

以網路協定為基礎的隱密通道-其威脅與防制

陳奕明

cym@mgt.ncu.edu.tw

中央大學資訊管理學系電腦網路實驗室

游啟勝

s0423035@cc.ncu.edu.tw

摘要

本文旨在介紹一種新的隱密通道 (Covert channel) 實作方式。文中先從文獻中對隱密通道的種類及衡量隱密通道的方法作一說明。再提出一個隱密通道的實作方式 – 以網路協定為基礎的隱密通道。這個實作方式是利用協定標準中未清楚定義的封包格式欄位來挾帶資料。我們將以一個網路上的木馬程式證明此實作方法的可行性，並對此木馬程式運作方式進行分析。最後，我們對以網路協定為基礎的隱密通道提出可行的防制方式。
關鍵詞：隱密通道、木馬程式溝通、網際網路協定、封包格式。

1. 前言

木馬程式 (Trojan horse) 的使用為網際網路安全帶來前所未有的威脅。根據警政署刑事局的調查，國內約有十分之一的主機被植入後門[1]。而在木馬程式的寫作中有三大重點，即木馬程式的植入方式、隱藏方式及溝通方式。植入方式為木馬程式應該如何引誘受害者在受害者電腦上執行木馬程式。隱藏方式則強調木馬程式如何在執行時，隱藏自己，使之不被使用者或偵測程式所發現。而溝通方式主要是探討木馬程式操控端及受控端要如何溝通。

在木馬程式的領域中，隱密通道最常用在木馬程式的操控端與受控端的溝通上。主要的兩大功能為：由木馬程式操控端傳送指令至受控端執行及將回應訊息由受控端回傳至操控端。新型的木馬程式為了避免被發現，漸漸使用更隱密的隱密通道作為傳輸指令及結果之用。因此，瞭解隱密通道的種類及發展趨勢，有助於中斷木馬程式操控端與受控端之間的通訊，藉此降低木馬程式對企業網路安全所造成的威脅與傷害。

本文共分六節。在第二節介紹隱密通道，包含隱密通道的定義、分類方法、木馬程式常用的隱密通道及衡量隱密通道的要素。第三節說明以網路協定為基礎的隱密通道的原理。第四節以網路上取得的木馬程式對以網路協定為基礎的隱密通道做可行性展示。第五節則對此類隱密通道提出可行的防制方式。最後，我們對以網路協定為基礎的隱密通道做一簡要的結論。

2. 隱密通道回顧

隱密通道一詞出現於 1985 年十二月，美國國防部公佈的 Trusted Computer Systems Evaluation Criteria 一文[2]，也就是俗稱的橘皮書。隱密通道在此文中指的是被電腦系統中的程序 (process) 用來傳送違反系統安全政策的傳輸通道。

根據 ISO/IEC 的定義[3]，任何用來傳送違反安全政策資料的傳輸通道都可稱為隱密通道。隱密通道基本上是一個傳輸資料的管道。它對企業所帶來最大的威脅即是造成機密資料的外洩。攻擊者可利用隱密通道來突破企業的安全措施將竊取到的企業內資料傳輸至外部。

2.1 隱密通道的分類

隱密通道可依其傳送的資料種類分為二類 [1]：(1) 操控指令通道、(2) 回應訊息通道。操控指令通道所傳送的資料是控制者傳送給受控端的指令；回應訊息通道傳送的資料則為受控端回傳的結果訊息。

而每一類還可分為主動式及被動式。主動式代表該方直接與另一方連結，被動式則表示以間接方法與另一方溝通。其結果可歸納成下表：

表 1：隱密通道的分類

回應訊息通道	操控指令通道	
	主動式	被動式
主動式	主從式架構連線	受控端主動發出反向連結
被動式	控制端直接發指令至受控端，受控端再將結果置於外部主機	控制指令及回應結果置於外部主機

(資料來源：[1])

