防範資料隱碼攻擊之方法與探討

作者／邱柏雄上尉

提要

資料隱碼(SQL Injection)攻擊已行之有年；但現今仍然是網站的威脅來源，原因在於形成這種攻擊弱點的主要因素是人為造成的，而攻擊原理乃利用資料庫結構式查詢語言（Structure Query Language, SQL）特性，針對程式撰寫上的弱點，以組合字串對資料庫管理系統進行非法的查詢或動作。現今國軍內部網頁應用系統的開發；隨著資訊人力的不足，易造成程式撰寫不嚴謹，存在「資料隱碼」漏洞之隱憂。

本文探討國軍網頁應用程式對於資料隱碼的相關安全議題，並提出防範對策及修正作法，以期降低資料隱碼攻擊威脅，使得國軍整體網路安全防禦機制更加周全。

關鍵字：資料隱碼、SQL Injection、非法的查詢。

前言

國軍推行資訊化作業之數項作為，其中如通資電學校以線上鑑測系統進行班隊學員生階段測驗及期末測驗(如圖一)，教官只需於線上批閱，電腦即能夠自動將成績計算；此外，各班隊學員採用教學網站進行教學的方式(如圖二)，學生對於印製紙本講義需求量將大幅減少。以上做法不僅提昇作業效率，也減少了紙張的用量，更符合了國家節能省碳的政策。

圖一 線上測驗系統
(資料來源：陸軍通資電校線上測驗)

圖二 教學網站

(資料來源：陸軍通電校資作組教學網站)

隨著國軍作業資訊化作業的腳步的邁進，應用程式系統結合資料庫的架構在將來必更日蓬勃發展，而現今系統資料庫大多是以關聯式資料庫管理系統(Relational Database Management System, RDBMS)為主，此類型資料庫系統的操作皆遵循 SQL 語法，透過語法的編排可便利的完成各項工作，因此程式設計師在存取資料庫時，往往利用程式語言像是 Visual Basic、C#等組織 SQL 語言，再傳遞給資料庫管理系統，執行建立、刪除或是賦與、移除權限或者是查詢、新增、修改、刪除資料記錄。

正因為 SQL 語言的便利性，讓惡意使用者有機可乘，因此，本論文將探討資料隱碼的形成原因及防範機制，以確保網頁應用系統與資料的安全性。

本文

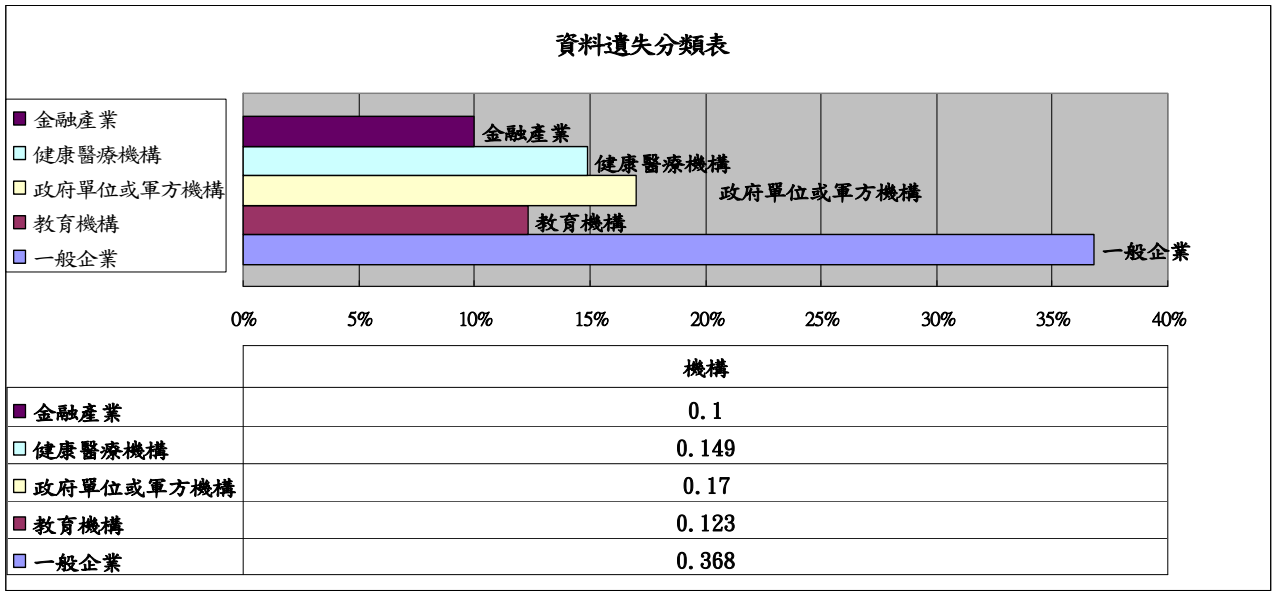
一、資料遭竊取案例分析

根據 IBM 發佈的 X-Force2008 年中趨勢統計報告，結果顯示數位犯罪正採用新的自動化技術和策略，因此較能比以往更快速的利用網路漏洞進行攻擊。報告中顯示有 94%針對瀏覽器的攻擊都發生在漏洞公布的 24 小時內，即人們意識到自已的系統存在漏洞之前就發生攻擊。而已公布的漏洞攻擊有一半以上都是和 Web 應用程式有關；在 2007 年至 2008 年上半年，其中和 Web 伺服器應用相關的攻擊中，隱碼攻擊所佔的比例從 25%成長至 41%，威脅程度可見其分量。

國軍網路類型為封閉型網路，且樣本空間過小，對於資料遺失相關統計資

料不易取得，以下為美國身份竊盜資源中心(Identity Theft Rrsource Center, ITRC)截至2008年8月的全球個資外洩的產業類型分類統計資料；如表一。

表一 全球分類產業資料遺失分析



	機構
金融產業	0.1
健康醫療機構	0.149
政府單位或軍方機構	0.17
教育機構	0.123
一般企業	0.368

(資料來源：ITRC,http://www.idtheftcenter.org/artman2/publish/lib_Survey/ITRC_2008_Breach_List.shtml)

