網站安全概念

1. 網站安全的基本原則
2. 安全開發生命週期
3. 安全網站開發要點

網站安全的基本原則

資訊安全的三大支柱：機密性、完整性和可用性

1. **機密性**：確保敏感資訊不被未經授權的個人、組織或系統存取、泄露或竊取。例如：加密、訪問控制、密碼保護等措施。
2. **完整性**：確保資訊在儲存、傳輸或處理過程中不被竄改、刪除或破壞。例如：數位簽章、校驗碼、資料備份和恢復、版本控制等措施。
3. **可用性**：確保資訊和系統在需要時可供使用，且能夠正常運作。例如：備援、災難恢復、負載平衡等措施。
4. **機密性：不被濫用，完整性：不被竄改，可用性：可供使用。**

安全開發生命周期（SDL）

用系統化的方法將安全性融入軟體開發過程的每個階段

1. 教育和訓練：確保開發人員了解安全性和 SDL 的重要性。例：要求所有開發人員參加安全性訓練課程，學習如何編寫安全的程式碼和如何識別安全漏洞。
2. 需求和設計：在需求和設計階段，將安全性納入考量，包括風險評估和安全目標。例：在設計一個新的網路應用程式時，開發團隊需要考慮安全性需求，例如如何保護用戶的敏感資料，如何防止 XSS 與 SQL 注入攻擊等。

安全開發生命周期

3. 實作：在實作階段，實施安全編碼實踐和安全措施，包括錯誤處理和輸入驗證。例：開發人員在編寫程式碼時，需要遵循安全編碼實踐，例如使用安全的函式庫，避免使用已知的安全漏洞的函式庫，例如 strcpy 與 strncpy（strncpy 則提供了更多控制，避免了緩衝區溢出）。
4. 驗證：在驗證階段，執行安全測試和弱點掃描，以識別和修復安全漏洞。例：開發團隊需要執行安全測試和弱點掃描，以識別和修復安全漏洞，例如使用工具如 `SQLMap`、`Burp Suite` 或 `OWASP ZAP` 來識別網路應用程式的安全漏洞。

安全開發生命周期

5. 發布：在發布階段，確保軟體的安全配置和安全更新機制。例：在發布軟體時，需要確保軟體的安全配置，例如設定正確的權限，啟用安全功能等。
6. 維護：在維護階段，持續監控和更新軟體的安全性，包括修復安全漏洞和應對安全事件。

這些步驟可以幫助開發人員在整個開發生命周期中關注安全性，從而減少安全漏洞和風險。

安全網站開發要點

輸入與輸出安全

1. **使用輸入資料過濾：** 過濾所有使用者輸入的資料，輸出資料時進行適當的清理，以防止 SQL 注入、XSS 攻擊等。
2. **安全的錯誤處理：** 進行安全的錯誤處理，以防止攻擊者利用錯誤信息進行攻擊。

安全網站開發要點

身份驗證與授權

3. **使用驗證和授權：** 確保所有使用者訪問與操作都經過 驗證 和 授權 ，未經授權的資料或操作將被拒絕。
4. **使用安全的密碼儲存：** 避免使用明碼儲存密碼，並使用安全的加密方式，以防止密碼被竊取。
5. **防止 CSRF 攻擊：** 使用 Token 或其他方法防止 CSRF 攻擊，以防止用戶被迫執行未經授權的動作。

安全網站開發要點

資料傳輸與儲存安全

6. **使用 HTTPS：** 使用 HTTPS 協議傳輸資料，以防止資料被竊取和竄改。
7. **設定安全的 Cookie：** 設定安全的 Cookie，例如 HttpOnly、Secure 等，以防止 Cookie 被竊取和竄改。
8. **使用環境變數儲存機敏資料：** 使用環境變數儲存機敏資料，以防止機敏資料被直接傳輸至原始碼。
9. **使用安全的資料庫：** 使用並安全配置資料庫，以防止資料被竊取和竄改。

安全網站開發要點

資料傳輸與儲存安全

10. **設定安全的檔案權限：** 設定安全的檔案權限，以防止攻擊者利用檔案權限進行攻擊。
11. **控制檔案上傳目錄權限：** 僅允許必要的權限給檔案上傳目錄（讀、寫、禁止執行），避免執行檔案上傳後可被執行，並限制可上傳的檔案類型，以防止惡意檔案上傳與執行。

安全網站開發要點

資料傳輸與儲存安全

12. **限制上傳的檔案被以網址的方式讀取：** 限制上傳的檔案不能直接透過網址存取。
13. **使用檔案流或 URL Rewrite 輸出檔案：** 避免未經授權的檔案直接透過連結下載，以防止檔案被竊取和竄改。

安全網站開發要點

安全防護與監控

14. **使用 Web 應用防火牆 (WAF)**： 使用 WAF 以防止攻擊者利用已知漏洞進行攻擊。
15. **使用安全的第三方庫**： 使用安全的第三方庫和框架，以防止漏洞和攻擊。
16. **使用安全的網路協議**： 使用安全的網路協議，如 TLS 等，以防止資料被竊取和竄改。

安全網站開發要點

安全防護與監控

17. **監控和記錄：** 監控和記錄網站的安全事件，以便及時發現和處理安全問題。
18. **定期進行安全審計：** 定期進行安全審計，以便及時發現和處理安全問題。
19. **更新和修補漏洞：** 定期更新和修補漏洞，以防止攻擊者利用已知漏洞進行攻擊。

安全網站開發要點

負載安全

20. **防止阻斷服務 (DoS) 和分散式阻斷服務 (DDoS) 攻擊：** 實施速率限制、流量限制、WAF等機制，以防止惡意流量淹沒伺服器，導致服務中斷。**例：** 設定每個 IP 位址在特定時間內的請求次數限制，部署能夠識別和過濾惡意流量的 WAF。

21. **資源限制：** 對伺服器的資源使用（例如 CPU、記憶體、網路連線數）設定合理的限制，防止單一請求或惡意行為耗盡系統資源，影響其他正常用戶。**例：** 在伺服器配置中設定每個使用者會話的最大記憶體使用量，限制同時開啟的資料庫連線數、Timeout等。

安全網站開發要點

負載安全

22. **處理大量並發請求的安全考量：** 在設計系統架構時，考慮如何安全地處理大量並發請求，避免因負載過高而導致安全漏洞被觸發或利用。**例：** 使用安全的負載平衡器來分散流量，確保後端伺服器不會過載，並仔細評估在高負載情況下可能出現的競爭條件 (race condition) 等安全風險。

23. **API 速率限制與配額管理：** 對公開 API 設定合理的速率限制和配額，防止 API 被濫用或惡意攻擊，影響後端服務的穩定性和安全性。**例：** 限制每個 API 金鑰在每分鐘或每天可以發出的請求次數，並實施配額管理以防止單一用戶過度消耗資源。

安全網站開發要點

負載安全

24. **非同步處理與佇列：** 對於可能導致長時間阻塞的操作，採用非同步處理和消息佇列等機制，避免這些操作直接佔用請求處理線程，提高系統在高負載下的可用性和安全性。**例：** 將檔案上傳、複雜的資料處理等操作放入消息佇列中異步處理，避免阻塞 Web 伺服器的請求處理。

這些是網站開發中一些重要的安全要點，需要注意的是，每個網站的安全需求都不同，需要根據自己的網站進行安全評估和加強。