國立臺灣大學「台大工程」學刊　第九十期　民國九十三年二月　第 127–136 頁
Bulletin of the College of Engineering, N.T.U., No. 90, February 2004, pp. 127–136

結合分散式防火牆與代理伺服器技術之安全性文件存取設計

SECURE DOCUMENT ACCESS ARCHITECTURE BASED ON DISTRIBUTED FIREWALL AND PROXY TECHNOLOGIES

潘 啟 諫[*]　楊 凱 翔[*]　李 肇 林[†]
Chi-Chien Pan[*]　*Kai-Hsiang Yang*[*]　*Tzao-Lin Lee*[†]

[*]博士班研究生　[†]副教授
國立台灣大學資訊工程學研究所
[*]Ph.D. student　[†]Associate Professor
Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan 10617, R.O.C.

Abstract

With the rapidly development of the Internet, the Intranet topologies of enterprises are getting more and more complicated and distributed, and therefore the security problem of the Intranet is gradually valued. Enterprises usually deploy firewalls to defense the outside attacks, however novel viruses and Trojans use the tunnel techniques to communicate with outside machines via the valid ports, and this attacks can pass the firewall to make the Intranet easy to be attacked. Besides, attacks from the inner enterprise members also break the Intranet security. On the other hand, document access and sharing is the primary behavior in each enterprise, and relates to the operation and development of enterprise. It is an important issue to ensure the document security and prevent any possible attacks from the Internet or Intranet. This paper focuses on the document security in the enterprise Intranet, and proposed one security architecture which combines the distributed firewall technology for filtering and control packets, and the proxy server for application level access control. Besides, the proposed architecture also applies the Secure Socket Layer (SSL) and Virtual Private Network (VPN) technique to secure the connection, and controls the network behavior according to different user applications in order to build a multi-level secure document access mechanism.

Keywords: firewall, distributed firewall, proxy, document access, network security.

摘　要

由於網際網路的快速發展，使得企業組織所建構的內部網路日趨複雜且所及範圍增加，內部網路的安全性問題逐漸受到重視。以往企業組織運用建置防火牆來抵擋外部可能之攻擊行爲，但是新穎的病毒及木馬等惡意程式運用網路穿隧技術，藉由標準通訊埠對外溝通，此種技術可以突破防火牆之限制，使得內部網路容易遭受攻擊。此外來自內部惡意員工的攻擊也會破壞內部網路的安全性。另一方面，文件的存取與分享已成爲各企業組織主要的內部活動行爲，關係整個企業組織的運作與發展，如何能夠確保其安全，避免遭受可能的網路攻擊行爲，便成爲主要的研究課題。本論文研究以企業組織內部網路的文件安全性爲範圍，結合分散式防火牆技術在封包層次的過濾與控制功能，以及代理伺服器在應用層次的存取分析與控制功能，輔以傳統的 SSL 安全連線及 VPN 虛擬私有網路技術，並以使用者的應用爲導向，來控制網路行爲，以形成一個多層次的安全性文件存取機制。

關鍵詞：防火牆、分散式防火牆、代理伺服器、文件存取、網路安全。

1. 緒　論

隨著網路的快速發展，愈來愈多的企業組織皆利用網路的便捷與分散處理的優點，來建立企業內部的應用系統，以增加工作效率及提升競爭力。而目前此方面需求大多透過網際網路以及虛擬私有網路 (Virtual Private Network，以下簡稱 VPN) 等技術來構成，兼具節省成本及良好的擴充彈性。然而在此架構下，整體網路環境範圍比以前廣泛許多，網路裝置可能分散在遠處，安全性是首先必須考量的重要因素。

由於原有 TCP/IP 網路通訊協定本身就存在著一些先天設計之缺陷 [1]，缺乏考量相關安全機制的問題，因此造成許多網路駭客 (Hacker) 的可乘之機，利用各種假造、攔截、監聽等方式，竊取企業組織中的各種重要之資訊。此外知識經濟已成爲現代世界經濟之主流，知識是未來各企業組織主要的競爭力量，而文件是其呈現、記錄及傳遞之主要媒介，文件機密往往涉及商業競爭，並同時影響企業整體生存之關鍵。如何能夠透過網路之便利性使得內部成員能夠方便、有效地使用與分享文件且兼顧安全性，已成爲各企業組織一項迫切的需求與問題。

在網路安全的領域方面，以往許多研究大多是著重在防禦外部駭客的攻擊。典型做法是建置企業防火牆來區隔內部與外部網路，藉由完全地隔離來防護內部網路安全，然而安全性的威脅卻仍然存在。其一是由於病毒 (Virus)、木馬 (Trojan) 的技術發展，駭客首先藉由電子郵件 (E-mail) 或其他作業系統漏洞等方式入侵內部裝置，再由內部裝置利用一些穿隧 (tunnel) 技術向外穿透防火牆，造成內部網路的暴露，讓外部網際網路可以直接存取到內部網路。其次是威脅的行爲，來自於公司內部的員工，由於對公司制度或上級主管的不滿等種種因素，而進行威脅企業文件安全之行爲。另外，企業組織中眞正重要的安全性文件與資訊通常也不會對外界公開，而是置於企業組織內部，僅提供具有權限之內部員工來存取，因此，問題的關鍵在於內部安全，而不是一般對外的網頁資料。同時在網路安全的特性部分，內部網路所需要的安全措施與外部網路的情形也不相同，由於一般企業組織外部都設置有防火牆，提供基本的封包過濾功能，增加外部直接攻擊的困難度，相對地內部網路往往不受限制，可進行的攻擊行爲及種類較外部網路多。但是外部網路其存取端行爲較難去控制，而內部網路因爲範圍較小，比較有機會可以控制所有網路存取端的行爲。

