如何設置“虛擬私有網路(VPN)”

作者: Arpad Magosanyi <mag@bunuel.tii.matav.hu>
譯者: 蔣大偉 <dawei@sinica.edu.tw> v0.2, 7 August 1997 翻譯完成日期: 20 Feb 1999

如何建立虛擬私有網路(Virtual Private Network)。

1 更正

'no controlling tty problem' -> -o 'BatchMode yes' 是由 Zot O'Connor <zot@crl.com> 所更正。

核心 2.0.30 的警告訊息，是由 mag 所更正。

2 推荐廣告

這份文件是 Linux VPN howto，它收集了如何在 Linux (以及一般的 UNIX) 上建立 一個虛擬保護式網路的相關資訊。

2.1 版權聲明

2.2 免責聲明

一如往常：本文對你所造成的任何危害，作者一概不負責任。正確的條文，請參閱 GNU GPL 0.1.1 的相關部分。

2.3 鄭重聲明

我們所面臨的是安全性的問題：如果你沒有形成一個好的安全策略，以及做好相關的配套 措施，你將無法獲得眞正的安全。

2.4 功勞

感謝所有提供工具程式給本文使用的人仕。

感謝 Zot O'Connor <zot@crl.com> 不僅指正“no controlling tty”的問題，而且 還提供了解決方法。

2.5 本文的現況

在閱讀本文前，你應該已具備完整的 IP 管理知識，至少要對“防火牆(firewall)”ppp 和 ssh，等知識，有一些瞭解。如果你要設定一個 VPN 環境，無論如何一定得知道這些 東西。我只是將我的經驗寫下來，以免日後忘記相關的內容。所以，我相信一定會有安全 的漏洞存在。爲了清楚起見，我試著以主機設置爲路由器方式，而不是以防火牆的方式，來 說明整個內容，希望大家輕輕鬆鬆就能夠瞭解本文。

2.6 相關文件

- 檔案 /usr/doc/HOWTO/Firewall-HOWTO 上的 Linux Firewall-HOWTO 文件
- 檔案 /usr/doc/HOWTO/PPP-HOWTO.gz 上的 Linux PPP-HOWTO 文件
- 目錄 /usr/doc/ssh/* 中的 ssh 文件
- Linux “網路管理指引(Network Admins' Guide)”
- 國家標準及技術委員會 (National Institute Standards and Technology，簡寫爲 NIST) 在電腦安全方面的出版品，請參考網址 http://csrc.ncsl.nist.gov/nistpubs/
- “防火牆通信論壇(Firewall list)” (majordomo@greatcircle.com)

3 介紹

由於網路安全問題日益受到重視，所以，防火牆的技術越來越廣泛地被應用在，網際網路 和“公司內部網路(intranet)”上，防火牆能力的優劣，對 VPN 的安全性有著舉足輕重的 影響。這只是我個人的體會。歡迎大家提出自己的看法。

3.1 命名慣例

我將會使用到“主防火牆(master firewall)”和“次防火牆(slave firewall)”這兩個專 有名稱，然而，VPN 的建置與主僕式架構之間沒有任何關聯性。我只是把它們看成，兩端在 建立連線時，它是個主動的參與者或被動的參與者。發起建立連線的主機，會被當作主防火 牆；然而，被動的參與者，就會被當作次防火牆。

4 開始建置

4.1 規劃

在你開始設定系統前，你應該要先瞭解一下網路連接的細節。現在，我假定你有兩個防火牆，各保護一個公司內部網路。所以，現在每個防火牆應該會有兩個網路界面（至少）。拿一 張紙，寫下它們的

IP 位址和網路遮罩。每個 VPN 的防火牆，將會使用到數個 IP 位址區 段。這些 IP 位址區段，應該設定在你公司現有的子網路的範圍以外。我建議使用“私有” IP 位址區段的範圍。如下所示：

- 10.0.0.0 - 10.255.255.255
- 172.16.0.0 - 172.31.255.255
- 192.168.0.0 - 192.168.255.255

爲了說明，此處我舉了一個設定的案例：有兩台 bastion [譯註] 主機，分別被稱爲 fellini 和 polanski。它們各有一個界面連接網際網路 (-out)，一個界面連接公司內部網路 (-in)，以及，一個界面連接 VPN (-vpn)。所有的 IP 位址和網路遮罩，如下：

- fellini-out: 193.6.34.12 255.255.255.0
- fellini-in: 193.6.35.12 255.255.255.0
- fellini-vpn: 192.168.0.1 點對點
- polanski-out: 193.6.36.12 255.255.255.0
- polanski-in: 193.6.37.12 255.255.255.0
- polanski-vpn: 192.168.0.2 點對點

