逢 甲 大 學

資訊工程學系專題報告

以逢甲大學為基礎之網路流量統計暨異常行為偵測

學 生： 張 煉 謙 (四乙)

指導教授 ： 劉 安 之

中華民國九十三年五月

目　　錄

圖 表 目 錄

第一章 導論

1.1 研究動機

近來網際網路上所流傳之病毒多如牛毛，網路使用者一不小心便會中毒，而此類大量流傳於網路上之病毒，大多都具有強大的攻擊行為及感染行為，於是逢甲大學校園網路這種開放的區域網路架構便首當其衝。

逢甲大學校園網路便常苦於無法有效迅速的即時解決病毒的破壞行為，一但有病毒發作，產生攻擊行為或是破壞行為，與遭受攻擊之電腦同網段或是同樓層之使用者們都會遭受影響，輕者網路緩慢，重則整棟大樓網路癱瘓，發作情況不勝枚舉。

由於筆者在資訊處系統維運組擔任工讀生，從事校園網路維護之工作，對於校內網路有相當程度之熟悉，在這環境之下，對於從事校內網路流量統計及病毒偵測之研究有很好的助益。

還沒開始作專題之前，就常聽見有關宿舍網路的問題。資源有限而每個人的慾望是無窮。所謂的網路資源浪費，其實包括了頻寬的浪費以及不符合經濟成本的服務設施的建置。而許多人是自已先浪費了資源後，才在怪罪所得到的資源不足。而且浪費的結果導致資源的不足，資源不足則會使成本提昇，成本提昇會轉嫁回使用者身上，造成一種

惡性的循環，所以，先做好節流的功夫再想開源才是最好的辦法。

學校的網路頻寬也是如此，頻寬有限，但是使用者的使用慾望卻是無窮的。在這有限的頻寬，若是所有使用者都是合法且合理的使用之下，校園網路頻寬仍是不夠，則真的就要朝開源的方向努力了。不過，以我們的觀察，目前頻寬的不足，一大部分是出在使用者的不當濫用。架設地下站時有所聞，在校園網路上的大批資料的共享，大批資料的抓取，都是使校園網路速度慢的原因。常見使用者一邊大量抓取網路芳鄰資料同時也在執行P2P軟體，這類使用者常常抱怨校園網路速度為何那麼慢，試想，大家如果都像這樣使用，校園網路怎麼會快的起來，所以宿舍網路的管理，變成了一項很重要的事。透過監控和管制，才能讓校園網路保有品質，才能保障合法使用的使用者。本系統就是希望能做出可以有效分析各種通訊協定之流量，並作出報表，供網管人員使用。讓網管人員透過本系統去針對異常行為作控管之動作。把影響頻寬的最大黑手－私設地下站給抓出來，使校園網路能最佳化使用。

1.2 現況說明

逢甲大學校內網路使用者眾，但是對於來自網路上的駭客及病毒攻擊有足夠防範之使用者並不多，常見 windows 使用者未定期作 windows update 甚或未安裝 service pacck 等修補程式，尤其以校內宿舍網路為最嚴重。

近年駭客及病毒常見針對 windows 漏洞來做攻擊行為，使用者防不勝防，也發生過有定期作 windows update 之電腦一樣遭受病毒魔掌。對於網管人員來說，要維持網路使用品質真是一大挑戰。

現今學術網路內使用之網路流量統計系統大多出於交大 Netflow 文件(http://netflow.nctu.edu.tw/netflow.html) ，經過評估結果決定採用交大 Netflow 之架構，再對資訊處系統維運組所提之需求來做系統之開發。使用跨平台之網頁來做呈現，不會發生因網管人員使用之作業系統不同，而需切換介面才可瀏覽之情況。

1.3 發展目標

實做具有即時檢測異常行為及監看網路流量之管理系統，使用網頁介面作呈現，加上易讀的圖表，使網管人員很快的能獲得所要的資訊。

第二章 系統分析

2.1 網路管理

網路管理目的在於維持良好的網路服務品質、及時排除網路障礙、增進網路維運效能，並進一步作為提供網路建置規畫的參考。網路的服務品質需隨時掌握與維持，運用預防式（Proactive）網路管理方式，使網路服務品質維持一定水準。

2.1.1 網路管理五大功能

◆ **組態管理** (Configuration Management)

大致包含網路拓樸資訊管理、自動搜尋網路設備、路由資訊管理與組態資訊管理。網路拓樸資訊是網管的第一步，要清楚網路拓樸之後，才能明確範圍進一步加以管理與監控。就像設備資產管理一般，網管人員可管理設備的數量、其上的軟硬體相關配備及設備上的組態及路由資訊；而透過網路管理系統就可自動搜尋出網路設備、路由資訊管理與組態資訊管理。

◆ **異常管理** (Fault Management)

即偵測、隔離及修復等管理，包含告警監視、故障定位和錯誤處理功能等。告警監視是以即時方式，經由網路監視重要網路元件

並將訊息傳給維運系統，以判斷屬一般或嚴重的故障。當障礙發生，同時還得即時判斷發生故障的設備是否有關聯事件，進而將其隔離，讓其餘系統能正常運作。

如何找出故障所在？依據告警監視或例行性維護時所發現的異常現象加以分析。維運系統分析許多網路元件的相關故障訊息，據以判斷故障位置所在，執行隔離或修復工作。在多重故障的情況下，還需借助專家系統進行故障定位。性能監視是連續收集網路元件性能相關資料，但不影響對客戶的服務。當服務品質降低時，提供性能資料給故障定位機制，以找出故障點。

網管人員還可量度整體的連線通訊品質，在連線品質降到預設臨界（Threshold）點前發現問題，並將資料存到網路性能共同資料庫作為評鑑網路品質的參考。

◆ **安全管理** (Security Management)

配合企業的營運需要制定安全政策(Security Policy)，並依據此政策制定安全法則(Security Rule) 及建置資訊系統，並隨時稽查，以杜絕各種非法活動。一般的安全管理包括機房管制，文件管理、Service 管理、密碼管理、防火牆管理等。

◆ **效能管理** (Performance Management)

指網路效能資料的收集與分析、報表的產生、問題通告與效能

提升的規畫。包含網路流量監測、統計分析與報表產生、伺服器品質監控、效能告警與網路性能及可靠度之提升。網路流量監測可隨時知道網路的使用狀況與品質，管理者收集流量數據並進行統計分析產生管理報表，除了可作為提升網路效能的參考，還可針對特殊的使用情況加以控管。另外當網路流量達到預設的臨界點時，需適時提出告警，讓管理者能適時處理以維持網路正常運作。

◆ **使用量管理** (Accounting Management)

使用量管理係指對資訊資源的使用情形的記錄，若有使用費的考量時，可做為會計計算的基礎。一般來說，使用量管理主要著重於網路頻寬及伺服器 CPU 使用量的管理。

2.1.2 系統建置

本系統以網路管理五大功能為基礎，進行 NetFlow 建置。

◆ **組態管理：**

掌握合法 IP 使用情況，查獲非法 IP 使用者。

◆ **異常管理：**

流量封包異常現象，可疑地下站偵測。

找出私設地下站之使用者，防止頻寬被佔用。

異常發生即時以 E-mail 通知。

◆ **安全管理：**

判斷網路病毒及駭客入侵可疑行為偵測。

以即時阻絕原兇，預防擴散之現象。

◆ 效能管理：

分析各單位網路使用效能，診斷網段是否瓶頸之現象。

◆ 使用量管理：

統計各單位流量使用率統計。

對外網路使用率。

以達到計時計價之目的。

2.1.3 流量統計階層關係

◆ 校園流量統計

逢甲校區流量：行政教學單位。

福星校區流量：男生宿舍、女生宿舍、學人宿舍。

包租宿舍流量：台逢學園、翰林學園、寶贊大樓。

校外宿舍流量：廣昱資通。

◆ 單位流量統計

行政單位流量：各行政單位。

教學單位流量：各教學單位。

宿舍流量：各樓層流量、各寢室流量。

◆ 個人流量統計

2.2 NetFlow

為了配合校園網路管理，及有效的運用頻寬，了解網路的使用情況是必要的。因此使用 NetFlow 技術來觀察校園網路的使用狀況，避免網路被使用在不正當的方向，浪費了網路頻寬，藉以提升校園網路速度及便利性。

2.2.1 NetFlow 簡介

NetFlow是Cisco 發展的流量統計協定，這些統計被有效的使用在網路管理、計數及網路規劃等，目前廣受各家廠商支援。

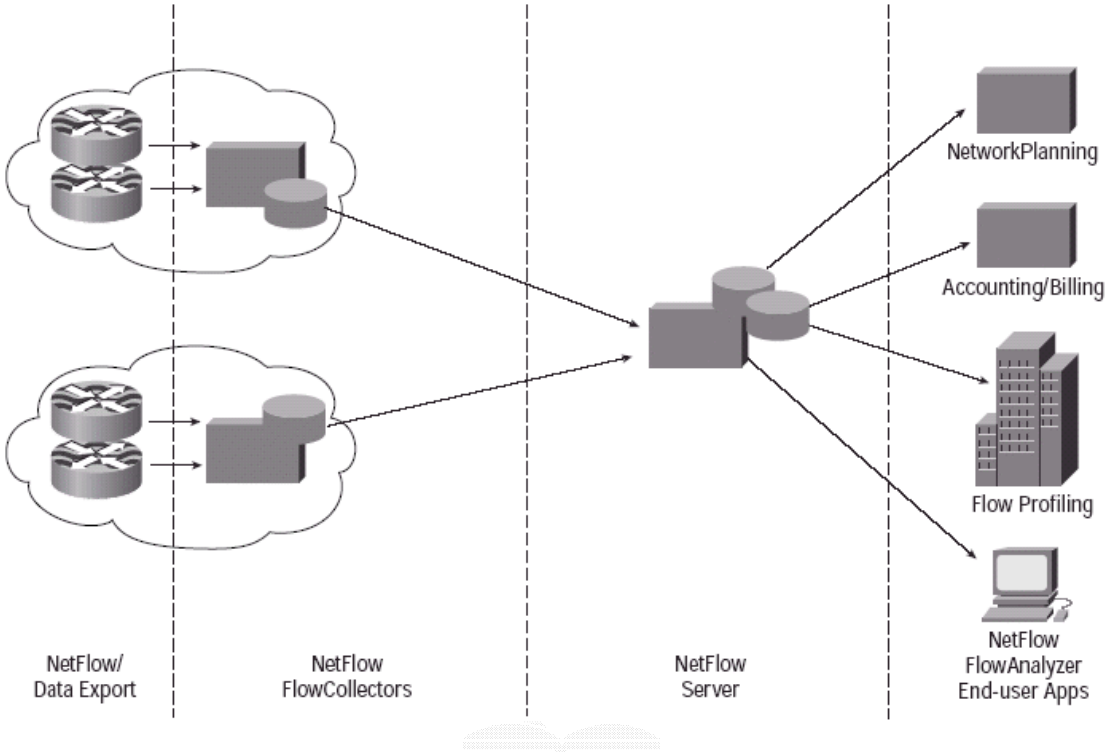
Flow 指的是特定來源和目地的單向流量資料，也就是來源IP、目地IP、來源Port、目地Port 四個屬性相同的封包之資料傳送量總和為一個Flow。

2.2.2 NetFlow 架構

NetFlow架構共可分為四部分, 如圖表 1 所示。NetFlow Data Export 支援NetFlow的設備，如Router、 Sniffer、 NTop。

Net Flow FlowCollector提供快速、準確與便利的資料收集方式，接收UDP協定的NetFlow封包，存進NetFlow Server。 NetFlow Server可同時由多個FlowCollectors收集datafiles，但無法每次由單一FlowCollector取得一個以上的datafile，以避免不當的執行影響NetFlow

FlowCollector，**並降低磁碟與**CPU**的負荷。**NetFlow FlowAnyalyzer**是網路傳輸分析工具，利用**NetFlow Server**中的資料做統計分析，如各別**IP**傳輸量、協定分佈、封包大小分佈、異常偵測。**



圖表 1 Netflow **架構**

NetFlow**可提供下列資訊，分析網路異常狀況：**

種類	**功能／內容**
來源IP**、目地**IP**、來源**port**、目的**port	**主機、服務傳輸量**
協定種類	TCP、UDP、ICMP **等等**
入口介面、出口介面	**流量在實體層的來源和目的，因**IP **層可以偽造。**

TCP Flag	◎可以判斷連線的性質，開始(SYN)、過程(SYN+ACK)、結束(FIN) ◎可以判斷攻擊的性質，如SYN Flood、ACKFlood、OOB Attack
packet 數目、byte 數目、開始時間、結束時間	Workload分析，例如：攻擊行為通常封包小數量多
ASN (Autonomous System Number)	在與多個ISP 互連時可直接統計互連流量，不需依照IP 判斷ISP

圖表 2 NetFlow 可提供之資訊

2.2.3 NetFlow 封包結構介紹

NetFlow 送出的封包大小接近 1500 bytes，一般來說每送一次包含有 20 – 50 個 flow 數，會依照當時網路忙碌程度提高送出的頻率。Netlfow 目前的版本有 Version 1、Version 5、Version 7、Version 8、Version 9。以下介紹 Version 1 及 Version 5。

2.2.3.1 Version 1

Version 1 封包包含標頭和流量資料, 其中流量資料數紀錄在標頭的 count 參數中，最多 24 筆。

Version 1 標頭結構

Bytes	Content	Description
0-3	version and count	NetFlow export format version number and number of flows exported in this packet (1-24).
4-7	SysUptime	Current time in milliseconds since router booted
8-11	unix_secs	Current seconds since 0000 UTC 1970.
12-15	unix_nsecs	Residual nanoseconds since 0000 UTC 1970.

