基於網路訊務動態基線分析之網路蠕蟲偵測機制

陳彥錚 張嫄媗 張家瑋 王士豪
國立暨南國際大學 資訊管理學系

摘　　要

許多新型網路蠕蟲主要利用作業系統或伺服器漏洞所設計，由於網路蠕蟲使用網際網路作為其活躍的舞台與傳播的媒介，因此往往能夠引發大規模氾濫，除了影響電腦正常運作，也直接耗用網路資源，形成另一種形式的網路阻斷服務攻擊，造成網路管理上很大的困擾。目前最常被採用的網路蠕蟲偵測方式是針對每一種蠕蟲已知的特徵，從網路訊務找出符合特徵的資料。使用這種蠕蟲偵測方式，我們必須事先知道每一蠕蟲的特徵，並為每一蠕蟲針對其特徵開發專屬的蠕蟲偵測程式。由於蠕蟲種類越來越多，對網管人員而言，一蠕蟲一偵測程式的管理方式延展性差，蠕蟲偵測工作漸成為網管人員沉重的負擔。另一方面，新型蠕蟲出現初期，其特徵尚未被了解，專屬的蠕蟲偵測程式付之闕如，這是使用蠕蟲特徵偵測另一個缺點。本論文提出一個利用網路訊務動態基線分析的蠕蟲偵測通用架構，利用每五分鐘的各通訊埠訊務基線之建立與分析，便可篩選出符合蠕蟲行為的訊務資料。此架構無須得知蠕蟲特徵以及開發專屬蠕蟲偵測程式，只利用 NetFlow 網路訊務監測工具，以其通訊埠、時間、流量三個維度所建立的訊務基線過濾與基線偏離的訊務，進而找出可能的蠕蟲及受感染的節點。使用本論文所提蠕蟲偵測機制，網管人員無須為每一蠕蟲尋求或自行開發專屬偵測程式，且在新型蠕蟲開始感染發作之際，此機制便能發揮作用，提供網管人員重要的訊務異常資料。我們將以最近爆發的殺手蠕蟲(Worm Sasser)為實驗測試對象，證明本論文所提蠕蟲偵測機制，能夠有效協助網管人員偵測網路蠕蟲。

關鍵詞：網路安全、網路蠕蟲、基線分析、網路管理
投稿領域：網際網路技術 - 資訊安全、網路管理

聯絡人：　陳彥錚 助理教授
國立暨南國際大學 資訊管理學系
南投縣 545 **埔里鎮大學路** 303 **號**
TEL: (049) 2910960 ext. 4849
FAX: (049) 2915205
E-mail: ycchen@ncnu.edu.tw

1. 前言

近年來網際網路蓬勃發展，隨著網路間相互連接，使網際網路觸角無遠弗屆，同時人們也對網路依存度日深。然而在現今日益複雜的計算環境之下卻也充斥著各種作業系統、網路服務、及應用程式的漏洞。別有居心的駭客便利用這些漏洞發展惡意程式企圖破壞或竊取電腦資源，而網際網路本身成為大量迅速散佈這些惡意程式的最佳途徑。根據電腦網路危機處理暨協調中心(CERT/CC)之統計發現[2]，近年來電腦系統漏洞被發現的數目連年成長，而同時網路安全事件之回報數量也隨之逐年攀升，這顯示依附系統漏洞而生的惡意程式，對網際網路安全威脅日趨嚴重。這些利用系統漏洞所設計並利用網路主動散佈感染的惡意程式，我們稱之網路蠕蟲(Network Worm)[10][11]，受網路蠕蟲感染的電腦，也會扮演傳播者主動感染網路上其他節點，這樣反覆不斷地複製感染行為，常產生大量的封包傳送，造成網路頻寬資源耗用，形同網路的阻斷服務攻擊(Denial of Service Attack, DoS)[26]，進而影響到的各項日常的作業並造成程度相當大的損失。

以近年來發生的蠕蟲感染事件為例，微軟於 2002 年 7 月發佈其資料庫軟體 MS SQL Server 2000 漏洞及其修補程式[20]，在六個月後的西元 2003 年 1 月 25 號開始，一隻針對 MS SQL 2000 資料庫軟體漏洞的網路蠕蟲 SQL Slammer 開始在網路上蔓延。這個蠕蟲造成網際網路上的流量暴增，數以百萬計的 UDP/IP 偵測封包大量傳送，試圖利用 SQL Server 解析服務溢位漏洞(SQL Server Resolution service buffer overflow, CVE CAN-2002-0649)進行入侵。SQL Slammer 蠕蟲本身只有 376 位元組，比紅色代碼(Code Red)4096 位元組 及 Nimda 61440 位元組小很多[27]。然而電腦一旦被感染，此蠕蟲就會再隨機選取目標主機進行進一步散播，透過自我繁殖使得電腦間無法通訊，結果造成全球網路大塞車，耗盡了網路的資源。另一蠕蟲例子是發生於 2003 年 8 月的疾風蠕蟲(Worm Blaster)，它攻擊一個才發現 26 天的微軟作業系統漏洞[20]，同樣造成網路擁塞，嚴重影響網路頻寬。無獨有偶，在 2004 年的 5 月初，一隻名為「殺手」(Worm Sasser)的蠕蟲病毒同樣讓受感染的網際網路無法正常運作。令人驚訝的是，殺手蠕蟲之誕生僅在它所利用的漏洞被公佈後 18 天。我們不難發現，這些曾造成網際網路巨大傷害的網路蠕蟲都依循以下相同的氾濫模式：

(1). 電腦系統廠商發佈漏洞消息。
(2). 針對此漏洞所發展的蠕蟲誕生。
(3). 蠕蟲開始於網路上自我複製散播。
(4). 網路流量激增造成擁塞，使用者及網管人員發覺網路異常。

因蠕蟲感染造成網路異常後，一開始網管人員懷疑網路設備或線路本身造成網路障礙，等這些因素被排除後，才會開始更進一步監看網路訊務詳細資料，從中發現不尋常的大量訊務資料，進而懷疑新型蠕蟲的感染。新型蠕蟲之感染蔓延，經專家學者分析找出蠕蟲在網路訊務資料中所呈現的特徵後，網管人員才能針對其特徵使用自動化的蠕蟲偵測程式找出感染源，進而實施斷網及漏洞修補的工作。在網管實務上，通常網管人員必須經歷從蠕蟲出現至找到有效防治之道的艱辛過程，這顯示目前網路管理工作在蠕蟲偵測上急需一個有效的監測管理機制。

欲監測網路上蠕蟲造成的異常行為，傳統上，通常使用嗅探器(Sniffer)來監聽過往的每一個網路封包[35]，或是觀察蠕蟲的行為特徵，再根據特徵做出針對單一種蠕蟲攻擊的偵測系統[21]，因為各種蠕蟲型態不一、特徵不定、表現出來的攻擊行為也就各有所異。然而，針對特定蠕蟲特徵所進行的監測往往會耗用很大的電腦運算資源，使得傳統監測主機在攻擊開始就因為負載過重成為第一個損壞的犧牲者，再者網管人員要針對每一種蠕蟲病毒個別去開發一套偵測系統，在未來蠕蟲越來越多的網路環境下，更是增添網路管理的複雜度與困難度。本論文研究目的並不在於如何針對某一蠕蟲找出最精確的

