OWASP TOP 10:2017

行政院國家資通安全會報技術服務中心

大綱

- 近年網站安全弱點與安全系統發展生命週期
- OWASP 組織與相關專案簡介
- OWASP Top 10 2017 項目
  - 弱點說明
  - 攻擊手法
  - 防範方式

近年網站安全弱點趨勢

新聞

Nike旗下網站被爆有漏洞遲未修補，可能外洩密碼等敏感資訊

由於回報三個月仍未獲得回應，研究人員遂向媒體揭露Nike旗下的MyNikeTea...存在XML外部實體攻擊
漏洞，可讓駭客存取伺服器上的密碼等機密資訊，遠端執行程式碼或存取Nike...

文/ 林妍溱 | 2018-03-07 發表

XML外部實體(XXE)

中國挖礦駭客轉移攻擊目標至Jenkins伺服器

駭客利用Jenkins上的CVE-2017-1000353安全漏洞進行JenkinsMiner挖礦攻擊，這是一個反序列化漏洞，影響Altoros Jenkins for PCF 10.2以前的版本，這幾個月以來駭客已開始利用遠端存取木馬（RAT）及XMRig挖礦程式開採該漏洞，波及全球Jenkins用戶。

文/ 陳曉莉 | 2018-02-19 發表

不安全的反序列化

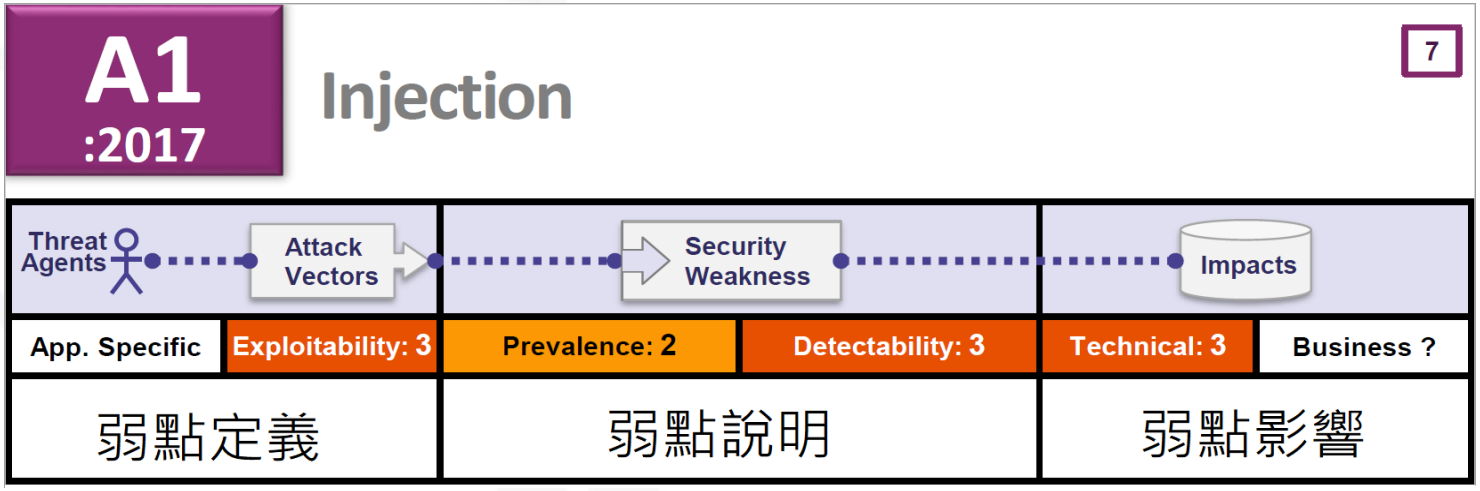
特斯拉失控衝撞店面，車主怪煞車失靈 特斯拉：Log檔紀錄根本沒踩煞車

janus 發表於 2014年10月25日 10:00 | 收藏此文

記錄與監控

網站漏洞趨勢-OWASP Top 10

- 發佈近年網頁應用程式威脅及常見的十大弱點
- 讓開發者了解網頁應用安全漏洞問題及防禦技術



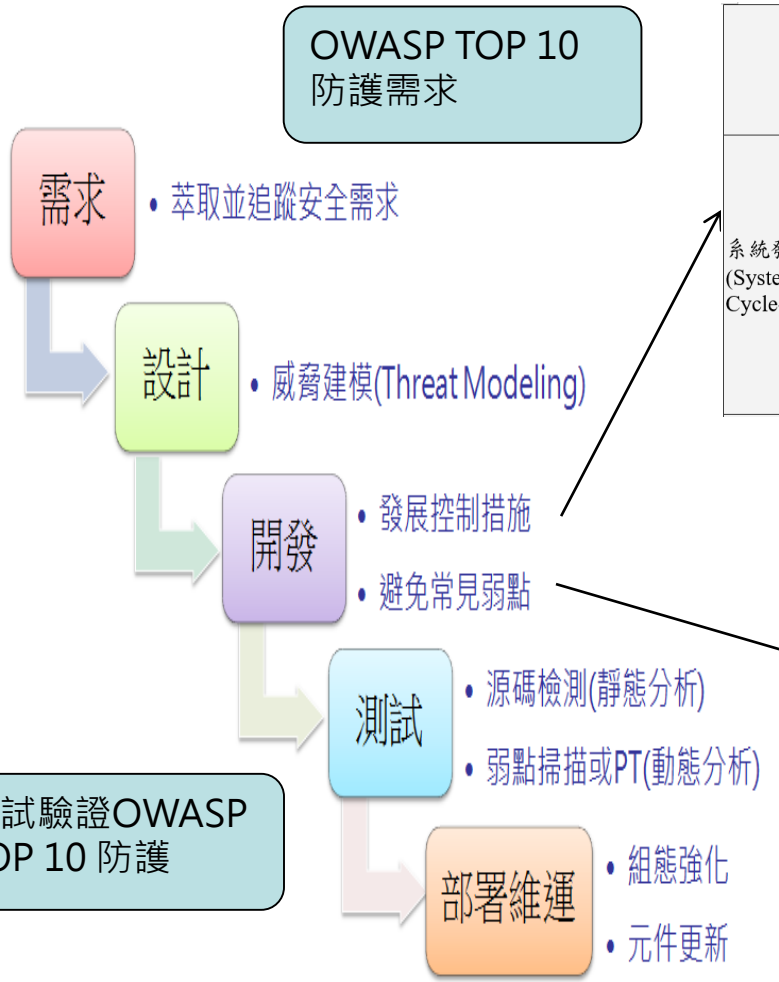
- 機關目前常見系統設計主要為網頁應用程式

OWASP Top 10 2017 項目

項次	中文名稱	英文名稱
A1	注入	Injection
A2	身分鑑別相關功能缺陷	Broken Authentication
A3	敏感資料暴露	Sensitive Data Exposure
A4	XML外部實體(XXE)	XML External Entity
A5	存取控制相關功能缺陷	Broken Access Control
A6	不當的安全組態設定	Security Misconfiguration
A7	跨站腳本攻擊 (XSS)	Cross-Site Scripting
A8	不安全的反序列化	Insecure Deserialization
A9	使用具有已知弱點的元件	Using Components with Known Vulnerabilities
A10	記錄與監控不足	Insufficient Logging & Monitoring

安全系統發展生命週期(SSDLC)

- 在系統開發的過程中，各階段加入安全上的考量



資訊系統分級與資安防護基準作業規定

控制措施	安全等級			參考文件
	普	中	高	
系統發展生命週期開發階段(System Development Life Cycle-Develop)	1.應針對安全需求實作必要控制措施。 2.應注意避免軟體常見漏洞(如 OWASP TOP 10)及實作必要控制措施。 3.應針對資訊系統之錯誤與例外進行適當處理。	執行等級「普」之所有控制措施。	1.執行等級「普」之所有控制措施。 2.執行「源碼掃描」安全檢測。 3.具備系統嚴重錯誤之通知機制(例如電子郵件或簡訊)。	安全控制措施參考指引附件 18 SA 安全軟體發展流程參考指引 3.3 安全軟體開發

項次	OWASP Top 10 項目
A1	注入
A2	身分鑑別相關功能缺陷
A3	敏感資料暴露
A4	XML外部實體(XXE)
A5	存取控制相關功能缺陷
A6	不當的安全組態設定
A7	跨站腳本攻擊 (XSS)
A8	不安全的反序列化
A9	使用具有已知弱點的元件
A10	記錄與監控不足

OWASP簡介

- OWASP(The Open Web Application Security Project)
  是一個非營利的全球性組織，專注在增進軟體安全
  - https://www.owasp.org/

- OWASP發起眾多與安全相關的專案
  - OWASP Top 10(十大網站安全風險)
  - OWASP Mobile Top 10(十大行動裝置安全風險)
  - OWASP Internet of Things Project (十大物聯網安全風險)
  - Application Security Verification Standard(系統安全驗證)
  - Software Assurance Maturity Model(軟體安全成熟度)
  - OWASP Enterprise Security API (ESAPI)
    - 免費、開放原始碼的網頁安全控制函式庫，輔助開發人員產出低風險的應用程式

OWASP Top 10 2017新版差異

- OWASP版本演進
  - OWASP Top 10 2010
  - OWASP Top 10 2013
  - OWASP Top 10 2017

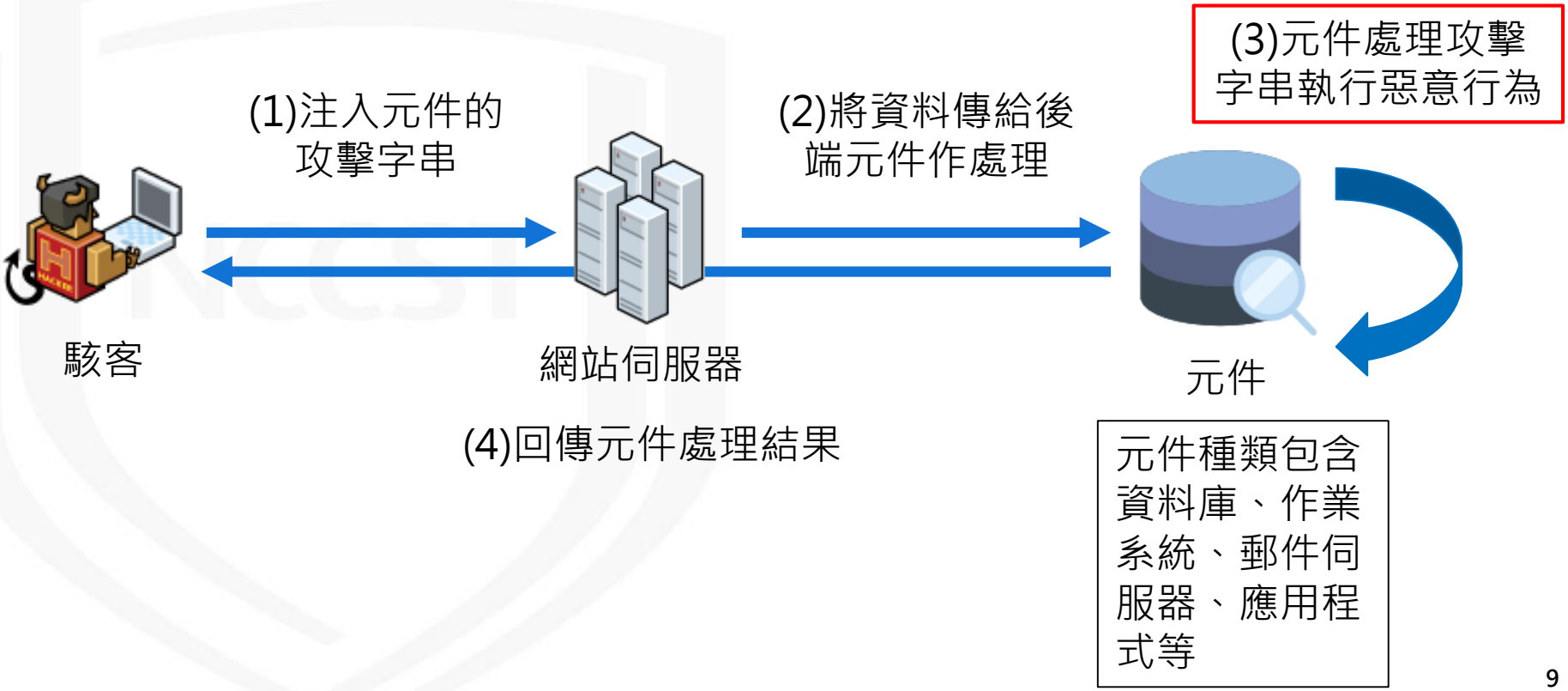
OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

合併
移除
新增

A1.注入(Injection)

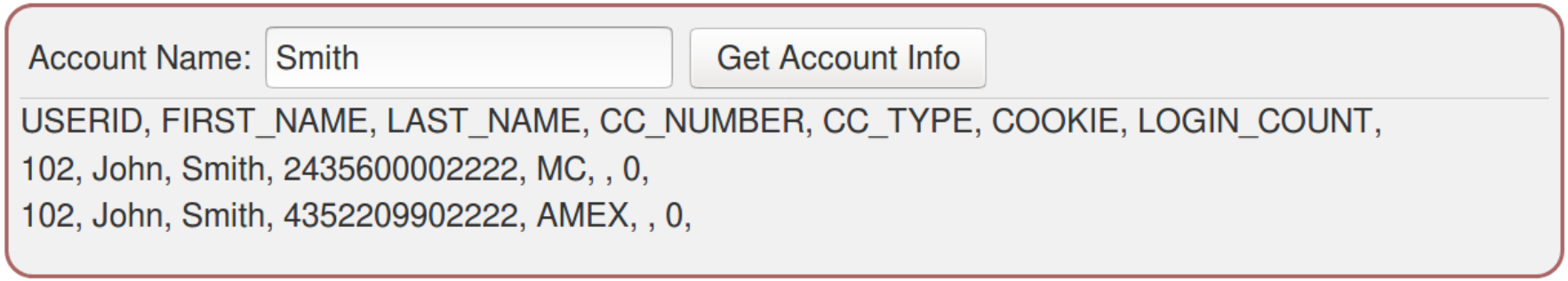
A1.注入-弱點說明

- 駭客將攻擊字串注入正常的語法中，使得網站伺服器出現非預期的結果，並執行駭客輸入的指令



A1.注入-攻擊手法

- 輸入帳號搜尋資訊



- 透過SQL Injection 語法做搜尋

A1.注入-防範方式(1/2)