目前在木馬程式中較常見的隱密通道其中幾種實作方式如下：

1. 直接連結
這是最常見的實作方式，通常是木馬程式受控者端以網路伺服器的方式存在，選擇一特定的通訊埠等待木馬程式操控端連結。連結之後再以此連線進行指令及結果的傳送。
此方式與一般網路程式運作方式無異，但多半會被防火牆阻擋。且操控者端與受控端直接連

線，容易被追蹤。受控端程式容易被偵測及留下連線記錄。

2. 以電子郵件挾帶資料
通常被用在受控端對攻擊者的溝通上，將所要傳送的資料置於電子郵件內容中，寄至特定電子郵件信箱帳號。由於目前有許多免費的電子郵件信箱，不需嚴格審核即可使用，對攻擊者來說，較直接連結方式不易被追蹤。
3. 以公眾留言板留下資料
可被用在操控端及控端的互相溝通上。木馬程式受控端可被設計由公眾留言板接收指令，也可於公眾留言板留下回應訊息。這個方式最大的優點是利用目前防火牆最容易開啟的 http 協定作為通訊之用。只要企業允許內部網路主機存取外部的 www 資源，此方法便可運作。
4. 以 IRC 接受指令及傳送結果
可被用在操控端及控端的互相溝通上。木馬程式受控端可以 client 形式連結至外部 IRC 伺服器，攻擊者便可即時控制木馬受控端行動。
5. 以 FTP 取得指令或上傳結果
可被用在操控端及控端的互相溝通上。木馬程式操控端及受控端以外部 FTP 伺服器作為中介站，進行指令及資料的交換。
6. 以特殊格式封包進行木馬程式之間的溝通
通常被用在攻擊者對受控端溝通上。攻擊者可利用預先定義好的網路封包格式來傳送指令至受害者端。這個方式也是我們本文所提以網路封包為基礎的隱密通道。在本文中會有詳細的介紹。
7. 以 ICQ 作為溝通的工作
ICQ 是一個即時的傳訊軟體，使用者可以利用它來即時傳送訊息、檔案等資料至另一台具有 ICQ 的電腦。隨著 ICQ 的盛行，木馬也開始利用 ICQ 訊息來傳送資料。

著名的木馬程式 Back Orifice 2000 (BO2K) [4]便可以伺服器附加程式 (plugin) [5]: Rattler、iICQ、gBot 使木馬程式受控端分別利用 E-Mail、ICQ 及 IRC 來傳送主機 IP address 的變化情形等資訊。

2.2 衡量隱密通道

要衡量一個隱密通道所使用的方法的好壞，我們可以從以下幾點來著手[6]：

1. 容錯率
該方法是否容易被其他設備更改，造成結果改變。例如：使用 TCP Sequence Number 的值當作指令編號，其結果可能因為防火牆、NAT 伺服器更改其內容而造成錯誤，因此，容錯率較低。相反的，使用 payload，因正常設備不會更改 payload 的內容，容錯率也就較高。
2. 頻寬

每個封包所能傳送的訊息量。例如：利用 TCP Sequence Number 來攜帶資料，每個封包的頻寬即為 4 bytes。頻寬乘於封包數即為挾帶資料量的最大值。

3. 偵測容易度
該方法是否容易被偵測到。

對攻擊者來說，偵測容易度可能是最重要的選擇要素。因為一旦使用的方法被偵測到後，大部份的隱密通道將會被中斷。但是這三個衡量要素的重要性並不是絕對的，而是要依實際應用環境及種類而定。對於操控指令通道來說，頻寬可能不是最重要的要素，攻擊者通常會為指令定義編號，藉此減少頻寬使用量。相反的，頻寬對於回應訊息通道可能就來得重要的多，一旦回應訊息太多，太少的頻寬會使回應的時間拉長，因而容易引起注意而被偵測到。

3. 以網路協定為基礎的隱密通道

由於任何可以傳送資料至外部的傳輸管道都有可能變成隱密通道。因此，隱密通道實作方法五花八門。常見的有利用網路協定中的 payload 挾帶傳送資料 (HTTP tunnel、IRC tunnel) 利用圖片進行資料隱藏...等。另外一個趨勢是利用網路協定標準無法對實作細節完全定義的缺失，利用協定格式各欄位的值或其值的排列組合來傳送資料。一旦攻擊者所使用的方法符合標準規定，偵測難度會大大提高。

