網站入侵偵測系統之分析與研究

施東河、黃于爵
雲林科技大學資訊管理研究所

摘要

網路安全對 MIS 資訊人員來說非常重要，然而技術人員的技術與知識越來越難跟上不斷出現的安全漏洞與攻擊手法。在日新月異的資訊安全問題中，如何及時找出網路安全的弱點，適時地、有效率的定期評估稽核自我網路安全狀況，成了當前企業與 MIS 資訊人員首要關切的議題。本文除廣泛搜集現有市面上所有的網站入侵行爲與攻擊軟體外，並根據國內外入侵偵測系統的探討，提出綜合 Network-based misuse model 與 Host-based anomaly model 的 WIDS 入侵偵測系統。本研究採用類神經網路中的自組織映射圖網路架構，並試圖提出一套具有學習能力的 WIDS 網站入侵偵測系統，期望能解決日新月異不斷翻新的攻擊手法，使得系統得以自我學習保護，使駭客攻擊傷害能降到最小。經過實證，本研究之入侵偵測系統正確率高達 86% 以上。

關鍵詞：入侵偵測系統、駭客、類神經網路、自組織映射圖網路

Analysis and Study of Web Intrusion Detection System

Dong-Her Shih, Yu-Chei Hwang
Department of Information Management, National Yunlin University
of Science and Technology

ABSTRACT

Network security to MIS personnel is very important. However, the technique and knowledge of the technician is getting hard to catch up with more and more secure leak and attack skill. During the improving of information security, to find out the weakness of network security instantly is very important. Also, to efficiently and correctly estimate and examine one's own security condition has become the first important theme for current enterprise and MIS personnel.

Our research, adopt the neural network type of SOM (Self-Organizing Map) structure, and try to propose a self-learning WIDS (Web Intrusion Detection System) which have the learning ability to detect the invade of network system. Our expectation is to solve the continuously changed invading attack problems. Through our WIDS (Web Intrusion Detection System), provided our system self-study ability so learn to protect system itself, also minimize hackers' attack. After testing and verifying, our research of WIDS (Web Intrusion Detection System) can be successfully detected up to 86% correctness.

Keywords: Intrusion detection system、Hacker、Neural network、Self-Organizing Map(SOM).

壹、研究背景與動機

根據 [CERT/CC, 2001] 的統計資訊圖 1，我們可發現，隨者網際網路的盛行與普及化，由各地所傳出的入侵攻擊事件從 1990 年逐年遞增，尤其到了 1999 與 2000 年攻擊事件更是趨向於直線遞增，這種現象實在令人擔心。今年最讓網管人員印象深刻的莫過於 7 、 8 月間爆發的紅色警戒病毒與隨之而來的娜坦病毒幾乎讓網路世界與各大企業人心惶惶，根據美國加州一獨立研究機構 Computer Economics 估計，在 7 、 8 月間肆虐全球電腦的代號紅色 (CodeRed) 病毒，已造成高達 26 億美元的損失 [Computer, 2001] 。

一般企業或資訊人員在進行網路與系統安全弱點評估時，通常會選擇購買商用的網路套裝軟體，較知名的如 CyberCOP 、 eEye Retina Scanner 、 ISS Internet Scanner 、 Kane Security Analyst 等工具。某些較有經驗的網管技術人員，也會採用開放原始碼程式 (Open Source) 如： Nessus 、 Nmap 、 Saint 等工具來進行弱點評估與掃瞄。但就使用的角度而言，不論是套裝或免費，這些工具多半無法滿足使用者眞正的需求 [林宗瀛，2001] 。因爲技術人員的技術與知識越來越難跟上不斷出現的

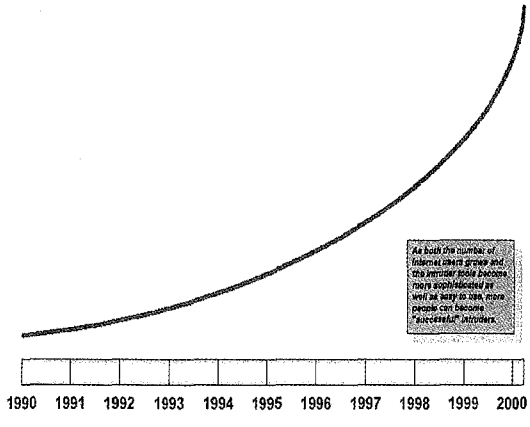


圖1：攻擊事件數據記錄
(資料來源：CERT/CC, 2001)

安全漏洞與攻擊手法。在日新月異的資訊安全問題中，如何及時找出網路安全的弱點，適時地、有效率的定期評估稽核自我網路安全狀況，成了當前企業與 MIS 資訊人員首要關切的議題。

本研究試圖提出一套具有學習能力的 WIDS(Web Intrusion Detection System) 入侵偵測系統。根據本研究所提架構期望能做到解決日新月異不斷翻新的攻擊手法，使得系統得以自我學習保護，而使駭客攻擊傷害能降到最小。

貳、網站入侵與攻擊行爲探討

網際網路的核心是 TCP/IP 所構成，亦是網路快速成長的關鍵因素之一。許多 RFC 文件都指出安全因素並未納入考慮，也就是說 TCP/IP 協定本身存在著許多漏洞。目前已知的攻擊方式實在難以估計。也因爲如此本章節重點主要是整理網站入侵的常見手法，包括今年 2002/4 月底發生的資料隱碼 (SQL Injection) 入侵事件。而 WEB 網站可說是一個服務提供者的門面，故常常是造成駭客攻擊的主要對象，但是如果這些網站本身有重要的資料，當被攻擊成功時，損失的可是公司的商譽以及客戶的隱私，實在不可不愼。底下內容將對常見之 WEB 網站入侵手法作各別詳細探討。

一、Web探索Probing〔McClure, 2001〕

本小節所要探討的內容是針對攻擊目標作蒐集資訊。網路環境中的駭客一定會事先做好探勘的動作。以 Web 探索 Probing 來說，假設您的攻擊目標有設立網站，可先從精讀其網頁著手，很多組織的網頁或多或少都會提供一些有用的駭客資訊，有些甚至相當誇張，例如有些組織將其防火牆的安全設定選項，直接列在網頁

上，且未設防任何網頁密碼保護。

此外 Web 探索 Probing 的方法，尚可試著觀看其網頁原始碼，將可能發現一些網頁程式設計師所留下的蛛絲馬跡，這些都將可能被埋藏在 HTML: 原始碼的註解中，如 "<" 、 "!" 、 "_" ，等註解標籤。且離線瀏覽網頁會比線上瀏覽來得有效率，因此我們可先把對方網站內容映射(mirror) 一份回自己的電腦，工具有 UNIX 上的 Wget 與 Windows 上的 Teleport Pro ，都是複製網站內容相當好用的工具。

另外我們也可利用一些主流搜索引擎所提供的進階搜尋功能例如 AltaVista ，可以讓您搜尋於 Host 目標網址下的所有目錄結構中的特定字串網頁。此時若有暴力法的攻擊工具夾著字典檔下去猜測，將會有相當高成功入侵機會。另外除了由網站下手探索外，網路上也有一些現成的工具可針對整個 WWW Server 系統作整體性的掃瞄，包括 Nmap 、 SATAN 、 SAINT 、 Nessus 等皆是這一類 Probing 探索的有名工具。

二、資料隱碼(SQL Injection)〔鮑友仲, 2002〕

利用資料庫查詢程式撰寫的漏洞，可以把「資料庫程式碼」當作一般資料丟給伺服器處理，使得伺服器錯把這些駭客丟來的「資料庫程式碼」僞裝資料，當成正常的資料庫程式來執行，藉而達到入侵的目的。例如，【 ExecSQL(" select * from table where id='$a' ") 】有一資料庫程式要做驗證密碼的工作，程式中他先將 User id 資料調出來，然後再準備下一步驗證動作，但是因爲沒有作好「資料輸入」的查核，使得駭客可以夾帶 SQL 語法到資料內闖關。

實際攻擊手法爲 Hacker 故意讓 $a 變數變成 " or [sql 語法] " 此時 Hacker 將可任意存取得 DB 中資料甚至修改、刪除皆可。承上例 Hacker 可使用 get 方式竄改 $a = " 'xxx' or where id != 'kkk' " ，此時程式所接收到的 SQL 執行碼將變成 ExecSQL(" select * from table where id = 'xxx' or where id != 'kkk' ") ，此時 Hacker 將可於 Web 瀏覽器中很輕易的看到特定資料庫中所有資料。

三、利用已知的程式漏洞：

由於現今資訊傳播相當迅速，任何系統程式有了安全上的漏洞很快就會透過各種管道傳遞，尤其是程式漏洞，大部分程式漏洞皆發生在程式設計師未全盤考量的設計與程式中不當的使用記憶體之情況而發生的 [TWCERT, 2000] 。例如：緩衝區溢滿 (Buffer Overflow) 攻擊。當然此部分程式漏洞也不是程式設計師所能控制的，例如 PHP3 此架構在 UNIX 系統中相當熱門的解譯程式，此是 Open source 之軟體，於 PHP3 系列版本就曾發生嚴重的程式漏洞，使 Hacker 能夠藉由 PHP3 透過瀏覽器以 root 權限任意存取瀏覽系統中任意目錄。因此各個系統管理者應該定期瀏覽相關的資訊例如目前網路中有許多站台是專門介紹攻擊防禦資訊的例如：CERT 、 Rootshell 、和 NetworkeICE... 等等。或是訂閱相關的 mailing list ，例如 TW-CERT 的 Advisory ，如果本身系統發現相關的問題，應該儘速修正。

四、安裝系統時，內定安裝或啓動的程式及服務

根據美國 FBI 聯邦調查局與 SANS 機構聯合發佈的一份 20 大軟體漏洞排行榜 [SANS, 2002][CNET, 2001] ，可發現微軟在 2001 年度漏洞排行榜上高居榜首，且除了一些特殊漏洞外（如 Code Red 所利用的 IIS 系統漏洞），此份名單也一些網管人員平時便可進行的網路保全措施。

例如，名單中指出多數採預設方式安裝的軟體都缺乏安全性；許多企業也沒有定期進行備份；沒有使用密碼，或者密碼組合太容易遭破解。我們可發現不管是 Windows NT 或者是 UNIX 皆有一些內定選項，當管理者未適當加以修改或者修正時，就會造成 Hacker 有可乘之機，例如 IIS 中預設安裝皆會發現 /scripts 目錄下有相當多的漏洞，且微軟也在後來坦承存在漏洞，故我們可發現於 IIS5 之後的版本皆不存在此 /scripts 目錄了。

五、U2R(User to Root)與R2L (Remote to Local)

指的是於 Telnet 中最常發生的兩種攻擊行爲集合，U2R 表示當入侵者藉由 Telnet 進入系統後，攻擊者已經取得一般 User 帳號，此時攻擊者將會採用各種攻擊行爲模式，設法要取得 Root 權限，此類衆多的攻擊方式通通被稱作是 U2R (User to Root)。而 R2L(Remote to Local) 指的是後門程式 (back door program) 之攻擊行爲集合稱之。

六、TCP SYN Flooding：〔CERT /CC, 2001〕

TCP SYN Flooding 是利用 TCP/IP 協定的弱點，正常的 TCP/IP 連結必須經過三筆訊息的交握 (handshaking) 過程方能完成。一般在連線過程當中，Host 傳送一個同步封包 (SYN) 給 Server，Server 回傳一個承認訊號 (ACK) 後，Host 再回傳自己的 ACK 並建立連線。如果 Host 主機不進行第三個步驟，Server 會預留暫存空間已等待接收資料，如果在同一時間發生太多建立連線的要求，且暫存空間也一直被建立而不釋放出來處理資料，則系統會發生當機或癱瘓的情形。

請參閱圖 2，這種手法就是所謂的 SYN Flooding，而且駭客通常會以假冒 IP 的方式來攻擊，讓系統無法追溯攻擊者。這時網站遭受到 TCP SYN Flood 攻擊時會有無法接收 WWW 或 FTP 服務等請求的現象。

七、Smurf 攻擊：〔CERT/CC, 2001〕

Ping 會送出單一的資料封包，然後等待回應。如果對於同一部主機連續給予大量的 Ping 封包，即 Ping Flood，則會造成網路及主機的大量資源耗盡。一種類似於 Ping Storm，但威力更強大的阻斷服務攻擊稱爲 Smurf(依攻擊程式之名稱命名) 請參閱圖 3 Smurf Attack。

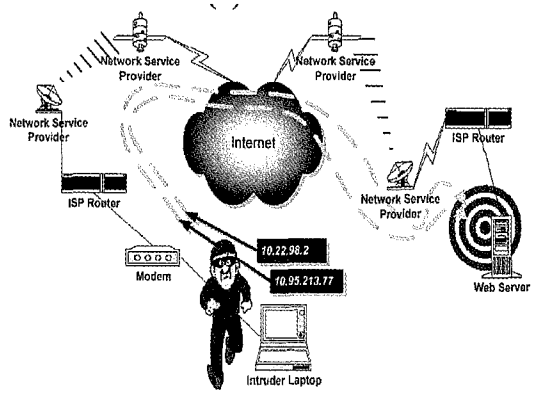


圖2：TCP SYN Flooding
[資料來源：CERT/CC，2001]