- 不要使用串接的查詢語法，改為使用參數化的查詢方式 (例如 : Prepared statements)

```
String sql= "select * from users where LAST_NAME = '" + userName + "'";
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery(sql);
```

```
String sql= "select * from users where LAST_NAME =?1 ";
PreparedStatement pstmt = con.prepareStatement(sql);
pstmt.setString(1,userName);
ResultSet rs = pstmt.executeQuery(sql);
```

A1.注入-防範方式(2/2)

- 以參數化的查詢方式使用Stored procedures (資料庫)

```
try
{
    int age = 39;
    String poetName = "dylan thomas";
    CallableStatement proc =
        connection.prepareCall("{ call set_death_age(?, ?) }");
    proc.setString(1, poetName);
    proc.setInt(2, age);
    cs.execute();
```

- 組成查詢字串時，跳脫特殊字元

```
Codec ORACLE_CODEC = new OracleCodec();
 String query = "SELECT user_id FROM user_data WHERE user_name = '" +
   ESAPI.encoder().encodeForSQL( ORACLE_CODEC, req.getParameter("userID")) + "' and user_password = '"
   + ESAPI.encoder().encodeForSQL( ORACLE_CODEC, req.getParameter("pwd")) +"'";
```

A1.注入-需求驗證項目

資安防護基準作業規定		需求項目	需求範例
系統與服務獲得 (System and Services Acquisition)	系統發展生命週期開發階段 (System Development Life Cycle-Develop)	應注意避免軟體常見漏洞 (如OWASP TOP 10) 及實作必要控制措施	以源碼檢測確認排除常見漏洞
		應針對資訊系統之錯誤與例外進行適當處理	設置客製化錯誤頁面
系統與資訊完整性 (System and Information Integrity)	軟體及資訊完整性 (Software, Firmware, and Information Integrity)	使用者輸入資料合法性檢查應置放於應用系統伺服器	以正規表示式規範使用者輸入欄位，並實作於伺服器端

安全等級

- 普級
- 中級
- 高級

A2.身分鑑別相關功能缺陷 (Broken Authentication)

A2.身分鑑別相關功能缺陷-弱點說明

- 網站應用程式中使用的身分鑑別相關功能(如帳號密碼等)有缺陷，讓駭客可以繞過驗證機制存取系統或是冒用身分

- 帳號密碼驗證弱點
  - 密碼被猜測
  - 未正確實作密碼重設機制
  - 密碼明文儲存

A2.身分鑑別相關功能缺陷-攻擊手法(1/2)

- 猜測密碼方式
  - –暴力攻擊 
    - ➢利用工具將所有字母、數字及字元的可能組合去破解
  - –字典攻擊 
    - ➢載入字典檔到破解程式
  - –混合攻擊法   
    - ➢利用單字列表加上數字及字元組合去進行破解
  - –外洩密碼 
    - ➢利用其他地方外洩的密碼進行猜測

A2.身分鑑別相關功能缺陷-攻擊手法(2/2)

- 密碼重設機制問題
  - 網站主動將密碼重新設定後，以電子郵件寄送給使用者
    - 產生格式過於簡單容易被破解
    - 產生格式太複雜則使用者容易再次忘記
    - 密碼明文傳輸儲存
  - 密碼重設時未再次確認使用者身分
  - 認證問題太過簡單

- 密碼儲存問題
  - 使用明文方式將資料儲存在伺服器
  - 駭客透過其他管道取得資料即可獲得密碼

A2.身分鑑別相關功能缺陷-防範方式(1/8)

- 猜測密碼
  - 增加駭客猜測密碼的時間跟難度
    - 帳戶鎖定(Account lockout)機制
    - 防自動化程式機制(Captcha)
  - 使用多重因素認證
- 密碼重設機制問題
  - 正確實作密碼重設機制
- 密碼明文儲存問題
  - 改為儲存密碼雜湊值

A2.身分鑑別相關功能缺陷-防範方式(2/8)

- 帳戶鎖定(Account lockout)機制
  - 超過可錯誤次數後鎖定時間
  - 鎖定對象：帳號、來源IP

- 防自動化程式機制(Captcha)

A2.身分鑑別相關功能缺陷-防範方式(3/8)

- 密碼複雜度 (建議)
  - 密碼長度大於12個字元
  - 包含英文大小寫、數字，以及特殊字元(四取三)

dfjiWEW4782@@ ✓

123456 ✗

PASSWORD ✗

- 強制密碼定期更換
  - 建議定期(如90天)更換密碼，且新密碼不可以與前三次使用密碼相同

A2.身分鑑別相關功能缺陷-防範方式(4/8)

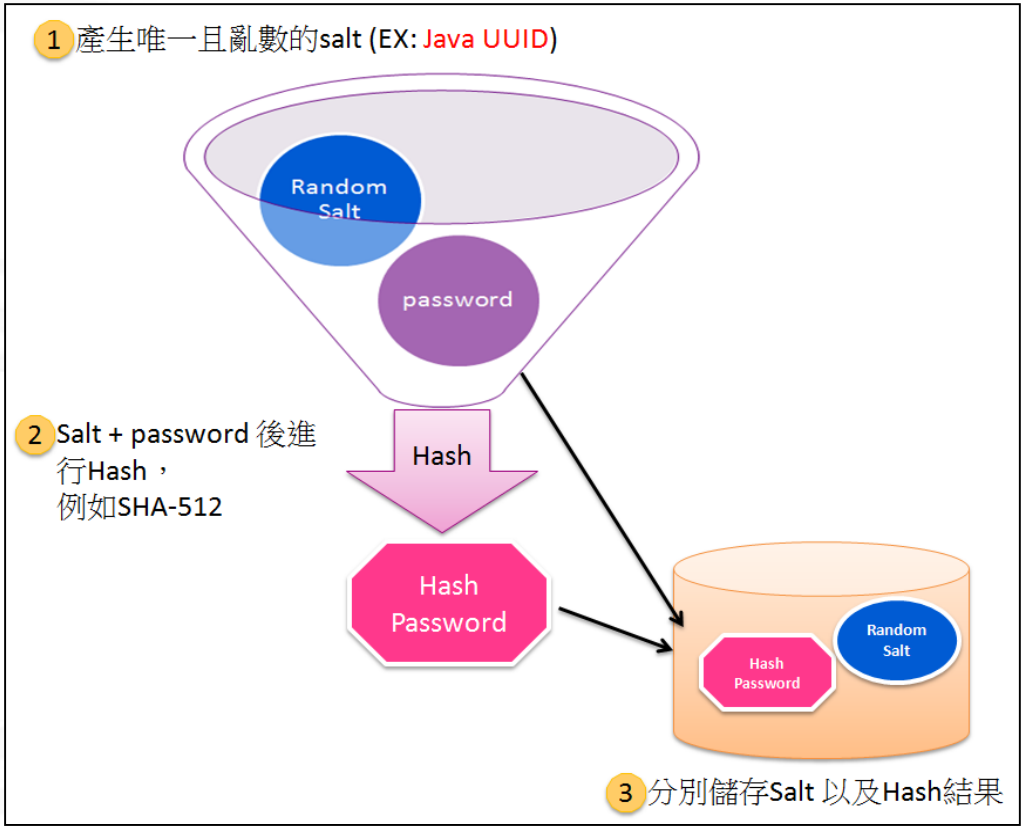
- 多重因素認證(Multi-factor)
  - 你知道的(Know)
    - PIN
    - 密碼
  - 你擁有的(Have)
    - 金融卡
    - 自然憑證
  - 你本身有的(Own)
    - 指紋
    - 虹膜
  - 你的位置(Where)
    - IP位址

A2.身分鑑別相關功能缺陷-防範方式(5/8)

- 正確實作密碼重設機制
  - 採用其他管道(Email、SMS)，重新身分驗證後，允許重設密碼
  - 常見作法為先要求使用者輸入郵件信箱並驗證，後續透過Email寄送一次性的令牌(Token)，檢查其時效性後，才允許使用者重設密碼

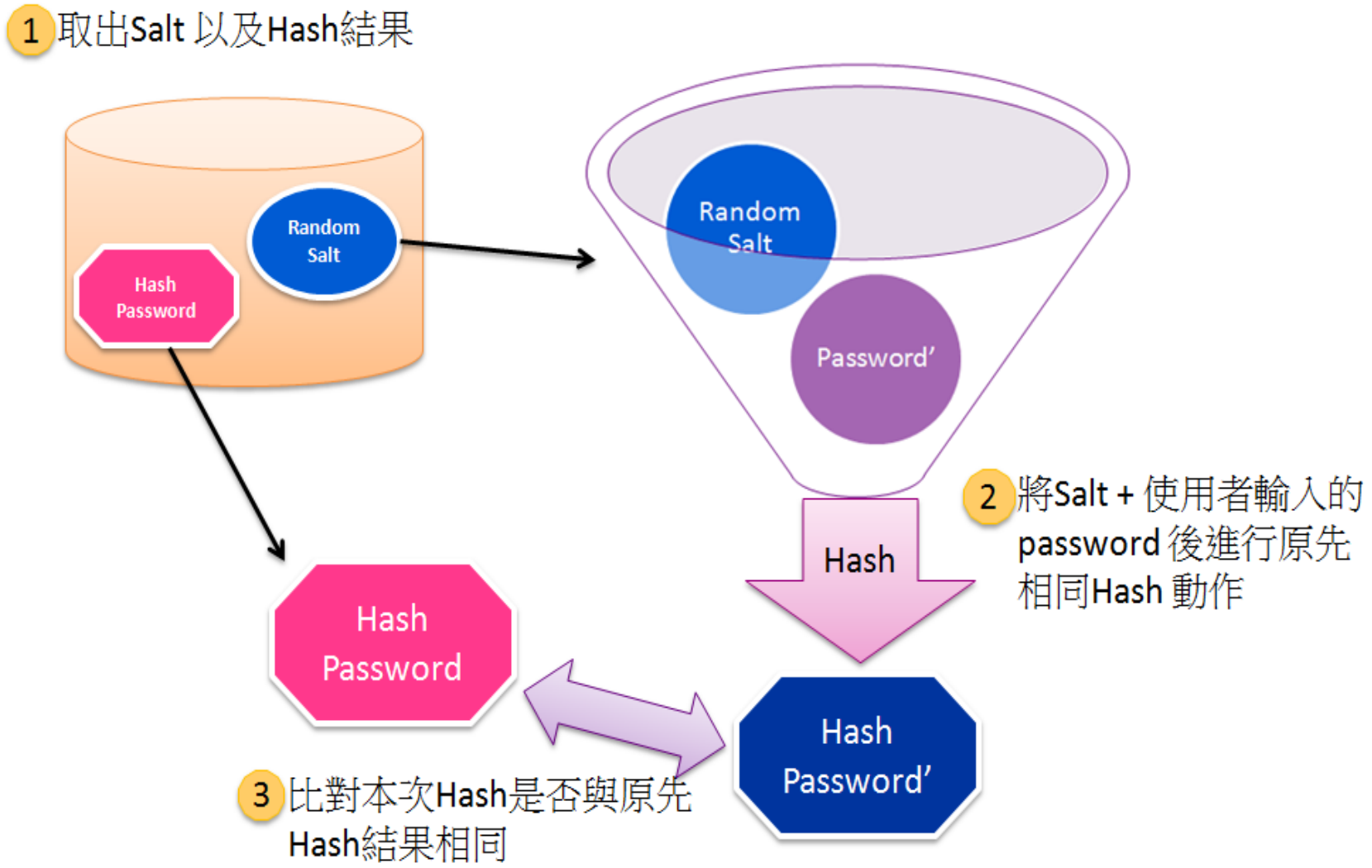
A2.身分鑑別相關功能缺陷-防範方式(6/8)

- 儲存密碼的雜湊數值
  - 不要儲存密碼明碼
  - 正確使用雜湊儲存(Salted)
- 初次設定



A2.身分鑑別相關功能缺陷-防範方式(7/8)

- 比對密碼



A2.身分鑑別相關功能缺陷-防範方式(8/8)

Wrong : 儲存明文

User	Password
Stephen	auhsoJ
Sarah	auhsoJ
Lisa	hsifdrowS
James	1010NO1Z
Harry	sinocarD tupaC

明文

明文

Not good : 未使用salt

User	Password Hash
Stephen	39e717cd3f5c4be78d97090c69f4e655
Sarah	39e717cd3f5c4be78d97090c69f4e655
Lisa	f567c40623df407ba980bfad6dff5982
James	711f1f88006a48859616c3a5cbcc0377
Harry	fb74376102a049b9a7c5529784763c53

結果相同

Good : Salted hashing

User	Random Salt	Password Hash
Stephen	06917d7ed65c466fa180a6fb62313ab9	b65578786e544b6da70c3a9856cdb750
Sarah	711f51082ea84d949f6e3efecf29f270	e3afb27d59a34782b6b4baa0c37e2958
Lisa	51f2e43105164729bb46e7f20091adf8	2964e639aa7d457c8ec0358756cbffd9
James	fea659115b7541479c1f956a59f7ad2f	dd9e4cd20f134dda87f6ac771c48616f
Harry	30ebf72072134f1bb40faa8949db6e85	204767673a8d4fa9a7542ebc3eceb3a2

結果相異

A2.身分鑑別相關功能缺陷-需求驗證項目(1/3)

資安防護基準作業規定		需求項目	需求範例
存取控制 (Access Control)	帳號管理 (Account Management)	資訊系統具備帳號管理機制，包含帳號之申請、開通、停用及刪除之程序	帳號申請、審核、停用、刪除功能
		資訊系統具備刪除或禁用已逾期之臨時或緊急帳號之功能	臨時帳號機制， 逾期需有自動停用功能
		資訊系統具備禁用閒置帳號之功能	帳號未登入達半年以上， 系統自動停用
		當超過機關所規定之預期閒置時間或可使用期限時， 資訊系統應自動將使用者登出	逾時自動登出、手動登出確實失效

A2.身分鑑別相關功能缺陷-需求驗證項目(2/3)

