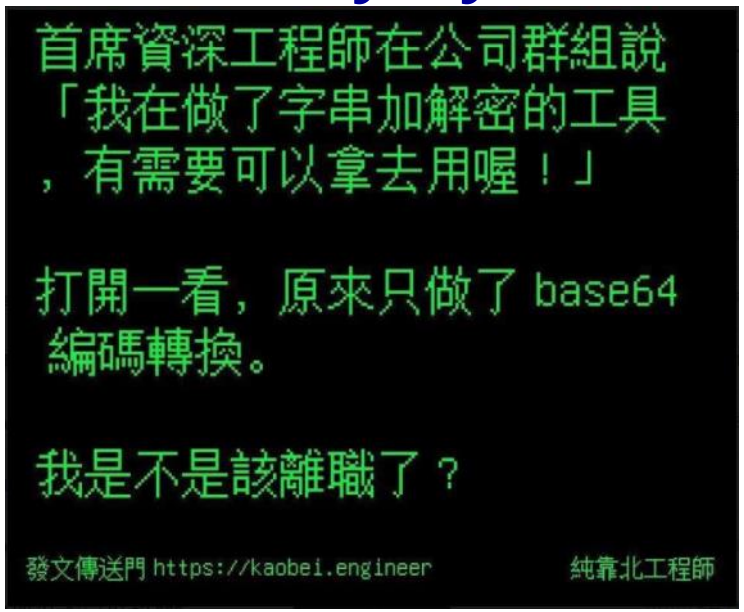
```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <int name="ITS_DeletePass" value="0" />
    <string name="OBUid">49676351</string>
    <string name="MsgType">31</string>
    <string name="OTAHost">http://xxx.hinet.net:80</string>
    <string name="COUNT">14</string>
    <string name="Date">2015/03/17 14:47:35</string>
    <string name="UserAuthResult">0</string>
    <int name="ITS_ReStartAPP" value="0" />
    <string name="ID">chttest@smartphone</string>
    <int name="ITS_NOT_RESENDLOCATION" value="0" />
    <int name="ITS_Notes" value="1" />
    <string name="MediaServer">http://xxxx.hinet.net:80</string>
    <string name="PASS">chtpassword</string>
    <int name="GpsSpeed" value="0" />
    <string name="TIME">2015/03/17 14:47:36</string>
    <string name="Punchin_Picture_Url4"></string>
    <int name="speed_limit_1" value="0" />
```

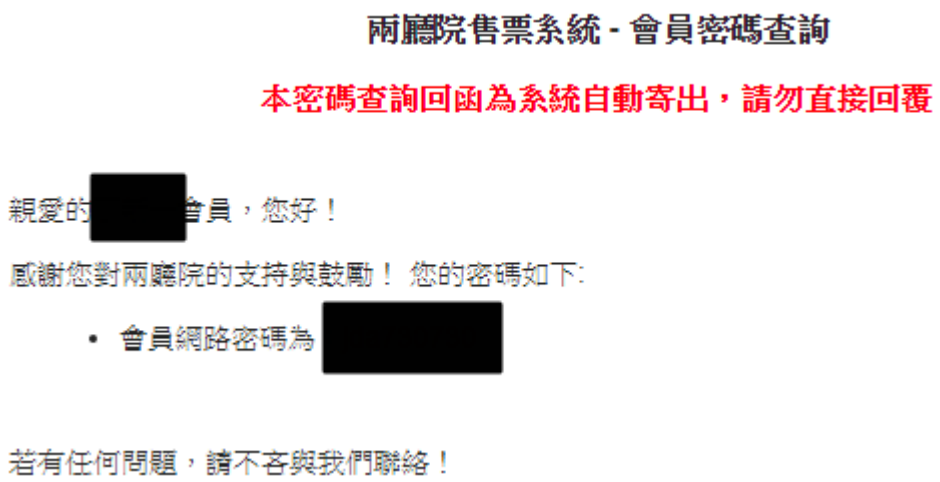


Reverse is very easy!!!

軟體系統安全設計原則

■ Avoid security by obscurity(避免安全隱患)





安全敏感性資料應採用適當且有效之金鑰長度與加密演算法

- 採用金鑰有效長度為128位元（含）以上之先進加密標準（AES）
- 採用三重資料加密演算法（Triple DES）
- 符合ANSI X9.17、FIPS 140-2、NIST SP 800-22 及 SP 800-90A (CAVP Testing: Random Number Generators) 至少其中一項安全的亂數產生函式

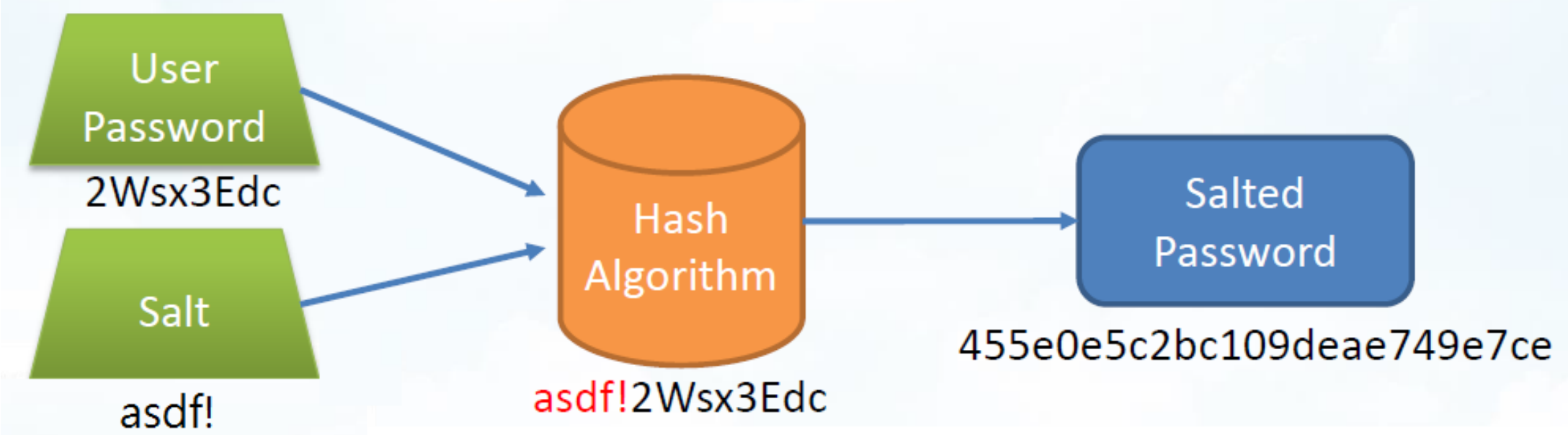
軟體系統安全設計原則

JAVA程式範例
```
MessageDigest digest =
MessageDigest.getInstance("SHA-256");
```

■ Avoid security by obscurity(避免安全隱患)

➢ Salted Password Hashing

- 產生使用者密碼時，使用Salted Password Hashing，降低遭Rainbow table破解的機率



密碼儲存進行Hash時，應採用單向雜湊的方式進行(例如SHA-256)，避免被回推
**儲存方式不建議採用加解密，可進行加密就代表可被解密回來

軟體系統安全設計原則

■ Keep security simple(保持簡單)

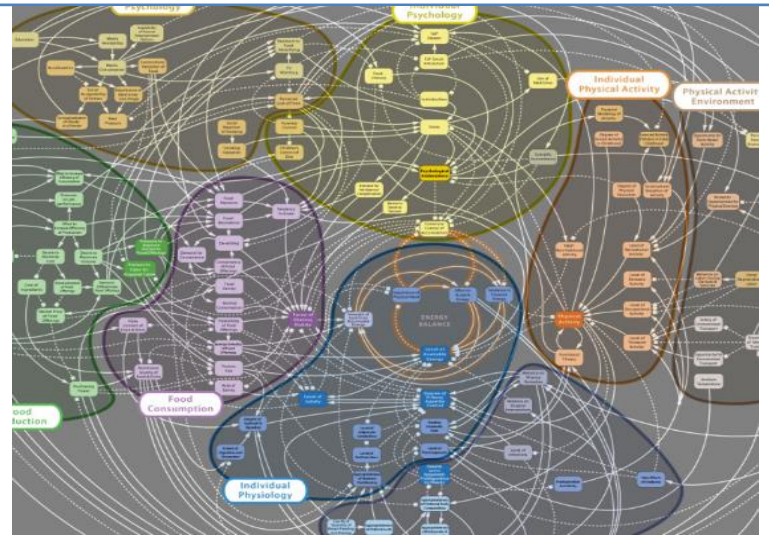
Why?