偵測技術，而是著眼於如何提供一個較為可行而通用的蠕蟲偵測機制，亦即不管蠕蟲為何，網管人員可利用此偵測通用機制發覺符合蠕蟲行為的訊務。此蠕蟲偵測通用機制基於平時對網路訊務的基線(Baseline)分析，因此能夠即時查覺訊務異常變動，進而找出蠕蟲。使用網路訊務基線分析偵測蠕蟲，是因為網路蠕蟲發作蔓延時，網路流量通常會有變高的情形，因此藉由網路流量的監測可能可以早期發現網路蠕蟲的行蹤。然而使用流量變化偵測蠕蟲，必須注意的是網路訊務實際是由眾多使用者於網路上各種行為所產生訊務的總和，在網路使用上因為不同網路服務的提供、網路應用不同的行為模式，以及使用者使用網路的地點與時間的差異性，在實務上我們很難界定流量正常與異常。不過在一個成員具有集體相似行為的群體所在之網路環境，基本上其網路訊務的變化較具規率性，或有明顯的網路流量尖離峰分布情況。因此，藉由平時網路訊務基線之建立，我們可以更精確地了解網路上某一位置在某一時段的正常流量。當蠕蟲感染網路時，網路整體行為一異於尋常便會反應於流量的變化，由流量偏離基線的情況，我們可以較清楚地判斷是否有蠕蟲發生。從過去幾個蠕蟲引起的重大網路安全事件，蠕蟲發作蔓延時，網路流量確實發生變化，我們幾可確定使用網路訊務基線分析定可發現蠕蟲所引起的流量變化。然而相反地，我們使用基線分析發現訊務異常，並不能就直接判定是否有蠕蟲出現，因為，流量發生異常並不見得是由蠕蟲所引起。為降低因其他因素所造成流量異常導致蠕蟲偵測機制無法正確判斷，我們所提蠕蟲偵測機制所使用的訊務基線分析，將個別針對每一個蠕蟲所利用的漏洞之通訊埠及通訊協定建立訊務基線，而不是以網路整體訊務基線判斷蠕蟲之存在。例如，SQL Slammer 蠕蟲使用 UDP 1434 通訊埠，我們平時針對 UDP 1434 通訊埠網路流量建立基線資料，了解在所監測的網路上平常 SQL Server 通訊情形，因此一旦 UDP 1434 通訊量異常，該蠕蟲偵測機制便幾可斷定網路訊務異常是由 SQL Slammer 蠕蟲所引起，因為目前並沒有其他理由能解釋為何 UDP 1434 通訊量會異於尋常。使用個別通訊埠建立基線確實可以增進蠕蟲偵測的正確性，然而基於偵測系統效能考量，我們無法將 65536 (2^{16})通訊埠號空間內的每一通訊埠建立個別的訊務基線資料，事實上大部份的通訊埠並未有漏洞被發現，沒有必要為這些通訊埠建立基線。使用本論文所提的蠕蟲偵測機制，可只針對已知蠕蟲所使用的通訊埠訊務建立基線分析，更可以採更積極的預防式(Proactive)網管觀念，在電腦系統廠商發佈漏洞消息時，即把漏洞所在的通訊埠新增至蠕蟲偵測機制中，一旦新型蠕蟲被製造出來開始危害網路時，我們已有相關的通訊埠訊務基線資料可以來偵測新型蠕蟲。

2. 相關文獻探討

本節我們將先探討網路蠕蟲之發展及分類，以期了解蠕蟲典型的行為模式，其次我們將探討幾種與蠕蟲偵測相關之技術與管理工具，包括防火牆、入侵偵測系統、以及網路訊務監測工具。

2.1. 網路蠕蟲

蠕蟲(Worm)指的是一種能在機器與機器間傳播散布的程式，而能夠藉由網路自我複製、自我感染而危害網路的惡意程式則稱為網路蠕蟲(Internet Worm)，本論文所指蠕蟲，實為網路蠕蟲。第一隻蠕蟲是由全錄(Xerox) 帕洛亞托研究中心(PARC)的兩位研究員於 1980 年寫出[11]，其名取自於 John Brunner 的 1975 年科幻小說(The Shockwave Rider)中，這支原始的蠕蟲程式原只是希望用來在分散式的環境中，能在電腦間遊走並做些有幫助的工作；然而時至今日，網路蠕蟲一詞更幾乎被用來泛指那些未經使用者所允許，而經由網路自我傳播複製的「惡意蠕蟲」，例如 2003 年 1 月發生的 SQL Slammer 蠕蟲、2003 年 8 月發生的 Worm Blaster 蠕蟲、以及 2004 年 5 月所發生的 Worm Sasser 蠕蟲等網路蠕蟲的氾濫，有些造成系統被濫用，有些則是造成作業系統重複的重新開機，但共同的行為則是讓網路的流量突然間爆漲，且持續不斷進行大量的網路掃瞄動

作，形成對網路使用量的阻斷攻擊。多數網路蠕蟲具有下列特徵[4][17]：

(1). 使用快速掃瞄機制找尋下一個目標。
(2). 以已知的漏洞做為網路攻擊的起點。
(3). 通常都能夠對網路上的漏洞作感染與影響。
(4). 利用平行與多執行緒的方式感染。
(5). 通常體積很小使之能夠迅速感染。

近年來網路蠕蟲之發展，變種速度快、感染方式多樣化，我們可依網路蠕蟲之傳播途徑簡單區分為三類[9] [15] [16]，有些蠕蟲甚至同時具有多重傳播途徑，例如：Worm Nimda，以下簡介三類網路蠕蟲：

(1). 使用 E-Mail 及用戶端軟體傳播之網路蠕蟲：

此類蠕蟲是一些惡意程式碼專為收集使用者的個人資料以及通訊錄資料，使用 E-Mail 的方式將自己複製傳播；此外，有一些則利用即時聊天應用程式傳播以及點對點傳輸軟體，例如 1999 年的 Mellisa、ExploreZip、2001 年的思砍病毒，以及 2003 年的 Bibro 等[9]。

(2). 利用 Windows 檔案分享傳播之網路蠕蟲：

利用 Windows 檔案分享功能的蠕蟲又稱為 SMB 類蠕蟲。Samba 是 UNIX 終端上模仿視窗系統中網路文件共用的伺服器軟體。運行 Samba 能夠使 UNIX 終端成為與視窗系統相容的文件伺服器及列印伺服器，由於 Samba 各版本大多存在緩衝區溢出的安全漏洞，因此也往往被蠕蟲所利用，SMB 類的蠕蟲除了利用 Samba 伺服器漏洞更利用網路上文件共享的讀寫權限未設定完善的缺點，加以自我複製感染，例如 1999 年 ExploreZip 與 2001 年的 Nimda 都是會透過檔案分享作為傳播途徑。

(3). 利用 TCP/IP 通訊協定傳播之網路蠕蟲：

此類蠕蟲為典型傳統式網路蠕蟲，為本論文所要探討的重點，利用 TCP/IP 通訊協定傳播之網路蠕蟲能夠直接利用作業系統以及應用程式的缺失，在網路上以 TCP/IP 通訊協定感染複製傳播，不必經由使用者同意或是人為操縱便可進行攻擊動作，例如 2001 年的 Namda、2003 年的 SQL Slammer 蠕蟲與 Worm Blaster 蠕蟲，以及 2004 年的 Worm Sasser 蠕蟲，都是屬於此類蠕蟲。通常此類蠕蟲會使用固定的 TCP 或 UDP 通訊埠進行感染。

2.2. 防火牆(Firewall)