現有的網路安全性機制常見的有安全連線 (Secure Socket Layer，以下簡稱 SSL) 與 PPTP、IPSec 等 VPN 技術，這些技術強調連線中途的安全，可確保封包在傳送途中不會遭受監聽竊取、竄改等，然而對於連線兩端點之安全性卻完全無法保障。如果駭客先利用病毒、木馬等技術先行植入端點，來進行遠端監看、操控的行爲，則原有的安全機制完全無效。此外，許多攻擊行爲基於 TCP/IP 協定天生的缺陷，如阻絕服務 (Denial of Service，以下簡稱 DoS)、中間人攻擊 (Man-in-the-Middle Attack) 等，相關安全機制亦無法發揮作用。

另外傳統的防火牆技術是採取保護企業組織網路進入點的基本做法，其主要的目的是在於防止 Internet 外部的攻擊進入，但其模式之前提是假設位於內部網路的所有成員都是善意的，但是事實上多數的攻擊行爲來自內部。因此，單純的外部防火牆技術對於問題並沒有太大的幫助。

本論文研究範圍主要著重於企業組織內部文件存取安全，針對前述各項問題及可能之攻擊方式，予以適當防護。各項研究目標包括：

1. 防止掃描攻擊：掃描攻擊是駭客發起攻擊的準備，如果能夠有效避免主要伺服器遭受掃描探測，就可以減少被攻擊的機會。
2. 防止重要文件遭監看、篡改：保護文件在存取過程中之機密性 (Confidentiality) 與完整性 (Integrity)。
3. 防止中間人攻擊：中間人攻擊係利用各種假造技術，來欺騙達到從中竊取密碼或資料的目的。
4. 防止阻絕服務：避免主要伺服器遭受大量無效連線或封包佔去資源，而導致無法運作。
5. 防止木馬監控：木馬攻擊爲最近常發生的攻擊手法，利用木馬監控使用者裝置，攔截鍵盤輸入及螢幕輸出，以取得使用者帳號密碼，及所存取的文件資料。
6. 簡化使用者負擔：各項設計必須兼顧安全性與便利性，儘量減少軟體的設定與安裝，避免造成使用者不便及負擔，而降低使用者配合的意願或破壞安全性裝置。

爲了能夠達成上述目標，單憑任一種安全性機制之技術是絕對無法達成。現有的各項網路安全性機制往往個別獨立運作，缺乏彼此溝通整合，無法做到垂直不同層次的完全控制，然而攻擊行爲卻是只要突破一處就足以造成整個安全性瓦解。因此，在系統架構上我們結合了分散式防火牆 (Distributed Firewall) 與代理伺服器 (Proxy) 的技術，輔以傳統的 SSL 及 VPN 網路安全機制。

分散式防火牆技術兼有一般企業防火牆與個人防火牆之優點，具有集中式控管存取規則的機制，能夠即時地反映狀況，並有效地管理及限制個別機器的網路存取行爲。因此利用分散式防火牆對於網路各端點進出封包的過濾以及所有連線之控制，便可以確保所有連線端點上之安全，並簡化使用者的設定。在代理伺服器技術方面，以往多用於增進網路存取效能，但由於其做爲應用層面的通道，也可以作爲應用層面的控制點，掌握高階服務的動向。我們結合這兩種不同的安全性控制機制，以分散式防火牆技術進行網路各端點封包層次的過濾及連線控制，以代理伺服器技術來提供高階應用服務之需求分析與控制。

本論文架構大綱如下：第二章敘述各相關研究領域與技術發展之現況，包括可能之攻擊、防火牆、代理伺服器與相關安全性連線機制等，第三章敘述整個

系統架構及運作方式，第四章敘述整個系統的安全性機制設計及如何有效抵擋各種可能之攻擊，第五章敘述實作的細節與相關考量，最後是結論與後續研究發展之方向。

2. 相關研究

2.1 常見的攻擊類型

目前企業組織多利用內部網路 (Intranet) 來建立 Web-based 系統作爲彼此溝通及文件資訊交換分享的平台，使用者透過瀏覽器等支援標準 HTTP 通訊協定 [2,3] 的工具來連結存取。然而此架構常面臨許多攻擊，常見攻擊類型包括掃描攻擊、中間人攻擊、阻絕服務攻擊、木馬程式等，如圖 1 所示：

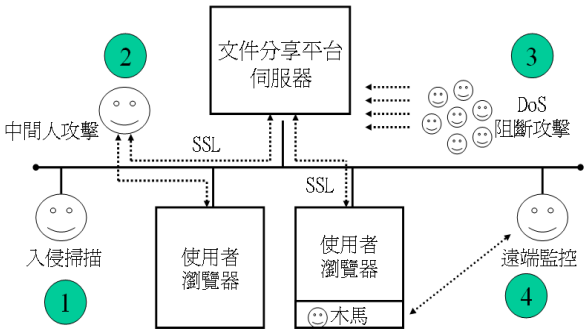

圖 1 四種常見攻擊類型

2.2 防火牆、個人防火牆、分散式防火牆

爲了能夠充分運用相關技術來滿足安全性需求，我們首先必須針對傳統防火牆、個人防火牆與分散式防火牆作一比較分析，作爲規劃整個系統架構之基礎。此研究方面以往已經有許多的研究文獻作歸納探討 [4~6]。另外根據 NIST 美國國家標準 800-41 [7]，一般國家或是企業組織需要安裝不同等級的防火牆來加強安全性，利用 Rule-based 的過濾規則，指定合法的來源或目的地 IP 位址、來源或目的地的 port，避免遭受非法的網路攻擊。