(一)2008年5月企業遭受大規模資料隱碼攻擊

案例內容：調查局近來發現駭客集團利用來自中國為主的中繼站，對台灣企業網站進行大規模的資料隱碼攻擊，預估有數萬網頁受害，而整體受害情況難以估算。以攻擊成功累積速度來看，調查局判斷這波攻擊應是以程式自動化攻擊，受害網站數量仍在持續累積中。

調查局指出，這次大規模攻擊事件，目的應為藉由資料隱碼攻擊，於企業網站放置掛馬網頁，再以此攻擊瀏覽網站的使用者。民眾連上受害的企業網站後，將會連結特定網址下載惡意程式，導致個人資料可能因此外洩。由於此次攻擊透過大量遙控方式，短時間遠端控制多台跳板電腦，藉由已知的漏洞，對台灣企業進行攻擊，由於此次攻擊非是直接竄改網頁，而是直接修改資料庫內容，受害企業很難發現攻擊行為。

案例分析：此次攻擊不再是以停擺伺服器服務或篡改網頁內容為攻擊目的，而是以SQL注入工具[1]或是自動機器人之類的工具進行攻擊，在資料庫的VarChar、Text等欄位內加上攻擊程式的連結，將伺服器當成感染源，攻擊用戶端的電腦，藉以竊取用戶端電腦資料或把其當成跳板，便以進行下一波的攻

[1] SQL 注入工具攻擊,http://blog.darkthread.net/blogs/darkthreadtw/archive/2007/06/01/816.aspx

勢，這類攻擊不只國內，就連國外知名網站-美國商業週刊網站(Business Week)也於2008年9月遭同樣手法攻擊，波及廣大的客戶群和企業，可知在自動化的資料隱碼攻擊後是不挑受害者對象的，防範稍一鬆懈就可能淪為目標；此外，遭受攻擊的影響不只是資料被篡改或遺失而已，而是造成波及的受害者範圍更大，且可能遭利用成為下一波攻擊的工具，所造成的損失更是難以估算。

(二)大學入學考試、國中基測中心遭駭客入侵案[2]

案例內容：2005年4月25日，台北刑事局偵九隊偵破大學入學考試、國中基測中心遭駭客入侵案，蘇嫌涉嫌入侵大考中心、基測中心學生個資資料庫，竊取電腦系統學生個資，將取得的一百多萬筆全國國、高中生個人資料交給補習班，作為招攬生意之用，所幸為考生資料及成績未被篡改。據調查，蘇嫌也曾入侵總統府網站、悠遊卡系統及某百貨公司網站(瀏覽公司會員資料約八萬筆)。

案例分析：此案中個資被竊取幸好只是淪為補習班作為招攬生意之用，而非是藉以篡改成績協助考生考取學校以謀取巨利，或者落入詐騙集團手中，後果更是不堪設想。

(三)駭客集團竊個資五千萬筆3

案例內容：2008年8月26日，刑事局偵九隊經一年多蒐證，偵破以陳姓男子為首的駭客集團，陳嫌涉嫌以成立工作室入侵中華郵政網路銀行，並將客戶存款盜轉新台幣數百萬元，此外還入侵健保局、教育部及多家電信公司資料庫竊取個人資料，進而整合全台超過五千萬筆個資資料庫，其包含政府官員、民意代表及企業界人士，甚至一般百姓等幾無一倖免。

案例分析：便利的網路服務連結了個人資訊所在的資料庫系統，一旦資料庫保全的防線潰敗，牽連受害者層面將是廣大的客戶群，而數位罪犯型式從個人模式發展至集團行動，攻擊的效率更高，試想如果要面對此類組織化的數位攻擊，正面的重視資料庫安全防衛措施必為首項要點。

二、就國軍資訊作業環境資料隱碼入侵可能模式分析

資料隱碼入侵手法繁多，筆者就以下四種資料隱碼的攻擊模式作分析：

◆ 利用註解符號配合SQL陳述式使用已知帳號或未知帳號情況下進行攻擊。

◆ 藉由某些網頁會傳遞參數的特性進行攻擊。

◆ 試探過程中利用錯誤訊息進行資料表單結構分析。

---

[2] 大考中心入侵案,http://mag.udn/com/mag/campus/stroypage.jsp?f _ART_ID=11599

[3] http://www.odnews.com.tw 2008-08-26

◆ 利用後端資料庫管理系統預設延伸預存程序進行攻擊。

本段會以ASP.NET及ASP語法撰寫具有資料隱碼弱點的網頁(Default.aspx及Default.asp)，並搭配MS SQL Server2000資料庫管理系統進行資料隱碼可能模式分析，並於下段說明國軍程設人員如何針對此類程式弱點進行防護。

首先以MS SQL Server2000建立一人員資料表Users，如表二。

表二 人員記錄表

使用者編號	帳號	權限	密碼	備考
sysadmins	sysadmins	99	abc+1234	Administrators
Poweruser	Admin	88	def+1234	Authorized
Users	stu	66	12345678	Guests

(資料來源：作者整理)

資料隱碼攻擊可能模式：

(一)利用註解符號配合SQL陳述式使用已知帳號或未知帳號情況下進行攻擊。

1.未知帳號配合登入嘗試

在未知帳號的條件下可以「' or 1=1--」等組合字串嘗試輸入；由於「or」運算子兩端運算式只要其中之一為真時，此敘述運算結果將為真，而在「--」之後的敘述則被註解掉。因此帳號將被忽略而密碼會被註解掉，此語法運算結果將會通過SQL Server的驗證達成登入。

圖三 登入嘗試

(資料來源：作者整理)

2.利用已知帳號名稱登入

假設攻擊者使用社交手法，竊知其他使用者帳號，或是以預設或慣用帳號名稱，像是Administrator、Sysadmin、Admin或Sa配合註解符號(--)進行嘗試