防制蠕蟲感染較簡單的方法是使用防火牆，防火牆主要用來將企業內部網路隔絕於外界的網際網路，以保護內部網路防止遭外來不正常的入侵與攻擊。由於蠕蟲感染行為類似網路入侵行為，目前各單位組織也大多有架設防火牆，因此我們可以考慮使用現有的防火牆來防制蠕蟲感染。防火牆依其對網路傳輸的處理方式不同，包括以下四類，分別提供不同程度的安全防護與彈性：

(1). 傳統網路層防火牆：路由器(Router)

藉由控制網路層的路由功能，我們可以依主機位址決定是否轉送封包來保護內部網路。單獨使用路由器並不能提供完整的安全防護，且缺乏完善防火牆所提供的彈性和功能。

(2). 封包過濾型型防火牆(Packet Filtering)

封包過濾是一種很簡單的防火牆。封包過濾常與路由器結合，且大部分的主要廠商都把封包過濾作為內定的組態。這種防火牆會檢查封包的目的地和來源的 IP 位址、

TCP/UDP 埠，並根據使用者設定的簡單規則來決定是否接受或拒絕封包。

(3). 狀態檢視型防火牆(Stateful Inspection)

狀態型封包防火牆(又稱智慧型封包過濾)使用與封包過濾類似的方法來控制網路傳輸，但會進一步地檢查資料封包流的內容，而不只是單純地過濾封包而已。狀態檢視封包防火牆根據封包的來源和目的地 IP 位址、通訊埠號及所要求的服務來作判斷過濾。這種防火牆之所以稱作「狀態型」的原因是它們會記住之前的連線狀態，目的是在記憶體中建立每一個資料流中封包的前後關聯。防火牆會根據此前後關聯來檢查每一個新收到的封包，並判斷此封包是新連線或是現有連線的延續。如果是後者，防火牆所進行的檢查動作會比對新連線的檢查少上許多。

(4). 應用層代理程式(Application Proxy)

位於應用層的代理程式是在防火牆上執行的一種軟體，能夠模擬網路連線的來源和目的地兩端。每台電腦跟其他電腦的網路傳輸都必須通過此代理程式，進行資料檢查並檢查連線的合法性，如此一來在檢查資料的過程中能夠有效地將內部網路和外界隔開。代理程式會檢查用戶端電腦送過來的資料，並判斷是否要轉送出去或是丟棄。

由於防火牆通常由管理者手動設定一組通用的過濾規則來決定封包是否轉送，對於蠕蟲防制上較缺乏彈性，通常只能針對已經發生的蠕蟲感染進行阻檔的動作，嚴格而言，防火牆是防止已知蠕蟲擴散的工具，而非蠕蟲偵測工具。另外，由於一般防火牆也會將經過防火牆的封包相關資訊記錄於防火牆的日誌(log)中，從日誌的分析也可了解詳細的訊務資訊，進而從中發現惡意入侵或蠕蟲散布的蹤跡[29]。

2.3. 入侵偵測系統(Intrusion Detection System, IDS)

IDS 專為偵測網路上惡意行為[18][32]之系統，其廣泛使用的三種技術為特徵比對偵測、異常行為偵測、以及異常通訊協定偵測[28]，茲分別說明如下：

(1). 特徵比對偵測(Signature Based Detection)

商用 IDS 所常採用的入侵偵測技術是檢查網路訊務中是否有特定攻擊特徵。亦即針對每一種型態的入侵行為，IDS 廠商必須事先了解其攻擊特徵才能偵測到它，因此這些攻擊必須是已知的。採用特徵比對偵測方式的 IDS 通常需要一個龐大的攻擊特徵資料庫，IDS 會試圖將每一個封包與資料庫中的所有攻擊特徵作比較，以找出符合特徵的封包。由於特徵資料庫龐大，當網路流量增加時，IDS 的偵測器無法逐一檢查每個封包，因此某些封包會被迫丟棄，導致部份入侵無法被查覺。目前大多數的 IDS 偵測器最多只能偵測每秒 60 Mb 的流量。在更快的資料傳輸速度時，IDS 的偵測率明顯地增降低。另一個使用特徵比對的已知問題是：IDS 廠商需花一段時間來辨識新的攻擊、建立攻擊特徵以及發佈更新。例如 Code Red 及 Nimda 攻擊[6] [7]，初期 IDS 並無法偵測出來。

(2). 異常行為偵測(Behavioral Anomaly Detection)

另一個較不普遍的入侵偵測方式是偵測統計上異常的行為。異常行為的偵測統計架構是以某些系統的統計值為基準，或者持續追蹤系統的行為模式。透過偵測這些模式的改變以偵測出攻擊。偵測的方式包括過度、非正常時間的使用系統，以及使用者程序造成的系統呼叫改變等[5]。這種解決方案的優勢是：它無須深入地瞭解導致異常背後的原因，就能偵測到異常行為；但是，合法的系統使用亦會引發異常行為，而導致大量的誤報。

(3). 異常通訊協定偵測(Protocol Anomaly Detection)

異常通訊協定偵測通常在應用層通訊協定執行。它關注的是通訊的結構與內容。許多攻擊會針對應用層通訊協定所設計，例如：Telnet、 HTTP、 RPC、SMTP 及 Rlogin 等

通訊協定。當直接在偵測器裡制定通訊協定的規則時，就很容易辨識違反規則的通訊流量，例如非預期的資料、額外的字元以及不合法的字元。這解釋為什麼某些攻擊可以被辨識出來。例如異常通訊協定偵測型的 IDS 可以偵測 Code Red，因為它們根據 RFC 規範精確地檢查 HTTP 通訊之內容，而 Code Red 攻擊違反 HTTP 通訊埠的標準，因為它使用 GET 指令以要求在感染的伺服器上放置並執行惡意程式碼。

2.4. 網路訊務監測工具—NetFlow

網管人員欲分析網路訊務內容，除了使用標準的 SNMP(Simple Network Management Protocol)[19]網管通訊協定至網路設備讀取網管資訊，最常使用的網路訊務監測工具便是 NetFlow [3][22][24][25]。NetFlow 為 Cisco 公司所發展，主要功能是將所有經過 Cisco 路由器的封包標頭資訊進行彙整統計，提供以訊務流(Flow)形式的詳細訊務傳輸記錄。使用 NetFlow 的網路訊務記錄具固定資料格式，由於廣被採用，漸成為共通的網管標準，並使網管人員得以依 NetFlow 標準自行發展所需的訊務分析與統計應用。另外，NetFlow 所記錄訊務的資訊並非逐一完整的封包記錄，而是經過整理的訊務流(Flow)資訊，資料量遠比原始通訊資料少。NetFlow 訊務記錄中的每一筆記錄記載一個訊務流，一筆訊務流代表從相同來源 IP 位址及通訊埠到相同目的 IP 位址及通訊埠的單向封包序(unidirectional packet sequence) [23]。NetFlow 判斷一訊務流的起始與終止，分為以下三種情況：

(1). 基於 TCP 之訊務流(TCP based Flows)：當一串 TCP 封包序的旗標欄位(Flag)出現了 FIN(Finish)或是 RST(Reset)時，表示結束一次的傳輸，也表示一筆訊務的結束。

(2). 基於時限之訊務流(Timeout based Flows)：當一個封包序列的最後一個封包到達後，預設 30 秒之內未再有新的封包傳遞，便自動結束此訊務，作為一筆訊務。

