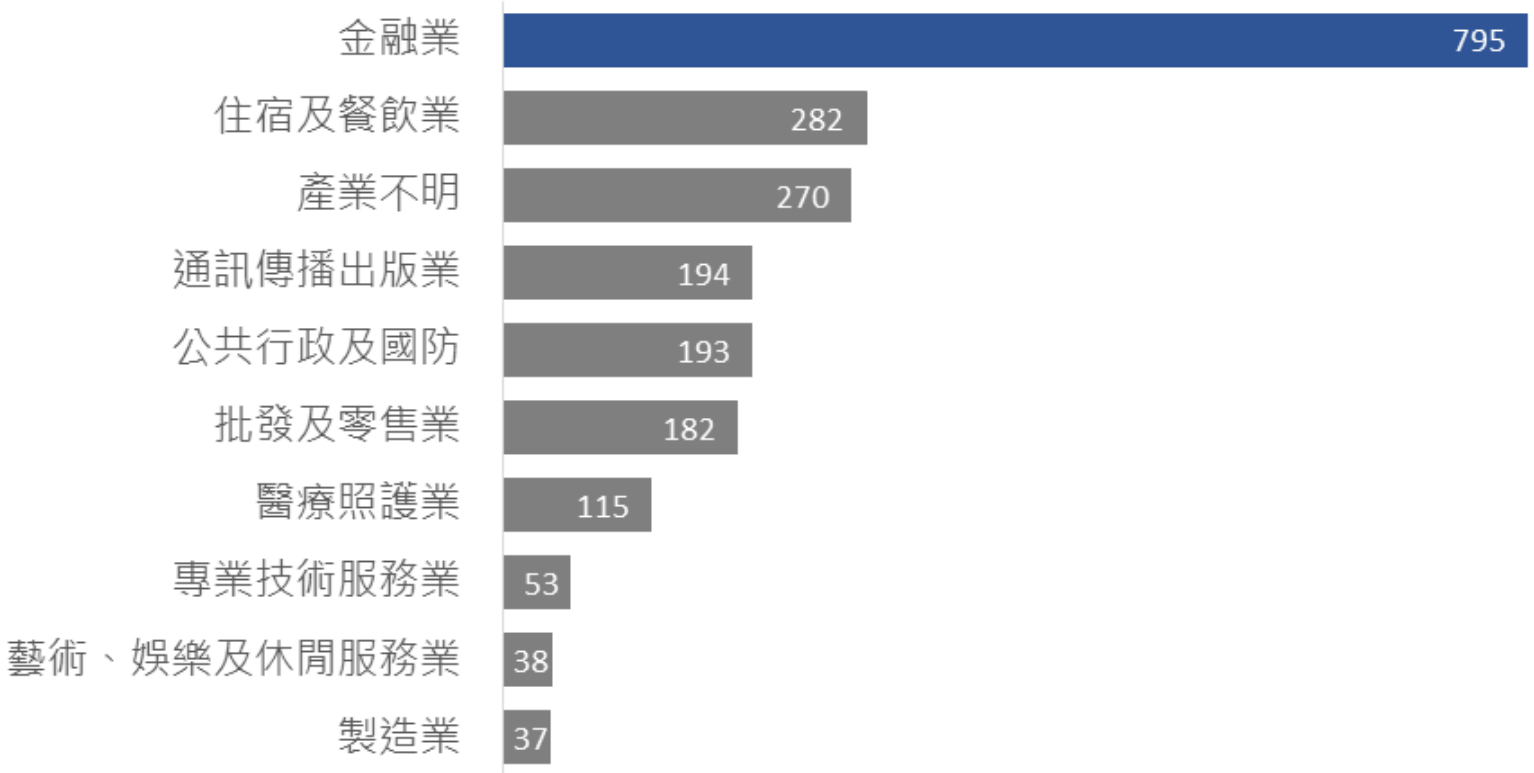
Java網站安全防護

建構安全的網站與應用程式

資料外洩災情嚴峻

2015各產業資料外洩件數



* 資料來源: Verizon 2016 Data Breach Investigations Report

金融業資料外洩災情最慘，
2016年金融業資安預算成長

率達**22.5%**

平均整體成長率為14.5%

盡早帶入軟體安全概念

軟體在上線營運後，
發現問題進行修補的相對成本，
為需求階段的

30倍

* 資料來源: NIST(National Institute of Standards and Technology)

我們的目標

1. 學習軟體開發安全技術的九大黃金準則。
2. 導入安全軟體發展生命週期各個階段實務，提升軟體品質。

九大黃金安全準則

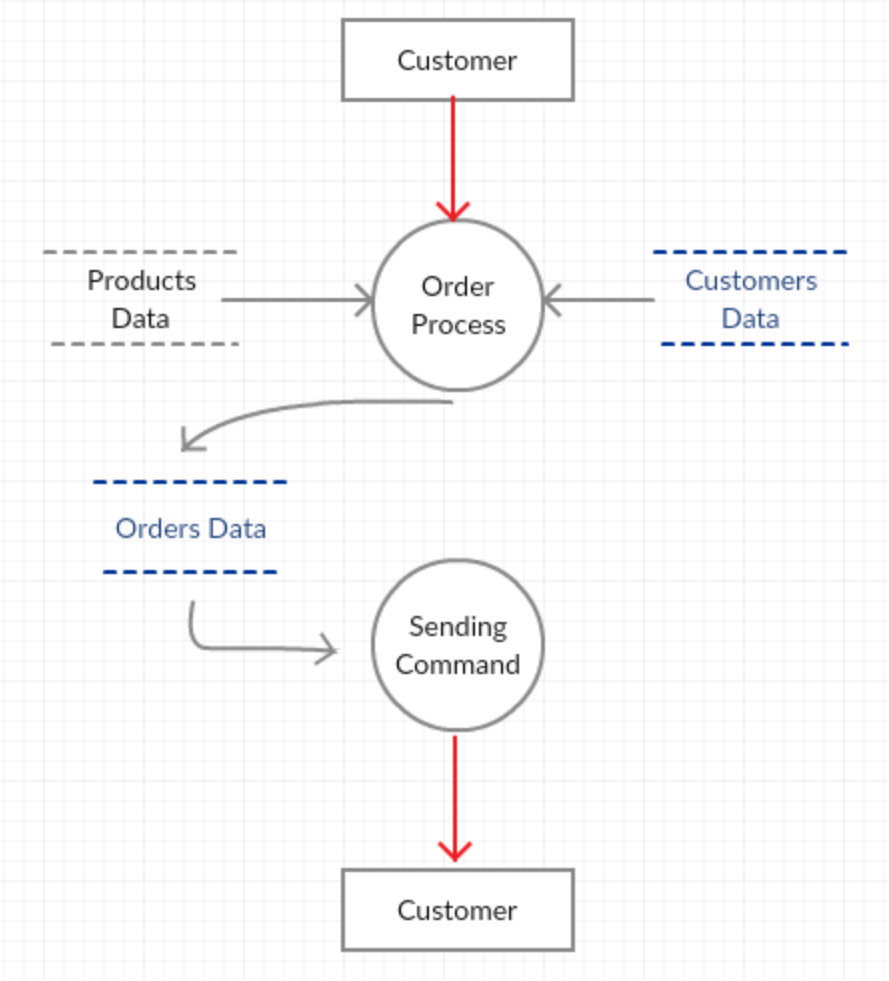
1 機密性(Confidentiality)
2 完整性(Integrity)
3 可用性(Availability)
4 身分認證(Authentication)
5 授權(Authorization)與存取控制(Access Control)
6 稽核(Auditing)與紀錄(Logging)
7 會談管理(Session Management)
8 錯誤與例外控制(Error and Exception Handling)
9 組態管理(Configuration Management)

