國 立 交 通 大 學

管理學院（資訊管理學程）碩士班

碩　士　論　文

運用線上分析處理與資料探勘於網路流量分析

Applying On-line Analytical Processing and Data Mining for Analyzing NetFlow Data

研 究 生：陳美君

指導教授：劉敦仁　博士

中　華　民　國　九　十　六　年　七　月

運用線上分析處理與資料探勘於網路流量分析

Applying On-line Analytical Processing and Data Mining for Analyzing Netflow Data

研 究 生：陳美君　　Student：Mei-Chun Chen

指導教授：劉敦仁 博士　　Advisor：Dr.Duen-Ren Liu

國 立 交 通 大 學

管理學院（資訊管理學程）碩士班

碩 士 論 文

A Thesis

Submitted to Institute of Information Management

College of Management

National Chiao Tung University

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science

in

Information Management

July 2007

Hsinchu, Taiwan, the Republic of China

中　華　民　國　九　十　六　年　七　月

運用線上分析處理與資料探勘於網路流量分析之實作

研究生：陳美君　　指導教授 劉敦仁 博士

國立交通大學管理學院（資訊管理學程）碩士班

摘要

隨著網路的蓬勃發展，使得各類的網路攻擊行為、病毒威脅與垃圾訊息等愈趨增加，而網路管理的問題也因為服務的多樣化而日趨複雜，於是網路頻寬、效能、服務品質、安全等便顯得更為重要。

本研究利用 Cisco 所提供之路由器、交換器等設備上的 NetFlow 技術，其所記錄的網路流量基本資訊，進行 OLAP 即時線上分析，藉以了解整體網路的即時與歷史狀態，期能即時發現網路異常狀況的發生，並藉由歷史資料的分析來發現異常狀況之蛛絲馬跡。另外，本研究分析曾經發生過 CodeRed、MSBlast 等攻擊的歷史 NetFlow 資料，透過決策樹模型來找出異常攻擊之單位時間內的流量臨界值，並將此臨界值應用於偵測網路攻擊之系統實作，以驗証該臨界值的準確性。

關鍵字: 線上分析處理、資料倉儲、決策樹、NetFlow、分散式阻斷服務、CodeRed、MSBlast、Cube

Applying On-line Analytical Processing and Data Mining for Analyzing Netflow Data

Student：Mei-Chun Chen　　Advisor：Dr. Duen-Ren Liu

Institue of Information Management

National Chiao Tung University

Hsinchu, Taiwan, Republic of China

Abstract

This study focuses on analyzing internet traffic using NetFlow technology. We use the OLAP to analyze flow traffic information and detect the real time network status of the network platform. This study aims to find the signature of network abnormal behavior through analyzing the historical netflow traffic information, which was incurred by the attack of CodeRed and MSBlast worm. Decision tree is applied to find the threshold of the abnormal network behavior. The threshold and techniques of the proposed analysis are implemented to detect the abnormal behavior of netflow traffic. Besides, experiments are conducted to verify the accuracy of the threshold.

keyword：OLAP、Data warehouse、Decision Tree、DDoS/DoS、CodeRed、MSBlast、Cube

誌 謝

感謝我的指導教授劉敦仁博士，在劉老師的悉心指導下，讓我能對各個相關領域有了更深的認知。

特別要感謝的是合勤科技與聯華電子的同事們，在碩士班上課期間，給我支持及鼓勵，其次要感謝交大資管所裡的老師和同學們，他們的指導與叮嚀，讓我能在各方面有所成長。

最後要感謝我的先生宜哲及可愛的女兒幸岑，因為他們的全力支持，我才得以安心的投注在課業上。

目 錄

圖 目 錄

表 目 錄

1. 簡介

1.1 研究動機與目的

由於網際網路的普及與電子商務的掘起，人們對於網路所帶來的商機存在著無限美好的願景，在仰賴資訊科技的同時，還是存在著許多潛在的駭客攻擊與網路安全威脅。這些威脅當然也包括了利用病毒、蠕蟲或木馬等造成 DoS/DDoS 攻擊的網路安全問題。除此之外，網路頻寬雖然大幅的增加，但網路使用的效能卻沒能成正比的提供，網路使用者常常為網路塞車所苦，主要的原因是因為網路使用者不當的網路使用行為對整體網路造成的影響，許多消耗頻寬的應用程式（例如：網路檔案分享軟體：edonkey…）等，均會造成網路阻塞，使得正常的使用者無法順利存取網路，而造成網路使用效能的低落。

傳統上網路管理人員要檢測內部網路是否有中蠕蟲或網路病毒，需要透過 IDS（入侵偵測系統）或者是 IPS（人侵防禦系統）的幫助，但由於 IDS 的假警報過多且沒有主動式防禦功能，而 IPS 雖然可以主動式保護阻擋異常網路封包的流通，但是，需在每個網段都架設一台 IPS，且會有網路頻寬的限制，這不但造成購置成本大幅增加，也會造成未來網路架構更趨複雜，而影響網路運作的效能；另外，利用掃毒軟體或入侵偵測/防禦系統還有一個潛在的缺點，就是必須有類似行為特徵發生過，或等到問題原因或特徵被發現並定義，才能充分發揮效用。

在近幾年的網蟲攻擊史裡，我們可以看到 2000 年 DDoS 的攻擊造成了 yahoo、eBay、CNN、Amazon 等商業網站癱瘓達數小時之久。2001 年 7 月 19 日首隻”紅色警戒”蠕蟲（CodeRed）出現後，就不斷在網路上搜尋及感染有安全漏洞的電腦和網路設備，並以每小時感染超過五千台電腦的速度擴散，據估計，約有四十萬台 IIS 伺服器遭到 CodeRed 的毒手，全球損失則在 40 億美元左右。2003 年 SQL Slammer 蠕蟲攻擊企業網站資料庫，造成美洲及亞洲地區的網路嚴重癱瘓。2004 年 1 月底 Mydoom 病毒更以最高一秒鐘感染 1200 封電子郵件，造成全球網路的重大損失。以上這些造成重大災情的病毒多數都屬於阻斷服務攻擊（Dos/DDoS），而阻斷服務攻擊的模式是攻擊者會發送大量的網路封包，使得被害者的網路頻寬耗盡，至使網路完全癱瘓。由此可知，在短時間內出現網路封包異常多的情況下，多數是網路出現了可疑的狀況，據此，我們可以使用網路流量的資訊來判斷是否有異常的侵入或攻擊產生。

由 Cisco 公司所開發的 NetFlow 就是一個提供網路管理者網路流量相關資料的協定，利用 NetFlow 產生的流量記錄將其與應用程式配合，可以記錄網路平常在不同時間的流量

或主機連線使用狀況，確切掌握目前所管轄的網路狀態，當發現網路或某應用程式流量異常，或是服務主機連線狀況異常大量增加或減少時，在第一時間提出警告，讓管理者可以立即採取對應措施，以在最短時間內恢復或維持網路服務的正常運作。

本論文將以 NetFlow 的資料流為基礎，透過 OLAP 的技術偵測並分析 NetFlow 的異常行為，並使用決策樹模型針對較著名的蠕蟲病毒 CodeRed、MSBlast 以及 DDoS 攻擊來找出這三種病毒最佳的 flow 臨界值，同時利用既有的異常資料來驗証此異常行為之流量，並實作分析各種情況之下的流量，提供網路管理者使用與分析，以保障網路的安全與品質。

1.2 論文架構

本論文架構如圖 1 所示，共有六個章節。第一章是簡介，說明了本論文的研究動機與目的以及論文的撰寫架構。第二章是文獻探討，針對 NBAD、NetFlow、病毒攻擊模式、Decision Tree、OLAP、Data warehouse、Cube 等主題做探討。第三章是系統分析與設計，將對實作之資料庫與資料倉儲維度與量值之設計做一說明。第四章則是系統實作部份，在此會使用決策樹模型來決定 MSBlast、CodeRed 蠕蟲病毒與 DDoS 網路攻擊的流量臨界值，並加以驗証，除此之外，還有對流量(flow)之線上即時分析系統與客製化之流量管理網頁程式之實作，第五章結論以及最後一章的參考文獻與附錄。

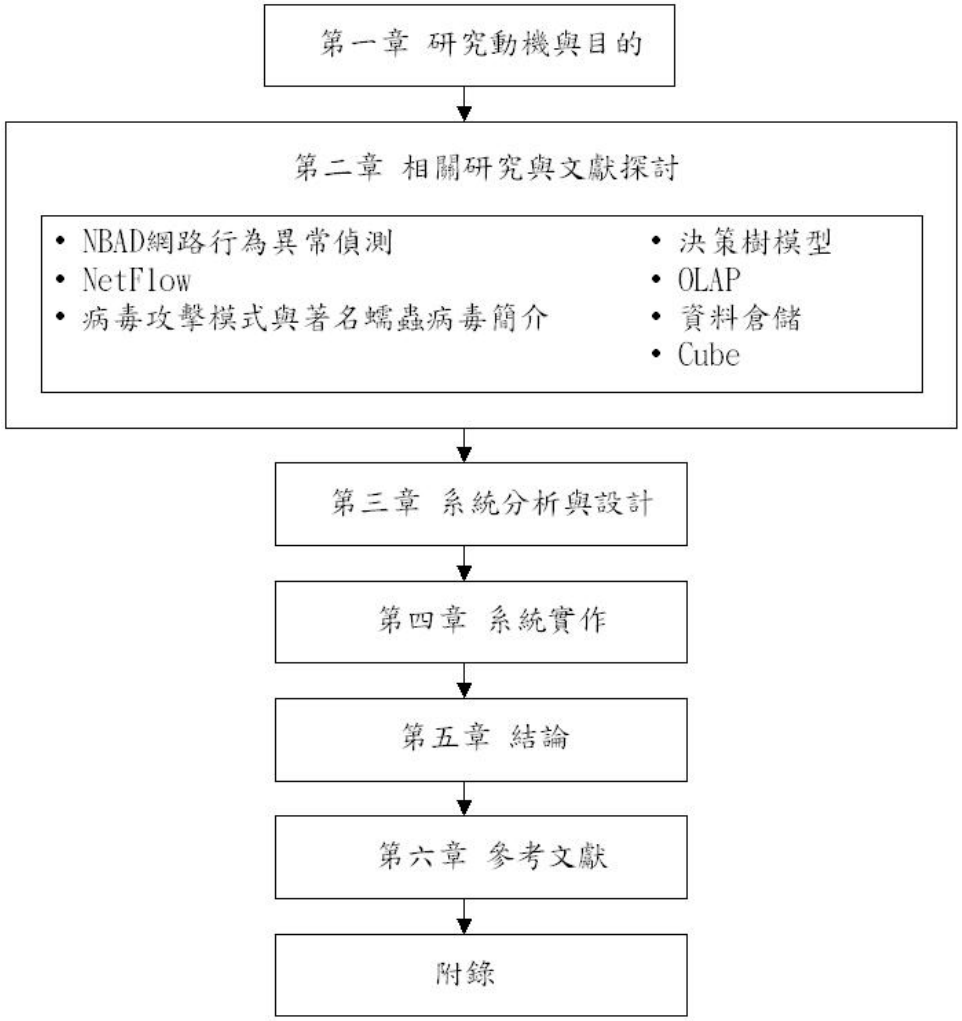


圖 1 論文架構圖

2.相關研究與文獻探討

資安事件禍首除了一些網路外部的入侵與攻擊事件外，其它大部份都是來自於網路內部，近年來如防火牆、IDS／IPS 防毒系統早已大量的被建置，隨著電腦的普及，Notebook 等行動用戶大量的增加，雖然提高了工作效率及生產力，但原本架構在安全保護網中的固定用戶端點，卻已漸漸的在改變了。使得多數原本固若金湯的企業網路，自此大門洞開。更別提木馬、間諜軟體等惡意程式，原本就較易避過防火牆及 IDS／IPS 等防護設備的檢測而直接侵入企業網路。

而 IDS／IPS(入侵偵測／入侵防禦系統)的各種設備又必須在有病毒特徵碼的情況下才能安全的防堵，但若遇新種病毒的侵入，在來不急建立特徵值的情況下，網路的安全性可能就要受到考驗了。所以本研究採用了 Cisco 的 NetFlow 資料流為研究之依據，針對幾個較著名的病毒透過 OLAP 做即時的線上分析，並提出警示。

2.1 網路行為異常偵測 NBAD (Network Behavior Anomaly Detection)

2.1.1 什麼是網路行為異常偵測

有鑑於當前 IDS／IPS 設備仍無法在第一時間內對網路上可能的異常入侵情形做快速的回應與分析，於是就有了網路行為異常偵測（Network Behavior Anomaly Detection；NBAD）的新興技術方案問世；此技術與採用病毒特徵碼的 IDS／IPS 產品所不同的是，它是一種透過多樣性網路行為特徵的辨識技術，例如某網路行為，在單位時間內的 Session 多，但傳輸量卻很少，或是具備攻擊某特定傳輸埠（Port）的動作時，即可判定為惡意的異常流量。

而網路行為異常偵測的解決方案為使用 NetFlow 的資料收集方法，從用戶的路由器設備(L3 Switch)中獲取流量和封包資料，進而建立一個高效能的資料庫。網路行為異常的偵測是透過資料庫所累計的資訊，建立偵查網路蠕蟲的模型；這樣一來，就可以把這個模型即時收到的資料，比對偵測流量及 Sessions 規則的改變。

2.1.2 異常流量分析

在曹乙帆的“資訊安全流量管理系統”〔29〕一文中有提到這種異常現象偵測方法，提供一種非常可靠的途徑，便於偵測和追蹤源於惡意攻擊，或網路配置而造成的合法流量變動，以及新服務的啟動等原因所導致的網路變化，把這種獨特的方法用於偵測網路的異常現象。當網路的通訊量突然急劇增長，超過平常的極限值時，此時就一定要提高警覺，主動去檢測一下；當網站的某一特定服務總是失敗時，也需要多加留意；總之，當機器出現異常情況時，就可以利用異常流量來查出那些 Service Port 被佔用最多及可疑 IP 有那些，並加以防範。

當然，最好的方式為防護設備還沒發現蠕蟲攻擊時可以透過網路異常的流量來即時發現異常蠕蟲活動，立即將有問題的 IP 隔離，使可能的危害降到最低，以解決網路中棘手的未知性病毒問題。

NetFlow 提供了合適的解決方案，不但可監控網路上使用者或應用層的情況，更可進一步分析網路中哪些網路服務佔用的比例最多，哪些 IP 佔用頻寬最大等，都可以藉此改善並提升網路設備的效能以及發揮最大的效益。透過建立一個正常網路使用的標準值，當發現網路流量明顯偏離標準值時，就會發出警訊。以下就針對 NetFlow 部份做一概述。

2.2 NetFlow

一種提供網路管理者網路流量相關資訊的協定，網管人員可透過 NetFlow 更快速有效的掌握目前所管轄網路的狀態。

2.2.1 NetFlow 運作機制

NetFlow 是由 Cisco 公司在 1996 年由 Darren Kerr 和 Barry Bruins 所發展的一套網路流量監測技術，在大部分 Cisco 路由器上都已內建 NetFlow，同時 Juniper、Ex-treme 等其他網路設備供應商也支援此技術，使其逐漸成為大家都能接受的標準。

在 TWCERT/CC 的“NetFlow 與網管之關係與應用”〔26〕一文中提到 NetFlow 本身主要是一套網路流量統計協定，其主要的原理是根據網路封包傳輸時，連續相鄰的封包通常是往相同目的地 IP 位址傳送的特性，配合 cache 快取機制，當網路管理者開啟路由器介面的 NetFlow 功能時，路由器介面會在接受到網路封包時分析其封包的標頭部分來取得流量資料，並將所接受到的封包流量的資訊彙整成一筆一筆的 Flow，在 NetFlow 協定中 Flow 是被定義為兩端點間單一方向連續的封包流，這意味著每一個網路的連結都會被分別紀錄成兩筆 Flow 資料，其中一筆記錄從客戶端連到伺服器端，另外一筆紀錄則是從伺服器端連回到客戶端的資訊。

路由器透過以下的幾個欄位來加以區分每一筆 Flow〔26〕：來源 IP 位址(source IP address)、來源埠號(source port number)、目的 IP 位址(destination IP address)、目的埠號(destination port number)、路由器輸入介面(router input interface)、服務種類(type of service)、以及協定種類(protocol type)，當路由器接受到新的封包時，路由器便會檢視這七個欄位來判斷這個封包是否屬於任何已記錄的 Flow，有的話則將新收集到的封包的相關流量資訊整合到對應的 Flow 記錄中，如果找不到封包對應的 Flow 記錄，便產生一個新的 Flow 記錄來儲存相關的流量資訊。由於路由器內快取記憶體的空間有限，無法無限制的容納持續增加的 Flow 紀錄，所以 NetFlow 協定也定義了終止 Flow 記錄的機制，來維持網路設備中儲存 Flow 資訊的空間。只要下面三種情況任何一個成立，路由器就會透過 UDP 封包將終止的 Flow 紀錄匯出到使用者事先指定的 NetFlow 資料收集處：

(1) 當封包內旗標欄位 (flag) 顯示傳輸協定中傳輸完成的訊息如 TCP FIN 時。

(2) 流量停止超過 15 秒。

(3) 流量持續傳送，每 30 分鐘會自動終止。

雖然目前大部分的網路硬體供應商都有支援 NetFlow 的技術，但是各個廠商還是實作了自己版本的 NetFlow，其中 NetFlow Version 5 是最常見的 Netflow 資料格式，它包含了以下幾個欄位，如表 1：

表 1 NetFlow 資料格式

欄位名稱	說　明
Source IP Address	來源主機之 IP 位址
Destination IP Address	目的主機之 IP 位址
Source TCP/UDP Port	來源主機所使用的埠號
Destination TCP/UDP Port	目的主機所使用的埠號
Next Hop Address	下一個端點的位址
Source AS Number	來源主機所屬的 AS 編號
Destination AS Number	目的主機所屬的 AS 編號
Source Prefix Mask	來源主機所屬網域的子網路遮罩
Destination Prefix Mask	目的主機所屬網域的子網路遮罩
Protocol	使用之通訊協定
TCPFlag	封包控制旗標
Type of Service	QoS 需求參數
Start sysUpTime	起始時間
End sysUpTime	終止時間
Input ifIndex	資訊流流入介面編號
Output ifIndex	資訊流流出介面編號
Packet Count	封包數量
Byte Count	Byte 數量

支援 NetFlow 功能的網路設備將其所收集到的 Flow 資訊以 UDP 封包送往預先設置好的流量接收主機，配合 NetFlow 相關收集軟體，將這些原始流量資料作適當的處理、儲存以提供後續的相關應用；由於 NetFlow 只有單純分析封包的標頭，所以 NetFlow 的紀錄只包含了流量的相關資訊，雖然如此 NetFlow 仍然能夠提供足夠的資訊來協助網路管理者掌握所管轄網路中異常的網路行為。另外 NetFlow 並未對封包內容進行分析，這樣可減輕網路設備運算處理的負擔，所以 NetFlow 的效率會比傳統的方式更好，也就更適合用來分析高速、忙碌的網路環境。由於 NetFlow 資料來源是網路中的核心元件－路由器，所以透過從路由器所蒐集到的 NetFlow 資訊可以協助掌握整體網路的情況，若再

透過適當的分析 NetFlow 資訊，更可以協助管理者在蠕蟲爆發或不正常網路行為的初期快速的偵測出網路的問題。

2.2.2 NetFlow 在網路安全上相關的應用

使用 NetFlow 來分析網路狀況其實是一種「異常偵測」(anomaly detection)的應用，藉由分析網路狀態找出與正常情況不同的異常狀況，而不像特徵式入侵偵測系統那樣，需利用網路封包的負載程度來偵測攻擊行為。

在 TWCERT/CC “NetFlow 與網管之關係與應用”一文中也有提到〔26〕網路攻擊行為存在著某些可供辨識的特徵，例如針對某個特定埠或利用某些特定網路的 IP 位址等等特徵；我們可以透過這些特徵來與所獲得的 NetFlow 資料進行比對，進而找出可能的異常行為，透過分析 NetFlow 資料中目的主機所使用的埠號（Port）欄位，例如 SQL Slammer 就是利用 1434 port 進行感染，利用目的主機所使用埠號這個欄位等於某個特定埠號，來過濾 NetFlow 資料找出相對應的攻擊，另外我們也可以利用不合邏輯的來源或目的 IP 位址來找出異常的 IP，再利用 NetFlow 資料中資訊流流入介面編號（Input IFindex）欄位的資訊，找出連接這個介面的上游路由器，再請網路管理者協助調查或處理。

某些異常行為可能會連到某個或某些特定位址，例如我們對造成嚴重網路擁塞的 CodeRed 蠕蟲對其 NetFlow 資訊加以分析便可發現此蠕蟲的攻擊行為有一個特性，每筆 Flow 的 destination TCP/UDP port 欄位值會等於 80，Packet Count 欄位值等於 3，Byte Count 欄位值等位 144 bytes，網路管理者便可以撰寫程式進一步的分析所蒐集的 NetFlow 資料，找出具此特徵的 Flow，於是便可找出網路內有可能感染 CodeRed 蠕蟲的主機，對其進行封鎖或下線的動作，以降低蠕蟲造成的危害。利用已收集到攻擊的特徵與 NetFlow 資訊中的相關欄位進行比對找出可能的攻擊，可以在攻擊造成網路嚴重傷害之前，做適當的反應措施來降低形成嚴重問題的可能性。