由於網路協定的定義往往只針對協定的欄位格式及其欄位代表的意義作規範，而鮮少對欄位的值的範圍及產生方式等細節作明確的規範。因此，網路協定中可用作隱密通道的部份可分為以下幾種：

1. payload
這是最常被使用的方式。攻擊者在協定封包的 payload 放在指令或結果訊息傳給受控端。使用 payload 為傳輸媒介的隱密通道大部份會使用各類加密法來提高其隱密性。另外，這個方法頻寬通常也最高。
2. 未定義的欄位值
此方式利用協定標準未定義的欄位設定值來進行資料傳輸。由於標準未定義的欄位值在正常情況下並不會產生。因此，這個方法具有高容錯率，但因為破壞了協定行為，所以容易被偵測出來。
以 ICMP 為例，在 ICMP 封包標準中，封包 type 欄位值 1 或 2 並沒有於標準定義。因此，攻擊者可定義當木馬受控者端收到 type 欄位值為 1 的 ICMP 封包時，便取 code 欄位的值作為指令編號，再去執行預先定義好符合該編號的指令。
3. 正常範圍內的欄位值
這個方式使用標準定義的欄位範圍值來傳輸

資料。因為所使用的方式完全正常，因此最難被偵測到。但是由於可能有其他正常行為可能產生同樣的封包，因此容錯率也最差。但是可以使用較少使用的欄位值、範圍較大的欄位、或是以多個欄位排列組合來減少碰撞。目前已被人使用的幾個例子有：TCP Checksum [6]、IP Identification [7]、TCP Sequence Number [7] 欄位。這幾個欄位，標準上都沒有或是無法規定其範圍值及產生方式。以 TCP Checksum 來說，Checksum 被設計用來確保資料的正確性，其值根據封包內容值而有不同。也因此我們可以藉由給予設計過的內容值來產生特定的 Checksum。然後解讀 Checksum 來得知資料內容。使用 Checksum 的另一個好處因為其欄位範圍較大，其碰撞機率比使用其他欄位來得小。因而可以確保收到的大部份資料都是正確的。另外，也可再加上多個正常範圍內的欄位值排列組合來再降低碰撞率。

表 2 為利用隱密通道衡量要素來對三種實作方式做評估。

表 2：以網路協定為基礎的隱密通道實作方式評估

使用方法	衡量要素		
	容錯率	頻寬	偵測容易度
payload	佳	佳	中
未定義欄位	佳	中	易
正常欄位	差	差	難

(資料來源：本研究整理)

以 payload 來挾帶資料的隱密通道，由於頻寬較大，因此較適合用來當作資料量較大的訊息回應通道。而欄位值及其排列組合可挾帶的資料量有限，較適合被用來當作操控指令通道。攻擊者只要在木馬程式中定義好執行指令的種類，傳送指令時僅需傳輸指令編號，即可大大地減少資料量。一些只需傳回指令執行成功與否的訊息回應通道也可以使用欄位值來當作傳輸媒介。

目前已知被使用的協定有 ICMP、IP、TCP、HTTP。另外 SMTP、DNS 也是有可能發展的方向，因為這些協定往往較其他協定容易通過防火牆傳送至內部網路。

4. 攻擊實例

這一小節主要藉由網路取得一個以 ICMP 協定為溝通管道的木馬程式[8]來說明以網路協定來作為隱密通道的可行性。

此木馬程式以 ICMP Echo Reply 封包做為傳送指令的媒介，以封包 payload 來傳輸指令，而以 ICMP 檔頭的 Sequence Number 當作密碼作為檢查此封包是否為木馬操控端送來的命令封包之

用。圖 1 為 ICMP Reply 的封包格式及木馬所利用的欄位。

Type(0)	Code(0)	Checksum
Identification		Sequence Number (密碼)
payload (指令)		

