

國 立 中 央 大 學

資 訊 管 理 學 系 碩 士 班

碩 士 論 文

合 作 式 防 火 牆 之 設 計 與 應 用

指 導 教 授：陳 奕 明 博 士

研 究 生：游 啟 勝

中 華 民 國 九 十 二 年 六 月



國立中央大學圖書館 碩博士論文授權書

(91年5月最新修正版)

本授權書所授權之論文全文與電子檔，為本人於國立中央大學，撰寫之碩/博士學位論文。(以下請擇一勾選)

() 同意 (立即開放)

() 同意 (一年後開放)，原因是：_____

() 同意 (二年後開放)，原因是：_____

() 不同意，原因是：_____

以非專屬、無償授權國立中央大學圖書館與國家圖書館，基於推動讀者間「資源共享、互惠合作」之理念，於回饋社會與學術研究之目的，得不限地域、時間與次數，以紙本、光碟、網路或其它各種方法收錄、重製、與發行，或再授權他人以各種方法重製與利用。以提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

研究生簽名：_____ 游 啟 勝 _____

論文名稱：_____ 合作式防火牆之設計與應用 _____

指導教授姓名：_____ 陳 奕 明 博 士 _____

系所：_____ 資 訊 管 理 研 究 所 _____ 博士 碩士班

學號：_____ 9 0 4 2 3 0 3 5 _____

日期：民國 _____ 92 _____ 年 _____ 6 _____ 月 _____ 20 _____ 日

備註：

1. 本授權書請填寫並親筆簽名後，裝訂於各紙本論文封面後之次頁(全文電子檔內之授權書簽名，可用電腦打字代替)。
2. 請加印一份單張之授權書，填寫並親筆簽名後，於辦理離校時交圖書館(以統一代轉寄給國家圖書館)。
3. 讀者基於個人非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

合作式防火牆之設計與應用

研究生：游啟勝

指導教授：陳奕明 博士

國立中央大學資訊管理學系碩士班

論文摘要

隨著網路應用的普及與多元化，網路的安全問題逐漸被人們所重視。目前防火牆已經成為大多數企業的第一道網路安全防線，同時也是最重要的攻擊回應機制，且未來幾年內，防火牆仍然會是相當重要的網路安全防禦機制。但現有的防火牆因為部署位置及運作架構的限制，遭遇愈來愈多的問題，也漸漸無法防禦日新月異的攻擊手法。

本研究首先整理及分析防火牆的演進及目前的問題，進而以分散式防火牆為基礎，加上縱深防禦及合作防禦的概念，提出一套合作式防火牆系統，各合作式防火牆主機與其它防禦機制可進行合作防禦來達到入侵預防的目的。本研究將探討合作式防火牆的數種合作防禦方式及其中的困難點，並提出對應的解決方案，包括提出一種以 XML 為基礎的通用規則來解決合作防禦時的溝通及分散式防火牆的管理問題，及一種網蟲防禦方法以解決網蟲擴散時的內部網路癱瘓問題。

論文中也將說明合作式防火牆的系統架構、運作流程及模組設計，並以系統雛型展示解決網蟲的內部網路癱瘓問題及與入侵偵測系統進行合作防禦來抵禦攻擊，藉此說明合作式防火牆系統的效用及應用方式。

關鍵字：分散式防火牆、入侵預防、縱深防禦、合作防禦、網路安全、XML

The Design and Applications of Cooperative Firewalls

Author: Chi-Sheng Yu

Advisor: Dr. Yi-Ming Chen

Department of Information Management
National Central University, Taiwan

Abstract

Because of the popularity and variety of network applications, network security is getting respected by people. Today, firewalls are the first line of defense of network security in most enterprises, and are also the most important mechanism of attack response. However, firewalls that are restricted by deployed positions and their architectures now suffer more and more challenges, and they also can't defend more and more new attacks.

In this thesis, we analyze the evolutions and problems of firewalls, and then develop a cooperative firewall system which is based on the distributed firewall and the concepts of defense in depth and cooperative defense. All firewalls in the cooperative firewall system can cooperate with other defense mechanisms to achieve intrusion prevention. We first present some possible schemes of cooperative defense with cooperative firewall system and discuss their difficulties. Then we propose solutions to solve these difficulties. The solutions include a new generic rule based on XML to solve the communication problems in cooperative defense and the management problem of distributed firewalls, and a detection and defense method of internet worm to solve the problem of network jam when worms spreading.

We also propose the system architecture, operating procedures, and module design of our cooperative firewall system and build a prototype system that is able to solve the network jam of internet worm and make cooperative defense with intrusion detection system to explain the efficiency and applications of the cooperative firewall system.

Keyword : Distributed Firewall, Intrusion Prevention, Defense in Depth, XML, Cooperative Defense, Network Security

致謝辭

本論文能夠順利完成，最要感謝的是我的指導教授陳奕明老師，在論文撰寫過程中花費許多時間與我討論，並不吝給予建議及修正研究方向，也解決許多我提出的疑惑。感謝口試委員辜國隆主任、黃世昆博士、周立德博士在論文口試時，提出許多確切的建議及給予的鼓勵。感謝林熙禎老師在實驗室報告時對論文的建議及指導。

在中央大學六年的求學生涯即將畫下句點，最後兩年的研究生涯中尤其印象深刻，雖然都是在忙碌中度過，但也感覺自己成長了不少。在這六年的日子中，感謝許多人陪我走過各個時期。首先感謝管院計中陳麗玉助教六年來在各方面的照顧及關心，尤其是助教對我的肯定及信心。感謝實驗室政耀學長、大為學長、及勁頤學長在專案、生活、及論文上給予的協助，沒有你們，長達兩年的研究生生活真不知道會變成如何。感謝同實驗室的曾韻一起陪我同甘共苦克服課業、專案及生活上的挑戰，我永遠記得我們曾在專案報告前一天徹夜不眠，拼命地努力趕進度。感謝柏嘉、清文、佳生、英嘉陪我一起走過在實驗室的時光，尤其是最後趕論文的苦日子中，每晚犧牲陪我吃宵夜及聊天。

也感謝其他實驗室同學惠名、雅涵、文雄、典正有事沒事陪我閒聊、逼促我運動。感謝大學時代的好友沛宏、孟佳、介仁、仲元常常陪我閒聊、出遊及消磨時間，伴我度過無數無聊及心情不好的時間。感謝志緯學長給予的論文建議，及與長成學長共同致贈的畢業禮物及祝福。

感謝宇瑞學長在專案計畫中給予的具體建議，及自己的有趣經驗。感謝中央大學計算機中心的各老師在我做計中計畫時給予的照顧。

最後要感謝的是我的父母親，沒有你們的養育及照顧，就沒有現在的我。

謹將此論文獻給我的父母及所有關心及照顧過我的人…謝謝你們！

游啟勝 謹誌

2003 年夏夜 于中央大學

目錄

| | |
|-------------------------------|-----------|
| 目錄..... | I |
| 圖目錄..... | III |
| 表目錄..... | IV |
| 第一章 緒論 | 1 |
| 第一節 研究背景..... | 1 |
| 第二節 研究動機及目的..... | 3 |
| 第三節 研究範圍與限制..... | 4 |
| 第四節 研究流程..... | 4 |
| 第五節 章節架構..... | 5 |
| 第二章 相關研究 | 7 |
| 第一節 防火牆演進分析..... | 7 |
| 2.1.1 網路式及主機式防火牆..... | 8 |
| 2.1.2 分散式防火牆..... | 10 |
| 2.1.3 防火牆面臨的挑戰..... | 12 |
| 第二節 縱深防禦與合作防禦..... | 15 |
| 第三節 入侵預防..... | 16 |
| 第四節 本章小結..... | 18 |
| 第三章 合作防禦方式與困難點分析 | 20 |
| 第一節 防火牆之間進行合作防禦..... | 20 |
| 第二節 防火牆與入侵偵測系統進行合作防禦..... | 21 |
| 第三節 防火牆與漏洞掃描系統進行合作防禦..... | 22 |
| 第四節 通用規則..... | 24 |
| 3.4.1 採用通用規則的原因及目的..... | 24 |
| 3.4.2 通用規則概念..... | 25 |
| 3.4.3 相關作法回顧..... | 26 |
| 3.4.4 通用規則語法..... | 27 |
| 3.4.5 通用規則的套用方式..... | 33 |
| 3.4.6 通用規則的支援模組..... | 36 |
| 第五節 網蟲的內部網路癱瘓問題..... | 37 |
| 3.5.1 網蟲的偵測方式..... | 38 |
| 3.5.2 門檻值的建立..... | 40 |
| 第六節 本章小結..... | 41 |
| 第四章 系統架構與設計 | 42 |
| 第一節 設計原則..... | 42 |
| 第二節 系統概觀..... | 43 |
| 第三節 運作流程..... | 44 |
| 第四節 內部模組設計..... | 49 |
| 4.4.1 防火牆節點..... | 50 |
| 4.4.2 註冊節點..... | 52 |
| 4.4.3 管理節點與DNS Server | 52 |

| | |
|-----------------------|-----------|
| 第五章 模擬實驗 | 54 |
| 第一節 模擬實驗一 | 54 |
| 第二節 模擬實驗二 | 56 |
| 第三節 模擬實驗結果討論 | 58 |
| 第六章 結論 | 61 |
| 第一節 研究結論 | 61 |
| 第二節 研究貢獻 | 61 |
| 第三節 未來研究方向 | 62 |
| 參考文獻 | 63 |
| 中文參考文獻 | 63 |
| 英文參考文獻 | 63 |

圖目錄

| | |
|--|----|
| 圖 1-1：2001 年至 2005 年全球資訊安全產品市場規模 | 1 |
| 圖 1-2：本研究之研究流程圖 | 5 |
| 圖 2-1：以應用程式為基礎的規則設定方式 | 10 |
| 圖 2-2：縱深防禦及合作防禦概念示意圖 | 16 |
| 圖 3-1：防火牆之間進行合作防禦示意圖 | 21 |
| 圖 3-2：分散式防火牆與漏洞掃描系統合作防禦示意圖 | 23 |
| 圖 3-3：通用規則概念示意圖 | 25 |
| 圖 3-4：通用規則組成示意圖 | 28 |
| 圖 3-5：References 元素組成示意圖 | 28 |
| 圖 3-6：RuleValid 元素組成示意圖 | 29 |
| 圖 3-7：RuleContent 元素組成示意圖 | 29 |
| 圖 3-8：系統平台描述資訊組成示意圖 | 30 |
| 圖 3-9：應用程式描述資訊組成示意圖 | 32 |
| 圖 3-10：網路連線描述資訊組成示意圖 | 32 |
| 圖 3-11：合作式防火牆通用規則套用流程圖 | 34 |
| 圖 3-12：規則套用決策樹 | 35 |
| 圖 3-13：通用規則範例部份內容 | 35 |
| 圖 3-14：判斷規則是否保留之決策樹 | 35 |
| 圖 3-15：通用規則語言支援模組示意圖 | 37 |
| 圖 3-16：網蟲偵測門檻值計算範例之網路架構圖 | 40 |
| 圖 4-1：合作式分散防火牆系統概觀示意圖 | 44 |
| 圖 4-2：改良式架構註冊流程圖 | 45 |
| 圖 4-3：改良式架構規則傳送流程圖 | 46 |
| 圖 4-4：註冊方式傳輸量比較圖 | 48 |
| 圖 4-5：新節點註冊運作流程圖 | 49 |
| 圖 4-6：分組傳送註冊資訊示意圖 | 49 |
| 圖 4-7：防火牆節點內部設計架構 | 51 |
| 圖 4-8：註冊節點內部設計架構 | 52 |
| 圖 4-9：管理節點內部設計架構 | 53 |
| 圖 4-10：DNS Server 節點內部設計架構 | 53 |
| 圖 5-1：Slammer 網蟲偵測畫面 | 55 |
| 圖 5-2：Slammer 網蟲阻擋規則 | 55 |
| 圖 5-3：snort 偵測到 Nmap 時之警示訊息 | 57 |
| 圖 5-4：由攻擊警示產生的防火牆規則 | 57 |
| 圖 5-5：Nmap 受防火牆阻擋後之部份掃描畫面 | 58 |
| 圖 5-6：防火牆節點與入侵偵測系統之合作防禦示意圖 | 60 |

表目錄

| | |
|--------------------------------------|----|
| 表 1-1：攻擊演進趨勢及其意義 | 2 |
| 表 1-2：2002 年前後五年前五大攻擊事件 | 3 |
| 表 2-1：三種防火牆綜合比較表 | 15 |
| 表 3-1：通用規則回應動作列表 | 30 |
| 表 3-2：系統平台描述資訊各元素及屬性之意義與範例 | 31 |
| 表 3-3：應用程式描述資訊各元素及屬性之意義與範例 | 31 |
| 表 3-4：網路連線描述資訊各元素及屬性之意義與範例 | 32 |
| 表 3-5：三種作業系統下之系統平台描述資訊蒐集模組實作方式 | 36 |

第一章 緒論

隨著網際網路的快速普及，網路攻擊及入侵事件日益增加，網路安全防護逐漸受到重視。由於防火牆的概念簡單且部署後的效果顯著，目前已成為大多數企業網路安全的第一道防線及最重要的網路安全防禦機制。不過隨著各種網路應用的發展、網路環境的變化及攻擊手法的推陳出新，目前的防火牆已無法適合某些應用環境及阻擋日新月異的攻擊手法。本研究針對現有防火牆的盲點加以改進，提出一套合作式防火牆系統，能與其他安全防禦機制進行合作防禦來阻擋新型態的攻擊手法及減少管理人員的負擔。

第一節 研究背景

根據 1999 年的 Computer Security Institute/FBI Computer Crime and Security Survey[7]指出，約 91% 的組織已部署了防火牆。Datamonitor 於 2002 年 7 月的研究報告也預測防火牆與虛擬私有網路 (VPN) 在未來幾年仍會是資訊安全產品中市場規模最大的產品 (圖 1-1)，防火牆儼然成為企業組織最普遍的網路安全防禦設備。防火牆藉由控制網路的重要入口，如同公司大門的警衛，檢查來往的網路通訊，再根據組織制訂的通行規則決定放行與阻擋，成為網路安全防禦的第一道防線。

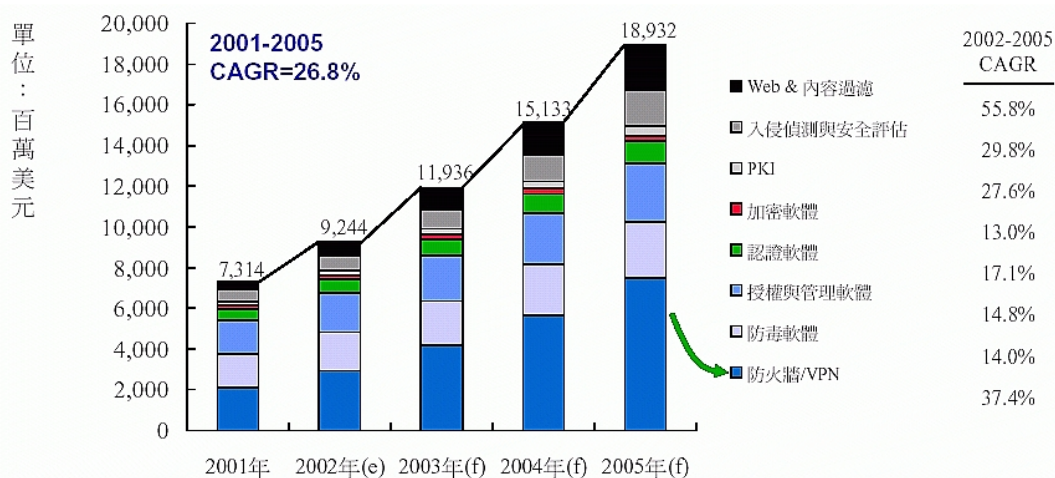


圖 1-1：2001 年至 2005 年全球資訊安全產品市場規模

(資料來源：[1]、DataMonitor)

根據 CERT/CC 於 2002 年的攻擊趨勢報告[8]指出，目前的攻擊正朝攻擊工具自動化、攻擊工具精密化、漏洞被發現的速度增快、對防火牆穿透能力增加、不對稱攻擊增加、及對網路基礎建設的攻擊增加六大趨勢演進。第一、第二項趨勢使得攻擊及擴散的速度增快，且更難被網路安全機制所偵測。第三項趨勢使得管理者更難維持系統的安全性。第四項趨勢證明目前的防火牆逐漸無法阻擋新式的攻擊。

表 1-1：攻擊演進趨勢及其意義

| 趨勢 | 說明 | 對策 |
|--------------------------|---|--|
| 攻擊工具自動化程度增加 攻擊工具的速度增快 | <ul style="list-style-type: none"> ● 掃描動作自動化 ● 攻擊動作自動化 ● 散播動作自動化 ● 整合性攻擊工具的增加 | <ul style="list-style-type: none"> ● 偵測到攻擊後，需要即時阻擋及防治機制來防禦快速的攻擊行為 |
| 攻擊工具精密化程度增加 | <ul style="list-style-type: none"> ● 反蒐證能力增加 ● 動態攻擊行為 ● 攻擊工具模組化 | <ul style="list-style-type: none"> ● 整合更多資訊來偵測攻擊 ● 需要整合其他安全機制進行合作防禦 |
| 漏洞被發現的速度增快 | <ul style="list-style-type: none"> ● 漏洞被發現的數量增加及發現速度增快，管理人員很難維持系統的安全性 ● 漏洞公佈與被攻擊的時間間隔縮短 | <ul style="list-style-type: none"> ● 建立自動化漏洞稽核、修補、防護機制 ● 對所有主機採取重大漏洞主動防禦措施 |
| 對防火牆穿透能力增加 | <ul style="list-style-type: none"> ● 可穿透防火牆的技術及協定增多 | <ul style="list-style-type: none"> ● 傳統防火牆逐漸無法阻擋攻擊 |
| 不對稱攻擊增加 | <ul style="list-style-type: none"> ● 主機遭受的攻擊與網路其他主機安全性相關 | <ul style="list-style-type: none"> ● 建立主機的信任關係，拒絕無信任主機的通訊 |
| 對網路基礎建設的攻擊增加 | <ul style="list-style-type: none"> ● 分散式阻斷服務 ● 網蟲攻擊的增加 ● 對 DNS 的攻擊 ● 對 Router 的攻擊 | <ul style="list-style-type: none"> ● 運用合作式防禦減少大規模的攻擊 ● 重要主機都應具備自身防禦機制 |