(3). (Time excess Flows)：當一傳封包序列連續傳遞，超過 30 分鐘未結束，便將此訊務自動的結束。

表 1 顯示 NetFlow 每一筆網訊務所包含的資訊，表 1 的例子顯示 4 月 5 日晚上 23:53，IP 位址為 10.10.22.163 的主機透過路由器之編號為 0 的介面卡，使用 UDP (17 為 UDP 的協定號碼)之 138 通訊埠，經由編號為 0 的介面卡與 IP 位址為 10.10.22.225 的主機建立連線，並產生封包大小為 1、流量為 229 位元組之訊務流。

表 1 NetFlow 單筆訊務流記錄

NetFlow 欄位	數值
Start time	0405.23:53:59.979
End time	0405.23:53:59.979
Source If	0
Source IP	10.10.22.163
Protocol	17
Source Port	138
Destination If	0
Destination IP	10.10.22.225
Destination Port	138
Flag	0
Packets	1
Octets	229

從此例子我們可以了解 NetFlow 記錄通訊內容非常詳盡，然而 NetFlow 以訊務流方式記錄訊務也不至於使整體訊務記錄資料量太大，因此使用 NetFlow 分析訊務資料，可以兼顧效能與分析的精確性。此外，許多蠕蟲之特徵在其通訊協定、通訊埠號、封包大小、以及封包次數，例如 SQL Slammer 蠕蟲傳送 376 位元的資料到目的主機的 1434 UDP 通訊埠[8]；Worm Blaster 蠕蟲使用 TCP 的 135 通訊埠來探測 DCOM RPC 弱點，每個封包長度固定為 40 位元組；而 Worm Sasser 蠕蟲則是針對 TCP 的 445 通訊埠作漏洞掃瞄，在一筆網路訊務上留下封包大小為 3，網路流量為 144 位元組的資料。這些蠕蟲特徵在 NetFlow 均有記錄，因此網管人員很容易利用 NetFlow 為特定蠕蟲撰寫偵測程式。而一旦符合蠕蟲特徵的訊務流經偵測程式篩選出來，我們還可利用記錄中之來源主機網路位址(source IP address)與目的主機網路位址(Destination IP Address)找到網路蠕蟲的來源與受害主機。使用網路蠕蟲的特徵作為網路蠕蟲偵測的依據，也是目前網路管理人員所常使用的方法[27][30][31]，雖然使用 NetFlow 固定的特徵欄位來辨別特定的單一種蠕蟲，可有較高的偵測效能，卻花費較多的運算比對資源。另外，蠕蟲特徵在初始氾濫未經研究發現之時，其實並不明確，且目前的網路環境已經就是充斥著蠕蟲的環境，未來，隨著作業系統與應用程式的漏洞越來越多，網路蠕蟲發作的機會越多[1][2]、種類也會越多，針對單一特定蠕蟲使用 NetFlow 作特定的偵測，反而會增加網路管理的複雜度與網管人員的不方便性。另外，有些研究則不使用特徵比對，而是藉由觀察網路進出雙向流量過度不對稱[34]，或從每一對來源與目的主機之間的網路流量統計[33]，進而判斷異常訊務之存在。無論採用特徵比對或流量統計，訊務正常或異常之分野，大部份都藉由是否超過預先設定的臨界值來判斷，臨界值之訂定影響偵測系統之精確性。此外目前這些利用 NetFlow 分析的惡意行為偵測方法並不具通用性，都是針對特定的蠕蟲及入侵行為所設計。

3. 網路訊務動態基線分析

由前節介紹，我們可以發現利用特徵比對配合流量統計是目前網路蠕蟲偵測最常見與有效的方法，然而這類型的蠕蟲偵測方法在網管實務上有以下問題：

(1). 新型網路蠕蟲不斷出現，從找尋蠕蟲特徵至開發偵測程式過程將一再地重演。
(2). 網路蠕蟲越來越多樣性，並伴隨變種變形出現，越來越難有效定義其特徵。
(3). 影響網路流量因素複雜，流量監測臨界值不易訂定。
(4). 不同時段、不同通訊埠的訊務流量，具有不同的尖、離峰流量特性，單一臨界值無法有效界定異常的流量。
(5). 偵測系統的運算資源有限，易成為攻擊下第一個犧牲者。
(6). 開發各式蠕蟲偵測機制，造成管理複雜化。

為改善上述問題，我們提出一個基於通訊埠訊務基線分析的蠕蟲偵測機制，此機制並不直接使用蠕蟲的特徵來篩選異常訊務，而是假設我們只知道蠕蟲所存在的通訊埠，藉由平常對該通訊埠的訊務建立基本的基線資訊，判斷突然偏離基線的訊務，進而找出蠕蟲。基線(Baseline)分析是將一天分成多個時段並將每天相同時段的正常流量計算其平均值，這些連續不同時段的流量平均值便形成基線；基線反應網路正常行為下所呈現的流量變化，是一重要的流量指標。而一旦網路有蠕蟲發生，網路訊務中由於加入了蠕蟲散佈的封包，網路之異常將直接反應於流量變化上。然而，不同的蠕蟲影響流量程度不一，因此單從流量改變的觀察很難判斷是否有蠕蟲存在，使用傳統以網路整體流量為基礎的基線分析，更難從中發現個別蠕蟲氾濫的情況。因此本論文所提蠕蟲偵測機制之基線分析對象為個別通訊埠流量，這是因為大部份的蠕蟲藉由固定的通訊埠散佈。使用個別通訊埠流量基線分析，將可直接查覺蠕蟲影響網路訊務的程度。至於哪些通訊埠需要進行流量基線分析，我們可由目前一些已知的蠕蟲的相關報告了解已知蠕蟲所在的通訊

埠，我們也可從電腦廠商公布系統漏洞的消息，了解哪些通訊埠有潛在蠕蟲的威脅。

為更清楚說明訊務基線分析，我們以圖 1 及圖 2 為例解釋如何利用基線偵測異常訊務進而發現蠕蟲。首先我們依據不同時段正常之網路平均流量建立動態基線(圖 1 中之黑線)，然後以動態基線值為基礎提升 K 倍($K > 1$)固定比率位置建立動態臨界線(圖 1 中之藍線)，當某一時段之流量超過臨界線，我們判定此時段為流量異常之時段。圖 1 紅色水平虛線則是傳統的流量異常偵測所使用的固定臨界線。圖 2 顯示針對實際流量進行異常偵測時，有些因網路尖峰時段之正常流量高於固定臨界值而被誤判為流量異常；使用以基線為基礎的動態臨界機制，網路尖峰所造成正常流量增加仍低於臨界值，與流量基線明顯偏離的時段 (如圖 2 所標示三處)才會被視為異常。

