以預防為基礎之網路聯防機制
Prevention-Based Network Cooperative Defense Mechanisms

周立德 林柏昇 洪茂元 梁宏漢

國立中央大學資訊工程學系

Email: cld@csie.ncu.edu.tw

摘要

隨著網路的發展，網路安全也逐漸受到所有人的重視，各種網路入侵事件層出不窮，使傳統的資訊安全市場產生巨變，不斷翻新的網路入侵技術、類型多變的病毒與網蟲攻擊，散佈於連接全世界的網際網路。資訊安全的重要性與攻擊者入侵問題，近年持續受到企業與政府關切；其中又以分散式阻絕服務 (Distributed Denial of Service，簡稱 DDoS) 攻擊對網路所造成威脅及損害最為嚴重。

本文提出階層式聯合防衛 DDoS 攻擊系統架構；聯合網路型入侵預防系統 (WallGuard)，主機型入侵預防系統 (WallAgent) 及區域派送員 (Domain dispatcher) 三個元件，組成階層式聯合防衛機制。WallGuard 負責多網域間聯防工作，實作流量統計與控制路由設備過濾攻擊。同時利用區域劃分的概念，WallGuard 可以進一步的透過所管轄之 Domain dispatcher 通報子網路下的 WallAgent 共同防衛 DDoS 攻擊，將攻擊阻絕在最近攻擊者端。另外提出分析系統記錄檔之預防機制防止 DDoS 攻擊發生，達到事前的預防效果。

關鍵詞：分散式阻絕服務攻擊、入侵預防系統、網路安全、聯防機制

1. 緒論

隨著全世界電腦用戶的激增，以及網路建設日益普及，國家、公司、個人都免不了會接觸網路，網路已經是生活中不可或缺的一部分，像是網路銀行林立、網路購物日趨普及與線上遊戲遽增等，但是網路時代來臨也產生新的社會問題。無論是智慧型網路犯罪、侵害電腦智慧財產權或是人為刻意的消耗網路資源行為，都可能造成國家機密外洩、造成商業公司龐大的金錢損失與為使用者帶來不便等等；其中消耗網路資源的攻擊與攻擊者入侵的行為近年屢見不鮮，再網路上的任何一台主機都有可能成為受害者，變成攻擊的跳板，使得網路安全需求量漸增。

雖然網路安全一再被強調其重要性，但攻擊事件逐年增加的趨勢，許多次的攻擊皆造成的重大損失與傷害，尤其是近幾年幾件大宗網路蠕蟲病毒，更造成了大量的經濟損失，浪費網路資源，也使得網路安全受到的重視。西元 1998 年美國 CERT[4] (Computer Emergency Response Team) 的成立即是希望透過積極的教育訓練、研究發展、入侵事件收集、線上諮詢服務等方式，希望能夠更加順利的處理未來危害網路安全的相關事件，同時利用這些經驗提高網路社群對電腦安全相關議題的注意，並且對於現有的電腦系統進行研究，以提高電腦系統的安全性，進而預防未來可能發生的電腦入侵與攻擊事件。最終目的則是降低因電腦緊急事故所引發的傷害和損失，使全民皆能享受社會資訊化的好處。同時台灣政府亦不落人後，自西元 2000 年正式成立台灣電腦網路危機處理中心 [11] (TWCERT/CC)，提高台灣企業、民眾與教育單位對網路安全的重視。

網路安全的最高境界便是能預測攻擊者動向，讓安全防護的機制能動敵機先。不過在真實的世界中，常常事與願違。新的攻擊事件依然層出不窮，網際網路每一處依然經常受到各類型的電腦病蟲攻擊。專門提供資安技術廠商不斷研究新的電腦病蟲與新的攻擊手法，制定新的機制來應付新的威脅，但往往等到知道如何消滅新的病蟲之後，新的變種病毒又開始在網路上蔓延了[3]；其中又以圖 1. 所示之 DDoS 攻擊對網路衝擊最為巨大。許多攻擊發作所利用的弱點都是事先早已公佈開來的，要在與攻擊者的戰爭中取得主動優勢，預防的機制和事