圖 1：木馬程式 ICMP Reply 封包格式

整個木馬程式運作流程如圖 2 所示，說明如下：

1. 木馬程式操控端將要木馬程式執行的指令放至 ICMP Echo Reply 封包的 payload 中，並將事先定義好的密碼填至 ICMP 檔頭的 Sequence Number 欄位中，接著送出封包。
2. 木馬程式受控端收到 ICMP Echo Reply 後，檢驗密碼是否正確。接著由 payload 取出指令，並顯示於螢幕上。

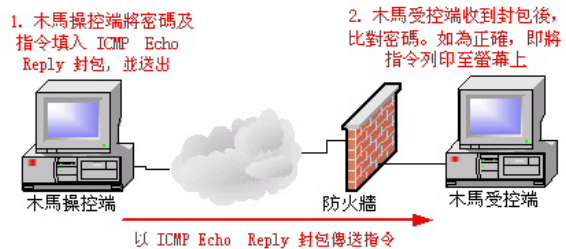


圖 2：木馬程式架構圖

木馬程式執行畫面如下：

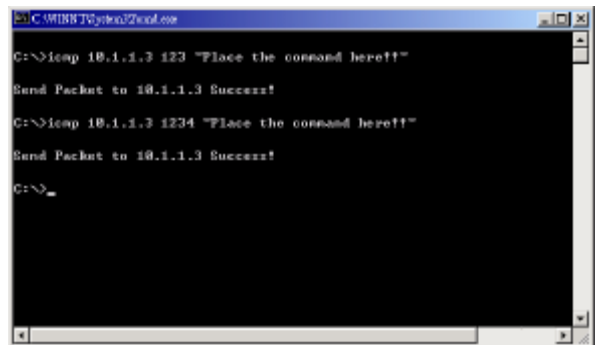


圖 3：木馬操控端執行畫面

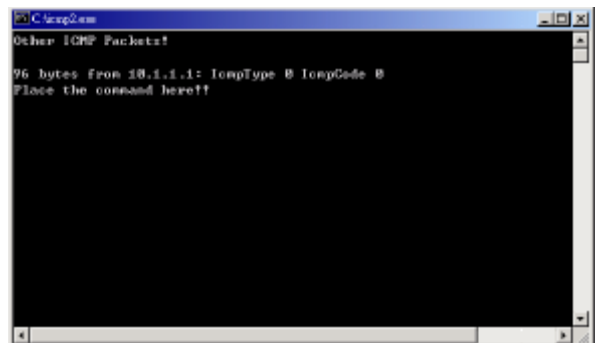


圖 4：木馬受控端執行畫面

圖 3 及圖 4 分別為木馬操控端與受控端的執行畫面。木馬操控端可輸入木馬受控端 IP、密碼、及欲執行指令。第一行表示當密碼為錯誤時，木馬受控端會顯示「Other ICMP Packets!」的訊息。當密碼正確無誤時，木馬受控端會列出此 ICMP 命令封包的來源位置、Type、Code 及 payload 所藏的指令。目前此木馬程式並不會執行指令及回傳任何結果。但是可以加以修改即可執行所接受的指令。值得注意此方法可以避過及穿透部分入侵偵測系統及防火牆的偵測及阻擋。

使用 ICMP 的優點有以下數點：

1. 部份企業會基於便利的原因開啟 ICMP 協定的通訊，以供內部網路主機對外部主機執行 ping 指令。相較於 TCP 及 UDP，ICMP 較不易引起注意。
2. 部份防火牆或入侵偵測系統對 ICMP 協定的功能較 TCP 及 UDP 不完整。舉例來說，許多低階的防火牆無法設定阻擋 ICMP 協定、或是無法對 ICMP 協定的各個種類個別設定允許或拒絕通過。另外，許多防火牆未具備 ICMP Stateful 的功能，因此攻擊者可偽造假的回應訊息通過防火牆。

