行政院國家科學委員會專題研究計畫 成果報告

可防禦分散式 DDoS 攻擊的異質性追蹤器布建問題之研究
研究成果報告(精簡版)

計 畫 類 別 ：個別型
計 畫 編 號 ：NSC 97-2221-E-216-035-
執 行 期 間 ：97 年 08 月 01 日至 98 年 07 月 31 日
執 行 單 位 ：中華大學資訊工程學系

計 畫 主 持 人：王俊鑫

計畫參與人員：碩士班研究生-兼任助理人員：陳弘昌
碩士班研究生-兼任助理人員：陳文誠

處 理 方 式 ：本計畫可公開查詢

中 華 民 國 98 年 10 月 30 日

行政院國家科學委員會補助專題研究計畫 □成果報告

可防禦分散式DDoS攻擊的異質性追蹤器布建問題之研究

計畫類別：□ 個別型計畫
計畫編號：NSC 97-2221-E-216-035
執行期間：97 年 8 月 1 日至 98 年 7 月 31 日

計畫主持人： 王俊鑫
計畫參與人員： 陳文誠, 陳弘昌

成果報告類型(依經費核定清單規定繳交)：□精簡報告

本成果報告包括以下應繳交之附件：
□發表之論文一份

處理方式：本計畫可立即公開查詢

執行單位：中華大學資訊工程學系

中 華 民 國 98 年 10 月 25 日

行政院國家科學委員會專題研究計畫成果報告

可防禦分散式DDoS攻擊的異質性追蹤器布建問題之研究

計畫編號：NSC 97-2221-E-216-035
執行期限：97 年 8 月 1 日至 98 年 7 月 31 日
主持人：王俊鑫助理教授　中華大學資訊工程學系
E-mail: chwang@chu.edu.tw

中文摘要

要有效解決 DoS/DDoS 的問題，首先需找到攻擊來源，並協同鄰近的具有封包過濾功能的路由器，即時的過濾異常封包，才能避免攻擊來源與受害者之間的網路頻寬被佔用。但原有的路由器，並不具備追蹤、過濾封包等功能，我們必須在路由器增加新的功能來支援，新增功能的路由器，我們以追蹤器來統稱。我們依據追蹤器的特性，以成本及必要性，歸類選擇三種異質性的追蹤器，tunneling-enabled tracers、marking-enabled tracers 及 filtering-enabled tracers，用來防禦 DoS/DDoS 的攻擊。其中 tunneling-enabled tracers 的成本最低且可輕易的將封包繞路導向，因此在計畫中，我們研究如何有效的利用 tunneling-enabled tracers 適時的將封包導向最佳的 marking-enabled tracers 或 filtering-enabled tracers 以進行來源追蹤及即時過濾異常封包，我們提出四種方法並與最佳解相互比較。藉由模擬結果，發現藉由 marking-enabled tracers 輔助的第四種方法有最佳的執行成效。

Abstract

To solve the DoS/DDoS problems efficiently, the first things is to locate the attack origins and then cooperate the filtering-enabled routers nearby to filter the abnormal packets in time. But the original routers can't provide these functions such as tracking, filtering, and etc. They have to be enhanced with additional functions to defense DoS/DDoS attacks. We refer the enhanced routers to as tracers. According to the characteristic, cost and necessity of tracers, we classify and select three kinds of heterogeneous tracers, namely tunneling-enabled tracers, marking-enabled tracers and filtering-enabled tracers to defense DoS/DDoS attacks. The tunneling-enabled tracers have the lowest cost than the others and they can alter the path of the passing packets to destination easily. In this project, we study how to use tunneling-enabled tracers efficiently to forward packets to the best candidate of marking-enabled tracers or filtering-enabled tracers for locating attack origins and filtering abnormal packets in time. Four methods are proposed and compared with the optimal solution. The fourth method with the assistance of marking-enabled tracers has the best performance by simulation result.

關鍵字：DoS/DDoS, tracers

一、前言與研究目的

隨著網路技術的蓬勃發展，網路成為現今人們重要的溝通工具。但是隨伴而來的網路安全問題，確日益嚴重。從內部網路安全問題(Intranet security)，到網際網路安全問題(Internet security)；如阻斷服務攻擊(Denial of Service, DoS)、分散式阻斷服務攻擊

(Distributed of Service, DDoS)[1-3]、蠕蟲式網路病毒的蔓延、電腦系統的漏洞等。在各種問題中，以佔用主機的資源或網路頻寬，導致主機無法對正常使用者提供服務的阻斷服務攻擊，最為嚴重與難解。因此，如何有效的解決阻斷服務攻擊及分散式阻斷攻擊的網路安全問題，實刻不容緩，為一重要研究課題。

追蹤攻擊來源，除可作為事後舉證外，但更重要的目的是可協同在攻擊來源附近具有封包過濾器(filter)，即時的過濾攻擊封包，以避免攻擊來源與受害端之間的網路頻寬，遭受攻擊封包的佔用，而達減緩 DDoS 攻擊的目的。有許多的文獻[4-21]，探討如何追蹤攻擊來源，稱為 IP 來源追蹤(IP Traceback)，透過 IP 來源追蹤的技術，不一定能找到真正幕後的攻擊者，但至少能找到幾近於攻擊來源，直接進行攻擊的主機。

在現行網路，路由器只負責封包轉送的動作，為了支援追蹤來源 IP 的方法，必須增加路由器原有的功能，例如記錄封包內容摘要(Digest)資訊，或者將路由器資訊寫入到封包內等功能，具有支援追蹤來源 IP 功能的路由器，我們稱為追蹤器(Tracer)。在本計劃中，我們檢視 IP 來源追蹤的相關文獻[4-21]，依追蹤器的實際可行性及必要性，考量下列的異質性的追蹤器的環境下，探究如何有效的防禦 DDoS 的攻擊：

- Tunneling-enabled tracers : 具有 IP-in-IP tunneling 將封包封裝 (encapsulation)及解封裝(decanpulation)的功能，如文獻[19-21]。
- Filtering-enabled tracers : 接受受害端發佈命令，更新及辨識攻擊封包的特徵值，依封包的特徵值或來源端的 IP 位址等規則，進行封包過濾，如文獻[14,20,21]。
- Marking-enabled tracers : 可將部份的路由器 IP 位址或連結(link)的資訊註記在經過的封包內，受害端收到足夠被標記的封包，可以還原攻擊路徑，如文獻[4-10]。

依追蹤器的成本考量及升級的難易度，tunneling-enabled tracers 的成本最少，所以數量應最多，marking-enabled tracers 次之，而 filtering-enabled tracers 成本最高，但有其必要性，所以只有少部份的路由器升級為過濾器。tunneling-enabled tracers 可將封包 IP tunneling 封裝後做繞路導向傳送至 marking-enabled tracers 或 filtering-enabled tracers， 而 marking-enabled tracers 與 filtering-enabled tracers 可將接收的封包，進行解封裝後(decapsulation)，再依路由表(routing table)將封包往目的端傳送。tunneling-enabled tracers，可輕易的將封包導向，但值得我們注意的是 tunneling 的邊際效應，封包可能會因此，繞行較遠的路徑才到達目的地，為了追蹤攻擊來源或過濾異常封包，將封包導向 marking-enabled tracers 或 filtering-enabled tracers，迫使正常的封包，需繞行較遠的路徑到達目的地，所負出的代價，為佔用網路頻寬的資源，因此我們必須思考，如何適時的利用 tunneling-enabled tracers 將封包做導向，如何選擇最佳的 marking-enabled tracers 及 filtering-enabled tracers 作為導向目標，才能以最少的代價，有效的追蹤攻擊來源，及有效的過濾攻擊封包，而且 marking-enabled 與 filtering-enabled tracers 的布建問題，亦是影響效能的關鍵問題，值得我們深入的探究。

二、 文獻探討

有許多的文獻[4-19]，探討如何追蹤攻擊來源，稱為 IP 來源追蹤(IP Traceback)；但透過 IP 來源追蹤，不一定能找到真正幕後的攻擊者，但至少能找到幾近於攻擊來源，甚至直接進行攻擊的主機。目前有關 IP 來源追蹤的研究，大致可區分為四類：(1)IP Marking

[4-10] (2)ICMP Traceback [11-14] (3)Logging-based[15-18] (4)Overlay Networks [19] 。第一類的方法[4-10] ，路由器以機率的方式選擇性的在經過的封包中加入適當的標記，這類方法優點是路由器的負擔較輕，但最大的缺點，受害者需要收集足夠被標記的封包與需較長的時間來重建攻擊來源的路徑，有部份的學者，提出改進的方法，只在邊界的路由器(edge router)進行封包的標記[4]，以減少須要收集被標記的封包，但這樣的方法不適用當攻擊來源來自於網路的核心部份(如 ISP (Internet Service Provider)的內部節點)。第二類的方法[11-14]，每一個路由器以機率的方式，選擇性的將經過的封包與路由器前後的連結的路徑的相關資料透過 ICMP 的訊息傳送到受害者，但 ICMP 的訊息可能被網路上的封包過濾器(如防火牆)過濾掉，而且攻擊者易假造錯誤的 ICMP 的訊息來干擾受害者重建攻擊來源的路徑。第三類的方法[15-18]，則路由器須記錄經過封包的資料，雖然可以做攻擊事後的追蹤，但路由器的負擔過重且需要較多的硬體資源來存放封包的資料。第四類，以 IP-in-IP tunneling 的方式建構覆蓋網路(Overlay Networks)[19]，所有的封包經過邊界路由器，會因 IP tunneling 轉送到網路內部的 Tracking routers，由 tracing routers 最後將封包轉送到目的端，如此可藉由 tracing routers 追蹤封包進入網路的邊界路由器，但當攻擊來源，來自於網路的內部節點，亦無從追蹤。

