

適用於電子商務環境之網蟲即時偵測及防禦系統

摘要

順暢的網路與安全的交易進行是維持電子商務營運不可或缺的條件。因此，企業無不想盡辦法維持網路連線的正常及加強交易過程的安全。最近幾年，網蟲（Internet Worm）的盛行嚴重威脅著電子商務的正常運作，但目前的網路安全防禦機制，例如入侵偵測系統與防火牆，仍無法即時對網蟲採取應變動作來確保網路的通暢，亦無法防禦未知網蟲。本文提出一個網路連線異常偵測方法，可用來偵測已知及未知網蟲，在偵測網蟲後能即時採取應變措施來維持網路連線的暢通。除介紹系統工作原理與系統架構之外，我們也建置網蟲攻擊模擬環境，以最新的 Blaster 網蟲為例證明此偵測方法的確能偵測出目前的網蟲。

關鍵字：網蟲偵測、網蟲防禦、Blaster 網蟲、網路安全

壹、前言

根據 Mi2g 的調查[4]，Slammer 網蟲[3]的生產損失在開始流行的前五天就已達十億美元，CodeRed 及其變種網蟲[2]更達 26 億美元，加上預防感染及事後修復的成本更是驚人。網蟲盛行之下，可能癱瘓電子商務賴以生存的電腦主機及網路連線，造成嚴重的生產損失，事後的修復成本及事前的預防成本也不容小覷。但目前的網路安全防禦機制，例如防火牆與入侵偵測系統，難以防治網蟲大量散播攻擊所造成的網路服務癱瘓，這是因為它們仍無法偵測未知網蟲攻擊及缺乏即時的回應機制。

為解決此問題，本文提出一種網蟲擴散攻擊連線偵測方法，其原理是利用歸納現有網蟲擴散時的共同特徵，再將流量異常偵測方法加入更多的限制條件來找出網蟲攻擊連線，並採取回應加以防禦。我們不但實作此網蟲偵測方法，同時也建模擬環境，並以最新的 Blaster（疾風）網蟲證明此方法的可行性。

本文共分為五節。第二節對歸納已知網蟲的相同行為特徵，並依此提出我們

的網蟲偵測方法。第三節說明網蟲偵測及防禦系統的實作及應用方式。第四節以 Blaster 網蟲實證提出的網蟲偵測方法。第五節做一簡要之結論。

貳、網蟲偵測方法

本節對網蟲擴散時的特性進行歸納，並依此提出我們的網蟲偵測方法。

一、網蟲簡介

網蟲與病毒一樣都屬於惡意程式的一種，兩者的最主要的差別在於擴散時的主動及被動性，以往的病毒必須依賴使用者加以執行之後才能潛伏於系統中，並於特定時間點發作。而網蟲則會主動攻擊其他主機，利用漏洞進行自我複製，因此只要有漏洞的主機置於網路上，即使完全沒有使用者在此主機上操作也可能會感染網蟲。

近來，網蟲有愈來愈盛行的趨勢[8]，至 2001 年至今短短二年，就出現了 CodeRed、CodeRed II、Nimda、Slammer、Blaster 等危害重大的著名網蟲，除了新網蟲不斷出現之外，每個網蟲也出現愈來愈多的變種網蟲，以 Blaster 網蟲為例，就有高達七種以上的相似變種。更值得注意的是：愈來愈多的怪客（Cracker）以利用網蟲散佈私人的訊息為樂，使得網蟲種類愈來愈多。在一個新的網蟲出現後，也往往在不久的時間後就出現更多的使用類似手法的變種網蟲，也因此網蟲的偵測及防禦[6,9,10]顯得格外重要。

二、網蟲擴散時之共同特性

我們可以從目前的網蟲歸納出一些網蟲進行擴散具有的共同特性，茲整理如下：

- 短時間內攻擊眾多不同受害者

目前的網蟲爲了要快速地擴散，多會不斷地攻擊不同的目標，而且攻擊的速率相當頻繁。因此，一旦有主機感染網蟲，可以觀察到感染主機發出許多擴散攻擊連線至許多 IP Address 不同的主機。擴散攻擊連線愈頻繁的網蟲，通常擴散的也愈快：Slammer 網蟲是目前擴散速度最快的網蟲，根據統計[7]，感染 Slammer 網蟲的主機數目每 8.5 秒會增加一部，發作開始十分鐘內就可以掃描 Internet 上超過 90% 以上的主機，其中一個重要的原因就