5. 防制策略

對於以網路協定為基礎的隱密通道，主要的防制策略著重在偵測及阻斷通訊兩方面。而阻斷通訊又比偵測來得重要。

我們歸納出三種防制策略：阻擋不必要的網路連線、阻止狀態不正常的連線、阻擋及偵測不正常的協定封包。

5.1 阻擋不必要的網路連線

正確的防火牆設定可以阻擋大部份木馬程式建立的連線。最重要的原則是阻擋所有不應使用的通訊協定。只開啟網路運作必要的協定。一些傳遞網路資訊而對網路正常運作沒有影響的協定，可依需要的安全性決定開啟或阻擋。另外，內部網路應儘量避免與外部網路直接通訊，拒絕所有外部往內部網路的連線。針對 ICMP 協定的防火牆設定原則可參考 ICMP Usage in Scanning 一文[9]第九段。

5.2 阻止不正常的連線狀態

防火牆及入侵偵測系統應能對連線狀態不正常的封包加以阻擋及警告。舉例來說，防火牆及入侵偵測設備應具有 stateful TCP/UDP/ICMP 的能力，以阻擋攻擊者客製的網路封包。例如沒有 request 卻忽然出現的 reply 封包，或是一個 request 有多個 reply 回來，都有可能是狀態異常的連線，應給予警告及處理。圖 5 為 BlackICE Defender 偵測到異常 ICMP Echo Reply 時所出現

的畫面。另外，常用的舊版 Checkpoint Firewall-1 並不具有 stateful ICMP 的檢查功能，對應的加強設定可以由網路上找到[10]。

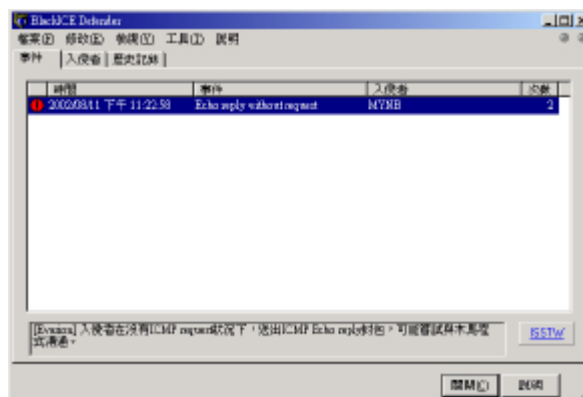


圖 5：入侵偵測系統偵測到不正常的連線狀態

5.3 阻擋及偵測不正常的協定封包

防火牆及入侵偵測系統應能阻擋及偵測單一不正常封包。

所謂不正常的封包可分為：

1. 未定義的協定欄位值及欄位組合
每個封包應檢查各個欄位的值及欄位組合是否已於協定標準中定義。未定義的協定欄位值及欄位組合最常被用來傳送指令編號。根本的解決方法是依協定標準建立正向列表，找出所有欄位值允許的設定值範圍，只有在列表上的欄位值組合才允許通過，一旦發現異常就阻擋封包。一個簡單的例子是 Sys-Security 對 snort 所寫的加強規則檔[11]，它可以使 snort 偵測到 ICMP 中所有 Type 及 Code 欄位組合未定義的封包。ICMP 所有 Type 及 Code 欄位定義值可參考 ICMP TYPE NUMBERS 一文[12]。

其實作方式簡單介紹如下：

根據協定標準，ICMP 所有定義的 Type、Code 欄位值與其用途如表 3：

表 3：ICMP Type 及 Code 欄位列表

Type	Code	用途
0	0	Echo Reply
3	Destination Unreachable	
	0	Network Unreachable
	1	Host Unreachable
	2	Protocol Unreachable
	3	Protocol Unreachable
	4	Fragmentation Needed and DF bit was set
	5	Source Route Failed
	6	Destination Network Unknown
	7	Destination Network Unknown
	8	Source Host Isolated
	9	Communication with Destination Network is Administratively