1 比較加密相關密碼學

特性	對稱性	非對稱性
加密/解密金鑰	相同	不同(公鑰/私鑰)
速度	快	慢
密文大小	與原始內文相同或更少	超過原始內容大小
金鑰協議	金鑰交換問題	沒問題
保管金鑰數目	參與者數量	1
演算法	~~DES~~、RC系列、3DES、IDEA、AES	RSA、E

1 資料加密的時機

使用Data Flow Diagram (DFD)
確認加密時機，
信任邊界做傳輸加密，
資料儲存庫中機密資料做儲存加密。



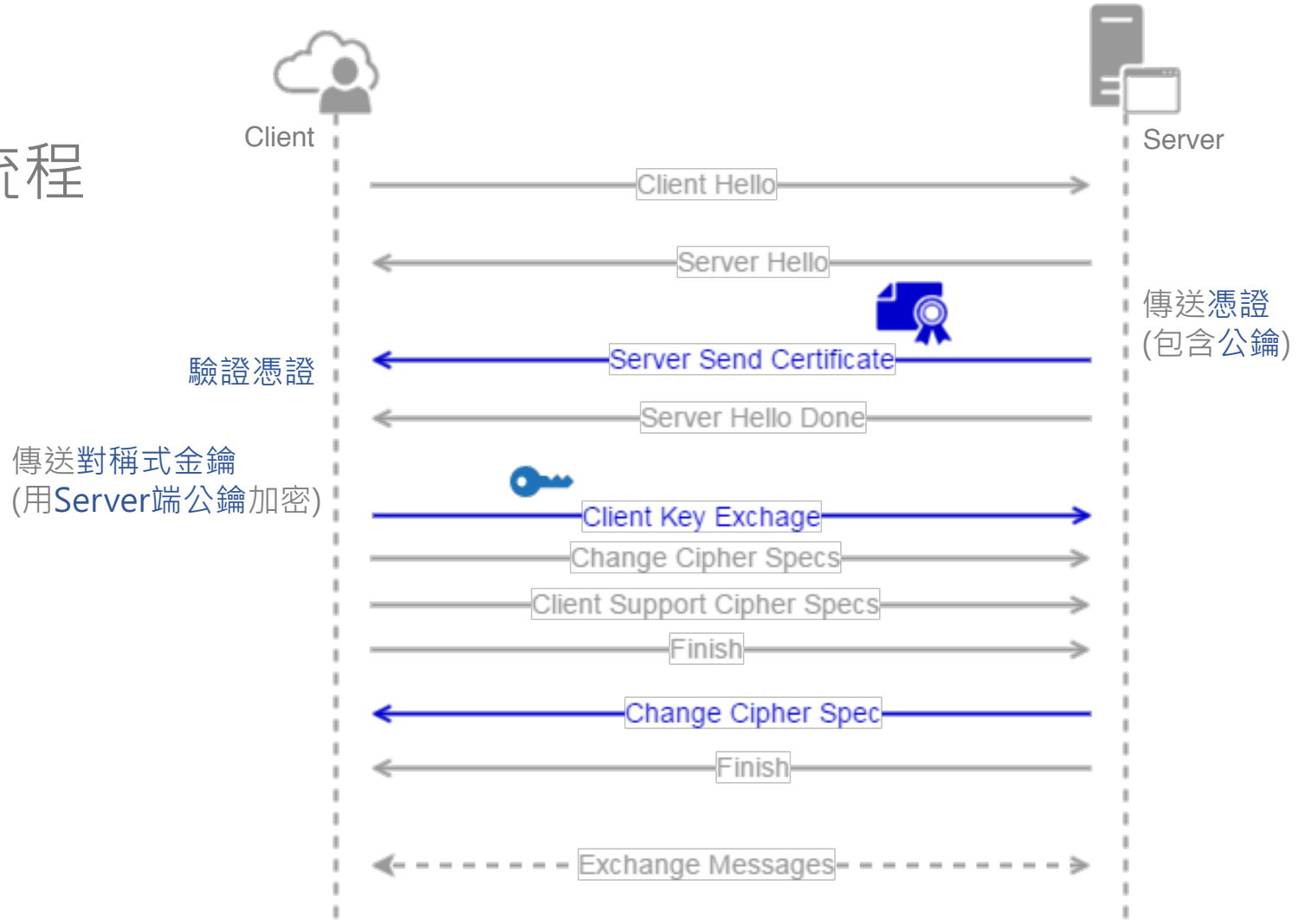
1 常見的分層安全性協定

<table>
<tr><td colspan="2">應用層 S/MIME、SET、PGP、HTTP、LDAP...</td><td>應用層</td></tr>
<tr><td colspan="2" rowspan="2"></td><td>表現層</td></tr>
<tr><td>會議層</td></tr>
<tr><td>SSL握手協定</td><td>SSL警告協定</td><td>.....</td></tr>
<tr><td colspan="3">SSL 紀錄協定</td></tr>
<tr><td colspan="2">傳輸層 TCP/UDP</td><td>傳輸層</td></tr>
<tr><td colspan="2">網際層 IP、IPSec</td><td>網路層</td></tr>
<tr><td colspan="2" rowspan="2">網路介面層</td><td>資料連結層</td></tr>
<tr><td>實體層</td></tr>
</table>