登入，當使用者輸入字串「sysadmin ' --」於登入帳號欄位於頁面，實際上傳給SQL Server的字串為「Select count(*) From Uers Where UserName='sysadmin' -- And Passwords="」；由於帳號sysadmins這一筆記錄已在資料表Users中存在，而在「--」之後的語法一律視為註解，因此SQL Server對於驗證密碼部分的語法將不予理會，這個敘述將永遠為真；會通過資料庫管理系統的驗證而達到登入的目的。

以上2種利用註解符號配合SQL陳述式的攻擊方式會成功的原因是直接以參數組合SQL語法，交由資料庫系統執行而造成的，程式直接以取得的參數來組裝SQL語法方式是不安全的寫法。

(二)藉由某些網頁會傳遞參數的特性進行攻擊

接著在ASP.NET語法建構的網頁Default.aspx，於IF區塊用來驗證帳號/密碼的判斷式中，於其中以程式碼「Response.Redirect ("Default2.aspx?" & "AccountTxt='" & AccountTxt.Text & "'", True)」加以驗證，使得程式驗證成功時，會利用URL傳遞使用者帳號並將頁面導至Default2.aspx，並於該頁面進行查詢該使用者帳號資料記錄且將結果顯現於網頁之上。

其中Default2.aspx程式碼關鍵部分為「Dim strSQL As String=“Select * From Users Where UserName=” & Request(“AccountTxt”).ToString( )」；以QueryString的方式取得由前一頁PostBack過來的參數AccountTxt。由於網頁會傳遞參數的特性，將原本傳遞變數的網址後端加上字串「?AccountTxt= 'sysadmins' union all select null,UserName, null,null,null from sysobjects where xtype='u'」，企圖欺騙程式於Default2.aspx頁面顯示時，不僅是列出使用者帳號資料，並且將資料庫中其它使用者自行定義表單也一併列出。

由於網頁Default.aspx會傳遞參數，將本來欲給資料庫管理系統執行的SQL語法，修改為一併查詢資料庫中所有使用者自訂資料表清單。語法字串使用了MS SQL Server資料庫中的預設資料表sysobjects特性，其內會記錄該資料庫內資料表清單相關資料。可再配合查詢的語法將所有資料表列出。

在獲取資料表清單資訊後便可查知資料庫其它資訊，重複上述作法便有可能查得該資料庫底下欲得知的資料表記錄，以Default2.aspx為例可再獲取Users資料表內所有帳號密碼明細資料，如圖四。

UserId	UserName	Priority	Passwords	Comment
sysadmins	sysadmins	99	abc+1234	Administrators
poweruser	admin	88	def+1234	Authorized
sysadmins	sysadmins	99	abc+1234	Administrators
Users	stu	66	12345678	Guests

查詢帳號密碼成功!!

圖四 查詢Users資料表明細
(資料來源：作者整理)

(三)試探過程中利用錯誤訊息進行資料表單結構分析

當程式因為執行語法錯誤的SQL陳述式而引發未處理的例外狀況，若是管理人員未適當的加以屏蔽，則會透露出程式潛在的弱點，此時配合註解符號(--)及查詢分隔符號(;)，即可進行非法查詢或插入記錄。

1.試探欄位數量及名稱

Microsoft在預設上，當ASP網頁的Script發生錯誤時，會透過磁碟的>\WINNT\Help\issHelp\common\500-100.asp，將網頁錯誤訊息傳回前端，對於開發者而言，可提供為除錯的依據，但也同時提供了非法使用者進行分析資料表單的機會。作者以網頁Default.asp為例；於帳號欄位輸入「' having 1=1 --」，由於網頁Default.asp設計上會檢查欄位是否為空，因此需要在密碼欄位輸入任意值以通過檢查；利用故意造成的錯誤輸入使得網頁傳回「資料行'Users.UserId'在選取清單中…」的執行錯誤訊息，提示了此網站使用的資料表為「Users」；且查詢的資料表中第一個欄位為「使用者」。
分析所獲得訊息後；進一步再利用語法字串「' Group by UserId having 1=1 --」可進行下一個欄位的試探，實際上傳送給MS SQL Server的語法字串為「SELECT * FROM Users WHERE UserName='' GROUP BY UserId Having 1=1 --' AND Passwords= 'aaa'」。執行錯誤回傳的訊息會如「資料行'Users.UserName'在選取清單中…」的格式；可知在SQL語法欄位數量不足時，系統訊息會提示下一個欄位，藉以得知第二個欄位為「UserName」。

重複以上動作，直到不造成執行時期錯誤，即可獲知完整的表單結構。其Users資料表結構欄位為：UserId、UserName、Priority、Passwords、Comment。因此程式若不屏蔽錯誤訊息顯示時，資料表結構便很容易為人所查知。

2.獲得欄位資料型態

除了獲得資料欄位的數量及名稱外，可使用故意造成錯誤的SQL語法組合字串；利用union關鍵字會比對欄位數量及型態的特性，像是「'union select count(*),1,1,1,1 from Users union select UserName,1,1,1,1 from Users --」的語法，來獲得資料表欄位資料型態資訊，以網頁Default.asp為例；於帳號輸入欄位鍵入上述字串。交由程式執行後，分析網頁回傳之錯誤訊息「SQL Server將varchar數值'Admin'轉換成資料型別為int的資料行語法錯誤」，可得知資料表Users的UserName欄位的資料型態為VarChar，除此之外連查詢時的第一筆資料的UserId欄位值都可能獲取，於此例為「Admin」，如圖五。

圖五 獲取資料型態
(資料來源：作者整理)

重複上述動作，利用先前取得的資料表欄位資訊，可查知帳號「admin」的密碼，只要設定查詢對象為「Passwords」欄位，而條件設定為符合UserName等於「Admin」，並於帳號欄位鍵入語法「'union select count(*),1,1,1,1 from Users union select Passwords,1,1,1,1 from Users where UserName = 'Admin'--」，在故意造成如圖六的錯誤訊息回傳的情況下，在頁面上可得知密碼欄位Passwords的值為「def+1234」的訊息。綜合以上；可得知Users資料表中，有一筆記錄為帳號「Admin」，密碼「def+1234」的資料，且帳號欄位的資料型態為varChar；密碼欄位的資料型態為int。因此，可知經由錯誤訊息不僅可能會洩露資料表結構資訊，也可能將整個資料庫資料拱手送人。