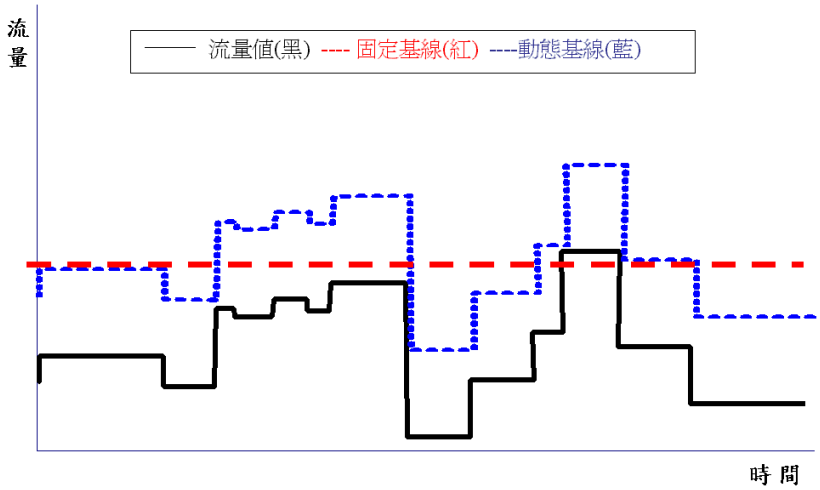

圖 1 動態基線與臨界線概念圖

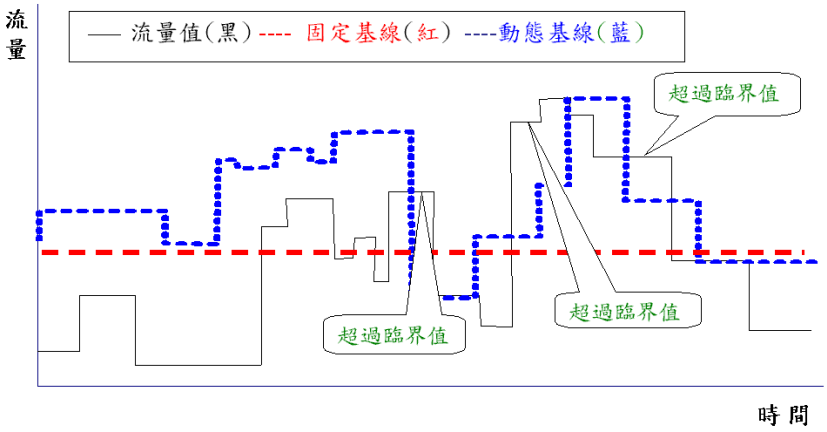

圖 2 網路流量與動態臨界線比較概念圖

當我們選定蠕蟲可能所在的通訊埠進行網路訊務基線分析與蠕蟲偵測時，除了依時段從 NetFlow 搜集訊務流記錄，尚包括四個工作：(1). 決定基線分析項目，(2). 計算訊務基線值，(3). 訂定動態臨界值，(4). 找出受感染之主機。茲分述如下：

(1). 決定基線分析項目

我們從 NetFlow 的訊務流記錄之目的通訊埠(Destination Port)欄位可判別某一筆訊務流記錄是否為基線分析的對象，此外我們尚需決定依訊務流記錄之哪一欄位進行基線分析。訊務流記錄中記載某一訊務流的封包數目(Packets)及位元組數(Bytes)，蠕蟲的感染行為是發送大量的封包，對網路流量而言，在蠕蟲感染時，不管封包數目或位元組數均會增

加，如果蠕蟲所散佈的封包非常小，而平常使用該通訊埠的正常訊務封包較大，蠕蟲對位元組數目統計的影響較不顯著，不過封包數目則會明顯增加。相反地，如果蠕蟲所散佈的封包比正常訊務封包大，位元組數目總數增加的情形會更明顯，此時封包數目也會明顯增加。目前發現許多蠕蟲的封包較小，此種情形使用封包數目建立基線較容易偵測蠕蟲的存在。

(2). 計算訊務基線值

在一個公司或組織之企業內部網路，其網路訊務量是由一群具有集體相似行為的組織成員所貢獻。隨著組織成員之作息，網路訊務通常會以一天為週期呈現規率性變化，或有明顯的尖離峰分布情況。因此我們將每一天分成 m 個時段：$I_1, I_2, \ldots I_m$，假設 p_i 為今日 I_i 時段之正常情況之訊務量(例如封包數目)，而 b_i 為至昨日止的 I_i 時段之基線值，經過今日之 I_i 時段後，I_i 時段新的基線值 b_i 更新如下：

$$b_i = a \times p_i + (1-a) \times b_i \text{ , } 0 < a < 1 \qquad (1)$$

a 代表今日的訊務量在基線基線所佔的比重，如果我們只針對週一至週五的訊務進行基線分析，我們可以直覺地將 a 設為 0.2 (即 1/5)，意謂 p_i 代表一週五天統計的其中一天之訊務，原 b_i 則代表五天中之前四天的加權平均。事實上，a 用於式(1)的加權平均是反應我們對最近一次訊務的重視程度，這可使 b_i 很快地反應最近的正常網路訊務變化。在進行 b_i 更新計算時，我們必須先確定 p_i 是正常訊務的統計值，如果 I_i 時段為異常時段，則 b_i 不更新。由於 b_i 將被用來作為判斷訊務是否正常的依據，在訊務基線計算初期，由於統計資料尚不夠多，b_i 值可能不具代表性，此時若使用 b_i 篩選正常訊務，很有可能將正常的 p_i 視為異常，而無法將 p_i 反應至 b_i 中，導致 b_i 本身的計算不正確，進而影響往後的異常偵測及 b_i 更新。因此，我們建議在訊務基線計算初期不使用式(1)計算 b_i 以及使用 b_i 判斷訊務是否正常，而是先從 NetFlow 搜集一段期間經網管人員判斷為正常的訊務資料，例如某一週之週一至週五的訊務資料，然後計算此段期間所有 I_i 時段 p_i 之平均作為開始使用式(1)計算的初始 b_i 值。

(3). 訂定動態臨界值

假設 Th_i 為 I_i 時段用來判別訊務是否異常的臨界值，很顯然，b_i 值愈大，Th_i 也應愈大，亦即 $b_i \propto Th_i$，因此我們令 Th_i 為 b_i 之固定倍數值，以反應動態基線之特性。假設 K 為一大於 1 的常數，式(2)為 Th_i 計算公式：

$$Th_i = K \times b_i \qquad (2)$$

K 值可由網管人員依偵測結果的正確性適度調整，一般而言，K 值越大時，感染行為導致訊務變化非常明顯的蠕蟲才容易被發現，因此一旦訊務被判為異常，幾可確定為蠕蟲，然而 K 值越大也會使警報誤失(False Positive)率增加，亦即有些蠕蟲沒被發現。另一方面，如果 K 值越小，被判為異常的時段越多，大部份的蠕蟲確實出現在被判為異常的時段，但警報誤判(False Positive)率也會升高，亦即有些正常的訊務也被誤以為蠕蟲。本論文所提供的是一個通用的偵測架構，不同蠕蟲之偵測的 K 最佳值可能不同，網管人員利用此架構偵測蠕蟲時，可以從先前的偵測結果調整 K 值，提昇偵測的正確性。

(4). 找出受感染之主機

當某一時段之訊務量超過臨界值時，該時段會被標示為異常時段，針對異常時段我們必須進一步分析該時段的 NetFlow 記錄，找出受蠕蟲感染的主機。一個受蠕蟲感染的主機會大量發送封包至網路上，因此在 NetFlow 留下多筆訊務流記錄。我們以三個步驟來辨識蠕蟲來源，分別是(a). 計算訊務偏離值、(b). 依 NetFlow 訊務量排序、(c). 累計至偏離值找出疑似蠕蟲攻擊主機群，茲分述如下：

