

網上應用系統保安

2008 年 2 月

© 香港特別行政區政府

這份文件的內容為香港特別行政區政府的財產，未經香港特別行政區政府的明文批准，不得轉載全部或部份文件內容。

免責聲明：政府盡力確保本文資料準確，但不明示或隱含保證資料準確無誤。對於因使用本文資料而引致或與之有關的任何錯誤或遺漏，香港特別行政區政府概不承擔任何法律責任。

目錄

摘要.....	2
I. 管理.....	3
帶動改變的技術與威脅發展.....	3
行政控制.....	3
技術控制.....	4
網上應用系統的保安指引.....	4
II. 資訊科技從業人員.....	5
網上應用系統的一般保安漏洞.....	5
網上應用系統保安的提示.....	7
網上應用系統的驗收清單.....	12
附錄一：瀏覽互聯網時如何保護自己.....	14
附錄二：如何消除程式碼的十大關鍵保安漏洞.....	15

摘要

網絡技術的發展加上企業環境的改變，意味著現今網上應用系統在公司、公共與政府服務中變得更加普遍。雖然網上應用系統可帶來便利和提升效率，但也產生新的保安威脅，若未作好妥善處理，對機構中的資訊科技架構可能會造成顯著的潛在風險。

過去十幾年，機構依賴網絡外圍設備的保安措施，以保護其資訊科技的基礎架構。然而，傳統網絡保安措施與技術已不足以保障網上應用系統和避免新的威脅，尤其是現在的攻擊瞄準了網上應用系統設計上的保安漏洞。因此，在發展應用系統時需同時考慮技術和管理兩者的保安。

若要克服這些新的保安威脅，必須瞭解常見的網上應用系統保安漏洞。本文討論關鍵性的網上應用系統保安漏洞，以及在系統發展周期的不同階段中，應如何處理這些保安漏洞。因為互聯網瀏覽者往往是網上應用系統資訊保安上最弱的一環，所以本文也會給他們有關如何安全地瀏覽互聯網的提示。

I. 管理

帶動改變的技術與威脅發展

網絡技術的發展加上企業環境的改變，使網上應用系統在公司、公共與政府服務中變得更加普遍，雖然網上應用系統可帶來便利和提升效率，但也產生新的保安威脅。如未作好妥善處理，對機構中的資訊科技架構可能會造成顯著的潛在風險。

網上應用系統的快速增長，使分散式的資訊科技基礎架構更加複雜，並且更難防衛。過去十幾年，機構依賴網絡外圍設備，如防火牆的保安措施來保護其資訊科技基礎架構。然而，現今越來越多攻擊是瞄準網上應用系統設計上的保安漏洞，例如植入式弱點漏洞，這種威脅有可能使傳統的網絡保安不足以保護網上應用系統。

這些威脅起源於不可信任的客戶接達點、無對話的規約（*session-less protocols*）、複雜的網絡技術以及不安全的網絡層。在網上應用系統中，應用系統通常不能控制客戶軟件，因此不能完全信任並直接處理來自客戶軟件的輸入。攻擊者會編造身份使其看起來像合法的客戶，複製使用者的身份或建立欺騙訊息與 *cookies*。此外超文件傳輸規約是一個無對話的規約，因此容易受到重複與插入弱點（*Injection flaws*）攻擊，超文件傳輸規約的訊息很容易被修改、仿冒與採取。

因此，機構必須了解和完全察覺潛在的威脅，方能實施適當的防衛策略。為了加強保護重要設施，網上應用系統的部署也需要技術上和行政上的額外保安控制。

行政控制

為了強化網上應用系統的保安與協助數據處理的保護，以下是行政控制的建議。

1. 為發展與維護網站和/或網上應用系統提供一個方向並列入關鍵的指引，例如由香港政府編製的一系列關於使用政府網站發布資訊的指引。
2. 把網上應用系統編碼與發展的作業實務列入關鍵的指引。軟件發展小組應遵守一系列網上應用系統編碼的安全作業實務，以抵抗一般的網上應用系統保安漏洞。
3. 收集和管理敏感資訊及使用者數據時，要符合政策與法規。
4. 編製保安及品質保證計劃，並採用品質保證方法，如重新檢查原始碼、滲透測試、用戶驗收測試等等。
5. 在網上應用系統推出前或系統作出任何重大更改或升級後，進行完整的資訊科