在傳統防火牆與分散式防火牆比較方面：

1. 傳統防火牆是利用控制網路進出的進入點，以達到控制整個網路上進出封包的目的。在此種防火牆環境下，內部網路成員是完全被信任的，然而實際上有許多攻擊行爲都是由內部發起，而傳統防火牆無法控制。然而在分散式防火牆環境下，組織內的每一台機器都安裝了防火牆，而其過濾管理的規則是由另一台主機作集中式的控管。各個使用者機器開機後即處於防護狀態，不論來自 Internet 或 Intranet 的封包都會被阻擋掉。
2. 內部機器利用 VPN、tunnel 技術以及點對點加密技術，可以從內對外建立連線，造成外部網路能直接通過傳統防火牆進入 Intranet，而完全無法阻擋。另一方面，分散式防火牆因爲在每一台機器上都加裝防火牆，即使使用上述技術進入網路內，也無法對其他機器造成影響。
3. 傳統防火牆的封包過濾控制主要是以 IP、MAC 位址、Port 等作爲控制規則之條件，缺乏足夠的完整資訊。而分散式防火牆如同個人防火牆一般，可以提供更細緻的控制程度，例如以應用程式爲導向的控制規則。

另外在個人防火牆與分散式防火牆的比較部分：

1. 個人防火牆是安裝在每一台機器上，但是規則也必須每一台的使用者自行設定，缺乏一個集中管理之機制。就整體網路而言，控制規則的變更需要每部機器逐一更新設定，非常麻煩。而分散式防火牆則將控制規則集中處理，使用者不必再自行控制，而由網路管理者統一設定。
2. 個人防火牆的安全性規則是以機器的 IP 爲主，而分散式防火牆則可以動態的以使用者爲主，作爲控制的策略。例如使用者雖然在不同的機器 login，但是仍然可以擁有一致地動態安全性設定，獲得其所被允許的相同規則。
3. 分散式防火牆如同傳統防火牆一般，可具備集中的 log 紀錄及計算統計的功能，而個人防火牆則缺少此部分。

儘管分散式防火牆與傳統防火牆及個人防火牆比較，分散式防火牆具有相當多優點，但在應用方面卻仍具有一些需要克服的問題，包括：

1. 對於能具有 administrator 權限的使用者的環境，使用者可能會 uninstall 或是將分散式防火牆關閉，則那台機器的安全性便無法確保。不過在我們的研究中，一般組織中的機器可以限制使用者不具有 administrator 登錄權限，來避免此狀況的發生。
2. 對於新加入網路的設備不容易保障其安全性，例如內部人員私下使用可上網的裝置，如 Personal Digital Assist (PDA) 等。另外，因爲單純的分散式防火牆無法加以控管，尤其是在具備無線網路的環境，這時就必須依賴其他網路安全的機制輔助，例如傳統防火牆等。在我們的研究環境下，由於主要是保護其他機器可以安全地連上主機端存取文件，並不需要保障其他機器的安全性。
3. 封包過濾的工作由個別的機器來進行，勢必對每台機器造成一些系統額外的負擔。

在技術方面，建立分散式防火牆必須綜合傳統防火牆與個人防火牆之許多技術，包括防火牆的存取規則設計，個人防火牆的封包過濾技術等。例如在 Windows 下的個人防火牆開發技術包括許多細節，Vadim V. Smirnov [8] 對於 Windows 下製作個人防火牆的各種技術做了完整的分析與比較，透過其中的內容我們可以選擇合適的實作方式與技巧。另外包括 Windows DDK 等驅動程式開發技術 [9]，Network Driver Interface Specification (NDIS) 封包收送等也是必備的知識技術。

2.3 代理伺服器之功能

代理伺服器 (Proxy server，以下稱 proxy) 是網路下載網頁的中間人。當使用者要去瀏覽網頁時，如果設定透過 proxy，則 proxy 會代替使用者機器去抓取網頁，之後再回傳網頁給使用者端。使用 proxy 優點包括：

1. 下載網頁速度加快：因爲 proxy 代替使用者去抓取網頁，便會儲存一份在 Client 端，以便後續其他使用者需要同樣網頁時，proxy 便可以直接回傳所需要的網頁，而不用費時間去重新抓取，因此速度會加快。
2. 增加下載續傳支援：如果 proxy 支援續傳，當目的地伺服器不支援續傳時，而 proxy 就可以將 cache 內的資料以續傳方式提供給使用者端。
3. 上網更安全：一般透過 Internet 傳送出資料，就一定會讓伺服器知道你的 IP，如果使用 proxy 主機傳送資料到企業外部的伺服器時，就不會暴露出實際的 IP 位址。

Proxy 主要的功能就是代理使用者端的動作，一方面讓他可以速度加快，另一方面就是爲了增加使用者端的安全性。而在我們的研究架構下，使用 proxy 的另一個原因是希望提供在應用層次的存取分析與控制功能，並且同時擔任區域上的控制端，確保只有合法的使用者才可以透過 proxy 存取到重要文件。

2.4 相關保護機制

在文件系統的安全性方面，一般研究方向可以分爲下列幾類：

1. 保障文件檔案在主機上的安全性：對於文件儲存在 Web 主機上的安全性問題，有不少研究提出解法，一般使用文件加密的方法，包括使用者對文件的加密，或是 Web 主機儲存檔案時的後加密方式，另外，重要的文件也可不以檔案的方式來儲存，更可以加強文件在主機上的安全性。
2. 保障 Web 伺服器主機的安全性：對於 Web 主機的安全性而言，主要是針對防止 DoS、DDoS 之類的攻擊，也已經有許多不同的研究提出解法 [10~14]。
3. 確保文件的存取權限控制：一般 Web 網站的安全性大多使用基本 HTTP [2,3] 的驗證方式，此種方式會在 Client 端跳出一個要求輸入帳號密碼的視窗，待使用者輸入帳號密碼後，傳送給 Server 以決定是否有權限可以開啓此文件。此種驗證應用於 Web 目錄上是很快速而有效，但仍有其缺點：
   - 此種方式只能抵禦外來的接觸嘗試，無法保護本機的 Web 目錄不會被未使用 Client 程式的其他本機使用者，直接透過檔案系統或其他服務存取文件資料。
   - 在預設的情形下，此種系統並不提供密碼鎖定的機制，因此反而易引起持續、反覆或粗暴的攻擊。攻擊者可以隨意嘗試各種使用者名稱及密碼而不受任何限制。
   - 此機制最大的弱點在於其密碼是以未加密的編碼格式傳送的，因此攻擊者可以利用 sniffer 等程式偷聽到密碼。