另外我們也可以在 NetFlow 資料中找出建立 session 數目最多的主機，因為如果一台主機對特定主機產生不正常的大量連結需求時，這可能代表著新的蠕蟲、阻斷服務攻擊、網路掃瞄等的情形發生，因為一般正常的主機對外連結會有一定正常的頻率。我們可以從感染蠕蟲的主機的 NetFlow 資訊中發現到大量的對外連結需求，例如我們就可以從感染 SQL Slammer 的主機上發現大量對外 1434 port 的連結需求。同樣的原理，如果所管轄網路中的使用者從網路上下載阻斷服務攻擊之工具程式企圖對外發動攻擊時，或是使用者利用 Nmap 之類的掃瞄工具掃瞄特定網址，以找出目標主機所可能存在弱點或是漏洞時，我們都可以從 NetFlow 資料中發現從網域中某個特定位址送出的大量 session。除了偵測網路的攻擊外，我們也可以透過分析 session 的方式找出網路濫用的行為，並進行適當處置以降低其所造成的傷害。

2.3 網路惡意攻擊模式與著名蠕蟲簡介

2.3.1 DoS（Denial of Service）阻斷服務

DoS 是一對一的網路攻擊方式，攻擊者藉由不當方式佔用系統分享資源(CPU、網路、硬碟…)，達到干擾正常系統運作的進行，和一般網路入侵不同的是，DoS 不一定需要取得系統使用的權力即可達到目的。最常見的 DoS 方式就是向攻擊對象送出大量且無意義的網路訊息，不管被攻擊的對象是否有回應，都會因頻寬的被佔用而導致不正常的運作。例如駭客試圖用大量封包攻擊一般頻寬相對小的 ADSL 使用者，受害者就會發現他要連的網站連不上或是反應十分緩慢。

2.3.2 DDoS（Distributed Denial of Service）分散式阻斷服務

DDoS〔25〕也算是 DoS 的一種，它的攻擊模式並非一對一，而是以多對一的方式同時對一個攻擊目標發動攻擊，而這些發動攻擊的點，通常是已遭受入侵而不自知的電腦系統。由於這種攻擊多以遙控的方式，利用替死鬼行兇，因此不僅難以防範，追查更是不易。更糟的是，這些被用來發動攻擊的程式不僅可在網站中取得，即使不是電腦高手也可輕易使用。

2.3.2.1 DDoS 攻擊模式

從技術上來看，DDoS〔25〕攻擊模式大致分成幾個不同的階段，並透過不同的網站來進行，底下我們大略簡單介紹一下，並請參考圖 2：

1. Client（攻擊者所在的系統，簡稱 C）；
2. Host（攻擊者發號施令的監控系統; 攻擊者可以直接控制，簡稱 H）；
3. Broadcaster（放大攻擊直接來源; 被殖入攻擊程式者，簡稱 B）；
4. Target（被攻擊者，簡稱 T）。

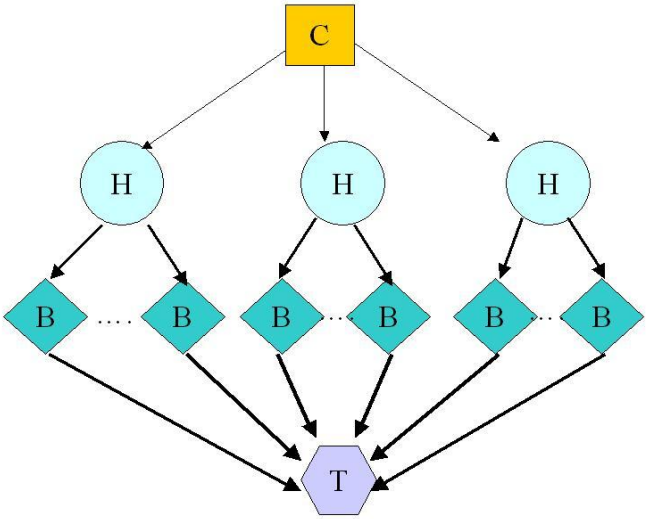


圖 2 DDoS 攻擊式意圖

第一階段, 攻擊者從自己的系統(定為 Client 者)出發，在網路上找到幾個管理鬆散的網站，入侵系統管理者的帳號(root、super user 等)，並利用許多已知系統的安全漏洞，入侵系統取得管理者權限，並將這幾個系統變成往後攻擊、發號施令與監看成果的主機(定為 Host 者)。

第二階段，透過同樣的 port scanning、www server searching、DNS searching 等技巧，攻擊者再次在網路各地找尋更多有安全漏洞的 Unix、Windows 等系統的網站，然後再殖入一些特定的網路服務程式(Daemon)，潛藏在該系統中，伺機等候前述攻擊者所在的 Host，發動網路攻擊的命令。這一些機器，也就成為潛在的放大攻擊者(定為 Broadcaster 者)，糟糕的是，這些特定的攻擊程式，往往都被設計成具有將封包放大的特異功能，同時也會隱藏封包的真實來源，並偽裝成其他來源，以誤導系統防護與追查的方向。

2.3.2.2 被 DDoS 攻擊時的現象

- 被攻擊主機上有大量等待的 TCP 連接。
- 網路中充斥著大量無用的封包，且來源位址為偽造的。
- 大量且無用的封包，造成網路擁塞，使受害主機無法正常和外界通訊。
- 利用受害主機提供的服務或傳輸協定上的缺陷，反覆高速的發出特定的服務請求，使受害主機無法及時處理所有正常請求，嚴重時會造成系統當機。

2.3.3 蠕蟲（Worm）

蠕蟲(Worm)會在電腦之間以自動的方式不斷的自我複製，所以系統一旦被蠕蟲感染，就會自動蔓延。而其最危險之處就是擁有大量複製的能力；例如，蠕蟲可將自己複製並傳給電子郵件通訊錄內所有的人，而收件者電腦也會繼續以相同的動作將自己複製並傳給其

它的電腦，最後造成大量網路流量的連鎖效應，進一步降低整個企業網路和網際網路的速度。新蠕蟲一旦出現就會快速地散播出來，不但會消耗網路頻寬，降低瀏覽網頁的速度，嚴重時還會造成電腦當機。

2.3.4 CodeRed 病毒簡介與特徵

在參考了台灣賽門鐵克網站的病毒應變中心裡的病毒簡介後我們知道了 CodeRed〔27〕是一隻會自我繁殖入侵系統的惡意程式碼，利用微軟 IIS Web Server 的安全性漏洞入侵，並在受害者的主機上自我繁殖，借機會取得主機的控制權，然後再隨機產生 IP，並利用 DoS 攻擊的模式，嘗試入侵尚未安裝修補程式的 IIS Web Server，由於持續變換目地 IP 位置，消耗路由器資源，如此便會造成網路頻寬壅塞，並導致路由器當機或效能下降。

特徵：

在 CodeRed 流量統計中每個 flow 代表一次 destination port=80, packets=3, size=144 bytes 的行為，雖然在 internet 上，符合上述特性的正常行為還是存在（如使用 ICQ），但是一般正常使用的電腦並不會在短時間內出現符合此特徵的流量異常多的情況，通常會界定一個流量臨界值，當超過此流量臨界值時就需特別注意了。

2.3.5 SQL Slammer 病毒簡介與特徵

SQL Slammer 主要的攻擊目標為 Microsoft SQL Server 2000 及 Microsoft Desktop Engine（MSDE），它會侵入受害主機並以 UDP port 1434 高速傳送封包，造成網路的滿載，形成分散式阻斷服務攻擊（DDoS）。

在參考了台灣微軟網站關於“Slammer 病毒最新消息”一文中瞭解了 SQL Slammer〔20〕是利用 SQL Server 解析服務(Resolution Service)緩衝區溢出(buffer overflow)的弱點而允許在 SQL Server 上執行任意程式碼，以及利用 keep-alive 功能對其它主機展開阻斷服務(Denial of Service)攻擊。且攻擊者可以偽造來源 IP 為其中一台區域 SQL Server，並且傳送封包給鄰近的 SQL Server，造成兩台 SQL Server 不停的交換封包，以降低受害 SQL Server 執行效率進一步地消耗大量的系統資源及網路頻寬。

特徵：

在 SQL Slammer 的流量統計中，每個 flow 代表一次 destination port = 1434 的行為，由於此種病毒會使用 DoS 的攻擊方式，所以若出現符合此特徵的流量大於預設的臨界值，即需請網路管理者特別注意。

2.3.6 Nimda 病毒簡介與特徵

於台灣微軟“關於 Nimda 病毒資訊”中提到，Nimda 病毒〔21〕會透過三種感染管道在網路上大量散播，包含了電子郵件、網路資源分享及微軟 IIS 伺服器感染。此病毒的主要破壞行為是透過電子郵件大量散播夾帶檔名為 Readme.exe 的電子郵件，造成網路頻寬的壅塞，使用者會明顯發現網路的速度變慢。另外，Nimda 病毒會自動尋找網路上的芳鄰及微軟 IIS 網頁伺服器進行感染，如果不小心連上中毒網站，將遭受病毒的感染。如此，災情會以一傳十、十傳百的倍增速度感染。

Nimda 散播速度快且佔用了許多網路的頻寬，破壞能力超強，它還提供了擁有 root 權限的 guest 使用者操作，影響程度除了降低電腦效能，拖慢連線速度，更動受害主機中的檔案及登錄資料外，還會藉由 email 傳遞。

被感染後的現象:

- 大量電子郵件寄送：此病蟲會以 Readme.exe 的名稱，利用 MAPI 格式自行複製寄送，值得注意的是，Readme.exe 這個檔案並不會以附件的格式出現在電子郵件中。
- 改變檔案格式：此病蟲會自行取代正常的檔案。
- 降低效能：可能會造成電腦系統緩慢。
- 降低資訊安全系統：病蟲會將使用者的 C 槽開放成網路共享的資源。
- 在使用 word 或 excel 時，會出現記憶體不足的錯誤訊息。

特徵：

由於 Nimda 病毒使用了多種的漏洞來做感染，並且被攻擊的主機會有不同的行為，無法透過特定的 Packet 數量或 Bytes 數量來作為判斷，所以使用 Flow 的 destination port = 80 來做為判斷的特徵。疑遭感染 Nimda 病毒的統計分析中每個 Flow 代表一次 destination port=80(http)的行為，如果 10 分鐘內 Flows 數超過所設定的臨界值，除非是 proxy server，在大部份的情況下很有可能該主機已遭 Nimda 感染或是其它會送出大量 http 攻擊封包的病毒所侵害，必須儘快檢查。

2.3.7 Scan Port 137 病毒簡介與特徵

這是一個在 NetBIOS service 上所發現的漏洞，台灣微軟網站”NetBIOS 的瑕疵可能導致資料被洩露”一文提及 NetBIOS〔19〕為一檔案分享的協定，包含 netbios-ssn(139/tcp)以及 netbios-ns(137/tcp)，NetBIOS 會洩漏出遠端主機的卡號以及網域內的主機名稱，入侵者便可以使用密碼破解工具，對分享的磁碟機進行字典攻擊的破解。

NetBIOS Name Server (NBNS)及 NetBIOS over TCP/IP (NBT)可在 Windows Internet Name Service(WINS)此服務上設定，WINS 接受不經授權的客戶端登錄，惡意使用者藉由

假冒電腦名稱造成同一網路上相同名稱的正常電腦無法登錄 WINS 而無法透過電腦名稱與網路上的電腦溝通。

特徵：

在 Scan Port 137 流量統計中每個 Flow 代表一次 destination port=137 或 port=139(NETBIOS Name Service 網路上的芳鄰)的行為，如果在短的時間內 Flow 數超過設定的臨界值代表該主機正在進行 scan port 137 的動作，不是被安裝後門程式就是有人利用 scan port 的軟體正在嘗試入侵別人的主機。

2.3.8 MSBlast 病毒簡介與特徵

台灣微軟網站“關於 MSBlast 蠕蟲與變種病毒的警告”中有提到 MSBlast 病毒〔18〕是針對微軟視窗作業系統漏洞攻擊，目前發現使用微軟 NT4.0 以上版本（含 Win2000、XP 及 2003(.Net)系統），因作業程式本身安全性問題，將造成系統不穩定或應用程式無法執行，及自動關機等異常情形，而無法正常使用電腦。

MSBlast 攻擊的模式是利用 Microsoft RPC(Remote Procedure Call)(port 135) DCOM(Distributed Component Object Model) overflow 的漏洞，借由此漏洞取得受害電腦的完整權限後，在該電腦上執行任何程式碼，並持續掃描攻擊網路上仍有此漏洞的電腦的 135 port。

被 MSBlast 攻擊時的現象

- 出現 RPC 服務意外終止倒數 60 秒重新啟動的訊息，造成系統不斷重開機；
- 無法拖曳圖示；
- 無法執行複製或貼上的動作；
- 新增移除程式呈現空白狀態；
- 某些應用程式無法執行，如 Internet Explorer、Microsoft Outlook、Outlook Express、MS Office；
- 系統與網路速度明顯地變慢。

特徵：

MSBlast 主要是利用的 TCP/IP 連接埠 135、5554、9898，以大量 icmp 封包攻擊，packet 大小為 92 bytes，傳送資料至 TCP 埠號 135，藉以探測 DCOM RPC 弱點。

2.4 決策樹演算法（Decision Tree）

決策樹採用了樹狀分岔的架構來產生規則，由於決策樹所產生的規則很容易的就可以讓人理解，所以它常被用來處理一些分類的問題。

以下圖來說明決策樹演算法的運算邏輯：

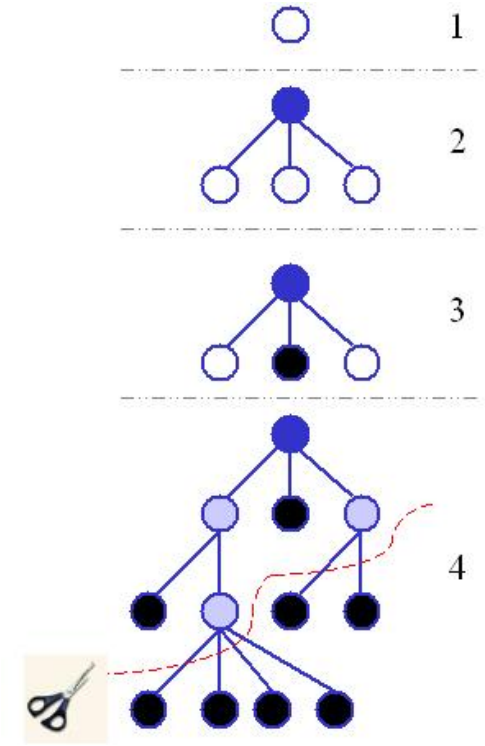


圖 3 決策樹 （參考 SQL Server 商業智慧聖經，16）

1. 使用訓練用的資料母體來當做決策樹的根節點；

2. 決策樹演算法會逐步根據每個輸入變數試算，找出哪個變數能夠產生最佳分類效果，然後依據此產生子節點；

3. 根據每個子節點預測結果的分佈狀況來產生預測機率；

4. 決策樹持續生長，最後採用修剪技術修剪去不必要的規則。

決策樹演算法〔16〕透過衡量資料的亂度來進行資料分類，當分類越準確時，資料亂度就會越低。演算法會逐步地搜尋每一個輸入的變數，以找出最佳的分岔變數，如果輸入的變數是連續變數，如身高，那麼決策樹演算法就會自動根據資料內容找出該連續變數的最佳切割點。

接著，持續以上的步驟，讓每一個產生的子節點都視為根節點，再繼續試算下一個最佳的分岔變數，繼續產生新的分岔。就這樣，決策樹會開始生長，直到子節點再也無法產生亂度更低的分岔為止。但是為了避免決策樹無限生長，在演算法中會加入修剪的機制，將不準確的規則或者是太瑣碎的規則加以去除。

2.5 OLAP 線上分析處理 (On-line Analytical Processing)

線上分析處理〔15〕是利用多維度資料的模型架構，提供決策者操作決策資料的分析查詢功能，讓決策者從分析資料的角度進一步理解決策資訊；主要功能有：向下探勘維度階層(Drill-Down)及向上匯總維度階層(Roll-Up)，利用各種圖表改變展線方式等。一般來說其種類有：

- 關聯式線上分析處理(ROLAP:Relational OLAP)；
- 多維式線上分析處理(MOLAP:Multi-dimensional OLAP)；
- 混合式線上分析處理(HOLAP:Hybrid OLAP)；

以上這三種的差異是實際資料在儲存模式上的不同，同時維度與資料量的大小也影響其分析與展現的效能。

ROLAP 關聯式線上分析處理

針對建立在關連式資料庫中的資料進行分析：

定義：資料儲存於關聯性資料庫(RDBMS)，不事先作運算。

優點：可大範圍隨性查詢。

缺點：查詢效率差（數分鐘到數小時），維護成本高，無法處理複雜運算。

MOLAP 多維度線上資料分析

針對建立在多維度資料庫中的資料進行分析：

定義：資料方塊存放在多維度資料處理伺服器端，事先做彙整計算並把結果寫入資料方塊(多維度資料庫)。

優點：查詢效率佳，可處理複雜運算功能，適合多人使用，適合網路應用結構

缺點：需大量的儲存容量，需要轉成資料方塊。

HOLAP 混合式線上資料分析

對儲存在兩種資料庫中的資料進行分析：

定義：原始資料儲存於關聯性資料庫，事先彙整計算處理過的資料存在資料方塊中，以操作邏輯規劃使用者應用模式來發揮效率；也就是在檢視合計資料時可以透過 MOLAP 資料庫，而需要明細資料時則使用 ROLAP 資料庫。

優點：查詢效率佳，可有範圍隨性查詢，可處理複雜運算功能，適合多人使用，適合網路應用結構。

缺點：建置成本高，維護成本高，維護技術複雜。

維度模式

在多維度查詢中維度模式的建立是重要的一環，維度模式的架構中所有的表格被歸納為兩個類型：事實表格（Fact Table）以及維度表格（Dimension Table）；事實資料是能夠反應過去事實的資料，而維度資料則是為了使查詢更加快而建立的索引參考資料，其特性如表 2 所示。

表 2 事實資料與維度資料特性

事實資料	維度資料
幾百萬筆／上億筆資料	遠比事實資料少
擁有多個外部索引鍵(Foreign key)	擁有單一主索引鍵(primary key)
數字資料	文字敘述資料
不會變更	經常變更

另外維度的模式一般可分為星狀模式（Star Schema）、雪花狀模式（snowflake schema）及星狀雪花模式（star-flake schema）三種類型。

2.6 資料倉儲（Data warehouse）/ 資料超市（Data Mart）

資料倉儲〔15〕從定義來說，是具有主題導向、資料整合與時間性的資料庫，可進一步利用線上分析、資料探勘等各種知識發現工具，來提供決策者或研究者快速、視覺化且具分析性的資訊，以提供精確之決策與分析之用。它和一般線上交易系統(OLTP)不同之處在於它儲存的不是目前營運交易的資料，而是經過處理、匯總後的資料，讓不同來源的資料擁有一致性的格式、名稱，以免造成混亂。

資料超市其概念和資料倉儲相似，但規模較小些，是個應用目的更為明確的資料匯整方式。雖然成本低、建置快是資料超市的一個優點，但是企業如果需要跨不同的資料超市才能查詢結果時，就會是個瓶頸。

2.7 Cube - OLAP 的資料儲存體

Data Cube〔15〕是 OLAP 中最基本的建構單元，是從資料倉儲子集合中所建立的資料集合，也是提供快速回應查詢資料的機制。Cube 是由維度(Dimension)與量值(Measure)所定義的多維度結構(Multi-dimensional Structure)，可提供使用者快速而複雜的查詢。

1. 維度(Dimension)：是 Cube 內屬於敘述性的資料映射成不同階層的資料表示方式，例如時間，地區，使用者性別等變數即是常見的維度。

2. 量值(Measure)：是 Cube 資料內屬於計量化的欄位，如 packet 總數，flow 總數，packet size 總數等，也是資料分析中最感興趣的項目。

Cube 中的每個維度可依需要再建立數個階層關係，像是時間維度可以分為的年、季、月、週、日等階層，而營業單位又可以再細分成區域別、縣市別、營業單位等層級，如此的設計可以讓使用者不需要透過資訊人員撰寫程式就可以用最簡單的方式操作這些維度階層，分析各階層的數值，查詢較複雜的問題。

3. 系統分析與設計

3.1 系統架構

資料的來源是由 Cisco router dump 出來的 NetFlow 資料直接以文字檔(.txt)的方式儲存後，再以人工處理的方式轉換至 Excel 檔案中，方便轉入 MS SQL Server 的資料庫。對轉入資料庫後的資料再做進一步的資料粹取與整理成資料倉儲，於是就成了後端分析與查詢應用程式的基礎了。另外，本論文會針有特定行為與無特定行為之入侵攻擊病毒做分析，對於有特定行為之病毒我們挑選了 CodeRed、MSBlast 二種較著名的蠕蟲病毒之入侵資料，透過線上分析報表的呈現做分析與驗証，對於無特定行為之入侵攻擊我們則歸類為疑似入侵之行為。整個系統分成了七大部份，在分析部份我們將分為流量統計分析與異常行為之監控二層面，最後並以網頁報表的方式來呈現。詳細的系統架構圖請參考圖 4：