(資料來源：[2,8] 及本研究整理)

Information Security 雜誌於 2002 年 11 月提出了過去五年及未來五年最嚴重的攻擊事件 [9] (表 1-2)，可以看見其結果與 CERT/CC 所提出的攻擊趨勢相去不遠：具有自動化攻擊及擴散能力的網蟲逐漸流行、不對稱攻擊及對網路基礎建設的攻擊增加。可預見的是未來的攻擊勢必將更快、更難偵測、規模更大，損害也將更嚴重。

表 1-2：2002 年前後五年前五大攻擊事件

| 1997~2002 | 2003~2008 |
|---------------------------------|-------------------|
| CodeRed (2001) | "Super" Worms |
| Nimda (2001) | 隱密性攻擊 |
| Melissa/Loveletter (1999, 2000) | 對自動更新程式的攻擊 |
| 分散式阻絕服務攻擊 (2000) | 對 Routing/DNS 的攻擊 |
| 木馬程式的遠端控制 (1998-2000) | 結合實體及虛擬世界的攻擊 |

(資料來源：[9])

第二節 研究動機及目的

雖然目前網路安全防禦機制相當眾多，除了防火牆之外，還有入侵偵測系統、誘補系統等。但防火牆仍是目前最有效的攻擊回應機制，藉由阻擋攻擊連線，才能確實中斷攻擊的進行。舉例來說，入侵偵測系統偵測到攻擊後，仍需要與防火牆配合來對攻擊連線進行阻擋。防火牆對網路安全防禦的重要性可見一斑。防火牆雖然整合愈來愈多的功能，但是運用的基本原理仍然相同：控制網路重要入口，藉由檢查網路封包及比對事先設定的政策規則來達到網路安全防禦的目的。但由於網路環境的變化，防火牆用來當作運作基礎的原理已面臨挑戰，使得防火牆漸漸無法阻擋新式的攻擊手法。其中最明顯的例子為防火牆無法對網蟲擴散時所造成的分散式阻斷服務攻擊及網蟲的大量感染做一有效防禦。

本研究的目的是在於解決目前防火牆所遭遇的問題，包含網路式防火牆的主機移動、內部攻擊、通訊加密、對動態協定處理困難、錯誤影響程度高、效能要求程度高、設定複雜等問題，及主機式防火牆管理困難的問題。另一個目的是在防

火牆系統內加入合作防禦的概念，使得防火牆系統能與其他安全機制進行合作防禦達成入侵預防。因此，本研究採用分散式防火牆的架構，並加入了通用規則及適當的軟體模組來協助進行合作防禦，最後提出一套合作式防火牆系統，並對其系統架構、運作流程，及模組設計加以說明。

第三節 研究範圍與限制

本研究的研究範圍在提出一套由分散式防火牆為基礎的合作式防火牆系統來彌補目前防火牆的不足，並就合作式防火牆與其他安全機制可能的合作防禦方式進行探討。

本研究藉由更改防火牆架構及運作方式來解決目前防火牆所面臨的問題。但部份安全威脅，如軟體漏洞問題，並不屬於防火牆可解決之安全問題範圍，故本研究仍然無法解決該類問題。在合作防禦方面，本文對合作式防火牆進行合作防禦的對象、目的及方式進行探討，但合作防禦的效用會受限於所合作對象的功能，例如與入侵偵測系統合作防禦時，入侵偵測系統無法偵測到的攻擊便無法進行防禦。本研究亦不涉及合作防禦對象的研究範圍，例如入侵偵測系統的誤報率及漏報率高低，便不屬於本研究的範圍。

第四節 研究流程

本研究的研究流程如圖 1-2 所示，分為以下數個階段：

1. 防火牆的演進及問題分析

此階段討論防火牆的演進原因，並針對各種防火牆所遭遇的問題進行分析及探討。

2. 入侵預防概念探討

此階段探討入侵預防及入侵偵測與預防的概念及目前作法。

3. 合作防禦方式研究

此階段探討數種合作式防火牆進行合作防禦的對象、目的及方式進行。

4. 合作防禦問題及對策研究

此階段對各合作防禦方式可能出現的問題及對應解決方式進行研究。

5. 系統設計

依據前述之概念，提出合作式防火牆系統，並對系統架構、角色功能及內部設計、及運作流程進行解說。

6. 模擬實驗

以模擬攻擊情境證明合作式防火牆的可行性及功用。

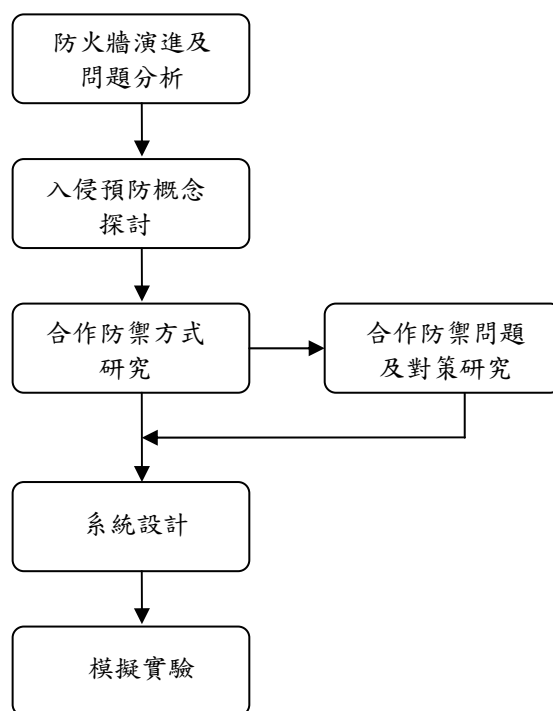


圖 1-2：本研究之研究流程圖

第五節 章節架構

本文第二章為相關研究，對防火牆的演進及目前面臨的問題作一整理，並對縱深防禦、合作防禦及入侵預防的概念加以討論。第三章為合作防禦方式與困難點分析，提出合作式防火牆可能進行合作防禦的對象、運作方式及困難點，並提

出本研究的解決方案。第四章為系統架構與設計，提出合作式防火牆的系統架構、運作流程及模組設計。第五章為模擬實驗，以解決網蟲內部網路癱瘓問題及與入侵偵測系統進行合作防禦為例說明合作式火牆的效用及運作方式。第六章為結論，對本研究做一結論，並提出貢獻及未來研究方向。

第二章 相關研究

本章就防火牆的演進及現有的問題加以探討，並解釋縱深防禦、合作防禦、及入侵預防的概念及其作法。第一節以網路式防火牆、主機式防火牆及分散式防火牆說明防火牆的演進及目前所面臨的挑戰。第二節對縱深防禦及合作防禦的概念及作法進行探討。第三節解釋入侵預防的意義，並提出合作防禦如何用來達成入侵預防。第四節則為本章小結，綜合各節內容下一簡要結論。

第一節 防火牆演進分析

根據 Cheswick 及 Bellovin 的定義 [10]，防火牆是位於兩個網路間的系統，並具有以下特性：

1. 所有由內到外及由外到內的通訊都必須經過防火牆。
2. 只有經過安全政策 (security policy) 授權通過的通訊才可以通過防火牆。
3. 防火牆本身不會被侵入。

防火牆本質上是做存取控制的動作，對來往的通訊進行檢查再依安全政策來決定讓通訊通過或是加以阻攔。防火牆的安全政策的定義通常有兩種策略[3]：預設允許 (default allow) 及預設拒絕 (default deny)。

預設允許策略代表防火牆預設對所有通訊都採取允許通過的動作，藉由定義額外的拒絕政策規則 (policy rule) 來阻擋危險通訊。而採用預設拒絕策略時，防火牆預設阻擋所有通訊，對於想要允許通過的通訊必須制訂與其對應的允許政策規則。

預設允許策略的防火牆設定較容易，只需要對想要拒絕的通訊制訂對應的阻擋政策規則即可。相對的，採用預設拒絕策略，在設定防火牆時必須針對所有允許的通訊制訂允許策略規則。因此在無法以應用程式為基礎進行政策規則設定的防火牆上，當想讓某應用程式的通訊能通過防火牆時，必須清楚地瞭解此應用程式運作時所使用的協定及通訊埠。要讓一台主機的網路功能正常運作，通常需要

設定許多允許策略規則。

但採用預設拒絕策略通常會比採用預設允許策略來得安全，原因是預設拒絕策略時，會阻擋大部份通訊，只有少數管理者設定的已知通訊允許通過。因此，我們可以專注防守這些少數允許通過的通訊是否含有攻擊行為。而採用預設允許策略時，由於能通過的通訊眾多，防守起來也特別困難。

根據防火牆部署的位置，可以將防火牆分為兩大類。第一類是網路式防火牆，另一類是主機式防火牆。網路式防火牆位於內部網路對外的重要入口上，而主機式防火牆則安裝於各台欲受保護的主機上。兩者的基本功能及原則相同，都是利用檢查來往的通訊，再比對政策規則後，判斷對此通訊進行阻擋或放行。不過主機式防火牆只對安裝主機送出或接收的連線封包進行檢查。最近幾年，有人提出分散式防火牆來加強多個主機式防火牆的管理功能。以下對三種防火牆演進做一整理。

2.1.1 網路式及主機式防火牆

網路式防火牆，又稱為閘道式防火牆，其位置位於外部網路與內部網路之間，對兩個網路間來往的通訊，依安全政策規則加以檢查來決定是否放行或阻擋通訊。由於防火牆只能針對經過的通訊加以檢查，所以網路式防火牆對於同一個網路內進行的通訊便束手無策，因為同一個網路內的通訊根本不會經過防火牆，防火牆便無法檢查通訊。因此，網路式防火牆無法防禦來自內部的攻擊。此外，若是有主機離開內部網路，此主機也無法受到防火牆的保護。

主機式防火牆直接安裝於需要受保護的電腦主機上，個人式防火牆即屬於此類防火牆。主機式防火牆運作原則與網路式防火牆相同，一樣對經過的通訊加以檢查，並依據安全政策規則來決定是否放行。但是由於主機式防火牆直接安裝於主機中，所以只能檢查此主機送出及收到的連線通訊。主機式防火牆解決了內部攻擊的問題，因為即使在同一個網路內的通訊，仍會經過主機式防火牆，藉由對

每個主機防火牆設定適當的規則，我們可以決定每台主機的通訊對象及通訊種類，藉此來達到防禦內部攻擊。主機式防火牆也解決了主機可能會到處移動的問題，不論主機在不在網路式防火牆所保護的內部網路中，主機式防火牆都可以隨時隨地保護安裝主機。

此外，因為主機式防火牆的使用對象可能是一般的使用者，而非網路式防火牆的網路管理人員，主機式防火牆發展了一種以應用程式為基礎的防火牆規則設定方法。網路式防火牆通常使用網路連線的角度來設定防火牆規則，依通訊協定種類、通訊主機、及通訊使用的通訊埠來制訂規則，網路管理人員藉由找到服務的通訊協定、通訊埠，及允許的通訊主機來達到通訊管理的目的。但是這些設定需要一定的背景知識，一般使用者可能只知道哪些應用程式想要可以通訊，而不清楚這些應用程式所用的協定、通訊埠等。而主機式防火牆直接安裝於主機中，藉由連線與程式關聯技術，可以知道連線是由哪個應用程式所發出。因此，可以直接設定哪些應用程式的連線可以通過防火牆。圖 2-1 是 Kerio Personal Firewall [34]以應用程式的角度來設定 Internet Explorer 瀏覽器可對外連線防火牆規則。利用此設定，只要連線是由 Internet Explorer 發出，或是傳送至 Internet Explorer 的連線，都可以通過個人式防火牆。

雖然主機式防火牆解決了內部攻擊及主機移動的問題，但是必須在所有想要保護的主機上安裝及設定防火牆。因此，在想要保護的主機一多時，如何管理眾多的主機式防火牆是一個問題。

分散式防火牆就像將多個主機式防火牆安裝於各台欲保護的電腦上，並加上一個中央管理的機制讓管理人員可以更新軟體及派送防火牆規則，使得管理動作可以在集中到一個中央管理節點進行，來對各個分散式防火牆進行規則設定。

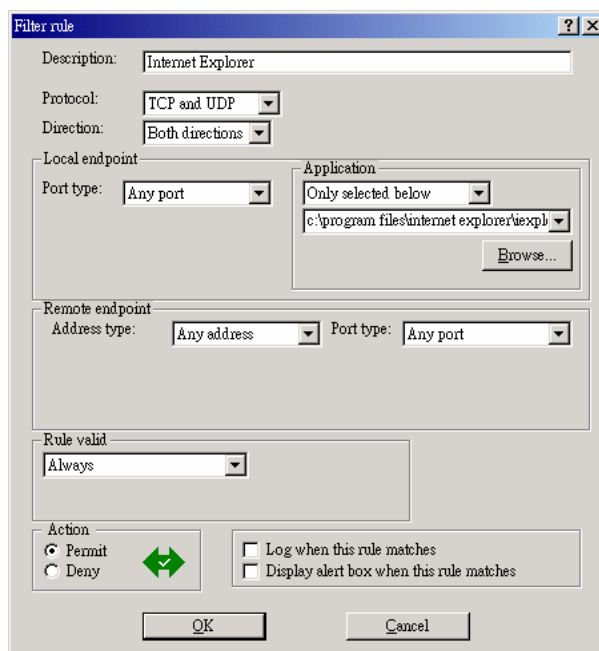


圖 2-1：以應用程式為基礎的規則設定方式

2.1.2 分散式防火牆

分散式防火牆[11,39]的概念[12,13]首先由 Steven M. Bellovin 於 1999 年所提出。Bellovin 認為傳統防火牆的功能是建立在網路拓樸及控制網路重要入口的基礎上，入口的一端被認為是可以被信賴的，另一端則是可能是敵人。但是由於網路應用的快速發展，這種假設已面臨挑戰：內部攻擊事件與日俱增，內部網路主機不一定是全部皆可信任。因此他提出一個由中央控制端制訂連線規則，而將實際阻擋連線的動作置於主機端的分散式解決方案。

Bellovin 所提出的分散式防火牆主要基於三個概念：政策語言（policy language）、系統管理工具、及 IPsec。基本概念是利用政策語言來描述各種連線允許或拒絕的規則，再使用系統管理工具將規則檔由中央控制端傳至所保護的各個主機上，實際判斷封包是否應該通過的責任則落在各個主機上。各個主機使用 IPsec 中的憑證來認定通訊主機的身份，再配合規則來判斷允許通過或阻擋該連線。因為使用了 IPsec 的憑證來確認身份，將比現在的防火牆使用 IP 位址來判斷通訊對方的身份更為安全，同時不再以網路拓樸為判斷敵我的基礎，主機不論

位於何處，這個身份都不會改變，因此可以隨時保護主機。

此外，分散式防火牆與主機式防火牆相同，將連線阻擋的執行置於安裝主機上進行，因此具有以下特性[12,13,14]：

- 避免內部攻擊

網路式防火牆並無法防止同在內部網路的兩台主機之間的攻擊，但分散式防火牆可分別對各台主機設定規則，可有效阻擋無信賴關係主機的內部攻擊行為。

- 效能及有用性的提高

網路式防火牆位在網路的重要入口上，若是防火牆發生故障會造成內部網段全部無法向外連接，同時防火牆的效能會直接影響內外通訊的品質。分散式防火牆各個主機只需處理自己有關的連線。一旦發生錯誤，也不會影響其他主機的正常運作。

- 可作到網路式防火牆做不到或是難以達到的功能

首先是分散式防火牆的執行動作，可以不再侷限於限制網路連線，而可以對應用程式進行設定。例如：設定某些主機的瀏覽器不可使用 javascript。分散式防火牆直接安裝於主機上，可取得機器的狀態，並藉由這些資訊來轉助判斷連線是否合法。例如：利用取得允許對外通訊的應用程式來找出 TCPACK 掃描封包，因為只有防火牆允許對外通訊的應用程式，才可能收到 ACK 封包，而網路式防火牆需要有 stateful 狀態檢查功能才能達到相同的目的。另外，分散式防火牆也可以藉由即時取得主機開啟的通訊埠來解決動態通訊埠開啟的問題。

最後是分散式防火牆可以隨時保護主機，不管受保護主機實體連接的位置為何，分散式防火牆仍只允許特定少數主機進行連結，能隨時保護主機。

而分散式防火牆與主機式防火牆最大的不同在於以下兩點：

1. 分散式防火牆以憑證來識別通訊對象，而主機式防火牆仍以 IP 位址為識別基礎。

2. 在分散式防火牆裡，中央制訂安全政策不可缺少的組成要件。而主機式防火牆各自制訂每台主機的規則。

雖然有上述兩點的不同，我們認為分散式防火牆其實是主機式防火牆的延伸。分散式防火牆利用中央管理的方式加強管理功能，並加入了憑證的概念來驗證規則的來源及通訊對象的身份。

2.1.3 防火牆面臨的挑戰

不論是網路式防火牆、主機式防火牆、或是分散式防火牆，由於部署位置的及運作架構的不同，所以適用的環境也會不同。以下針對目前防火牆所遭遇到的挑戰說明三種防火牆適用的情況。

1. 移動性 (mobility)

由於無線通訊及行動電腦的普及，電腦主機有可能在多個不同的網路上移動。目前的網路式防火牆並無法隨時隨地對主機進行保護，一旦主機移動到其他網路，便無法受到防火牆的保護。但是實際上該台主機所扮演的角色及用途並沒有改變，因此我們希望不論此主機的位置在何處，都能給予同樣的保護。網路式防火牆只能保護在防禦網路內的主機免於遭受外來攻擊，而主機防火牆或分散式防火牆本身就在主機上，因此可以隨時隨地保護主機。