圖表 3 Version 1 **標頭結構**

Version 1 流量紀錄格式

Bytes	Content	Description
0-3	srcaddr	Source IP address.
4-7	dstaddr	Destination IP address.
8-11	nexthop	Next hop router's IP address.
12-15	input and output	Input and output interface's SNMP index.
16-19	dPkts	Packets in the flow.
20-23	dOctets	Total number of Layer 3 bytes in the flow's packets.
24-27	First	SysUptime at start of flow.
28-31	Last	SysUptime at the time the last packet of flow was received.
32-35	srcport and dstport	TCP/UDP source and destination port number or equivalent.
36-39	pad1, prot, and tos	Unused (zero) byte, IP protocol (for example, 6=TCP, 17=UDP), and IP type-of-service.
40-43	flags, pad2, and pad3	Cumulative OR of TCP flags. Pad 2 and pad 3 are unused (zero) byte.
44-47	reserved	Unused (zero) bytes.

圖表 4 Version 1 **流量紀錄格式**

Version 1 標頭格式圖

Byte 3	Byte 2	Byte 1	Byte 0

Version	Count
Sysuptime	
Unix seconds	
Unix nanoseconds	

圖表 5 Version 1 **標頭格式圖**

Version 1 流量紀錄格式圖

Byte 3	Byte 2	Byte 1	Byte 0

Source IP address		
Destination IP address		
Next hop IP address		
Input physical interface index	Output physical interface index	
Packet count for this flow		
byte count for this flow		
Start of flow timestamp		
End of flow timestamp		
Source TCP/UDP application port	Destination TCP/UDP application port	
Padding	IP Protocol	Type of Service
Padding		
Padding		

圖表 6 Version 1 **流量紀錄格式圖**

2.2.3.2 Version 5

Version 5 封包包含標頭和流量資料, 其中流量資料數紀錄在標頭的 Count 參數中，最多 30 筆。和 Version 1 不同的是 Version 5 在 Header 中加入了封包的順序號碼，如此一來便可以知道封包是否有遺失而造成流量資料不正確。在流量資料中也加入了 Border Gateway Protocol (BGP) AS (autonomous system 自主系統)號碼及遮罩長度。

Version 5 標頭結構

Bytes	Content	Description
0-3	version and count	Netflow export format version number and number of flows exported in this packet (1-30).
4-7	SysUptime	Current time in milliseconds since router booted
8-11	unix_secs	Current seconds since 0000 UTC 1970.
12-15	unix_nsecs	Residual nanoseconds since 0000 UTC 1970
16-19	flow_sequenc	Sequence counter of total flows seen.
20-23	reserved	Unused (zero) bytes.

圖表 7 Version 5 **標頭結構**

Version 5 流量紀錄格式

Bytes	Content	Description
0-3	srcaddr	Source IP address.
4-7	dstaddr	Destination IP address.
8-11	nexthop	Next hop router's IP address.
12-15	input and output	Input and output interface's SNMP index.
16-19	dPkts	Packets in the flow.
20-23	dOctets	Total number of Layer 3 bytes in the flow's packets.
24-27	First	SysUptime at start of flow.
28-31	Last	SysUptime at the time the last packet of flow was received.
32-35	srcport and dstport	TCP/UDP source and destination port number or equivalent.
36-39	pad1, tcp_flags, prot, and tos	Unused (zero) byte, Cumulative OR of TCP flags, IP protocol (for example, 6=TCP, 17=UDP), and IP type-of-service.
40-43	src_as and dst_as	AS of the source and destination, either origin or peer.
44-47	src_mask, dst_mask, and pad2	Source and destination address prefix mask bits, pad 2 is unused (zero) bytes.

圖表 8 Version 5 **流量紀錄格式**

Version 5 標頭格式圖

Byte 3	Byte 2	Byte 1	Byte 0

Version	Count
Sysuptime	
Unix seconds	
Unix nanoseconds	
Flow sequence number	
reserved	

圖表 9 Version 5 **標頭格式圖**

Version 5 流量紀錄格式圖

Byte 3	Byte 2	Byte 1	Byte 0

Source IP address			
Destination IP address			
Next hop IP address			
Input physical interface index		Output physical interface index	
Packet count for this flow			
byte count for this flow			
Start of flow timestamp			
End of flow timestamp			
Source TCP/UDP application port		Destination TCP/UDP application port	
Padding		IP Protocol	Type of Service
Source AS		Destination AS	
Source netmask length	Destination netmask length	Padding	

圖表 10 Version 5 **流量紀錄格式圖**

2.2.3.3 封包格式圖

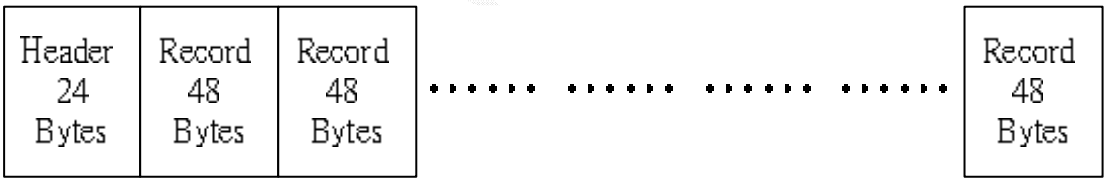
Version 1 封包圖

圖表 11 Version 1 **封包圖**

Version 1 **共包含** count **個** record。Count **最大值為** 24**，也就是說** Version 1 **最多可以包含** 24 **筆流量資料，封包最大為**

16 bytes(Header) + 48 bytes * 24 records = 1168 bytes

Version 5 封包圖



圖表 12 Version 5 **封包圖**

Version 5 **共包含** count **個** record。Count **最大值為** 30**，也就是說** Version 5 **最多可以包含** 30 **筆流量資料，封包最大為**

24 bytes(Header) + 48 bytes * 30 records = 1464 bytes

2.3 病毒及駭客攻擊行為

現今會對網路造成影響之病毒是越來越多，大多為 Worm 之型態，為及早發現病毒攻擊行為及防止網路遭受病毒攻擊而崩潰，本系統具有偵測病毒之特定及不特定異常行為功能。

特定行為之病毒採用該筆 Flow 特徵來辨識，例如：protocol = udp destination port = 1434 為 SQL Slammer 之行為。

不特定行為之病毒採用 Flow 數量之臨界點來做定義，目前使用之臨界點為每 10 分鐘產生的 Flow 數量超過 10000 筆，可視情況作調整之動作。

2.3.1 一般性的流量特徵

因 Worm 的特性將造成異常流量，因此，應先找出正常與異常流量之特徵，才可比較分析何處發生 Worm 的感染。依照 Worm 所利用的安全弱點型式，有直接和間接的流量特徵。

直接流量特徵為攻擊者對受害者的直接連線，利用作業系統漏洞(例：Remote Procedure Call)、不安全的權限設定(例：網路芳鄰開放任意讀寫)、伺服器的漏洞(例：透過 Web Server 入侵)。已被感染的主機會產生大量對外攻擊的連線，因此可以先找出封包數量或 Flow 數量異常之主機，再分析相關之資料。

間接流量特徵以 E-mail Worm **為代表，利用的是應用程式弱點(例：** mailclient **自動開啟附加檔案並執行)、不安全使用習慣(例：使用者開啟來源不名的郵件附加檔案)，表現在流量上為** E-mail **相關的流量異常增加。**

茲介紹各種常見病毒及其特定行為如下：

2.3.2 Internet Worm 簡介

Worm **是利用主機的安全弱點進行自動化入侵與繁衍的惡意程式碼。利用的弱點包括作業系統的弱點、應用程式的弱點、不安全的權限設定、伺服器的弱點、人類不安全的使用習慣。**

2.3.2.2 Code Red 簡介

Code Red **是一隻會自我繁殖入侵系統的惡意程式碼，利用微軟**IIS WebServer **的安全性漏洞入侵，並在受害者主機上自我繁殖，入侵後留下後門，同時產生**600 **個執行緒，隨機產生**IP **嘗試入侵沒有安裝修補程式的** IIS WEB**伺服器，被植入**Code Red **的主機可能同時多次掃瞄同一台主機，沒有受到感染的主機可能會成為**Code Red **的攻擊目標，受到**DOS(denial of service)**的攻擊。由於持續變換目地**IP **位置，消耗路由器資源，導致低階路由器當機，中高階路由器效能下降。**

2.3.2.3 Nimda 簡介

Nimda (別名W32/Nimda.A@mm, I-Worm.Nimda, Readme,Readme.exe 等)會利用現有網站提供受感染檔案下載，並利用 End User 的主機掃描其他有漏洞的網站，這使得Nimda 可輕易地入侵受防火牆屏障的Intranet，強大的破壞力透過網際網路在全球迅速傳播，破壞力不遜於Code Red。

Nimda 混合使用多種傳播方式：

1. 和Code Red 一樣使用IIS 之漏洞
2. 被攻擊主機之前遭Code Red worm 或sadmid/IIS worm 感染留下之後
3. Internet Explorer 以及Outlook Express 自動開啟檔案的漏洞
4. 修改Web Server 之網頁使得拜訪者自動下載Nimda
5. 上傳至網路芳鄰可任意讀寫之資料夾
6. E-Mail

Nimda 結合上述多種常見的攻擊模式，是其迅速擴張肆虐的主因，一天之內全球都受到感染，Nimda 會針對系統漏洞進行複製與散播，將C 磁碟機設為資源共享，任意複製、修改、刪除重要檔案文件、破壞受感染的系統，並加重網路承載。另外因為使用網路芳鄰，因此除了路由器效能受影響外，LAN 的效能也受影響。如果受到感染後沒

有徹底清除並修補系統漏洞，將會對網路安全造成重大危害。

2.3.3 病毒特徵

2.3.3.1 Code Red Worm

感染紅色警戒病毒後，主機至少會被植入第一代木馬，這支木馬會持續送出攻擊封包，企圖感染別的主機。這些攻擊行為有一個特性，就是每個 flow 的 desitonation port=80, packets=3, size=144

2.3.3.2 Nimda Worm

Nimda 由於使用多種漏洞，並且依照被攻擊主機有不同行為， 無法以特定 Packet 數量和 Byte 數量作為判斷條件，因此採用 desitonation port = 80 且五分鐘內同一 IP 之 flow 數大於threshold 判斷。

2.3.3.3 SQL Slammer

自2003/01/25起網際網路開始遭受SQL Slammer蠕蟲的威脅，受害主機以UDP Port 1434高速溢滿網際網路，形成分散式阻斷服務攻擊(DDOS)。

SQL Slammer蠕蟲的感染對象是微軟SQL server 2000和微軟SQL桌

面引擎 (MSDE) 2000。它利用Microsoft SQL Server 2000的緩衝區滿溢漏洞進行傳播。這個漏洞的成因是SQL沒有妥善處理傳送往UDP port 1434的數據。

當SQL伺服器收到惡意要求，會引致緩衝區滿溢而令蠕蟲可以執行。當它入侵了有漏洞的系統後，便開始在網路上循環的尋找其他有漏洞的主機，再利用UDP連接埠1434將自已傳送到有漏洞的主機，一傳十、十傳百，進而癱瘓整個網路。

2.3.3.4 疾風病毒 MSBLAST(DCOM_RPC)

受感染的電腦會出現「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、無法執行複製貼上或拖曳圖示的動作、新增移除程式呈現空白狀態、某些應用程式且無法執行（例如Internet Explorer、Microsoft Outlook、Outlook Express、MS Office...等），也會讓網路與系統速度都變慢。不過，此病毒不具大量寄送郵件的症狀。病毒發作為1 - 8月的16 - 31日，9 - 12月的任一天。該病蟲試圖對Windows Update執行「拒絕服務」（DoS）攻擊。其目的是阻止你使用針對DCOM RPC弱點的修正程式。

其 DoS 流量具有下列特徵：

1. 在 windowsupdate.com 的埠號 80 上有 SYN 流量。

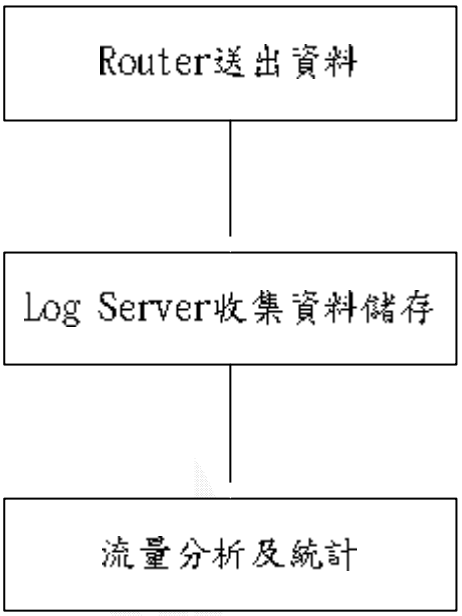
2. 試圖每秒傳送 50 個 RPC 封包及 50 個 HTTP 封包。
3. 每個封包長度為 40 位元組。
4. 如果病蟲在 DNS 裏找不到 windowsupdate.com，它會使用 255.255.255.255 作　目標位址。