   除了此種機制外，Apache 也支援使用 MD5 [15] 的摘要式密碼編譯驗證法。利用 MD5 演算法將密碼轉換成 128bits 的訊息摘要後，傳送給 Web 主機做確認，則可以避免被偷聽的攻擊產生。
4. 保障文件在傳輸上的安全性：對於文件在網路上傳輸的安全性方面，有相關研究 [16~18] 提出使用 VPN，IPSec，SSL 等各種傳輸加密方法，以確保傳送資料時的正確性，但卻很難防止中間人攻擊發生。
5. 保障文件在端點的安全性：一般當合法的使用者在 Client 端開啓或是編輯文件檔案時，如何有效避免被其他非法的使用者讀取，或是被木馬程式偷偷將資料傳送出去，是一大問題。一般研究是利用安裝個人防火牆來防止木馬程式偷偷將資料送出，另外搭配防毒軟體將可能是木馬的程式找出來刪除。

3. 系統架構

本論文結合分散式防火牆與代理伺服器技術，設計機制來確保文件的安全性，並且以儘量減少使用者及管理者因安全性措施所造成的不便爲目標。系統架構如圖 2 所示，主要可分成使用者端 (Client，以下用 #C 表示)，代理控制端 (Proxy & Control，以下用 #P 表示)，文件伺服器端 (Server，以下用 #S 表示)：

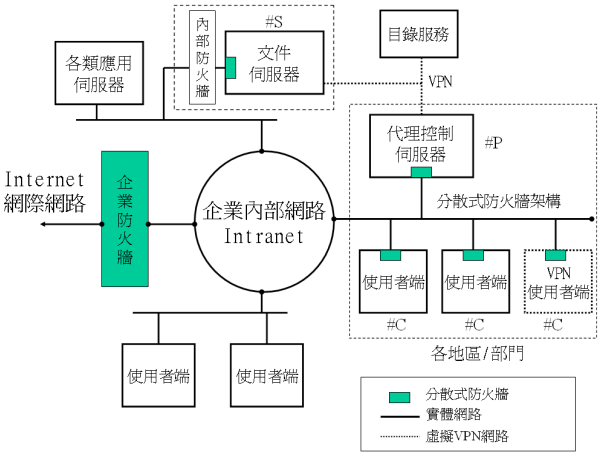

圖 2　安全文件存取機制

3.1　Server 端 (#S)

#S 端以一般化 Web Service 爲基本的文件分享平台，存取以 WebDAV [19] 協定爲標準。在安全性考量，#S 端應儘量減少被端點直接連結，且所有連結的端點必須是預先被許可及被控制，以減少攻擊的可能。所有文件存取動作皆需經由 #P，一方面可以認證及控制 #C 端，另一方面也提供 #S 被適當隱藏，其餘未經授權與未被控制的端點即使利用掃瞄工具，也會被 #S 底層分散式防火牆抵擋掉，甚至無法發現 #S 存在。

由於以往的 Web Server 存取控管多基於連線 IP 等資訊，屬於應用層面的過濾功能，無法進行快速的底層封包過濾，對於遭受如 DoS 等攻擊行爲時，利用 TCP/IP Protocol 缺點時，即造成系統癱瘓。因此我們必須要能夠快速的過濾封包，以拒絕不當的連線來源。對於此點在 #S 的進入封包控管部分，除了可以搭配獨立運作的內部防火牆，使得其它機器完全無法連到 #S 外，也可以使用個人／分散式防火牆，只允許 #P 及少數要用到的埠 (port) 通過。另外爲了達到 #S 隱藏及通訊加密的目的，#S 與 #P 端使用 VPN 作爲連線基礎，封包控管過濾只允許 VPN 所用的埠進出，如此 #S 相關網站服務埠 80/443 等並不公開對外，所有使用者都必須透過已經建好 VPN 連線的 #P，才能夠使用主機上的服務，以防止駭客掃描等攻擊。我們設計的整合架構如圖 3 所示。在此架構下，#S 原來有的存取控管機制並不受影響，仍可以照常運作 (如存取時間限制等)。

另外考量最壞的情況，即是駭客已經控制 #C 端，甚至以人工直接操作機器，此時我們在 #S 端增加下載時亂數鍵值加密之功能，使用者事先在目錄服務中註冊 E-mail 位址或行動電話簡訊號碼，當下載特定文件時，#S 會以鍵值加密文件，並經註冊之管道傳送解密的鍵值 (如圖 4 所示)，與文件採用不同發送管道，以增加應用面的安全性。