1 傳輸加密

SSL/TLS單向憑證流程

- 單向身分認證
- 機密性
- 完整性
- 不可否認性



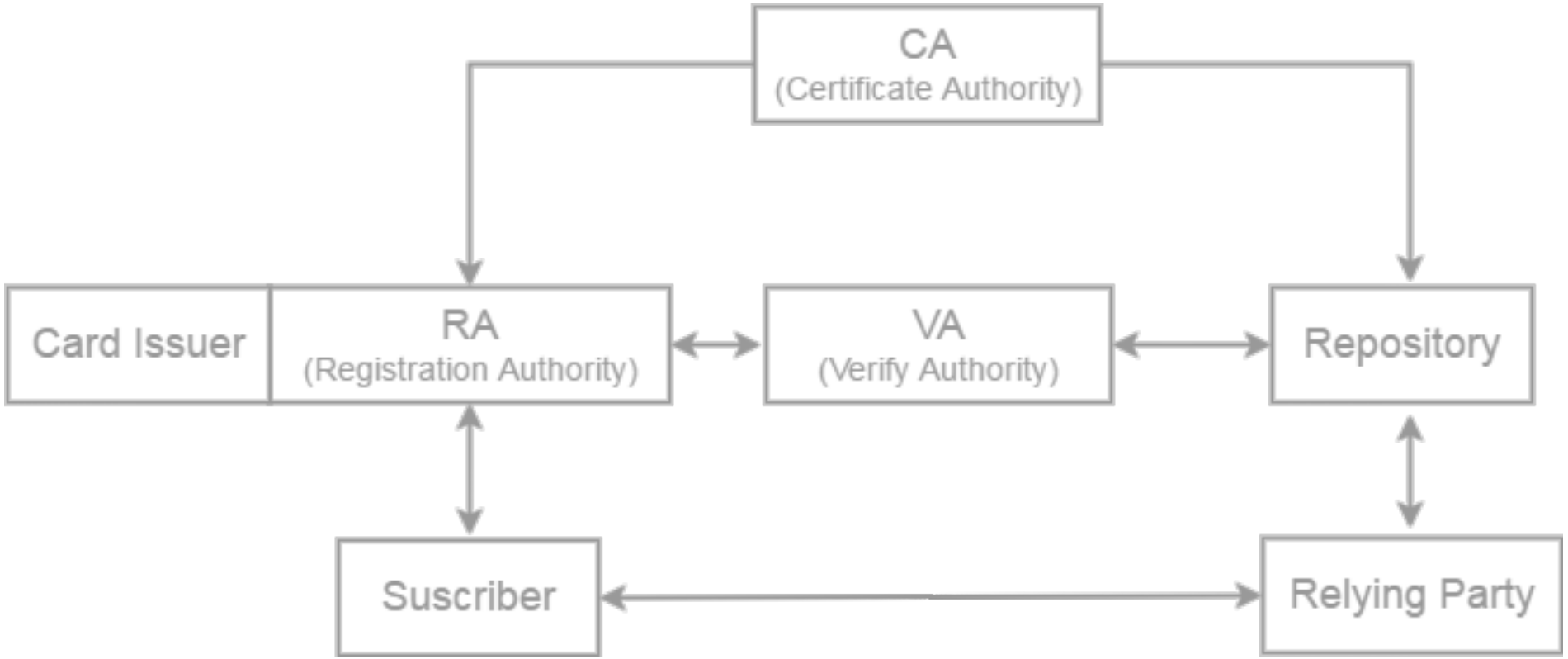
1 如何管理憑證

Trust Store：存放預設信任的憑證。
Key Store：存放本身的金鑰對(私鑰、公鑰、憑證)。

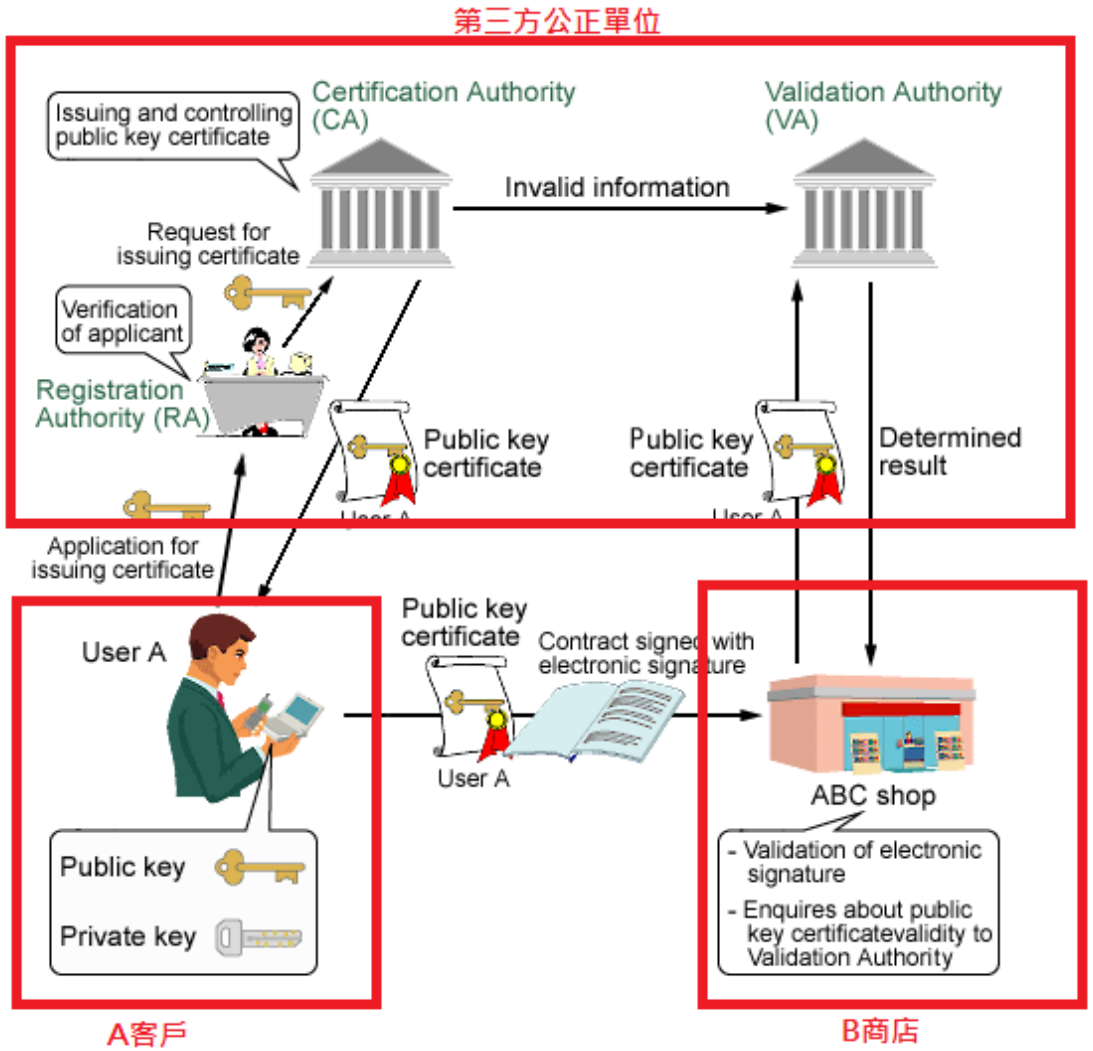
瀏覽器憑證鏈 (Certificate Chain)

