東海大學
資訊工程與科學研究所

碩士論文

UDIDT 下之多階段入侵偵測系統

Multi-Phase Intrusion Detection System under UDIDT

指導教授：呂芳懌博士
研 究 生：鄭　真　真

中華民國九十二年八月

誌謝

大學畢業後，有機會能夠再回到學校進修，重溫學生生活，令我更加珍惜！特別感謝恩師呂芳懌教授的指導，專案製作、研究網路安全與探討網路入侵攻擊模式，培養學生思考邏輯與組織能力，提昇分析報告與寫作論文的能力，使學生再面對問題時，能有不同的思考與見解。老師對教學與研究的執著嚴謹與認真熱忱的態度，著實讓學生由衷敬佩，這也都是學生所要學習的。

論文審查及口試期間，承蒙林宜隆教授、林熙禎教授、連志誠教授及張文貴教授等提供許多寶貴的意見，使本論文的內容更加嚴謹而充實，在此致上最誠摯的謝意。

感謝與我共同研究 UDIDT 系統的嘉鴻，讓我能對建構系統有整體的構思，使我獲益匪淺。其次，實驗室所有陪我成長的夥伴們，政瑋、義樺、俊維、子逸、清健、品璁、仁傑、耀中及國煒等的關心與交換研究心得；實作時提供技術支援的好朋友們，信聰、揚凱、坤義、惟翔、宜親與秀佾等；以及為我鼓勵打氣的姊妹們，琪琳、沛妤、莉莉和淑真等，讓我留下許許多多美好的回憶，我都銘感於心。

最後，要感謝的是默默為我加油的家人，尤其是我的父母，給予我最大的空間與支持，使我能夠順利完成此論文及研究所學業，謹以此論文獻給我最摯愛的家人。

摘要

近幾年來，資訊安全的觀念雖然逐漸受到重視，然而，許多系統的管理者對於網路安全的防範並未落實，再加上駭客工具越來越容易取得，攻擊者可以輕易的入侵主機或是透過某些攻擊方式癱瘓主機或網路設備，種種方式都造成公司單位或個人難以估計的損失。一些被動式的防禦與偵測工具，例如，Firewall、IDS（Intrusion Detection System）等，對於攻擊者的入侵行為只有警示作用，沒有嚇阻效果。事實上，惟有找到攻擊者，訴諸法律，才能有效的嚇阻攻擊事件的發生。

本文提出一個入侵偵測與追蹤機制，稱之為"區域聯防入侵偵測與追蹤系統（UDIDT, Union Defense of Intrusion Detection and Traceback System）"。UDIDT 係在其所在區域內以多階段式入侵偵測系統（Multi-phase IDS）偵測入侵攻擊，透過紀錄在該區域內封包之 Digests，及與其他區域的相互合作，而以「區域聯防」的方式追蹤大部分類型的攻擊來源。

本研究中首先蒐集歸納多種攻擊模式封包特性、入侵偵測系統及入侵追蹤系統，分析其優缺點，再以網路區域聯合防禦的觀念設計本系統，其中設計 MIDS 為到即時的入侵偵測系統，並提供 UDIDT 追蹤入侵者時所需之資料。最後以實驗來驗證 MIDS 的偵測效率。

關鍵字：網路安全、入侵偵測、入侵追蹤

ABSTRACT

In recent years, people have paid more and more attention on information security. However, illegal intrusions seriously prevail over the network due to widespread hacking tools and lots of insecure hosts. An intrusion of the system causes great financial(s) for a company or people. Tradition security tools such as Firewall, Intrusion Detection System only focus on warning, prevention and detection. In order to prevent a system from an illegal attack, finding and punishing malevolent hackers should be an effective way.

In this paper, we proposed an intrusion detection and traceback system, called “Union Defense of Intrusion Detection and Traceback System (UDIDT)”. This system actively detects intrusions by a multi-stage detecting IDS named Multi-phase IDS. It keeps hash codes for packets flowing through a network section with which the traceback system can trace hackers of an attack with a union defense approach.

In this research, we first sum up the characters of attacking packets, then the advantage and the lacks of current intrusion detection systems and trace back systems. We use the concept of union defense to design UDIDT. MIDS is a real-time intrusion detection system. It also supports the pre-recorded data for UDIDT to trace back the source of an intrusion. Finally, Experiment is involved to validate the efficiency and the availability of the Detecting Queue in MIDS.

Keyword: Network security, Intrusion detection, Intrusion traceback

目錄

圖目錄

表目錄

第1章 緒論

本章將介紹研究動機、研究目標、研究方法，並對本論文之整體架構做簡單的描述。

1.1. 研究動機

隨著電子商業的日益蓬勃，網路通訊改變了個人的日常生活及企業、組織的商業行為，人們愈來愈依賴網路的傳輸速度及其所提供的資源。今日的科技亦提供不少技術支援，如寬頻技術、更高階的伺服器等；但網路頻寬及電腦處理速度卻永遠也追不上實際的需求。這樣一個有限資源的網路世界便成為攻擊者發動 DoS（Denial of Service）、DDoS（Distributed Denial of Service）等攻擊的可利用之處[18]。

駭客工具越來越容易取得，駭客入侵行為也越來越頻繁，攻擊手法更是不斷翻新，目前有許多網站紛紛提供駭客攻擊程式與教學題材供網友參考；甚至有人設置線上病毒產生器（virus generator）的網站[3]，只要登錄此網頁，點選所要的功能，即可獲得所想要的攻擊程式；攻擊者不需具備足夠的系統或網路等專業知識，即能使用自動化工具發動攻擊，甚至可以聯合多部攻擊主機發動一場大規模的攻擊行動。

駭客的攻擊行為往往造成個人、企業及社會難以估計的損失，2000 年 2 月的 Yahoo、amazon.com、ebay.com、CNN.com、E-trade 等知名網站遭到攻擊的事件，造成服務中斷數小時，由 Yankee Group 評估顯示各商業網站之營收、市場資本損失及安全維護費用超過十二億美元。2001 年 5 月的中美網路駭客大戰，也使我們認識到資訊戰的冰山一角[18]。除此之外，盛行的線上交易使得愈來愈多個人穩私資料或商業機密直接暴露於網路上，利用網路的隱藏性、成本低而且跨國性的法律問題等，也使得攻擊者被發現的風險甚低。

以上的說明在在都顯示目前大部分的網路安全防護仍是相當薄弱的，根據資訊科技研究公司 Computer Economic 在 2002 年公佈的研究數據[3]顯示，因為非法攻擊而造成資料或生產量損失所產生的全球經濟影響，從 1995 年的 5 億美金逐年增加至 2000 年的 171 億，見圖 1-1。在 2001 年卻首次呈現負成長，表示網路安全已受到政府、企業或個人的重視，業界也開始注意這個市場的開發；然而，這並不代表目前所有在網路安全方面的努力已成功地嚇阻或阻止網路駭客的攻擊。防火牆是目前最基本的網路安全配備，

卻已經無法滿足企業由被動防禦到主動預防的需求；不論是軟體或硬體式的防火牆最主要的防禦措施不外乎是關閉不必要的通訊埠、設定拒絕通訊的 IP 位址或設定過濾條件等，雖然能阻斷一些來意不明或惡意的網路封包，但是有經驗的駭客仍可輕易地設計惡意程式，堂而皇之地入侵企業網站。

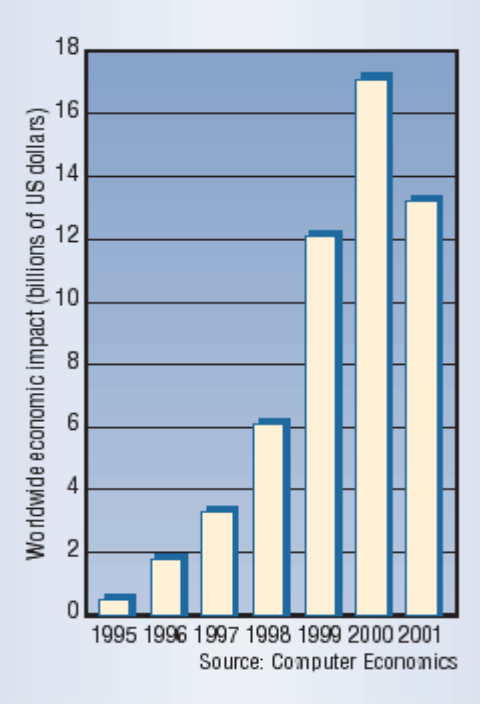

圖 1-1　非法攻擊所造成全球經濟的影響

（附圖來源：[3]）

近兩年來逐漸受重視的 IDS（Intrusion Detection System）之流，常用來與防火牆搭配，共同組成一道防止入侵的防線。然其大多也僅根據預建的攻擊行為模式資料庫比對網路連線或封包型態，產生警訊和報告；一旦查覺出可疑的攻擊行為，則對網管人員發出警訊，俾在被攻擊之初即時地隔離、解決可能造成嚴重災害的攻擊事件。但不論是防火牆或 IDS 皆屬被動式的防禦與偵測，如今駭客們仍在網路上橫行，網路管理者也僅能疲於奔命地處理防火牆及 IDS 所發出的警訊，卻無法主動追查出真正的攻擊者並將之繩之以法，更遑論有效地嚇阻惡意的入侵行為。既然這些裝置皆難以保障網路世界的安全，發展能夠主動反制與遏止的入侵追蹤系統將是必然的趨勢。

1.2. 研究目標

入侵追蹤的目標在於能夠正確、快速地針對惡意的封包，找出其真正的入侵者。雖然，有不少入侵追蹤研究陸續發表，其中有些方法是要改變現有通訊協定、設備功能，

有些則僅能追踪部份的攻擊，有一部份甚至仍在理論研究階段。本研究則以目前可取得的設備，規劃出通用、可行的入侵追蹤機制，稱之為”區域聯防入侵偵測與追蹤系統（UDIDT, Union Defense of Intrusion Detection and Traceback System）”[15]。

UDIDT 係以區域聯防的方式，透過事先紀錄在各「區域」的封包之 hash 值，追蹤大部分類型的攻擊來源。方法是在各區域入口的 Backbone Router（BR）處建置多階段入侵偵測系統（MIDS, Multi-phase Intrusion Detection System），藉由攻擊模式的比對，分析其封包規則，以階段偵測的方式，分別在 IDP 與 RS 設置 ID、AD 及 Analyzer，三階段進行不同的攻擊模式偵測，盡可能即時地發現攻擊行為，將偵測出疑似攻擊的資訊提供給區域聯防機制，並由入侵偵測與入侵追蹤兩子系統共享所擷取之資訊，以減少雙方均搜集所浪費的效能，並相互配合以達到強健的防禦效果。期能以 UDIDT 主動追蹤所偵測出來的攻擊封包之攻擊來源，迅速逮捕入侵駭客，訴諸法律，以達嚇阻的效果。

1.3. 研究範圍

入侵偵測結合入侵追蹤的機制，勢必是最有效的防禦方式；一般而言，入侵追蹤乃利用一些網路上的傳輸資訊做為逆向反查，直到找到攻擊封包的來源，但其最大的困難在於駭客會偽造封包的來源 IP 位址來隱藏自己的真實位置，所以在建立追蹤機制時，須避免被駭客偽造的 IP 位址所誤導。本論文所提出的 UDIDT 是透過「刻意製造的痕跡」來達到反向追蹤的目的。這個痕跡包括各區域所紀錄行經該區域的封包及行經的途徑。另外，各區域亦蒐集其他封包（包括由外對內及由內對外）資訊，提供入侵偵測系統，研判是否有攻擊該區域或由該區域發動的攻擊行為，俾即時地發出警示。這裡所謂的區域是指獨立的網路管理單位。之所以要劃分區域，主要的目的是配合網路所在之地理位置，一方面分散各區域設備所紀錄的資訊量，一方則以較小的管控單位有效地偵測入侵行為，再以各區域相互合作的方式共同追蹤攻擊來源，以達成小區域防守，大區域追蹤的安全原則。

事實上若偵測系統無法即時發出入侵警示，則追蹤系統將無法適時阻止入侵，勢必對區域內的主機造成傷害，屆時僅能做事後補救及追蹤，因此，入侵偵測部份應盡量偵測出各種來犯之攻擊。唯攻擊手法日新月異，且部份攻擊以正常連線或電子郵件等方式為之，受到這些封包並無特殊特性的限制，而無法有效偵測出來。這一部份的補救措施便是配合 HIDS 對於系統連線日誌中分析各連線行為，近來常討論的法醫鑑識（Forensics）[12]技術，例如，下指令／命令之習慣、打字速率、連線狀況或使用軟體習慣等，做為辨識攻擊者的特徵。

1.4. 研究方法

本研究之過程與方法如圖 1-2 所示。由於入侵行為常是利用網路協定、設備、系統、服務及程式的漏洞或疏忽，因此，研究初期必須先探討攻擊行為、入侵偵測系統及入侵追蹤系統等，以瞭解現有功能技術及限制，進而設計務實可行的 UDIDT 入侵追蹤系統架構，再研製 MIDS 多階段入侵偵測系統，期達到主動偵測與追蹤的功效。MIDS 的運作效率將以實驗分析驗證之。

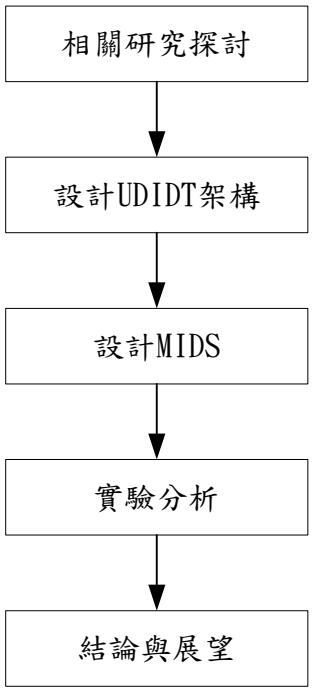

圖 1-2　研究過程與方法

1.5. 本論文架構

本論文第二章將探討各種攻擊行為，乃分類介紹各入侵偵測系統及重要的入侵追蹤相關研究，並說明其設計概念及優缺點。第三章介紹 UDIDT 架構、元件、入侵追蹤流程及其貢獻與限制。第四章描述多階段入侵偵測系統的設計規劃，包括，各元件設計，及如何從事即時偵測與事後偵測。第五章中則介紹實驗設計及分析結果，以證明本系統之可行性。UDIDT 對區域內的防禦策略為何？而此系統是否反而成為攻擊對象？又該如何預防？MIDS 的限制及對偽裝 Router 之攻擊模式等討論都將在第六章描述。第七章為本論文之結論，並說明未來的展望。

第2章 文獻探討

為能有效地從事入侵偵測，首先必須分析駭客所採用的各種攻擊模式，以解析其所用的各種網路架構、設施、作業系統或應用軟體等之缺失與漏洞，以歸納出各種攻擊手法及攻擊封包的特徵，以做為偵測防禦的依據。

以下將以偵測方式與建置方式分別探討入侵偵測系統的基本觀念。最後列舉近年來各重要的的入侵追蹤研究，介紹其所能防禦的攻擊模式與追蹤系統的架構及各別的優缺點。

2.1. 攻擊模式

一般而言，依照攻擊的行為來區分攻擊模式，可分為：單一封包、分段式及超載攻擊三類。單一封包攻擊是攻擊者所發出的單一封包即包含有攻擊行為稱之。通常是利用系統漏洞來癱瘓被害主機或取得系統使用者的權限，例如：Ping of death 及緩衝區溢位等均是。分段式攻擊則是利用封包分段（Fragment）所進行的攻擊，因為 IP 層在實作上，對於封包重組演算法有漏洞而讓駭客有機可趁；例如：Teardrop 等。超載攻擊是要讓系統資源耗竭而無法為其他合法使用者服務稱之，也就是常見的 DoS 或 DDoS 攻擊。此類攻擊又可分為造成系統資源耗竭及網路資源耗竭兩種；前者是以消耗 CPU 效率、記憶體、檔案系統配額或其他作業程序等系統資源， TCP SYN Flood 攻擊即是一例。後者是以消耗網路頻寬，使得網路使用率達到飽合，常見的有完全地佔用網路頻寬等。

若依據封包的攻擊特性來區分，攻擊模式可分成 DoS 攻擊和非 DoS 攻擊兩類。前者係以攻擊封包直接令被害主機喪失正常的運作能力或無法提供正常的服務；後者係傳送具有破壞系統的程式或指令的封包，以遂行目標主機之攻擊，使其提供服務之機制陷於癱瘓。

2.1.1. DoS 攻擊

DoS 攻擊又可分成廣義和狹義兩種，上述者為廣義的定義，而狹義的 DoS 攻擊則是以「一部」攻擊主機所進行者。若以多部主機進行者謂之 DDoS。換言之，廣義的 DoS 攻擊＝｛狹義的 DoS 攻擊｝∪｛DDoS 攻擊｝。

以下若未特別強調，則本論文所說的 DoS 是指廣義者，一般而言，完全地佔用網路頻寬、ICMP flood、ICMP Smurf flood、Ping of death、UDP flood、SYN flood、Land 及

Teardrop 等類攻擊均屬之[5-9]，以下分述之。

1、 完全地佔用網路頻寬

利用一台或多台主機同時對一個網域發出大量的封包，塞滿該網域的某一網段（section）的頻寬，使正常要求服務的封包無法抵達伺服主機或提供服務的封包無法到達使用者端。例如，從 T1（1.544Mbps）或者更快的網路發出大量封包，就可以讓與其串聯的 56Kbps 或 128Kpbs 的連線處有大量封包被棄置，換言之，有大量的封包無法順利到達對方，自然也無法獲得應有的服務；反之，也以並聯多個 56Kbps 連線對被害網路，發出大量封包，即被攻擊端使用 T3（45Mbps）網路頻寬也可以將其塞爆，其現象仍是數量龐大的封包被丟棄。

2、 ICMP flood

ICMP（Internet Control Message Protocol）封包的功能是用以測試與回報網路設備的狀態，最常用的就是 ping 指令。ping 指令會對目的主機 S 發出「ICMP Echo」的 ICMP 封包，來詢問 S 是否存在。而 S 收到訊息，則會回覆「ICMP Echo Reply」，通知使用者本機是存在的。

若以假造來源 IP（例如說是 A）發出大量 ping 要求，送至一個或多個伺服器主機 S，則 S 會回應等量的 ICMP 封包給 A，造成 S 與 A 之間的網路流量大量增加，而流量大到沒有多餘的頻寬可以讓一般正常使用者使用，又是一種 DoS 攻擊。

3、ICMP Smurf flood

Smurt 攻擊亦是以假造的 IP（例如，A）對一個網域廣播（Broadcast）IP 位址（xxx.xxx.xxx.255 for class C）發出 ping 要求；於是此網域的所有主機都會回送 ICMP ECHO REPLY 封包給 A 主機，因而造成該網域壅塞。手法和 ICMP flood 相同。

4、 Ping of death

對目標主機發送過長的 Ping 封包（ICMP Echo Request 封包），當這個封包重新組合時即超出緩衝區（Buffer）限制的大小而造成超載，可能會使伺服主機失去正常的功能，甚或當機；例如，乙太網路封包最大傳輸單元有 1518 位元組的限制，使用者卻給予超過 65535 位元組的資料，若伺服主機沒有做好資料長度過長的處理機制，則會覆蓋到其它部份資料，就有可能造成伺服主機發生錯誤而當機。

此類攻擊僅會發生在內部網域的攻擊，因為外部駭客發出的攻擊封包必須經過路由器轉送，每個路由器（Router）所能接受的資料封包大小都不一樣，也就是每個路由器的 MTU（Maximum Transmission Unit）[31]不一樣；如果一個路由器接到一個比本身的 MTU 更大的封包，則會將之切成幾個較小的封包傳送，傳到目的主機時才會被拼湊回原來的大小。所以此類攻擊不可能發生於來自外部網域的攻擊。

5、 UDP flood

UDP（User Datagram Protocol）是一種使用者（Client）端送出封包給伺服主機 S 前不需要事先建立連線，只需要對 S 送出資料即可的協定。攻擊者若發送一連串或大量的 UDP 封包到伺服主機，以塞滿伺服主機端的網路頻寬，稱為 UDP flood DoS 攻擊。

6、 SYN flood

在 TCP 協定裡，如果雙方要建立連結，必須先完成 three-way handshake 的程序。在第一階段，使用者會向伺服主機 S 發出一個連線的請求（SYN），並等候 S 的回應。如果 S 同意這個連線的請求，則會向使用者送出 SYN+ACK 的封包，是為第二階段。當使用者收到後，還需回送一個 ACK 封包，告知 S 目前已經建立連線，這就是第三個階段。接著才是後續的資料傳送。

然而一個系統會分配一定數量的資源，來建立連線。SYN flood 便是利用此種特性耗盡伺服器的資源，使其無法提供正常服務。攻擊者以一個假冒的 IP 位址 A 送出 SYN 封包到系統 B，B 會回送 SYN+ACK 封包。A 有三種可能性：(1)存在且在開機狀態；(2)存在，但未開機；(3)不存在。第一種情況 A 會立刻回應一個 RST 封包告知 B，意思是沒有打算建立這個連線，系統 B 也不會等候；後兩種情形 B 會將這個連線保持在 SYN_RECV 狀態，並放入等候佇列（backlog）中等待，卻永遠收不到回應，一直到連線建立計時器（connection establishment timer）逾時（一般最短是 75 秒，最長可能到 23 分鐘），才終止連線要求。但在此時間之內，攻擊者又陸續送出更多的連線請求，造成 B 上的連線佇列持續爆滿。因為一般的等候佇列都很小（視各 O.S.或系統管理者設定而有所不同，其值可為 5、32、64，最大為 127；但加大等候佇列將消耗更多的資統資源（記憶體），故不宜無限制擴張），攻擊者只要每 10 秒送出幾個 SYN 封包，就可以把一個連接埠完全癱瘓，而伺服主機已無法再接收任何正常的服務要求。

7、 Land

此攻擊方式是利用攻擊軟體發送一個來源和目的位址與通訊埠都相同的連線請求（SYN）封包，使得被攻擊者誤以為是其本身送此封包給自己，因此造成其主機解析 land 封包時，佔用大量系統資源，而使網路功能完全癱瘓。

8、 Teardrop

乃是利用 IP 層重組封包之程式瑕疵。當資料經由網路傳送時，IP 封包經常會因為路由器的最大傳輸單位的原因，而被切割成許多較小片段，這些小片段和原來封包的結構除了一些位移（offset）欄位之外大致都相同。位移資訊主要是用來正確重組 IP 封包。此攻擊方式為發送一對重疊的位移值封包片段到目標主機，重疊的位移在封包重組時，會產生主機誤判封包大小，某些舊版本的 Linux 系統會因而當機。原因是： Linux 核心可以正確地檢查每個分割長度是否過長，卻沒有檢查分割是否過短。Win NT/95 也同樣有這樣的漏洞。

2.1.2. 非 DoS 攻擊

此類的攻擊方式包括：單一封包直接攻擊，即時性跳板攻擊及非即時性跳板攻擊三種。跳板攻擊是：先入侵某主機 S，再於 S 中植入木馬（Trojan），其次透過木馬，亦或以其他方式由 S 直接攻擊目標主機 T 之謂。對駭客 H 而言，攻擊是間接地透過 S 進行的，目的在逃避被攻擊端的偵測。這種攻擊方式可由入侵者直接下指令給跳板主機，做立即性（即時性）攻擊。亦可設定特定時間或是設定一段時間後才進行攻擊，是為非即時性攻擊，這類攻擊非常難以掌握 H 於何時下達什麼攻擊命令給 S，在龐大的網路封包日誌紀錄裡更難追蹤 H 和 S 之間的連線關係，因此，目前對這種攻擊仍難以逆向追蹤到 H，S 只有一直擔任待罪羔羊了。事實上，非 DoS 攻擊可透過下列的方式進行攻擊[1,3,5]。

1、 電子郵件 SMTP（Simple Mail Transfer Protocol）與 POP（Post Office Protocol）等通訊協定

攻擊者可以藉由含有後門程式的病毒電子郵件，進入系統後建立後門，而形成一個通道。

2、 ICMP ECHO、ICMP ECHO REPLY 與 UDP 封包

攻擊者將攻擊指令封裝在特定 ICMP 或 UDP 封包的資料欄內，傳送到已被植入木馬的伺服器中，木馬會執行此指令，再將執行結果封裝在 ICMP ECHO REPLY 內，回傳給攻擊者。著名的 Loki 與 Lokid（分別是用戶端與伺服器端程式）即利用此觀念運作。

3、 TCP 序號預測

攻擊者 H 對目標主機 T 發動小型 TCP SYN Flood 攻擊，造成 T 暫時無法回應其信任主機 S，此時，H 會嘗試傳送一個假冒主機 S 的要求連線封包給 T，並將網路卡設為雜亂模式，再利用 Sniffer 機制觀察其回應的 TCP 封包序號，接著偽造 S 的位址向 T 發送預測序號的 TCP 封包，使 T 誤判為係來自 S，而取得與 T 的正常連線關係，使 H 取代原來 S 的地位，進而入侵 T。就算 S 稍後與 T 恢復連線後，H 仍可繼續和 T 交換訊息。

4、 TCP 連線劫持

是利用 TCP/IP 協定的疏忽，在共享式媒體（Shared Media）上進行，方法是利用 CSMA/CD（Carrier Sense Multiple Access/Collision Detection）傳輸資料時，會將目的封包傳送到同一個區段的所有節點的特性，進行其攻擊，步驟如下：(1)、追蹤連線取得目前封包的 SEQ 與 ACK 編號；

(2)、辨識目前連線的狀態，計有

(i) 正在連線，

(ii) 穩定（stable）沒有資料正在傳輸，

(iii) 主機端發送封包的 SEQ 是否與發送端回應的 ACK 序號相等，

(iv) 用戶端發送封包的 SEQ 是否與主機端回應的 ACK 序號相，

以確定攻擊對象正處於連線狀態；

(3)、取得其封包 SEQ 編號，劫持其正常連線，插入攻擊者的封包而取得連線。

5、 設備、應用程式或服務系統之後門漏洞

攻擊者會先掃描遠端主機 D，找出 D 目前所提供的服務，亦或掃描遠端的網路設備，並存取其系統資訊（如：共幾台路由器及廠牌），依此比對各廠牌各版本的已知後門漏洞或預設帳／密碼。例如：guest、anonymouse、Cisco 的通用密碼為 cisco、3COM 的共用帳號為 admin 和密碼則是 synnet，若使用者並未加以改變，則可順利登入之，以遂行其攻擊。

至於系統漏洞方面，微軟公司 NT 上的 IIS（Internet Information Server） Server 或是 IE 瀏覽器，linux 上的 snmp 等，都有其已知的漏洞或缺失，若管理者或使用者未以最新的 patch 程式修補之，駭客也會利用這些漏洞或缺失攻擊之。

6、 暴力破解密碼

和“後門漏洞”相同，先掃描遠端主機 D，找出其目前所提供的服務，如果 D 是網路設備，則存取其系統資訊，再以兩種方式進行攻擊，第一種，是利用其他的攻擊方式，如後門漏洞等，得到系統的密碼檔或編碼過的密碼檔之後，以特殊的程式（例如：John the ripper）加以解碼，進而得到更高的權限，再搭配其他的攻擊方式（例如，DNS Hijacking，其細節將述於後）進行攻擊。第二種是對 D 所提供的不同服務，用不同的程式進行暴力破解，例如：http 服務，使用 wwwHack，如為 linux 則採用 DES 演算法。若得到加密後的帳號密文檔，亦可破解，只是較費時而已；至於 NT 上的密碼檔也可以用 DumpACL 以及 pwdump 等工具來破解 SAM 密碼檔；目前，也有軟體可以在網路上竊聽 SMB（Server Message Block）的密碼 hash。

「字典攻擊法」是典型暴力破解的工具之一，它是以所附之字典上的單字逐一登入，以單字爲密碼的系統則會被登入，這種攻擊法，每秒約可完成 800 萬筆測試組合。2003 年 3 月趨勢科技發表新聞稿[32]指出，「網路芳鄰」病毒（WORM_DELODER），就是字典攻擊法的傑作，令人驚訝的是「網路芳鄰」僅靠 84 組密碼清單，包含 123、123456、123abc、abc、abc123、Admin、Administrator 等，就滲透上萬部系統，很難想像企業所用系統的密碼，竟然僅是常見的英文單字及簡單數字排列。另外，將生日、姓名、身份證、電話號碼等個人資料，或以空白、阿拉伯數字排列等當作密碼，也是常見的。也曾發生網管人員將所有系統皆用同一組密碼，或使用原廠預設的帳號和密碼，讓駭客不費吹灰之力就能長驅直入這些系統。