2. 內部攻擊

由於防火牆只能對經過的通訊進行檢查，且需要一定要在通訊的重要通道上才能進行回應動作。不經過防火牆的通訊便無法受到防火牆的檢查，因而產生了內部攻擊問題。網路式防火牆是以網路為一個防護區域單位，只能藉由將網路分割成更小的網路，減少每個網路內的主機數目來儘量避免內部攻擊問題。而主機防火牆或分散式防火牆則是以單一主機為防護區域單位，所有連接或傳送至該主機的連線都會受到檢查，因此可避免內部攻擊問題。

3. 加密

現在的通訊為了安全的理由多會使用加密機制來保護資料，防火牆在判斷規

則時需要使用各層協定檔頭來與規則比對，但加密會使防火牆可用來與規則比對的資訊減少。例如使用 IPsec [5]的 transport mode 時，防火牆無法取得通訊的 TCP 檔頭資訊，所以無法得知使用的通訊埠。使用 tunneling mode 時，則視 security gateway 與防火牆的部署位置也會產生不同的問題[6]。網路式主機依部署位置不同，有可能會產生此類問題，目前一個常見的解決方法是將防火牆與安全閘道器結合為一，但是這個方法無法解決類似主機對主機使用 tunneling mode 時的加密問題。主機防火牆或分散式防火牆因為直接安裝在主機上，可在主機對通訊進行解密後取得解密的通訊資料來解決加密後判斷資訊無法取得的問題。

4. 防火牆設定複雜度

防火牆的政策規則設定好壞關係著防火牆的防禦能力，愈符合運作環境的政策規則設定可以減少開放不必要的通訊種類，提高防火牆的保護。目前政策規則的方式多以網路連線為基礎，藉由限制主機位址、協定種類、及通訊埠號等網路連線特性來制訂規則。另一種是使用應用程式為基礎來設定政策規則，藉由設定哪些應用程式的連線需要被限制或開放。網路式防火牆使用以網路連線為基礎的設定方式，而主機式防火牆與分散式防火牆可以同時使用兩種設定方式。以設定複雜度來說，以網路連線為基礎的設定方式需要具有較多的背景知識，而應用程式的設定方式則較為直覺。以網路連線為基礎的設定方式的另一個問題是：沒辦法確定實際使用該通訊埠的是何種服務，其他服務可藉由更改通訊埠來通過防火牆，雖然可以利用應用層防火牆（application-layer firewall）來加以解決，不過成本頗高。

5. 管理難易度

要用防火牆來保護具有數目相同主機的網路之管理困難度，我們定義為管理難易度。網路式防火牆只需要管理單點即可防禦整個網路，最為簡單，而主機式防火牆需要管理每一台受保護主機上的防火牆。分散式防火牆的管理難易度則介於兩者之間，因為它提供一個中央管理的機制，但仍必須針對各別主機設定規則。

6. 效能要求程度

效能是指防火牆比對來往通訊與設定的政策規則來決定阻擋或放行通訊的速度。隨著網路的速率愈來愈快，防火牆的效能一直是被受人注意的重點之一。以網路式防火牆來說，因為位居網路的重要通道，若是效能太差可能會影響兩個網路之間的整體傳輸速率。目前其中的一種解決方式是使用特殊應用積體電路（Application-specific Integrated Circuit, ASIC），將防火牆常用的功能以硬體取代以往的軟體處理來加快處理的速度。但是隨著超高速網路的產生及防火牆的規則及功能日漸複雜，這個方式還是有其極限。對於主機式及分散式防火牆來說，它們只需要處理單一主機的來往通訊，效能要求要比網路式防火牆來得寬鬆。

7. 錯誤影響程度

錯誤影響程度，是指系統發生錯誤時所造成的影響程度。網路式防火牆採用單點架構，一旦此點發生錯誤時，防火牆的防禦機能會完全失效，可能還會造成內外網路無法互相通訊。現有的解決方式多是利用備援的概念，利用使用多台的防火牆來提高降低錯誤影響程度。在防火牆發生錯誤時，備份防火牆能接手原有防火牆的功能。但是如何偵測錯誤、何時切換防火牆、及如何還原至防火牆發生錯誤前一刻的運作環境，每一個都需要複雜的方法來達成。而主機式防火牆與分散式防火牆將通訊檢查及規則比對的動作交由各台主機進行。因此，萬一有主機發生錯誤，也不會影響其他防火牆的防禦功能，整體來說，防火牆的功能還是能正常運作，錯誤影響程度也就較網路式防火牆來得小。

8. 動態行為協定的處理

目前許多的協定，如：FTP 協定或是其他多媒體協定，都會動態開啟具有隨機埠號的通訊埠[15]。目前的網路式防火牆多以解析應用層協定來解決此問題，但此方法需要耗費較多的時間在協定的解析上。而主機式及分散式防火牆可以利用判斷連線是哪個應用程式所發出，或是即將傳送至哪個應用程式來決定是否允許通行，此方式不用解析協定內容，運作成本較低。

表 2-1：三種防火牆綜合比較表

| | 網路式防火牆 | 主機式防火牆 | 分散式防火牆 |
|-----------|--------|-----------|--------|
| 部署位置 | 網路之間 | 受保護主機上 | |
| 規則設定基礎 | 網路連線 | 網路連線、應用程式 | |
| 內部攻擊 | 差 | 佳 | 佳 |
| 移動性 | 中 | 佳 | 佳 |
| 加密 | 中 | 佳 | 佳 |
| 動態行為協定的處理 | 差 | 佳 | 佳 |
| 效能要求程度 | 高 | 中 | 中 |
| 錯誤影響程度 | 高 | 中 | 中 |
| 設定複雜度 | 中 | 佳 | 佳 |
| 管理難易度 | 佳 | 差 | 中 |

(資料來源：本研究整理)

第二節 縱深防禦與合作防禦

縱深防禦 (defense-in-depth) [4,16] 的概念是結合多種不同的安全防禦機制互相彌補個別安全機制的不足來加強防禦的強度。另外，縱深防禦也假設任何個別的防禦機制都可能會失效，此時仍然可以有其他的防禦機制來進行防禦。

圖 2-2 是縱深防禦的概念示意圖，此架構共有防火牆、網路式入侵偵測系統、主機式入侵偵測系統及陷阱系統四道防線。其中，A 類攻擊會用防火牆直接阻擋。B 類及 C 類攻擊可以通過防火牆，但會被網路式入侵偵測系統所偵測，然後加以阻擋或導入陷阱系統。D 類攻擊可通過防火牆及網路式入侵偵測系統的阻擋及偵測，但會被主機式入侵偵測系統所偵測。只有 E 類的正常連線才可以通過各道防線存取網路資訊系統。四道防線結合可以防禦較多的攻擊種類，而即使有其中一道機制失效，例如：防火牆失去防禦功能，A 類攻擊仍可能會被兩種入侵偵測系統所偵測到。合作防禦 (cooperative defense) 則強調各防禦機制間藉由交換資訊共同對某一攻擊進行防禦，兩個防禦機制可以是相同機制或是不同機制。舉例來說，多個防火牆可進行合作防禦來防治分散式阻斷服務攻擊。另外，在圖 2-2 中，入侵偵測系統發現 C 類與 D 類攻擊時，入侵偵測系

統必須與防火牆或路由器溝通，將連線導入陷阱系統中觀察，也是另一種合作防禦。

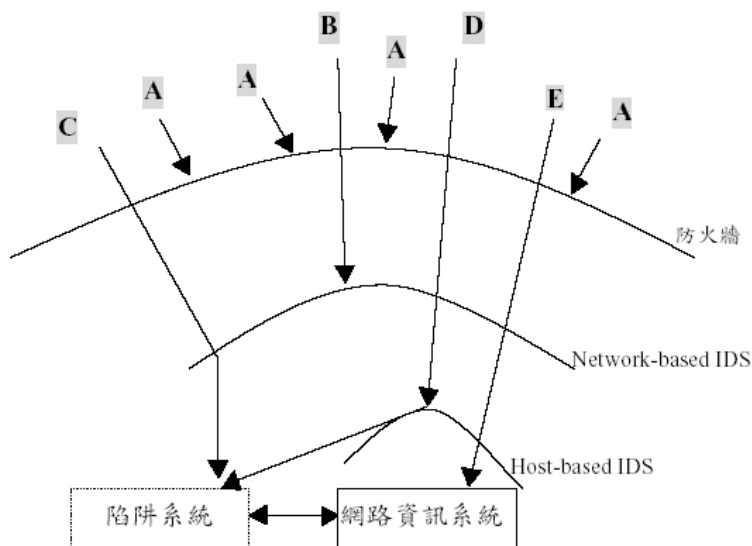


圖 2-2：縱深防禦及合作防禦概念示意圖

(資料來源：[4])

縱深防禦與合作防禦最大的不同是縱深防禦強調防禦機制的多元性，但不一定需要互相交換資訊，而合作防禦則強調防禦機制間的溝通與合作。

第三節 入侵預防

入侵預防 (intrusion prevention) 字面意思是事先利用各種方法使攻擊失效，讓攻擊者無法入侵成功。RCL & Associates 的總裁 RRobert Lonadier，對入侵預防的廣泛定義[17]為：「任何可以防止沒有經過授權的網路存取之措施及機制」。Snort 入侵偵測系統 [36] 的發展者 Martin Roesch 則指出「入侵預防是存取控制，而入侵偵測是監視」[17]。

入侵預防系統 (Intrusion Prevention System, IPS) 泛指能用來阻擋攻擊的系統，例如防火牆即是其中一種入侵預防系統。入侵預防的概念與入侵偵測結合，衍生出了入侵偵測及預防 (Intrusion Detection and Prevention, IDP)，藉由在入

入侵偵測系統偵測到攻擊行為之後，對攻擊採取防禦措施來阻止攻擊。業界採用較小的定義[17]來定義入侵預防系統，認為入侵偵測與預防系統就是入侵預防系統，也是下一代的入侵偵測系統，能「在偵測到攻擊行為後，自動採取回應動作來阻止或減小攻擊行為所造成的傷害」。

因此，我們將入侵預防分為兩類：第一類的入侵預防強調利用存取控制減少被入侵的可能。另一類的入侵預防則藉由找出攻擊行為加以中斷來防止入侵。防火牆利用預先設定的政策規則對連線來源、目的、及連線種類進行限制，屬於第一類的入侵預防。而利用入侵偵測系統抓出攻擊，再利用主動回應動作或是與防火牆進行合作防禦來阻擋攻擊，則屬於第二類的入侵預防。

以往的入侵偵測系統強調如何找出攻擊行為，而入侵偵測及預防系統則同時強調找出攻擊行為及中斷攻擊行為。在第一章中的攻擊趨勢中曾提及攻擊工具及手法的自動化，使用整個攻擊流程的進行時間縮短，現在的入侵偵測系統多是提供攻擊警訊告知管理人員。但在攻擊自動化後，這樣的回應動作已無法阻擋攻擊，整個攻擊可能在管理人員處理之前就已經結束。入侵偵測及預防系統藉由強化入侵偵測系統的回應動作，在偵測到攻擊行為後，自動做出回應動作來阻止攻擊行為繼續進行。

目前入侵測系統的回應動作通常分為主動式回應與被動式回應兩類[18]。

1. 被動式回應

入侵偵測系統在偵測到攻擊後以各種方式告知負責人員。常用的方式有螢幕訊息、電子郵件、呼叫器或手機簡訊、及警示報告。入侵系統本身不介入攻擊防禦動作，而由管理人員自己處理。目前大多數的入侵防火牆都以被動式回應做為偵測到攻擊行為的主要回應動作。

2. 主動式回應

入侵偵測系統在偵測到攻擊後會採取修正的動作來防止攻擊繼續發生。常見的有劫持連線 (session hijacking)、中斷連線 (session termination)、重新設定防火牆組態、重新設定路由器及交換器、欺騙技術 (Deception

technique)、修正錯誤設定。部份主動式回應需要與其他網路設備或安全機制進行合作防禦才能達成。

以目前自由軟體中最著名的網路式入侵偵測系統 Snort 為例，除了提供被動式回應的攻擊警訊記錄外，還可以使用發送 TCP Reset 方式及 ICMP Unreachable 封包來中斷連線[19]。但是這兩種方式實際上是使用假造封包來達到中斷連線的目的，不一定每次都會成功，且無法適用於會對通訊對象認證的未來 IPv6 應用環境。但由於入侵偵測系統無法控制連線通行與否，因此只能採用這些方式來做主動回應。

由於入侵偵測系統部署的位置不一定在網路的要道上，所以無法直接操控網路連線的通過與阻擋。snort-inline [20]將入侵偵測系統與防火牆結合，加強了入侵偵測系統的主動式回應動作，能使入侵偵測系統在偵測攻擊後，能使用防火牆來阻擋攻擊，而成為更有用的入侵偵測及預防系統。我們則認為入侵偵測及預防系統本質上是在防火牆中加上入侵偵測的功能。它延用網路式防火牆的架構，掌握網路重要入口，偵測來往通訊是否含有攻擊行為來決定放行與否，原有防火牆規則比較則換成了入侵偵測規則。

入侵偵測系統除了使用主動式回應來達到入侵預防目的之外，另一個可能的方式是與其他安全機制進行合作防禦，例如將攻擊資訊傳給防火牆來阻擋攻擊連線或是將攻擊連線導入至陷阱系統。

第四節 本章小結

不論是網路式、主機式或分散式防火牆都面臨不同的問題。網路式防火牆與主機式防火牆目前已發展得相當成熟。但我們認為分散式防火牆能解決較多關鍵性的問題，但是必須再簡化管理動作，才能使分散式防火牆普遍地被使用。

因為攻擊手法日新月異，已經沒有一個單一機制能夠處理所有安全威脅，所以縱深防禦與合作防禦是未來的趨勢。藉由結合多種各種安全防禦機制，一方面

加強防禦程度，另一方面也可防止單一防禦發生錯誤。安全防禦的最終目的不只要抓到攻擊行為，而是更進一步地採取主動式回應來阻止攻擊行為成功入侵系統。

第三章 合作防禦方式與困難點分析

本章主要在探討可與分散式防火牆進行合作防禦的安全防禦機制、合作方式及困難處，最終目的是要達成入侵預防，在入侵者攻擊成功前阻擋其發生。本章前三節針對合作防禦的對象進行討論，共提出防火牆與防火牆、入侵偵測系統與防火牆、及漏洞掃描系統與防火牆之間三種合作防禦方式及問題點探討。第四節提出用來解決合作防禦的通用規則。第五節提出一個方法解決網蟲向外擴散時所造成的網路癱瘓問題，目的是讓合作防禦機制得以在網蟲大量感染時仍可以保持正常運作。

第一節 防火牆之間進行合作防禦

我們認為網路式防火牆和分散式防火牆是未來最重要的兩類防火牆，它們之間可以進行以下的合作防禦。基本上，分散式防火牆與其他防火牆之間的合作防禦主要利用互相傳遞政策規則來達成。

- 分散式防火牆與網路式防火牆

如圖 3-1 所示，分散式防火牆可以與網路式防火牆互相交換政策規則。網路式防火牆能在規則過多時，將部份規則交由網路式防火牆來負責。或是分散式防火牆在使用動態開啟通訊埠的協定時，將相關規則傳送給網路式防火牆，使得該連線能正常通過網路式防火牆。使用完畢時再使用相同的流程，移除此規則。但是分散式防火牆與網路式防火牆之間只能使用以網路連線為基礎的政策規則，因為網路式防火牆只能使用此種規則。

- 分散式防火牆之間

各主機上的分散式防火牆之間也可以進行合作防禦。例如：某一台主機上的防火牆可以統計哪些政策規則阻擋最多的連線，我們可以假設這些政策規則可能阻擋了最多的攻擊或是不必要的連線，所以將這些規則傳送給其他分散式防火牆，使其他主機也能防禦相同的攻擊。

不過各分散式防火牆之間進行合作防禦的最大問題是各主機的運作環境不同，現有的防火牆政策規則無法直接套用在各個主機上。最明顯的例子是同樣的服務可能使用不同的通訊埠號，此時若是直接套用固定通訊埠號的政策規則，反而會造成套用不適當的規則。

解決方式是使用較抽象化的政策規則語言，避免描述會隨每台主機不同的運作細節，而這些資訊再由主機的實際狀況決定。FireHOL 計劃[21] 實現了通訊埠的抽象化，藉由只描述服務名稱，再由主機設定來取得實際的通訊埠號。另一個方式是以應用程式為基礎來設定政策規則，再判斷系統上的應用程式環境是否符合。

此外，我們認為還需要加入對系統平台資訊的描述，因為分散式防火牆可能運作於不同平台的主機上，加入系統平台描述，可以使規則能更精確地套用於合適的分散式防火牆主機上。

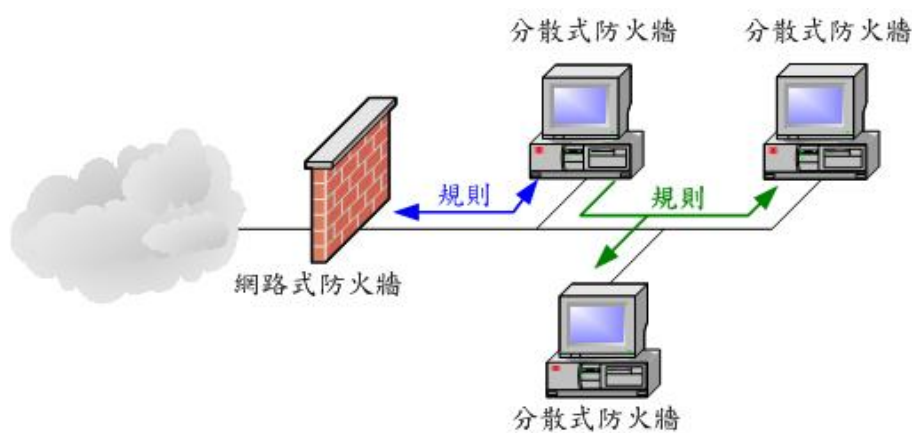


圖 3-1：防火牆之間進行合作防禦示意圖

第二節 防火牆與入侵偵測系統進行合作防禦

入侵偵測系統與防火牆進行合作防禦的最終目的是在入侵偵測系統發現攻擊時，能及時用防火牆來阻擋攻擊。而可能的實作方式有二種，第一種方式是將入侵偵測系統及防火牆分為兩個不同的系統，在發現攻擊時，入侵偵測系統與防火牆溝通，利用防火牆來阻擋攻擊。第二種是將入侵偵測系統與防火牆合而為