1 公開金鑰基礎建設 (Public Key Infrastructure, PKI)管理非對稱式加密公鑰

憑證應用在PKI的角色組成圖



1 PKI交易流程


第三方公正單位
Certification Authority (CA)
Validation Authority (VA)
Issuing and controlling public key certificate
Invalid information
Request for issuing certificate
Verification of applicant
Registration Authority (RA)
Public key certificate
Public key certificate
Determined result
Application for issuing certificate
User A
Public key certificate
Contract signed with electronic signature
User A
ABC shop
Public key
Private key
- Validation of electronic signature
- Enquires about public key certificatevalidity to Validation Authority
A客戶
B商店

1 儲存加密

- Java加解密架構：Java Cryptography Architecture (JCA) 。
- Java標準版密碼學應用預設強度被限制，需下載相對應版本之Java Cryptography Extension (JCE) Unlimited Strength。

九大黃金安全準則

1 機密性(Confidentiality)
2 完整性(Integrity)
3 可用性(Availability)
4 身分認證(Authentication)
5 授權(Authorization)與存取控制(Access Control)
6 稽核(Auditing)與紀錄(Logging)
7 會談管理(Session Management)
8 錯誤與例外控制(Error and Exception Handling)
9 組態管理(Configuration Management)

2 對資料產生訊息摘要(Message Digest, MD)

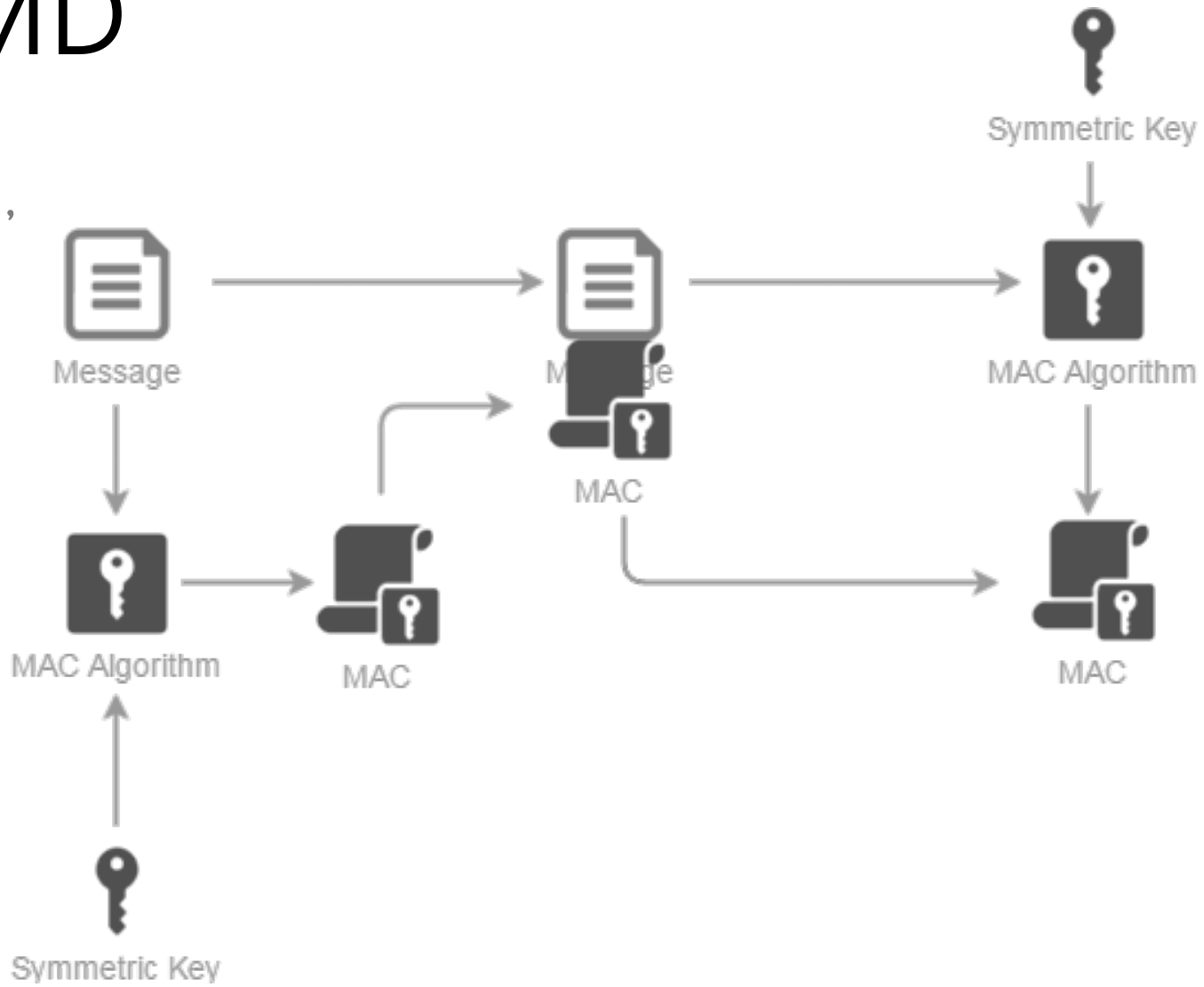
常見的方式有雜湊函數(Hash Function)和訊息驗證碼(Message Authentication Code, MAC)。