圖六 包含密碼的錯誤訊息
(資料來源：作者整理)

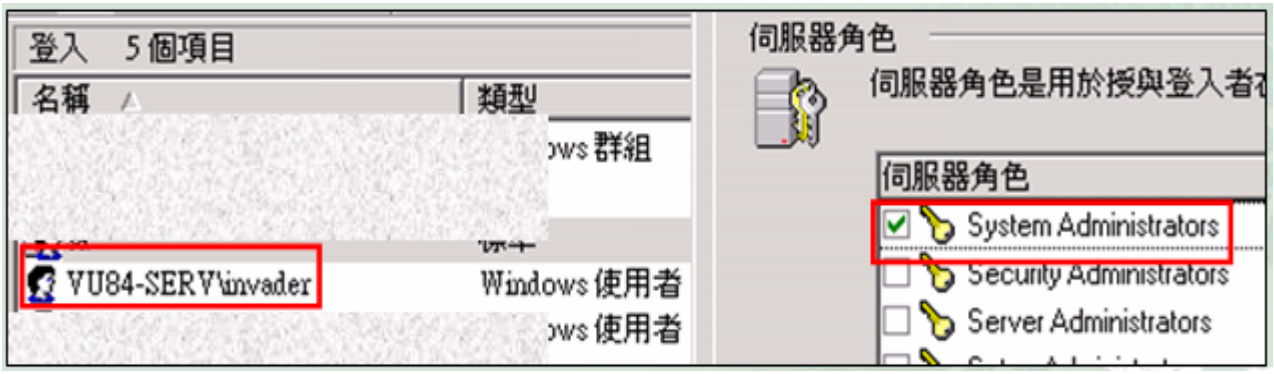
(四)利用後端資料庫管理系統預設預存程序進行攻擊

MS SQL Server中預設存在功能強大的預存程序，本來是為了便利管理者進行各種管理作為，若是反遭利用，有可能利用其預存程序配合組合字串進行非法查詢，或者進行權限的偷盜。假設如欲獲得SQL Server管理者帳號密碼，可利用Master資料庫中延伸預程序xp_cmdshell，於網頁Default.asp之可注入點如帳號欄位鍵入如「' ;exec master.. xp_cmdshell 'net user invader /add' --」的語法，於資料庫管理系統所在的平台上新增作業系統使用者。

雖然此種作法往往於網頁的驗證上不會成功，也可能不會出現任錯誤訊息(除了帳號、密碼之類的訊息)，但是以網頁Default.asp為例，利用此一延伸預存程序可在資料庫管理系統所在平台的作業系統新增一名為「invader」的帳號，如圖七。

圖七 新增使用者invader
(資料來源：作者整理)

光是取得作業系統帳號可能還無法達到目的，接著可以預存程序sp_grantlogin 將帳號invader加至資料庫管理系統上，再以延伸預存程序sp_addsrvrolemember將invader帳號加入SQL Server的sysadmin帳號群組；賦與Sysadmin群組權限，藉以獲得資料庫系統管理者權限，如圖八。侵入者可藉此帳號登入MS SQL Server，進行各項操作或竊取資料。



圖八 新增帳號權限賦與
(資料來源：作者整理)

在資料庫管理系統中可利用之預存程序為數不少，像是以xp_reg開頭的延伸預存程序，可進行登錄檔相關編輯，或是以sp_OA開頭的一組可建立執行OLE物件的預存程序等等，都可能利用來作為入侵系統的工具。

三、國軍因應資料隱碼威脅之防範對策及修正作法

防範對策作法可分別由以下幾點進行：

◆ 資料隱碼弱點檢測步驟

◆ 系統及架構防範作為

◆ 程式撰寫注意事項

(一)資料隱碼弱點檢測步驟

對於已存在的應用系統，可進行以下步驟檢測是否具有資料隱碼的弱點：

1.利用SQL Injection檢測工具進行檢測

欲檢測系統是否具有資料隱碼弱點，可先行使用民間廠商開發出的檢測工具進行測試；以下為幾款檢測工具[4]：

[4] Top 15 free SQL Injection Scanners,http://www.security-hacks.com/2007/05/18/top-15-free-sql-injection-scanners.

表三 工具列表

工具名稱	下載位址	特性
Acunetix Web Vulnerability Scanner	http://www.acunetix.com/wvs/vulnerability-scanner.htm	可連線更新 Plug-IN 及弱點
Automagic SQL Injector	http://www.indianz.ch/tools/attack/automagic.zip	為一自動化工具，設計上可節省穿透測試的時間，但它主要針對 vanilla Microsoft SQL injection 漏洞測試。
SQLBrute	http://www.justinclarke.com/security/sqlbrute.py	以 blind SQL injection 測試系統弱點，支援對 MS SQL Server(Time based、Error Based 測試弱點)及 Oracle(Error Based 測試弱點)，以 Python 寫成
BobCat	http://www.northern-monkee.co.uk/projects/bobcat/bobcat.html	可將使用者使用中的資料取回，並列出連結的伺服器，資料庫概要資料。
SQL Power Injector	http://www.sqlpowerinjector.com/	可自動的在頁面進行 SQL 語法的注入，作穿透測試。
Absinthe	http://www.0x90.org/releases/absinthe/download.php	圖形化的使用界面，可針對 Blind SQL Injection 防護能力較弱的系統下載資料庫相關資料。
SQL Injection Pen-testing Tool	http://sqltool.itdefence.ru/indexeng.html	圖形化的使用界面，主要可用來針對網頁應用程式作檢測。
SQID	http://sqid.rubyforge.org/	和 Lilith 相同是命令列的使用者界面，主要可用在對網頁的 Submit Form 作 SQL Injection 命令的測試。
Blind SQL Injection Perl Tool	http://www.unsec.net/download/bsqlbf.pl	以 Perl Script 寫成的工具，主要用在獲知網站關於 SQL Injection 弱點的資訊。
Lilith	http://angelo.scanit.biz/	以 Perl 寫成的免費工具，只要平台可執行 Perl5 以上版本皆適用。
FJ-Injector Framwork	http://sourceforge.net/project/showfiles.php?group_id=183841	為一開放原始碼的工具，主要針對網頁應用程式，具有類似 Proxy 的功能，可攔截並修改 HTTP Requests。
SQLNinja	http://sqlninja.sourceforge.net/	可針對網頁應用程式及 MS SQL Server 作測試。
NGSS SQL Injector	http://www.indianz.ch/tools/attack/sqlinjector.zip	可針對多樣資料庫系統作弱點測試：Access, DB2, Informix, MSSQL, MySQL, Oracle, Sysbase.
SQLMap	http://sqlmap.sourceforge.net/	以 python 寫成的自動化 blind SQL injection 工具，它可對資料庫管理系統作 fingerprint，並可針對網頁應用系統的弱點作入侵動作。