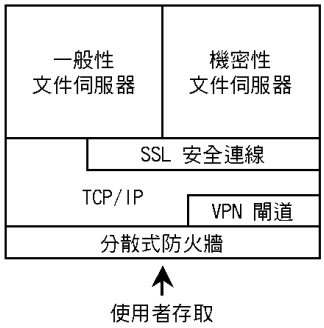

圖 3　文件分享主機端架構

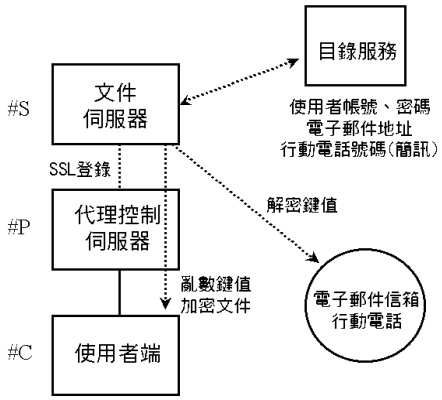

圖 4　動態文件下載加密

3.2　Client 端 (#C)

此部分的需求目標爲對使用者的不便減到最少，且達成有效的控管機制。使用者透過標準的瀏覽器與 WebDAV Client 等即可進行文件的存取，所需動作爲設定瀏覽器 proxy 爲 #P，以及安裝分散式防火牆的被控制端，藉由與控制端的密切聯繫來保障 #C 的安全性。#C 必須先執行登錄 #P 的動作來啓動相關的分散式防火牆控管規則，當正進行對 #S 的連線時，#P 便藉由分散式防火牆來阻絕 #C 的其他連線，只能允許連線到 #P，以避免駭客木馬遠端進行監控 (如圖 5 所示)。

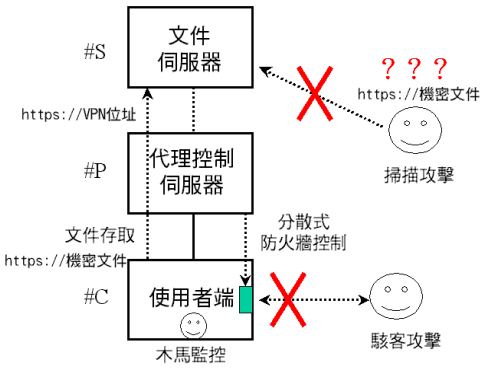

圖 5　相關安全機制

此外，企業組織更可以藉由企業防火牆的規則設置 (#P 可通過)，限定所有 #C 必須經由 #P 才可以連接 Internet，達到資訊過濾與控管的目的。

3.3 Proxy 端 (#P)

此部分包含代理與控制兩部分，代理部分主要是針對 #C 的需求，予以轉送。爲了進一步考慮安全性及便利性，我們設計代理允許如「https://機密文件」等非 DNS 所定義之名稱，利用代理程式予以轉換，連線至正確 #S。在控制方面，主要執行分散式防火牆之控管，依據不同的 #C 登錄權限及使用狀況來變更 #C 的防火牆規則。相關權限控制資訊集中置於企業的目錄服務 (如 LDAP [20] 伺服器) 中，此目錄服務亦可以作爲與文件伺服器的交換橋樑，例如在 #S 輸入存取密碼錯誤過多、禁止其使用 #P、並限制 #C 的行爲等，屬於多層次的安全性措施。

3.4 系統運作過程

整個系統運作的過程描述如下：

1. #S、#P、#C 安裝設定完成。
2. #S 與 #P 進行 VPN 認證與連線。
3. #C 開機啓動預設分散式防火牆規則，此時尚無法使用 #P、進行存取 #S 及 Internet 等。#C 進行登錄 #P 動作，#P 藉由目錄伺服器檢查權限，並傳送防火牆控制規則。
4. #C 可以經由 #P 進行一般 Internet 存取。
5. 當 #P 檢查到 #C 傳來位址如 https://機密文件，屬於內部非 DNS 名稱時，自動轉換 #S 位址。
6. #P 立刻啓動 #C 嚴密防火牆控制規則，避免 #C 與其他機器連線，以杜絕木馬遠端即時監看鍵盤及螢幕。
7. #C 與 #S 經由 #P 代理進行 SSL 連線動作。
8. #S 檢查 #C 之存取權限，如同一般普通網站的存取控制。
9. 文件存取完畢由 #P 控制還原 #C 防火牆控制規則。

4. 安全性分析

針對上述所設計之架構，我們對各種可能之攻擊行爲加以分析，以確認系統設計之有效性：

4.1 網路監聽

在 #S 與 #P 方面，由於形成 VPN 連結，封包經過有效加密保護，監聽也無法得知內容。在 #P 與 #C 方面，所傳送的資料分成兩種，一種是分散式防火牆的控制封包，另一種是經由 #P 所代理的文件資料封包，分散式防火牆的控制封包是由我們設計非對稱性加密機制，透過 Public/Private key 的運作機制 (如圖 6 所示) 來保護封包內容，而 #C 與 #S 透過標準 SSL 連線機制，使得經由 #P 至 #C 的文件資料封包同樣具有保護內容的功能。

4.2 掃描攻擊

掃描攻擊主要作爲其他攻擊的準備，#S、#P、#C 端皆有分散式防火牆裝置以過濾來路不明的連線。在 #S 方面更是關閉多數的埠，僅留 VPN 的通道，將標準的 Http 80 埠僅對 VPN 的固定裝置開放，如 #P，目錄服務等。另外我們設計以特殊名稱來取代伺服器的 IP 進行連結，由於此部分工作在 #P 內部進行替換，再經由 #S 至 #P 連線，對一般 #C 端而言，即無法得知 #S 端相關資訊 (IP，MAC 等)，而其他 Intranet 的機器裝置更是完全無法得知 #S 服務的存在，避免遭受掃描。

4.3 中間人攻擊

在 #S 與 #P 間，藉由 VPN 予以連結，我們使用標準的 IPSec 結合雙向認證和共享密鑰來抵擋中間人攻擊的可能。在 #P 與 #C 間，一方面由於所有控制封包均經過非對稱加密演算法加密，而 #P 的公開金鑰採用事前直接安裝至 #C 端，並無金鑰之交換，中間人攻擊無法發生。另一方面 #P 回應 #C 的封包資料，以封包內容經加密之特殊欄位的 IP 及 MAC，並不以標準封包的欄位爲主，因此中間人攻擊也無效。

