使用 FreeBSD 建立安全的郵件系統 – Sendmail 篇

中央研究院 計算中心 張毓麟 ylchang@sinica.edu
(本文刊載於 中央研究院 計算中心通訊 17 卷 13 期)

前言

傳統的郵件傳送協定(SMTP)並不具備使用者身分認證的功能，相當容易被有心人士濫用爲廣告郵件或垃圾郵件的轉送點。而沒有身分認證功能的郵件傳送機制，也造成系統管理或是網路管理人員，處理問題郵件追蹤時的困難。

因此大多數的郵件系統都拒絕爲內部可信賴區域以外的使用者轉送郵件(mail relay)，但這種限制也造成了合法使用者使用上的不方便。例如出差或是下班回家之後，無法繼續使用公司的郵件伺服器發信。學生放假離開學校之後也不能使用學校的郵件主機發出信件。

以往要解決這個問題，必須購買一些昂貴的商業郵件伺服器(例如微軟的 Exchange Server)，以便在使用者發出信件前先進行身份的認證。但是現在使用新版本的 Sendmail 郵件伺服軟體，搭配 Cyrus SASL 認證程式庫，即可達成往日商業軟體才有的身分認證功能。這種具有身分認證的郵件伺服器稱爲 Authenticated Mail Server。

爲了進行使用者身分認證，勢必採用帳號(account)與通行碼(password)或類似的認證機制。如果進行認證的時候，帳號與通行碼以明文(clear text)的方式在網路上傳遞，則十分容易被有心的駭客擷取，認証功能反而爲駭客開一個更方便的大門。基於保護合法使用者以及網路資源的前提之下，保密連線(Encrypted Connection)就成爲一個必備的功能。

本文所採用的 Sendmail 郵件伺服軟體亦具備與 OpenSSL 安全程式庫搭配的能力。利用 OpenSSL 程式庫的功能，不只提供使用者安全連線(SSL/TLS)避免帳號密碼外洩。在 Email Server 與其他機器連線傳送郵件的時候，也會儘可能的使用最安全的保密連線方式，讓使用者的郵件在 Internet 上傳送的時候更加安全。

接下來，將要詳細的解說使用 FreeBSD 4.3-STABLE 版作業系統，架設具有保密連線功能的 Authenticated Mail Server 的詳細操作步驟。

爲什麼選用 FreeBSD

在幾個免費的 PC-UNIX(或 Linux)作業系統中，*BSD 家族(包括 FreeBSD/ OpenBSD/ NetBSD) 擁有其他 OS 望塵莫及的優點。當其他的 PC-UNIX(或 Linux)作業系統還在爲系統整合煩惱，並且爲系統中大小漏洞疲於奔命的時候，*BSD 家族早已完成系統整合，並將眼光放在打造鋼鐵般強健的多平台作業系統的目標上。

*BSD 除了以架構嚴謹、管理便利著稱以外，設計時對系統安全性近乎瘋狂偏執的追求，使他成爲 PC 上最難入侵的作業系統。再加上沿襲 4BSD 在網路與高系統負荷方面優異表現的血統，*BSD 不只適合擔任個人 DeskTop 的工作平台，作爲 Internet Service 伺服器更是不二人選。

FreeBSD 是在國內最容易取得的*BSD 家族作業系統。也有大量的中英文文件以及書籍可供參考(其實*BSD 的文件幾乎是完全通用的)。此外，FreeBSD 提供的安裝介面操作上較爲便利，對於習慣微軟軟體的使用者而言，轉換上比較沒有障礙。

在本文中所需要使用的軟體與程式庫，都已經被收錄在 FreeBSD 超過 5700 套軟體的軟體庫(ports)之中。只需要下幾個簡單的指令即可安裝，並且都提供設定的範例與詳細說明文件。

預備程序

由於 Sendmail 與 OpenSSL 都是 FreeBSD 4.3-STABLE 版作業系統內建的功能，因此不需要額外的安裝手續，只需要根據我們的需要進行調整即可。爲了進行調整，需要有 FreeBSD 4.3-STABLE 的 OS Source Code。如果您的系統安裝時沒有將 OS Source Code 安裝進去，請執行 /stand/sysinstall 這支工具程式進行補充安裝，或是查閱 http://www.freebsd.org/ 網站上的說明來安裝。