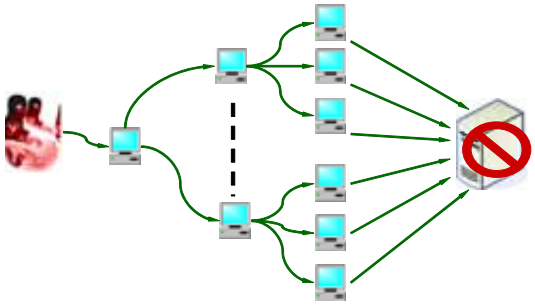

圖 1. DDoS 攻擊示意圖

前的準備工作將是相當重要的一環。

隨著技術進步，網路頻寬變大，硬體速度提升，DoS 攻擊只靠一台電腦已經無法造成目標資源耗盡，因此演變出 DDoS 攻擊模式[5][6]。DDoS 攻擊之所以成功，主要因為網路散佈太多安全性薄弱的電腦主機，使得攻擊者輕易取得大量的攻擊跳板，DDoS 攻擊所控制的攻擊來源更多，攻擊跳板也可能分為很多層級，攻擊者遙控這些電腦同時發動攻擊，造成的網路衝擊遠比 DoS 攻擊來得更大。除此之外，攻擊者可以使用 IP 偽裝 (IP spoofed) 技術或利用反射器 (reflector)[1]技巧，使得追查與防堵攻擊來源困難度增加。一個沒有任何防護措施的伺服器，可能在數分鐘內就被 DDoS 攻擊癱瘓，所以抵禦 DDoS 攻擊於事前作預防的措施的概念很重要。在攻擊發生時，減低防禦機制失效之可能性並過濾異常行為的封包。本論文的目標就是要設計一整合性的防禦系統運用在抵禦 DDoS 攻擊，使得網路服務能夠繼續正常運作。

本論文將於第二節對階層式聯合防衛 WallAgent、WallGuard, Domain dispatcher 各子系統之功能與設計作一介紹，而於第三節介紹系統實作部分，第四節為系統之測試，最後於第五節討論本系統之結論。

2. 階層式聯合防衛系統之功能與設計

圖 2.表示階層式聯合防衛系統架構，分成三個子系統 WallAgent, WallGuard 和 Domain dispatcher 構成，分別詳細介紹如下：

2.1 WallAgent

安裝於用戶端主機之應用軟體，負責監控主機網路資源之使用，並對於先前定義的異常偵測規則做出反應。同時與 Domain dispatcher 交換協防訊息。階層式聯合防衛系統中是屬於最基層的元件。WallAgent 的模組功能架構由五個模組組合成： (1) 註冊管理模組 (Registration Management Module) 用來發出請求到 Domain dispatcher 做註冊動作，之後若有更新政策訊息或者聯合防衛訊息才信任已註冊過的 Domain dispatcher；(2) 攻擊偵測模組 (Attack Detection Module) 會監視系統的資源使用狀態，這些資訊主要用來判別異常行為的參數，例如封包的傳輸量、CPU 使用率或記憶體使用率。並對所有的監控的封包與內建的攻擊規則做比對，當符合攻擊的特徵時會產生記錄檔並執行回應動作；其中特徵比對使用 Snort；(3) 防火牆模組 (Firewall Module) 使用防火牆規則決定流經封包是否同意其通過；(4) 政策管理模組 (Policy Management Module) 管理自身的防火牆規則與異常偵測規則。並接受 Domain dispatcher 的更新政策訊息。(5) 政策決定模組 (Policy Making Module) 藉由事先定義的規則做決策，例如偵測模組發出異常現象的警報，此模組會對防火牆下達命令來阻絕攻擊，並且事先定義的規則可以依實際的需求可能會有不一樣的設定。