7、 DNS Hijacking

駭客曾利用提供 DNS（Domain Name Service）服務的機構所制定的不當政策，(例如：一些網域名稱代管伺服器接受申請帳號的手續太過簡單)，或是藉由各種方式入侵 DNS 主機來達到其目的。其典型的例子便是改掉導向的 IP 位址，這種攻擊常利用 IP 位址對應主機名稱的程序，例如：攻擊者可以試圖讓目標 DNS 伺服主機儲存一筆資訊，而將某 Domain Name，例如，www.xxx.com，對應到一個不存在的 IP 位址，例如，0.0.0.10。當使用者想要查詢 www.xxx.com 時，DNS 回應的是 0.0.0.10，因此所有要送到 www.xxx.com 的資料均送不到對方，使用者也當然永遠都收不到回應，也就阻斷了該 Domain Name 的服務是為 DNS Hijacking。

8、 木馬程式

木馬一般分為伺服器端（Server）和用戶端兩部份，在伺服器端的是攻擊者刻意安裝在目的主機 T（即攻擊跳板）以進行攻擊的部份的，其功能是：監視 T 的封包，一旦收到用戶端下達的命令，便對指定目標（在命令中描述或事先已定義在伺服器端的部份中）進行攻擊。安裝在用戶端的則是用來下命令或控制伺服器端的木馬程式。此攻擊方式需先利用其他攻擊方式，如 UDP 封包、後門漏洞或暴力破解密碼等方法，設法進入 T，在 T 中植入木馬程式，繼而進行實際攻擊（如 DoS 攻擊）或逐行系統的入侵（如竊取機密資料等）。

9、 緩衝區溢位（Buffer Overflow）[22]

此類攻擊是由於程式原始碼設計不當，導致未授權的使用者可以透過執行中的程序，存取系統記憶體中的資料，而造成無法預期的損失。一般可以把緩衝區溢位攻擊分成三個階段：(1)字串長度溢位、(2)覆蓋返回位址及(3)執行攻擊程式。

大部份的電腦程式會在記憶體配置一個區域來儲存或暫時儲存資料，這個記憶區即稱為緩衝區。若在緩衝區中放入超過它能容納的資料，裝不下的資料會溢出到無法預期的地方，通常是會覆蓋下一個鄰近的資料區塊。這種現象最常發生在 C 語言所開發的軟體，因其未提供自動檢查資料界限的功能（例如： strcpy()、sprintf()、gets()、fgets()、scanf()、vscanf()等函式都沒有做界限檢查)，因此緩衝區往往會溢位，(所以程式設計者必須在程式中自行檢查）所造成的結果有：(1)程式以奇怪的方式運作、(2)程式完全當掉或(3)程式繼續運作而不出現顯著的變化等；但最重要的是有可能在溢位發生時，修改其內容。

以上是此類攻擊的第一階段，第二階段則是要插入程式碼，一般是在攻擊程式中插入一小段能產生 shell 的指令，改變程式執行的順序，覆蓋返回位址，並寫入攻擊程式碼的記憶體位址，於是就可以達到第三階段執行攻擊程式了。

若只有單純的緩衝區溢位現象並不會造成安全問題，但如果是以系統管理者的權限運作的程式發生緩衝區溢位的問題，攻擊者便可進一步以系統管理者的權限控制電腦，

造成更大的破壞。另一種狀況是伺服器可能因為所連結的檔案受到緩衝區溢位而停止服務。

10、SQL Injection

網頁的程式撰寫時，並未做好對使用者的輸入做妥善的過濾與查驗，使將其組合成 SQL 指令，傳送給後端資料庫系統執行。若使用者輸入的字串中含有某些對資料庫系統有特殊意義的符號或命令時，便可以直接對資料庫系統下達指令，而造成對資料庫的破壞。

以上說明常見的攻擊模式，此外，也有以找出遠端主機目前所提供的服務，而做不同的攻擊。常見的例子是攻擊 Apache 及 IIS 服務的漏洞，例如，IIS 的驗證不夠嚴謹，利用 URL 的要求：http://target/_vti_bin/shtml.dll/xxx.html，可以得到一個錯誤的訊息：“C:\AbsolutePath\xxx.html”: no such file or folder，從回傳的內容便可得知架設該 IIS 的目錄位置，接著便可以用其他的漏洞來做進一步的攻擊。

2.2. 入侵偵測系統

自 Denning[2]在 1987 年發表入侵偵測系統的論文後，IDS 的研究持續至今，目前已發展出即時且架構複雜的系統。IDS 係針對網路上或主機上的可疑活動進行偵測與分析，藉由「不當使用」（Misuse Detection）及「異常使用」（Anomaly Detection）方式[2]，偵測出外部攻擊者及內部人員對未經授權之資訊系統所做的不正當存取或攻擊行為。

2.2.1. IDS 種類

一般而言，IDS 可以三種方法來分類，一是依入侵偵測所使用的資訊“來源”；二是依所蒐集資訊的分析方法；三是資訊的分析型態[24]。

IDS 依所使用的資訊來源可分為：主機型入侵偵測系統（Host-based Intrusion Detection System，簡稱 HIDS）、網路型入侵偵測系統（Network-based Intrusion Detection System，簡稱 NIDS）及混合型入侵偵測系統（Hybrid Intrusion Detection System，簡稱 Hybrid IDS）三大類[24]。HIDS 通常安裝於一部主機，是藉由分析該主機上的各種日誌檔及各項活動程序來偵測該主機是否遭到攻擊。NIDS 以分析網路封包的方式來偵測所保護的網域是否遭到攻擊，通常是建置在網路的骨幹上。而 Hybrid IDS 以 HIDS 為基礎，監測系統的日誌及重要檔案的變化，亦有 NIDS 監督網路流量的功能，但其布署的位置有很大的爭議性，故此類型 IDS 通常不被建議。

以分析方法區分計有：歸納／分析連線日誌方式和根據網路流量／封包特性兩類。

前者通常是 HIDS 所採取之分析方法,但是 NIDS 也可以將蒐集的封包組合為連線資訊,再加以歸納分析，以研判是否為入侵攻擊。

IDS 的分析型態計包括：不當使用、異常使用與混合偵測三類，前者需先蒐集攻擊活動的規則，再採用比對方式，與建立的規則相同或相似者即判定為疑似攻擊行為。中者則以統計的方法歸納系統或網路正常使用的活動樣本（pattern），依此找出異常活動，而入侵行為其實就是異常活動的子集合。後者則以不當使用為偵測基礎，輔以異常使用確認疑似攻擊，以提升正確偵測率，降低誤判率。

一般而言，IDS 是以不當使用及異常使用或 HIDS 及 NIDS 來分類類，以下敘述之。

2.2.2. 不當使用 vs. 異常使用

「不當使用」的偵測方式又稱為「特徵比對」（Signature-based matching）方式[21]，為正面表列，只有符合攻擊特徵的封包或數個封包，才會遭到攔截；比對的方式是將封包特徵與系統事先定義的攻擊模式資料庫進行分析比對，一旦發現是相似的攻擊模式，即視為攻擊行為。

「異常使用」的偵測方式又可稱為「政策比對」（Policy-based matching）方式[21]，為負面表列，凡是不符合正常規則者即認為是攻擊行為，而予以攔截；換言之，是以識別不尋常的主機運作或網路行為的方式來偵測是否有攻擊行為發生。

此兩種方式各有其優缺點：

1. 網路上攻擊模式日新月益，因此，「不當使用」的攻擊模式資料庫亦需不斷更新，才能防範新的攻擊手法。但此方式可藉由手動微調警示敏感度，以避免過多的誤判警訊。

2. 「異常使用」偵測方式需事先建立系統正常使用狀態的基本模型，再進行網路上的各種行為的比對，惟使用者行為及網路活動非常難以掌握，因此可能會產生過多的誤判警訊。然因正常行為以外者均屬異常使用，以致於具有偵測新的攻擊方式的能力。

2.2.3. 主機型 IDS vs. 網路型 IDS

HIDS 是一軟體程序（Process），在 UNIX 系統，檢查的項目通常包括 Syslog、Messages、Lastlog、Wtmp 等。在 Window 系統上，則檢視系統、安全事件日誌紀錄等檔案。由系統管理者設定或 HIDS 程式預設固定間隔時間 t，HIDS 每隔 t 時間，就會讀取日誌紀錄，並與事先定義的規則比對；若為「不當使用」偵測方式，一旦發現某筆日

誌紀錄符合某項攻擊規則，就會發出警告。若依「異常使用」偵測方式，則發現某筆日誌紀錄不在事先建立的正常使用模型中，才發出警告。

近來，發展出一種以作業系統執行系統呼叫（System Call）的日誌紀錄，配合資通安全鑑識技術[12]，以下指令／命令之習慣、打字速率、連線狀況或使用軟體習慣等，做為 HIDS 辨識攻擊的特徵。同樣地其辨識方式也是可以分成「不當使用」或「異常使用」兩類。

NIDS 是以軟體程序或硬體形式建置於特定的硬體系統上，通常是將網路介面設定成錯亂模式，俾接收到該網域的所有傳輸封包，再利用 Sniffer 機制擷取有用的封包資訊，其次和事先定義的攻擊規則相比對，來判斷可能的攻擊封包。通常，NIDS 是以「不當使用」偵測方式為原則；當然，也可以根據來源位址、目的地位址、來源通訊埠及目的地通訊埠來檢查封包，判斷其是否為攻擊封包。

1.HIDS 之優缺點

優點：

(1) 只要有攻擊行為就會產生紀錄訊息或是作業系統呼叫，不會遺漏任何針對其所在主機的攻擊。

(2) 可以依據日誌紀錄或檢視系統檔案或設定檔是否被竄改、存取權限是否得宜及企圖暴力破解密碼等行為，來判斷是否已經遭到攻擊。

(3) 安裝於主機上，較能和作業系統整合。

(4) 在意合法使用者的異常使用甚於外部攻擊者可採用此法。

缺點：

(1) HIDS 亦為作業系統的程序之一，需佔用系統資源，可能會影響其他軟體的執行效率，特別是檢查系統呼叫的 HIDS。

(2) 監控範圍僅限於所在主機。

2.NIDS 之優缺點

優點：

(1) 通常 NIDS 是配置虛擬 IP，NIDS 可以完全在網路上隱形，所以攻擊者並不會知道自己的攻擊行為已經被監控了。

(2) 一部 NIDS 可以監控其內部網域對外及對內的所有封包，成本效益較高，不像 HIDS 僅監控其所在主機。

(3) 網路上的封包有遵循的各項通訊協定，不會產生 HIDS 因為電腦平台或作

業系統不同而有差異的問題

缺點：

(1) 可能因網路流量太大，來不及蒐集封包，而遺漏了部份封包資訊。

(2) 無法判斷疑似攻擊封包是否已對內部的主機進行攻擊。

(3) 以流量或封包特性來判別是否遭受攻擊，會有比較多的誤判情形；例如：網域內使用者使用 TV/IP 時，網路中會有大量的 UDP 封包，NIDS 可能將此誤判為一攻擊事件。

(4) 無法檢視加密過的封包，如：IPSec 等。

兩種 IDS 的選用各有其風險，必須視組織所要防範的重點而定，當然，在經濟許可下，兩種均選用，可互補其優缺點，而做全面性的防護，唯須考慮所需經費及對系統績效的影響。

然而若駭客取得合法使用者帳號密碼而竊取資料，對於 NIDS 來說此為正常之網路行為，而對 HIDS 亦為合法之存取，兩者皆不會發出被入侵警告，即使在系統日誌檔，仍會保存此紀錄（如果未被駭客刪除），只是被害組織發現被入侵時，亦無從得知駭客的真實身份。

2.2.4. NSMS 入侵偵測系統

筆者曾與實驗室夥伴們共同發展一套搭配 BorderWare 防火牆的入侵偵測系統，稱為 NSMS（Network Security Monitoring System）。系統建置於防火牆之後的主機上，偵測功能包括：

1.檢視本機系統日誌檔

2.檢查重要檔案是否被更動

3.偵測本機所提供的系統服務及安全弱點

4.檢測密碼的安全程度

5.定時抓取防火牆的日誌資料

6.分析流量

7.提示其警示日誌

茲將 NSMS 系統主要功能分述如下：

1.系統設定

(1) 使用者管理：對系統使用者之新增、修改、刪除及權限設定。

(2) 設定 Firewall Profile：指定 Firewall Log 之路徑、指定重要檔案，並設定過濾條件。

(3) 設定抓取 Log 時程：定期抓取 Log file 之設定。

(4) 維護攻擊規則資料庫：對攻擊規則之資料庫新增、修改、刪除及查詢。

2.服務列表及檢測

(1) 系統服務列表：列舉系統現有開放之服務（port）。

(2) 安全弱點報告：

(i)安全弱點檢測報告（包含 WWW、FTP、Mail、NFS、具問題的 CGI 程式、Protocol Stack 等）、對系統版本可能的漏洞提出警告、系統提供的指令（資訊）可能有利於意圖攻擊系統者。

(ii)重要檔案異動偵測：將設定欲保護的系統或其他重要檔案、目錄，利用 Hash function 計算出 Hash 值，若這些被保護的資料遭到篡改，其 Hash 值即與原值不同，依此偵測。

(iii)系統權限檢查：檢查欲保護的系統或其他重要檔案之檔案屬性，是否唯讀或是否隱藏及各帳號對其存取權限等。

(iv)帳號密碼安全度檢查：蒐集各式常用的密碼，使用暴力破解方式，以檢查本機所設之帳號與密碼是否容易被破解。

3.防火牆日誌查詢及異常警示

(1)各項 Log 查詢：包括系統日誌、防火牆日誌之內容查詢。

(2)誤判比對：誤判比對查詢功能。

(3)攻擊警示：對於防火牆偵測出的疑似攻擊，顯示警告提示。

由於此系統具有 HIDS 檢視本機系統日誌檔、偵測重要檔案異動、檢查系統權限、檢查帳號密碼安全度及自我安全弱點檢測等功能；而抓取防火牆的日誌檔，統計傳輸資料量，因此具有部份 NIDS 流量及攻擊警示功能，故為混合式的入侵偵測系統。

2.3. 入侵追蹤相關研究

近幾年學術上發表了不少入侵追蹤系統的文章，初期的研究是針對特定攻擊手法提出追蹤的方式，特別是針對 DoS 及 DDoS 攻擊。近期也陸續發表一些以整體 Internet 聯防架構為基礎的入侵追蹤方法，不過大多仍停駐在理論研究階段。入侵追蹤系統通常是以主動式和被動式來分類。前者是基植特殊機制在封包所經過的網路設備中，封包傳送之中即填入可追蹤之資訊或發出追蹤資訊封包，當發現入侵行為時，只須將攻擊的封包或資訊蒐集組合起來即可找出攻擊來源，例如：IP Marking[6]及 ICMP Message[1]等均為其例。後者則不需改變網路機制，至多僅需做傳輸紀錄，在攻擊發生後，藉傳輸紀錄之資訊來追蹤攻擊來源， Hop by hop[9]、AMN[6]、Sleepy Watermark tracing approach[10]、Digests[8]及 HRIDS[12]等均是。

以下我們將這些追蹤系統依照攻擊模式分為針對 DoS／DDoS 攻擊與適用多種攻擊兩類，分別探討系統架構的及優缺點。

2.3.1. DoS／DDoS 攻擊追蹤

針對 DoS／DDoS 攻擊的追踪方式有下列三種：

1、IP Marking

本方法由 S. Savage 和 D. Wetheral 等[6]所提出，是以目前的 IP 協定為基礎，在封包經過每個路由器時，由路由器以 1/25 機率對轉送的封包進行 Marking 的，做法則是在封包標頭的 Identification 欄位內紀錄經 XOR 計算後的路由器位址；當攻擊發生時，只要蒐集一定數目的封包，就能反算 XOR 路由器位址，以重建攻擊的路徑。然，此方法需在路由器增加 Marking 功能，將會提高路由器的工作負載（Overhead）。

2、ICMP Message

是由 B.M. Leech 和 T. Taylor 在[1]2001 年 10 月所提出，是利用路由器轉送封包 P 時，產生另一目標 IP 相同的 ICMP 封包 C，傳送到相同的目的端，而 C 的內容有 P 的部份資訊；為避免造成網路上的流量負擔，路由器每轉送 20,000 個 P 才產生一個 C。當受害者發現遭到 DoS 或 DDoS 攻擊時，必須在攻擊封包夠多的條件下，途經的路由器才能發出足夠的 ICMP 封包，以重組攻擊之途徑，進而追蹤到入侵者的位置。其缺點與 IP Marking 方法相同，也有路由器負載增加的問題。

3、Hop by hop[9]

是由距離被害主機最近的區域網路邊緣路由器（edge router），判斷攻擊封包是由哪

些鄰近的路由器轉送過來，決定出這些相鄰的路由器集合 R 後，再由 R 中的每一個路由器以同樣的方式找到更外層的轉送路由器，如此遞迴地執行，一直追蹤至網路邊緣，或定義出攻擊封包的可能路徑為止。當有大規模多路徑攻擊時，本方法難以分辨出多條入侵路徑，若是對每一條可能路徑皆進行追蹤，則追蹤效率將不盡理想，且必須於每個路由器紀錄各轉送封包之特徵，會造成路由器相當大的負載，而且，目前亦無統一規格之紀錄檔，對於 DDoS 這種大規模且攻擊機器眾多的追蹤亦難以達成。

2.3.2. 適用多種攻擊

適用多種攻擊之入侵追蹤方法有下列幾類：

1、AMN

是由 T. Baba 和 S. Matsuda 於[6]2002 年初所提出。因為 Internet 的範圍廣大，不可能做到集中式資料搜尋與追蹤，而且眾多網路區段所使用通訊協定也不盡相同，追蹤訊息的交換亦有困難，故作者提出分散式管理架構，將整個 Internet 區分區段（Section）稱為 AMN（Autonomous management network）。AMN 是由 Sensor、Monitoring manager（MM）及 Tracer 三部份所組成；其運作流程：功能和 IDS 相似的 Sensor 一旦偵測出攻擊封包，即刻通知所屬 MM，由 MM 驅動其管轄下之 Tracer，Tracer 比對封包紀錄，辨別出此攻擊封包係由相鄰的設備 N 轉送而來，則將結果回傳給 MM。MM 根據其管轄所有 Tracer 的回覆結果，找出攻擊封包來源，可能是同一區段內之設備或主機，亦或是相鄰的 AMN。如為前者，則可依 MAC address 直接找到該主機（甚至是其使用者／擁有者）；如為後者，MM 乃通知相鄰 AMN 之 MM 繼續向前追蹤，如此重覆 Tracer 的比對、回覆及通知相鄰之 MM，而由多個 AMN 之相互合作，最後定能找出攻擊者真正的位置。

然而，Tracer 儲存各封包所有資訊，在比對資料欄位花費較多時間，因此比對效率較差。且攻擊封包若是通過防火牆對外進行攻擊，此方法所追蹤到的 IP 位址應該就是該防火牆了；另外對於跳板攻擊也只能追蹤到離受害者最近的跳板機器；此方法除了這兩個入侵追蹤上的限制，也未考慮追蹤資訊在 MM、Sensor 及 Tracer 之間傳遞時的保密及安全性。

2、Sleepy Watermark tracing approach[10]

這個追蹤機制是由一個 IDS 觸發，當有攻擊行為發生時，才會喚醒 Watermark 的機制，以避免平常佔用系統資源。該機制是嵌入在應用程式端，會在回傳給攻擊者的封包中加入一段 Watermark 訊息 W，尤如在封包上做一個記號，攻擊者不易發現，只有安裝在各個區域內的追蹤系統才會查覺其存在，並發出警示，且只在回傳訊息時加入，因此只適用於對稱式的攻擊，而如何在封包中加入 W 而不讓駭客發覺，也是一個困難點。

3、Digests

Digests 是一個根據單一個封包就能夠追蹤的架構[8]，稱之為 SPIE(Source Path Isolation Engine)，是屬於 Log-Based 的 IP Traceback 方法，其在各個區域的 Backbone Router 端均設置一個 DGA（Data Generation Agent），負責將每個封包轉成一筆筆的 Digests，而 SCAR（SPIE Collection And Reduction Agent）則位於每個區域內，負責蒐集各個 DGA 的資訊。當攻擊發生時，SCAR 會依照所控管的 DGA 產生出攻擊路徑的一部分，網域上所有 SCAR 通力合作就能產生整個攻擊路徑。而負責控管所有 SCAR 及彙整 SCAR 傳來的區域攻擊路徑，而整合成一完整的攻擊路徑的是 STM（SPIE Traceback Manager），STM 也負責認證每個 Traceback 的請求，以避免遭受 DoS 或 DDoS 攻擊。SPIE 則利用 Bloom filter 的資料結構，以一組 Hash function 來加速查詢和減少碰撞，並且用硬體的 DGA 和 SPIE 來加速資料的處理。SPIE 的優點是針對單一的封包就能追蹤到封包來源，也能克服 IP Spoofing 的問題，缺點是每次進行追蹤時均需整合各 SCAR 的部份追蹤路徑，故追蹤所需時間較長。

4.HRIDS[12]

是由呂芳懌與楊子逸於 2003 年初提出。在作業系與應用程式之間建置智型監視器（Intelligent Monitor，IM），蒐集使用者的所有執行系統呼叫（System call）紀錄，藉此定義使用者的系統使用特徵（user profiles），利用資料挖掘（Data mining）技術，發現不當使用或異常使用的情形，一旦有這類情形，立即主動發出登出（log out）命令，中斷此攻擊連線，並結合入侵追蹤機制 LID（Lightweight Invisible Detector）以找出攻擊來源。

第3章 區域聯防入侵偵測與追蹤系統（UDIDT）

本入侵追蹤系統之架構是以分散式區域聯防為基礎，以 NIDS 為入侵偵測機制，蒐集與內部網域相關的封包，並進行攻擊特徵的比對。由於本論文探討主題為 UDIDT 內的入侵偵測部份，故先介紹此背景架構，再於第四章引出其中入侵偵測 MIDS。

3.1. UDIDT 架構

本論文之背景系統『區域聯防入侵偵測與追蹤系統』(Union Defense of Active Intrusion Detection and Traceback System，UDIDT)，建構 UDIDT 的目的是希望能夠發展一個可以迅速確實地找到入侵者的系統。

UDIDT 架構見圖 3-1，由“追蹤管理者”（Traceback Manager，TM）和“區域網路管理者”（LAN Manager，LM）所組成。

UDIDT 以各網路管理單位（Network Manager Unit, NMU）為單元，TM 位於 NMU 中，是整個 UDIDT 的樞紐，負責接收被攻擊的申訴、發動入侵追蹤及聯合防守的聯繫，採區域聯防，在發生入侵事件時，透過彼此的通力合作，快速地追蹤到攻擊者所在位置。大學校園或企業網路等具有區域管理單位特性之系統即為一個 NMU；以下係以大學校園為例，在發生駭客攻擊時，NIDS 偵測後會通知學校單位所屬之 TM，發動入侵行為的追蹤。NIDS 未發現時，則須由被害者直接向網管人員申訴，由網管人員啟動 TM 之追蹤功能。

LM 用以監督整個區域網路(稱為 NMU 單元)的主幹（Backbone），負責蒐集、儲存及偵測封包資訊。每個 NMU 的封包進、出都是透過一個或多個 BR 傳送，每一 BR 對外的每一連接埠前面分別設置一個交換器（Switch）S，S 必須有 mirror port 的功能，S 會將所經過的封包複製一份到 mirror port，而導入 LM。

LM 是由 Header Processor(HP)、Immediate Detecting Processor(IDP)及 Rear Service（RS）所組成。為了避免這些主機直接暴露在網路上，遭受有心人士的攻擊，我們以虛擬 IP 的方式（Dynamic Host Configuration Protocol，DHCP）來配置這些主機。

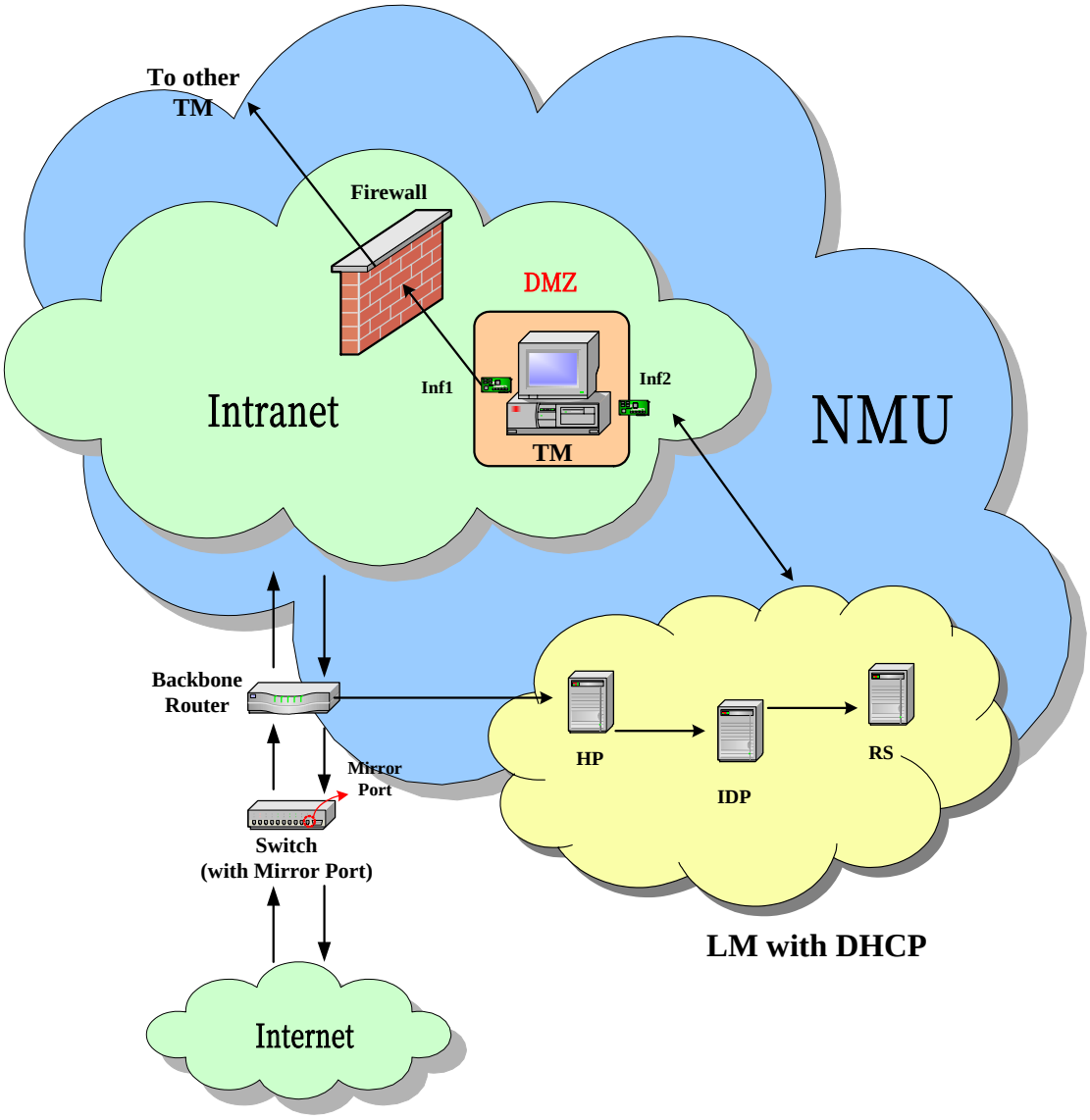

圖 3-1 UDIDT 架構

TM 是唯一能夠直接存取 LM 存放資訊之資料庫的子系統，因此，必須有兩個網路介面（interface），inf0 和 inf1。前者是和外界網路溝通之介面，後者則直接存取資料庫，以查詢入侵追蹤的資料。

TM 也是整個追蹤機制中唯一具有真實 IP 的子系統，否則，TM 和 TM 之間就無法傳遞訊息，當然也無法進行聯合防守。為了保障 TM 免於被駭客直接攻擊，應將之置放於網路中受防火牆保護的 DMZ（Demilitarized Zone）區域內。

3.2. 元件介紹

UDIDT 主要由 TM、LM 及認證機制 TA（Trace Authority）組成，依序針對其內部元件簡單介紹功能之運作。

3.2.1. TM 各元件

TM 負責管理各 NMU，並與其他 NMU 溝通，內部有 NMU 的 ARP （Address Resolution Protocol）Table Set（ATS）、SSL 機制、DHCP 服務、NTP 服務及追蹤單元（Trace Agent）等機制。