資安防護基準作業規定		需求項目	需求範例
識別與鑑別 (Identification and Authentication)	內部使用者之識別與鑑別 (Identification and Authentication for Organizational Users)	資訊系統應識別及鑑別非機關使用者 (或代表機關使用者行為的程序)	系統使用者具有唯一帳號，且須通過身分驗證機制
	身分驗證管理 (Authenticator Management)	使用預設密碼登入系統時，應於登入後要求立即變更	系統若有預設密碼，登入後要求立即變更
		基於密碼之鑑別資訊，系統應強制最低密碼複雜度；強制密碼最短及最長之效期限制	設定/修改使用者密碼功能，限制使用者密碼強度 (密碼長度12個字元以上、包含英文大小寫、數字
		使用者更換密碼時，至少不可以與 前3次使用過之密碼相同	登入時檢查密碼效期，必須定期90天更換密碼，且至少不可以與前3次使用過之密碼相同
		具備帳戶鎖定機制，帳號登入進行身分驗證失敗達5次後，至少15分鐘內不允許該帳號繼續嘗試登入或使用機關自建之失敗驗證機制	帳戶鎖定機制，帳號登入進行身分驗證失敗達5次後，至少15分鐘內不允許該帳號繼續嘗試登入

A2.身分鑑別相關功能缺陷-需求驗證項目(3/3)

資安防護基準作業規定		需求項目	需求範例
識別與鑑別 (Identification and Authentication)	身分驗證管理 (Authenticator Management)	身分驗證機制應防範自動化程式之登入或密碼更換嘗試，如採用圖形驗證碼(CAPTCHA)機制	採用圖形驗證碼(CAPTCHA)機制
		密碼重設機制對使用者重新身分驗證後，發送一次性及具有時效性令牌(Token)，例如:簡訊驗證碼、EMAIL驗證連結，檢查傳回令牌有效性後，才允許使用者進行重設密碼動作	密碼重設機制
		對帳號之網路或本機存取採取多重認證技術	多重因素認證技術 (憑證、IP、OTP等)
識別與鑑別 (Identification and Authentication)	鑑別資訊回饋 (Authenticator Feedback)	資訊系統應遮蔽在鑑別過程中之資訊(如密碼)，以防止未授權之使用者可能之窺探/使用	密碼輸入欄位不以明文顯示
	加密模組鑑別 (Cryptographic Module Authentication)	資訊系統若以密碼進行鑑別時，該密碼應加密或經雜湊處理後儲存	使用者密碼添加亂數(Salt)進行雜湊函式(HASH Function)處理後，分別儲存亂數及雜湊後密碼

A3.敏感資料暴露
(Sensitive Data Exposure)

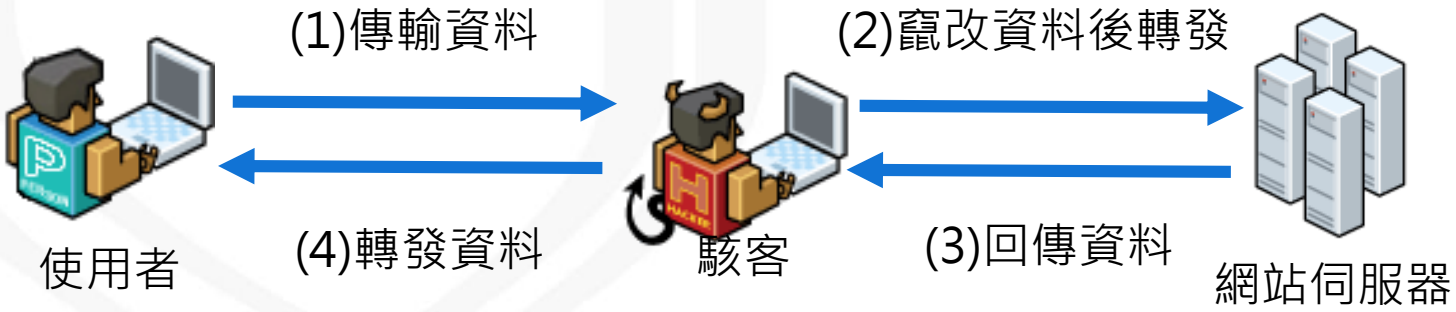
A3.敏感資料暴露-弱點說明

- 資料傳輸或儲存時未使用妥善的保密機制造成資料暴露
- 駭客可以透過監聽或是攔截方式取得傳輸資料
- 駭客透過攻擊手法取得資料時，可直接讀取未加密之敏感資料

A3.敏感資料暴露-攻擊手法

- 監聽傳輸資料

- 攔截、竄改傳輸資料
  - 中間人攻擊 (man-in-the-middle attacks)

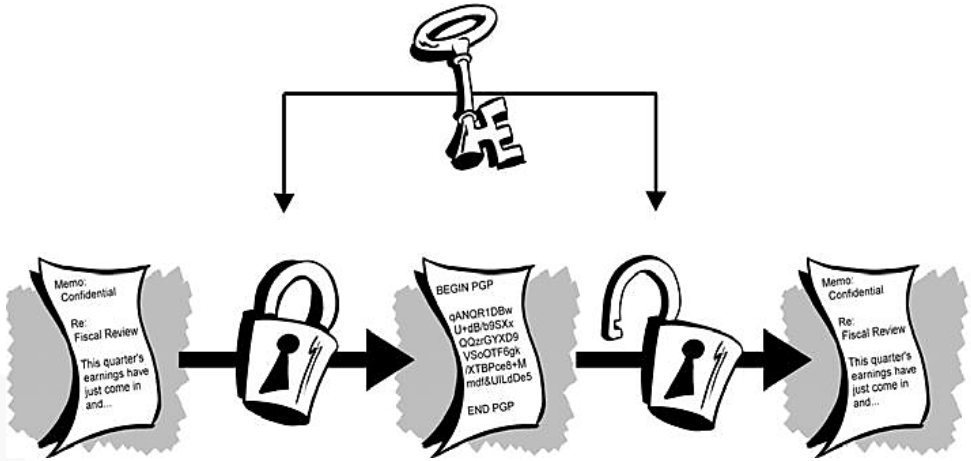


A3.敏感資料暴露-防範方式(1/6)

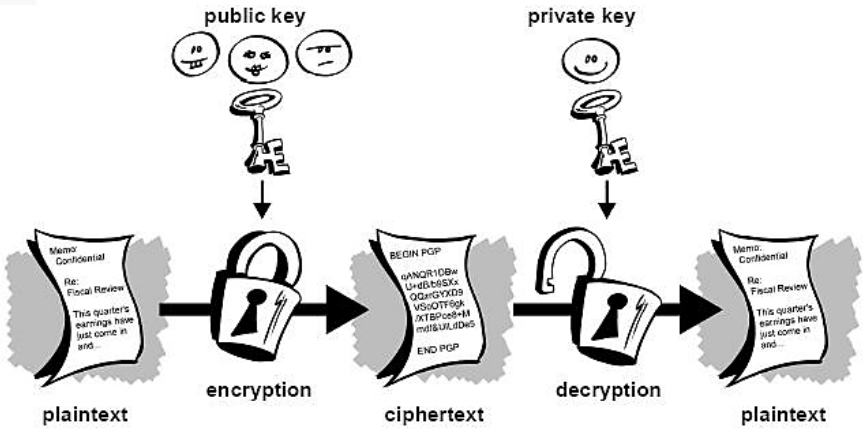
- 常見控制措施
  - 傳輸加密
    - 採用通道加密與傳輸層加密
  - 資料加密
    - 資料加密後儲存
    - 採用儲存體的加密機制

A3.敏感資料暴露-防範方式(2/6)

- 對稱式加密
  - 單一金鑰加解密
  - 演算法例如：
    - AES



- 非對稱式加密
  - 需要一對金鑰，一個是私人金鑰，另一個則是公開金鑰
  - 演算法例如：
    - RSA
    - ElGamal
    - ECC

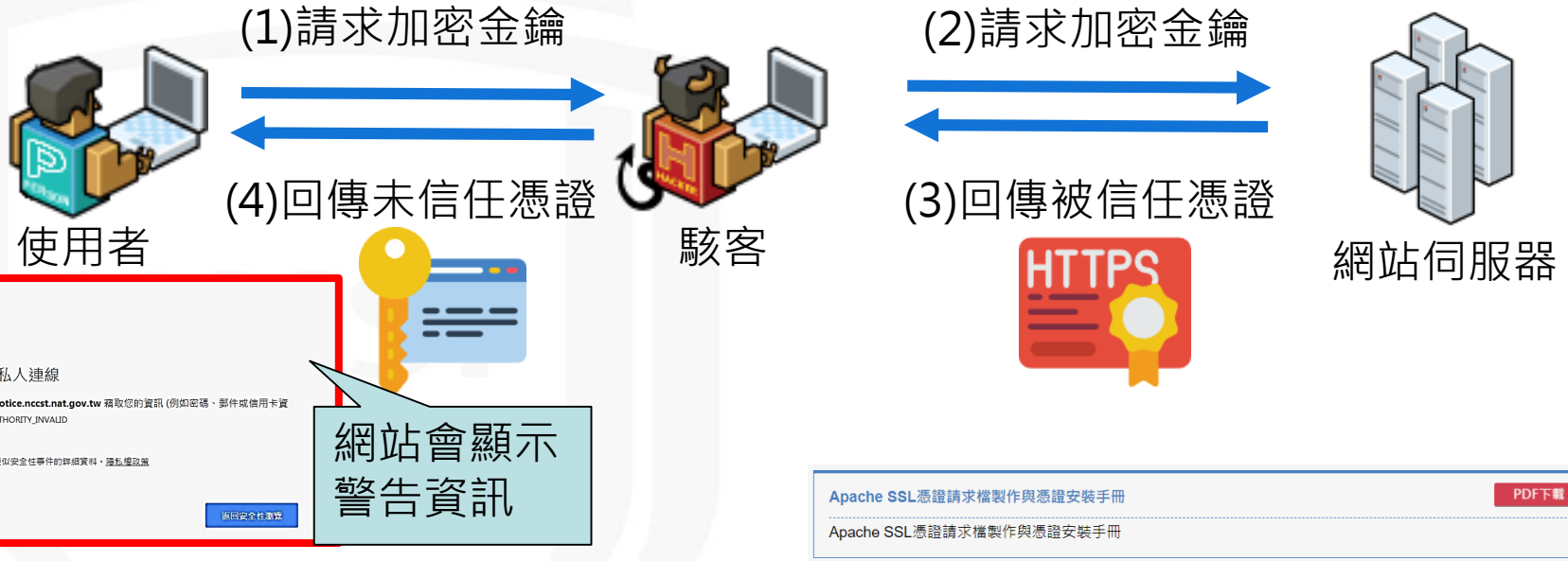


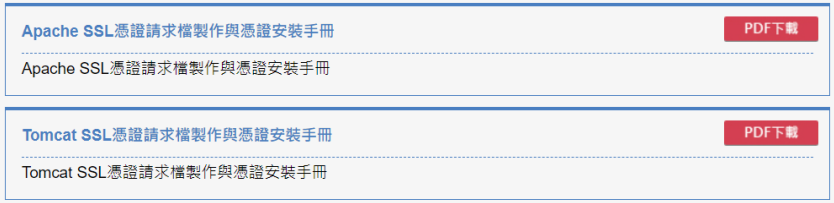
A3.敏感資料暴露-防範方式(3/6)

- 傳輸加密
  - 讓駭客無法理解監聽訊息
  - 仍然無法抵擋中間人攻擊，須採用合法CA核發之憑證以減少可能之風險

A3.敏感資料暴露-防範方式(4/6)

- 採用合法CA(Certificate Authority)核發之憑證
  - 若駭客無法取得合法CA核發之憑證，使用者頁面就會顯示不信任的警告資訊





- 政府憑證管理中心 GCA (https://gca.nat.gov.tw/web2/form02.html)

A3.敏感資料暴露-防範方式(5/6)

- 使用 HTTPS 代替 HTTP
- HTTPS使用強度足夠的協定及演算法
  - SSL3.0及TLS1.0之前的版本可被攻擊
  - 建議使用 TLS 1.1 以上的版本

```
<Connector SSLEnabled="true" URIEncoding="UTF-8" acceptCount="300"
           keystoreFile="XXX.jks" keystorePass="XXXXX" maxSpareThreads="75" maxThreads="500"
           minSpareThreads="25" port="443" compression="on" compressionMinSize="2048"
           noCompressionUserAgents="gozilla,traviata"
           compressableMimeType="text/html,text/xml,text/javascript,application/x-javascript,application/javascript"

           ciphers=
           "TLS_RSA_WITH_AES_256_GCM_SHA384,
           TLS_RSA_WITH_AES_128_GCM_SHA256,
           TLS_RSA_WITH_AES_256_CBC_SHA256,
           TLS_RSA_WITH_AES_256_CBC_SHA,
           TLS_RSA_WITH_AES_128_CBC_SHA256,
           TLS_RSA_WITH_AES_128_CBC_SHA,
           TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
           TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
           TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
           TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA"

           protocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https"
           secure="true" sslProtocol="TLSv1.2"
           sslEnabledProtocols="TLSv1.1,TLSv1.2"
/>
```

指定演算法

指定 Protocol

A3.敏感資料暴露-防範方式(6/6)