技保安審計。

技術控制

本文“網上應用系統保安的提示”一節可用作網上應用系統的詳細及重要保安措施參考之用。

網上應用系統的保安指引

為了改善網上應用系統的保安，一個開放且可供自由接達，名為 **Open Web Application Security Project (OWASP)**¹ 的社群已建立，並協調全世界的力量來減低網上應用系統軟件相關的保安風險。

一些主要的機構與政府部門針對網上應用系統的開放特性所帶來的管理風險，投入資源來制定策略、政策和指引。為了確保網上應用系統保安達到最低標準上的保障，某些機構已經制定了清單來評估網上應用系統在推出前保安之全面性。為了這個目的，美國國防部已制定了他們的應用系統保安清單²。

香港政府資訊科技總監辦公室 (OGCIO) 也發布了一系列保安政策與指引，作為政府部門和機構的參考文件，這些文件已經被發布在以下的 OGCIIO 網站:

<http://www.ogcio.gov.hk/chi/prodev/csecpol.htm>

「政府資訊科技總監辦公室 (OGCIO) 資訊科技保安指引」中第 10.1.1 節「應用系統設計及發展的保安考慮事項」，描述了一般應用系統設計與發展所需之保安考慮和原則。這些保安考慮事項也可應用在發展網上應用系統上。因為網上應用系統經常受到額外的保安威脅，指引中第 10.7 節「網上應用系統保安」也描述到軟件發展小組應遵循一套網上應用系統安全編碼作業實務，以便防禦一般網上應用系統的保安漏洞攻擊。

¹ <http://www.owasp.org/>

² <http://iase.disa.mil/stigs/checklist/app-security-checklist-v2r19-24Nov06.doc>

II. 資訊科技從業人員

這一節主要描述為相關程式與系統提升可靠性與保安性上的技術觀點。

網上應用系統的一般保安漏洞

OWSAP 是一個遍佈世界的自願參加者社群，目標是讓大眾知道網上應用系統的保安問題。人們和機構可以根據應用系統保安風險³的資訊作出決策。OWASP 在文件中列出了最關鍵的網上應用系統保安漏洞，這文件的標題為 “*The Ten Most Critical Web Application Security Vulnerabilities 2007 Update*”⁴。

1. 跨網址程式編程攻擊 (Cross Site Scripting, XSS)

跨網址程式編程攻擊 (XSS) 的潛在威脅使受害者的瀏覽器執行手稿程式 (Script)，導致劫持用戶對話、竄改網站並可能植入蠕蟲等等的保安威脅。由於應用系統讀入了沒有經過驗證或加密的數據，並將它傳給瀏覽器，從而造成這些保安漏洞。

2. 植入式保安漏洞

此保安漏洞的潛在威脅使攻擊者可透過非預期的命令或改變系統數據，以欺騙應用系統。在植入式保安漏洞中，以網上應用系統的 SQL 插入漏洞最為常見。當使用者提供的數據被傳送到詮釋器作為命令或查詢時，就可能構成植入式攻擊。

3. 惡意程式執行

受到遠程文件包含 (Remote File Inclusion, RFI) 破壞的程式碼，其潛在威脅可讓攻擊者有機會引入惡性程式碼與數據，導致毀滅性的攻擊如入侵整個伺服器。執行惡意程式會影響到 PHP、XML 以及任何程式架構，因它們可接收使用者傳送的檔名或檔案。

³ http://www.owasp.org/index.php/Main_Page

⁴ https://www.owasp.org/index.php/Top_10_2007#Summary

4. 不安全的直接物件參照

這個潛在威脅使攻擊者不需要透過授權便可以操作接達其它物件的參照。當開發者展示內部完成物件的參照時，例如檔案、目錄、數據庫記錄或鍵值作為劃一資源定位（URL）或表單文件的參數，直接物件參照便會發生。