- Demand unit：I need all the features on the market!!!
- Complex System.

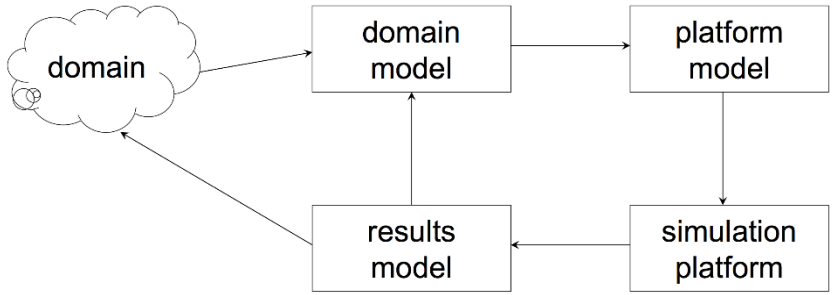
Protection

- 系統所需的功能愈少，所運用的資源愈少，檢查的內容就會相對容易與設計。
- 複雜系統應切模組區分，針對各模組有各自獨立的檢查項目。

Example



VS



$E=MC^2$????

Error = (More Code)^2

軟體系統安全設計原則

Fix security issues correctly(正確解決安全議題)

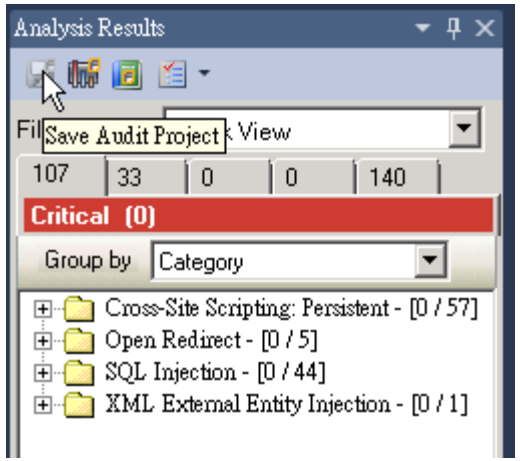
Why?

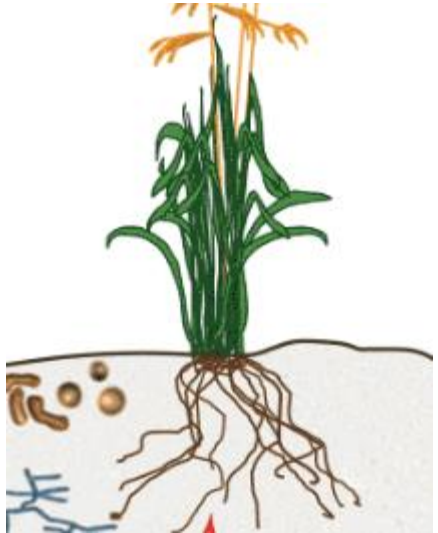
- 先上版在說!
- Develop time not enough...

Protection

- 弱點發生進行修復時，須確認弱點影響範圍與資料流路徑，避免盲目的修復弱點
- Code Review

Example





You see

The real problem

3. 軟體系統安全測試方法

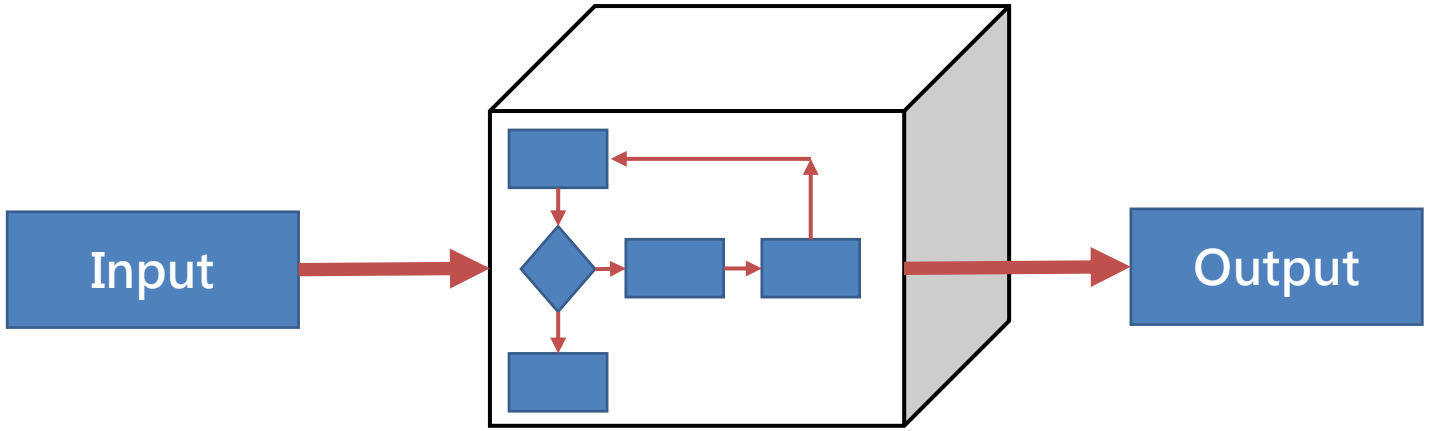
包含原始碼、網頁、主機、開源等弱點掃描之重點及其價值

軟體系統安全測試方法

■ 靜態檢測分析(Static Application Security Testing，SAST)

簡稱白箱、原始碼弱點檢測說明：靜態分析的一種方式，藉由分析程式碼可能的所有執行路徑，找出其中的風險(Data Flow → Control Flow → Tainted Analysis)。

- 必須要：完整的程式碼；支援檢測的語言與編譯器；每次更版都掃瞄。
- 限制：容易有誤報(False Positive)，需針對檢測結果再確認。



軟體系統安全測試方法

- **動態檢測分析(Dynamic Application Security Testing，DAST)**

  簡稱黑箱、網頁弱點檢測、DAST

  - 說明：動態分析的一種方式，透過找出系統所有頁面、所有輸入欄位，針對輸入欄位進行攻擊、驗證，檢驗系統弱點(Send Request → Check Response)。
  - 必須要：線上環境或與線上環境相同之測試環境；每次更版都掃瞄；錄製腳本後才得以測試。
  - 限制：容易有漏報(False Negative)；對系統效能將造成影響；會寫入資料進入系統當中

軟體系統安全測試方法

■ 主機弱點檢測(Host-based Vulnerability Assessment, VA)

簡稱主機檢測、弱點評估

說明：屬於動態分析的一種方式，掃描網路環境中各種網路設備與系統主機，檢測是否存在已知的弱點

必須要：對系統中能連上網的主機設備所開放的服務及其相關環境設定進行弱點掃描，送出各種類型之網路封包或攻擊指令，以測試目標系統之回應，並從回應之訊息來判斷各項系統漏洞及其他資訊

限制：無法測試企業防火牆的強度，需要開通防火牆才能進行較準確的掃描。(有權限與無權限檢測結果差很多!!)

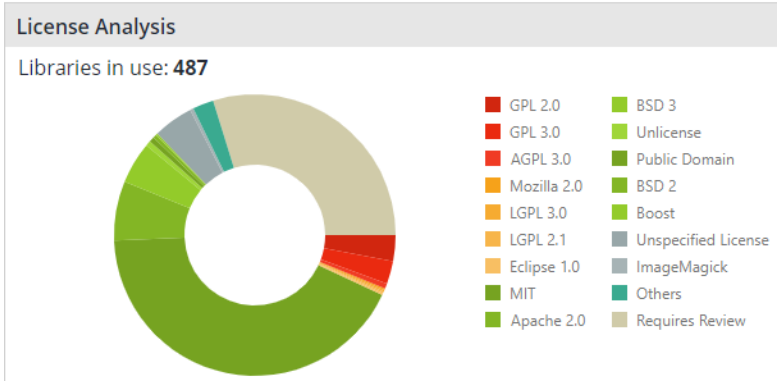
軟體系統安全測試方法

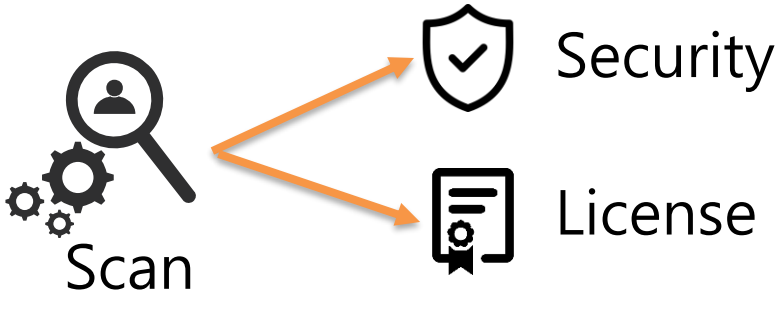
■ 開放原始碼安全檢測(Open Source Security，OSS)