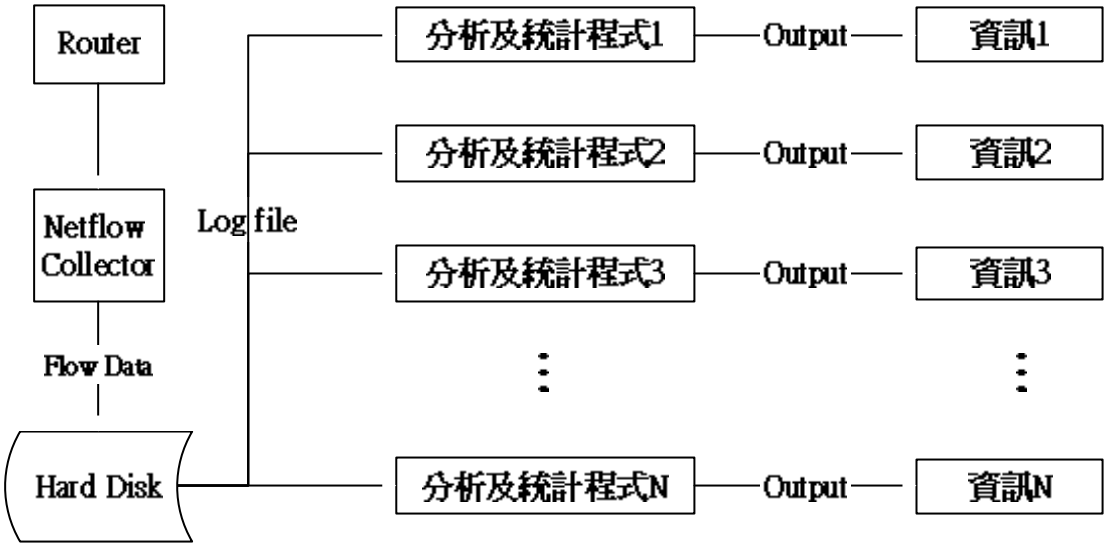
2.3.3.5 Welchia 假好心病毒

此病毒為「疾風」病毒的變種，檔案名稱為「dllhost.exe」（在Window系統中也有另外一個同名的系統檔案，但大小只有6 KB。）。它會試圖從Microsoft的Windows Update網站下載更新程式，並移除先前的「疾風」病毒，然後開啟電腦的666 ~ 765 Port以進行感染，所以被稱為「假好心病毒」。此病毒當電腦的系統時間為2004年時，會自動從系統內移除。此病蟲會使用兩種不同的方式來選取受害者的IP位址。它會從受感染機器的IP (A.B.C.D)中使用A.B.0.0並往上累加，或者根據某些內建的位址來隨機建立IP位址。當選定開始位址後，它會在Class C網路的位址範圍內往上累加。例如，若從A.B.0.0開始，它將往上累加到至少A.B.255.255。此病蟲將傳送ICMP回應或PING來檢查所建立的IP位址是否為網路上啟用中的機器。一旦病蟲辨識出此位址屬於網路上啟用中的機器，它將傳送資料至TCP port 135以探測DCOM RPC弱點，或者傳送資料至TCP port 80以探測WebDav弱點。

2.4 系統架構

本系統主要分三個步驟完成：




Router
Netflow Collector
Flow Data
Hard Disk
Log file
分析及統計程式1
分析及統計程式2
分析及統計程式3
分析及統計程式N
Output
Output
Output
Output
資訊1
資訊2
資訊3
資訊N

2.5 系統功能

收集 Router **所送出之** Flow **資訊，加以分析運算，將結果放入** Database**，以網頁呈現。**

網頁可查詢之項目：

1. **當日全校流量：查詢前一日全校** TopN **之流量比例**
2. **異常行為列表：查詢前十分鐘異常行為之發生，採用圖片表示，易讀，容易判斷**
3. **各系所當日之流量資訊：查詢前一日各系所流量資訊及比例**
4. **各協定之流量分析：查詢各種已設定監控之協定流量分析比例**
5. **各** IP **之流量分析：查詢個別** IP **之流量資訊比例**
6. **歷史總流量：查詢過去到現在之總流量及比例**
7. **歷史異常行為** IP **列表：查詢** IP **是否有過異常行為**
8. **異常行為之說明：解釋各種異常行為偵測之方式**

第三章 系統分析

3.1 環境說明

3.1.1 硬體環境

主機型號：DELL PowerEdge 2600

CPU：**雙** Xeon 2.4Ghz **支援** HyperThread

記憶體：1G

硬碟：SCSI 36.4G 10Krpm *3 RAID-5

3.1.2 軟體環境

Mandrake 9.2，Kernel 2.4.22，gcc 3.3.1，g++ 3.3.1，MySQL 4.0.18，

PHP 4.3.4，Apache 2.0.48，GD 2.0.15，zlib 1.1.4，Perl 5.8.1，

DBD-mysql-2.9003

硬碟分割

/ ：1024MB
swap ：1024 MB
/home ：5120 MB
/netflow ：40960 MB **存放** Flow **原始資訊及** MySQL **資料表**
/usr ：10240 MB
/var ：10240 MB

3.2 相關工具

3.2.1 Perl 介紹

Perl 是什麼東西呢? Practical Extraction Report Language 的縮寫，是一種能掃視任意文字檔，並且能從中取出資訊製成報表的解譯語言（Interpreter)。它的目的就是用來取代 UNIX 原有的 sed、awk 與 shellscript 的組合，用來匯集資訊、產生報表的一個工具語言（程式)。Perl 的作者 Larry Wall，他設計 Perl 時的哲學是以實用為第一優先(所謂的實用就是語言容易使用、有效率，而且完整)，而不是設計一個看起來很漂亮的語言（漂亮就是程式非常的小，語法幽雅，而且只由最少的語法基本元素構成)。

一般而言，由 awk，sed，shell programming 寫的東西可以輕易用 Perl 來處理，而且速度更快。在 C program 中，特別是有關抽取文字檔中資訊, 加以轉化處理, 用 Perl寫會更加方便, 而且不用處處擔心 pointer 和 memory allocation 的問題。當然，Perl 執行時會有一道轉成內部表示式的關係，真正 optimized 過的 C code 仍會比 Perl code 快。一般而言，Perl code 會比 C code 短很多。

Perl 在 regular expression，list（array）和 associative array 有很方便語法和不錯的執行效率。連 yacc（Berkeley yacc）也可有產生 Perlcode

選項，這對於 prototyping 十分方便。

正由於 Perl 的特性，容易處理字串參數的傳遞，自然成了某些特殊用途的新寵，尤其是 CGI，彷彿 Perl 天生是為了 CGI 而設計的！事實上 CGI 本身定義是和使用的程式語言種類無關的，你可以用 C、Fortran、甚至 Visual Basic、 Delphi 來寫 CGI 程式。

那大家為什麼要採用 Perl 原因有三點：

1. Perl 有非常傑出文字處理能力，能輕易的產生 HTML 文件，尤其它具有完整 Regular Expression 語法，使得在 C 語言看來必需要寫一大串程式碼的功能，Perl 幾行就解決了。
2. Perl 是一種直譯式語言，因此可以避免在各種不同系統上，還需要重新編譯的麻煩。
3. Perl 的檢查旗標(-T)可以保護你的 Web Server，避免被未經授權的 client 破壞。

由於 Perl 處理文字的能力很強，所以拿來處理 Flow 資訊的速度也是很快，自然拿 Perl 來實作是最佳選擇。

3.2.2 Apache 介紹

Internet 最近幾年成為極熱門的話題之一，造成這股熱潮的主因便是 World Wide Web(以下簡稱 WWW or Web)。世界各地的 Web Page 的製作者將自己精心製作的網頁放到網站上，讓所有其他的人參觀，

透過這種互動，整個 Internet 形成一個龐大的資料庫，我們可以在上面找到各式各類我們想要的資訊。

那麼要如何建立一個網站呢？除了主機，作業系統與使用者所製作的網頁外，我們還需要安裝一套能將網頁放到網路上讓其它人來存取的軟體，也就是所謂的 Web Server。Web Server 比較有名的有免費的 Apache，Microsoft 的 Internet Information Server，Netscape 的 Enterprise Server 等等。由於我們使用的作業系統平台是 Linux，因此我們介紹的是在 Unix 系統上最受歡迎的 Apache Web Server，Apache 也有 Windows 的版本。免費的 Apache Web server 具有比商業 Web server 不惶多讓的功能與速度，同時安裝與設定也十分地容易，由於這些特性使得 Apache 成為佔有率最高的 Web Server 軟體。

3.2.3 PHP 介紹

在 1994 的時候，有一位 Rasmus Lerdorf 利用了 perl 寫了一個小程式，想知道他放在網站上的簡歷有多少人來看過，於是他便將其包裝成一個 **Personal Home Page Tools** 這就是 PHP 的前身囉 一開始的 PHP 當然不如現今的功能來的強大，只能算是一個語法解析的引擎，大概只能用來處理一些小程式如計數器之類的。之後 Rasmus 先生又將 Personal Home Page Tools 結合了 FI(form interperter, 表單直譯器)==> PHP/FI(PHP2) 誕生了。PHP 推出之後一直獲得不錯的評價，與不少人

的使用，到了 PHP3 的時代，PHP 已經有一個團隊來開發了它，並且重新改寫了 PHP，替 PHP3 奠定了良好的基礎。到了 PHP4 結合了 zend，與眾人的努力，PHP 的功能變的更強大，如今的 PHP 除了能應付網站上的應用需求外，亦能開發視窗介面的 GUI 程式。最重要的是 PHP 是免費的!!

PHP 是一種嵌入在 HTML 並由伺服器解釋的腳本語言。它可以用於管理動態內容、支援資料庫，甚至構建整個電子商務站點。它支援許多流行的資料庫，包括 MySQL、PostgreSQL、Oracle、Sybase、Informix 和 Microsoft SQL Server。

另外 PHP 也支援以下的東西

- 支援 HTTP Authentication
- 支援 FastCGI
- 支援 Access Control
- 支援 Access Logging
- 支援 GD library
- 支援 File Upload (RFC 1867)
- 支援 cookie
- 支援 Regular expression

3.2.3.1 PHP 與 GD 結合的安裝設定

本系統有使用到 GD Library**，所以使用上最需要注意的就是在安裝** PHP **時要一並作設定，同時也需要安裝** linpng **及** zlib**。設定方式如下：**

```
./configure --with-apxs2=apxs 之路徑  --with-mysql --with-gd
--with-zlib-dir=zlib 路徑
```

3.2.4 MySQL 介紹

MySQL **是一個小巧靈瓏的資料庫伺服器軟體，對於小型（當然也不一定很小）應用系統是非常理想的。除了支援標準的** ANSI SQL **語句，它還支援多種平臺，而在** Unix **系統上該軟體支援多線程運行方式，從而能獲得相當好的性能。**

MySQL **的功能特點如下：**

- **可以同時處理幾乎不限數量的用戶；**
- **處理多達** 50,000,000 **以上的記錄；**
- **命令執行速度快，也許是現今最快的；**
- **簡單有效的用戶權限系統。**

3.2.5 PerlDBI DBD::mysql 介紹

3.2.5.1 DBI 介紹

DBI 的建構者及作者 Tim Bunce 說: "DBI 是給 Perl 語言用來連接資料庫的程式設計介面(API)。DBI API 規格中定義了一系列函數、變數以及用法，他們提供了一套連接資料庫的一致介面，不論實際上你所要連接的資料庫是那一種。"

簡單的說，DBI 介面容許使用者透通地 (transparently) 連接多種類形的資料庫。所以如果你連接到 Oracle、Informix、mSQL、Sybase 或任何其它的資料庫，你都無須了解介面後運作的機制。因為 DBI 定義的 API 在這些資料庫上都可以運作。

3.2.5.2 DBD-mysql 之安裝及使用

DBD-mysql-2.9003 這檔案可由

http://www.cpan.org/modules/by-module/DBD/ 這裡取得

解開檔案之後丟在任意暫存目錄下即可開始安裝，本系統採用手動安裝，也就是重新從 source code Make 過。Make 所下之參數為

```
perl Makefile.PL --testdb=<db> --testuser=<user>
--testpassword=<password> --testhost=<hostname>
```

其餘若是有所不了解可以查看此套件的說明黨。

以下說明此 DBI 在程式碼裡的使用方式

宣告使用 DBI 及相關變數：

```
use DBI;
my  $dsn              = "DBI:mysql:dbname:localhost";
my $user               = "dbuser";
my $password        = "password";
my ($dbh, $rows);
```

與資料庫連線：

```
$dbh = DBI->connect($dsn, $user, $password);
```

對資料庫下指令：

```
$dbh->do("INSERT INTO……")
```

與資料庫中斷連線：

```
$dbh->disconnect();
```

3.2.6 Flow-tools 介紹

本系統使用 Flow-tools **來接收** Router **所送出之** Flow **資訊，** Flow-tools **可以從** http://www.splintered.net/sw/flow-tools/**取得，以下介紹本系統所使用到的兩支程式。**

3.2.6.1 flow-capture

這支程式是用 daemon **的方式在系統裡常駐，在開機時執行，在** /etc/rc.d/rc.local **加入下面這段**

```
/usr/local/netflow/bin/flow-capture -n 143 -V 5 -z 6 -N 0 -e 600
-w /netflow reciver ip/0/9991 &
```

程式本身有支援許多參數，以下解說本系統有使用到之參數

/usr/local/netlfow/bin **為** flow-capture **所在路徑**

-n **為一天留存幾份，**143 **為每** 10 **分鐘產生一個檔案。**

-V **使用的** NetFlow **版本例如** 5 **就是** Version 5

-z **壓縮率，一般來說使用** 6 **就差不多了，比率再高也不見得更有效率**

-N **使用之檔名格式**

-e **最多留存多少份資料**

-w Flow **檔案儲存的地方**

local ip/remote ip/port **本機**(collector)**的** IP**，以及接收** Port **的設定**

3.2.6.2 flow-print

本系統使用 flow-print 來印出流量資訊供分析程式擷取使用，日後若是有未知異常行為需要判定也是需要由此方式印出原始資料來做確認。以下為本系統所使用之指令

```
Flow-print –f0 < ft-v05.2004-05-04.190000+0800
```