4.4 阻絕服務攻擊

阻絕服務攻擊的重點主要在於造成伺服器的無法正常運作，在 #S 端因爲僅留下少數 VPN 埠及限定的連結對象 (#P，目錄服務)，我們可以快速過濾封包，不會造成服務的受阻中斷。另外在 #P 部分，由於分散式防火牆的控制功能，所有要使用 Proxy 功能的機器，必須先經由分散式防火牆的登錄及許可。

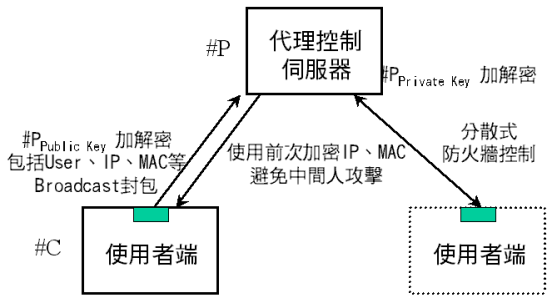

圖 6 分散式防火牆之控制加密機制

在分散式防火牆的登錄時我們可以記載有效的 IP 及 MAC，提供 #P 的快速封包過濾使用，同樣可以避免不明來路的大量連線，造成阻絕服務。

4.5 木馬監控

木馬的入侵管道主要是透過人為直接放入機器、電子郵件、網頁瀏覽、系統漏洞等，我們考量 #C 端的入侵可能，當 #C 端透過 #P 與 #S 連線時，在最終端點皆會解密，因此木馬遙控監視，就可以得知相關文件資料。為了避免此情形，我們在 #P 端代理 #C 端存取 #S 時，就會透過分散式防火牆機制控制，禁止 #C 端其他網路連線的封包外出，以避免木馬監控。另外在 #S 端，我們可以定義相關文件的屬性，加入下載加密的措施，當文件下載時先經亂數鍵值加密，將解密鍵值透過事先定義的 E-mail 位址或行動電話簡訊送出，使得木馬程式即使自 #C 端取得文件，也無從得知文件的內容。

4.6 分散式防火牆的破壞

由於分散式防火牆是保護前面一些攻擊行為的基礎，若遭受攻擊破壞則會影響其他保護的機制。為了避免分散式防火牆使用者端機制遭受破壞、卸載解除，我們在 #C 端限制 Administrator 權限的使用，使用者不能具有 Administrator 的權限，以保護分散式防火牆使用者端在核心驅動程式。另外在控制命令的傳送部分，除了我們以非對稱式加密機制保護內容外，我們更加入動態的單次控制鍵值做法 (如圖 7 所示)，控制端傳送資料時前，會先要求 #C 端傳回亂數產生一個鍵值，回送至 #P，然後將此鍵值加入控制封包傳送，#C 端若是收到不正確的鍵值則完全忽視該命令，以避免駭客錄製舊有的命令封包，來發生重送攻擊，造成系統的混亂。另外在封包的保護方面，以 ICMP Echo Reply 作為主要封包架構，將重要部分以加密形式置於資料後端，一般無法解密只會認為是 ping 的回應，並不會特別顯著地易遭受攔截。另外我

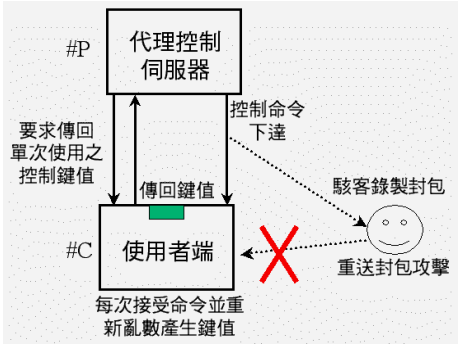

圖 7 分散式防火牆之單次控制鍵值

們採用 RSA 的加密機制，在原始封包的內部不足長度的部分皆以亂數補足一定長度，再經過加密後，形成相同命令但是每次封包皆不同的狀況，使得駭客根本無法從封包攔截進行分析，確保資料的安全。

5. 實作相關技術

本節依照系統架構之各部分相關實作技術細節加以詳細說明：

5.1 Server 端 (#S)

#S 端以 Apache Web Server 作為平台。Apache Web Server 具有系統穩定、模組化設計擴充性高等優點，且目前支援眾多作業系統，如 Windows 系列、Linux 等，符合多數企業組織的需求。在模組部分，利用 mod_ssl 來支援 SSL 通訊協定，mod_ldap 來支援 OpenLDAP [20] 等。我們同時製作動態下載加密的模組 mod_encode，可視需求定義文件於下載時，藉由亂數產生的鍵值對文件加密，並透過使用者事先登錄在目錄服務的 E-mail 信箱或行動電話簡訊傳送解密鍵值，來加強文件的安全性保護。另外在目錄服務方面，我們使用 OpenLDAP 作為使用者權限資料之儲存核心。在封包控制部分，Windows 作業系統環境下，我們設計分散式防火牆使用者端來進行過濾，只允許少數對象的 VPN 封包通過。在 Linux 作業系統部分配合 iptable 作為過濾機制，同時我們使用 Open Source Project 的 FreeS/WAN 及 Poptop 來作為 VPN 伺服器使用。

5.2 Client 端 (#C)

由於目前 Windows 作業系統仍是使用者端最為普及的作業平台，為了符合實際企業組織應用的需要，本研究在 #C 端主要以 Windows 為主，支援 Win98、Win2000、WinXP。#C 端主要的程式即為分散式防火牆的使用者端，為了避免使用者能夠任意卸載防火牆，以造成任意連線之漏洞，整個控制程式以 Device Driver 的形式置於系統的核心中，利用作業系統的權限來保護。另外由於基於 Windows 平台設計，必須與整個 Windows 網路設計架構密切配合，利用 NDIS Hooking 及 NDIS Intermediate 的技術來進行封包過濾控管。