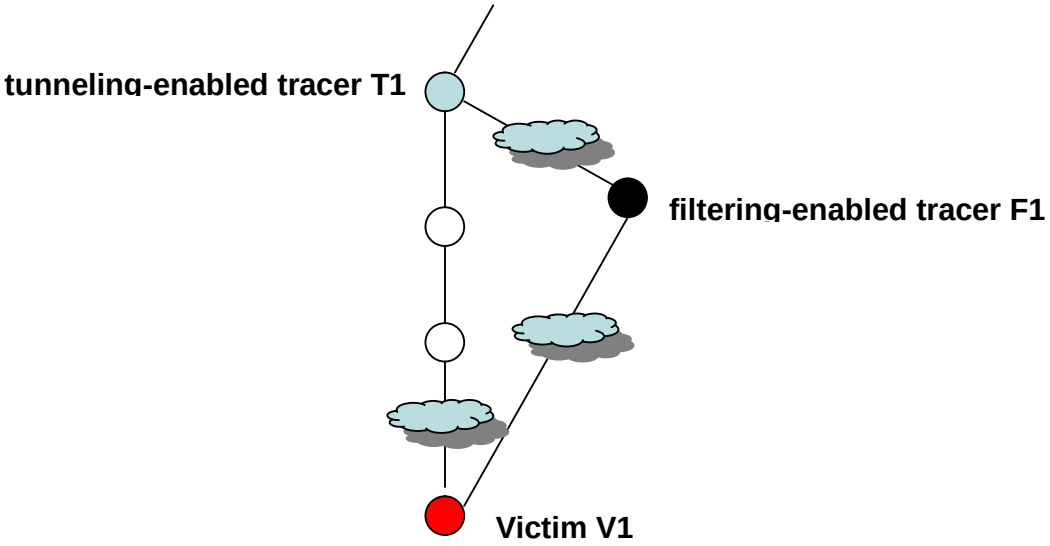
追蹤攻擊來源，除可作為事後舉證外，但更重要的目的是可協同在攻擊來源附近具有封包過濾器(filter)，即時的過濾攻擊封包，以避免攻擊來源與受害端之間的網路頻寬，遭受攻擊封包的佔用，而達減緩 DDoS 攻擊的目的。目前僅有少數的文獻中[14, 22, 23]，探討如何利用封包過濾器，來防禦 DDoS 的攻擊。

由上述的文獻的探討中，我們可以看出，攻擊封包可以被過濾的先決條件為，攻擊封包，需經過封包過濾器，才有機會被過濾掉。若封包過濾器設置在欲保護對象的防火牆，可以確保攻擊封包一定會經過封包過濾器，但太靠近受害端無法保護攻擊來源與受害端之間的網路頻寬，換言之無法有效的減緩 DDoS 攻擊，因此，如何有效的減緩 DDoS 攻擊，封包過濾器的布建問題實為一大挑戰，雖然我們可以藉由 IP 來源追蹤的方法，找到攻擊來源(或接近攻擊來源)，但當過濾器不在攻擊的路徑上的環境下，如何協同鄰近的封包過濾器來進行過濾，為本計畫的研究重點。

三、 研究方法與模擬結果

(a)研究方法

我們假設在受害端可偵測到攻擊現象，而且可以分析出攻擊封包的特徵值，受害端可將封包的特徵值通知相關的 filtering-enabled tracers，讓 filtering-enabled tracers 可依封包的特徵值進行封包過濾，但當攻擊封包的傳送路徑未經過任何 filtering-enabled tracers，我們可以利用網路上佔有較大比例的 tunneling-enabled tracers 將封包導向 filtering-enabled tracers 進行封包過濾，雖然異常封包在 filtering-enabled tracers 可因此被過濾，但同時會使得正常封包繞路， 因此若接受通知 tunneling-enabled tracers 不當的選擇 filtering-enabled tracers 做為封包導向的目標，可能因正常封包繞路，佔用網路頻寬資源過多，反而得不償失。因此，我們首先訂定下列的成本函數，作為 tunneling-enabled tracers 如何選擇最佳的 filtering-enabled tracer(s)的依據。



圖一. 過濾追蹤器的選擇

- 假設 T1 選擇 F1 做為封包導向目標
- 假設經過 T1 到達 vitcim 的交通流量為 $\mathbf{D_{t1}}$，其中正常交通流量的比例為 α，則異常的交通流量的比例為(1-α)
- 網路節點 x, y 的距離(hops)，以 d(x,y)表示
- 異常的交通流量因被導向而過濾，可節省的網路頻寬的量定義為 Ra
- 正常交通流量因被導向而繞路，須額外佔用網路頻寬的量定義為 An

以圖一的例子，我們可以算出

- Ra= $\mathbf{D_{t1}}$*(**1**-α)*(d(T1,V1)-d(T1,F1))
- An= $\mathbf{D_{t1}}$ *α*(d(T1,F1)+d(F1,V1))-d(T1,V1))

Ra 愈大愈好，An 愈小愈好，而(Ra-An)表因封包導向過濾，最後可節省的網路頻寬資源，換句話說，以最大化(Ra-An)的值作為 filtering-enabled tracer 的最佳選擇。

為避免過多的正常封包被 tunneling-enabled tracers 導向 filtering-enabled tracer，導致正常封包因繞路而額外佔用網路頻寬，尤其當攻擊封包的量少時，大部份的正常封包繞路，反而得不償失，正常的封包事實上不應被導向，但封包的異常與否對 tunneling-enabled tracers 無法得知，所以問題的關鍵在 tunneling-enabled tracers 如何決定是否對流經的封包進行導向到鄰近的 filtering-enabled tracer，因此我們提出下列的決策方法:

❒ All-tunneling:

追蹤器只有兩種類型,tunneling-enabled tracers 與 filtering-enabled tracers，all-tunneling 的方法為 tunneling-enabled tracers 對經過的封包均進行導向至鄰近的 filtering-enabled tracer，用以觀察當網路攻擊量少時，事實上，all-tunneling 反而增加網路的負擔。

❒ 50%-tunneling:

追蹤器亦只有兩種類型,tunneling-enabled tracers 與 filtering-enabled tracers，50%-tunneling 的方法為 tunneling-enabled tracers 對經過的封包以 50%的機率進行導向至

鄰近的 filtering-enabled tracer，封包有一半的機率會被導向。

❒ Dynamic-tunneling:

只有 filtering-enabled tracers 才有能力識別封包的異常並過濾，若在 filtering-enabled tracers 上對被 tunneling-enabled tracers 導向的封包，作簡單的統計，計算異常封包的比例,並將統計結果回饋提供給相關的 tunneling-enabled tracers 作為決定是否要將經過的封包進行導向的依據，換言之，tunneling-enabled tracers 可藉由其鄰近的 filtering-enable tracers 的幫忙，以經過封包中異常封包比例的歷史資料動態的來決策。

❒ Marking assistance:

若追蹤器只有兩種類型,tunneling-enabled tracers 與 filtering-enabled tracers,是無法得知攻擊來源的落點，因此我們加上 marking-enabled tracers 的幫忙，利用我們的研究成果[20]所提出的方法，可以追蹤至離攻擊來源最接近的 marking-enabled 的追蹤器，亦可得知攻擊封包經過的路徑，所以可以關閉不在攻擊路徑上的 tunneling-enabled tracers，以節省路由器的負擔並避免正常封封包被導向。

在模擬上述的方法中，我們統計所有封包從來源端，往目的端(victim)傳送所經過的總 hop count 的個數為網路成本(攻擊封包經過或被導向 filtering-enabled tracers 會被過濾掉)，來評量所提出的方法之效能。除了以不布建任何追蹤器的環境下，封包所經過的總 hop count 的個數(original method)為基礎，進行比較外，我們亦假設若 tunneling-enabled tracers 有識別封包的異常的能力，換句話說，tunneling-enabled tracers 不會對正常封包進行導向，只會對攻擊封包進行導向，所量測的結果可視為最佳解(optimal solution)，用以評估所提出方法的好壞的程度。

(b)模擬結果

我們採用由 skitter[22]所量測的實際網路拓樸資料庫，來產生我們所需要的網路拓樸結構，每次以 15 張網路拓樸結構進行模擬，每一個網路節點代表一個路由器，每邊的成本都是 1 hop 距離，封包以最短路徑來傳送;在我們的模擬實驗不考量傳送路徑改變的影響。

模擬的網路節點數從 1000 到 5000 點，每次隨機取 100 個攻擊節點，模擬 10000 個封包,對每一封包, 若正常封包, 隨機選擇來源點, 若攻擊封包, 則由攻擊點中隨機選擇作為來源點，攻擊封包的比例由 30%到 90%，藉以觀察所提出方法的效能。

因追蹤器採用部份布建的方式，所有追蹤器佔總節點數的比例為 50%，各種追蹤器的位置為隨意布建，三種不同性質的追蹤器，再以下列的比例進行模擬實驗:

Kinds of Tracers / test	Test 1	Test 2	Test3	Test4	Test5	Test 6	Test 7
F (Filtering-enabled tracers)	20%	20%	20%	20%	30%	30%	40%
M (Marking-enabled tracers)	0%	20%	30%	40%	30%	40%	40%
T (Tunneling –enabled tracers)	80%	60%	50%	40%	40%	30%	20%

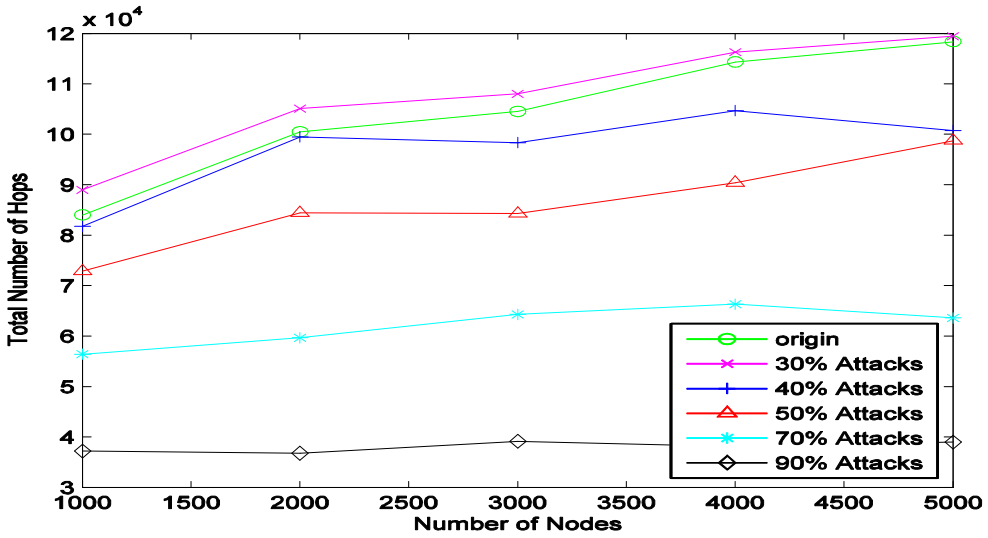


圖 3.1：比例 80%T, 20% F 之下，各種攻擊流量比例下, all- tunneling 的成本比較

●All-tunneling 方法的結果:

假設網路上的追蹤器只有兩種：Tunneling -enabled tracers 及 Filtering-enabled tracers，在無從得知封包的好壞情況，Tunneling -enabled tracers 將所有經過的封包都導向轉送至鄰近的 Filtering-enabled tracers，由圖 3.1，可觀察在攻擊比例不高的情況 (30%的攻擊封包)，all-tunneling 後網路成本卻超過原成本，當攻擊比例增加 40%以上，總成本才會大幅減少。這是由於 all- tunneling 的方式，在攻擊比例不高迫使大部份正常的封包，需繞行較遠的路徑才能到達目的地，因此增加了總成本的關係。

在 30%的攻擊封包下，我們試著改變 Tunneling -enabled tracers 的比例，觀察 all-tunneling 是否會得到改善。圖 3.2 顯示各種比例，網路成本都是在原成本附近跳動，亦會出現多於原成本的情況,可預見很難以調整 Tunneling -enabled tracers 的比例來改善在 30%攻擊封包，all- tunneling 造成成本過高的問題。

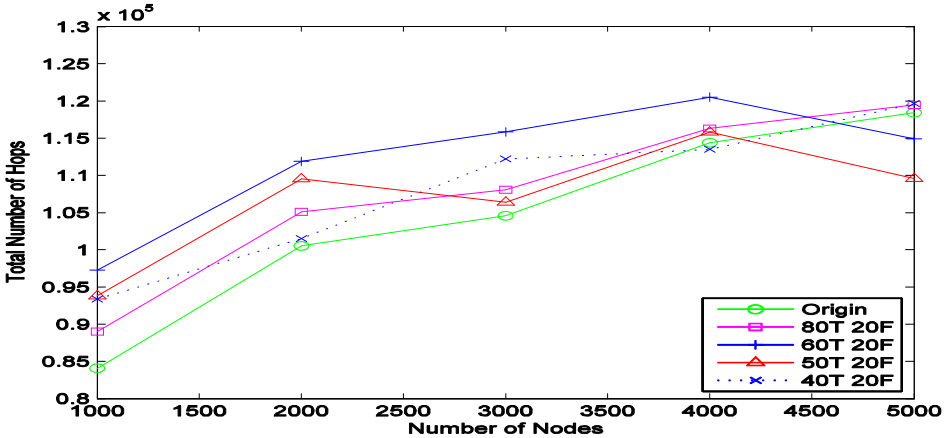


圖 3.2: 各種 tunneling -enabled tracers 比例，all tunneling 在 30%的攻擊封包下的結果

●50%-tunneling 方法的結果:

為改善攻擊比例不高的情況，all- tunneling 的網路成本卻超過原成本的問題，50% -tunneling 方法是 tunneling –enabled tracers 以一半的機率將封包轉送至鄰近的 filtering-enabled tracers，圖 3.3 顯示在低攻擊比例環境之下(30%)，50%- tunneling 優於

all- tunneling 的方法。而不同比例的 tunneling –enabled tracers 的個數，50%- tunneling 的方法，大致上來說，有較好的總成本。但與最佳解的成本的比較，可以看出 50% -tunneling 方法仍有很大的改善空間。

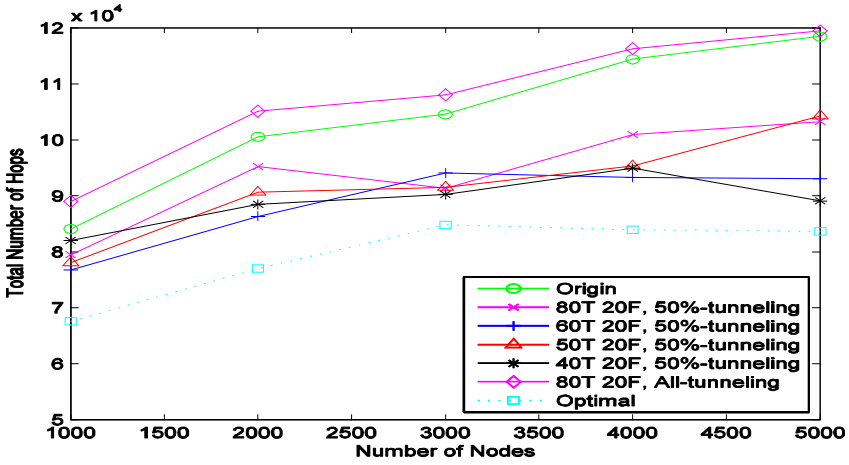


圖 3.3：50%-tunneling 在各種 tunneling –enabled tracers 比例與最佳解的比較

●Dynamic-tunneling 方法的結果:

當攻擊發生時，並非所有的路由器上都會有相同比例的攻擊封包經過，因此 tunneling –enabled tracer 以其鄰近 filtering-enabled tracer 統計收到的封包中為攻擊封包的比例，動態機率決定是否要將經過的封包進行導向轉送，由圖 3.4 顯示在攻擊比例為 30 %的情況，dynamic-tunneling 比 all-tunneling 及 50%-tunneling 來的好，而由改變 tunneling-enabled tracers 數量的比例，進行 dynamic-tunneling 的實驗，並不能選出適用於不同網路節點數的最佳比例，但 40%的 tunneling-enabled tracers 較接近最佳解。

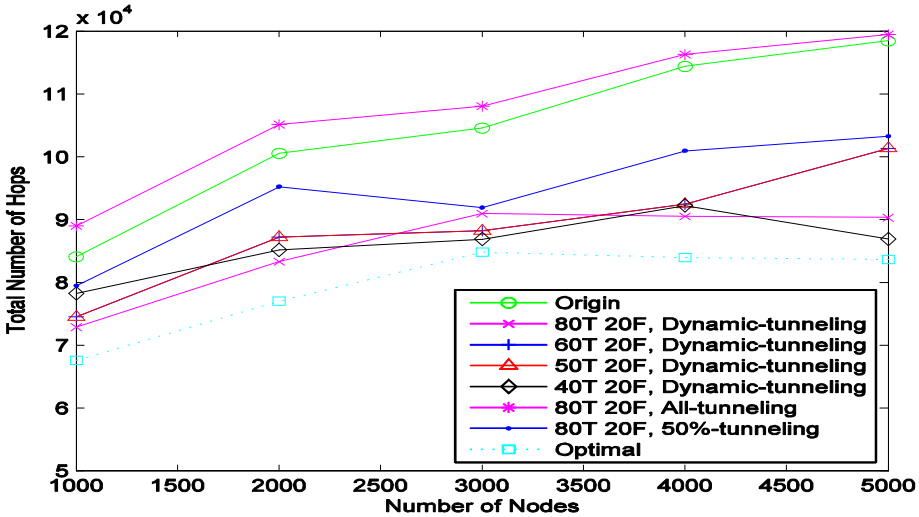


圖 3.4：在攻擊量 30%之下，dynamic-tunneling 各路由器比例比較

●Marking assistance 方法的結果:

在路由器中布建了部份比例 marking-enable tracers，marking-enable tracers 可對經過的封包做標記，在受害端收集足夠被標記的封包，可找出攻擊路徑，進而找到最靠近攻擊來源的 marking-enable tracer，利用這項資訊，讓不在攻擊路徑上的 tunneling-enabled

tracers 關閉導向轉送的功能，以節省路由器的負擔。

圖 3.5 為加入了 marking-enable tracers，在攻擊封包 30%比例下，60%, T20%, F20M 比例之下，與其它各種方法做比較，dynamic-tunneling 與 marking assistance 的網路成本非常相近，但加入 marking-enable tracers 後會更好一些，也更為接近最佳的成本。