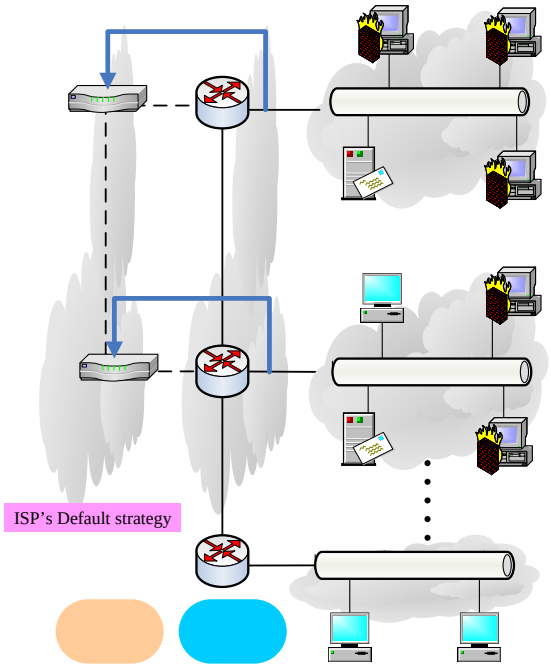

圖 2. 階層式聯合防衛系統架構圖

2.2 WallGuard

為一般應用軟體，安裝主機上或者整合到硬體網路設備中，若是安裝於主機須與網路設備相連接並擁有其管理權限可以做流量限制與監測。負責全域的協防工作，從攻擊目的端之 WallGuard 將聯合防衛訊息傳播開來，使在攻擊來源的 WallGuard 可以協助將訊息抵擋。同時對於內部的所屬的 WallAgent 主機可以透過 Domain dispatcher 下達聯合防衛訊息。在階層式聯防偽系統中是屬於最高層的元件。WallGuard 的模組功能架構由六個模組組合成：(1) 註冊管理模組 (Registration Management Module)用來發出請求到 WallGuard 做註冊動作;(2) 攻擊偵測模組 (Attack Detection Module) 會監視遠端管控的網路設備，如路由器或交換機，這些資訊主要用來判別異常行為的參數，例如封包的傳輸量、CPU 使用率或記憶體使用率。並對所有的監控的封包與內建的攻擊規則做比對，當符合攻擊的特徵時會產生記錄檔並執行反應動作；其中特徵比對使用 Snort；(3) 防火牆獲控制器模組 (Firewall/Controller Module) 使用防火牆規則決定流經封包是否同意其通過；(4) 政策管理模組 (Policy Management Module) 管理自身的的防火牆規則與異常偵測規則。並接受其他信任之 WallGuard 的更新政策訊息。(5) 政策決定模組 (Policy Making Module) 藉由事先定義的規則做決策，例如偵測模組發出異常現象的警報，此模組會

對防火牆下達命令阻絕攻擊，並且事先定義的規則可以依實際的需求可能會有不一樣的設定。政策決定模組的請求對象可能在遠端，例如 WallAgent 下達防火牆指令是於自己主機上，而對於 WallGuard 就可能直接對路由器下達命令 (6) 紀錄預防模組 (Trace Prevention Module) 分析系統記錄檔之預防演算法產生需要被觀察的名單。

2.3 Domain dispatcher

為一般應用軟體，安裝之主機必需與管理之 WallAgent 相同網域。負責提供 WallGuard 與 WallAgent 聯防界面，當 WallAgent 非常多的時候可以利用 Domain dispatcher 劃分多個子區域，主要功能降低 WallGuard 直接管理大量 WallAgent 的複雜性與負擔，同時也可以給予各個子區域不同之執行政策。在階層式聯防偽系統中是屬於中間層的元件。Domain dispatcher 模組由兩個模組組合而成：(1) 註冊管理模組 (Registration Management Module) 用來發出請求到 WallGuard 做註冊動作，並會幫忙 WallGuard 遞送更新政策訊息或者聯合防衛訊息。另外也會負責接受 WallAgent 的註冊訊息；(2) 政策管理模組 (Policy Management Module) 管理區域下之 WallAgent 的規則，保持 WallAgent 的防禦規則是符合區域的政策。