5. 跨網站請求偽造（Cross Site Request Forgery, CSRF）

這保安漏洞的潛在威脅強迫登入者的瀏覽器向具有保安漏洞的網上應用系統寄發預先驗證的要求，攻擊者便可利用使用者的瀏覽器執行惡意行為而獲利。CSRF的威力跟被攻擊的網上應用系統一樣。

6. 資料洩漏和不適當誤差處理

這個保安漏洞的潛在威脅使攻擊者可以利用這個弱點竊取敏感數據，或導致更多嚴重的攻擊。應用系統存在的問題很多，一不小心便會洩露有關內部組態的資訊、工作或違反私隱權。

7. 身份驗證功能和對話管理的缺失

此點的潛在威脅使攻擊者可組合密碼、密碼匙或授權的權標，以假裝成其他使用者的身份。當帳戶憑證與交談的權標沒有受到合適的保護時，便會造成這個保安漏洞。

8. 不安全的加密儲存設備

這個潛在的威脅在於當攻擊者使用未經妥善保護的數據來進行身份盜竊与其它犯罪行為，例如偽造信用卡。因網上應用系統沒有做好加密功能來保護數據和憑證，所以便造成這個漏洞。

9. 未加密的網絡通訊

當網絡通訊架構傳輸數據時，這個保安漏洞可能洩漏敏感性資料。原因是當需要保護傳輸敏感數據的網絡通訊時，該網絡的傳輸並沒有加密。

10. 無權限的劃一資源定位（URL）接達

這一個保安漏洞使攻擊者透過直接接達 URL 而有機會進行未授權的操作。當應

用程式避免顯示連結或 URL 給未授權的使用者時，應用程式只保護某部份的敏感功能，因而形成這個保安漏洞。

系統的開發者應該注意這些一般性的保安漏洞，並制定程式開發的標準，避免這樣的問題在編碼階段發生。透過 OWASP 的指導以建立網上應用系統保安是一個不錯的參考⁵。

網上應用系統保安的提示

如同本文上一節所提，部署網上應用系統帶來好處亦帶來了新的保安風險。為了有效地處理這些風險，在整個發展周期中，應該考慮多樣化的保安控制。為了幫助了解所建議的保安控制與生命周期中可能相關的地方，本節將透過生命周期的各階段指出需要特別注意的保安考慮。

需求階段

在本階段中，應用系統發展小組應該收集和項目有關的部門系統需求與保安規格。通過系統需求，發展小組便知道這應用系統的主要目的概覽，該做的及不該做的都包含在內。此資訊將協助發展小組制訂應用系統關鍵的保安控制。

此外，一般性的保安控制或機制是需要加入應用系統中以遵守規例或需求，例如 Payment Card Industry Data Security Standard (PCI DSS) 的第 6 個需求，標題為“Develop and Maintain Secure Systems and Applications”，焦點在於建立控制以減低系統和軟件中現存的保安漏洞，並詳細指出軟件開發的保安以及避免受攻擊的需求。

在設計、發展和測試階段中，正確地建立系統和使用者的保安需求是非常重要的，這將增加網上應用系統全面性的保安，並確保使用者對結果滿意。

設計階段

設計階段所牽涉的不只要求應用系統的設計須和第一階段所提的規格一致，還須制訂安全的編碼標準、執行 threat modelling 並發展應用系統的保安架構。

制訂安全的編碼標準

安全的編碼標準是告訴開發人員如何去撰寫應用系統的程式碼、開發保安程式碼時的指引，對識別高風險區域、數據輸入與其它界面和誤差處理等作出適當的評論。不同的機

⁵ http://www.owasp.org/index.php/OWASP_Guide_Project

構包括 OWASP 和電腦緊急應變小組（Computer Emergency Response Team, 或 CERT）⁶ 建議一些安全的編碼作業實務。政府資訊科技總監辦公室（Office of the Government Chief Information Officer; OGCIO）在其資訊科技保安指引中也包含了一般性的安全編碼作業實務。