說明：是一套針對Open Source元件管理工具，與資料庫交叉比對的結果(雜湊值運算)，辨識您的第三方元件是否有弱點，版本更新狀況及License有效性。

必須要：提供系統所引用的第三方元件(Library、Source)，並定期追蹤專案所引用的套件是否含有已知安全性風險與漏洞(CVSS，CVE)。

限制：多數開發人員完全不清楚引用多少Open Source與是否違反GPL/AGPL。將近 40% 的開發人員每個月花20~60個小時在處理Open Source弱點(GSS提供)。





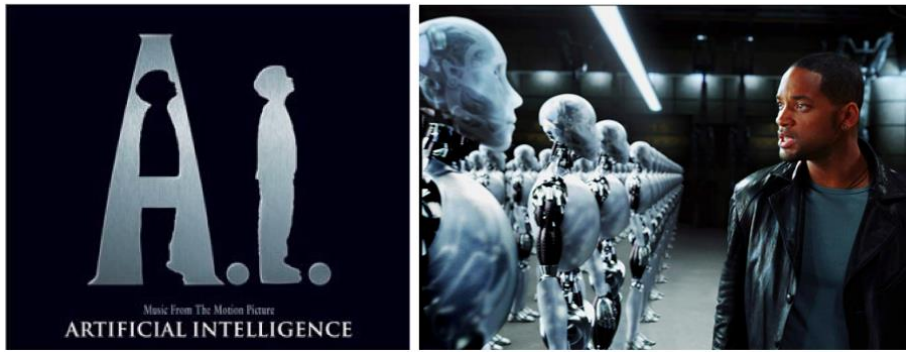
軟體系統安全測試方法

■ 弱點檢測確認事項

		原始碼	網頁	主機
01	確認掃描工具版本(Version)	V	V	V
02	程式語言是否支援	V	-	-
03	確認掃描規則(Rule packs)	V	V	V
04	檔案之檔案數、行數	V	-	-
05	未出現非必要之Error或Warning	V	V	V
06	資料流會互相影響之系統，應在同一檢測下完整執行	V	-	-
07	掃描頁面與實際系統頁面相符合	-	V	-
08	確認掃描之執行日期	V	V	V
09	確認掃描標的包含所有關聯之伺服器IP	-	-	V
10	確認各掃描標的開啟之服務，與用戶使用之服務一致	-	-	V

軟體系統安全測試方法

那就採用自動化檢測工具替公司軟體安全進行防護!!



但..如果有這一天就是例外狀況

- 只能說將風險降低!!
- 因為駭客不是軟體！軟體是死的、駭客卻是活的。每天都有非常多新的駭客手法、安全漏洞等，如果沒有與時俱進，該如何與駭客對抗呢？
- 雖然資安工具會定時更新掃描規則，然而在你等待更新的同時，你的網站、主機也許已經被攻破了
- 工具無法精確的對系統「邏輯思維」進行測試，因而無法進一步找出邏輯漏洞。唯有依賴人工的精準測試，才能真正找出問題

軟體系統安全測試方法

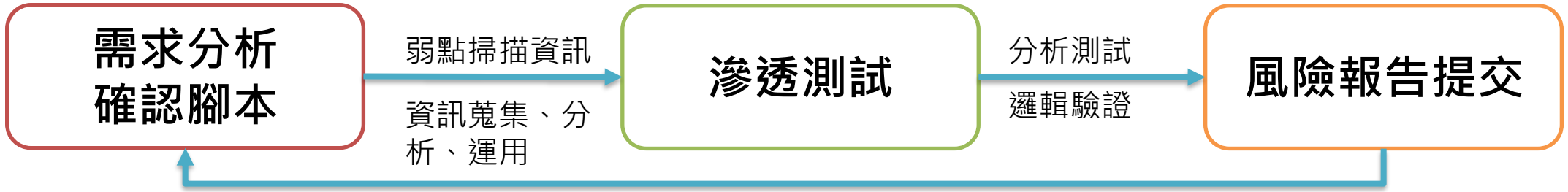
■ 滲透測試(Penetration Test，PT)

➢ 白帽駭客

說明：由具開發經驗與資安技術背景的專業人員，透過模擬駭客的攻擊方法與思維，利用各種工具與途徑嘗試入侵，評估伺服器及網路環境的整體安全性，提供修補建議並輔助修補漏洞。

必須要：弱點掃描無法找出人類才能辨識及驗證的問題，例如：商業邏輯漏洞、身份權限跨越漏洞、表單上傳漏洞等。

限制：滲透測試人員的知識與能力。檢測時程相對較長。



4.資訊系統委外開發RFP資安需求
政府組態基準(GCB)

應用程式的安全性與系統管理

■ 資訊系統委外開發RFP資安需求

- ➢ 主要為提供政府機關於資通系統委外開發時，撰寫建議書徵求說明書(RFP)，**訂定資通系統資安需求**之參考。包含「存取控制」、「稽核與可歸責性」、「識別與鑑別」、「系統與服務獲得」、「系統與通訊保護」及「系統與資訊完整性」等構面。

■ 政府組態基準(Government Configuration Baseline，GCB)

- ➢ 政府組態基準(GCB)目的在於**規範資通訊終端設備**(如:個人電腦、伺服器)的**一致性安全設定**(如:密碼長度)，以降低成為惡意使用者入侵的管道。包含「帳戶安全性」、「個人電腦系統」、「防火牆設定」、「使用者設定」等。

資訊系統委外開發RFP資安需求

重於開發

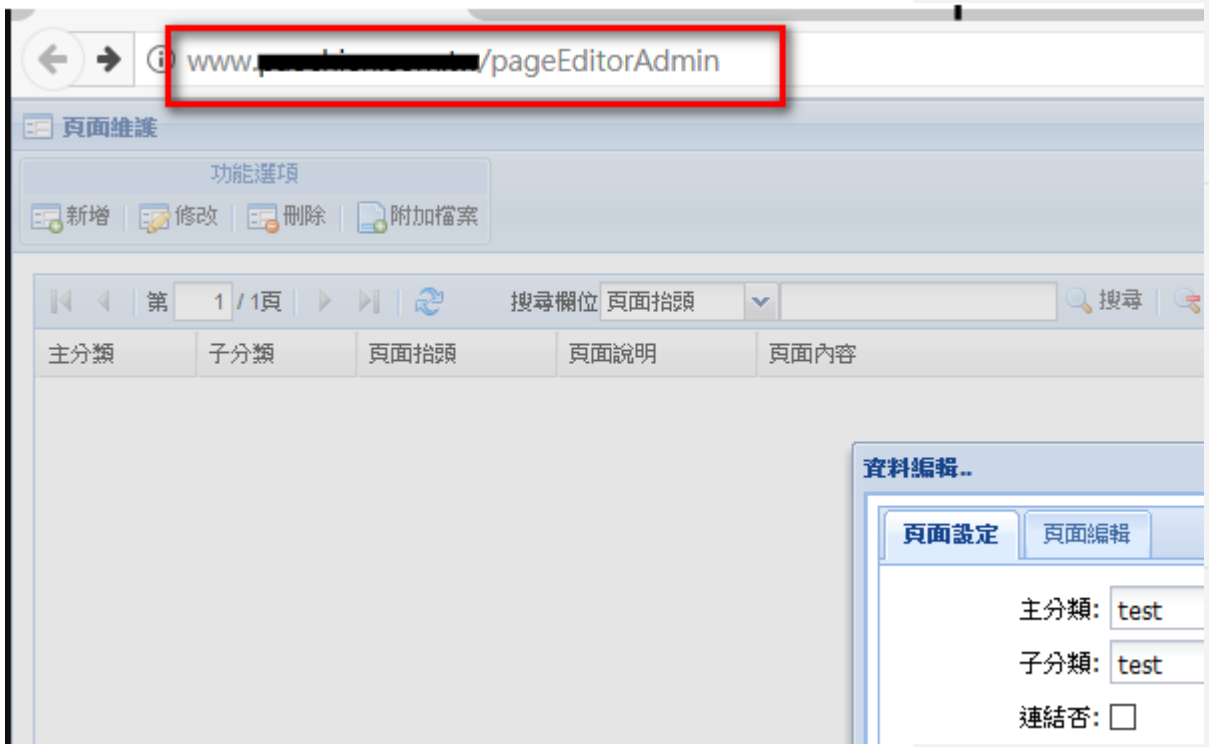
- **資訊系統委外開發RFP資安需求(V1.0)**
  - 以資訊系統安全需求為主，依軟體安全特性各項類型，包含「機密性」、「完整性」、「可用性」、「身分驗證」、「授權與存取控制」、「日誌紀錄」、「會談管理」、「錯誤與例外處理」、「組態管理」，訂定共50項安全需求項目
- **資訊系統委外開發RFP資安需求(V2.0)**
  - 針對資通系統之防護需求等級(普、中、高)訂定資安需求項目，包含「存取控制」、「稽核與可歸責性」、「識別與鑑別」、「系統與服務獲得」、「系統與通訊保護」及「系統與資訊完整性」等構面，共計41項技術面及13項管理面資安需求項目