(a). **計算訊務偏離值：訊務偏離值是指此異常時段訊務量與基線值間的距離，假設異常時段 I_i 之訊務偏離值為 f_i，**

$$f_i = p_i - b_i \tag{3}$$

f_i 代表因蠕蟲感染所增加的訊務量最大值， f_i 越大可能表示受蠕蟲感染的主機數目越多，因此我們必須將更多的主機列入蠕蟲可疑名單之中。

(b). **依** NetFlow **訊務量排序：貢獻較多訊務量的主機，較可能是受感染的主機，因此我們從** NetFlow **中該異常時段的訊務流記錄，將來源** IP **位址相同的訊務流之訊務量加總，並依加總結果遞減排序，顯示蠕蟲可能性由高至低排列的所有主機名單。**

(c). **累計至偏離值找出疑似蠕蟲攻擊主機群：步驟**(b)**執行後所列出的主機排名名單中，究竟前幾名最有可能是受蠕蟲感染的主機，我們可以推測 f_i 主要是由列於名單前面的主機所貢獻。假設 H_j 為主機名單中排名第 j 的主機之訊務量，我們將名單中之前 n 個主機列為最有可能感染蠕蟲之主機群，n 可由式**(4)**決定：**

$$n = \min\{r \mid \sum_{j=1}^{r} H_j \geq f_i\} \tag{4}$$

式(4)**意謂 n 為存在最小的 r 使得前 r 個主機的訊務量總和超過訊務偏離值 f_i。顯然地，f_i 越大，滿足式**(4)**最小的 r 將越大。**

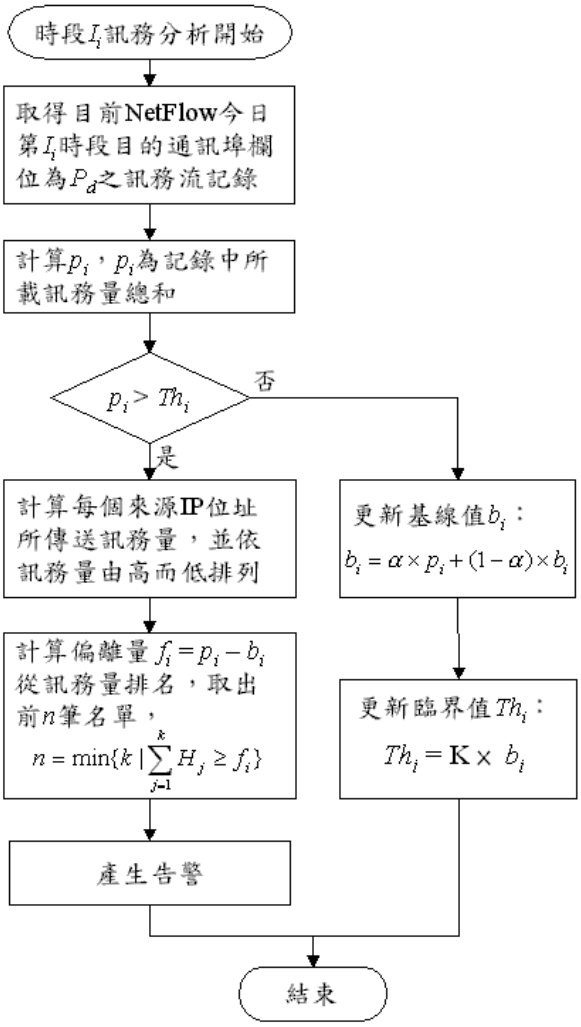

圖 3 **使用基線分析偵測蠕蟲流程圖**

綜合以上訊務基線分析方法，我們於圖 3 顯示蠕蟲偵測之流程圖。對於每一個時段 I_i 及每一個被分析的通訊埠 P_d，我們均必須執行一次如圖 3 的偵測流程。

4. 實驗測試–以殺手蠕蟲為例

為驗證本論文所提蠕蟲偵測機制，我們依據本論文所提設計構想於實驗室環境建立網路訊務動態基線分析之實驗雛型，該實驗雛型使用一部 HP tc4100 網路伺服器作為訊務搜集伺服器，其網卡開啟為雜亂模式(Promiscuous Mode)接收經過該網卡之所有訊務以便進行基線分析，為使該網卡能搜集到實驗室網路內之所有封包，我們將此網卡連接至實驗室之乙太網路(Ethernet)交換器，並開啟交換器之鏡射(Mirror)功能，使所有經過交換器的封包都能複製一份至訊務搜集伺服器。所有為訊務搜集伺服器所接收之訊務會先經過 Fprobe 軟體[13]產生符合 NetFlow 訊務流記錄格式之訊務資料。Fprobe 是一套植基於 libpcap 封包擷取程式庫上之訊務搜集工具，它主要的功能是將原始的訊務整理成符合 NetFlow 記錄格式的訊務流資料。此外，我們還利用 flow-tools [14]套件中之 flow-capture 工具[12]，接收由 Fprobe 所送來之 NetFlow 訊務流資料，並以五分鐘為一時段分別儲存訊務資料以便進一步進行基線分析。此外，我們並使用 MySQL 資料庫儲存基線分析與蠕蟲偵測統計資料。

建立實驗雛型後，我們以殺手蠕蟲(Worm Sasser)為基線分析對象進行實驗測試。2004 年 4 月 13 日，微軟公司發佈了編號 MS04-011 的系統弱點公告，指出微軟作業系統之 LSASS(Local Security Authority Subsystem Service)服務有緩衝區溢位之漏洞，在不到 20 天的時間，此漏洞便被利用設計成為殺手蠕蟲。殺手蠕蟲是針對目的主機之 TCP 445 通訊埠作漏洞掃瞄，送出特殊的封包導致緩衝區溢位，使得受感染的主機開啟 TCP 5554 通訊埠將大小為 15,872 位元組殺手蠕蟲複製到受感染的主機上。從網路訊務的觀察，殺手蠕蟲的感染過程，初期是漫無目的主機掃瞄，其封包傳送行為較為異常；而在發現有漏洞之主機後使用 FTP 將蠕蟲植入的行為則與一般主機間通訊沒有兩樣。因此，我們的實驗測試將針對殺手蠕蟲在主機掃瞄時所使用的 TCP 445 通訊埠之流量進行實驗分析。相關的實驗參數列於表 2：

表 2 實驗測試參數

參　數	設定值
每一分析時段長度	5 分鐘
目的通訊埠 P_d	445
基線計算加權 a	0.15
臨界倍數 K	5

本實驗進行測試時間在 2004 年 5 月中旬，由於殺手蠕蟲大量感染行為發生於 5 月初，許多受感染主機此時已完成漏洞修補及掃毒，在實驗測試期間，雖然沒有大量的殺手蠕蟲發作，不過仍有零星的殺手蠕蟲感染情形。此外，實驗所監測的網路環境為系上之網路通訊實驗室，實驗室約有 15 部個人電腦，大部份為研究生個人所使用。因此，本身網路訊務量並不多，訊務搜集伺服器所搜集的訊務除了實驗室本身所產生的訊務，尚包括由外部經路由器至實驗室交換器的訊務，這些外來訊務包括廣播、群播、以及目的 IP 位址為實驗室內主機之封包。為方便實驗進行，我們並不使用每 5 分鐘執行一次的即時蠕蟲偵測，而是先搜集一星期週一至週五的訊務資料，再模擬 5 分鐘執行一次的基線分析。圖 4 顯示第五天早上 8:00 至 10:30 之間的實際流量、基線、以及臨界線。