1. ATS

定時蒐集該 NMU 所有路由器該時間之 ARP Table，提供各時間點的 ARP 查詢，避免路由器的 ARP Table 更新後，而查詢不到舊有的紀錄，TM 可依 ATS 判斷攻擊封包係來自內部或外部。

2. SSL

乃確保兩 TM 溝通資料的機密性與可靠度。其提供了連線私密性、身份確認與連線可靠性，並搭配認證機制來保障 TM 與 TM 及 TM 與認證機制間資料與傳遞訊息的安全性。

3. DHCP

LM 和 TM 藉由 DHCP 之架設，可避免 LM 資料遭受攻擊。DHCP 之服務可藉由 TM 之一網路介面來設定。

4. NTP

TM 藉由 NTP 服務與最接近的 NTP 伺服器校準網路時間，並由 TM 去校準 HP 的時間，以便能夠給予一致且正確的時間戳記。

5. Trace Agent

負責追蹤要求的發出與處理來自各 NMU 的追蹤要求。Trace Agent 有兩種追蹤方式。一是自主式追蹤，在接獲偵測機制的追蹤要求時由 Agent 自動發出追蹤要求；另一為人工式追蹤，在偵測機制無法偵測出來，且經確認證實有遭攻擊之情形時，網管人員可藉此方式由 Trace Agent 的應用程式介面輸入條件查詢與追蹤。

3.2.2. LM 各元件

LM 是形成多階段入侵偵測系統的所在，其細部結構將於第四章中介紹。

1. Header Processor

利用 Sniffer AP 機制收取所有經由 mirror port 送來之封包，並擷取標頭及部份資料內容，以 Hash 方式產生一 Hash 值（即 Digests），再結合其他封包資訊彙整成一筆 Packet Record（PR）送交 IDP 處理。包括：封包到達 HP 的 Timestamp、MAC Source Address、MAC Destination Address、Digests 及由各層擷取的資料。

2. Immediate Detecting Processor

將 HP 送來的 PR，使用 IP Mask 機制篩檢封包來源及目的地，根據內部網域各 IP 位址及封包進出的方向，排入指定偵測佇列（Detecting Queue）中，並以即時偵測器（Immediate Detector）檢查偵測佇列中各封包的資料，預判可能的攻擊。如為代轉封包（即來源 IP 和目的 IP 均不是本 NMU 者），則進入獨立的外部佇列，不加以偵測立即進入 RS。

3. Rear Service

經過 IDP 分類及偵測後，每個封包的資訊根據封包的傳送方向先存放至 In-bound Temporary Area（IBTA）、Out-bound Temporary Area（OBTA）及 Foreign Temporary Area（FTA），待緩衝區填滿後才全部存入資料庫中，目的是調節資料庫的交易處理；同時也由 Auxiliary Detector（AD）及 Analyzer 執行 IDP 無法偵測的若干入侵行為。

3.2.3. TA 各元件

認證機制主要負責各 TM 間的認證與協助 TM 往下一個 hop 追蹤。TA 包含有 TCA（Trace Certificate Authority）與 TQA（Trace Query Agent）等元件。

1.TCA

TCA 是一 PKI 機制，負責發憑證給各個 NMU 之 TM，並接受各 NMU 的註冊。也為各 NMU 的 TM 與其他 TM 之間認證，確認各 TM 的身份，為一階層式的架構，上層可以為下層做認證。因此，TCA 所發的憑證有兩種，一是為 TM 認證，另一則是替下層 TCA 的認證。

2.TQA

TQA 是蒐集向 TCA 註冊的部份資訊，包含有 NMU 名稱、NMU 管轄下的 BR 之 MAC 位址與其 TM 之 IP 位址。TQA 負責提供 TM 資訊，讓查詢之 TM 知道鄰近 NMU 之 TM 位址。

3.3. 入侵追蹤流程

首先制定 TM 與 LM 之間訊息交換的協定；再以三種入侵情形，來說明整個區域聯防入侵追蹤如何啟動、追蹤訊息如何交換、如何追蹤攻擊來源。

3.3.1. 訊息協定

整個 UDIDT 系統運作上，各個單位之間必須相互傳遞訊息。而，我們的訊息協定包含有三類型的協定，分別為追蹤、查詢、同步等，以下分述之：

1.追蹤協定

主要是追蹤要求處理時各元件的溝通，包括有 Trace Request from Detection System、Trace Request from TM、Trace Request between TM、Trace Notice、Trace Result 及 Trace Alert 等六種訊息格式，其名稱、訊息之內容參數和描述如表 3-1 所示。

表 3-1 追蹤協定的訊息格式

訊息名稱	訊息內容	訊息描述
Trace Request from DS	**Digests, Source MAC address, Source IP address**	**Detect System -> TM(Start)**
Trace Request from TM	**Digests, TM(Start) IP Address**	**TM(Start) -> TM(Next hop)**
Trace Request between TM	**Digests, TM(Start) IP Address**	**Between TM and TM**
Trace Notice	**Next NMU name, Next TM IP**	**Between TM and TM**
Trace Result	**Trace Result Description**	**TM(Start) -> TM(Next hop)**
Trace Alert	**Exception Description**	**Between TM and TM**

2.查詢協定

此類協定是用於 TM 對 LM 的 RS 資料庫查詢，及向 TM 的 TQA 查詢所用。查詢協定包含 Query Digests、Query Digest Result、Query Next NMU、Query Next NMU Result 及 Query Alert 等五種訊息協定，其名稱、訊息之內容參數和描述如表 3-2 所示。

表 3-2　查詢協定的訊息格式

訊息名稱	訊息內容	訊息描述
Query Record	**Hashcode**	**TM->RS**
Query Record Result	**Type1: No such Result** **Type2: Have such Result,** **Source MAC Address, Source IP Address**	**RS->TM**
Query Next TM	**Source MAC Address**	**TM->TQA**
Query Next TM Result	**NMU name, Next TM IP Address**	**TQA->TM**
Query Alert	**Query Exception Description**	**TQA->TM**

3.同步協定

各 TM 對 HP 要求校正系統時間所用，其包含 Synchronous System Time 及 Synchronous Finish 等兩種訊息格式，其名稱、訊息之內容參數和描述如表 3-3 所示。

表 3-3　同步協定的訊息格式

訊息名稱	訊息內容	訊息描述
Synchronous System Time	**System Time**	**TM->HP**
Synchronous Finish	**Finish**	**HP->TM**

當 TM 接收到 IDP 或 RS 發出的 Trace Request from DS 的要求時，TM 以 ATS 先判斷此攻擊是否發自內部，若為內部，則透過應用程式介面通知網管人員；若需要其他 TM 協助追蹤，則會先傳送一個 Trace Request from TM 訊息給上一個 hop 的 TM。循序而下，TM 之間藉由 Trace Request between TM 訊息一路追查，在 TM 發出此訊息同時，同時發出 Trace Notice 訊息通知最開始的 TM，告知其目前欲追蹤之下一個 TM 的 IP 位址。當追蹤完畢，會以 Trace Result 訊息傳回最開始 TM。

3.3.2.　追蹤流程

系統對於入侵追蹤的流程，可分成對外發動 DoS／DDoS 攻擊、外部網路以 DoS／DDoS 攻擊本網域及自我偵測或使用者提出被攻擊申訴三種情形。以圖 3-2 為 UDIDT 追蹤流程，圖 3-3 為例說明追蹤之運作方式。

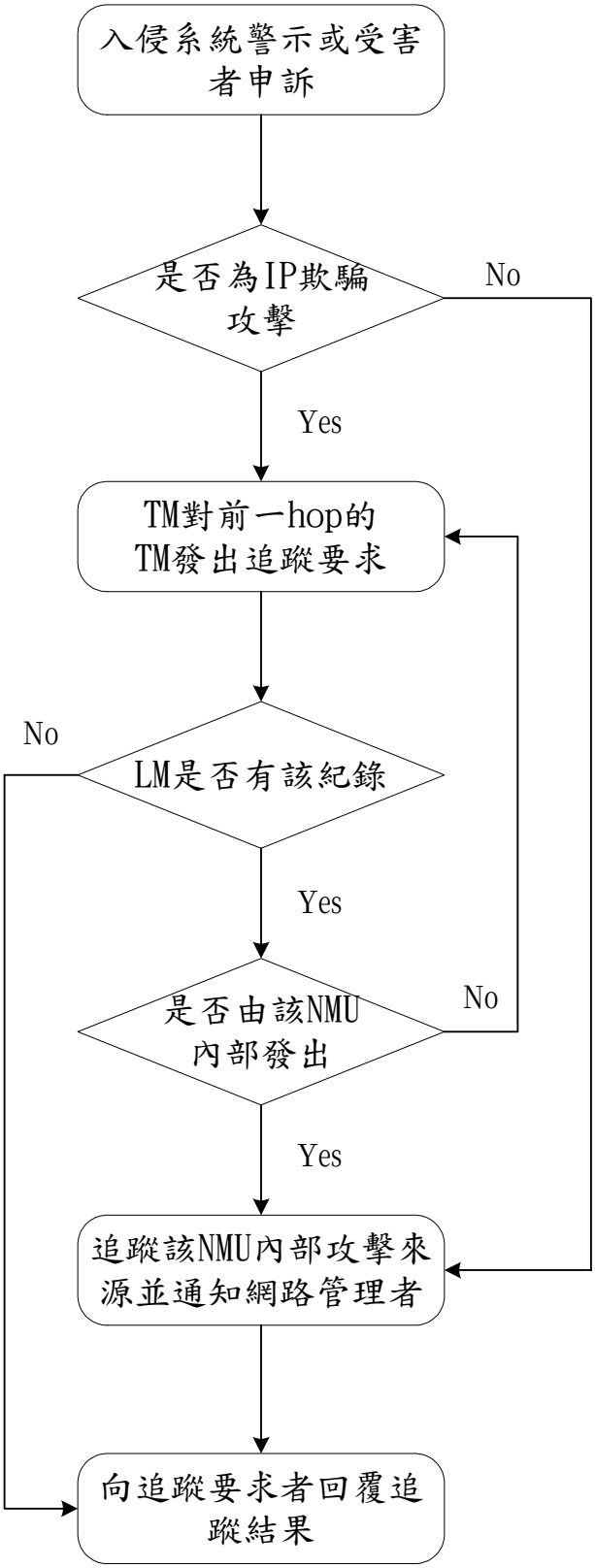

圖 3-2　UDIDT 追蹤流程

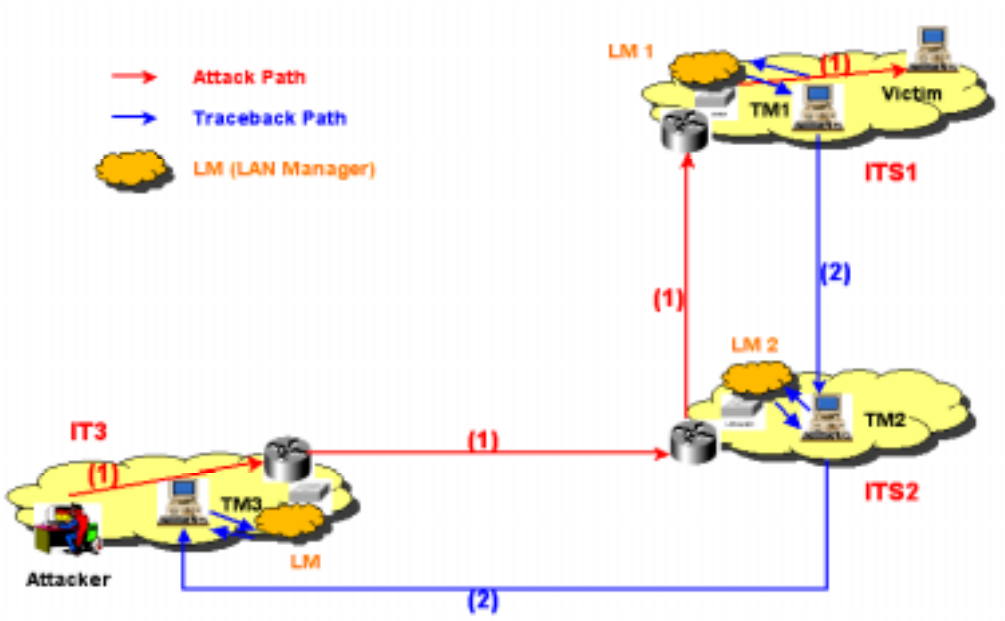

圖 3-3　UDIDT 追蹤示意圖

(1) 對外發動 DoS／DDoS 攻擊

位於 TM3 管轄網域的攻擊者試圖對 TM1 發動 DoS 攻擊，此時，TM3 的 IDP 偵測出來，而發出 Trace Request from DS 訊息通知 TM3。TM3 對照該網域的 ATS，發現為內部攻擊，即可追查出發出攻擊封包之 MAC Source Address 及 IP 位址，立即找到攻擊者，並由其網域之網路管理人員自路由器封鎖攻擊封包。

(2) 外部網路以 DoS/DDoS 攻擊本網域

若每個 NMU 皆設置 UDIDT 系統，通常在 DoS 攻擊封包發出時，攻擊者所在之 UDIDT 即可偵測出來並予以阻止；跨網域的 DDoS 攻擊，在各別網域之 UDIDT 可能會也可能不會偵測出其為攻擊封包，如為後者，就必須依賴被攻擊網域中的 MIDS 來偵測了。假設 TM2 及 TM3 所在的網域同時向 TM1 網域發出 DDoS 攻擊；此時 TM1 的 IDP 偵測出來，便發出 Trace Request from DS 通知 TM1，TM1 對 TM2 發出 Trace Request from TM，TM2 再以 Query Digests 通知其管轄下的 RS，經封包 Digests 比對，若攻擊封包係由 TM2 網域內所發出者，即可查出其攻擊來源主機 IP 位址，RS 將此結果以 Query Digests Result 傳回給 TM2，比對 ATS 得知其 IP，即可找到攻擊主機 A。如為跳板攻擊再以 A 之系統日誌紀錄配合法醫鑑識技術（Forensics），例如，下指令／命令之習慣、打字速率、連線狀況或使用軟體習慣等，繼續在跳板主機所在的 RS 中找出控制封包的來源，再由該 RS 通知其 TM 向前追蹤或比對 ATS（如為來自該 NMU 之駭客），直到找到真正的攻擊主機或真正的攻擊者。

若 TM1 通知 TM2 的攻擊封包係由 TM3 網域內所發出，則 TM2 查詢比對其轄

下 RS 資料，得到前一節點（BR）的 IP 位址，TM2 再發出 Trace Request from TM 給該路由器所在之 TM3，由 TM3 以相同模式繼續追查。

(3) 自我偵測或使用者提出被攻擊申訴

由 IDP 或 RS 偵測到攻擊而通知 TM（假設為 TM1）或 TM1 之應用程式介面接受管理人員下達追蹤指令，便對其轄下 RS 發出 Query Digesst 訊息，比對方式同前述，結果若發現攻擊係來自外部，則依 RS 所提供的來源 IP 位址，發出 Trace Request from TM 給該 IP 位址之相鄰路由器所在之 TM，運作模式依此類推，假設追蹤到 TMK，並發現攻擊來自內部，TMK 再經由 ARP Table 比對即可追蹤到攻擊封包來源主機。

3.4. UDIDT 貢獻及限制

關於提出本系統架構是否能達到較令人滿意的入侵追蹤效果，以下將以系統比較來說明，至於本系統未能解決的部份也在此分別陳述，最後提出另外兩個方案，可以使本系統更為精簡。

3.4.1. 系統比較

本系統和 T. Baba 和 S. Matsuda[6]提出的 AMN 的分散式架構，都是以區域聯防的概念來設計，以下根據其所敘述之演算法直觀地為兩者比較說明：

1、 該系統架構適用於一般企業、小型 Intranet；本系統則不在此限。

2、 該系統之 Tracer 採用記憶體存放封包資訊，限用於即時性（real time）入侵的追蹤。本系統除了即時性地比對檢測之外，存放於資料庫中的封包資訊，亦提供事後之追查，並進行封包之間的交叉關聯性比對。

3、 本系統以 Digests 來追蹤封包，一來加速了封包比對的效率，二來避免侵犯個人網路隱私的法律問題，也不會因而洩漏個人資訊。而該系統必須對所蒐集的欄位都做比對，較無效率。

4、 本系統使用 CA 與 SSL 連線等安全機制，確保在 TM 與 RS 及 TM 與 TM 之間訊息傳遞不被竊取或篡改，並從事身份的識別。該系統並無此保護機制。

5、 本系統在偵測入侵攻擊部份，遇有加密封包（如，IPsec），就無法利用 HP 完成封包解析的動作；若是只有封包資料內容加密（如，SSH），則對本系統之追蹤機制並無影響；在偵測機制方面，因為已加密的內容無法辨識，故單一攻擊則無法偵測，其

它方式攻擊則僅能以封包標頭資訊來判別。

3.4.2. UDIDT 的限制

提供 mirror port 的交換器的運作原理是：各通訊埠不論流進或流出的封包，皆複製一份於緩衝區，待緩衝區資料填滿，再從 mirror port 傳送出去。但 mirror port 的頻寬與其他通訊埠是相同的，若此交換器的每個通訊埠網路流量都非常大，緩衝區將來不及消化，因此會有封包遺失的狀況，而影響本系統蒐集封包資料的完整性。若交換器的通訊埠頻寬各為 k Mbps，則 mirror port 之頻寬應該是 n * k，其中 n 為 port 數量（不含 mirror port），才能應付所有通訊埠的流量。

若是在本系統前端的交換器成為攻擊對象，而造成前端設備的癱瘓，後端 LM 必然無法接收到封包資訊，當然更無法發揮入侵偵測及追蹤的功能。至於 NMU 內部機器攻擊內部主機的部份，因為攻擊封包並未經過 BR，所以本系統無法偵測及追蹤此攻擊行為。

對於跳板攻擊，目前已發表的偵查／追蹤封包的機制均稱薄弱，未來擬繼續研究跳板攻擊追蹤的部份，讓系統能更加完善。

3.4.3. 其他方案

由於 UDIDT 架構是在 BR 對外的每個通訊埠都建置配備 mirror port 功能的交換器，並以切割網域 IP 的方式平行架構 HP 及 IDP，這樣 LM 的系統設計稍嫌繁雜；若 BR 有 n 個通訊埠對外界連接，則需要建置 n 組交換器、HP 及 IDP，如圖 3-4，所以較適合 BR 對外連接較少的網路系統，另外，提出兩種方案以茲參考。

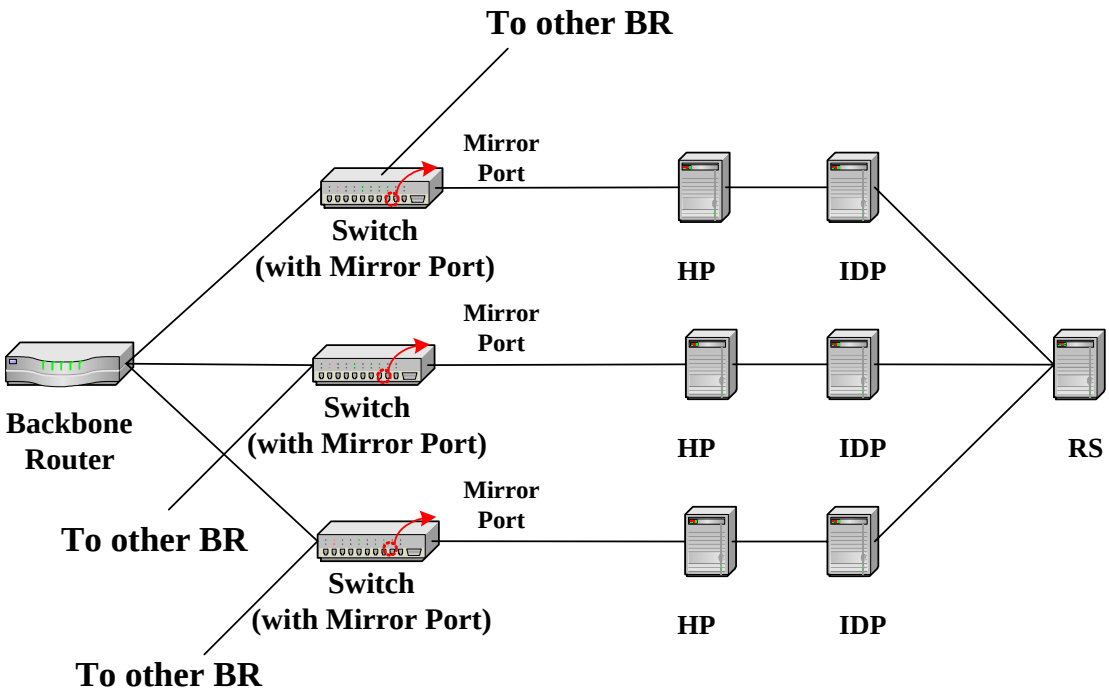

圖 3-4　LM 的硬體架構

1.利用路由器提供的網管工具－Netflow

修改 Netflow 部份功能，(1)將封包複製一份傳至 HP，即模仿 mirror port 的功能，如此，則不需使用 Switch；(2)將所經過的封包皆寫入自身的 MAC address，如此封包本身即可知其前一途經的路由器。系統架構中的每個 BR 僅各需要一組 LM。

2.假設未來路由器增加 mirror port 功能

LM 可直接接收由路由器自 mirror port 傳來的封包，則系統架構同 1.。因為目前業界尚未有此類 Router 問市，故此方案僅提供業者一個新的想法。

第4章 多階段入侵偵測系統

（Multi-Phase IDS）

多階段入侵偵測系統是搭配 UDIDT 中 LM 的 NIDS 入侵偵測機制，負責蒐集所有經過 BR 的封包，比對攻擊特徵；本章將介紹其元件架構的規劃、偵測規則的設計及理論探討。

4.1. MIDS 架構

將蒐集的封包資料分類為流進、流出及 BR 轉送封包三種，並根據所屬網域內 IP 置於不同的資料結構中，分別為偵測佇列（Detecting Queue，DQ）、暫存區（Temporary Area，TA）及資料庫，利用封包分類及資料結構的特性，在此三階段分別進行偵測。第一階段是 IDP，即時地偵測出由外向內的 DoS/DDoS 攻擊、由內向外的 DoS 攻擊及根據單一封包可判斷之攻擊，第二階段是在 RS 中的 AD，將 IDP 資料彙整流進、流出與轉送三種封包集合放置於 TA，AD 即時偵測由內向外的 DDoS 攻擊，以彌補在 IDP 階段無法偵測的部份，最後一個階段係在資料庫的 Analyzer，非即時性根據封包之相關性來檢查是否有攻擊的發生。圖 4-1 所示是 MIDS 之組成架構。

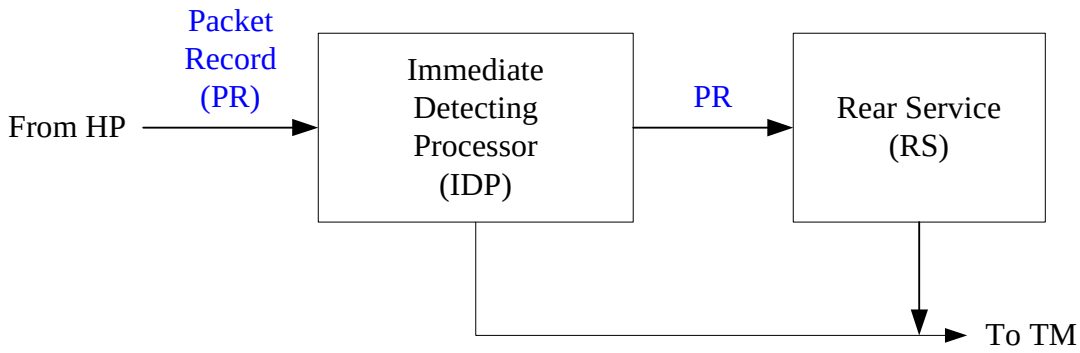

圖 4-1 MIDS 架構

4.1.1. HP 元件

HP 的組成元件如圖 4-2 所示，由 Sniffing Application Program（Sniffing AP）和 Hash Processor 所組成，前者的網路卡係設定在錯亂模式（Promiscuous Mode），以收取所有經由 mirror port 送過來之封包，而將每個封包的第二、三、四層標頭中的某些欄位和部

分的資料內容擷取出來，包含：

1.第二層標頭的 Source Mac Address 和 Destination Mac Address；

2.傳送過程中不會被改變的第三層欄位，如圖 4-3 中的白色部分；

3.第四層的 Source Port 和 Destination Port；

4.資料內容 40bytes；

而將這些資料彙整成一筆 packet data，送交 Hash Processor 處理。

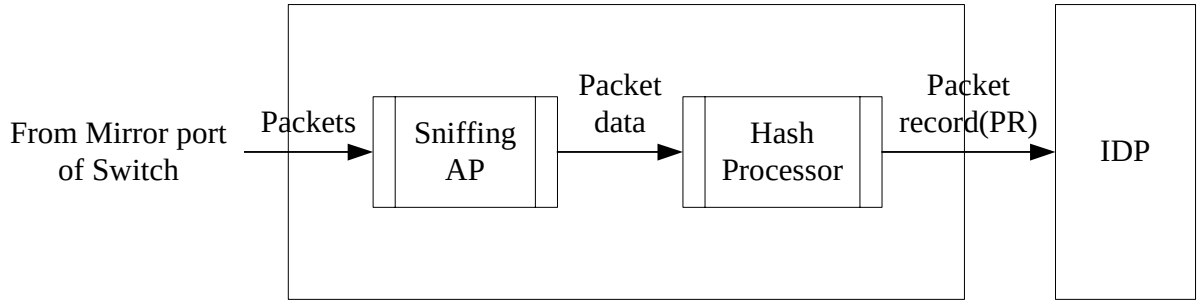

圖 4-2　Header Processor 架構

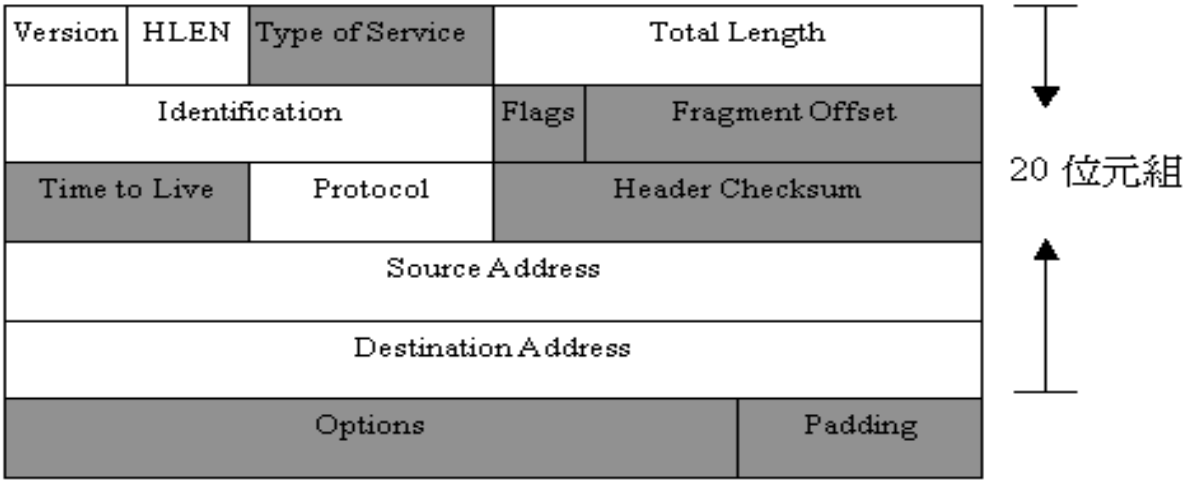

圖 4-3　白色欄位是要紀錄的資料(layer3 header)

Hash Processor 的工作是將第三層標頭資料加上 40bytes 的封包資料內容產生一筆 Hash 值（Digests），再與上述的其他資料(第二、四層的標頭資料及前 40bytes 資料)，彙整為一筆 PR，送交給後端的 IDP 處理。

4.1.2.　IDP 元件

IDP 由 IP Mask、偵測佇列（Detecting Queue，DQ）及即時偵測器（Immediate

Detector，ID）所組成，如圖 4-4 所示。將 HP 傳送來的 PR，經過 IP Mask 區分為流進／流出／轉送，將封包排入相對應的 DQ 中，以 ID 進行平行偵測各 DQ。若發現疑似攻擊行為，ID 則將 PR 的來源 MAC 位址、來源 IP 位址及 Digests 傳送給 TM 執行後續的追蹤，而該封包則丟棄不再處理。亦可以加註標記，之後的偵測機制一旦發現此一標記即不再檢查該封包，檢查通過的封包則送至後端的 RS。

