三個網路代理伺服器：FWTK, AMaViS, SpamAssassin

李志祥 曹世強 林盈達

國立交通大學資訊科學系

新竹市大學路 1001 號

TEL：(03)5712121-56667

E-mail: **{lizx, weafon, ydlin}@cis.nctu.edu.tw**

摘要

網際網路的快速普及，使用人數大量的增加，為節省網路頻寬，縮短使用者等待時間，而有代理伺服器(Proxy server)架構的提出。他具有快(Speed)、通透性(Transparency)、安全性(Security)等好處。代理伺服器目前在網路應用層(Application Layer)上常見的種類有三，包括快取服務(Cache)，應用層防火牆(Application Firewall)及內容過濾(Content Filter)。在應用層防火牆方面，著名的 TIS FWTK 套件可用來抵擋應用層上安全的漏洞。而內容過濾方面又可分為病毒偵測和垃圾郵件過濾兩種。郵件處理介面 AMaViS 套件搭配掃毒引擎 Clam Anti-Virus 套件，可進行郵件病毒的偵測。而 SpamAssassin 套件則提供垃圾郵件的過濾處理。文中對 FWTK, AMaViS,及 SpamAssassin 這三個套件，進行深入追蹤及運作流程分析。FWTK 方面，設定檔的重複讀取及內容過濾時一次讀取一個字元等等是影響處理效能的關鍵。Anti-Virus 方面，解壓縮及病毒碼的比對是消耗處理時間的關鍵。Anti-Spam 方面，信件表頭、信件內容及線上黑名單資料庫的比對是影響處理效能的關鍵。

關鍵字：FWTK,AMaViS,SpamAssassin,Proxy,Clam,Openantivirus,Firewall

1. 簡介

很多人在不知不覺中，默默接受著代理伺服器的服務。一個典型的代理伺服器的運作流程如圖 1 所示，他位於使用者端(Workstation)與遠端伺服器(Remote System)的中間。當使用者要傳輸資料到遠端系統時，先將資料傳到代理伺服器上，代理伺服器再將使用者的資料傳輸到遠端的系統。而當遠端系統回傳資料

時，代理伺服器會先接收，再將內容傳送到使用者端。如此即完成了一次資料傳輸的步驟，當資料連續傳輸時，傳輸的方式跟上面的方法相同，都是透過代理伺服器來傳送及處理。由於，代理伺服器處理資料封包的層級，相當於國際標準組織 (International Standards Organization, ISO) 所訂定的開放式系統互連模組 (Open System Interconnection Model, OSI Model) 中的應用層(Application Layer)網路協定，所以又有人稱之為應用層代理伺服器 (Application Proxy)。

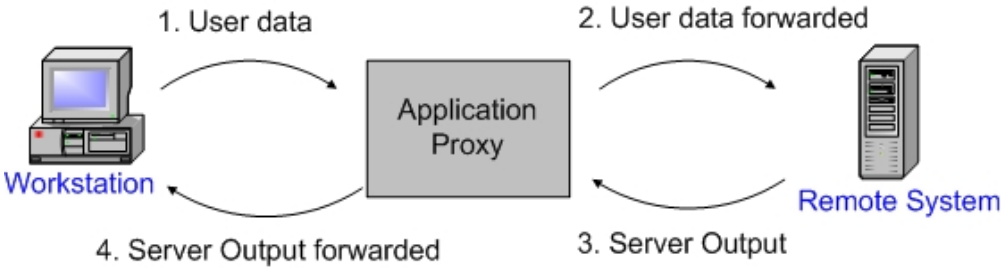

圖 1：代理伺服器的運作流程圖

究竟代理伺服器夾在使用者與遠端伺服器間，能帶來哪些便利呢？根據提供服務的不同，好處也有所區別，不過大致上有『快』、『通透性』、『安全性』、『加值服務』等四個優點，以下針對各項分別說明。

- **快**

代理伺服器處理資料的遞送時，能將資料儲存起來。當使用者端第二次再讀取同樣的資料時，代理伺服器就不用再去詢問遠端系統，而可以直接將儲存的資料傳遞給使用者，如此即可讓使用者更快取得所需的資料，又可以節省骨幹網路的頻寬，達到『快』的好處，故有人稱此類伺服器為快取伺服器(Cache Server)。

- **通透性**

當系統管理者為了提供如網頁內容的過濾、快取的應用、病毒的偵測…等特定的服務，卻又不想影響到使用者端的設定，那就可以採用代理伺服器通透性的技術。代理伺服器居中處理，利用網路資料流的重導，默默的處理資料後，傳送到使用者的手上，使用者不需額外作任何設定，達到『通透性』的好處，又有人稱此類伺服器為通透性代理伺服器(Transparent Proxy)。

- **安全性**

當公司的決策者或校園網路系統管理員，為了提高公司內部網路或校園網路的安全性，除了網路層防火牆外，另一個可以使用的方法就是採用應用層代理伺服器，對於網路封包中應用層的內容，進行嚴格的過濾，移除有可能危害系統安全的內容，如 Java、JavaScript、ActiveX…等等。如此即可保護內部網路或校園網路的安全。

- **加值服務**

當系統管理者為了提供系統內額外的服務，而不打算影響到使用者目前設定時，即可以採用代理伺服器具有『加值服務』的特點，居中處理。如系統管理者想要提供病毒的偵測，即可在代理伺服器中事先將資料掃描過濾，經過適當的處理再將資料傳送到使用者手上，達到提供病毒偵測的目的，除了病毒偵測外，亦可以提供垃圾郵件過濾、網頁資料內容過濾…等等。