1. 驗證所有輸入參數的有效性，以防止 SQL 插入攻擊及跨網址程式編程攻擊：

程式設計師應發展一個集中模組，以驗證輸入參數，並根據允許輸入參數類型的格式，檢查各項輸入參數。過濾“~!#\$%^&*[]<>' \r\n”等特殊字符的輸入，或以換碼序列取代這些字符。不應依賴客戶端手稿程式進行驗證。

應用系統必須只能接受那些嚴格限制與預期的字符數據，若預期的是數字，只有數字的字符才被接受，若預期的是文字，則只有字母才被接受。輸入數據的形式是否合適也應作出驗證，若預期是一個電郵地址，則應透過適當安排的文字、數字、@符號、破折號及點號作出驗證。所有進入應用系統的數據應強制最小或最大長度。此技術應使用在帳戶號碼，交談憑證與使用者名稱等等。所有這些技術限制了入侵攻擊潛在缺口的數目。

2. 對應用程式回應進行淨化（Sanitised application response）

應發展一個集中模組，以進行淨化工作。檢查所有輸出、調用返回碼及錯誤碼（例如後端數據庫的調用），以確保根據預期的處理程序執行。舉例說，在回應中不必要的內部系統資料，如內部 IP 地址、主機名稱、目錄結構、伺服器的詳細錯誤信息等資料，便不應在客戶端洩露。大多數應用系統／網站伺服器允許自訂錯誤頁面。

3. 超文本傳輸規約（HTTP）的可信性

程式編寫員不應信賴及依賴 HTTP REFERER 標頭、表單字段或 cookies 程式來確定保安措施，因這些數據均可被偽冒。除非採用了有效的加密算法核實 HTTP 標頭是否完整，否則不要信賴這些來自客戶端瀏覽器的參數。此外，由於隱藏參數容易被攻擊者操縱，故不能認為用戶無法修改隱藏參數。

4. 保存伺服器上敏感的對話以防止客戶端修改

不應將敏感資料儲存在任何客戶端瀏覽器的 cookies 程式中。如果一定要的話，應採用較強的加密技術保護數據的機密性及完整性。

⁶ <http://www.cert.org/secure-coding/>

5. 加密含有敏感資料的頁面及防止快取記憶

在傳輸過程中，含有敏感資料的頁面應使用適當的算法及密碼匙進行加密，例如保密插口層 (Secure Socket Layer, SSL)、傳輸層保安 (Transport Layer Security, TLS)。使用已簽署的 Java 微應用程式或 ActiveX 獲取及顯示敏感資料，及設置適當的 HTTP 標頭屬性，以防止瀏覽器或代理伺服器對含有敏感資料的個別頁面進行快取記憶。

6. 對話管理

對話識別碼應比較長、複雜並包含難以預測的隨機數字。在對話中，應經常更改對話識別碼，以縮短對話識別碼的有效期。此外，對話識別碼不應儲存在劃一資源定位、永久性 cookies 程式、隱藏的超文本標示語言字段及 HTTP 標頭中。程式編寫員可考慮在客戶端瀏覽器的對話 cookies 程式中儲存對話識別碼。透過保密插口層／傳輸層保安保護對話識別碼，使攻擊者無法從網絡中竊取。應用系統應提供登出功能及執行閒置對話超時。當用戶登出後或閒置對話過期時，須確保不但刪除客戶端的 cookie 程式（如可能的話），亦清除瀏覽器的伺服器端對話狀態及與後端伺服器的連接。

7. 接達限制

確保終端用戶賬戶僅有權接達獲授權的操作程式，並限制接達後端數據庫，或運行結構化查詢語言指令、操作系統指令。如應用系統向接達程式作出系統調用，不應直接調用真實檔案名稱及目錄路徑。倘若攻擊者獲取源碼，則可發現系統資料。利用網站伺服器的對映功能作為過濾。

