利用 Netflow 即時偵測蠕蟲攻擊

徐偉智　王明輝
高雄第一科技大學電腦與通訊工程系
weichih@ccms.nkfust.edu.tw
eagle@mail.nkmu.edu.tw

摘要

在這篇論文，我們發展了一個利用 Cisco Netflow 資料來即時偵測蠕蟲攻擊的方法。利用這個方法，可以在蠕蟲發動掃描攻擊時，有效辨識出發動攻擊的電腦 IP 位址，接著就可以在路由器上將發動攻擊的 IP 位址阻擋掉，避免這些大量的掃描動作影響網路的正常運作。藉由在收到 Netflow 資料之後即時分析，只針對異常的 Netflow 資料進行統計，若統計結果大於預先設定的臨界值，就發出蠕蟲攻擊警報。我們所發展的方法只需要少量的記憶體和 CPU，並且不需蠕蟲的特徵資料。經實驗證實，我們所提出的方法是一個實用的方法。

關鍵字: 蠕蟲，異常偵測，即時

Abstract

In this paper, we develop one method that utilizes Cisco Netflow packets to real-time detect the worm attacks. By this way, we can identify the IP address which launch the worm attack, and block them on the router using access list control to keep the network work smooth. We process the received Netflow data in memory and drop useless data. The method that we develop only needs a small amount of memory and CPU, and does not need the signatures of the worm. By some experiments, the proposed method is proven to be an efficient one.

Keywords: Worm, Anomly Detection, Real-time

1. 前言

自從 CodeRed [1] 這支蠕蟲出現之後，病毒的傳播漸漸不再經由傳統磁片傳染的方式，而是透過網路，因此可能在一夕之間就可以散佈到全球，而且也會降低網路效能。這些病毒為了達到快速傳播的目的，一定要先尋找大量的目標主機，而尋找大量目標的方式不外乎 ICMP Echo 和 SYN scan 兩種方式。雖然現在許多路由器和第三層的交換器都強調使用 ASIC 架構，但是在設計時，都認為 ICMP 的封包量不會很大，所以許多的設備都還是用軟體的方式來處理 ICMP 封包。這時候若是遇上蠕蟲利用 ICMP Echo 的方式來尋找目標，因為一下子有太多的 ICMP 封包需要處理，就會造成設備的 CPU 使用率飆高到幾乎滿載，有些設備就會因此而無法提供服務，而造成網路停擺。另外，如果蠕蟲是利用 SYN scan 的方式來尋找目標，對於那些在路由的處理上是使用 Route Cache [2] 機制的設備，在遇上 SYN scan 的時候，因為每一個封包的目的 IP 都不一樣，使得 Route Cache 失去效用，造成每一個封包都要查詢路由表，因而使得效能大幅下降，甚至到停擺的地步。

傳統的入侵偵測系統是專注在封包層級的分析，需要使用到大量的硬體資源，若運用在高速的骨幹網路上，更需要用到等級非常高且價格昂貴的硬體設備。在這篇論文中，我們提供了一個以分析資料流為基礎的方法，只需要使用一般的電腦即可即時分析高速骨幹網路的資料流。

本論文的編排方式如下：第二節我們會討論如何利用資料流來偵測異常流量。第三節說明我們利用異常的資料流來偵測蠕蟲的方法。第四節應用我們的方法來偵測蠕蟲的攻擊。第五節則是結論。

2. 資料流的異常流量偵測

2.1 資料流的定義

一個資料流是某個來源端與目的端之間的單方向的封包串流。在 IETF 中所定義的資料流記錄 [3] 包含許多欄位，如來源 IP，目的 IP、來源埠號、目的埠號、協定類型、ToS、TCP 旗標等等，其中有七個主要欄位用來識別一個資料流，分別是：來源 IP 位址、目的 IP 位址、來源埠號、目的埠號、協定類型、ToS 值及封包進入路由器的介面。