目前代理伺服器的種類大致分成快取式代理伺服器，應用層防火牆及內容過濾三種。快取式代理伺服器主要在加快瀏覽速度，最具代表性的套件為 Squid，本文因篇幅的關係，則不詳加介紹，有興趣的讀者可以參考[1]，至於應用層防火牆，代表性的套件有 FWTK[2]與 DansGuardians [3]，此兩套皆支援網址過濾(URL Filter)及內容過濾(Content Filter)之功能。由於 FWTK 另外還有支援郵件代理伺服器、檔案傳輸代理伺服器等之功能，所以在下一節，我們選用 FWTK 套件來當範例，詳細分析及介紹代理伺服器。至於內容過濾方面，除了網頁內容過濾外，郵件系統內容的過濾也是很重要的一環。我們將於第三及四節，分別選取代表性的病毒過濾程式 AMaViS 搭配 Clam-av 及垃圾郵件分析程式 SpamAssassin 進行深入的探討。最後是我們的結論。

2. FWTK 應用層防火牆

所謂 FWTK 是 FireWall ToolKit 的縮寫，是一套設計在 UNIX 環境下使用 TCP/IP 運作的應用層防火牆工具套件，在 1970 年代，網路開始發展，ARPA-NET 實驗性網路的成功，當時即有人注意到網路安全的問題，於是在擴建計劃中，委託 Trusted Information Systems(TIS)這家公司研究網路安全的問題，發展出現今這套應用層防火牆工具套件。以下將對這個工具套件進行深入的研究，了解其整體

的軟體設計架構、設計的目標、資料結構、系統運作流程及程式流程分析。

2.1 設計目標

為了達到網路安全，FWTK 在設計上，根據筆者整理，可列出六大目標：

第一、 讓使用者遠離閘道器，也就是所謂的應用層代理伺服器，使其不能登入此主機。

第二、 在代理伺服器上執行的程式，不要具有特殊的管理者權限，如 root ，也就是說如果程式不需要用到特殊的權限執行時，盡量的降低程式執行的身份。

第三、 利用變更根目錄(Chroot)的機制，隱藏系統檔案資源及目錄結構。

第四、 程式執行時能夠支援使用者身份確認。

第五、 代理伺服器在處理重要的交易時，能夠利用紀錄檔將事件、時間及使用者紀錄下來。

第六、 設計套件中的程式碼能夠簡單，使得任何人都能夠快速的檢查是否有系統安全性上的漏洞。

2.2 FWTK 軟體組成架構

FWTK 的全名 FireWall ToolKit 暗示著 FWTK 是由一些小工具組成的軟體套件。各個小工具能夠提供特定的服務，當系統管理者在設定應用層代理伺服器時依照需求來選取這些小工具，然後經過適當的組合，即可達到提供應用層防火牆的功能。如私有網路(private network)內部想提供 HTTP 及 MAIL 的服務，以利內部使用者可以使用外部網路的服務，可以選取 smap、smapd 及 http-gw 來搭配組合，即可達成需求，若想更嚴格管制，要經過帳號密碼認證才能使用，則在搭配 authsrv 模組即可，由此可知 FWTK 使用上的彈性。

表 1 列出 FWTK 所提供的模組。一般來說 smap 與 smapd 是合併使用的，前者在 port 25 等待連接，然後將接收的郵件放到特定的目錄，後者則以 Daemon 的模式運作，檢查特定目錄是否有郵件儲存，如果有新郵件則將此郵件傳送給系統的 Mail Transfer Agent (MTA)，MTA 則照正常的程序傳送 Mail，如此的好處是 MTA 程式如果有漏洞不會直接暴露在外面，達到安全性的效果。

	Main service	Statement
1.	smap	SMTP service
2.	smapd	SMTP service
3.	Netacl	Network Access Control Lists (all inetd daemon)
4.	ftp-gw	A proxy server for FTP
5.	tn-gw	A proxy server for TELNET
6.	rlogin-gw	A proxy server for RLOGIN
7.	Plug-gw	A TCP Plug-Board Connection Server (Usenet news)
8.	http-gw	A proxy server for HTTP
9.	x-gw	A proxy server for X Window
10.	authsrv	network authentication service

表 1：FWTK 軟體組成架構

而 Netacl 則與 TCP wrapper 相當，提供連線位址的檢查，使得系統多一層的保護，剩下的工具如 ftp-gw、tn-gw、rlogin-gw、http-gw 及 x-gw 就類似代理伺服器的功用，利用封包的過濾及重導，使得內部網路有一定的安全性。至於 authsrv 則是提供使用者認證的功能。由以上介紹，可以看出 FWTK 已經提供網路上大部分的應用功能。