一，也就是入侵偵測及防禦系統。

以第一種方式而言，優點是可以使用多個不同的入侵偵測系統來增加偵測的攻擊種類，但是面臨的最大問題是偵測到攻擊後，可能無法及時利用防火牆來阻擋攻擊。而第二種方式最大的優點是被判斷為攻擊連線可以完全被阻擋，但因為入侵偵測及防禦系統本質上是防火牆，也會面臨前述的效能要求高及發生錯誤時影響程度大等的防火牆問題。此外，目前最普遍的入侵偵測方式仍使用誤用偵測（misuse detection）方式。因此，來往的通訊可能需要比對眾多的攻擊特徵才能確定此通訊不是攻擊行為，比對的動作會比原本的防火牆更多，對效能的影響會比防火牆來得更嚴重。

最後，入侵偵測的偵測率與正確性是決定入侵偵測系統與防火牆合作防禦的關鍵。過高的誤報率（false positive rate）會使防火牆阻擋過多的正常連線，過高的漏報率（false negative rate）會減低合作防禦的效用，許多攻擊將會無法進行防禦被偵測及防禦。

第三節 防火牆與漏洞掃描系統進行合作防禦

漏洞掃描系統，如：Nessus [22]、SARA[23]、ISS Internet Scanner [24]，最主要的功能是用來偵測一個系統存在的漏洞或是錯誤設定。因為大多數的攻擊都是針對某個漏洞或是錯誤設定而來，漏洞掃描系統的目的就是找出這些可能遭受攻擊的漏洞或設定，再利用修正這些漏洞及錯誤設定來避免攻擊成功達成入侵的目的。目前在做完漏洞掃描動作之後，仍沒有一個好的機制能在修正漏洞前，防止這些漏洞遭受攻擊。這是因為以往的網路式防火牆只能以網路連線為基礎來阻擋連線。而使用分散式防火牆之後，能藉由阻擋漏洞應用程式的所有連線來防止此漏洞遭受攻擊。加上漏洞修正判斷機制後，甚至能在漏洞或是錯誤修正之後，自動恢復程式的正常通訊。

圖 3-2 說明了分散式防火牆如何與兩種不同的漏洞掃描系統進行合作防

禦。與網路式漏洞掃描系統合作時，漏洞掃描系統結束漏洞掃描動作時，可產生漏洞應用程式列表，再轉換成漏洞規則傳送至分散式防火牆中，分散式防火牆就可以對漏洞應用程式連線進行阻擋，達到防禦攻擊的目的。使用主機式漏洞掃描系統時，則可以發送漏洞描述給分散式防火牆，防火牆再觸發主機式漏洞掃描系統檢查該漏洞，如果該漏洞存在，則阻擋該漏洞應用程式連線。主機式漏洞掃描系統也可以作為分散式防火牆的漏洞修正判斷機制，防火牆可定時呼叫主機式漏洞掃描系統檢查之前有漏洞或含有錯誤設定的應用程式是否已做修正，來決定是否移除連線阻擋規則。如此一來，我們可以得到一個自動化的漏洞防禦機制，能在應用程式有漏洞時阻擋其連線，防止被攻擊，而在漏洞應用程式進行修正後，恢復其連線通訊。

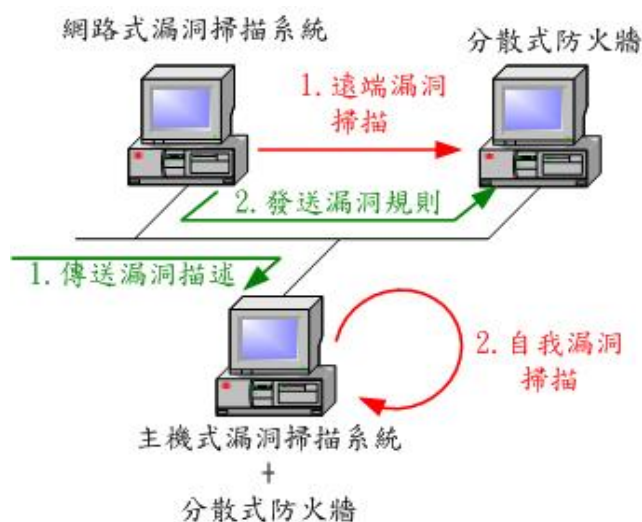


圖 3-2：分散式防火牆與漏洞掃描系統合作防禦示意圖

此合作防禦的重點在需要一個能夠描述漏洞應用程式與執行回應動作所需資訊的規則，供防火牆執行回應動作及主機式漏洞掃描系統檢查漏洞。例如網路式漏洞掃描系統發送給分散式防火牆的漏洞規則，至少需要包含漏洞編號來供主機式漏洞掃描系統將來再次檢查此漏洞、及漏洞應用程式的相關資訊來讓分散式防火牆進行阻擋動作。

第四節 通用規則

通用規則利用描述系統平台及應用程式資訊使規則可以套用到合適的系統中，也解決分散式防火牆的管理困難問題。通用規則中並保留了對網路連線的描述，用來與網路式防火牆整合。本節提出通用規則的概念、設計及運作方法。

3.4.1 採用通用規則的原因及目的

通用規則主要的目的有以下兩點：

1. 幫助達成合作防禦

如果規則中描述會隨主機而不同的運作細節，便無法合適地套用到各台主機上。即使各個防火牆節點能夠互相交換規則，也無法達成合作防禦，因為每台主機的運作環境都不相同。舉例來說，用來阻擋 Web Server 通訊的 port 80 規則不一定適用於所有主機，因為不是所有 Web Server 都是用 port 80 來通訊。因此，規則必須要能夠忽略運作細節，以較高階的資訊來描述目標，再由主機端由高階資訊來取得運作細節。

2. 簡化分散式防火牆的管理

以往分散式防火牆雖然可由中央管理介面制訂規則，再將規則派送至各個分散式防火牆中套用，但是在制訂規則時還是需要瞭解各台主機的運作細節，或是決定規則在哪些主機上要套用。當要為 10 台主機制訂規則時，必須對 10 台主機的運作環境加以瞭解及制訂規則。因此，以往的中央管理介面的遠端規則制訂及派送功能只解決了制訂規則的「地點」問題，將制訂規則的地點集中至中央管理主機，但是實際的操作動作與到每台主機上制訂規則無異。

本研究希望藉由使用通用規則，為每種應用程式都建立規則範本，將來我們只需要知道每台主機上有哪些應用程式，就可以取出各應用程式的規則範本來建立每台主機的防火牆規則。另一方面藉由對系統運作環境的描述，主機可利用判

斷自身的運作環境是否與規則描述相符來決定是否套用該規則。如此一來，管理者也不用自行判斷規則套用的主機，只需要將規則發給所有主機，主機會以運作環境異同來決定套用與否。

3.4.2 通用規則概念

通用規則的基本概念為：「避免描述隨主機不同的運作細節，而改以描述較高階的資訊，再利用高階資訊由主機取得運作細節，同時加上規則的運作環境，在主機端判斷該規則是否適用於該主機。」以網路連線的允許通訊設定為例，通用規則會利用描述什麼應用程式的連線可以被允許對外通訊，來取代直接描述允許的通訊埠號，因為通訊埠號會隨主機不同而改變。防火牆節點由規則中所描述的資訊來判斷來找出符合的應用程式，再對該應用程式進行連線阻擋動作。通用規則概念如圖 3-3 所示。

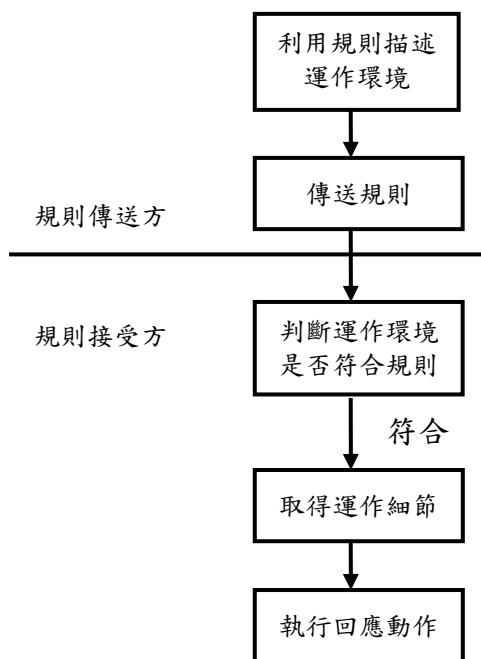


圖 3-3：通用規則概念示意圖

3.4.3 相關作法回顧

目前已有許多作法與解決合作防禦的溝通及多台防火牆的管理問題相關，以下對現有相關做一解說，並說明為什麼要本研究仍要提出通用規則的原因。

1. High Level Firewall Language (HLFL) [42]

HLFL 提供了一個較高階的防火牆語言，目的是用來解決不同種類防火牆管理困難的問題。藉由使用 HLFL 來描述防火牆規則，可以將同一個規則轉換成不同實際的防火牆規則，例如同一個 HLFL 規則可以轉換成可在 ipfw、Cisco ACL、IPFilter、IPFWadm、IPChains、NetFilter/IPTables 實際運作的規則。對於各種防火牆具有的特殊功能，也可以藉由 HLFL 提供的條件命令註解來加以描述。

2. FireHOL [21]

FireHOL 也是一種高階的防火牆語言，但只能轉換成 iptables 規則，它的主要目的是增加規則的易讀性及易學性，使得規則的設定變得更容易及更直覺。同時，他可以視服務被設定為客戶端(client)或是伺服器端(server)來決定這個服務實際的通訊埠為何。就某種程度來說，它已經具有一定的抽象化概念，能避免描述會隨主機不同運作細節。

3. VulXML [43]

VulXML 提出了一個 XML 格式的漏洞描述語言，目的是用來描述漏洞與安全通報。利用 VulXML 來描述的漏洞可以直接被漏洞掃描系統直接用來檢查漏洞，或是其他工具來取得此漏洞的相關資訊。但目前 VulXML 只能描述 Web 應用程式的漏洞，藉由真正建立連線，送出檢查要求，並視對應的回應訊息來找出漏洞。

4. Open Vulnerability Assessment Language (OVAL) [44]

OVAL 提供一種類似 SQL 語法的漏洞描述語言，可用來描述漏洞資訊，漏洞掃描系統。與 VulXML 最大的不同是檢查的漏洞不限於 Web 應用程式，而

可以是 Windows NT、Windows 2000、Solaris 作業系統上的任一漏洞。OVAL 利用類似主機式漏洞掃描系統中檢查應用程式狀態、應用程式的執行身份等動作來檢查漏洞是否存在。

5. Application Vulnerability Description Language (AVDL) [45]

AVDL 的目的是使一致的 XML 描述來描述相同的漏洞，這個漏洞描述並能成為各種防禦機制溝通的媒介。例如同一個漏洞描述可供漏洞掃描系統找出漏洞、供防火牆阻擋漏洞應用程式、供事件管理系統產生事件報告、及漏洞修正系統修正漏洞。但是目前此語言還在草擬階段，並未提出真正範例。

根據以上現有的相關研究，AVDL 比較接近本研究的需要，因為它提供了各安全機制整合之間的溝通語言，但是它並未提出實例，我們無法得知是否真正足夠描述主機的運作環境來支援多個防火牆的管理。HLFL 及 FireHOL 雖然解決防火牆的管理問題，使得防火牆之間的管理及規則的制訂變得更容易，但是仍然少了對運作環境描述，無法讓防火牆自動判斷規則是否應該套用，但 FireHOL 已將避免直接描述通訊埠號，而以服務名稱代替再取得實際通訊埠的概念符合部份通用規則的目的。VulXML 及 OVAL 提供了二種漏洞描述語言，但它們仍偏重於供漏洞掃描系統找出漏洞，而事實上，檢查漏洞不是合作防禦的重點，我們只需要知道漏洞的編號，就可以利用漏洞掃描系統檢查漏洞，我們需要的是漏洞掃描系統在找到漏洞後，能傳出什麼資訊供防火牆來阻擋漏洞攻擊，而不在乎漏洞的檢查方法。基於以上原因，本研究仍提出自己的通用規則，來解決分散式防火牆的管理與合作防禦的溝通問題。

3.4.4 通用規則語法

本小節對通用規則的語法做一解說。因為 XML 具有結構性及彈性等特性，使得 XML 已成為目前常用的資訊交換媒介，所以本研究提出的通用規則

以 XML 格式來表示防火牆規則。以下針對通用規則內容的各個部份加以介紹。

- Rule 元素

圖 3-4 是一個完整的通用規則，整個規則可分為 References、RuleValid、及 RuleContent 三大組成元素 (element)。屬於整個規則 (Rule) 的屬性有規則名稱 (Title)、此規則使用的規則語言版本 (SchemaVersion)、規則版本 (RuleVersion)、規則釋出日期 (DateReleased)、及規則最後更新日期 (DateRevised)。規則名稱是此規則的標題，記錄了此規則的描述內容。規則語言版本則是解析此規則的方式，不同版本的規則語言，會有不同的關鍵字，也具有不同的功能。規則版本則是此規則的釋出版本號碼，用來識別規則的新舊。規則釋出日期及規則最後更新日期代表此規則第一次提出和最後一次修改的日期。

```
- <Rule Title="" SchemaVersion="1.0" RuleVersion="1.0" DateReleased="2003/06/01" DateRevised="2003/06/03">
+ <References>
+ <RuleValid>
- <RuleContent Action="BLOCK">
+ <Platforms>
+ <Applications>
+ <Connections>
</RuleContent>
</Rule>
```

圖 3-4：通用規則組成示意圖

- References 元素

References 元素描述了此規則的參考資料。可包含一個以上的安全通報 (Advisory) 元素。每個安全通報元素具有發行組織及通報編號 (VulID) 兩個屬性。漏洞掃描系統藉由取得此元素來得知漏洞編號。

```
- <References>
  <Advisory organization="CERT" VulID="CA-2002-12" />
</References>
```

圖 3-5：References 元素組成示意圖

- RuleValid 元素

為了避免在防火牆上累積太多的規則，我們設計了一個丟棄不適用規則的方法。RuleValid 元素則描述了合作式防火牆該如何判斷此規則是否保留，可以有一個或多個 Attribution 元素。以圖 3-6 為例，規則告訴合作式防火牆需比對系統與規則內的 OSName 屬性值。若是符合則保留此規則，不符合則丟棄此規則。這個設計主要是要讓規則制訂者能夠自行決定規則套用的環境。不過若是設定太寬鬆，系統內會保留過多無用的規則。若是設定太嚴格，系統環境一經變化，此規則便會失效。

```
- <RuleValid>  
  <Attribution Name="OSName" />  
</RuleValid>
```

圖 3-6：RuleValid 元素組成示意圖

- RuleContent 元素

RuleContent 元素是整個規則的重點，用來描述此規則適合的「運作環境」及「回應動作」。如圖 3-7 所示，運作環境描述分為系統平台（Platforms）、應用程式（Applications）、網路連線（Connections）三大類。回應動作（Action）為 RuleContent 元素的屬性，分為應用程式類及網路連線類回應動作。Rule 元素中可以包含多個 Platforms、Applications、及 Connections 元素，但是最少需要有 Platforms、Applications、Connections 三個元素中的其中一個。

```
- <RuleContent Action="BLOCK">  
  - <Platforms>  
    + <Platform OSName="Microsoft Windows 2000">  
  </Platforms>  
  - <Applications>  
    + <Application APName="Microsoft Internet Explorer" APTYPE="CLIENT" APPROCESSNAME="IEXPLORE.EXE">  
  </Applications>  
  - <Connections>  
    + <Connection Direction="BOTH">  
  </Connections>  
</RuleContent>
```

圖 3-7：RuleContent 元素組成示意圖

- Action 屬性

Action 屬性描述了規則條件符合時的執行動作。合作式防火牆因為安裝於受保護主機上，可能回應的動作除了網路連線類回應動作外，還可以直接對應用程式的執行進行阻擋及中止。因此，回應動作分為網路連線及應用程式兩類。網路連線類回應動作與現有網路式防火牆動作類似，如表 3-1，有允許連線通過 (ALLOW)、拒絕網路連線 (DENY)、對網路連線進行速度限制 (RATE-LIMIT)。應用程式類回應動作則有阻止及中止程式執行 (BLOCK)。此動作會在新程式開始執行時，依據規則判斷該程式是否可執行，若為限制的程式則停止其執行動作，防止使用者執行危險程式。另外，規則套用同時也會對現有執行程式列表進行掃描比對，並停止執行中的限制程式，避免已經在執行的程式可逃避該類規則。

表 3-1：通用規則回應動作列表

| 回應動作名稱 | 說明 |
|------------|-----------|
| ALLOW | 允許網路連線 |
| DENY | 阻擋網路連線 |
| RATE-LIMIT | 限制連線速度 |
| BLOCK | 阻止及停止程式執行 |

- Platforms 元素

Platforms 元素用來描述此規則適用的軟硬體平台資訊。如圖 3-8 所示，系統平台描述資訊可由一個以上的系統平台 (Platform) 元素所組成。系統平台元素中的 OSVersion、OSLanguage、OSPatch、及 Architecture 元素可以有 multiple，也可以完全沒有。各元素及屬性意義與格式範例整理如表 3-2。

```

- <Platforms>
- <Platform OSName="Microsoft Windows 2000">
  <OSVersion>5.00.2195</OSVersion>
  <OSLanguage>zh-tw</OSLanguage>
  <OSPatch>Service Pack 3</OSPatch>
  <Architecture>AT/AT COMPATIBLE</Architecture>
</Platform>
</Platforms>