5.3 Proxy 端 (#P)

#P 端的程式主要包括分散式防火牆控制端及代理伺服器兩大部分。如同 #S 端一般，為了滿足企業

組織的應用，我們考量 Windows 及 Linux 兩大作業平台。在分散式防火牆控制端，我們不需要像 #C 端完全置於系統核心中，考量開發及測試的便利性，我們以應用程式層爲主，封包的收送則利用 Packet Driver 作爲介面，加上與 #S 端相同的封包過濾機制。另外在代理伺服器部分，我們主要以改寫 Apache mod_proxy 爲主，增加伺服器名稱轉換對應功能及加強與分散式防火牆之整合控制。

在程式開發工具方面，我們主要以 C 及 C++ 語言爲主，在 Windows 作業系統使用 Visual C++ 6.0，搭配開發工具如 Windows Device Driver Development Kit [9]、Windows Resource Kit [21] 等，在 Linux 作業系統主要以標準 C 編譯器。其他跨平台的程式部分，包括 Apache API [22]、Packet Driver Library [23]、OpenLDAP API 和 OpenSSL Library [24] 等。

5.4 分散式防火牆之控制

由 #P 端至 #C 端是我們設計分散式防火牆主要的控制機制，爲了能夠有效保護控制命令的傳送及防火牆規則的散佈，我們設計封包相關格式及欄位如圖 8 所示。各欄位說明如下：

首先爲了安全性考量，避免封包顯眼遭受注視，控制封包以 ICMP echo reply 爲基本加以擴充，一般不詳細分析會認爲是 Ping 的回應。封包主要內容分成 ICMP Header 及 ICMP Payload 兩部分。在 ICMP Header 部分，利用 Type 0 (Echo Reply)，及特殊選定的 Code 作爲識別。在 ICMP Payload 部分，整個封包經過加密保護，其中 Signature 爲固定之兩個字元，其主要目的在於核對封包是否經過正確之解密動作，一旦解碼錯誤則 Signature 不正確，整個封包會被丟棄。Command 爲主要的指令代碼，Mode 爲 #C 端模式代碼。Key Password 爲 32bits 之亂數產生，用以作爲單次控制鍵值的識別。#P 必須給定正確的數值，否則將會被視爲無效的命令，以避免駭客錄製命令重送的攻擊。另外 Sender IP、Sender MAC 則是保護在封包之內眞正的 IP 及 MAC，提供 #P 端回應使用，使得像中間人攻擊等無法發揮。最後，Optional Data 部分則是包括防火牆控制規則的資料、相關使用者憑證的資料等。

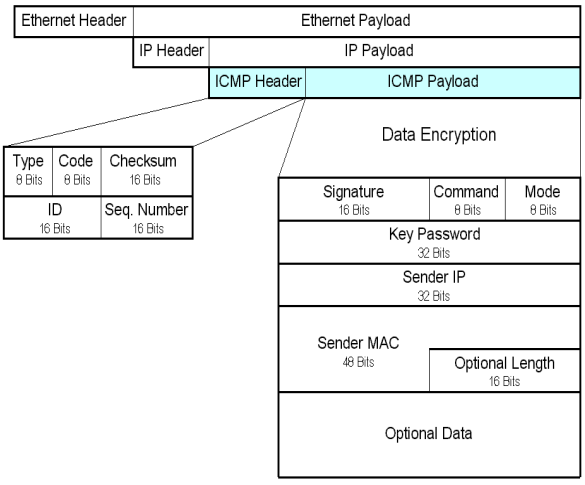

圖 8 控制封包之格式

除了上述欄位之外，在資料加密部分，我們實作 1024 bits RSA 加解密機制作爲封包傳送之基礎，由於 #C 端考量實作時系統安全，整個程式建構於系統核心中，受限於核心程式的一些限制，例如有限的可呼叫函數庫、獨立的記憶體管理等，許多現成的加解密 library 將無法使用 (如 SSLeay、Microsoft Encryption 等)。我們以 PuTTY [25] 的原始碼爲基礎，改寫其中記憶體處理及修正相關的 memory leak bug，作爲此部分 RSA 演算法之基礎。

6. 結 論

在本研究中我們應用結合分散式防火牆與代理伺服器的技術，來建立安全性文件存取機制，可以提供企業組織此方面的需求與後續研究之參考，然而所有的安全性研究永遠沒有止境，攻擊者隨時都在尋找可乘之機來突破防護。在經過實際的運作測試，我們得到以下的結論，並認爲未來仍有許多值得繼續研究改進的地方：

1. 安全性的措施必須是多層次的結合，由於攻擊型態的複雜化，單憑任一種安全性機制皆無法有效抵擋各種攻擊，唯有結合各網路層面及從使用者應用導向的網路控制並即時依狀況反應，整體安全機制才能確保安全。
2. 在木馬技術的防堵方面，本研究有效切斷駭客遠端即時監控的可能性，大幅減少攻擊的威脅，但仍有可能是長期潛伏的木馬配合事前精心規劃的批次作業攻擊，例如長期性攔截畫面輸出文字及鍵盤輸入字元加以分析，取得關鍵密碼等。這方面的防護可以藉由另外的一些額外措施來加強：例如以圖形或語音來取代文字的詢問，不定期要求使用者依亂數產生的文字圖形來鍵入文字等。這些加強性的防護措施都是未來非常值得繼續研究的方向。
3. 加強與其他安全機制的整合：包括加強系統相關紀錄與稽核的功能，能夠對各存取行爲有效的加以監控紀錄，遇有安全性的事件可以立即反應並加以處理，同時相關紀錄並可以作爲重要的佐證。另外研究結合像入侵偵測系統 (Intrusion Detection System，IDS) 等機制，增加多方面的同步監控分