由這七個欄位，就可以識別出一個獨一無二的資料流，這當中只要有一個不一樣，就是一個不同的資料流。

2.2 異常流量的判斷規則

現今 Cisco 的 Netflow 普遍被用來做學術網路的流量統計工作，一般也都留有歷史資料。Netflow 可以很容易地運用在大型的網路上，因為它不會佔用太多的資源。而 IDS 就需要很強的 CPU 處理能力或是龐大的儲存空間。就 Netflow 的記錄來看，我們雖然不能看到每個封包的內容，但是我們確可以分析每一個 IP 的行為模式，藉此找出行為不正常的 IP。

異常流量分析最常用的方式就是使用基準線的方式。所謂基準線就是設定一個門檻值，只要統計資料超過這個門檻值就可能是異常流量。例如假設一個 IP 在正常情況下，一分鐘所會產生的資料流最高不會超過 300 個，就可以將門檻值設定為 300。如果某一個 IP 在一分鐘之內所產生的資料流超過 300，就可以認定這個 IP 的行為不正常。

這種常用的異常流量判斷方式在現今 P2P 軟體盛行的情況下會容易產生誤判，因為許多 P2P 軟體在短時間之內也都會產生大量的資料流。因此，除了根據基準線的方式來判斷異常流量之外，在我們的方法中還加入了用 TCP 旗標欄位來判斷異常流量的方法。正常的 TCP 通訊要完成 3-Way Handshaking 這三個步驟，才能開始傳送資料。而異常的流量可能無法完成這三個步驟，在 Netflow 的 TCP 旗標欄位中就記錄為 2。所以也可以根據這個欄位值來找出異常的資料流。

2.3 文獻探討

包蒼龍先生於 2004 年發表了一篇「Netflow Based Intrusion Detection System」論文 [4]，是利用 Netflow 的資料來做入侵偵測，先將收集到的 Netflow 資料存放於 MySQL 資料庫中，然後再去分析資料庫中的資料，計算異常流量是否超過門檻值來判斷是否有蠕蟲的攻擊產生，可以偵測 Ping sweep，DOS 和 TCP、UDP 埠號掃描等蠕蟲活動。在偵測到蠕蟲活動之後，這個系統會透過 Telnet 連線的方式，連接到路由器或路由交換器上，設定相關的存取控制指令，將不正常的 IP 阻擋掉。這不是一個即時的系統，而且 Netflow 的資料量長久累積下來是很大的，MySQL 資料庫的效能會受到影響，在論文中也有提到要分析 30 分鐘的資料就不太可能。

另外 Thomas 等人於 2005 年發表的論文 [5] 中也是利用 Netflow 資料來做蠕蟲攻擊的偵測，不過他們主要是提出一個 UPFrame 平台架構，並不是在蠕蟲攻擊的偵測，而且他們所使用的 Cisco 路由器因為硬體上的限制，無法產生 TCP 旗標欄位的值，所以在判斷上就失去了某些依據。

3. 即時蠕蟲偵測

3.1 蠕蟲病毒傳播模式分析

一般蠕蟲的基本程式結構主要分為以下三個模組：

(一) 傳播模組：負責蠕蟲的傳播。

(二) 隱藏模組：侵入主機後，隱藏蠕蟲程序，防止被用戶發現。

(三) 目的功能模組：實現對電腦的控制、監視

或破壞等功能。

而傳播模組又可以分為三個子模組：掃描模組、攻擊模組和複制模組。蠕蟲程式一般的傳播步驟如下：

(一) 掃描：由蠕蟲的掃描功能模組負責探測存在漏洞的主機。當蠕蟲向一個主機發送探測漏洞的封包，並成功收到回應後，就得到一個可傳播的對象。

(二) 攻擊：攻擊模組按漏洞攻擊步驟自動攻擊步驟(一)中找到的對象，取得該主機的權限。

(三) 複製：復製模組通過原主機和新主機的連線，將蠕蟲程式複製到新主機並啟動。

現在流行的蠕蟲採用的傳播技術一般是盡快地傳播到盡量多的電腦中，於是掃描模組採用的掃描策略是隨機選取某一段 IP 位址，然後對這一段 IP 位址上的主機掃描。比較差一點的掃描程序可能會不斷重複上面這一過程。這樣，隨著蠕蟲的傳播，新感染的主機也開始進行這種掃描，這些掃描程序不知道那些位址已經被掃描過，只是簡單地隨機掃描網際網路。於是蠕蟲傳播的愈廣，網路上的掃描封包就愈多。即使掃描程序發出的封包很小，積少成多，大量蠕蟲的掃描引起的網路擁塞就非常嚴重了。