階層式聯合防衛系統之訊息傳遞機制主要的分類成註冊訊息、更新政策訊息和聯合防衛的訊息三種。以下是系統訊息傳遞的說明：

2.4 註冊訊息傳遞

聯防系統都是一個元件以上組成，註冊是為了讓彼此之間相互知道對方的存在，目的為加強子系統間安全性。

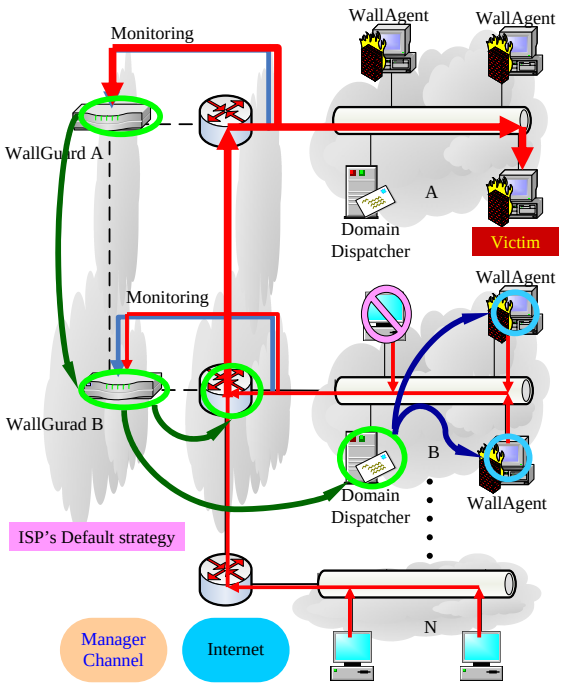

圖 3. 聯合防禦訊傳遞示意圖

2.5 更新政策訊息

WallGuard 和 WallAgent 的決策都是利用一些事前定義好的規則來檢驗，例如防火牆的規則和入侵偵測皆是利用特徵化的規則比對所有的封包，當封包特性符合規則時將會觸發偵測事件產生。

2.6 聯合防禦訊息

該訊息有別一般的情況，這個訊息平時並不會產生，只有當系統偵測出攻擊或者經過預防策略監控之後需要發動聯防才會產生。這個訊息主要是傳遞判斷的規則與相對應的動作規則。當偵測系統能夠分析出攻擊封包的特徵時，那麼就可以將此特徵傳遞給其他協防的系統，同時可以告知如何處理。圖 3.為例表示攻擊發動被 WallGuard A 偵測到後，WallGuard 經由判斷後發現流量來自外部，接著將協防訊息傳送到 WallGuard B，WallGuard B 如果發現攻擊流量時，會先將攻擊流量過濾，接著再將訊息傳給底層之 Domain dispatcher 讓協防訊息傳到子網路下的 WallAgent。

另外本論文提出分析系統紀錄之預防演算法，此演算法是屬於 WallGuard 子系統功能之一。利用攻擊者會找尋遭受感染的來源特性，進而感染源會被操控找尋更多的受害者，並將攻擊的工具散播下去，這些被感染源接觸的對象群組稱作通聯清單 (Contact list)。本演算法在已知感染源是誰的情況下 (感染自行通報、網管人員發現或系統紀錄等)，利用系統記錄檔盡可能找尋所有通聯清單，如郵件病毒會在攻擊散播郵件時在郵件伺服器保有紀錄，或者攻擊者進入系統可能會留下未清掉的連線資訊。通聯清單中所有嫌疑者都有可能繼續感染其它受害者或者發動 DDoS 攻擊。因此本演算法針對這些嫌疑者加以分類與監控，進一步讓更有嫌疑的人用過濾或限流加以管制，預防嫌疑者將病毒感染他人或發動 DDoS 攻擊。分析系統紀錄之預防演算法先從已知的感染者，輸入相關的系統記錄檔，找出曾經與使用者接觸之對象，這些對象都可能會遭到攻擊者入侵或被植入後門，之後 WallGuard 系統將這些接觸對象的狀態設為嫌疑者加入到通聯清單，如果通聯清單有找出嫌疑者，將對其進行監測，WallGuard 會試圖檢查嫌疑犯是否有異常的行為，如有開啟有嫌疑的通訊埠 (MaybeSlave) 或不合法執行程序存在 (Slave) 時，WallGuard 將會發出政策更新訊息給 Domain dispatcher 傳給 WallAgent 自行隔離處理，若該主機