譯註： bastion 是指暴露在公司網路外部的防火牆閘道。

所以我們有個計劃。

4.2 搜集工具

你將會需要

- Linux 防火牆
- 核心
- 非常少的設定
- ipfwadm 程式
- fwtk 程式
- VPN 所使用的工具
- ssh 程式
- pppd 程式
- sudo 程式
- pty-redir 程式

目前使用的版本：

- 核心： 2.0.29。請使用穩定的核心，而且，必須比 2.0.20 還新，因爲 ping'o'death 的錯誤。在撰寫本文時，最後一個穩定的核心是版本 2.0.30，但是它有一些錯誤。如果，你想要使用最新版核心所提供，既快又酷的網路程式碼，你自己可以嘗試看看，版本 2.0.30 對我而言，已經很好用了。
- 基本的作業系統：我比較喜歡 Debian 所發行的版本。你絕對使用不到任何大型的 軟體套件，當然，也包含 sendmail 在內。你也絕對不能像其它的 UNIX 主機一樣，允許 telnetftp和 'r' 命令，等功能的使用。
- ipfwadm 程式： 我使用的是 2.3.0。
- fwtk 程式： 我使用的是 1.3。
- ssh 程式： >= 1.2.20。較舊的版本，下層的協定會有問題。
- pppd 程式： 我測試的是 2.2.0f，但是我無法確定它是否安全，這就是爲什麼我會 將它的 setuid 位元拿掉，並透過 sudo 來執行它的原因。
- sudo 程式： 我所知道的最新版本是 1.5.2。
- pty-redir 程式： 這是我寫。請至 ftp://ftp.vein.hu/ssa/contrib/mag/pty-redir-0.1.tar.gz 取得。現在的版本是 0.1。如果使用上有任何問題，請來信告知。

4.3 編譯與安裝

你現在的工作不是編譯就是安裝所搜集到的工具。 並參閱其（以及 firewall-howto） 詳細的說明文件。現在，我們已經安裝好這些工具了。

4.4 其它子系統的設定

設定防火牆以及其它的項目。你必須在兩台防火牆主機之間，允許 ssh 資料的流通。這 是指，主防火牆會有網路連線到次防火牆的埠 22。在次防火牆上啓動 sshd，來驗証是否 允許你“登入(login)”。這個步驟尚未測試過，請告訴我你的測試結果。

4.5 設定 VPN 的使用者帳户

以你日常使用的工具（例如，vimkdirchownchmod）在次防火牆上建立一個使用者帳 户，你也可以在主防火牆上建立一個使用者帳户，但是，我認爲在開機階段設定連線就可以 了，所以，使用原始的 root 帳户就已足夠。有任何人可以爲我們說明一下，在主防火牆上 使用 root 帳户，會有什麼危險性？

4.6 爲 master 帳户，產生一個 ssh key

你可以使用 ssh-keygen 程式。如果，你要自動設置 VPN，你可以設定一個沒有密碼的 “私人鑰匙(private key)”。

4.7 為 slave 帳户，設置自動的 ssh 登入環境。

在次防火牆中，複製你剛才產生的“公共鑰匙(public key)”到，使用者帳户 slave 中 的 .ssh/authorized_keys 檔案裡，並且，設定檔案的使用權限，如下：

```
drwx------ 2 slave slave 1024 Apr 7 23:49 ./
drwx------ 4 slave slave 1024 Apr 24 14:05 ../
-rwx------ 1 slave slave 328 Apr 7 03:04 authorized_keys
-rw------- 1 slave slave 660 Apr 14 15:23 known_hosts
-rw------- 1 slave slave 512 Apr 21 10:03 random_seed
```

其中，第一行是 ~slave/.ssh，第二行是 ~slave。

4.8 加強 ssh 在 bastion 主機上的安全性。

請按照我在 sshd_conf 上的設定：

```
PermitRootLogin no
IgnoreRhosts yes
StrictModes yes
QuietMode no
FascistLogging yes
KeepAlive yes
RhostsAuthentication no
RhostsRSAAuthentication no
RSAAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
```

密碼認證(PasswordAuthentication)被關閉了，所以，你只有使用授權過的 key，才能夠 完成登入的動作。（當然，你也已經關閉了，telnet 與 'r' 命令）。

4.9 允許 ppp 的執行，和這兩個帳户的路由。

當你的 master 帳户是 root 時（以我的例子而言），你不必做任何事情。至於 slave 帳户，則會在你的 /etc/sudoers 的檔案中出現一行：

```
Cmnd_Alias VPN=/usr/sbin/pppd,/usr/local/vpn/route
slave ALL=NOPASSWD: VPN
```