```

圖 3-8：系統平台描述資訊組成示意圖

表 3-2：系統平台描述資訊各元素及屬性之意義與範例

| 名稱 | 說明 | 格式範例 |
|--------------|----------|--|
| OSName | 作業系統名稱 | Microsoft Windows 2000 FreeBSD Linux |
| OSVersion | 作業系統平台版本 | 5.00.2195 4.7-Release 2.4.19 |
| OSLanguage | 作業系統語系 | zh-tw zh_TW.Big5 |
| OSPatch | 作業系統修正程式 | Service Pack3 |
| Architecture | 硬體架構種類 | AT/AT COMPATIBLE i386 |

- Applications 元素

Applications 元素記錄了此規則所適合的應用程式種類。如圖 3-9 所示，Applications 元素包含一個以上的應用程式（Application）元素所組成。應用程式元素內的各元素也可以有多個，也可以完全沒有。各元素及屬性意義與格式範例整理如表 3-3。

表 3-3：應用程式描述資訊各元素及屬性之意義與範例

| 名稱 | 說明 | 格式範例 |
|----------------|-----------|----------------------------------|
| APName | 應用程式名稱 | Microsoft Internet Explorer |
| APType | 應用程式種類 | CLIENT |
| APProcessName | 應用程式執行緒名稱 | IEXPLORE.EXE |
| APProcessOwner | 應用程式執行者身份 | 16 ,root |
| APPatch | 安裝修正程式名稱 | SP1, Q324929 |
| APLanguage | 應用程式語系 | zh-tw |
| APVersion | 應用程式版本 | 6.0.2800.1106 |
| APFile | 應用程式檔案 | C:\Program Files\Internet |
| FileName | 應用程式路徑 | Explorer\IEXPLORE.EXE |
| FileChecksum | 雜湊值 | d2c45c6c41585a50f43938b266d700f9 |
| HashMethod | 使用的雜湊函式 | MD5 |

```

- <Applications>
  - <Application APName="Microsoft Internet Explorer" APTYPE="CLIENT" APPROCESSNAME="IEXPLORE.EXE">
    <APPROCESSOWNER>114</APPROCESSOWNER>
    <APPATCH>SP1</APPATCH>
    <APLANGUAGE>zh-tw</APLANGUAGE>
    <APVERSION>6.0.2800.1106</APVERSION>
    <APFILE FileName="C:\Program Files\Internet Explorer\IEXPLORE.EXE"
      FileChecksum="d2c45c6c41585a50f43938b266d700f9" HashMethod="MD5" />
    </Application>
  </Applications>
    
```

圖 3-9：應用程式描述資訊組成示意圖

● Connections 元素

Connections 元素是網路連線類描述資訊，用來描述網路連線。Connections 元素可由一個以上的網路連線（Connection）元素所組成。網路連線元素內的各元素也可以有多個。各元素及屬性意義與格式範例整理如表 3-4。

表 3-4：網路連線描述資訊各元素及屬性之意義與範例

| 名稱 | 說明 | 格式範例 |
|-----------|-------------|--------------------------|
| Direction | 連線方向 | Incoming, Outgoing, Both |
| Protocol | 通訊協定種類 | TCP, UDP, ICMP,... |
| SrcIP | 來源/目的 IP 位址 | 10.0.0.1/255.0.0.0 |
| DstIP | | 10.2.2.2 |
| SrcPort | 來源/目的通訊埠號 | 21, 80, 1000-1024, |
| DstPort | | |

```

- <Connections>
  - <Connection Direction="BOTH">
    <Protocol>TCP</Protocol>
    - <SrcIP>
      <IP Address="10.2.2.2/255.255.0.0" />
    </SrcIP>
    - <DstIP>
      <IP Address="10.1.1.1" />
    </DstIP>
    - <SrcPort>
      <Port Number="1024-" />
    </SrcPort>
    - <DstPort>
      <Port Number="80" />
    </DstPort>
  </Connection>
</Connections>
    
```

圖 3-10：網路連線描述資訊組成示意圖

3.4.5 通用規則的套用方式

本小節說明防火牆如何處理通用規則。防火牆在接到通用規則後，所進行的流程如圖 3-11 所示。

本研究將防火牆接收到的規則分為三種類型。第一類是接收到的規則永遠不可能在該主機上適用，最常出現的狀況是作業系統不同，因為作業系統不同時，通常適合的規則也會完全不同。第二類是現在環境不符合，但是以後可能符合，例如某應用程式在接收規則時並未安裝，但以後可能會被安裝，所以可能就需要相關的規則。第三類是系統環境完全符合。第一種類型的規則不會被保留，因為它們不可能適用於該系統。第二類規則應加以保留並定時重新檢查。如此一來，系統環境改變時，規則也能同時改變來加以保護。第三類規則會被直接套用，並定時檢查是否仍然適合套用。

因此，防火牆接到規則時，首先會判斷此規則是否有可能適用於目前的系統，對於不可能適用的規則，即進行丟棄。可能適用的規則會被存放於規則存放區，防火牆並會定時取出規則存放區的所有規則重新檢查是否符合目前的系統環境，如果符合即會進行規則套用。這個動作可以達到因應系統環境變化動態更改套用的規則。例如：當新的應用程式被安裝後，符合該應用程式的規則就會被套用。接下來，防火牆節點會視規則內回應動作的種類，將規則複製至網路連線類規則存放區或應用程式類規則存放區。真正負責執行回應動作的應用程式回應模組及網路連線回應模組則會取得各自存取區的規則加以套用。

那麼該如何判斷哪些規則可能適用於目前的系統及符合目前的運作環境呢？防火牆節點接到通用規則時，首先會將規則內容依元素種類、名稱建成如圖 3-12 的決策樹。

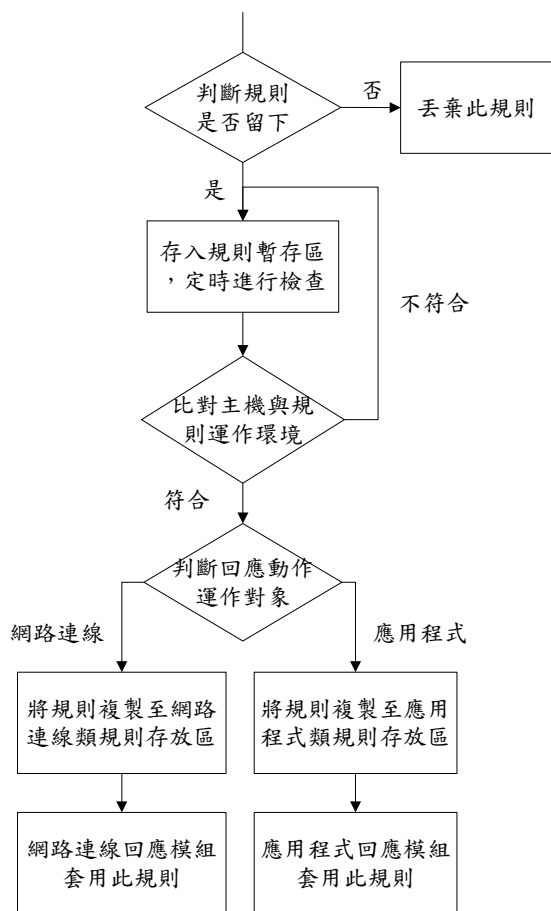


圖 3-11：合作式防火牆通用規則套用流程圖

判斷規則是否留下時，首先會取出所有 RuleValid 元素裡的 Attribution 元素 Name 屬性值。以圖 3-13 的通用規則為例，規則中的 RuleValid 元素描述應判斷主機的 OSName 是否符合規則內的任一 OSName 值來決定此規則是否被留下或是被丟棄。於是防火牆取出所有 OSName 的值，建立如圖 3-14 的決策樹，規則內沒有描述到的元素或屬性，表示不用加以比較，決策樹中對應的部份結果會為真代替。假設安裝主機的主機的作業系統是 Windows 2000，圖 3-14 決策樹最後的結果會為真，因此圖 3-13 的規則會被留下。

而在判斷運作環境是否相同時，基本流程與判斷是否保留規則時相同，但由規則取出較多的元素及屬性值加以比較。取出的部份有系統平台元素中的 OSName、OSVersion、OSLanguage、OSPatch 及 Architecture。及應用程式元素中的 APName、APPatch、APVersion、APLanguage 及 APFile。將規則內的這些

部份取出建成決策樹，再與主機運作環境比較，可以判斷規則與主機運作環境是否相同。

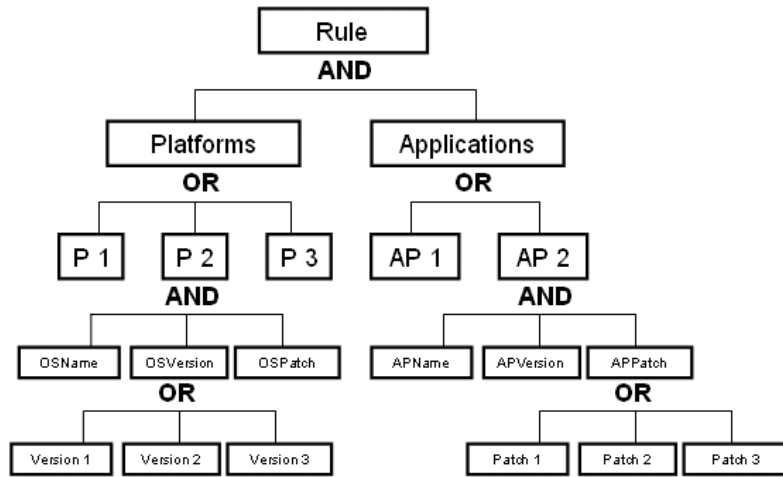


圖 3-12：規則套用決策樹

```

- <RuleValid>
  <Attribution Name="OSName" />
</RuleValid>
- <RuleContent Action="BLOCK">
- <Platforms>
- <Platform OSName="Microsoft Windows 2000">
  <OSVersion>5.00.2195</OSVersion>
  <OSLanguage>zh-tw</OSLanguage>
  <OSPatch>Service Pack 3</OSPatch>
  <Architecture>AT/AT COMPATIBLE</Architecture>
</Platform>
- <Platform OSName="FreeBSD">
  <OSVersion>4.7-Release</OSVersion>
  <OSLanguage>zh_TW.Big5</OSLanguage>
  <Architecture>i386</Architecture>
</Platform>
</Platforms>
<Applications />
+ <Connections>
</RuleContent>

```

圖 3-13：通用規則範例部份內容

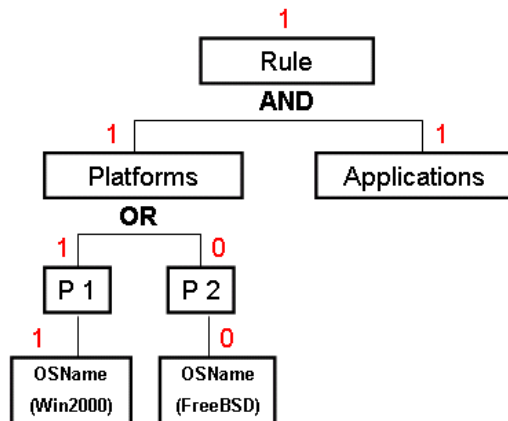


圖 3-14：判斷規則是否保留之決策樹

3.4.6 通用規則的支援模組

不同的系統平台與應用程式需要具有不同的資訊蒐集模組來取得規則內的各個屬性值。如圖 3-13 所示，每個作業系統都會有一個支援模組輔助防火牆取得通用規則內的各種系統平台屬性值，才能與規則內的屬性值加以比對，達到判斷運作環境是否相同的目的。

表 3-5 是以 Windows 2000、FreeBSD、與 Linux 三種作業系統為例，實際說明資訊蒐集模組如何實作取得各個系統平台屬性值。

表 3-5：三種作業系統下之系統平台描述資訊蒐集模組實作方式

| | Windows 2000 | FreeBSD | Linux |
|--------------|--|--------------|--------------|
| OSName | Registry : HKEY_LOCAL_MACHINE\SOFTWARE \Microsoft\Windows NT\CurrentVersion\ ProductName | uname -s | uname -s |
| OSVersion | Registry : HKEY_LOCAL_MACHINE\SOFTWARE \Microsoft\Windows NT\CurrentVersion\ CurrentVersion HKEY_LOCAL_MACHINE\SOFTWARE \Microsoft\Windows NT\CurrentVersion\ CurrentBuildNumber | uname -r | uname -r |
| OSLanguage | Registry : HKEY_LOCAL_MACHINE\SYSTEM\ CurrentControlSet\Control\Nls\Language\ InstallLanguage | LANG 環境變數 | LANG 環境變數 |
| OSPatch | Registry : HKEY_LOCAL_MACHINE\SOFTWARE \Microsoft\Windows NT\CurrentVersion\ CSDVersion | N/A | N/A |
| Architecture | Registry : HKEY_LOCAL_MACHINE\HARDWAR E\DESCRIPTION\System\Identifier | uname -p | uname -i |

| 通用規則語言 | | |
|-------------------|--------------|------------|
| Windows 2000 支援模組 | FreeBSD 支援模組 | Linux 支援模組 |
| Windows 2000 | FreeBSD | Linux |

圖 3-15：通用規則語言支援模組示意圖

應用程式屬性值的取得方式則由作業系統名稱及應用程式名稱共同決定。同一個應用程式在不同的作業系統下可能使用不同的方式來取得同一個屬性值。因此，每種應用程式在不同作業系統下有不同模組取得各個應用程式屬性值。

第五節 網蟲的內部網路癱瘓問題

為了使各防禦機制能夠進行合作防禦，必須確保各主機的網路連線暢通。但近幾年來，網蟲所造成的內部網路癱瘓問題，假如使用目前的解決方式，不論是以代理伺服器或路由器對網蟲連線加以阻擋，仍然無法解決內部網路主機無法互相連線的問題，因此合作防禦機制便無法正常運作。針對這一點，本研究設計了一種網蟲偵測方式實作於各合作式防火牆中，能在感染網蟲時，保持各主機的網路通暢以利合作防禦的運作。

在合作式防火牆主機上偵測及阻擋網蟲的原因有以下幾點：

1. 在主機端阻擋網蟲連線的效益最佳

在愈靠近主機端阻擋網蟲向外擴散所得到的效益愈大。利用網路式防火牆或路由器[32,33]進行封包阻擋時，雖然內部網蟲無法向外部網路擴散，但是與網路感染主機的所在網段仍充斥著大量的網蟲擴散連線，並造成網路壅塞。在網路的各交換器上，對有問題的卡號進行阻擋的方法，同於該埠以下的主機仍然可能因為該埠以下的網路段壅塞而無法連外，而且網蟲感染主機的所有連線，不論正常或是不正常連線都將受到阻擋。如果能在主機端加以阻擋，擴散的攻擊連線便完全無法對外發送，所以沒有其他主機會受到該主機感染網蟲的影響。另外，也可只針對有問題的連線加以阻擋，網蟲感染主

機的其他正常連線仍可繼續保持通暢。因此，我們希望在主機端進行阻擋網蟲攻擊連線的動作。

2. 網蟲感染主機可能無法取得外部資訊

由於網蟲感染後可能大量地向外擴散，而擴散所使用的攻擊連線常常造成網路癱瘓。因此，在網蟲向外擴散時，網蟲感染主機可能無法與其他機器進行溝通取得外部資料，因此網蟲感染主機必須自行進行偵測及阻擋網蟲。

3. 可取得主機資訊輔助偵測

在主機端，我們可以更容易取得主機資訊來輔助判斷出網蟲連線。例如：CPU 使用率增加、開啟連線數目暴增、主機流量變大、系統記錄檔出現攻擊特徵等，都是網蟲感染時常見的現象，有些甚至是網路端無法取得的資訊。利用這些資訊，有助於判斷出網蟲攻擊。

3.5.1 網蟲的偵測方式

現在的網蟲偵測方式可分為單點式與分散式兩種。單點式偵測方法利用單一主機資訊偵測網蟲。而分散式偵測方法必須集合多台主機的資訊，加以分析來找出網蟲連線，例如 GrIDS [37,38] 及 Thomas Toth 提出的網蟲偵測法[40]即屬於分散式偵測。由於網蟲可能造成主機無法對外連線，因此在分散防火牆上的網蟲偵測必須採用單點式方法。本研究提出一個簡單的網蟲偵測方法，假設「單一主機在短時間內，不應產生大量目的位址不同、但目的通訊埠與內容都相同的連線」，如果有連線違反這個原則，即認定為網蟲擴散連線。

這個偵測方法主要是基於目前網蟲的兩個特性：

1. 密集的擴散攻擊連線

目前的網蟲為了要快速地擴散，多會在短時間內產生大量的擴散攻擊連線。攻擊連線愈密集，擴散速度就愈快，也愈容易造成網路癱瘓，而此類網蟲的嚴重性也愈大。以目前擴散最快速的 Slammer 網蟲來說，其單一主機的擴散連線依主機效能不同，有可能佔滿 100 Mbps 的頻寬，在開始十分鐘

內就可以掃描 Internet 上超過 90% 的主機，因此在短時間內就造成相當大的傷害[41]。而擴散速度慢的網蟲，嚴重程度較小，也不易造成網路癱瘓。

2. 擴散行為固定

目前已知的網蟲多使用相同的攻擊連線方式來進行擴散。因此，只要將連線封包的檔頭資訊去除，同一個網蟲的攻擊連線的封包內容部份都會相同。這個特點有助於用來偵測網蟲。在連線封包內容相同的情形下，封包的大小也會相同。