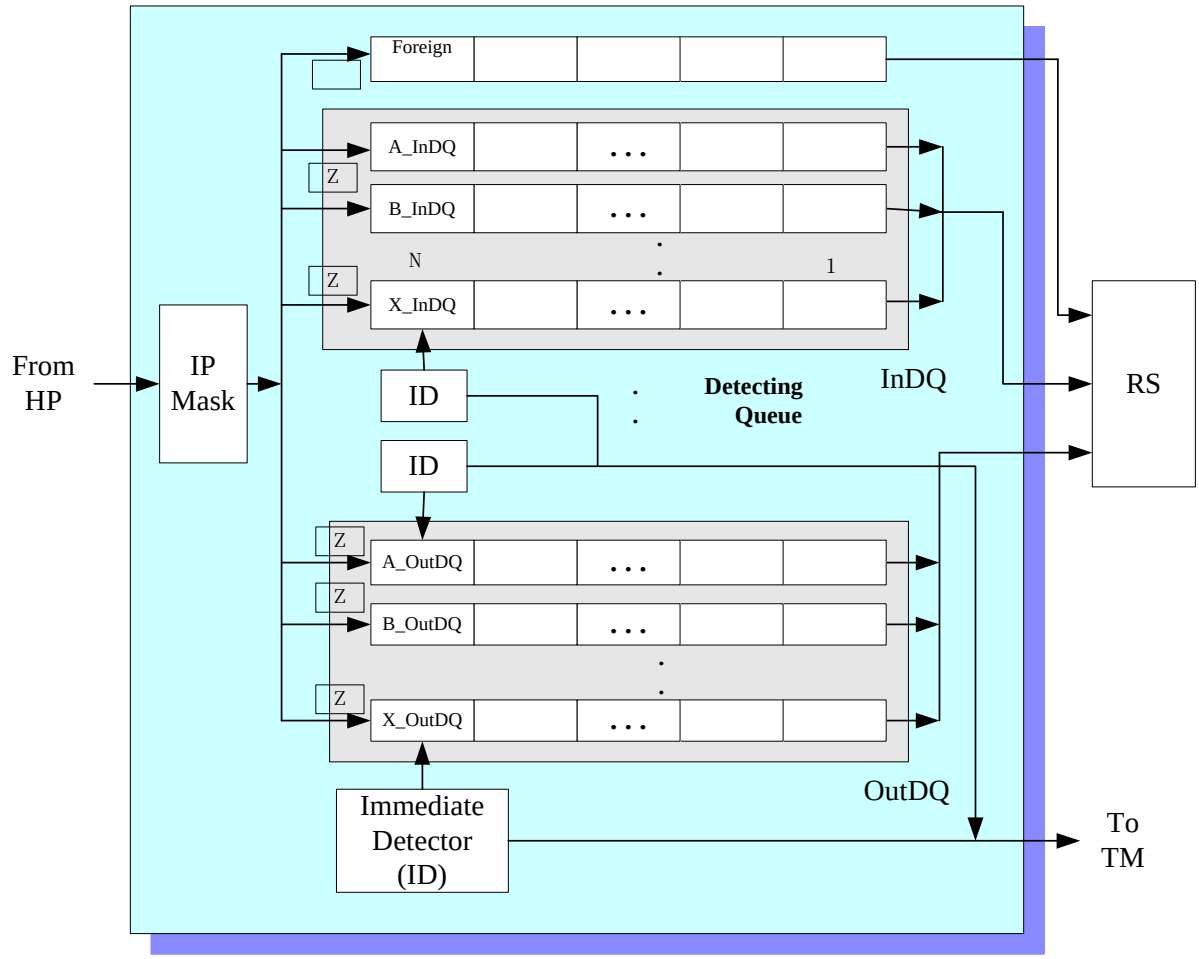

圖 4-4　IDP 架構圖

1.IP Mask 與 DQ 配置

當 TM 蒐集到 BR 最新的 Routing table 亦傳送一份給 IP Mask，依此可界定出該 NMU 之網域範圍。若 PR 之目的 IP 為 NMU 內部 IP，則 PR 之目的 MAC 位址為該 BR 之位址，此資料對於判斷攻擊模式或進行追蹤皆無助益，故刪除該欄位。相同地，若 PR 之來源 IP 為 NMU 內部 IP，則 PR 之來源 MAC 位址為該 BR 之位址，亦將刪除該欄位，以減少不必要的資料處理與儲存空間。

網域內的主機凡有所屬封包通過（流進／流出）BR，即使此機器處於未開機狀態，均依其 IP 動態地配置 DQ（IP 以 A 為例），每一 IP 最多有兩個 DQ，分別為 A_OutDQ

和 A_InDQ，存放來源位址為 A 及目的位址為 A 的封包資訊。DQ 係以記憶體設計之，俾加速比對速度與偵測的即時性。IP 運作流程與 DQ 配置的演算流程如圖 4-5 所示。

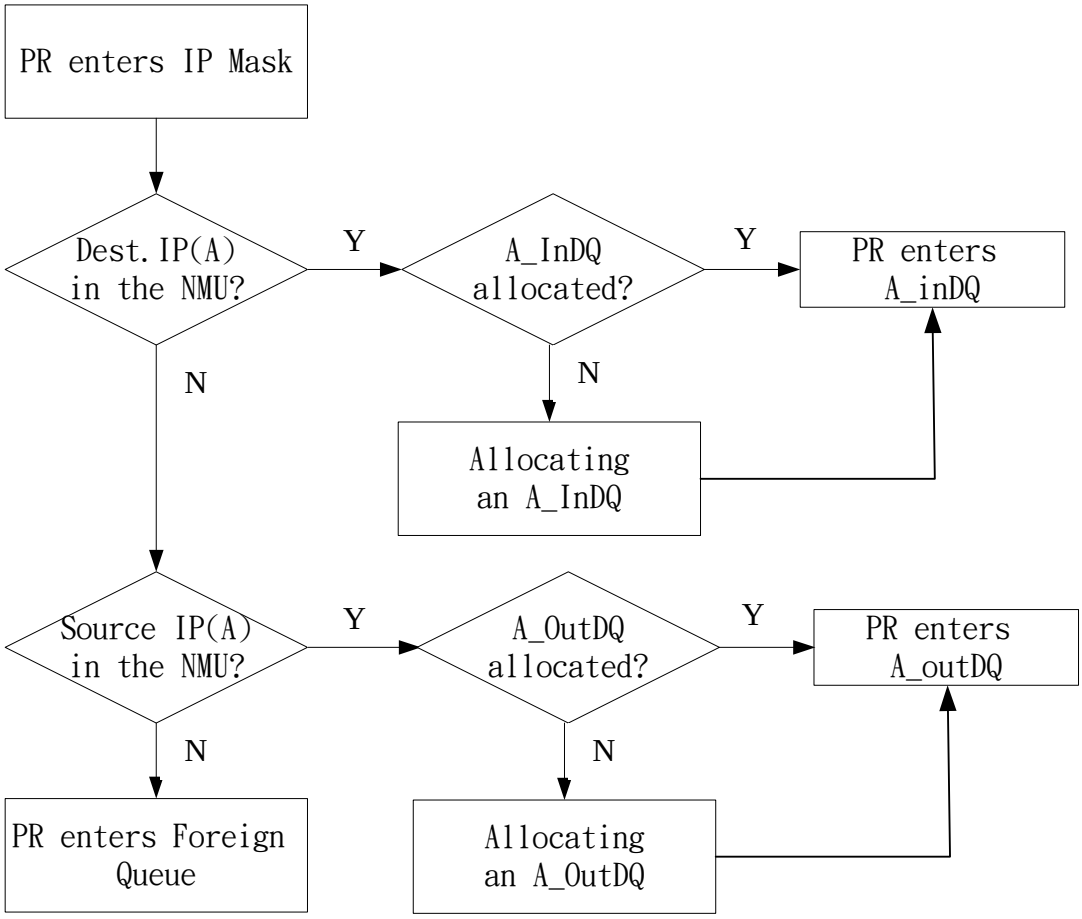

圖 4-5　IP 運作流程與 DQ 配置的演算流程

即使被攻擊者未開機，仍必須將此封包存入佇列中，加以偵測與追蹤；目的是嚇阻攻擊者。

而 DQ 之操作乃依滑動視窗（Sliding Window）為之，讓 DQ 一直維持 n 個最新的封包資訊。為避免某一 IP_DQ 封包流（數）量很少，封包資訊停駐在 DQ 過久，讓所使用之記憶體空間閒置，而設定一 timer，在 timeout 時，強制將此 DQ 中之封包資訊寫入資料暫存區，以釋出該佇列，增加資源之使用率。

2.ID

是第一階段的偵測機制。與 DQ 相同，亦是採動態配置方式；假設網域內有 m 個 IP，偵測佇列數量最多為 2m 個（不計 Foreign DQ），偵測器的數量則可少於 2m 個，一般而言，可統計正常使用數量 k（取平均值或最大值），而設置 k 個偵測器（$k < 2m$）。一旦 PR 進入 In/OutDQ 內，即觸發配置一個 ID 偵測該 PR 是否具入侵特徵。若面對新的 DQ，其佇列中僅有一個 PR，為避免 ID 浪費比對之時間，故僅比對單一封包攻擊模

式。以事先建立之各項攻擊封包特性規則，檢查 DQ 中之 PR，判斷是否有疑似攻擊之封包，換言之，是一個不當使用的偵測方式。

ID 偵查的方式分為比例檢查及單一封包檢查兩類，平行地進行偵測；單一封包攻擊是根據 ID 事先建立的攻擊特徵（Signature）來判別，例如，IIS 服務的漏洞，可依其常見之攻擊指令判斷之。比例檢查以封包之來源、目的及協定等來判斷是否為 DoS 攻擊，根據 W. Lee 的理論[4]，在短暫的時間內，相同封包連續傳至相同目的端，極有可能為一 DoS 攻擊。所以在 DQ 中不論 A_In 或 A_Out，n 個單元裡，2 秒鐘之內有 75% 以上的封包之來源位址、來源埠、目的埠及通訊協定相同，則應判定為一疑似 DoS 攻擊；細節將在下一節中詳細介紹。

如前述，由外對內部網路的 DoS/DDoS，攻擊型態可分成佔用頻寬及佔用系統資源兩類，佔用頻寬的特徵是大封包或封包數量多，封包協定相同（如：ICMP），故可用比例偵測來判別；佔用系統資源的攻擊又可分為癱瘓服務及癱瘓主機，前者的特徵與佔用頻寬相同，加入 port 編號即可加以判斷；後者可依主機的連線上限加以判別；而內部對外的 DoS 的判別方式與由外對內之 DoS 相同，只是一個在 IP_InDQ(即 A_InDQ)，一個在 IP_OutDQ(A_OutDQ)中偵測。

網路流量通常是 TCP 或 IP 封包，一旦發現大量 ICMP、UDP 封包，極有可能是攻擊行為，但可能亦是大量合法的 UDP 封包，例如：IP／TV 即是；唯在偵測器比對時，可以加上 Digests 一起判斷。但是部份 DoS/DDoS 乃利用攻擊程式，例如：w32ddos 等，可以產生大小不同的封包。因此，若是有大量相同協定但內容不同（Digests 不同）的封包，不判定為攻擊，將讓駭客有機可趁；是故我們寧可誤判，而讓網路管理者與使用者自行判斷是否確實為攻擊。

4.1.3. RS 各元件

RS 是由三個暫存區、輔助偵測器、資料庫、分析器及查詢處理器（Querying Processor，QP）所組成，如圖 4-6 所示。由 IDP 中 InDQ、OutDQ 及 Foreign DQ 傳送來的 PR 分別先放置於 IBTA、OBTA 及 FTA 緩衝，再寫入資料庫。IBTA 存放來自 InDQ 的封包資料。OBTA 集中來自 OutDQ 的資料。FTA 放置來自 ForeignDQ 的資料。另設置 AD 彌補 ID 無法偵測 NMU 內部對外發動 DDoS 之攻擊行為。

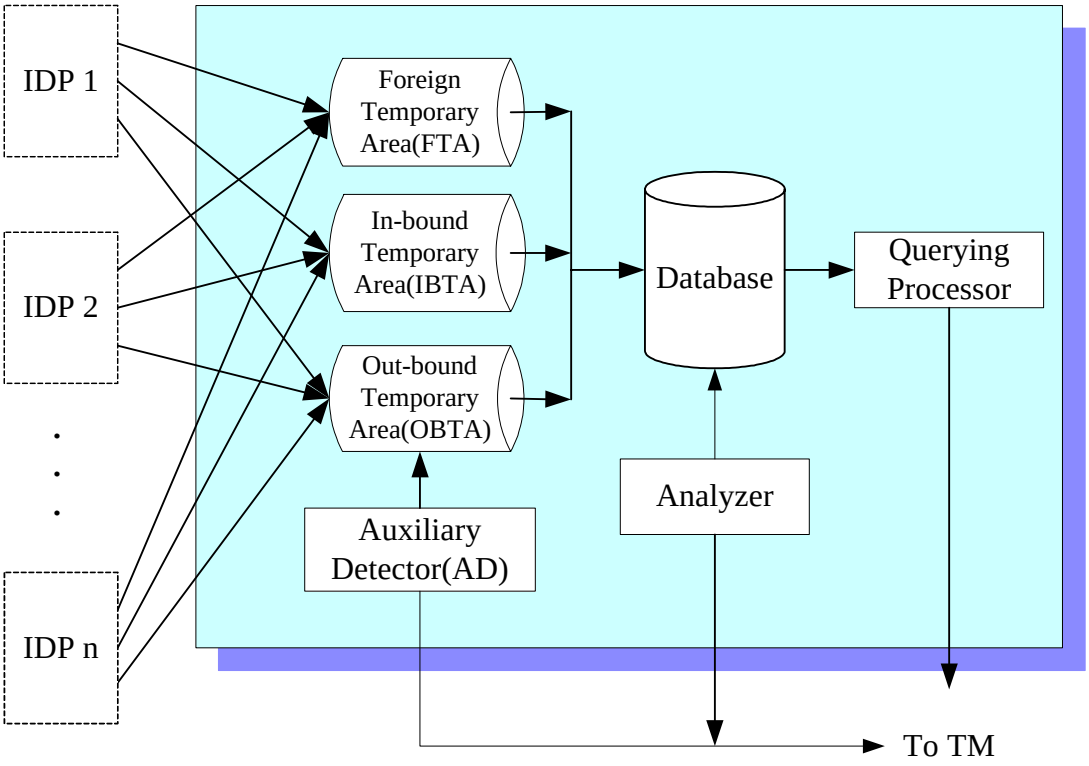

圖 4-6　RS 架構圖

1. TA

所有從 DQ（不論 A_InDQ 或 A_OutDQ）滑出（slide out）的 PR，都先存放至 TA，待 TA 填滿後才全部存入資料庫中，以緩衝在同一時間或短暫的時間內大量封包湧入資料庫中，減少資料庫交易（transaction）的數量，避免有一個封包滑出便寫入一次，俾調節資料庫的交易處理績效。在 TA 資料移往資料庫時，可能會有 PR 擬自偵測佇列進入緩衝區，本系統係利用 Double Buffer 方式處理，一方面亦加速處理速度。

2.AD

此為第二階段偵測機制。在 OBTA 設置一個 AD，目的是偵測內部對外發出的 DDoS 攻擊，原因是這些攻擊封包分散在若干 A_OutDQ，不易偵測出來，然而若將之彙集在 OBTA，雖然封包來自內部不同 IP，但其目的端相同，判斷由內對外的 DDoS 攻擊。這個判斷器判斷的對象不是 Digests，而是 PR 本身，因為來源 IP 不同，Digests 亦不同。我們將根據 PR 中之來源埠、目的位址、目的埠及通訊協定是否相同來研判。

3.資料庫

各 TA 填滿後，則存入資料庫中，分別以對內傳輸封包資料檔（INTB）、對外傳輸封包資料檔（OUTTB）及外部轉送封包資料檔（FTB）放置來自 IBTA、OBTA 及 FTA 之資料。資料庫中原則上存放一週的 PR，以逐日的方式，將一週以前的 PR 移至歷史資

料庫，再儲存一個月的資料，而每一個月由網管人員定期將資料備份至磁帶予以保存。

4.分析器

資料庫中設置一分析器，是第三階段偵測機制，利用關聯性資料庫特性，進行封包資料交叉關連性比對。以期找出：Half-opened connection、Teardrop、Trojan 及 IIS 網頁伺服器漏洞等攻擊模式。

5.查詢處理器

QP 為 TM 向資料庫查詢封包資訊之介面。TM 傳送”Query Digests”訊息至 QP，QP 依 Digests(Hash 值)查詢(訊息內容於 3.3.1 中定義)，所得結果即以”Query Digests Result”訊息回覆。

4.2. 偵測器

4.2.1. 攻擊偵測規則探討

又根據 W. Lee 的理論[4]，可利用主機的連線日誌檔（Connection Log），找出可能攻擊的連線。作者建立了一些挖掘模型（mined pattern）及演算法，希望自日誌檔中大量的連線紀錄，找出循序相關性（sequential correlation）、異常使用（anomaly）連線或不當使用（misuse）連線。經過他們的實驗結果，大致上準確度可達 99.1%。本系統仍依其理論，只是將連線日誌檔轉換為僅用若干封包欄位等資料。

針對第二章中介紹的攻擊模式，在此針對以封包格式相同的 flood 攻擊，分析及歸納各模式之攻擊特徵。

1.完全地佔用網路頻寬

此類模式是利用大量封包塞爆被害主機之網域頻寬，令其無法提供正常服務，而攻擊封包可歸納出以下三個特性：

(1) 目的位址相同

(2) 來源位址相同（若為 DDoS 攻擊，此項不符合）

(3) 封包協定相同

2.ICMP flood

此類模式係利用傳送大量 ICMP Echo 封包，令被害主機疲於回覆 ICMP Echo reply，使其癱瘓而無法提供正常服務，而攻擊封包可歸納出以下三個特性：

(1) 目的位址相同

(2) 來源位址相同（若為 DDoS 攻擊，此項不符合）

(3) ICMP 封包（IP 封包 PROTOCOL=01）

3.ICMP Smurf flood

此類模式的手法亦是傳送 ICMP Echo 封包，但其利用傳送給網域廣播位址（xxx.xxx.xxx.255）進行網域廣播的方式，不必發出大量封包，即可讓被害主機疲於回覆 ICMP Echo reply，令其癱瘓而無法提供正常服務，攻擊封包可歸納以下二個特性：

(1) 目的位址：xxx.xxx.xxx.255

(2) ICMP 封包（IP 封包 PROTOCOL=01）

4.Ping of death

此類攻擊乃以過長的 ICMP 封包，使被害主機失去正常功能，故檢查封包大小是否符合其正常範圍，例如：Token ring 的封包最大為 65535bytes，FDDI 為 4500bytes，Ethernet 的封包正常為 46~1500bytes，至於封包大小若為 8~32bytes 已是不正常封包。特徵是：

(1) 不符合各協定封包 size 大小範圍

(2) ICMP 封包

5.UDP flood（IP 封包 PROTOCOL=11）

此類攻擊以發出大量無需建立連線的 UDP 封包，塞滿被害主機的網路頻寬，可歸納出以下三個特性：

(1) 目的位址相同

(2) 來源位址相同（若為 DDoS 攻擊，此項不符合）

(3) UDP 封包

6. TCP SYN flood

此類攻擊則是利用 three-way handshake，系統會分配資源來建立連線的特點，而耗用被害主機的資源，使其無法提供正常服務。故歸納以下三個特性：

(1) 目的位址相同

(2) 來源位址相同（若為 DDoS 攻擊，此項不符合）

(3) TCP SYN 封包

7. Land

此攻擊方式是使得被攻擊主機解析封包時，無法辨識，系統卻以大量資源繼續不斷地解析之，而使被害主機癱瘓。共歸納出以下五個特性：

(1) 目的位址相同

(2) 目的埠相同

(3) 來源位址相同

(4) 來源埠相同

(5) TCP SYN 封包

8. Teardrop

利用 IP 層重組封包之程式對於重疊的位移在封包重組時，產生主機誤判封包大小，某些版本的 Linux 系統會因而當機。計歸納出以下二個特性：

(1) IP 封包片斷（flag：b1=0 封包可被切割）

(2) 相同 identification 之封包片斷，判斷 fragment offset 欄位加上其封包大小應該等於下一封包片斷之 fragment offset，例如：fragment offset#1+total length(1) = fragment offset#2。

9. IIS 網頁伺服器漏洞攻擊

採用節錄 SATAN Attack command 81 條規則（請參考附錄 A）

10. Buffer Overflow

因為此類攻擊有兩種方式，可以畸形封包或夾帶攻擊指令發出攻擊，故可以第 4. 及第 9.兩類方式之特徵。

(1) 若是以封包不規則大小方式攻擊，則同 4

(2) 若是包含指令的封包，則同 9

11.後門、木馬攻擊

根據常見特洛伊木馬之常用的通訊埠（如附錄 B）為其特性。

本系統在哪個位置偵測 DoS/DDoS 攻擊整理如表 4-1。「個數及方向」表示攻擊主機與被害主機的個數及方向，←代表封包流向為由內對外，→代表封包流向為由外對內；例如，第 3 項中的 1 ← n，乃是 NMU 內多部主機對外發動攻擊，被害主機為單一機器。第 6 項中的 n → 1，則是 NMU 外部多台主機對內單一主機發動攻擊。

表 4-1　DoS/DDoS 攻擊與偵測位置

項目	個數及方向	偵測位置
1	1 ← 1	OutDQ
2	1 → 1	InDQ
3	1 ← n	OBTA
4	1 → n	InDQ
5	n ← 1	OutDQ
6	n → 1	InDQ
7	n ← n	OBTA
8	n → n	InDQ

其中項目 1、2、4 及 5 類屬 DoS 攻擊，項目 3、6、7 及 8 則屬 DDoS 攻擊，前者中項目 4，以 NMU 內部多部被害主機個別來看，攻擊來源為單一主機，故屬 DoS 攻擊，反之，項目 5 亦為相同原因。後者中項目 7，以 NMU 內部多部被害主機個別來看，攻擊來源為多部主機，故屬 DDoS 攻擊，反之，項目 8 原因亦相同。

而項目 1 與 5 可歸為相同攻擊，攻擊方向及偵測位置亦同，因為後者是 NMU 內部單一主機對外部多台主機發出攻擊，對各別被害主機而言實為 DoS 攻擊。依此類推，2

與 4 同，3 與 7 同，6 與 8 亦同。

而從表 4-1 中可歸納出凡是 1 → X（X ← 1）者為由外（內）對內（外）之 DoS 攻擊，n → X（X ← n）為由外（內）對內（外）之 DDoS。

4.2.2. ID 偵測流程

ID 對 InDQ 的偵測對象及其規則，請參考表 4-2，對 OutDQ 的偵測及其規則，見表 4-3。攻擊模式欄位中之括弧內標號代表上一節中所歸納的 11 種攻擊特徵。

表 4-2 ID 針對 InDQ 的偵測規則

編號	對象	攻擊模式	偵測規則
A	InDQ	DoS(1, 2, 5)	source IP、port# destination port# protocol
B		DDoS(1, 2, 5)	source port# destination port# protocol
C		Smurf(3)	Protocol is ICMP Destination IP is xxx.xxx.xxx.255
D		Packet size(4)	Maximum size: Token ring 65535 bytes Ethernet 46~1500 bytes Minimum size: 8~32 bytes
E		Land(7)	Source IP = Destination IP Source port# = Destination port#

表 4-3 ID 針對 OutDQ 的偵測規則

編號	對象	攻擊模式	偵測規則
F	OutDQ	DoS(1, 2, 5)	source port# destination IP、port# protocol
G		Smurf(3)	Protocol is ICMP Destination IP: xxx.xxx.xxx.255
H		Packet size(4)	Maximum size: Token ring 65535 bytes Ethernet 46~1500 bytes

			Minimum size: 8~32 bytes
I		Land(7)	Source IP = Destination IP Source port# = Destination port#

ID 對 InDQ 的偵測流程分為單一封包比對（One way check）及各封包相比對的比例偵測（Ratio check）兩模組。

One way check 共計只有 ICMP Smurf、Packet size 及 Land 等，均只要以單一條件即可偵測是否為攻擊封包，故此三個程序採平行處理，以增加執行速率。一旦在此程序發現攻擊封包，因為攻擊特性非常明確，乃將其 DR（Dangerous Rank）設定為 High。如圖 4-7 所示。

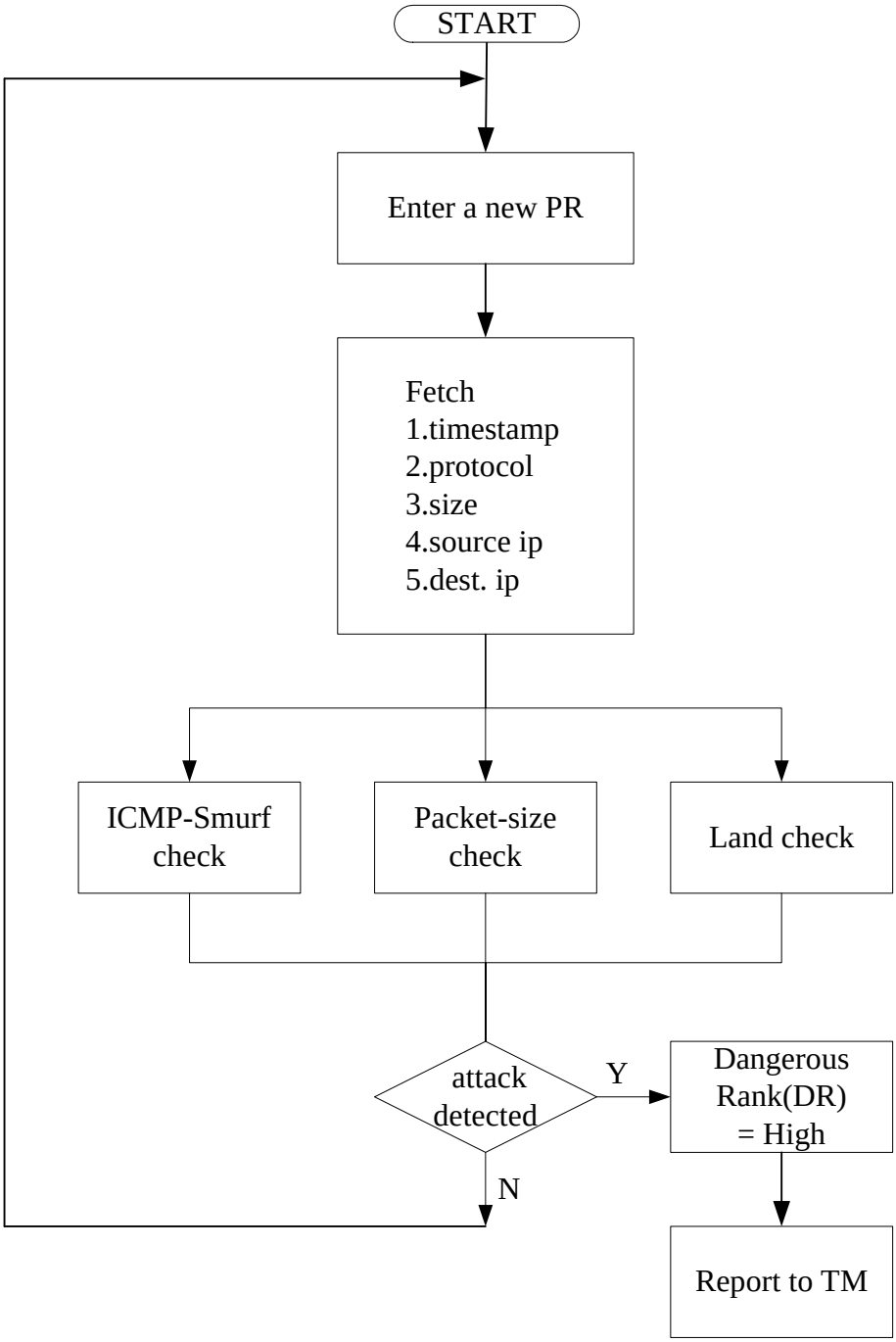

圖 4-7　ID 偵測之 one way check 流程

在 Ratio check 偵測模組中，將新進 PR 先放置於 Buffer Z，若佇列未被全部填滿，則先取出 Buffer Z 的 timestamp、protocol、source ip、source port、dest port 及累計計數參數（counter、bytes 及 timer），再進入模組（Check Previous PRs）內，採用循序比對的方式，新進的 PR 逐一與佇列中其他 PR 比對，而分別計數 DoS 及 DDoS 攻擊封包之個數（counter）及大小（Bytes），由此二項指標數字與總合數相除得其比例。反之，若佇列是滿的，則直接呼叫 Front-Rear 模組，直接減去移出 PR 的個數及大小，並加上將移入佇列之 PR 的個數及大小。根據 W. Lee 理論[4]，在兩秒鐘之內，有相同連線達 75%