-f0 **為印出簡單格式其中包含** source interface，source ip，destination interface，destination ip，protocol，source port，destination port，packets，size。

ex：

Sif	SrcIPaddress	Dif	DstIPaddress	Pr	SrcP	DstP	Pkts	Octets
0009	10.30.143.6	0000	90.216.201.228	06	9ee	1bd	3	144
0009	172.17.57.66	0009	193.61.121.101	06	1318	6662	1	40

分析程式再依印出位元格式來擷取所要的資訊轉換成可計算的數值。

使用-f1 **參數印出之結果為兩行格式，多了每筆** Flow **的起始時間**(StartTime)**、終止時間**(EndTime)**、持續時間**(Active)**、每個** Packet **多少** Byte(B/Pk)**、** Type of Service(Ts)**及** flags(fl)**。**

Sif	SrcIPaddress	DIf	DstIPaddress	Pr	SrcP	DstP	Pkts	Octets
StartTime		EndTime		Active		B/Pk	Ts	Fl
0009	10.30.143.6	0000	90.216.201.228	06	9ee	1bd	3	144
0506.09:16:38.678		0506.09:16:47.574		8.896		48	00	02

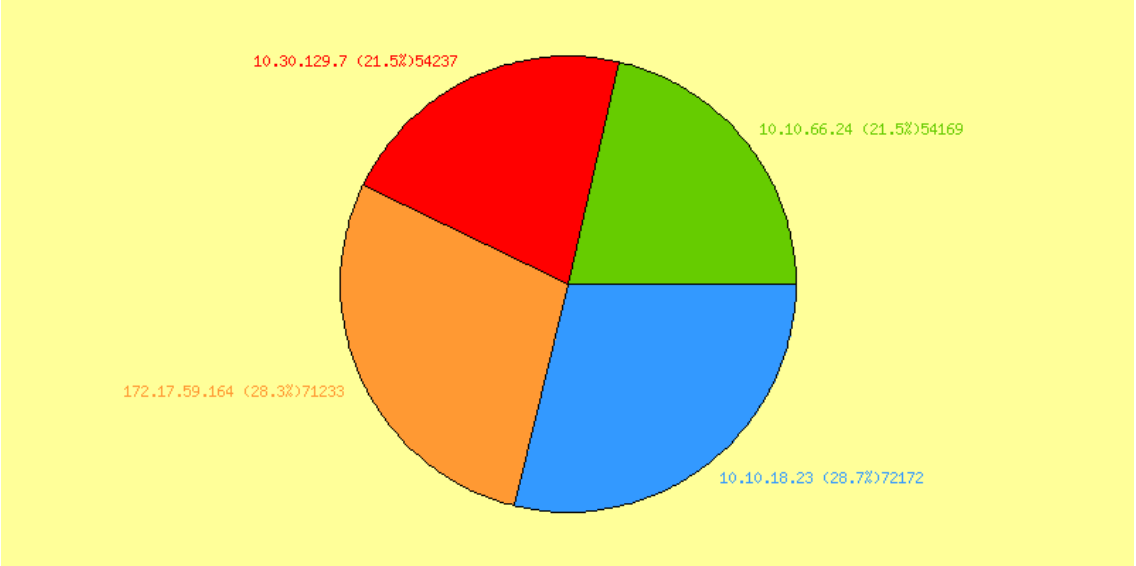
3.2.7 phPie() PHP script 介紹

PHP 要怎麼有效率的畫出大餅圖呢？這個 script 就是很好的方式。只要將值與名稱傳送給這個 php script，這個 php script 就會幫您畫出大餅圖，讓您可以輕鬆的達成目標。

使用方式為：

```
http://webpath/phPie.php?data[10.10.18.23]=72172&data[172.17.59.164]=7123
3&data[10.30.129.7]=54237&data[10.10.66.24]=54169
```

其產生出來的圖片為：



圖表 15 phPie **繪圖範例**

3.2.8 CSS 介紹

CSS **是「**Cascading Style Sheets**」的縮寫，**Cascading **是串接、連接之意；**Style **則是風格、款式之意；**Sheets **則是一頁紙、表的意思。所以呢，要以中文來解釋** CSS **的話呢，稱之為「串接樣式表」。簡單來說，**CSS **令我們的網頁更美麗。**

使用 CSS **架構的網頁有個很大的特點，就是可以一次更改許多數量的網頁配色而不用一個一個網頁去修改，對於網頁的整體性及日後維護都相當的方便。也不用擔心修改中的網頁會有瀏覽上的問題。**

第四章 系統實作

4.1 網路流量統計暨異常行為管理系統

本系統使用 Netflow 技術，用來對校園網路作流量之統計、各種通訊協定之分析及異常行為之監控。分為三個層次：流量分析、異常行為分析及網頁呈現。

4.1.1 流量分析

流量分析其作法為每天固定時間做分析運算, 分析運算之結果放入 Database 供網頁來做瀏覽，以及備分。

每日運算時間大概約一小時(假設無大量異常行為發生之情況下)，以目前的欄位定義每天放入 Database 的檔案大小為 110MB。

4.1.1.1 通訊協定

系統目前所使用分析之通訊協定

service	port	service	Port
Ftpdata	20	Edonkey4665	4665
Ftp	21	Proxy	3128
Telnet	23	EzPeer6677	6677
Smtp	25	EzPeer6678	6678
Www	80	Kuro	6699
Pop3	110	Bt6881	6881
Tcp135	135	Bt6882	6882
NetbiosNs	137	Bt6883	6883

NetbiosDgm	138	Bt6884	6884
NetbiosSsn	139	Bt6885	6885
Tcp445	445	Bt6886	6886
Tcp593	593	Bt6887	6887
Edonkey	4661	Bt6888	6888
Edonkey4662	4662	Bt6889	6889
Edonkey4663	4663	BTServer	6969
Edonkey4664	4664	others	All other port

圖表 16 系統使用之分析協定

若要更改所要分析的協定，只要修改 services **檔案內的通訊協定及** port **即可成程式產生資料庫表都是自動的，無須煩惱萬一更改通訊協定之後資料庫欄位並沒建立的問題**

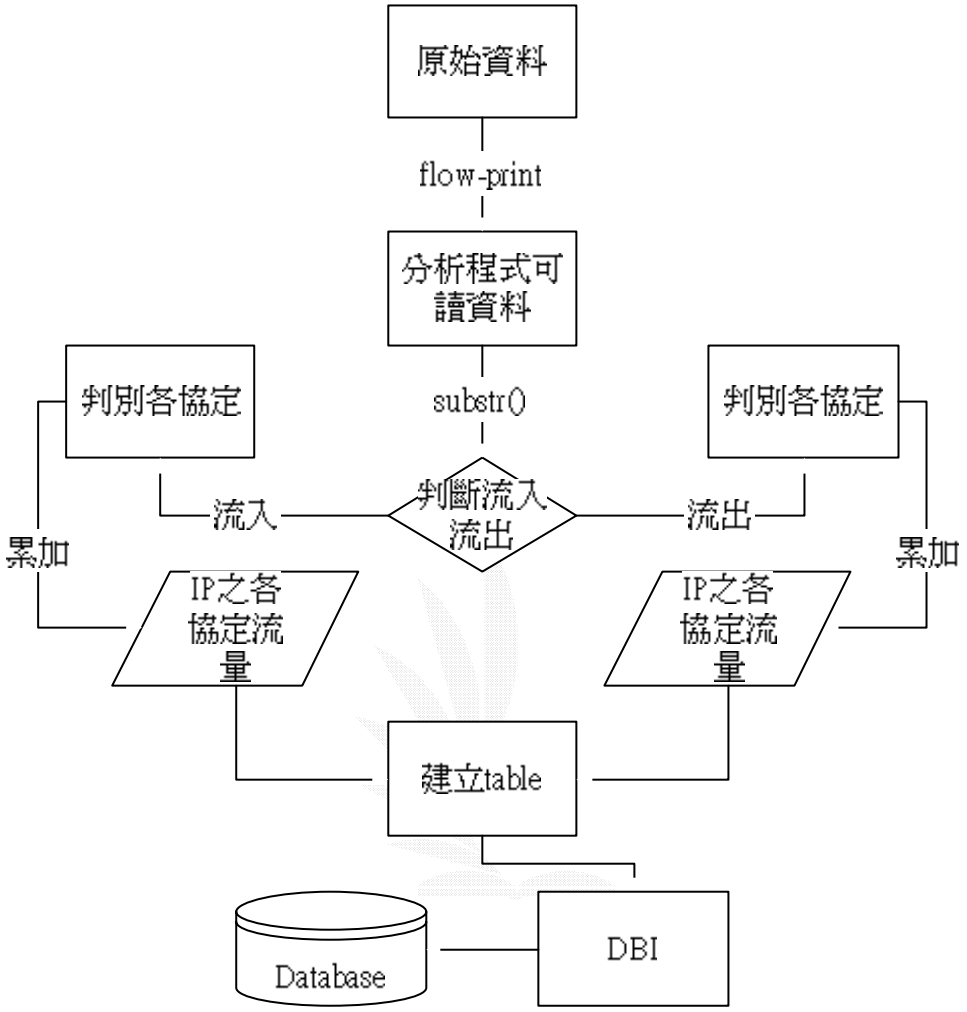
程式抓取 services **定義之副程式**

```
sub init_serv {
    my($file) = @_;
    open(FN, $file) || die $!;
    while (<FN>){
        my($name,$port) = split;
        $service{"$port"} = $name;
    }
    $service{"total"} = "total";
    $service{"others"} = "others";
    $service{"icmp"} = "icmp";
}
```

services **檔案內容** port **之定義為** 16 **進位**

Ftpdata 14
Ftp 15
Telnet 17
Smtp 19
Www 50
Pop3 6e
Tcp135 87
NetbiosNs 89
NetbiosDgm 8a
NetbiosSsn 8b
Tcp445 1bd
Tcp593 251
Edonkey 1235
Edonkey4662 1236
Edonkey4663 1237
Edonkey4664 1238
Edonkey4665 1239
Proxy c38
EzPeer6677 1a15
EzPeer6678 1a16
Kuro 1a2b
Bt6881 1ae1
Bt6882 1ae2
Bt6883 1ae3
Bt6884 1ae4
Bt6885 1ae5
Bt6886 1ae6
Bt6887 1ae7
Bt6888 1ae8
Bt6889 1ae9
BTServer 1b39

4.1.1.2 分析流量之流程



4.1.1.3 判斷流入流出方式

判斷流入流出的流量主要是由下面這兩段程式來做判斷，與交大 Netflow 文件所說使用 interface 來判斷流入流出的比較方式不同，此種比較方式是 Source IP 為校內 IP，Destination IP 為校外 IP 即為流出，反之 Source IP 為校外 IP，Destination IP 為校內 IP 則為流入。

```
sub checkinout {
    if ( isInNet($src) && !isInNet($dst) ){
        $iplist{$src}= "$src";
        &update("FCU","OUT");
    }
    elsif ( !isInNet($src) && isInNet($dst) ){
        $iplist{$dst}= "$dst";
        &update("FCU","IN");
    }
}
```

```
sub isInNet {
    my($ip) = @_;
    if ( $ip =~ /^140\.134\./ || $ip =~
/^172\.((1[6-9])|(2[0-9])|(3[0-1]))\..*/ || $ip =~ /^10\./){
        if ( $ip !~ /^140\.134\.242\./){
            return 1;
        }else {
            return 0;
        }
```

4.1.1.4 資料庫欄位表

欄位名稱	資料型態
ip	char(15)
FCUINFLOWBt6881	double(8,1)
FCUINFLOWBt6882	double(8,1)
FCUINFLOWBt6883	double(8,1)
FCUINFLOWBt6884	double(8,1)
FCUINFLOWBt6885	double(8,1)
FCUINFLOWBt6886	double(8,1)
FCUINFLOWBt6887	double(8,1)
FCUINFLOWBt6888	double(8,1)
FCUINFLOWBt6889	double(8,1)
FCUINFLOWBTServer	double(8,1)
FCUINFLOWEdonkey	double(8,1)
FCUINFLOWEdonkey4662	double(8,1)
FCUINFLOWEdonkey4663	double(8,1)
FCUINFLOWEdonkey4664	double(8,1)
FCUINFLOWEdonkey4665	double(8,1)
FCUINFLOWEzPeer6677	double(8,1)
FCUINFLOWEzPeer6678	double(8,1)
FCUINFLOWFtp	double(8,1)
FCUINFLOWFtpdata	double(8,1)
FCUINFLOWicmp	double(8,1)
FCUINFLOWKuro	double(8,1)
FCUINFLOWNetbiosDgm	double(8,1)
FCUINFLOWNetbiosNs	double(8,1)
FCUINFLOWNetbiosSsn	double(8,1)
FCUINFLOWothers	double(8,1)
FCUINFLOWPop3	double(8,1)
FCUINFLOWProxy	double(8,1)
FCUINFLOWSmtp	double(8,1)
FCUINFLOWTcp135	double(8,1)
FCUINFLOWTcp445	double(8,1)