析與控制來確保安全。

4. 改進控管規則：在存取控制及各項控管方面，也是未來值得研究的重要方向。如何能配合整個企業組織的安全政策機制，使得高階的政策能夠有效且一致地轉換至底層之各項安全控制裝置配合，如分散式防火牆、伺服器、代理伺服器等，使得控管更爲實際且便利。例如，禁止非上班時間存取機密文件此一政策，轉換成各項裝置的控制規則等，形成多層次的防堵控制。
5. 分散式防火牆之加強：由於分散式防火牆部分程式位於使用者端，最大問題在於遭受破壞卸載、分析等。我們雖然可以從行政命令 (要求使用者必須安裝) 及作業系統控制 (使用者沒有管理密碼等)，但是因爲使用者端難以完全掌握，甚至使用者就是惡意的駭客。此部分的問題涉及系統程式保護部分，包括保護執行檔案的完整，禁止利用逆向工程技術來反組譯取得重要程式邏輯等，這些都是在系統保護技術方面長久以來的問題，也是需要進一步研究的課題。

參考文獻

[1] S. M. Bellovin, “Security problems in the TCP/IP protocol suite,” *ACM Computer Communications Review*, Vol. 19, No. 2, Mar. 1989.

[2] Hypertext Transfer Protocol HTTP/1.0, RFC 1945.

[3] Hypertext Transfer Protocol HTTP/1.1, RFC 2068.

[4] S. M. Bellovin, *Distributed Firewalls*, November 1999, http://www.research.att.com/~smb/papers/distfw.html

[5] D. Wan, *Distributed Firewall*, May 2001, http://www.giac.org/practical/gsec/Daniel_Wan_GSEC.pdf

[6] W. Li, *Distributed Firewall*, December 5th, 2000. http://www.cs.helsinki.fi/u/asokan/distsec/documents/li.ps.gz

[7] J. Wack, K. Cutler, and J. Pole, “Guidelines on firewalls and firewall policy,” *NIST Special Publication*, 800-41, Jan. 2002.

[8] V. S. Vadim, “Firewall for Windows 9x/NT/2000,” http://www.ntkernel.com/articles/firewalleng.shtml

[9] Microsoft Windows Driver Development Kits, http://www.microsoft.com/whdc/ddk/winddk.mspx

[10] R. Peteanu, “Best Practices for Secure Development,” http://citeseer.nj.nec.com/peteanu-01best.html

[11] J. Wang and Chien, “Using Overlay Networks to Resist Denial-of-Service Attacks,” http://citeseer.nj.nec.com/wang03using.html

[12] F. Kargl, J. Maier and M. Weber, “Protecting web servers from distributed denial of service attacks,” *Proceedings of World Wide Web*, 2001, pp. 514–524.

[13] T. Peng, C. Leckie and K. Ramamohanarao, “Detecting distributed denial of service attacks using source IP address monitoring,” 2002, http://citeseer.nj.nec.com/peng02detecting.html

[14] A. Hussain, J. Heidemann and C. Papadopoulos, “A framework for classifying denial of service attacks,” 2003, http://www.isi.edu/~hussain/pubs/Hussain03a.pdf

[15] R. Rivest. “The MD5 message-digest algorithm,” RFC 1321, Apr. 1992.

[16] V. Pathak, “Encryption servers: a scalable distributed method for internet security,” http://citeseer.nj.nec.com/pathak01encryption.html

[17] P. Chodowiec, P. Khuon and K. Gaj, “Fast implementations of secret-key block ciphers using mixed inner- and outer-round pipelining,” FPGA 2001, pp. 94–102.

[18] D. S. Alexander, W. A. Arbaugh, A. D. Keromyts and J. M. Smith, “A secure active network environment architecture: realization in switchware,” *IEEE Network Magazine*, 1998.

[19] Y. Goland, E. Whitehead, A. Faizi, S. Carter and D. Jensen, “HTTP extensions for distributed authoring – WEBDAV,” Feb. 1999.

[20] OpenLDAP Project, http://www.openldap.org

[21] Microsoft Windows 98 Resource Kit, http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/win98/reskit/win98rk.asp

[22] The Apache Software Foundation, http://www.apache.org

[23] WinPcap: the Free Packet Capture Architecture for Windows, http://winpcap.polito.it/

[24] OpenSSL Project, http://www.openssl.org/

[25] PuTTY: A Free Win32 Telnet/SSH Client, http://www.chiark.greenend.org.uk/~sgtatham/putty/

潘 諫 **(Chi-Chien Pan)** 曾任職於交通部運輸研究所副工程司、工程司，民國 81 年取得國立台灣大學資訊工程研究所碩士學位，目前爲國立台灣大學資訊工程研究所博士班學生，同時任職於交通部民用航空局飛航服務總台資管中心副主任。研究興趣爲資訊安全與管理、個人防火牆、網路通訊。

楊 凱 翔 **(Kai-Hsiang Yang)** 民國 87 年由國立台灣大學資訊工程研究所碩士直攻博士，目前爲國立台灣大學資訊工程研究所博士班學生。主要研究領域爲代理伺服器、資訊安全、網路系統。

李 肇 林 **(Tzao-Lin Lee)** 民國 67 年取得國立台灣大學數學研究所碩士學位，民國 70 年取得美國卡內基美濃大學統計系碩士學位，民國 74 年取得美國卡內基美濃大學統計系博士學位，目前爲國立台灣大學資訊工程學研究所副教授。主要研究領域爲軟體工程、電腦網路、資料庫系統整合、資訊安全。

收稿日期 92 年 11 月 30 日、修訂日期 93 年 2 月 10 日、接受日期 93 年 2 月 12 日
Manuscript received November 30, 2003, revised February 10, 2004, accepted February 12, 2004