另外，網蟲的攻擊連線的目的通訊埠皆相同也是非常重要特徵。目前攻擊連線的目的通訊埠多為漏洞服務的預設通訊埠。例如：CodeRed 網蟲攻擊 IIS 時，通訊埠皆為 80。Slammer 網蟲攻擊 MS SQL Server 時，通訊埠皆為 1434。我們認為短時間內，網蟲並不會改變這個攻擊特性，開始攻擊非預設通訊埠。因為網蟲無法找出漏洞服務所開啟的通訊埠，除非使用類似 Nmap [35]的「服務辨視」的技術，但是此方法成本太高，也會讓網蟲更容易被偵測出來。所以短期之內，網蟲仍會攻擊最多人使用的服務預設通訊埠。

我們可以監視所有主機對外的連線，找出在一定時間內是否有封包內容及目的通訊埠相同的連線快速傳送至許多位址不同的主機。為了提高偵測正確率，可以利用以下兩點來加強此偵測方法：

1. 擴散攻擊連線的封包具有一定長度

網蟲的程式必須包含攻擊及擴散等步驟，所以我們假設每個網蟲都具有一定程度的封包長度。以現有擴散攻擊連線封包最短的 Slammer 網蟲[30,31]為例，各層封包檔頭長度，網蟲程式碼仍有 376 bytes。其他較大的網蟲的攻擊連線還可能分為數個封包，而除了最後一個封包外的其他封包的大小都為最大封包長度。因此，即使最後一個封包的長度很短，我們也能利用前面的封包來找出網蟲攻擊連線。

2. 擴散攻擊連線以 IP 位址為基礎

網蟲擴散時的目的位址選擇策略，不論是隨機選擇或是 Localized

Scanning [25]，都是以 IP 位址為基礎 [26]。這些被選到的 IP 位址會被直接用來傳送擴散攻擊連線，但不會在主機上留下網域名稱 (domain name) 轉換記錄。但是一般的使用習慣並不會在短時間內，直接使用 IP 位址進行大量不同主機的連線。因此，對於有網域名稱轉換記錄的 IP 位址，在網蟲連線數目統計時並不會加入計算，可進一步提高精確率。

3.5.2 門檻值的建立

使用上述的偵測方法時，很重要的一個關鍵是如何決定門檻值，也就是在多短時間內，有多少相同的連線封包出現時，就認定該連線為網蟲攻擊連線。在本研究中，因為此偵測法最主要的目的是找出網蟲連線再加以阻擋來解決網蟲的網路癱瘓問題。因此，我們使用類似於頻寬分配的方式計算門檻值來找出可能癱瘓網路的連線。

計算攻擊連線封包速率門檻值的公式可表示如下：

$$R = \frac{B}{(N \times S)}$$

其中 B 代表網路重要通道的頻寬，意指最容易發生壅塞的網段所具有頻寬大小。N 代表重要通道以下網路的主機數目。S 代表網蟲攻擊連線的假設封包大小值，可以封包最大值代替。R 代表攻擊連線封包速率的門檻值。

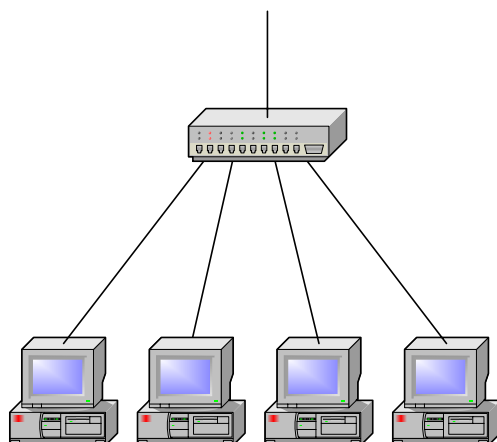


圖 3-16：網蟲偵測門檻值計算範例之網路架構圖

以圖 3-7 的網路架構為例，最容易發生網路壅塞的網段是交換器的對外連線通道，而不是交換器與各個主機間的連線，假設其頻寬為 100 Mbps。對外通道下的網路主機數目為 4。網蟲攻擊連線封包大小假設為 1000 Bytes (8000 bits)。

此時，門檻值約為：

$$R = \frac{100000000}{(4 \times 8000)} \approx 3125$$

此時，只要交換器下的四台機器每秒發送 3125 個以上的封包，就足以使要道頻寬全被佔滿，而造成對外網路癱瘓。

第六節 本章小結

本章對分散式防火牆可能進行合作防禦的對象、方式、及困難點進行探討，並提出本研究的解決方式。首先，本研究認為分散式防火牆可與網路式防火牆、及其他分散式防火牆節點交換規則來抵禦平衡負載及抵禦相同攻擊。可與入侵偵測系統合作變成入侵偵測及預防系統，在偵測到入侵後進行防禦動作。可與漏洞掃描系統組成一個漏洞管理及防禦機制，對有漏洞的應用程式進行連線控制，防止遭受漏洞攻擊。

本研究認為上述合作防禦最大的問題有二：溝通問題及內部網路癱瘓問題。溝通問題指的是目前的防火牆規則語言無法支援合作防禦的進行。內部網路癱瘓問題則會讓合作防禦機制完全無法運作。本研究提出通用規則語言，使得防火牆規則能夠套用至合適的主機上，並且同時解決分散式防火牆難以管理的問題。另外，藉由提出一個網蟲偵測方式，能對可能癱瘓內部網路的網蟲擴散連線加以偵測及阻擋來維持內部網路任兩台主機可正常溝通。

第四章 系統架構與設計

本研究以分散式防火牆為基礎，加入合作防禦的概念，提出合作式防火牆系統 COFS (Cooperative Firewall System)。系統中的各個分散防火節點可以互相溝通，或與其他網路安全防禦機制來進行合作防禦。本章將提出合作式分散防火牆系統的概念、架構、系統內各角色及其功能。

第一節 設計原則

下面提出合作式防火牆的設計原則，並對提出這些原則的目的加以說明。

- 擴充性

只要某台主機安裝了防火牆軟體，該主機就能受到防火牆的保護。且在防火牆群組中新增主機時，不需要再更改其他主機的防火牆設定。因此，可以很容易地在防火牆群組加入新的主機共同進行合作防禦。

- 獨立性

各防火牆能獨立對安裝主機進行防禦，其功能不受其他主機上防火牆發生故障所影響。此外，合作防禦的功能也不會因某個主機發生問題而停擺。目的是使合作式分散防火牆具有一定程度的容錯能力。

- 即時偵測及回應能力

分散式防火牆應具有即時異常偵測及對異常狀況進行即時回應的能力。因為部份的攻擊行為無法藉由監測網路連線的網路式偵測方法來偵測到，而在主機上進行偵測不但可以擁有最多的資訊輔助判定攻擊，還可解決以往網路式偵測方法所無法偵測的項目（如：連線加密問題）。此外，某些攻擊唯有由主機端直接進行回應動作才能確實降低攻擊所造成的威脅及傷害。其中最著名的例子就是內部網路的分散式阻斷服務攻擊，唯有從主機端中斷送出攻擊封包，才能解決區域網路內充滿攻擊連線封包所造成網路壅塞情形。

- 容易使用

管理人員能夠不用考慮各主機的系統資訊及運作環境，只需要針對問題以規則進行描述，就能將規則套用在合適的主機上。

- 合作防禦能力

各主機上的防火牆能進行合作防禦。一旦某台主機偵測到攻擊，可將此資訊回報給其他主機進行事先防禦。

- 能與現有網路安全機制整合

主機上的分散式防火牆能與現有網路安全機制結合，如：入侵偵測系統、網路式防火牆，都可以與合作式防火牆系統溝通。如此一來，可以使以往的安全投資不至於浪費，並藉由結合現有的網路安全機制縱深防禦及合作防禦的功效。

第二節 系統概觀

本研究提出的合作式分散防火牆系統，架構如圖 4-1 所示，共分為四種角色，分別為防火牆節點、註冊節點、管理節點，及 DNS Server。各種節點可共同組成防火牆群組，群組的名稱為 DNS 網域名稱 (domain name) 格式，例如：fwgroup.domain.org。防火牆群組中必須至少具有一個以上的註冊節點，而管理節點及防火牆節點的數量則沒有限制，可以完全沒有，也可以有多個。整個群組至少需要具有一台 DNS Server 來輔助溝通，也可以多個群組共用同一台 DNS Server。同一個防火牆群組內的各個防火牆節點可互相交換訊息進行合作防禦。具有不同系統平台的主機可以區分為不同的防火牆群組，因為系統平台不同通常受到的攻擊也會不同。

四種節點角色都具有其獨特的功能，但可同時安裝於同一台主機中，並共用相同的功能模組，茲分列如下：

1. 防火牆節點

安裝於需要受保護的主機上，主要功能為偵測異常狀態、蒐集系統資

訊、接受外部資料、執行回應動作及與其他節點進行溝通。

2. 註冊節點

具有與 DNS Server 溝通及進行節點註冊的能力，負責處理節點註冊及移除動作。

3. 管理節點

提供使用者介面供管理人員制訂規則發送至群組內其他節點。

4. DNS Server

DNS Server 負責提供網域名稱查詢服務及與註冊節點溝通，當註冊節點提出註冊要求時，會將資料加入 DNS 的資料庫中。

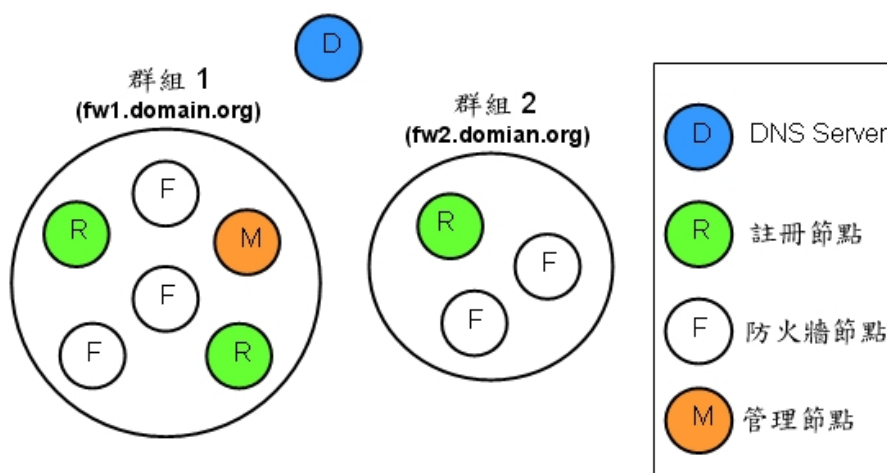


圖 4-1：合作式分散防火牆系統概觀示意圖

第三節 運作流程

防火牆群組的功能是進行合作防禦，同一個群組的防火牆可以互相交換規則來共同防禦攻擊。加入防火牆群組的主機，可以接收到同一群組其他防火牆的規則進行合作防禦。也可以對同一群組中的一個或多個防火牆發送規則。在達到這個目的，主要要解決兩個問題：第一是節點如何加入或離開防火牆群組。第二是如何取得目前防火牆群組內所有防火牆位址來交換資訊。我們利用 DNS 機制來解決上述兩個問題，將群組名稱制訂網域名稱格式，網域名稱所對應的 IP 位址即為群組內的各個防火牆位址。再利用更新及查詢網域名稱來解決上述問題。

要組成一個新的防火牆群組，首先要先有一台註冊節點，並將註冊節點的 IP

位址加入該群組名稱的網域名稱中。接下來，便可以進行新節點的註冊。

- 節點加入及移除

若是有新節點想加入該防火牆群組，首先利用該防火牆群組的名稱向 DNS Server 查詢出所有節點位址，並將註冊資料傳送給所有主機。如圖 4-2 所示，想要加入防火牆群組的主機，首先向 DNS Server 查詢 fwgroup.domain.org (步驟一)。DNS Server 便會回應 fwgroup.domain.org 目前所有主機 IP 位址 (步驟二)，這些主機就是目前防火牆群組內所有的主機。接著，想要加入防火牆群組的主機會傳送註冊資訊給所有防火牆群組內的主機 (步驟三)。最後，註冊節點一接到註冊資料後，便會檢查 DNS Server 是否已有該筆記錄。如果結果為否，則向 DNS Server 加入該註冊主機，新註冊的節點便會被加入此防火牆群組。

節點移除流程與註冊流程相同，想要離開防火牆群組的節點，將含有主機資訊的節點移除要求傳送所有主機，註冊節點收到後會處理節點移除的動作。

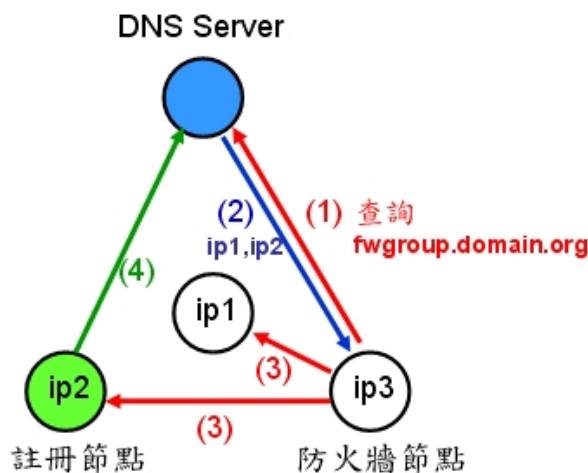


圖 4-2：改良式架構註冊流程圖

- 規則傳送流程

若是任一防火牆節點需要傳送規則給群組內的其他防火牆節點，流程與註冊前三個步驟相同。首先會利用群組名稱向 DNS Server 查詢目前群組內所有節點位址 (步驟一)。接下來，DNS Server 會回應此群組內所有節點的 IP 位址 (步

驟二)。最後，防火牆節點可選擇將規則傳送給 IP 列表中的一個或多個節點（步驟三）。

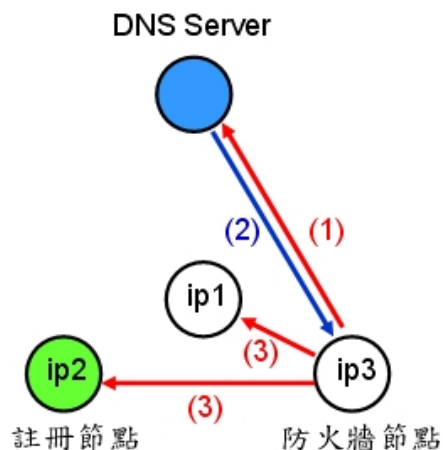


圖 4-3：改良式架構規則傳送流程圖

本研究提出的合作式防火牆在系統架構上使用 DNS Server 來輔助管理各防火牆群組內節點位址，而不採用目前最普遍的單一管理伺服器的架構。以下對採用此架構的理由及對可能面臨的問題進行探討。

我們決定採用此架構的原因如下：

1. 提高容錯程度

合作式防火牆架構利用 DNS 機制來取得群組內的各節點的 IP 位址，比起使用單一中央管理伺服器容錯程度較佳。因為 DNS 屬於網路基礎服務，目前已有許多方式可以輕易地增加 DNS 機制的容錯程度。最常見的方法是建置多個備份 DNS Server，能在主要 DNS Server 發生錯誤時，仍然可以保持 DNS 機制的正常運作。比起單一管理伺服器方法需要額外設計其容錯架構的方式簡便許多。

2. 隱藏註冊節點的位址，減少被攻擊的可能

在整個系統運作流程中，註冊節點的位址幾乎是隱藏的。利用防火牆群組名稱進行 DNS 查詢時，DNS Server 回應的是所有防火牆群組的所有種

類節點的位址。

要癱瘓防火牆節點間的合作防禦功能運作時，可能的方式有二：一是癱瘓節點間溝通，使得節點間無法互相連絡；二是藉由癱瘓註冊功能，使得新節點無法加入群組進行合作防禦。要使節點無法與其他節點溝通，必須癱瘓 DNS 機制，在建置多台備份主機的情形下，困難度頗高。若是想癱瘓防火牆群組的註冊功能，首先要找到註冊節點。但是當群組內的防火牆節點愈來愈多，要找到註冊節點也就愈困難。合作式防火牆的架構隱藏了註冊節點的真正位址，減少被攻擊的可能。因此，此架構的註冊及節點互相溝通的功能，都很難被中斷。

3. 運作方式簡單

不論是節點進行註冊或是已在群組內的防火牆節點想要傳送規則給其他節點，都只需要先進行一次 DNS 查詢動作，即可進行傳送註冊資訊或是規則。在系統設計裡也只要加入簡單的 DNS 查詢模組，不論在運作流程上或是實作上都相當簡單。

但是這個架構會造成傳輸資料量的增加，以下分為規則傳送及節點註冊兩部份來做探討。

- 在傳送規則時

在傳送規則至群組內所有節點時，使用單一管理伺服器的傳輸資料量包含了將規則檔傳送至管理伺服器，及管理伺服器將規則檔傳送至其他節點所耗費的資料量。假設規則檔的大小為 s ，群組內有 n 個節點，所有傳輸的資料量共為 $s \times 1 + s(n-1)$ 。

而改良式架構傳送規則時的傳輸資料量包含了一開始的 DNS 查詢及回應流量，加上將規則檔傳送至其他節點所耗費的資料量。DNS 查詢及回應流量並不太大，可以忽略不計。假設規則檔大小為 s ，群組內有 n 個節點，要將規則傳

輸至群組內所有節點時，傳輸的資料量共為 $s(n-1)$ 。

因此，在傳送規則時，此架構的傳輸資料量並不會與單一管理伺服器的方法有太大的差別。

- 在節點註冊時

如圖 4-4 所示，在節點註冊時，若是使用單一中央管理伺服器的架構時所需要傳送的資料，只需將註冊資訊傳給中央管理伺服器。但在目前的架構中，若只考慮傳送註冊資訊時的資料量時，因為註冊資料需要傳送給群組內的所有主機，隨著主機數目愈多，傳輸資料量也會愈多。