是 Slammer 網蟲在 100 Mbps 的網路每秒能發出高達約 30000 次擴散攻擊。

- 使用 IP Address 做為擴散目標選擇策略

網蟲在選擇擴散對象時，多以特定的演算法直接計算出擴散對象的 IP Address，再加以攻擊。但是這樣的方式違背一般正常的使用習慣，多數的使用者會使用網域名稱（domain name）來連結大多數的網站，而不是大量地使用 IP Address 來連結，因此，如果發現太多的連線目的主機都不具 domain name 查詢記錄的話，可將其視為異常狀況。

- 擴散連線的通訊協定欄位固定

網蟲在擴散時多會攻擊漏洞應用程式的預設通訊埠，這是因為網蟲現階段仍無法判斷漏洞應用程式使用哪一個通訊埠，退而求其次攻擊最多人使用的預設通訊埠。目前已知的網蟲對一個或幾個固定的通訊埠進行攻擊，因此網蟲感染主機可被觀察出發出許多相同目的通訊埠的連線。

另外，目前的網蟲每次擴散攻擊使用的方式皆相同，因此每個攻擊連線封包的資料欄位（payload）都會是相同的。

其他欄位例如來源通訊埠、TCP 協定中的 Sequence Number、Acknowledge Number、Code Bits 欄位等，則視網蟲寫作方式而有可能相同或不同。

- 擴散攻擊封包不會太小

網蟲在做擴散攻擊時，其攻擊連線必須包含漏洞攻擊、網蟲複製等複雜動作，因此網蟲擴散連線封包大小會有一定程度，Slammer 是目前擴散連線封包總合最小的網蟲，長度只有 404 Bytes。其他網蟲，如 CodeRed 及 Nimda，分別是 4 KB 及 60 KB。

三、網蟲偵測及防禦方法

本文依前述的網蟲擴散時具有的相似特性提出一個新的網蟲偵測方法。此方法是改進流量異常偵測方法而來，過去流量異常偵測方法將網路流量依通訊協定種類（TCP 或 UDP）與目的通訊埠加以區分，將相同通訊協定及相同目的通訊埠的流量大小加總來發現異常。一個簡單的例子是某天管理人員發現 TCP 協定

80 目的通訊埠的流量突然增加，此時可以假設有異常狀況發生，甚至用它來偵測出 CodeRed 網蟲。

本文提出的網蟲偵測方法除了通訊協定種類及目的通訊埠之外，亦依前述網蟲特性加入了更多的限制條件，使得異常偵測更為準確。本偵測方法假設一部正常的主機「不會在短時間內發送大量通訊協定、目的通訊埠、及資料欄位三個條件相同的封包給不同位址的其他主機」。若有主機違反此假設，快速傳送相同的封包給不同的主機，此連線則視為網蟲擴散攻擊連線。部份應用的連線起始時，會發送相同的連線要求封包給多部主機，為了避免誤判此種連線，我們可以加入封包大小的限制，因為網蟲的封包具有一定程度以上的大小，而連線要求封包為求效率，多會設計較為簡潔，封包長度因此會較短。如果能取得每個主機的 DNS 查詢記錄，我們可以將有查詢記錄的主機連線忽略不計，因為網蟲所攻擊的對象以 IP Address 為主，而這種作法，主機並不會查詢 DNS。

利用上述的網蟲偵測方法偵測到網蟲後，便可配合防火牆即時阻擋網蟲進行擴散，更可以持續監視異常狀況，一旦狀況解除，系統便可自動移除規則，如此一來，我們得到一個不需人力介入的網蟲防治解決方案。

參、系統實作與部署方式

一、系統解析

此偵測方法目前已實作於 Linux 作業系統，並配合 Linux Bridge Firewalling 套件[1]發展出一個網蟲即時偵測及防禦系統。此系統主要功能為在網蟲盛行時，在偵測出網蟲後，以阻擋網蟲的擴散攻擊連線減低對外網路及內部網路的癱瘓，維持網路連線的暢通。

