透通式外寄郵件閘道器之建置與應用

范修維
世新大學資訊管理學系
fan@cc.shu.edu.tw

摘要

垃圾郵件防治是當前重要的資安工作之一，然而絕大部分的垃圾郵件防治均是針對內收郵件進行過濾，鮮少將外寄郵件納入考量，以致於外寄郵件常造成資安管控的死角。除了郵件過濾外，郵件的備分與監控也漸漸受到重視，尤其是具有敏感資料的單位，對此需求更甚。本文中，我們將提出一種透通式的外寄郵件閘道器之建置方案，在該閘道器上不僅可以套用現有的垃圾郵件防治技術對外寄郵件進行過濾，而且還可將這些郵件進行備分，便於日後進行稽核與追蹤。整個系統的建置成本非常低廉，只需一到數台伺服器即可完成。由於透通式的特性，使得既有的伺服器以及個人電腦完全不必更改設定，在推行上幾乎不會受到習慣改變而產生任何阻力。因此，本文所提之方案是一個可行而又值得推廣的方案。

關鍵詞：透通式、郵件閘道器、垃圾郵件、郵件備分、郵件監控、郵件稽核

Abstract

Today, anti-spam is the most important topic of information security. But, most of the anti-spam only filter with incoming mails. Therefore, the outgoing mails become the death corner of information security controlling. Moreover, backup and monitoring for the mails also are the important topics. In this paper, an implement strategy for transparent outgoing mails gateway is proposed. Our proposed mechanism could not only backup the mails, but also audit and trace the mails. Besides, the cost of our proposed system is very cheap and only uses one or several servers to implement. Furthermore, owing to the property of transparency, all servers and pc do not change any setup data. Therefore, the proposed mechanism is very practical for the real implementation and application.

Keywords: transparent mail gateway, anti-spam, mail backup, mail monitoring, mail auditing.

1. 前言

以現今的垃圾郵件辨識技術，大部分的垃圾郵件都能分辨出來，而且具有極低的誤判率。但是絕大多數的垃圾郵件防治工作都是針對內收郵件，鮮少針對外寄郵件進行防治。在此情形下，一旦單位內部對外發送垃圾郵件，不但難以即時偵測，而且也不易找到發送垃圾郵件的源頭。

試想下列四種情形：

1. 當使用者拿著隨身碟到公用電腦安裝垃圾郵件發送程式以賺取外快，如何偵測與防治？
2. 當使用者練習架設郵件伺服器時，卻忘記設定 Mail Relay 的限制，以致於被當作垃圾郵件轉寄站，該如何偵測與防治？
3. 當伺服器被駭客入侵而開啟 Mail Relay 的功能，該如何偵測與防治？
4. 當使用者誤觸病毒或被植入木馬，以致於對外大量發送垃圾郵件，該如何偵測與防治？

上述四種情形均為本校實際案例，為了處理這些事件，網管人員必須花費許多工夫才得以解決。在解決之前，這些事件對外已造成一定程度的傷害。如果我們可以建立一種透通式的外寄郵件閘道器，並且將垃圾郵件辨識技術套用其上，這些問題便可迎刃而解。

再考慮以下情形：

1. 當使用者寄出一封郵件之後，同時也把備分郵件刪除，但稍後使用者卻想找回剛剛所寄之郵件。如果使用者求助於網管人員，此時網管人員是否能給予適當的協助？
2. 當單位內部某個部門的主管懷疑機密資料已外洩，想要調查對外的通聯紀錄，此時網管人員能否協助列印電子郵件通聯清冊？
3. 如果已有明顯事証可以推斷使用者透電子郵件寄送不法資料，網管人員能否協助進行証據保全，把該使用者之外寄郵件備分起來，作為未來的呈堂証供？

如果我們可以建立透通式外寄郵件閘器，並且在上面設定郵件備分的功能，這些問題就可解決。

再考慮另一案例，某生利用學校對外的高速頻寬在校內大量下載不當檔案，隨後將這些檔案進行打包與切割，以附件的方式寄到多個 gmail 帳號，利用 gmail 巨大的郵件空間作為暫存地，回家後再透過 ADSL 慢慢下載。這種行為不但嚴重耗損網路頻寬，同時也嚴重浪費網路資源。此案例為本校真實案例，我們透過本文所提之機制迅速地偵測出此事件，並即時遏止此行為。