2 使用Hash Function做MD

同一演算法函式，原始訊息不變，則雜湊值相同(唯一)。
不可逆、雜湊碰撞機率低。
不建議使用MD5與SHA1演算法(Rainbow Table)，
盡量採用SHA-256或更高強度的演算法，並且加鹽雜湊。

2 使用MAC做MD

MAC與HASH不同在於，
訊息摘要產生及驗證
需要一把對稱式金鑰。



2 防範輸入資料變成命令的一部分

OWASP A1-注入攻擊(Injection)

- 改變組成 SQL 命令的方式(SQL Injection)。
- 黑/白名單過濾(OWASP ESAPI)。

OWASP A3-跨站腳本攻擊(Cross-Site Scripting, XSS)

- 黑/白名單過濾。

OWASP A10-未經驗證的重新導向(Un-validated Redirects and Forwards)

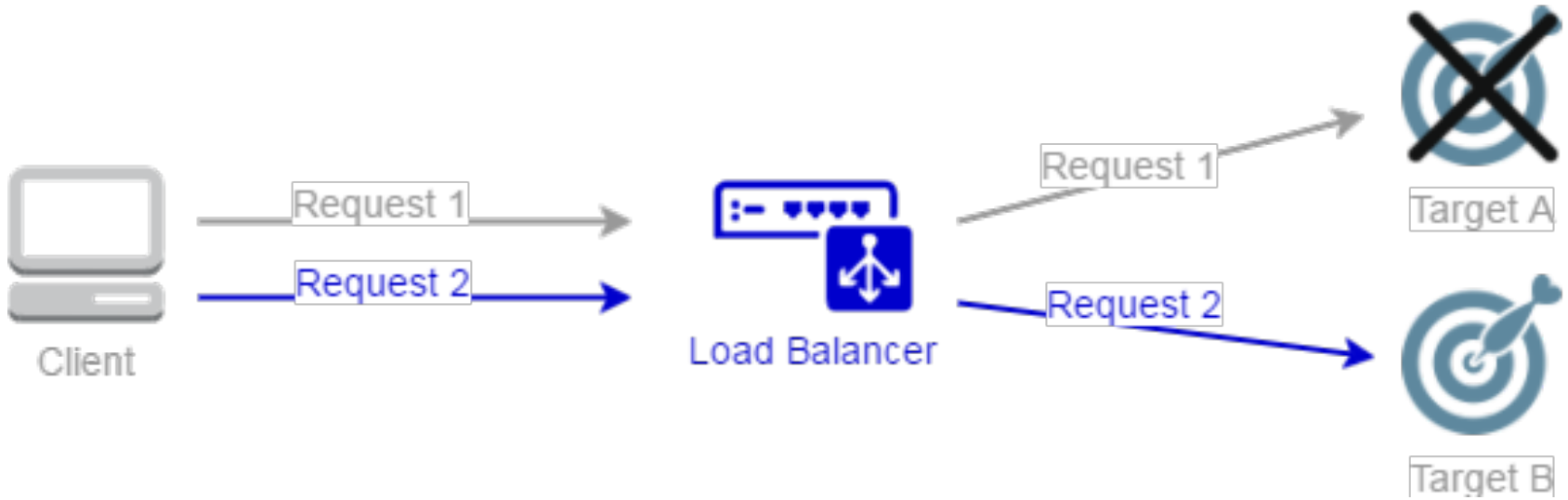
- 白名單(OWASP ESAPI - RandomAccessReferenceMap)。
- 隱藏導向網址。

九大黃金安全準則

1 機密性(Confidentiality)
2 完整性(Integrity)
3 可用性(Availability)
4 身分認證(Authentication)
5 授權(Authorization)與存取控制(Access Control)
6 稽核(Auditing)與紀錄(Logging)
7 會談管理(Session Management)
8 錯誤與例外控制(Error and Exception Handling)
9 組態管理(Configuration Management)

3 建立負載平衡(Load Balancing)

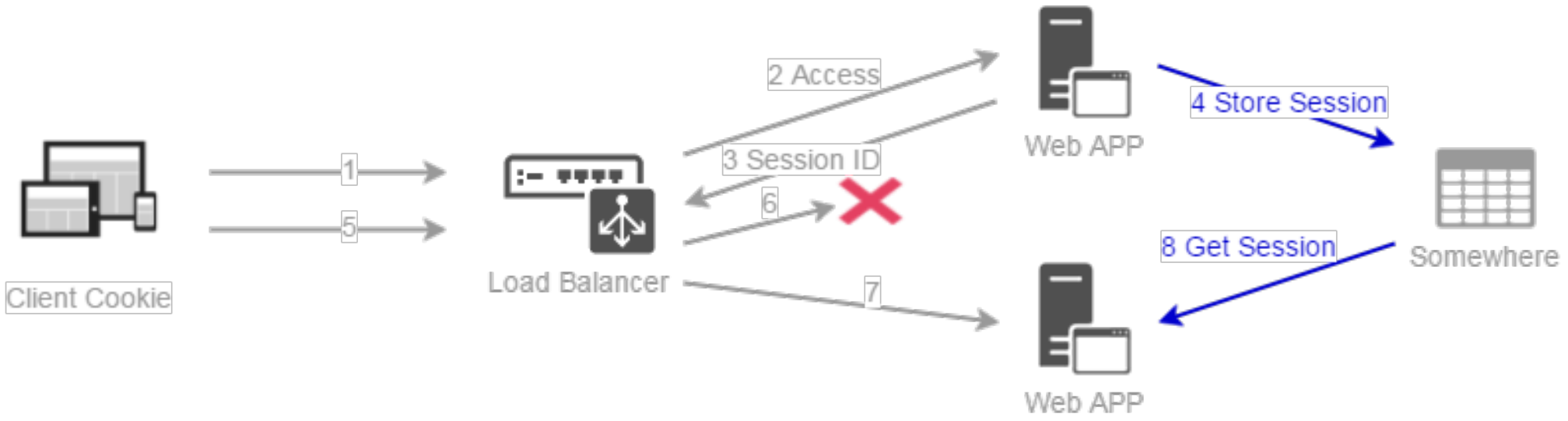
建立Load Balancing可以做到Session Stickiness、Heartbeat、Failover。



* 分散式的目標物件中的功能才可以做Load Balancing和Failover，例如JSP & Servlet、DB Object、EJB Object、JMS、JNDI Object、Web Service...。