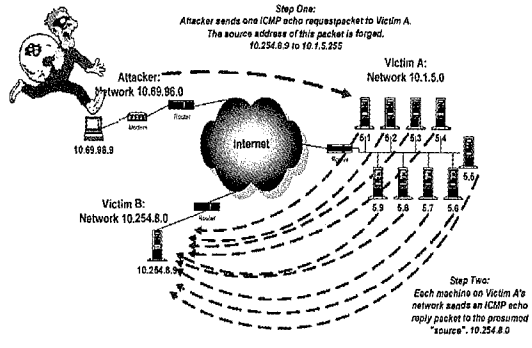


圖3：smurf attack
[資料來源：CERT/CC, 2001]

Smurf 利用 ICMP(Internet control Message Protocol) 的 echo 回應，因爲 Smurf 的 echo 要求是向整個網路上全部的主機廣播，所以會衍生大量的反應封包，所有的反應封包送給單一主機，即被攻擊的目標，造成服務被阻斷。如圖 3 駭客送出 ICMP echo 的 request 封包給 Victim A，而這個封包的來源 IP 被修改成 10.254.8.9 到目的地的 IP:10.1.5.255，這時當 Network 10.1.5.0 的機器收到後，便對 10.254.8.9 的這台機器發動 Ping flood，造成該機器和網路的資源耗盡。所以駭客可以利用不知情的網路做爲跳板，有效的放大攻擊能量，因此 smarf 也是相當有威力的阻斷服務攻擊。

八、分散式阻斷服務攻擊DDOS(Distributed denial of Service Attack)(CERT/CC, 2001)

DDOS 就是將一些 DOS 的攻擊程式分散於多台電腦，一起攻擊目標網站（如圖 4），每一台被控制的電腦，都發動 Ping Flood 或 SYC Flood 等等的 DDOS 攻擊，這樣聯合起來由攻擊者下達單一的、簡潔的命令，指揮數百部或數千部的電腦同時向目標主機投擲資料封包造成目標主機整個網路和主機的癱瘓，當時攻擊 Yahoo ！網站的能量達到每秒 1Gagabits 以上，可以想像 DDOS 威力的強大。這些發動 DOS 攻擊的機器，大多是利用一些駭客程式，如 TFN 、 TFN2K 等，但須先入侵數十台電腦，並直入後門程式及 DDOS 程式，通常，這些電腦的防護能力都相當的弱，甚至在這些電腦的網路上根本沒有部署防火牆或入侵偵測系統，因此，很容易成爲駭客下手的目標。

圖 4 DDOS Attack 說明了 DDOS 的攻擊模式，駭客入侵了網際網路上一堆電腦後，選擇其中一台或多台電腦作爲 Master Server ，而其他台電腦被當作

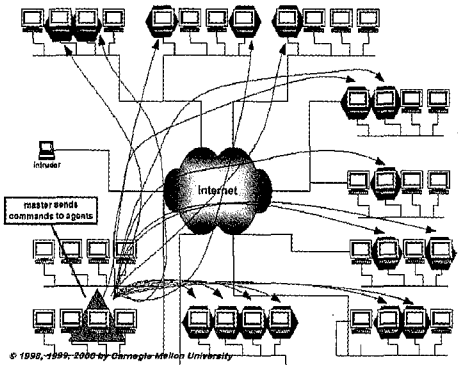


圖4 DDOS Attack

（執行者），也就是攻擊者在（執行者）的電腦上會放入 agents 也稱爲 daemons 的 DOS 程式。此時只要駭客下達命令給 Master 發動攻擊，Master 在下達給這些 daemons ，同時向目標投擲大量封包，形成暴風雨式的 DDOS 攻擊。像是 TFN （ Tribe Flood Network ）、 TFN2K 都是這樣的架構。 Trinoo 則多了一層架構於 Master Server 和 daemons 的中間，其攻擊的封包是爲 UDP 封包。其他的 DDOS 工具則使用不同型態的封包。

TFN 及 stacheldraht 的（執行者）可執行強大的 smurf 攻擊，或選用 UDP 、 TCP 、 SYN 、 ICMP echo 等三種型態之一的封包， TFN2K 則有混和不同型態封包的能力，包含了 UDP Flood 、 ICMP echo flood 、 ICMP broadcast flood (Smurf) ， TCP SYN attack 。而這些 DDOS 的程式，也有愈來愈完整的機制，像是 Trinoo 甚至於使用了 password 作爲溝通， TFN2K 則更進一步對控制通訊於以加密，未來的 DDOS 程式的發展將愈來愈難捉摸。

九、特洛伊木馬(Trojan Horse attacks)：(McClure et al., 2001)

特洛伊木馬大致可分成兩種類型，一

是藉著藏在看是無害甚至很有趣的程式之中，而到處散佈的控程式，當收到這個包裝程式的使用者被騙執行程式之後，這個特洛伊木馬便會開始暗中進行一些非法或是未經授權（含有惡意）的動作，例如傳回一些使用者資訊到某個預設主機上、顯示系統訊息、刪除檔案或將磁碟格式化。另一類型則是由攻擊者手動安裝，此種類型需要攻擊者先入侵系統且取得適當權限後方可能成功，包括有些木馬程式會在您的系統留下後門，作爲下次入侵的獨立專屬窗口，潛在的特洛依木馬技術的數量只會受限於攻擊者的想像力與創造力。

十、主機假冒(spoofing)

主機假冒攻擊是相當常見的網路攻擊方式，攻擊者假冒合法的主機，以欺騙要攻擊的主機，並藉此得到該主機的信任，取得想要的資源或權限。目前常見的主機假冒攻擊方式有下列幾種：

1. 硬體位址假冒 (ARP spoofing)
2. IP 位址假冒 (IP address spoofing)
3. TCP 連線假冒 (Connection spoofing)
4. 繞路路徑假冒 (Routing information spoofing)
5. 主機名稱假冒 (Domain Name spoofing)
6. 服務假冒 (Service spoofing)

而其他市面上網路知名攻擊軟體與攻擊手法本研究搜集整理如表 1 所示：

表1：網路上知名攻擊軟體與攻擊手法介紹

Attack Tools	Classification	說　　明
Nmap	Probing (Normal)	Nmap 可以判斷遠端作業系統核心 (Kernel) 與版本、開啓哪些傳輸埠、甚至可針對整個網域進行掃瞄，且支援正常模式 (Normal) 與偷偷掃瞄模式 (stealth)。
	Probing (Stealth)	
SATAN	Probing (Vulnerability)	撒旦 (SATAN) 於 1995 現身，當時受到相當矚目，它的功能涵蓋網路上主機的全面掃瞄，找出有哪些主機存在，並可以發現這些主機提供什麼服務，接著找出這些服務所存在的漏洞，最後提出建議書以供管理者改善參考。
SAINT (Web 版)	Probing (Vulnerability)	SAINT是由 SATAN 所改進而來的，透過瀏覽器的介面使用，它的目標在設計出一套簡單好用的系統安全偵測工具，讓想要使用的人能夠自由的使用它。
Nessus	Probing (Vulnerability)	1998 年由 Renaud Deraison 設計提出，Derasion 建立了一個 CVS 伺服器，可以每天、甚至每小時進行更新，更新速度相當快，2002 年後 Nessus 採用 Plug-in 的支援方式來提供各式弱點分析，目的是全方位的系統弱點偵測與分析。
Storm-DDOS	DOS (1min)	此軟體具殺傷力，特色在於可自訂封包大小，DDOS 並無限制，Server 端 Action 後將會自動尋找 Client 端同時發動攻擊。
	DDOS (2pc.1min)	
Raw-clones	DOS (1min)	可以遠控 8 台僵屍電腦進行攻擊，支援跳躍攻擊及隨機選擇攻擊等。
	DDOS (2pc.1min)	

Attack Tools	Classification	說 明
DDOS Ping V2.00	DOS (1min)	可以對一連串的網絡位置主機和服務器進行查詢指令攻擊，同時支援 4 種不同的連接速度，可以自行設定每秒傳送封包的多少。
	DDOS (2pc.1min)	
Udp Flooder	DOS (1min)	同 DDOS Ping 一樣介面，可以設定封包的內容和攻擊同標設定 Windows 或 Linux，可以選定 3 種攻擊的手法，包括 Trinno，Stacheldraht，Tribe Flood Network，又支援 UDP 傳送，可以測試您的服務器安全性。
SYN Flooder	DOS (1min)	SYN Flooding是利用 TCP/IP 協定的弱點以假冒 IP 的方式來攻擊，當同一時間發生太多建立連線的要求，且暫存空間也一直被建立而不釋放出來處理資料，則系統會發生當機或癱瘓的情形
Smurf2k	DOS (1min)	此是SYN flooding的變形攻擊。Smurf利用ICMP(Internet control Message Protocol)的echo回應，因爲Smurf的 echo 要求是向整個網路上全部的主機廣播，所以會衍生大量的反應封包，所有的反應封包送給單一主機，即被攻擊的目標，造成服務被阻斷。
CodeRed	系統漏洞(CodeRed V1)	這個病蟲利用的技術就是 IIS「緩衝區溢位」的漏洞。
Nimda Worm	系統漏洞	這隻 worm 除了修改web內容以促進它的繁殖外並無毀滅性的動作，但卻有因網路掃描和電子郵件增殖所而造成拒絕服務 DOS (denial of service) 的報告。
PHF attack	程式漏洞	此是早期 UNIX 系統內定安裝 CGI-bin 程式之漏洞。
HTTP Bomber	程式漏洞	可以選擇使用代理伺服器進行攻擊，攻擊手法分爲 Normal (支援亂線攻擊)和 POST 請求攻擊，會自動統計攻擊次數和輸出體積。
snot-0.91a.tar.gz	程式漏洞	此工具可牽制 snort rule 的檢測，使攻擊封包闖關成功。
ShutDownSQL	程式漏洞	利用程式漏洞使服務端口 (port)，無法提供服務而休克。不需要戶名和密碼就可以停掉SQL Server服務。
Access Diver V4.76	程式漏洞	多功能的破解程式，包括 GET、HTML、PROXY、SOCK 等破解及連接功能，支援字典檔破解功能。
Hackware V1.0 Beta Build 43	程式漏洞	多功能的破解程式，包括支援 Frontage Extension、CGI、Proxy 等破解程序，可自行生成字典檔內容，轉寄和匿名電郵等功能。
資料隱碼	程式漏洞	於http竄改 SQL 參數試圖闖關獲取、刪除or更改資料。
竄改參數	程式漏洞	於http竄改參數試圖闖關攻擊。

Attack Tools	Classification	說　　明
php-exploit	程式漏洞	於 http 使用參數 +256bit 攻擊 ex: http://phpweb.yuntech.edu.tw/php/reg.php?aaaaaaaaaa~128~aaaaaa ，此行為將被 SOM 攔下判斷合法否。a
Buffer Overflow	程式漏洞	

（資料來源：本研究整理）

參、入侵偵測系統定義與分析

自從 Anderson 於 1980 年提出入侵偵測系統 (intrusion detection) 的技術報告後，入侵偵測系統的研究至今已超過二十年。根據 Graham[2000] 的定義，「入侵行為 (intrusion)」意指：「某人（通常被稱為 "hacker" 或 "cracker"）企圖潛入或濫用 (misuse) 系統」，而「入侵偵測系統」意指：「負責偵測這些入侵行為的系統」。[Theuns et al., 2002] 則定義入侵偵測是使用自動及智慧型的工具，及時偵測入侵者意圖。這樣的工具我們稱之為入侵偵測系統 IDS(Intrusion Detection System)。一般而言，入侵偵測系統就是一種網路安全監測工具，藉由解讀系統稽核檔或網路封包內容，即時偵測出對系統所進行的攻擊行為，並回報給系統管理者，以加強維護系統之安全。

根據 [Michael, 2001] 所收集整理的 IDS 入侵偵測系統共有 92 種。雖然各式各樣的 IDS 入侵偵測系統不斷的因應市場需求而產生，但針對其運作模式可分成 host-based 與 network-based [Biswanath et al., 1994][Bace and Mell][鄭有倫 ,2000]，分別說明如下：

一、網路型(network-based)入侵偵測系統

主要是涉及擷取網路中所有封包，然後根據所擷取的封包作分析與偵測的動作，此類的 IDS 入侵偵測系統有：Network Security Monitor(NSM)、Distributed Intrusion Detection System (DIDS)、Network Flight Recorder(NFR)…等等 [Biswanath et al., 1994]。

主要可以擁有下列優點：

1. 成本較低
2. 可偵測到主機式入侵偵測系統偵測不到的攻擊行為
3. 駭客消除入侵證據較困難
4. 與作業系統無關

由於僅需在區域網路中增設一台監視主機，因此無須更動原網路架構中的網路作業系統。但相對的，網路型入侵偵測系統也有其一定的缺點如下：

1. 若網路的型態過大，網路型入侵偵測系統往往會漏失掉許多封包，無法完全監控網路上所有流通的封包數。
2. 若要擷取大型網路上的流量並分析，往往需要更有效率的 CPU 處理速度，以及更大的記憶體空間。
3. 如果封包已經加密，則網路型入侵偵測系統就無法調查其中的內容，可能會錯失包含在封包中的某些攻擊資訊。
4. 對於攻擊者直接坐在受害主機前的攻擊行為是無能為力的。

二、主機式(host-based)入侵偵測系統

主機式入侵偵測系統發展始於 80 年代早期，通常只觀察、稽核系統日誌檔是

否有惡意的行爲，用以防止類似事件再發生。主要是以本機電腦系統上的稽核資料(audit data)爲基礎所進行的入侵偵測工作，這類的IDS入侵偵測系統有：Computer Watch、Discovery、Haystack、Intrusion-Detection Expert System(IDES)、Multics Intrusion Detection and Alerting System(MIDAS)…等等[Biswanath et al., 1994]。在Windows NT/2000的環境下，通常可以藉由監測系統，事件及安全日誌檢視器中所記載的內容來加以過濾、比對，從中發現出可疑的攻擊行爲；在UNIX環境下，則監測系統日誌。當有事件發生時，主機式入侵偵測系統即做入侵行爲的比對，若有符合，則由回應模組通知系統管理員，以對攻擊行爲進行適當的反應。