8. 建立應用系統審計及報告的中央模組

9. 使用最適當的認證方法識別及驗證輸入的用戶查詢

執行 Threat Modelling 模式

要建立安全的應用系統，便需要瞭解此應用系統所面對的威脅，Threat Modelling 的過程可協助我們確定在整體應用程式方案上的威脅、攻擊、漏洞與抵禦措施。Threat Modelling 可透過以下步驟達成：

步驟一：確認關鍵的保安目的。

步驟二：將應用系統的重要特性列出並建立整體的看法。

步驟三：解構應用系統，以確認需要被評估而會影響安全的特性與模組。

步驟四：確認所有威脅。

步驟五：確認所有保安漏洞。

設計網上應用系統保安的架構

網上應用系統結構一般包括三個層級，把對外網站伺服器、內部的應用系統伺服器及數據庫伺服器隔開。藉著這些層級結構，即使攻擊者可入侵對外網站伺服器，他也需另尋方法攻擊內部網絡。這是縱深防禦的原則，縱深防禦對資訊保安而言是一個實用的方法，其基本概念主要是在於多層次的保安，以保護重要資產。保安的層次包含輸入驗證、數據庫層的概念、伺服器的組態、代理伺服器、網上應用系統防火牆、數據加密與作業系統強化等。

發展階段

就減低程式碼保安問題而言，這是最重要的階段。注意到保安編碼標準可協助改善保安狀況，並減少導致保安事故常見錯誤的發生次數。此外，執行保安風險評估也有助於確認所需的保安控制。

測試與品質保證階段

在任何應用系統推出前，全面測試是非常重要的。除用戶驗收測試之外，尚有其它測試，例如系統測試、壓力測試、迴歸測試、和單元測試等，對於驗證系統功能的效能和準確性是有用的。本節敘述一些測試，可用來增加已發展的程式或系統的可靠性與安全程度。

網上應用系統單元測試

網上應用系統單元測試是發展階段中的一個重要部份，其設計是用來協助確認網上應用系統中的保安漏洞。單元測試包含了個別程式或模組的測試，以確保程式內部的操作或模組的執行與規格一致。單元測試應包括對一般保安問題的測試，例如緩衝區溢出，尤其當模組与其它組件合併，此測試更為重要。如果沒有執行單元測試，便很難在發展階段中執行自動保安測試程序。

有許多不同的工具，能幫助找出並消除網上應用系統的保安漏洞。但值得注意的是，這些工具只能應付一小部份有效的網絡保安應用程式所需之測試。只是依賴這些工具，而不集中改善軟件發展生命周期的保安，這是錯誤的保安觀念，因為自動掃描工具的能力仍只限於找出及確認某種類的保安漏洞。

編碼覆檢

編碼覆檢能幫助確認出保安漏洞，確保維持保安發展標準及整體程式設計的一致性。一般而言，系統發展經理、系統管理員、與數據庫的管理員會一起檢查應用程式原始碼的運作，並作出適當的建議加以改善。此外，藉著對原始碼的檢視，可以確認及評估隱藏或保安性的內容，如密碼匙和密碼所採用的保護措施是否適當。

市場上可以找到一系列的自動掃描工具，協助我們進行編碼覆檢。然而若結合網上應用系統掃描工具，這些工具只能用來確定一般的錯誤，而不是較複雜的保安問題。因此不應以這些工具取代人的分析。

在任何編碼覆檢開始之前，項目小組應先制訂編碼的哪一部份是屬於高風險和可能易受攻擊。一般來說，能提供接達控制、組態管理、審計、記錄、授權驗證、提供與第三者或操作系統連結的界面，以上具備這些功能的程式或作業系統，都應接受覆檢。這項覆檢動作，通常參考 **threat modelling** 或風險評估和設計分析的資料，以決定此項目的哪一個範圍應該是編碼覆檢的一部份。