由以上各種事証顯示，對於外寄郵件之偵測與過濾顯然有其必要性。然而，這種外寄郵件之過濾

是否可直接套用現有的垃圾郵件過濾器，將內收郵件過濾器反向當作外寄郵件過濾器？其實不然，主要有下以幾點困難：

1. 許多內收郵件過濾器需要依賴 MX 紀錄的設定，才能將郵件導入過濾器進行分析。這類的郵件過濾機制將無法適用於外寄郵件之過濾，因為我們無法修改所有收信人主機的 MX 設定。
2. 許多內收郵件過濾器必須事先列出後端允許收信的郵件伺服器名稱，未在列表上的郵件伺服器一律不進行過濾，甚至直接丟棄郵件。這種設定在內收郵件過濾上不會造成太大的困擾，但是在外寄郵件過濾時，這種設定就變成無法完成的任務。
3. 某些郵件過濾的機制必須事先匯入收信人的帳號，不在列表的帳號一律不予收信。這類的過濾機制亦無法適用於外寄郵件的過濾。

綜合以上各點說明，重新設計一套外寄郵件的過濾機制是有其必要性。本文中，我們將提出一套透通式外寄郵件閘道器的建置方案，在此機制下，所有外寄郵件都會強制導入我們的郵件閘道器。在此閘道器上，我們可以套用既有的垃圾郵件辨識技術，以進行外寄郵件的過濾與隔離。此外，只要在該閘道器上作些適當的設定，就可進行外寄郵件的備分，以利日後的追蹤與稽核。

在接下來的章節裡，我們將介紹整個系統的架構。接著，在第三節中，我們將針對系統實作進行說明，包括透通式外寄郵件閘道器之建置方式、郵件過濾、郵件備分、效能調校、除錯機制與系統監控各項重點。最後，我們也將以本校的實作經驗進行探討，並說明未來的發展方向。

2. 系統架構

完整的系統架構如圖 1 所示，各部分的功能說明如下：

1. 郵件導入：
   由 Layer 4 Switch 或由 Policy Route 將對外 TCP Port 25 的傳輸導入郵件閘道器。
2. 位址轉換：
   將郵件傳輸的目的地位址轉換為本機位址，使得本機的 SMTP 程式得以攔截該郵件。
3. 郵件接收
   攔截外送郵件，以便進行第一階段之郵件備分，並將郵件派送到辨識引擎進行垃圾郵件辨識。必要時可將收件人地址進行改寫，以便利用內收郵件過濾器作為郵件辨識引擎。