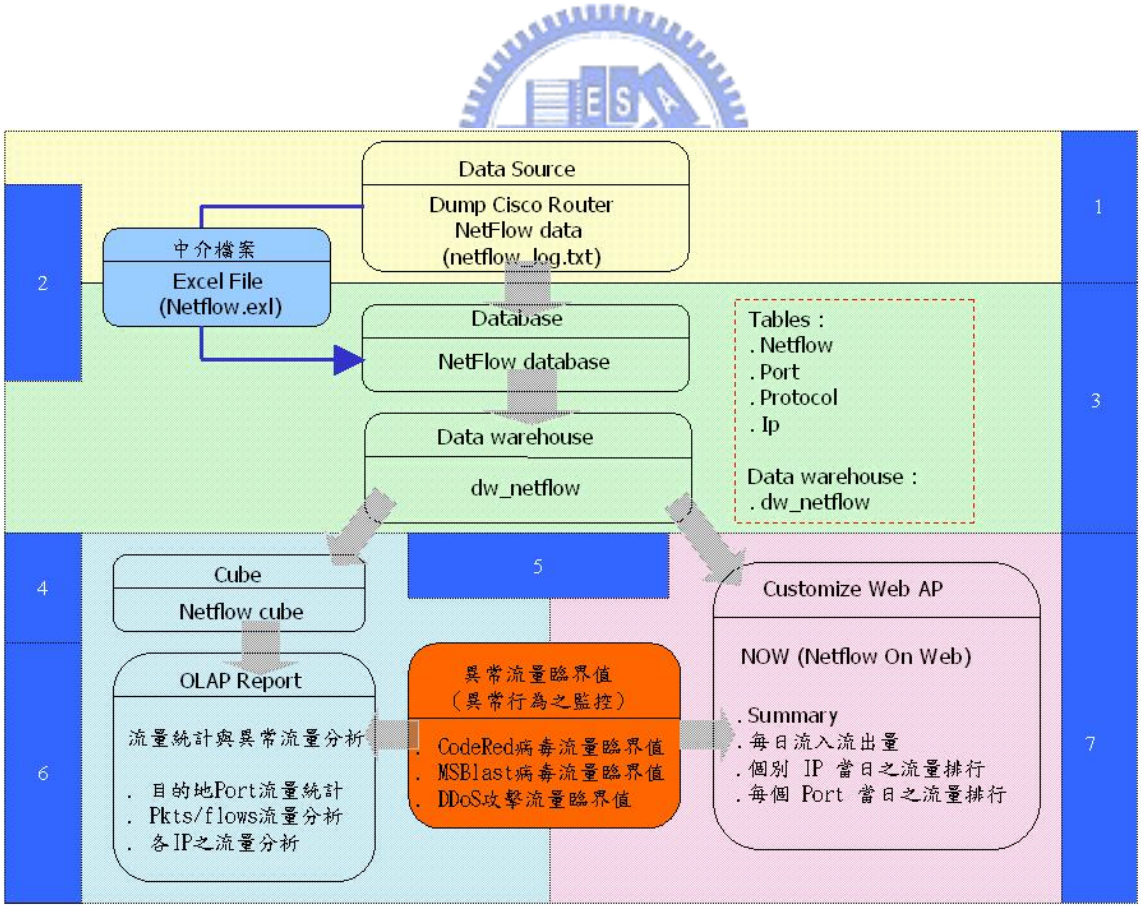


圖 4 系統架構圖

1. Data Source：由於 netflow 是 Cisco 自訂的網路流量統計格式，所以必須將 netflow 資料由 Cisco Router 中 dump 出來，先暫存成文字檔。

2. 中介檔案：由於 data source 部份我們是先 dump 出來存成文字檔(.txt)，而文字檔內的資料最終就是要轉入資料庫，在沒有設計自動轉檔程式的情況下，將文字檔先轉成 Excel 檔案格式，再轉入資料庫。

3. Database & Data warehouse：轉入資料庫後的 netflow 原始資料，先存成名為 netflow 的資料表格，然而，此資料表格內的資料是我們使用 terminal 將資料 dump 出來，所以，大約每 30 秒到 1 分鐘的時間，我們就會將其存成一個一個的 log 檔案。這些 log 檔案經過整理後，粹取出一筆一筆的 flow 資料並將其 insert 至 netflow 資料表格中，就成為了最基礎的資料。

   當然，這個 netflow 資料表格內的資料我們還是必須經過加工，處理成為另一個名為 dw_netflow 的 data warehouse。

4. Cube：cube_dwnflow_sifip_dport_area 是我們為本論文所設計的一個 cube，它所依據的資料來源就是 dw_netflow 裡的資料。

5. 異常流量臨界值：由於網路攻擊性的蠕蟲病毒種類多得不勝其數，本論文使用決策樹演算法對較著名的 CoreRed 與 MSBlast 兩種蠕蟲病毒做流量的臨界值分析，用此臨界值界定是否感染了此種類的病毒；至於其它會影響到網路流量具攻擊性的病毒我們則將其歸類為 DDoS 式攻擊型態，並對其做流量臨界值的分析。

6. OLAP Report：在這裡，我們使用了 Analyzer 2005 快速的依據我們所設計好的 cube，來產生 OLAP 線上分析報表，我們可以透過精美的圖文設計來查看所要查詢的資料，同時，也可以利用它來設計各種不同的排行規則(遞增／減，80/20 法則，監測值的特別處理，Top N 等等的規則設定（台灣睿智，22））來滿足我們對報表的需求。

7. Customize Web AP：最後，再加強就是客製化網頁應用程式的部份，有些比較簡單、容易的查詢(不太需要透過 cube 的方式來產生線上分析的報表，例如每日個別 IP 流量排行等)，我們會比較傾向於自行設計，因為，這樣較能掌握資訊的需求與變動的彈性度。

3.2 系統設計

我們依據系統架構圖內的七大部份，分別說明設計的詳細內容：

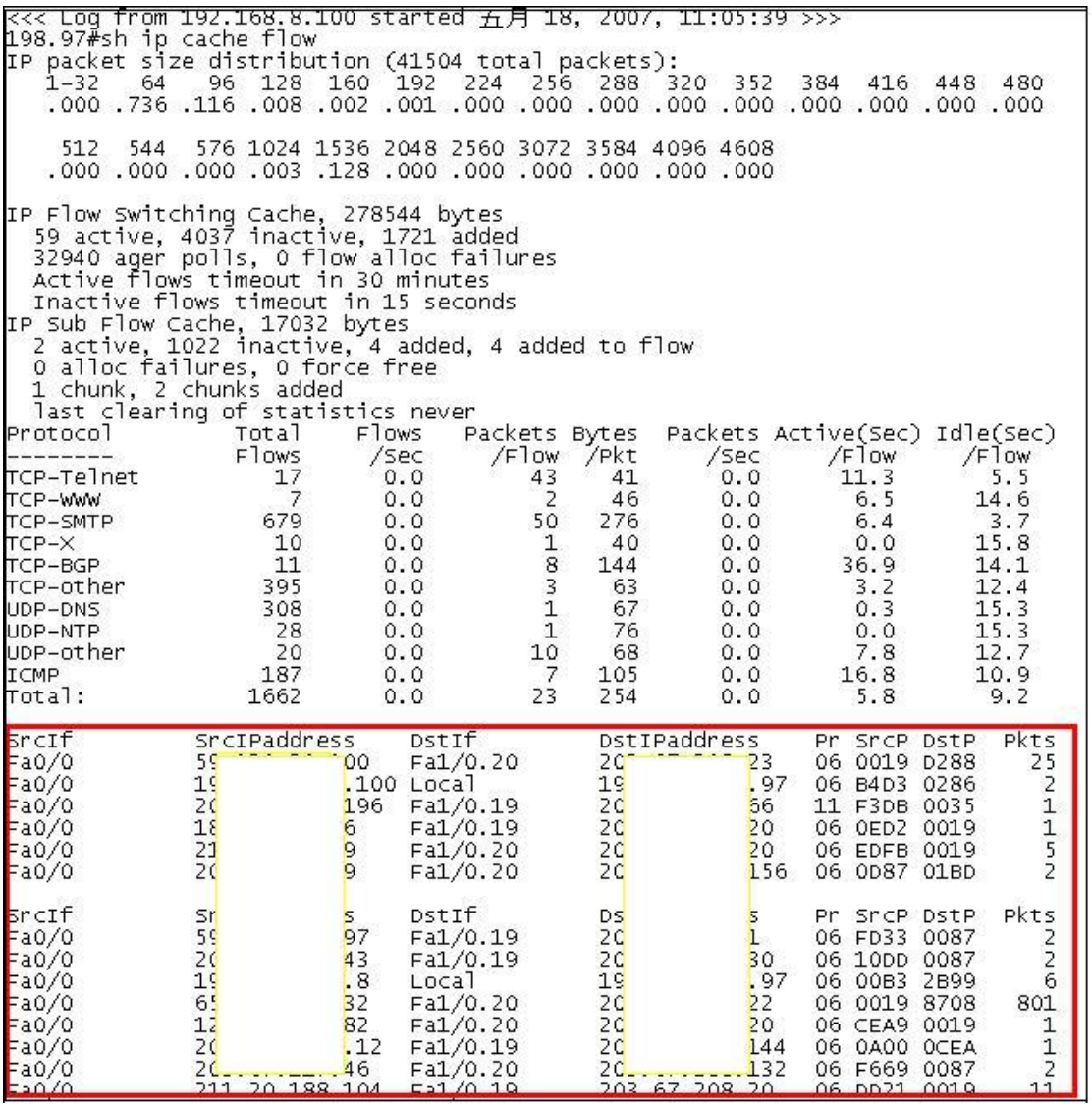
3.2.1 Data Source

資料的來源是具有 NetFlow 功能的路由器設備，先透過以下的指令，在指定的 interface 啟動統計的功能

ip flow ingress

啟動之後，再使用下面的指令,將累計的結果顯示出來，如圖 5 所示；

show ip cache flow



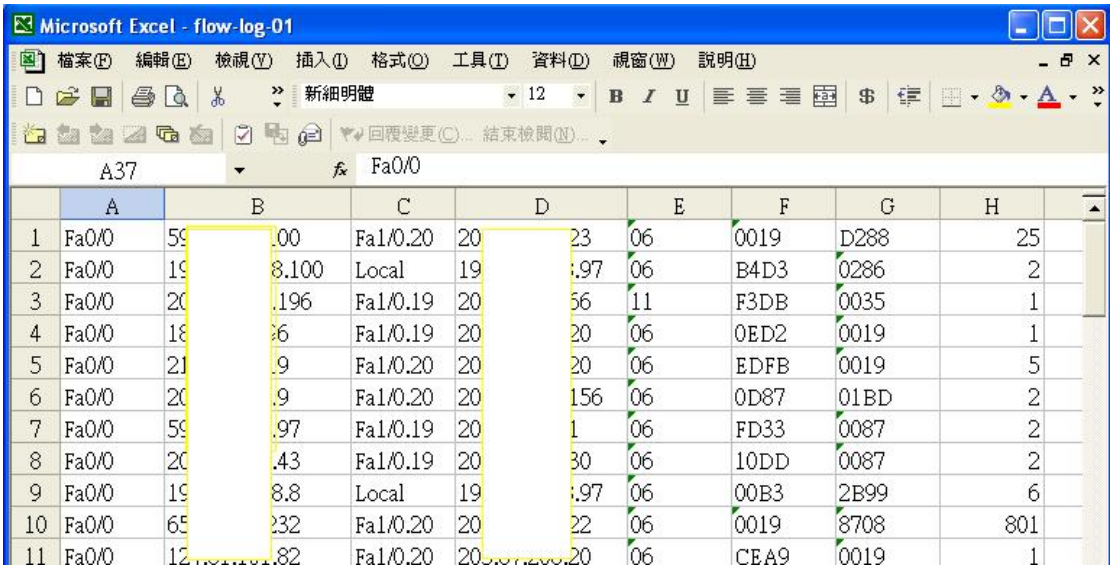
```
<<< Log from 192.168.8.100 started 五月 18, 2007, 11:05:39 >>>
198.97#sh ip cache flow
IP packet size distribution (41504 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .736 .116 .008 .002 .001 .000 .000 .000 .000 .000 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .000 .003 .128 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
  59 active, 4037 inactive, 1721 added
  32940 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 17032 bytes
  2 active, 1022 inactive, 4 added, 4 added to flow
  0 alloc failures, 0 force free
  1 chunk, 2 chunks added
  last clearing of statistics never
Protocol         Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------         Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet          17      0.0        43    41      0.0      11.3       5.5
TCP-WWW              7      0.0         2    46      0.0       6.5      14.6
TCP-SMTP           679      0.0        50   276      0.0       6.4       3.7
TCP-X               10      0.0         1    40      0.0       0.0      15.8
TCP-BGP             11      0.0         8   144      0.0      36.9      14.1
TCP-other          395      0.0         3    63      0.0       3.2      12.4
UDP-DNS            308      0.0         1    67      0.0       0.3      15.3
UDP-NTP             28      0.0         1    76      0.0       0.0      15.3
UDP-other           20      0.0        10    68      0.0       7.8      12.7
ICMP               187      0.0         7   105      0.0      16.8      10.9
Total:            1662      0.0        23   254      0.0       5.8       9.2

SrcIf         SrcIPaddress    DstIf         DstIPaddress    Pr SrcP DstP  Pkts
Fa0/0         59              Fa1/0.20      20              06 0019 D288    25
Fa0/0         19              Local         19              06 B4D3 0286     2
Fa0/0         20              Fa1/0.19      20              11 F3DB 0035     1
Fa0/0         18              Fa1/0.19      20              06 0ED2 0019     1
Fa0/0         21              Fa1/0.20      20              06 EDFB 0019     5
Fa0/0         20              Fa1/0.20      20              06 0D87 01BD     2

SrcIf         Sr              DstIf         Ds              Pr SrcP DstP  Pkts
Fa0/0         59              Fa1/0.19      20              06 FD33 0087     2
Fa0/0         20              Fa1/0.19      20              06 10DD 0087     2
Fa0/0         19              Local         19              06 00B3 2B99     6
Fa0/0         65              Fa1/0.20      20              06 0019 8708   801
Fa0/0         12              Fa1/0.20      20              06 CEA9 0019     1
Fa0/0         20              Fa1/0.19      20              06 0A00 0CEA     1
Fa0/0         20              Fa1/0.20      20              06 F669 0087     2
Fa0/0         211.20.188.104  Fa1/0.19      203.67.208.20   06 DD21 0019    11
```

圖 5 由 Cisco Router dump 出之 netflow data source

3.2.2 中介檔案

由於 Data Source 部份是文字檔案(.txt)，有些資訊並非我們所要的（圖 5 中框框以外的部份)，必須先將之刪去，留下 netflow 格式部份，然後將其存成 Excel 格式的檔案。

檔案格式如下：



	A	B	C	D	E	F	G	H
1	Fa0/0	59	Fa1/0.20	20	06	0019	D288	25
2	Fa0/0	19	Local	19	06	B4D3	0286	2
3	Fa0/0	20	Fa1/0.19	20	11	F3DB	0035	1
4	Fa0/0	18	Fa1/0.19	20	06	0ED2	0019	1
5	Fa0/0	21	Fa1/0.20	20	06	EDFB	0019	5
6	Fa0/0	20	Fa1/0.20	20	06	0D87	01BD	2
7	Fa0/0	59	Fa1/0.19	20	06	FD33	0087	2
8	Fa0/0	20	Fa1/0.19	20	06	10DD	0087	2
9	Fa0/0	19	Local	19	06	00B3	2B99	6
10	Fa0/0	65	Fa1/0.20	20	06	0019	8708	801
11	Fa0/0	12	Fa1/0.20	20	06	CEA9	0019	1

圖 6 NetFlow data source 轉入資料庫過程中之 Excel 中介檔案格式

由於時間部份並不在我們所取得的 netflow 完整的格式裡，所以日期欄位是最後再使用 SQL 指令去 update。

SQL update 指令（範例）：

```
UPDATE       netflow

SET           StartTime = CONVERT(datetime, '2007-05-18 11:05:39'),

               EndTime = CONVERT(datetime, '2007-05-18 11:05:49')

WHERE       (StartTime IS NULL)
```

3.2.3 Database / Data warehouse 設計

3.2.3.1 netflow 資料庫之資料表格關連圖

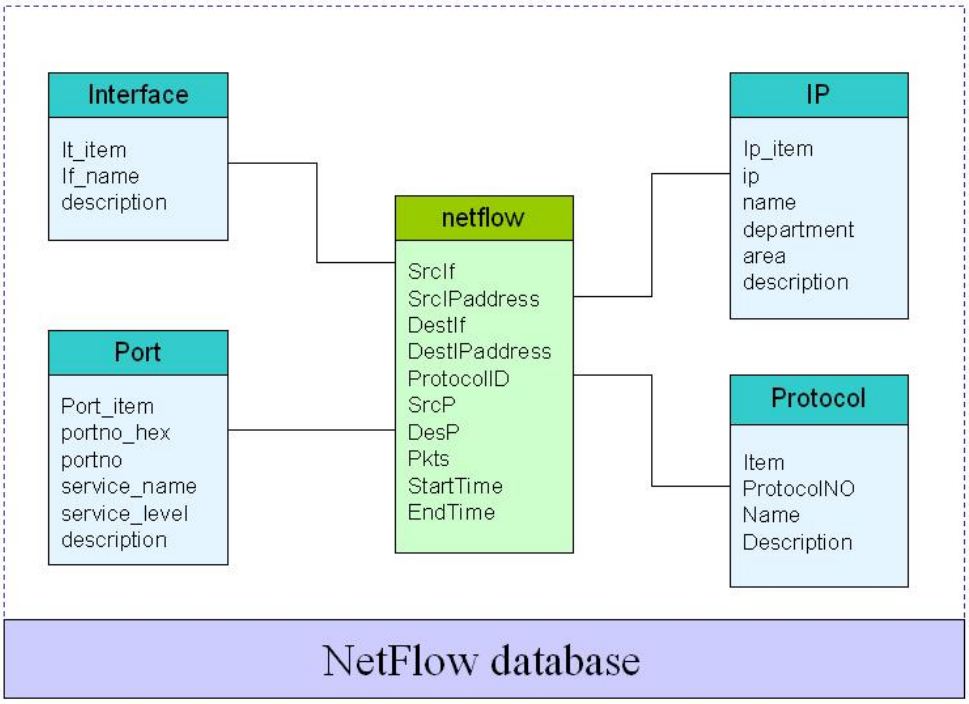


圖 7 netflow 資料庫之資料表關連圖

netflow 是會經常變動的資料表，一旦路由器收集到了一定程度的流量資料後，不管是用自動或是人工的方式，這些流量資料將被塞入 netflow 資料表中，所以它是會不斷地增加的資料表。另外還有用來記錄所有路由器介面的 Interface 資料表，負責記錄 IP 基本資料的 IP 資料表，記錄常用到的 Port Service 的 Port 資料表，以及用來記錄封包所使用的網路協定之 Protocol 資料表。下一章節將會對以上這些資料表的詳細內容與用途做說明。

3.2.3.2 table schema

NetFlow 資料庫內有五個重要的資料表格，分別為 netflow、IP、Port、Interface 以及 Protocol。

(a)table name : netflow

資料表格主要功能：主要儲存由路由器 dump 出來的 NetFlow 原始資料，包含了 11 個欄位。

表 3 netflow table schema

欄位名稱	資料型態	資料長度	是否允許 Null 值	欄位說明
SrcIf	Varchar	30		來源主機之流入介面
SrcIPaddress	Varchar	15		來源主機之 IP 位址
DestIf	Varchar	30		目的主機之流入介面
DestIPaddress	Varchar	15		目的主機之 IP 位址
ProtocolID	Int	4		使用之通訊協定
SrcP	Nvarchar	6		來源主機所使用的埠號
DesP	Nvarchar	6		目的主機所使用的埠號
Pkts	Numeric	8		封包數量
StartTime	Datetime	8		此筆 flow 開始累計的時間
EndTime	Datetime	8	V	此筆 flow 結束累計的時間

(b)table name : IP

資料表格主要功能：主要儲存網路位址 IP 所屬人之相關基本資料，包含了 6 個欄位。

表 4 IP table schema

欄位名稱	資料型態	資料長度	是否允許 Null 值	欄位說明
Ip_item	Numeric	9		流水號
Ip	Varchar	15		IP 位址
Name	Varchar	100	V	使用此 IP 之使用者名稱
Department	Varchar	100	V	使用者所屬部門
Area	Varchar	10	V	1 : 公司內部 (192.168.xxx.xxx) (139.175.238.xxx) 0 : 公司外部
Description	Varchar	150	V	備註用