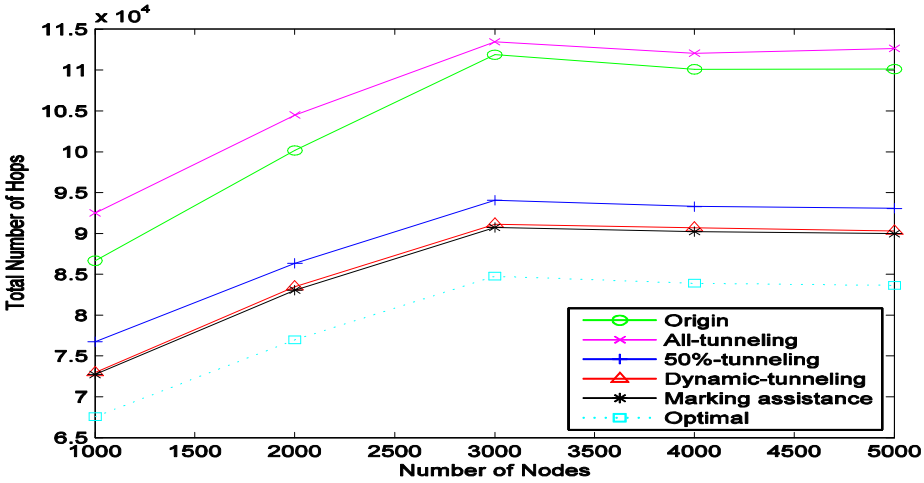


圖 3.5：60%, T20%, F20M 比例之下，marking-assistance 的效能

圖 3.6 為網路節點 5000 點時，攻擊封包的比例從 30%到 90%，各種方法的總成本表現。60%T、20%F、20%M，在攻擊量 30%時 all-tunneling 的成本多於原成本，但攻擊量 40%以上，all-tunneling 情況會有大幅的改善，dynamic-tunneling 以及 marking assistance 則更為趨近於最佳化的狀況。

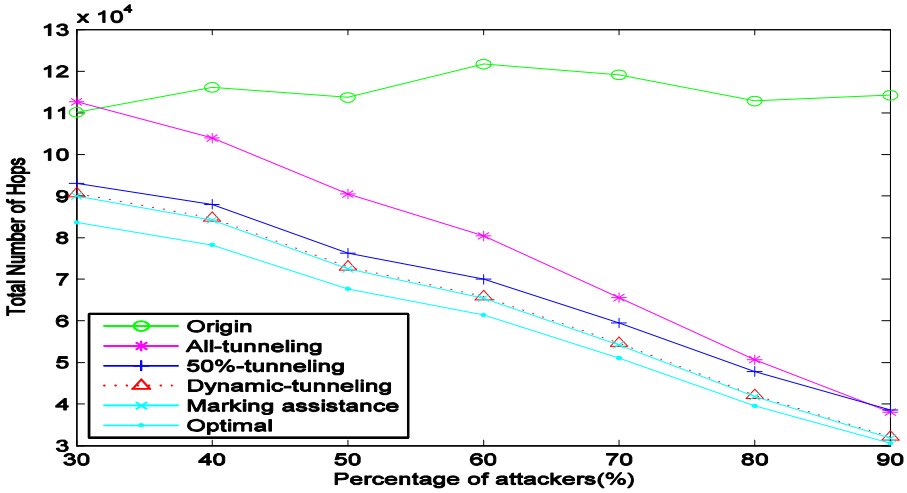


圖 3.6：60T20F20M 中，在不同攻擊比例下，四種方法的總成本與最佳解的比較

藉由 marking-enabled tracers 的幫助下，分辨出可能的攻擊路徑，進而決定將不在攻擊路徑上 tunneling-enabled tracers 轉為「turn off」的狀態，因此除了降低總成本，也可以減少追蹤器的啟用率（active ratio of tracers）。如圖 3.7，除了 marking assistance 外，其它方法的啟動率都是 100%。圖 3.8 中在點數為 5000 點時，不同的攻擊比例底下，tunneling-enabled tracers 的啟動率的可減少多於 20%。

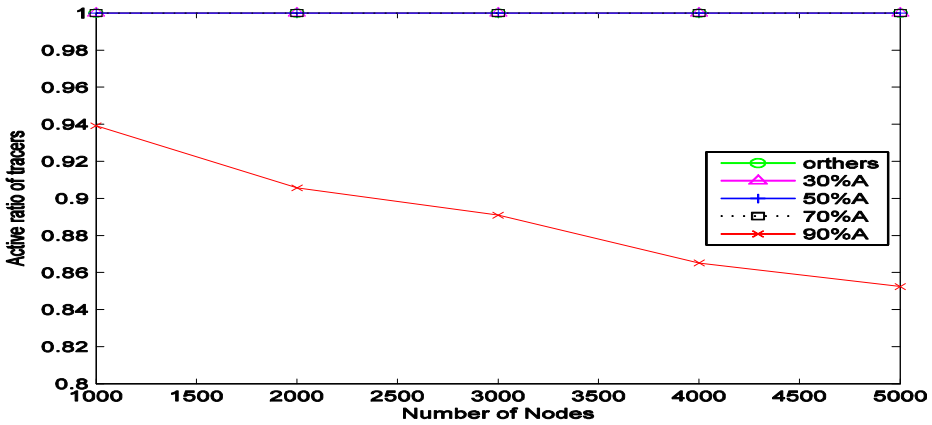


圖 3.7：60%T, 20%F, 20%M，在 50%攻擊量之下，追蹤器的啟動率

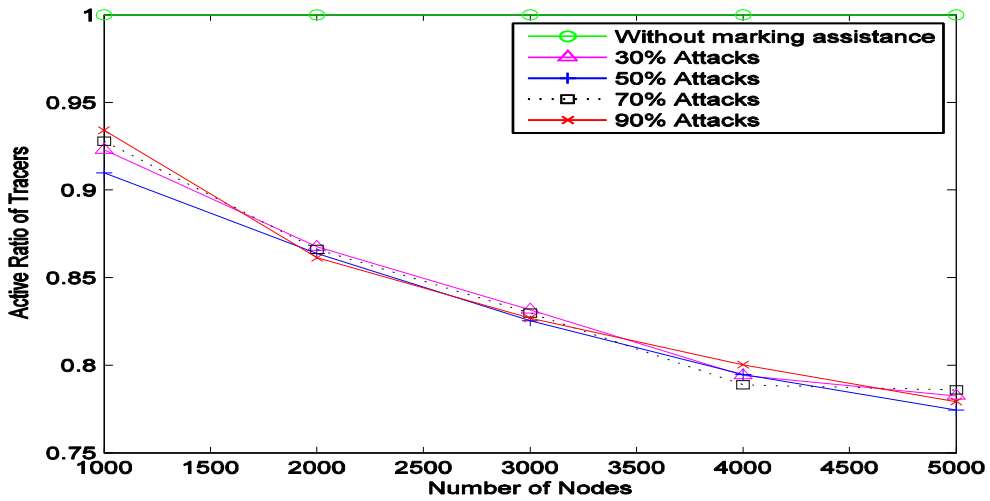


圖 3.8：50%T, 20%F, 30%M 中，各種攻擊比例下追蹤器的啟動率

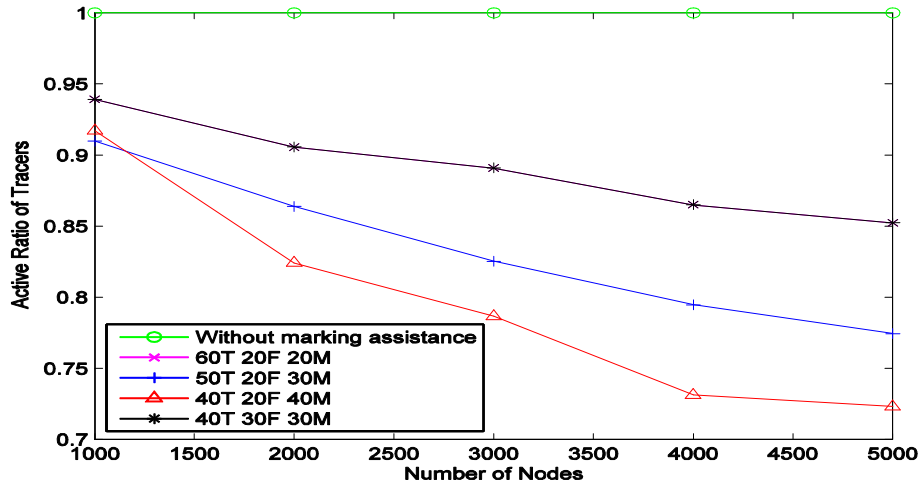


圖 3.9：在 50%攻擊量中，不同 marking-enabled tracers 比賽，追蹤器啟動率的比較

圖 3.9，為網路節點 5000 點，增加 marking-enabled tracers 的比例，觀察對追蹤器的啟動率的影響，由圖可知 marking-enabled tracers 愈多的情況， 降低追蹤器的啟動率的效果愈好。