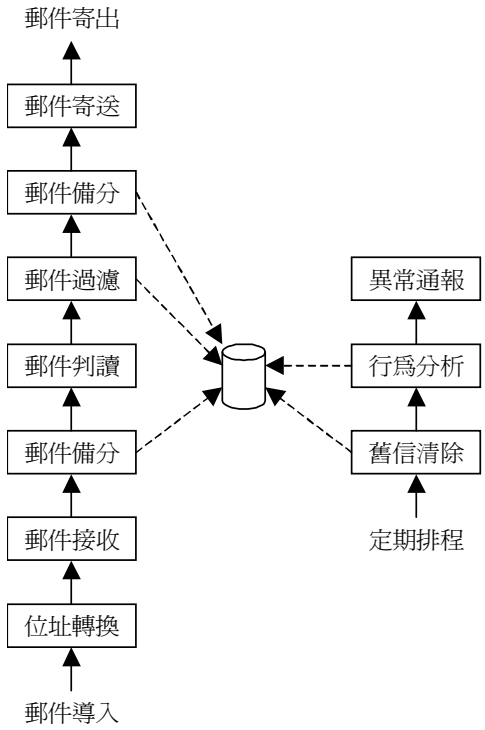

圖 1 外寄郵件閘道器之系統架構圖

4. 郵件備分
   此為第一階段之郵件備分，此處之備分將包含正常郵件與垃圾郵件。
5. 郵件判讀
   將所收到的郵件進行判讀，以辨識是否為垃圾郵件 (或病毒郵件)。此部分可套用既有的郵件辨識引擎進行辨識。
6. 郵件過濾
   根據郵件辨識引擎的分析結果，進行郵件的過濾。這部分可依照管理者之政策決定是否刪除、隔離或放行。一些系統的黑白名單、特殊規則也可在此階段套用。
7. 郵件備分
   此為第二階段之備分，對象為系統所外寄之郵件 (不包含被過濾之垃圾郵件)。
8. 郵件寄送
   此為外寄郵件之 SMTP 服務程式，負責對外寄送郵件。如果之前曾改寫收信人地址 (為了套用既有的內收郵件過濾器進行過濾)，在此階段也必須將該地址還原。
9. 定期排程
   利用作業系統的排程機制定期執行舊信清除、行為分析、異常通報等工作。
10. 舊信清除
    此部分可按照管理者之政策，決定保留舊信備分的天數，超過的部分則予以清除。
11. 行為分析
    針對郵件的傳輸量、郵件數、寄信人、收信人等數量進行分析，以判斷是否出現異常現象。此部分主要為經驗法則，根據管理者之經驗所

進行的分析與判斷。

12. 異常通報
當異常事件發生時，可採取適當的通報機制，如電子郵件通知、手機簡訊通知。

上述各個功能並非需要全部實作，可視實際情形進行增刪，例如：第一階段之備分與第二階段之備分二者擇一施行即可，毋須二者同時施行。

在實作上，這些功能可以在 Linux、FreeBSD 等系統進行建置。在工具的選擇方面，我們以 sendmail [8]作為郵件收發的服務程式，搭配 procmail [7] 的郵件處理能力，進行郵件的備分與過濾。此外，我們亦採用 apache [3]網頁伺服器搭配 openwebmail [2] 作為郵件檢視工具。至於垃圾郵件辨識系統，我們採用 nopam [1]系統作為核心引擎。整個系統功能可以作適當的切割，放置在不同的伺服器，以降低系統的負荷。例如：分割成四台伺服器是個不錯的方法 (圖 2)：

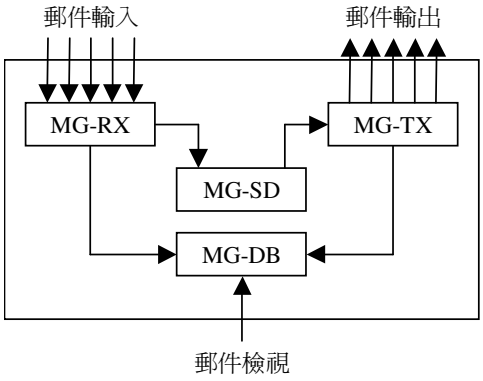

圖 2 以四台伺服器建置外寄郵件閘道器之示意圖

1. 郵件攔截伺服器 (MG-RX)
此部分主要工作為攔截外寄的郵件，可視需要決定是否進行第一階段之備分，並送往後端的郵件辨識引擎進行垃圾郵件判讀。必要時，可以對郵件收信地址作改寫，以符合垃圾郵件過濾器之要求 (圖 3)。

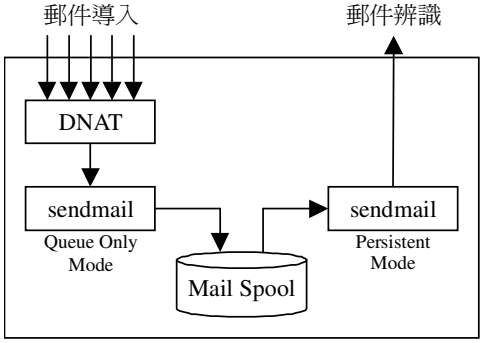

圖 3 郵件攔截伺服器 (MG-RX) 內部功能示意圖

2. 郵件辨識伺服器 (MG-SD)
此部分主要工作為進行郵件辨識，以判斷是否為垃圾郵件或病毒郵件。可套用既有之商用郵件過濾器或免費的郵件過濾引擎來實作。在此我們以 nopam 系統作為辨識引擎 (圖 4)。

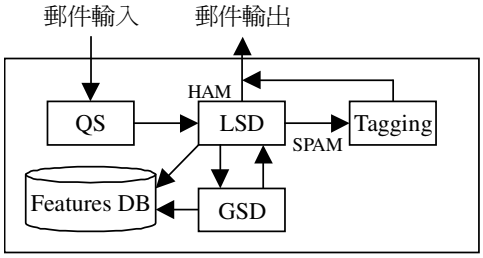

圖 4 郵件辨識伺服器 (MG-SD) 內部功能示意圖

3. 郵件備分伺服器 (MG-DB)
此部分主要功能為提供充足的儲存空間，以存放所備分之郵件。此外，亦可對所備分之郵件進行分析，以偵測是否有異常事件發生 (圖 5)。