重於開發、維運

資訊系統委外開發RFP資安需求

- **存取控制**
  - 確訂定資通系統之存取限制

猜測路徑(Dirb)



詳細資料

ZDID：ZD-2017-01036

通報者：elijiachen (Elijia)

風險：低

類型：存取控制缺陷 (Broken Access Control)

http://[redacted]/pageEditorAdmin

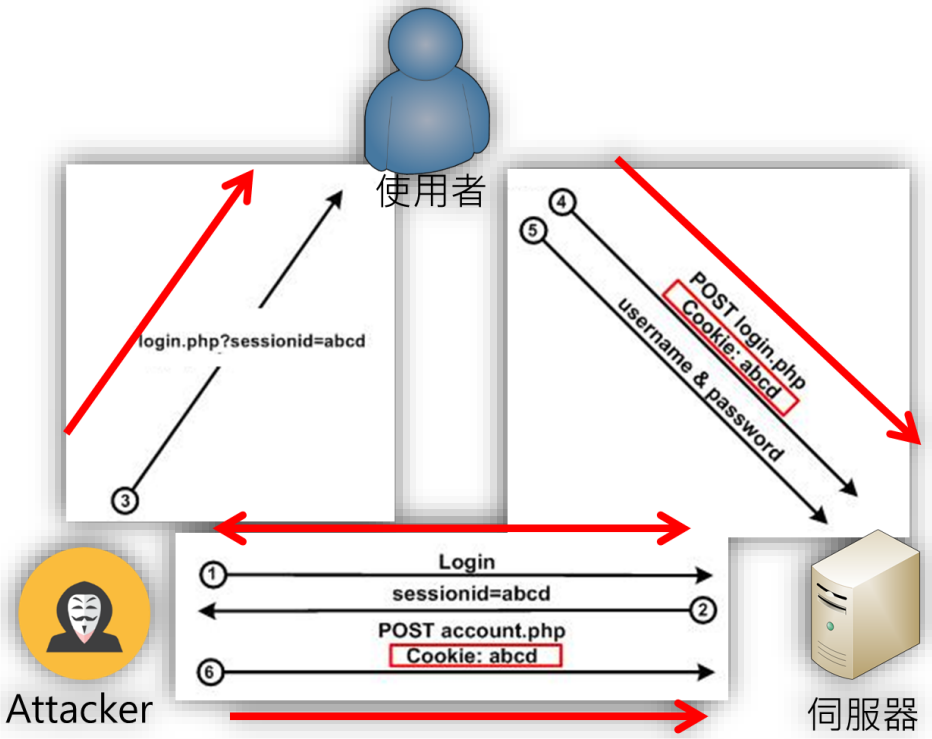
網站後台存取設置不當

資訊系統委外開發RFP資安需求

■ 存取控制

➢ 會談(Session)機制目的

Session Fixation



Session Timeout

```
<configuration>
   <system.web>
      <sessionState timeout="10" />
   </system.web>
</configuration>
```

session_unset
(PHP 4, PHP 5, PHP 7)

session_unset — Free all session variables

Description
session_unset (void) : bool

The session_unset() function frees all session variables currently registered.

Return Values
Returns TRUE on success or FALSE on failure.

所有

session_destroy
(PHP 4, PHP 5, PHP 7)

session_destroy — Destroys all data registered to a session

Description
session_destroy (void) : bool

session_destroy() destroys all of the data associated with the current session. It does not unset any of the global variables associated with the session, or unset the session cookie. To use the session variables again, session_start() has to be called.

當前

資訊系統委外開發RFP資安需求

- 系統與服務獲得
  - 發生錯誤時，使用者頁面不包含詳細之錯誤訊息
  - 系統上線要將註記刪除

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.web>
        <customErrors mode="On">
    <error statusCode="404" redirect="/NotFound/SystemWeb404.html"/>
        </customErrors>
    </system.web>
    <system.webServer>
        <urlCompression doDynamicCompression="true" />
        <httpErrors>
            <remove statusCode="404" subStatusCode="-1" />
            <error statusCode="404" prefixLanguageFilePath=""
              path="/NotFound/SystemWebServer404.html" responseMode="ExecuteURL" />
        </httpErrors>
    </system.webServer>
</configuration>
```

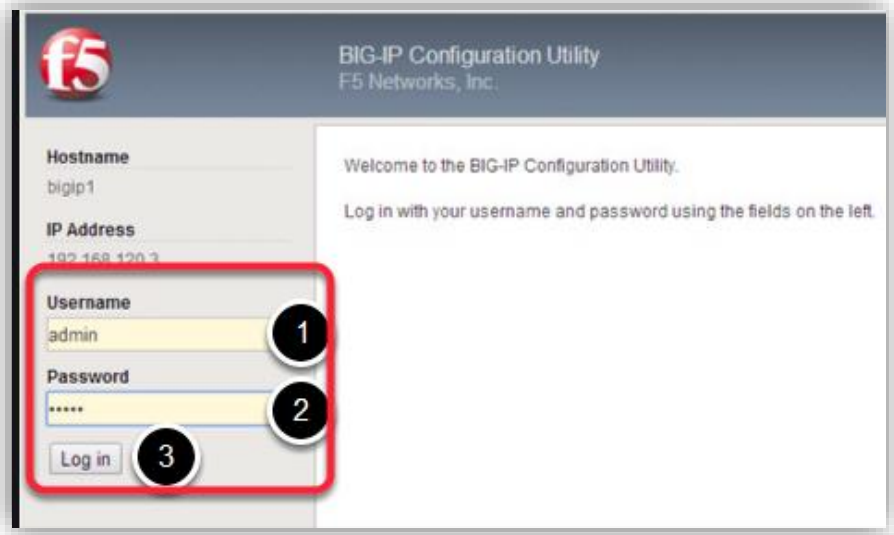
顯示過多錯誤資訊給使用者

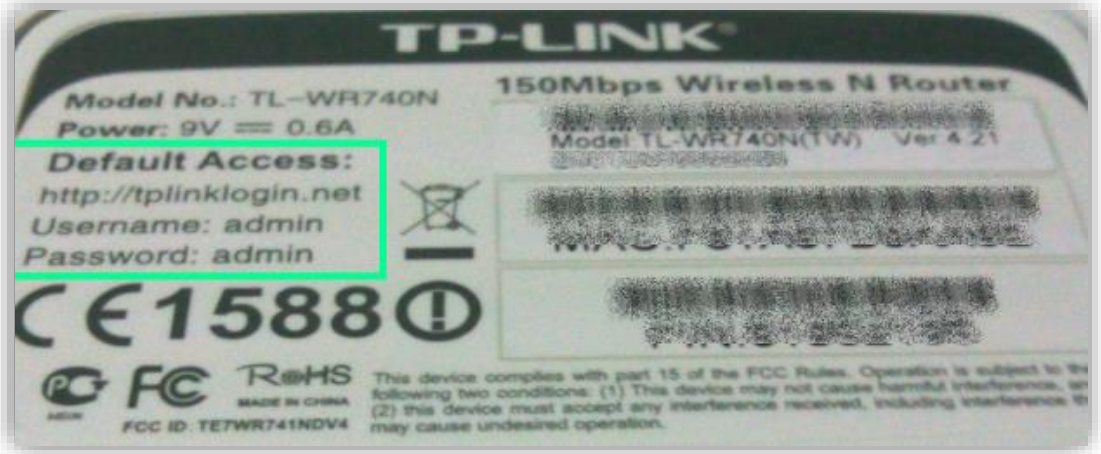
```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
Set-Cookie: ASP.NET_SessionId=oaobe3ddwvi2pjgs4xemxdyp; path=/CHT; secure; HttpOnly
X-AspNetMvc-Version: 5.2
X-Powered-By: ASP.NET
Strict-Transport-Security: max-age=31536000
Date: Wed, 27 May 2020 08:24:39 GMT
Connection: close
Content-Length: 1323