一般而言，主機式入侵偵測系統有下列的優點：

1.確定駭客是否成功入侵

2.監測特定主機系統的活動

3.較適合有加密及網路交換器(switch)的環境

4.不需另外增加硬體設備

然而主機式入侵偵測系統也具有某些功能上的限制：

1.各種作業系統有各種不同的稽核紀錄檔，因此必須針對各不同主機不同版本或完全不同的作業平台安裝各種主機式入侵偵測系統。

2.入侵者可能經由其他系統漏洞入侵系統並得到系統管理者的權限，導致主機式入侵偵測系統失去其效用。

3.主機式入侵偵測系統可能會因爲DoS而失去作用。

4.主機式入侵偵測系統並不能用來做網路的監測與掃描主機所在的整個網域，因爲僅能看到經由該主機所接收到的網路封包資訊。

5.由於當主機式入侵偵測系統處於監測狀態時亦消耗該主機的系統資源，亦會影響被監測主機本身的效能。

如果再進一步針對其偵測方式加以分類又可分成misuse model和anomaly model [Bauer et al., 2001]，本研究將之綜合整理成圖5。Misuse model係指系統資料庫中已經事先預存著各式各樣的攻擊行爲樣本(attack pattern或attack signature)，然後藉由收集分析所有使用者行爲與攻擊樣本比對，如果發生符合或相似，即有可能是攻擊行爲。至於anomaly model則指的是系統資料庫中已經事先預存著正常行爲(normal behavior)的特徵樣本，往後藉由收集分析所有使用者行爲與正常行爲樣本比對，如果發生行爲偏離很大，即很有可能發生入侵攻擊事件。

Anomaly detection或misuse detection IDS其效能優劣主要由兩因素決定：[李駿偉等，2002]

(1)資料特徵（feature）-衆多收集資料中並非每一筆資料對IDS偵測過程有幫助，因此必須找出可用的特徵來進行過濾與分類。此外feature也可用來表示每一筆資料的行爲描述，代表每一user在連線或使用期間的行爲描述。因此feature的適當選定有助於IDS對使用者的行爲分析。

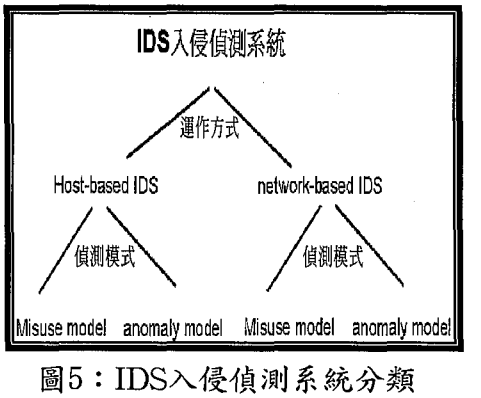


圖5：IDS入侵偵測系統分類
（資料來源：本研究整理）

(2)分析方法與工具-其作用如下：

1. 用來建立攻擊規則(Attack Rules)或行爲模式 (Behavior Model)。確定特徵後就是據此進行過濾與分析，必須依賴各種分析工具，建立對應的攻擊規則與行爲模式。
2. 用來進行比對工作。當收集到現行行爲資料時，需要如同建 model 的過程，根據特徵將行爲轉換，專爲與先前建立行爲模式相符的比對格式。

特徵和分析工具的選取是影響 IDS 效能的主要因素，歸納衆多文獻，IDS 使用的分析工具可概略分成六類：

(1)統計分析工具-利用統計方法建立規則或行爲模式（rule, behavior model）。是用統計學上的 classification 方法來進行資料分類，如 K-MEANS、ADC 等方式。常應用於建構異常偵測器(anomaly detector)。如 ARGUS, SPADE, W&S, NIDAS 等

(2) Neural network 分析工具-具學習能力的演算法，經由適當訓練可使其具有辨識 intrusion 的能力。使用 neural network 方法來進行資料分類，代表方法有 self-organization、backpropagation network 等。亦常用於建構 anomaly detector。如 HyperView, ACME 等。

(3) Rule-based 分析工具-利用各種語法敘述目的事件，以建立事件規則庫的分析工具。將資料建成 rule，然後將建立的 rule 與現行 behavior 進行比對。代表方法有 P-Best rule language。常用來建立 misuse 專家系統。如 IDES, NIDS, ASAX 等。

(4) Bayesian network 分析工具-利用貝氏條件機率所延伸出來的一種學習演算法。使用 Bayesian network 方式表示入侵事件關係，以進行分析與攻擊問題偵察。常用於 misuse 專家系統。如 ICE。

(5) Finite-state-machine 分析工具-利用狀態轉移的觀點來描述目的事件的分析工具。將系統的狀態或行爲利用有限狀態的方式表述。常用來建立 misuse 專家系統。如 STAT, USTAT 等。

(6) Data Mining 分析方法- 利用 Data mining 與 Meta-learning 有效的將資料分類並且減少不必要的資料比對，且能產生新的 Rule 以偵測未知入侵。適用於 misuse 與 anomaly 專家系統如 FIRE。

Misuse detection 所依賴的是利用已知的事件建立各種攻擊模式，再比對判定行爲是否具侵略性。這樣的方法不易誤判（將正常視爲異常），但偵測率不高。若現行攻擊行爲不存在於攻擊模式資料中，將無法偵測此行爲。通常使用 rule-based、bayesian network 及 finite-state-machine 三種分析工具。另一個方式是 anomaly detection。對比於 misuse detection 方法，後者是建立負面行爲模式 (Negative Behavior Model)，前者則是建立正面行爲模式 (Positive Behavior Model)，亦即在系統建立正常行爲模式資料庫，將所偵測到的現行行爲與現存模式進行比較，若差異甚大則視爲異常行爲。此方法偵測率高，但同時誤判率也很高，易把正常行爲誤判爲異常的攻擊。通常使用 statistic analysis 及 neural network 等方法工具，主因是在使用上限制較多，且使用的特徵必須可量化才可使用。

由於 misuse 與 anomaly detection 方式各有優缺點，因此本文之研究動機爲，擬建立一高效能之智慧型入侵偵測系統，將 misuse 和 anomaly detection 兩種方式混合使用，即混合式偵測 (Hybrid detection)，以對二者缺陷產生互補作用，達成高偵測率同時也能保持低誤判率。

肆、國內外知名入侵偵測系統探討

李勁頤，陳奕明 [2002] 曾針對國外著名入侵偵測系統做比較分析，本節則再針對國內外著名的入侵偵測系統做一整理與介紹：

一、RD-NIDS〔鄭有倫,2000〕

RD-NIDS 便是一個以分散式架構爲基礎所設計的分散式網路入侵偵測系統。

RD-NIDS 採用分散式架構，在每一區段內安裝一套軟體，本地入侵偵測引擎，並於網管中心架設一套中央入侵偵測管理員 (CIDM) 軟體。藉由各區段 LIDE 與 CIDM 間相互的分工及配合，以達到偵測入侵的目的。LIDE 首先對區段網路上的封包加以監督並擷取、過濾、分析與比對，判斷該區段網路是否遭受攻擊，或發掘可疑的封包。若判斷遭受攻擊或發現可疑，則將相關資料往 CIDM 傳送，由 CIDM 做進一步的分析與判斷。其分工方式爲 LIDE 做初步的過濾，再交由 CIDM（有較強的計算能力）做更精確的判斷。由於將入侵偵測機制分散於各網路區段中，因此對於新增或移除受保護網路區段，或是一旦網路架構有重大變動，RD-NIDS 皆可隨之作適當的調整，具有較大的彈性。

二、NeuroIDS〔丘偉權, 2001〕類神經網路式之入侵偵測系統

NeuroIDS 是一個應用類神經網路發展而成的網路型入侵偵測系統。NeuroIDS 主要有兩種操作模式，一個是訓練和測試模式；另一個是應用模式。訓練和測試模式主要是讓使用者自己訓練並產生自己想要之入侵偵測引擎。應用模式則是讓使用者眞正執行線上即時入侵偵測。NeuroIDS 提供三種偵測模式，即特定服務模式、攻擊類型模式及一般 TCP 模式。特定服務模式主要用於偵測針對某特定服務之攻擊；攻擊類型模式可用以偵測同一群組中之相類似攻擊；一般 TCP 模式可以用以偵測廣泛之 TCP 攻擊。目前在實作上多採用 TCP 模式，但特定服務模式可用以偵測 FTP 之攻擊；攻擊類型模式則用以偵測 Probe 攻擊爲主。

三、EMERALD〔Porras and Neumann 1997〕

EMERALD 由 SRI 於 1997 年所提出，其目的在於建立一套具 scalability 的分散式入侵偵測架構，以整合各類資訊安全系統。EMERALD 主要由許多 EMERALD Monitor 組成。如圖 6 所示，每個 EMERALD Monitor 都已完整包含分散式入侵偵測系統的主要元件，而這些 EMERALD Monitor 主要透過階級分層法加以組織，包括了服務層、領域層級與企業層級。位於服務層級的 EMERALD Monitor 將進行「服務層級分析」以找出發生於特定服務的入侵行爲；而更高層級的 EMERALD Monitor 則分別執行「領域層級分析」與「企業層級分析」的關聯分析工作，以找共同合作找出分散式的攻擊行爲。

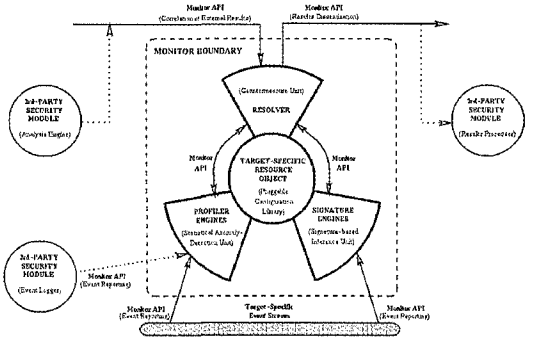


圖6：EMERALD Monitor架構

在協調機制方面，每個 EMERALD Monitor 都擁有功能完整的協調器，透過發行 (publish)/ 訂閱 (subscribe) 的協商協定來共同合作。由於 EMERALD 的主要目標在建立分散式的入侵偵測環境，甚至希望能整合其他入侵偵測系統，所以其強調的重點是該架構對於異質性統的整合能力，而非建立一套關係緊密、分工細緻的系統，因此管理元件的角色在 EMERALD 並不明顯，關於 EMERALD 如何描述完整的分散式入侵偵測任務，如何將這個任務拆解、分派至各 EMERALD Monitor 執行的資料也相對缺乏。

四、NSTAT〔Kemmerer, 1997〕

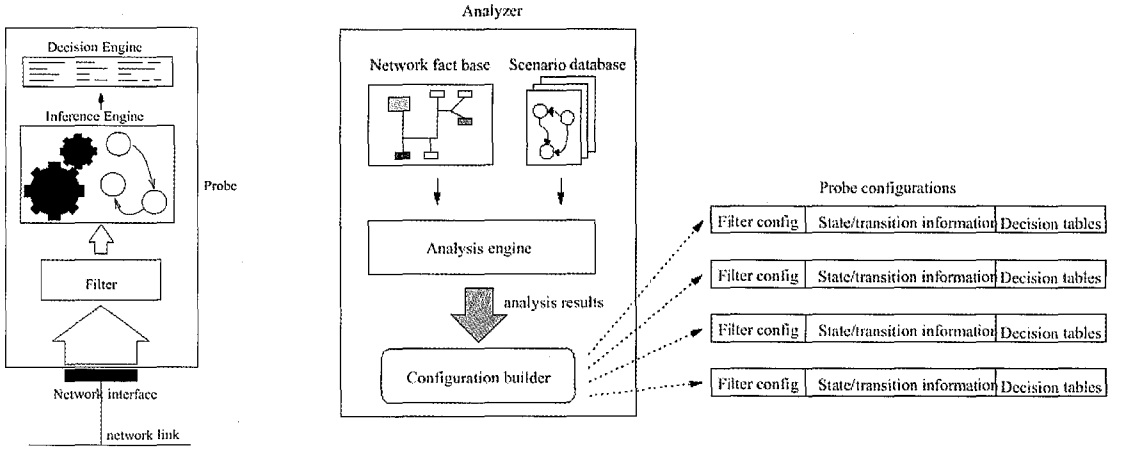
NSTAT 由美國加州大學 Santa Barbara 分校 (UCSB) 於 1997 年提出，爲其早期系統 USTAT 的分散式版本。USTAT 爲一主機式入侵偵測系統，利用狀態轉換圖來描述入侵行爲，並使用 Rule-based 的專家系統 STAT 來進行入侵偵測。UCSB 將原有單點式入侵偵測系統 USTAT，擴充成分散式入侵偵測系統 NSTAT 的主要目的是爲了要偵測網路檔案系統上的合作式入侵行爲。NSTAT 的系統架構仍是最原始的集中式架構，各主機的系統稽核資訊都送到單一分析器進行分析。如同 DIDS，NSTAT 分析型態爲集中分析，且該系統沒有（也無需）協調器，資料型態集中處理無工作分派問題，所以無所謂協調型態、管理型態。