(c)tablename : Port

資料表格主要功能：主要儲存常用的 Network Port service 之基本資料，包含了 6 個欄位。

表 5 Port table schema

欄位名稱	資料型態	資料長度	是否允許 Null 值	欄位說明
Port_item	Numeric	9		流水號
Portno_hex	Nvarchar	6		Port 之十六進位表示法 例如：0050
Portno	Int	4		Port 之十進位表示法 例如：80
Service_name	Varchar	100	V	該 Port 所屬之服務名稱 例如：http
Service_level	Varchar	50	V	記錄該 service 所使用之協定 例如：udp 或 tcp
Description	Varchar	250	V	備註用

(d)tablename : Interface

資料表格主要功能：主要儲存公司現有路由器中所設定之所有 Interface 之基本資料，包含了 3 個欄位。

表 6 Interface table schema

欄位名稱	資料型態	資料長度	是否允許 Null 值	欄位說明
If_item	Numeric	9		流水號
If_name	Nvarchar	30		路由器之 Interface 名稱
Description	Nvarchar	100	V	備註用

(e)tablename : Protocol

資料表格主要功能：主要儲存網路封包常用之 Protocol 其網路協定的編號與名稱，包含了 4 個欄位。

表 7 Protocol table schema

欄位名稱	資料型態	資料長度	是否允許 Null 值	欄位說明
Item	Numeric	9		識別碼
ProtocolNO	Int	4		網路協定所使用的協定編號 (例如：6；代表的就是 tcp 的協定編號)
Name	Varchar	150		網路協定的名稱 (例如：tcp)
Description	Varchar	200	V	備註用

3.2.3.3 dw_netflow 資料倉儲與各維度量值之設計

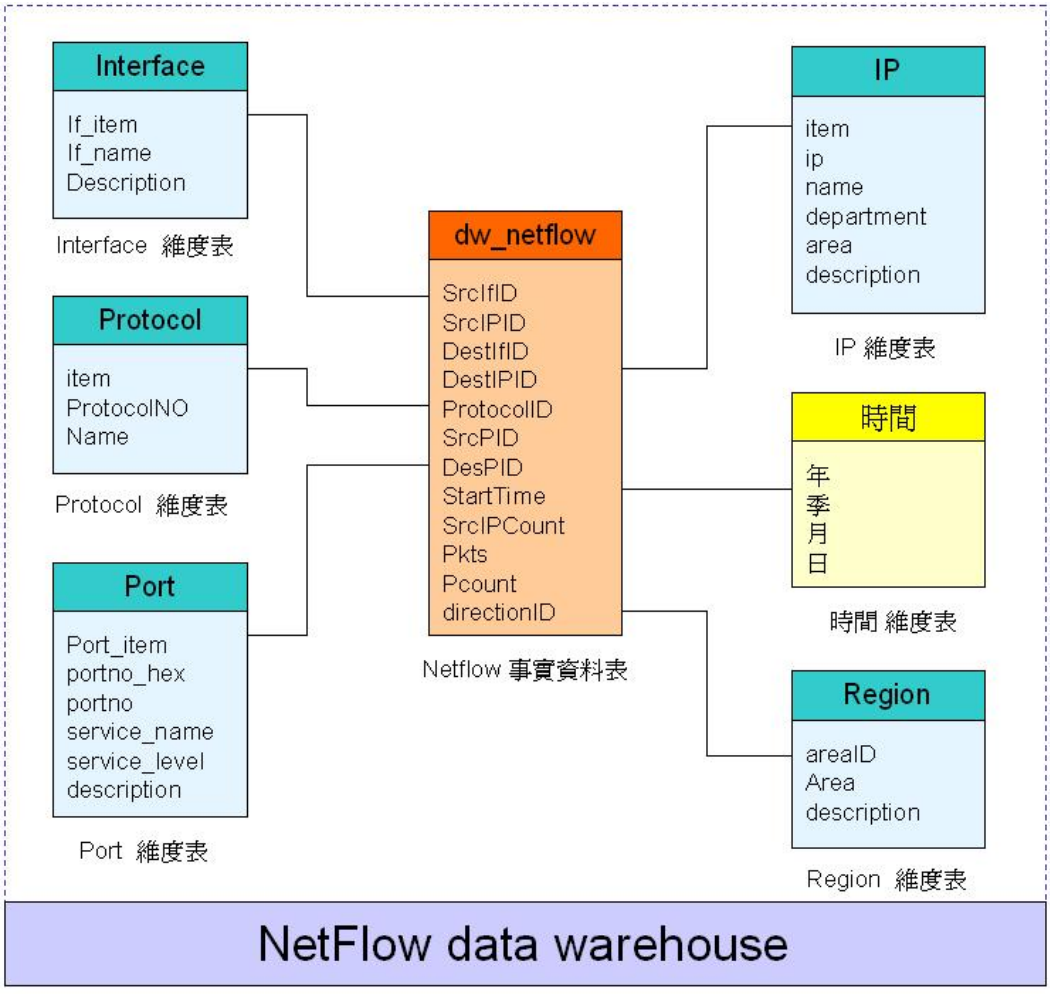


圖 8 dw_netflow 資料倉儲各維度架構圖

在 dw_netflow 資料倉儲裡有一個名為 dw_netflow 的事實資料表，用來儲存 netflow 資料表內經過整理過後的資料，insert_exdw_netflow store procedure 就是設計專門用來將 netflow 內的資料轉化為 dw_netflow 格式的程式；另外還有 6 個維度資料表，分別為 Interface、Protocol、Port、IP、時間以及 Region，(如圖 8 所示)。

Interface 的 If_item 欄位是用來和 dw_netflow 資料表內的 SrcIfID 欄位做關連的。
Protocol 的 item 欄位是用來和 dw_netflow 資料表內的 ProtocolID 欄位做關連。
Port 的 Port_item 欄位是用來和 dw_netflow 資料表內的 DesPID 欄位做關連。
IP 的 item 欄位是用來和 dw_netflow 資料表內的 DestIPID 欄位做關連。
Region 的 areaID 欄位是用來和 dw_netflow 資料表內的 directionID 欄位做關連。
時間維度則是使用 dw_netflow 資料表內的 StartTime 做為依據。

3.2.3.3.3 事實資料表設計

(a)事實資料表名稱：dw_netflow

資料來源：ex_netflow

索引欄位：SrcIfID，來自 Interface 資料表

SrcIPID，來自 IP 資料表

DestIfID，來自 Interface 資料表

DestIPID，來自 IP 資料表

ProtocolID，來自 Protocol 資料表

SrcPID，來自 Port 資料表

DesPID，來自 Port 資料表

StartTime，來自 ex_netflow 資料表

Fcount，來自 ex_netflow 資料表

Pkts，來自 ex_netflow 資料表

Pcount，來自 ex_netflow 資料表

DirectionID，來自 area 資料表

SQL 敘述：使用 Store Procedure 產生

Store Procedure Name : insert_exdw_netflow

Insert_exdw_netflow Store Procedure 程式碼：(請參考附錄一)

下面的表格，列出了 dw_netflow 事實資料表的資料綱要以及主索引鍵欄位。

表 8 dw_netflow table schema

欄位名稱	資料型態	資料長度	是否允許null值	欄位說明
SrcIfID	Numeric	9		.來自 Interface 資料表 .來源 Interface 的代號
SrcIPid	Numeric	9		.來自 IP 資料表 .來源 IP 的代號
DestIfID	Numeric	9		.來自 Interface 資料表 .目的地 Interface 的代號
DestIPid	Numeric	9		.來自 IP 資料表 .目的地 IP 的代號
ProtocolID	Numeric	9		.來自 Protocol 資料表 .使用通訊協定的代號
SrcPid	Numeric	9		.來自 Port 資料表 .來源 Port 的代號
DesPid	Numeric	9		.來自 Port 資料表 .目的地 Port 的代號
StartTime	Datetime	8		資料流的啟始時間
Fcount	Int	4		資料流的 flow 值
Pkts	Numeric	9		資料流的 packet 值
Pcount	Int	4		預設值皆為 1，方便統計用
DirectionID	Int	4		計錄此資料流為流入或流出； 0：流出 1：流入

3.2.3.3.4 維度資料表設計

(a) 維度資料表名稱： Port

請參考下面的表格，列出 port 維度資料表的綱要以及主索引鍵欄位。

表 9 Port 維度資料表資料欄位

欄位名稱	資料型態	資料長度	是否允許null值	欄位說明
Port_item	Numeric	9		.主索引鍵 .流水號
Portno_hex	Varchar	4		Port之十六進位表示法 例如：0050
Portno	Int	4		Port之十進位表示法 例如：80
Service_name	Varchar	100	V	該Port所屬之服務名稱 例如：http
Service_level	Varchar	50	V	記錄該service所使用之協定 例如：udp 或 tcp
Description	Varchar	250	V	備註用

(b)維度資料表名稱： Interface

請參考下面的表格，列出 Interface 維度資料表的綱要以及主索引鍵欄位。

表 10 Interface 維度資料表資料欄位

欄位名稱	資料型態	資料長度	是否允許null值	欄位說明
If_item	Numeric	9		.主索引鍵 .流水號
If_name	Nvarchar	30		路由器之Interface名稱
Description	Nvarchar	100	V	備註用

(c)維度資料表名稱： IP

請參考下面的表格，列出 IP 維度資料表的綱要以及主索引鍵欄位。

表 11 IP 維度資料表資料欄位

欄位名稱	資料型態	資料長度	是否允許null值	欄位說明
Ip_item	Numeric	9		.主索引鍵 .流水號
Ip	Varchar	15		使用者的 IP 位址
Name	Varchar	100	V	使用此 IP 的使用者名稱
Department	Varchar	100	V	使用者所屬部門
Area	Varchar	10	V	1：公司內部 (192.168.xxx.xxx) (139.175.238.xxx) 0：公司外部
Description	Varchar	150	V	備註用

(d)維度資料表名稱： Protocol

請參考下面的表格，列出 Protocol 維度資料表的綱要以及主索引鍵欄位。

表 12 Protocol 維度資料表資料欄位

欄位名稱	資料型態	資料長度	是否允許null值	欄位說明
Item	Numeric	9		.主索引鍵 .流水號
ProtocolNO	Int	4		網路協定所使用的協定編號 (例如：6；代表的就是 tcp 的協定編號)
Name	Varchar	150		網路協定的名稱 (例如：tcp)
Description	Varchar	200	V	備註用

(e)維度資料表名稱： Region

資料表主要功能：主要提供 flow 之流入流出及代表公司內部或外部之 IP 的代號，包含了 3 個欄位。

請參考下面的表格，列出 Region 維度資料表的綱要以及主索引鍵欄位。

表 13 Region 維度資料表資料欄位

欄位名稱	資料型態	資料長度	是否允許null值	欄位說明
AreaID	Int	4		.主索引鍵 0：流出的資料／公司外部 1：流入的資料／公司內部
Area	Char	30		OUT：流出的資料 IN：流入的資料
Description	Char	50	V	備註用

3.2.3.3.5 資料移轉部份

設計每日由系統自動將 source database 轉至 data warehouse。

dw_netflow 資料表： 將 Store procedure 設定每 10 分鐘自動由 netflow table 將近 10 分鐘的 raw data 轉入 dw_netflow table。

其它的資料表： 包括了 Port、Interface、IP、Region、Protocol 等五個資料表，由於資料的變動性較小，管理者可以自行透過資料庫軟體做新增或修改的動作，或是撰寫新增修改的網頁介面，方便更新與管理。

3.2.4 建立 Cube

由於事實資料表與維度資料表都已經設計好了，要實做 Cube 就變得非常的容易。本論文使用 Microsoft SQL Server 所提供的 Analysis Services 工具來做為我們資料倉儲的線上分析處理的決策支援工具，我們將透過它來建立 Cube。

3.2.5 OLAP Report

前面我們使用了 Analysis Services 建立了一個強大的後端分析系統，再來就是提供使用者透過 Internet/Intranet 的技術來存取並呈現線上分析的結果，本論文將使用 Strategy Companion 的 Analyzer 2005 來做為 OLAP 前端分析的工具。

Analyzer 2005 提供多種形式的分析表，我們可以依需要，將資料來源以拖拉的方式將資料拖曳進報表中，即可簡單地完成一張分析表，透過系統提供的任意維度間階層組合交叉分析功能，可更快速地達到不同角度來分析資料的目的。另外，若希望以統計圖的方式呈現，我們也可以利用建立統計圖形的功能迅速地產生對應資料的統計圖形。

本論文希望透過 Analyzer 2005 來製作一個“目的地 Port 流量統計與監測“的分析報表，透過報表來設定我們使用決策樹所分析出來的臨界值，透過此臨界值來提醒管理者網路可能出現了某些異常狀況，同時藉由此報表來為有興趣的量值設定資料鑽研（DrillDown）的功能，以及設定決策分析圖的樣式。

(a) 報表名稱：目的地 Port 流量統計與監測報表(area)

報表功能：針對目的地 Port 之流量加以統計，並對該值設定蠕蟲病毒的臨界值，借以提醒網路管理者，流量異常情況的發生

使用 cube 名稱：cube_dwnflow_sifip_dport_area

選取維度：時間、port no 、in or out、ip、protocol name、start time

選取量值：fcount、pkts

3.2.6 Customize Web AP

本論文為網路流量統計設計了一些功能較簡單的報表，最主要的用意是給網路管理者一個想法，免費的工具固然好用，但有時候卻非完全盡如人意。這時候，做個自行開發流量統計的系統也是個好主意，如此對於功能需求的彈性度會更大些。圖 9 為自行開發流量統計系統 Netflow On Web（NOW）之開發環境。

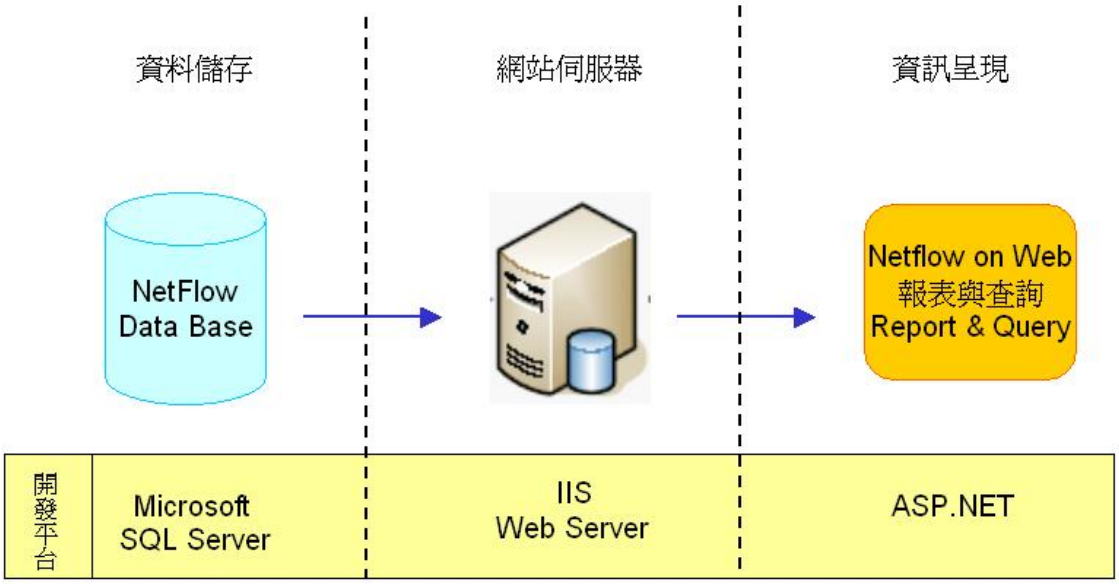


圖 9 Netflow On Web 開發環境

Netflow On Web（NOW）是本論文自行開發之網路流量統計網頁程式，主要是依據 netflow 資料庫中的每日流量資料產生一些統計性的查詢報表，供網路管理者查詢。

Netflow On Web 開發環境：

開發軟體：ASP.NET

開發平台：.NET Framework 1.1 以上

網站伺服器：IIS

資料庫：Microsoft SQL Server 2000

Netflow On Web 主要功能：

(a)Summary：列出每日流量之每筆 flow 的詳細資料。

(b)每日流入流出量：依據 flow 列出每日流入流出之總量

(c)個別 IP 當日之流量排行：依據使用者選定之日期，列出當日各個 IP 之流量，並依流量做降冪的排行。

(d)每個 Port 當日之流量排行：依據使用者選定之日期，列出當日各個 Port 之流量，並依流量做降冪的排行。

(e)疑似病毒感染之 IP 列表：依據各種蠕蟲病毒所界定的流量臨界值，來判斷疑似感染蠕蟲病毒之 IP 列表。

4. 系統實作

4.1 Decision Tree 在流量(flow)臨界值之分析

本論文將使用決策樹分析法來決定 netflow 流量的臨界值，以判別是否遭受病毒的攻擊。在決定出幾種病毒的臨界值後，後面章節的系統設計與實作部份都將使用我們所定義出來的臨界值。

由於網路上多數的病毒攻擊方式都採用 DDoS，所以我們將焦點放在 flow 的封包量上；受攻擊的次數愈多，封包量就會愈大。

流量臨界值分析環境：Microsoft Analysis Services (SQL Server 2000)。

4.1.1 MSBlast 病毒臨界值分析

在我們所擁有的歷史資料裡，依據 MSBlast 病毒的特性，我們將 pktsize 與 flowcount 以及 isWorm 這三個欄位放入決策樹中，讓決策樹來產生合理的臨界值。

(a) 決策樹所依據的資料表名稱：worm_msblast

(b) worm_msblast 資料表資料欄位

表 14 worm_msblast table schema

欄位名稱	資料型態	資料長度	是否允許null值	欄位說明
Item	Numeric	9		流水號
IP	Varchar	15		flow 之來源 IP 位址
Protocol	Varchar	10		flow 所使用之通訊協定
Port	Numeric	9		flow 之目的地 Port
Pktsize	Numeric	9		Packet 的大小(bytes)
Pktscount	Numeric	9		Packet 的數量
Flowcount	Numeric	9		此筆記錄在單位時間內，出現的 flow 次數
wormType	Varchar	20		感染病毒的形態
IsWorm	Char	10		此筆記錄是否感染病毒 0：沒有感染病毒 1：已感染病毒
ddate	Datetime	8		此筆 flow 所產生之日期時間

(c) 資料筆數

資料總筆數：719

中毒資料筆數：163

沒有中毒資料筆數：556

(b)決策樹所需之值

Predictable columns：isWorm

Input columns：pktsize、flowcount

(c)決策樹產生之結果

依據以上的條件值，決策樹產生的結果如下：

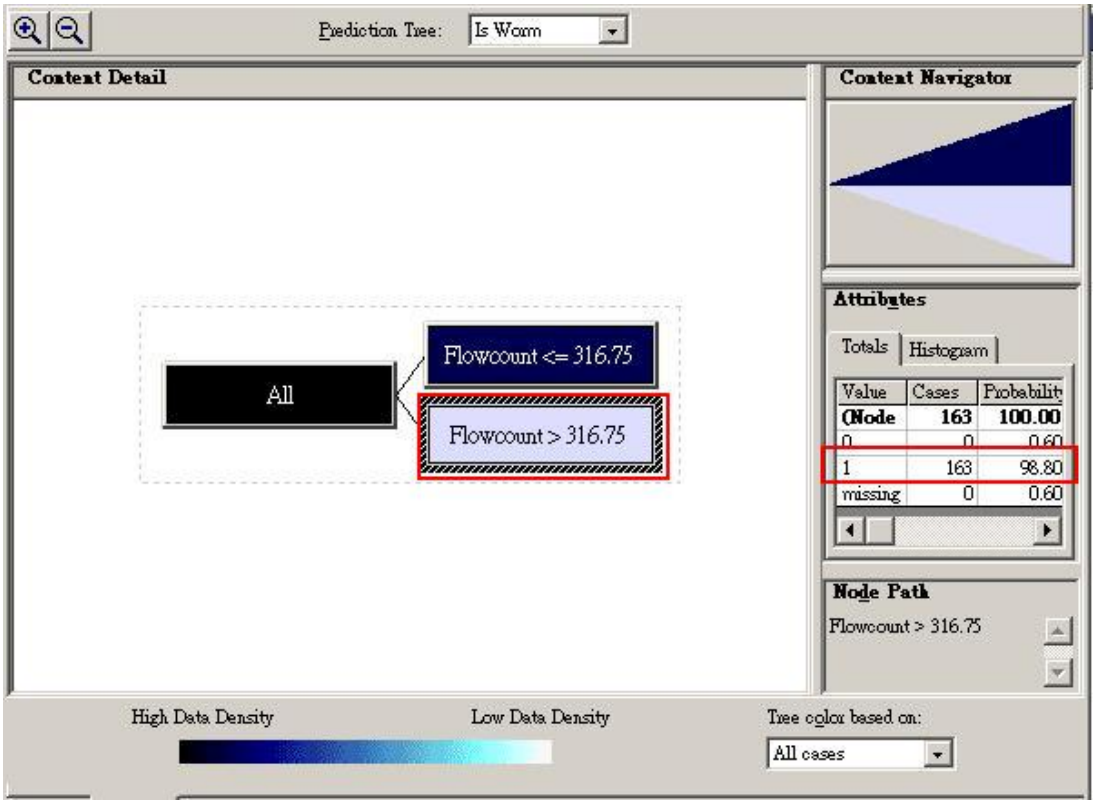


圖 10 使用決策樹產生 MSBlast 病毒之決策樹狀圖 (1)