err:System.Web.Services.Protocols.SoapException: 伺服器無法處理要求。 ---> MySql.Data.MySqlClient.MySqlException: Data
too long for column 'HospName' at row 1
   於 MySql.Data.MySqlClient.MySqlStream.ReadPacket()
   於 MySql.Data.MySqlClient.NativeDriver.GetResult(Int32& affectedRow, Int64& insertedId)
   於 MySql.Data.MySqlClient.Driver.NextResult(Int32 statementId, Boolean force)
   於 MySql.Data.MySqlClient.MySqlDataReader.NextResult()
   於 MySql.Data.MySqlClient.MySqlCommand.ExecuteReader(CommandBehavior behavior)
   於 MySql.Data.MySqlClient.MySqlCommand.ExecuteReader()
   於 MySql.Data.MySqlClient.MySqlCommand.ExecuteNonQuery()
   於 SQLCommonOrder.doSQLNonQuery(String[] cloud, String connectionString) 於
c:\inetpub\code\Maria\20200527\API\App_Code\DatabaseLayer\Common\SQLCommonOrder.cs: 行 423
   於 HealthRecordMaintainAccess.updateHospital(String ls_index, String EmpID, String HospitalId, String
HospitalName) 於 c:\inetpub\code\Maria\20200527\API\App_Code\DatabaseLayer\Admin\HealthRecordMaintainAccess.cs: 行
260
```

資訊系統委外開發RFP資安需求

■ 系統與服務獲得

➢ 資通系統相關軟體，不使用預設密碼



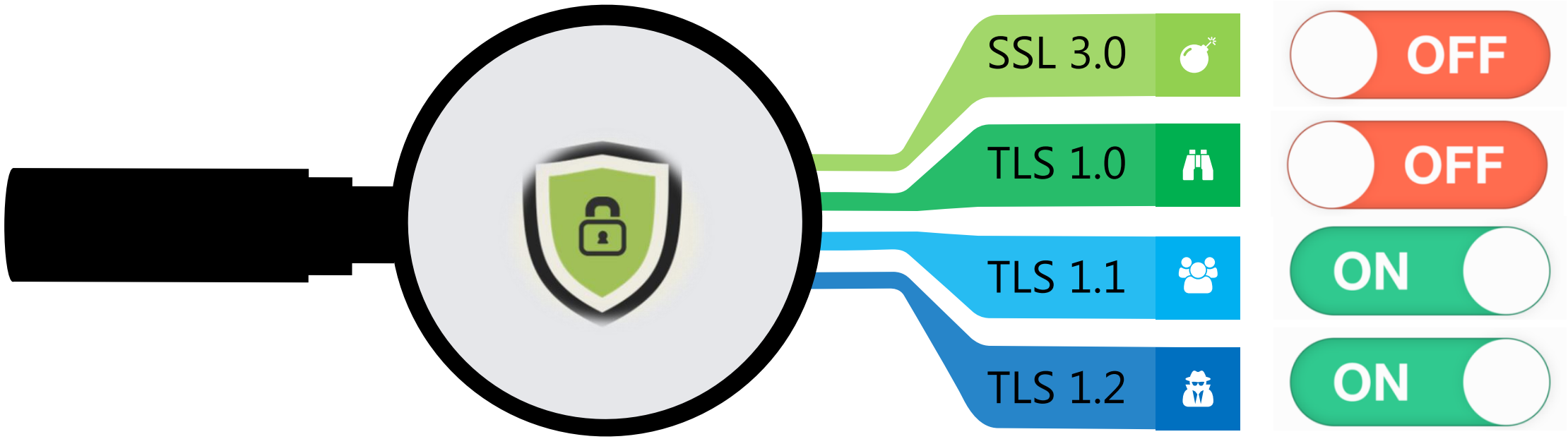


資訊系統委外開發RFP資安需求

- **系統與通訊保護**
  - ➢ 資訊系統傳輸機敏資料時，應避免明文傳輸

補充說明：Server端憑證支援檢查可透過SSL LABS或者Nmap指令進行檢查
nmap –p 443 –script ssl-enum-ciphers <host>