五、NetSTAT〔Vigna and Kemmerer, 1998〕

NetSTAT 由 UCSB 於 1998 年提出，爲一網路式分散式入侵偵測系統，其設計目的在於偵測網域中的入侵行爲。NetSTAT 利用狀態轉換圖來描述網路入侵行爲與分散式攻擊，並使用 Rule-based 的專家系統 STAT 來進行入侵偵測。

NetSTAT 的系統架構，基本上是由許多由許多分布於各地的 probe 與 Analyzer 共同組成。probe 爲一自治型入侵偵測模組，不但可以獨立偵測一般型式的入侵型行爲，也可以與其他 probe 共同合作，偵測分散式的攻擊行爲。

圖 7(a) 爲 probe 的架構圖，probe 主要包括 Filter、Inference Engine 與 Decision Engine 等模組，Filter 的主要工作爲收集網路通訊資料，經過濾後交給 Inference Engine 處理。Inference Engine 的主要任務爲分析網路通訊資料，執行入侵偵



(a) (b)

圖7：Probe與Analyzer架構圖

測的工作。當 Inference Engine 完成分析後，會將結果再將交給 Decision Engine，讓它判斷應要採用什麼回應機制，若發現偵測結果爲分散式入侵的一部份，Decision Engine 還會依 decision table 中的資訊，主動與其他 probe 聯繫，交換分析結果，以合作偵測分散式入侵行爲。

而 NetSTAT Analyzer 爲一管理元件，其可以幫助管理者分析與描述（分散式）入侵攻擊行爲，更重要的是 Analyzer 的 Configuration Buidler 模組，還可將完整的分散式攻擊描述加以分解，產生對於感應器、分析器、協調器與回應元件的設定資訊，充分提供任務描述與工作分派的管理功能。圖 7(b) 爲 Analyzer 的架構圖。據悉，NetSTAT 爲最早採用「動態協商、集中管理」的系統。

六、AFFID[Balasubramaniyan, et al.,1998]

AFFID 由 Purdue 大學於 1998 年提出，其主要的目的在建立一套具有 scalability 的分散式入侵偵測系統。該系統採用階層式的建構，並使用了自治型代理人程式的技術。圖 8 爲 AFFID 的系統架構圖，AFFID 主要包含四類元件，Agents、Filters、Transceiver 與 Monitors。Agents 爲安裝在遠端電腦上，特別設計來執行特定用途的程式，例如其可能是監視受保護主機上的 telnet 連線數目，也可能是如 IDIOT 等大型分散式入侵偵測系統，不過無論 Agents 的功能強弱，在 AFFID 中 Agents 並不能直接相互溝通，而必須將它們所產生的訊息送至 Transceiver。Filters 的主要功能爲替 Agents 提供資料選取與資料抽象化的服務，以方便 Agents 由資料來源處取得所需資訊。Transceivers 爲安裝在每台受保護電腦上，負責管理其上 Agents 與提供對外溝通能力的程式。Transceivers 的主

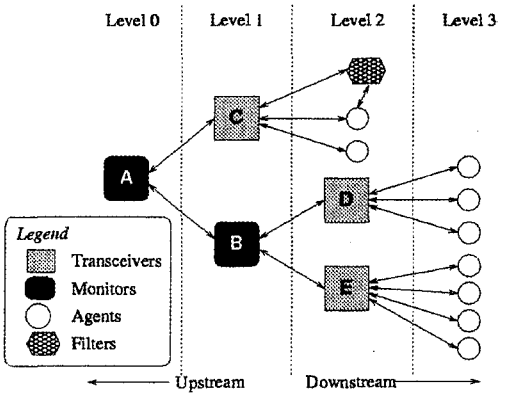


圖8：AFFID系統架構

要任務爲啓動、停止代理人程式與回應 Monitors 的命令，以及接收與處理代理人程式所產生之資訊，並將處理結果送交其 Agents 或 Monitors。Monitors 爲 AFFID 中最高層的實體 (entities)，主要負責控制多個主機上的 Transceiver（或 Monitors），並要處理來自多個 Transceivers（或 Monitors) 的資訊。

基本上 AFFID 警示訊息的傳遞主要仍是依循階層式架構進行，因此其協調型態應仍屬固定式協調。此外，AFFID 由於主要目的建立分散式入侵偵測系統架構，因此十分缺乏關於其如何進行分散偵測的任務描述與工作分派。

七、IDIAN(Feiertag, et al.,2000)

IDIAN 由 Feiertag 等學者於 2000 年提出，本研究的重點主要在探討異質入侵偵測系統間的溝通、協商機制，各入侵偵測系統間可透過協商機制，提供服務或請求其他入侵偵測系統的服務。此系統主要利用 CIDF 的 General Intrusion Detection Object (GIDO) 作爲各系統間資訊間換的標準，各入侵偵測系統並可利用該系統定義的 GIDO Filter 來描述自己的能力，與表達服務請求。

圖 9 爲 IDIAN 協商協定（消費者觀點）的概念示意圖，該協定的運作流程大

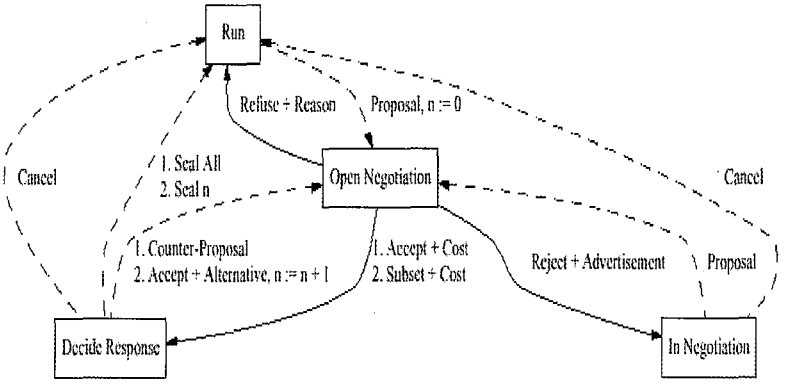


圖9：協商協定

致如下，由消費者向供應者提出其需求(proposal)，供應者判斷後若發現自己並未提供該服務，則拒絕該提，並告知消費者其能提供之服務。若供應者有提供該服務，則會在評估自己的能力（依目前系統資源與要求服務者的多寡）後，告知消費者自己所能提供的部分與該服務的成本，消費者在接供應者回應後，可決定是否全部接受或部份接受服務內容。如此反覆此程序，直到雙方達成協議。

透過協商協定，不同入侵偵測元件可以了解彼此的能力並取得相關的服務，而達成共同運作的目標，而入侵偵測元件的協調器也可依各 IDS 達成的契約，來決定應如何轉送警示，所以 IDIAN 很明顯的具備了分散分析與動態協商能力。

八、MADAM

MADAM 由 Lee 等學者於 2000 年提出，MADAM 是一套建構於 CIDF 之上的分散式入侵偵測系統。此研究有兩大目標，(1) 即時收集資訊，並利用資料挖礦技術加以分析後，線上產生新的偵測方法，分送至各入侵偵測系統；(2) 透過入侵偵測資訊的交換，使分散式入侵偵測系統能合作偵測分散式入侵。

圖 10 爲 MADAM 的架構圖，在 MADAM 目前的雛形系統中包括了兩台入侵偵測系統 Bro 與 NFR 、一台 Modeling Engine 與 Match Maker 。各 IDS 所產

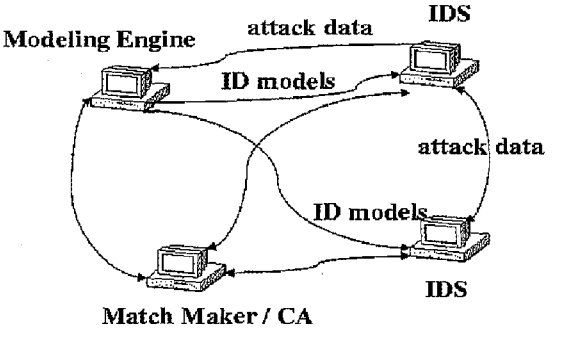


圖10：MADAM系統架構

生的攻擊資訊都會包裝成 GIDOs ，會送到 Model Engine 進行分析，以產生新的偵測模式。而 Match Maker 爲一 CIDF matchmaker ，其任務在於協助入侵偵測系統的各元件找到其所需服務的提供者。

在新偵測模型的方面，MADAM 中主要展示了以下的實驗。入侵偵測系統 Bro ，原本沒有偵測 SYN Flood 的方法，但其 Anomaly 入侵偵測模組卻透過網路流量的變化發現出異常現象，於是便將相關資訊 (tcpdump data) 送往 Model Engine 進行分析。在 Model Engine 找出了 SYN-Flood 的特徵後，就將新偵測模式以 GIDOs 包裝後送到 Bro 、NFR ，Bro 與 NFR 都能分別將新偵測模式轉會成該系統內部可使用的指令檔 (scripts) ，用以偵測 SYN-Flood 。而在合作偵測分散式入侵方面，MADAM 中主要展示了下述實

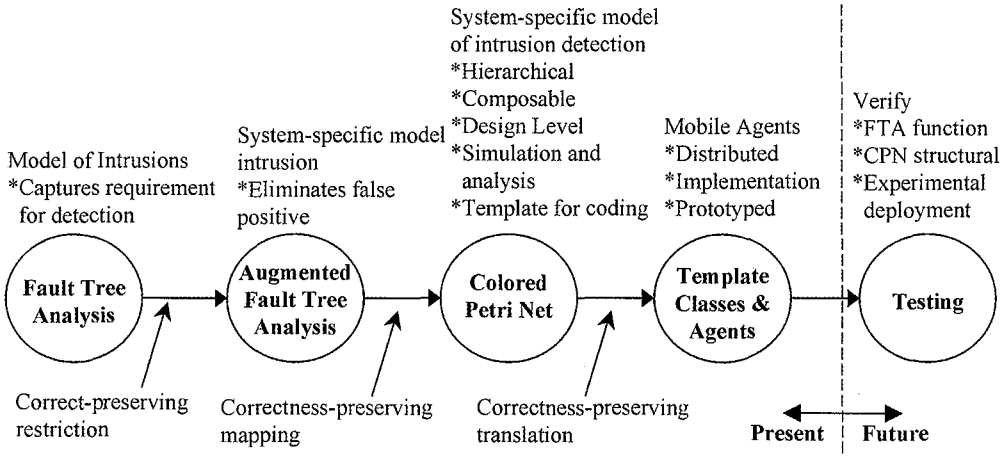


圖11：入侵偵測系統發展模式

驗：Bro 所偵測的子網路內，某台伺服器正遭受 TFN 分散式攻擊工具的攻擊，當 Bro 偵測到 DDoS 攻擊後，便會將相關攻擊報告包裝成 GIDOs 後送給負責監控另一各網域的 NFR 。NFR 在解讀 GIDOs 後，由攻擊的來源位址發現，許多攻擊的程式 (slaves) 都是在位於自己負責的網路內，因此開始執行相關回應措施，如 block 掉 TFN master 與 slaves 間的控制訊息，與通知各主機上的回應單元刪除 slaves 程式。

由於 MADAM 仍屬於雛形階段，只有兩台入侵偵測系統，因此目前並未實作完整的協調器，目前雖已提供強大的偵測模式更新的管理功能，但仍欠缺對於複雜分散式入侵偵測任務之描述與工作分派能力。不過 MADAM 卻提供了管理元件能力未來發展的方向 --- 自動、即時產生的分散式入侵偵測任務，並立即完成偵測任務之分派。

九、MAIDS(Helmer, et al.,2000)

MDIDS 由 Iowa 州立大學於 2000 年所提出，整個系統由許多單一功能的代理人程式共同組成。該系統利用 Software Fault Tree (SFT) 與 Colored Petri Net (CPN) 來描述分散式攻擊行為，其 CPN 分析結果並能直接對應到所需的代理人程式，甚至自動產生 Java classes 。圖 11 為 MAIDS 所採用的入侵偵測系統發展程序概念圖，首先是利用 SFT 建立入侵行為模型，以求得偵測需求；接下來再利用加強型 SFT 進一步描述信任關係 (trust) 、時間性關係 (temporal) 與前後關係 (context) ，以精煉入侵行為模型，減少可能的誤報；而以加強型 SFT 表達的入侵模型可進一步對應成以 CPN 表示的入侵偵測模型，並轉換成 Java 模板與代理人程式，經實際實作與分派代理程式後，完成入侵偵測系統的建構；最後是系統的測試，不過目前該研究尚未完成這部分的工作。

十、CARDS

CARDS 由學者 Ning 等人於 2001 年提出，其目的在建構一套非集中式的入侵偵測系統，以偵測分散式入侵行為。

CARDS 系統主要包括了 signature manager 、monitor 與 directory service 等元件， signature manager 主要負責產生入侵行為的特徵 (signatures) 、分解該特徵與將分解後的特徵散佈到相關的 monitors 。monitor 為該系統負責入侵偵測的元件，如圖 12 所示 monitor 上有許多 probes 會分別自資料來源收集資訊，較特別的是該系統會為每個 probe 建立 system