由上述模擬實驗的觀察，我們發現以下兩點：

- 最佳導向轉送方式：利用封包經過追蹤器的歷史紀錄來決定是否要轉送下一個經過追蹤器的封包(dynamic tunneling)，其成效與最佳化的情況非常相近。

- 最佳搭配方式：運用 Marking-enabled tracers 可以關閉不必要的 tunneling-enabled tracers 可以節省超過 20%的追蹤器啟用量，更重要的是可以追蹤攻擊的落點，有效的利用鄰近的 Filtering-enabled tracers 及時過濾攻擊封包。

四、計畫自評

本計劃中，我們研究如何有效的利用 tunneling-enabled tracers 適時的將封包導向最佳的 marking-enabled tracers 或 filtering-enabled tracers 以進行來源追蹤及即時過濾異常封包，藉由模擬結果，在 50%的路由器升級為不同比例的追蹤器，各種追蹤器的位置為隨意布建的環境下，我們發現 marking assistance 的方法，除可追蹤攻擊來源的落點之外，亦可節省超過 20%的追蹤器啟用，能有效的將攻擊封包導向 filtering-enabled tracers，效能接近最佳解。本計畫相關的研究成果，可追蹤離攻擊來源最接近追蹤器的論文發表在 IEEE Aina 2008 的會議[23]，可共同防禦 DDoS 的網路安全平台發表於 IEEE NISS 2009 的會議[24]，在此平台中我們利用個人電腦，實作可收集區域網路內節點的傳輸資料並過濾異常封包的虛擬閘道器(virtual gateway)，將來可以移植到交換器上或路由器做為 filtering-enabled 的追蹤器，而上述異質性的追蹤器的相關結果亦整理完成，近期將投稿於國際會議。本計畫原本預計兩年的時間，但只獲一年的補助，計畫雖已結束，但後續有關異質性追蹤器之布建，對攻擊來源追蹤的影響與過濾異常封包的效能的影響，更值得我們深入的探討，預計繼續提後續的計畫。而參與本計畫之研究人員，藉由規劃目標、執行過程、結果分析、延伸應用，培養出網路安全與具備實作能力的科技人才，以落實前瞻產業技術建立及人才培育的目標。

參考文獻

[1] J. Mirkovic and P. Reiher, ”A Taxonomy of DDoS Attack and DDoS Defense Mechanisms,” *ACM SIGCOMM Computer Communications Review(CCR)*, vol. 34, no. 2, April 2004, pp.39-54.

[2] L. Garber, ”Denial-of-service rip the internet,”*IEEE Computer, vol. 33 no. 4*, pp. 12-17, April 2000.

[3] R. K. Chang,“Defending against flooding-based. Distributed denial-of- service attacks: a tutorial,” *IEEE. Communications Magazine*, Volume: 40 Issue: 10, pp. 42-51, Oct. 2002.

[4] A. Belenky and N. Ansari, ”IP Traceback with Deterministic Packet Marking,”IEEE Communications Letters, vol. 7, Issue 4, April 2003, pp. 162 - 164.

[5] S. Savage, D. Wetherall, A. Karlin and T. Anderson, ”Network Support for IP Traceback,” *IEEE/ACM Trans. Net.*, vol. 9, no. 3, June 2001, pp.226-37.

[6] H. Sozaki, S. Ata, I. Oka, and C. Fujiwara, ”Performance Improvement on Probabilistic Packet Marking by using History Caching,” 6th Asia-Pacific Symposium on Information and Telecommunication Technologies, APSITT 2005 Proceedings, 09-10 Nov. pp.381 - 386.

[7] L. Wu, H. X. Duan, J. P. Wu, and Xing Li, ”Improved marking model ERPPM tracing back to DDoS attacker,” Third International Conference on Information Technology and Applications, ICITA, vol. 2, July 2005, pp.759-762.

[8] U. K. Tupakula and V. Varadharajan, ”A practical method to counteract denial of service attacks,” in Proc. Australasian Computer Science Conference (ACSC2003), Adeliaide,

australia. Conference in Research and Practice in Information Technology, vol. 16.
[9] A. Yaar, A. Perrig, and D. Song, ”FIT: Fast Internet Traceback,” *Proc. INFOCOM,* 2005, pp.1395-1406.
[10] M. Muthuprasanna, G. Manimaran, M. Alicherry, and V. Kumar, ” Coloring the Internet: IP traceback,” 12th International Conference on Parallel and Distributed Systems, ICPADS 2006, Vol.1, July 2006.
[11] S. M. Bellovin, ”ICMP Traceback Messages,” IETF draft, 2000; htttp://www.research.att.com/smb/papers/draft-bellovinitrace-00.txt.
[12] Henry C. J. Lee, Vrizlynn L. L. Thing, Yi Xu, Miao Ma, ICMP Traceback with Cumulative Path, an Efficient Solution for IP Traceback, International Conference on Information and Communications Security, Oct. 2003, (Springer Lecture Notes in Computer Science, Vol. 2836, pp. 124-135, Sept. 2003).
[13] V.L.L. Thing, H.C.J. Lee, M. Sloman and J. Zhou , “Enhanced ICMP traceback with cumulative path,”*2005 IEEE 61st Vehicular Technology Conference, Volume 4*, pp. 2415 - 2419 , May-1 June 2005.
[14] J. Lee and G. d. Veciana, ”Scalable Multicast Based Filtering and Tracing Framework for Defeating Distributed DoS Attacks,” *Internation Journal of Networking Management* 2005, pp.43-60.
[15] A. C. Soneren et al., ”Single-packet IP Traceback,” IEEE/ACM Transactions on Networking, vol. 10, December 2002, pp.721-34.
[16] C. Gong, T. Le, T. Korkmaz, and K. Sarac ”Single packet IP traceback in AS-level partial deployment scenario,” IEEE Global Telecommunications Conference, vol. 3, 28 Nov.-2 Dec, 2005.
[17] W. Timothy Strayer, C. E. Jones, B. I. Schwartz, J. Mikkelson, and C. Livadas ”Architecture for multi-stage network attack traceback,” The IEEE Conference on Local Computer Networks, Nov. 2005.
[18] Y. N. Jing, P. Tu, X. P. Wang, and G. D. Zhang, ” Distributed-log-based scheme for IP traceback,” The Fifth International Conference on Computer and Information Technology, Sept. 2005, pp. 711-715.
[19] R. Stone, “CenterTrack: An IP overlay network for tracking DoS floods,” in *Proc.2000 USENIX Security Syrup*., July 2000, pp. 199-212.
[20] A. Greenhalgh, M. Handley, and F. Huici, “Using Routing and Tunneling to Combat DoS Attacks,” Steps to Reducing Unwanted Traffic on the Internet Workshop, SRUTI;05.
[21] F. Huici and M. Handley, “An Edge-to-Edge Filtering Architecture Against DoS,” ACM SIGCOMM Communication Review, vol. 37, no. 2, pp. 39-50, April 2007.
[22] http://www.caida.org/tools/measurement/skitter/
[23] Chun-Hsin Wang and Yen-Chih Chiang, “Multi-Layer Traceback under the Hierarchical Tracers Deployment,” *IEEE AINA workshop* 2008.
[24] Chun-Hsin Wang and Chun-Wei Huang, “A Collaborative Network Security Platform in P2P Networks, ” *IEEE NISS workshop NetCom*, pp.1251-1256, 2009.

2009 International Conference on New Trends in Information and Service Science

A Collaborative Network Security Platform in P2P Networks

Chun-Hsin Wang and Chun-Wei Huang
Department of Computer Science and Information Engineering
Chung Hua University, Hsinchu, Taiwan 30012, R.O.C.
E-mail: chwang@chu.edu.tw, kkjj_tw@msn.com

***Abstract*— Network security problems emerge in an endless stream and cause the inestimable damage. To solve network security problems efficiently, it is not enough to make good protection at nodes or protect networks from outside attacks. Many network security problems should be solved efficiently in collaborative approaches which can integrate various resources over internet to defense network security. In this paper, we have designed and implemented a collaborative network security platform based on P2P system. The nodes participated in the P2P system can publish their designed defensible services against network security problems. Based on the published services, collaborative network applications can be developed easily to solve the network security problems on demand. An experiment against TCP SYN flooding attack is demonstrated by the designed defensible services including packets sniffing, forwarding, filtering, and logging services, which can trace the attack origins and filter malicious traffic efficiently.**

***Keyworkds*— P2P, Collaborative Network Security**

I. INTRODUCTION

Network security problems emerge in an endless stream and cause inestimable damage. To solve Network security problems, it is not enough to make good protection at nodes or just set *firewall* or *IDS* (Intrusion Detection System) to protect edge networks from outside attacks. Many network security problems should be solved efficiently in collaborative approaches which can integrate various defensible resources over internet. For example, to find attack origins, the technology of IP traceback [1], [2] needs to cooperate among the enhanced routers or nodes which can provide tracing service such as logging the passing packets or marking their address into them. It is a trend and challenge how to integrate possible resources over networks and apply them to solve the various of network security problems.

Many serious network security problems are caused by Distributed Denial of Service (DDoS) ([3], [4]) attacks and virus worms-spreading. DDoS attacks always paralyze the services which network nodes can provide and occupy the network bandwidth by flooding volumes of traffic to the victims. One attack node may contribute low-rate malicious traffic but attack traffic from widely distributed attack nodes is aggregated toward to the victim. Typical single-point defense system near attack origin, setting IDS at the entrance of individual edge networks, can not recognize low-rate attack traffic destined to victim. For the similar reason, single-point worms monitoring can not detect the signature of worms fast and prevent worms spreading effectively. Therefore, these facts show how important that the cooperation among defense systems over internet is.