- 資料加密
  - 使用對稱式加密
  - 使用公開、具國際機構驗證且未遭破解的演算法
    不使用自創之加密方式
  - 使用最大限度之金鑰長度

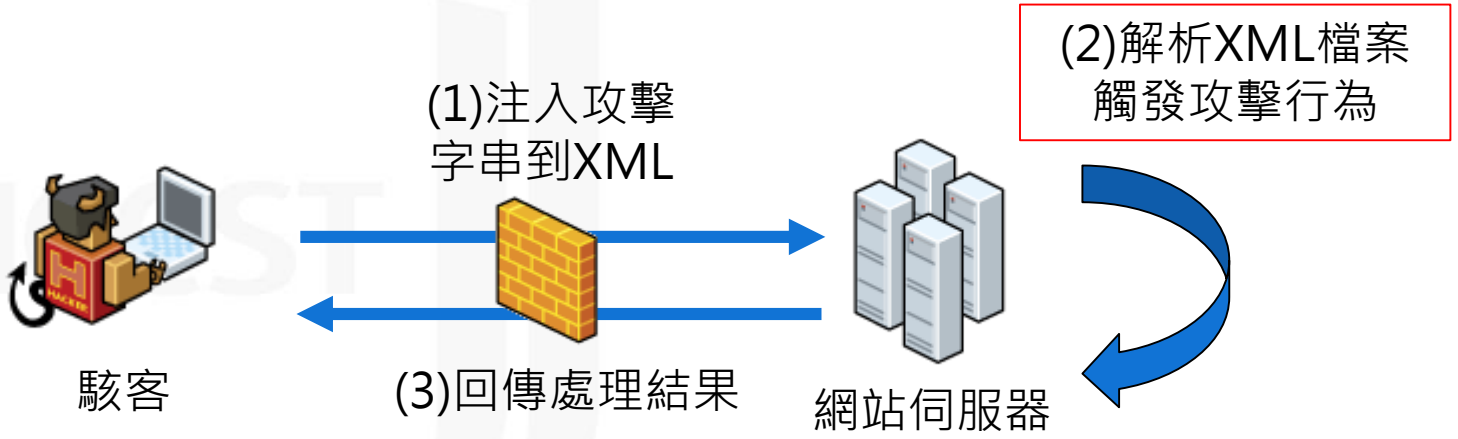
A3.敏感資料暴露-需求驗證項目

資安防護基準作業規定		需求項目	需求範例
系統與通訊保護 (System and Communications Protection)	內部使用者之識別與鑑別 (Identification and Authentication for Organizational Users)	身分驗證相關資訊不以明文傳輸(如HTTPS)	使用HTTPS加密協定
		資訊系統應採用加密機制來保護遠端存取連線的機密性	
		傳輸過程中除非有其他替代之實體保護措施，否則資訊系統應採用加密機制(如HTTPS) 以防止未授權之資訊揭露或偵測資訊之變更	
		避免使用已遭破解之加密演算法	HTTPS加密協定設定 高強度演算法
	資料儲存之安全 (Protection of Information at Rest)	靜置資訊指資訊位於資訊系統特定元件，如儲存設備上之狀態。與系統相關需要保護的資訊，例如：資訊系統組態設定。該等資訊應予以保護	禁止目錄尋訪， 正確實作檔案下載功能
		機密資訊應加密儲存	系統機密資料，加密儲存資料庫

A4.XML外部實體
(XML External Entity, XXE)

A4. XML外部實體-弱點說明

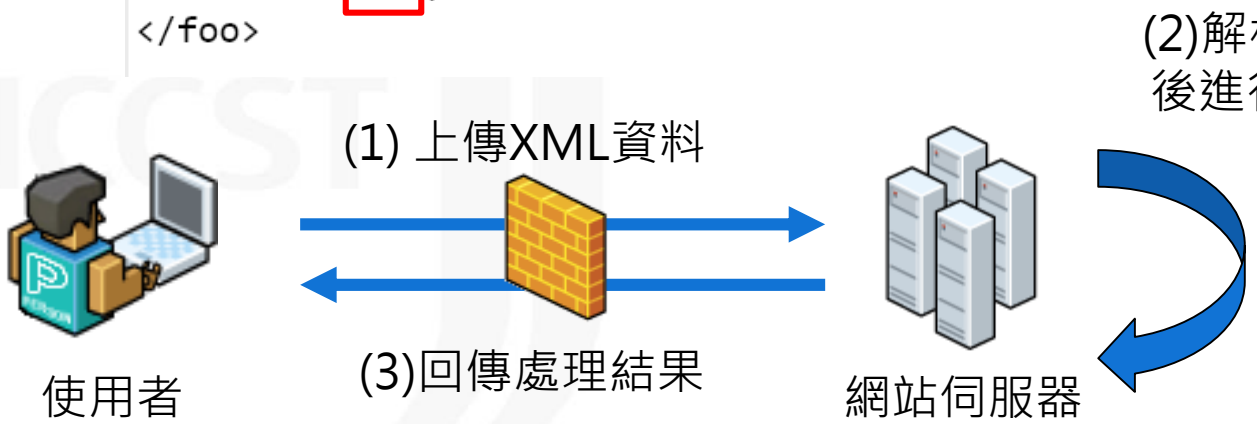
- 駭客將攻擊字串注入到XML檔案中，並且上傳到網站伺服器，使得網站伺服器解析XML時觸發了攻擊行為



A4. XML外部實體-攻擊手法(1/2)

- 正常行為範例
  - 網頁伺服器透過解析上傳XML內容得到資訊再進行後續處理

```
<?xml version="1.0" encoding="UTF-8"?>
<foo>
    <user>ian</user>
</foo>
```

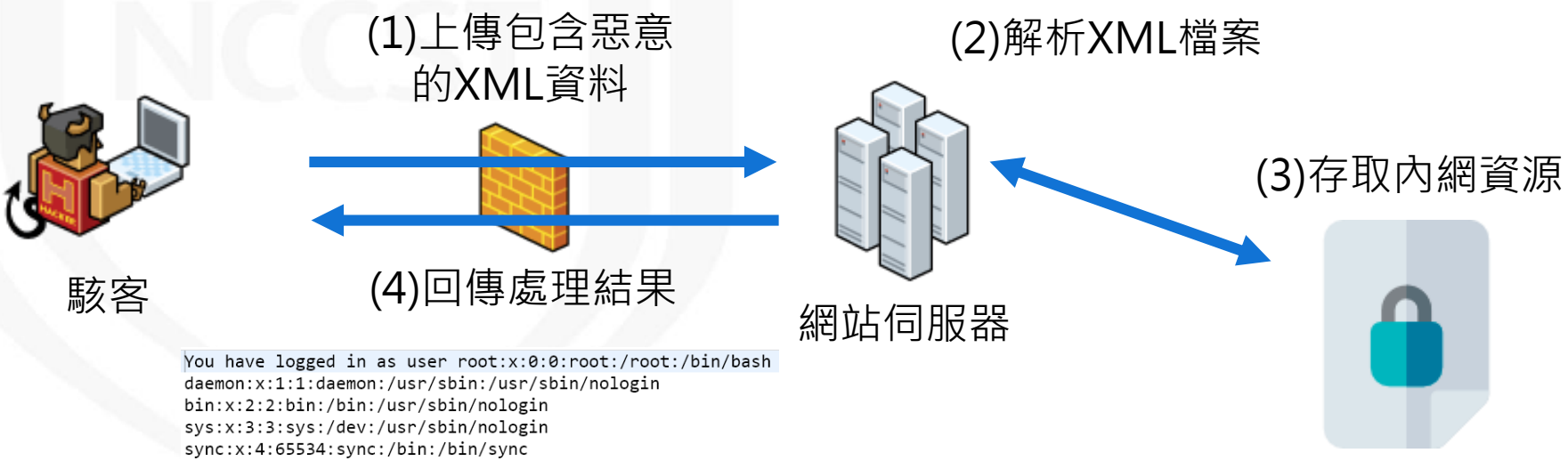



```
You have logged in as user ian
```

A4. XML外部實體-攻擊手法(2/2)

- 攻擊行為範例
  - 透過XML提供的型態定義語法(Document type declaration)存取本地端資源，達到資料外洩的效果

```
<?xml version="1.0"?>
<!DOCTYPE foo [
    <!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
<foo>
    <user>&xxe;</user>
</foo>
```




```
You have logged in as user root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
```

A4. XML外部實體-防範方法

- 不同XML Parser有各自的設定禁止方式
  - https://www.owasp.org/index.php/XML_External_Entity_(XXE)_Prevention_Cheat_Sheet

- JAVA
  - 使用XMLReader元件

```
XMLReader spf = XMLReaderFactory.createXMLReader();
spf.setFeature("http://xml.org/sax/features/external-general-entities", false);
spf.setFeature("http://xml.org/sax/features/external-parameter-entities", false);
spf.setFeature("http://apache.org/xml/features/nonvalidating/load-external-dtd",false);
```

  - 使用XPathExpression元件

```
DocumentBuilderFactory df = DocumentBuilderFactory.newInstance();
df.setAttribute(XMLConstants.ACCESS_EXTERNAL_DTD, "");
df.setAttribute(XMLConstants.ACCESS_EXTERNAL_SCHEMA, "");
DocumentBuilder builder = df.newDocumentBuilder();
xPathExpression.evaluate( builder.parse(new ByteArrayInputStream(xml.getBytes())) );
```

- .NET Framework

XML Parser	Safe by Default?
LINQ to XML	Yes
XmlDictionaryReader	Yes
XmlDocument	
...prior to 4.5.2	No
...in versions 4.5.2 +	Yes
XmlNodeReader	Yes
XmlReader	Yes
XmlTextReader	
...prior to 4.5.2	No
...in versions 4.5.2 +	Yes
XPathNavigator	
...prior to 4.5.2	No
...in versions 4.5.2 +	Yes
XslCompiledTransform	Yes

  - XmlDocument
    - xmlDoc.XmlResolver = null;
  - XmlTextReader
    - reader.ProhibitDtd = true;

A4. XML外部實體-需求驗證項目

資安防護基準作業規定		需求項目	需求範例
系統與服務獲得 (System and Services Acquisition)	系統發展生命週期開發階段 (System Development Life Cycle-Develop)	應注意避免軟體常見漏洞 (如OWASP TOP 10) 及實作必要控制措施	以源碼檢測確認弱點， 關閉XML外部引入功能

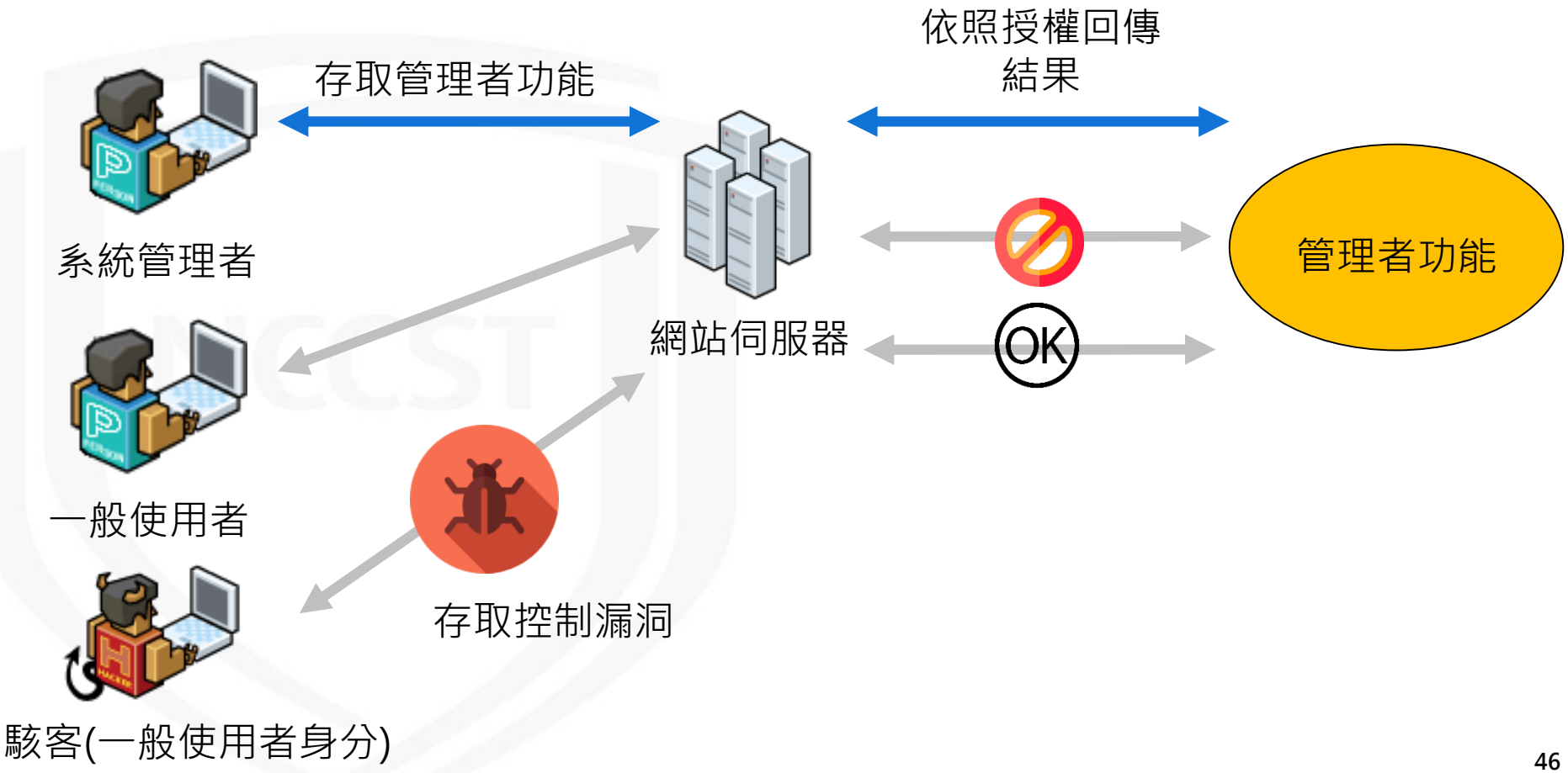
安全等級

- 普級
- 中級
- 高級

A5.存取控制相關功能缺陷 (Broken Access Control)