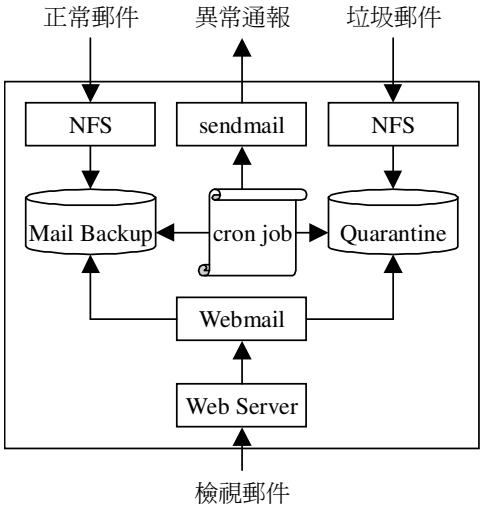

圖 5 郵件備分伺服器 (MG-DB) 內部功能示意圖

4. 郵件寄送伺服器 (MG-TX)
此部分主要工作為接受郵件辨識引擎分析後的郵件，再按管理者之政策決定郵件放行與否 (圖 6)。

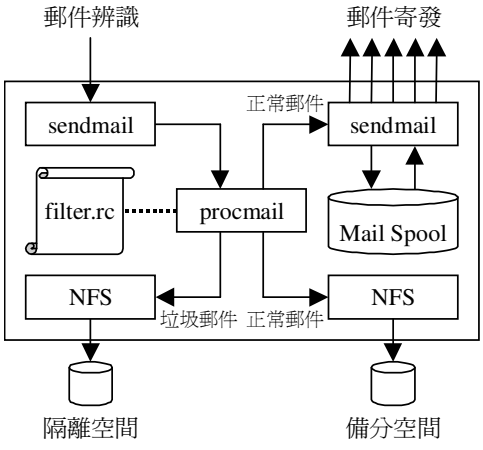

圖 6 郵件寄送伺服器 (MG-TX) 內部功能示意圖

透通式外寄郵件閘道器雖然可採用線上模式

(inline mode)，但此種方法會有一些缺點：

1. 多了一層網路設備會增加所有網路封包傳輸的延遲。
2. 一旦郵件閘道器故障 (如斷電)，所有網路傳輸均因而中斷。
3. 不利於維護與測試，一旦更改閘道器組態，很可能造成其它網路傳輸的副作用。

因此，建議由 Layer 4 Switch 或具有 Policy Route 的路由器將外寄郵件的傳輸全部導入郵件閘道器。此時，整體網路架構如圖 7 所示。

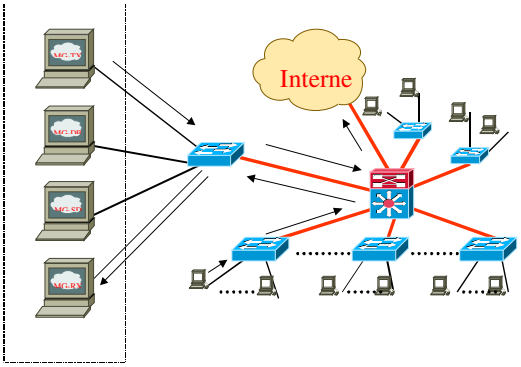

圖 7 套用外寄郵件閘道器之網路架構

對於流量較小的單位 (例如中小學或中小企業)，不一定要建置四台伺服器，可以將所有功能實作於一台伺服器。只是在此情形下，其設定會比較複雜 [6]。

3. 系統實作

以下我們針對系統實作的各個細節進行討論。

3.1 透通式外寄郵件閘道器之建置

在郵件導入方面，以 Cisco 6509 為例，只要使用簡單的 Route-Map 即可完成此項工作[4]：

```
access-list 101 permit ip any 192.192.148.0 0.0.3.255
access-list 101 permit ip any 192.192.152.0 0.0.7.255
access-list 101 permit ip any 172.16.0.0 0.15.255.255
access-list 170 deny tcp host 192.192.150.77 any
access-list 170 deny tcp host 192.192.150.78 any
access-list 170 deny tcp host 192.192.150.79 any
access-list 170 deny tcp host 192.192.150.80 any
access-list 170 permit tcp any any eq smtp
route-map SPAM permit 10
  match ip address 101
  set ip precedence routine
route-map SPAM permit 16
  match ip address 170
  set ip precedence routine
  set ip next-hop 192.192.150.77
```

