

第三章 即時傳送架構下Mail的防匿方法

3.1 Mail的運作方式

目前已被應用在TCP/IP網路上的Mail系統如 sendmail、zmailer等，其傳輸方式皆遵循SMTP(Simple Mail Transfer Protocol)[1]，而傳輸文件之格式亦以RFC822(Standard for the format of ARPA Internet text messages)為依歸。下圖即為在SMTP架構下Mail的運作方式：

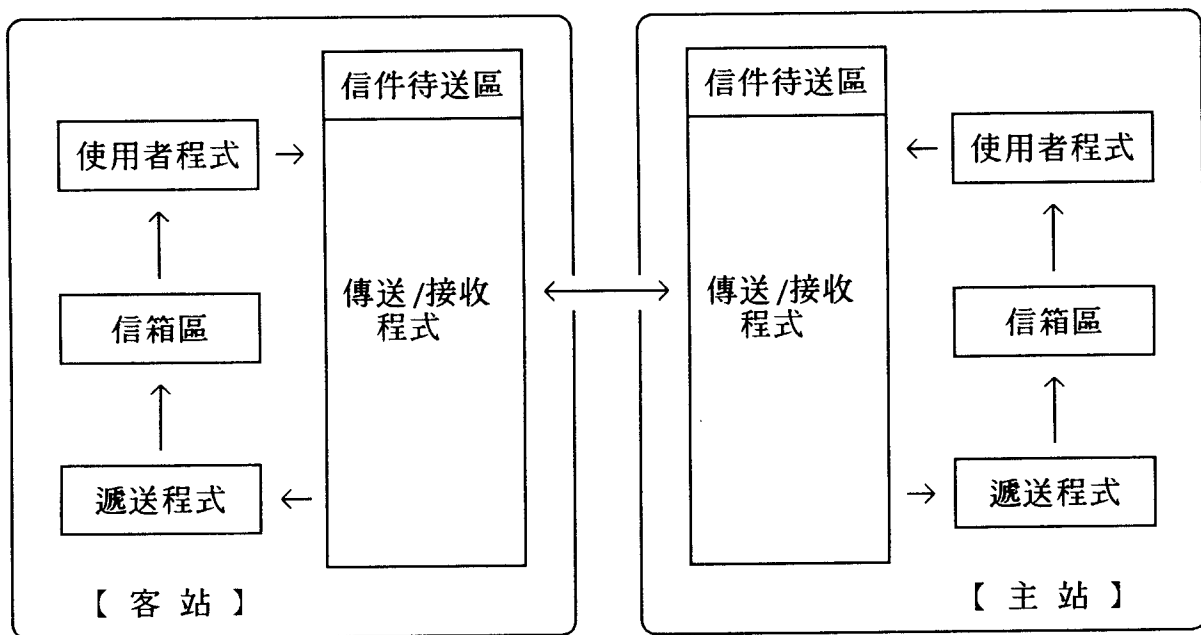


圖3.1 SMTP架構下Mail的運作方式

傳送/接收程式：如 sendmail、smail、zmailer等。

遞送程式：如 mail、rmail、deliver等。

使用者程式：如 mail、mailx、elm等。

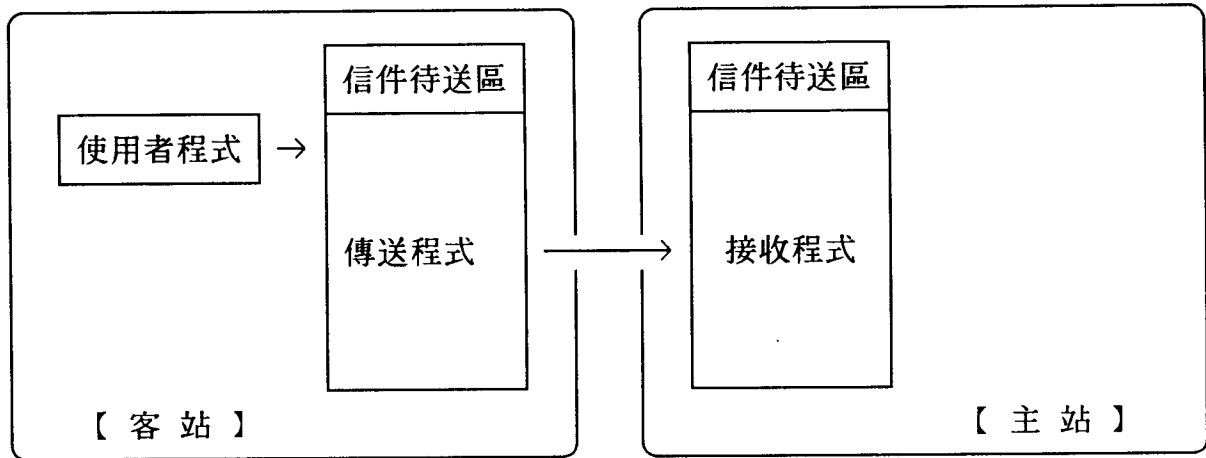
從上圖我們可以很清楚地看出Mail的防匿關鍵：在客站為使用者程式與遞送程式；在主站為接收程式。

3.2 在Mail客站上的防匿方法

一篇使用者所寫的信函從編寫完成到被傳送至Mail主站前，Mail系統

[1]詳見RFC821, Simple Mail Transfer Protocol。

會先作如下的處理：



使用者程式：任意編寫稿頭與文稿。

```
To: ruser@rhost.edu.tw
Subject: test

Here comes the mail body
```

傳送程式：檢查稿頭，確定各欄位如 From 等是否齊全，若有缺則予補齊，並加蓋郵戳 (Received)，再將信函傳送給主站。

```
To: ruser@rhost.edu.tw
Subject: test
Received: by shost.edu.tw
Message-Id: <datanumber@shost>
From: suser@shost.edu.tw
Date: April 15, 1994

Here comes the mail body
```

接收程式：接收客站送來的信函，並加蓋郵戳 (Received)。

```
To: ruser@rhost.edu.tw