正如你所看到的，我在次防火牆主機上，使用了一些命令稿(scripts)，來設定 ppp 和路由表。

4.10 撰寫命令稿程式

在主防火牆主機上，我使用了一個成熟的啓始命令稿：

```
#! /bin/sh
# 程式架構  這個檔案是個建立在 /etc/init.d/ 目錄下的命令稿實例。
#               你應該在 /etc/init.d 目錄下使用這個命令稿。
#
#               作者 Miquel van Smoorenburg <miquels@cistron.nl>.
#               Debian GNU/Linux 修訂版作者
#               Ian Murdock <imurdock@gnu.ai.mit.edu>.
#
# 版本:              @(#)skeleton  1.6  11-Nov-1996  miquels@cistron.nl
#

PATH=/usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/bin/X11/:
PPPAPP=/home/slave/ppp
ROUTEAPP=/home/slave/route
PPPD=/usr/sbin/pppd
NAME=VPN
REDIR=/usr/local/bin/pty-redir
SSH=/usr/bin/ssh
MYPPPIP=192.168.0.1
TARGETIP=192.168.0.2
TARGETNET=193.6.37.0
MYNET=193.6.35.0
SLAVEWALL=polanski-out
SLAVEACC=slave

test -f $PPPD || exit 0

set -e

case "$1" in
  start)
        echo setting up vpn
        $REDIR $SSH -o 'Batchmode yes' -t -l $SLAVEACC $SLAVEWALL sudo $PPPAPP >/tmp/device
        TTYNAME=`cat /tmp/device`
echo tty is $TTYNAME
        sleep 10s
        if [ ! -z $TTYNAME ]
        then
        $PPPD $TTYNAME ${MYPPPIP}:${TARGETIP}
        else
                echo FAILED!
                logger "vpn setup failed"
        fi
        sleep 5s
        route add -net $TARGETNET gw $TARGETIP
        $SSH -o 'Batchmode yes' -l $SLAVEACC $SLAVEWALL sudo $ROUTEAPP
    ;;
```

```
  stop)
        ps -ax | grep "ssh -t -l $SLAVEACC " | grep -v grep | awk '{print $1}' | xargs kill
    ;;
  *)
    # echo "Usage: /etc/init.d/$NAME {start|stop|reload}"
    echo "Usage: /etc/init.d/$NAME {start|stop}"
    exit 1
    ;;
esac

exit 0
```

slave 帳户可以使用命令稿來設定路由 (/usr/local/vpn/route)：

```
#!/bin/bash
/sbin/route add -net 193.6.35.0 gw 192.168.0.1
```

而其 .ppprc 的內容，如下：

```
passive
```

5 讓我們檢視執行的結果：

master 會登入到 slave 帳户裡啓動 pppd以及，將所有的資料重導至本機的 pty（ 虛擬終端機）。整個執行流程如下：

- 配置一個新的 pty
- 透過 ssh 登入 slave 帳户
- 在 slave 帳户中執行 pppd
- master 在本機的 pty 執行 pppd
- 並且在用户端設定路由表。

此處我們考慮到了時序的問題（不是太嚴格的要求），這就是爲什麼我們會使用到 'sleep 10s' 這個敘述的原因。

6 著手執行。

6.1 登入

現在，你應該已經測試過 ssh 是否能夠正常地工作。如果，slave 拒絕你登入，請閱讀 記錄檔。也許是檔案使用權限或 sshd，在設定上的問題。

6.2 啓動 ppp

登入到 slave 帳户，並執行：

```
sudo /usr/sbin/pppd passive
```

此時，如果工作正常你應該會看到一些亂碼。假設，沒有出現亂碼，不是 sudo 就是 pppd 有問題。請參考，記錄檔 */etc/ppp/options* 和 *.ppprc*，等檔案，以便找出是那個命令出了問題。問題排除後，將 'passive' 這個字寫到 .ppprc 裡，然 後再試一次。以壓下 enter'~'和 '^Z'等按鍵的方式，清除螢幕上的亂碼，繼續 工作。現在，你應該會看到 master 的"輸入提示符號(prompt)"，然後執行 kill %1。如果你想知道更多有關"逸出字元(escape character)"的説明，請參閱"調整(tuning)" 那一節。

6.3 一次完成兩個動作

當然，你也可以這麼做

```
ssh -l slave polanski sudo /usr/sbin/pppd
```

如果工作正常，它就會當著你的面，傳送一些看似亂碼的資料。

6.4 Pty 的重導功能

這次，我們試著重導上面的動作：

```
/usr/local/bin/pty-redir /usr/bin/ssh -l slave polanski sudo /usr/sbin/pppd
```