為了讓郵件閘道器得以攔截郵件，必須為其加上 DNAT 的轉換，以便將目的位址轉換成本機位址。這部分的工作可在 Linux 上藉由 iptables [5] 來完成：

```
iptables -t NAT -A PREROUTING -p tcp --dport 25 -j REDIRECT --to-ports 25
```

在郵件服務程式方面，我們採用傳統的 sendmail 進行實作。這部分不需要額外的設定，只要開放接受單位內部的郵件轉寄 (Mail Relay) 即可。

至此，我們已經擁有透通式外寄郵件閘道器，只是此閘道器至此僅能作為 Mail Relay，尚無其它附加功能。接下來，我們將要在此閘道器上繼續增加郵件過濾與郵件備分等功能。

3.2 郵件過濾

當外寄郵件閘道器攔截到外送的郵件後，必須設法將該郵件送往郵件辨識引擎作判讀，才能分辨是否為垃圾郵件。如果判讀為一般郵件，則予以放行；如果判讀為垃圾郵件，則可按管理者之政策決定是否過濾、隔離或放行。這部分需要修改 sendmail.mc 設定檔：

```
FEATURE(stickyhost)dnl
define(`LOCAL_RELAY', `esmtp:[192.192.150.79]')dnl
define(`MAIL_HUB', `esmtp:[192.192.150.79]')dnl
define(`SMART_HOST', `esmtp:[192.192.150.79]')dnl
define(`ESMTP_MAILER_ARGS',`TCP $h 10024')dnl
```

在郵件辨識方面，我們採用了 nopam 垃圾郵件過濾引擎。該軟體是一套建構於 FreeBSD 之免費軟體，對垃圾郵件的辨識率相當高、且誤判率極低。在辨識之後，可在郵件主題加上 ***Spam*** 的標籤，以方便讓後端的過濾機制進行處理。

在後端的郵件過濾部分，必須在 sendmail.mc 引入 procmail 的郵件處理機制：

```
MODIFY_MAILER_FLAGS(`PROCMAIL', `+f')dnl
LOCAL_CONFIG
CPprocmail
LOCAL_RULE_0
R$*<@$+.procmail.>$*    $@ $1<@$2.>$3
R$*<@$+.>$*          $#procmail          $@/etc/mail/filter.rc
                     $:$1<@$2.procmail>$3
```

引入 procmail 郵件處理機制後，即可在 filter.rc 中加入過濾敘述：

```
SUBJECT=`formail -xSubject: \
        | sed -e 's/[;\`\\]/ /g' \
        | expand | sed -e 's/^[ ]*//g' -e 's/[ ]*$//g`
```

```
:0
* $SUBJECT ?? "^\\*\\*\\*Spam\\*\\*\\*"
{
  :0 c
  ! -C /etc/mail/sendmail.cf -f "$SENDER"
                    "yy@xx.shu.edu.tw.procmail"

  :0 c:
  $DIR/spam/$FROM

  FOR = `formail -xReceived: | grep "for "`

  :0 : $DIR/spam.mbx.lock
  * ! $FOR ?? "shu.edu.tw"
  * ! $SUBJECT ?? "This is an autoreply"
  | formail >> "$DIR/spam.mbx"
}

:0 w
! -C /etc/mail/sendmail.cf -f "$SENDER" "$@"
```

在上述的範例中，我們把判定為垃圾的郵件全部寄送一份到 yy@xx.shu.edu.tw 帳號，並且在本機的 spam 目錄亦保留一份副本。對於單位內部以 .forward 轉寄到外部的郵件、以及開啟自動回覆功能的垃圾郵件都作了特別的處理，因為這些郵件是來自於外部，非源自內部之不當使用。