(資料來源：http://www.security-hacks.com/2007/05/18/top-15-free-sql-injection-scanners.)

檢測工具只是作為先行過濾應用系統是否具有資料隱碼的弱點，但掃描結果安全不代表是完全沒問題，所以接著要進行手動的檢測。

2.過濾所有開放使用者可能輸入資料的頁面或表單(Web Form)

由於資料隱碼是透過正當管道利用組合字串的輸入，造成資料庫系統非預期的執行結果，所以欲檢視應用系統是否具有資料隱碼的弱點，先要將所有可能的資料輸入頁。像是可以輸入資料的登入頁面，或者是遞交的查詢頁面等等，先予以列出。

3.利用輸入SQL Injection字串進行測試

使用前述攻擊模式所提之單引號、註解符號、關鍵字having、Group by或Union進行檢測，或者使用某些已整理好的SQL Injection CheatingSheet[5]內所列檢測語法對所列出頁面一一進行測試。

(二)系統及架構防範作為

1.針對MS SQL Server

(1)避免過多的System Administrator權限帳號。

(2)刪除不必要的預存程序；以下為需要刪除及注意的預存程序表，如表四。

表四 延伸預存程序表

預存程序名稱	說明	建議刪除	建議檢視(具風險性)
xp_cmdshell	將指令字串當成作業系統的命令 shell 執行，並以文字列傳回任何輸出。	V	
xp_regaddmultistring	可進行登錄檔相關編輯	V	
xp_regdeletekey	可進行登錄檔相關編輯	V	
xp_regdeletevalue	可進行登錄檔相關編輯	V	
xp_regenumkeys	可進行登錄檔相關編輯	V	
xp_regenumvalues	可進行登錄檔相關編輯	V	
xp_regread	可進行登錄檔相關編輯	V	
xp_regremovemultistring	可進行登錄檔相關編輯	V	
xp_regwrite	可進行登錄檔相關編輯	V	
sp_OACreate	在 Microsoft® SQL Server™ 執行個體建立 OLE 物件的執行個體	V	
sp_OADestroy	銷毀建立的 OLE 物件。	V	
sp_OAMethod	呼叫 OLE 物件的方法	V	
sp_OASetProperty	將 OLE 物件屬性設定為新值。	V	
sp_OAGetErrorInfo	獲得 OLE Automation 錯誤資訊	V	

[5] SQL Injection CheatSheet,http://ferruh.mavituna.com/sql-injection-cheatsheet-oku .

sp_OAStop	停止整個伺服器 OLE Automation 預存程序執行環境。	V	
sp_OAGetProperty	取得 OLE 物件的屬性值	V	
sp_executesql	執行可以重新使用許多次或已經動態建立的 Transact-SQL 陳述式或批次。		V
sp_grantdbaccess	在目前的資料庫新增 Microsoft® SQL Server™ 登入或 Microsoft Windows NT® 使用者或群組的安全性帳戶，並授權此安全性帳戶執行資料庫中的活動。		V
sp_grantlogin	允許 Microsoft® Windows NT® 使用者或群組帳戶使用 Windows NT 驗證連線至 Microsoft SQL Server™。		V
xp_availablemedia	顯示系統上可用磁碟機		V
xp_dirtree	顯示目錄下檔案架構。		V
xp_enumdsn	列出系統上的 ODBC 資料來源名稱		V
xp_enumgroups	列出作業系統上使用者群組		V
xp_getfiledetails	獲取指定檔案的相關屬性		V
dbo.xp_makecab	壓縮多個目標檔案至指定位置		V
xp_ntsec_enumdomains	列出伺服器所在網域名稱。		V
xp_servicecontrol	停掉或啟動某個服務。		V
dbo.xp_subdirs	只列某個目錄下的子目錄。		V
xp_terminate_process	以 pid 為參數強制停止執行中程序。		V
xp_unpackcab	解壓縮檔。		V

(資料來源：http://www.microsoft.com/Taiwan/sql/SQL_Injection_G2.htm , SQL Server線上說明/ Transact-SQL 程式語法的參考說明)

(3) 刪除多餘資料庫

MS SQL Server安裝後，會有預設存在的範例資料庫，這些資料庫是供程式開發、測試使用，像是「NorthWind」或「Pubs」，由於這些資料庫架構是公開的，易遭有心人的利用，要將其刪除。

2.針對網頁伺服器

(1)架設網頁應用程式防火牆（Web Application Firewall, WAF）

應用此類商規現貨本身的防禦機制，由於此類產品對於資料隱碼攻擊已累積一定程度的防禦方法與對策，因此架設網頁應用程式防火牆後，將可阻檔大部分已知的SQL Injection攻擊；這是最快也最有效的防禦方式。