		Prohibited
10		Communication with Destination Host is Administratively Prohibited
11		Network Unreachable for Type of Service
12		Host Unreachable for Type of Service
13		Communication Administratively Prohibited
14		Host Precedence Violation
15		Precedence Cutoff in effect
4	0	Source Quench
5	Redirect	
5	0	for Network or Subnet
	1	for Host
	2	for TOS and Network
	3	for TOS and Host
6	0	Alternate Host Address
8	0	Echo Request
9	Router Advertisement	
	0	Normal router advertisement
	16	Does not route common traffic
10	0	Router Selection
11	Time Exceeded	
	0	Time-To-Live Exceeded in Transit
	1	Fragment Reassembly Time Exceeded
12	Parameter Problem	
	0	Pointer indicates the error
	1	Missing a Required Option
	2	Bad Length
13	0	Timestamp Request
14	0	Timestamp Reply
15	0	Information Request
16	0	Information Reply
17	0	Address Mask Request
18	0	Address Mask Reply
30	0	Traceroute
31	0	Datagram Conversion Error
32	0	Mobile Host Redirect
33	0	IPV6 Where-Are-You
34	0	IPV6 I-Am-Here
35	0	Mobile Registration Request
36	0	Mobile Registration Reply
39	0	SKIP
40	Photuris	
	0	Bad SPI
	1	Authentication Failed
	2	Decompression Failed
	3	Decryption Failed
	4	Need Authentication
	5	Need Authorization

(資料來源：[12])

Type 及 Code 欄位都有一個位元組，因此值的範圍可由 0 至 255 由此可看出有定義的值只佔了全部範圍的一小部份，我們可以利用 snort 來幫助我們找出沒有定義的欄位值封包。根據上

表，以 Type 為 12 為例，我們可看出 Code 值只有定義為 0、1、2。所以我們可以寫出以下的 snort 規則[11]：

```

alert icmp any any -> any any (msg:"ICMP
Parameter Problem Code 0 (Pointer indicates the
error)"; itype: 12; icode: 0;)
alert icmp any any -> any any (msg:"ICMP
Parameter Problem Code 1 (Missing a Required
Option)"; itype: 12; icode: 1;)
alert icmp any any -> any any (msg:"ICMP
Parameter Problem Code 2 (Bad Length)"; itype:
12; icode: 2;)
alert icmp any any -> any any (msg:"ICMP
Parameter Problem (Undefined Code!); itype:
12;)

```

前三條規則是針對正常 Code: 0、1、2 作定義，最後一條規則則抓出其他 Code 沒有定義且 Type 為 12 的封包。其餘的規則可以依此類推。

在此規則檔的最後應加上：

```

alert icmp any any -> any any (msg:"ICMP
Unknown Type";)

```

來抓到在之前規則檔沒定義到的 Type 值封包。

- 異常的封包大小及 payload 內容
雖然有些欄位的內容在標準中並沒有規定。但是部份協定欄位在實作上會有相同的格式（例如：固定的封包大小及 payload。）針於此類協定，應於平時蒐集其特徵，之後對其封包大小及內容進行列表比對，找出異常的封包。舉例來說：在同種作業系統下的 ping 程式所產生的 ICMP Echo Request/Reply 的封包大小及 payload 內容都是相同的。
圖 6 為 windows 系列作業系統 ICMP Request 的固定 payload：

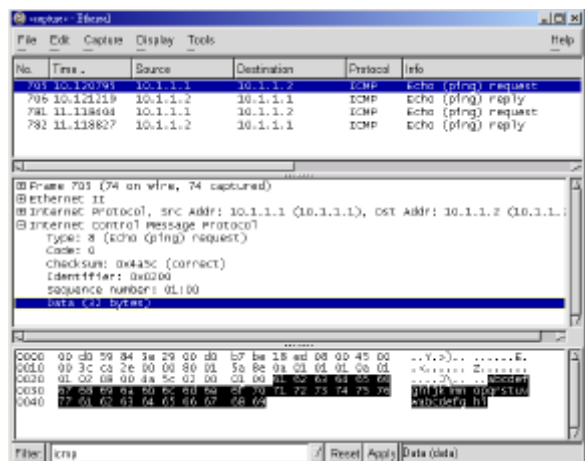


圖 6：windows 系列作業系統 ICMP request payload

而攻擊實例中的木馬程式所使用的 ICMP Echo Reply 封包格式(圖 7), 可明顯看出其封包總大小與封包 payload 不同於正常 ping 程式所產生的封包(圖 6)。因此, 如果平常有對這些標準沒有定義的欄位特徵進行蒐集, 很容易可以偵測到異常狀況的發生。