圖 10 依據現有資料庫中的資料說明了 flowcount >= 316.75(單位時間內 flow 的數量)，有 98.8%的機會受到 MSBlast 病毒的攻擊。而另一個子節點則說明了若 flowcount <= 317.75 有 99.64%的機會不會受到此種病毒的攻擊。

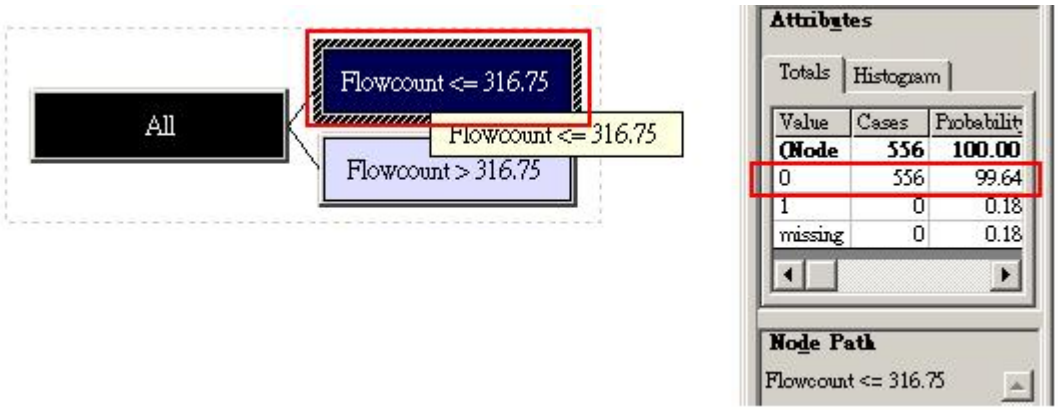


圖 11 使用決策樹產生 MSBlast 病毒之決策樹狀圖 (2)

(e) MSBlast 臨界值的設定：flowcount = 330

如圖 11 所示，由決策樹所決定出來的 flowcount 為 316.75，為方便取整數，我們將值設定為 330，並將此臨界值進一步的做驗証，在驗証中若發現有不適合的現象，我們再來修正此臨界值。

4.1.2 CodeRed 病毒臨界值分析

在我們所擁有的歷史資料裡，依據 CodeRed 病毒的特性，我們將 pktsize 與 flowcount 以及 isWorm 這三個欄位放入決策樹中，讓決策樹來產生合理的臨界值。

(a)決策樹所依據的資料表名稱：worm_codered

(b)worm_codered 資料表資料欄位

表 15 worm_codered table schema

欄位名稱	資料型態	資料長度	是否允許null值	欄位說明
Item	Numeric	9		流水號
IP	Varchar	15		flow之來源 IP位址
Protocol	Varchar	10		flow所使用之通訊協定
Port	Numeric	9		flow之目的地Port
Pktsize	Numeric	9		Packet的大小(bytes)
Pktscount	Numeric	9		Packet的數量
Flowcount	Numeric	9		此筆記錄在單位時間內,出現的flow次數
wormType	Varchar	20		感染病毒的形態
IsWorm	Char	10		此筆記錄是否感染病毒 0：沒有感染病毒 1：已感染病毒
ddate	Datetime	8		此筆flow所產生之日期時間

(c) 資料筆數

資料總筆數：523

中毒資料筆數：40

沒有中毒資料筆數：483

(c)決策樹所需之值

Predictable columns：isWorm

Input columns：pktsize、flowcount

(d)決策樹產生之結果

依據以上的條件值，決策樹產生的結果如下：

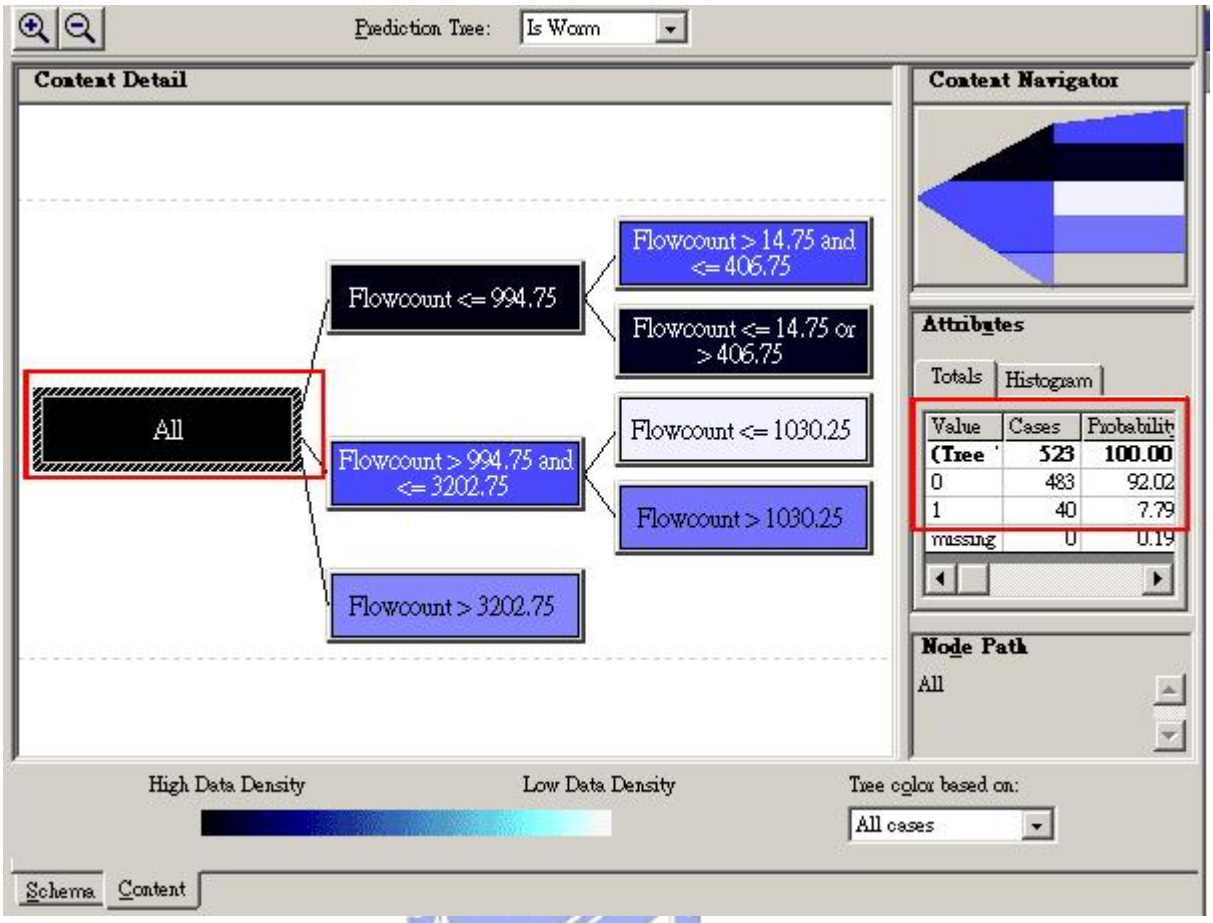


圖 12 使用決策樹產生 CodeRed 病毒之決策樹狀圖

如圖 12 所示，在“A11”根節點下分出了二層的子節點，每一個子節點其所佔的資料筆數與機率值如下表：

表 16 CodeRed 決策樹狀圖中每個子節點所佔之資料筆數與機率值

Sub tree contents	Isworm 值 / 資料筆數	probability
Flowcount <= 994.75	0 / 447 1 / 6	98.25% 1.54%
Flowcount > 14.75 and <= 406.75	0 / 38 1 / 6	82.98% 14.89%
Flowcount <= 14.75 or > 406.75	0 / 409 1 / 0	99.51% 0.24%
Flowcount > 994.75 and <= 3202.75	0 / 36 1 / 7	80.43% 17.39%
Flowcount <= 1030.25	0 / 5 1 / 7	40.00% 53.33%
Flowcount > 1030.25	0 / 31 1 / 0	94.12% 2.94%
Flowcount > 3202.75	0 / 0 1 / 27	33% 93.33%

(e)CodeRed 臨界值的設定：flowcount = 1000

由表 16 的樹狀圖表中我們可以看到”flowcount <= 994.75”子節點佔了 453 筆的資料筆數，不過大部分都是屬於 isworm = 0(沒有中毒的資料)，所以我們再看看另一個子節點”flowcount > 994.75 and <= 3202.75”，它的總筆數有 43 筆，屬於中毒的筆數有 7 筆，沒有中毒的筆數有 36 筆，和 flowcount <= 994.75”此子節點總筆數加起來共有 490 筆，佔絕大多數的資料了，所以，我們決定以將 994.75 取整數 1000 為 CodeRed flowcount 的臨界值。

4.1.3 一般 DDos 攻擊病毒臨界值分析

在我們所擁有的歷史資料裡，依據 CodeRed 病毒的特性，我們將 pktsize 與 flowcount 以及 isWorm 這三個欄位放入決策樹中，讓決策樹來產生合理的臨界值。

(a)決策樹所依據的資料表名稱：worm_ddos

(b)資料表資料欄位

表 17 worm_ddos table schema

欄位名稱	資料型態	資料長度	是否允許null值	欄位說明
Item	Numeric	9		流水號
IP	Varchar	15		flow之來源 IP位址
Protocol	Varchar	10		flow所使用之通訊協定
Port	Numeric	9		flow之目的地Port
Pktsize	Numeric	9		Packet的大小(bytes)
Pktscount	Numeric	9		Packet的數量
Flowcount	Numeric	9		此筆記錄在單位時間內,出現的flow次數
wormType	Varchar	20		感染病毒的形態
IsWorm	Char	10		此筆記錄是否感染病毒 0：沒有感染病毒 1：已感染病毒
ddate	Datetime	8		此筆flow所產生的日期時間

(c)資料筆數

資料總筆數：482

中毒資料筆數：392

沒有中毒資料筆數：90

(d)決策樹所需之值

Predictable columns：isWorm

Input columns：pktsize、flowcount

(e)決策樹產生之結果

依據以上的條件值，決策樹產生的結果如下：

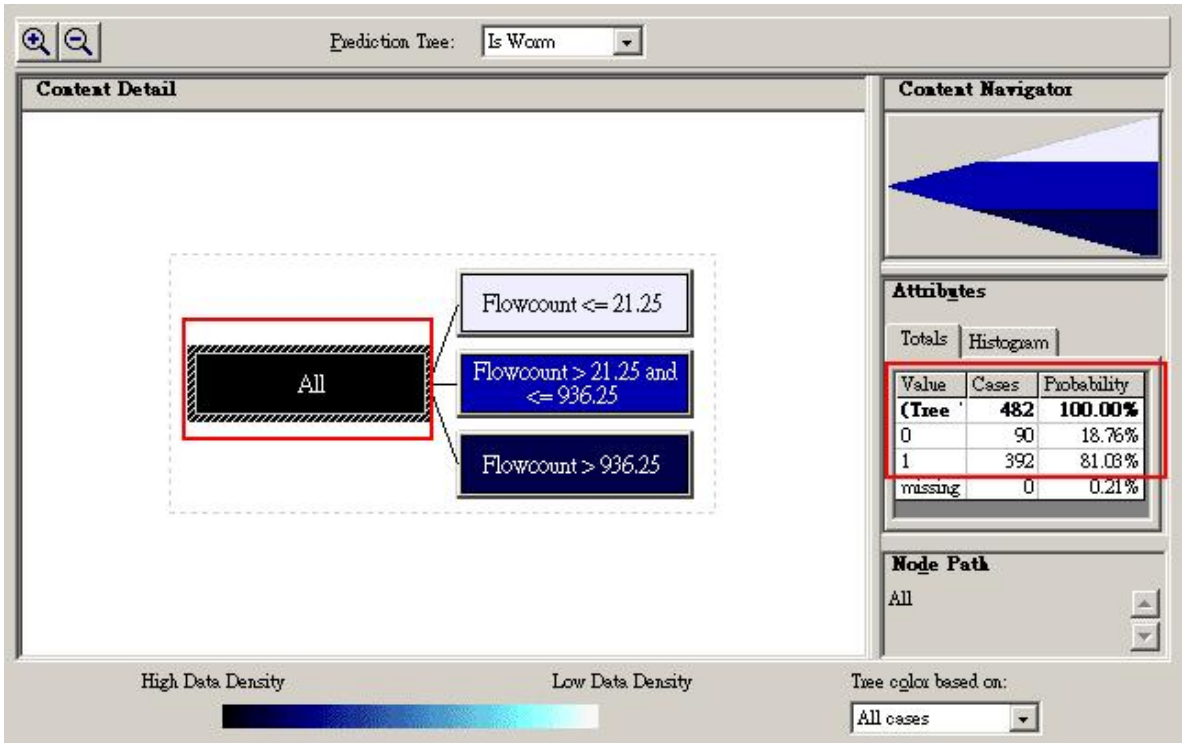


圖 13 使用決策樹產生 DDoS 病毒之決策樹狀圖

如圖 13 所示，在“All”根節點下分了三個子節點，每一個子節點其所佔的資料筆數與機率值請看表 18：

表 18 DDoS 決策樹狀圖中每個子節點所佔之資料筆數與機率值

Sub tree contents	Isworm 值 / 資料筆數	probability
Flowcount <= 21.25	0 / 22 1 / 0	92.00% 4.00%
Flowcount > 21.25 and <= 936.25	0 / 60 1 / 99	37.65% 61.73%
Flowcount > 936.25	0 / 8 1 / 293	2.96% 97.71%

(f)DDoS 臨界值的設定：flowcount = 30

由表 18 的樹狀圖表中我們可以看到”flowcount <= 21.25”子節點 isworm=0 的筆數佔了 22 筆(沒有中毒的資料),所以我們再看看另一個子節點”flowcount > 21.25 and <= 936.25”，它的總筆數有 159 筆，屬於中毒的筆數有 99 筆，沒有中毒的筆數有 60 筆，和 flowcount <= 21.25”此子節點沒有中毒的筆數加總起來已經有 82 筆，對沒有中毒的筆數來說已佔絕大多數的資料了，所以，我們決定

將 21.25 取整數 30 為 DDoS flowcount 的臨界值。

4.2 流量(flow)臨界值之驗証

在 4.1 節中我們為幾個俱攻擊性的病毒使用了決策樹模型分別為其單位時間內 netflow 之 flow 數量決定了臨界值；

表 19 決策樹模型分析出各病毒種類之所屬 flow 臨界值一欄表

病毒種類	單位時間內 flow 數量之臨界值
MSBlast	330
CodeRed	1000
DDoS 攻擊	30

以下我們將依據上面的臨界值來做驗証，以確保臨界值是有意義的。

4.2.1 驗証 MSBlast 病毒

資料內容包含了已確定中毒及未中毒之 flow 資料；

- 驗証資料日期：2004/04/09 ~ 2004/04/12
- 驗証臨界值：330
- 驗証資料方式：

  使用 SQL command 至資料庫中將歷史資料撈出，再做進一步的確認 SQL Command 與結果：

```
select ip,port,pktsize,flowcount, isworm,ddate
from worm_msblast
where item >= 1000
order by flowcount
```

	ip	port	pktsize	flowcount	isworm	ddate
67	163.26.148.69	135	48	297	0	2004-04-09 15:52:17.000
68	163.26.148.73	135	48	305	0	2004-04-10 11:32:11.000
69	163.26.107.144	135	48	305	0	2004-04-12 00:22:01.000
70	163.26.148.60	135	48	307	0	2004-04-10 11:32:11.000
71	163.26.148.113	135	48	313	0	2004-04-09 16:52:11.000
72	163.26.148.117	135	48	330	0	2004-04-09 16:32:26.000
73	163.26.107.144	135	48	336	1	2004-04-12 01:32:01.000
74	163.26.148.46	135	48	343	1	2004-04-09 15:52:17.000
75	163.26.148.30	135	48	402	1	2004-04-10 11:32:11.000
76	163.26.148.82	135	48	417	1	2004-04-09 15:52:17.000
77	163.26.107.144	135	48	419	1	2004-04-12 03:12:00.000
78	163.26.148.46	135	48	448	1	2004-04-09 16:52:11.000
79	163.26.148.98	135	48	468	1	2004-04-10 11:42:11.000
80	163.26.148.113	135	48	471	1	2004-04-09 16:32:26.000
81	163.26.148.104	135	48	476	1	2004-04-09 17:02:10.000
82	163.26.148.113	135	48	530	1	2004-04-09 16:42:12.000
83	163.26.148.43	135	48	548	1	2004-04-10 12:02:14.000
84	163.26.148.100	135	48	636	1	2004-04-09 17:02:10.000
85	163.26.148.64	135	48	637	1	2004-04-09 16:32:26.000
86	163.26.148.49	135	48	689	1	2004-04-10 11:52:14.000
87	163.26.148.46	135	48	708	1	2004-04-09 16:32:26.000

圖 14 MSBlast 資料驗証

如圖 14 所示，在資料驗証上之說明：第 72 筆以後的資料其 flowcount 值皆大於 330，isWorm 欄位皆為 1(表示遭受病毒攻擊)，所以 330 是可以信任的臨界值。

4.2.2 驗証 CodeRed 病毒

資料內容包含了已確定中毒及未中毒之 flow 資料；

- 驗証資料日期：2004/04/15 ~ 2004/04/22
- 驗証臨界值：1000
- 驗証資料方式：

  使用 SQL command 至資料庫中將歷史資料撈出，再做進一步的確認 SQL Command 與結果。

```
select * from worm_codered
where item >= 1000
order by flowcount
```

	item	ip	protocol	port	pktsize	pktscount	flowcount	wormType	isWorm	ddate
38	1036	163.26.194.36	tcp	80	144	11	3	CodeRed	0	2004-04-17 22:22:02.000
39	1037	163.26.194.36	tcp	80	144	11	3	CodeRed	0	2004-04-17 22:12:02.000
40	1038	163.26.194.36	tcp	80	144	11	3	CodeRed	0	2004-04-17 22:02:03.000
41	1039	163.26.194.36	tcp	80	144	11	3	CodeRed	0	2004-04-17 21:52:02.000
42	1040	163.26.194.36	tcp	80	144	11	3	CodeRed	0	2004-04-17 21:42:02.000
43	1041	163.26.194.36	tcp	80	144	12	4	CodeRed	0	2004-04-17 21:32:01.000
44	1015	163.26.194.36	tcp	80	144	12	4	CodeRed	0	2004-04-18 02:02:01.000
45	1049	163.26.107.253	tcp	80	144	46	15	CoreRed	1	2004-04-22 16:02:04.000
46	1046	163.26.151.160	tcp	80	144	1388	462	CodeRed	0	2004-04-16 16:02:07.000
47	1043	163.26.151.160	tcp	80	144	2832	944	CodeRed	0	2004-04-16 16:32:08.000
48	1044	163.26.151.160	tcp	80	144	2866	955	CodeRed	0	2004-04-16 16:22:04.000
49	1045	163.26.151.160	tcp	80	144	2899	966	CodeRed	0	2004-04-16 16:12:05.000
50	1042	163.26.151.160	tcp	80	144	2900	966	CodeRed	0	2004-04-16 16:42:03.000
51	1056	163.26.107.253	tcp	80	144	3001	1000	CoreRed	1	2004-04-22 15:32:07.000
52	1050	163.26.107.253	tcp	80	144	3048	1016	CoreRed	1	2004-04-22 15:56:03.000
53	1051	163.26.107.253	tcp	80	144	3048	1016	CoreRed	1	2004-04-22 15:52:07.000
54	1052	163.26.107.253	tcp	80	144	3048	1016	CoreRed	1	2004-04-22 15:51:11.000
55	1053	163.26.107.253	tcp	80	144	3048	1016	CoreRed	1	2004-04-22 15:50:35.000
56	1054	163.26.107.253	tcp	80	144	3092	1030	CoreRed	1	2004-04-22 15:49:38.000
57	1055	163.26.107.253	tcp	80	144	3092	1030	CoreRed	1	2004-04-22 15:42:08.000

圖 15 CodeRed 資料驗証

如圖 15 所示，在 DDoS 資料驗証之說明：第 50 筆以後的資料其 flowcount 值皆大於 1000，isWorm 欄位皆為 1(表示遭受病毒攻擊)，所以 1000 是可以信任的臨界值。