保密連線的金鑰(key-pair)與授權憑證(CA; Certification Authority)

通常，我們會向獨立公正單位(例如 VeriSign 或 GlobalSign 等等公司)購買安全金鑰以及授權憑證。但如果不願意花錢購買，也可以自行製作金鑰以及授權憑證。自行製作的安全金鑰與授權憑證，在功能上與買來的相同，但是向獨立公正單位購買，會讓使用者心理上覺得比較有保障。

安全連線至少需要三個檔案才能啓用. 請將公正單位核發的 key-pair 與 CA 放置於下列目錄

```
server 端的 key-pair, 存放於 /etc/mail/cert/mykey.pem
server 端的 CA, 存放於 /etc/mail/cert/mycert.pem
公正單位的 CA, 存放於 /etc/mail/cert/cacert.pem
```

請注意！如果使用由公正單位所發出的 key-par，務必通知公正單位不可將 mykey.pem 做 DES 編碼，否則 sendmail 將無法於開機時自動啓動。

如果想自行製作 key-pair 與 CA，請依照下列指令操作

```
# mkdir /usr/local/CA
# cd /usr/local/CA
# mkdir certs crl newcerts private
# echo "01" > serial
# cp /dev/null index.txt
# cp /etc/ssl/openssl.cnf openssl.cnf
```

編輯 openssl.cnf 檔案, 將檔案中約第 38 行的路徑設定由 ./demoCA 改成 /usr/local/CA。接著執行以下指令，假裝自己是公正單位，做一個 cacert.pem 出來。請按螢幕上的指示，輸入相關的系統資料，當螢幕上提示輸入公正單位密碼(PEM pass phrase)的時候，請自行設定一個密碼，並請牢記這個密碼，以便日後使用。

```
# cd /usr/local/CA
# openssl req -new -x509 -keyout private/cakey.pem \
-out cacert.pem -days 365 -config openssl.cnf
```

執行以下指令，建立 server 端的 CA 與 key-pair。請特別留意，在這個步驟中，當畫面上提示輸入 Common Name 的時候，請務必輸入機器的全名(FQDN)，否則以後使用者連線的時候將會出現警告訊息，造成使用者的困擾。當螢幕提示輸入 PEM pass phrase 的時候，請輸入上一步驟中的公正單位密碼。

```
# cd /usr/local/CA
# openssl req -nodes -new -x509 -keyout mykey.pem \
-out myreq.pem -days 365 -config openssl.cnf
# openssl x509 -x509toreq -in myreq.pem -signkey mykey.pem \
```

```
-out tmp.pem
# openssl ca -config openssl.cnf -policy policy_anything \
-out mycert.pem -infiles tmp.pem
# rm -f tmp.pem
```

以下列指令，將 key-pair 與 CA 複製到 /etc/mail/cert 目錄之下，並設定正確權限

```
# mkdir /etc/mail/cert
# cp /usr/local/CA/mykey.pem /etc/mail/cert/
# cp /usr/local/CA/mycert.pem /etc/mail/cert/
# cp /usr/local/CA/cacert.pem /etc/mail/cert/
# chmod og-rwx /etc/mail/cert/mykey.pem
# chmod og=r /etc/mail/cert/mycert.pem
# chmod og=r /etc/mail/cert/cacert.pem
```

使用以下的指令建立 CA 的 hash link，請特別注意引號的方向(建議剪貼以下指令，以免不小心打字失誤)

```
# cd /etc/mail/cert
# ln -s cacert.pem `openssl x509 -noout -hash < cacert.pem`.0
```

這樣就完成了自行建立 key-pair 與 CA 的程序。這組 key-pair 與 CA 將可被 sendmail 使用於保密連線的資料加密功能上。

軟體與程式庫安裝設定

以下所有的操作程序，請特別留意大小寫的不同。大小寫與空白的混淆，將造成操作上的失敗。

Cyrus SASL 程式庫

Cyrus SASL 程式庫由卡內基美隆大學(Carnegie Mellon University)發展，被收錄在軟體庫的系統安全分類中，安裝的指令如下

```
# cd /usr/ports/security/cyrus-sasl
# make install
```

Cyrus-SASL 程式庫安裝之後，會自動建立 Sendmail 運作時需要的設定檔，完全不需要進行任何調整，只要以下列指令啓動 password checking 介面即可

```
# /usr/local/etc/rc.d/pwcheck.sh start
```