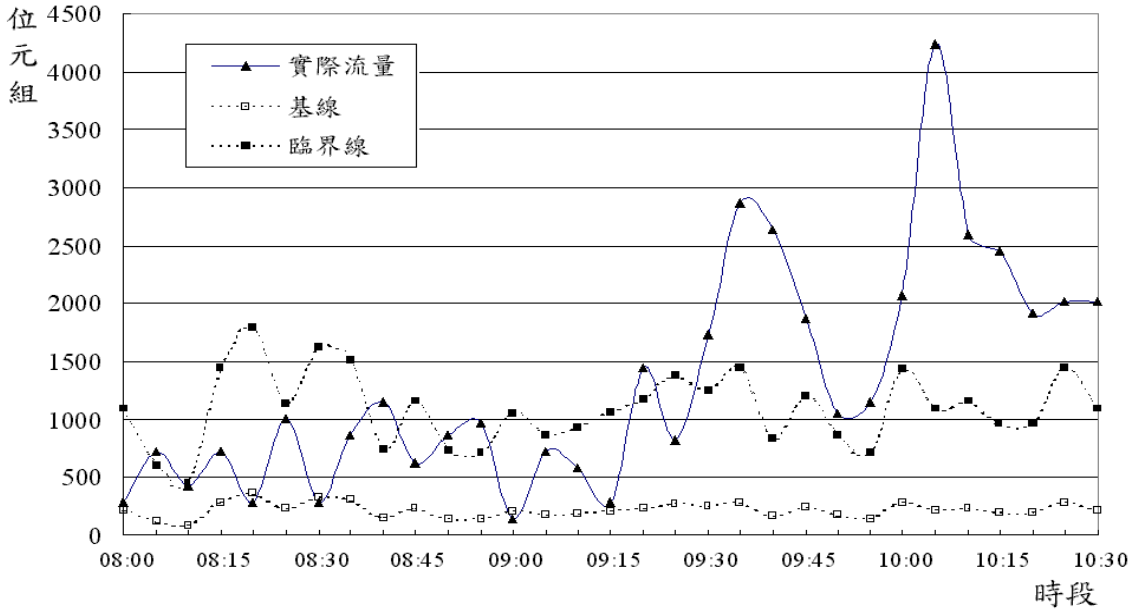

圖 4 TCP 445 **通訊埠流量統計**

表 3 **時段** 10:05 **可疑主機之偵測**

來源主機	流量 (位元組)	累計流量 (位元組)	可疑主機
17.111.179.43	768	768	✓
10.10.24.220	432	1200	✓
10.10.16.210	313	1513	✓
10.10.23.9	288	1801	✓
10.10.24.73	288	2089	✓
10.10.24.97	288	2377	✓
10.10.15.133	144	2521	✓
10.10.23.84	144	2665	✓
10.10.14.117	144	2809	✓
10.10.24.88	144	2953	✓
10.10.23.70	144	3097	✓
10.10.21.213	144	3241	✓
10.10.9.171	144	3385	✓
10.10.23.108	144	3529	✓
10.10.16.160	144	3673	✓
10.10.21.157	144	3817	✓
10.10.21.140	96	3913	✓
10.10.15.191	96	4009	✓
10.10.24.115	88	4097	
10.10.23.186	48	4145	
10.10.20.155	48	4193	
10.10.15.150	48	4241	

由圖 4 我們可以知道在 9:30 以前，流量大致正常，18 次的流量發生 5 次實際流量超過臨界線的情形(8:05、8:40、8:50、8:55、及 9:20)，9:30 以後幾乎所有時段流量都超過臨界線。對於每一超過臨界線之時段，我們必須進一步找出可能受感染之主機，我們以圖 4 中發生流量最高的 10:05 時段為例說明實驗過程。10:05 時段之基線值為 216，實際流量為 4241 位元組，超過臨界值 1080，因此我們將此時段判為異常時段。實驗程式因此將 10:05 時段之訊務流記錄將來源 IP 位址相同的訊務流之流量加總，並依流量由高而低排序產生主機列表。此時段流量偏離量 f_i 為 4241－216＝4025 位元組，因此我們從列表由上而下找到累積流量大於 4025 處，將此處前之主機標示為可疑主機。表 3 顯示實驗結果。

在上述實驗過程，我們呈現了使用訊務基線分析偵測蠕蟲的基本架構，然而蠕蟲偵測之正確性與基線的計算是否準確有關，基線的建立必須植基在更多具代表性的訊務之分析，此次實驗由於無法直接從骨幹網路的路由器取得足以代表所處網路全貌的網路訊務，再加上只擷取 445 通訊埠訊務計算基線，我們認為此次實驗所搜集的訊務量尚不足以完全反映平常的網路訊務。另外，在此實驗中我們發現使用訊務偏離量決定應列入可疑主機的數目可能過多，雖然如此，比起傳統使用固定數目列表方式，我們的方法較具彈性，所列可疑名單對網管人員仍有相當大的參考價值。

5. 結語與未來展望

網際網路的發展帶動資訊科技之日新月異、無遠弗屆，給人一種天涯若比鄰的感覺，然而相反的，資訊科技的進步也加速網際網路的發展，若將資訊科技與網際網路相互結合，往善的方面為之，所帶給我們的是經濟與社會更加的發達，另一方面，往惡的方向走，則會給我們的世界帶來無法想像的損失，例如惡意網路蠕蟲的產生，便是一項資訊科技與網際網路相結合之下的產物，其帶來的危害小至個人資料的遺失，大至整個世界的金融、經濟秩序的影響都相當可觀，然而要完全杜絕網路蠕蟲之威脅，是件非常困難的工作，因為系統上、應用程式上之漏洞層出不窮，網路蠕蟲之威脅非但是不見減少，反而更加快速度越來越多，再加上變形以及舊漏洞新蠕蟲(多蠕蟲)之舊瓶裝新酒的攻擊手法，不僅讓網路蠕蟲攻擊成為網路生活的一部份，更在在顯示網管人員與網路蠕蟲之間道高一尺魔高一丈的關係。

因此，本研究提出使用網路訊務監測方式，利用建立不同時段、不同通訊埠之網路流量動態基線，並以此動態基線為基礎，分析各時段的網路流量，藉以偵測網路蠕蟲的發生，正當我們進行實驗設計時，適逢微軟再度發佈系統漏洞警訊，而此漏洞也在短時間內被利用設計成網路蠕蟲，特別是殺手蠕蟲(Worm Sasser)的產生，在我們實際偵測的環境下，也確實被我們監測到其爆發感染影響網路的行為，這證明本文所提之偵測機制確為可行的方法，特別是在於新型的網路蠕蟲開始掃瞄、感染時。最後，針對我們所提的監測機制，茲條列其優點，並論述如下：

(1). 資料來源標準化

本論文所提訊務基線分析所使用 NetFlow，為思科公司所提出之標準，此標準也已列為 RFC 標準草案，由於其資料格式公開，使得 NetFlow 漸漸成為業界共同認定與支援之標準。

(2). 分析方法單一化

本論文所提監測機制之分析方式採用依據不同時段來建立網路流量基線的方法，不僅在監測時使用基線作為網路異常的分析，更在偵測時使用基線累積的網路流量作為來源判斷的標準，使得整個機制所採用的分析方法單一化。