3 建立叢集(Cluster)架構

建立Cluster可以達到系統可擴展性(Scalability)、高可用性(High Availability)、容錯(Fault Tolerance)。



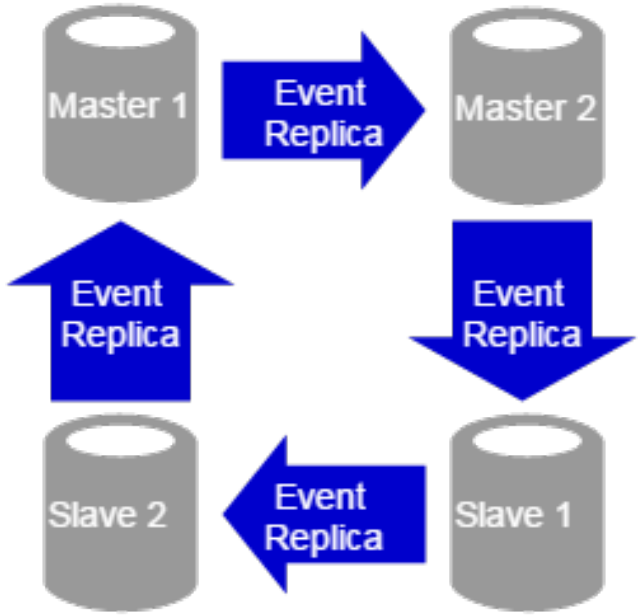
* 供應商實作HTTP Session failover各有不同，以下幾點區分實作方法，如何備份Session狀態?備份的頻率(frequency)跟粒度(granularity)為何?

3 建立資料庫備份機制

參考每家供應商做的DB HA機制，Oracle、SQL Server...。

Chaining Replication

Circular Chaining Replication



3 使用雲端服務平台

容易動態擴充，提高網站可用性。例如Amazon EC2、Azure。

九大黃金安全準則

1 機密性(Confidentiality)
2 完整性(Integrity)
3 可用性(Availability)
4 身分認證(Authentication)
5 授權(Authorization)與存取控制(Access Control)
6 稽核(Auditing)與紀錄(Logging)
7 會談管理(Session Management)
8 錯誤與例外控制(Error and Exception Handling)
9 組態管理(Configuration Management)

4 採用多重機制驗證

綜合有多種方式確認使用者身分識別。

例如帳號密碼(密碼加鹽雜湊)、憑證、One Time password(OTP)..。

4 建立帳戶鎖定機制

防止暴力破解法(Brute Force)、字典攻擊法(Dictionary Attack)。

帳戶鎖定機制基本上要能夠做到，

- 紀錄已錯誤次數及上次成功登入時間。
- 設定可錯誤次數及鎖定時間(或人為解鎖)。
- 帳戶登入超過錯誤次數時，進行鎖定。
- 鎖定對象限制帳號或來源。

4 重要交易要求再次進行身分驗證

許多網站會使用密碼加上OTP再次確認使用者身分。

4 使用全自動區分電腦和人類的公開圖靈測試(CAPTCHA)

防止自動化程式。

有許多線上資源，例如Google 的reCAPTCHA。

4 建立定期更換密碼與重設密碼機制

降低暴力破解法和猜測密碼風險。

密碼原則通常包含，

- 密碼可使用期間。
- 密碼複雜度及長度。
- 限制密碼使用歷程。

重設密碼機制應採用其他管道(例如電子信箱)重新驗證身分後，在一定時間內(Token機制)讓使用者重新設定密碼。

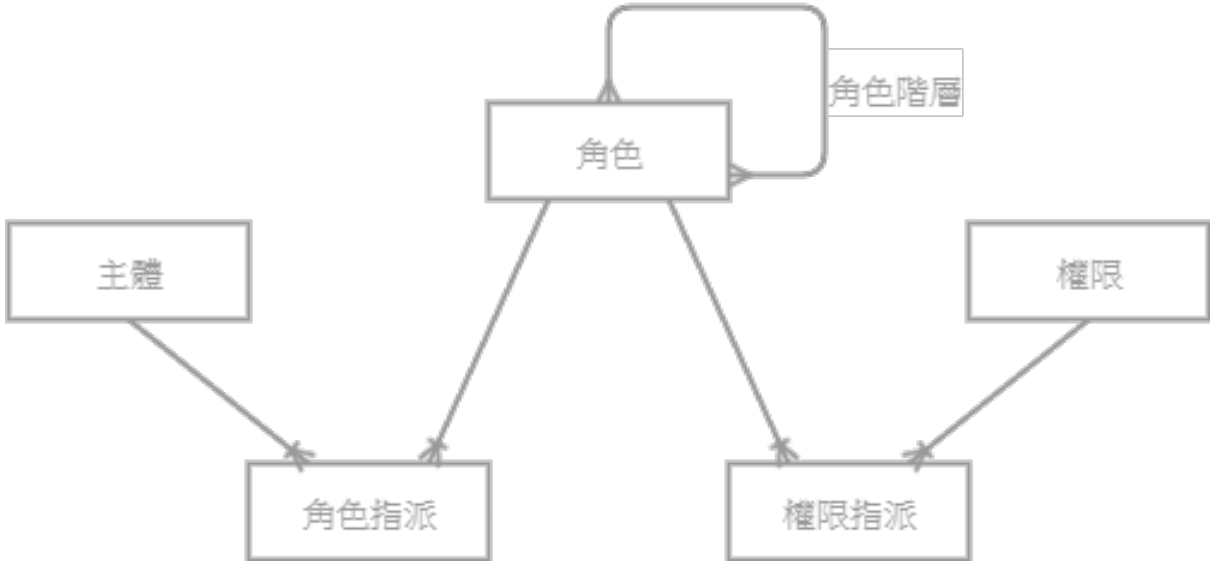
4 網站連線來源為固定對象時，可建立雙向驗證

九大黃金安全準則

1 機密性(Confidentiality)
2 完整性(Integrity)
3 可用性(Availability)
4 身分認證(Authentication)
5 授權(Authorization)與存取控制(Access Control)
6 稽核(Auditing)與紀錄(Logging)
7 會談管理(Session Management)
8 錯誤與例外控制(Error and Exception Handling)
9 組態管理(Configuration Management)