3.3 郵件備分

當 sendmail 引入了 procmail 郵件處理機制後，我們即可利用 filter.rc 進行郵件的備分。備分時，可以針對每個寄信人進行不同的檔案備分，也可以針對不同的收信人進行不同的檔案備分：

```
DIR = "/mnt1/MAIL/current"
FROM="$1"
SHIFT = 1
TO="$1"

:0 c
$DIR/from/$FROM

:0 c
$DIR/to/$TO
```

同時啟用這二種備分模式將會損耗二倍的儲存空間，但是在進行郵件的稽核與除錯時，會有較大的幫助。

我們可以藉由定期排程機制固定每天執行下列命令，讓每天的郵件備分到不同的目錄：

```
mkdir -p /mnt1/MAIL/`date +%F`/{from,to}
rm /mnt1/MAIL/current
ln -s /mnt1/MAIL/`date +%F` /mnt1/MAIL/current
```

當使用者想要調閱資料時，管理者可以很快速地把信夾調閱出來，例如：

```
cd /mnt1/MAIL/2006-07-01/from
more fan@cc.shu.edu.tw
```

3.4 效能調校

雖然郵件攔截伺服器與郵件寄送伺服器都提供了 SMTP 服務，但這二者的服務對象、連線來源、郵件傳送速度、服務可用度的要求…幾乎都迥然有異。前者的服務對象是單位內部的使用者，連線來源幾乎均在 LAN 之內，郵件傳送速度快，服務可用度要求較高。在此情形下，應該避免讓伺服器的負載增高而暫停接受 SMTP 的連線請求。此時，我們可以修改 sendmail 的送信模式，由預設的 foreground 模式改為 Queue Only 模式，再以獨立單一的程序一封封地將郵件佇列郵件送給後方的郵件辨識引擎。這個方法的好處是在相同的 CPU 負荷下提供更高的收信能力，以服務更多用戶端的寄信請求，並且可以很平順地將郵件依序地送給後端郵件辨識引擎，不易產成負荷瞬間爆增。這部分可藉由下列步驟來達成：

```
# vi sendmail.mc
define(`confDELIVERY_MODE',`q')dnl
# m4 sendmail.mc > sendmail.cf
# /usr/sbin/sendmail -bd -qp
```

至於郵件寄送伺服器則不適宜採用 Queue Only 模式，因為在寄送郵件時，常可能因為網路擁塞或其它因素而造成連線時間過久、甚至逾時，若採用 Queue Only 模式將導致排在其後的郵件產生過久的寄信延遲，因此仍然必須採用預設的 foreground 模式。但是在此模式下，可能會產生過多的 sendmail 程序而造成系統負荷昇高，此時可考慮降低關閉服務的門檻。即使系統因負荷昇高而暫停接受 SMTP 的連線請求，也不會造成太大的困擾，因為該伺服器的前一級是郵件辨識伺服器，它可以在 SMTP 的服務恢復後繼續寄出郵件。另外，除了降低關閉服務的門檻值之外，其它的連線參數也可以適度地調整，以縮短逾時的等待時間。這部分可藉由修改 sendmail.mc 的設定來達成：

```
define(`confTO_ICONNECT', `10s')dnl
define(`confTO_CONNECT', `30s')dnl
define(`confTO_INITIAL', `1m')dnl
define(`confTO_HELO', `30s')dnl
define(`confTO_MAIL', `30s')dnl
define(`confTO_RCPT', `30s')dnl
define(`confTO_DATAINIT', `30s')dnl
define(`confTO_DATABLOCK', `30s')dnl
define(`confTO_DATAFINAL', `30s')dnl
define(`confTO_RSET', `30s')dnl
define(`confTO_QUIT', `30s')dnl
define(`confTO_MISC', `30s')dnl
```

```
define(`confTO_QUEUERETURN', `12h')dnl
define(`confTO_IDENT', `0')dnl
```

3.5 除錯機制

在系統建置之後，可能會面臨一些用戶的挑戰，例如：郵件遺失、郵件延遲過久。若要進一步追蹤郵件的傳輸情形，可以開啟 sendmail 的全文紀錄功能進行除錯，它可套用在下列幾個階段：