FCUINFLOWTcp593	double(8,1)
FCUINFLOWTelnet	double(8,1)
FCUINFLOWtotal	double(8,1)
FCUINFLOWWww	double(8,1)
FCUINPKTSBt6881	double(8,1)
FCUINPKTSBt6882	double(8,1)
FCUINPKTSBt6883	double(8,1)
FCUINPKTSBt6884	double(8,1)
FCUINPKTSBt6885	double(8,1)
FCUINPKTSBt6886	double(8,1)
FCUINPKTSBt6887	double(8,1)
FCUINPKTSBt6888	double(8,1)
FCUINPKTSBt6889	double(8,1)
FCUINPKTSBTServer	double(8,1)
FCUINPKTSEdonkey	double(8,1)
FCUINPKTSEdonkey4662	double(8,1)
FCUINPKTSEdonkey4663	double(8,1)
FCUINPKTSEdonkey4664	double(8,1)
FCUINPKTSEdonkey4665	double(8,1)
FCUINPKTSEzPeer6677	double(8,1)
FCUINPKTSEzPeer6678	double(8,1)
FCUINPKTSFtp	double(8,1)
FCUINPKTSFtpdata	double(8,1)
FCUINPKTSicmp	double(8,1)
FCUINPKTSKuro	double(8,1)
FCUINPKTSNetbiosDgm	double(8,1)
FCUINPKTSNetbiosNs	double(8,1)
FCUINPKTSNetbiosSsn	double(8,1)
FCUINPKTSothers	double(8,1)
FCUINPKTSPop3	double(8,1)
FCUINPKTSProxy	double(8,1)
FCUINPKTSSmtp	double(8,1)
FCUINPKTSTcp135	double(8,1)
FCUINPKTSTcp445	double(8,1)
FCUINPKTSTcp593	double(8,1)
FCUINPKTSTelnet	double(8,1)

FCUINPKTStotal	double(8,1)
FCUINPKTSWww	double(8,1)
FCUINSIZEBt6881	double(8,1)
FCUINSIZEBt6882	double(8,1)
FCUINSIZEBt6883	double(8,1)
FCUINSIZEBt6884	double(8,1)
FCUINSIZEBt6885	double(8,1)
FCUINSIZEBt6886	double(8,1)
FCUINSIZEBt6887	double(8,1)
FCUINSIZEBt6888	double(8,1)
FCUINSIZEBt6889	double(8,1)
FCUINSIZEBTServer	double(8,1)
FCUINSIZEEdonkey	double(8,1)
FCUINSIZEEdonkey4662	double(8,1)
FCUINSIZEEdonkey4663	double(8,1)
FCUINSIZEEdonkey4664	double(8,1)
FCUINSIZEEdonkey4665	double(8,1)
FCUINSIZEEzPeer6677	double(8,1)
FCUINSIZEEzPeer6678	double(8,1)
FCUINSIZEFtp	double(8,1)
FCUINSIZEFtpdata	double(8,1)
FCUINSIZEicmp	double(8,1)
FCUINSIZEKuro	double(8,1)
FCUINSIZENetbiosDgm	double(8,1)
FCUINSIZENetbiosNs	double(8,1)
FCUINSIZENetbiosSsn	double(8,1)
FCUINSIZEothers	double(8,1)
FCUINSIZEPop3	double(8,1)
FCUINSIZEProxy	double(8,1)
FCUINSIZESmtp	double(8,1)
FCUINSIZETcp135	double(8,1)
FCUINSIZETcp445	double(8,1)
FCUINSIZETcp593	double(8,1)
FCUINSIZETelnet	double(8,1)
FCUINSIZEtotal	double(8,1)
FCUINSIZEWww	double(8,1)

FCUOUTFLOWBt6881	double(8,1)
FCUOUTFLOWBt6882	double(8,1)
FCUOUTFLOWBt6883	double(8,1)
FCUOUTFLOWBt6884	double(8,1)
FCUOUTFLOWBt6885	double(8,1)
FCUOUTFLOWBt6886	double(8,1)
FCUOUTFLOWBt6887	double(8,1)
FCUOUTFLOWBt6888	double(8,1)
FCUOUTFLOWBt6889	double(8,1)
FCUOUTFLOWBTServer	double(8,1)
FCUOUTFLOWEdonkey	double(8,1)
FCUOUTFLOWEdonkey4662	double(8,1)
FCUOUTFLOWEdonkey4663	double(8,1)
FCUOUTFLOWEdonkey4664	double(8,1)
FCUOUTFLOWEdonkey4665	double(8,1)
FCUOUTFLOWEzPeer6677	double(8,1)
FCUOUTFLOWEzPeer6678	double(8,1)
FCUOUTFLOWFtp	double(8,1)
FCUOUTFLOWFtpdata	double(8,1)
FCUOUTFLOWicmp	double(8,1)
FCUOUTFLOWKuro	double(8,1)
FCUOUTFLOWNetbiosDgm	double(8,1)
FCUOUTFLOWNetbiosNs	double(8,1)
FCUOUTFLOWNetbiosSsn	double(8,1)
FCUOUTFLOWothers	double(8,1)
FCUOUTFLOWPop3	double(8,1)
FCUOUTFLOWProxy	double(8,1)
FCUOUTFLOWSmtp	double(8,1)
FCUOUTFLOWTcp135	double(8,1)
FCUOUTFLOWTcp445	double(8,1)
FCUOUTFLOWTcp593	double(8,1)
FCUOUTFLOWTelnet	double(8,1)
FCUOUTFLOWtotal	double(8,1)
FCUOUTFLOWWww	double(8,1)
FCUOUTPKTSBt6881	double(8,1)
FCUOUTPKTSBt6882	double(8,1)

FCUOUTPKTSBt6883	double(8,1)
FCUOUTPKTSBt6884	double(8,1)
FCUOUTPKTSBt6885	double(8,1)
FCUOUTPKTSBt6886	double(8,1)
FCUOUTPKTSBt6887	double(8,1)
FCUOUTPKTSBt6888	double(8,1)
FCUOUTPKTSBt6889	double(8,1)
FCUOUTPKTSBTServer	double(8,1)
FCUOUTPKTSEdonkey	double(8,1)
FCUOUTPKTSEdonkey4662	double(8,1)
FCUOUTPKTSEdonkey4663	double(8,1)
FCUOUTPKTSEdonkey4664	double(8,1)
FCUOUTPKTSEdonkey4665	double(8,1)
FCUOUTPKTSEzPeer6677	double(8,1)
FCUOUTPKTSEzPeer6678	double(8,1)
FCUOUTPKTSFtp	double(8,1)
FCUOUTPKTSFtpdata	double(8,1)
FCUOUTPKTSicmp	double(8,1)
FCUOUTPKTSKuro	double(8,1)
FCUOUTPKTSNetbiosDgm	double(8,1)
FCUOUTPKTSNetbiosNs	double(8,1)
FCUOUTPKTSNetbiosSsn	double(8,1)
FCUOUTPKTSothers	double(8,1)
FCUOUTPKTSPop3	double(8,1)
FCUOUTPKTSProxy	double(8,1)
FCUOUTPKTSSmtp	double(8,1)
FCUOUTPKTSTcp135	double(8,1)
FCUOUTPKTSTcp445	double(8,1)
FCUOUTPKTSTcp593	double(8,1)
FCUOUTPKTSTelnet	double(8,1)
FCUOUTPKTStotal	double(8,1)
FCUOUTPKTSWww	double(8,1)
FCUOUTSIZEBt6881	double(8,1)
FCUOUTSIZEBt6882	double(8,1)
FCUOUTSIZEBt6883	double(8,1)
FCUOUTSIZEBt6884	double(8,1)

FCUOUTSIZEBt6885	double(8,1)
FCUOUTSIZEBt6886	double(8,1)
FCUOUTSIZEBt6887	double(8,1)
FCUOUTSIZEBt6888	double(8,1)
FCUOUTSIZEBt6889	double(8,1)
FCUOUTSIZEBTServer	double(8,1)
FCUOUTSIZEEdonkey	double(8,1)
FCUOUTSIZEEdonkey4662	double(8,1)
FCUOUTSIZEEdonkey4663	double(8,1)
FCUOUTSIZEEdonkey4664	double(8,1)
FCUOUTSIZEEdonkey4665	double(8,1)
FCUOUTSIZEEzPeer6677	double(8,1)
FCUOUTSIZEEzPeer6678	double(8,1)
FCUOUTSIZEFtp	double(8,1)
FCUOUTSIZEFtpdata	double(8,1)
FCUOUTSIZEicmp	double(8,1)
FCUOUTSIZEKuro	double(8,1)
FCUOUTSIZENetbiosDgm	double(8,1)
FCUOUTSIZENetbiosNs	double(8,1)
FCUOUTSIZENetbiosSsn	double(8,1)
FCUOUTSIZEothers	double(8,1)
FCUOUTSIZEPop3	double(8,1)
FCUOUTSIZEProxy	double(8,1)
FCUOUTSIZESmtp	double(8,1)
FCUOUTSIZETcp135	double(8,1)
FCUOUTSIZETcp445	double(8,1)
FCUOUTSIZETcp593	double(8,1)
FCUOUTSIZETelnet	double(8,1)
FCUOUTSIZEtotal	double(8,1)
FCUOUTSIZEWww	double(8,1)

圖表 18 資料庫欄位表

4.1.1.5 欄位解說

FCU	IN/OUT	FLOW/PKTS/SIZE	Services
prefix	**流入或是流出**	FLOW **為** Flow **數**	**各種通訊協定之名稱**
		PKTS **為** packet **數，每** 1000 **個** packet **為一單位**	
		SIZE **為流量大小，單位為** MB	

圖表 19 **資料庫欄位解說**

4.1.1.6 **節將說明如何建立** Table。

4.1.1.6 建立 Table 之副程式

```
sub createdb {
      my($net,$io,$sd,$port,$ip,$i,$j);
      $i = 0;
      foreach $net ("FCU"){
        foreach $io ("IN","OUT"){
          foreach $sd ("SRC","DST"){
            foreach $port (keys(%service)) {
              foreach $fps ("FLOW", "PKTS", "SIZE") {
                  @tablefield[$i] = sprintf("$net$io$sd$port$fps");
             if (($io eq "IN" && $sd eq "DST") || ($io eq "OUT" && $sd eq
"SRC")) {
             @tablelist[$i] = sprintf("$net$io$fps$service{$port}
DOUBLE(8,1) NOT NULL,");
             @tablefield1[$i] = sprintf("$net$io$fps$service{$port}\,");
             $i++;
             }
              }
            }
          }
        }
      }
      chop @tablefield1[$i-1];
      $create = sprintf("CREATE TABLE $tablename(ip char (15) NOT
NULL,@tablelist)TYPE=MyISAM");
      $dbh ->do($create);
}
```

4.1.2 異常行為分析

異常行為分析其作法為：在單位時間內去抓取 Flow **原始檔來做分析，持續執行以達到有效的監控。**

本系統使用之時間為 10 分鐘，每十分鐘分析程式會執行一次，經由分析 Flow 原始資料來判定是否有異常行為的發生。

異常行為分析包括有特定行為及無特定行為，以下分別介紹對有特定行為之異常行為及無特定行為之異常行為要如何來做分析。

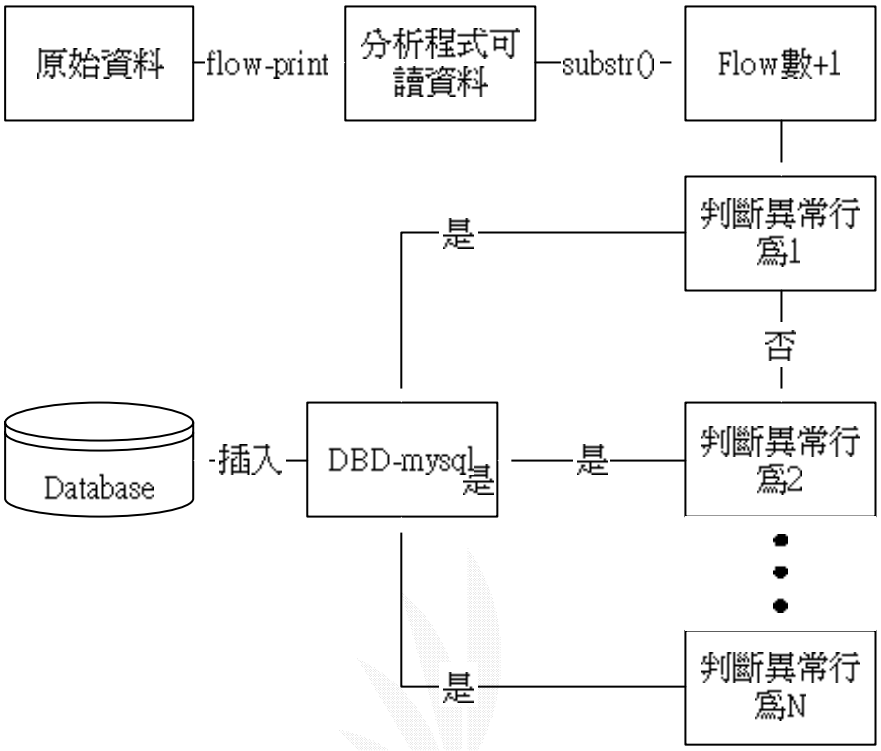
4.1.2.1 有特定行為

由於有些異常行為(像是 CodeRed)的 Flow 可以很明顯的判定其行為是病毒行為 EX：desitonation port=80, packets=3, size=144。這種便是 CodeRed 的樣式，藉由異常行為是否持續一段時間來判斷。