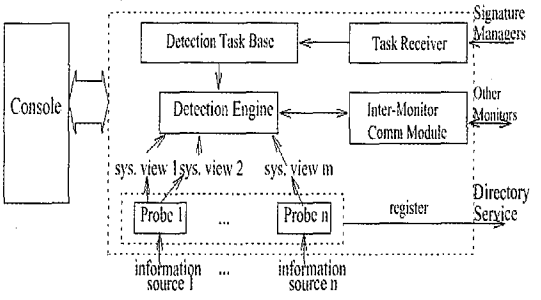


圖12：CARDS Monitor架構圖

views，並將該 views 註冊於 directory service 中，使各 monitors 能從抽象層次了解各 probe 所能提供的資訊，根據文中的描述 probes 除了可能是感應器外，也可能包含了簡單的分析器，如有些 probes 能提供 TCPDoSAttacks 或 Land attacks 等 system views。此外 monitor 還具備直接與其他 monitors 溝通的能力，以共同合作完成分散式入侵偵測之任務。CARDS 利用 labeled directed-graph 來描述分散式入侵行為。各 nodes 間的 edges 代表各事件間的時間關係。在完成入侵特徵的一般性描述後，signature manager 可依各 system views 實際的存在位置（相關的 system views，由哪些 Monitors 提供），將一般性描述轉為明確性描述，並利用各事件間的時間關係，找出事件間的相依關係，並以 work-flow tree 以繪出，以協助完成該偵測任務之工作分派。

十一、IDML〔Tseng, 2001〕

IDML 為交通大學資訊科學實驗室所發展的入侵偵測語法語言，整體架構分為兩大部份，網路封包經第一線 Snort 防禦後，將封包經 Parser 轉入 IDML 機制內處理與分析，以期找出入侵模式進而抵禦入侵，防治效果不錯，本研究即以 IDML 為比較對象。

以下，我們介紹以類神經網路的學習功能來實現混合式的入侵偵測系統。

伍、學習機制探討

洪蘭 [1996] 認為，學習 (learning) 是自然生物或人造系統之所以有智慧的一極為重要特徵。儘管對學習行為的解釋及分類有各種不同的理論，但學習與記憶是密不可分的，因為有學習行為的發生才導致記憶的形成；能夠記憶才會產生學習的效果。本研究試圖實作出具有即時線上偵測能力的 IDS 入侵偵測系統。根據文獻收集中發現，類神經網路系統，具有相當良好的學習與辨識效果，且文獻中共同展現出一致的特性如下：[Lee et al., 2001][蘇木春等 ., 2000] [Bonifacio et al., 1998]

1. 學習能力：由於神經元間的連結是透過神經節，而神經節本身是可特過權重值加以調整的，因此神經網路具有強大的學習能力。
2. 聯想記憶能力：神經網路如果有少數神經元或連節受損，並不損及其正常功能，其原因在於神經網路資訊儲存是分散式記憶 (Distributed Memory)，也就是資訊散佈在許多連節（神經節）上。因此即使小部份連結受損，並不會造成嚴重的後果，而僅是造成功能略為降低。也因為分散記憶的關係，對於不完整或有雜訊的輸入也能正確的處理，亦即具有模糊推論 (Fuzzy Reasoning) 的能力。
3. 平行處理能力：類神經網路採用大量的平行計算，經由許多不同的人工神經元來做運算處理，因此資料在網路中是在同一時段中，以平行的方式被處理。
4. 錯誤容忍能力：類神經網路在運作時具有很高的錯誤容忍度，如果輸入資料混雜少許的雜訊干擾，仍然不會影響其運作的正確性。而且即使有部分的人工神經元失效，整個神經網路仍能繼續運作。

也因如此，本研究選擇類神經網路來實做 IDS 系統中的學習與辨識部分。根據 [Haykin, 1994] [蘇木春 等 ., 2000] 將類神經網路依網路架構分成單層前饋網路、多層前饋網路、循環式網路與晶格狀

網路，而學習方式的部分也被分成監督式學習與非監督式學習，將分別介紹如下：

1.監督式學習(supervised learning)

從範例中學習，它是目前應用最廣泛的神經網路架構，主要是由問題領域中取得訓練範例，從中學習輸入變數與目標輸出變數的內在對應規則，以應用於新的案例，根據此一概念所發展的演算法有：感知機網路、倒傳遞網路、機率神經網路與學習向量化網路 ... 等等，應用領域包含：分類、診斷、預測 ... 等等。

2.非監督式學習(unsupervised learning)

從觀察及發現中學習，與監督式最大的差異在於前者必須在訓練期的時候提供網路正確的輸出資料給網路修正學習；而後者則不需提供正確的學習結果。非監督式學習是由網路本身，去發掘出資料本身的重要特徵、結構、規則或是其間的關係。

監督式學習中的倒傳遞神經網路與非監督式學習中的自組織映設圖網路 (SOM) 和霍普菲爾神經網路 (Hopfield neural network)，皆具有相當好的學習與辨識能力，但霍普菲爾神經網路大都應用在電子電路、動力學與統計力學上，且此網路有記憶容量的限制，且其輸入值分成離散型與連續型，離散型僅可輸入二元值，連續型輸入也被限制在 0 與 1 間的任意實數，故如果要將霍普菲爾神經網路應用於本研究的 IDS 入侵偵測系統可能不適合，本研究搜索國外的一些相關研究發現倒傳遞神經網路於 IDS 入侵偵測已經有相關研究陸續出現 [Lee et al., 2001][Bonifacio et al., 1998]，並且也發現國外的研究較偏向理論的研究，且大部分皆以攻擊行為作為學習對象，但本研究認為現今社會上的攻擊行為日新月異，實在很難對其攻擊行為定義，有可能今天所學習的攻擊行為樣本，明日就不適用了。為了突破此瓶頸故採用自組織映設圖網路 (SOM) 將重點放在正常行為的特徵擷取，實際測試自組織映設圖網路 (SOM) 應用於 IDS 入侵偵測系統的實際成效如何？底下內容就針對自組織映設圖網路做一簡單介紹。自組織映設圖網路 (SOM) 基本的架構請參考圖 13，網路結構包含了輸入層、輸出層與網路連結層。其中輸入層：用以表現網路的輸入向量，其處理單元 (Neuron) 數依問題大小而定。輸出層：用以表現網路的輸出向量，一般由一維或二維的 Neuron 組成，模擬腦細包的排列，其處理單元數依問題大小而定。至於網路連結層：每個輸出層單元和輸入層單元相連結的加權值所構成的向量，表示一個訓練資料對映樣本點聚類之形心座標。

自組織映設圖網路 (SOM) 的運作過程，分為學習過程與回想過程兩部分。

※學習過程 (Learning Phase)：

1. 計算輸入向量與各輸出處理單元的歐氏距離平方。
2. 具有最小的歐氏距離平方的輸出處理單元稱為優勝單元 (Winner)，找出優勝單元。
3. 以優勝單元作為臨近中心，並根據臨近半徑定出臨近區域。
4. 調整臨近區域中的輸出處理單元與輸入處理單元之間的網路連結加權值，使得在臨近區域內所有 Neuron 座標 (即這些加權值) 向輸入向量的座標方向靠近，而且越靠進臨進中心的 Neuron，其移動的允許幅度就越大。
5. 依續將所有的輸入向量餵入 SOM 網路中，重覆步驟 1 ～ 4。

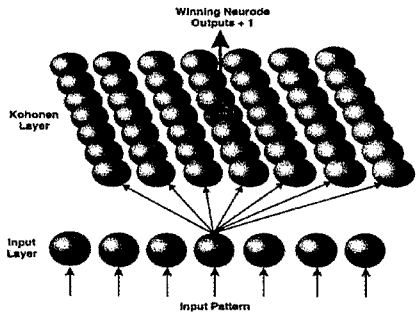


圖13：二維 Kohonen 模型

6. 步驟 1 ～ 5 稱爲一次 Iteration ，在每次的 Iteration 結束後，縮小臨近半徑的大小，並開始另一次新的 Iteration ，如此反覆訓練，直到輸入訓練樣本的聚類座標收斂爲止。

※回想過程 (Retrieving Phase) ：

1. 將在學習過程所得到的網路連結矩陣載入到 SOM 網路中。
2. 輸入測試向量到 SOM 網路的輸入層。
3. 計算輸入向量與各輸出處理單元的歐式平方距離。
4. 找出輸出層中的優勝單元。

陸、本研究 WIDS 入侵偵測系統架構

由於目前網路上各種攻擊方式日新月異，各種新的攻擊手法不斷翻新，因此常常使得入侵樣式資料庫的內容追隨不上最新的攻擊手段。爲了更有效的提高偵測率，我們往往必須隨時更新入侵資料庫的內容。如此一來，將會導致入侵樣式資料庫會隨著攻擊型態的增加，而呈現線性狀態的擴大。這將可能會影響系統的執行效率與系統資源。因此，在目前入侵測系統的研究領域中，多朝向引入人工智慧系統的概念，希望藉由人工智慧的學習理念，使得入侵偵測系統具有自動學習的功能，針對目前現有之攻擊樣式進行學習動作，以期一旦新型態的攻擊模式產生時，可以在第一時間發佈警告訊息。基於上述的文獻探討與整理，本研究將嘗試提出一套綜合坊間方法將 misuse model 和 anomaly model 結合在一起的入侵偵測系統。由於網路上的封包變化頗大，較無法確知是否爲正常或異常封包，因此網路防護上，本文採用依異常攻擊行爲整理成規則的網路式 misuse model 。而主機上的使用者，其正常使用行爲卻是可判斷的，因此在主機防護上，本文採用依正常行爲判斷的 anomaly model 。其中 misuse model 將採用 Snort 所提供的攻擊行爲 rule ，採用封包擷取的方式做爲防禦攻擊之第一道防線， anomaly model 將採用類神經網路 SOM Neural network 的學習機制實做，整體結合成本研究之 WIDS 入侵偵測系統。此架構將可以有效預防駭客使用翻新手法入侵。圖 14 即爲本研究 WIDS 入侵偵測系統架構：

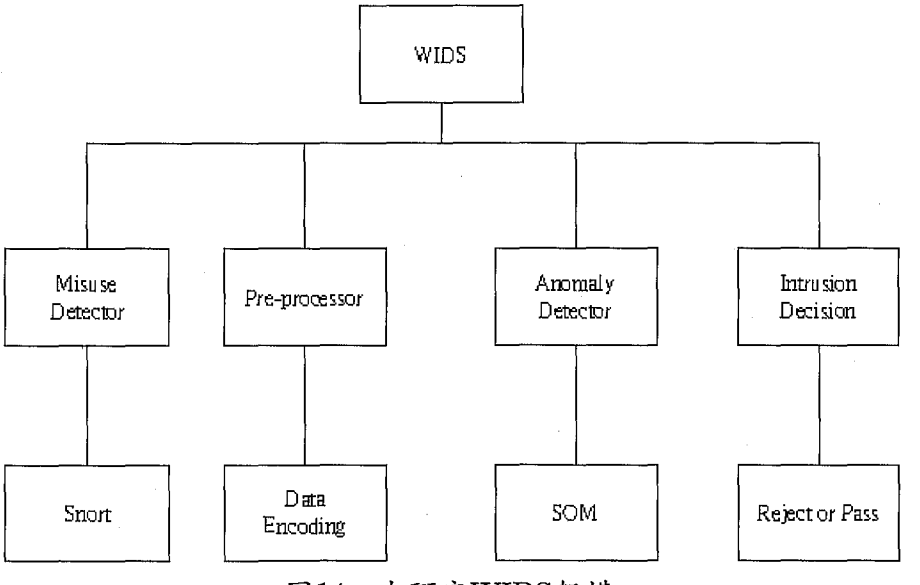


圖14：本研究WIDS架構
(資料來源：本研究整理)

本研究 WIDS 入侵偵測系統架構主要分成兩階段，一是離線 SOM 類神經學習階段，一是即時線上入侵偵測階段。其中共有四大組成要件，分別介紹如下：

第一道機制 *Misuse detector* ：

Misuse detector 收集我們想要之網路封包並讀取每個 TCP 封包的封包標頭與資料內容。例如指定欲收集之連到某主機位址之封包、封包類型等。此機制採用 Snort V1.83 版本，Snort 雖然自稱其爲陽春型入侵偵測系統，但安裝與設定上相當複雜繁瑣，尤其在設定規則時。Snort 網站尚提供規則資料庫 (rule database) 供人直接使用，也提供線上規則產生器，當選擇所需要的規則後，便可以自動產生之。規則範例如下：

範例 1 ：

alert tcp any any ->192.168.1.0/24 143 (msg:"IMAP Buffer overflow!"; content: "|90E8 C0FF FFFF|/bin/sh";)

以上 Rule 規則表示：如果偵測到通訊埠 143 遭受攻擊時 (IMAP) ，這個 Rule 規則指令會指定 Snort 應該產生警告，請注意劃底線的內容，這類型攻擊與 [Taeho, 1998] 破解內容有關。

範例 2 ：

alert tcp any any -> 192.168.1.0/24 80 (msg:"PHF attempt"; content:"/cgi-bin/phf";)

以上 Rule 規則表示：讓 Snort 去觀察每一個封包流，檢查是否有 PHF 漏洞 [Paul, 2000] ，當 Snort 過濾到攻擊封包時即會產生警告。截至 2002/05/31 ， Snort 共提供 1195 種攻擊行爲 Rule 。 Snort 是個僅需些許系統資源，即可做到快速、可靠的網路型入侵偵測工具，它可以利用竊聽 sniffit 功能一一過濾個別封包流，並藉由特徵、編譯及在執行不利於系統的行爲時，加入攻擊戳記並產生產生警告。本研究即是採用 Snort V1.83 版本加上 Snort 所提供的規則庫作爲入侵偵測系統之第一道防線。

第二道機制 *Preprocessor* ：

主要是利用 Misuse detector 收集到之網路封包，從中擷取所想要之輸入變數，並將這些輸入變數編碼轉成類神經網路之輸入格式。基於需要用到字串比對之考量，實作上，這個部份使用 Java 語言撰寫。