5 建立Role-Based Access Control, RBAC

採取於伺服器驗證且集中控管授權。



5 RBAC角色階層建立更新

找出某一子節點的所有父節點，

左鍵值小於查詢節點左鍵值，

且右節點大於查詢節點右鍵值

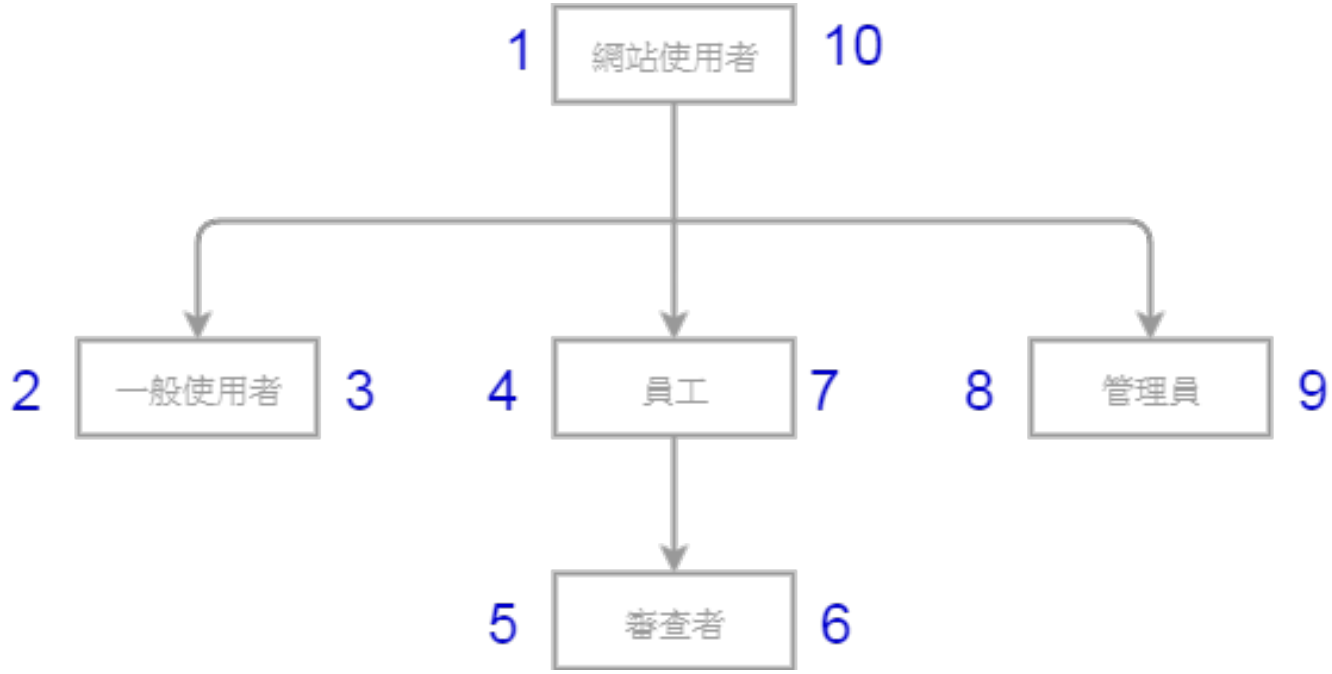
的所有節點

找出某一父節點的所有子節點，

左鍵值大於查詢節點左鍵值，

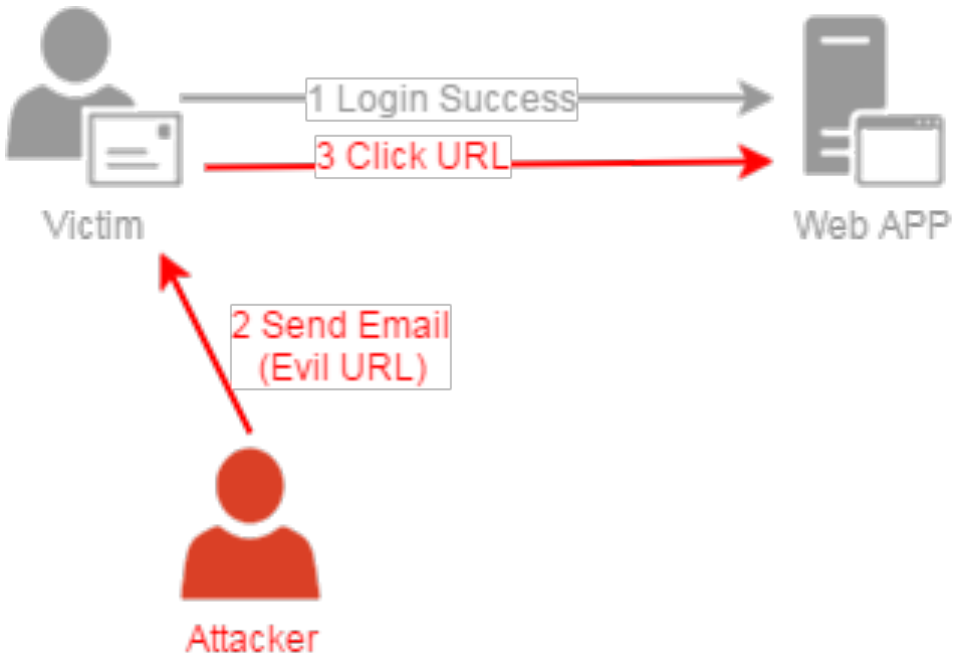
且右節點小於查詢節點右鍵值

的所有節點



5 防範跨站偽造請求 (Cross-Site Forgery, CSRF)

動態產生Synchronizer Token Pattern(By Session or Page)、重新授權 、CAPTCHA。



九大黃金安全準則

1 機密性(Confidentiality)
2 完整性(Integrity)
3 可用性(Availability)
4 身分認證(Authentication)
5 授權(Authorization)與存取控制(Access Control)
6 稽核(Auditing)與紀錄(Logging)
7 會談管理(Session Management)
8 錯誤與例外控制(Error and Exception Handling)
9 組態管理(Configuration Management)

6 紀錄內容應排除所有個資項目

記錄需要包含以下資訊，

- 時間
- 身分識別
- 地點(通常為IP位置)
- 物件(存取甚麼資源物件)
- 錯誤代碼
- 優先權
- 行為

記錄需要考慮保留時間、關聯性、效能。

6 紀錄輸出方案-Log4J

設定

- 程式碼範圍
- 紀錄層級：TRACE、DEBUG、INFO、WARN、ERROR。
- 輸出對象：主控台、檔案、資料庫、郵件傳輸、Syslog協定。
- 輸出格式