(3). **網路監測簡易化**

因應未來網路蠕蟲種類更多、變形加速而產生不同蠕蟲擁有不同的特徵與行為，若使用特徵偵測法勢必依各不同的網路蠕蟲建立個別的偵測系統，很顯然，在網路蠕蟲越來越多的情況下各種偵測系統也會越來越多，造成網管人員管理上的困難；然而，使用動態基線分析來偵測網路蠕蟲之發生，雖然無法保證能夠完全正確偵測所有的蠕蟲，使用單一的監測機制，卻可簡化監測的工作，也能能在網路蠕蟲發生的初期發現訊務的變化進而找出來源，這使網路監測的工作在實務上較為可行。

本論文研究主要目的不在於找出最精確的蠕蟲偵測方法，而是考量偵測方法的通用性，從網管實務的觀點提出一套確實可行的蠕蟲偵測機制，使網管人員能夠依循此機制，從漏洞發佈有可能造成氾濫影響網路時便加以監控，並在真正發生網路蠕蟲攻擊事件的初期，在尚未有特徵出現時，便能有網路蠕蟲之警訊，並判斷疑似的受感染問題主機來源。未來，我們希望能依據本論文所提機制開發一套自動化的蠕蟲監測系統，提供圖形化操作介面使網管人員只要經由簡單的設定，就可啟動某一通訊埠的基線分析與蠕蟲偵測，並能即時收到監測系統告警，在第一時間掌握蠕蟲的蹤跡。此外，目前我們只使用 NetFlow 訊務流記錄中所記載之原始封包數目及流量進行基線分析，未來我們將研究其他可能作為基線分析的對象，以期提高蠕蟲偵測之精確性。可能的基線分析對象包括：前後時段的流量變化、前後時段的封包數目變化、平均封包大小(即流量除以封包數)、每時段平均訊務流數目等。此外，基線計算加權參數 a 以及臨界倍數 K 值之設定對蠕蟲偵測精確性之影響，也有待進一步研究。

參考文獻

[1] CERT Advisory CA-2001-09 Statistical Weaknesses in TCP/IP Initial Sequence Numbers, *http://www.cert.org/advisories/CA-2001-09.html*.

[2] CERT/CC Statistics 1988-2003, *http://www.cert.org/stats/index.html*.

[3] Cisco NetFlow, *http://www.cisco.com/warp/public/732/Tech/nmp/NetFlow/index.shtml*.

[4] Cliff Changchun Zou, Lixin Gao, Weibo Gong, and Don Towsley, Monitoring and Early Warning for Internet Worms, *CCS' 03*, Washington, DC, USA, October 27–31, 2003.

[5] Cliff Changchun Zou, Weibo Gong, and Don Towsley, Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense, *WORM' 03*, Washington, DC, USA, October 27, 2003.

[6] Cliff Changchun Zou, Weibo Gong and Don Towsley, Code Red Worm Propagation Modeling and Analysis, *CCS' 02*, Washington, DC, USA, November 18-22, 2002.

[7] D. Moore, C. Shannon, and J. Brown, Code-Red: A Case Study on The Spread and Victims of an Internet Worm, *Proc. 2nd ACM Internet Measurement Workshop*, ACM Press, pp. 273–284, 2002.

[8] D. Moore, V. Paxson, Colleen Shannon, Stuart Staniford, and Nicholas Weaver, Inside the Slammer Worm, *IEEE Security & Privacy*, pp.33-39, 2003.

[9] Darrell M. Kienzle and Matthew C. Elder, Recent Worms: A Survey and Trends, *WORM ' 03*, Washington, DC, USA, October 27, 2003.

[10] Denning, P. J., (ed.), *Computers Under Attack: Intruders, Worms, and Viruses*, ACM Press, Addison-Wesley, New York, 1990.

[11] Eugene H. Spafford, The Internet Worm Incident, *Technical Report CSD-TR-933*, Purdue University, Sep. 1991.

[12] FlowFront, *http://www.ratel.ru/flow/*.

[13] Fprobe, *http://fprobe.sourceforge.net/*.

[14] Flow-tools, *http://www.splintered.net/sw/flow-tools/*.

[15] Nicholas Weaver, Vern Paxson, Stuart Staniford, and Robert Cunningham, A Taxonomy of Computer Worms, *WORM ' 03*, Washington, DC, USA, October 27, 2003.

[16] Prabhat K. Singh and Arun Lakhotia, Analysis and Detection of Computer Viruses and Worms-An Annotated Bibliography, *ACM SIGPLAN Notices* 29 V. 37(2), February 2002.
[17] Vincent Berk and George Bakos, Designing a Framework for ActiveWorm Detection on Global Networks, *IEEE International Workshop on Information Assurance* (IWIA'03), 2003.
[18] Xinzhou Qin, Wenke Lee, and Lundy Lewis, Integrating Intrusion Detection and Network Management, *IEEE NOMS* Session 8 Proactive Security Management, pp.330-343, 2002.
[19] James D. Murray**著，**SNMP**網路管理協定，**O'REILLY, 1999**。**
[20] **微軟安全性公告** MS04-011**，**
http://www.microsoft.com/taiwan/security/bulletins/MS04-011.asp**。**
[21] NetFlow **，**http://NetFlow.nctu.edu.tw/NetFlow.html**。**
[22] **蕭漢威，陳年興，「網路流量統計分析及其管理意函」，**1998**台灣學術網路研討會。**
[23] **劉大川，張傑生，楊子翔，「台灣學術網路出國線路壅塞狀況之分析研究」，**1998 **台灣學術網路研討會，民國**87**年**10**月。**
[24] **蕭漢威，陳年興，「**TANet**骨幹網路流量監測技術」，**1999**台灣學術網路研討會，民國**88**年**10**月。**
[25] **林佳毅，蕭漢威，陳年興**(1999)**，「**TANet**網路骨幹流量統計分析」，**2000**台灣學術網路研討會，民國**89**年**10**月。**
[26] **楊子翔，蔡錫鈞，「**Network DoS/DDoS**攻擊及預防方法之研究」，**2000**台灣學術網路研討會，民國**89**年**10**月。**
[27] **黃文穗，林守仁，「利用**NetFlow**建置**Code Red Worm**偵測系統」，**2001**台灣學術網路研討會，民國**90**年**10**月。**
[28] **林鳳銘，吳守豪，李蔡彥，「**Intrusion Detection: a Network View**」，**2001**台灣學術網路研討會，民國**90**年**10**月。**
[29] **嚴大中，廖百齡，俞齊醒，何應魁，江清泉，「以防火牆日誌分析之網路攻擊偵測系統」，**2002**台灣學術網路研討會** pp.828-834**，民國**91**年**10**月。**
[30] **周文正，「校園區域網路病毒攻擊自動偵測與阻斷系統」，**2002**台灣學術網路研討會，**pp.840-846**，民國**91**年**10**月。**
[31] **陳嘉玫，黃世昆，陳年興，鍾明勳，「即時偵測防治**Internet Worm**」**
http://www.cert.org.tw/document/docfile/InternetWorm.pdf, 2002**。**
[32] **王岳忠，「入侵偵測系統的運作方式」，***http://taiwan.cnet.com***，民國**92**年**10**月。**
[33] **楊素秋，曾黎明，「**Nimda **攻擊訊務的檢測」，**2003**台灣學術網路研討會，台北，民國**92**年**10**月。**
[34] **楊素秋，曾黎明，「**X-Attack **攻擊訊務的監測與自動阻絕」，**2003**台灣學術網路研討會，台北，民國**92**年**10**月。**
[35] **范修維，廖鴻圖，伍啟錄，「網路攻擊來源追蹤技術」，**2003**台灣學術網路研討會，台北，民國**92**年**10**月。**