可視為 DoS 攻擊，本論文依之。流程如圖 4-8 所示，其中 n 代表佇列長度。

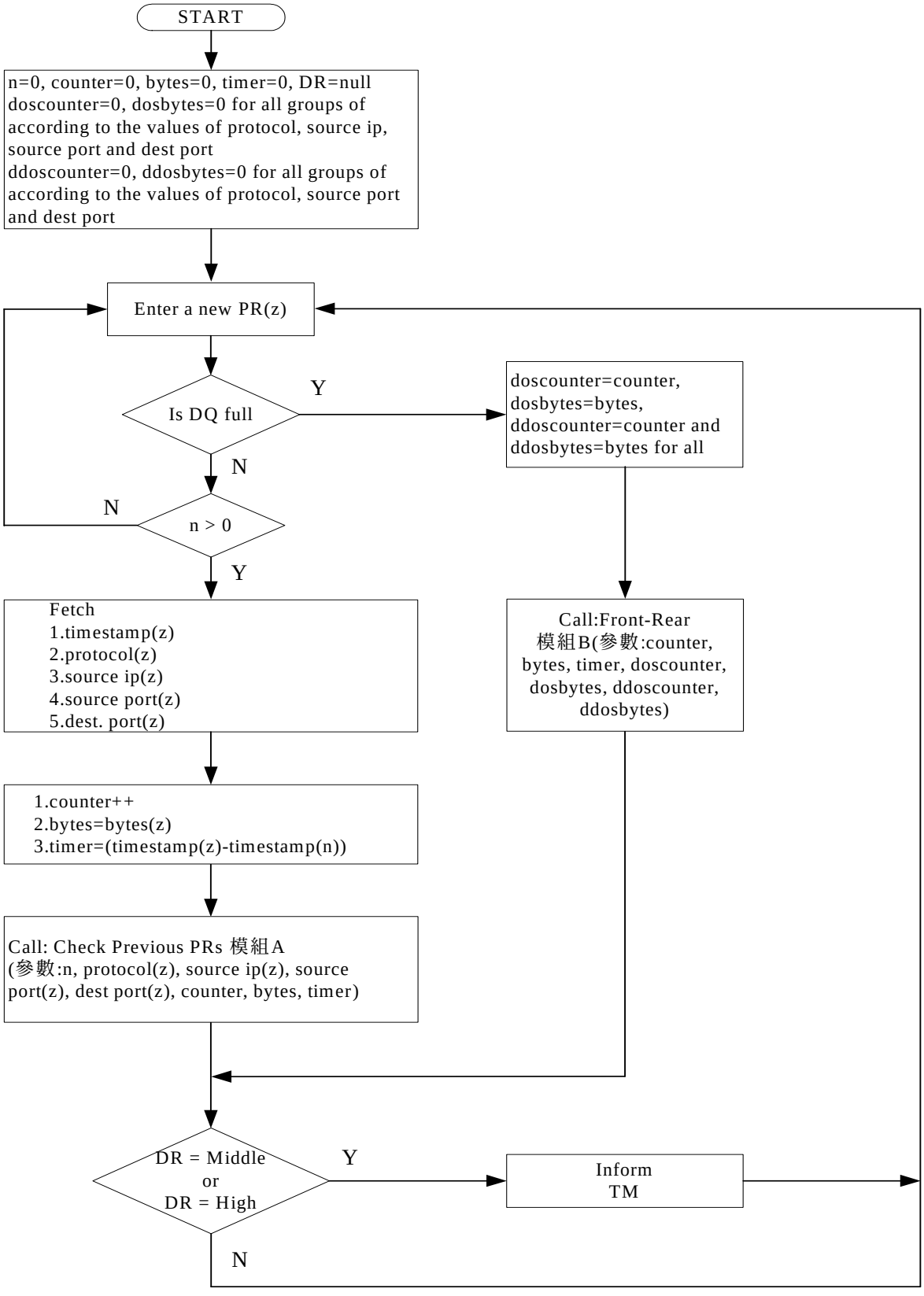

圖 4-8　ID 針對 InDQ 之 Ratio check 偵測流程

Check Previous PRs 模組之流程如圖 4-9 所示，持續累計進入該佇列所有封包個數（totcounter）、大小（totbytes），總時間（tottimer）則是由每個封包進入佇列時之時間標記，前後封包兩兩相減累計而得；若偵測符合結果 DDoS 攻擊之規則，則以 ddos_counter 和 ddos_bytes 分別累計該攻擊之封包個數、大小，若偵測符合結果 DoS 攻擊之規則，則分別以 dos_counter、dos_bytes 累計 DoS 攻擊之二項指標數字；當總時間大於或等於 2 秒鐘，則以累計計數參數與總合數相除（例如：ddos_counter/counter）得其比例來判斷是否為攻擊行為。

Front-Rear 模組之流程如圖 4-10 所示，係以記憶體空間換取比對時間的方式，佇列最後一筆 PR 將離開 DQ 而進入 TA 時有三個步驟，一是將其 DDoS 計數參數（依 protocol、source port 及 dest port 區分），包括 ddoscounter 及 ddosbytes 從累計參數中扣除，二是將其 DoS 計數參數（依 protocol、source ip、source port 及 dest port 區分）doscounter 及 dosbytes 從累計參數中扣除，三是扣除該佇列之總封包數、大小及時間（totcounter、totbytes 及 tottimer），之後才是將 Buffer Z 的 DDoS、DoS 計數參數加到對應的累計計數參數中，並立即判斷其比例數值，而不必逐一比對。

最後得出的結果，UDP 封包可能會產生誤判，例如：IP/TV，所以僅將 DR 設為 Middle。至於 ICMP 及 TCP 封包則無這方面的疑慮，所以這兩種封包一旦符合偵測規則，均視為攻擊行為。

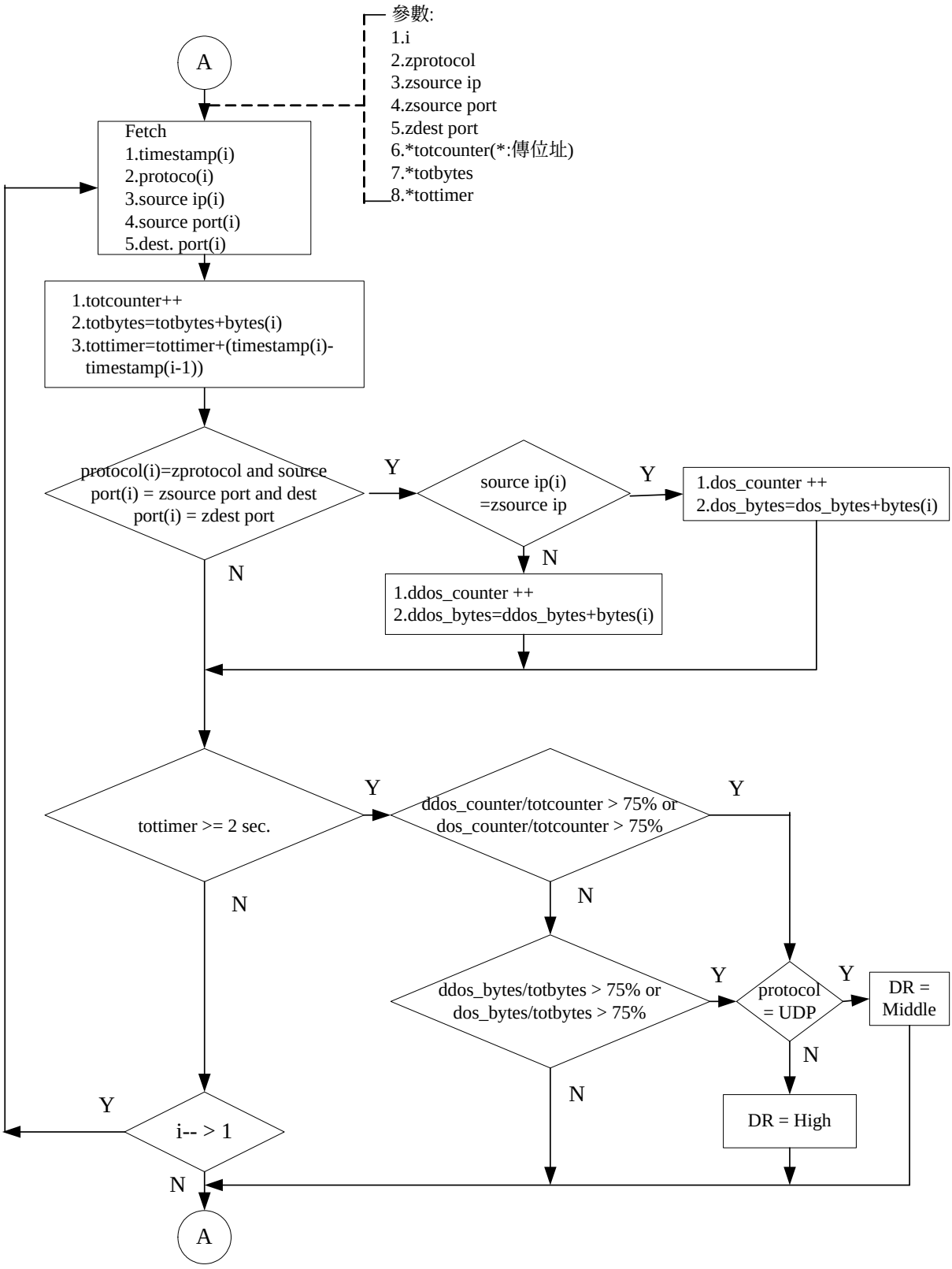

圖 4-9 Ratio check 之 check previous PRs 模組

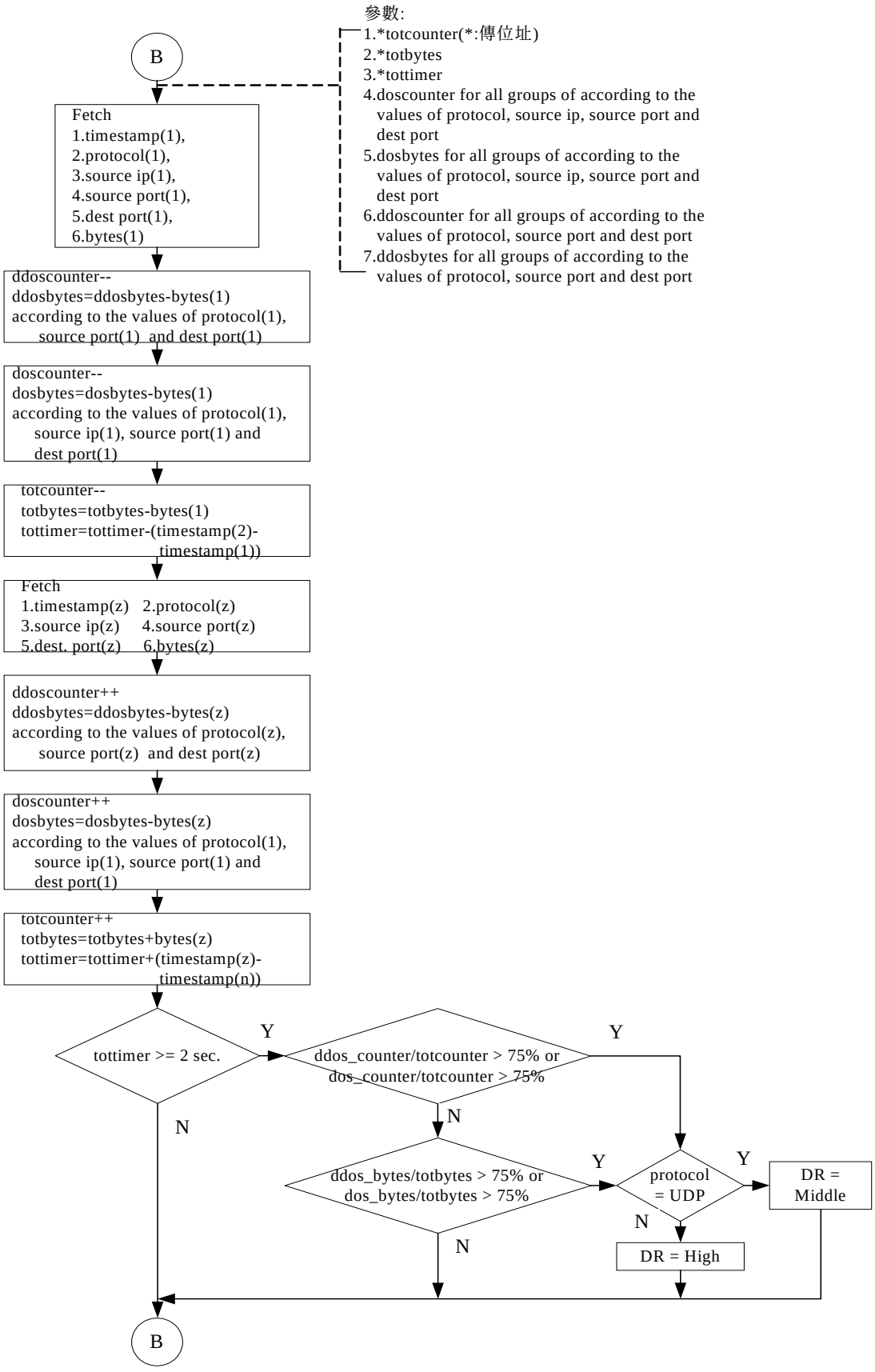

圖 4-10 Front-Rear 模組

ID 對 OutDQ 的偵測流程與對 InDQ 相同，同樣分為 One way check 與 Ratio check；其中僅 Ratio check 有差異，原因是在 Out_DQ 只能偵測出由 NMU 內發出攻擊的 DoS 攻擊，所以 ID 針對 Out_DQ 的 RatioCheck 偵測流程和圖 4-9 相同，僅其 check previous PRs 模組由原先之圖 4-10 改成圖 4-11 之流程，圖 4-8 之演算法及圖 4-10 之 Front-Rear 模組均不變。

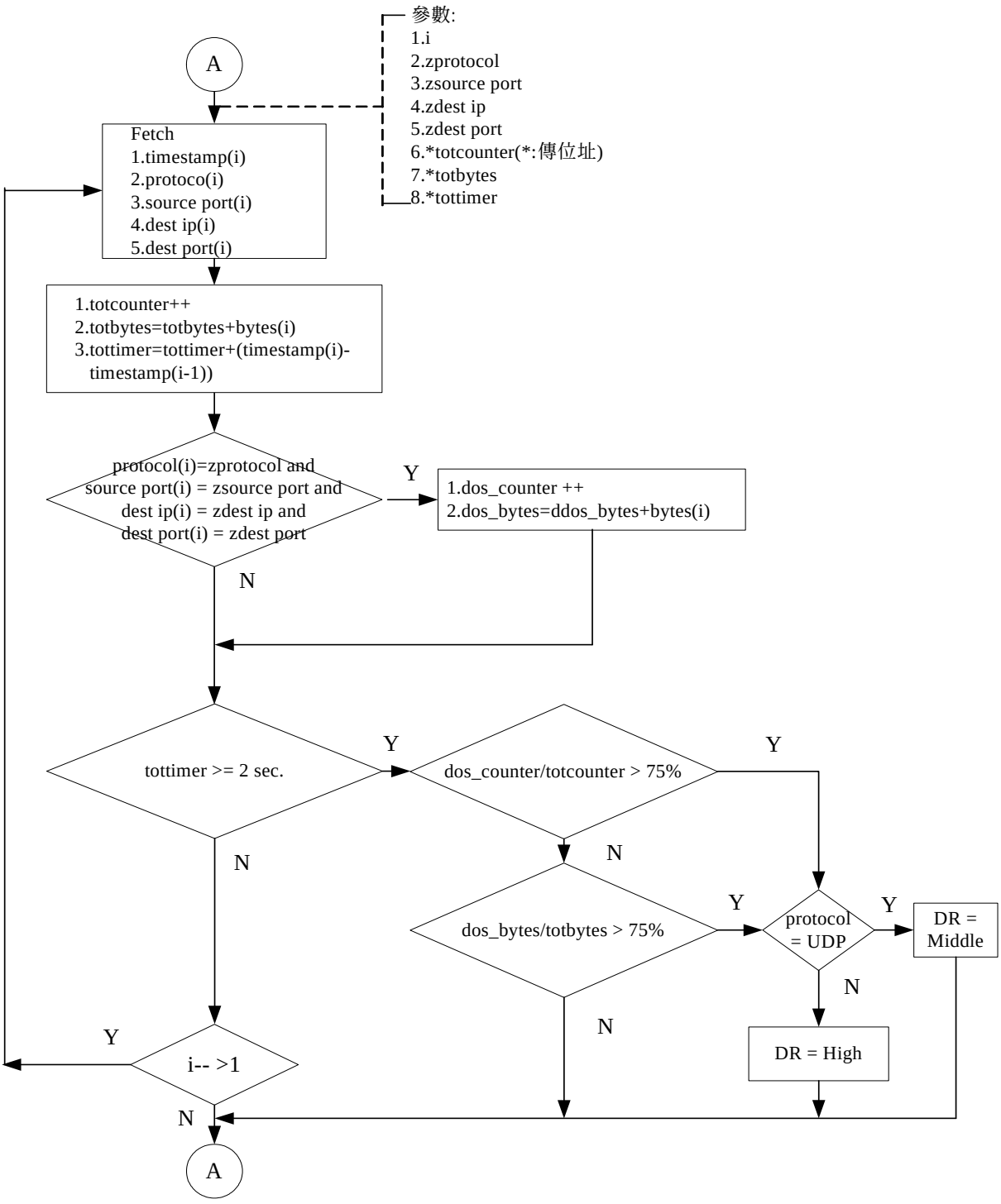

圖 4-11　ID 針對 OutDQ 的 check previous PRs 模組

4.3. 資料暫存區與資料庫

本節將描述輔助偵測器及分析器如何判斷入侵封包，並表列資料庫綱要之設計。

4.3.1. AD 偵測流程

表 4-4 所示為 AD 針對 OBTA 的偵測及其規則。由 NMU 內部對外發出的 DDoS 攻擊封包係分散在若干 OutDQ 中，ID 無法即時偵測出來，故在彙集對外封包之 OBTA 來偵測。

表 4-4 AD 針對 OBTA 的偵測規則

編號	對象	攻擊模式	偵測規則
J	OBTA	DDoS(1, 2, 5)	source port# destination IP、port# protocol

OBTA 集合了所有 NMU 對外的封包，所以可以在此進行由網域內對外的 DDoS 攻擊之偵測。AD 對 OBTA 的偵測流程與 ID 對 OutDQ 中偵測 DoS 攻擊的方式相同亦是採取 Front-Rear 模組及 check previous PRs 模組流程。步驟是：當新封包進入 OBTA 後，立即驅動 AD，接下來流程同 ID 針對 InDQ 之 Ratio check 偵測流程(圖 4-8)，取出 Buffer Z 的 timestamp、protocol、source port、dest ip、dest port 及累計計數參數（counter、bytes 及 timer），check previous PRs 模組則原先的 ID 針對 OutDQ 之 Ratio check 的 check previous PRs 模組圖 4-11 改成圖 4-12 之流程。

圖 4-12 之流程與圖 4-11 之流程相似，唯此處乃判斷 DDoS 攻擊之模式，所累計的是 ddoscounter 及 ddosbytes，再依其所佔比率來判斷是否為攻擊行為。

當某一 OutDQ 佇列填滿時，則 OBTA 必須不斷接收從 OutDQ 一個個移過來的封包，此時，OBTA 的封包到達速率與 OutDQ 相同，但是，當愈來愈多的 OutDQ 陸續填滿後，OBTA 的封包到達速率將快速成長，且其所需容納的資料亦愈來愈多，勢必也會影響 AD 偵測的速率。在最嚴重時，其偵測速度必須是 ID 之 K 倍，K 為 OutDQ 之數量。否則，當 OBTA 的封包到達速率大於 AD 的偵測速率時，將會造成封包的遺失。

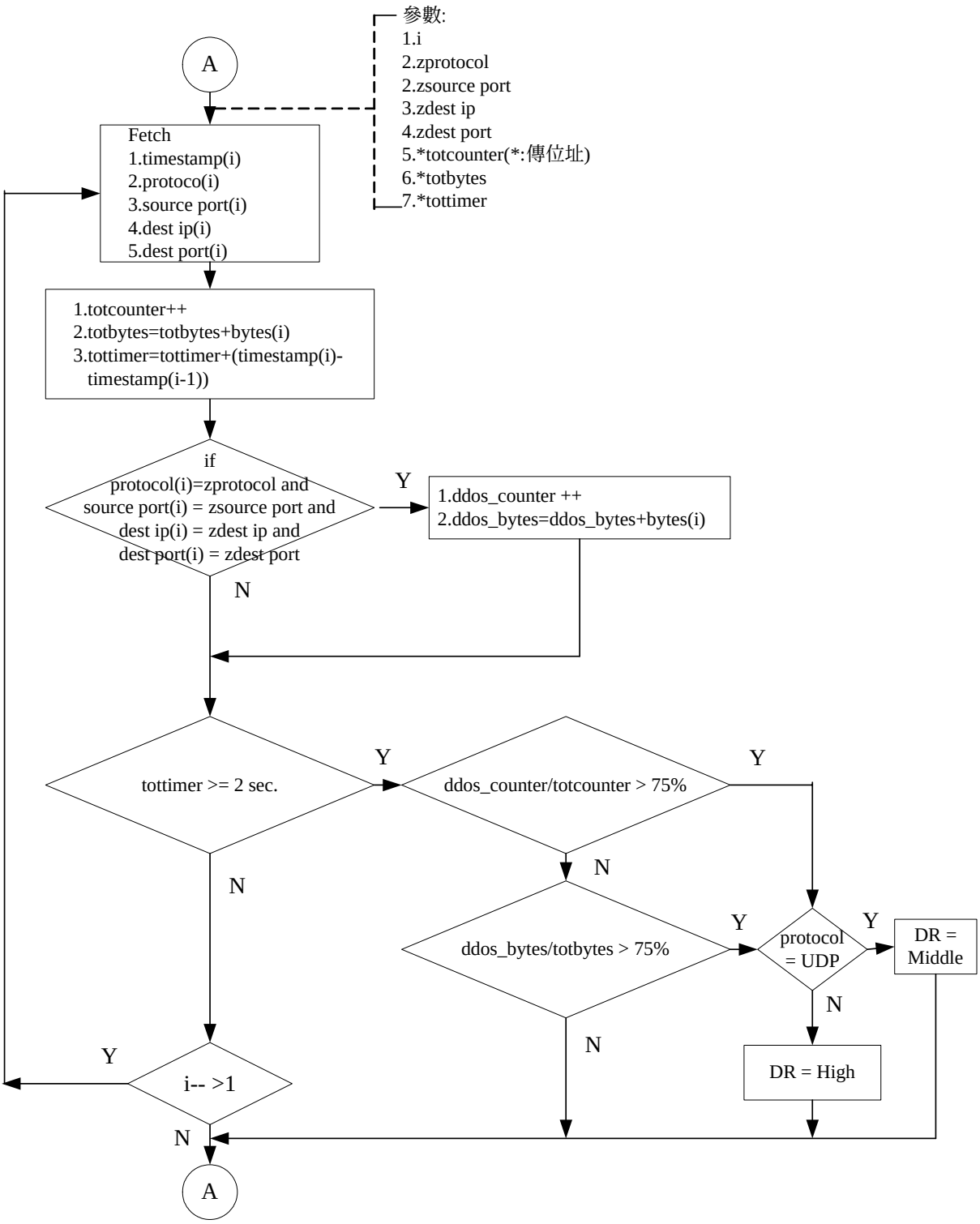

圖 4-12　AD 的 check previous PRs 模組

4.3.2.　封包資料庫設計

在資料庫之前的資料緩衝區分為 IBTA、OBTA 及 FTA，而資料庫中亦以之設計其

綱要，表格名稱各自為 INTB、OUTTB 及 FTB，其細部綱要敘述如下。

1.INTB：為對內傳輸封包資料檔，存放 NMU 網域外部對 NMU 內部 IP 的傳輸封包資料，因其目的 MAC 位址皆為 NMU 之 BR，故不儲存，共包含：Digests、Source_mac、Tlen 等 15 個屬性，其關聯綱要定義如表 4-5。

表 4-5　對內傳輸封包資料檔之關聯綱要

序號	欄位名稱	描述	型態	長度	鍵值
01	Digests	HP 所產生的 Hash 值	char	32	PK
02	Source_mac	來源 MAC 位址	char	12	
03	Tlen	封包長度	char	4	
04	Ident	IP 封包編號（Identification）	char	4	
05	Fragment	IP 標頭之 flag 及 fragment	char	4	
06	Protocol	封包類別（01 ICMP, 06 TCP, 17 UDP）	char	2	
07	Source_ip	來源 IP 位址	char	8	
08	Dest_ip	目的 IP 位址	char	8	
09	Type	ICMP 標頭的 Type 欄位	char	2	
10	Code	ICMP 標頭的 Code 欄位	char	2	
11	Source_port	來源 port	char	4	
12	Dest_port	目的 port	char	4	
13	SYN	Synchronizing 號碼	char	8	
14	ACK	Acknowledge 號碼	char	8	
15	Timestamp	時間戳記	datetime	12	PK

2.OUTTB：為對外傳輸封包資料檔，存放 NMU 網域內部 IP 對外的傳輸封包資料，因其來源 MAC 位址皆指 NMU 之 BR，故不儲存，共包含：Digests、Destination_mac、Tlen 等 15 個屬性，其關聯綱要定義如表 4-6。

表 4-6　對外傳輸封包資料檔之關聯綱要

序號	欄位名稱	描述	型態	長度	鍵值
01	Digests	HP 所產生的 Hash 值	char	32	PK
02	Dest_mac	目的 MAC 位址	char	12	
03	Tlen	封包長度	char	4	

04	Ident	IP 封包編號	char	4	
05	Fragment	IP 標頭之 flag 及 fragment	char	4	
06	Protocol	封包類別（01 ICMP, 06 TCP, 17 UDP）	char	2	
07	Source_ip	來源 IP 位址	char	8	
08	Dest_ip	目的 IP 位址	char	8	
09	Type	ICMP 標頭的 Type 欄位	char	2	
10	Code	ICMP 標頭的 Code 欄位	char	2	
11	Source_port	來源 port	char	4	
12	Dest_port	目的 port	char	4	
13	SYN	Synchronizing 號碼	char	8	
14	ACK	Acknowledge 號碼	char	8	
15	Timestamp	時間戳記	datetime	12	PK

3.FTB：為外部轉送封包資料檔，存放經 NMU 網域 BR 轉送的傳輸封包資料，僅紀錄 BR 收到時的封包(Destination MAC address 為 BR 之位址),轉送出去的封包資料(Source MAC address 為 BR 之位址）不再紀錄；欄位包含：Digests、Dest_mac、Source_ip 及 Dest_ip 等 4 個屬性，其關聯網要定義於表 4-7。

表 4-7　外部轉送封包資料檔之關聯網要

序號	欄位名稱	描述	型態	長度	鍵值
01	Digests	HP 所產生的 Hash 值	char	32	PK
02	Dest_mac	來源 MAC 位址	char	12	
03	Source_ip	來源 IP 位址	char	8	
04	Dest_ip	目的 IP 位址	char	8	

當 TM 執行追蹤封包來源時,可直接查詢其管轄之資料庫中存放的各類封包資料檔。

4.3.3. Analyzer 偵測

Analyzer 對 DB 的偵測模式及其規則，見表 4-5。它是利用封包之間的關聯性來判斷是否為攻擊行為。

表 4-8　Analyzer 對 DB 的偵測規則

編號	對象	攻擊模式	偵測規則
K		Half-opened Connection(6)	source IP destination IP 在 75 秒內, SYN 沒有對應的 ACK(SYN+2) 封包數>5
L	DB	Teardrop(8)	Flag.b1=0 identification Offset 循序計數 fragment#1+total length(1)=fragment#2 fragment#2+total length(2)=fragment#3 …..
M		Trojan or Stepping stone attack(11)	Trojan 常用的 port#（如附錄 B）
N		IIS 網頁伺服器漏洞(9)	Attack command rule(SATAN 81 rules)