第三道機制 *Anomaly detector* ：

主要是在訓練與測試模式下執行。經過 Preprocessor 處理後的輸入將會送到這個元件來作訓練與測試。在此，我們使用 php 撰寫 SOM 程式以利線上即時偵測判斷。此機制共分成兩個階段，一是離線類神經 SOM 學習階段，一是線上類神經 SOM 異聯想階段，描述如下：

1. 類神經 *SOM* 離線學習階段

根據 [蘇木春 等 , 2000] 提到 SOM 演算法的主要目標，就是以特徵映射的方式，將任意維度的輸入向量，映射至一維或二維的特徵映射圖上。當 SOM 趨向於收斂的狀態時，特徵映射圖會試著盡量伸展以佈滿整個輸入空間。本研究試圖利用此方法於離線階段中將使用者執行系統程式的正常行爲學習出來，當離線學習階段完成後，此時類神經網路中神經元與神經元中的加權值已經配置完畢。實際建置過程與結果將分別於下節做詳細說明。

2. 類神經線上異聯想偵測階段

假設攻擊者能夠成功騙過 Snort 通過第一道防線，此時系統尚會將這些疑似攻擊封包導入類神經 SOM 中，進一步判斷。由於學習階段中我們已習得正常使用者行爲，故此時類神經的異聯想辨識過程，若發現使用者已經偏離正常行爲特徵，即會輸出 1 表示攻擊行爲成立，如果

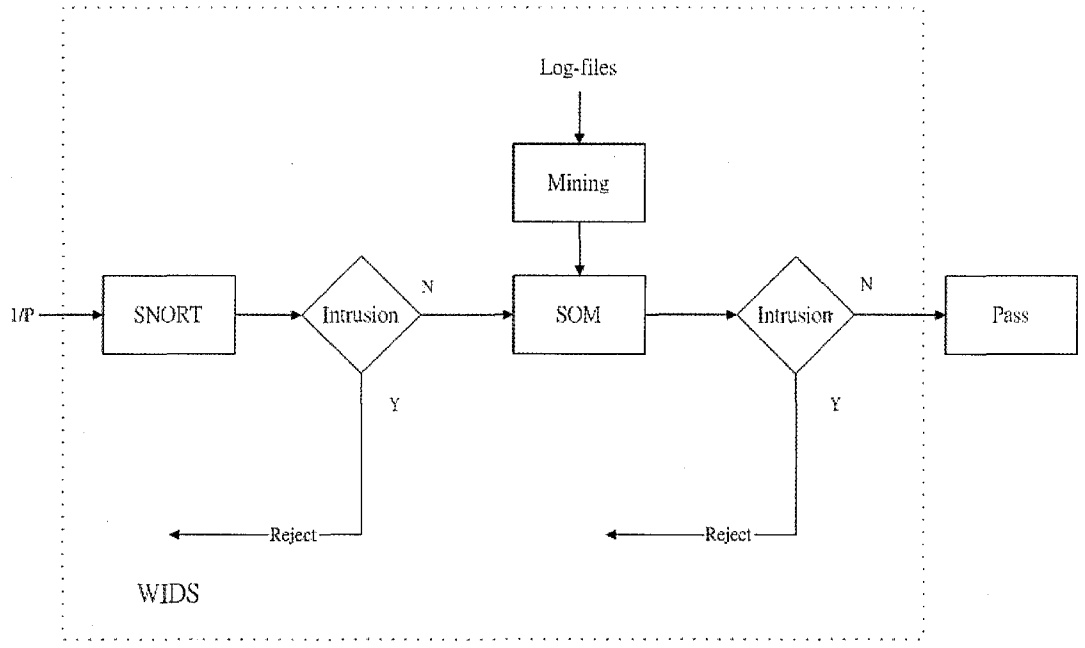


圖15：WIDS系統運作流程

辨識後發現疑似攻擊行為尚在正常行為特徵的範圍內，類神經將輸出 0 表示攻擊行為不成立，此行為仍在正常許可範圍內。攻擊行為的封包將於轉往下一機制。

第四道機制 *Intrusion Decision* ：

Intrusion Decision 是用以偵測入侵之重要元件。封包經由 Anomaly detector 訓練與測試後，得出兩種可能之輸出：0 或 1 。若輸出為 0 則判斷為正常；若輸出為 1 ，則系統判斷輸入為入侵，此時將會有相關之警告訊息顯示在螢幕上。當攻擊特定行為特徵發生時，我們必需制定相關的行動政策 [Stardust, 1999] 。

本研究的 Intrusion Decision 動作即是當攻擊行為確定發生後，可執行的一連串自動 Reject 動作，此 Reject 動作包括將其 IP 位址加入 Snort rule 規則中，達到即時拒絕攻擊者的入侵，並將攻擊者 IP 寫入 TCP Wrappers 中的 host.deny 中，拒絕此 IP 存取 Server 中所有資源包括 FTP Server ，且將攻擊封包摘要內容與攻擊者 IP 動態寫入 Mysql 資料庫中，且當一波攻擊完畢後，系統會將此波攻擊過程摘要使用 E-Mail 通知管理者。

1. 系統開發環境

本研究 WIDS 系統開發環境主系統為計算機中心 Personal Web 網頁申請主機 Linux 系統，開發工具包括： Java JDK1.4 + Apache1.3 + php4.0 + Database Mysql 3.23 。

2. 系統運作

系統操作的過程主要是利用 Misuse detector 來取得區域網路上每個 TCP 封包，在所蒐集到的網路封包送到 Anomaly detector 或送到 Intrusion Decision 之前，先利用 Preprocessor 將有用之網路封包資訊擷取並轉成類神經之輸入格式。Anomaly detector 則是用來訓練與測試 WIDS 。Intrusion Decision 則是完成訓練與測試後，由 Anomaly detector 產生之結果及其他程式所構成，而其系統運作流程則如圖 15 所示。

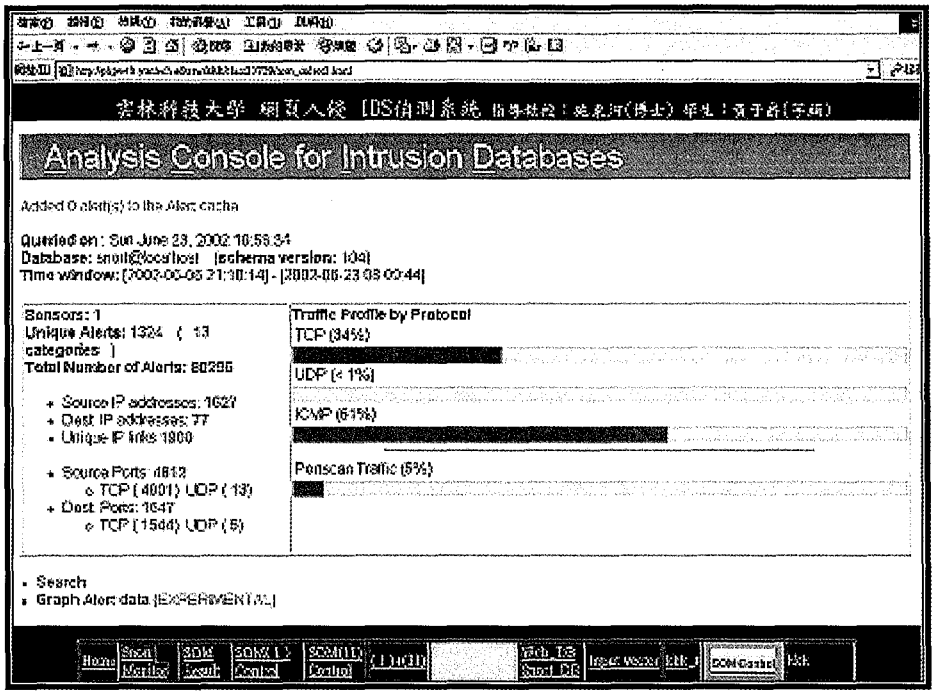


圖16：本研究 WIDS系統主畫面

圖 16 即是本研究 WIDS 系統主畫面，此 WIDS 監控管理畫面，僅有 MIS 管理員方可進入維護，此處尚有設置兩道密碼保護鎖，使用 JAVA 所寫的一套 RSA 密碼防火牆程式，僅允擁有 Private Key 之使用者方可進入維護，且僅允許使用者輸入 ID 與 Password 三次，即會將 IP 鎖定，Reload 將無效，嚴格禁止一般 User 不法潛入造成系統危機。此 WIDS 第一道防禦機制的主要功能包括線上監控所有攻擊封包流量（請參考圖 16）、監控已經判斷出來的攻擊 IP 行爲（請參考圖 17）、察看特定攻擊封包的詳細內容包括封包表頭 (Header)（請參考圖 18）與查詢 Search 追蹤攻擊者的特定資料（請參考圖 19）。本系統設定 Snort 擷取所有封包進行過濾與偵測，且會將 Snort 所偵測過濾到的攻擊封包完整寫入 Mysql 資料庫中。Snort 管理介面，採用美國危機處理中心 (CERT) ACID(Analysis Console for Intrusion Database) 軟體，此軟體使用 PHP 撰寫且開放原始碼，故本研究使用 PHP 將之改寫使之與資料庫 Mysql 搭配，本研究第三道 SOM Neural Network 防禦機制使用 Java JDK + PHP4 撰寫亦搭配 Mysql 資料庫，故整合後的整體系統將使管理人員可相當輕易的維護整套 WIDS 系統。

圖 20 爲線上資料庫維護系統，內容包括計算機中心 Personal WEB 會員所有資料、會員所有網站行爲、SOM 判斷結果與攻擊 IP 警告區。學習正常行爲執行步驟如下，1. 先選擇系統學習的人數，2. 點選 Random 按鈕，系統即會 Random 至資料庫中取出會員個人相關資料。3.mining 按鈕，至資料庫中擷取出會員執行系統程式之正常行爲，4. 決定類神經相關的控制變數，包括 Output 神經原與學習次數。設定完成後點選 Exit 按鈕，畫面，將可看到動態學習過程，包括總執行時間計時。圖 21 爲 Intrusion decision 拒絕服務機制之輸出畫面：