好長的句子，不是嗎？你應該使用 ssh 執行檔的完全路徑名稱，爲了安全的理由，pty-redir 程式只允許你使用這種方式。現在，你會透過這個程式取得一個裝置名稱。假設，你取得的 是 */dev/ttyp0* 。你可以使用 ps 命令來檢視目前的狀況。請找尋 'p0' 這個裝 置的相關敘述。

6.5 這個裝置上面，會有些什麼東西？

試著執行

```
/usr/sbin/pppd /dev/ttyp0 local 192.168.0.1:192.168.0.2
```

來建立連線。然後，檢視 ifconfig 命令的輸出結果，看是否已經建立了這個裝置，然後，使用 ping 來檢查你的虛擬網路。

6.6 設定路由

除了設定主防火牆主機的路由，次防火牆主機也要設定。現在，你應該能夠從公司的一個 內部網路上的主機，ping 到其它內部網路上的主機。接著，設定額外的防火牆規則。現 在，你已經擁有了 VPN 的環境，你可以設定公司兩個內部網路之間的連接規則。

7 調整

7.1 設定的調整

正如我所說的，這份文件只是我個人設定 VPN 的備忘錄而已。設定中有部分的內容，我還 未測試過。等到我測試過後，會給它們正確的定位，或有任何人告訴我“它是如何工作的”。有個最重要的事情大家必須銘記在心，ppp 網路連線尚未使用 8-bit。我自己也覺得 ssh 或 pty 的設定，一定還有要加強的地方。在 ssh 的設定中，使用了“顎化符號(tilde)” (˜) 字元做爲逸出字元。它可以停止或減緩兩端之間的通訊，當任何的“新行符號- 顎化符號(newline-tilde)”逸出順序的出現，會使得 ssh 跳到輸入提示符號的模式。ssh 的文件上說： < 在大部分的系統上，若設定不使用逸出字元，則就算是你使用了 tty，也會造成通訊對話的透通化。> 這個功能相對於 ssh 的選項標記是 ’-*e*’，你也可以在設定檔中設定它。

7.2 頻寬與安全誰重要

不論建置任何的虛擬網路，都會浪費掉實際資源。VPN 會吃掉頻寬和計算的資源。你的目標 應該是如何取得雙贏的局面。你可以使用 ’-C’ 開關或 ’CompressionLevel’ 選項，來調整 它。你也以嘗試使用另一種加密法，但是，我並不建議這麼做。也請注意，如果你使用越高 的壓縮等級，你傳送資料的來回時間就越長。歡迎提供任何相關的測試報告。

8 分析易受攻擊的弱點

我試著在此處說明一下，這個特別的設定和 VPNs 一般有那些易受攻擊的弱點。熱誠地歡迎 各位發表任何意見。

- sudo 程式：我承認，我過度地使用了 sudo。我深信目前它仍然比使用 setuid bits 還安全。Linux 上仍然沒有好的存取控制機制，是個不爭的事實。只有等到相容 POSIX.6 標準的核心正式發行了< http://www.xarius.demon.co.uk/software/posix6/>。更 糟糕的是，我居然透過 sudo 來呼叫執行 shell 的命令稿程式。實在糟糕透了。你有任何 建議麼？
- pppd 程式：它也會使用 suid root (譯註) 的執行方式。你可以透過使用者的 .ppprc 來設定它。留心，它可能會有“緩衝區超限運轉(buffer overrun)”的狀況發生。底限是：盡可能地保護你的 slave 帳户的安全性。
- ssh 程式：當心，ssh 在 1.2.20 以前的版本有安全的漏洞。更糟糕的是，我們的 設定是，當我們對 master 帳户的安全性做出了讓步，相對地，也棄守了 slave 帳户的安 全底限，而且，我們使用了兩個透過 sudo 啓動的程式，也大開了攻擊之門。那是因爲，爲 了能夠自動設定 VPN，我們選擇讓 master 使用沒有密碼的“私人鑰匙(secret key)”。
- firewall 程式： bastion 主機上的防火牆，若規則設定的不恰當，就等於是大開公 司內部網路的方便之門。我建議大家使用 IP“僞裝(Masquerading)”的技術（此時，就算是 路由設定不正確，所造成的影響也是微不足道的），以及，在 VPN 的界面上做嚴格的控制。

譯註： suid root 是指任何執行該程式的人，在執行的當時會取得 root 的權限。其中，suid（ 設定使用者識別代碼）是指設定檔案屬性的第 11 個位元，讓執行該檔案的人，成爲檔案的 擁有者。