To defense DDoS attacks and worm containment from internet, some collaborative approaches [5-8] are proposed in literatures. They focus on integration of IDSs or worm monitoring over internet to coordinate distributed detection and defense activity. In [5], a DHT-based chord [9] P2P system is used to integrate IDS (snort [10] or Bro [11]) for fast worm containment and prevent flooding attacks. In [6], an "Worminator" platform is implemented, in which the alter information is detected by the Antura network detection system [12] and encoded in Bloom Filters [13] for sharing in multiple domains. The global alter information can speed up to identify where the malicious traffic is from. In [7], the distributed detection module of DDoS attacks is deployed at transit routers. In each *ISP*, a server named as *CAT* (change aggregation trees) takes responsibility to aggregate the flooding alters from routers. A tree-based CAT tree is constructed for fast detecting DDoS flooding attacks. In [8], a DHT-based Pastry protocol [14] is adopted to cooperate multiple IDS against DDoS attacks.

From the discuss as above, we note that P2P overlay networks can promise to exploit the distributed defensible resources against network security problems. But the proposed collaborative defense systems in literatures only focused on internet security problems and the defensible resources are limited to complex IDS or worms-monitoring systems. We know the attack origins are hidden in some intranet finally. The practical and fundamental problem is how to solve the network security problems in intranet. It can be expected that network security problems could be solved efficiently when more possible resources are involved. The defensible resources integrated by P2P system should contain all possible resources including routers/switches with traceable function, firewalls, IDS, and even the personal computers (PCs).

In this paper, we have designed and implemented a collaborative network security platform based on P2P system to solve intranet network security problems especially. Ubiquitous PCs are the main member of the proposed platform. The joining nodes in the p2p system can design defensible services and publish them for sharing under secure control. Based on the published services, collaborative network applications can be developed easily to solve the network security problems on demand. To let PC nodes with defense ability, we have designed defensible services including packets sniffing, forwarding, monitoring, and filtering services. An experiment against TCP SYN flooding attack is demonstrated by the

*This research was supported by the National Science Council, Taiwan, R.O.C., under grant NSC 97-2221-E-216-035.

978-0-7695-3687-3/09 $25.00 © 2009 IEEE
DOI 10.1109/NISS.2009.222

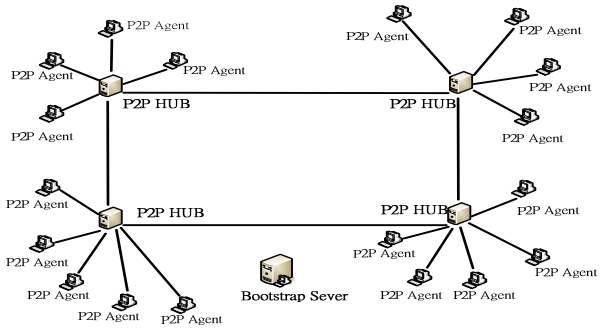


Fig. 1. The model of network security platform

designed services, which can trace attack origins and filter abnormal traffic efficiently. To our best knowledge, we are the first paper to study how to integrate resources of ubiquitous PCs against network security problems in P2P networks.

The rest of the paper is organized as follows. The proposed collaborative network security platform is described in section II. The designed defensible services are described in III. The implementation of collaborative application and experiment are presented in section IV. Finally, some concluding remarks and future work are given in section V.

II. COLLABORATIVE NETWORK SECURITY PLATFORM

In this section, we first introduce the model of the proposed network security platform. Then we describe the function of components in it, system operation and implementation in detail. Consider the trade-off between centralized and distributed P2P systems, the proposed network security platform (Figure I) is adopted to be similar to KaZaA [15]. The members of platform are bootstrap server, P2P hub nodes, and P2P agents nodes. The bootstrap server is initialized first when system is starting. It maintains the authorities of joining nodes, the list of P2P hub nodes, and (private/public) keys for secure information exchange. The role of P2P hub nodes are as cluster leaders in KaZaA, which maintain the IP addresses of their owned P2P agent nodes and associated resources-sharing. To prohibit the misuse of resources-sharing, peers in the proposed P2P system are classified into three categories, namely Service Passive Agent (SPA), Request Active Agent (RAA), and Publish Active Agent (PAA). SPA can provide services for RAA which can issues the request to solve network security problems. Besides the function of SPA and RAA, PAA can publish new defensible services to enhance the capability against network security problems for the proposed platform.

A. Secure Message Exchange

To protect message exchange among the peers in the proposed platform, typical technologies of symmetric and asymmetric keys protection are applied. The messages of requesting to collaborate on network security from RAA and publishing new defensible services from PAA are protected with high precedence by asymmetric keys. The protection of the others message exchange is adopted by symmetric keys. The peers joining the platform are to be as SPAs by default. SPAs can request bootstrap server to be as RAAs or PAAs. All kinds of agents have to register in bootstrap server and then get the corresponding keys from it as follows.

- SPA get a private key to decrypt request messages issued by RAA, a private key to decrypt the published messages issued by PAA, and a pair of symmetric keys for others messages exchange.
- In addition to the keys SPA have, RAA has a private key to encrypt request messages in asymmetric way before they are transmitted.
- PAA has the function of SPA and RAA. Besides those, it can publish new defensible services for sharing in the platform. PAA will get an extra private key to encrypt the message of publishing new service.

Every key is with a default value of time-to-live (TTL). The default value is 24 hours. Public keys used to decrypt the receiving messages are managed by bootstrap server. P2P hub nodes take responsibility to get new public keys from bootstrap server and forward them to their owned agents before TTL of public keys is ten minutes left. In this way, public keys in agents can be renewed when TTL of public keys is time out. Due to the private keys are assigned by the role of agents, agents need to get their new private keys from bootstrap server under password authentication. In similar renewal of public keys, agents (SPA, RAA, and PAA) will get their new private keys before TTL of private keys is five minutes left.

B. System operation

The bootstrap server is initialized first when the collaborative network security platform is starting. One peer can request it to acquire an account to join the platform and to be SPA by default. Simultaneously SPA can get an identification (ID) number and the list of hub nodes existing in platform from bootstrap server. If none of hub nodes exists, the SPA will become the first one hub node. Otherwise, SPA measures hop counts between it and hub nodes in the list. Then SPA selects exactly one hub node closest to it and to be its member. But when the selected hub node is too far from the SPA or the members of it is too many (ex. over than 50), the SPA will become a new hub node under load balance consideration. When hub node leaves the platform, members of it will rejoin the platform after random time. SPA can request bootstrap server to be RAA or PAA and get related keys.

C. Implementation of Agents and Dynamic link library

In this paper, the bootstrap server, P2P Hub nodes, and three kinds of agents (SPA, RAA, PAA) are implemented by *BIGSPEED* peer-to-peer SDK [17] in Microsoft XP operating system. The programs are coded in Borland Delphi 7 and the functions of Winpcap [18] library are applied to capture packets passing through network interface. For convenient to make use of the proposed platform, a shell-like command-line is designed to send messages for requesting services or publishing new defensible services. In addition, a dynamic link library, "P2Pdefense.dll", is provided such that users

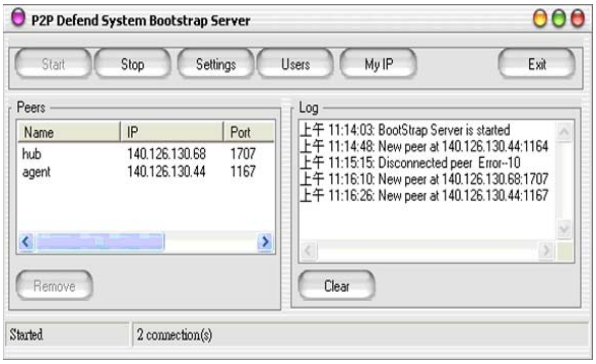


Fig. 2. Bootstrap Server

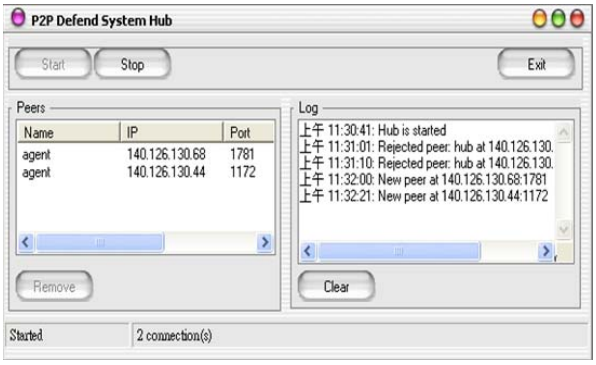


Fig. 3. P2P Hub node

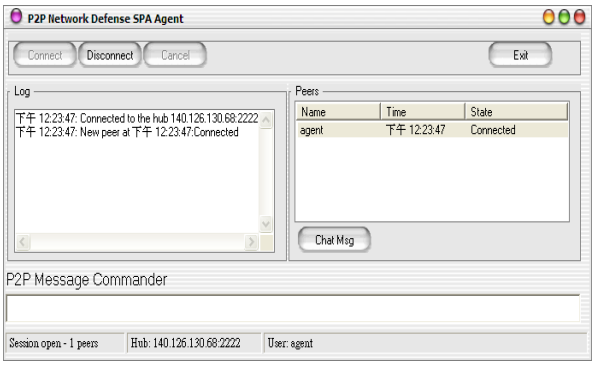


Fig. 4. SPA agent

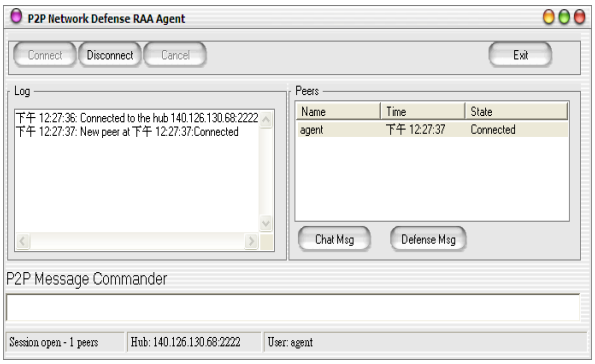


Fig. 5. RAA agent

can develop collaborative network security applications easily by calling the developed functions in it. Users joining the proposed platform can share their designed defensible services or package them into ”P2Pdefense.dll”. It can be expected that network security problems can be solve efficiently when more users (peers) participate and share their defensible services in the platform. We briefly show our implementation as follows.