```
<system.webServer>
  <rewrite>
    <rules>
      <rule name="HTTP to HTTPS redirect" stopProcessing="true">
        <match url="(.*)" />
        <conditions>
          <add input="{HTTPS}" pattern="off" ignoreCase="true" />
        </conditions>
        <action type="Redirect" redirectType="Found"
              url="https://{HTTP_HOST}/{R:1}" />
      </rule>
    </rules>
  </rewrite>
</system.webServer>
```



資訊系統委外開發RFP資安需求

■ 系統與資訊完整性

➢ 使用者輸入資料合法性檢查應置放於應用系統伺服器端

科技董座扮「白帽駭客」　竄改高鐵票價...賠15萬寫悔過書

正文　網友評論　友善列印

ETtoday新聞雲　說這專頁讚　117　讚

《蘋果日報》報導，許嫌和洪嫌從104年4月起至5月間開始扮演駭客，在其經營的數位科技公司等地，利用電腦網路連線登入台灣高鐵公司的網路訂票系統網頁訂購車票時，充分運用其網路通訊科技知識，於網路中修改台灣高鐵公司的網路購票資訊，**將原定價1仟餘元之高鐵票價，篡改成1百餘元之票價後，竟僅以1百餘元之金額即得以購買原價1千餘元之高鐵車票乘車使用**。

ibon售票系統遭爆有漏洞，專家：曝露網站設計不嚴謹的老問題