應用系統推出前階段

在應用系統推出前和作出重大改動後，應執行資訊科技保安審查。每個修復的保安漏洞都需更新及反映在程式碼內。接下來，每一項保安漏洞修復都需要一次程式碼更新。因此，要持續維護安全的應用系統，必須評估每項修復所帶來的影響。

維修與支援階段

保安是持續的過程，保安問題在應用系統推出後仍不斷出現。應繼續執行保護與偵測的機制，以確保應用系統安全地與順暢地執行。重要的持續保安措施如下：

應用系統記錄的覆檢

為了偵測網上應用系統的不正常狀況，記錄覆檢是必要的。許多網絡伺服器支援詳盡的記錄，它們保存了所有在網上應用系統的指令。藉著定期審視網絡接達記錄和指令，便有可能預見網上應用系統未來的保安問題。一旦發現不正常的劃一資源定位（URL），可能代表某種網絡攻擊已經發生。

此外，應用系統的擁有人也可要求執行網上應用系統的審計追蹤，定期查閱那些可疑的指令或顯示不正常情況的報表。

版本控制與獨立發展環境

維護應用系統的完整性應以適當的保安控制，例如版本控制與各項獨立環境供系統發

展、系統測試、驗收測試與現場操作等。生產與發展環境應同時進行。發展應用系統的工作人員除了某些必要狀況外不可取得生產資訊。

網上應用系統防火牆 (Web Application Firewalls, WAF)

標準的防火牆可以限制或允許經過機構授權的接達行為，但在機構的網上應用系統防火牆仍然無法理解在網上應用系統的特定內容。根據 Web Application Security Consortium (WASC)，網上應用系統防火牆 (WAF) 是「*an intermediary device, sitting between a web-client and a web server, analyzing OSI Layer-7 messages for violations in the programmed security policy*」⁷，即 WAF 是一個中間設備，介於網絡客戶端和網絡伺服器端之間，作為分析違反程式保安政策之 OSI 第七層訊息之用。

一般而言，網上應用系統防火牆通常安裝於網絡伺服器之前，這些網上應用系統防火牆的形式如同標準防火牆，它可以是軟件或者是硬體，目的是讓網絡伺服器免受攻擊為主。防火牆有兩種保護方式：

1. 以辨識碼為基礎：WAF 透過檢查網絡請求方式，對照攻擊辨識碼特徵檔以確認攻擊。
2. 以不正常行為作為基礎：網上應用系統防火牆透過偵測不正常的傳輸模式確認攻擊。

網上應用系統的驗收清單

網上應用系統一經驗收後，應執行獨立的保安評估來評估網上應用系統和原始碼，以確保完全符合公司政策或項目的保安需求。對網上應用系統項目來說，這類評估是必要的，且可能會外判至外部人員。保安控制的測試案例，需要在項目的初步階段實行，且應包含用戶驗收測試。

應盡早在項目的初步階段考慮其保安性。必須與開發人員溝通對保安的期望與需求，特別是授權機制、數據輸入的認證、與審計追蹤。

以下是在評估網上應用系統保安時，需要檢視範圍的一些例子：

身份識別及認證

1. 使用者與其使用程序如何認證？
2. 認證程序是否遵照規格的指示，以及是否遵守機構的保安政策？
3. 如果認證過程是以密碼為基礎，使用者的密碼如何處理與保存？這些密碼處理

⁷ <http://www.webappsec.org/projects/glossary/>

機制，是否符合機構的保安政策？這些密碼是寫定的密碼，或是內建於原始程式碼中？

4. 這些應用系統是否需要在每次連線時針對每一個連結作授權認證？

數據保護

1. 數據保護機制是否符合機構的保安政策？
2. 所有受保護的數據是否經由正確的方式傳輸？
3. 如果使用加密，如何操作？加密的操作程序是否完全遵照機構的保安政策？