1) Bootstrap Server: The graphic user interface (GUI) of bootstrap server is implemented as Fig. 2. The joining agents and existing hub nodes are maintained by bootstrap server. Their names, IP address, and port number are displayed on the left part of GUI. The log of connections from peers is showed on right part of GUI. The login accounts of Peers can be set by the GUI of bootstrap server.

2) P2P Hub Nodes: Agents connecting to the hub node are displayed on right part of GUI in P2P hub node as Fig. 3. The log of connections from agents is showed on the right part of GUI.

3) Agents: The GUI of SPA, RAA, and PAA are showed in Fig. 4, Fig.5, and Fig. 6 respectively. All of agents have the common way to send messages by a shell-like command-line named as ”P2P Message Commander”. Compound messages are allowed in command-line. A semicolons is used to separate two different sending messages. For different kinds of agents, the commands of sending messages by command-line are listed in table 7. SPAs can only chat with others agents connecting to the same P2P hub node. The chatting message can be composed of the reserved word ”MSG”, < Name of Agent>, which agent wants to talk, and < Text>, the transmission message. RAAs can request other agents to provide services but they can not publish new services. The message of requesting service is composed of the reserved word ”RUN”, <hop count>, <service>, and <parameters list>. When hub node receives the request message, it will decrease the value of <hop count> by one and forwarding it to their neighboring hub nodes if that value is not zero. The request message will be broadcast when the value of <hop count> is set by ”-1”. The name of requesting service and its related parameters are encoded in <service> and <parameters list> respectively. When agents receive the request message, they will automatically search and get the request services in the proposed platform if they still don’t have the services. To collect the log of services execution from agents, we have designed the command ”BACK Log”, which can transmit log file back to the RAA (or PAA). The log file is named as the IP address of requesting agent (RAA or PAA) plus ID of agent executing the request service. Therefore, the requesting agent can distinguish where the log files are from. In addition, RAA can control the duration of executing service in agents by the command ”Time limit <number of minutes>”, which can limit the service execution time for an assigned number of minimums. Besides the commands of RAA, PAA can use the command ”Publish <Name of Service>” to publish new service. For example, a PAA with IP address ”140.126.130.5” wants to publish a new service named as ”sniff.exe” and ask other agents in platform to run this service for three minutes. The log file of sniff.exe is need to transmit back to the PAA. The messages of command-line in the PAA are to be ”Publish

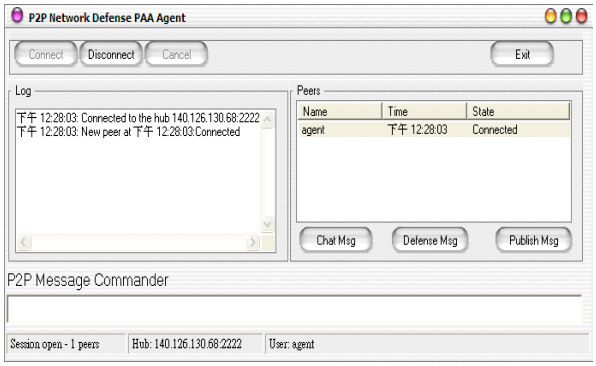


Fig. 6. PAA agent

Agents	Commands	Parameters	Functions Description
SPA	MSG	< Name of Agent> <Text>	SPA can chat with agents belonging to the same hub node.
RAA/ PAA	RUN	<Hop Count> <Service > <parameter list>	RAA/PAA can request agents within a limit distance to run the assigned service.
RAA/ PAA	Time limit	<Number of Minutes>	RAA/PAA can limit the duration of executing service.
RAA/ PPA	Back Log	None	The log file of executing service will be transmitted back to the RAA/PAA
PAA	Publish	<Name of Service>	PAA can publish new service to the platform

Fig. 7. The commands in command-line

Functions	Descriptions
Int Startnetwork(char *loginame, char *password)	Login the platform
Int SendCommand(char * command)	Sending commands
Int Chat(char *AgentName, char *chatmessage)	Chatting with agents
Int SendFile(char *AgentName, char *filename)	Sending files to agent
Int Download(char *AgentName, char *filename)	Download files from agents
Int Search(char *AgentName, char * keyword,char *fileLists)	Searching files
Int Stopnetwork()	Logout platform

Fig. 8. The functions of P2Pdenfense.DLL

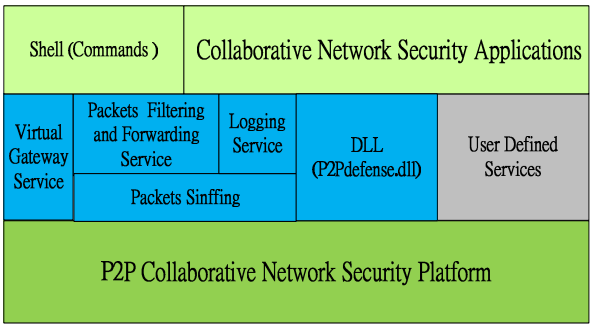


Fig. 9. Architecture of P2P Network Security Platform

sniff.exe;RUN -1 sniff.exe 140.126.5.100;Time limit 3;Back Log", where the "140.126.5.100" is the parameter of service "sniff.exe".

The command-line in agents can provide an convenient way to use the collaborative network security platform to defense network security problems. To make the development of collaborative network applications easily, we package basic functions of the proposed platform into a dynamic library named as "P2Pdefense.dll" in Fig. 8.

III. Implementation of Defensible Services

The basic idea of this paper is that the joining nodes in the proposed platform can design defensible services and publish them for sharing under secure control. The published services can also be packaged into a dynamic library for sharing. Based on the published services and dynamic library, collaborative network applications can be developed easily to solve the network security problems on demand. The software architecture of developing collaborative network applications is showed in Fig. 9.

The main members of the collaborative network security platform are ubiquitous PCs. A general PC can be given with defense ability by installing some services (software) such as PC-based IDS, anti-virous system, and so on. Pcs joining the proposed platform could be the defense nodes. We hope a defense node can secure not only itself but also its neighboring nodes to defense network security collaboratively. To realize the capability, the defense node must have the ability to investigate packets passing through network interfaces of its neighboring nodes. By a network switch with mirror functions, packets can be copied and forwarded to an assigned mirror port which the defense node connects to. But its not practical because not all of network switches have the mirror function and mirror port should be configured manually or by network management protocols. Without the assistance of network switches, how to forward packets needed to be investigated to the defense node is a challenge.

By cheating of APP and RARA [19] protocols, we have successfully designed the virtual gateway service which can let the defense node with capability to investigate packets passing through its neighboring nodes. The defense node with virtual gateway service can reply ARP packets to cheat their neighboring nodes of the gateway is the defense node. Therefore, all packets from neighboring will be forwarded to the defense node. To let the packets destined to neighboring nodes forward to the defense node, we can cheat the real gateway of the MAC addresses of neighboring nodes are the defense node by ARP packets. All packets from or to the neighboring nodes will be forward to the defense node with virtual gateway service as the display of Fig. 10

After the packets are forwarded to the defense node by virtual service, the defense node still needs to capture the receiving packets and then investigate which packets are malicious. If the packets are normal traffic, they should be forwarded to their original destination nodes. Otherwise, malicious packets should be filtered and the related information such as source IP address of packets and their types should be logged. Thus, we implement the function of packets sniffing by winpcap library. Based on the packets sniffing, useful services of packet forwarding, filtering, and logging are developed for

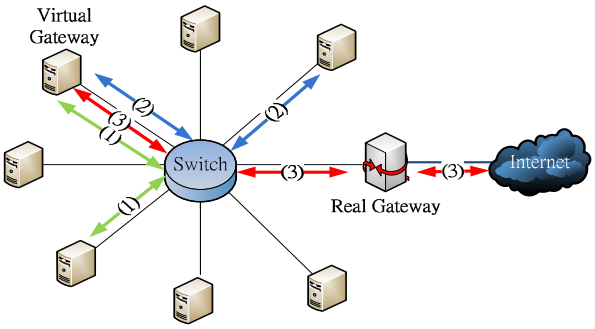


Fig. 10. Packets forwarding by virtual gateway service

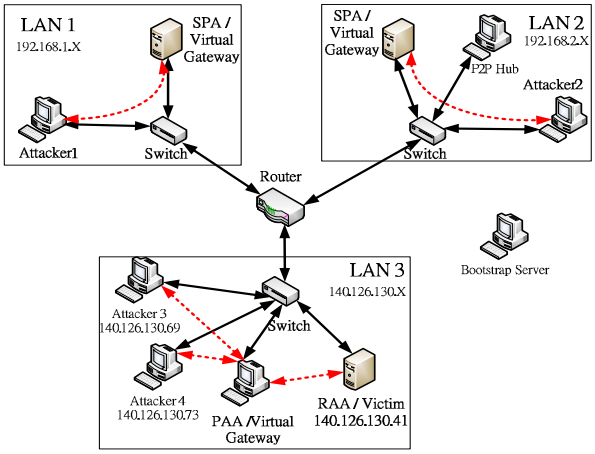


Fig. 11. Experiment Environment

the proposed platform.