沒有安裝 WallAgent，WallGuard 將會主動控制路由器將其隔離。WallGuard 將會一直監控通聯清單中的嫌疑者，嫌疑者可設定觀察一段時間之後變成 Normal 的狀態，並移出通聯清單。另外如果使用者不在線上時，或無法偵測到時會以 Offline 的狀態顯示。

3. 系統實作

本系統實作於國立中央大學，目前仍在持續發展與實驗階段。實驗網路環境架構如圖 4.所示；其中 L 字母開頭為用戶端，共分成三個虛擬子網路，安裝 WallAgent 子系統；R 字母開頭為路由器，負責轉送封包；P 字母開頭負責建立網管網路，並安裝 WallGuard 子系統。除此之外，由於本實驗網路

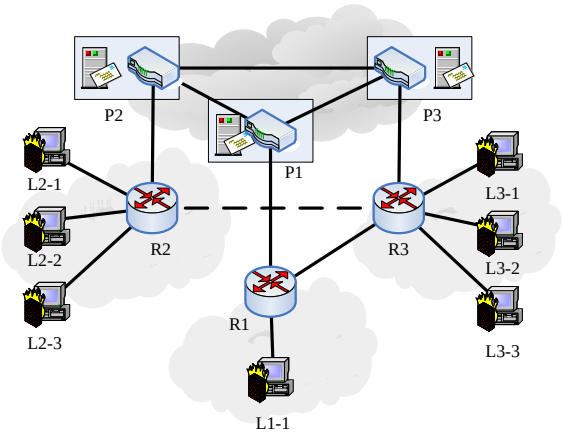

圖 4. 實驗網路架構圖

的環境不大，所以將 Domain dispatcher 與 WallGuard 安裝在同一台主機上。本論文建構出來的實驗網路設備，共用了 13 部個人電腦主機；其中 7 部主機為一般主機，L1-1, L2-1, L2-2, L2-3, L3-1, L3-2, L3-2, L3-3 分別裝 WallAgent，運算處理器為 Intel Celeron 2.0 Ghz，記憶體 DDR 256MB；三部主機為 P1, P2, P3 並裝 WallGuard 與 Domain dispatcher，運算處理器為 Intel Pentium(R) 4 2.8 Ghz，記憶體 DDR 256MB；3 部主機為 R1, R2, R3 做為路由器，運算處理器為 Intel Pentium(R) 4 2.8 Ghz，記憶體 DDR 512MB；以上作業系統皆為 Redhat 9.0；另外有一台 24 埠的 SMC6724A-TWN Switch 串接所有的電腦。之所以建立實驗網路環境，一方面可以防止攻擊流量影響現有的網路，另外一方面可以依需求改變實驗網路架構提高實驗的彈性。

使用者可以透過子系統提供的圖形介面操作，先將系統的設定檔初始化，將欲信任主機填於設定檔中，並且於 WallAgent 與 WallGuard 設定是否啟動發現攻擊之後自動回應。圖 5.為聯合防衛系統操作示意圖。當系統發出協同防禦 DDoS 攻擊時，其執行的步驟如下：

Step 1. 系統使用者對 WallGuard 進行初始設定，管轄區域下的 Domain Dispatcher IP 位址與整個區域協防 WallGuard IP 位址。