一般企業的網路架構如圖 1 所示：各式主機連接至終端交換器上，終端交換器再連接至功能較強大的核心交換器，核心交換器再連結至路由器，最後路由器以 ISP 線路連結 Internet。通常內部網路的頻寬會比連結至 Internet 的頻寬為大，在圖中以線的粗細來表現，線條愈粗表示頻寬愈大。

在企業內部有主機感染網蟲後，會不斷發出擴散攻擊連線掃描外部及內部網路主機，此時內部及對外網路線路會充滿攻擊連線（由網蟲感染主機至 Internet 的路徑），因為與 ISP 連結的對外線路因頻寬較小，會首先被攻擊連線塞滿造成

其他正常主機無法連外。如果有更多的主機感染網蟲，則會進一步造成內部網路線路雍塞或是內部交換器當機而癱瘓內部網路。

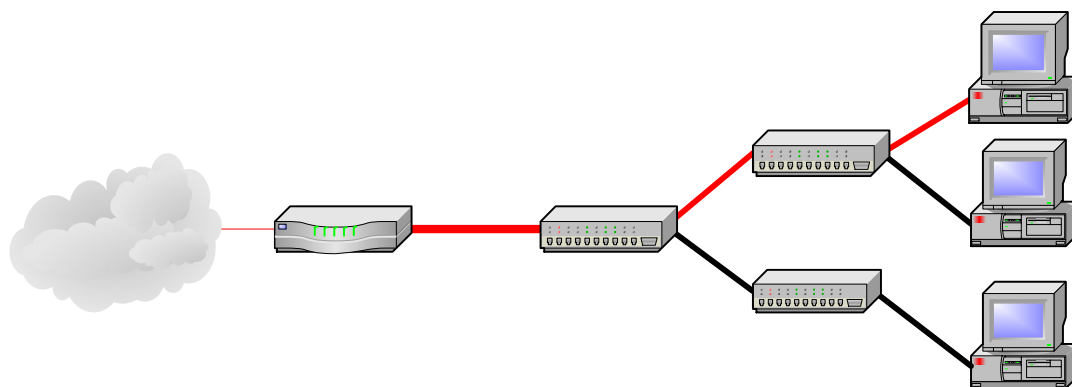


圖 1：網蟲擴散造成網路癱瘓示意圖

二、部署位置

如圖 2 及圖 3 所示，本系統（圖 2 及圖 3 Firewall 處），可部署於兩個網路設備之間，系統的主要功能是偵測及反應。偵測功能對通過的網路封包進行蒐集並加以統計，並以網蟲偵測方法偵測網蟲。反應功能在偵測到網蟲後，以防火牆對可疑的網蟲擴散攻擊連線進行阻擋。本系統以 **bridge** 模式運作，因此能在不更改任何網路設備的設定下，即時偵測及防禦網蟲。