A5.存取控制相關功能缺陷-弱點說明

- 系統服務有存取控制漏洞導致駭客可以進行未授權的行為



A5.存取控制相關功能缺陷-攻擊手法

- 駭客竄改傳送內容，將其送到網頁伺服器

存取網址

請求參數

cookie

駭客

(1)傳送惡意竄改資料

(3)回傳處理結果

網站伺服器

(2)存取控制有缺陷導致允許進行未授權之行為

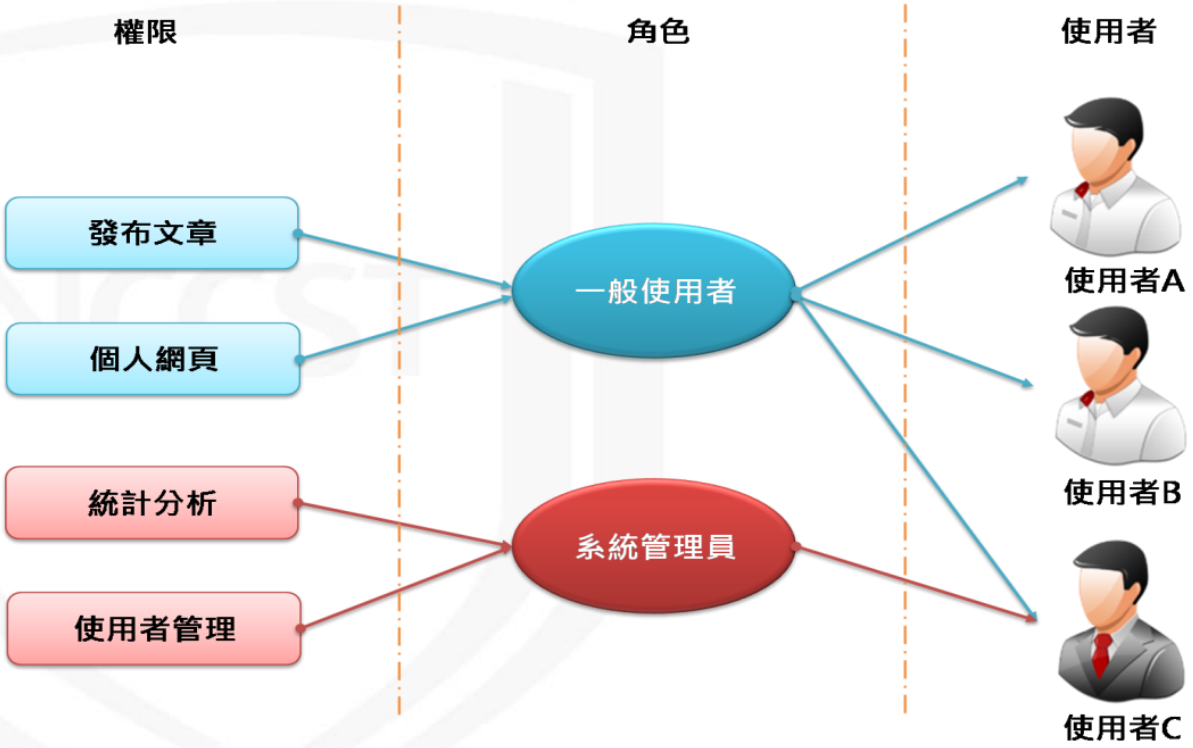
http://example.com/example.jsp?account_id=1

A5.存取控制相關功能缺陷-防範方式(1/4)

- URL修改
  - 採用RBAC集中控管授權機制
- 客戶端參數修改
  - 請求參數必須在伺服器端進行驗證

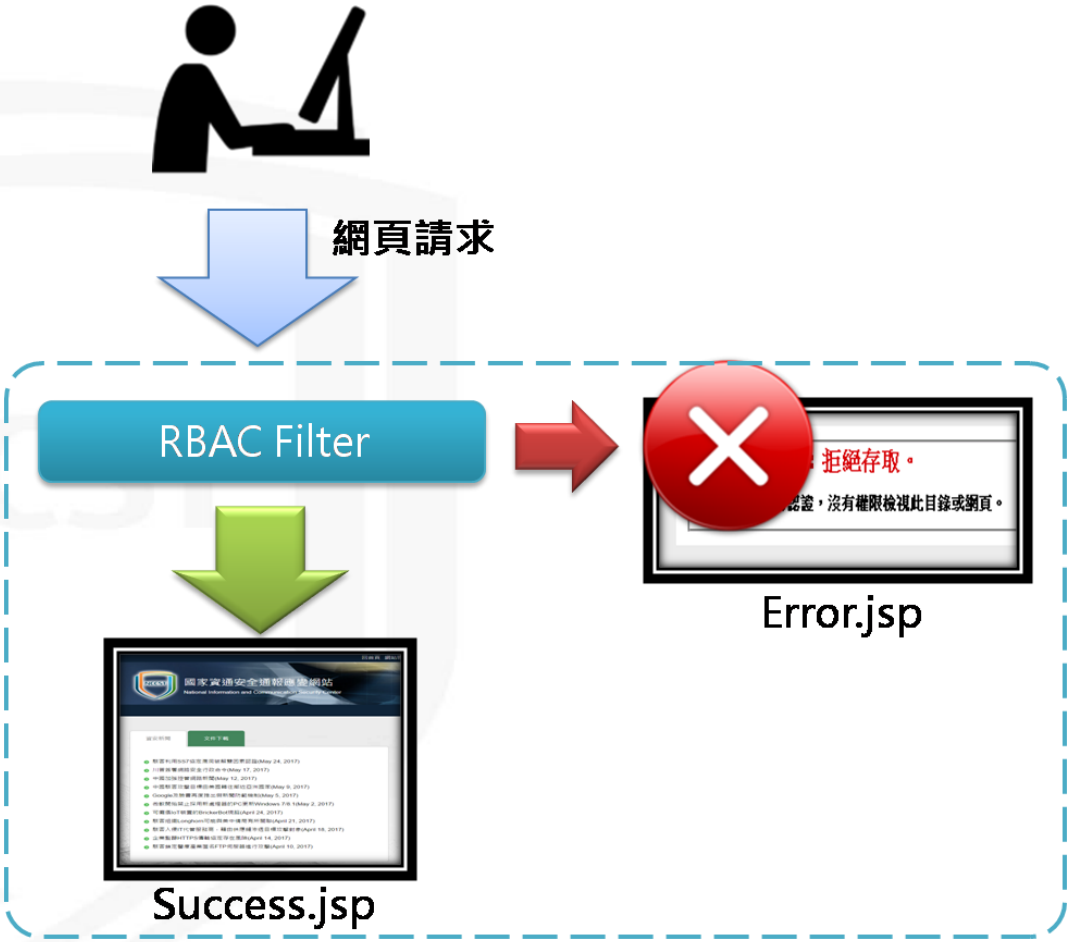
A5.存取控制相關功能缺陷-防範方式(2/4)

- 採用RBAC集中控管授權機制
  - 以角色為基礎的存取控制。不直接賦予使用者權限，而是將權限賦予角色



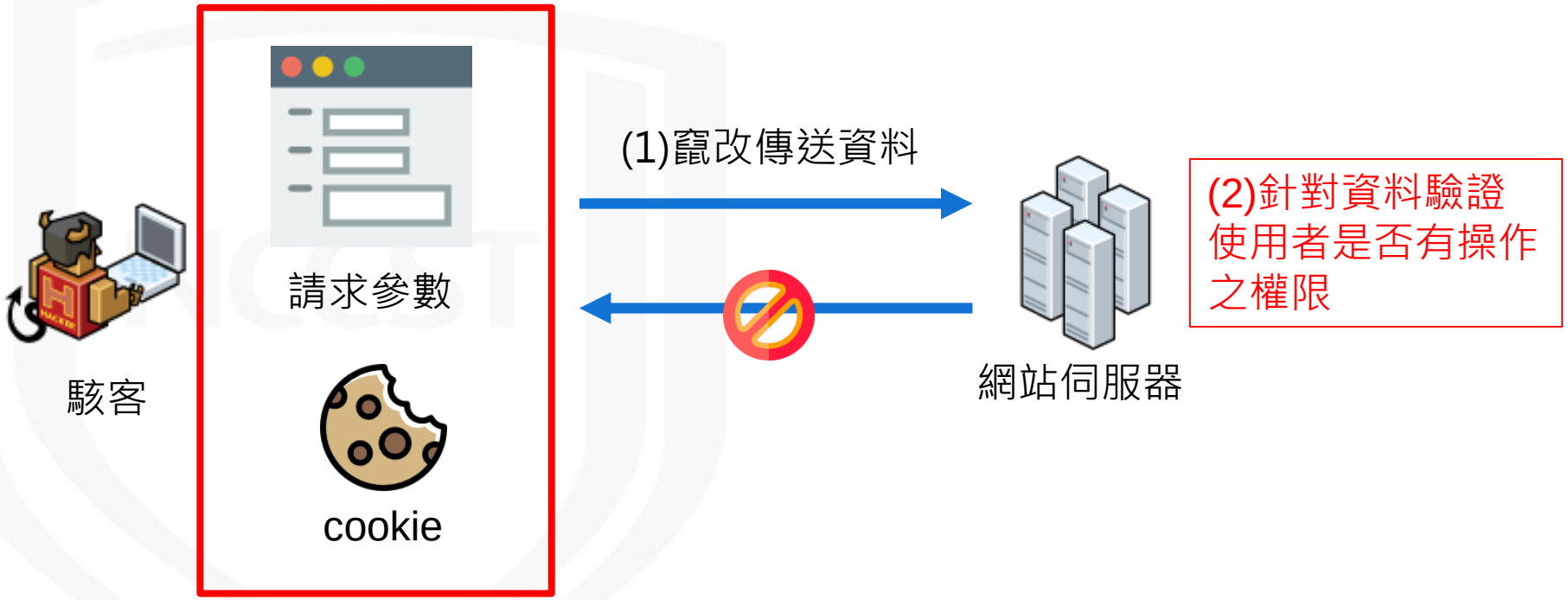
A5.存取控制相關功能缺陷-防範方式(3/4)

- 可利用Filter來檢查該帳號是否具備所請求資源的權限



A5.存取控制相關功能缺陷-防範方式(4/4)

- 請求參數必須在伺服器端進行驗證
  - 前端送上來的任何資訊皆可以被修改



A5.存取控制相關功能缺陷-需求驗證項目

資安防護基準作業規定		需求項目	需求範例
存取控制 (Access Control)	最小權限 (Least Privilege)	對於每一種允許之遠端存取類型，都應先取得授權，建立使用限制、組態需求、連線需求及文件化	伺服器端RBAC Filter檢查授權
		資訊系統使用者授權檢查應於伺服端集中過濾	
系統與資訊完整性 (System and Information Integrity)	軟體及資訊完整性 (Software, Firmware, and Information Integrity)	使用者輸入資料合法性檢查應置放於應用系統伺服器	以正規表示式規範使用者輸入欄位，並實作於伺服器端

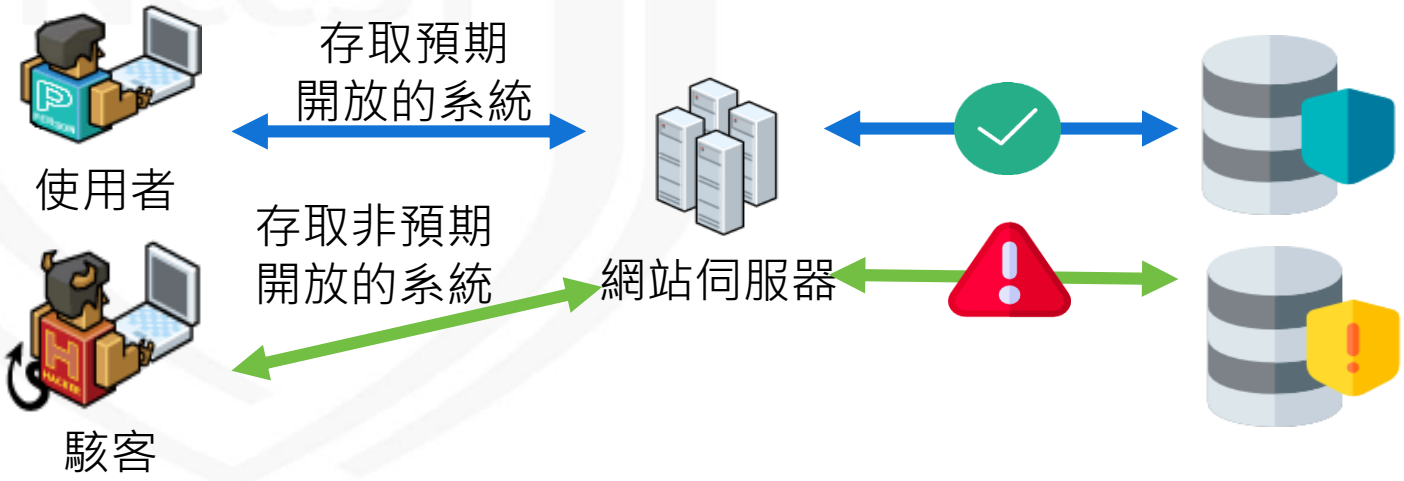
安全等級

- 普級
- 中級
- 高級

A6.不當的安全組態設定 (Security Misconfiguration)

A6.不當的安全組態設定-弱點說明

- 與安全功能機制相關的系統組態或Web應用程式參數被錯誤的設定或是強度不足
- 組態問題通常有
  - 預設帳號密碼
  - 對安全配置不清楚或設定不正確
  - 開放不必要的功能



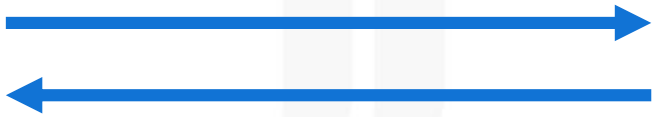
A6.不當的安全組態設定-攻擊手法(1/2)

- 預設帳號密碼
  - 駭客得到特定裝置的預設密碼後，利用預設密碼嘗試對系統進行登入

Vendor	Model	Version	Access Type	Username	PASSWORD
3COM	CoreBuilder	7000/6000/3500/2500	Telnet	debug	synnet
3COM	CoreBuilder	7000/6000/3500/2500	Telnet	tech	tech
3COM	HiPerARC	v4.1.x	Telnet	adm	(none)
3COM	LANplex		2500 Telnet	debug	synnet
3COM	LANplex		2500 Telnet	tech	tech
3COM	LinkSwitch	2000/2700	Telnet	tech	tech
3COM	NetBuilder		SNMP		ANYCOM
3COM	NetBuilder		SNMP		ILMI
3COM	Netbuilder		Multi	admin	(none)

駭客

(1)透過預設密碼
嘗試登入系統



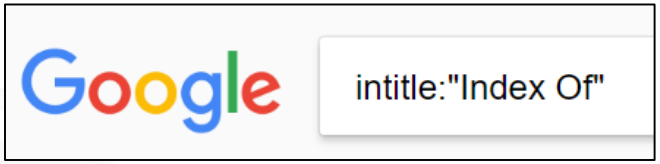
網站伺服器

(2)若系統有未修改之預設密碼，則可以進行登入動作

A6.不當的安全組態設定-攻擊手法(2/2)