4.1.2.2.無特定行為

採用總 Flow 數量來判斷，若是單位時間內 Flow 數過大(EX：每秒有 50 筆 Flow 數)則大概可以判定這個 IP 為異常，有可能此 IP 使用者有在跑 P2P 軟體或是大量的抓取資料，最大的可能便為中毒，像是最近出現的 Sasser 病毒，中毒之 IP 根據觀察其 Flow 峰值大概是在每 10 分鐘有兩萬次以上的 Flow 數且流出量的大小並不大(即為單位時間內 Flow 數量大，但是每個 Flow 傳送的 Packet 很小。經由這些方式來協助判定有異常行為發生之 IP 提供給網管人員做進一步的處理。

4.1.2.3 異常行為判斷執行流程



4.1.3 網頁呈現

網頁部分可以使用的功能有

1. **異常行為查詢**

   查詢前十分鐘異常行為之發生，採用圖片表示，易讀，容易判斷，可從查詢結果進一步再去查詢個別 IP 之流量訊息。

2. **全校流量查詢**

   查詢前一日全校 TopN 之流量比例，可從查詢結果進一步再去查詢個別 IP 之流量訊息。

3. **各系所流量查詢**

   查詢前一日各系所流量資訊及比例

4. **個別 IP 流量查詢**

   查詢個別 IP 之流量資訊比例

5. **各通訊協定查詢**

   查詢各種已設定監控之協定流量分析比例，可從查詢結果進一步查詢出每種設定的通訊協定之 TopN

6. **過去總流量查詢**

   查詢過去到現在之總流量及比例

7. **過去異常行為 IP 查詢**

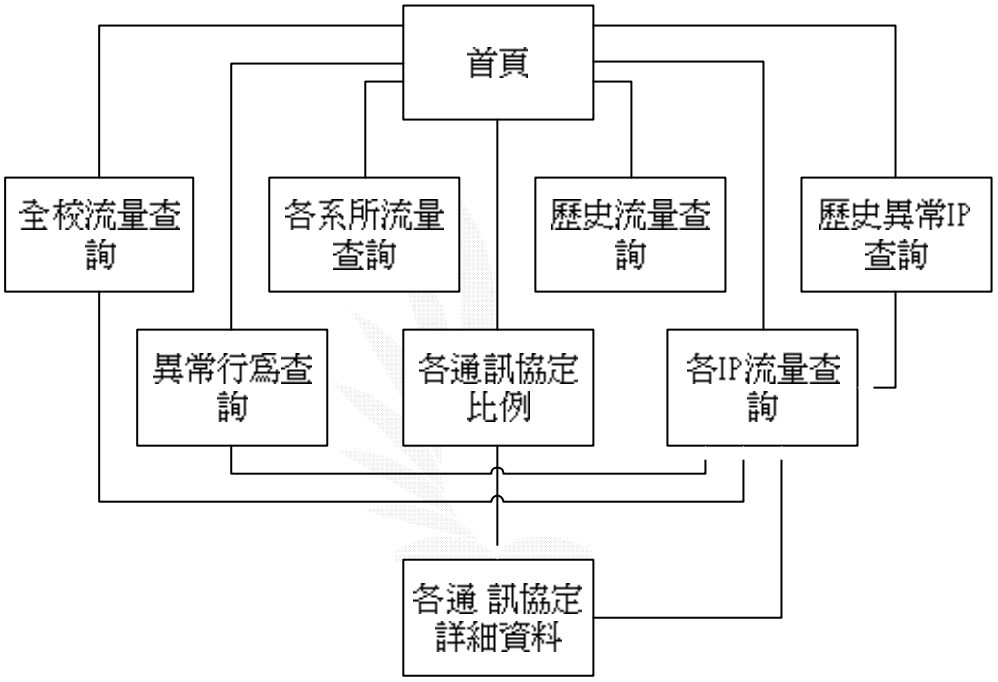
   查詢 IP 是否有過異常行為，可從查詢結果進一步再去查詢個別 IP

之流量訊息。

8. **異常行為說明**

解釋各種異常行為偵測之方式

4.1.4 網頁關係圖



圖表 21 網頁關係圖

4.2 系統展示

4.2.1 網站首頁

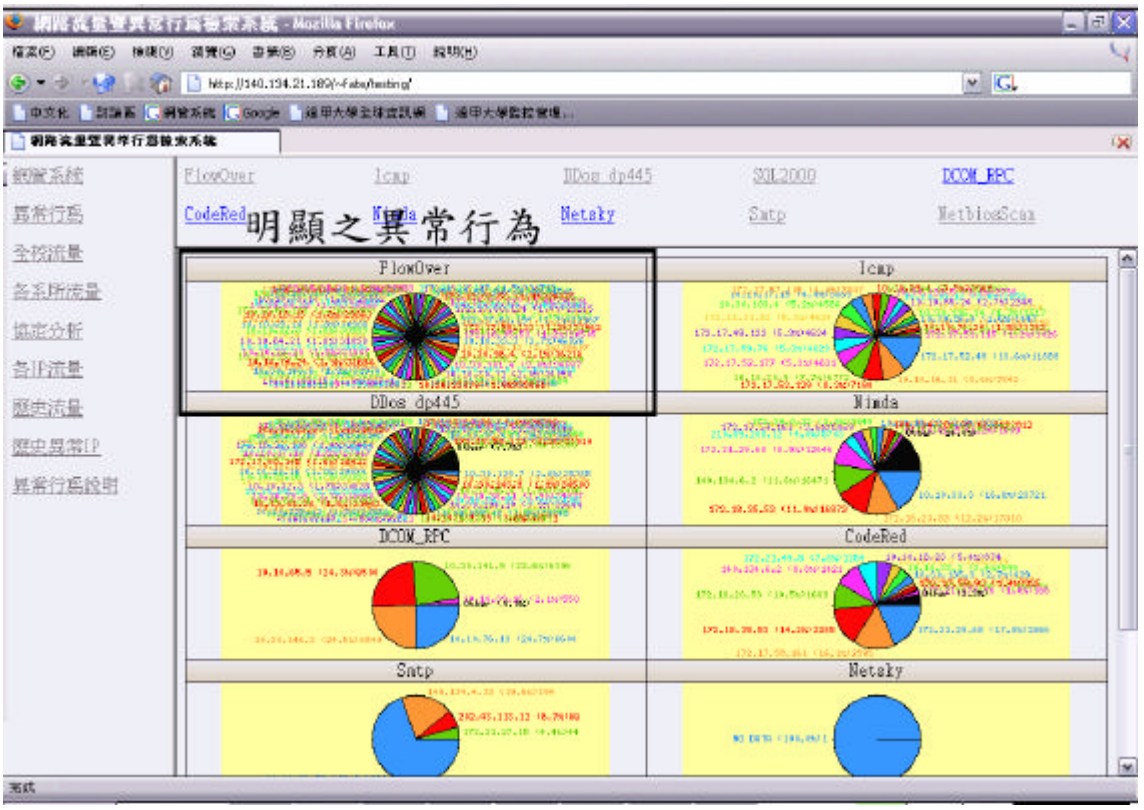
首頁放置了一些製作本系統所使用的的資源連結

若想進一步了解更多資訊可以點選到該網站去瀏覽

圖表 22 參考資源

4.2.2 異常行為

本頁放置監控中的異常行為表，若想看更詳細的資料可以點圖進入獲得更進一步的資訊



圖表 23 異常行為-1

進一步的資訊：(Next page)

點選 IP 可查詢該 IP 各通訊協定之流量。

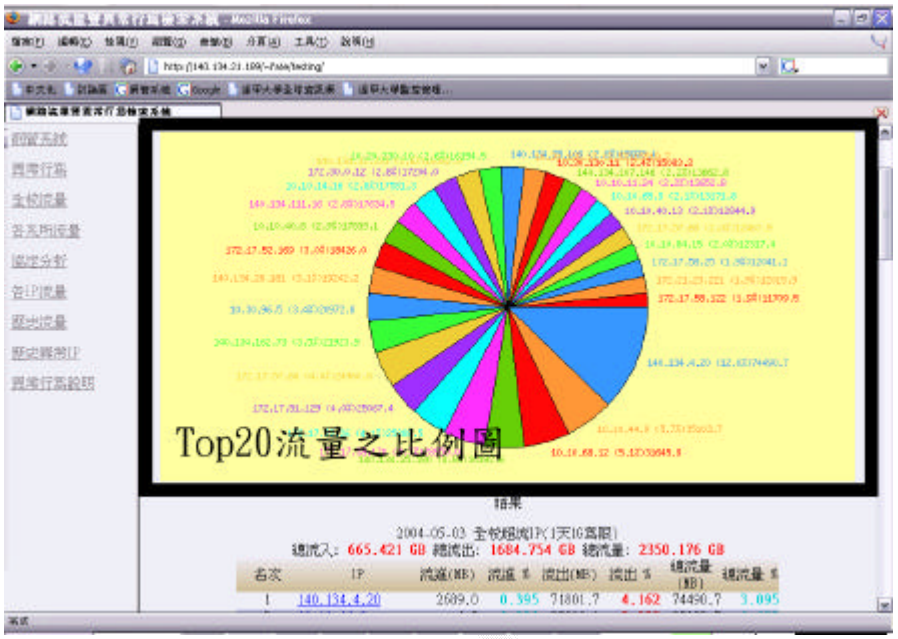
圖表 24 異常行為-2

圖表 25 異常行為-3

4.2.3 全校流量

本頁放置全校流量查詢的功能，可查詢過往日期之流量，大餅圖顯示的是 Top20 **流量之比例。**

查詢出之列表可以點選做進一步的分析。流量比例佔全校比例超過 1%**者會以紅色顯示。點選** IP **可以查詢該** IP **各通訊協定之流量。**



圖表 26 **全校流量**-1

圖表 27 **全校流量**-2

4.2.4 各系所單位流量

本頁放置各系所單位當日流量比例關係圖以及該系所總流量及佔全校總流量比例列表。

圖表 28 各系所單位流量-1

圖表 29 各系所單位流量-2

4.2.5 協定分析

本頁放置各種通訊協定佔全校流量之比例，點選通訊協定可查詢該通訊協定之 TopN。

圖表 30 協定分析

4.2.6 查詢各 IP 流量

本頁面可以輸入所想要查詢日期之 IP 流量及各通訊協定比例。

圖表 31 查詢各 IP 流量

4.2.7 歷史流量查詢

本頁面可以查詢歷史流量紀錄，同時也顯示歷史流量紀錄比例圖。

圖表 32 歷史流量查詢

4.2.8 歷史異常 IP 查詢

此頁面提供輸入查詢 IP 是否有過異常行為之紀錄，對於查詢該 IP 是否被鎖定挺好用，圖例是已經被封鎖的 IP，中的是 SQL Slammer 這隻 Worm。由圖中可看到該 IP 的 Flow 數每 10 分鐘有過上百萬次，且送出的資料量為 500 多 MB。

圖表 33 歷史異常 IP 查詢

4.3 心得感想

早先嘗試開始實做的時候是都使用交大 Netflow 文件所說的方式來做，由於找不到學長可以問，許多都是自己懵懵懂懂在跌跌撞撞中一直嘗試，差不多將交大 Netflow 文件所說的系統架起來之時，卻發生不小心將檔案都刪除的意外，於是所有都歸零。同時間有再另外搜尋與 Netflow 相關的文章，發現在國外有個 Flow-Tools 可以拿來使用，經過裝起來時使用之後發現 Flow-Tools 的效能比起交大 Netflow 文件所提供的收集程式好，交大 Netflow 提供的收集程式會持續讓 CPU Loading 在 100%，早先以為是所使用的機器等級太差(Pentium-166)，經過換上 Flow-Tools 之後發現 CPU Loading 大大降低，且其輸出格式也與交大 Netflow 文件使用的格式類似，於是便採用新的收集程式來做。

有了新的收集程式之後便開始重新再架過一次系統，由於早先實驗用收集的流量資訊為逢甲大學校園網路對外骨幹中的一小部分，且其流量大都是流出，這個缺點讓我無法確定自己的程式執行結果是否有架設成功。在詢問系統維運組余組長之後，余組長答應使用較好的 PC(Pentium3-1G)來實作且收集的流量資訊來源為校內工學院的 Router(Cisco Catalyst 5500)，從這台機器的流入流出很明顯可以看出差異，在經過一段時間的嘗試之後，便開始參考交大 Netflow 文件裏的分析程式來實作專題所要使用的程式。

在經過一段時間的摸索與測試及改進之後，整個系統的運作算大致上 OK，接下來所要作的就是將運算後資料作儲存及呈現，於是有了放在資料庫裡的想法，藉由 PHP 與 MySQL 的結合讓我可以對網頁的呈現做很大的彈性，也做出之前沒想到的功能。

後來指導老師建議應該有個圖表來表示各流量之間的關係圖會更好。於是便開始找尋如何去使用 PHP 畫圓餅圖，發現 GD 是在 PHP 上畫圖唯一的選擇，雖然知道 GD 在畫圖方面的能力很強，但是使用上太過複雜。後來在 http://www.silisoftware.com/找到 phPie()這個 php script，終於解決了困擾我許久關於如何畫圓餅圖的問題。

自從網站架構越來越完整，細部的需求也一值增加，實作期間也碰過好幾次病毒流行，如開學病毒潮及最近的 Sasser 病毒肆虐，驗證本系統可以對於異常行為的 IP 作有效的觀察，進而可以提供網管人員去做處理，實在很欣慰，辛苦做的專題可以派上用場真的很高興。

參考資料

[1]Jon Orwant, Perl 5 Interactive Course, The Waite Group,Inc 1996
[2] Schwartz, Christiansen, Learning Perl, 2/e, O'REILLY 1999