2.3 設定檔資料結構

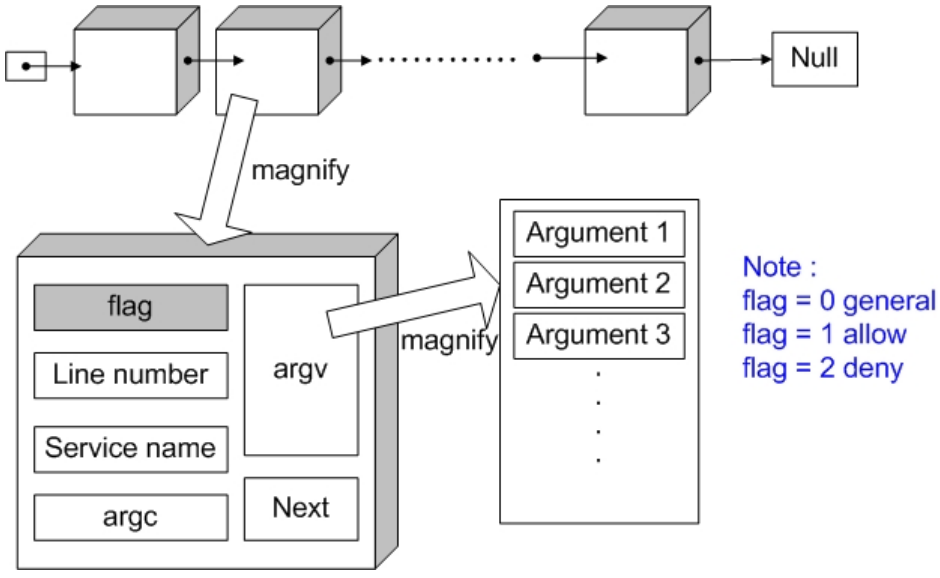

圖 2：FWTK 設定檔資料結構

FWTK 與其他的應用程式相較之下沒有那麼複雜，如 Squid，所以也就沒有用到很特別的資料結構，但所有的小工具皆圍繞在一個設定檔上，每次一有新的連線就會去檢查設定檔的內容，而利用特定資料結構儲存設定檔的內容，如圖 2，設定檔的內容以『一行』為一個單位，存入一個結構中，如圖 2 上方中的一個方塊。而指標 next 則指向下一行的設定，如此即完成設定檔的儲存。至於每一個結構中，則儲存著是否可通行的旗標、位於設定檔的行數、服務名稱及一些特殊的引數與網路位址…等等。

2.4 運作流程圖

FWTK 的運作流程大致上與圖 1 介紹之應用層代理伺服器相同，只有某些部分有些許的差異。如圖 3 中步驟 2 的部份，FWTK 會嚴格的檢查連線機器的網際網路位址(IP Address)及其網域名稱(Domain Name)的正解、反解是否相同，如果都正確無誤才能完成正常的連接及重導程序，並將這事件紀錄下來。除此之外，FWTK 的 http-gw 模組亦提供網路內容過濾(Content Filter)及特殊網址過濾(URL Filter)的功能，如圖 3 中步驟 5 的部分，其他的部分則與圖 2 相同。

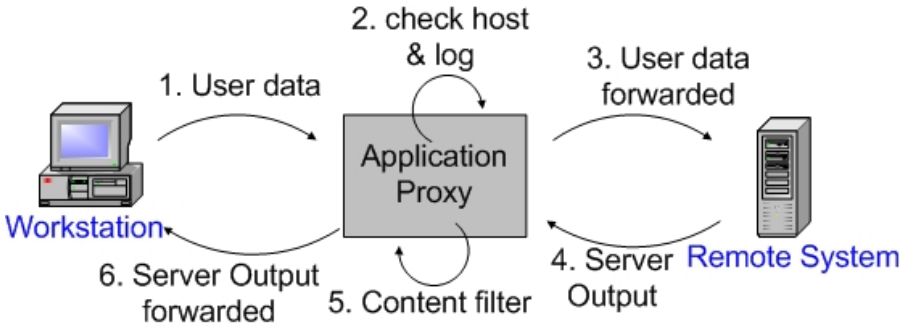

圖 3：FWTK 運作流程圖

2.5 netacl, ftp-gw, http-gw 程式流程分析

接下來我們挑選比較常用到的三個模組，來深入的探討 FWTK 內部的運作流程。並檢測該應用程式的好壞，判斷是否有拖累系統效能的關鍵步驟。

- **Netacl**

Netacl 與 TCP Wrapper 的功能類似。其運作的流程，如圖 4 所示。一開始管理者可以選擇要讓其以 Daemon 的形式提供服務還是利用系統內原有

inetd 的形式喚起 Netacl，兩者在設定上有些微差異，接著進入圖 4 中第 1 階段，讀取設定檔，然後將設定檔的內容存入前幾節所提到的資料結構，再來進入到第 2、第 3 階段，檢查網域名稱的正解、反解及網際網路位址是否准許通行，如果都通過檢查，系統會記錄連線資訊，然後進入第 4 階段，執行變換根目錄及降低執行權限的動作，最後執行系統呼叫，喚起預先設定的服務，如此即完成整個程式的動作。