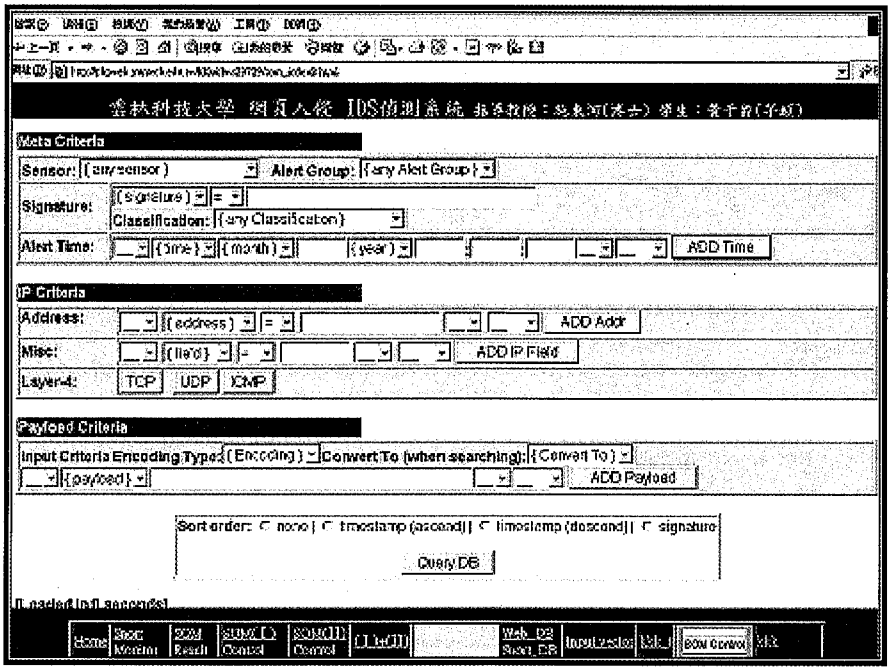


圖19：搜尋特定IP 之搜尋介面

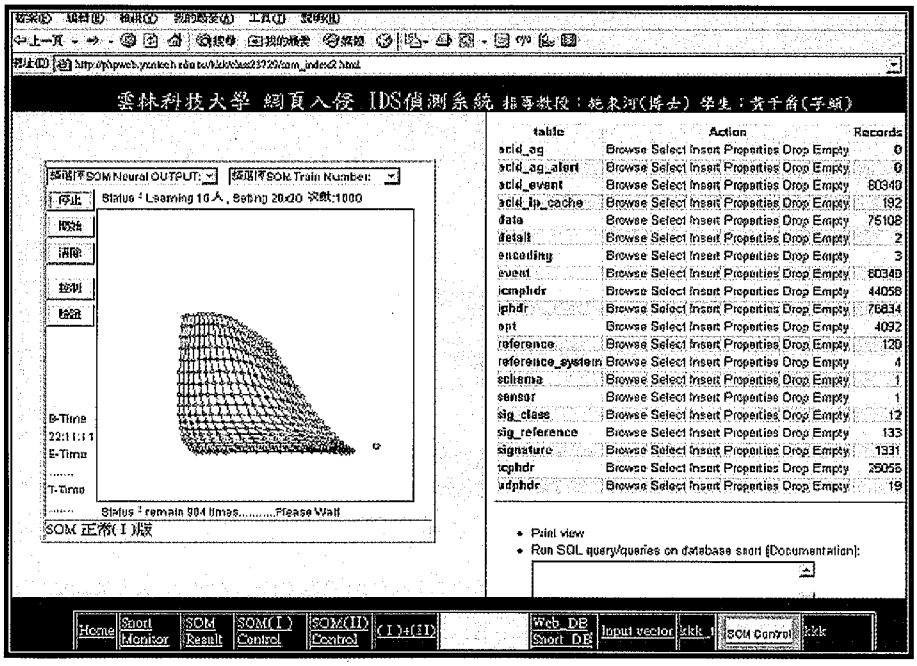


圖20：線上資料庫維護系統

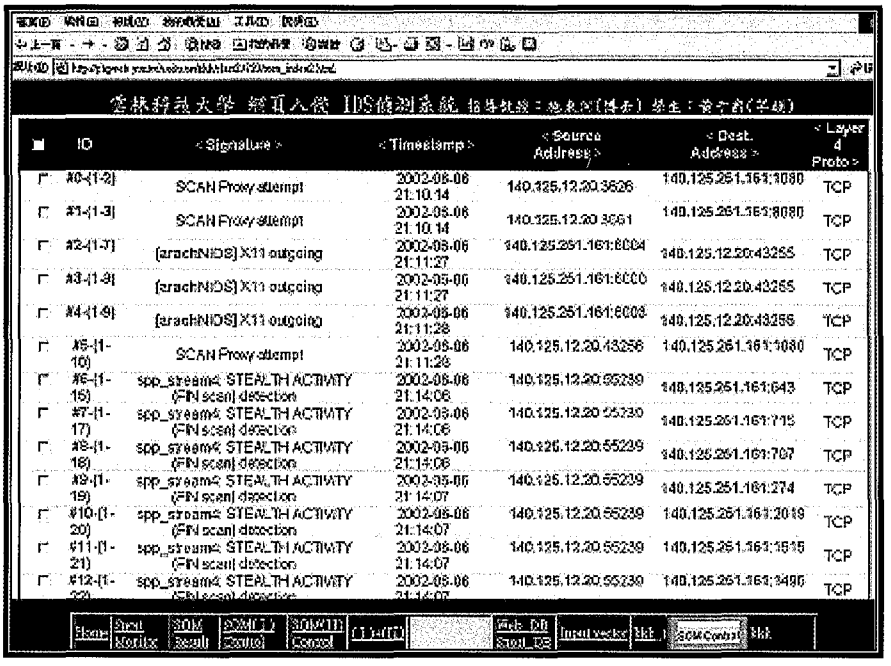


圖17：監控IP行為

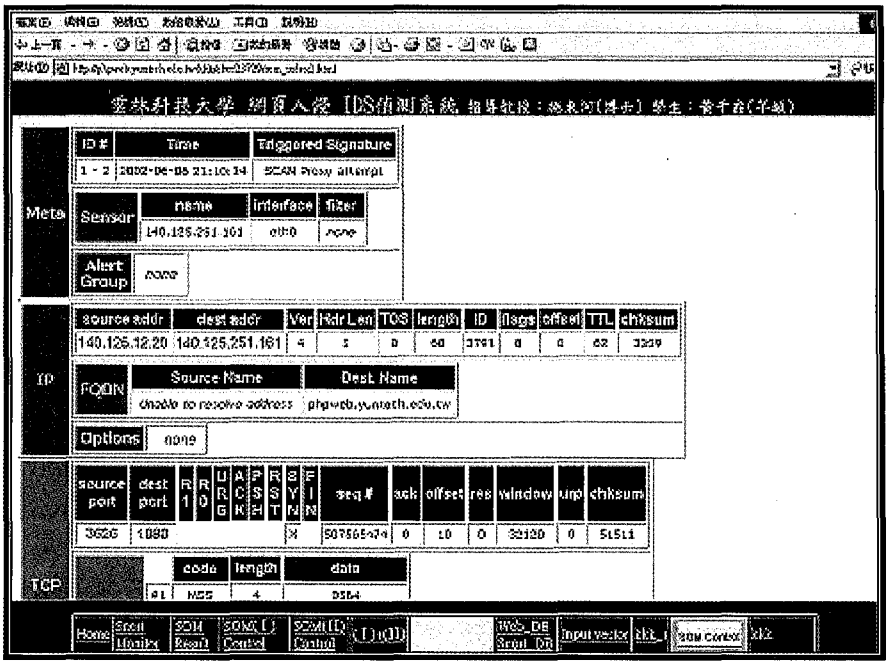


圖18：察看IP 詳細內容

圖21：為Intrusion decision 拒絕服務機制之輸出畫面

柒、實驗設計

本研究 WIDS 網頁入侵偵測系統以雲科大計算機中心個人網頁系統 (Personal Web) 為保護與防禦的對象，實際測試防禦入侵的效果。實驗設計重點在測試使用 SOM 能否成功與正確的將一般使用者的特徵擷取出來，接著驗證其所擷取的 SOM 特徵能否於線上成功的執行異聯想過程將偏離正常行為的使用者找出來。本研究即是根據以上重點設計出以下實驗，實驗一主要在驗證 SOM 能否成功學習出正常行為特徵，實驗二則使用目前網路上流通較知名的攻擊軟體工具實際攻擊計算機中心個人網頁系統 (Personal Web)，驗證安裝本研究之 WIDS 入侵偵測系統其成效如何？

一、樣本說明

雲林科大計算機中心個人網頁系統 (Personal Web) 正式服務日期為 2001/07/15，本研究即是以此日期為開端至 2002/05/25，共 10 個月的稽核日誌檔 (如圖 22) 當作總樣本，內容包括完整的使用者行為、程式執行時間，與程式執行方法。且本研究已經將這 10 個月的稽核日誌檔輸入資料庫中，此 10 個月的稽核日誌檔所記錄的使用者行為共有 951187 筆個人記錄。截至 2002/05/25，雲林科大計算機中心個人網頁系統共有 404 位通過認證的使用者共同使用此個人網頁系統服務。

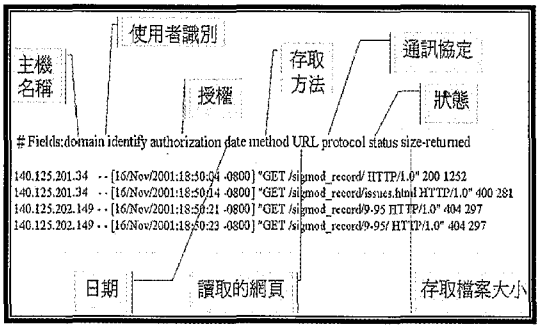


圖22：稽核日誌檔範例

二、實驗一

實驗目的：驗證類神經 SOM 是否已經學習出正常行爲特徵。

實驗步驟：

1. 藉由圖 20 SOM 學習行爲控制畫面，以亂數 Random 擷取出 5 ,10 ,20 , ... ,100 ,200,...,404 位使用者，並將其執行系統程式的正常行爲找出來，經過向量轉換的動作，並丟入類神經 SOM 中進行特徵擷取與學習的動作，並找出最佳收斂模型。
2. 透過控制介面，將所有 404 位正常使用者行爲丟入 SOM 中進行驗證，驗證 SOM 是否已經成功的將使用者正常行爲特徵學習出來。
3. 根據稽核日誌檔中的 404 人正常行爲總值計算本階段正確率。

實驗一結果分析

經過實驗，以隨機抽取 50 位使用者行爲當成訓練資料其收斂情形最佳，實驗結果顯示，總會員 404 位正常執行程式的行爲共有 17178 筆，共有 14 筆誤判記錄，正確率爲 (17178-14) / 17178= 0.9991 ，僅有 0.0009 的誤判率。故本研究認爲以類神經 SOM 學習出來的行爲特徵足以代表總會員 404 位使用者執行系統程式之正常行爲。

※實驗一結果【 0.0009 誤判率】追蹤說明：

追蹤發現 14 筆誤判記錄中每筆資料對伺服器的要求 (Request) 行爲皆爲空值，且要求執行之程式檔案大小 (Size) 皆爲零，經過與專家討論後認定此發生原因有以下兩項：

1. 網路不穩：例如當使用者正在執行個人網頁系統 (Personal Web) 之網頁申請行爲時，突然斷線，可能發生此狀況。
2. 使用者不熟悉使用方式：例如當使用者正在執行個人網頁系統 (Personal Web) 之網頁申請行爲時，因爲不熟悉其介面操作方式，或者程式執行中點選 Reload 鍵，可能發生此狀況。

三、實驗二

實驗目的：實際攻擊測試，審視本系統防禦效果如何。

實驗步驟：

1. 使用目前網路上流通較知名的攻擊軟體工具 (如表 1) ，加上手動攻擊共 22 種攻擊方式，其中手動攻擊方式共有三種包括資料隱碼、竄改參數與緩衝區溢滿攻擊 (Buffer Overflow attack) 。實際攻擊計算機中心個人網頁系統，驗證安裝本研究之 WIDS 入侵偵測系統其成效如何？
2. 計算 WIDS 入侵偵測系統的正確率。

實驗二攻擊結果分析

表 2 即爲實驗二實際攻擊 WIDS 系統之結果，其中 Snort 之正確率 =61.78% 而本研究之 WIDS 正確率 =99.14% 。

表2：實驗二實際攻擊WIDS系統之結果

Attack Tools	Classification	Snort accuracy	WIDS accuracy
Nmap	Probing (Normal)	1167/1713 = 68.1%	68.1% + 539 / 1713 = 99.5%
	Probing (Stealth)	855/1614 = 52.9%	52.9% + 759 / 1614 = 99.9%
SATAN	Probing (Vulnerability)	110/110 = 100 %	100% + 0 / 110 = 100 %
SAINT (Web版)	Probing (Vulnerability)	83/122 = 67.2%	67.2% + 39 / 122 = 99.1%
Nessus	Probing (Vulnerability)	443/623 = 71.1%	71.1% + 171 / 623 = 98.5%
Storm-DDOS	DOS (1min)	6351/6526 = 97.3%	97.3% + 118/ 6526 = 99.1%
	DDOS (2pc.1min)	12749/13252 = 96.2%	96.2% + 130/ 13252 = 97.2%

Attack Tools	Classification	Snort accuracy	WIDS accuracy
Raw-clones	DOS (1min)	5221/5409 = 96.5%	96.5% + 146/ 5409 = 99.2%
	DDOS (2pc.1min)	11334/11818 = 95.9%	95.9% + 142/ 11818 = 97.1%
DDOS Ping V2.00	DOS (1min)	4173/4329 = 96.4%	96.4% + 43/ 4329 = 97.3%
	DDOS (2pc.1min)	8285/8712 = 95.1%	95.1% + 118/ 8712 = 96.4%
Udp Flooder	DOS (1min)	3759/3908 = 96.2%	96.2% + 82/ 3908 = 98.3%
SYN Flooder	DOS (1min)	4707/4784 = 98.4%	98.4% + 53/ 4784 = 99.5%
Smurf2k	DOS (1min)	6216/6455 = 96.3%	96.3% + 65/ 6455 = 97.3%
CodeRed	系統漏洞(CodeRed V1)	221/221 = 100 %	100% + 0 / 221 = 100 %
	系統漏洞(CodeRed V2)	2876/2876 = 100 %	100% + 0 / 2876 = 100 %
Nimda Worm	系統漏洞	4412/4412 = 100 %	100% + 0 / 4412 = 100 %
PHF attack	程式漏洞	60/60 = 100 %	100% + 0 / 60 = 100 %
HTTP Bomber V1.001b	程式漏洞	1536/3000 = 51.2%	51.2% + 1440 / 3000 = 99.2%
snot-0.91a.tar.gz	程式漏洞	987/1719 = 57.4%	57.4% + 721 / 1719 = 99.3%
ShutDownSQL	程式漏洞	0/109 = 0 %	0% + 109 / 109 = 100 %
Access Diver V4.76	程式漏洞-密碼破解	0/1958 = 0 %	0% + 1958 / 1958 = 100 %
Hackware V1.0 Beta Build 43	程式漏洞-密碼破解	0/1542 = 0 %	0% + 1542 / 1542 = 100 %
資料隱碼 (手動60times)	程式漏洞	0/60 = 0 %	0% + 60 / 60 = 100 %
竄改參數 (手動60times)	程式漏洞	0/60 = 0 %	0% + 60 / 60 = 100 %
php-exploit (手動60times)	程式漏洞	0/60 = 0 %	0% + 60 / 60 = 100 %
Buffer Overflow (256bits) (手動60times)	程式漏洞	0/60 = 0 %	0% + 60 / 60 = 100 %
		總平均：61.78%	總平均：99.14%

(資料來源：本研究整理)

※實驗二正確率算法說明：

正確率的算法共包含兩個部分，一是 Snort accuracy 一是本系統 WIDS(Snort + SOM) 之整體正確率。發動攻擊者皆爲本作者故本研究可根據攻擊時間點與作者 IP 輕易統計出總共的攻擊封包量 T ，也可輕易的於資料庫中查詢出 WIDS 所偵測到的攻擊封包數 A ，故 A/T 即可獲得偵測正確率。

例如表 3 Nmap Probing(Normal) 情況，1713 表示所有攻擊封包 T ，1167 表示本系統第一道防禦機制 (即 Snort) 所偵測到的攻擊封包數目 A ，故 T/A 即可求得 Snort accuracy 等於 68.1% 。至於攻擊

表3：偵側正確率－計算範例

Attack Tools	Classification	Snort accuracy	WIDS accuracy
Nmap	Probing(Normal)	1167/1713=68.1%	68.1%+539/1713=99.5%
	Probing(Stealth)	855/1614=52.9%	52.9%+759/1614=99.9%