[3]劉國正,林景毅,PHP4 入門實務手冊,松崗電腦圖書資料股份有限公司,2000

[4]陳俊宏,PHP4 網站實作-深度研究篇,旗標出版股份有限公司,2000

[5]位元文化編著,PHP4+MySQL4 動態網頁入門實務,文魁資訊股份有限公司,2002

參考網址

[1] Flow-Tools http://www.splintered.net/sw/flow-tools/
[2] How to install and configure DBD::mysql
http://search.cpan.org/src/JWIED/DBD-mysql-2.1026/INSTALL.html

[3]交大 Netflow 文件 http://netflow.nctu.edu.tw/netflow.html

[4]如何建置 Netflow http://www.tn.edu.tw/sammy/netflow/setup.htm

[5] NetFlow Services Solutions Guide
http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/nfwhite.htm
[6] NetflowExporter Project http://sunsite.cc.ncu.edu.tw/NetflowExporter/
[7]phPie php sript
http://www.silisoftware.com/scripts/index.php?scriptname=phPie
[8]Perl Doc http://www.perldoc.com/

[9]PHP5 網管實驗室 http://www.perldoc.com/

[10]網站建置百寶箱 http://dob.tnc.edu.tw/

程式碼

每日計算總流量之程式原始碼

```
#!/usr/local/bin/perl5

use DBI;
my  $dsn            =  "DBI:mysql:dbname:localhost";      #data  source  name
my  $user           =  "dbuser";                            #username
my  $password       =  "password";                           #password
my  ($dbh,  $rows);                                         #database  handle

$dbh = DBI->connect($dsn, $user, $password);

($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time-86400);
$mon++;
$year+=1900;

#format date to 2-digit. Ex: 2 -> 02
$mon = sprintf("%02d",$mon);
$mday = sprintf("%02d",$mday);

#setting up path
$rawdir = "/netflow";
$flowprint = "/usr/local/netflow/bin/flow-print";
$dir            =  "/netflow/prog";
$outputdir = "/netflow/log";
$protfile = "$dir/protocols";
$servfile = "$dir/services";
$tablename = "ff$year$mon$mday";

@gotfilename = `ls $rawdir/ft-v05.$year-$mon-$mday.*`;

#log
open LOG, ">$outputdir/process.log.$year$mon$mday" or die $!;
```

```
$starttime = localtime();
&init_prot($protfile);
&init_serv($servfile);

$dbh->do("DROP TABLE IF EXISTS `$tablename`");

#create tables
&createdb;

for ($i=0;$i<scalar @gotfilename;$i++) {
    $processtime = localtime();
    chomp @gotfilename[$i];
    printf LOG "NOW %3s %45s AT %20s\n",$i,@gotfilename[$i],$processtime;
    open FLOWDATA, "$flowprint -f0 < @gotfilename[$i] |" || die $!;
    while (<FLOWDATA>) {
        chomp;
        $sif   = substr($_, 0,   index($_, ' ', 0));
        $src   = substr($_, 5,   index($_, ' ', 5)  - 5);
        $dif   = substr($_, 22, index($_, ' ', 22) - 22);
        $dst   = substr($_, 27, index($_, ' ', 27) - 27);
        $proto = substr($_, 45, index($_, ' ', 45) - 45);
        $sp    = substr($_, 48, index($_, ' ', 48) - 48);
        $dp    = substr($_, 53, index($_, ' ', 53) - 53);
        $pkts  = substr($_, 59, index($_, ' ', 59) - 59);
        $size  = substr($_, 70, index($_, ' ', 70) - 70);

        $pkts /= 1000;
        $size /= 1048576; #(1024*1024);

        &checkinout;
    }
}


#===============
$insertdbtime = localtime();
print LOG "Start insert data to DB at $insertdbtime\n";
```

```
&insertdata;
&dailysizetotal;
#execute other perl program
do "$dir/ipdeny.pl";
do "$dir/flowchart.pl";

$dbh->disconnect();

$endtime = localtime();
print LOG "start :: $starttime\n";
print LOG "end   :: $endtime\n";

sub insertdata {
    my($i,$j);
    $i = 0;
    $j = 0;
    $k = 0;
    @sorted_hash = sort keys %iplist;
    $length = scalar @sorted_hash;
    $tablelistlength = scalar @tablelist;
    for ($ipno;$ipno<$length;$ipno++){
        for ($j=0;$j<$tablelistlength;$j++){
            $tt = ${@tablefield[$j]}{@sorted_hash[$ipno]};
            @value[$j] = "\"$tt\",";
        }
        chop @value[$j-1];
        $dbh->do("INSERT INTO $tablename(ip\,@tablefield1)
VALUES(\"@sorted_hash[$ipno]\",@value)");
    }
}

sub isInNet {
    my($ip) = @_;
    if ( $ip =~ /^140\.134\./ || $ip =~ /^172\.((1[6-9])|(2[0-9])|(3[0-1]))\..*/ || $ip =~
/^10\./){
        if ( $ip !~ /^140\.134\.242\./){
            return 1;
```

```
            }else {
                return 0;
            }
        } else {
            return 0;
        }
}

sub checkinout {
        if ( isInNet($src) && !isInNet($dst) ){
            $iplist{$src}= "$src";
            &update("FCU","OUT");
        }
        elsif ( !isInNet($src) && isInNet($dst) ){
            $iplist{$dst}= "$dst";
            &update("FCU","IN");
        }
}

sub update{
        my($net,$io) = @_;
        if ($proto == '06' || $proto == '11') {
            if ($service{"$sp"}) {
                $port = $sp;
            }
            elsif ($service{"$dp"}) {
                $port = $dp;
            }
            else {
                $port = "others";
            }
        }
        elsif ($proto == 01) {
            $port = "icmp";
        }

        if ($io eq "IN"){
#           my($a, $b, $c, $d) = split(/\./,"$src");
```

```
#           $src = sprintf("%d.%d.%d", $a, $b, $c);
#           ${$net.$io.SRC.$port.FLOWC}{"$src"} ++;
#           ${$net.$io.SRC.$port.PKTSC}{"$src"} += $pkts;
#           ${$net.$io.SRC.$port.SIZEC}{"$src"} += $size;
            ${$net.$io.DST.$port.FLOW}{"$dst"} ++;
            ${$net.$io.DST.$port.PKTS}{"$dst"} += $pkts;
            ${$net.$io.DST.$port.SIZE}{"$dst"} += $size;
            ${$net.$io.DST.total.FLOW}{"$dst"} ++;
            ${$net.$io.DST.total.PKTS}{"$dst"} += $pkts;
            ${$net.$io.DST.total.SIZE}{"$dst"} += $size;
        }
        else {
#           my($a, $b, $c, $d) = split(/\./,"$dst");
#           $dst = sprintf("%d.%d.%d", $a, $b, $c);
            ${$net.$io.SRC.$port.FLOW}{"$src"} ++;
            ${$net.$io.SRC.$port.PKTS}{"$src"} += $pkts;
            ${$net.$io.SRC.$port.SIZE}{"$src"} += $size;
            ${$net.$io.SRC.total.FLOW}{"$src"} ++;
            ${$net.$io.SRC.total.PKTS}{"$src"} += $pkts;
            ${$net.$io.SRC.total.SIZE}{"$src"} += $size;
#           ${$net.$io.DST.$port.FLOWC}{"$dst"} ++;
#           ${$net.$io.DST.$port.PKTSC}{"$dst"} += $pkts;
#           ${$net.$io.DST.$port.SIZEC}{"$dst"} += $size;
        }
    }

    sub init_prot {
        my($file) = @_;
        open (FN,$file) || die $!;
        while(<FN>){
            my($name,$proto) = split;
            $protocol{"$proto"} = $name;
        }
    }

    sub init_serv {
        my($file) = @_;
        open(FN, $file) || die $!;
```

```
        while (<FN>){
                my($name,$port) = split;
                $service{"$port"} = $name;
        }
        $service{"total"} = "total";
        $service{"others"} = "others";
        $service{"icmp"} = "icmp";
}

sub createdb {
        my($net,$io,$sd,$port,$ip,$i,$j);
        $i = 0;
        foreach $net ("FCU"){
         foreach $io ("IN","OUT"){
           foreach $sd ("SRC","DST"){
            foreach $port (keys(%service)) {
             foreach $fps ("FLOW", "PKTS", "SIZE") {
                    @tablefield[$i] = sprintf("$net$io$sd$port$fps");
              if (($io eq "IN" && $sd eq "DST") || ($io eq "OUT" && $sd eq "SRC")) {
              @tablelist[$i] = sprintf("$net$io$fps$service{$port} DOUBLE(8,1) NOT
NULL,");
              @tablefield1[$i] = sprintf("$net$io$fps$service{$port}\,");
              $i++;
              }
              }
             }
            }
           }
        }
        chop @tablefield1[$i-1];
        $create = sprintf("CREATE TABLE $tablename(ip char (15) NOT
NULL,@tablelist)TYPE=MyISAM");
        $dbh ->do($create);
}

sub dailysizetotal {
        my($insize,$outsize,$totalsize);
        $sth = $dbh->prepare("SELECT
```

```
sum(fcuinsizetotal),sum(fcuoutsizetotal),sum(fcuinsizetotal)+sum(fcuoutsizetotal)
FROM $tablename");
        $sth->execute();
        while(($insize ,$outsize ,$totalsize) = $sth->fetchrow_array()) {
                $dbh->do("insert into DailyTotalSize(date,insize,outsize,totalsize)
values('$year$mon$mday','$insize','$outsize','$totalsize') ");
        }
}
```