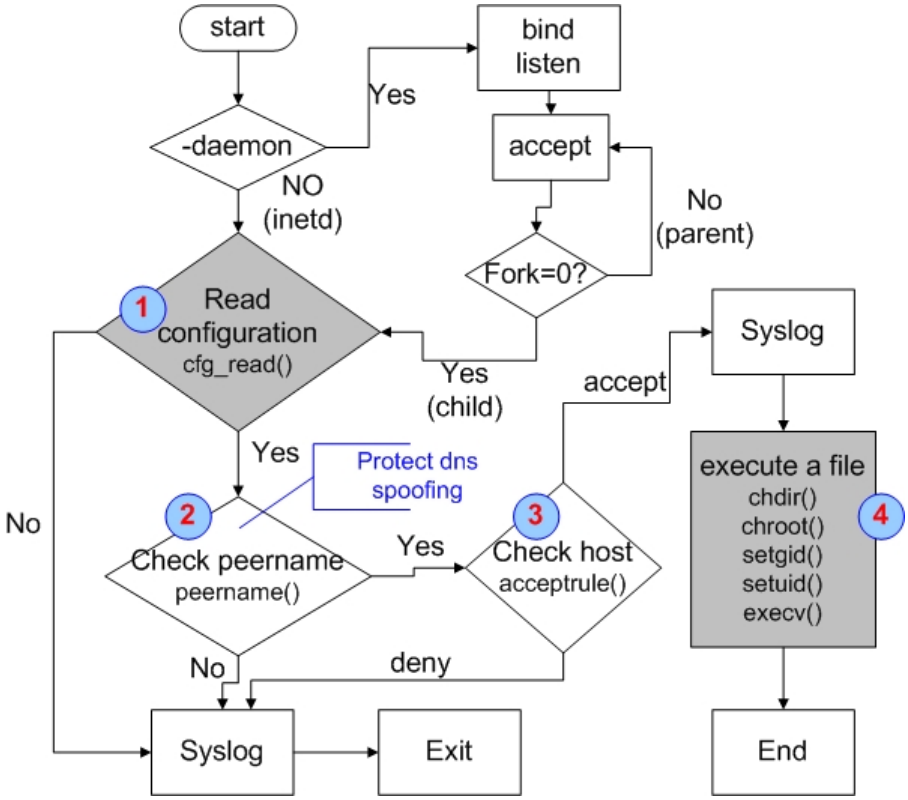

圖 4：FWTK Netacl 程式流程圖

■ **ftp-gw**

如圖 5 所示，第 1 階段到第 4 階段，大致上與 Netacl 相同，差別在於第 3 階段與第 4 階段的順序，主要是因為 ftp 的連線模式有資料傳輸模式及控制模式，有可能開啟兩個以上的連線。當第一個連接檢查過網路位址後，程式會先記憶起來，之後如果有同網路位址的連接，則可以跳過位址檢查，所以才將順序對調。接著進入第 5 階段，首先檢查使用者的帳號及密碼，如果不正確，則一直停留在第 5 階段。如果正確則將 Authenticated 旗標設為 True，進入第 6 及 7 階段。代理伺服器幫忙把資料重導到另一端的網路，並

等待及接收伺服器傳回的內容。當資料傳回後，進入第 8 階段，代理伺服器將資料傳回給原使用者，如此即完成一次資料的傳輸。當資料連續傳輸時，如圖 5 淡黃色的部分依照紅色箭頭，一直在 5～8 階段循環，直到資料傳輸完畢，此即為 FWTK ftp-gw 的程式流程，至於其他的代理程式，如 tn-gw、rlogin-gw …等等，與此流程類似則不另外介紹。

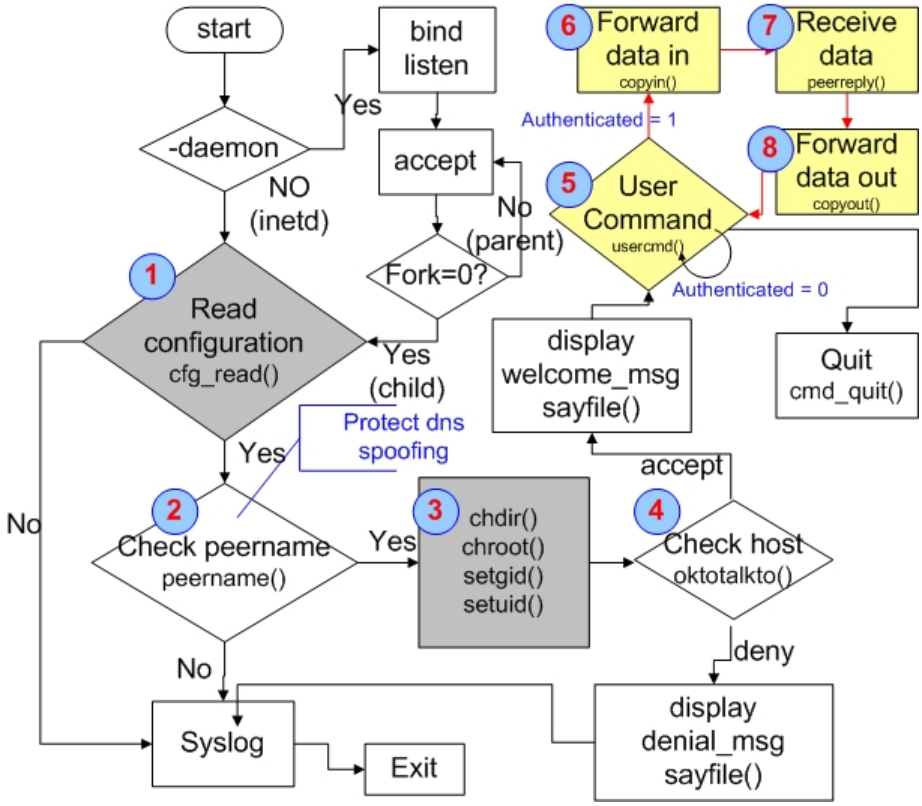

圖 5：FWTK ftp-gw 程式流程圖

■ **http-gw**

接下來是 FWTK 最複雜、最精采的部分，HTTP 模組，如圖 6 所示。此模組提供代理伺服器處理 HTTP 協定的運作，以達到特殊網址過濾(URL Filter)及網頁內容過濾(Content Filter)的目的。程式的運作流程一開始與其他模組類似，讀設定檔、檢查網域名稱及檢查網際網路位址，如果檢查正確的話，則進入第 4 階段，讀取使用者欲連接的網址，然後進入第 5 階段，檢查此網址是否為禁止的網址，如果非禁止的，則進入第 6 階段，將請求的網址傳遞到遠端機器並等待回應，當接收到回應的內容時，先檢查檔案的格式，如果不認得此檔案格式，則中斷連線，如果認得檔案格式，但非網頁的格式，則直