4.2.3 驗証 DDoS 攻擊

資料內容包含了已確定中毒及未中毒之 flow 資料；

- 驗証資料日期：2003/10/02 ~ 2003/11/12、2004/04/23 ~ 2004/08/31
- 驗証臨界值：30
- 驗証資料方式：

  使用 SQL command 至資料庫中將歷史資料撈出，再做進一步的確認 SQL Command 與結果。

```
select ip,port,pktsize,flowcount, isworm,ddate
from worm_mix
where item >=1000
order by flowcount
```

	ip	port	pktsize	flowcount	isworm	ddate
1	163.26.156.65	NULL	144	2	0	2004-04-23 10:52:11.000
2	163.26.156.65	NULL	144	3	0	2004-04-23 11:32:13.000
3	163.26.156.65	80	144	4	0	2004-04-23 08:12:04.000
4	163.26.156.65	80	144	5	0	2004-04-23 08:02:03.000
5	163.26.182.3	137	78	41	1	2004-08-31 14:32:23.000
6	163.26.182.3	137	78	42	1	2004-08-31 14:52:11.000
7	163.26.182.3	137	78	53	1	2004-08-31 14:42:10.000
8	163.26.182.3	137	78	57	1	2004-08-31 14:02:10.000
9	163.26.182.3	137	78	58	1	2004-08-31 14:22:09.000
10	163.26.182.3	137	78	60	1	2004-08-31 13:32:05.000
11	163.26.182.3	137	78	62	1	2004-08-31 14:12:10.000
12	163.26.182.3	137	78	72	1	2004-08-31 13:52:09.000
13	163.26.163.220	137	78	73	1	2003-10-02 17:12:08.000
14	163.26.182.3	137	78	82	1	2004-08-31 13:42:07.000
15	163.26.163.210	NULL	144	110	0	2004-04-27 04:12:01.000
16	163.26.107.253	NULL	144	248	1	2004-04-23 14:22:09.000
17	163.26.107.253	NULL	144	263	1	2004-04-23 14:42:09.000
18	163.26.107.253	80	144	277	1	2004-04-23 10:22:11.000
19	163.26.163.182	137	78	292	1	2003-11-11 16:22:09.000
20	163.26.163.182	137	78	425	1	2003-10-02 16:22:05.000
21	163.26.163.182	137	78	428	1	2003-11-12 10:02:10.000
22	163.26.107.253	NULL	144	431	1	2004-04-23 14:32:22.000
23	163.26.163.182	137	78	524	1	2003-11-12 13:22:09.000
24	163.26.163.210	NULL	144	555	0	2004-04-27 03:12:01.000
25	163.26.182.28	137	78	809	1	2003-10-10 09:22:02.000

圖 16 DDoS 資料驗証

如圖 16 所示，在 DDoS 資料驗証之說明：第 5 筆以後的資料其 flowcount 值皆大於 30，isWorm 欄位皆為 1(表示遭受病毒攻擊)，所以 30 是可以信任的臨界值。

4.3 OLAP 系統實作

本論文系統實作部份的資料將以 NetFlow dump 出來之原始資料為基礎，基於此資料將其應用於線上即時分析與自行開發之網頁應用程式上。

4.3.1 實作環境

作業系統 : Windows 2000 Server

資料庫 : Microsoft SQL Server 2000

網站伺服器：IIS

程式開發工具 : SQL Language 、ASP.NET

商業智慧軟體 : MS Analysis Services、Analyzer 2005

在我們提供的整個系統實作部份，首先，要將 NetFlow 的原始資料擷取出來，並依據要分析的主題將資料轉化成有意義的資訊，載入資料倉儲或資料超市的資料庫中，以做為將來知識或智慧的資料源頭。本系統整個所涵蓋的內容包括了(1)資料粹取／轉換／載入工具(ETL)，(2)資料分析平台設計工具（Analysis services)，(3）線上分析呈現工具（OLAP）及後端之查詢應用程式，請參考圖 17 資料轉化為最後智慧之流程。

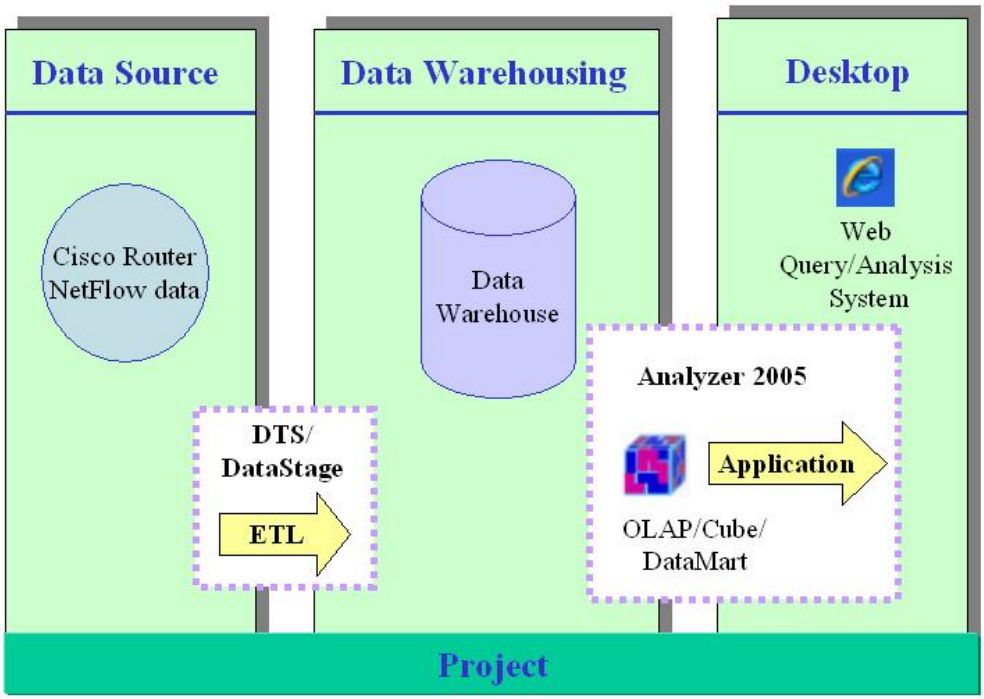


圖 17 資料轉化成智慧的流程

當然，從資料轉化成智慧的流程必得經過許多系統化或人工操作的加值部份（像是資料的整理、篩選、擷取等等），並輔以工具的使用及對系統的探索，才能完成一個完整的資料倉庫與商業智慧的解決方案。

4.3.2 系統實作步驟

在本節系統實作步驟裡，將分成 DataSource 部份、ETL 資料轉換部份、Data Cube、Data Mining 決策樹模型建立臨界值運用於線上即時分析報表以及 Web 查詢報表等五大部份，以下將一一詳細說明：

Data Source 部份

由 Cisco Router dump 出來的 NetFlow 資料封包原始格式；如圖 18 所示：

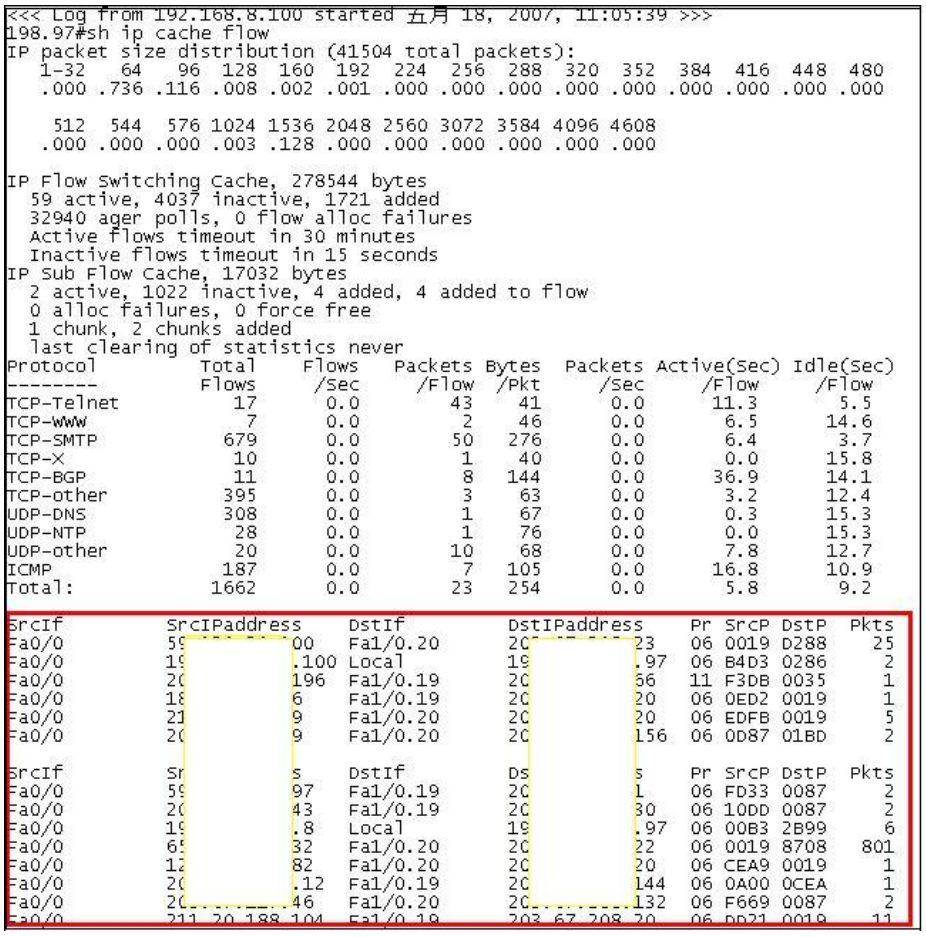


圖 18 由 Cisco Router dump 出的 NetFlow 格式

圖 18 的資料格式為 Cisco Router dump 出來的 netflow 資料，而框框內的資料才是我們需要的 netflow 格式，其中包括了 SrcIf、SrcIPaddress、DestIf、DestIPaddress、Pr、SrcP、DetP 以及 pkts。

ETL（資料擷取與轉換至資料庫）

由 Data Source 之資料轉入資料庫，我們必須先在資料庫為其建構一個 netflow 資料表格，netflow 資料表結構請參考前章節之表 3。

在有了資料表的結構後，我們必需使用 SQL 指令，將其建立於資料庫中，以儲存資料。

建立 netflow 資料表格之 SQL 指令：

```
Create TABLE netflow(
        SrcIf varchar(30) NOT NULL,
        SrcIPaddress varchar(15) NOT NULL,
        DestIf varchar(30) NOT NULL,
        DestIPaddress varchar(15) NOT NULL,
        ProtocolID int NOT NULL,
        SrcP nvarchar(6) NOT NULL,
        DesP nvarchar(6) NOT NULL,
        Pkts numeric(18,0) NOT NULL,
        StartTime datetime NULL,
        EndTime datetime NULL
)|
```

圖 19 建立 netflow 資料表格之 SQL 指令

在原始資料表的部份，除了歷史資料外，其它每日新增的資料受限於路由器儲存容量與資源的限制，我們是以每分鐘為單位將資料擷取後再轉入 netflow 資料表格，也就是該資料表是以累計的方式來運作，所以最新的資料會在該資料表的最後端，所以我們必須再透過程式進行資料之整理與萃取以找出最乾淨之資料，供之後做出正確與完整之分析。經由萃取後之資料，我們分別存在名為 ex_netflow 與 dw_netflow 之新資料表格內，接下來要做的任何分析與資料探勘都將依據此新整理過後的資料表格，該資料表格之欄位說明如下：

(a) 資料表格名稱：ex_netflow

表 20 ex_netflow table schema

欄位名稱	資料型態	是否允許null值	預設值	欄位說明
Item	Numeric		0	.主索引鍵 .流水號
SrcIf	Varchar(30)		0	來源之路由器介面
SrcIPaddress	Varchar(15)		0	來源之 IP 位址
DestIf	Varchar(30)		0	目的地之路由器介面
DestIPaddress	Varchar(15)		0	目的地之 IP 位址
ProtocolNO	int		0	封包傳輸時所使用之通訊協定
SrcP	Nvarchar(6)		0	來源之 Port 號(以十六進位表示)
DesP	Nvarchar(6)		0	目的地之 Port 號(以十六進位表示)
StartTime	datetime			該筆 flow 開始記錄之日期時間
EndTime	datetime	V		該筆 flow 停止記錄之日期時間
Fcount	Numeric(18,0)		0	計錄單位時間內，該 flow 出現的次數
Pkts	Numeric(18,0)		0	該筆 flow 所接收到的 packet 數量

ex_netflow 的資料主要是 netflow 資料表格經過整理與萃取出的資料，此資料表已經將單位時間內的 flow 統計出其出現的次數，以供 dw_netflow 資料表 fcount 欄位之用；同時，也是本論文自行開發之流量統計程式之依據。

(b) 資料倉儲之資料表格名稱：dw_netflow

表 21 dw_netflow table schema

欄位名稱	資料型態	資料長度	是否允許null值	欄位說明
SrcIfID	Numeric	9		.來自 Interface 資料表 .來源 Interface 的代號
SrcIPid	Numeric	9		.來自 IP 資料表 .來源 IP 的代號
DestIfID	Numeric	9	V	.來自 Interface 資料表 .目的地 Interface 的代號
DestIPid	Numeric	9		.來自 IP 資料表 .目的地 IP 的代號
ProtocolID	Numeric	9		.來自 Protocol 資料表 .使用通訊協定的代號
SrcPid	Numeric	9		.來自 Port 資料表 .來源 Port 的代號
DesPid	Numeric	9		.來自 Port 資料表 .目的地 Port 的代號
StartTime	Datetime	8		資料流的啟始時間
Fcount	Int	4		資料流的 flow 值
Pkts	Numeric	9		資料流的 packet 值
Pcount	Int	4		預設值皆為 1，方便統計用
DirectionID	Int	4		計錄此資料流為流入或流出； 0：流出 1：流入

dw_netflow 之資料欄位多參考自 ex_netflow 資料表，此資料表主要提供做 cube 之依據，為製作 cube 之事實資料表（fact table）之用。

在產生 dw_netflow 之前，必需先執行一支 Store procedure 程式來整理與萃取出要的資料，準備做為 data warehouse 內 fact table 之用。

Store procedure 名稱 : insert_exdw_netflow

執行 Store procedure 方式: 在 SQL command line 下執行下方之指令

exec insert_exdw_netflow

(Store Procedure 程式部份請參考附件一)

OLAP-Data Cube

我們透過 Analysis Services 建立了一個名為 cube_dwnflow_sifip_dport_area 之 cube，並將 StartTime、In or Out、Protocol No、Portno Hex、Portno、IP、IP user name、Protocol Name 此 8 個欄位設定為維度值(Dimension)，另外 Fcount、Pkts 這 2 個欄位設定為量值(Measure)；我們以 Analysis Service 中 Meta Data 頁面所成現的資訊來說明這 8 個維度值與 2 個量值。

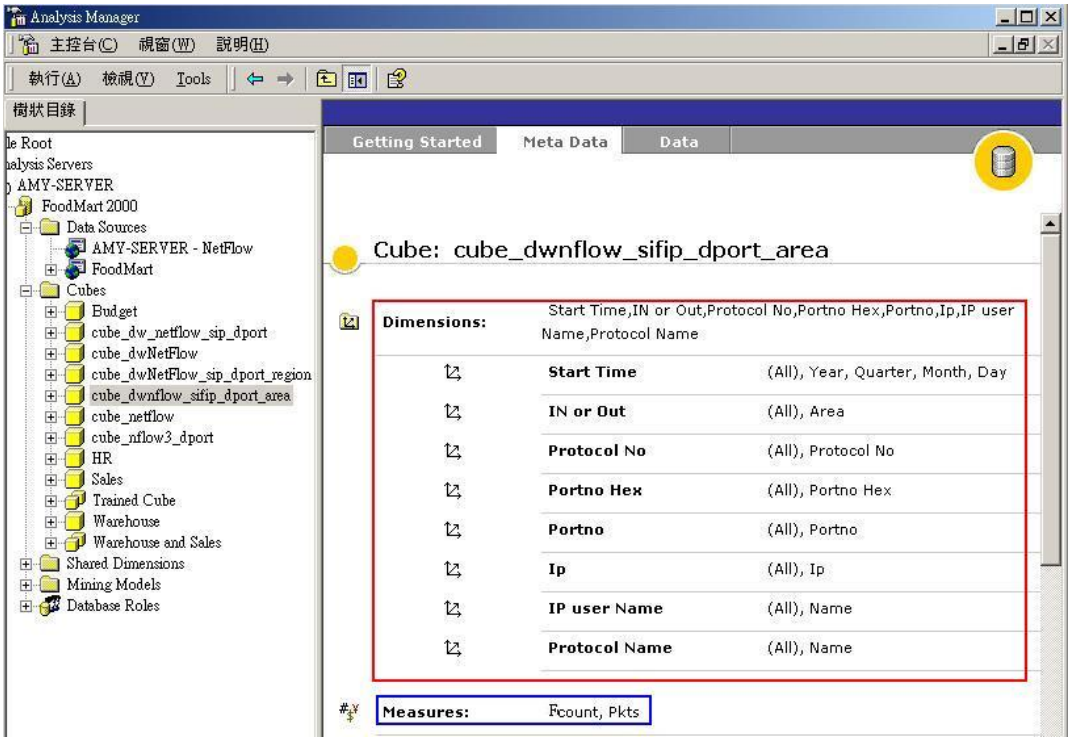


圖 20 cube_dwnflow_sifip_dport_area cube 之 Meta Data 資訊

此 8 個維度值與 2 個量值之說明如下：

維度值(Dimensions)

表 22 cube_dwnflow_sifip_dport_area cube 之維度值(Dimensions)

Start Time	記錄該筆 flow 的日期時間
In or Out	該筆 flow 的流向（流入或流出）
Protocol No	Flow 所使用之通訊協定編號
Portno Hex	以十六進位表示之 Port 號
Portno	以十進位表示之 Port 號
Ip	IP 位址
IP user name	該 IP 之使用者
Protocol Name	Flow 所使用之通訊協定名稱

量值 (Measures)

表 23 cube_dwnflow_sifip_dport_area cube 之量值(Measures)

Fcount	Flow 量
pkts	Packet 大小

依據以上之維度值與量值建立好之 cube，在所有維度值之條件為 all 之情況下，OLAP 系統將 pcount、pkts 做彙總的數值，參考圖 21。

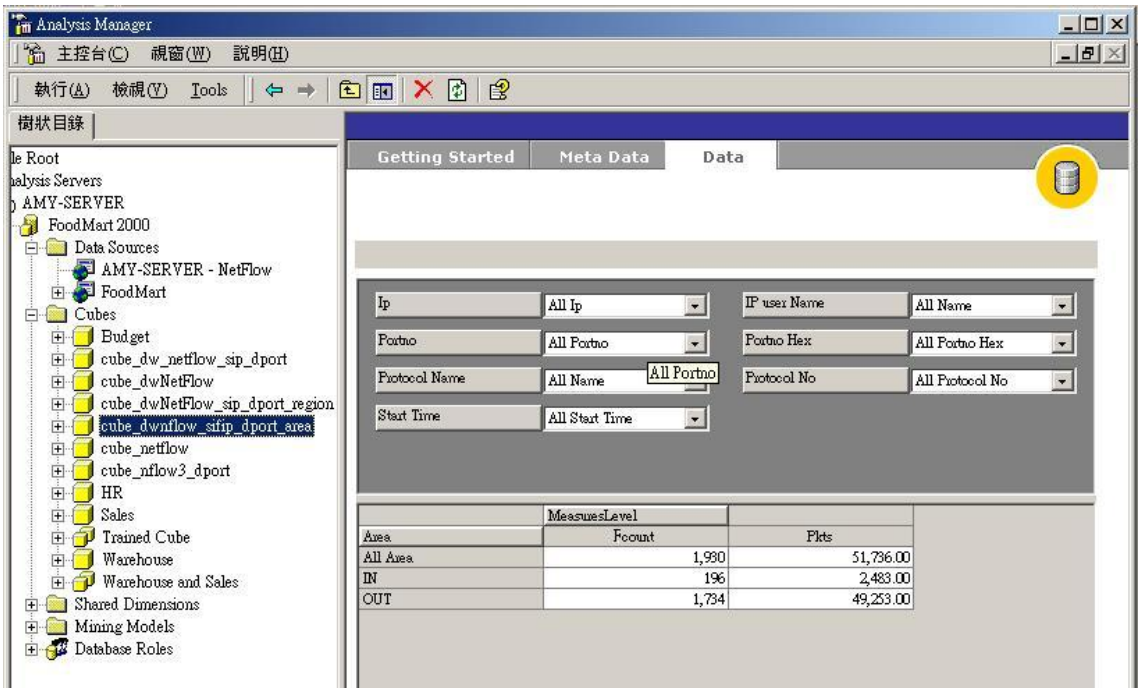


圖 21 在所有維度值為 all 的條件下之量值之總合結果

當然，我們可以依據自行想要定訂的條件，去變更維度的條件值，在下方的量值總合就會跟著條件值的不同，而快速的呈現結果。

例如，我們可以將條件過濾為 Protocol ID = 6 且 StartTime 為 5 月 18 日的資料，這時只需去調整 Protocol ID 與 StartTime 為我們想要的條件值，調整好之以後，量值的總合就會自動呈現以上條件的值了，請參考圖 22。