統一超商的ibon售票系統旅遊專區遭發現有漏洞，利用瀏覽器檢視程式碼就能竄改票價、結算金額，就能以1元買到8張票。資安專家認為這曝露出國內常見的不夠嚴謹問題，系統設計上忽視對回傳數值的驗證，才會導致這樣離譜的問題發生。

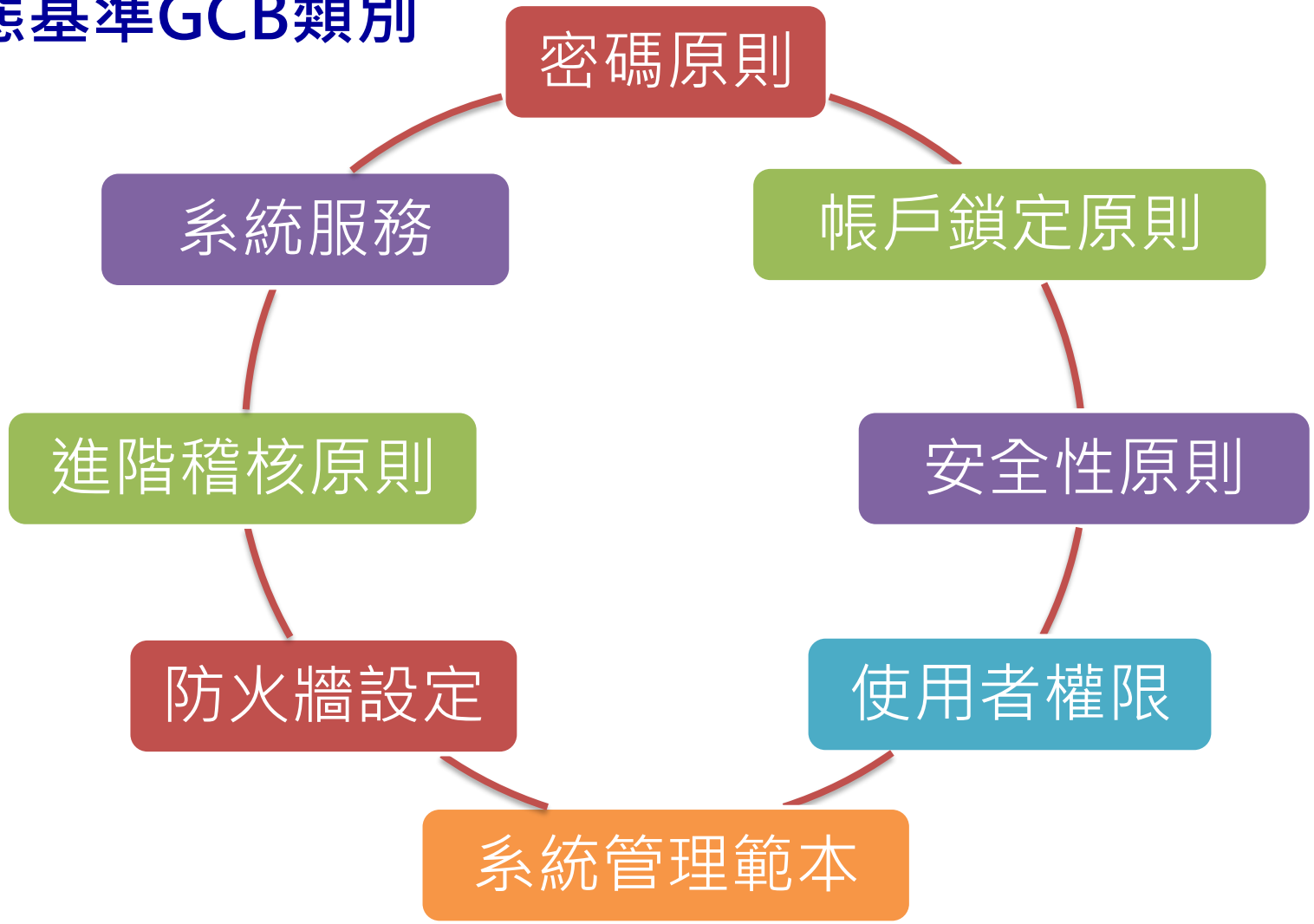
文/ 蘇文彬 | 2015-09-17 發表　讚 3.5萬 按讚加入iThome粉絲團　讚 分享 595　G+1 10

駭客怎麼有辦法破解台鐵的訂票系統呢？

匿名 1年前 · 9045 瀏覽

應用程式的安全性與系統管理

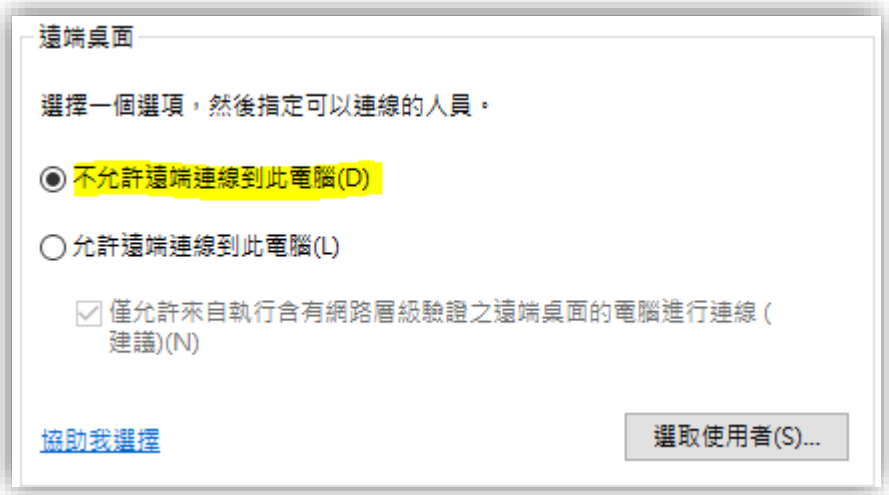
- 政府組態基準GCB類別

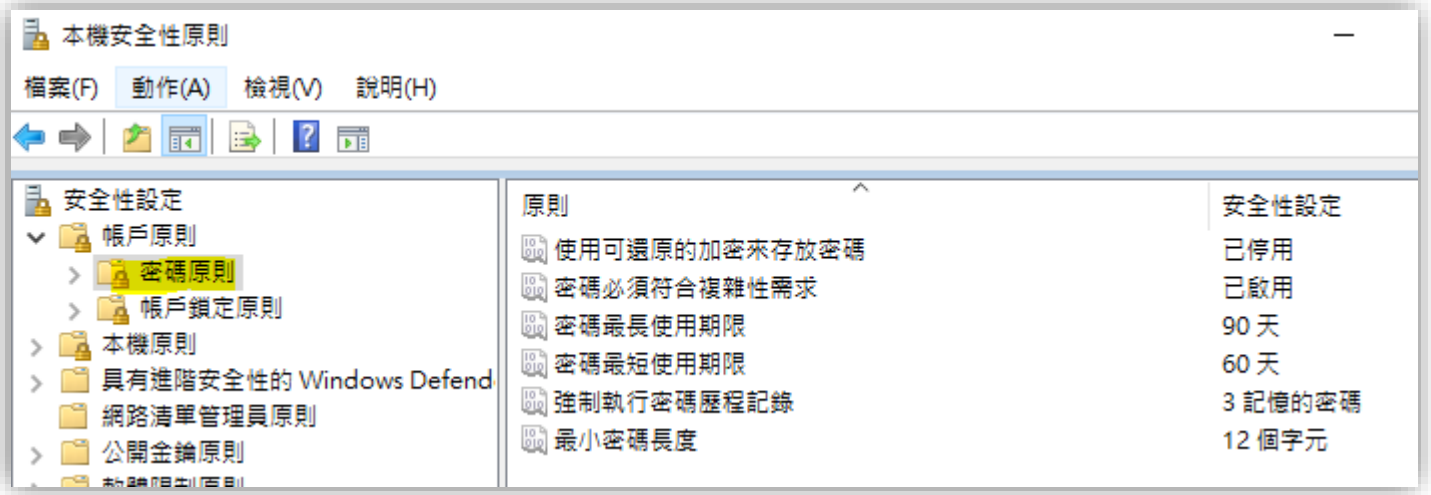


政府組態基準GCB

■ 密碼原則

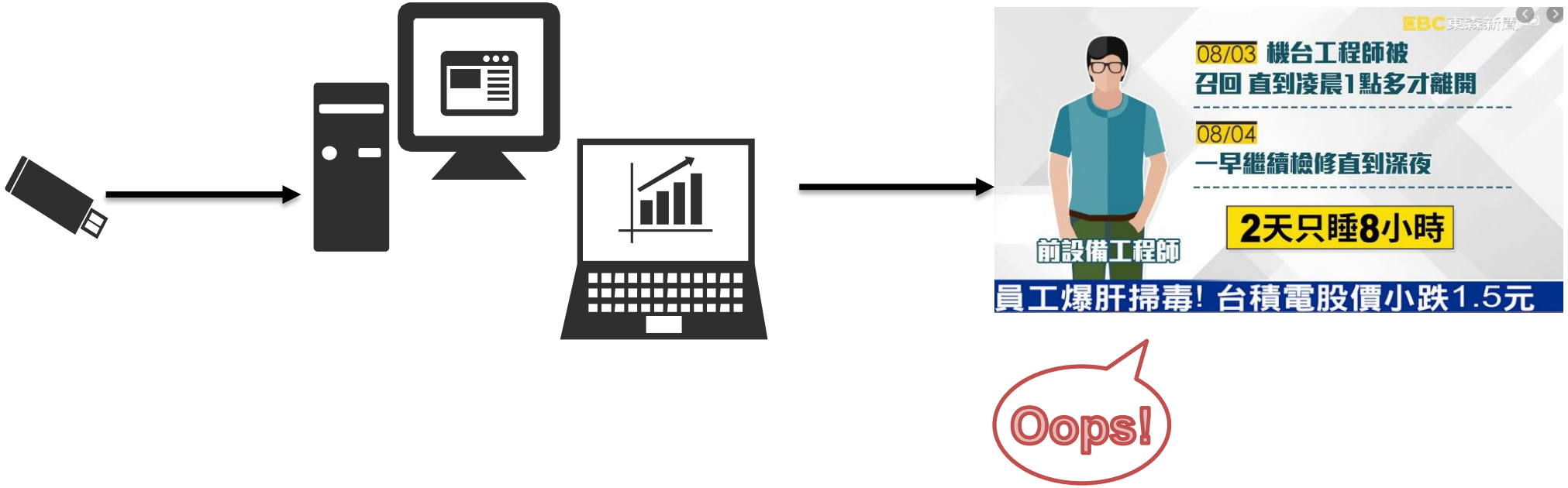
- 組態設定「帳戶：重新命名系統管理員帳戶」變更系統內建的Administrator帳戶名稱，並搭配組態設定「最小密碼長度」指定使用者帳戶的密碼最小長度需在12個字元以上，減少被駭客猜測到特殊權限的使用者名稱和密碼被猜到或暴力破解的機率降低。
- 若非必要，請直接關閉遠端登入權限。否則設定允入白名單。





政府組態基準GCB

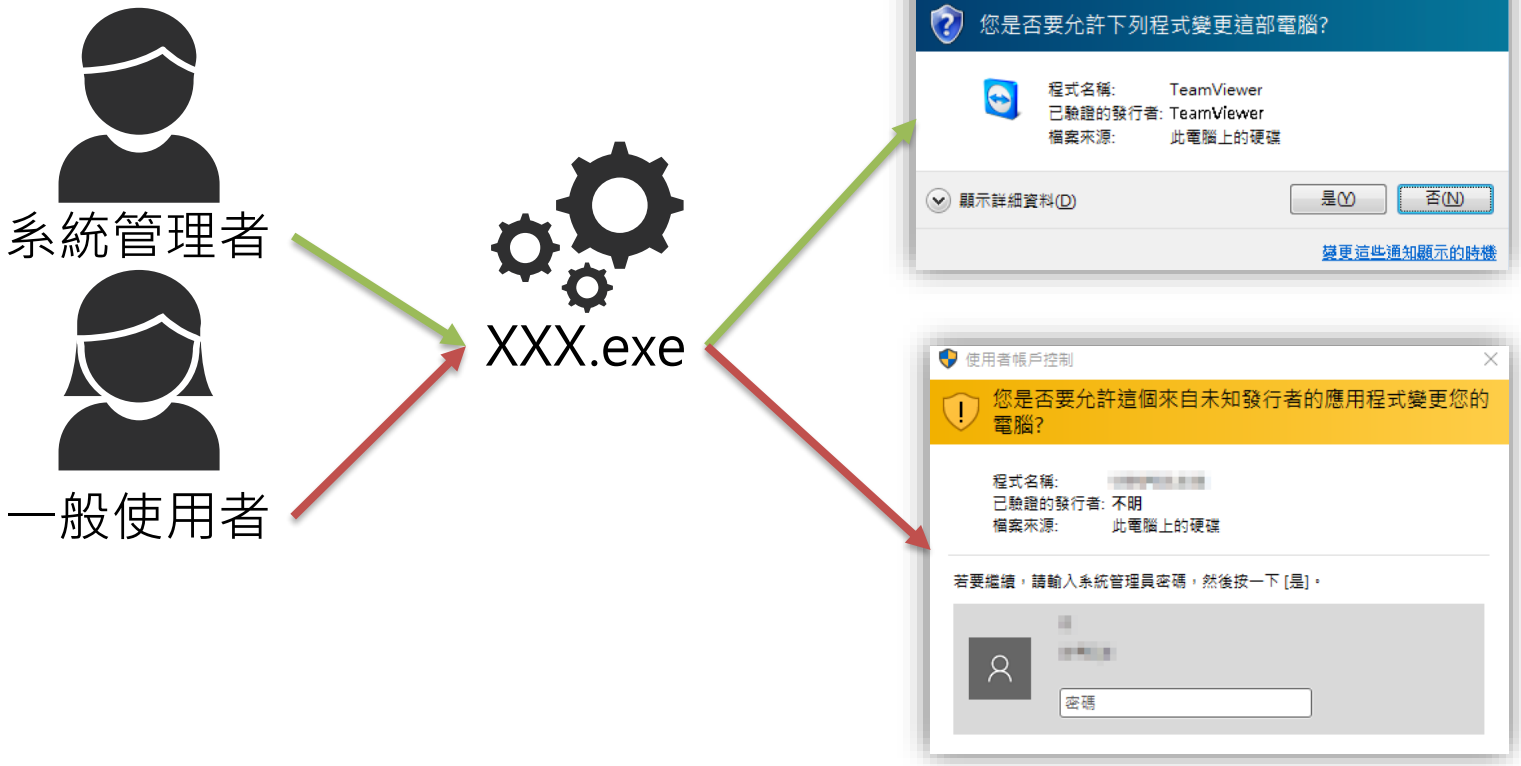
- **安全性原則**
  - 電腦設定\系統管理範本\Windows元件\自動播放原則\AutoRun的預設行為(建議關閉)。
  - 多數企業已將USB透過專屬維運的方式進行，或者直接進行限制。



政府組態基準GCB

■ 使用者權限