(2)對網頁伺服器本身進行安全性的設定

針對IIS可設定目錄安全性設定，設定「IP位址及網域名稱限制」限制存取網頁服務的IP範圍，藉以過濾存取服務使用者族群。避免未經核可的使用

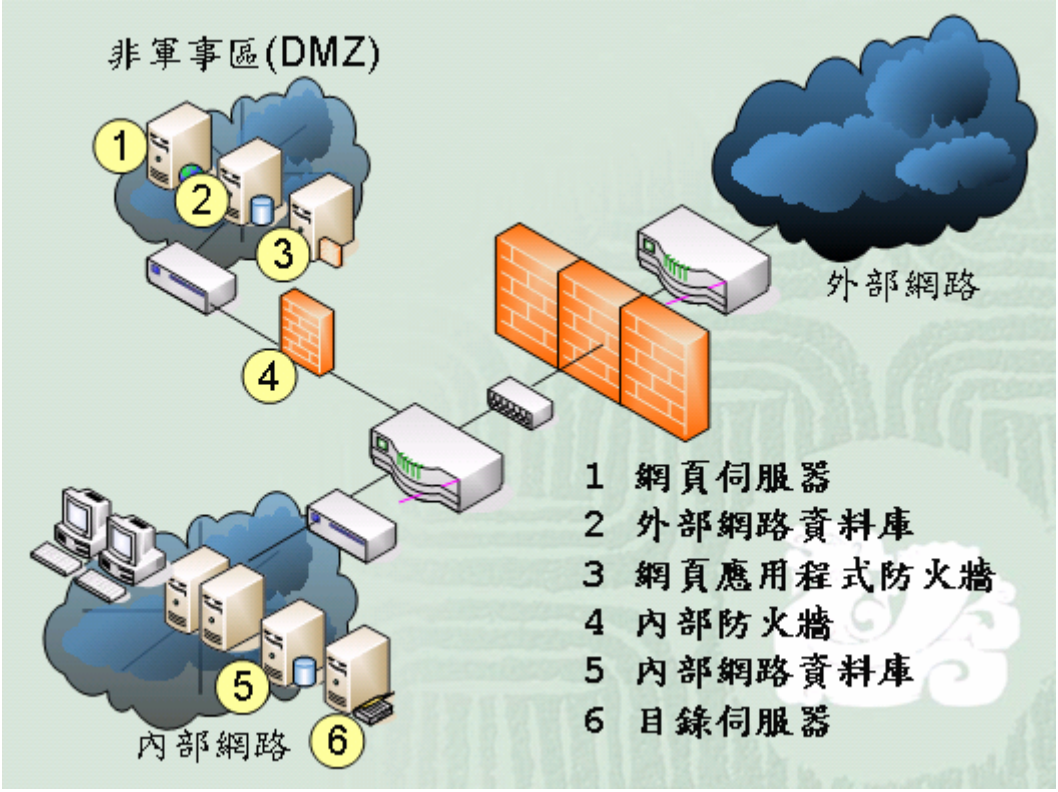
者對應用程式的存取，將資料隱碼攻擊的威脅範圍限制在已獲准存取的IP網段。

(3)伺服器上應用程式漏洞的修補與防堵

以2008年5月間大量發生於企業網站的資料隱碼攻擊為例，其中仍有多數網站以資料隱碼的手段入侵而遭植入木馬程式，卻尚未被發現，直至2008年8月藉以發起另一波攻擊，利用Adobe、Flash等應用程式漏洞對入侵網站的用戶進行攻擊，因此，適時的修補程式漏洞與更新，雖不能遏止SQL Injection攻擊，卻可使整體的防禦體制更加完整。

3.針對架構

以下為架構佈署防禦；如圖九。



圖九 架構佈署圖

(資料來源：作者整理)

(1)架設內部網路防火牆

建議於網頁伺服器與資料庫伺服器間架設一網路防火牆；基本上資料庫伺服器以提供服務，供用戶端或其它系統存取，很少對用戶端主動發出連線要求。因此可限制資料庫伺服器主動發出的連線要求。

(2)配合目錄伺服器服務(Active Directory Services)進行管制

使用目錄服務，針對應用程式系統所提供的服務進行權限管制，只分別對有權力使用該服務的用戶帳號開放存取；限制用戶能使用的服務範圍，以

防止非法的使用者進行對系統的資料隱碼弱點測試；再者考慮相關的資安政策套用至群組原則物件(GPO)中，可同時落實資安政策面及技術面的安全考量。

(3)針對網路架構設置非軍事區(De-Militarized Zone, DMZ)

非軍事區的設置，將提供外部網路服務和供內部成員使用較具有機密性的資料庫分隔開來，如此設計的考量在於提供外部用戶存取的服務，較難去加以規範使用者的使用方式，資料隱碼的隱憂至今無法消除的原因，是在於程式設計上的弱點，攻擊者的所有攻擊都是藉以合法的管道輸入來達成目的，只要程式的撰寫上不夠周延，這個問題必一直存在，因此在考量將風險降到最低的作法，即將不需提供給外部使用者的資料庫加以保護在內部網路，即使提供外部存取之資料庫被入侵，也不會立即洩露重要資訊，因此，設置非軍事區是個相當保險的作法。

(三)程式撰寫注意事項

1.避免註解符號組合SQL語法的攻擊

(1)在ASP.NET語法中，不建議以ADO.NET的SqlCommand類別直接用參數組合SQL語法字串來執行查詢，而採用搭配SqlParameter來處理參數，在攻擊可能模式所述敘的ASP.NET語法的網頁Default.aspx範例中，可修改成如下的語法：「

……

```
Dim strSQL As String="Select Count(*) From Users Where UserName=@userName And Passwords=@password"
Dim objCmd as SqlCommand=New SqlCommand(StrSQL,objconn)
objCmd.Parameters.Add("@userName",sqldbType.NvarChar).Value=AccountTxt.Text)
```

…….