3.2 即時偵測方法

為了達到即時偵測蠕蟲的目的，我們設計了如圖 1 的架構，在路由器上做適當的設定之後，就可以令路由器將 Netflow 資料送到我們的系統中。首先 Netflow duplicator 會先收到路由器送來的封包，然後原封不動地將封包複製兩份，一份送給 Netflow Capture，一份送給 Netflow Receiver。Netflow Capture 會將收到的 Netflow 封包寫入檔案長久保存，可以做為日後分析或產生流量報表之用。Netflow Receiver 則會將 Netflow 封包中的每一筆資料流記錄取出，然後送給 Online Analysis 做即時的線上分析。

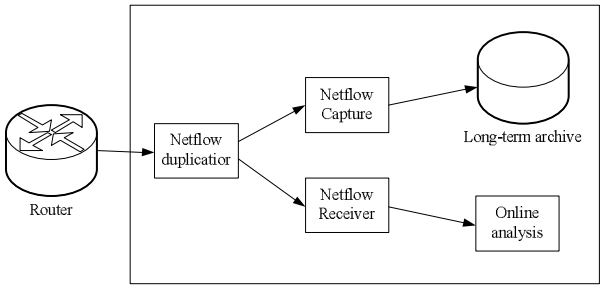

圖 1 系統架構

而線上即時分析的運作流程圖如圖 2 所示。收到的資料流封包立即作檢查和統計的工作，超過時效無用的資料就立即丟棄，以節省記憶體的使用。在系統中，為了達到快速運算的目的，所以在接收到 Netflow 資料後，會先經過一個過濾器，將正常的資料流過濾掉，不要讓它們進入攻擊偵測的程序中，這樣可以大幅加快偵測的速度。而我們在過濾器設置三個過濾條件，分別說明如下：

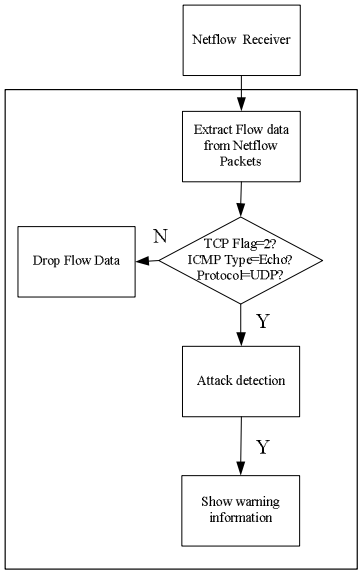

圖 2 系統運作流程圖

(一) **是否為 ICMP 封包，而且 ICMP Type 為 Echo**。蠕蟲為了達到快速傳播的目的，一定要先透過掃描的方式尋找大量的目標，此時就會產生大量的資料流。掃描的方式不外乎兩種，一是利用 ICMP 的 Echo 封包來找出那些 IP 有人在使用，另一種方式是透過 TCP 的

SYN 封包來找出那些 IP 有人在使用。因為一般電腦在正常使用下，並不會產生大量的 ICMP Echo 封包，所以我們會對發出 ICMP Echo 的 IP 加以研判是否發出攻擊。