根據 TCP 協定 Three-way handshaking 的連線程序，找尋對應的 SYN 及 ACK 序號，如表 4-8 編號 K 的攻擊規則，通常連線建立計時器逾時時間一般最短是 75 秒，最長可能到 23 分鐘，而等候佇列則視各 O.S.或系統管理者設定而有所不同，其值可為 5、32、64，最大為 127，故取其逾時時間最短及等候連線最小來判別此類攻擊行為。

Teardrop 攻擊是製造數個 offset 不合理的小片斷，故需在資料庫中找出相同辨識編號（Identification）者，將這些封包依 offset 值排序後，判斷其 offset 是否重疊或不連續。

通常跳板攻擊是利用植入木馬程式為前置動作，所以若能有效判斷出是否遭到木馬攻擊，就能防止主機成為“跳板”。

另外，節錄 SATAN（掃瞄主機漏洞之軟體）掃瞄 IIS 網頁伺服器漏洞所採用的 81 條規則，將所蒐集的封包，找出相同的辨識編號視為同一封包資料，將這些辨識編號相同的封包資料欄位組合起來與規則比對，藉以判別是否為攻擊行為。

此四類攻擊模式偵測方法是在資料庫中分別對 INTB 及 OUTTB 建立 Trigger（觸動器）程式，當資料自 IBTA 及 OBTA 整批寫入時，即驅動 Trigger 程式。其中，針對所欲寫入之紀錄逐筆與表格內已存在資料相比對，比對規則如表 4-8 編號 L，即可偵測 Teardrop 攻擊模式。至於 Trojain 之類之攻擊模式，則是將已蒐集其常用之通訊埠，預先在資料庫中建立表格，亦是在 Trigger 程式裡，進行關聯比對，藉此判斷攻擊行為。

第5章 實驗與分析

由於本系統未接收實驗室外部駭客之攻擊封包，僅於實驗室內部進行模擬攻擊，所蒐集到之攻擊程式也有限，但是，仍可以就網路流量及傳輸資料量之變化，分析其意義。

5.1. 實驗過程與方法

本實驗之過程與方法如圖 5-1 所示。首先建立兩組子網路環境，將於 5.2 節詳細介紹實驗架構，利用 Sniffer 軟體蒐集兩子網路流進／流出之封包，設定該軟體僅採集 UDP、TCP 及 ICMP 三種通訊協定封包，蒐集 120 秒平均網路封包數量及 Bytes 量，觀察進行攻擊時，網路各通訊協定的封包數量及 Bytes 數的變化。

其次，是蒐集攻擊程式，有一些網站可以下載駭客程式，例如：fetag.org[30]，即是一個提供攻擊程式、防護程式及程式破解的交流站。我們以 DoS／DDoS 攻擊及非 DoS 攻擊分別實驗，並分析其結果。

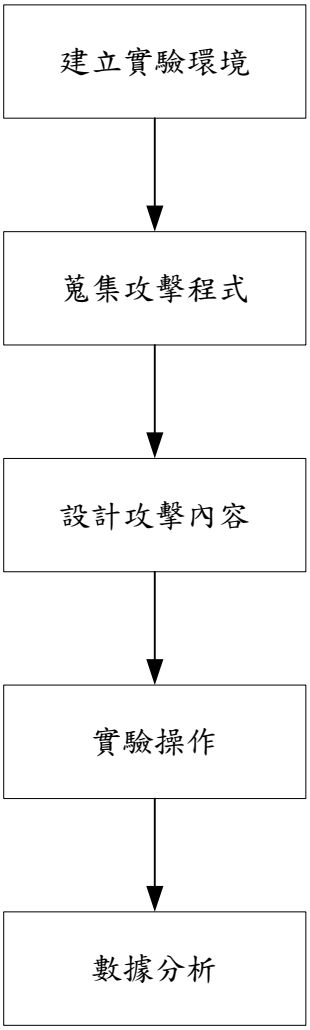

圖 5-1　實驗過程與方法

5.2. 實驗架構與攻擊模式

實驗架構見圖 5-2，其中，A 和 B 為外部機器，C 和 D 為 NMU 內部機器，C 必須設置兩個網路介面，形成 DHCP 架構，D 則是一個虛擬 IP，而 A 與 B 皆無法對其進行攻擊，只能攻擊 C。機器 C 中裝設 HP 元件，具有 Sniffer 蒐集封包與產生 PR 之功能，C 則將所蒐集的 PR 傳送至後端 MIDS，進行偵測。

至於由內對外之 DDoS 攻擊可以架設一配備兩張網路卡的 Linux 主機，作為路由器使用，連接兩邊子網路，在其裝設 HP 元件，蒐集兩端子網路之封包後，傳給 MIDS 系統。表 5-1 為所蒐集與實驗所用之攻擊程式及其描述。

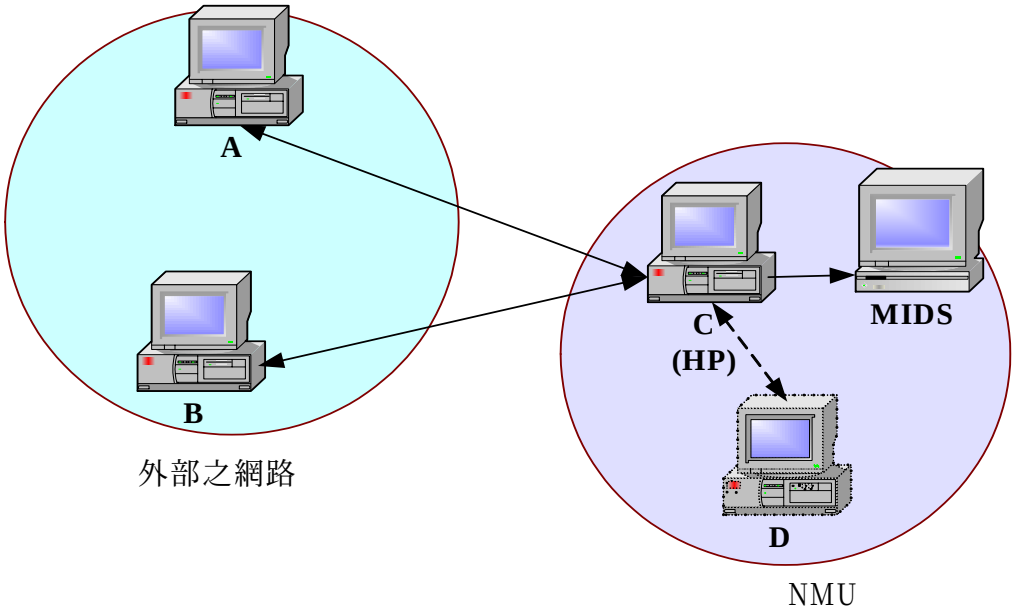

圖 5-2　本實驗架構

表 5-1 實驗所用之攻擊程式及其描述

攻　擊　程　式	攻　　擊　　描　　述
Icmp.exe	設定被攻擊主機、封包大小及發出封包之間隔時間，即可發動 ICMP flood 或 Ping of death 攻擊。
Xicmp.exe	設定被攻擊主機、封包大小及傳送數量，即可發動 ICMP flood、Ping of death。

本研究利用此兩支攻擊程式產生 ICMP Flood 攻擊模式項目，進行實驗，包括（根據圖 5-2）：

1. 由外對內進行 DoS/DDoS 攻擊：由 A 與 B 發出對 C 的攻擊封包。
2. 由外對內進行單一封包攻擊：由 A 或 B 發出對 C 的攻擊封包。
3. 由內對外進行 DoS 攻擊：由 C 或 D 發出對 A 或 B 的攻擊封包。
4. 由內對外進行單一封包攻擊：由 C 或 D 發出對 A 或 B 的攻擊封包。

5.3. 實驗操作與分析

分別實驗 DoS/DDoS 攻擊與非 DoS/DDoS 攻擊兩類，於下分述之。

5.3.1. DoS／DDoS 攻擊實驗

以 ICMP flood 之 DoS/DDoS 攻擊分別進行三項實驗，包括：由外對內之 ICMP Flood（DoS）攻擊、由內對外之 ICMP Flood（DoS）攻擊及由外對內之 ICMP Flood（DDoS）攻擊，將執行十次之統計平均封包個數及平均封包 Bytes 數表列比較之，並說明偵測結果。

1.由外對內之 ICMP Flood（DoS）攻擊：由於該攻擊程式會發出 ICMP echo request 封包，而內部受害機器會因而對外產生 ICMP echo reply 封包。而本實驗連續發動 120 秒攻擊，期間每 5 ms 發送一次，並在此期間蒐集發出及回覆之封包。利用 Sniffer 軟體蒐集封包個數及 Bytes 數，統計如表 5-2 及表 5-3。

表 5-2 由外對內進行 ICMP Flood（DoS）攻擊所蒐集流進 NMU 的封包平均個數及 Bytes 數與其所佔比例

	封包個數	個數比例	Bytes 數	Bytes 比例
UDP	0	0	9	0
ICMP	5746	0.999	735462	0.999
TCP	5	0.001	543	0.001

表 5-3 由外對內進行 ICMP Flood（DoS）攻擊所蒐集流進 NMU 的封包平均個數及 Bytes 數與其所佔比例

	封包個數	個數比例	Bytes 數	Bytes 比例
UDP	1	0	139	0
ICMP	5794	0.999	741568	0.995
TCP	4	0.001	3483	0.005

MIDS 偵測結果經統計，在 InDQ 中平均偵測出 DoS 攻擊有 5537 次，偵測率為 96.36%（5537/5746），但同時在 OutDQ 中平均偵測出 5579 次 DoS 攻擊，後者屬於前者的回應封包，對攻擊端亦是一種 DoS 攻擊，其偵測率為 96.29%（5579/5746）。圖 5-3 為偵測結果之局部畫面。

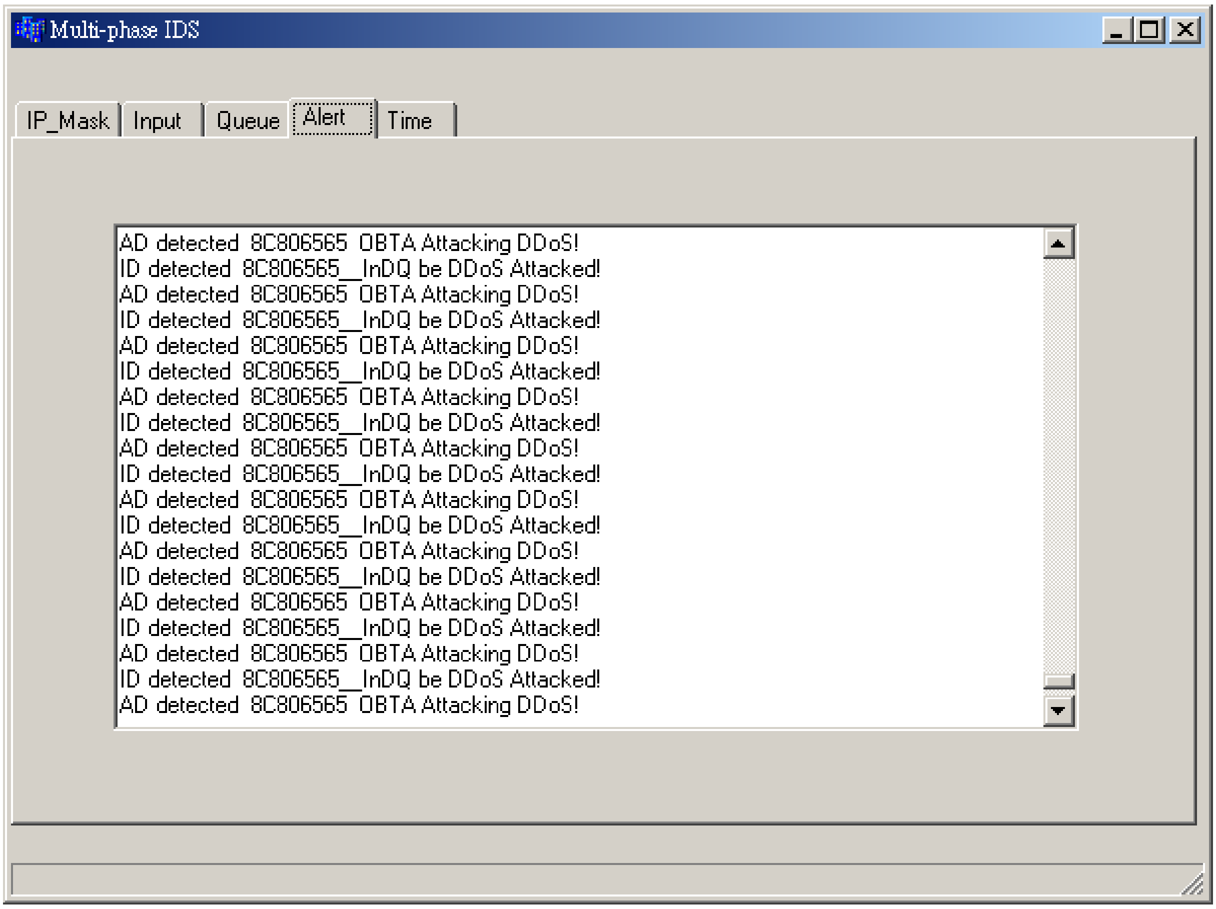

圖 5-3　由外對內之 ICMP Flood（DoS）攻擊之偵測結果（局部）

2.由內對外之 ICMP Flood（DoS）攻擊：由於該攻擊程式會發出 ICMP echo request 封包，而外部受害機器會因而對內產生 ICMP echo reply 封包。而本實驗連續發動 120 秒攻擊，期間每 5 ms 發送一次，並在此期間蒐集發出及回覆之封包。利用 Sniffer 軟體蒐集封包個數及 Bytes 數，統計如表 5-4 及表 5-5。

表 5-4　由內對外進行 ICMP Flood(DoS)攻擊所蒐集流進 NMU 的封包平均個數及 Bytes 數與其所佔比例

	封包個數	個數比例	Bytes 數	Bytes 比例
UDP	0	0	0	0
ICMP	2552	0.999	326605	1
TCP	2	0.001	153	0

表 5-5　由內對外進行 ICMP Flood(DoS)攻擊所蒐集流出 NMU 的封包平均個數及 Bytes 數與其所佔比例

	封包個數	個數比例	Bytes 數	Bytes 比例

UDP	1	0	109	0
ICMP	2544	0.999	325594	0.999
TCP	1	0	326	0.001

由 MIDS 得出，在 OutDQ 中平均偵測到 DoS 攻擊有 2436 次，偵測率為 95.75%（2436/2544），但同時在 InDQ 中平均偵測出 2449 次 DoS 攻擊，對攻擊端亦是一種 DoS 攻擊，其後者屬於前者的回應封包，偵測率為 95.96%（2449/2552）。圖 5-4 為偵測結果之局部畫面。

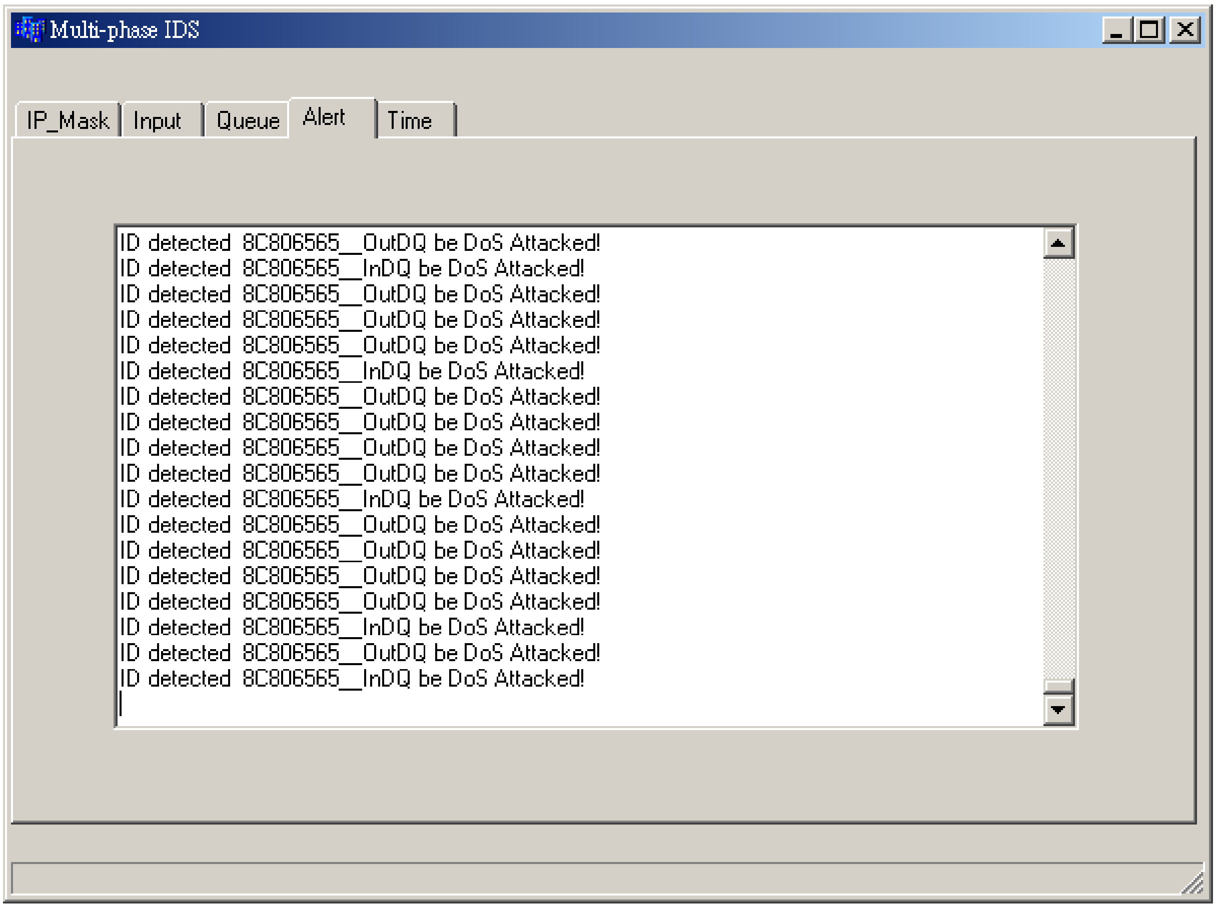

圖 5-4　由內對外之 ICMP Flood（DoS）攻擊之偵測結果（局部）

3.由外對內之 ICMP Flood（DDoS）攻擊：由於該攻擊程式會發出 ICMP echo request 封包，而內部受害機器會因而對外產生 ICMP echo reply 封包。而本實驗由 A、B 同時連續發動 120 秒攻擊，期間每 5 ms 發送一次，並在此期間蒐集發出及回覆之封包。利用 Sniffer 軟體蒐集封包個數及 Bytes 數，統計如表 5-6 及表 5-7。

表 5-6　由外對內進行 ICMP Flood（DDoS）攻擊所蒐集流進 NMU 的封包平均個數及 Bytes 數與其所佔比例

	封包個數	個數比例	Bytes 數	Bytes 比例

UDP	0	0	0	0
ICMP	14316	1	1832393	0.999
TCP	4	0	1015	0.001

表 5-7　由外對內進行 ICMP Flood（DDoS）攻擊所蒐集流出 NMU 的封包平均個數及 Bytes 數與其所佔比例

	封包個數	個數比例	Bytes 數	Bytes 比例
UDP	1	0	176	0
ICMP	14435	1	1847680	1
TCP	4	0	645	0

MIDS 偵測結果經統計，在 InDQ 中平均偵測出 DDoS 攻擊有 14094 次，偵測率為 98.45%（14094/14316），但同時在 OBTA 中平均偵測出 14219 次 DDoS 攻擊，後者屬於前者的回應封包，對攻擊端亦是一種 DoS 攻擊，其偵測率為 98.5%（14219/14435）。圖 5-5 為偵測結果之局部畫面。

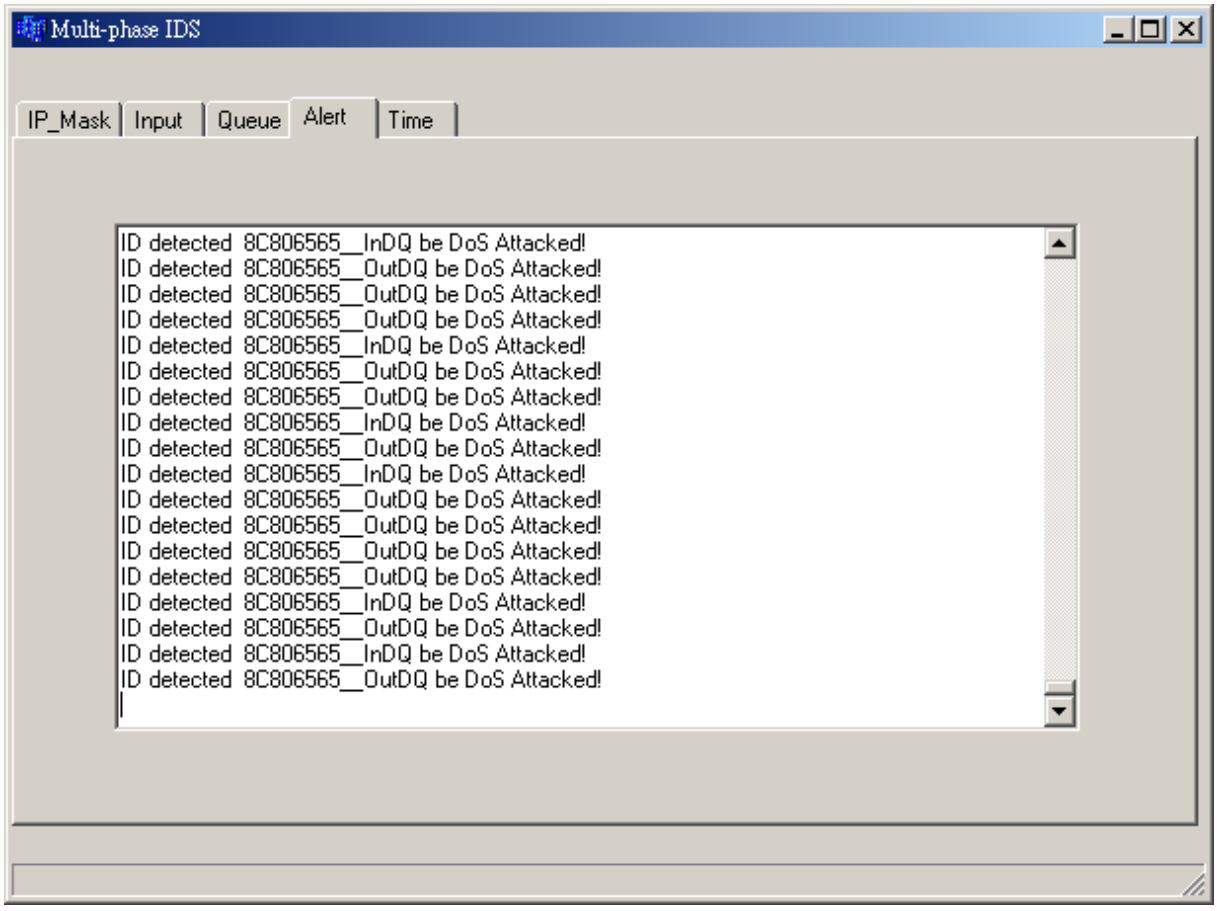

圖 5-5　由外對內之 ICMP Flood（DDoS）攻擊之偵測結果（局部）

5.3.2. 非 DoS／DDoS 攻擊實驗

對於非 DoS/DDoS 攻擊模式，共執行 Ping of death 及 IIS 網頁伺服器 Unicode 漏洞攻擊。

1.Ping of death 攻擊

利用攻擊程式 Xicmp.exe，由外對內之 ping of death 攻擊，封包大小本實驗設定為 500000 bytes，在兩分鐘內共發出 10 個封包，由於該攻擊程式會發出 ICMP echo request 封包，而內部受害機器會因而對外產生 ICMP echo reply 封包。經 MIDS 偵測，在 InDQ 發現 2 個 Ping of death 攻擊，亦在 OutDQ 發現 2 個 Ping of death 攻擊。由於受害機器（PIII 500 + Windows 2000 Professional 繁體中文版）呈現暫時當機狀態，停頓些許時間才恢復正常服務，後續的封包因此無法接收到，故僅就其收到的 2 個攻擊封包偵測出來。如圖 5-6。

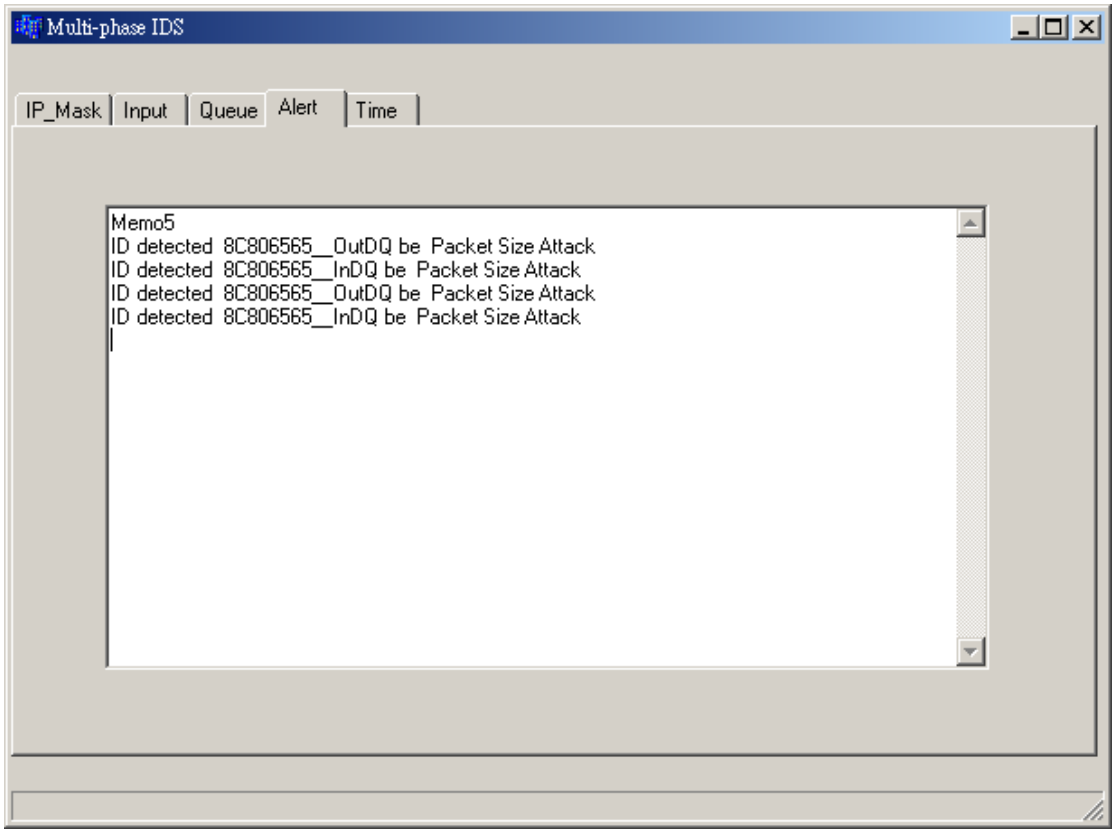

圖 5-6 由外對內之 ping of death 攻擊之偵測結果

2.IIS 網頁伺服器 Unicode 漏洞攻擊[14]

瀏覽器在(PIII 500 + Windows 2000 繁體中文版 + IIS 5.0 + SP1)平台環境下，輸入URL:http://127.0.0.1/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir，IIS 回傳找不到該畫面，如圖 5-7 所示。因該輸入到達了伺服器，所以在 IIS 伺服器的系統日誌檔C:\WINNT\system32\LogFiles\W3SVC1\ex030720.txt 中，可以發現該筆連線紀錄，如圖5-8。