本文提出此系統可部署的兩個網路位置：一個是位於路由器與交換器之間，另一個是內部網路的各交接器之間。二種部署方法都有其特性及優缺點，茲說明如下：

- 部署於路由器與交換器之間

此部署方式將網蟲偵測及防禦系統主機部署於路由器與核心路由器之間，此方式的優缺點分析如下：

本方式的優點：

1. 可阻止內部網蟲向外擴散

將系統部署於此位置能阻擋網蟲對外部網路擴散，如圖 2 所示，網蟲的擴散連線在路由器與交換器之間被阻擋，與 ISP 連結的對外連

線不會被內部往外的網蟲攻擊連線所佔滿，因此其他未感染網蟲的主機仍可以對外。

網蟲偵測及防禦系統在蒐集足夠的外部網蟲對內部網路的擴散攻擊連線時，亦能阻擋由外部網蟲進入內部網路，但無法阻止外部網蟲連線影響連外線路速度。

2. 防禦系統部署容易

由於只要部署一台主機於路由器與交換器之間，部署相當容易，但網路速度將取決於此主機的效能。

本方式的缺點：

1. 內部網路仍遭受網蟲攻擊

不過在網蟲偵測及防禦系統以下的內部網路仍有攻擊連線，一旦內部網路有多台主機感染網蟲，內部網路仍可能癱瘓。

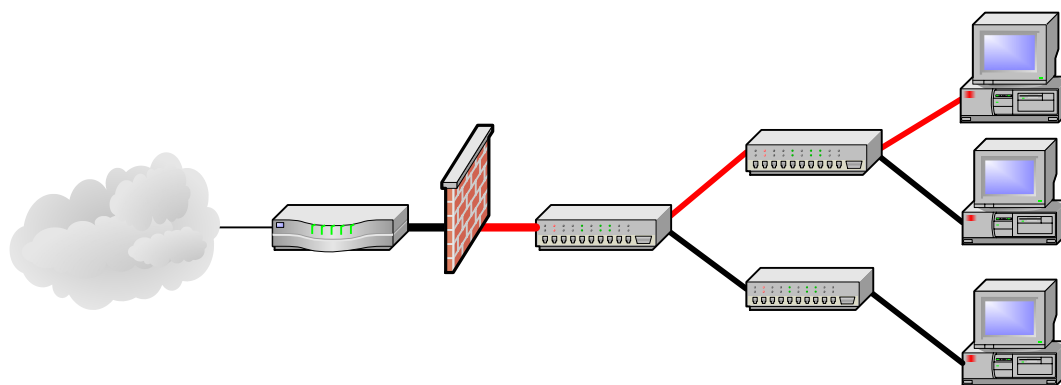


圖 2：將網蟲偵測及防禦系統部署於路由器與交換器

- 部署於各交接器之間

這個部署方式在核心交換器與每個終端交換器之間都部署一台網蟲偵測及防禦系統，此方式的優缺點分析如下：

本方式的優點：

1. 可有效解決內部網路網蟲攻擊問題

此方法將內部網路切分成數個小網路，如果有為網蟲感染主機也只會影響其所有的網路區域，其他的正常的區域內主機仍可正常對外。

2. 防禦系統效能不影響整體網路速度

由於有多台主機一起分擔處理網路流量，即使主機效能不佳也只會影響所負責區域的主機，而非所有主機。

本方式的缺點：

1. 部署成本較高

由於核心交換器與每個終端交換器之間都需部署一台主機，若是內部網路龐大，成本會相當高昂。

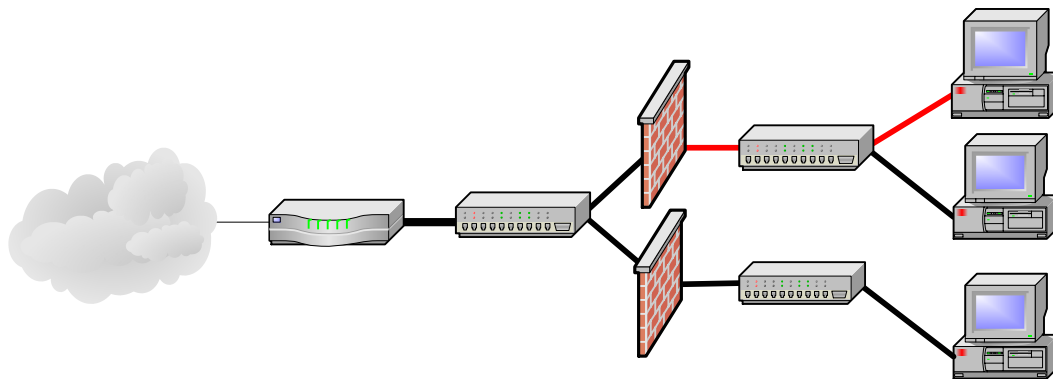


圖 3：將網蟲偵測及防禦系統部署於交換器之間

另一種較周密的部署方式是將網蟲偵測及防禦系統放置於每台主機與終端交換器之間，一個可行的方式是在網路卡或主機式防火牆中建置網蟲偵測及防禦模組[5,11]。如此一來，任一網蟲感染主機都無法大量擴散，發出的攻擊連線也會被阻擋，網蟲造成的對外連線及內部網路的癱瘓問題則得以根本解決，但此方法的成本也可能是最高的。