Step 2. 當內部攻擊發生時，WallGuard 產生聯合防衛訊息給所管轄 Domain dispatcher。

Step 3. Domain dispatcher 收到來自 WallGuard 的聯合防衛訊息之後，會再將該訊息傳到所管轄的 WallAgent。

Step 4. WallAgent 處理聯合防衛訊息無錯誤發生後，回應一個成功的訊息。

Step 5. Domain dispatcher 派遣聯合防衛訊息至其管轄之 WallAgent 完成之後，會回應一個工作完成的訊息

Step 6. 當外部攻擊發生時，WallGuard 產生聯合防衛訊息給鄰近之 WallGuard。

Step 7. WallGuard 處理聯合防衛訊息無錯誤發生後，回應一個成功的訊息。

Step 8. 使用者可以從 WallGuard 觀看聯防後的結果，由使用者介面呈現。

圖 6.為 WallAgent 進入畫面，Resource Monitoring 頁籤是監測本機系統狀態。Resource monitoring configuration 為設定 SNMP 代理人相關資訊，由於 WallAgent 是監測本機端，所以該項 Monitoring IP Address 使用預設即可，但要填入本機端之 SNMP community 以便取得系統資源參數。使用者需要於 Monitoring Interface configuration 下拉選單選擇監視的介面名稱。System status 顯示監視主機之統計圖，目前可以監視流量傳輸率、頻寬傳輸率、CPU 使用率以及記憶體使用率。圖 7.為

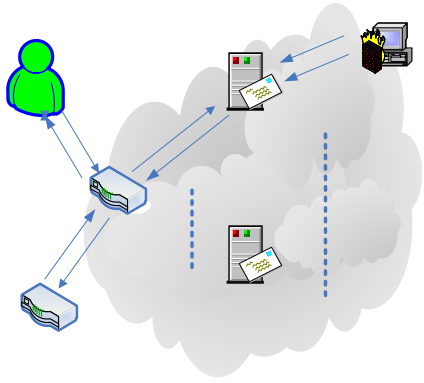

圖 5. 聯合防衛系統操作示意圖

Abnormal Configuration 頁籤是提供設定偵測異常的門檻值與觀看入侵事件。在此頁籤的下面兩個文字方框分別會顯示 Snort[9]的警報內容與分析 snort 的警報內容後的攻擊排行。Firewall Configuration 用來管理主機內部防火牆的狀態，在這個頁籤可以增加防火牆規則，重讀防火牆設定檔以及移除防火牆規則三個操作。Registration Page 設定相互信任的主機，WallAgent 註冊設定檔內容會預設尋找 wallagent.conf 的檔案。在 wallagent.conf 只要填入信任之 Domain dispatcher 的 IP 位址即可。為 Events 可以查看 WallAgent 系統運作時產生的事件。事件有三個基本欄位，分別是 Timestamp、Type 與 Action。Timestamp 為事件發生的時間，型態有分