- 當偵測到應用程式安裝封裝需要提升權限時，會提示使用者輸入系統管理使用者名稱與密碼，輸入有效的認證，操作會以適用的權限繼續。



政府組態基準GCB

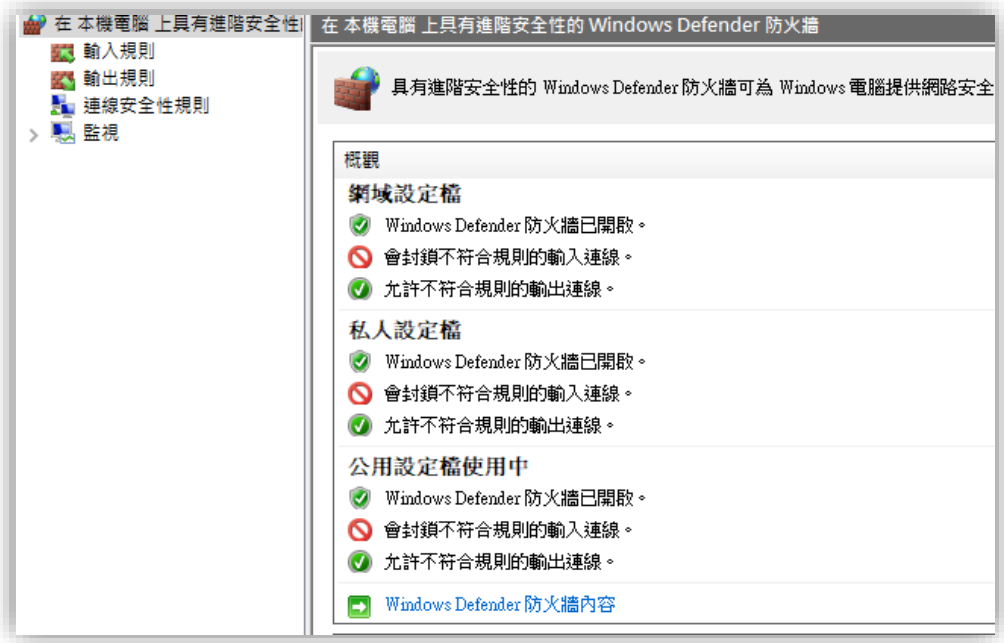
■ 防火牆設定

➢ 輸入設定

- ➢ 阻擋防火牆外部往防火牆內部之網路連線
- ➢ 建議預設關閉

➢ 輸出設定

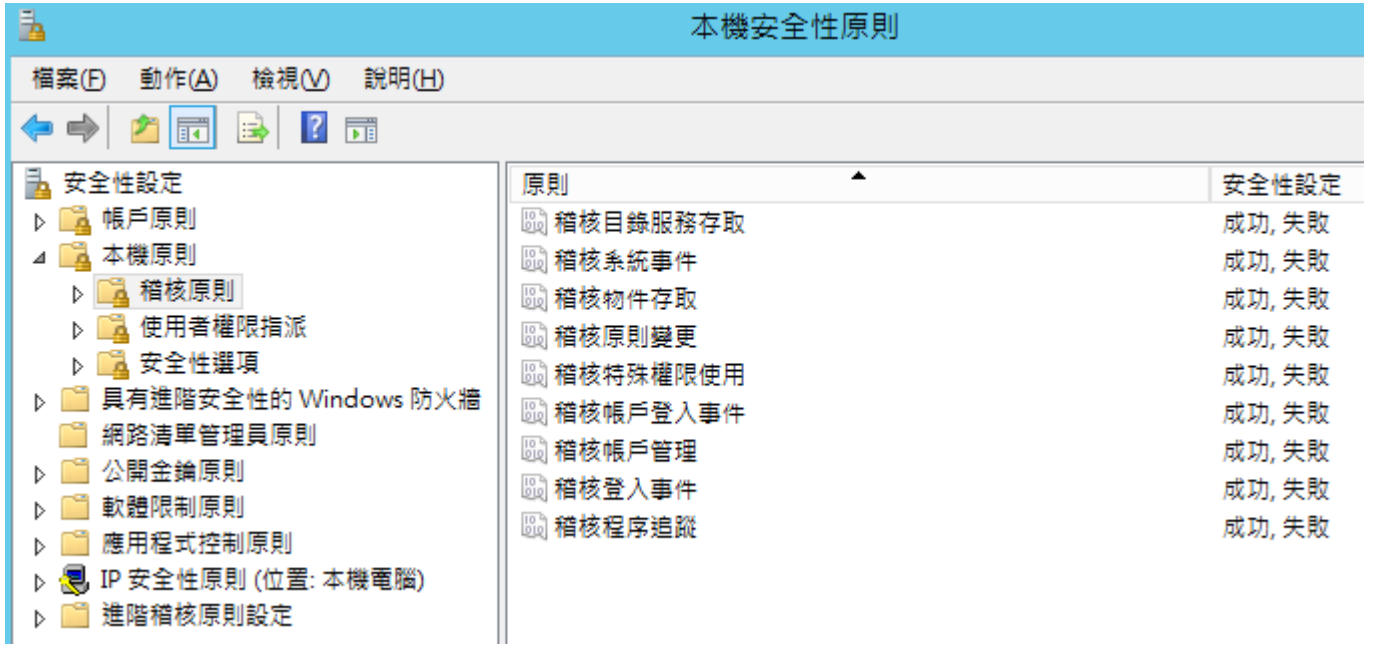
- ➢ 允許防火牆內部往防火牆外部之網路連線
- ➢ 預設值

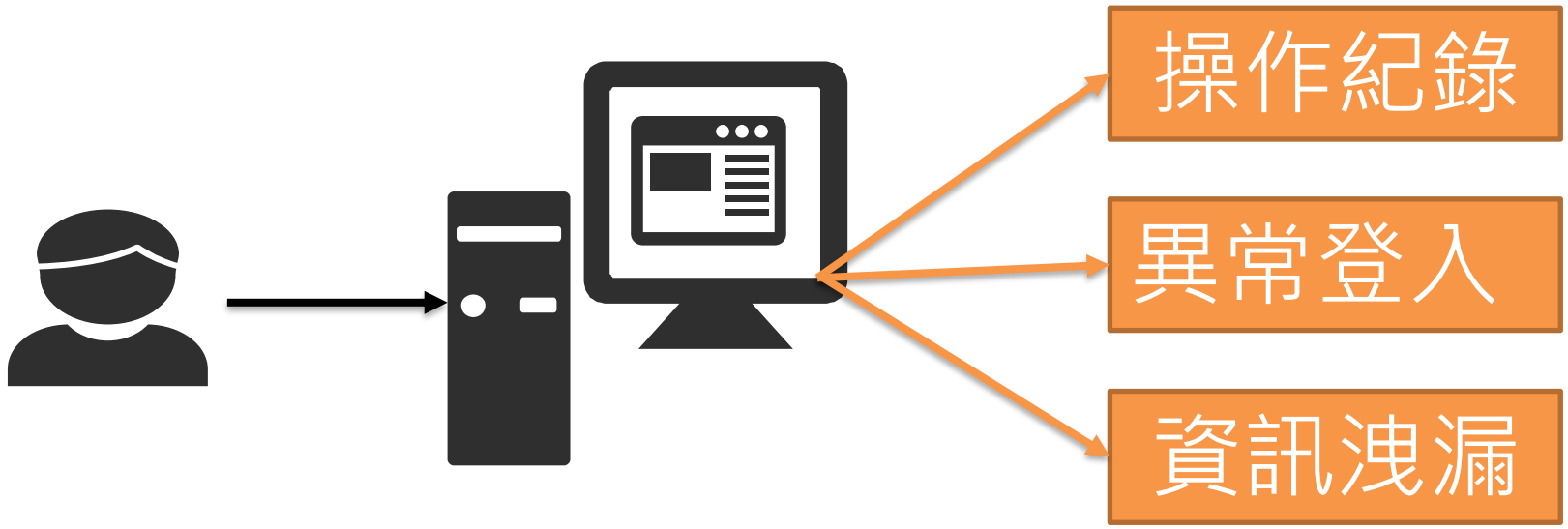


關閉非必要服務埠!

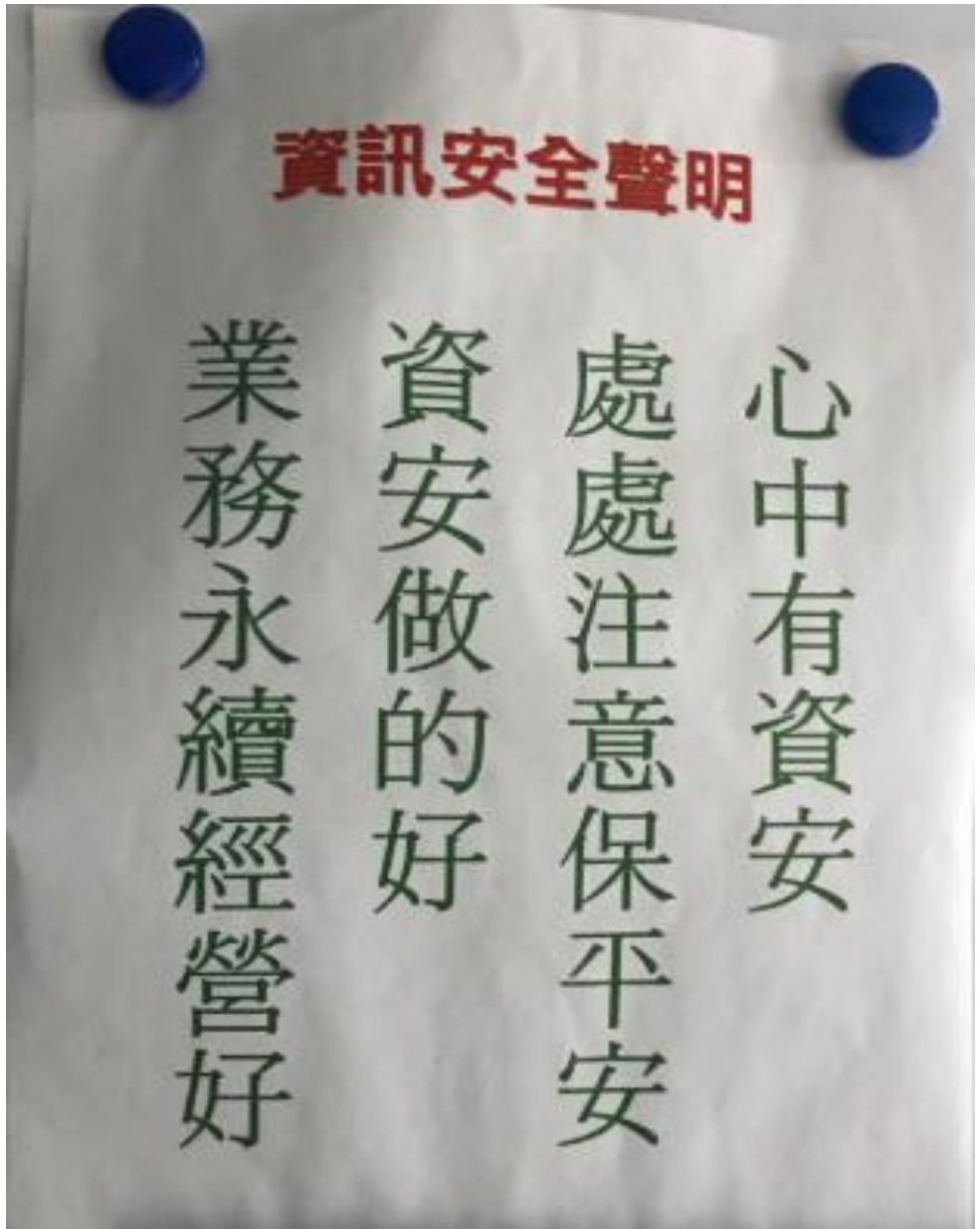
政府組態基準GCB

- 進階稽核原則設定
  - 檢查及審查可能影響系統安全性之活動，讓系統管理員記錄及檢視 指定之安全性相關活動事件的功能和服務





總結



~重要觀念宣導~

局內資訊系統在查詢時均會留下紀錄。
請勿進行非公務之查詢，或基於私人目的為不當查詢。
以免違反個資保護規範或觸犯相關法令。

Do right thing（做對的事情）/ Do thing right（把事情做好）

01 **對的時機做對的事**

安全的系統開發，絕不會只由上線前的安全檢測就可以達成，必須於需求、設計，甚至人員訓練階段就該持續要求。

02 **及早預防、持續維護**

弱點的修改成本隨著時間指數成長，若能提早避免掉問題，可以大幅降低系統修改成本。然而永遠會有新的問題被發現，也會有舊的漏洞被疏忽，因此需要持續維護。

03 **層層把關、面面防護**

一套資訊系統，就如同木桶一樣，必須面面俱到的防護，否則攻擊將會從最弱的一個面滲透