Syslog協議可以將紀錄資料儲存在遠端伺服器中，
有些作業系統有提供Syslog服務，例如CentOS內建的RSYSLOG。

九大黃金安全準則

1 機密性(Confidentiality)
2 完整性(Integrity)
3 可用性(Availability)
4 身分認證(Authentication)
5 授權(Authorization)與存取控制(Access Control)
6 稽核(Auditing)與紀錄(Logging)
7 會談管理(Session Management)
8 錯誤與例外控制(Error and Exception Handling)
9 組態管理(Configuration Management)

7 一般防護Session機制

- 不採用Session ID預設名稱。
- Session ID長度及安全產生方式，儘量使用框架提供的內建Session ID產生管理方式。
- Session Cookie的Domain與Path，預設使用網站應用程式路徑。
- Session Cookie的Max-age，預設在Session失效或是客戶端關閉瀏覽器時，Session Cookie無法再次使用。

7 防護會談劫奪(Session Hijacking)

使用以下機制防範Session Hijacking，

- Cookie存放Session ID。
- 使用Session Timeout自動登出機制。
- 不將Session ID顯示於URL。
- 設定Secure Flag強制Cookie必須使用SSL(HTTPS)傳送。

7 防護置換會談ID(Session Fixation)

於身分認證成功後，置換Session ID以防範Session Fixation(自動化Session Fixation保護機制)。

九大黃金安全準則

1 機密性(Confidentiality)
2 完整性(Integrity)
3 可用性(Availability)
4 身分認證(Authentication)
5 授權(Authorization)與存取控制(Access Control)
6 稽核(Auditing)與紀錄(Logging)
7 會談管理(Session Management)
8 錯誤與例外控制(Error and Exception Handling)
9 組態管理(Configuration Management)

8 錯誤與例外控制安全措施

- 所有功能都進行錯誤與例外處理。
- 防止直接顯示錯誤例外訊息。
- 紀錄詳細資訊，顯示錯誤代碼。
- 考慮是否再次拋出錯誤。
- 一致Log機制並區分嚴重等級。
- 避免「安靜無聲」錯誤，嚴重錯誤採用通知機制、正確釋放資源。

九大黃金安全準則

1 機密性(Confidentiality)
2 完整性(Integrity)
3 可用性(Availability)
4 身分認證(Authentication)
5 授權(Authorization)與存取控制(Access Control)
6 稽核(Auditing)與紀錄(Logging)
7 會談管理(Session Management)
8 錯誤與例外控制(Error and Exception Handling)
9 組態管理(Configuration Management)

9 良好的組態設定

避免使用以下設定，

- 預設安裝功能/錯誤頁面。
- 過高的系統權限。
- 網站目錄瀏覽。
- 未記錄網站存取行為。

完整的組態管理，

- 整理記錄軟體專案中使用到的元件及底層軟體版本資訊。
- 參考安全警訊來源，例如CVE、CVE Details、US-CERT Alerts、Microsoft Security Alert、Bugtraq。
- 使用安全性檢測工具，例如OWASP提供Dependency-Check工具。

了解九大安全黃金準則後，
接下來將準則帶入到軟體安全實務中!

1 機密性(Confidentiality)
2 完整性(Integrity)
3 可用性(Availability)
4 身分認證(Authentication)
5 授權(Authorization)與存取控制(Access Control)
6 稽核(Auditing)與紀錄(Logging)
7 會談管理(Session Management)
8 錯誤與例外控制(Error and Exception Handling)
9 組態管理(Configuration Management)

安全軟體生命週期(SSDLC)發展階段實務

1 需求階段

2 設計階段

3 測試階段

4 部屬與維運階段

1 需求階段實務

在需求訪談及工作會議中，可以利用資料分級、誤用案例、問券與查檢表提出概念化的安全需求。

* 問券與查檢表(Checklist)可以參考，
  - SANS組織提供Securing Web Application Technologies(SWAT)
  - OWASP組織提供Application Security Verification Standard(ASVS)。

設計階段活動的回饋，

- 建立威脅建模
- 安全風險分析
- 安全設計審查，使用追溯矩陣(Traceability Matrix)/問題追蹤管理系統(Issue Tracking System)，確認需求沒被遺漏。

安全軟體生命週期(SSDLC)發展階段實務

1 需求階段

2 設計階段

3 測試階段

4 部屬與維運階段

2 設計階段實務

安全設計法則

- 縱身防禦 (Defense in Depth)。
- 防範安點失效 (Single Point of Failure)。
- 保持簡單愚蠢 (KISS)，使用模組化設計簡化系統架構。
- 開放設計 (Open Design)。
- 使用現存元件 (Leveraging Existing Components)。
- 安全故障 (Fail Secure)。
- 完全仲裁 (Complete Mediation)，應在重要交易再次要求使用者身分驗證。
- 職責分離 (Separation of Duties)。
- 最少權限(Least Privilege)。

架構風險分析

- 抗攻擊能力分析 (Attack Resistance Analysis)，使用已知攻擊清單比對架構圖終節點或連線。
- 模糊分析 (Ambiguity Analysis)，釐清系統中模糊不清或有爭議的部分可能的安全問題。
- 底層框架弱點分析 (Underlying Framework Weakness Analysis)。

安全設計審查，

蒐集相關文件(需求與設計階段產出文件)，分析安全設方式，進一步修正需求與設計。

安全軟體生命週期(SSDLC)發展階段實務

1 需求階段
2 設計階段
3 測試階段
4 部屬與維運階段

3 測試階段實務

安全性測試，使用持續(CI&CD)靜態源碼掃描及Code Review，配合動態漏洞掃描和滲透測試，驗證安全需求。

漏洞掃描工具可以分為網路型、主機型及針對特定應用，常見網站漏洞掃描工具有，Nessus、IBM AppScan、HP WebInspect、Acunetix Web Vulnerability Scanner、Nikto。

安全軟體生命週期(SSDLC)發展階段實務

1 需求階段
2 設計階段
3 測試階段
4 部屬與維運階段

4 部屬與維運階段實務

進行攻擊面分析。

評估軟體變更管理對安全性的影響，可以針對變更範圍、安全機制相關、設計變更、機敏資料、安全開發實務、安全測試、部屬環境進行考量

將安全準則及SSDLC階段實務導入下一個專案，提升軟體安全品質!

1 需求階段

2 設計階段

3 測試階段

4 部屬與維運階段