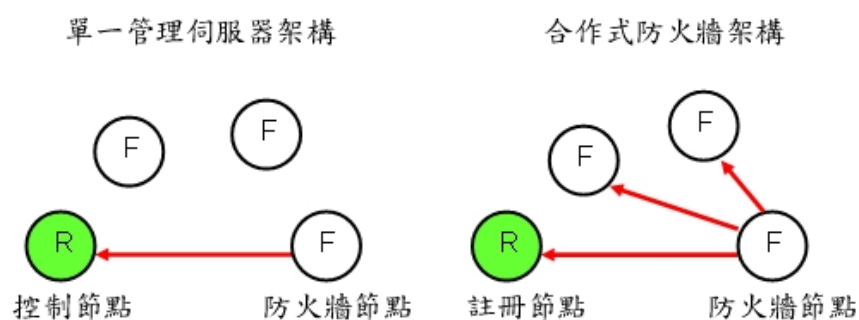


圖 4-4：註冊方式傳輸量比較圖

針對此問題，本研究提出一個解決方式來改善傳輸資料量遽增的問題。藉由分組及增加註冊節點，可以有效減少註冊時的資料傳輸量。分組是將群組內的所有節點等分為 n 個小組進行逐次傳輸，直到註冊成功為止。同時，也增加註冊節點數目來提高找到註冊節點的機率，進一步減少註冊時的傳輸量。

新節點在進行註冊時的運作流程如圖 4-5 所示。以圖 4-6 為例，藉由 DNS Server 查詢得知防火牆群組內有五個節點，若是分成二組，可以將查詢結果中的前二個節點分為第一組，其他三個節點分為第二組。接著將註冊資訊傳送給第一組內的所有節點，等待一段時間後再度查詢 DNS，檢查自己的位址是否已加入群組。若結果為否，再將註冊資料傳送至第二組節點。圖 4-6 的例子中，群組內有兩個註冊節點，在分組時分別分至不同的組別。第一次傳送後，因為第一

組內有一個註冊節點，所以它將會幫新節點註冊。此時，新節點便不用傳送註冊資訊給第二組內的節點了。因此，可達到減少傳輸量的效果。

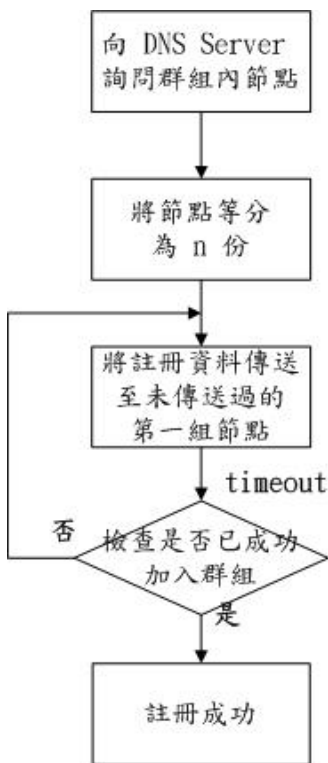


圖 4-5：新節點註冊運作流程圖

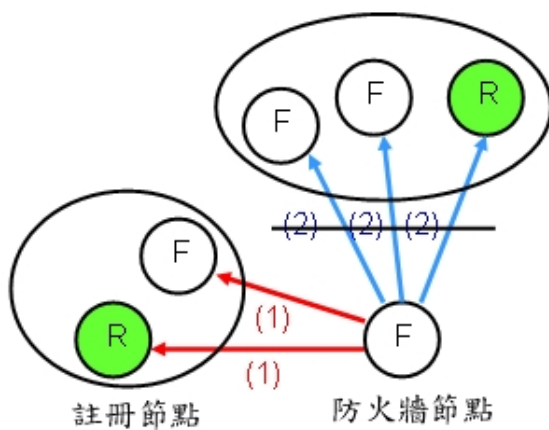


圖 4-6：分組傳送註冊資訊示意圖

第四節 內部模組設計

本節對各角色的內部模組設計加以解釋，並說明各模組的功用。各節點角色

的模組目的為達到各角色所負責的功能。以下依序說明防火牆節點、註冊節點、管理節點及 DNS Server 的內部模組設計。

4.4.1 防火牆節點

如圖 4-7 所示，防火牆節點主要可分為：「認證與存取控制」、「資料蒐集與防火牆規則存放」、「資料分析及規則判斷」、及「回應執行」四大部份。

認證與存取控制部份負責在節點對外交換資訊時，進行身份驗證與存取控制，所有對外傳送或接送資訊的動作都會先經過身份驗證確認後才會開始進行。

資料蒐集與防火牆規則部份中包含了「自我偵測模組」、「外部資訊輸入模組」、「節點資訊蒐集模組」、及「規則存放區」。

- 自我偵測模組

此模組負責偵測主機的各種異常狀況，尤其是會影響防火牆節點運作或是與其他節點進行合作防禦的異常狀況。在效能允許範圍內，可針對不同的異常狀況加入不同的偵測方法來加以偵測。自我偵測模組在偵測到異常後，會蒐集攻擊情境相關的資料來產生防火牆規則，再交由資料分析及規則判斷模組加以套用來中斷異常行為的發生。例如網蟲的偵測方法最後就是實作在此模組中。

- 外部資訊輸入模組

用來接收其他系統的資訊，如其他入侵偵測系統的警告訊息或防火牆的記錄檔，目的是使防火牆節點能與現存的安全機制結合共同防禦攻擊。防火牆節點利用此模組與其他安全機制進行合作禦，對於不同安全機制傳來的資訊也會有不同的解譯方法來取得需要的資訊。

- 節點資訊蒐集模組

此模組負責蒐集節點主機的各種相關資訊，利用通用規則的高階資訊取得執行回應動作時需要的運作細節。例如：作業系統種類版本、修補程式安裝狀況、開啟的服務及其通訊埠等。

● 規則存放區

此模組負責檢查接收到的規則是否需要被留下及管理所有防火牆規則，並定時呼叫資料分析及規則判斷模組檢查規則是否符合運作環境。

資料分析及規則判斷模組負責判斷規則是否符合目前的運作環境來決定是否要採取回應動作，並與節點資訊蒐集溝通取得判斷規則是否套用時所需要的資訊。一旦決定套用，此模組會將回應動作傳送至適合的回應模組。

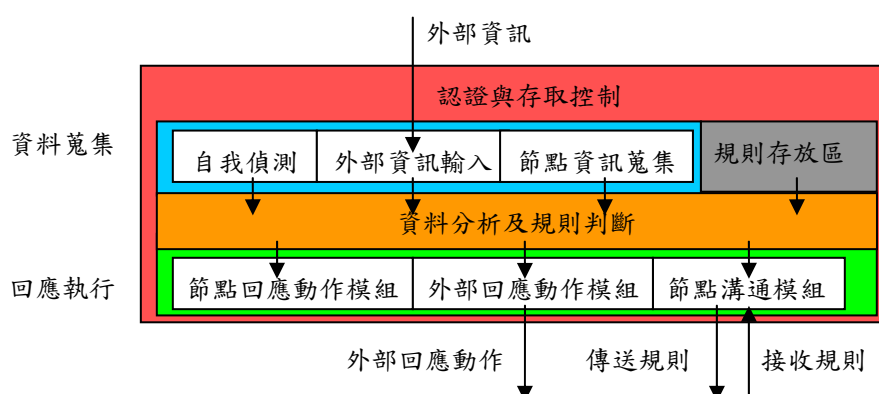


圖 4-7：防火牆節點內部設計架構

回應執行部份分為節點回應動作執行模組、節點溝通模組、及外部回應動作模組。節點回應動作模組負責對做出節點回應動作，目前可分為網路連線類回應動作及應用程式類回應動作。網路連線回應動作包含對某類連線做阻擋的動作，及減慢連線的傳輸速度[27,28]。減慢連線傳輸是比阻擋連線較為「仁慈」的方法，可以有效減低類似網蟲攻擊的傷害，而發生誤判時，正常連線也不會完全中斷，直到管理人員加以判斷及處理。節點溝通模組與其他節點溝通，進行傳送及接受規則及註冊資訊動作，並具有 XML Parsing 功能處理 XML 格式的溝通資訊及規則。外部回應動作模組可與其他回應機制進行溝通，藉由加入不同的回應動作樣本，防火牆節點能與交換器溝通進行鎖卡，或是產生警告訊息至其他監控系統。

4.4.2 註冊節點

註冊節點的主要功能是接收防火牆節點的註冊資料，並向 DNS Server 進行節點註冊及移除動作。

認證與存取控制模組及節點溝通模組與防火牆節點功能完全相同。註冊模組負責檢查註冊資料格式是否正確，查詢是否註冊資料是否真的未註冊後，將註冊資料轉換成 DNS 註冊時的格式，再向 DNS Server 註冊。

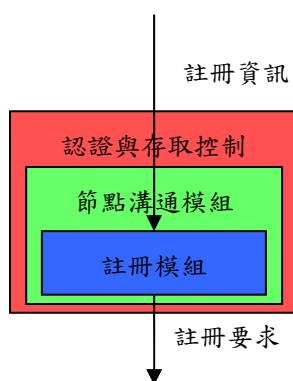


圖 4-8：註冊節點內部設計架構

4.4.3 管理節點與 DNS Server

管理節點提供管理人員一個使用者介面來制訂規則，並傳送至一個或多個防火牆節點。如圖 4-9 所示，認證及存取控制模組與防火牆節點功能完全相同。管理介面為設定規則時的使用者介面。規則轉換模組則接收管理人員於管理介面所制訂的規則內容，並將規則內容轉換成防火牆節點使用的通用規則。最後，利用節點溝通模組進行傳送。

DNS 設定模組負責接收註冊要求，更改 DNS Server 內資料，來加入註冊主機，模組設計如圖 4-10，只有認證及存取控制模組與 DNS 設定模組。

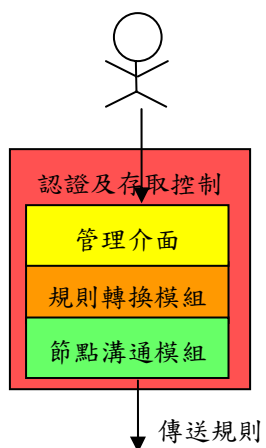


圖 4-9：管理節點內部設計架構

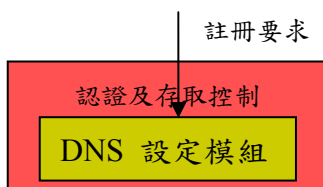


圖 4-10：DNS Server 節點內部設計架構

第五章 模擬實驗

本章以阻擋網蟲擴散及與 snort IDS 進行合作防禦兩個模擬實驗來說明合作式分散防火牆系統的效用，並對模擬實驗方式與結果進行討論。

第一節 模擬實驗一

- 實驗情境

內部網路中的主機感染 Slammer 網蟲，開始向外發送大量攻擊封包，造成對外網路壅塞。

- 實驗目的

提供一個自動化機制能夠自動偵測網蟲及阻擋網蟲擴散，並能在狀況解除時，自動解除對該類封包的阻擋。

- 實驗流程

1. 外部網蟲攻擊內部漏洞主機造成網蟲感染。
2. 內部網蟲感染主機開始向外擴散，造成網路壅塞現象。
3. 合作式防火牆中的自我偵測模組發現網蟲連線，並判斷有可能造成網路癱瘓。
4. 自我偵測模組產生相關規則阻擋網蟲擴散，網路恢復正常。
5. 網蟲感染主機經修復後，阻擋規則自動被移除。

- 實驗結果

圖 5-1 是偵測到網蟲的畫面。在啟動自我偵測程式時，可以指定只統計封包大小在 100 bytes 以上的連線。1000 則是門檻值，只要有封包內容、目的通訊埠相同的封包每秒傳送至 1000 台不同的主機，該類封包就會被認定為是網蟲的攻擊擴散封包，而被防火牆加以阻擋。若是連線速率連續低於門檻值 10 秒以上，我們就認為主機已恢復正常，則解除對該主機的阻擋動作。值得注意的是由圖 5-1 中可看到 Slammer 網蟲每秒鐘送出約 4000 個的擴散攻擊封包，所以只

需有數台主機感染網蟲就足以癱瘓對外網路。而管理者可藉由降低門檻值來增加防禦的強度。

```

root@labfw:~/demo/demo1
[root@labfw demo1]# ./show_worm 1 100 10 10
UDP 10.2.2.2:1050 -> *:1434 (3686)
UDP 10.2.2.2:1050 -> *:1434 (3486)
UDP 10.2.2.2:1050 -> *:1434 (4372)
UDP 10.2.2.2:1050 -> *:1434 (3721)
UDP 10.2.2.2:1050 -> *:1434 (4322)
UDP 10.2.2.2:1050 -> *:1434 (3809)
UDP 10.2.2.2:1050 -> *:1434 (4063)
UDP 10.2.2.2:1050 -> *:1434 (3555)
UDP 10.2.2.2:1050 -> *:1434 (4215)
UDP 10.2.2.2:1050 -> *:1434 (3663)
UDP 10.2.2.2:1050 -> *:1434 (4327)
UDP 10.2.2.2:1050 -> *:1434 (3350)
UDP 10.2.2.2:1050 -> *:1434 (4116)
UDP 10.2.2.2:1050 -> *:1434 (3821)
UDP 10.2.2.2:1050 -> *:1434 (4353)
UDP 10.2.2.2:1050 -> *:1434 (3668)
UDP 10.2.2.2:1050 -> *:1434 (3876)
UDP 10.2.2.2:1050 -> *:1434 (3859)
UDP 10.2.2.2:1050 -> *:1434 (3999)
UDP 10.2.2.2:1050 -> *:1434 (3780)
    
```

圖 5-1：Slammer 網蟲偵測畫面

```

root@labfw:~/demo/demo1
[root@labfw demo1]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination
DROP      udp  --  10.2.2.2              anywhere          udp dpt:ms-sql-m len
gth 404

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
[root@labfw demo1]#
    
```

圖 5-2：Slammer 網蟲阻擋規則

自我偵測模組在偵測到網蟲後，會產生阻擋規則並進行套用來阻擋擴散封包的傳送（圖 5-2）。由於之前網蟲偵測條件是「大量目的位址不同、但目的通訊埠及內容相同的封包」。因此，我們取出封包的協定種類、目的通訊埠、封包長度

(用來代替封包內容) 產生合適的阻擋規則。

第二節 模擬實驗二

● 實驗情境

某主機受到外部主機以 Nmap 進行主機掃描，入侵偵測系統發現此攻擊行為後，與合作式防火牆進行合作防禦阻擋此攻擊連線。

● 實驗目的

本實驗的主要目的在整合入侵偵測系統與合作式防火牆共同防禦攻擊，藉由入侵偵測系統監視是否有攻擊發生。一旦偵測到攻擊，藉由將攻擊警示傳送至合作式防火牆來即時進行攻擊防禦，達到入侵預防的目的。

● 實驗流程

1. 外部攻擊者以 Nmap 掃描受害主機。
2. 入侵偵測系統 snort 發現此攻擊，產生攻擊警示。
3. 入侵偵測系統將攻擊警示傳送給合作式防火牆，合作式防火牆再由攻擊警示中擷取適當資訊轉換為阻擋規則進行防禦。

● 實驗結果

利用 Nmap 對安裝合作式防火牆的主機進行掃描時，網路型入侵偵測系統 Snort 可偵測到 Nmap 掃描，產生的攻擊警示如圖 5-3 所示。Snort 偵測到攻擊後，馬上與被攻擊主機的合作式防火牆溝通，並傳送攻擊警示。合作式防火牆藉由 Snort 的攻擊警示資訊轉換為防禦規則，對 Nmap 掃描進行阻擋。

對於網路掃描攻擊，我們會採用阻擋來自攻擊主機的所有連線之防禦策略。所以由攻擊警示中取出攻擊來源 IP 位址，而被攻擊的位址因為是本機位址，則可有可無，產生出的規則範如圖 5-4 所示。進行合作防禦後，Nmap 已無法正確的掃描主機的通訊埠，部份掃描結果畫面如圖 5-5 所示，可以看見攻擊端的 Nmap 掃描結果已不正確。

```

root@labfw:/var/log/snort
[root@labfw snort]# cat alert
[**] [1:469:1] ICMP PING NMAP [**]
[Classification: Attempted Information Leak] [Priority: 2]
05/07-02:22:07.976099 10.1.1.1 -> 10.0.0.10
ICMP TTL:46 TOS:0x0 ID:47506 IpLen:20 DgmLen:28
Type:8 Code:0 ID:1397 Seq:0 ECHO
[Xref => http://www.whitehats.com/info/IDS162]
[root@labfw snort]# _
    
```

圖 5-3：snort 偵測到 Nmap 時之警示訊息

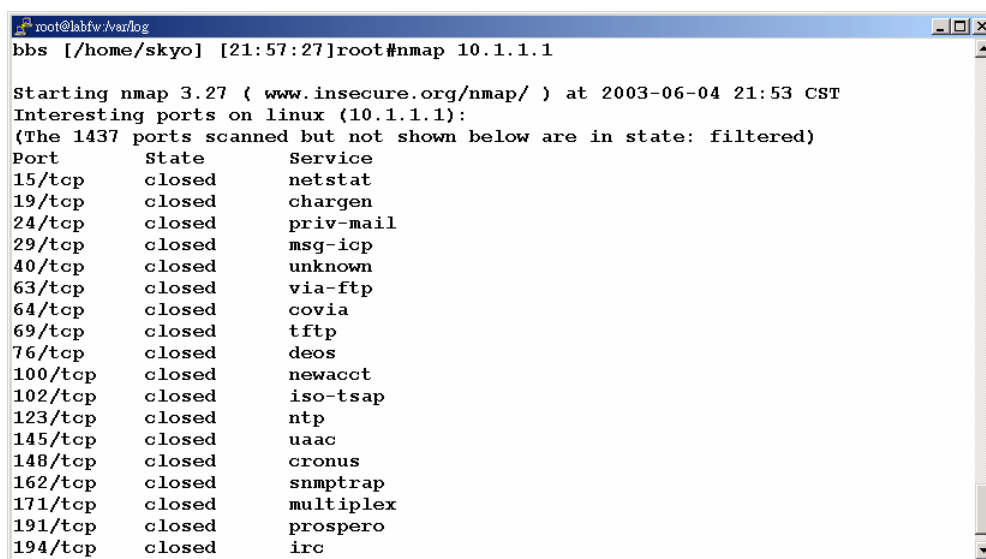
```

root@labfw:/var/log
[root@labfw log]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
DROP      all  --  10.2.2.2              10.1.1.1

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
[root@labfw log]# _
    
```