接將內容傳遞給使用者，如果為網頁格式，則對其內容做檢查，過濾掉一些特殊的內容。至於可以過濾哪些內容呢？端看設定檔而定，如 Java、JavaScript、ActiveX …等等，最後再將過濾後的內容傳遞給使用者，此即為 FWTK http-gw 的程式運作流程。

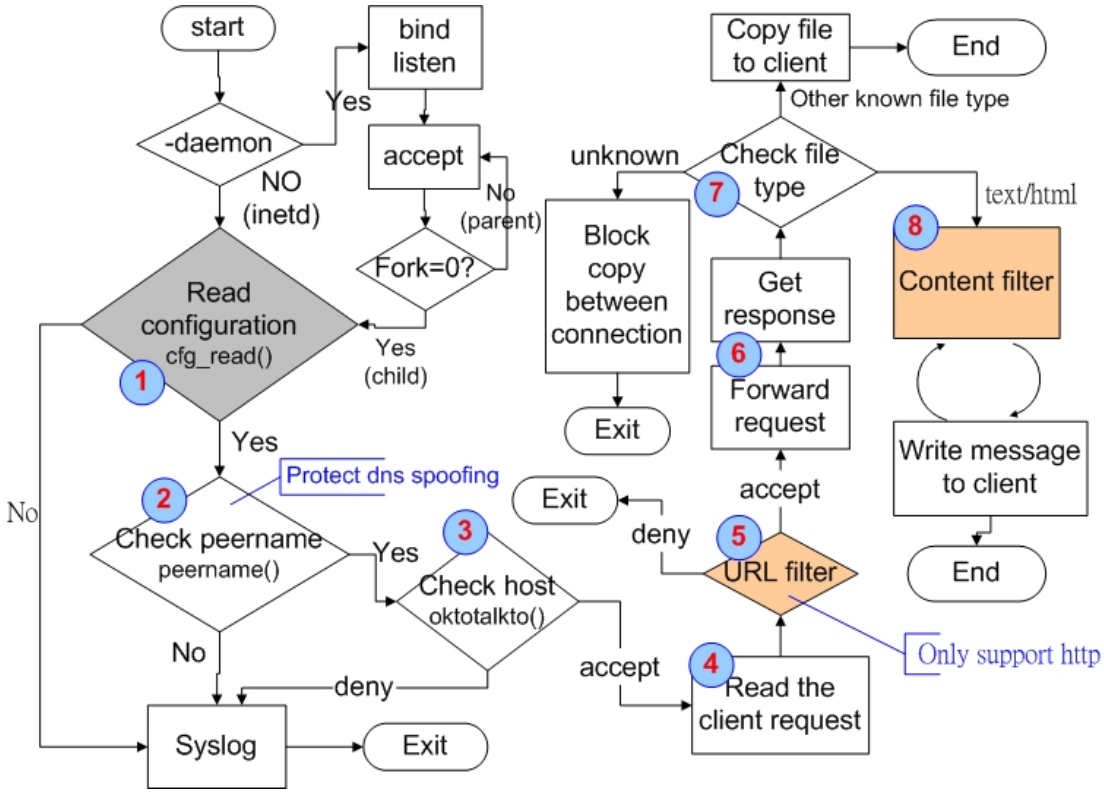

圖 6：FWTK http-gw 程式流程圖

3 AMaViS + Clam-AV 病毒偵測

AMaViS 是 A Mail Virus Scanner 的縮寫[4]，提供 Mail Transfer Agent (MTA) 與 Virus Scan Engine 溝通的橋樑，而 Clam-AV 為 Open Source 的病毒掃描引擎[5]，兩者搭配即能完成病毒偵測的效用，對於郵件系統管理者來說，瞭解這兩套軟體是非常重要的。以下將深入的介紹各個運作的模式。

3.1 Mail 與 AMaViS 的運作流程介紹

本節將從系統郵件伺服器運作的角度，介紹 AMaViS 與郵件伺服器是如何搭配運作的。如圖 7 所示，步驟 1-5 顯示正常的郵件的傳送過程。使用者一開始利用 Mail User Agent (MUA)寫信，然後將信件傳送出去，此時 MUA 會透過 SMTP

Protocol 與 Mail Transfer Agent (MTA)溝通，將信件傳遞給 MTA。MTA 此時會判斷接收信件者是否為本機使用者，如果是，則直接傳遞給 Mail Delivery Agent (MDA)處理，MDA 再將信件放到正確使用者的信箱，如果非本機使用者，則再利用 SMTP Protocol 將信件傳送到下一台 MTA，如此便完成郵件寄送。當要提供額外的病毒偵測時，可以在圖 7 中 A 或 B 的地方使用 AMaViS，中途攔截掃描郵件，然後再將郵件重導回 MTA，達成病毒偵測的目的，此即為 Mail 與 AMaViS 搭配的運作模式。