Sendmail 設定調整

系統雖然內建 Sendmail，但內建的版本並沒有與 Cyrus-SASL 程式庫連結，因此必須加以調整。以下的指令將爲調整 Sendmail 做準備

```
# killall -9 sendmail
# cat >> /etc/make.conf
SENDMAIL_CFLAGS=-I/usr/local/include/sasl -DSASL
SENDMAIL_LDFLAGS=-L/usr/local/lib
SENDMAIL_LDADD=-lsasl
^D （同時按 Ctrl 鍵與 D 字母鍵）
```

此外，我們還希望在系統的紀錄檔中，能明確的紀錄某位使用者在幾點幾分使用本 Server 發送信件，因此以下列指令取得筆者提供的小小修正檔，並將其與系統的 Sendmail Source Code 整合

```
# cd /tmp
# fetch http://beta.wsl.sinica.edu.tw/~ylchang/Email/patch/
sendmail.logfile.header.patch （請與上一行不空格連接）
# cd /usr/src
# patch < /tmp/sendmail.logfile.header.patch
```

以下列指令安裝新的 Sendmail 程式

```
# cd /usr/src/lib/libsmutil
# make clean
# make
# cd /usr/src/usr.sbin/sendmail
# make clean
# make all install
```

完成上述指令後，sendmail 的安全連線與身分認證功能已經被啓動，但仍需進一部的設定才能正常運作。

編輯 /etc/mail/freebsd.mc 檔案，在檔案末端加入以下 11 行設定

```
dnl The following lines are used to enable the STARTTLS function
define(`CERT_DIR', `/etc/mail/cert')dnl
define(`confCACERT_PATH', `CERT_DIR')dnl
define(`confCACERT', `CERT_DIR/cacert.pem')dnl
define(`confSERVER_CERT', `CERT_DIR/mycert.pem')dnl
define(`confSERVER_KEY', `CERT_DIR/mykey.pem')dnl
define(`confCLIENT_CERT', `CERT_DIR/mycert.pem')dnl
define(`confCLIENT_KEY', `CERT_DIR/mykey.pem')dnl
dnl The following lines are used to enable CYRUS-SASL function
TRUST_AUTH_MECH(`LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `LOGIN PLAIN')dnl
```

請特別注意大小寫，以及引號的方向，建議使用複製的方式將以上 11 行設定貼入 /etc/mail/freebsd.mc 檔案的末端，以免發生意外的打字錯誤。接下來以下列指令，建立新的 sendmail 控制檔案(cf file)

```
# cd /etc/mail
# make cf
# make install
```

完成以上設定程序之後，sendmail 的保密連線與身分認證功能已經被正確的設定完成，可以進行下一步驟，檢查是否正確運作。

檢查保密連線與身分認證的功能是否正確運作

首先，以下列指令重新啓動 sendmail

```
# killall -9 sendmail
# /usr/sbin/sendmail -bd -q30m
```

以下列指令檢查保密連線與身分認證功能是否正確啓動

```
# telnet localhost 25
ehlo localhost
```

若螢幕上出現的訊息包含以下兩行，則表示保密連線與身分認證功能已經正確啓動了。

```
250-AUTH LOGIN PLAIN
250-STARTTLS
```