」

利用SQL Parameters來替換語法中的條件值，而非是直接使用控制項的屬性組合SQL 語法字串來執行；而連接字串部分則可寫在Web.Config檔中。

採用SQL語法字串組合控制項文字屬性值，成為欲傳遞給資料庫管理系統執行的SQL命令的程式寫法，會因為輸入頁面無法規範使用者輸入「正常」的數值，將所輸入資料組裝為SQL命令的一部分，往往會產生非預計執行結果的查詢，因此，要使用SqlParameter處理參數，避免此類漏洞。

(2)ASP語法則是不建議以ADODB.Connection物件直接執行語法；取而代之以ADODB.Command物件執行預存程序來規範輸入及查詢，一來可規範輸入的參數型別，二者可限制參數長度，避免使用者輸入組合字串進行試探。建議寫法及使用之預存程序如下：

「

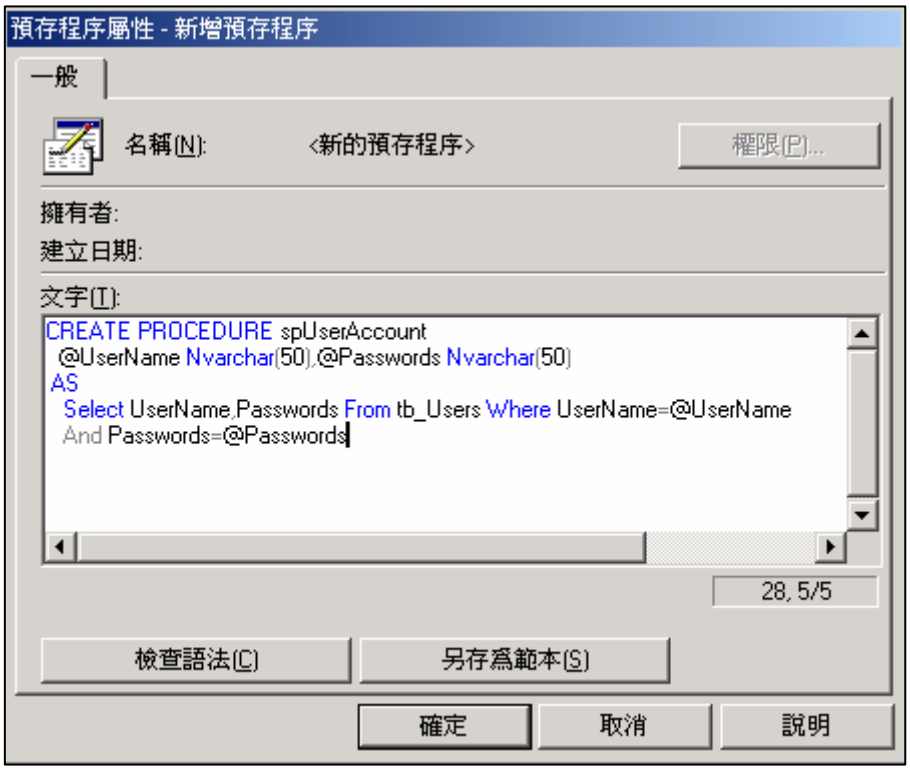
……

```
Set cmd = Server.CreateObject("adodb.command")
With cmd
  .ActiveConnection = conn
  .CommandText = "spUserAccount"
  .CommandType = 4
  .Parameters.Append .CreateParameter("UserName",202,1,50,Request("userName"))
  .Parameters.Append .CreateParameter("Passwords",202,1,50,Request("password"))
  Set Rec = .Execute( )
End With
```

……

」

預存程序 spUserAccount，如圖十。



圖十 預存程序

(資料來源：作者整理)

2.避免錯誤訊息的顯示

(1)語法的設計及網頁的重導

錯誤訊息的顯現，原本為開發時期可提供程式設計師除錯的依據，但同時也是提供非法使用者相關的參考資料來源，所以於網站開發期程結束正式上線時，就應避免此類訊息的顯現。

要避免 ASP.NET 語法撰寫的網頁出現錯誤訊息，請於網站開發完畢後，在 Web.Config 檔中將<customErrors>區段的 mode 設為 On 或 RemoteOnly 及將 Web.Config 檔中<compilation>區段的 debug 屬性設為 false。如圖十一。

```
<customErrors mode="On"   defaultRedirect="ErrorPage.aspx">
            <error statusCode="403" redirect="NoAccess.htm" />
            <error statusCode="404" redirect="FileNotFound.htm" />
</customErrors>

<compilation debug="false" strict="false" explicit="true"/>
```

圖十一 Web.Config 檔片段
(資料來源：作者整理)

除了避免錯誤訊息的顯示外，必需再針對程式執行時可能造成錯誤的程式碼區塊，使用 Try-Catch-Finally 語法，對錯誤輸出作防範及處理。

要避免 ASP 撰寫的網頁會於頁面顯現錯誤訊息，則可利用 VBScript 的 On Error Resume Next-On Error Goto 0 語法將程式碼含括其中，並使用 Err.Number 來作錯誤重導至適當的網頁。並且將磁碟中 C:\%systemroot%\ Help\iisHelp\ common \500-100.asp 預設網頁更名或刪除。

(2)以標準訊息回覆用戶端

攔截作業系統、資料庫管理系統或網站自身可能的錯誤訊息，並加以處理，統一改以明確的訊息對用戶端說明。

3.加強傳遞變數的處理

由於傳遞的變數是造成隱碼攻擊隱憂的主因之一，因此在接收變數的程式碼，有必要加強變數型別及長度的檢查部分，但不要只在前端作，像是 HTML 標籤 Input 的 MaxLength 屬性，或是以 Script 作限制，只要另存網頁，並修改原始碼，或者是直接以 QueryString 輸入，接著再次重新整理頁面，就可避免這些檢查。可在後端作以下限制：

(1)將傳遞參數中的單引號替代掉

在組合字串的程式碼部分，將使用的單引號替換成雙引號，以String.Replace("'","""")進行。

(2)過濾含有 SQL 字串的參數

過濾 SQL 語法字串，將輸入資料造成組合語法的可能加以限制；ASP.NET 語法可撰寫成如圖十二。

```
If String.IndexOf(SearchTemplateString) <> 0  then
'限制動作
End If
```

圖十二 ASP.NET 語法
(資料來源：作者整理)