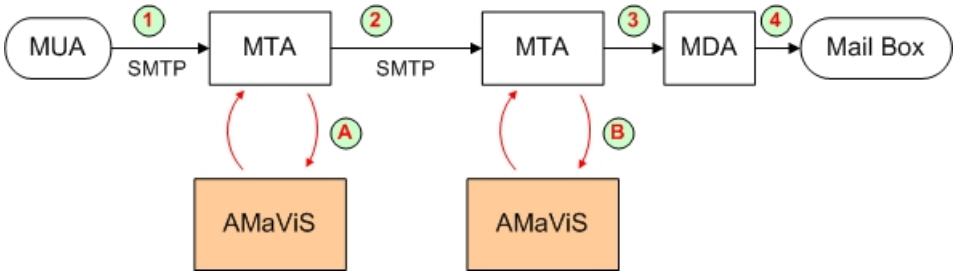

圖 7：Mail 與 AMaViS 的運作流程圖

3.2 AMaViS 的程式流程分析

前一節從系統郵件運作的角度，了解到 AMaViS 在郵件系統運作中的定位。而本節更深入介紹 AMaViS 所能提供的功能及其程式運作流程。

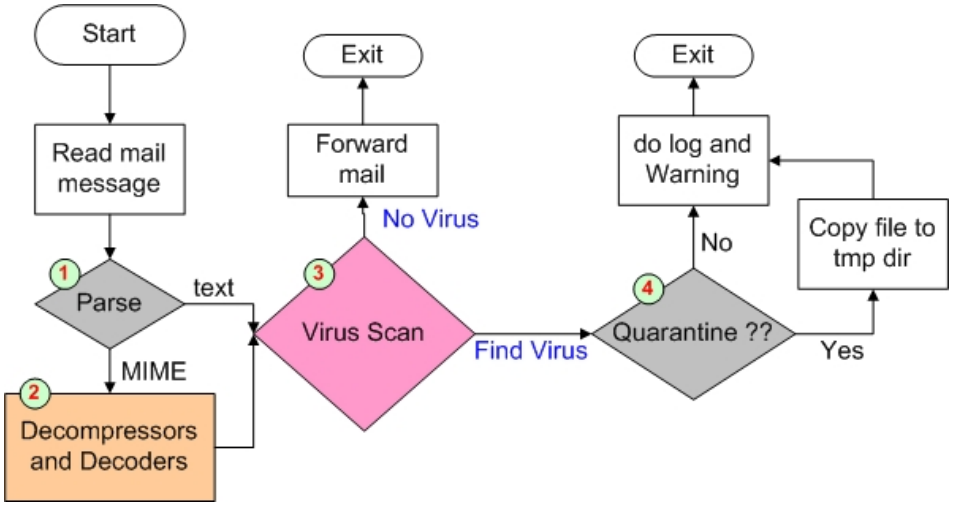

圖 8：AMaViS 程式流程圖

如圖 8 所示，當 AMaViS 接收到郵件時，進入第 1 階段，會先分析郵件的

格式，如果為純文字檔，則直接丟給掃毒引擎掃毒，如圖 8 中的第 3 階段，若為 Multipurpose Internet Mail Extensions(MIME)格式，則進入第 2 階段，解讀信件內容及對壓縮檔解壓縮，而 AMaViS 所支援的解壓縮套件如表 2 中所列，當完成此步驟，AMaViS 進入第 3 階段，喚起掃毒引擎執行掃毒工作，當發現有病毒存在時，會依設定檔設定之內容，判斷是否將信件隔離，如此即完成病毒偵測功能。

1. uudecode	2. compress
3. gunzip	4. unzip
5. unarj	6. unrar
7. xbin	8. LHarc
9. bunzip2	10. arc
11. freeze	12. tnef

表 2：AMaViS 使用之解壓縮套件

1. Sophos Sweep
2. Trend Micro FileScanner
3. Network Associates Virus Scan for Linux
4. CyberSoft VFind
5. Dr Solomon's AntiVirus
6. F-Secure Inc. (former DataFellows) F-Secure AV
7. H+BEDV AntiVir/X
8. Kaspersky Anti-Virus
9. CAI InoculateIT
10. GeCAD RAV AntiVirus 8 (preliminary support)
11. ESET Software NOD32 (preliminary support)
12. Command AntiVirus for Linux

表 3：AMaViS 支援之掃毒引擎

由上一段內容可了解到 AMaViS 套件本身並無掃毒能力，只是 MTA 與 Virus Scan Engine 溝通的橋樑。而 AMaViS 支援哪些掃毒引擎呢？如表 3 所列，AMaViS 在perl-11 的版本中已經能夠支援 12 種掃毒引擎[1]，但所列的套件當中，皆未有

[1] AMaViS-Next Generation 版本則內建支援 Clam-AV 掃毒引擎

程式碼可供研究，且目前開放原始碼中對於具有掃毒引擎的套件屈指可數，於是筆者修改了AMaViS程式，使其能夠支援第 13 種掃毒引擎 ---- Clam-AV，下一節將深入了解 Clam-AV 掃毒引擎的運作模式。

3.3 Clam-AV 的程式流程分析

目前 Open Source 的病毒掃描引擎不多，且大多是利用 OpenAntiVirus [6]所發表的病毒特徵進行比對，Clam-AV 也不例外。然而，OpenAntiVirus 從 2002 年 10 月後就沒有更新過病毒特徵檔，如此即無法偵測新的病毒。在那之後，Clam-AV 使用舊有的病毒特徵檔，並自行更新新版的病毒特徵檔，且提供線上下載。因此，我們選擇此套件來進行研究。