- 對安全配置不清楚或設定不正確
  - –駭客使用特定工具對系統進行探測，進而執行惡意行為

(1)透過搜尋引擎或是工具尋來協助



駭客

(2)透過組態弱點進行攻擊

網站伺服器

(3)依照組態弱點達到不同的攻擊效果


```
<servlet>
    <servlet-name>default</servlet-name>
    <servlet-class>
      org.apache.catalina.servlets.DefaultServlet
    </servlet-class>
    <init-param>
        <param-name>debug</param-name>
        <param-value>0</param-value>
    </init-param>
    <init-param>
        <param-name>listings</param-name>
        <param-value>true</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
```

A6.不當的安全組態設定-防範方式

- 針對不同層面的組態設定做防護

- 功能最小化
  - 關閉不必要的功能

- 設定正確
  - 修改預設帳密
  - 定期更新元件
  - 注意安全設定

A6.不當的安全組態設定-需求驗證項目

資安防護基準作業規定		需求項目	需求範例
系統與服務獲得 (System and Services Acquisition)	系統發展生命週期開發階段 (System Development Life Cycle-Develop)	應注意避免軟體常見漏洞 (如OWASP TOP 10) 及實作必要控制措施	以源碼檢測確認排除常見漏洞
		應針對資訊系統之錯誤與例外進行適當處理	設置客製化錯誤頁面
資料儲存之安全 (Protection of Information at Rest	資料儲存之安全 (Protection of Information at Rest)	靜置資訊指資訊位於資訊系統特定元件，如儲存設備上之狀態。與系統相關需要保護的資訊，例如：資訊系統組態設定。該等資訊應予以保護	禁止目錄尋訪，正確實作檔案下載功能

安全等級

- 普級
- 中級
- 高級

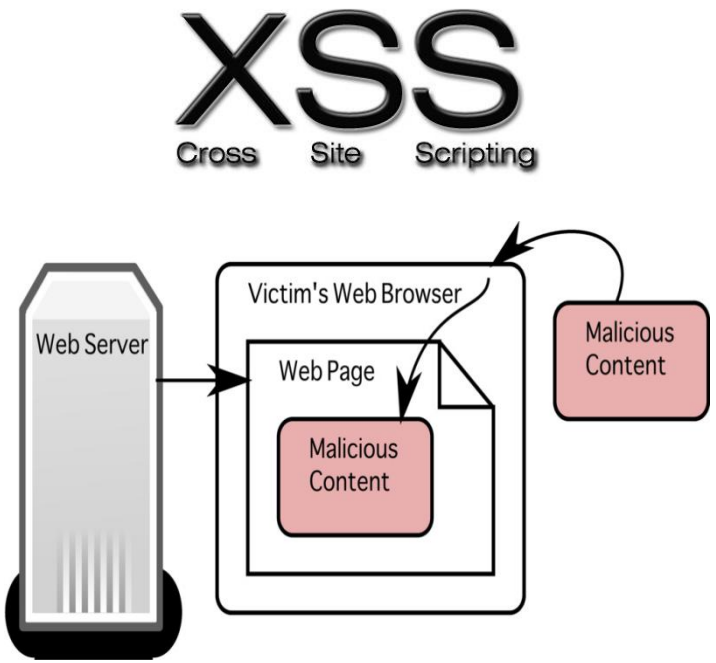
A7.跨站腳本攻擊
(Cross-Site Scripting, XSS)

A7.跨站腳本攻擊-弱點說明

- Cross-site scripting，通常簡稱為XSS

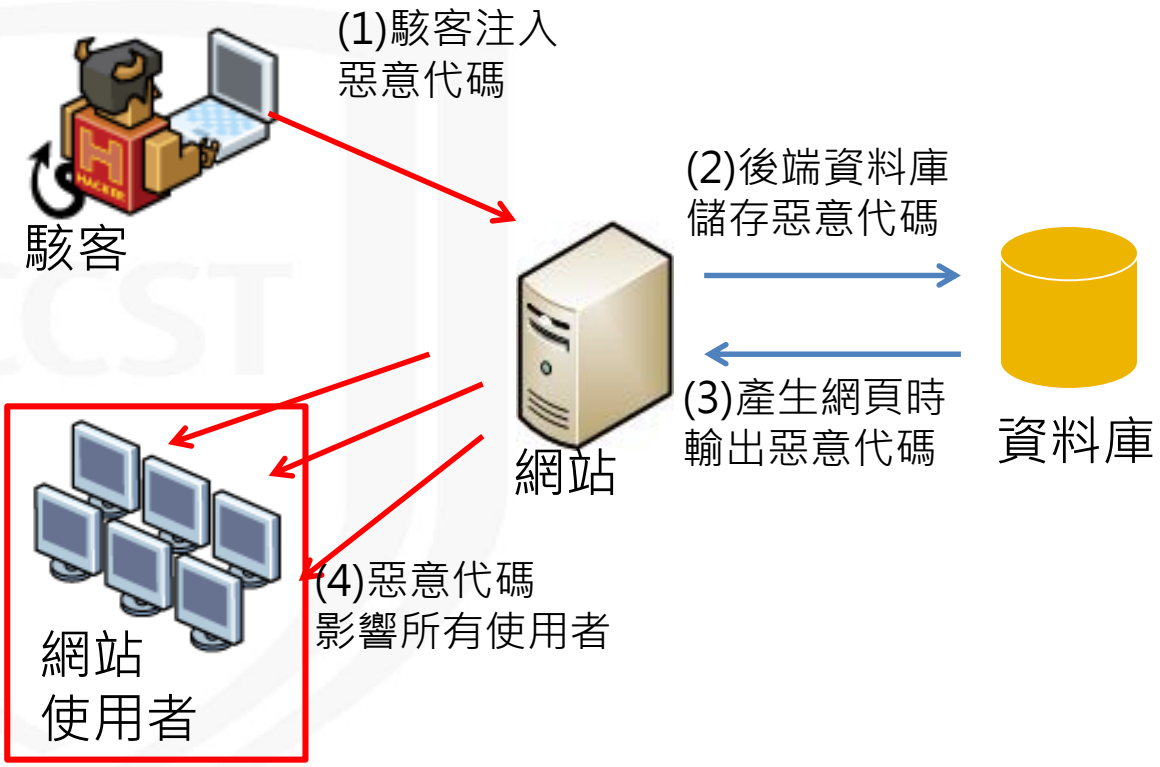
- XSS弱點允許惡意使用者將程式碼注入到網頁上，其他使用者在觀看網頁時就會載入並執行到惡意程式碼

- XSS風險包含
  - 盜用cookie
  - 網站轉址
  - 修改網站內容
  - 執行客製化腳本以提高權限等動作



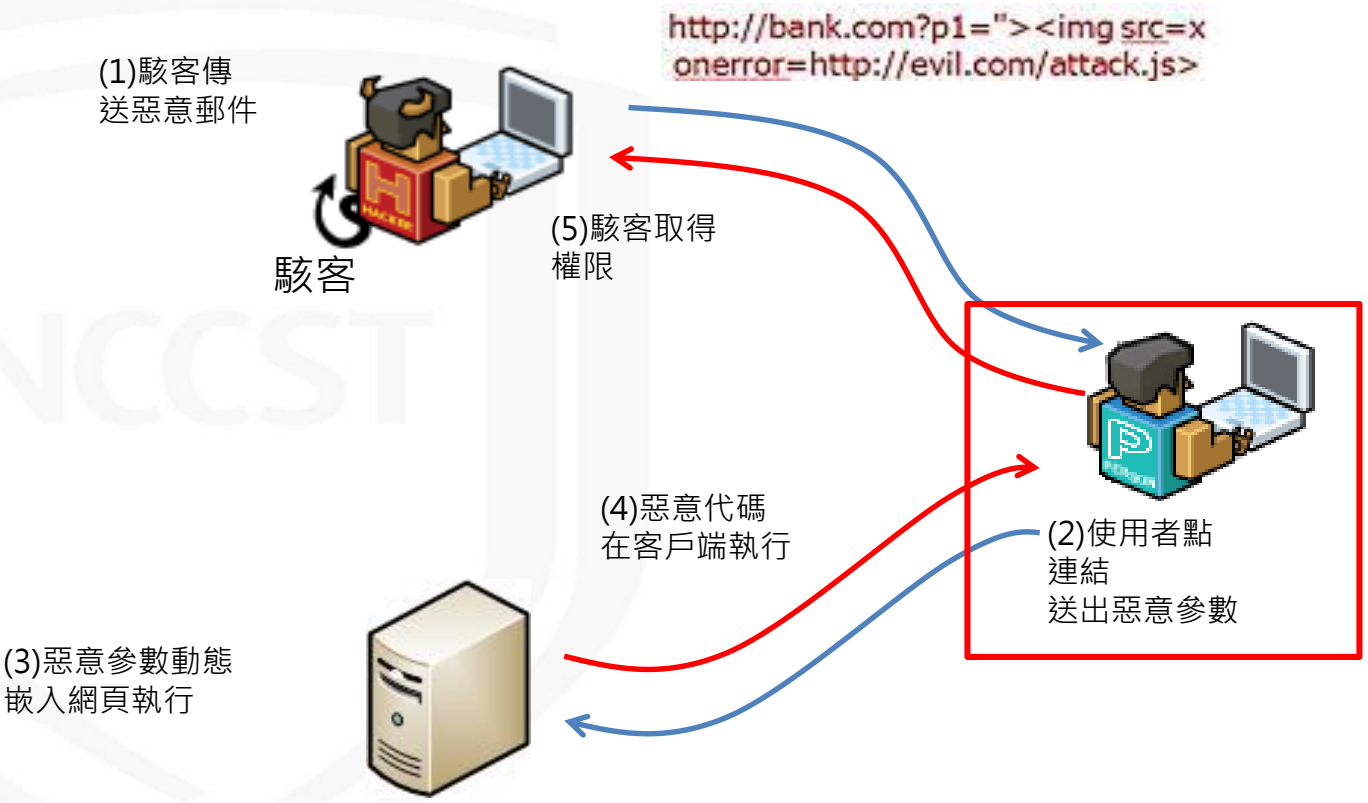
A7.跨站腳本攻擊-攻擊手法(1/3)

- 儲存式(Stored XSS)
  - –駭客將惡意腳本存入後端資料庫



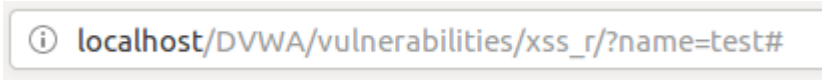
A7.跨站腳本攻擊-攻擊手法(2/3)

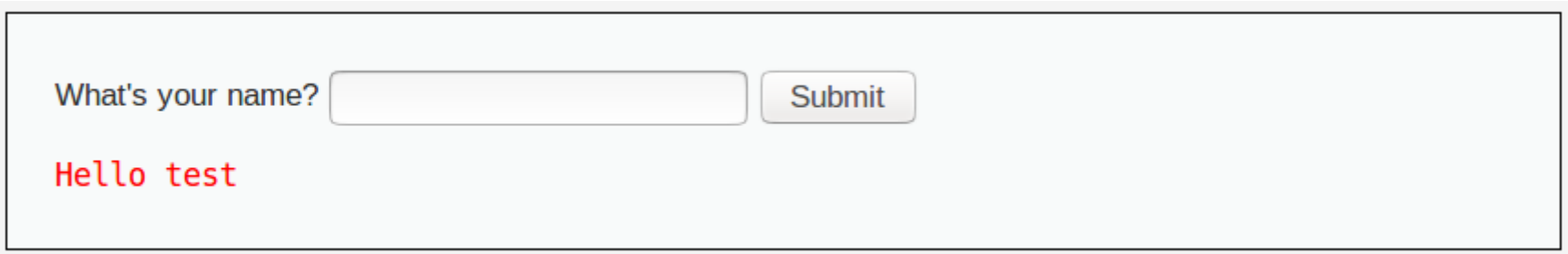
- 反射式(Reflected XSS)
  - 駭客誘騙受害者點選含有惡意腳本之連結



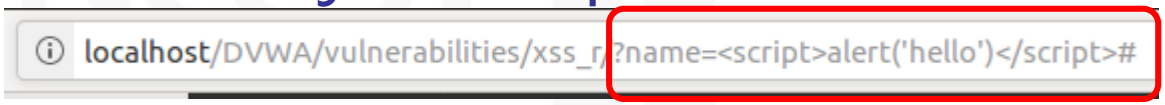
A7.跨站腳本攻擊-攻擊手法(3/3)

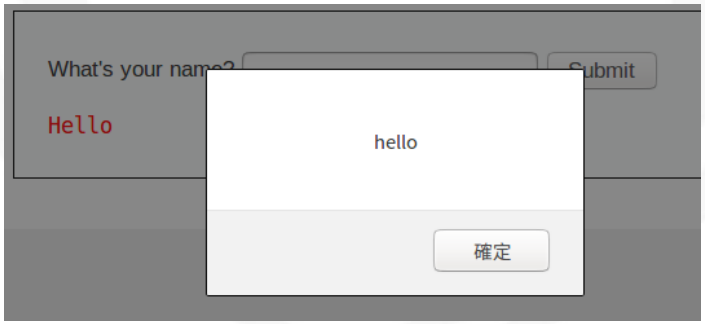
- 於參數帶入使用者帳號，顯示在頁面上





- 改為輸入 javascript 語法



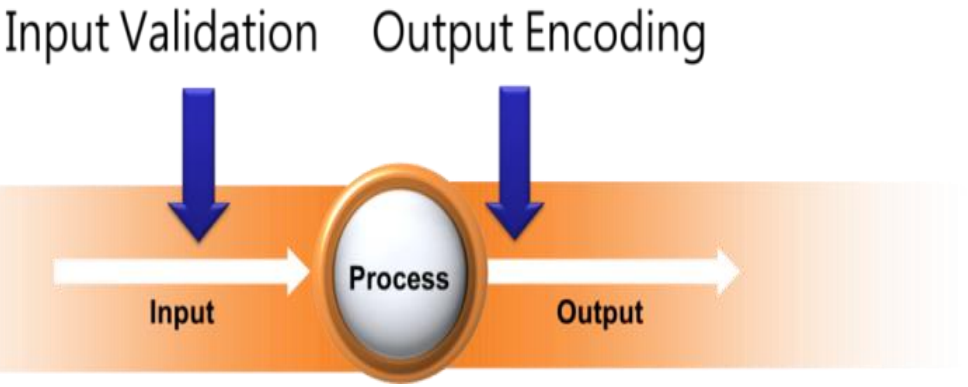


A7.跨站腳本攻擊-防禦方式(1/2)