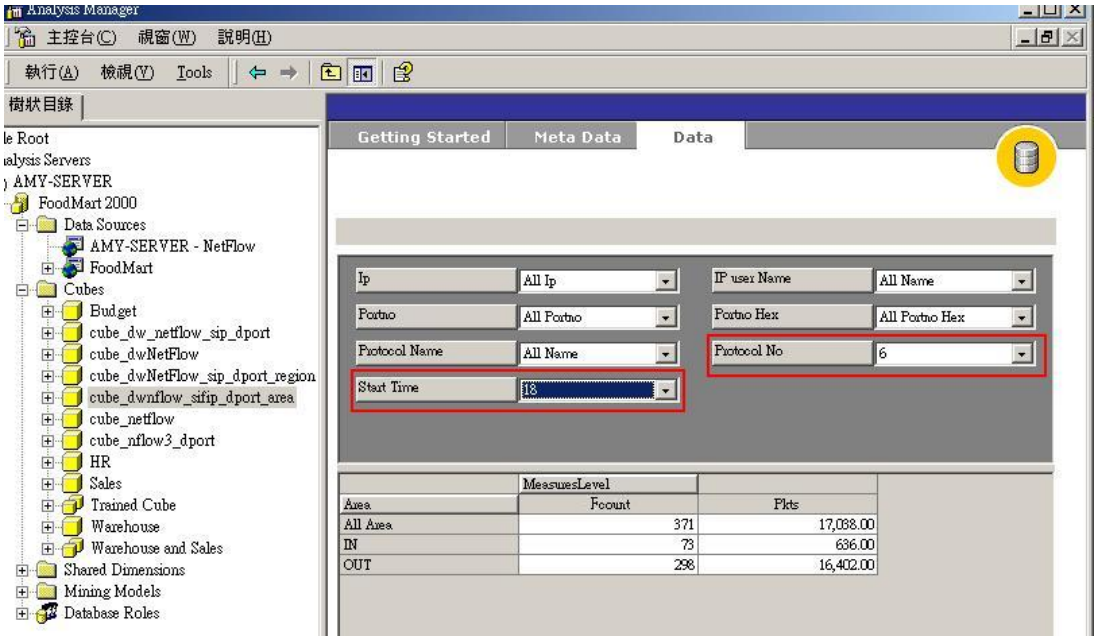


圖 22 設定好條件值之後的資料

當然，若要更清楚的知道在這些統計出來的量值背後，到底它的詳細資料是什麼，我們可以點選有興趣的數值，如圖 23 就會出現如下鑽研(DrillDown)後的資料。

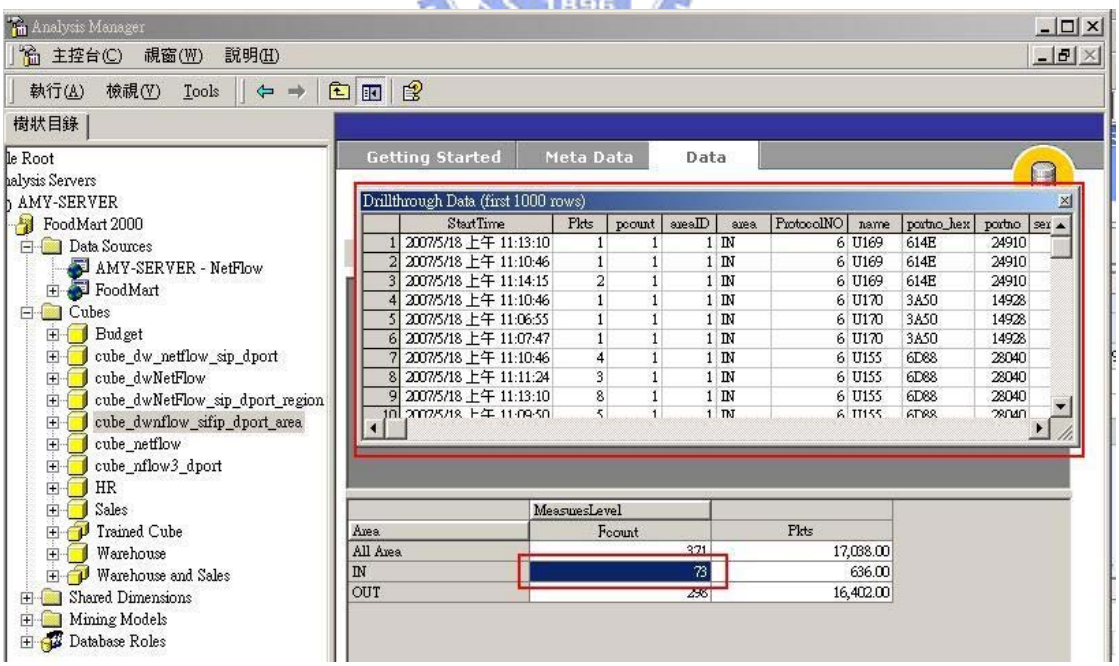


圖 23 DrillDown 後的詳細資料

OLAP AP

在 OLAP 線上即時分析部份，本論文製作了一個目的地 Port 流量統計的報表，藉以說明該如何運用前面設定的蠕蟲病毒臨界值於此報表，一但流量數超過此臨界值，即可快速的知道那個 Port 目前的流量有問題。

(1) 報表名稱：目的地 Port 流量統計與監測(area)

報表功能：針對目的地 Port 之流量加以統計，並對該值設定蠕蟲病毒的臨界值，借以提醒網路管理者，流量異常情況的發生。

使用 cube 名稱 cube_dwnflow_sifip_dport_area

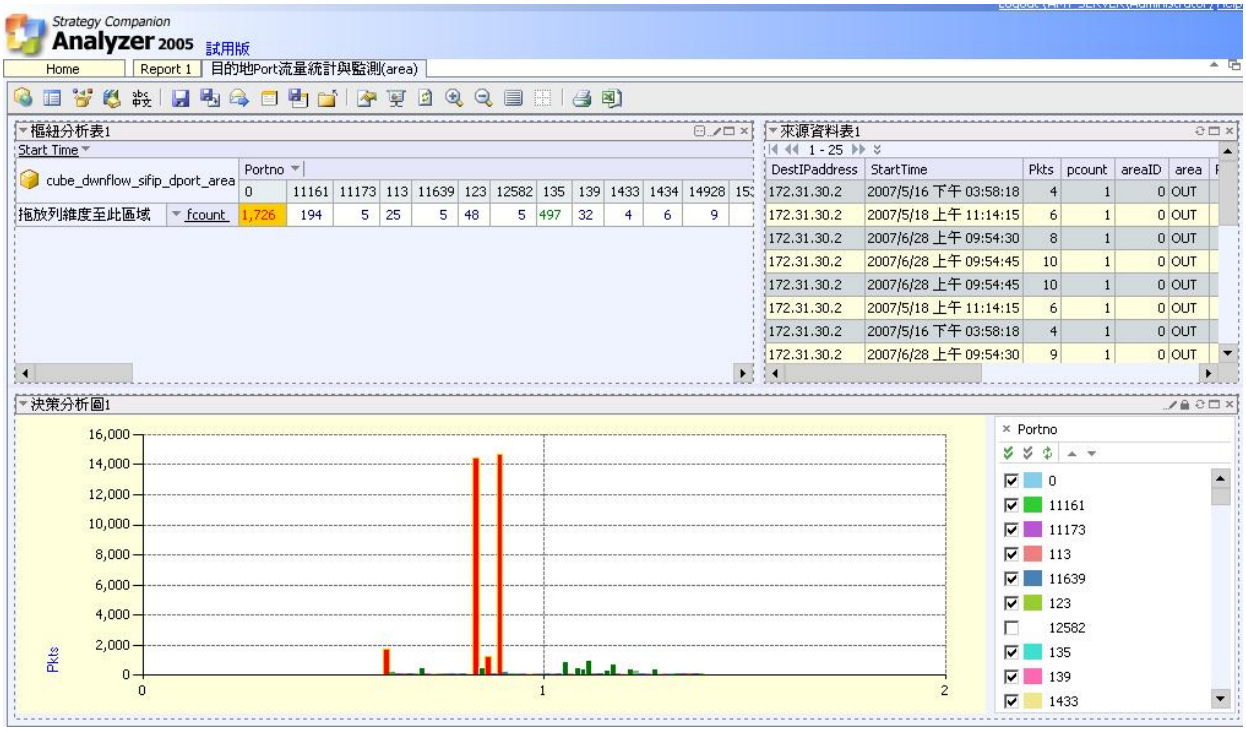


圖 24 目的地 Port 流量統計與監測報表

在圖 24“目的地 Port 流量統計與監測報表”中，可以看到報表分成了三大部份，有左上方的“樞紐分析表”，右上方的“來源資料表”，以及下方的“決策分析圖”；“樞紐分析表”主要是將所選擇的維度與量值之彙總結果呈現出來；“來源資料表”主要是提供對樞紐分析表內之量值，做更深入的鑽研(DrillDown)，也就是 raw data 的部份；“決策分析圖”部份則可提供管理者做各種圖狀(例如：直條圖、圓餅圖、折線圖、泡泡圖等等…)的分析。

(a) 針對量值設定監測值

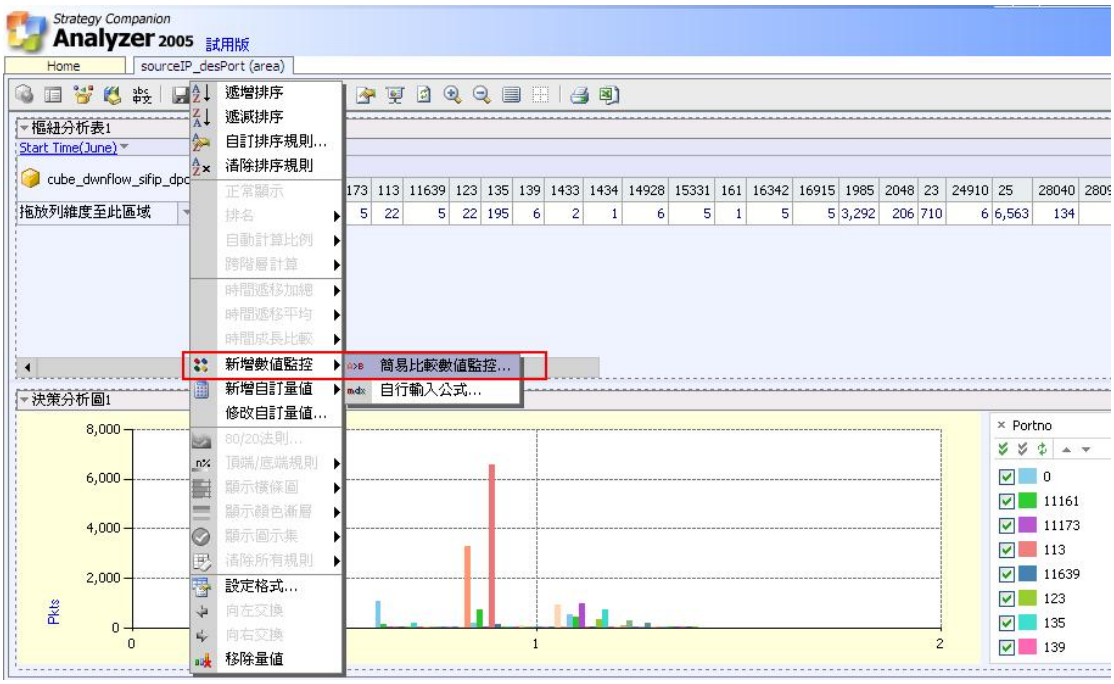


圖 25 設定監測值

例如，條件為 fcount >= 1000 之數值，1000 為本論文分析 CodeRed 病毒的 flow 臨界值，我們將該欄位的文字變成紅色，同時欄位用黃色加以填滿，以做為特別注意之標記。

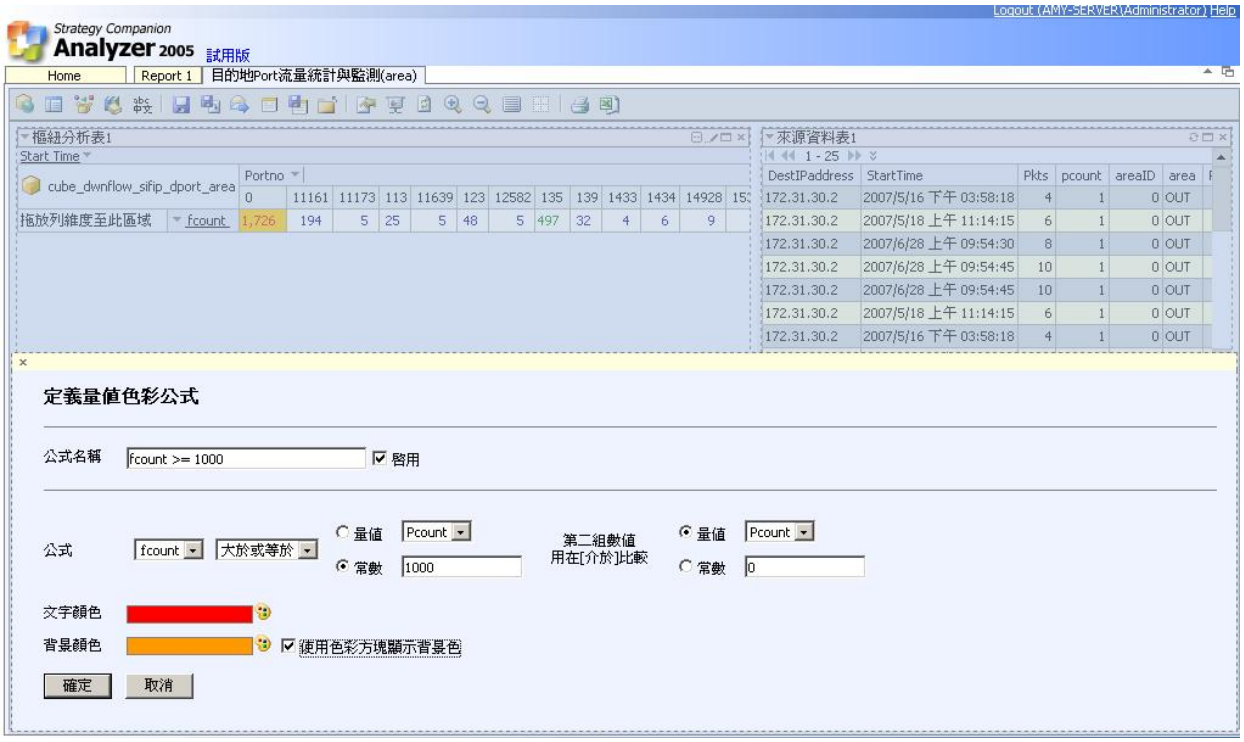


圖 26 設定監測值之條件

設定之後，所有大於等於 1000 的 fcount 量值皆會變成紅字，欄位底色為黃色（請參考圖 26）。

(b) 設定來源資料表（DrillDown 功能）

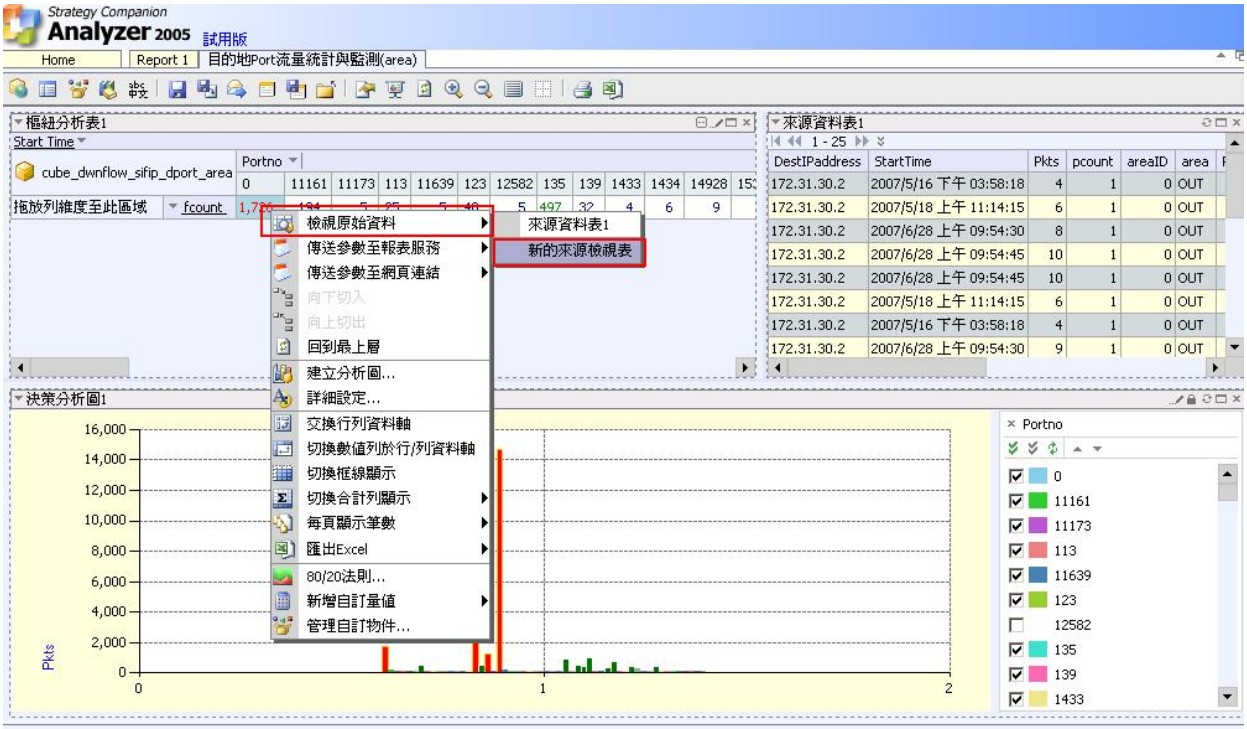


圖 27 設定 DrillDown 功能

在樞紐分析表內選定一個有興趣的量值，例如我們對 Portno = 0 的 fcount = 1726 這個量值有興趣，在選定這個量值之後，按一下滑鼠右鍵，會出現如圖 27 的選項可供選擇，針對 DrillDown 功能，只需依圖 27 使用紅色框線的功能（檢視原始資料－＞新的來源檢視表）即可，於是屬於該量值之詳細資料都會呈現於來源分析表內。

(c) 設定決策分析圖

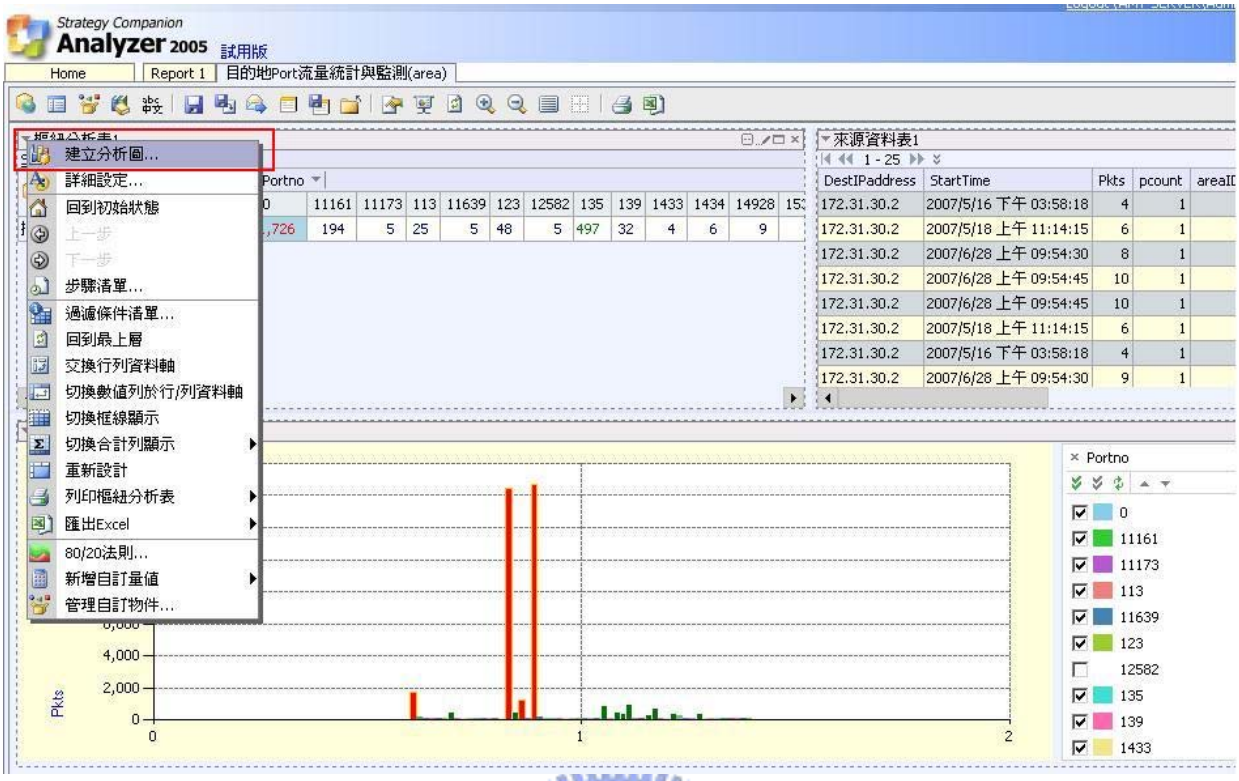


圖 28 設定決策分析圖

若要建立決策分析圖，只需在樞紐分析表處，如圖 28 所示，向下的箭頭按一下滑鼠右鍵，然後選定“建立分析圖”即可，接著，會出現如圖 29 建立新的分析圖設定的畫面；在這設定的畫面，需選定分析圖希望在報表中呈現的位置與種類(例如，長條圖、圓餅圖、折線圖等等…)，在選定好之後，分析圖的樣式就會呈現在報表中了，可參考圖 29。