1. 當郵件閘道器收到所攔截的郵件時。
2. 當郵件閘道器將郵件送給辨識引擎時。
3. 當郵件閘道器收到辨識後的結果時。
4. 當郵件閘道器將郵件對外寄出時。

透過這四個郵件全文紀錄的檢查，可以判斷出問題發生於哪一個階段。若要啟動 sendmail 全文紀錄，只要在該程序之後加入「-X」的參數即可，例如：

```
# /usr/sbin/sendmail -bd -qp -X /mnt1/MAIL/current/rx.log
```

除了 sendmail 的全文紀錄功能之外，亦可開啟 procmail 的詳細紀錄功能，以便偵測是否因為濾信條件撰寫不當而產生問題。這部分可在 filter.rc 中加入下列敘述：

```
DIR = "/mnt1/MAIL/current"
LOGFILE=$DIR/procmail.log
VERBOSE=on
LOGABSTRACT=on
```

3.6 系統監控

除錯機制的開啟之後，可能造成儲存空間的大量損耗。為了避免因儲存空間用盡而造成服務停擺，我們可設定簡單的監控制機，當硬碟空間頻臨用盡之前，立即通知系統管理人員。這部分可藉由定期排程機制檢視硬碟剩餘空間來預防：

```
if [ `df -k | grep /$        | sed 's/.* \([0-9]*\)% .*/\1/` -gt 90 ] ||
   [ `df -k | grep /usr$     | sed 's/.* \([0-9]*\)% .*/\1/` -gt 90 ] ||
   [ `df -k | grep /var$     | sed 's/.* \([0-9]*\)% .*/\1/` -gt 90 ] ||
   [ `df -k | grep /mnt1$ | sed 's/.* \([0-9]*\)% .*/\1/` -gt 90 ] ||
   [ `df -k | grep /mnt2$ | sed 's/.* \([0-9]*\)% .*/\1/` -gt 90 ]
then
     df -k | mail -s "`hostname` 主機硬碟空間快爆了" \
                                fan@cc.shu.edu.tw
fi
```

如果我們希望系統偵測到垃圾郵件時，就把最新的報告寄給管理者，則亦可透過排程機制來進行：

```
SPAM="/mnt1/MAIL/current/spam.mbx"
ATIME=`stat -c %X $SPAM`
MTIME=`stat -c %Y $SPAM`
if [ "$ATIME" -le "$MTIME" ]
then
    formail -l $SPAM -s < $SPAM | mail -s "今日最新外寄垃圾郵件報表" fan@cc.shu.edu.tw
fi
```

3.7 上線實測

本文所述之透通式外寄郵件閘道器已在本校實際運作四個月，對象含蓋了校內所有的電腦。建置之初採用三台伺服器，後來因為負載很輕而合併為一台伺服器。這台伺服器之規格非常普通，P4 2.8G CPU、1GB RAM、SATA 160GB x 2、100M bps NIC x 1，硬體經費不到六萬元。

建置之後，我們成功地攔截病毒或木馬直接對外的 SMTP 連線請求，因而防止了病毒郵件對外的寄送。在郵件稽核方面，我們也從郵件備分量的異常偵測出使用者在濫用郵件 (前文所述的 gmail 事件)，進而阻止事件的持續發生。在郵件備分上，也曾遇到校內利潤單位主管要求進行外寄郵件的証據保全，以便作為日後對質之依據。

4. 結論與未來發展方向

透通式外寄郵件閘道器可攔截單位內部對外所傳送之郵件，並且對這些郵件進行分析、過濾與備分。由於採用透通式技術，單位內部所有電腦均不必更改設定，也不容易感覺到閘道器的存在。本文所採用之軟體均無需費用，伺服器也沒有特殊需求，因此建置成本非常低廉。在管理者技能上，只要對 sendmail 與 procmail 有一定程度的瞭解，以及會撰寫簡易的 shell script，均可輕鬆完成建置。建置完成後，不僅可以降低管理者的負擔，也可以提供使用者更好的服務，因此非常值得推廣。

在異常偵測的部分，仍有非常大的發展空間。如何對所備分的郵件進行各種分析，以期更早發現問題，這是未來的發展方向。

參考文獻

[1] http://nopam.ccu.edu.tw/
[2] http://openwebmail.org/
[3] http://www.apache.org/
[4] http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fiprrp_r/ind_r/1rfindp2.pdf
[5] http://www.netfilter.org/
[6] http://www.ijs.si/software/amavisd/README.sendmail-dual
[7] http://www.procmail.org/
[8] http://www.sendmail.org/