每十分鐘計算異常行為之程式原始碼

```
#!/usr/local/bin/perl5

use DBI;
my $dsn          = "DBI:mysql:netflow:localhost";      #data source name
my $user         = "root";                              #username
my $password     = "sysnet";                           #password
my ($dbh, $rows);                                      #database handle

$dbh = DBI->connect($dsn, $user, $password);

($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time-600);
$mon++;
$year+=1900;

$nowtime = (time+60*60*8)%86400;
$suffix = ($nowtime-(86400/144))/(86400/144)%6;

if ($min >= 0 && $min <= 9 ){
        $suffix = 0;
}

$mon = sprintf("%02d",$mon);
$mday = sprintf("%02d",$mday);
$hour = sprintf("%02d",$hour);
$min = sprintf("%02d",$min);
```

```
$rawdir = "/netflow";
$flowprint = "/usr/local/netflow/bin/flow-print";
$dir           =  "/netflow/prog";
$outputdir = "/netflow/log";

$protfile = "$dir/protocols";
$servfile = "$dir/services";
$ipfile     =  "$dir/iplist";
$tablename = "virus";

$sth = $dbh->prepare("SELECT
`icmp`,`445`,`59a`,`87`,`codered`,`nimda`,`netsky`,`smtp`,`netbios` FROM
virus_cont");
$sth->execute();
($cont_icmp,$cont_445,$cont_59a,$cont_87,$cont_codered,$cont_nimda,$cont_netsky,
$cont_smtp,$cont_netbios) = $sth->fetchrow_array();

$sth->finish();

$filename = sprintf("ft-v05.%02d-%02d-%02d.%02d%d*", $year, $mon, $mday, $hour,
$suffix);
@gotfilename = `ls $rawdir/$filename`;

open LOG, ">$outputdir/abnormal.$mon$mday$hour-$suffix" or die $!;

$starttime = localtime();
&init_prot($protfile);
&init_serv($servfile);

for ($i=0;$i<scalar @gotfilename;$i++) {
      $processtime = localtime();
      chomp @gotfilename[$i];
      printf LOG "NOW %3s %45s AT %20s\n",$i,@gotfilename[$i],$processtime;
      open FLOWDATA, "$flowprint -f0 < @gotfilename[$i] |" || die $!;
      while (<FLOWDATA>) {
            chomp;
            $sif    = substr($_, 0,   index($_, ' ', 0));
            $src    = substr($_, 5,   index($_, ' ', 5)   - 5);
```

```
            $dif    = substr($_, 22, index($_, ' ', 22) - 22);
            $dst    = substr($_, 27, index($_, ' ', 27) - 27);
            $proto = substr($_, 45, index($_, ' ', 45) - 45);
            $sp     = substr($_, 48, index($_, ' ', 48) - 48);
            $dp     = substr($_, 53, index($_, ' ', 53) - 53);
            $pkts   = substr($_, 59, index($_, ' ', 59) - 59);
            $size   = substr($_, 70, index($_, ' ', 70) - 70);

#           $size /= 1024;
            &abnormal;
            $totalflow10min++;
        }
}


#================
$insertdbtime = localtime();


#$dbh->do("DROP TABLE IF EXISTS `$tablename`");

#&createtable;

#&expire;

&out;


$dbh->disconnect();

print LOG "10 min total flow count $totalflow10min\n";
$endtime = localtime();
print LOG "start :: $starttime\n";
print LOG "end    :: $endtime\n";

close LOG;

sub expire {
```

```
        my ($expired,$mon,$mday,$hour,$min,$year);
        ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time-3600);
        $mon++;
        $year+=1900;

        $year = sprintf("%02d",$year%100);
        $mon = sprintf("%02d",$mon);
        $mday = sprintf("%02d",$mday);
        $hour = sprintf("%02d",$hour);
        $min = sprintf("%02d",$min);

        print "$year$mon$mday$hour$min\n";

        $expired = sprintf("$year$mon$mday$hour$min");

        $dbh->do("delete from virus where date < \'$expired\'");
}

sub abnormal{
        if (isInNet($src)) {
                $flow_cont{$src}++;
                $flow_cont_pkts{$src} += $pkts;
                $flow_cont_size{$src} += $size;
        }
        if ( $proto == 01 && $sp == 0 && $dp == 800){
                $abnormal_cont_icmp_flow{$src}++;
                $abnormal_cont_icmp_pkts{$src} += $pkts;
                $abnormal_cont_icmp_size{$src} += $size;
        }
        if ( $proto == '06' && $dp == '1bd') {
                $abnormal_cont_445_flow{$src}++;
                $abnormal_cont_445_pkts{$src} += $pkts;
                $abnormal_cont_445_size{$src} += $size;
        }
        if ( $proto == '11' && $dp == '59a') {
                $abnormal_cont_59a_flow{$src}++;
                $abnormal_cont_59a_pkts{$src} += $pkts;
                $abnormal_cont_59a_size{$src} += $size;
```

```
        }
        if ( $proto == '06' && $dp == '87' && $pkts == '1' && $size =='48') {
            $abnormal_cont_87_flow{$src}++;
            $abnormal_cont_87_pkts{$src} += $pkts;
            $abnormal_cont_87_size{$src} += $size;
        }
        if ( $dp == '50' && $pkts == '3' && $size == '144') {
            $abnormal_cont_codered_flow{$src}++;
            $abnormal_cont_codered_pkts{$src} += $pkts;
            $abnormal_cont_codered_size{$src} += $size;
        }
        if ( $dp == '50' ) {
            $abnormal_cont_nimda_flow{$src}++;
            $abnormal_cont_nimda_pkts{$src} += $pkts;
            $abnormal_cont_nimda_size{$src} += $size;
        }
        if ( $dp == '19' && $pkts == '2' && $size == '96') {
            $abnormal_cont_netsky_flow{$src}++;
            $abnormal_cont_netsky_pkts{$src} += $pkts;
            $abnormal_cont_netsky_size{$src} += $size;
        }
        if ( $dp == '19') {
            $abnormal_cont_smtp_flow{$src}++;
            $abnormal_cont_smtp_pkts{$src} += $pkts;
            $abnormal_cont_smtp_size{$src} += $size;
        }
        if ( $dp == '8b' && $pkts == '1' && $size == '48') {
            $abnormal_cont_netbios_flow{$src}++;
            $abnormal_cont_netbios_pkts{$src} += $pkts;
            $abnormal_cont_netbios_size{$src} += $size;
        }
    }

    sub out {
        my($src);
        foreach $src(keys %flow_cont){
            if ($flow_cont{$src} > 10000){
                printf LOG ("on %15s occur %9s times flow too
```

```
much\n",$src,$flow_cont{$src});
                $dbh->do("INSERT INTO
$tablename(ip\,flow\,pkts\,size\,kind\,10minoutflow\,10minoutpkts\,10minoutsize)
VALUES(\'$src\'\,\'$flow_cont{$src}\'\,\'$flow_cont_pkts{$src}\'\,\'$flow_cont_size{$s
rc}\'\,\'10min
Flow\>10000\'\,\'$flow_cont{$src}\'\,\'$flow_cont_pkts{$src}\'\,\'$flow_cont_size{$src
}\')");
            }
        }
        foreach $src(keys %abnormal_cont_icmp_flow){
            if ($abnormal_cont_icmp_flow{$src} > $cont_icmp){
                printf LOG ("on %18s occur %8s times  icmp(dst:2048) pkts %10d,size
%5.2f\n",$src,$abnormal_cont_icmp_flow{$src},$abnormal_cont_icmp_pkts{$src},$a
bnormal_cont_icmp_size{$src});
                $dbh->do("INSERT INTO
$tablename(ip\,flow\,pkts\,size\,kind\,10minoutflow\,10minoutpkts\,10minoutsize)
VALUES(\'$src\'\,\'$abnormal_cont_icmp_flow{$src}\'\,\'$abnormal_cont_icmp_pkts{$
src}\'\,\'$abnormal_cont_icmp_size{$src}\'\,\'icmp\'\,\'$flow_cont{$src}\'\,\'$flow_cont_
pkts{$src}\'\,\'$flow_cont_size{$src}\')");
            }
        }
        foreach $src(keys %abnormal_cont_445_flow){
            if ($abnormal_cont_445_flow{$src} > $cont_445){
                printf LOG ("on %18s occur %8s times  445 port(1bd) pkts %10d,size
%5.2f\n",$src,$abnormal_cont_445_flow{$src},$abnormal_cont_445_pkts{$src},$abn
ormal_cont_445_size{$src});
                $dbh->do("INSERT INTO
$tablename(ip\,flow\,pkts\,size\,kind\,10minoutflow\,10minoutpkts\,10minoutsize)
VALUES(\'$src\'\,\'$abnormal_cont_445_flow{$src}\'\,\'$abnormal_cont_445_pkts{$src
}\'\,\'$abnormal_cont_445_size{$src}\'\,\'445\'\,\'$flow_cont{$src}\'\,\'$flow_cont_pkts{
$src}\'\,\'$flow_cont_size{$src}\')");
            }
        }
        foreach $src(keys %abnormal_cont_59a_flow){
            if ($abnormal_cont_59a_flow{$src} > $cont_59a){
                printf LOG ("on %18s occur %8s times  1434 port(59a) pkts %10d,size
%5.2f\n",$src,$abnormal_cont_59a_flow{$src},$abnormal_cont_59a_pkts{$src},$abno
rmal_cont_59a_size{$src});
```

```
                $dbh->do("INSERT INTO
$tablename(ip\,flow\,pkts\,size\,kind\,10minoutflow\,10minoutpkts\,10minoutsize)
VALUES(\'$src\'\,\'$abnormal_cont_59a_flow{$src}\'\,\'$abnormal_cont_59a_pkts{$src
}\'\,\'$abnormal_cont_59a_size{$src}\'\,\'59a\'\,\'$flow_cont{$src}\'\,\'$flow_cont_pkts{
$src}\'\,\'$flow_cont_size{$src}\')");
            }
        }
        foreach $src(keys %abnormal_cont_87_flow){
            if ($abnormal_cont_87_flow{$src} > $cont_87){
                printf LOG ("on %18s occur %8s times  135 port(87) pkts %10d,size
%5.2f\n",$src,$abnormal_cont_87_flow{$src},$abnormal_cont_87_pkts{$src},$abnor
mal_cont_87_size{$src});
                $dbh->do("INSERT INTO
$tablename(ip\,flow\,pkts\,size\,kind\,10minoutflow\,10minoutpkts\,10minoutsize)
VALUES(\'$src\'\,\'$abnormal_cont_87_flow{$src}\'\,\'$abnormal_cont_87_pkts{$src}\'
\,\'$abnormal_cont_87_size{$src}\'\,\'135\'\,\'$flow_cont{$src}\'\,\'$flow_cont_pkts{$sr
c}\'\,\'$flow_cont_size{$src}\')");
            }
        }
        foreach $src(keys %abnormal_cont_codered_flow){
            if ($abnormal_cont_codered_flow{$src} > $cont_codered){
                printf LOG ("on %18s occur %8s times  CodeRed pkts %10d,size
%5.2f\n",$src,$abnormal_cont_codered_flow{$src},$abnormal_cont_codered_pkts{$sr
c},$abnormal_cont_codered_size{$src});
                $dbh->do("INSERT INTO
$tablename(ip\,flow\,pkts\,size\,kind\,10minoutflow\,10minoutpkts\,10minoutsize)
VALUES(\'$src\'\,\'$abnormal_cont_codered_flow{$src}\'\,\'$abnormal_cont_codered_
pkts{$src}\'\,\'$abnormal_cont_codered_size{$src}\'\,\'CodeRed\'\,\'$flow_cont{$src}\'\,
\'$flow_cont_pkts{$src}\'\,\'$flow_cont_size{$src}\')");
            }
        }
        foreach $src(keys %abnormal_cont_nimda_flow){
            if ($abnormal_cont_nimda_flow{$src} > $cont_nimda){
                printf LOG ("on %18s occur %8s times  Nimda pkts %10d,size
%5.2f\n",$src,$abnormal_cont_nimda_flow{$src},$abnormal_cont_nimda_pkts{$src},
$abnormal_cont_nimda_size{$src});
                $dbh->do("INSERT INTO
$tablename(ip\,flow\,pkts\,size\,kind\,10minoutflow\,10minoutpkts\,10minoutsize)
```

```
VALUES(\'$src\'\,\'$abnormal_cont_nimda_flow{$src}\'\,\'$abnormal_cont_nimda_pkts
{$src}\'\,\'$abnormal_cont_nimda_size{$src}\'\,\'Nimda\'\,\'$flow_cont{$src}\'\,\'$flow_
cont_pkts{$src}\'\,\'$flow_cont_size{$src}\')");
            }
        }
        foreach $src(keys %abnormal_cont_netsky_flow){
            if ($abnormal_cont_netsky_flow{$src} > $cont_netsky){
                printf LOG ("on %18s occur %8s times  NetSky pkts %10d,size
%5.2f\n",$src,$abnormal_cont_netsky_flow{$src},$abnormal_cont_netsky_pkts{$src},
$abnormal_cont_netsky_size{$src});
                $dbh->do("INSERT INTO
$tablename(ip\,flow\,pkts\,size\,kind\,10minoutflow\,10minoutpkts\,10minoutsize)
VALUES(\'$src\'\,\'$abnormal_cont_netsky_flow{$src}\'\,\'$abnormal_cont_netsky_pkt
s{$src}\'\,\'$abnormal_cont_netsky_size{$src}\'\,\'Netsky\'\,\'$flow_cont{$src}\'\,\'$flow
_cont_pkts{$src}\'\,\'$flow_cont_size{$src}\')");
            }
        }
        foreach $src(keys %abnormal_cont_smtp_flow){
            if ($abnormal_cont_smtp_flow{$src} > $cont_smtp){
                printf LOG ("on %18s occur %8s times  Smtp pkts %10d,size
%5.2f\n",$src,$abnormal_cont_smtp_flow{$src},$abnormal_cont_smtp_pkts{$src},$ab
normal_cont_smtp_size{$src});
                $dbh->do("INSERT INTO
$tablename(ip\,flow\,pkts\,size\,kind\,10minoutflow\,10minoutpkts\,10minoutsize)
VALUES(\'$src\'\,\'$abnormal_cont_smtp_flow{$src}\'\,\'$abnormal_cont_smtp_pkts{$
src}\'\,\'$abnormal_cont_smtp_size{$src}\'\,\'Smtp\'\,\'$flow_cont{$src}\'\,\'$flow_cont_
pkts{$src}\'\,\'$flow_cont_size{$src}\')");
            }
        }
        foreach $src(keys %abnormal_cont_netbios_flow){
                if ($abnormal_cont_netbios_flow{$src} > $cont_netbios){
                    printf LOG ("on %18s occur %8s times  Netbios pkts
%10d,size
%5.2f\n",$src,$abnormal_cont_netbios_flow{$src},$abnormal_cont_netbios_pkts{$src
},$abnormal_cont_netbios_size{$src});
                    $dbh->do("INSERT INTO
$tablename(ip\,flow\,pkts\,size\,kind\,10minoutflow\,10minoutpkts\,10minoutsize)
VALUES(\'$src\'\,\'$abnormal_cont_netbios_flow{$src}\'\,\'$abnormal_cont_netbios_pk
```

```
ts{$src}\'\,\'$abnormal_cont_netbios_size{$src}\'\,\'NetbiosScan\'\,\'$flow_cont{$src}\'\,
\'$flow_cont_pkts{$src}\'\,\'$flow_cont_size{$src}\')");
            }
        }
}

sub createtable {
        $dbh->do("CREATE TABLE `virus` (`date` TIMESTAMP (10) DEFAULT '0', `ip`
CHAR (15) DEFAULT '0' NOT NULL, `flow` MEDIUMINT (8) UNSIGNED
DEFAULT '0' NOT NULL, `pkts` MEDIUMINT (8) UNSIGNED DEFAULT '0' NOT
NULL, `size` DOUBLE (8,1) DEFAULT '0' NOT NULL, `kind` CHAR (10) DEFAULT
'0' NOT NULL)  TYPE = MyISAM");
}

sub isInNet {
        my($ip) = @_;
        if ( $ip =~ /^140\.134\./ || $ip =~ /^172\./ || $ip =~ /^10\./){
            return 1;
        } else {
            return 0;
        }
}



sub init_prot {
        my($file) = @_;
        open (FN,$file) || die $!;
        while(<FN>){
            my($name,$proto) = split;
            $protocol{"$proto"} = $name;
        }
}

sub init_serv {
        my($file) = @_;
        open(FN, $file) || die $!;
        while (<FN>){
            my($name,$port) = split;
```

```
            $service{"$port"} = $name;
        }
        $service{"total"} = "total";
        $service{"others"} = "others";
        $service{"icmp"} = "icmp";
}

sub ipdeny {#added at 2004/02/25
        my($ip,$insize,$outsize,$totalsize,$low,$middle,$high);
        $low = 1000;
        $middle = 2000;
        $high = 3000;
        $sth = $dbh->prepare("SELECT ip, fcuinsizetotal,fcuoutsizetotal FROM
$tablename ORDER by ip");
        $sth->execute();
        while(($ip ,$insize ,$outsize) = $sth->fetchrow_array()) {
            $totalsize = $insize+$outsize;
            if ($totalsize > $low && $totalsize <= $middle){
                $dbh->do("INSERT INTO IpDeny(ip,status_id,date2)
VALUES('$ip','1','$year$mon$mday')");
            }elsif ($totalsize > $middle && $totalsize <= $high){
                $dbh->do("INSERT INTO IpDeny(ip,status_id,date2)
VALUES('$ip','2','$year$mon$mday')");
            }elsif ($totalsize > $high ){
                $dbh->do("INSERT INTO IpDeny(ip,status_id,date2)
VALUES('$ip','3','$year$mon$mday')");
            }
        }
        $sth->finish();
}
```