(二) **TCP Flag 欄位的值是否為 2**。若蠕蟲是透過 SYN 封包來掃描，就要配合 TCP 旗標欄位來找出這些異常流量。因為蠕蟲是對大量的 IP 發出 SYN 封包，但是並不是每個 IP 都會回應，所以此時就會產生許多沒有收到回應的 SYN 資料流。Cisco 的 Netflow 對於發出 SYN 封包但是沒有回應的資料流，會將其 TCP 旗標欄位的值設為 2，因此我們只要判斷這個欄位的值是否為 2，就可以知道是否為異常資料流。

(三) **是否為 UDP 資料流**。由於目前尚未找到可以立即有效判別 UDP 資料流是否為異常資料流的方法，所以 UDP 資料流一律要檢查。

經過這三個條件的過濾之後，可以大幅減需要過濾的資料流，除了加快偵測的速度外，也可以減少偵測時的記憶體使用量。接下來經過過濾之後的資料流資料會送入 Attack Detection 程序，在偵測程序中，我們準備了兩個 Hash Table，一個存放當前這一分鐘的統計資料，一個存放前一分鐘的統計資料。然後根據資料流中的時間欄位的值，算出資料流產生的時間，將其放入對應的 Hash Table 中，在一分鐘過後，會將前一分鐘的資料釋放掉，然後再產一個新的 Hash Table 來存放當前這一分鐘內的資料流資料。我們利用下列四種方式來判斷攻擊行為：

(一) ICMP 掃描。ICMP 掃描的判斷比較簡單。在收到資料流的資料時，會先判斷是否為 ICMP Echo 資料流，若是的話就進入 ICMP 掃描的偵測程序，若不是的話就接著執行其它的判斷程序。

因為一般電腦在正常使用下，並不會產生大量的 ICMP Echo 資料流，而我們也時實際用 ping 程式測試對某一台電腦持續發出 ICMP Echo 封包，在路由器所送出的 Netflow 記錄中，只計算為一筆資料流而已，所以只要使用基準線的方式，設定好適當的門檻值，就可以找出這些發動攻擊的 IP。經由歷史資料的統計分析，我們發現在正常使用下，一台電腦機乎不會發出 ICMP 封包，而蠕蟲在掃描時，一分鐘可以發出數百個 ICMP 封包。不過現在的蠕蟲更為聰明，它們會控制每分鐘發出的封包數不要太多，以免容易被使用者查覺，所以我們設計了一個可以每分鐘發出大約 60 個 ICMP Echo 資料流的程式，總共使用 50 個來源 IP 在實際的網路環境上做測試，將門檻值分別定為 10、20、30、40、50、60，用來測試那一個門檻值偵測的效果最好，得到的結果如圖 3 所示。因為我們只針對 ICMP Echo 資料流做判斷，其它類型的 ICMP 資料流都不採計，所以誤判率為 0。

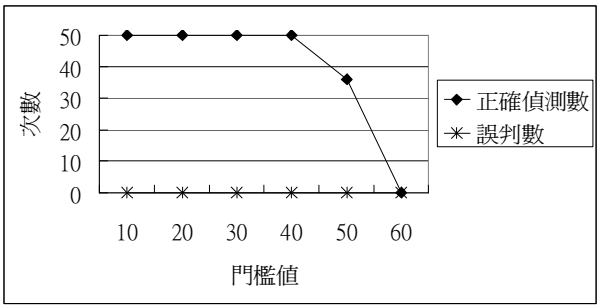

圖 3 不同的門檻值與偵測的結果

由實驗的結果得知，門檻值設為 10 到 40 之間都可以正確偵測到所有的 ICMP 掃描，而且不會有誤判，但是超過 40 之後就開始有漏失的情形產生。為了減少以後可能發生的誤判，並兼顧偵測的時效性，所以我們決定將門檻值設定為 20。

(二) TCP SYN 掃描。若蠕蟲是透過 SYN 封包來掃描，就會產生許多 TCP 旗標欄位為 2 的資料流，加上門檻值的適當設定，就可以找出這些異常的 IP。經由統計資料得知，一般電腦在正常使用下，不會有 TCP 旗標欄位為 2 的資