IV. An Experiment against TCP SYN Flooding Attack

To demonstrate the proposed collaborative network security platform and our implementation, we set the experiment of three LANs shown in Fig. 11. P2P hub node is located in LAN1. There exists a SPA in LAN1 and LAN2 respectively. A PAA and the victim which is also RAA are located in LAN3. These four agents belong the same P2P hub node.

To defense the TCP SYN attacks, we have designed the packet filtering service named as "packet_filtering" which can drop the SYN segments. The PAA can publish the designed services by command-line, i.e., "Publish virtual_gateway.exe;Publish packet_filtering.exe". The SPAs and RAA can get these two services by searching in P2P hub node and download them from PAA.

There are four attackers distributed in three LANs. Each attacker sends the TCP SYN attack to the victim for requesting TCP connections thirty times per ten seconds. To detect the attack of TCP SYN flooding, a simple detecting program of counting number of TCP SYN segments is developed. The attack of TCP SYN flooding is identified when the number of receiving TCP SYN segments is over 100. In our experiment scenario, four attackers are starting to attack the victim by sending TCP SYN segments. After 30 seconds, the victim starts the detecting program to detect the TCP SYN flooding

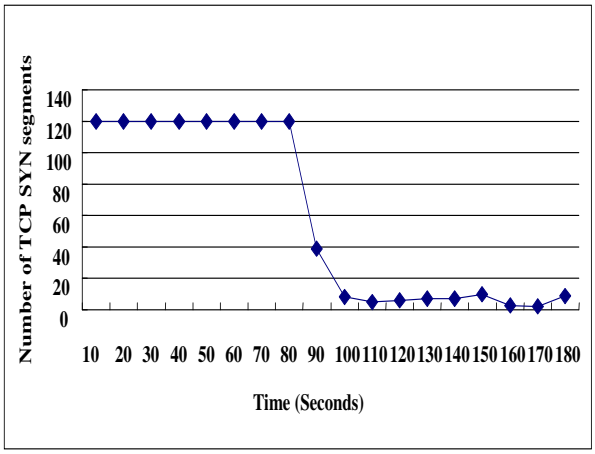


Fig. 12. The number of TCP SYN segment is received in the victim

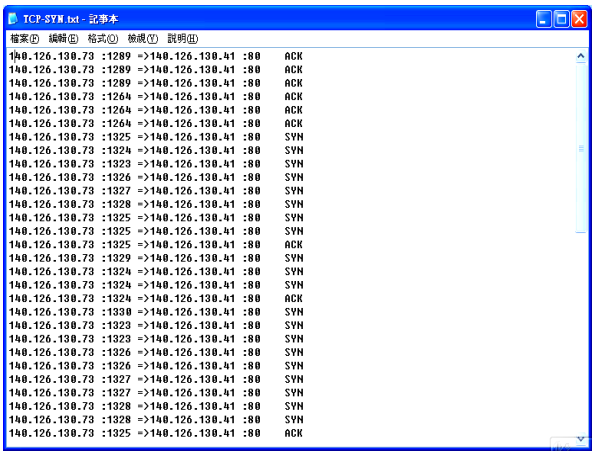


Fig. 13. The log file of from PAA in LAN3

attack and requests other agents to defense the attack collaboratively. Therefore, the victim sends the request message "RUN -1 packet_filtering 140.126.130.41 80 TCP_SYN;Back Log;Time Limit 3" in command-line. The first parameter "140.126.130.41" in the parameters of packet_filtering is the IP address of victim. The second parameter "80" means the port number and the last parameter is used to indicate which type of traffic is needed to be filtered. That is the victim requests agents to monitor the packets destined to the victim on port 80 and filter the TCP SYN segments.

When agents in different LANs receiving the request message by P2P hub node, they will try to be the virtual gateway by running "virtual_gateway.exe". If there exist one virtual gateway, agents will give up to be a virtual gateway. In the current version, multiple virtual gateway for load balance within a switch is not supported. In our experiment, each LAN has one the virtual gateway running as in Fig. 11.

Figure 12 shows the number of TCP SYN segments is received in the victim. The victim RAA executes the detecting program after attacks are starting for 30 seconds and then requests agents to defense the attack collaboratively. From this figure, we can observe that the number of TCP SYN segments is decreasing at 90 seconds because the malicious TCP SYN

segments are filtered by the virtual gateway distributed different LANs. Besides that, the attackers can be traced by the log file transmitted back by virtual gateways. Fig. 13 shows that the attacker 4 in LAN3 with IP address "140.126.130.73" is sending TCP SYN segments to the victim with IP address "140.126.130.41".

V. CONCLUSIONS AND FUTURE WORK

To solve network security problems efficiently, it is not enough to make good protection at nodes or protect networks from outside attacks. Many network security problems should be solved efficiently in collaborative approaches which can integrate various resources over internet to defense network security. We have designed and implemented a collaborative network security platform based on KaZaA-like P2P system to solve intranet network security problems especially. Ubiquitous PCs are the main member of the proposed platform. To let PC nodes with defense ability, we have designed defensible services including the virtual gateway service, packets sniffing, forwarding, filtering, and logging services. Without the assistance of network switches, a general PC joining the platform can be the virtual gateway to investigate the packets from other nodes and then filters the malicious packets. An experiment against TCP SYN flooding attack is demonstrated. The victim can be protected from the TCP SYN flooding attack efficiently. It can be expected that network security problems can be solved efficiently when more ubiquitous PCs join the proposed platform and share their designed defensible services and applications by the power of resources sharing in P2P system.

In the future, the GUI of agents will be improved. The load balance of virtual gateways will be considered. Multiple virtual gateways will be supported such that one virtual gateway can only take responsibility to investigate packets from part of nodes within a switch. Based on assistance of the distributed virtual gateways over networks, an interesting problem how to trace the attack origins as soon as possible is worth exploring. Besides that, we can also consider let network switches join the proposed platform. The virtual gateway service may be implemented in network switches. It will be convenient to trace the attack origins in intranet even only part of switches with virtual gateway service.

REFERENCES

[1] S. Savage, D. Wetherall, A. Karlin and T. Anderson, "Network Support for IP Traceback," *IEEE/ACM Trans. Net.*, vol. 9, no. 3, June 2001, pp.226-37.

[2] A. Belenky and N. Ansari, "On IP Traceback," *IEEE Communicatin Magazine*, July 2003, pp.142-153.

[3] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," *ACM SIGCOMM Computer Communications Review(CCR)*, vol. 34, no. 2, April 2004, pp.39-54.

[4] Rocky K. C. Chang, "Defending against Flooding-Based Distributed Denial-od-Service Attacks: A Tutorial," *IEEE Communicatin Magazine*, Oct. 2002, pp.42-51.

[5] Min Cai, Kai Hwang, Yu-Kwong Kwok, Shanshan Song, and Yu Chen, "Collaborative Internet Worm Containment,", *IEEE Security and Privacy*, May/June, 2005, pp. 25.-33.

[6] M. E. Locasto, J. J. Parekh, A. D. Keromytis, and S. J. Stolfo, "Toward Collaborative Security and P2P Intrusion Detection," IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, 2005, pp. 333-339.

[7] Yu Chen, Kai, Hwang, and Wei-Shinn Ku, "Collaborative Detection of DDoS Attacks over Multiple Network Domains," IEEE Transactions on Parallel and Distributed Systems, Vol. 18, No. 12, December 2007, pp. 1649-1661.

[8] Saad, Radwane; Nait-Abdesselam, Farid; Serhrouchni, Ahmed, "A collaborative peer-to-peer architecture to defend against DDoS attacks" Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on 14-17 Oct. 2008 Page(s):427 - 434.

[9] I. Stoica, R. Morris, D. Nowell, D. Karger, M. Kaashoek, F. Dabek and H.Balakrkshnan, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," *IEEE/ACM Transactions on Networking*, Vol.11, No. 1, February 2003.

[10] http://www.snort.org

[11] http://bro-ids.org

[12] S. Roberison. E. Siegel, M. Miller, and S. Stolfo, "Surveillance Detction in High Bandwidth Environments," in 2003 *DARPA DISCEX III Conference*, April 2003.

[13] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communication of ACM*, Vol. 13, pp. 422-426, July 1970.

[14] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," *Lecture Notes in Computer Science*, 2238:329, 2001.

[15] KaZaA http://www.kazaa.com

[16] CacheLogic Research: The True Picture of P2P File Sharing, http://www.cachelogic.com/research/

[17] BIGSPEED Computing Inc., http://www.bigspeed.net/

[18] The Windows Packet Capture Library, http://www.winpcap.org

[19] Network Working Group ARP RFC 826,1982.