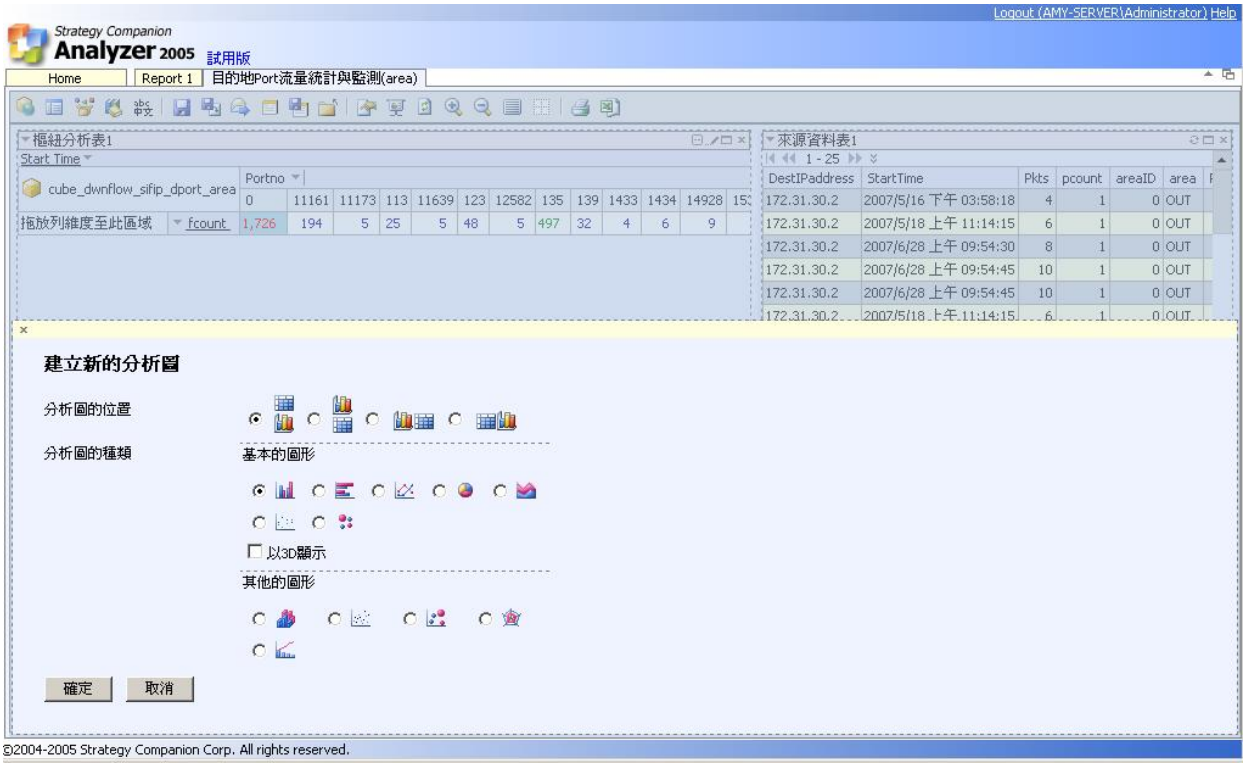


圖 29 建立分析圖樣式

在建立分析圖的選單裡，可以選擇分析圖所希望在報表中的那個位置（例如，報表的上方、下方、或左方、或右方），以及分析圖的種類（例如：長條圖、折線圖、圓餅圖、泡泡圖、區域圖等等…）。

Web 查詢報表程式

雖然市面上關於支援 Netflow 的軟體工具不少（例如，免費的 NetFlow Analyzer、FlowViewer，以及需付費的 NetView 等等…），而且圖文並茂，但這些終究是別人製定的條件與報表。在這些製示的報表不足我們使用的情況下，我們可能會需要一些自行定訂的條件與呈現方式，在這裡我們實作了一個 web 查詢的報表程式，主要的目的就是希望可以依據想要查詢的條件來自製報表，這樣的彈性度會比較大一點。

本論文中使用了 Microsoft ASP.NET 製作了一個簡單功能的 web 查詢報表，可供管理者自行查詢之用，本論文之網路流量統計網站位址：http://192.168.1.33/netflow/index.htm

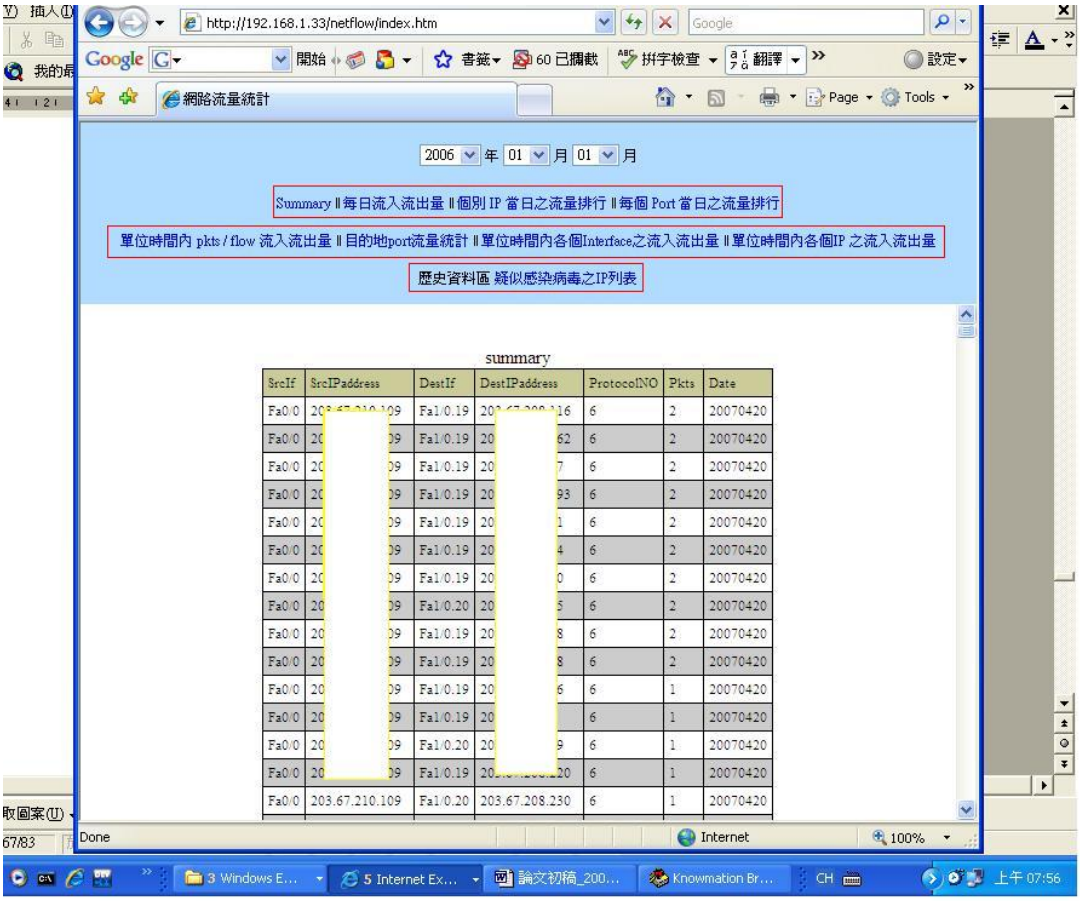


圖 30　web 網路流量統計程式呈現結果

本程式主要分 4 個子查詢程式，以及透過 Analyzer 2005 所設計出來的 1 個報表，還有歷史資料區的疑似受病毒攻擊之報表。

項目	功能	程式名稱	程式主要用途
1	Summary	Summary.aspx	每日所有流量明細
2	每日流入流出量	Everydaypkts.aspx	列出每日對外對內之流量
3	單位時間內個別 IP 之流量排行	EachIPSummary.aspx	單位時間內依 IP 所佔之流量做排序(大到小)
4	單位時間內每個 Port 的流量排行	EachPortSort.aspx	單位時間內依 Port 所佔之流量做排序(大到小)
5	目的地 port 流量統計與監測	http://192.163.1.33/analyzier	
6	疑似感染病毒之 IP 列表	Wormattack.aspx	依據決策樹分析出的最佳臨界值來判斷感染病毒與否

表 24 NOW 系統之子查詢程式列表

在網頁程式中另外 4 個分析報表，是使用 Analyzer 2005 所製作出來的分析報表，其中一個“目的地 port 流量統計與監測”是我們在前章節所特別提出來做設計的分析報表，在此將它一並呈現在此系統中。以及歷史資料區“疑似感染病毒之 IP 列表”的查詢報表，使用了在前章節所決定出的蠕蟲病毒 flow 之臨界值。在此附上“單位時間內個別 IP 之流量排行“程式，請件附件四。

5. 結論

若要單純的從每日所產生的 Netflow 資訊中取得有意義的資料，並不是件簡單的事。但是若能自行將 Netflow 的資訊，透過程式的開發與撰寫，卻可以讓我們在這一大堆流量資訊中，有效地排序、檢視，好讓網路管理者有效率地進行分析。本研究利用決策樹針對常見的 MSBlast、CodeRed 及一般的攻擊歷史做流量臨界值的分析，並對流量做線上即時分析與客製化之流量管理網頁。這些分析結果可讓網路管理者找出異常的網路狀態，進行更進一步的處理；除此之外，NetFlow 的資訊也可以被拿來分析網路、網路應用情況、使用者行為等等，藉此達到頻寬分配、安全分析、更可做為 ISP 之帳務應用等。

雖然目前已有許多自動化的工具程式可以用來處理各種不同種類的威脅，但實際上，將所有流量記錄下來，並且有能力進行各種分析，才能有效地鑑別出問題所在，並且處理之，如此才更能有效地運用 Netflow 資訊進行分析。

6.參考文獻

[1] Dan Zhu, G Premkumar, Xiaoning Zhang, Chao-Hsien Chu :Data Mining for Network Intrusion Detection : A Comparison of Alternative Methods. Decision Sciences; Fall 2001; 32,4; ABI/INFORM Global; 635-660

[2] Guy Helmer, Johnny S.K. Wong, Vasant Honavar, Les Miller; Automated discovery of concise predictive rules for intrusion detection; Department of Computer Science, Iowa State University, 226 Atanasoff Hall, Ames, IA 50011-1041, USA, February 2001

[3] S.Jha*, M.Hassan; Building agents for rule-based intrusion detection system; School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia; December 2001

[4] Cisco IOS NetFlow, http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html

[5] Cisco NetFlow 欄位說明, http://www.cisco.com/en/US/products/sw/iosswrel/ps1824/products_command_reference_chapter09186a0080080d98.html#xtocid23273115

[6] Ithome,企業資安技術應用專刊,2006

[7] NED LINDBERG,挖出 NetFlow 資訊,資安人科技網／Information Security,

http://www.isecutech.com.tw/feature/view.asp?fid=637,2006

[8] Stephen Northcutt、Judy Novak 著,陳正昌譯,網路入侵偵測教戰手冊,台灣培生教育出版社股份有限公司,ISBN:957-2054-57-0,2001

[9] 王曠銘,基於 NetFlow 之大型網路蠕蟲偵測系統,國立中山大學資訊工程學系,Jul 2005

[10] 李駿偉,入侵偵測系統分析方法效能之定量評估,中原大學資工所,2002

[11] 李駿偉、田筱榮、黃世昆,入侵偵測分析方法評估與比較,中原大學資工所,Communications of the CCISA Vol.8 No. 2,2002

[12] 黃悅民、陳順男,大學宿舍網路使用型態之應用層流量監測、分析與不當資訊管制之研究,國立成功大學,2004

[13] 賴森堂,「即時多觀點網路流量整合性監測工具,屏東商業技術學院資訊管理系(所),2004

[14] 劉士豪、蔡義昌,以 NetFlow 技術發展網際網路資料分析方法,中原大學資管所,Jul 2003

[15] 沈兆陽著,資料倉儲與 Analysis Services SQL Server 2000 OLAP 解決方案,文魁資訊股份有限公司,ISBN:957-466-165-2,2001

[16] 胡百敬、尹相志著,SQL Server 商業智慧聖經,學貫行銷股份有限公司,ISBN:986-7693-82-5,2004

[17] 中國傳媒科技,IPS 入侵防禦系統,

http://big5.xinhuanet.com/gate/big5/news.xinhuanet.com/newmedia/2005-11/22/content_3817846.htm,2005

[18] 台灣微軟,關於 MSBlast 蠕蟲及變種的病毒警告,http://support.microsoft.com/default.aspx?kbid=826955,2007

[19] 台灣微軟,NetBIOS 的瑕疵可能導致資料被洩露,http://support.microsoft.com/kb/824105/zh-tw,2007

[20] 台灣微軟,Slammer 病毒最新消息,http://www.microsoft.com/taiwan/security/slammer.asp,2003

[21] 台灣微軟,TechNet Flash 資訊技術人快訊－“Nimda”病蟲資訊,http://www.microsoft.com/taiwan/technet/flash/2001/0918.htm,2001

[22] 台中市教育局,DDoS 阻斷服務攻擊,http://netflow.tceb.edu.tw/virus/DDOS.html

[23] 台灣微軟 MicrosoftTech Net 網站,http://www.microsoft.com/taiwan/technet/columns/profwin/15-security_enterprise.mspx

[24] 台灣睿智,Analyzer 2005,http://www.analyzer.com.tw

[25] 台灣電腦網路危機處理暨協調中心（TWCERT/CC）,DDoS 攻擊的趨勢與防禦策略,http://www.cert.org.tw/document/column/show.php?key=73

[26] 台灣電腦網路危機處理暨協調中心（TWCERT/CC）,NetFlow 與網管之關係與應用,http://www.cert.org.tw/document/column/show.php?key=87

[27] 台灣賽門鐵克-安全機制應變中心,
http://www.symantec.com/zh/tw/enterprise/security_response/index.jsp

[28] 如何做 NetFlow 流量分析,
http://www.gpes.cy.edu.tw/study/network/%E5%A6%82%E4%BD%95%E5%81%9ANewflow%E6%B5%81%E9%87%8F%E5%88%86%E6%9E%90.htm

[29] 曹乙帆,CureLan FV-1000M 資訊安全流量管理系統, DigiTimes,
http://oa.digitimes.com.tw/print.aspx?zNotesDocId=AC8D23602A139FB148257164003D0E61,2006

[30] 蔡旻甫,新一代入侵防護機制的建置 - 理論與架構篇, 尚富煜科技,
http://www.sunfuin.com.tw/news/feature_safe_37.shtml

[31] 趨勢科技,病毒百科,http://www.trend.com.tw/vinfo/enduserdefault.asp

附錄

附錄一 程式碼 - insert_exdw_netflow store procedure

Store Procedure (insert_exdw_netflow) 程式碼

```
CREATE PROCEDURE insert_exdw_netflow

@idate varchar(6)

AS



DECLARE @vSrcIf char(30)

DECLARE @vSrcIPaddress char(15)

DECLARE @vDestIf char(30)

DECLARE @vDestIPaddress char(15)

DECLARE @vProtocolNO int

DECLARE @vSrcP char(4)

DECLARE @vDesP char(4)

DECLARE @vStartTime datetime

DECLARE @vFcount int

DECLARE @vPkts int

DECLARE @vpcount int



DECLARE @vitem int
```

```
DECLARE @vProtocolID int
DECLARE @if_sitem int
DECLARE @if_ditem int
DECLARE @isitem int
DECLARE @iditem int
DECLARE @psitem int
DECLARE @pditem int

DECLARE @vSareaID int
DECLARE @vDareaID int
DECLARE @vDirectionID int
DECLARE @vDirection char(5)

DECLARE nf_cursor CURSOR
FOR SELECT SrcIf, SrcIPaddress, DestIf, DestIPaddress, ProtocolID, SrcP, DesP, StartTime,
count(*) as fcount, sum(Pkts) as pkts, 1 as pcount
    FROM netflow
    WHERE CONVERT(varchar(10),cast(StartTime as smalldatetime),112) like '%' +@idate + '%'
    GROUP BY SrcIf, SrcIPaddress, DestIf, DestIPaddress, ProtocolID, SrcP, DesP, StartTime
    ORDER BY StartTime, SrcIPaddress
OPEN nf_cursor
FETCH NEXT FROM nf_cursor INTO @vSrcIf, @vSrcIPaddress, @vDestIf,
```

```
@vDestIPaddress,@vProtocolNO,@vSrcP,@vDesP,@vStartTime,@vFcount,@vPkts,@vpcount

WHILE @@FETCH_STATUS=0

BEGIN



   select DISTINCT @vSareaID =  p.areaID  from netflow n, ip p where n.SrcIPaddress = p.ip
and n.SrcIPaddress = @vSrcIPaddress

   select  distinct @vDareaID =  p.areaID from netflow n, ip p where n.DestIPaddress = p.ip
and n.DestIPaddress = @vDestIPaddress



   if ((@vSareaID = 0) and (@vDareaID = 1)) or ((@vSareaID = 1) and (@vDareaID = 1))

    BEGIN

     select @vDirectionID = '1'

     select @vDirection = 'IN'

    END

   else

    BEGIN

     select @vDirectionID = '1'

     select @vDirection = 'OUT'

    END



   SELECT @vitem = max(item)+1 from ex_netflow

   insert into ex_netflow
values(@vitem,@vSrcIf,@vSrcIPaddress,@vDestIf,@vDestIPaddress,@vProtocolNO,@vSrcP,@vDes
P,@vStartTime,@vFcount,@vPkts,@vpcount,@vDirection,NULL)
```

```
   SELECT @vProtocolID = item FROM protocol WHERE ProtocolNO = @vProtocolNO

   SELECT @if_sitem = if_item FROM interface WHERE if_name = @vSrcIf

   SELECT @if_ditem = if_item FROM interface where if_name = @vDestIf

   SELECT @isitem =  ip_item   FROM ip  WHERE IP = @vSrcIPaddress

   SELECT @iditem = ip_item   FROM ip  WHERE ip = @vDestIPaddress

   SELECT @psitem = port_item   FROM port  WHERE portno_hex = @vSrcP

   SELECT @pditem = port_item   FROM port   WHERE portno_hex = @vDesP



   insert into dw_netflow
values(@if_sitem,@isitem,@if_ditem,@vDestIPaddress,@iditem,@vProtocolID,@psitem,@pditem
,@vStartTime,@vFcount,@vPkts,@vpcount,@vDirectionID)



   FETCH NEXT FROM nf_cursor INTO @vSrcIf,  @vSrcIPaddress,  @vDestIf,
@vDestIPaddress,@vProtocolNO,@vSrcP,@vDesP,@vStartTime,@vFcount,@vPkts,@vpcount



END

CLOSE nf_cursor

DEALLOCATE nf_cursor



GO
```

附錄二 常用的協定號碼

協定名稱	協定號碼	協 定 全 名
IP	0	Internet Protocol
ICMP	1	Internet Control Message Protocol
IGMP	2	Internet Group Multicast Protocol
GGP	3	Gateway-Gateway Protocol
TCP	6	Transmission Control Protocol
PUP	12	PARC Universal Packet Protocol
UDP	17	User Datagram Protocol

附錄三 常見的服務通道(Port List)

服務名稱	port
ftp	21
telnet	23
smtp	25
dns	53
www	80
pop3	110
auth	113
nntp	119
ntp	123
https	443

附錄四 程式碼 - 單位時間內個別 IP 之流量排行

```
<%@ page language="VB" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SQLClient" %>
<script runat="server">

  Sub page_load(obj as Object, e as EventARGS)
    dim sdate as string
    sdate = Session("symd")
    Response.write(sdate)

    dim myConnection as new SQLConnection("server=localhost;uid=sa;pwd=adminsa;database=NetFlow")
    dim myCommand as new SQLDataAdapter("select SrcIPaddress, sum(Pkts) as pkts from ex_netflow _
        where CONVERT(varchar(10),cast(StartTime as smalldatetime),112) like '%" & sdate & "%' _
        group by SrcIPaddress order by pkts desc", myConnection)
    dim ds as DataSet = new DataSet()
    myCommand.Fill(ds, "ex_netflow")
    ListData.DataSource = ds.Tables("ex_netflow")
    ListData.DataBind()
  end sub
</script>

<html><body>
<form runat="server">
<center>
  <br> 個別 IP 的總流量排行 <br>
  <asp:DataGrid ID="ListData" Runat="server" BorderColor="black" CellPadding="4" Font-Size="8pt" _
    HeaderStyle-BackColor="#cccc99" ItemStyle-BackColor="#ffffff" AlternatingItemStyle-BackColor="#cccccc"
    AutoGenerateColumns="false">
   <columns>
    <asp:BoundColumn HeaderText="SrcIPaddress" DataField="SrcIPaddress" />
    <asp:BoundColumn HeaderText="Pkts" DataField="pkts" />
   </columns>
  </asp:DataGrid>
</center>
</form></body></html>
```