這裡的 SearchTemplateString 指的是找出輸入字串中是否有可能造成攻擊的關鍵字，像是 BETWEEN、HAVING、GROUP BY、INSERT 等等。
若是 ASP 語法則可撰寫成如圖十三。

```
If Instr(String,"SearchTemplateString) <> 0  then
'限制動作
End If
```

圖十三 ASP 語法
(資料來源：作者整理)

除了使用函數過濾關鍵字外，也可使用正規表示式(Regular Expressions)，對資料加以篩選。

(3)檢查獲得參數的長度

限制適當長度的參數長度，不要只在用戶端程式進行限制，這樣的作法是十分脆弱的，建議在伺服端程式作檢查，或者參數傳遞時檢查參數的長度並加以處理，可以降低組合字串的攻擊威脅。

4.妥善使用權限管理

在程式設計上，勿以較高權限帳號存取資料庫，例如以 SQL 資料庫系統管理員(System Administrator, SA)身份存取資料庫，因此在連線字串的設計上以設定好的帳號權限來提供程式存取資料庫，如連線字串「 Server = [ip 位址] ; Database =ForPaper;User Id=[帳號]; Password=[密碼]; TrustedConnection =true」可將標示「帳號」以適當權限帳號替換。

五、建議檢查表

因應目前國軍資訊網頁服務防範隱碼攻擊威脅，作者整理以下檢查事項提供國軍管理人員作為檢測隱碼攻擊漏洞之表格，如表五。

表五 資料隱碼建議檢查表

分類	處置作為		檢測處置
資料隱碼弱點檢測	以 SQL Injection 工具進行先期檢測		請檢視檢查報告進行防範
	針對網頁進行組合字串進行測試 (所有字串皆以「」括住，非字串的一部分，此外請注意注入點是否要求全部都要輸入資料)	測試字串如下各項：	
		以「'"」進行測試	執行結果異常，請檢查程式碼
		以「--'」進行測試	執行結果異常，請檢查程式碼
		以「'Or 1=1 --」及「'Or 1=2 --」進行測試	若前者執行結果正常及後者執行結果異常請檢查程式碼
		以「'Or 'x'='x' --」及「'Or 'x'='y' --」進行測試	若前者執行結果正常及後者執行結果異常請檢查程式碼
		以 'Having 1=1 -- 進行測試	執行結果異常，請檢查程式碼
系統防範作為	刪除 MS SQL Server 上不必要預存程序	xp_cmdshell	建議刪除
		xp_regaddmultistring	建議刪除
		xp_regdeletekey	建議刪除
		xp_regdeletevalue	建議刪除
		xp_regenumkeys	建議刪除
		xp_regenumvalues	建議刪除
		xp_regread	建議刪除
		xp_regremovemultistring	建議刪除
		xp_regwrite	建議刪除
		sp_OACreate	建議刪除
		sp_OADestroy	建議刪除
		sp_OAMethod	建議刪除
		sp_OASetProperty	建議刪除
		sp_OAGetErrorInfo	建議刪除
		sp_OAStop	建議刪除
	刪除 Pubs 及 NorthWind 資料庫		建議刪除
	架設網頁應用程式防火牆		無，建議架設

	以目錄伺服器進行存取網頁管控		無，建議以目錄伺服器進行管控
	架設 DMZ 分隔內部網路		無，建議建立
程式碼檢視	ASP	是否以 ADODB.Connection 物件直接 執行組合 QueryString 的 SQL 語法	是，建議以 ADODB.Command 配合預存程序執行查詢
	ASP.NET	是否直接以 SQLCommand 類別組合語法進行查詢	是，建議以 SQL Parameter 取得參數執行查詢
	兩者皆是	是否以 Try 語法進行程式撰寫	否，建議改寫
	兩者皆是	是否過濾 SQL 語法關鍵字，並將取得參數統一轉大小寫	否，建議過濾及改寫
	兩者皆是	是否在伺服器端程式碼檢查限制參數長度	否，建議改寫
	兩者皆是	是否刪除或屏蔽\%systemroot%\Help\iisHelp\common\500-100.asp	否，建議以統一錯誤訊息回應
	兩者皆是	是否以資料庫管理系統最高權限連接資料庫	是，建議不以最高權限帳戶連接

(資料來源：作者整理)

結論

提供線上互動式作業的網站，當成員身份進行認證與資料查詢作業流程時，皆採由用戶端輸入資料或查詢條件，經由程式藉以存取後端資料庫，並將執行結果回傳，呈現於用戶端網頁之上。由於 Web 應用程式的用戶群大，也容易成為攻擊的對象，而國軍目前資訊作業環境多以此類模式進行作業，所以同樣面臨資料隱碼攻擊威脅，而資料隱碼(SQL Injection)漏洞主要是人為因素造成的，因此，身為資訊人員，唯有以嚴謹的態度去檢視程式架構，及參考多方的防範建議，才能將潛在「資料隱碼」威脅降到最低。因此，作者於文中提供資料隱碼威脅檢查表供管理人員檢查索引之用，以期能提昇國軍網路整體安全。

參考資料

一、 Paul Litwin,Stop SQL Injection Attacks Before They Stop You, pp1-15.

二、 Sam M.S. NG,SQL Injection Protection by Variable Normalization of SQL Statement,2005.

三、 http://msdn.microsoft.com/Taiwan/sql/SQL_Injection.htm .

四、 財團法人台灣網路資訊中心，網路安全委員會第五次會議記錄，ttp://www.twnic.net.tw/member/sec5.htm .

五、 http://www.ettoday.com/2006/01/19/91-1896211.htm .

六、 http://www.ettoday.com/2002/04/22/399-1293697.htm .

七、 Microsoft TechNet 安全性摘要報，http://www.microsoft.com.twiwan/technet/security/advisory/954462.mspx..

八、 IBM X-Force 報告摘要，http://www.i-security.twlearn/sub_200810_4.asp。

九、 http://www.security-hacks.com/2007/05/18/top-15-free-sql-injection -scanners , http://www.microsoft.com/downloads/details.aspx?familyid=58a7c46e-a599-4fcb- 9ab4-a4334146b6ba&displaylang=en&tm .