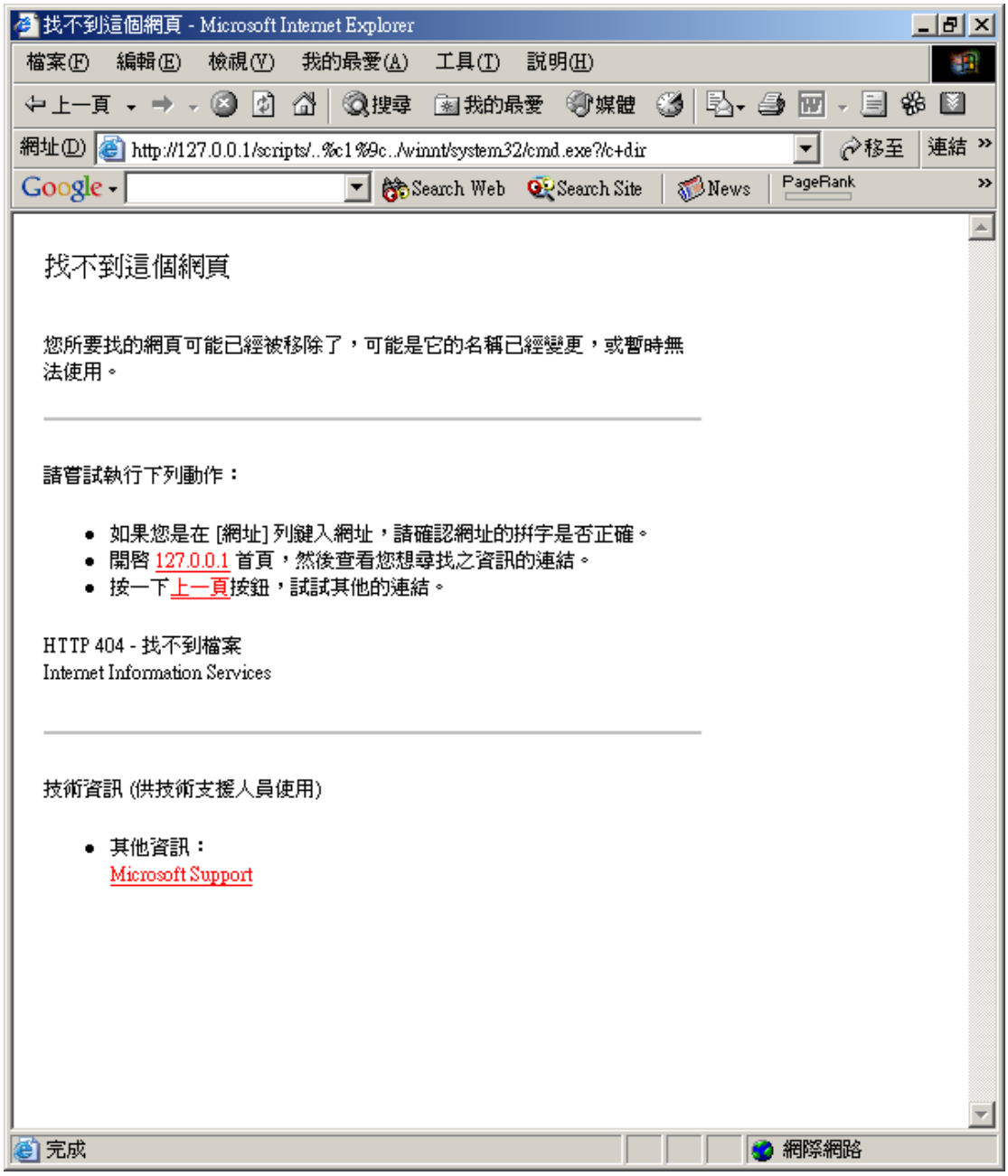

圖 5-7 輸入 URL: http://127.0.0.1/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir 之回傳畫面

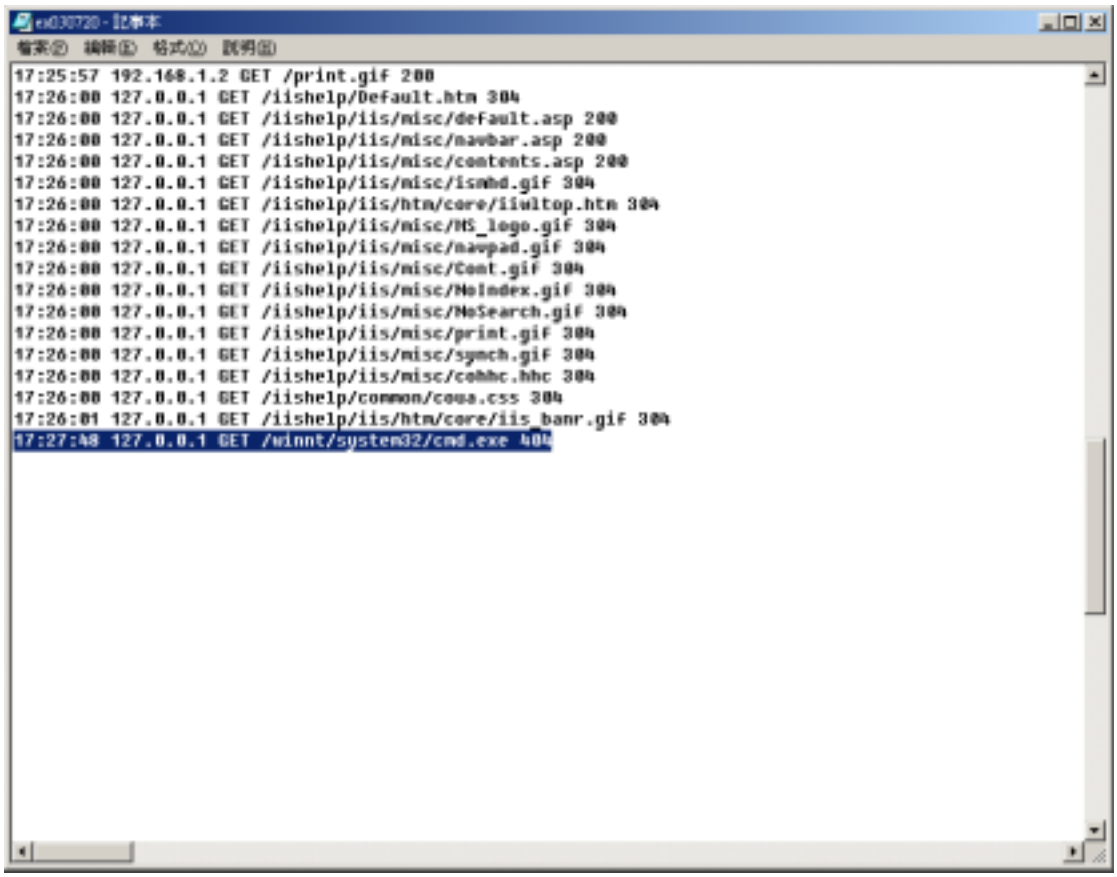

圖 5-8　在執行 IIS 網頁伺服器 Unicode 漏洞攻擊後之 IIS 伺服器系統日誌檔

現有許多個人免費的防護軟體皆可以阻擋這類攻擊，以 SecureIIS.exe 程式為例，可以設定哪些列為危險的指令及欲偵測的關鍵字，如圖 5-9 及圖 5-10 所示，列於圖 5-9 中之指令及圖 5-10 中之輸入，只要在輸入字串中發現則視為攻擊行為，而加以紀錄，如圖 5-11 所示，即是被攻擊後被偵測出來的畫面，此時網頁就會出現訊息，如圖 5-12，警告攻擊者。

這類攻擊指令的防護，技術上並不困難，我們的理論可以藉此程式證明是可行的，MIDS 系統亦可以利用這類防護程式，與其他類偵測工具平行地偵測攻擊行為。

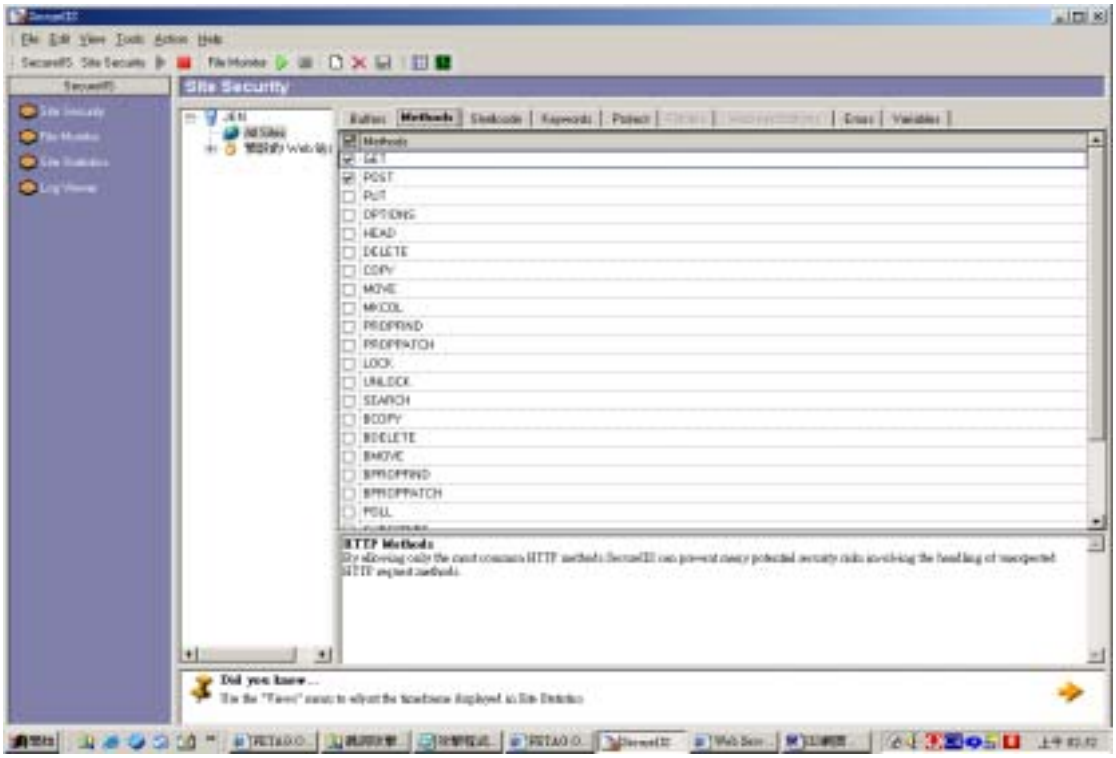

圖 5-9　SecureIIS 程式設定危險指令

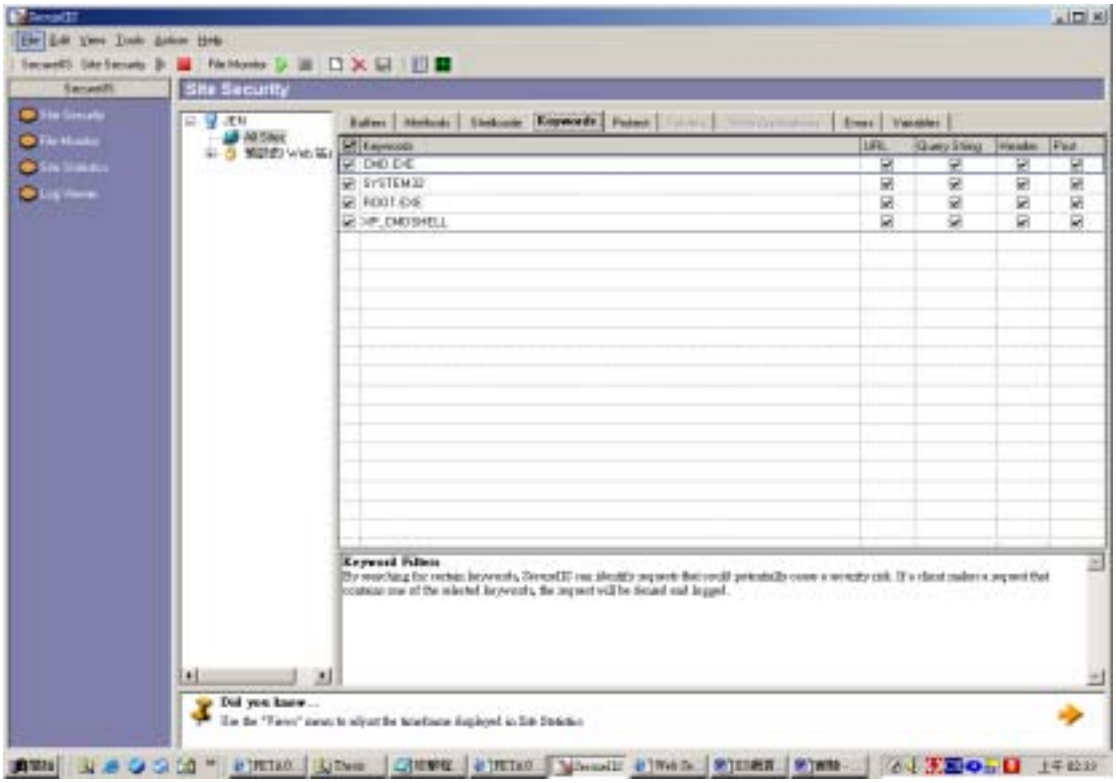

圖 5-10　SecureIIS 程式設定欲偵測的關鍵字

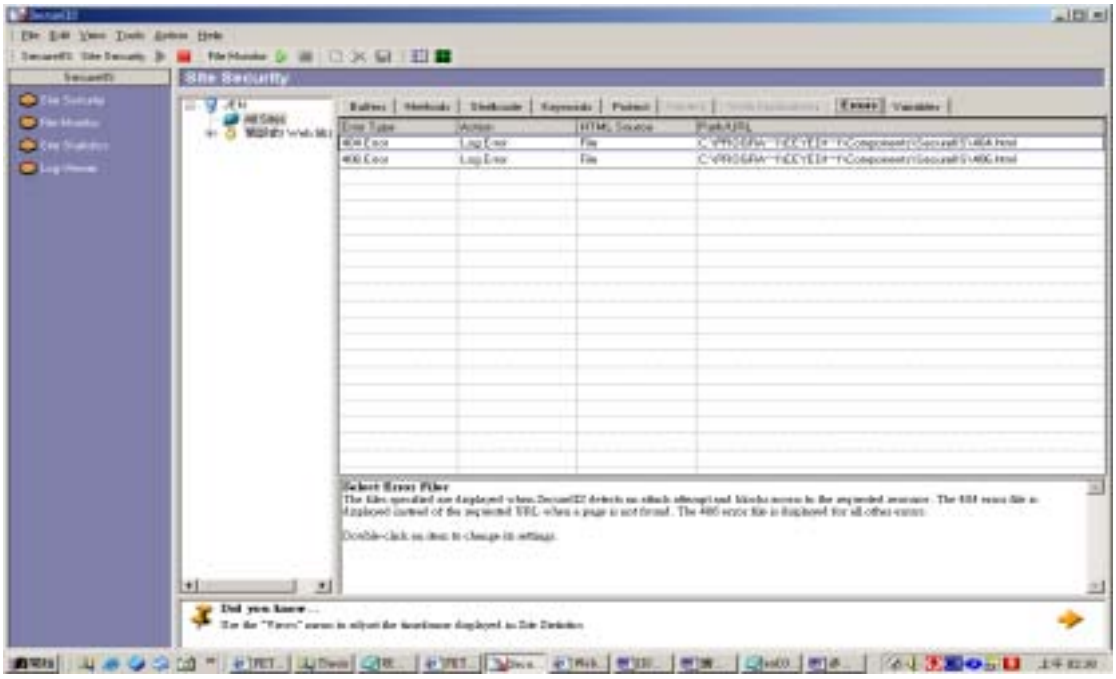

圖 5-11　SecureIIS 偵測出攻擊行為

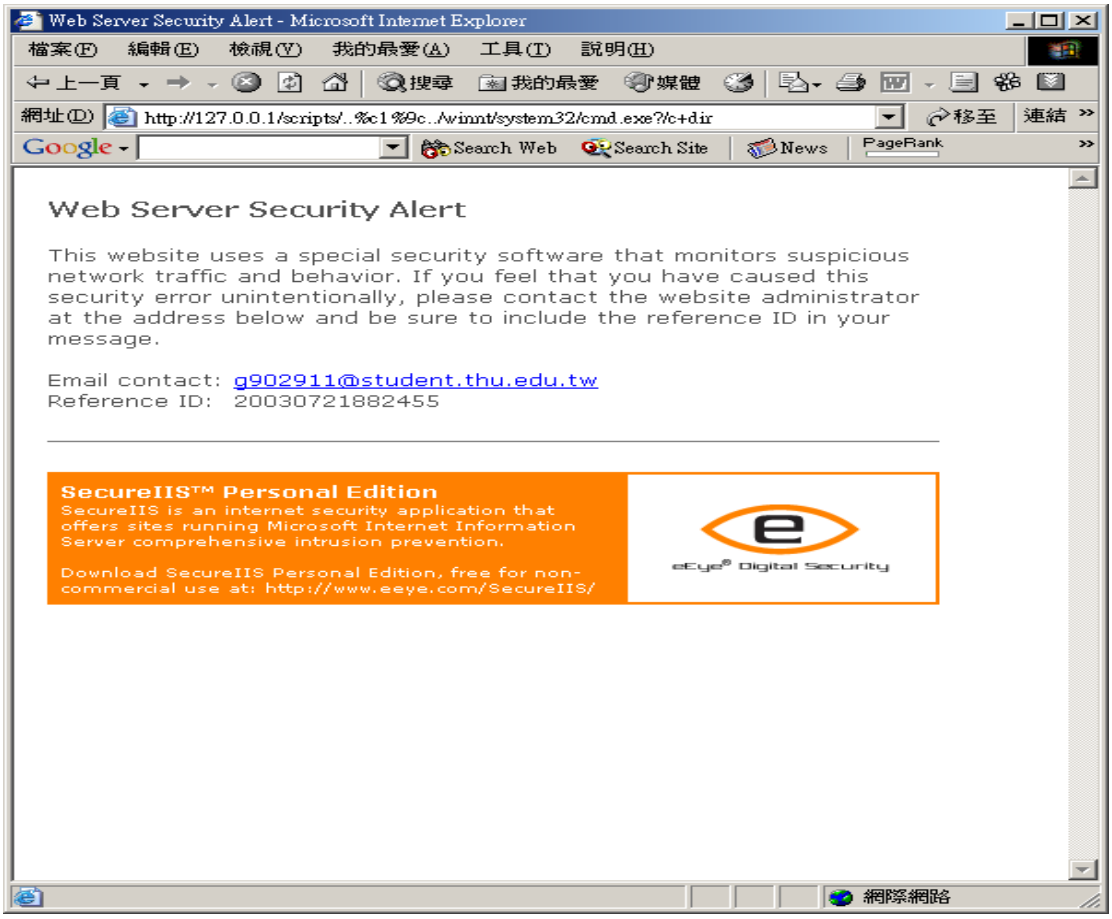

圖 5-12　網頁警告訊息

5.4. 實驗結論

常見的 NIDS 大都是以流量的變化為偵測的依據，我們的實驗亦是在單位時間內，依相同協定封包（UDP、TCP 及 ICMP）個數所占所有個數的比例，判斷其為一攻擊行為，但我們另外加入一個要件：Bytes；這是一個重要的觀念，若 DoS／DDoS 攻擊，TCP 封包最大 1500Bytes，10000 個 TCP 封包同時流進網域內時，此時在網路就有 15MB 的資料，同樣也可以暫時阻斷網路服務（假設頻寬僅 10 MB），但其封包數可能並未達到 DoS／DDoS 攻擊的比例。

將 5.3.1 節 DoS/DDoS 攻擊實驗，各執行 10 次後，取得 120 秒內遭受 ICMP flood 攻擊時的流進／流出之封包個數及 Bytes 的平均值，很明顯地，ICMP 封包數量及 Bytes 數皆佔 UDP、ICMP 及 TCP 三種通訊協定總合之最大量，故在第四章中，我們所引用的理論[4]而設計的系統是可以偵測出 DoS／DDoS 攻擊。

此外，針對 Ping of death 及 IIS 網頁伺服器 Unicode 漏洞攻擊，實驗中亦可以本系統或利用現有的免費軟體，達到我們所設計的偵測規則，也足以證明本系統之可行性。

第6章 UDIDT 與 MIDS 的防禦策略及限制

前兩章中設計以 UDIDT 與 MIDS 從事入侵偵測與追蹤，但是，此系統是否可能遭到攻擊？有哪些攻擊模式是 MIDS 無法偵測出來的呢？本章將逐一探討，亦介紹 UDIDT 的防禦措施，並預期未來可能發展的新攻擊模式及本系統之因應方法。

6.1. TM 的防禦策略

由於 LM 對於封包的處理是以複製，而非以 store and forward 方式，將封包導引至 MIDS。當 MIDS 偵測出攻擊行為，該封包也可能早已攻擊了目標主機；因此，TM 除了主動追蹤入侵來源，亦需搭配防禦策略，本論文提出之策略如下：

1.當偵測出 DoS/DDoS 攻擊封包，MIDS 即刻發出 Trace Request from MIDS 訊息通知 TM，追蹤其來源位址，再由網管人員以人工方式在 BR 內部建立黑名單管制清冊，以阻斷該來源位址之連線，保護 NMU 網域及內部主機免於繼續受到此類攻擊。人為地建立清冊的原因是希望由網管人員確定是否為一真實的攻擊或其實是一誤判。

2. MIDS 偵測出單一封包之類的攻擊模式時，須經 TM 人工介面反應給網管人員，俾通知該被害主機，令其盡速實行補救措施，例如，自我檢查是否已被植入木馬程式、是否有重要檔案內容被非法地修改過等。

一方面，MIDS 的設計原則是：一旦偵測出攻擊封包即刻通知 TM 進行追蹤。駭客卻可利用此一特性，故意發出連續性的攻擊封包，使 TM 機制不斷進行追蹤步驟，疲於奔命，進而癱瘓 TM，使其暫時失去追蹤的功能。

在這方面 UDIDT 的預防策略如下：

TM 設置一緩衝區，當 MIDS 偵測出攻擊封包後，立即傳送訊息給 TM，但 TM 將資料保持在緩衝區內一段時間，例如：每 5 秒鐘，在此時間內，若有重複之追蹤需求則不理會。

6.2. MIDS 的限制

MIDS 各階段所能偵測的攻擊模式，整理成表 6-1。

表 6-1　MIDS 三階段所偵測之攻擊模式

第一階段： ID(InDQ 及 OutDQ)	第二階段： AD(OBTA)	第三階段： Analyzer(Database)
1.ICMP flood(incoming DoS/DDoS & outgoing DoS) 2.UDP flood(incoming DoS/DDoS & outgoing DoS) 3.ICMP Smurf flood 4.Ping of death 5.Land	1.ICMP flood(outgoing DDoS) 2.UDP flood(outgoing DDoS)	1.TCP SYN flood 2.Teardrop 3.Trojan 4.IIS 網頁伺服器漏洞

MIDS 的規劃設計，期以封包特徵快速地偵測入侵行為，但也有其瓶頸也是未來需要繼續努力的地方：

1.流量太大時，ID 對 DQ 的偵測速率小於網路流量速率時，可能造成封包的遺失，這也是為什麼本系統部份偵測，例如，One-way check，採取平行處理的主要原因之一。

2.若是 DoS／DDoS 攻擊採用不同的攻擊工具時，攻擊封包格式將會發出封包協定、封包大小及來源 IP 不同之攻擊，本系統將難以判斷。

在第二章中所列出的 DoS/DDoS 攻擊模式，大都可以在 MIDS 中偵測出來，根據[20]所整理分類的十種網路犯罪手法及方式中，本系統可以偵測出其中的五項，有木馬程式、系統安全漏洞、緩衝區位溢位、網站入侵與攻擊及阻斷服務等。然而，也有若干攻擊受限於蒐集封包的位置或利用正常封包來進行，是 MIDS 無法偵測的，包括：

1.NMU 內部攻擊

因為 LM 的位置在 BR 的外面，所以內部機器攻擊內部主機的封包只要不經 BR 者，LM 皆無法蒐集到，當然就無法加以偵測了。在這方面擬建議 NMU 的網路管理人員對於內部 IP 的管理，做到實體與邏輯位址（MAC 與 IP 位址）對應鎖定，即，各內部機

器的一個網路介面指定一個 IP 位址，如此，內部駭客就無法使用 IP spoofing 方式攻擊同一網域內之其他主機了。一旦網管人員接受被攻擊申告時，發現攻擊是來自內部，則可以由 ARP Table 直接且正確地找出發動攻擊的機器，使駭客無所遁形。

2.附在電子郵件之攻擊

由於藉電子郵件所進行之攻擊封包都是正常封包，在傳送電子郵件時，難以檢查出其未來會有攻擊行為。這類攻擊可以事先蒐集各種可能的病毒或後門程式之名稱，例如，Nimda 病毒會發出夾帶 readme.exe 之電子郵件，可以之比對封包內容字串來判斷，一經發現，則將封包丟棄。然而，比對內容字串容易出現誤判的情形，而造成正常封包亦遭棄置的情形，且網路傳輸績效將會因字串比對而大受影響。

3.TCP 連線劫持

此類攻擊是發生在共享式媒體的架構上，屬於內部網路攻擊，情況與「NMU 內部攻擊」相同，對其之建議（IP 與 MAC 位址鎖定）亦同。

4.後門漏洞

是利用主機或網路設備的預設帳號或各版本的已知漏洞所進行的攻擊，其攻擊行為都是借用正常的封包來進行，故亦難以偵測出來；系統版本的漏洞改善（patch）及帳號密碼定期改變，都是預防此類攻擊的不二法門。因此，網路管理人員及使用者平日便需加強自身對網路安全的觀念。

5.暴力破解密碼

是以掃瞄系統漏洞或該系統所提供的服務，測試系統之密碼，再行入侵系統，所進行之攻擊封包都是正常封包，。對此項目之攻擊的建議與「後門漏洞」相同，亦是定期變更系統帳號密碼，尤其是密碼的選取是以：特殊、大小寫夾雜、非一般常見者為佳，以增加入侵之困難度。

6.DNS Hijacking

駭客是利用提供 DNS 主機的機構所制定的不當政策，例如：網域名稱代管伺服器接受申請帳號的手續太過簡單，設定的資訊審核未確實，或使用者可以自行修改自己網域名稱（Domain Name）等，這些行為都是採正常的封包傳輸。此類亦是需要網路管理人員加強其網路安全觀念，增加申請及審核之嚴謹性，不讓駭客有機會取得該網域名稱代管伺服器的控制權，否則，將防不勝防。

7.SQL Injection

SQL Injection 是以網站正常輸入的方式，將攻擊命令夾在傳送資料所進行的攻擊，因此，對使用者的輸入未做好妥善的過濾與查驗，則易遭到攻擊，而使資料庫遭受破壞。NMU 亦無法偵測此類攻擊，但是擬在此建議其防範的方法：

(1) 資料庫系統對於網頁輸入的字串建議過濾其長度、型態，以避免駭客輸入特殊字元或字串，而擷取帳號、密碼及資料，或是破壞資料庫。

(2) 若以 client 端的程式（如：java script 或 VB script 等）檢查輸入，使用者可以把網頁程式儲存於其本地（local）端，修改程式碼而略過輸入欄位檢查，即可進行 SQL Injection 攻擊。故應盡量將查核程式置於 server 端或介於 client 與 server 之間的 agent。

(3) 做好所有例外處理，勿讓使用者直接看到系統傳回的錯誤訊息，而且建議取消所有程式開發中錯誤訊息的顯示，而改以顯示錯誤代碼，因為這些都可以成為駭客用來獲取資料庫檔案結構或內容等資訊。

(4) 加強資料庫帳號與權限管理，限制哪些帳號擁有哪些讀寫（select、insert、delete 等）權限，讓網站的使用者不能以系統管理者帳號連結資料庫。

偵測此類攻擊，與「附在電子郵件之攻擊」相同，僅能檢查封包內容是否含有與資料庫指令有關的字串；然而，此舉亦增加其比對時間，降低傳輸績效，若只是單純地檢驗傳輸內容中是否含有此類字串便判斷是一項攻擊，反而增加系統誤判的機率。其實，最佳的防護方法是系統開發及管理時，投入多一些網路安全的觀念，例如，以上四點建議，即可減少甚至防止此類的攻擊了。

6.3. 偽裝 Router 之攻擊模式

Linux 之風日漸盛行，許多資訊玩家皆以其為作業系統，而在網路上，Linux 主機即具有 Router 的功能，若被有心人士利用，將有可能成為未來的攻擊模式之一。其可能的攻擊方法包括：