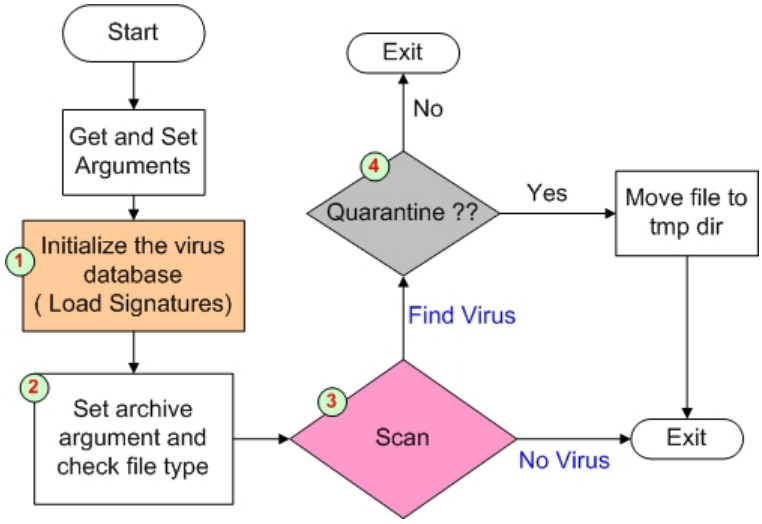

圖 9：Clam-AV 程式流程圖

接下來，我們介紹 Clam-AV 的運作流程。Clam-AV 提供兩種運作模式，一種是使用 Daemon 的模式運作，另一種則是命令列模式，為了與 AMaViS 搭配運作，必須採用命令列模式，以下將對命令列程式進行分析。如圖 9，當 Clam-AV 經由 AMaViS 喚起時，會先設定一些參數，接著進入圖 9 中的第 1 階段，讀取病毒特徵檔，再來進入第 2 階段，設定一些處理壓縮檔的特殊參數及檢查檔案的格式，當檔案為壓縮檔時，則將之解壓縮處理，然後進入第 3 階段，當開始要掃描以比對病毒特徵時，會先判斷要掃描的資料是目錄還是檔案，檔案則直接掃描，目錄則使用 TreeWalk 的方式掃描，最後若發現病毒時，再判斷是否需要隔離檔

案，如此即完成病毒偵測掃描的目的，然後回傳值給 AMaViS，AMaViS 再根據回傳值以判斷信件的處理方式，此即為 Clam-AV 的程式流程。

4. SpamAssassin 垃圾郵件過濾

SpamAssassin[7]是一套過濾郵件的程式，其能檢查郵件的內容是否符合 RFC 的相關規定，郵件中的檔頭是否正確，郵件中的內容是否出現特殊的關鍵字等疑似垃圾信的情況，然後查詢設定檔的設定得到一些特定的分數，最後將有檢查到疑似垃圾信的情況之分數累計若超過某個設定值，則視為垃圾郵件，除此之外，還支援線上資料庫比對，黑名單檢查…等等，功能相當齊全，以下將詳細介紹其運作流程及程式的運作模式。

4.1 Mail 與 SpamAssassin 的運作流程介紹

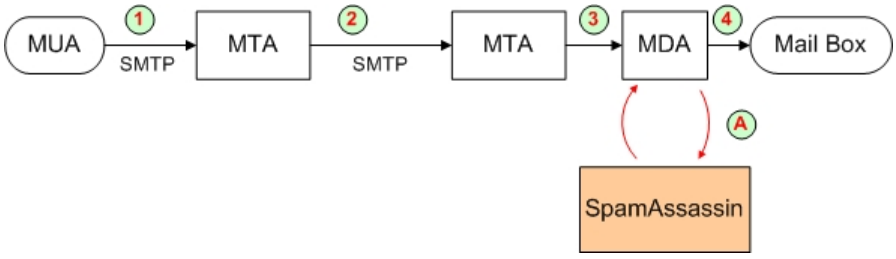

圖 10：Mail 與 SpamAssassin 的運作流程圖

SpamAssassin 目前是與郵件系統結合運作的。在圖 7 提到郵件系統正常寄送流程，此節延續圖 7 所提的方法，郵件系統採用如圖 10 第 1 到第 4 階段正常的寄送郵件，當要加入郵件過濾程式時，需利用 Mail Delivery Agent (MDA)[8]喚起 SpamAssassin 程式處理，當郵件經過處理後，會在信件的表頭檔中留下資訊，MDA 再利用表頭檔的資訊判斷是否刪除或對其信件的標題做標示以區別垃圾郵件，此即為整個 Mail 與 SpamAssassin 的溝通運作流程。

4.2 SpamAssassin 的程式流程分析

SpamAssassin 套件中有提供兩種模式可以讓 MDA 呼叫使用，一種是單一的命令列程式 SpamAssassin，另一種則採用 client–server 架構，MDA 呼叫一個

名為 Spamc 的 client 端程式使其連到名為 Spamd 的 Daemon 程式上，做內容過濾處理。如表 4 所示，SpamAssassin 處理 11689byte 大小的訊息，需要花費 3.36 秒，相對的 Spamc/Spamd 處理則只需 0.86 秒。當處理的訊息大小變大為 115855byte 時，SpamAssassin 需要花費 5 秒的時間，Spamc/Spamd 則只需要 2.5 秒。可以看出後者的執行效率比前者高出許多，在同一時間內所能處理的資料量也比前者多，故使用時皆建議採用後者。接下來，我們以表 5 描述 Spamc 與 Spamd 兩者是如何溝通運作的。整個運作模式與 HTTP protocol 類似，透過訊息的傳遞，達到溝通及處理的目的。

Program / Msg Size	SpamAssassin	Spamc/Spamd
11689 byte	3.36 (s)	0.86 (s)
115855 byte	5 (s)	2.5 (s)

表 4：SpamAssassin 與 Spamc/Spamd 效能比較表

conversation looks like:	
	spamc --> PROCESS SPAMC/1.2
	spamc --> Content-length: <size>
(optional)	spamc --> User: <username>
	spamc --> \r\n [blank line]
	spamc --> --message sent here--
	spamd --> SPAMD/1.1 0 EX_OK
	spamd --> Content-length: <size>
	spamd --> \r\n [blank line]
	spamd --> --processed message sent here--