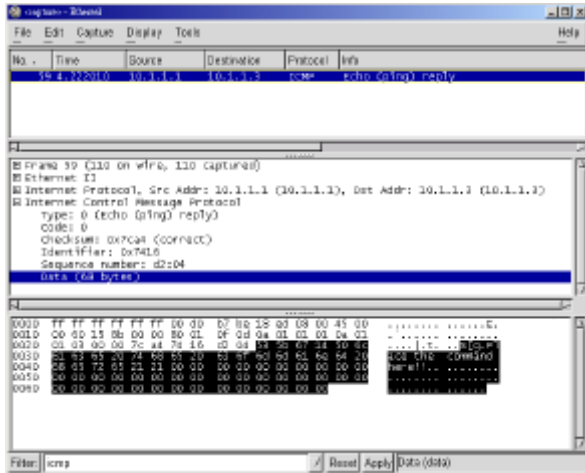


圖 7：不正常的 ICMP Echo Reply 封包

5.4 以 application proxy 進行身份認證並檢查連線封包

所有允許對外的協定, 應透過 application proxy 先以身份認證, 再檢查為正確的協定格式封包才允許通過, 避免以協定作為偽裝的隱密通道。

要防止以上三種以網路協定為基礎的隱密通道, 必須對各類協定標準細節相當熟悉, 根據標準找出哪裡是正常的。至於標準沒有明確規定的部份, 可以依使用系統的種類建立正向列表加以比對, 藉此找出看似正常, 實為異常的連線。

6. 結論

隱密通道最主要功用就是如何在不被發現的情形下進行資料的交換。隱密通道已被木馬程式用來當作操控端與受控端傳送指令及回應結果訊息的管道。愈來愈隱密的隱密通道實作方式及使用正常協定行為的趨勢, 使得偵測隱密通道日益困難。唯有利用更多的方法加以判斷及更強大的設備來找出所有可能的異常狀況, 才能偵測及阻擋隱密通道的通訊。雖然無法完全中斷隱密通道所造成的威脅, 但是還是可以減少通訊的頻寬, 拉長攻擊所需的時間, 避免更多的資訊外洩及增加偵測的機率。

7. 參考文獻

- [1] 黃世昆, 防止攻擊跳板主機的安全管理策略, 中央警察大學 2000 第二屆網際空間: 資訊、法律與社會研討會, 中華民國 89 年 12 月。
- [2] DoD 5200.28-STD, Department of Defense Trusted Computer System Evaluation Criteria, U.S., 1985.
- [3] ISO/IEC 2382-8, Information Technology - Vocabulary: Control, integrity, and security, 1998.
- [4] Back Orifice 2000 Web Site, <http://bo2k.sourceforge.net>.
- [5] Back Orifice 2000 Server Enhancement Plugins Web Page, <http://bo2k.sourceforge.net/software/bo2k10.html>.
- [6] Drew Hintz, Covert Channel, DEF CON 10.
- [7] Craig H. Rowland, Covert Channel in the TCP/IP Protocol Suite.
- [8] 揭開木馬的神秘面紗 (三), <http://www.yesky.com/20010525/181452.shtml>.
- [9] Ofir Arkin, ICMP Usage in Scanning, http://www.sys-security.com/archive/papers/ICMP_Scanning_v3.0.pdf.
- [10] Checkpoint FW-1, Version 4.0 and 4.1 ,Using the INSPECT language to implement stateful ICMP messages, <http://www.yassp.org/fw1/icmp.html>
- [11] Advanced ICMP Basic Rule Base on Snort , http://www.sys-security.com/archive/snort/icmp_rules/ICMP_basic_plus.
- [12] ICMP TYPE NUMBERS, <http://www.iana.org/assignments/icmp-parameters>