圖 5-4：由攻擊警示產生的防火牆規則



```
root@labfw:/var/log
bbs [/home/skyo] [21:57:27]root#nmap 10.1.1.1

Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-06-04 21:53 CST
Interesting ports on linux (10.1.1.1):
(The 1437 ports scanned but not shown below are in state: filtered)
Port      State      Service
15/tcp    closed    netstat
19/tcp    closed    chargen
24/tcp    closed    priv-mail
29/tcp    closed    msg-icp
40/tcp    closed    unknown
63/tcp    closed    via-ftp
64/tcp    closed    covia
69/tcp    closed    tftp
76/tcp    closed    deos
100/tcp   closed    newacct
102/tcp   closed    iso-tsap
123/tcp   closed    ntp
145/tcp   closed    uaac
148/tcp   closed    cronus
162/tcp   closed    snmptrap
171/tcp   closed    multiplex
191/tcp   closed    prospero
194/tcp   closed    irc
```

圖 5-5：Nmap 受防火牆阻擋後之部份掃描畫面

第三節 模擬實驗結果討論

在模擬實驗一中，合作式防火牆的自我偵測模組可使用本研究提出的網蟲偵測方法成功解決網蟲所可能造成的內部網路癱瘓問題，且比以往的方式效果更佳，不但沒有其他主機會因為網蟲感染主機而無法對外連線，同時網蟲感染主機的其他正常連線仍可對外連線。同時，本研究也提供了連線恢復機制，能在大量連線停止後，同時解除對該類連線封包的管制。因此，合作式防火牆在網蟲大量感染時仍可繼續與任何安全機制進行合作防禦，也減輕了管理人員在解決網蟲問題時所需要的人力介入負擔。

在模擬實驗二中，合作式防火牆能成功地與入侵偵測系統進行合作防禦，利用入侵偵測系統所提供的警示訊息，取出適當資訊來防禦攻擊。因此，只要入侵偵測系統能夠偵測到的攻擊都可以利用合作式防火牆來進行防禦。由於合作式防火牆的部署位置在攻擊行為的必經道路上，所以阻擋效益會比現有的入侵偵測系統的主動式回應來得高。因為類似 TCP Reset 與 ICMP Unreachable 的主動式回應不一定能夠阻止攻擊，而與網路式防火牆或路由器合作時，單一防火牆或路由器要處理較多的規則，也無法阻擋內部攻擊。

在兩個模擬實驗中，會依攻擊種類不同產生不同的防火牆規則，在模擬實驗一中，我們取出封包的協定種類、目的通訊埠、封包長度來產生防禦規則，而在模擬實驗二中，我們只取出攻擊來源位址來產生防禦規則。這是因為在攻擊偵測還無法完全正確時，採用自動回應機制來阻擋攻擊，可能在偵測動作產生誤判時，反而阻擋了正常的連線。對於這個問題，本研究利用結合更多的資訊來產生與偵測條件相近且能夠阻擋攻擊的規則，藉由縮小阻擋的連線範圍來減小誤判時所發生的傷害。而 guardian [29] 則阻擋所有來自攻擊來源位址的連線，在發生誤判時，反而會造成無法連線的狀況。

以模擬實驗二為例，合作式防火牆藉由接收入侵偵測系統產生的攻擊警示與入侵偵測系統進行合作防禦。如圖 5-6 所示，入侵偵測系統在偵測到攻擊之後，會產生攻擊警示來描述真實世界中的該攻擊情境。對於不同的攻擊情境，入侵偵測系統能夠產生不同的攻擊警示，且所產生的攻擊警示應該能夠完整的描述該攻擊情境。在真實世界中，我們可以從不同攻擊情境中，判斷出這是哪一種攻擊，進而得到對此攻擊該採取何種防禦動作。真實世界中的採取防禦動作決定了該從攻擊警示擷取何種資訊來產生回應動作敘述來防禦此攻擊。因為攻擊警示中的資訊不一定都與回應動作有關，必須視真實世界中所要採取的防禦動作加以選擇適當的資訊來產生回應動作描述。最後，依回應動作描述來真正防禦此攻擊。

因此，對於每一個入侵偵測系統產生的攻擊警示，都應視現實世界中所採取的防禦措施建立該攻擊的處理樣本，說明該取出何種資訊來產生回應動作描述，最後進行回應動作執行。每一個入侵偵測系統也會有不同的轉換規則，轉換規則描述了如何從攻擊警示中取得各種資料的方法，以 snort 為例，轉換規則描述了如何由警示訊息中，取出攻擊名稱、攻擊分類、攻擊時間、來源 IP 位址及通訊埠、目的 IP 位址及通訊埠等各種資訊。

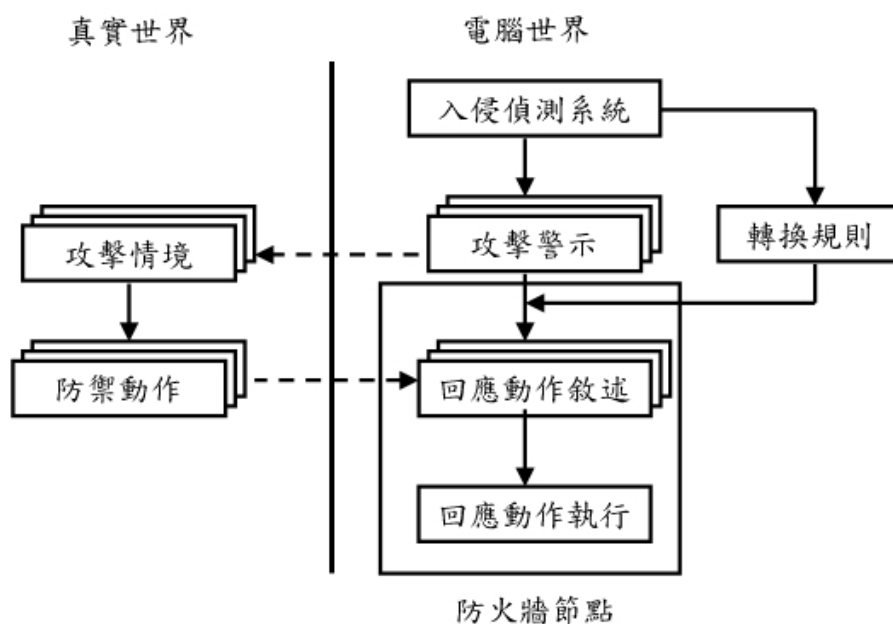


圖 5-6：防火牆節點與入侵偵測系統之合作防禦示意圖

攻擊發生時，入侵偵測系統會將攻擊警示傳給遭受攻擊的防火牆節點，防火牆節點利用外部資訊輸入模組接收入侵偵測系統的攻擊警示，再根據該入侵偵測系統的轉換規則及攻擊處理樣本，將不同的攻擊警示轉換成不同的回應動作描述，藉此與入侵偵測系統進行合作防禦。

最後，合作式防火牆與入侵偵測系統並不能阻擋所有偵測到攻擊行為。若是合作式防火牆與入侵偵測系統分屬兩台不同的主機上，偵測到攻擊後，還需要進行攻擊警示傳送、攻擊警示轉換成防禦規則等動作，若是攻擊在防禦規則生效前已成功便無法阻擋該攻擊。解決方式是將合作式防火牆與入侵偵測合而為一，所有連線在政策規則外，再作入侵行為檢查，確定不是攻擊行為連線才進行允許通過。但是目前最有效的入侵偵測方法仍為特徵比對，但一個連線需要比對所有攻特徵，會對防火牆的效能產生極大影響。目前本研究使用折衷的方式將入侵偵測系統與合作式防火牆置於同一主機，減少溝通所需的時間，經模擬實驗證明可以防止攻擊流程只需一秒的 Nmap 掃描攻擊。

第六章 結論

本章對整個研究作一簡單的結論，並提出本研究的貢獻所在。最後說明本研究可以繼續研究的方向供其他研究者參考。

第一節 研究結論

目前防火牆已成為大多數企業網路安全的第一道防線，也是最重要的攻擊回應機制，因此在未來數年仍是相當重要的安全防禦機制。網路式防火牆已經發展得相當成熟，並且整合許多功能。但是由於架構的限制，並無法解決所有目前防火牆所面臨的問題。而主機式防火牆與分散式防火牆可以用來彌補這些不足，但要達到最大的效益，必須針對管理的動作加以簡化。

攻擊者所使用的攻擊手法日新月異，已經沒有單一的安全機制能夠阻擋所有的攻擊行為。因此，根據縱深防禦與合作防禦的概念，結合多種安全機制共同防禦攻擊是未來的趨勢。防禦最終目的是能達成入侵預防，在攻擊行為成功入侵主機或是造成傷害前阻止其發生。而攻擊流程的進行速度愈來愈快，只依靠人工判斷及手動設定已不可行，必須依靠自動化的偵測機制及防禦流程在第一時間對攻擊進行防禦。同時也要儘量減少發生誤判時，自動化防禦所造成的影響。

第二節 研究貢獻

本研究的主要貢獻如下：

1. 歸納及整理防火牆的演進及所遭遇的問題。
2. 提出用來描述高階資訊的規則概念來解決分散式防火牆的管理及規則套用問題。
3. 提出一個網蟲偵測及防治方法，避免網蟲擴散攻擊時造成的內部網路癱瘓，幫助合作防禦機制運作。

4. 提出三種分散式防火牆可能的合作防禦方式及探討其困難點。

第三節 未來研究方向

對於本研究後續可能的研究方面，我們分為以下數點進行說明：

1. 其他合作防禦的對象及方式的探討

除了防火牆、入侵偵測系統、漏洞掃描系統以外，仍有許多其他的安全防禦機制可以一起進行合作防禦。而這些防禦機制該如何合作抵擋攻擊及可以達到的效果都還需要進一步的研究。

2. 通用規則的加強

在本研究中，我們使用通用規則來解決分散式防火牆的規則套用及管理問題。但是目前的通用規則所具有的屬性還相當的簡單，仍無法描述精細的系統運作環境。如果能有更完整的語言能對系統運作環境加以描述，規則將能夠更精確地套用到適合的系統上。管理者甚至可以不用指定規則要套用到哪些主機上，只要將規則傳給所有主機，主機就能自動判斷是否這個規則需要進行套用。如此一來，管理負擔將能大大地降低。

另外一點，目前通用規則的設定只對功能面進行考量，對於真正應用時的實際需求，例如：溝通時的安全性及規則信賴問題，仍未加以考量。但目前通用規則使用 XML 格式表示，可以輕易的加入需要的標籤再提出新的通用規則版本。

3. 攻擊偵測方式增加及加強

在合作式防火牆的自我偵測模組應具有許多偵測方法來偵測可能影響防火牆運作的攻擊行為。但目前我們只對網蟲影響合作防禦的問題進行探討。本研究假設合作式防火牆本身不會受到攻擊而失去功能。但實際上，應加入偵測方法來檢查合作防火牆本身的安全，來確保防火牆自身的安全無虞。

參考文獻

中文參考文獻

- [1] 王凱，「我國資訊安全市場發展現況與趨勢」，財團法人資訊工業策進會市場情報中心研究報告，民國 91 年 12 月。
- [2] 李勁頤，「利用程序追蹤方法關聯分散式入侵偵測系統之入侵警示研究」，國立中央大學資訊管理學系碩士論文，民國 91 年 6 月。
- [3] Simson Garfinkel 及 Gene Spafford 原著，林逸文、蔣大偉翻譯，「UNIX 與 Internet 安全防護－網路篇」，第 264 至 265 頁，美商歐萊禮台灣分公司，2001 年 12 月。
- [4] 曾宇瑞，「網路安全縱深防禦機制之研究」，國立中央大學資訊管理學系碩士論文，民國 89 年 6 月。
- [5] Carlton R. Davis 原著，劉良棟翻譯，「IPSec－VPN 安全架構與實作」，9-3 至 9-8 頁，麥格羅·希爾國際出版公司，民國 91 年 6 月。
- [6] 林宸堂，「IPsec VPN 的難題：Firewall 與 NAT 的配置」，<http://www.iii.org.tw/ncl/document/IPSecVPN.htm>，民國 90 年 9 月。

英文參考文獻

- [7] Power, Richard, “1999 CSI/FBI Computer Crime and Security Survey”, Computer Security Journal, Volume XV, Number 2. San Francisco, CA: Computer Security Institute, 1999.
- [8] CERT/CC, “Overview of Attack Trends”, Software Engineering Institute, Carnegie Mellon University, 2002.
(Available at http://www.cert.org/archive/pdf/attack_trends.pdf)
- [9] Ed Skoudis, “Infosec’s Worst Nightmares”, Information Security Magazine, November 2002.

- (Available at <http://www.infosecuritymag.com/2002/nov/nightmares.shtml>)
- [10] W. R. Cheswick and S. M. Bellovin, “Firewalls and Internet Security, Repelling the Wily Hacker”, Addison-Wesley Publishing Company, 1994.
- [11] DistributedFirewalls.com, <http://www.distributedfirewalls.com>.
- [12] Steven M. Bellovin, “Distributed Firewalls”, ;login:, November 1999, pp. 39-47.
- [13] Sotiris Ioannidis, Angelos D. Keromytis, Steven M. Bellovin, and Jonathan M. Smith, “Implementing a Distributed Firewall”, ACM Conference on Computer and Communications Security, Athens, Greece, November 2000.
- [14] Wei Li, “Distributed Firewall”, December 2000.
- (Available at <http://www.cs.helsinki.fi/u/asokan/distsec/documents/li.ps.gz>)
- [15] Utz Roedig, Ralf Ackermann, and Christoph Rensing et al., “A Distributed Firewall for Multimedia Applications”, Proceedings of the Workshop "Sicherheit in Mediendaten", September 2000.
- [16] Steve Bridge, “Achieving Defense-in-Depth with Internal Firewalls”, August 2001. (Available at <http://www.sans.org/rr/paper.php?id=797>)
- [17] Edward Hurley, " Intrusion prevention: IDS' 800-pound gorilla", http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci892744,00.html, April 2003.
- [18] Kathleen A. Jackson, “Intrusion Detection System Product Survey”, June 1999.
- [19] Martin Roesch and Chris Green, “Snort Users Manual”, http://www.snort.org/docs/writing_rules/chap2.html#tth_sEc2.3.22, 2003.
- [20] Snort inline Projects, <http://www.honeynet.org/papers/honeynet/tools/>.
- [21] FireHOL Project, <http://firehol.sourceforge.net/>, September 2002.
- [22] Nessus Project: A free, powerful, up-to-date and easy to use remote security scanner, <http://www.nessus.org/>.

- [23] SARA Project: Security Auditor's Research Assistant, <http://www-arc.com/sara/>.
- [24] ISS Internet Scanner, http://www.iss.net/products_services/enterprise_protection/vulnerability_assessment/scanner_internet.php.
- [25] Stuart Staniford, Vern Paxson, and Nicholas Weaver, "How to Own the Internet in Your Spare Time", Proceedings of the 11th USENIX Security Symposium, May 2002. (Available at <http://www.icir.org/vern/papers/cdc-usenix-sec02/>)
- [26] Gregory R. Ganger, Greg g Economou and Stanley M. Bielski, "Self-Secure Network Interfaces: What, Why and How", CMU-CS-02-144, School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213, May 2002.
- [27] Matthew M. Williams, "Throttling Viruses: Restricting propagation to defeat malicious mobile code", 18th Annual Computer Security Applications Conference, December 2002.
- [28] Anil Somayaji and Stephanie Forrest, "Automated Response Using System-Call Delays", Proceedings of the 9th USENIX Security Symposium, August 2000.
- [29] Guardian Project, <http://www.chaotic.org/guardian/>.
- [30] CERT/CC, "CERT Advisory CA-2003-04 MS-SQL Server Worm", January 2003.
- [31] Microsoft Corporation, "PSS Security Response Team Alert – New Worm: W32.Slammer", January 2003. (Available at <http://www.microsoft.com/technet/security/virus/alerts/slammer.asp>)
- [32] Cisco System Inc., "Using Network-Based Application Recognition and ACLs for Blocking the "Code Red" Worm", September 2002.
- [33] Cisco System Inc., "SAFE SQL Slammer Worm Attack Mitigation", January 2002.
- [34] Kerio Personal Firewall, http://www.kerio.com/us/kpf_home.html.

- [35] Nmap, <http://www.insecure.org/nmap/index.html>, 2003.
- [36] Snort: The Open Source Network Intrusion Detection System, <http://www.snort.org/>.
- [37] S. Cheung, R.Crawford, and M. Dilger et al., “The Design of GrIDS: A Graph-Based Intrusion Detection System”, Technical Report CSE-99-2, U.C. Davis Computer Science Department, January 1999. (Available at <http://seclab.cs.ucdavis.edu/arpa/grids/grids.ps>)
- [38] S. Staniford-Chen, S. Cheung, R. Crawford et al., “GrIDS: A graph based intrusion detection system for large networks”, In Proceedings of the 19th National Information Systems Security Conference, pages 361 ~ 370, 1996.
- [39] Robert Gwaltney, “Protecting the Next Generation Network – Distributed Firewalls”, http://www.sans.org/rr/firewall/next_gen.php.
- [40] Thomas Toth and Christopher Kruegel, “Connection-history based anomaly detection”, Proceedings of the 2002 IEEE Workshop on Information Assurance and Security, June 2002.
- [41] CNET Networks, Inc., “Counting the cost of Slammer”, January 2003.
(Available at <http://news.com.com/2100-1001-982955.html>)
- [42] High Level Firewall Language Projects, <http://www.hlfl.org> and <http://freshmeat.net/projects/hlfl/>.
- [43] VulXML Project: A Web Application Security Vulnerability Description Language, <http://www.owasp.org/vulnxml/>, October 2002.
- [44] OVAL, Open Vulnerability Assessment Language, <http://oval.mitre.org/>, October 2002.
- [45] AVDL, Application Vulnerability Description Language, <http://www.avdl.org/>, April 2003.