料流產生，而正在發動攻擊的電腦，平均一分鐘可以產生大約 3000 個 TCP 旗標欄位為 2 的資料流，因此我們將門檻值設定為 3000 的十分之一，300。一台電腦在一分鐘之內發出超過 300 個 TCP 旗標欄位為 2 的資料流，就判定其在發動攻擊。

(三) UDP Flood 攻擊。這是向某一台電腦大量發出 UDP 封包的 DOS 攻擊方式。因為在我們的歷史資料中並沒有找到 UDP Flood 攻擊的記錄，所以我們自已寫了一支可以發出 UDP Flood 攻擊的程式來統計一台電腦在一分鐘內可以發出多少個 UDP 封包。經由測試，一台電腦一分鐘內可以發出 260 萬個以上的 UDP 封包，但是 Netflow 的資料抓到的平均只有 18000 筆，因為目的埠相同的封包會被 Netflow 歸類為同一筆資料。當然，UDP Flood 不一定攻擊同一台電腦，也可能亂數攻擊不同的電腦，所以我們定了以下的規則來偵測 UDP Flood 攻擊：

(1) 同一組來源和目的 IP，但是目的埠不同的資料流超過 50 以上。
(2) 來源相同，但是目的 IP 和目的埠皆不同的資料流超過 700 以上。

50 和 700 是我們實驗出來的數據，我們同樣也是用 50 個 IP 來發動 UDP 攻擊，條件 (1) 的門檻值分別設為 10、20、30 到 60 來測試，條件 (2) 的門檻值分別設 200、300、400、500、600、700 來測試，最後得到的結果是條件(1)和條件(2)的門檻值分別設為 50 和 700 就可以找出發動攻擊的 IP，而且不會誤判。產生誤判的原因大多是有人使用 P2P 軟體所產生的資料流。

(四) SYN Flood 攻擊。這也是 DOS 攻擊的一種，就是利用假的來源 IP 對伺服器發出大量的 SYN 封包，讓伺服器無法提供正常的服務。由於我們主要是偵測內部的蠕蟲攻擊，所以只要判斷來源 IP 和目的 IP 都不在我們的內部 IP 範圍內，就可以肯定是 SYN Flood 的攻擊。

4. 偵測實際的蠕蟲攻擊

4.1 已知蠕蟲的偵測

首先，我們由圖 4 的統計圖表看出，在某一天的大約 9 點開始有攻擊 TCP 埠號 139 的蠕蟲出現，異常流量大量增加。

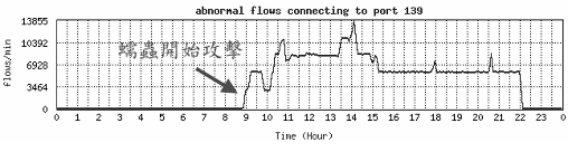

圖 4 連線到 TCP 埠號 139 的異常流量統計

會攻擊 TCP 埠號 139 的蠕蟲有不少個，如 W32.HLLW.Deborms.D [6] 和 W32.Secefa.A [7] 等等。是什麼樣的蠕蟲並不重要，重要的是找出那些 IP 在攻擊。本實驗利用 flow-tools [8] 中的工具程式 flow-receive 和 flow-print，配合 Perl 程式來做 Netflow 封包的即時處理。我們連續以三天的歷史資料來做測試，將 IP 發動攻擊的時間與被偵測到的時間差做成一個統計圖表，如圖 5 所示。由統計圖中可以看出，所有的攻擊 IP 都可以在收到 Netflow 資料後的 1 分鐘內被偵測到。