若沒有出現這兩個訊息，表示前述的操做發生了錯誤，請檢查 sendmail 的記錄檔 /var/log/maillog 內的訊息，了解錯誤發生的詳細狀況。

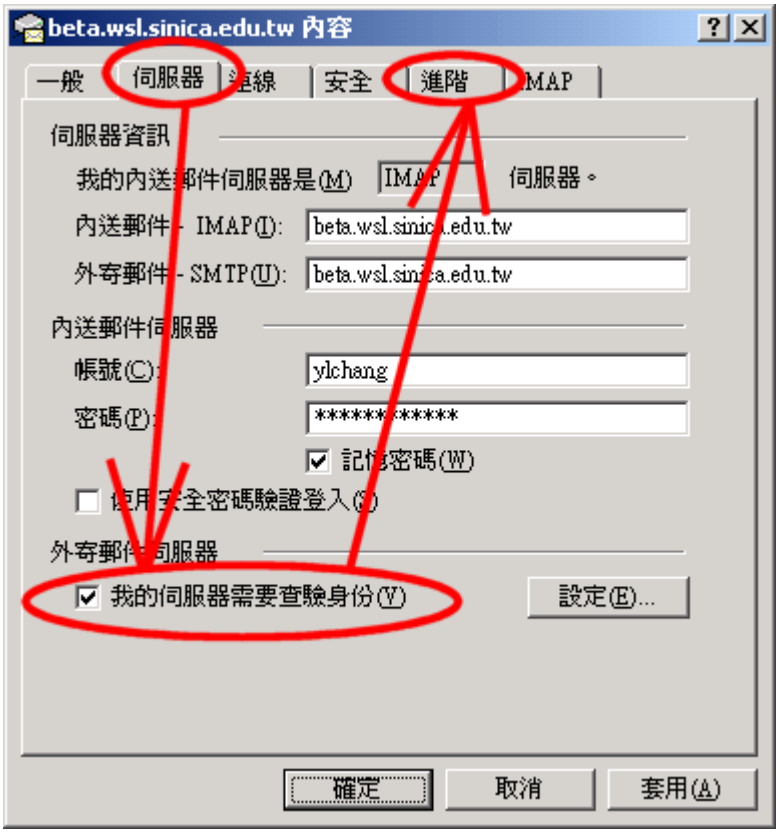
設定使用者的郵件程式使用保密連線與身分認證

本段落以 Microsoft OutLook Express 以及 Netscape Messenger 爲範例，以圖例說如何設定明使用者發送郵件時，受到保密連線以及身分認證機制的保護。

Microsoft OutLook Express 設定步驟

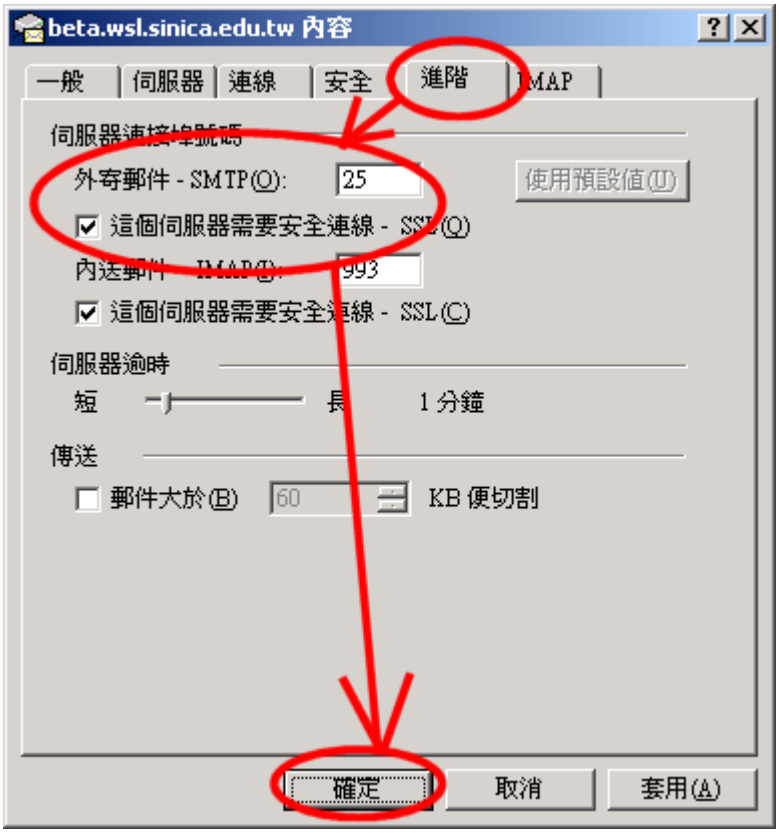
身分認證部分:

[OutLookExpress]-->[工具]-->[帳號]-->[郵件]-->[內容]-->[伺服器]-->[我的伺服器需要查驗身分] (打勾)



保密連線部分:

[OutLookExpress]-->[工具]-->[帳號]-->[郵件]-->[內容]-->[進階]-->[外寄郵件-SMTP]-->[這個伺服器需要安全連線-SSL] （打勾）