記錄

1. 審計追蹤記錄的機制，是否符合規格？
2. 這些應用系統的審計記錄，是否容易受到那些未授權的刪除、修正或洩露？

處理錯誤

1. 錯誤訊息是如何被處理？洩漏的數據，有沒有可能被後來的攻擊行為所利用？應用程式的失敗，會導致系統處於危險狀態嗎？

操作

1. 是否有強制執行職務分工（Segregation of duties）和最少權限原則？
2. 在啟動前，所有的內建使用者身份、測試的使用者身份、和預設密碼值的身份是否已從作業系統、網絡伺服器及其應用程式本身中移除了？
3. 系統管理程序、改動管理程序、運作復原程序、和備份程序，是否有完整且清楚的界定？

這份清單尚有不足之處，因應保安需求，與網上應用系統目的之特殊性，應根據特殊需求以涵蓋額外測試案例或檢查標準。

此外，當任何的資訊系統外判給第三方服務提供者時，須執行適當的保安管理程序來保護數據，並減輕資訊科技項目或服務外判時帶來的保安風險。建議讀者可參閱“資訊科技服務外判保安”一文，得知更多有關外判所需保安考慮的資訊。

附錄一：瀏覽互聯網時如何保護自己

當我們享受現代網上應用系統與服務的便利時，終端用戶應採取積極方法來保護自己。在許多網上交易應用系統中，簽署或同意使用條款的敘述對顧客而言是常見的，顧客同意網上應用系統提供者對有關因顧客帳戶的保安問題，而導致任何的遺失或損害不需負責任。

終端用戶的一般性防護：

1. 不要從公用電腦登入重要的網上應用系統。
2. 在工作站中不要設定使用者帳號和密碼快取。
3. 連線結束後記得登出。
4. 對於不同的網上應用系統與服務，使用不同組別的登入帳號與密碼。
5. 重要的網上應用系統若未支援一次性密碼，則須經常更換密碼。
6. 異常行為發生時，立即向服務提供者報告。
7. 確保作業系統與系統元件，如互聯網瀏覽器，是完整安裝且更新修正至最新版。
8. 安裝個人防火牆與更新了最新病毒碼的防毒軟件，任何防毒軟件應該可偵測出惡意軟件，例如鍵盤側錄程式。
9. 不要從未知的網址來源下載軟件或外掛軟件。

附錄二：如何消除程式碼的十大關鍵保安漏洞

OWASP 機構歸納出網上應用系統十大關鍵保安漏洞，項目小組不僅提供關鍵保安漏洞的詳細數據，也提供如何從程式碼中消除這些保安漏洞之建議。以下是由 OWASP 網站所摘錄的建議事項。

1. 使用標準的輸入驗證機制來驗證輸入數據。
2. 強化輸出編碼。
3. 詳細說明輸出編碼（如 ISO 8859-1 或 UTF 8）。
4. 當輸入或替輸出編碼時，不要使用黑名單式的驗證方法來偵測跨網址程式編程攻擊（XSS）。
5. 注意標準錯誤。
6. 使用強化的應用系統程式界面（API）查詢參數。
7. 強制最小權限原則（Least privilege principle）。
8. 避免過於詳細的錯誤訊息提示。
9. 使用預儲程序時要特別小心。
10. 不要使用動態查詢界面。
11. 不要使用簡單的程式跳出功能。
12. 使用間接的物件參照圖。
13. 使用詳盡的弱點檢查機制。
14. 增加防火牆規則以避免網絡伺服器製造外部網站與內部系統的新連線。
15. 檢查使用者提供的檔名或檔案。
16. 考慮部署「chroot jail」。
17. 無論何時，盡可能避免顯示私人的物件參照。
18. 以目前已知且可行的方法，廣泛驗證任何私人的物件參照。
19. 對所有參照物件確認授權。
20. 每一表單和 URL 插入訂製的隨機權標（custom random tokens）。
21. 進行重要交易與敏感數據時，需重新認證或使用交易簽章。
22. 進行重要交易與敏感數據時，不要使用 GET 語法請求 URLs。

詳細的內容請參考 OWASP 機構網站所公布的 Top Ten Project 主題網頁（http://www.owasp.org/index.php/OWASP_Top_Ten_Project）。