- Input Validation
  - 將使用者所提供的內容進行過濾，限制可使用的字元
    - 黑名單
      - 禁止使用 < > ’ ” 等符號
    - 白名單(建議使用)
      - 使用正規表示式(Regular Expression)
      - 檢查所有使用者輸入資料
      - 例如帳號欄位只允許輸入英文數字

```
中文:[\u4e00-\u9fa5]
英文字母:[a-zA-Z]
數字:[0-9]
中文，英文字母和數字_:
^[\u4e00-\u9fa5_a-zA-Z0-9]+$
同時判斷輸入長度：
[\u4e00-\u9fa5_a-zA-Z0-9_]{4,10}
```

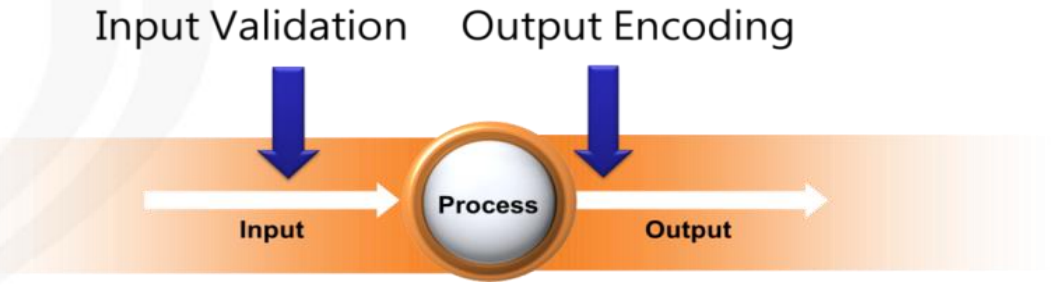
  - 對象例如：
    - URL parameters
    - 任何文字輸入欄位



A7.跨站腳本攻擊-防禦方式(2/2)

- Output Encoding
  - 讓使用者提供的資料不會被瀏覽器認為是指令碼
    - 例如<script>編碼後變成 <script>
    - 使用ESAPI提供之方法
  - 對象例如：
    - HTML
    - HTML Attribute
    - URL
    - JavaScript
    - CSS

- String encodeForHTML(String Input)
- String encodeForHTMLAttribute(String Input)
- String encodeForURL(String Input)
- String encodeForJavaScript(String Input)
- String encodeForCSS(String Input)



A7.跨站腳本攻擊-需求驗證項目

資安防護基準作業規定		需求項目	需求範例
系統與服務獲得 (System and Services Acquisition)	系統發展生命週期開發階段 (System Development Life Cycle-Develop)	應注意避免軟體常見漏洞 (如OWASP TOP 10) 及實作必要控制措施	以源碼檢測確認弱點，對輸入資料進行格式驗證，對輸出資料進行編碼
系統與資訊完整性 (System and Information Integrity)	軟體及資訊完整性 (Software, Firmware, and Information Integrity)	使用者輸入資料合法性檢查應置放於應用系統伺服器	以正規表示式規範使用者輸入欄位，並實作於伺服器端

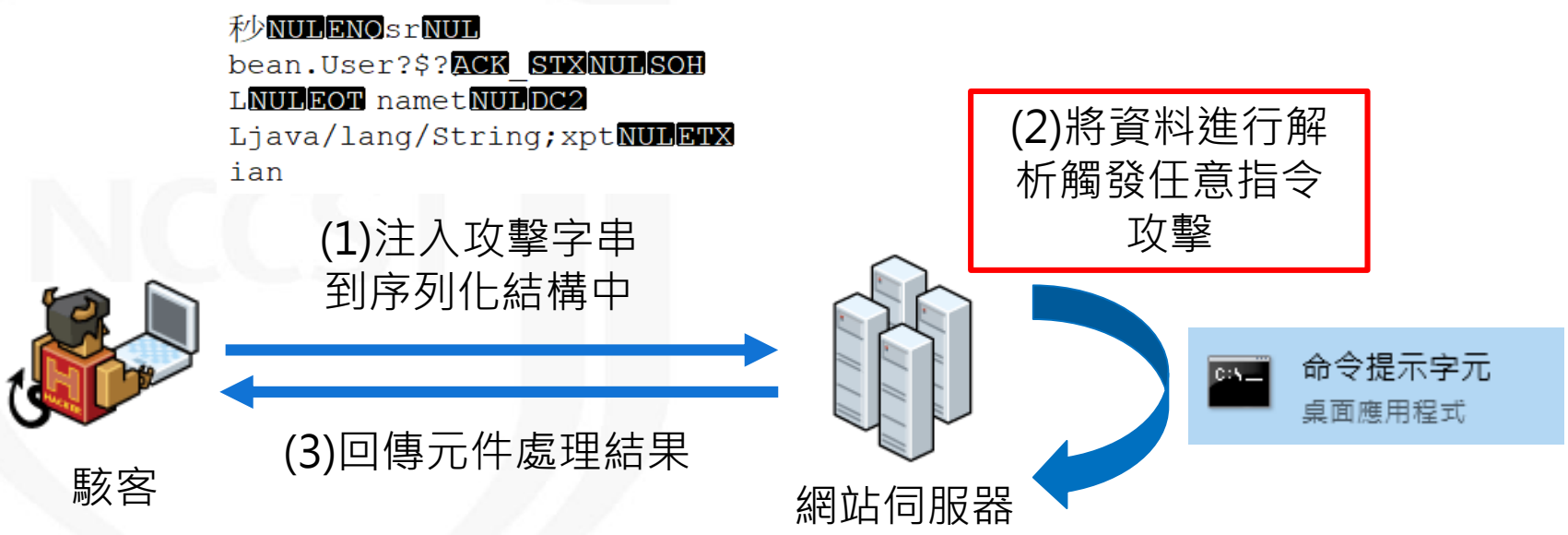
安全等級

普級
中級
高級

A8.不安全的反序列化
(Insecure Deserialization)

A8.不安全的反序列化-弱點說明(1/2)

- 駭客將攻擊字串注入序列化的結構中，使得網站伺服器解析後出現非預期的結果，並執行駭客輸入的指令



A8.不安全的反序列化-弱點說明(2/2)

- 序列化(Serialization)
  - 是一種用來轉換程式語言物件(Object)格式的動作
  - 可將物件轉換為適合網路傳輸或於檔案系統持續儲存的格式，例如JSON、YAML或XML
- 反序列化(Deserialization)
  - 將儲存的資料格式轉回程式語言物件的動作

```
public class User implements Serializable {

    private static final long serialVersionUID = 1L;

    private String name;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

Object

Serialization

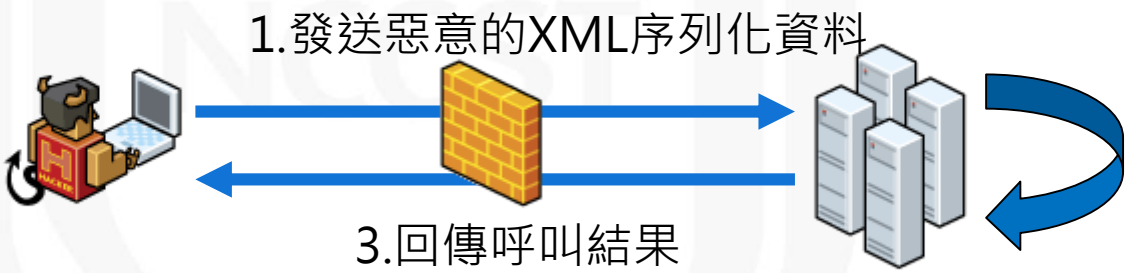
Deserialization

```
秒NULENOsrNUL
bean.User?$?ACK_STXNULSOH
LNULEOT nametNULDC2
Ljava/lang/String;xptNULETX
ian
```

Stream of Bytes

A8.不安全的反序列化-攻擊手法

```
<map>
  <entry>
    <jdk.nashorn.internal.objects.NativeString>
      <flags>0</flags>
      <value class="com.sun.xml.internal.bind.v2.runtime.unmarshaller.Base64Data">
        <dataHandler>
          <dataSource class="com.sun.xml.internal.ws.encoding.xml.XMLMessage$XmlDataSource">
            <is class="javax.crypto.CipherInputStream">
              <cipher class="javax.crypto.NullCipher">
                <initialized>false</initialized>
                <opmode>0</opmode>
                <serviceIterator class="javax.imageio.spi.FilterIterator">
                  <iter class="javax.imageio.spi.FilterIterator">
                    <iter class="java.util.Collections$EmptyIterator"/>
                    <next class="java.lang.ProcessBuilder">
                      <command>
                        <string>cmd.exe</string><string>/c</string><string>calc.exe</string>
                      </command>
                      <redirectErrorStream>false</redirectErrorStream>
                    </next>
                  </iter>
```



2.解析惡意資料並觸發攻擊指令(calc.exe)

```
<!DOCTYPE html><html><head><title>Apache Tomcat/8.0.47 - Error report</title>
erif;color:white;background-color:#525D76;font-size:22px;} H2 {font-family:Ta
;font-size:16px;} H3 {font-family:Tahoma,Arial,sans-serif;color:white;backgro
a,Arial,sans-serif;color:black;background-color:white;} B {font-family:Tahoma
 {font-family:Tahoma,Arial,sans-serif;background:white;color:black;font-size:
ht: 1px; background-color: #525D76; border: none;}</style> </head><body><h1>H
ecurity.Provider$Service : java.lang.String cannot be cast to java.security.P
```

A8.不安全的反序列化-防範手法

- 輸入驗證
  - 於反序列化時限制可以處理的物件名稱，透過白名單防禦(效果較佳)

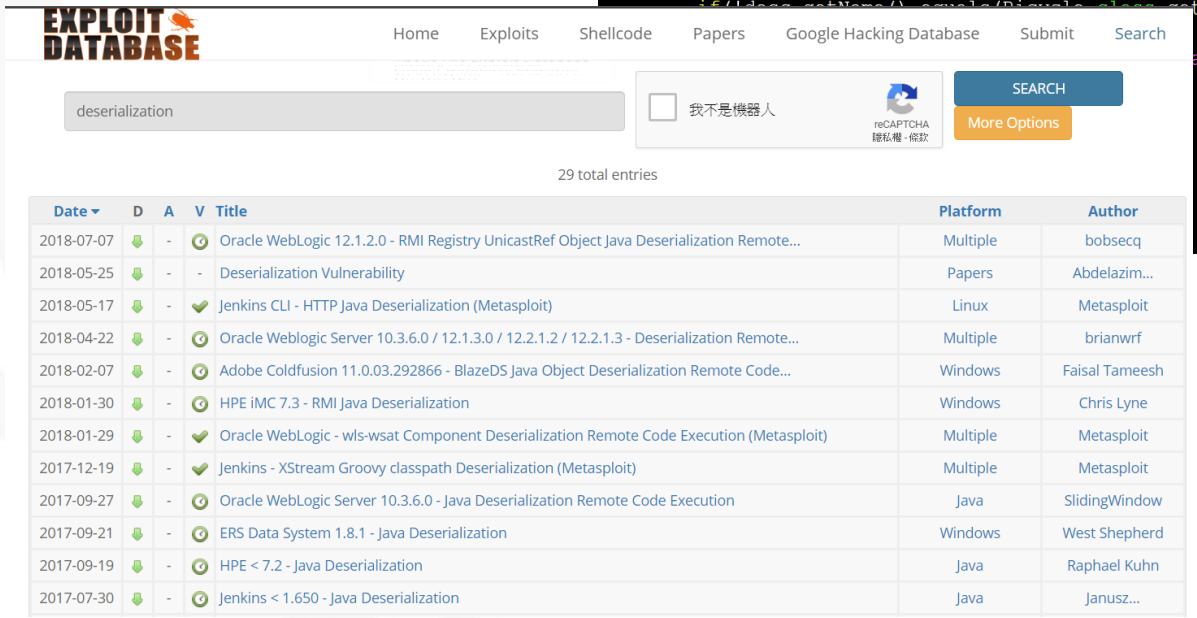
```
public class LookAheadObjectInputStream extends ObjectInputStream {

    public LookAheadObjectInputStream(InputStream input) throws IOException {
        super(input) ;
    }