1.利用路由資訊協定（Routing Information Protocol，RIP）最大路由值的限制：例如，RIP 最大的路由值為 16，也就是封包 TTL（Time to Life）欄位自 15 起，每經過一 hop 減 1，一旦至 Router R 時，TTL 值等於 0，則此封包將遭丟棄，因係避免太多無法送達的封包在網路間流竄，造成網路阻塞或大量消耗頻寬資源，亦會增加路由器之負擔；而與該封包來源相同的網段位址將一直都不會被加入 R 之路由表中，因為經過 R 的封包 P 在到達該網段之前，TTL 已等於 0，而被棄置，即 P 不可能到達該網段。而 RIP 是唯一可以讓路由器利用 Linux 機器可靠地交換資訊的路由協定，若攻擊者利用 Linux 系統偽裝成 Router，置於一個網路通道，如圖 6-1 中之 D，其路徑度量設為 15，則當 Router A、B、C 在建立路由表時，評估 D 之路徑度量已是 15 了，推算與 D 連接之網段 X（Sub Net X）傳送封包至 D 時，路由值已為 0，故網段 X 將不會被加入路由表中，所以經過 Router A、B、C 的封包皆無法到達網段 X 了。

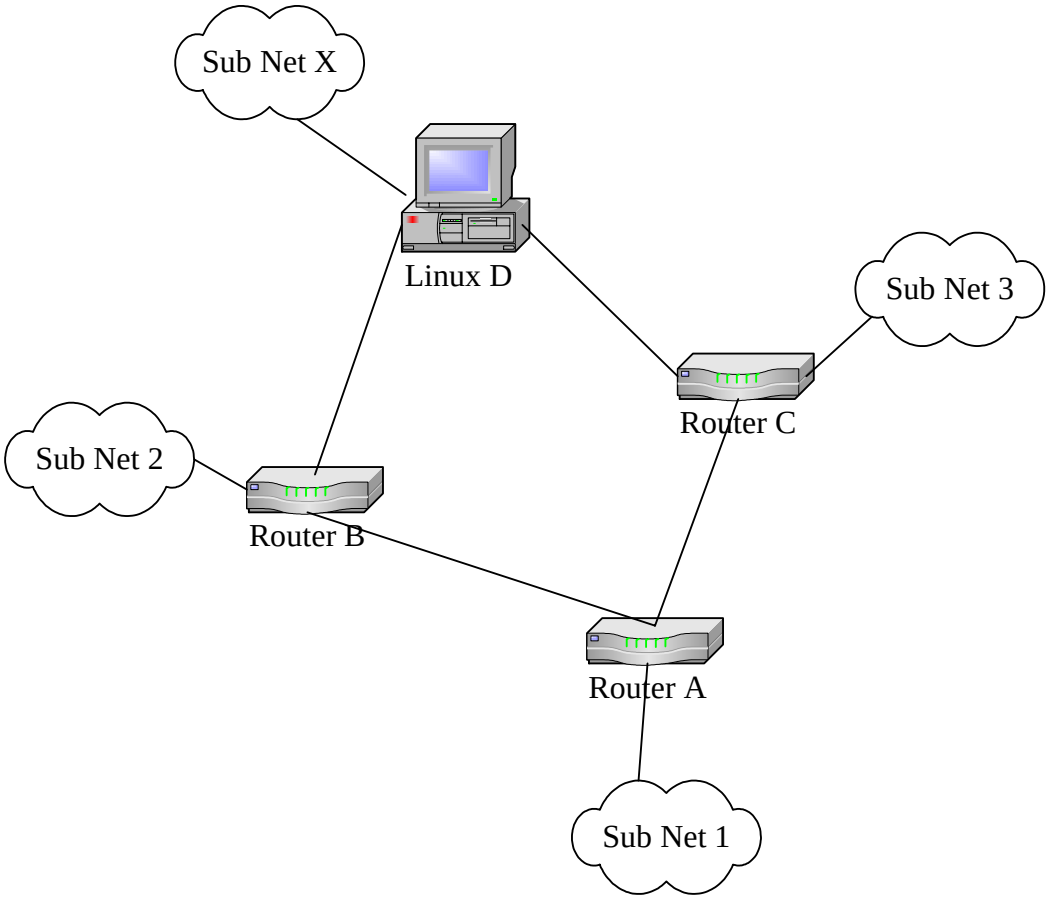

圖 6-1　Linux 系統偽裝為 Router 示意圖

2.通常 RIP 更新路由表時所佔用之頻寬比其他協定還大，因該協定是將整個路由表送送給對方的路由器。通常路由器是透過 IGP（Interior Gatway Protocol）、EGP（Exterior Gateway Protocol）或 BGP（Border Gateway Protocol）確定與之相鄰之路由器的位址後，每 30 秒發送一次 RIP；若以 Linux 偽裝成路由器，密集傳送大量 RIP 給鄰近路由器，就可能塞爆該路由器，而癱瘓其正常功能。

因為 RIP 是藉由 ICMP 封包傳送路由表，而我們已在 MIDS 中檢查 ICMP 封包，故可判斷出上述的第二類的 DoS/DDoS 攻擊；但是第一類者就無法依封包本身來判斷了。然而，駭客欲將偽造主機與各網域路由器相連實屬不易，也有可能利用類似 DNS Hijacking 攻擊模式的方法，對路由器進行攻擊；各區域網路管理人員應善加利用網管工具，監督網路上之異常狀況。

第7章 結論

本研究中，首先歸納常見的攻擊模式、IDS 的種類及幾個重要的入侵追蹤方法，啟發我們對於入侵追蹤的構想，而提出區域聯防入侵偵測與追蹤系統 UDIDT，希望能即時偵測入侵封包，主動追蹤攻擊來源。另外，設計了 MIDS 的架構，是 UDIDT 下的 NIDS，以攻擊封包特性，即時地偵測入侵行為，將偵測出的攻擊行為通知 TM，得以阻絕及追蹤攻擊來源，最後利用關聯式資料庫，提供與支援 TM 在追蹤時所需要的資料，亦利用封包之間的關聯性來判斷是否為攻擊行為。

多階段入侵偵測的設計乃是利用攻擊封包特徵及儲存空間特性；有些攻擊行為可以經由其封包特徵或行為方式得以判斷，有些則無法利用幾個特徵即可辨識出來；故利用記憶體空間小、速度快的特性及資料庫資料容量大、但執行速度較慢的特性，而以記憶體實作佇列結構進行即時性的偵測，以資料庫的關聯性特色，對有相關性、連續性的攻擊封包加以辨識。

以固定攻擊辨識規則來辨別攻擊封包，如節錄 SATAN 81 條規則，必須不斷增加及擴充規則資料庫，才能夠偵測出新的攻擊指令，顯然缺乏系統彈性及擴充性，可以將之改為關鍵字之形式，例如：動作為“GET”，命令為“cmd.exe”等，在封包資料字串中搜尋關鍵字，符合幾個關鍵字的組合即可辨別為攻擊封包，如此便可以增加 IDS 的彈性及擴充性。

「區域聯防」的觀念不僅用於入侵追蹤，亦適用入侵偵測系統，在被害 NMU 端的 MIDS 未能或不易偵測出的攻擊，例如，由內對外 DDoS 攻擊，亦可以由發動攻擊主機的 NMU 偵測發現，而對外發動之單一攻擊，亦可直接由攻擊者所在之 IP 偵測出來，以減少網路安全上之糾紛，是故藉由區域間同心協力地維護網路的安全使用，即為『區域聯防入侵偵測與追蹤系統』之本質。

7.1. 研究貢獻

歸納整理常見的攻擊模式、入侵追蹤方式，分類介紹入侵偵測系統，可以提供後續研究者參考。

以宏觀的角度來思考入侵的防禦工作，設計區域聯防的機制，並非所有的攻擊都以自我區域的防衛來思考，無論是由內部網域發出的攻擊行為，或是由外部對內部的入侵，各區域的偵測機制皆應阻絕其所能發現的攻擊行為，否則，亦可由其他區域的偵測機制共同防護，可謂之「分散式偵測」；除此之外，對於偵測出的攻擊，亦採區域聯合追蹤的方法，有效嚇阻危害網路安全的不肖分子，謂之「分散式追蹤」，以有效嚇阻危

害網路安全的不肖分子。這是當今建構網路安全的重要趨勢之一，也是從事網路安全之學者專家必備的思維與認知。

整理攻擊封包的傳輸特性，分類其特徵，採用記憶體存放封包資訊的方式，達到即時入侵偵測的目標，並以資料庫儲存封包資訊，做為非及時的入侵偵測及事後的追蹤。並依不同的攻擊類型採用不同的偵測方法及儲存體，形成階段式的偵測模型，俾提供較完善的入侵偵測機制。

7.2. 未來研究方向

對於跳板攻擊，目前偵查／追蹤封包的機制尚稱薄弱，未來將以系統日誌紀錄配合法醫鑑識技術，繼續在跳板主機所在的 RS 中找出控制封包的來源，再由該 RS 通知其 TM 向前追蹤，直到找到真正的攻擊主機或真正的攻擊者，俾使整體系統能更加完善。

目前越來越多新發展的攻擊模式，將攻擊程式與病毒程式結合在一起，造就傳播迅速、破壞力強大的攻擊，例如：Code Red（紅色警戒）病毒。2001 年六月微軟的網際網路伺服器軟體被發現有瑕疵，在同年七月「紅色警戒」就利用此瑕疵造成全球各地超過二十二萬五千台電腦遭到感染，特別是使用 Microsoft NT4.0 或 2000 作業系統的電腦 [33]。類似這樣的攻擊將會更嚴厲地衝擊目前網路安全的機制，以本研究的偵測攻擊探討的是以封包特徵為偵測目標，對於合法的傳輸封包並無法建立偵測規則，而紅色警戒即是利用 HTTP 的 80 通訊埠攻擊被害主機。所以對於這類型攻擊不該只是消極的修補系統漏洞，或遭感染後的事後補救，而應該更積極地加以防禦，目前對這類型之攻擊的防禦機制仍在萌芽階段，這必然也是未來網路安全的重要課題之一。

參考文獻

一、西文部份

[1] Bellovin, “ICMP Traceback Messages”, Network Working Group Internet Draft, draft-bellovin-itrace-00.txt.

[2] D. E. Denning, “An Intrusion-Detection Mode,l” IEEE Transactions on software engineering, volume SE-13, NO. 2, pp.222-232, February 1987.

[3] G. Lawton, “Virus Wars: Fewer Attacks, New Threats,” Computer, Technology News, pp. 22-24, December 2002.

[4] W. Lee & S.J. Stolfo & K.W. Mok, “Algorithms for Mining System Audit Data,” DARPA, F30602-96-1-0311.

[5] E. Maiwald, Network Security: A Beginner's Guide, McGraw-Hill Companies Inc., 2001.

[6] S. Savage & D. Wetherall & A. Karlin, & T. Anderson, “Network Support for IP Traceback,” IEEE/ACM Transactions on Networking, volume 9, NO. 3, June 2001.

[7] S. McClure & J. Scambray & G. Kurtz, Hacking Exposed Second Edition: Network Security Secrets & Solutions, Mc Graw Hill Companiew Inc, 2001.

[8] A. C. Snoeren & C. Partridge & LA. Sanchez & CE. Jones & F. Tchakountio & B. Schwartz & S. T. Kent & W. T. Strayer, “Single-Packet IP Traceback,” IEEE/ACM Transactions on networking, volume 10, NO.6 December 2002.

[9] R. Stone, “CenterTrack: An IP Overlay Network for Tracking DoS Floods,” 9th USENIX Security Symposium, Denver, Colorado, USA, 14-17 August 2000.

[10] X. Wang & D. S. Reeves & S. F. Wu & J. Yuill, “Sleepy Watermark Tracing: An Active Network-Based Intrusion Response Framework,” Proceedings of IFIP Conference on Security, pp.369-384, June 11-13, 2001.

二、中文部份

[11]方盈，TCP/IP 通訊協定、入門與應用，博碩文化股份有限公司，2000 年 4 月

[12]呂芳懌、楊子逸，“A Host-based Real-time Intrustion Detection System with Data Mining and Forensics Techniques,“，第三十七屆國際卡拉漢安全科技年會，2003 年 10 月。

[13]呂芳懌、楊子逸，“IIS 伺服器漏洞剖析與防備,”，第四屆 2002 年「網際空間：資訊、法律與社會」學術研究暨實務研討會，2002 年 11 月。

[14]呂芳懌、楊子逸，“IIS 網頁伺服器 Unicode 漏洞探討,”，2001 年第五屆資訊管理學術暨警政資訊實務研討會，2001 年 6 月 1 日。

[15]呂芳懌、鄭真真、洪嘉鴻，“UDAIDTS：以 Hash 為基礎的主動式區域聯防入侵偵測與追蹤系統,“，第十四屆國際資訊管理學術研討會，2003 年 7 月 11-12 日。

[16]呂芳懌、蘇俊維、許惟翔，“內部網路安全威脅分析與防制,”，2003 電子商務與數位生活研討會，2003 年 4 月 11-12 日。

[17]李志堅，“有效的動態機率封包標記”，逢甲大學資訊工程研究所碩士論文，2002 年 6 月。

[18]沈文吉，“網路安全監控與攻擊行為之分析與實作”，台灣大學資訊管理研究所碩士論文，2001 年 6 月。

[19]卓武雄，等候系統及其管理，六國出版社，1980 年 6 月。

[20]黃明凱，“網路犯罪輔助偵查專家系統雛型之建構”，中央警察大學資訊管理研究所碩士論文，2002 年 6 月。

[21]劉均緯，“未知入侵行為之偵測與預防,”，iSecuTech ，pp.88-89，2003 年 3 月。

[22]樊國楨，“從 Slammer 攻擊事件看緩衝區溢位問題,”，iSecuTech，pp.125-129，2003 年 3 月。

[23]閻雪，中國大陸的駭客技術，松崗電腦圖書資料股份有限公司，2000 年 8 月。

[24]賴冠州譯，R. G. Bace 著，入侵偵測專業手冊，旗標出版股份有限公司，2001 年 12

月。

三、網路部份

[25] http://www.ncert.nat.gov.tw/infosec/data.asp，國家資通安全應變中心。

[26] http://www.microsoft.com/security，微軟公司資通安全網。

[27] http://iss.com.tw，鈺松國際資訊股份有限公司。

[28] http://secureuni.com，諮安科技股份有限公司。

[29] http://trend.com.tw，趨勢科技股份有限公司。

[30] http://fetag.org，飛鷹系統。

[31] http://www.ncnu.edu.tw/~u8321026/CNI/HW4/CH19.html，暨南大學資訊工程學系／謝明泰資網報告。

[32] http://www.ettoday.com，東森新聞網／資訊科技，2003 年 3 月 28 日。

[33] http://www.libertytimes.com.tw/2001/new/jul/21/today-int5.htm，自由電子新聞網／國際新聞，2001 年 7 月 21 日。

附錄

附錄一　節錄 SATAN 81 條攻擊指令

cgi 攻擊

1."GET /cgi-bin/query?mss=../config HTTP/1.0" 404 273

2."GET /cgi-bin/bbs_forum.cgi?read=../../../../etc/group HTTP/1.0" 404 281
3."GET /cgi-bin/bbs/bbs_forum.cgi?read=../../../../etc/group HTTP/1.0" 404 285
// 關連->對不同的目錄 get 相同檔名的檔案->bbs_orum

4."GET /cgi-bin/dumpenv.pl HTTP/1.0" 404 278

5."GET /cgi-bin/test-cgi HTTP/1.0" 403 280
6."GET /cgi-bin/nph-test-cgi HTTP/1.0" 404 280
//關連->get 類似檔名的檔案->test-chi 和 nph-test-cgi

7."GET /cgi-bin/wwwboard.pl HTTP/1.0" 404 279
8."GET /cgi-bin/wwwboard.cgi HTTP/1.0" 404 280
9."GET /cgi-bin/wwwboard HTTP/1.0" 404 276
//關連->get 相同主檔名的檔案->wwwboard

10."GET /cgi-bin/wrap HTTP/1.0" 404 272
11."GET /cgi-bin/wrap.pl HTTP/1.0" 404 275
12."GET /cgi-bin/wrap.cgi HTTP/1.0" 404 276
//關連->get 相同主檔名的檔案->wrap

13."GET /cgi-bin/finger HTTP/1.0" 404 274
14."GET /cgi-bin/finger.pl HTTP/1.0" 404 277
15."GET /cgi-bin/finger.cgi HTTP/1.0" 404 278
//關連->get 相同主檔名的檔案->finger

16."GET /cgi-bin/phf HTTP/1.0" 404 271
17."GET /cgi-bin/handler HTTP/1.0" 404 275
18."GET /cgi-bin/info2www HTTP/1.0" 404 276
19."GET /cgi-bin/textcounter.pl HTTP/1.0" 404 282

20."GET /cgi-bin/glimpse HTTP/1.0" 404 275
21."GET /cgi-bin/aglimpse HTTP/1.0" 404 276
//關連->get 類似檔名的檔案->glimpse 和 aglimpse
22."GET /cgi-bin/webgais HTTP/1.0" 404 275
23."GET /cgi-bin/www-sql HTTP/1.0" 404 275
24."GET /cgi-bin/websendmail HTTP/1.0" 404 279

25."GET /cgi-bin/jj HTTP/1.0" 404 270
26."GET /cgi-bin/count.cgi HTTP/1.0" 404 277
27."GET /cgi-bin/imagemap.exe HTTP/1.0" 404 280
28."GET /catinfo HTTP/1.0" 404 267

29."GET /cgi-bin/saint_test_cgi HTTP/1.0" 404 282

30."GET /cgi-bin/csh HTTP/1.0" 404 271
31."GET /cgi-bin/bash HTTP/1.0" 404 272
32."GET /cgi-bin/zsh HTTP/1.0" 404 271
33."GET /cgi-bin/ash HTTP/1.0" 404 271
34."GET /cgi-bin/ksh HTTP/1.0" 404 271
35."GET /cgi-bin/sh HTTP/1.0" 404 270
36."GET /cgi-bin/tcsh HTTP/1.0" 404 272
//關連->是執行 get 和 shell 有關的檔案

37."GET /cgi-bin/perl HTTP/1.0" 404 272
38."GET /cgi-bin/perl.exe HTTP/1.0" 404 276
//關連->get 相同主檔名的檔案->perl

39."GET /cgi-win/uploader.exe HTTP/1.0" 404 280
40."GET /cgi-dos/args.bat HTTP/1.0" 404 276
41."GET /cgi-dos/args.cmd HTTP/1.0" 404 276
//關連->get 相同主檔名的檔案->args

42."GET /cgi-shl/win-c-sample.exe HTTP/1.0" 404 284

43."GET /cgi-bin/guestbook.cgi HTTP/1.0" 404 281
44."GET /cgi-bin/guestbook.pl HTTP/1.0" 404 280
//關連->get 相同主檔名的檔案->guestbook

45."GET /cgi-bin/excite HTTP/1.0" 404 274

46."GET /cgi-bin/w3-msql/index.html HTTP/1.0" 404 286

47."GET /cgi-bin/wais.pl HTTP/1.0" 404 275

48."GET /cgi-bin/wais/wais.pl HTTP/1.0" 404 280

//關連->對不同的目錄 get 相同檔名的檔案->wais

49."GET /ddrint/bin/ddicgi.exe HTTP/1.0" 404 281

50."GET /cgi-bin/db2www HTTP/1.0" 404 274

51."GET /cgi-bin/db2www.exe HTTP/1.0" 404 278

//關連->get 相同主檔名的檔案->db2www

52."GET /search97cgi/vtopic HTTP/1.0" 404 278

53."GET /cgi-bin/webplus HTTP/1.0" 404 275

54."GET /cgi-bin/webplus.exe HTTP/1.0" 404 279

55."GET /cgi-bin/webplus.cgi HTTP/1.0" 404 279

//關連->get 相同主檔名的檔案->webplus

56."GET /dsgw/bin/search HTTP/1.0" 404 275

57."GET /cgi-bin/statsconfig.pl HTTP/1.0" 404 282

58."GET /cgi-bin/wwwwais HTTP/1.0" 404 275

59."GET /cgi-bin/pi HTTP/1.0" 404 270

60."GET /cgi-bin/post-query HTTP/1.0" 404 278

61."GET /cgi-bin/ncommerce3/ExecMacro/orderdspc.d2w/report HTTP/1.0" 404 309

62."GET /cgi-bin/websync.exe HTTP/1.0" 404 279

63."GET /globals.pl HTTP/1.0" 404 270

64."GET /process_bug.cgi HTTP/1.0" 404 275

CGI 攻擊歸納	
Path 中間字串	/cgi-bin/ /cgi-win/ /cgi-dos/ /cgi-bin/w3-msql/ /dsgw/bin/ /ddrint/bin/ /cgi-bin/ncommerce3/ExecMacro/orderdspc.d2w/

	/search97cgi/ /.
Path	已陳列在上
File	*.pl ; *.cgi ; *.exe; *.html; *.bat; *.cmd
Relation	陳列在上的註解//部分

Asp 攻擊

65."GET /site/eg/source.asp HTTP/1.0" 404 278

66."GET /shop/product.asp HTTP/1.0" 404 276

67."GET /shop/product.ast HTTP/1.0" 404 276

//關連->去 get 相同主檔名的檔案->product.asp 和 product.ast

68."GET /query.asp HTTP/1.0" 404 269

69."GET /search/query.asp HTTP/1.0" 404 276

//關連->對不同的目錄 get 相同檔名的檔案->query.asp

Asp 攻擊歸納	
Path 中間字串	/site/eg/ /shop/ /search/
Path	已陳列在上
File	*.asp; *.ast
Relation	陳列在上的註解//部分

RPC 攻擊

70."GET /pccsmysqladm/incs/dbconnect.inc HTTP/1.0" 404 291

71."GET /_private/shopping_cart.mdb HTTP/1.0" 404 286

RPC 攻擊歸納	
Path 中間字串	/_private/ /pccsmysqladm/incs/
Path	已陳列在上
File	*.inc; *.mdb

Relation	與資料庫的檔案有關連

ISAPI Mapping 攻擊

72."GET /scripts/c32web.exe/ChangeAdminPassword HTTP/1.0" 404 298

ISAPI Mapping 攻擊歸納	
Path 中間字串	/scripts/
Path	已陳列在上
File	ChangeAdminPassword
Relation	通常包含/scripts/這個目錄

Servlet 攻擊

73."GET /servlet/sunexamples.BBoardServlet HTTP/1.0" 404 293

Servlet 攻擊歸納	
Path 中間字串	/servlet/
Path	已陳列在上
File	sunexamples.BBoardServlet
Relation	

php 攻擊

74."GET /piranha/secure/passwd.php3 HTTP/1.0" 404 286

Php 攻擊歸納	
Path 中間字串	/piranha/secure/
Path	以陳列在上
File	Passwd.php3
Relation	

ISAPI 攻擊

75."GET /scripts/cart32.exe/cart32clientlist HTTP/1.0" 404 295

76."GET /scripts/emurl/RECMAN.dll HTTP/1.0" 404 284

77."GET /pbserver/pbserver.dll HTTP/1.0" 404 281

78."GET /interscan/cgi-bin/FtpSaveCSP.dll HTTP/1.0" 404 292

79."GET /interscan/cgi-bin/FtpSaveCVP.dll HTTP/1.0" 404 292

//關連->get 類似檔名的檔案->FtpSaveCSP.dll 和 FtpSaveCVP.dll

80."GET /query.idq HTTP/1.0" 404 269

81."GET /search/query.idq HTTP/1.0" 404 276

//關連->在不同的目錄 get 相同檔名的檔案-> query.idq

ISAPI 攻擊歸納	
Path 中間字串	/scripts/cart32.exe/ /scripts/emurl/ /pbserver/ /interscan/cgi-bin/ /search/
Path	以陳列在上
File	*.dll; *.idq
Relation	陳列在上的註解//部分

附錄二　特洛伊木馬常用的通訊埠

http://www.fanqiang.com (2001-05-21 08:10:00)

Port number：Protocol：Name of Trojan(s)

2：TCP：Death 1.0
21：TCP：Blade 0.80 Alpha
21：FTP：Invisible FTP
23：TCP："Tiny Telnet Server, tRuVa Atl 1.2"
25：TCP："Antigen, Email Password Sender 1.03-1.6, Gip1.10,Kuang2 0.17A-0.30, Mail Bomb Trojan, Magic Horse Trojan, Moscow Mail, Taprias, WinPC "
31：TCP："Agent 31, Hackers Paradise, Masters Paradise"
58：TCP：DMSetup
59：TCP：DMSetup
79：TCP：Firehotcker
80：TCP：Exector
12：UDP：Jammer Killah 1.2
146：TCP：The Infector 1.3
170：TCP：A-Trojan
555：TCP："Ini-Killer, phAse zero 1.0 - 1.1, Stealth Spy "
667：TCP：SniperNet 2.1
669：TCP：DP Trojan 2.5
1066：TCP：B.F. Evolution 5.3.12
1212：TCP：Kaos 1.1 - 1.3
1234：Telnet：Ultor's Telnet Trojan
1243：TCP：SubSeven 1.0 - 1.8
1245：TCP：VooDoo 6
1256：TCP：Project nEXT 0.5.3 beta
1349：UDP：Back Ofrice DLL
1441：TCP：Remote Storm 1.2
1492：FTP：FTP99cmp
1777：TCP：Scarab 1.2c
1981：Telnet：Shockrave
1999：TCP："BackDoor 1.00-1.03, Transmission Scout 1.1-1.2"
2000：TCP：Transmission Scout 1.1-1.2

2001：TCP：Trojan Cow 1.0
2023：TCP：Ripper Pro
2115：TCP：BUGS
2140：UDP：The Invasor
2565：TCP：Striker 1.0
3000：TCP：Remote Shut 1.2 - 1.4
3250：UDP：The Invasor
3456：TCP：The Terror Trojan
3459：TCP：Sanctuary
3700：TCP：Portal of Doom V.3 Beta
4000：TCP：Skydance 2.16 Beta
4242：TCP：Virtual Hacking Machine
4321：TCP：Bo-Bo 1.0 Final Beta
4444：TCP：Swift Remote 1.06
4567：TCP：File Nail 1
5000：TCP：Bubbel
5031：TCP：Net Metropolian 1.00-1.04
5400：TCP：Blade Runner 0.80 Alpha
5401：TCP：Blade Runner 0.80 Alpha
5402：TCP：Blade Runner 0.80 Alpha
5569：TCP：Robo Hack 1.2
5882：UDP：Y3K RAT 1.0
5888：UDP：Y3K RAT 1.0
6000：TCP：The thing 1.6
6400：TCP：The thing 1.1 - 1.5
6669：TCP：Host Control 1.0
6712：TCP：Funny Trojan
6912：TCP：ShitHeep
6969：TCP：Gatecrasher 1.1 - 1.2
7000：TCP：Remote Grab 1.0
7001：TCP：Freak 88's DAS
7789：TCP：ICQ Killer
9000：TCP：Netministrator
9400：TCP：InCommand 1.0-1.4
9876：TCP：RUX
10085：TCP：Syphillis 1.18 Alpha
10101：TCP：BrainSpy Beta
10607：TCP：Coma 1.0.9

10666：UDP：Ambush 1.0
11223：TCP：Secret Agent 1.0
12345：TCP：NetBus 1.20-1.70
12349：TCP：BioNet
12624：TCP：ButtMan 0.9N
16484：TCP：MoSucker 1.0
16969：TCP：Priority
17166：TCP：Mosaic 2.00
20001：TCP：Millenium 1.0 - 2.0
20034：TCP：NetBus 2.0 Beta-NetBus 2.10
21544：TCP："GirlFriend 1.0 Beta-1.35, Kid Terror 1.0"
23023：TCP：Logged 1.0
23432：TCP：Asylum 0.2
23456：FTP：Evil FTP
29104：TCP：NetTrojan 1.0
29891：UDP：The Unexplained
30001：TCP：Err0r32 beta
30029：TCP：Aol Trojan 1.1
30100：TCP：NetSphere 1.27a
30101：TCP：NetSphere 1.27a
30102：TCP：NetSphere 1.27a
30947：TCP：Intruse 1.27b
31337：UDP："BackOfrice 1.20, Freak trojan 2k "
31338：UDP：DeepBO
32100：TCP：Peanut Britter 0.2 beta
33333：TCP："Blackharaz, Prosiak"
33577：TCP：Psychward small 02
33777：TCP：Psychward big - small
34324：TCP：Bigluck
40412：TCP：The Spy Beta 1
41666：TCP：Remote Boot Tool 1.0
41666：UDP：Remote Boot Too 1.0
47262：TCP：Delta source 0.5 - 0.7
47262：UDP：Delta source 0.5 - 0.7
50766：TCP：Fore 1.0 Beta
51966：TCP：CAFEiNi 0.8
53001：TCP：Remote Windows Shutdown 0.02b
54321：TCP：SchoolBus .6 - 1.60

61466：TCP：TeLeCoMMaNDo 1.5.40
65000：TCP：Devil 1.3
65535：TCP：RC