肆、模擬實驗

此節說明如何建置 TCP 網蟲擴散的模擬環境，並 Blaster 網蟲實證網蟲偵

測方法的可行性。

一、建置 TCP 網蟲擴散模擬環境

目前出現的網蟲可分為 TCP 及 UDP 兩類。UDP 網蟲數目較少，較著名為 Slammer 網蟲。TCP 網蟲數目較多，著名的 CodeRed I/II、Nimda、Blaster 皆屬 TCP 網蟲。由於 TCP 協定的特性，TCP 網蟲在發送擴散攻擊封包時，必須先完成 TCP Handshaking 動作，所以攻擊對象必須存在，網蟲才會送出擴散攻擊。但因無法確定網蟲要攻擊的對象的位址為何，要建立 TCP 網蟲的擴散模擬環境尤其困難。目前有二種方法：一是更改網蟲的程式碼，限制其攻擊範圍。另一是利用改寫目的位址的方法，將所有網蟲送出的攻擊連線轉送到同一台主機。

二、模擬實驗與結果

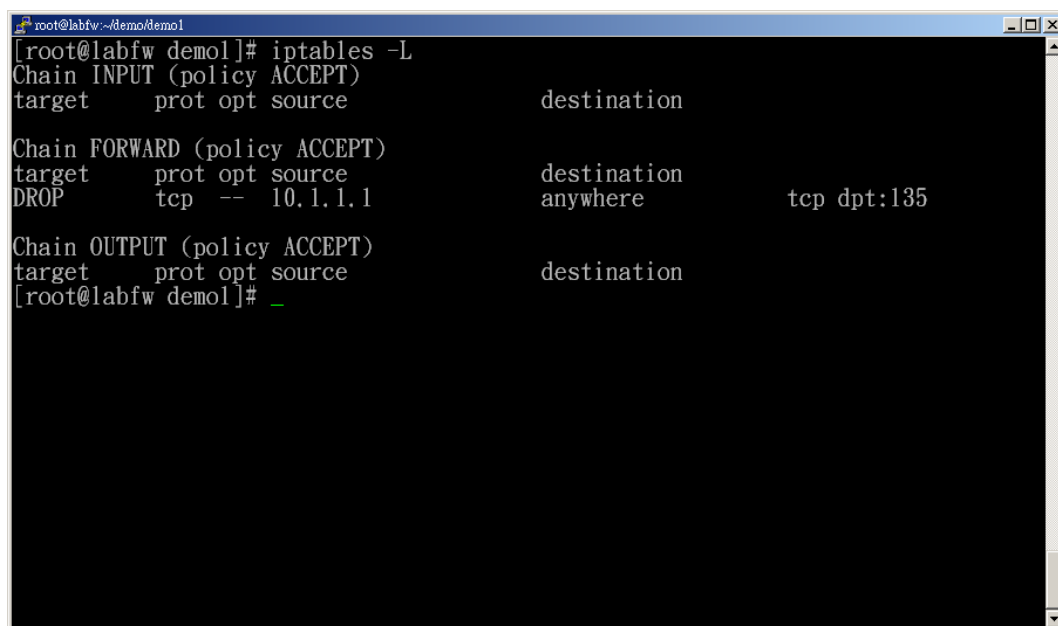
我們依網蟲偵測方法實作出一個封包統計程式，對所有經過的 TCP 和 UDP 連線進行統計，並記錄其連線時間、來源通訊埠、目的位址、目的通訊埠、及封包資料欄位長度、及資料欄位的雜湊值。只要在一定的時間間隔內，有相同通訊協定、目的通訊埠、資料欄位雜湊值的封包，傳送至 n 台不同位址的主機（n 為門檻值，可以重視程度來制定），就認定為異常情形。此時，程式會將異常封包加以存檔，同時以通訊協定、來源位址、及目的通訊埠產生網蟲的阻擋規則。圖 4 是 Blaster 攻擊被偵測時的存檔封包雜湊值及連線資訊。圖五則是自動產生的網蟲阻擋規則。

```

c0acafe561e1d9a59bb3450c49a55f6c 10.1.82.90:1038-10.1.0.90:135
fb36b35e2735938621ba7de4fd3dd6d2 10.1.82.90:1452-10.1.75.8:135
9f46a5ada0157e5eddc794102aa6fa6c 10.1.82.90:1453-10.1.75.9:135
a26b51f9dd5297b37e393ff0610c0dea 10.1.82.90:1774-10.1.34.178:135
73e09bdf6607ceaff36be48e33933454 10.1.82.90:1926-10.1.155.1:135
932a3b734a26ff7b7d9e94a0e8893ed8 10.1.82.90:2164-10.1.182.3:135
87bb11d0c6284c8832a8acad7703a751 10.1.82.90:2683-10.1.105.19:135
3cf0cba2ce9b3ce7741079862141ab65 10.1.82.90:2826-10.1.160.47:135
2b569237d69ae44d38cb5d4d93da4610 10.1.82.90:2868-10.1.182.3:135
26d3f552590822bfd585ac97ad5e9185 10.1.82.90:3368-10.1.0.90:135