    @Override
    protected Class<?> resolveClass(ObjectStreamClass desc) throws IOException,ClassNotFoundException {
        System.out.println("I get Class Name:"+desc.getName()) ;

        if(!desc.getName().equals(BigInteger.class.getName())) {
                                                                    attempt",
```

於 resolveClass 裡面檢查是否為白名單類別



Date	D	A	V	Title	Platform	Author
2018-07-07		-		Oracle WebLogic 12.1.2.0 - RMI Registry UnicastRef Object Java Deserialization Remote...	Multiple	bobsecq
2018-05-25		-	-	Deserialization Vulnerability	Papers	Abdelazim...
2018-05-17		-		Jenkins CLI - HTTP Java Deserialization (Metasploit)	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...	Multiple	brianwrf
2018-02-07		-		Adobe Coldfusion 11.0.03.292866 - BlazeDS Java Object Deserialization Remote Code...	Windows	Faisal Tameesh
2018-01-30		-		HPE iMC 7.3 - RMI Java Deserialization	Windows	Chris Lyne
2018-01-29		-		Oracle WebLogic - wls-wsat Component Deserialization Remote Code Execution (Metasploit)	Multiple	Metasploit
2017-12-19		-		Jenkins - XStream Groovy classpath Deserialization (Metasploit)	Multiple	Metasploit
2017-09-27		-		Oracle WebLogic Server 10.3.6.0 - Java Deserialization Remote Code Execution	Java	SlidingWindow
2017-09-21		-		ERS Data System 1.8.1 - Java Deserialization	Windows	West Shepherd
2017-09-19		-		HPE < 7.2 - Java Deserialization	Java	Raphael Kuhn
2017-07-30		-		Jenkins < 1.650 - Java Deserialization	Java	Janusz...

A8.不安全的反序列化-需求驗證項目

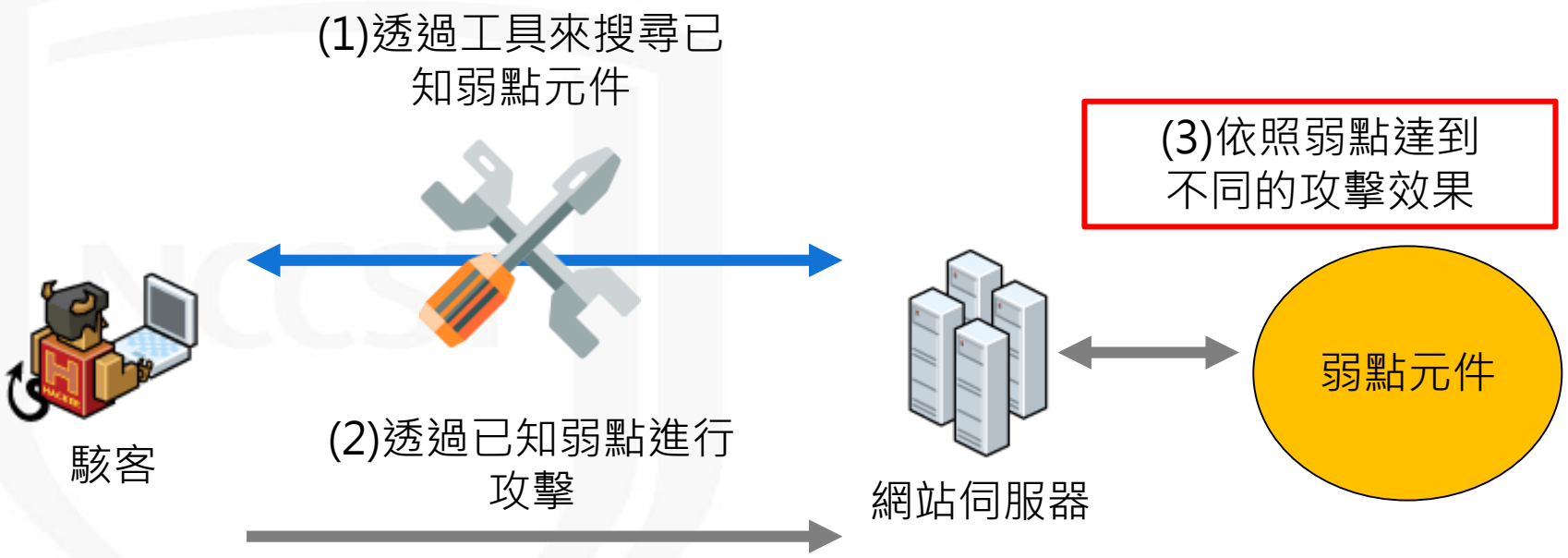
資安防護基準作業規定		需求項目	需求範例
系統與服務獲得 (System and Services Acquisition)	系統發展生命週期開發階段 (System Development Life Cycle-Develop)	應注意避免軟體常見漏洞 (如OWASP TOP 10) 及實作必要控制措施	以源碼檢測確認排除常見漏洞
系統與資訊完整性 (System and Information Integrity)	軟體及資訊完整性 (Software, Firmware, and Information Integrity)	使用者輸入資料合法性檢查應置放於應用系統伺服器	透過白名單方式驗證反序列化物件合法性

安全等級
- 普級
- 中級
- 高級

A9.使用具有已知弱點的元件 (Using Components with Known Vulnerabilities)

A9.使用具有已知弱點的元件-弱點說明

- 駭客透過掃描工具偵測到系統使用具有弱點之元件，並利用該弱點進行攻擊



A9.使用具有已知弱點的元件-防範方式(1/2)

- 識別整理專案中所有使用到的元件、底層軟體及其版本資訊

- 定期維護該清單並確保正確性
- 使用外部元件或軟體，注意其安全通告並且評估更新
  - 注意Security Alert網站，例如CVE (http://cve.mitre.org/cve/)
  - 訂閱廠商的安全通告

A9.使用具有已知弱點的元件-防範方式(2/2)

- CVE (Common Vulnerabilities and Exposures)
  - 國際著名的漏洞安全資料庫，其內容包含對已知安全漏洞的標準化名稱列表，讓使用者能夠更有效率的識別、發現和修復元件的漏洞問題

CVE-ID

CVE-2018-1327 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions

Description

The Apache
crafted XML
http://strut
from the Ap

Impact

CVSS v3.0 Severity and Metrics:
Base Score: 7.5 HIGH
Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H (V3 legend)

CVSS v2.0 Severity and Metrics:
Base Score: 5.0 MEDIUM
Vector: (AV:N/AC:L/Au:N/C:N/I:N/A:P) (V2 legend)

Impact
Exploit

Vulnerable software and versions Switch to CPE 2.2

Configuration 1

OR

✱ cpe:2.3:a:apache:struts:*:*:*:*:*:*:*:* ✚ versions from (including) 2.1.1 up to (including) 2.5.14.1

A9.使用具有已知弱點的元件-需求驗證項目

資安防護基準作業規定		需求項目	需求範例
系統與服務獲得 (System and Services Acquisition)	系統發展生命週期開發階段 (System Development Life Cycle-Develop)	應注意避免軟體常見漏洞 (如OWASP TOP 10) 及實作必要控制措施	以源碼檢測確認， 並更新或修補有漏洞之元件

安全等級

- 普級
- 中級
- 高級

A10記錄與監控不足 (Insufficient Logging & Monitoring)

A10.記錄與監控不足-弱點說明

- 未記錄到駭客的攻擊行為導致無法被偵測或是追蹤攻擊資訊

駭客

(1)透過不同手法攻擊
網站伺服器

(2)依照弱點達到
不同的攻擊效果

網站伺服器

A10.記錄與監控不足-防範方式(1/4)

- 目標
  - 確保資訊系統產生之記錄完整性，並滿足可被歸責性(Accountability)之標準
- 項目
  - 記錄時機點
  - 記錄內容
  - 保護記錄

A10.記錄與監控不足-防範方式(2/4)

- 身分驗證機制(Authentication)
  - 驗證成功及失敗
  - 帳戶鎖定
  - 密碼變更
- 資料存取(Data Access)
  - 敏感資料被讀取
  - 資料異動及刪除
  - 資料結構變更
  - 管理者行為
- 功能錯誤(All Error Conditions)
  - 查詢語法執行失敗
  - 檔案存取錯誤
  - 非預期的狀態
  - 連線失敗(資料庫或其他外部系統等)
  - 逾時、效能問題

A10.記錄與監控不足-防範方式(3/4)

- 人
  - 使用者ID
    - 唯一識別的ID，但個資類型(身分證字號、Email)應除外
    - 避免多人共用同一帳號，造成無法歸責狀況
- 事
  - 行為描述
  - 錯誤代碼
    - 便於搜尋及問題追查
    - 避免不同錯誤使用相同代碼
- 時
  - 系統應設定定期校時
  - 應記錄至毫秒(millisecond)
- 地
  - Network Address (src、dst ip)
- 物
  - 物件資訊

A10.記錄與監控不足-防範方式(4/4)

- 保留方式
  - –Log應保留多久，是否有相關要求
  - –相對應的儲存空間需求
  - –遠端備份紀錄
- 有效性
  - –運用加密或雜湊處理機制，以保護稽核資訊

	uuser_account text	action text	category text	descr text	logtime timestamp without time zone	hash text
1	irwsasdsad	sendTo...	發送OTP	註冊驗證-...	2018-07-04 20:08:49.666	qswy3sJoKKqZ8BxUZe82NvnU...
2	irwsasdsad	sendTo...	發送OTP	註冊驗證-...	2018-07-04 20:08:49.706	trA+uiH0EKNLg92WL7feHdEAe...
3	asdqweqwdqwd	sendTo...	發送OTP	註冊驗證-...	2018-07-05 11:11:02.981	yZncoZ5VpoFEsoLXmS2RY65q7...
4	asdqweqwdqwd	sendTo...	發送OTP	註冊驗證-...	2018-07-05 11:11:03.043	9cWOLN2LDkdnkcUUEaZEWP7...
5	asdqweqwdqwd	checkT...	驗證OTP	驗證登入O...	2018-07-05 11:11:14.798	Wo50t0itVZIYZRdNZjr5KnXbalu...

A10.記錄與監控不足-需求驗證項目(1/2)

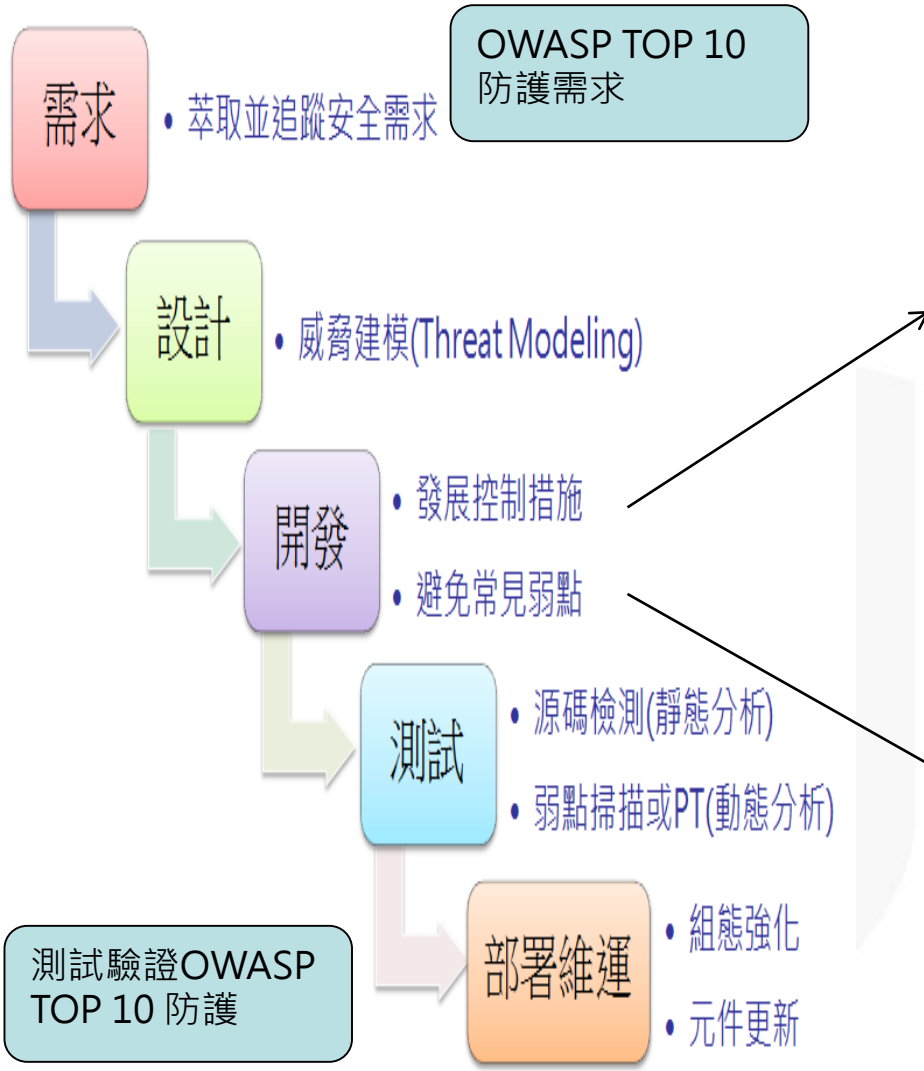
資安防護基準作業規定		需求項目	需求範例
稽核與可歸責性 (Audit and Accountability)	稽核紀錄內容 (Content of Audit Records)	確保資訊系統有稽核特定事件之能力，並決定有哪些特定事件在資訊系統中應該被稽核	具有單一Log機制，記錄帳號異動、密碼異動、登入失敗、帳號鎖定、管理者行為至稽核紀錄資料表 (包含人事時地物)
		應稽核資訊系統管理者帳號所執行之各項功能	
		資訊系統產生之稽核紀錄採用單一的日誌紀錄機制，確保輸出格式的一致性	
		資訊系統產生的稽核紀錄，應依需求納入額外的資訊	稽核紀錄依需求納入額外的資訊

A10.記錄與監控不足-需求驗證項目(2/2)

資安防護基準作業規定		需求項目	需求範例
稽核與可歸責性 (Audit and Accountability)	稽核儲存容量 (Audit Storage Capacity)	資訊系統應在稽核處理失效(如儲存容量不足)之情況下，採取適當之行動	寫入稽核紀錄資料表失敗時，發信通知管理員
	稽核處理失效之回應 (Response to Audit Processing Failures)	當機關規定需要即時通報的稽核失效事件發生時，資訊系統應在機關規定之時效內，對機關特定之人員、角色提出告警(如信件通知)	
	稽核資訊之保護 (Protection of Audit Information)	運用加密或雜湊處理機制，以保護稽核資訊	稽核紀錄表格內容留存雜湊值
		定期備份稽核紀錄到與原稽核系統不同之實體系統(如Log伺服器)	將log遠端儲存或以第三方機制寫出至log伺服器

總結

OWASP Top 10 對開發之重要性



資訊系統分級與資安防護基準作業規定

控制措施	安全等級			參考文件
	普	中	高	
系統發展生命週期開發階段(System Development Life Cycle-Develop)	1. 應針對安全需求實作必要控制措施。 2. 應注意避免軟體常見漏洞(如 OWASP TOP 10)及實作必要控制措施。 3. 應針對資訊系統之錯誤與例外進行適當處理。	執行等級「普」之所有控制措施。	1. 執行等級「普」之所有控制措施。 2. 執行「源碼掃描」安全檢測。 3. 具備系統嚴重錯誤之通知機制(例如電子郵件或簡訊)。	安全控制措施參考指引附件 18 SA 安全軟體發展流程參考指引 3.3 安全軟體開發

項次	OWASP Top 10 項目
A1	注入
A2	身分鑑別相關功能缺陷
A3	敏感資料暴露
A4	XML外部實體(XXE)
A5	存取控制相關功能缺陷
A6	不當的安全組態設定
A7	跨站腳本攻擊 (XSS)
A8	不安全的反序列化
A9	使用具有已知弱點的元件
A10	記錄與監控不足