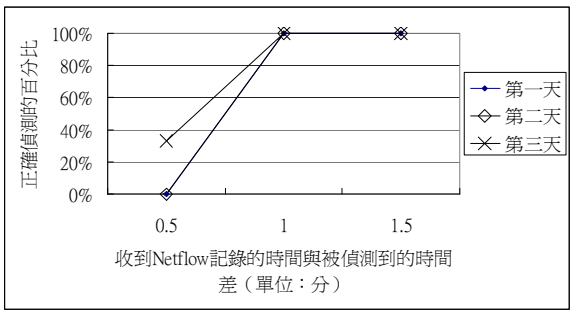

圖 5 收到已知蠕蟲的 Netflow 記錄的時間與偵測到的時間差

4.2 UDP 攻擊的偵測

除了偵測真實的蠕蟲之外，我們也自己寫了一個以亂數攻擊 UDP 埠號的程式來測試我們的偵測系統是否能正確偵測出來。我們用 50 個 IP 同時發動攻擊，連續做三次，畫出的平均統計圖如圖 6 所示。也都可以在 1 分鐘之內找出攻擊 IP。

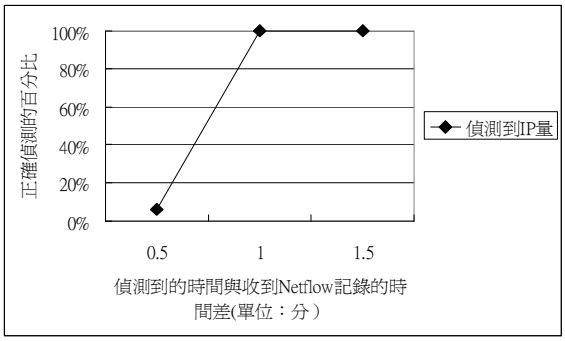

圖 6 收到 UDP 攻擊記錄的時間與偵測到的時間差

4.3 DOS 攻擊的偵測

DOS 攻擊的偵測很容易，我們自己寫了一個 SYN Flood 的程式來攻擊某一台主機，也可以在 1 分鐘之內偵測出來。我們同時對 50 個 IP 發動攻擊，將發動攻擊與偵測到的時間差畫出如表 1 之統計圖表。同樣都可以在 1 分鐘內找到被攻擊的 IP。

表 1 收到 DOS 攻擊記錄的時間與偵測到的時間差

偵測到的時間與收到 Netflow 記錄的時間差(單位：分）	0.5	1	1.5
累積正確偵測的比率	100%	100%	100%

5. 結論

在這篇論文，我們發展了一個利用 Cisco Netflow 資料來即時偵測蠕蟲攻擊的方法，並詳細說明了這個方式的運作過程及各種方式所設定的門檻值。利用這個方法，可以在蠕蟲發動埠號掃描攻擊時，辨識出發動攻擊的電腦 IP 位址，接著就可以在路由器上將發動攻擊的 IP 位址阻擋掉，避免這些大量的掃描動作影響網路的正常運作。我們所發展的系統硬體需求不高，而且不會影響網路的效能，可以用於第一線的偵測。經由實際的測試，確實可以在短時間內找出發動攻擊的電腦 IP 位址，並將其阻擋掉，所以網路的效能和伺服器群都不會受到蠕蟲攻擊的影響。

參考文獻

[1] CodeRed Worm. http://securityresponse.symantec.com/avcenter/venc/data/codered.worm.html, 2001.

[2] Route Cache. http://linux-ip.net/html/routing-cache.html.

[3] RFC 3954. http://www.faqs.org/rfc/rfc3954.txt

[4] Tsang-Long Pao and Po-Wei Wang, “Netflow Based Intrusion Detection System,” IEEE International Conference on Networking, Sensing & Control. Taipei, Taiwan, March 21-23, 2004

[5] Thomas Dubendorfer, Arno Wagner and Bernhard Plattner, “A Framework for Real-Time Worm Attack detection and Backbone Monitoring,” Proceedings of the 2005 First IEEE International Workshop on Critical Infrastructure Protection

[6] Symantec Security Response. W32.HLLW.Deborms.D. http://securityresponse.symantec.com/avcenter/venc/data/w32.hllw.deborms.d.html, 2003.

[7] Symantec Security Response. W32.Secefa.A. http://securityresponse.symantec.com/avcenter/venc/data/w32.secefa.a.html 2005.

[8] flow-tools. http://www.splintered.net/sw/flow-tools/