Subject: test
Received: by shost.edu.tw
Received: by rhost.edu.tw
      from shost.edu.tw
Message-Id: <datanumber@shost>
From: suser@shost.edu.tw
Date: April 15, 1994

Here comes the mail body
```

圖3.2 一般Mail主、客程式處理Mail的過程

從上圖我們可以看出一個給匿名者有機可乘的漏洞：由於傳送程式完全接受使用者所編寫的文稿 (包括稿頭中的 From 欄位)，所以 From 欄位

的真實性就非常值得懷疑。因此我們似乎必須對傳送程式或使用者程式加以修正，但是做這樣的修正工作並不恰當，原因有三：

- 爲了配合發函人可能會用別名 (alias name) 與隱域名稱 (hidden domain) 來組合通訊地址的習慣，許多使用者程式均提供了修改 From 欄位的功能。
- 使用者程式的種類太多 (如 mail、mailx、elm 等)，不值得我們逐一修改，而且使用者程式處理過的文稿還要再送傳送程式處理。
- From 欄位的重寫牽涉到各傳送站的重寫規則 (rewriting rules)。我們千萬不能只考慮單純的

起點 → 終點

的傳送，應該要顧及在

起點 → 中點 → 終點

的傳送過程中，各站處理 From 欄位的方式是否一致。在第二種情況下，若是 From 欄位在每一站上都被剔除然後再重寫，那麼當信件還未被送達終點前，From 欄位所記載的發函人資料早就失真了。

幸好絕大部份的 Mail 傳送程式都會在傳送信件前，檢查信件的第一行是否爲

From 發函人@發函人地址 [1]

若無，則補；若有，則不補。這項功能乃是因循傳統 UNIX 系統對信件的處理方式 [2]。這項功能在某些傳送程式中是可以選用的 (optional)，但在某些傳送程式中是屬於強制性的。因此，只要我們能讓這項功能表現出來，就不須再修改傳送程式，並達到防匿的效果。

[1] 請注意，是 "From 發函人@發函人地址"，而非 "From: 發函人@發函人地址"。

[2] 在 mail folder 中，"From 發函人@發函人地址" 代表著一封信件的開始。