量中有 546(1713 - 1167 = 546) 個封包未被偵測出來，這些 546 個封包將會被系統導入 SOM 防衛機制中進行偵測判斷，從表 3 中可發現，從未被發現的 546 個攻擊封包中被 SOM 判斷出 539 個攻擊封包。故本研究 WIDS 整體正確率即是 68.1% + 539 / 1713 = 99.5% 。

※實驗二攻擊結果分析

從表 2 實驗二結果可發現，本研究 WIDS 入侵偵測系統的整體平均偵測正確率有 99.14% ，遠比 Snort 的 61.78% 高出許多，現在本研究就分別針對 Snort 與 SOM 防禦機制其無法偵測的攻擊行爲作如下解析：

1. Snort 防禦機制中無法偵測的狀況

根據表 2 可發現 Snort 對於較知名的攻擊方式皆有不錯的偵測率例如 100% 偵測的包括：撒旦 (SATAN) 、 Code Red 、 Nimda Worm 與 PHF 攻擊，但是當 Snort 碰到變化型攻擊方式則就力不從心，例如撒旦 (SATAN)WEB 版 、 Nmap Probing (Stealth) 偷偷掃瞄方式與 HTTP Bomber ，這些工具 Snort 皆僅有平均 56% 左右的偵測正確率。另外尚有新的攻擊工具是針對攻擊 Snort Rule 而來的，此工具 Snort 本身僅有 57.4% 偵測率，對系統仍俱有相當威脅性的。另外從表 2 中也可發現 Snort 有偵測率 0% 的情況，此表示著這類型的攻擊行爲， Snort 根本無法偵測出來，再細部觀察表 2 察看 0% 的攻擊方法，可發現這些攻擊方法皆是利用程式漏洞的攻擊方式，包括資料隱碼 (Sql Injection) 、竄改參數 、 ShutDown SQL 、 Access Driver(密碼破解) 、 Hackware (密碼破解) 、 php-exploit 與緩衝區溢滿攻擊。雖然 Snort 偵測率不高，但從表 2 中我們可發現 Snort 對於阻斷式攻擊 DOS 與分散式阻斷式攻擊 DDOS 皆有不錯的偵測率，從 Storm-DDOS 至 Smurf 攻擊等 6 種阻斷式攻擊工具平均 Snort 有 96.4% 的偵測正確率， 本研究 WIDS 入侵偵測系統因安裝 Snort 在最前端幫整體系統檔掉大部分的垃圾攻擊封包。另外， Snort 本身提供監聽 (Sniffer mode) 封包功能，故本研究可輕易改寫程式將通過 Snort 防線的所有封包導入 SOM Neural Network 中進行第二階段處理，此也是本系統採用 Snort 當第一道防禦機制的原因之一。

2. 本研究 WEB IDS 入侵偵測系統無法偵測的狀況

根據表 2 可發現本研究 WIDS 入侵偵測系統整體偵測正確率達到 99.14% ，但系統爲什麼仍會發生誤判偵測不到攻擊者的情況？ 故本研究根據資料庫中攻擊封包資料與 Snort 資料庫中實際攻擊封包資料經過交叉比對的結果發現，此情況大部分都是發生在阻斷式攻擊 DOS 甚至是分散式阻斷式攻擊 DDOS 所造成的大量封包湧入系統時，造成封包遺失的情況導致，此部分的偵測率下降到 96.4% (由 DDOS Ping 攻擊工具所測試出來的數據) 。另外根據表 3 尚可發現於程式漏洞的攻擊手法中，本研究 WIDS 系統偵測正確率皆達到 100% ，此現象應屬正常，因爲本研究利用類神經 SOM 將正常使用者其使用程式的習慣與特徵學習出來，且經過實驗一驗證得知 SOM 已經將計算機中心 Personal Web 中的總會員 404 位執行程式

的習慣與特徵學習完成，事後驗證正確率亦達 99.91%，故本系統對於程式漏洞的防範俱有相當高的成效。

捌、結論與未來研究

本文除廣泛搜集現有市面上所有的網站入侵行為與攻擊軟體外，並根據國內外入侵偵測系統的探討，提出綜合 Network-based misuse model 與 Host-based anomaly model 的 WIDS 入侵偵測系統。本研究嘗試使用類神經 SOM 去學習並擷取一般使用者執行程式的行為特徵，實驗一結果分析顯示，SOM 具備學習正常行為的能力，誤判率低，擷取行為特徵效果不錯。而本研究為了更嚴謹的測試 WIDS，故設計實驗二使用 22 種攻擊手法實際攻擊本研究 WIDS 入侵偵測系統效能，結果發現本研究 WIDS 入侵偵測系統的整體平均偵測正確率可高達 99.14%。另外本研究將研究數據與交大資科曾憲雄教授所提的 IDML(Intrusion Detection Markup Language) [Tseng, 2001] 作一廣泛比較，整理如表 4 所示，由於 U2R 與 R2L 的 telnet 與後門程式攻擊方式較特殊，不適合 SOM 入侵判斷，故無法提升正確率，但從表 4 中可發現本研究整體平均正確率仍達 86.36%，皆優於其他方法。

一、研究限制

本研究雖經實際測試效果不錯唯仍有以下限制：

1. 本研究 WIDS 必須安裝於 Linux 中，因開發工具皆使用 Linux 版本之 Apache、PHP、Perl、Mysql 資料庫、JAVA JDK 與 SNORT。
2. 本研究 SOM 類神經部分僅讓系統學習網站中的個人網頁申請之使用者正常行為，並未包括 Telnet 或 FTP 等其它服務的正常使用者行為。
3. 本系統 WIDS 可阻擋 DOS 攻擊，但是無法阻擋癱瘓網路攻擊。

二、未來研究方向

未來研究可嘗試讓 SOM 學習系統中其它的服務行為，例如 FTP、Telnet 或者公司內部的特定網路行為模式，系統學習完成後，即可將資料庫中的權重值應用至線上機制中，讓系統去判定有否偏離的正常行為發生。本系統的最大好處就是不用更新病毒碼或稱攻擊特徵碼，如果發現系統偵測率有下降的趨勢時，可由管理者透過網站 (Web) 管理介面再將正常行為重新學習即可。

另外亦可朝向分散式系統設計著手，一般傳統的網路入侵偵測系統主要是安裝

表4：IDML與本系統WIDS正確率比較表：

	SNORT	SNORT with the IDML [Tseng, 2001]	本研究WIDS
Probing	67%	73%	99.4%
U2R	62%	68%	同SNORT (62%)
R2L	59%	70%	同SNORT (59%)
DoS	65%	74%	97.93%
系統漏洞			100%
程式漏洞			99.85%
Average	65.2%	73.6%	86.36%

在單一的網路區段內，因此其偵測的範圍往往僅止於該區段。對於如 DDoS 這種同時由多處發動攻擊的情形便難以偵測出來。因此，目前實務上多傾向於發展分散式的網路入侵偵測系統，將各區段的網路監控由各區的 IDS 所監測，但在中央系統則由一中央管理程式集中控管，進一步分析及偵測各區網路受攻擊的情況，以突破網路環境的受限。如知名入侵偵測系統 NetRanger 、 EMERALD 與 NetStat 皆是採用分散式概念設計出的成品。分散式 IDS 最大優點就是可讓系統負載平衡，尤其遇到分散式阻斷攻擊 DDOS 時，可有效將攻擊封包阻擋於區域網路外。若能將本系統提升至分散式系統，將可有效解決本研究所碰到的計算延遲、封包遺失等導致誤判的狀況。

備註：本研究 WIDS 系統現已實際應用於雲林科技大學個人網頁系統，除自動 reject 機制，因考量使用者的抱怨並未啓用外，餘皆運作正常。

網址： http://phpweb.yuntech.edu.tw/

參考文獻

1. 丘偉權，2001，"以類神經網路建構入侵偵測系統"，國立成功大學電機工程學系碩士論文。
2. 李勁頤、陳奕明，分散式入侵偵測系統研究現況介紹，資訊安全通訊，2002,Vol.8, No.2, pp. 38-61, 3 月
3. 李駿偉、田筱榮、黃世昆，" 入侵偵測分析方法評估與比較 "，資訊安全通訊， 2002,Vol.8, No.2, pp. 21-37, 3 月
4. 洪蘭（譯），天生嬰才：重新發現嬰兒的認知世界，遠流出版社， 1994 。
5. 鄭有倫，2000，" 具反偵察能力之分散式網路入侵偵測系統之設計與實現 "，國立成功大學電機工程學系碩士論文。
6. 蘇木春、張孝德，機器學習：類神經網路、模糊系統及基因演算法則 II 版，全華科技，台北， 2000 。
7. 鮑友仲，再談「資料隱碼」攻擊，Hackland 駭客資訊網， http://www.hackland.idv.tw/data_attack-1.htm ， 2002
8. Anderson J.P, "Computer security threat monitoring and surveillance," Technical Report, James P. Anderson Co., Fort Washington, PA, 1980.
9. Bace Rebecca and Mell Peter, "NIST Special Publication On Intrusion DetectionSystems", http://www.21cfr11.com/files/library/government/intrusion_detection_systems_0201_draft.pdf.
10. Balasubramaniyan Jai Sundar, Joe Omar Garcia-Fernandez and Isacoff David ., "An Archiecture for Intrusion Detection using Autonomous Agents," COAST Technical Report 98/05, 1998.
11. Bauer D.C., Cannady J. and Garcia R.C., "Detecting anomalous behavior: optimization of network traffic parameters via an evolution strategy", SouthEastCon Proceedings, IEEE, 2001, pp. 34 -39.
12. Biswanath Mukherjee, L. Todd Heberlein, Karl N. Levitt, "Network Intrusion Detection", IEEE network, 1994.
13. Bonifacio J.M.Jr. and Cansian A.M., "Neural Networks Applied in Intrusion Detection System", Neural Networks Proceedings, IEEE World Congress on Computational Intelligence, 1998, vol.1, pp. 205 -210
14. CERT/CC,"InternetSecurityOverview"2001 ,http://www.cert.org/present/cert-overview-trends/module-2.pdf
15. CERT/CC, "Statistics 1998-2002

Number of incidents reported", http://www.cert.org/stats/cert_stats.html, 2002.

16. CNET ，" http://taiwan.cnet.com/news/ec/story/, 2001,10 月。

17. Feiertag R., Rho S., Benzinger L. and Wu S., T. Redmond, "Intrusion Detection inter-component adaptive negotiation," Computer Networks, 2000, vol. 34, pp 605 ~ 621

18. Graham Robert, "FAQ: Network Intrusion Detection System," version 0.8.3, http://www.robertgraham.com/pubs/network-intrusion-detection.html, March 2000.

19. Haykin S., " Neural Networks :A Comprehensive Foundation", Macmillan College Publishing Company, Inc., 1994.

20. Helmer Guy, Wong Johnny and Slagell Mark., "A Software Fault Tree Approach to Requirements Analysis of an Intrusion Detection System," To appear, Proceedings of the 1st Symposium on Requirements Engineering for Information Security, October 2000.

21. Kemmerer R.A., "NSTAT: A Model-based Real-time Network Intrusion Detection System," Technical Report TRCS97-18, November 1997, Computer Science Dep., University of California Santa Barbara.

22. Kohonen T., Kangas J.A. and Laaksonen J.T., "Variants of self-organizing maps", Neural Networks, IEEE Transactions on , 1990, Volume: 1 Issue: 1, pp. 93 -99

23. Lee Susan C. and Heinbuch David V., "Training a Neural-Network Based Intrusion Detector to Recognize Novel Attacks" Systems, Man and Cybernetics, Part A, IEEE Transactions on , 2001, Volume: 31 Issue: 4 ,pp. 294 -299.

24. Lee Wenke, Numbalkar R.A. and Yee K.K., "A data mining and CIDF based approach for detecting novel and distributed intrusions," In Proceedings of 3rd International Workshop on the Recent Advances in Intrusion Detection, October 2000.

25. McClure, Joel Scambray and George Kurtz, 2001 ，駭客現形第二版，尤焙麟譯，麥格羅．希爾，台北。

26. Michael Sobirey, 2001, "Currently 92 Intrusion Detection Systems", http://www-rnks.informatik.tu-cottbus.de/~sobirey/ids.html

27. Ning P., Jajodia S. and Wang X.S., "Abstraction-based Intrusion Detection in Distributed Environments," ACM Transactions on Information and System Security (TISSEC), November 2001, 4 (4), pp 407 ~ 452.

28. Porras Philip A. and Neumann Peter G, "EMERALD: Event monitoring enabling responses to anomalous live disturbances," In Proceedings of the 20th National Information Systems Security Conference, October 1997, Baltimore, Maryland, USA, National Institute of Standards and Technology/ National Computer Security Center. , pages 353 ~ 365

29. SANS Resources, " The Twenty Most Critical Internet Security Vulnerabilities Updated The Experts Consensus " , http://www.sans.org/top20.htm ， May, 2002,

30. Theuns Verwoerd and Ray Hunt, " Security architecture testing using IDS-a case study", Computer Communica-

tions, 2002 ,Volume: 25, Issue: 15, pp. 1402-1412.

31. Tseng S.S., Lin Y.T. and Lin S.C., "An Intrusion Detection Model Based Upon Intrusion Detection Markup Language IDML", Journal of Information Science and Engineering , 2001,17, pp. 899-919.

32. Vigna G. and Kemmerer R., "NetSTAT: A Network-based Intrusion Detection Approach," in Proceedings of the 14th Annual Computer Security Application Conference, Scottsdale, Arizona, December 1998.