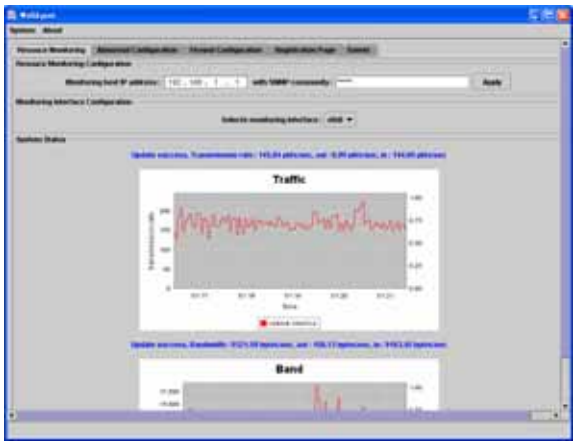

圖 6. WallAgent 之資源監控畫面

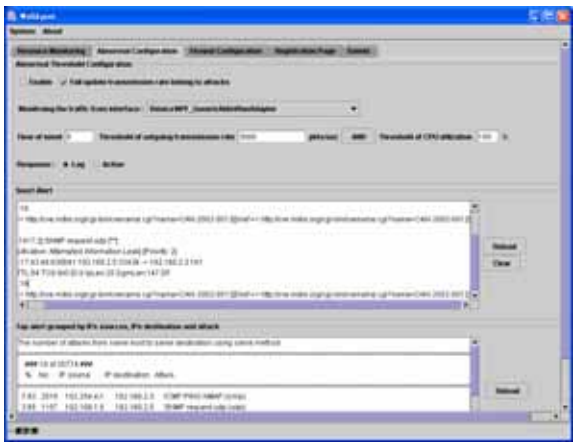

圖 7. WallAgent 之異常偵測設定畫面

INFO 表示系統資訊，ALARM 表示異常事件發生，如異常狀態被偵測出來時與協同防衛傳遞訊息的事件等。Action 描述該事件的行為動作。

WallGuard 的使用介面與功能大致與 WallAgent 相同，本節只提出不同之處介紹。Device Monitoring 的頁籤與 WallAgent 的 Resource Monitoring 操作相類似，也是 WallGuard 進入畫面；但不同的是 WallGuard 有可能是監視所管理之路由器，因此必須在 Monitoring device IP address 填入管理之路由器 IP 位址與 SNMP community。WallGuard 之 Registration Page 可以填入多個可信任之聯合防衛 WallGuard 與管轄之 Domain dispatcher 的 IP 位址，預設的設定檔為 wallguard.conf。分析系統紀錄畫面，表格列出通聯名單的使用者，該名單目前是針對 sendmail maillog 所產生。Rescan period 表示通聯名單掃描的間隔時間。

Domain dispatcher 顯示子系統 Domain dispatcher 之 Registration Page，預設設定檔為 dc.conf 同時它需要輸入可信任之 WallGuard 及管轄內的 WallAgent 的 IP 位址。它會開啟一個服務，等待 WallGuard 的訊息通知，傳遞訊息給底層之 WallAgent。

4. 系統測試

在本章節，以實驗驗證本系統之功能及預防演算法之機制。

實驗：階層式聯合防衛系統運作測試

本實驗目的為驗證 WallGuard 與 WallGuard 間協防機制是否正確地啟動，且是否能夠有效偵測出攻擊並能控制路由器將流量過濾在 WallAgent 端。選擇以 L2-1, L2-2, L2-3, L3-1, L3-2, L3-3 主機攻擊 L1-1 網頁伺服器服務，並且監測 Switch 上攻擊封包途徑對應埠之流量，於 WallGuard、Domain dispatcher 和 WallAgent 反應之後，再以網頁瀏覽器確認 L1-1 是否可以回應請求。圖 8.分別為聯防系統測試偵測異常時刻與反應時刻分佈圖。WallGuard 防禦機制啟動前使用連覽器瀏覽會出現無法連線之訊息。防禦機制啟動之後，可以瀏覽到網頁達到抵禦的效果。圖十八數據顯示 P1 之 WallGuard 偵測到異常的時刻平均差 7 秒，除了有網路的延遲可能產生誤差之外，還有 JAVA 本身的程式執行的週期

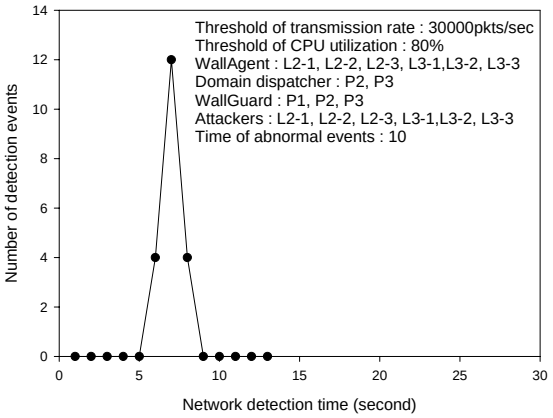

圖 8. 聯防系統之異常偵測時刻分佈圖

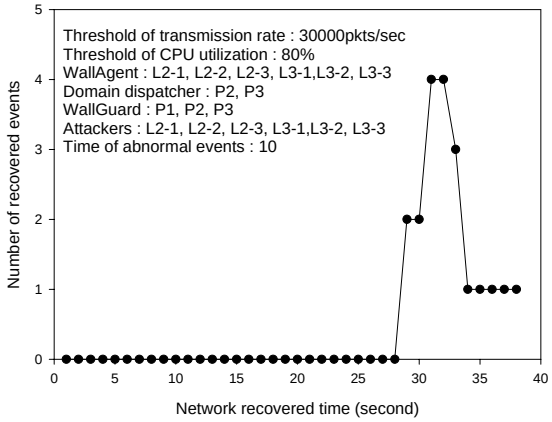

圖 9. 聯防系統之反應時刻分佈圖

誤差。而這個實驗 P1 之 WallGuard 的事件發生次數設定為 10 次的，在於整個網路的恢復可能需要較長時間。圖 9.顯示 WallAgent 從偵測到反應攻擊(下達防火牆指令) 需要平均 33.9 秒。主要在於事件發生次數之設定會拉長系統的反應時間。