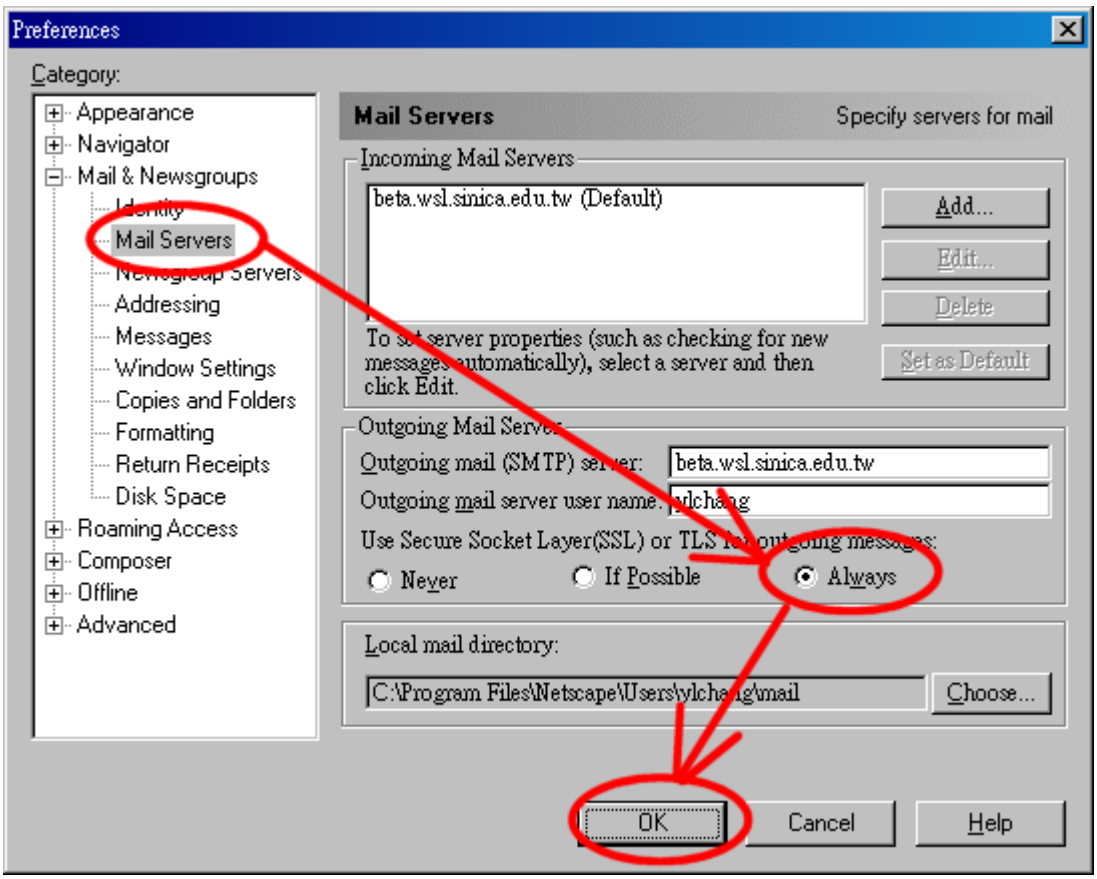
Netscape Messenger 設定步驟

身分認證部分:

Netscape Messenger 會自動偵測 mail server 是否提供身分認證功能，因此不需要針對身分認證做額外的設定。

保密連線部分

[NetscapeMessenger]→[Edit]→[Preference]→[Mail&Newsgroups]→[Mail Servers]→[Outgoing Mail Server]→[Use Secure Socket Layer (SSL) or TLS for outgoing messages]→選 Always



結語

以上，就是在 FreeBSD 4.3-STABLE 版作業系統上，以 sendmail 搭配 OpenSSL 與 Cyrus-SASL 建構具備保密連線與身分認證的郵件系統程序。搭配合適的用戶端軟體可以在使用者傳送郵件到主機的過程中，保護資料的安全。另一方面，如果收件人的主機有支援保密連線功能，則也會自動的以加密的方式將郵件送到收件人的信箱，達到全程保密目標。

礙於篇幅，筆者將於下幾期的中心通訊中，繼續說明如何藉由郵件的標頭 (Email Header)判讀保密連線的等級，以及如何在郵件標頭中加上發信記錄，並與 sendmail 記錄檔交叉比對，實現問題郵件全程追蹤、快速處理的功能。