表 5：Spamc/Spamd 之溝通模式

了解溝通原理後，再來我們介紹 Spamc/Spamd 整個程式的運作流程。如圖 11 所示，當 Spamc 被 MDA 喚起時，首先程式會先做參數的設定及決定執行程式的使用者，此步驟的主要用意在於可以讓各個使用者有各自的設定檔，然後進入第 2 階段，Spamc 讀取 MDA 傳來的郵件訊息，再來進入第 3 階段，Spamc 連

到 Spamd 上，並把相關資料傳給 Spamd，Spamd 則進行一些比對，比對之內容視設定檔而定，Spamd 比對後將結果回傳給 Spamc，Spamc 再以回傳之結果為根據，傳送訊息給 MDA，如此即完成郵件過濾之目的。Spamd 程式的運作流程，如圖 12 所示，一開始先等待 Spamc 的連線，當有連線進入時，則進入第 1 階段，判對是否處理個人設定檔，不需處理時，則直接進入第 4 階段，讀取 Spamc 傳來的訊息內容。需處理時則適時的載入，並轉換為此使用者之帳號，再來進入第 4 階段，然後讀取 Spamc 傳來的訊息內容。最後進行資料比對，判斷是否為垃圾信件，再將結果回傳給 Spamc，此為 Spamd 的程式運作流程。另一命令列程式則直接做一些比對的動作，在此則不詳加描述。以上是對代理伺服器提供郵件過濾功能的介紹。

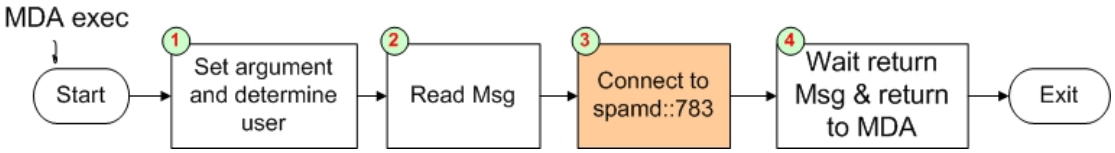

圖 11：Spamc 程式流程圖

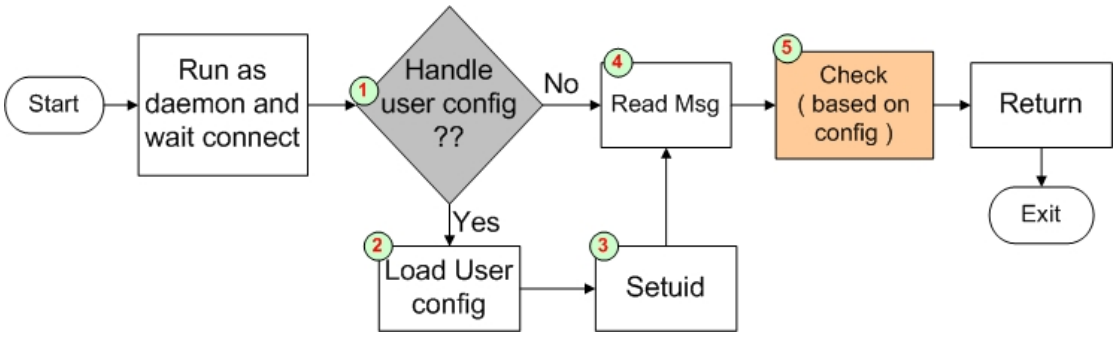

圖 12：Spamd 程式流程圖

5. 結論

在文章中，我們介紹了三個著名的代理伺服器。其中 FWTK 提供完整的套件功能，讓使用者方便選取套件搭配，輕易的達到想要提供的代理伺服器功能。而 AMaViS+Clam-AV 提供郵件預先偵測病毒的功能，讓使用者免於遭到病毒的危害。SpamAssassin 提供郵件預先偵測垃圾信的功能，讓使用者免於受到垃圾信的困擾。各個套件有其用途，就看使用者如何搭配使用。

由以上之內容，可以了解到使用代理伺服器所帶來的便利，但是不是所有的

服務皆能改成使用代理伺服器的運作模式呢？是不是病毒偵測的工作能由使用者端移到代理伺服器上呢？相信這是個值得思考的問題。若從各個程式的設計角度看來，則皆有改善的空間。如在 FWTK 方面，設定檔的重複讀取及內容過濾時一次讀取一個字元的做法，尚未讓處理效能發揮到淋漓盡致的境界。另外，網址過濾僅支援 HTTP 協定及僅能處理單一的網址過濾(URL Filter)變數，都是程式可以改善的地方。在 Anti-Virus 方面，解壓縮及病毒碼的比對是消耗處理時間的關鍵。換個角度思考，Clam-AV 使用 Aho-Corasick 字串比對演算法是否恰當呢？還有其他的演算法更適合嗎？在 Anti-Spam 方面，Spamc 與 Spamd 之間的檔案傳送，花費相當多的時間，是否有其他的替代做法呢？至於 Spamd 花在比對資料的時間相當可觀，且等待線上黑名單資料的回傳，更是影響處理效能的一大關鍵。

6. 參考文件

[1] 林逸祥、林盈達,「加速網頁讀取—快取軟體 Squid 測試」網路通訊,129 期，2002 年 4 月。

[2] FWTK, http://www.fwtk.org/ .

[3] DansGuardians, http://dansguardian.org/ .

[4] A Mail Virus Scanner, http://www.amavis.org/ .

[5] Clam Anti-Virus, http://clamav.elektrapro.com/ .

[6] OpenAntiVirus, http://www.openantivirus.org/ .

[7] SpamAssassin, http://spamassassin.org/ .

[8] Procmail, http://www.procmail.org/ .