5. 結論

為了因應近年來網路攻擊事件不斷增加，本論文探討 DDoS 攻擊的聯合防衛系統的議題，針對如何於攻擊發生之前做預防機制。首先分析 DDoS 攻擊事件流程，了解來源端發動異常的流量封包到受害者端的手法與造成可能的傷害。從網路管理的角度遇到攻擊事件發生時的解決步驟，分別觀察現有

的網路環境有幾種類型的抵禦位置，找尋最佳的防線。驗證與實作提出階層式聯合防衛與預防系統架構，提升現有的網路環境安全性。最後提出預防的機制，使用分析系統記錄的演算法降低監測網路的流量，針對郵件紀錄提供實作並與原有的方法比較，確實減少偵測的負擔。

本論文使用了區域聯防的概念抵禦 DDoS 攻擊，同時也利用分析系統紀錄的演算法達到預防的效果。設計了三個子系統 WallGuard, WallAgent 和 Domain dispatcher 階層式的相互合作抵禦 DDoS 攻擊。WallGuard 負責彼此間全域聯防工作及轉送協助抵禦攻擊之訊息。同時利用區域劃分的概念，WallGuard 可以透過所管轄之 Domain dispatcher 通報子網路下的 WallAgent 共同防衛 DDoS 攻擊，讓 WallGuard 與 WallAgent 組成一個協防體系，對於日後加進新的抵禦 DDoS 攻擊技術更為變化。並且本文提出的預防演算法，達到攻擊者可以在發動 DDoS 攻擊之前將其隔離，對於掃描主機之網路負擔也會降低。

參考文獻

[1] BroadWeb, *http://www.broadweb.com.tw*

[2] Byeong Kil Lee and Lizy John, "NpBench: A Benchmark Suite for Control Plane and Data Plane Applications for Network Processors," *Proceedings of the International Conference on Computer Design*, San Jose, Oct. 2003.

[3] CERT/CC Overview Incident and Vulnerability Trends, CERT Coordination Center, Pittsburgh, *http://www.cert.org/present/cert-overview-trends/*, 2002.

[4] Computer Emergency Response Team Coordination Center, CERT/CC, *http://ww.cert.org*.

[5] D. Moore, G..M. Voelker, and S. Savage, “Inferring Internet denial-of-service activity”, *Proceedings of 10th USENIX Security Symposium*, Washington, DC, May 2001.

[6] J. Mirkovic, G. Prier and P. Reiher, “Attacking DDoS at the Source,” *Proceedings of International Conference on Network Protocols*, Paris, France, pp. 312-321, Nov. 2002.

[7] Jelena Mirkovic, Janice Martin, and Peter Reiher, “A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms,” *UCLA Technical Report #020018*, 2002.

[8] Jelena Mirkovic and Peter Reiher, “A Taxonomy of DDoS Attack and DDoS Defense Mechanisms,” *ACM SIGCOMM Computer Communication Review*, Vol. 34,, No. 2, April 2004.

[9] M. Roesch, “Snort - Lightweight Intrusion Detection for Networks,” *Proceedings of the 13th Systems Administration Conference (LISA'99)*, Seattle, Washington, pp. 229-238, Oct. 1999.

[10] Rocky K. C., Chang, “Defending against flooding-based distributed denial-of-service attack: a tutorial,” *IEEE Communication Magazine*, vol. 40, pp. 42-51, Oct. 2002.

[11] Taiwan Computer Emergency Response Team Coordination Center, TWCERT/CC, *http://www.cert.org.tw*