```

圖 4：Blaster 網蟲擴散攻擊封包



```
root@labfw ~/demo/demo1
[root@labfw demol]# iptables -L
Chain INPUT (policy ACCEPT)
target    prot opt source                destination

Chain FORWARD (policy ACCEPT)
target    prot opt source                destination
DROP      tcp  --  10.1.1.1              anywhere           tcp dpt:135

Chain OUTPUT (policy ACCEPT)
target    prot opt source                destination
[root@labfw demol]# _
```

圖 5：Blaster 網蟲防禦規則

伍、結論

無論是前年的 CodeRed 或是今年的 Blaster (疾風)網蟲都對電子商務的經營企業產生有形或無形的損失及傷害，雖然各式網路安全產品及防禦機制不斷精進，但就現階段來說仍無法有效地在第一時間偵測及防禦網蟲。

本文歸納現有已知網蟲的擴散特性對流量異常偵測進行改良，加入更多的網蟲判定條件，可更精準地偵測出網蟲擴散連線。再結合開放原始碼的 Linux 作業系統及 Linux Bridge Firewalling 套件實作的網蟲偵測及防禦系統，能在不改變任何網路設定下，部署於內部網路中的數個位置，即時偵測及防禦網蟲，藉由阻擋網蟲擴散攻擊連線，能夠降低對外線路及內部網路因為充滿網蟲擴散連線而癱瘓的機率，來維持網路連線的暢通。本文亦以 Blaster 網蟲實證此偵測方法及系統的可行性。

電子商務時代中，網路連線的順暢是電子商務進行的必要因素之一，本文希冀藉由提出網蟲偵測及防禦系統，降低從事電子商務各個企業面臨層出不窮的網蟲攻擊事件所造成的傷害及損失。

陸、參考文獻

- [1] Bridge – Linux Ethernet Bridge Project, <http://bridge.sourceforge.net>.
- [2] CERT/CC, “CERT Advisory CA-2001-23 Continued Threat of the "Code Red" Worm”, July 26, 2001.
- [3] CERT/CC, “CERT Advisory CA-2003-04 MS-SQL Server Worm”, January 25, 2003.
- [4] CNET Networks, Inc., “Counting the cost of Slammer”, January 2003.
(Available at: <http://news.com.com/2100-1001-982955.html>)
- [5] Gregory R. Ganger, Greg g Economou and Stanley M. Bielski, “Self-Secure Network Interfaces: What, Why and How”, CMU-CS-02-144, School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213, May 2002.
- [6] Matthew M. Williams, “Throttling Viruses: Restricting propagation to defeat malicious mobile code”, 18th Annual Computer Security Applications Conference, December 2002.
- [7] CAIDA, ICSI, Silicon Defense, UC Berkeley EECS and UC San Diego CSE, “Analysis of the Sapphire Worm”, February 2003.
- [8] Jose Nazario, Jeremy Anderson, Rick Wash and Chris Connelly, “The Future of Internet Worms”, The Black Hat Briefings '01 July 11-12th Las Vegas, July 20, 2001.
- [9] Steven Cheung, et al, “The Design of GrIDS: a graph based intrusion detection system”, Department of Computer Science, University of California at Davis, January 1999.
- [10] 游啓勝、陳奕明, 「COFS：一個具備防治網蟲攻擊能力的合作式防火牆系統」, 第十三屆全國資訊安全會議, 民國九十二年。
- [11] 游啓勝, 「合作式防火牆之設計與應用」, 中央大學資訊管理學系碩士論文, 民國九十二年六月。